Outline

1. **More Operators for Binary Coded GA**
   - Selection Operators
   - Crossover Operators
   - Mutation Operators
2. **Simulation and Application**
   - Rosenbrock Function
   - Himmelblau Function
   - Rastrigin Function
   - Ackley Function
3. **Closure**

Recap

- Binary-coded genetic algorithm
  - Generalized framework
  - Solution representation
  - Working principles through an example
    - selection operator,
    - crossover and mutation operators,
    - survivor operator
  - Graphical example

Selection Operators

- The purpose is to identify good (usually above-average) solutions in the population.
- In this process, we eliminate bad solutions in the population.
- We make multiple copies of good solutions.
- Selection can be used either before or after search/variation operators.
  - When selection is used before search/variation operators, it is called as <u>reproduction</u> or <u>selection</u>.
  - After search/variation operators: The process of choosing the next generation population from parent and off-spring populations is referred to as <u>survivor</u> or <u>elimination</u>. It is also being referred to as <u>environmental selection</u>.
- Common selection methods are
  - Fitness proportionate selection
  - Tournament selection
  - Ranking selection

Fitness Proportionate Selection Operator

- **Probability** of selecting individual i from population P of size N is:

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}, \tag{1}$$

where f_i is the fitness of individual i.

- It is suitable for maximization problem.
- It is also known as roulette wheel selection method.

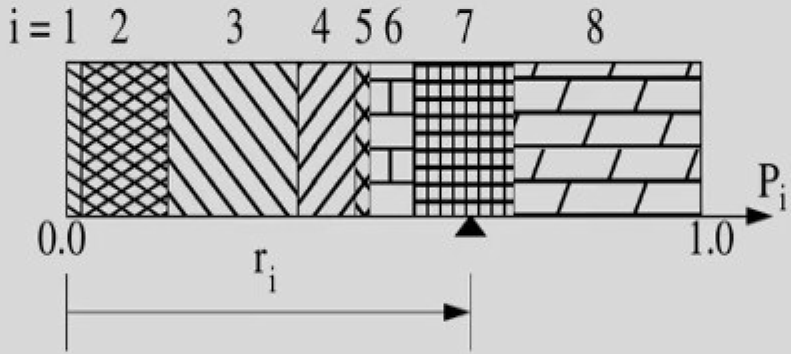
Fitness Proportionate Selection Operator

Index	$f(x_1, x_2)$	p_i	P_i
1	210.445	0.0041	0.0041
2	7536.407	0.1471	0.1512
3	14041.882	0.2740	0.4252
4	1546.603	0.0302	0.4554
5	189.732	0.0037	0.4591
6	671.924	0.0131	0.4122
7	7252.209	0.1415	0.6137
8	19793.183	0.3863	1.0000
sum	51242.386		

- Let us calculate probability of solution '1'.
  - Sum of all fitness values $= \sum_{j=1}^{N} f_j = 51242.386$
  - Probability, $p_1 = \frac{210.445}{51242.386} = 0.0041$
- Similarly, we can calculate probability of solution '2'.
  - Probability, $p_2 = \frac{7536.407}{51242.386} = 0.1471$
- We then calculate probability of other solutions.
- P_i is the cumulative probability.

Fitness Proportionate Selection Operator

- The figure is drawn using the cumulative probability.
- r_i is the random number between 0 to 1.



Index	$f(x_1, x_2)$	p_i	P_i
1	210.445	0.0041	0.0041
2	7536.407	0.1471	0.1512
3	14041.882	0.2740	0.4252
4	1546.603	0.0302	0.4554
5	189.732	0.0037	0.4591
6	671.924	0.0131	0.4722
7	7252.209	0.1415	0.6137
8	19793.183	0.3863	1.0000

- Let the first random number is $r_1 = 0.07218$. As per the cumulative probability, solution 2 is selected. 1
- The second number random number is $r_2 = 0.68799$. Solution 8 is selected. + 2

Fitness Proportionate Selection Operator

Index	$f(x_1, x_2)$	p_i	P_i
1	210.445	0.0041	0.0041
2	7536.407	0.1471	0.1512
3	14041.882	0.2740	0.4252
4	1546.603	0.0302	0.4554
5	189.732	0.0037	0.4591
6	671.924	0.0131	0.4722
7	7252.209	0.1415	0.6137
8	19793.183	0.3863	1.0

- The rest of the random numbers are
  - $r_3 = 0.49976$, $r_4 = 0.31172$, $r_5 = 0.51961$, $r_6 = 0.48610$, $r_7 = 0.87648$, $r_8 = 0.99177$.

