Genetic algoritm

Zsolt Sziklai

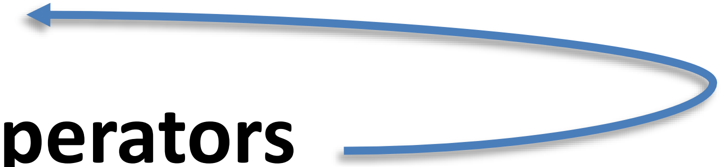
2017.

Biological inspiration

- The basis of the genetic programming gives Darwin's theory of evolution.
- „The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a *generation*. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population." - https://en.wikipedia.org/wiki/Genetic_algorithm [2016-03-05]

LOOP

- **Initialization**
- **Selection**
- **Genetic operators**
  - **Crossover**
  - **Mutation**
- **Termination**



The problem

$$a^2 + b^2 + c^2 + d^2 = 49$$

Parameters of genetic algoritm

- **numberOfPopulation = 10000;**
- **numberOfGenes = 4;**
- **numberOfSelection = 100;**
- **minValue = -100;**
- **maxValue = 100;**
- **memberMutationRate = 35;**
- **geneMutationRate = 35;**
- **targetValue = 49;**
- **maxIteration = 1000;**
- **epsilon = 0.1;**

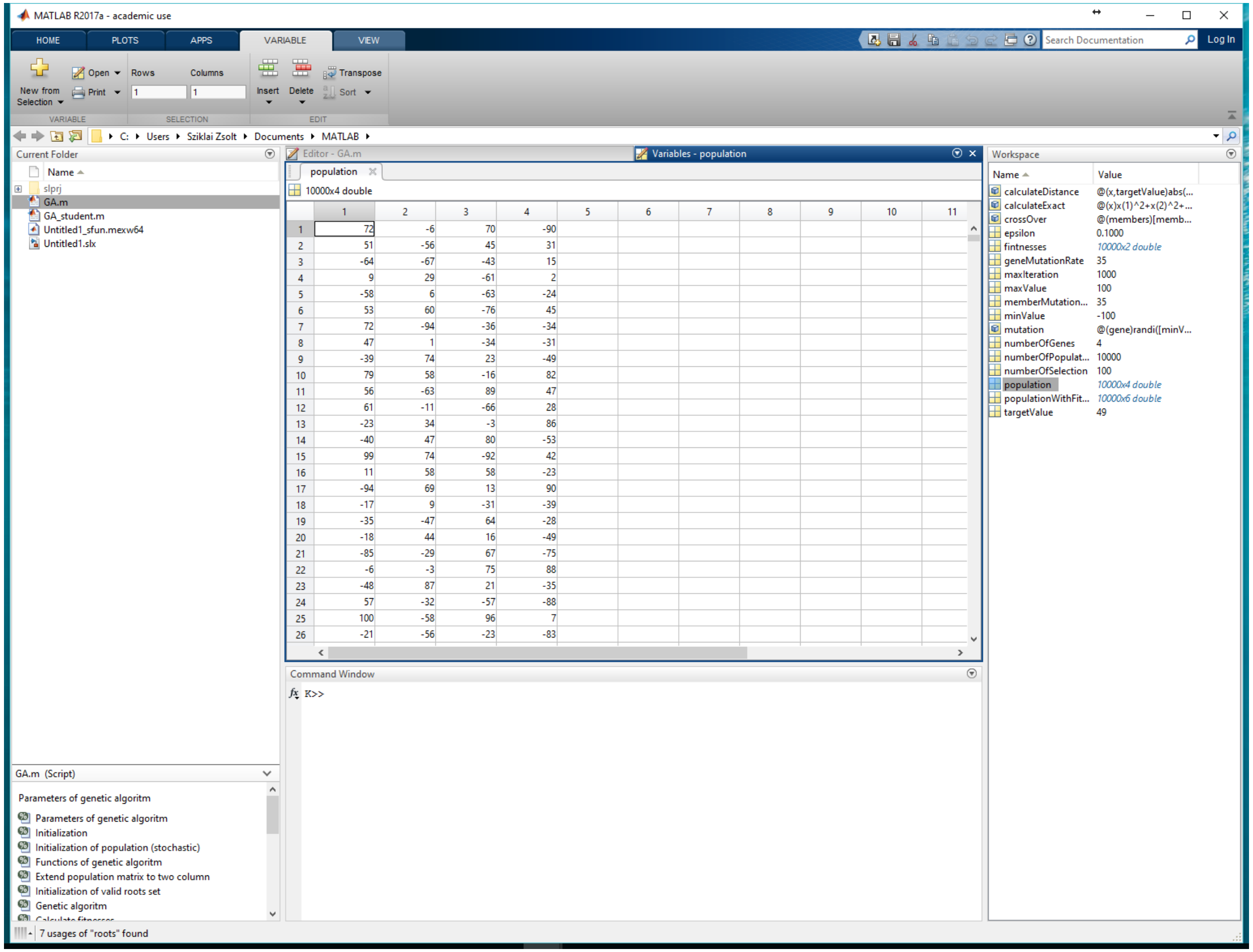
Initialization

- „The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Often, the initial population is generated randomly, allowing the entire range of possible solutions (the *search space*). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.” - https://en.wikipedia.org/wiki/Genetic_algorithm [2016-03-05]

Initialization of population (stochastic)

- For integer roots
  - – population = randi([minValue maxValue], numberOfPopulation, numberOfGenes);
    - randi : generate random integer value between minValue and maxValue
- For rational roots
  - – your business ;)

Initialization of population (stochastic)