  + N

- Selected solutions are '7', '3', '7', 7', '8', '8'.

Fitness Proportionate Selection Operator

Index	$f(x_1, x_2)$	p_i	P_i
1	210.445	0.0041	0.0041
2	7536.407	0.1471	0.1512
3	14041.882	0.2740	0.4252
4	1546.603	0.0302	0.4554
5	189.732	0.0037	0.4591
6	671.924	0.0131	0.4722
7	7252.209	0.1415	0.6137
8	19793.183	0.3863	1.0

- The rest of the random numbers are
  - $r_3 = 0.49976$, $r_4 = 0.31172$, $r_5 = 0.51961$, $r_6 = 0.48610$, $r_7 = 0.87648$, $r_8 = 0.99177$.
- Selected solutions are '7', '3', '7', 7', '8', '8'.

- Solution 2 gets 1 copy, solution 3 gets 1 copy, solution 7 gets 3 copies and solution 8 gets 3 copies.

Index	$f(x_1, x_2)$
2	7536.407
3	14041.882
7	7252.209
7	7252.209
7	7252.209
8	19793.183
8	19793.183
8	19793.183

- Two solutions become 'super-solution' in the population.

Stochastic Remainder Roulette-Wheel Selection Operator

- We calculate probability of each solution using the formula used with fitness proportionate selection operator.
- We then multiply the population size (N) with the probability of each solution.

Index	$f(x_1, x_2)$	p_i	Np_i
1	210.445	0.0041	0.0329
2	7536.407	0.1471	1.1766
3	14041.882	0.2740	2.1922
4	1546.603	0.0302	0.2415
5	189.732	0.0037	0.0296
6	671.924	0.0131	0.1049
7	7252.209	0.1415	1.1322
8	19793.183	0.3863	3.0901

- We assign a number of copies to the solution based on the integer value of product Np_i.
- For example, $Np_1 = 0.0329$ for solution '1', we do not assign any copy to it.
- For solution '2', $Np_2 = 1.17$. Therefore, we assign 1 copy of it.
- Similarly, solution '3' gets 2 copies, solution '7' gets 1 copy and solution '8' gets 3 copies.
- The total number of solutions selected is 7.
- We have to choose 1 more solution to keep the population size fixed, that is, $N = 8$.

Stochastic Remainder Roulette-Wheel Selection Operator

- We then use the fitness proportionate selection operator using only the decimal values.

Index	Np_i	Decimals	P_i
1	0.0329	0.0329	0.00329
2	1.1766	0.1766	0.2094
3	2.1922	0.1922	0.4017
4	0.2415	0.2415	0.6431
5	0.0296	0.0296	0.6728
6	0.1049	0.1049	0.7777
7	1.1322	0.1322	0.9099
8	3.0901	0.0901	1.0000
	sum	1	

- Let the random number is $r_i = 0.76335$. We now select solution 6.
- We can see that the first part of stochastic remainder roulette-wheel selection operator is deterministic because the number of copies is decided using the integer value.
- It is considered less noisy than fitness proportionate selection operator in the sense of introducing less variance in its realization.

Stochastic Universal Sampling (SUS)

Index	$f(x_1, x_2)$	p_i	P_i
1	210.445	0.0041	0.0041
2	7536.407	0.1471	0.1512
3	14041.882	0.2740	0.4252
4	1546.603	0.0302	0.4554
5	189.732	0.0037	0.4591
6	671.924	0.0131	0.4722
7	7252.209	0.1415	0.6137
8	19793.183	0.3863	1.0000

- Only one random number r is required for selecting solutions.
- Since N different solutions have to be chosen, a set of N equi-spaced number is created.

- $R = \{r, r+1/N, r+2/N, \ldots, r+(N-1)/N\}$ mod 1.
- Let $r = 0.40416$ and $1/N = 0.125$. Here, the population size is $N = 8$.
- Solution '3' is selected first using r.
- The second solution is selected using $r + 1/N = 0.40416 + 0.125 = 0.52916$, that is, solution '7'.
- The series of numbers for selecting solutions is $0.65416, 0.77916, 0.90416, \boxed{0.02916}, 0.15416, 0.27916$.
- The following solutions are then selected: '8' '8', '8', '2', '3', '3'.