10000x4 double

	1	2	3	4	5	6	7	8	9	10	11
1	72	-6	70	-90							
2	51	-56	45	31							
3	-64	-67	-43	15							
4	9	29	-61	2							
5	-58	6	-63	-24							
6	53	60	-76	45							
7	72	-94	-36	-34							
8	47	1	-34	-31							
9	-39	74	23	-49							
10	79	58	-16	82							
11	56	-63	89	47							
12	61	-11	-66	28							
13	-23	34	-3	86							
14	-40	47	80	-53							
15	99	74	-92	42							
16	11	58	58	-23							
17	-94	69	13	90							
18	-17	9	-31	-39							
19	-35	-47	64	-28							
20	-18	44	16	-49							
21	-85	-29	67	-75							
22	-6	-3	75	88							
23	-48	87	21	-35							
24	57	-32	-57	-88							
25	100	-58	96	7							
26	-21	-56	-23	-83							

Name	Value
calculateDistance	@(x,targetValue)abs(...
calculateExact	@(x)x(1)^2+x(2)^2+...
crossOver	@(members)[memb...
epsilon	0.1000
fintnesses	*10000x2 double*
geneMutationRate	35
maxIteration	1000
maxValue	100
memberMutation...	35
minValue	-100
mutation	@(gene)randi([minV...
numberOfGenes	4
numberOfPopulat...	10000
numberOfSelection	100
population	*10000x4 double*
populationWithFit...	*10000x6 double*
targetValue	49

```
K>>
```

GA.m (Script)

Parameters of genetic algoritm

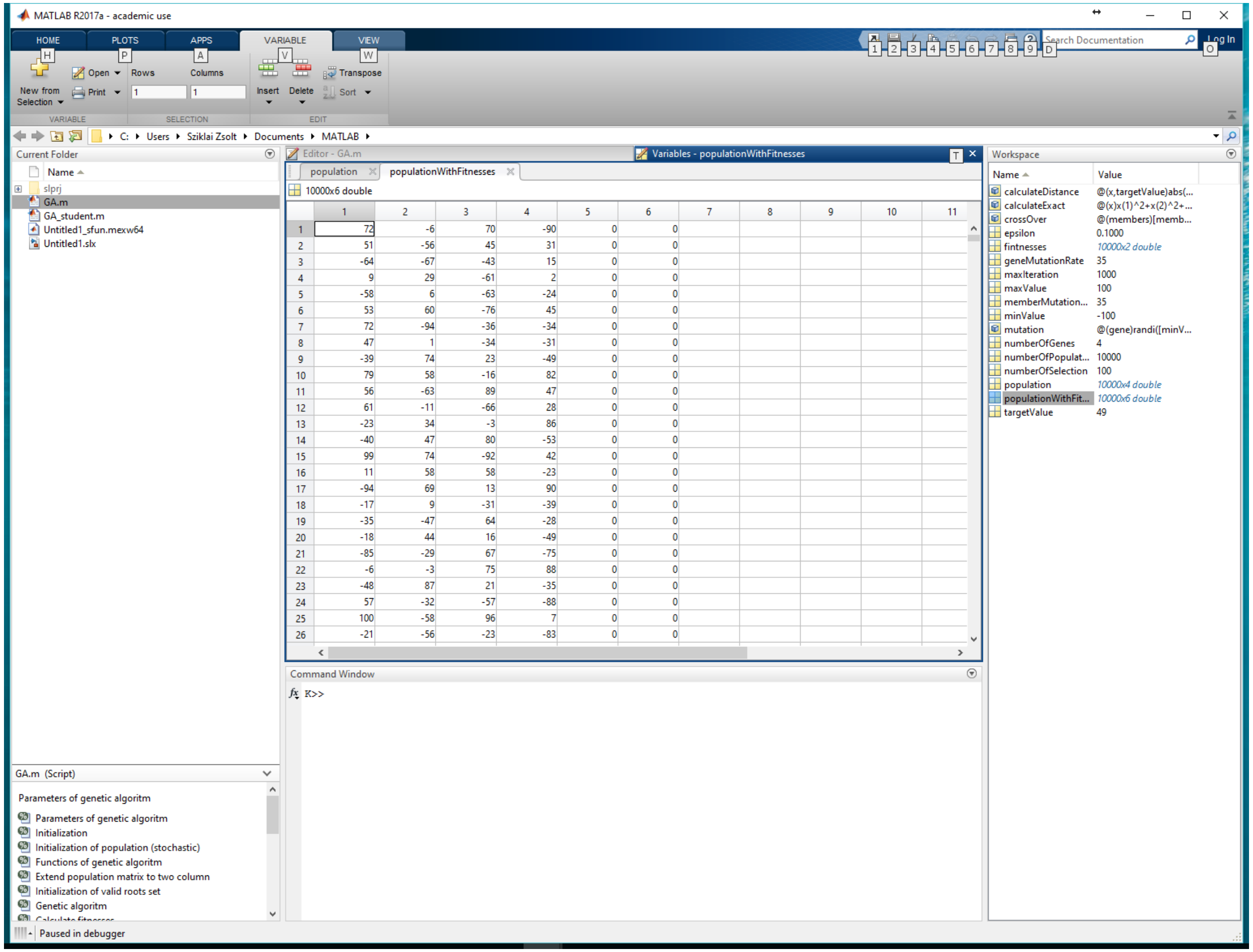
- Parameters of genetic algoritm
- Initialization
- Initialization of population (stochastic)
- Functions of genetic algoritm
- Extend population matrix to two column
- Initialization of valid roots set
- Genetic algoritm
- Calculate fitnesses

7 usages of "roots" found

Column: a, b, c, d genes

Row: member, Rows: population

Initialization of population (stochastic)



10000x6 double

	1	2	3	4	5	6	7	8	9	10	11
1	72	-6	70	-90	0	0					
2	51	-56	45	31	0	0					
3	-64	-67	-43	15	0	0					
4	9	29	-61	2	0	0					
5	-58	6	-63	-24	0	0					
6	53	60	-76	45	0	0					
7	72	-94	-36	-34	0	0					
8	47	1	-34	-31	0	0					
9	-39	74	23	-49	0	0					
10	79	58	-16	82	0	0					
11	56	-63	89	47	0	0					
12	61	-11	-66	28	0	0					
13	-23	34	-3	86	0	0					
14	-40	47	80	-53	0	0					
15	99	74	-92	42	0	0					
16	11	58	58	-23	0	0					
17	-94	69	13	90	0	0					
18	-17	9	-31	-39	0	0					
19	-35	-47	64	-28	0	0					
20	-18	44	16	-49	0	0					
21	-85	-29	67	-75	0	0					
22	-6	-3	75	88	0	0					
23	-48	87	21	-35	0	0					
24	57	-32	-57	-88	0	0					
25	100	-58	96	7	0	0					
26	-21	-56	-23	-83	0	0					

```
K>>
```

Name	Value
calculateDistance	@(x,targetValue)abs(...
calculateExact	@(x)x(1)^2+x(2)^2+...
crossOver	@(members)[memb...
epsilon	0.1000
fintnesses	10000x2 double
geneMutationRate	35
maxIteration	1000
maxValue	100
memberMutation...	35
minValue	-100
mutation	@(gene)randi([minV...
numberOfGenes	4
numberOfPopulat...	10000
numberOfSelection	100
population	10000x4 double
populationWithFit...	10000x6 double
targetValue	49

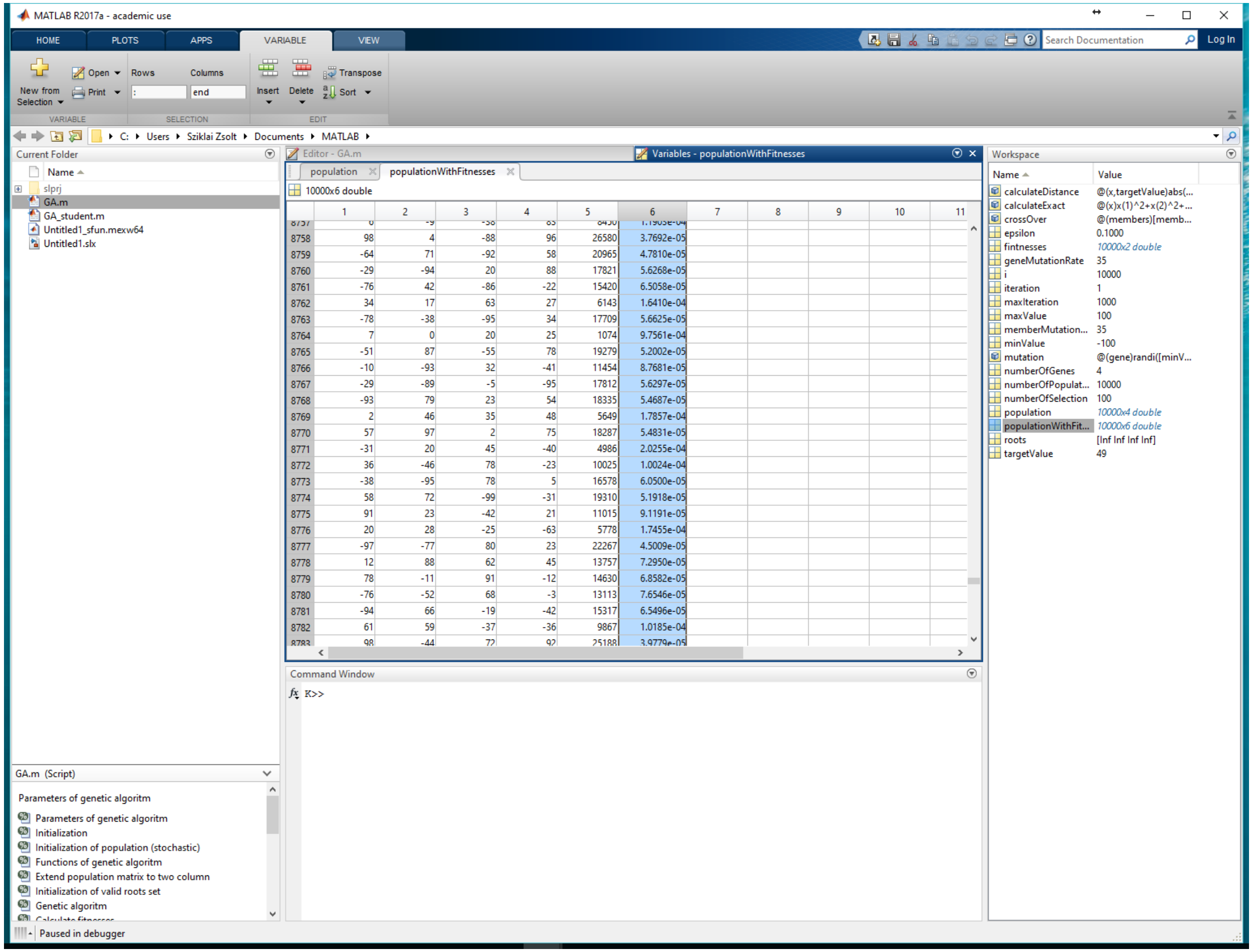
Parameters of genetic algoritm

- Parameters of genetic algoritm
- Initialization
- Initialization of population (stochastic)
- Functions of genetic algoritm
- Extend population matrix to two column
- Initialization of valid roots set
- Genetic algoritm
- Calculate fitnesses

Paused in debugger

Add two columns: column of exact value (y=f(x)) and column of fitness

Population in works



10000x6 double

	1	2	3	4	5	6	7	8	9	10	11
8757	0	-9	-38	83	8430	1.1905e-04					
8758	98	4	-88	96	26580	3.7692e-05					
8759	-64	71	-92	58	20965	4.7810e-05					
8760	-29	-94	20	88	17821	5.6268e-05					
8761	-76	42	-86	-22	15420	6.5058e-05					
8762	34	17	63	27	6143	1.6410e-04					
8763	-78	-38	-95	34	17709	5.6625e-05					
8764	7	0	20	25	1074	9.7561e-04					
8765	-51	87	-55	78	19279	5.2002e-05					
8766	-10	-93	32	-41	11454	8.7681e-05					
8767	-29	-89	-5	-95	17812	5.6297e-05					
8768	-93	79	23	54	18335	5.4687e-05					
8769	2	46	35	48	5649	1.7857e-04					
8770	57	97	2	75	18287	5.4831e-05					
8771	-31	20	45	-40	4986	2.0255e-04					
8772	36	-46	78	-23	10025	1.0024e-04					
8773	-38	-95	78	5	16578	6.0500e-05					
8774	58	72	-99	-31	19310	5.1918e-05					
8775	91	23	-42	21	11015	9.1191e-05					
8776	20	28	-25	-63	5778	1.7455e-04					
8777	-97	-77	80	23	22267	4.5009e-05					
8778	12	88	62	45	13757	7.2950e-05					
8779	78	-11	91	-12	14630	6.8582e-05					
8780	-76	-52	68	-3	13113	7.6546e-05					
8781	-94	66	-19	-42	15317	6.5496e-05					
8782	61	59	-37	-36	9867	1.0185e-04					
8783	98	-44	72	92	25188	3.9779e-05					

```
K>>
```

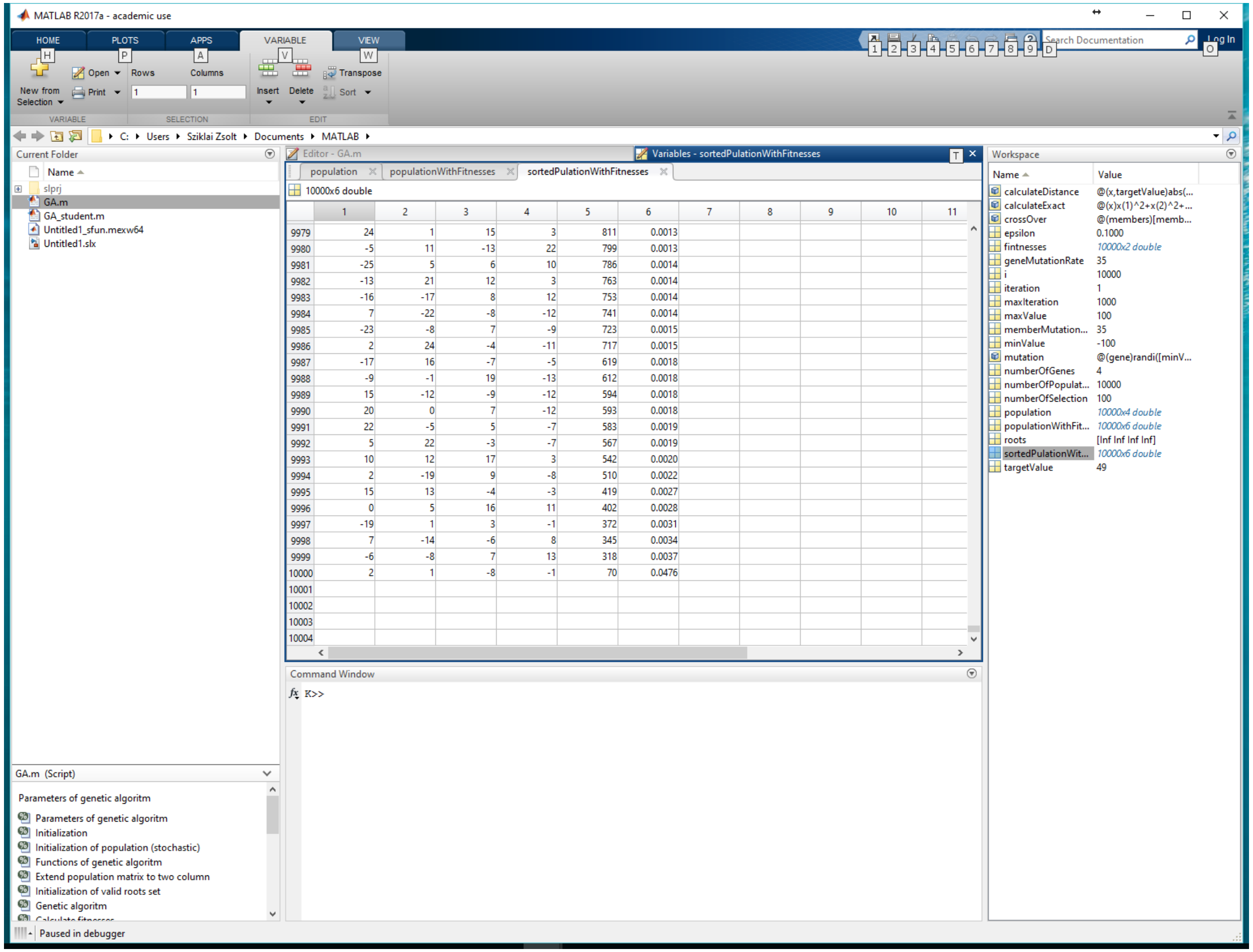
Name	Value
calculateDistance	@(x,targetValue)abs(...
calculateExact	@(x)x(1)^2+x(2)^2+...
crossOver	@(members)[memb...
epsilon	0.1000
fintnesses	10000x2 double
geneMutationRate	35
i	10000
iteration	1
maxIteration	1000
maxValue	100
memberMutation...	35
minValue	-100
mutation	@(gene)randi([minV...
numberOfGenes	4
numberOfPopulat...	10000
numberOfSelection	100
population	10000x4 double
populationWithFit...	10000x6 double
roots	[Inf Inf Inf Inf]
targetValue	49

Parameters of genetic algoritm

- Parameters of genetic algoritm
- Initialization
- Initialization of population (stochastic)
- Functions of genetic algoritm
- Extend population matrix to two column
- Initialization of valid roots set
- Genetic algoritm
- Calculate fitnesses

column of exact value (distance of target value) and column of fitness (big is better (1/x))

Sorted population



10000x6 double

	1	2	3	4	5	6	7	8	9	10	11
9979	24	1	15	3	811	0.0013					
9980	-5	11	-13	22	799	0.0013					
9981	-25	5	6	10	786	0.0014					
9982	-13	21	12	3	763	0.0014					
9983	-16	-17	8	12	753	0.0014					
9984	7	-22	-8	-12	741	0.0014					
9985	-23	-8	7	-9	723	0.0015					
9986	2	24	-4	-11	717	0.0015					
9987	-17	16	-7	-5	619	0.0018					
9988	-9	-1	19	-13	612	0.0018					
9989	15	-12	-9	-12	594	0.0018					
9990	20	0	7	-12	593	0.0018					
9991	22	-5	5	-7	583	0.0019					
9992	5	22	-3	-7	567	0.0019					
9993	10	12	17	3	542	0.0020					
9994	2	-19	9	-8	510	0.0022					
9995	15	13	-4	-3	419	0.0027					
9996	0	5	16	11	402	0.0028					
9997	-19	1	3	-1	372	0.0031					
9998	7	-14	-6	8	345	0.0034					
9999	-6	-8	7	13	318	0.0037					
10000	2	1	-8	-1	70	0.0476					
10001											
10002											
10003											
10004											

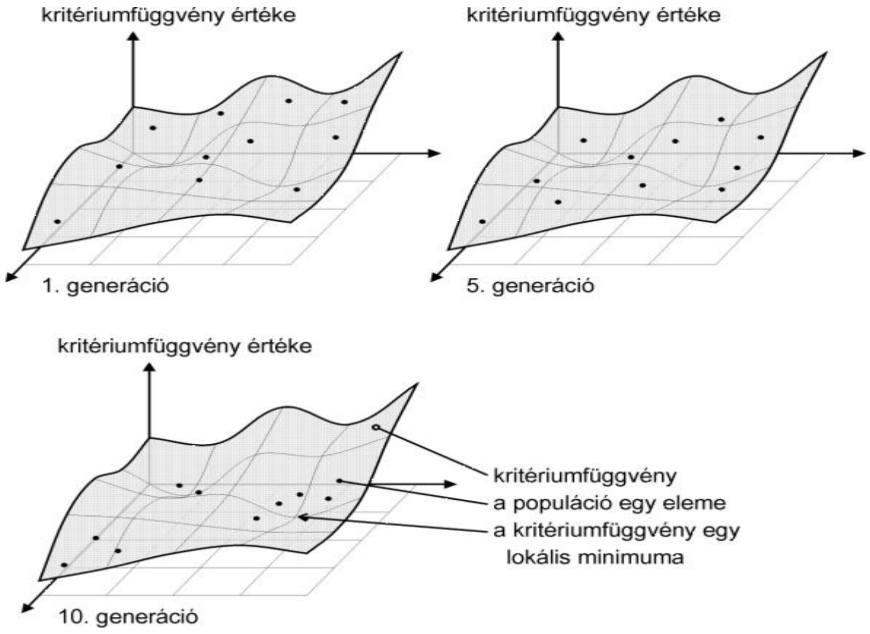
Name	Value
calculateDistance	@(x,targetValue)abs(...
calculateExact	@(x)x(1)^2+x(2)^2+...
crossOver	@(members)[memb...
epsilon	0.1000
fintnesses	10000x2 double
geneMutationRate	35
i	10000
iteration	1
maxIteration	1000
maxValue	100
memberMutation...	35
minValue	-100
mutation	@(gene)randi([minV...
numberOfGenes	4
numberOfPopulat...	10000
numberOfSelection	100
population	10000x4 double
populationWithFit...	10000x6 double
roots	[Inf Inf Inf Inf]
sortedPulationWit...	10000x6 double
targetValue	49

```
K>>
```

Parameters of genetic algoritm

- Parameters of genetic algoritm
- Initialization
- Initialization of population (stochastic)
- Functions of genetic algoritm
- Extend population matrix to two column
- Initialization of valid roots set
- Genetic algoritm

Criterion function



Criteria function

- calculateExact = @(x) x(1)^2+x(2)^2+x(3)^2+x(4)^2;
  - Matlab anonymous function : ‚x’ is input param and the return value is a mathematical expression’s result

Calculate fitness

- Close to target value are big differences between a small step (selection pressure)
  - populationWithFitnesses(i, 6) = 1 / calculateDistance(populationWithFitnesses(i, 1:4), targetValue);
  - See 1/x function

Sorting

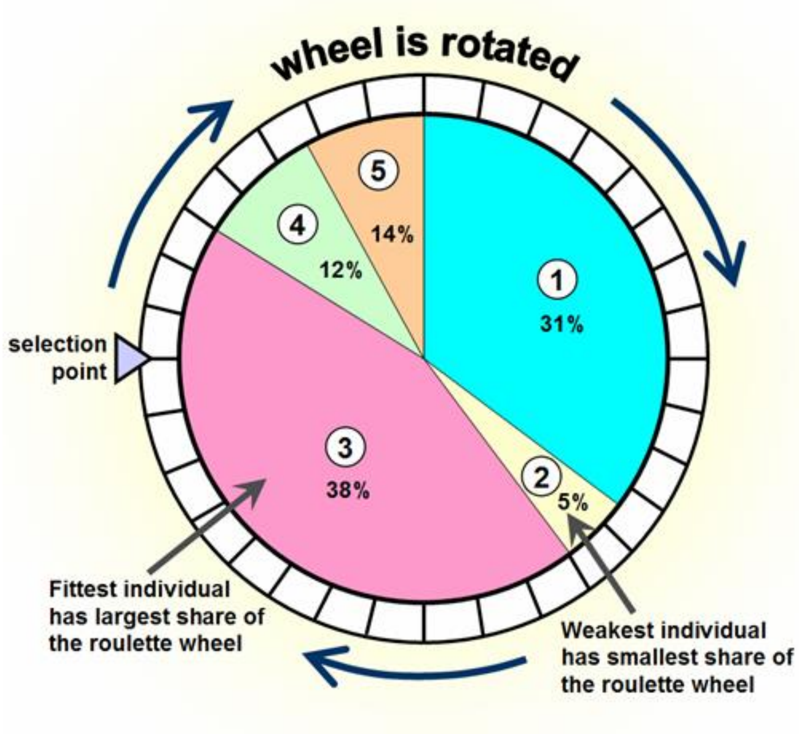
- sortedPulationWithFitnesses = sortrows(populationWithFitnesses, 6);
  - Sorting by row

SELECTION

- „During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a *fitness-based* process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as the former process may be very time-consuming.” - https://en.wikipedia.org/wiki/Genetic_algorithm [2016-03-05]

Selection

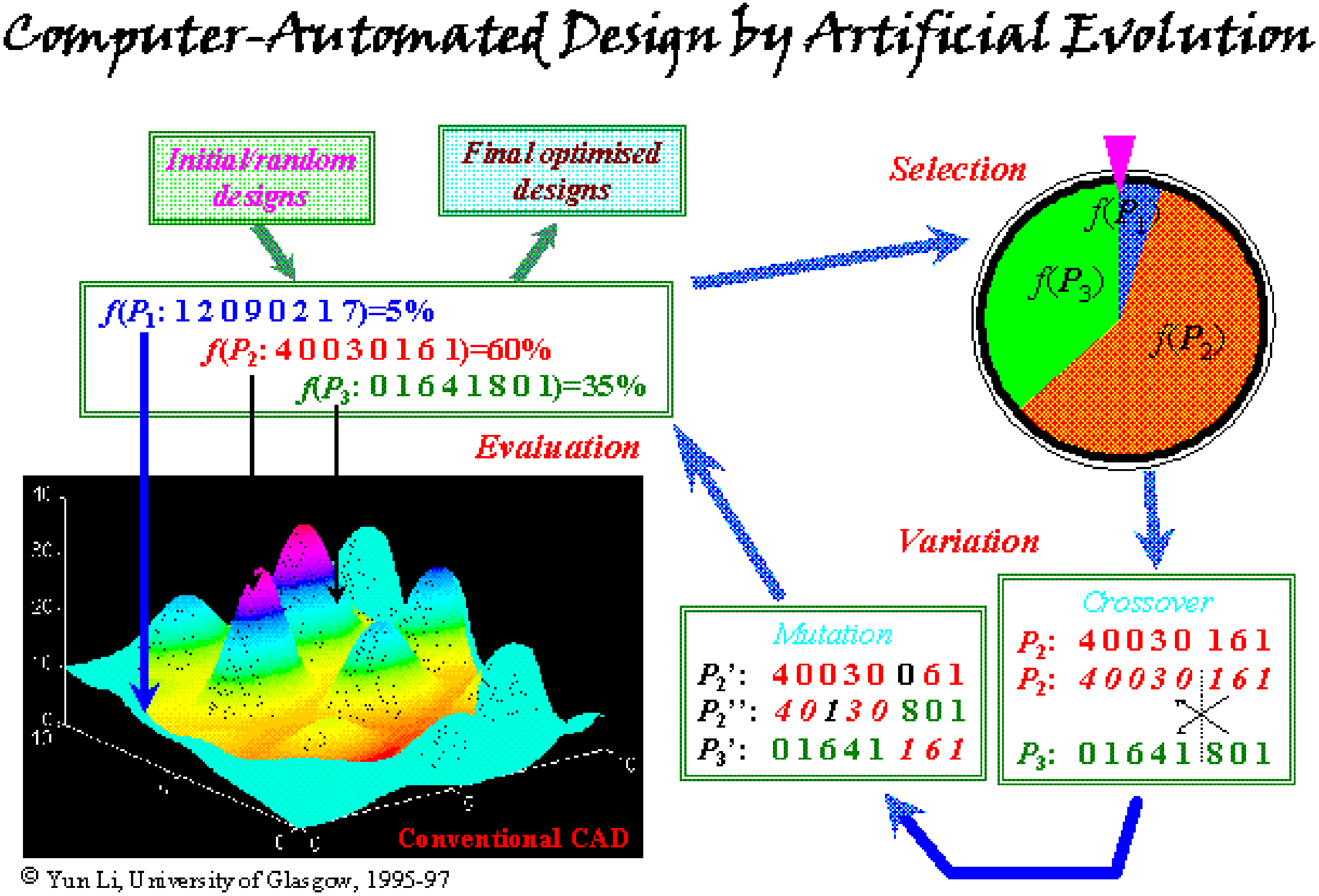
- Elitist
  - selectedPopulation = sortedPulationWithFitnesses(end:-1:numberOfSelection, 1:4);
    - Selecting the bests
- Roulette wheel
  - Your bussines ;)


wheel is rotated
selection point
5
14%
4
12%
1
31%
3
38%
2
5%
Fittest individual has largest share of the roulette wheel
Weakest individual has smallest share of the roulette wheel

Selection – FITNESS Function (*roulette-wheel selection*)

- „A generic selection procedure may be implemented as follows:
- The fitness function is evaluated for each individual, providing fitness values, which are then normalized. Normalization means dividing the fitness value of each individual by the sum of all fitness values, so that the sum of all resulting fitness values equals 1.
- The population is sorted by descending fitness values.
- Accumulated normalized fitness values are computed (the accumulated fitness value of an individual is the sum of its own fitness value plus the fitness values of all the previous individuals). The accumulated fitness of the last individual should be 1 (otherwise something went wrong in the normalization step).
- A random number *R* between 0 and 1 is chosen.
- The selected individual is the first one whose accumulated normalized value is greater than *R*.
- For a large number of individuals the above algorithm might be computationally quite demanding. A simpler and faster alternative uses the so-called stochastic acceptance." - https://en.wikipedia.org/wiki/Genetic_algorithm [2016-03-05]

Selection – Roulette-wheel



Selection by Rank

- Like roulette-wheel selection but…
- For example: 60% , 35% , 5%
  - Sorting by (reverse): 1st 5% , 2nd 35%, 3th 60%
  - Modifying by: (1 + 2 + 3 = 6)
    - 1/6 = **17%** , 2/6 = **33%** , 3/6 = **50% => 100%**

Selection - Competition

- Choose randomly two invidual
- Generate a random number between 0 and 1
- If [0, 0,5] choose first invidual Else choose second

Selection – Best (elitist)

- Choose two of the best for crossover

Selection - Randomly

- Choose randomly of population

Selection - interactive

- „**Interactive evolutionary computation** (IEC) or **aesthetic selection** is a general term for methods of evolutionary computation that use human evaluation. Usually human evaluation is necessary when the form of fitness function is not known (for example, visual appeal or attractiveness; as in Dawkins, 1986[1]) or the result of optimization should fit a particular user preference (for example, taste of coffee or color set of the user interface).” -

Selection - boltzman

$$f^*(i) = \frac{\exp(f(i)/T)}{< \exp(f(i)/T) >}$$

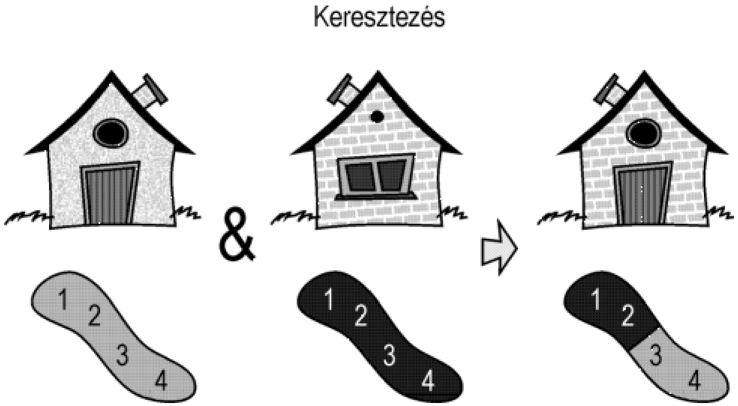
T is temperature and < > denotes the average over the population, as T decreases

Selection - Sigma scaling

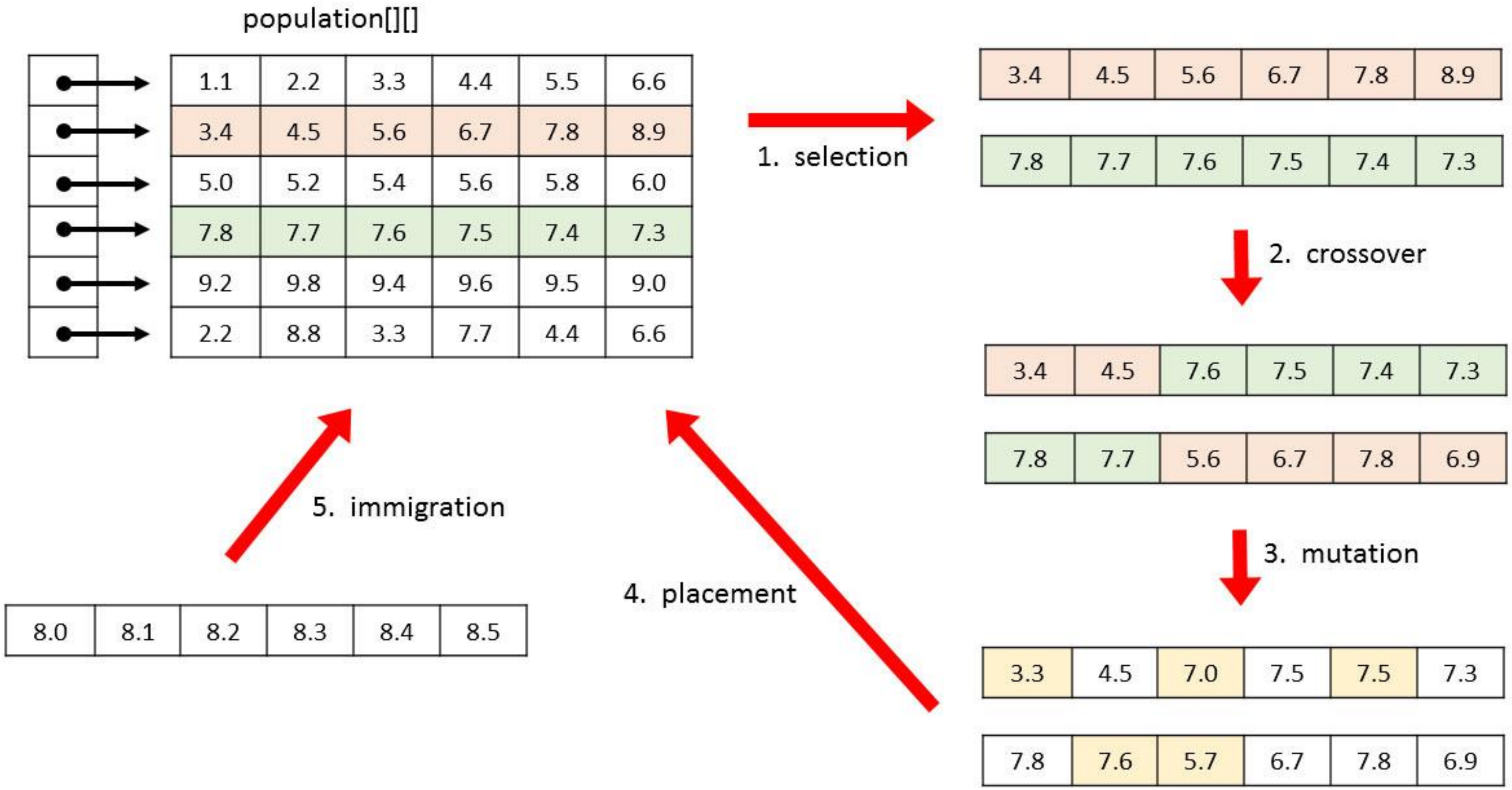
$$f^*(i) = \begin{cases} 1 + \dfrac{f(i) - F}{2s} & ha \;\; s \neq 0 \\ 1 & ha \;\; s = 0 \end{cases}$$

f(i) is the fitness of i, **F** is teh mean fitness of the population, **s** is the standard deviation of the population

Crossover



Crossover



population[][]

1.1	2.2	3.3	4.4	5.5	6.6
3.4	4.5	5.6	6.7	7.8	8.9
5.0	5.2	5.4	5.6	5.8	6.0
7.8	7.7	7.6	7.5	7.4	7.3
9.2	9.8	9.4	9.6	9.5	9.0
2.2	8.8	3.3	7.7	4.4	6.6

1. selection

3.4	4.5	5.6	6.7	7.8	8.9
7.8	7.7	7.6	7.5	7.4	7.3

2. crossover

3.4	4.5	7.6	7.5	7.4	7.3
7.8	7.7	5.6	6.7	7.8	6.9

3. mutation

3.3	4.5	7.0	7.5	7.5	7.3
7.8	7.6	5.7	6.7	7.8	6.9

4. placement

5. immigration

8.0	8.1	8.2	8.3	8.4	8.5

Crossover

- simple crossover
  - crossOver = @(member1, member2) [member1(1) member1(2) member2(3) member2(4)];
    - 1st and 2nd genes inherit to father and other to mother
- randomize crossover
  - crossOver = @(members) [members{randi([1 2])}(1) members{randi([1 2])}(2) members{randi([1 2])}(3) members{randi([1 2])}(4)];
    - use matlab array for rating 50% to choose father’s or mother’s gene(s)

Crossover (implement)

- Select two members selectedPopulation and use crossOver function. The result member put the new population matrix.
  - for i=1:numberOfPopulation

Mutation

- For integer roots
  - mutation = @(gene) randi([minValue maxValue]);
- For rational roots
  - Your business ;)

Mutation (implement)

- Use memberMutationRate and write the correct condition.
- If member has to mutate use geneMutationRate and write the correct conditions all of genes.
  - for i=1:numberOfPopulation
    - If depends memberMutationRate
      - If depends geneMutationRate
      - If depends geneMutationRate
      - If depends geneMutationRate
      - If depends geneMutationRate

Exit condition

- Integer and rational target value
  - Integer case
    - If sortedPulationWithFitnesses(end,5) == targetValue
  - General case
    - if abs(sortedPulationWithFitnesses(end,5) - targetValue) < epsilon
      - Epsilon has to be less than 1

Write out the result

- Last one is the best
  - disp(sortedPulationWithFitnesses(end,:));

Appendix

- rng('shuffle');
  - Help ;)
    - Random generator initialization. Every run generates different random values.
- get(gcf,'currentchar')
  - Help ;)
    - asynchronously keyboard handling
      - get(gcf,'currentchar') ~= ' '
      - Gives the pressed button, it doesn’t block your code running.