Fitness Proportionate Selection

Limitations

- Domination of "super solution" in early generations
  - Suppose, the fitness (maximization) of five solutions in the population is given as $f_1 = 10, f_2 = 5, f_3 = 70, f_4 = 7, f_5 = 8$.
  - Because of the fitness proportional selection, solution '3' will get more copies and will become a "super-solution".
- Slower convergence in later generations
  - Suppose, the fitness of five solutions in the population is given as $f_1 = 19, f_2 = 21, f_3 = 22, f_4 = 18, f_5 = 20$.
  - Every solution will get a copy. It means that selection operator has no role in selecting solutions and thus, it becomes dummy.

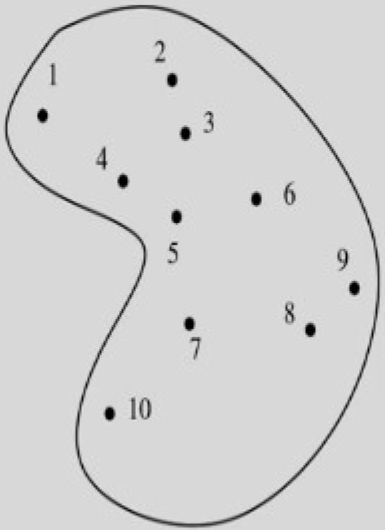
Possible remedies:

- Fitness scaling
- Use ranking selection to avoid fitness scaling
- Tournament selection: It does not have any scaling problem.

Tournament Selection Operator

Tournament selection with tournament size 'k'

Randomly, we sample a subset $\hat{P}$ of k solutions from the population P. Select solution in $\hat{P}$ with the best fitness.



- Suppose $k = 4$, we choose four solutions randomly, say, $\{2, 6, 3, 9\}$ and their fitness values are $f_2 = 10, f_6 = 5.9, f_3 = 16.7, f_9 = 9.4$.
- For maximization problem, who will win the tournament?
  Solution 3
- For minimization problem, who will win the tournament?
  Solution 6
- Often, tournament size $k = 2$ is used, which is known as binary tournament selection operator.

Operators for Survivor or Elimination

$(\mu + \lambda)$ selection scheme

- Let the size of parent population is μ.
- Let λ number of offspring solutions are generated after variation operator.
- The next population is filled by choosing the μ best number of solutions from the combined populations of parent and offspring solutions.

(μ, λ) selection scheme where $(\lambda > \mu)$

- Let the size of parent population is μ.
- Let λ number of offspring solutions are generated after variation operator. Note that $(\lambda > \mu)$.
- The next population is filled by choosing the μ best number of solutions from the offspring population only.

Crossover Operators for Binary Variables

- Crossover operator creates offspring/new solutions from **more than one parent**.
- Crossover probability (p_c)
  - $100p_c\%$ strings are used in the crossover operator.
  - $100(1-p_c)\%$ of the population are simply copied to the new population.

- Single-point crossover operator
  - Choose a random site on the two parents
  - Split parents at this crossover site
  - Create children by exchanging tails

parent 1:	1 0 1 0 1 1	1 0 1 0
parent 2:	0 1 0 1 0 0	1 1 1 0
offspring 1:	1 0 1 0 1 1	1 1 1 0
offspring 2:	0 1 0 1 0 0	1 0 1 0

Crossover Operators for Binary Variables

- $n-$point crossover:
  - Choose n random crossover sites
  - Split along those sites
  - Glue parts, alternating between parents
- $2-$ point crossover operator

parent 1:	1 0 1|0 1 1|1 0 1 0
parent 2:	0 1 0|1 0 0|1 1 1 0
offspring 1:	1 0 1|1 0 0|1 0 1 0
offspring 2:	0 1 0|0 1 1|1 1 1 0

x_1 01101

x_2 11100

+

Crossover Operators for Binary Variables

- Uniform crossover
  - Select a bit-string z of length n uniformly at random.
  - for all i in 1 to n
    - ★ if $z_i = 1$ then bit i in offspring−1 is x_i else y_i .
    - ★ if $z_i = 1$ then bit i in offspring−2 is y_i else x_i .

parent 1:	1 0 1 0 1 1 1 0 1 0
parent 2:	0 1 0 1 0 0 1 1 1 0
$z =$	1 0 1 0 0 0 1 1 1 0
offspring 1:	1 1 1 1 0 0 1 0 1 0
offspring 2:	0 0 0 0 1 1 1 1 1 0

Mutation Operator for Binary String

The mutation operator introduces small, random changes to a solution's chromosome.

Local Mutation

- One randomly chosen bit is flipped.
- 1 0 1 0 1 1 1 0 1 0
- 1 0 1 1 1 1 1 0 1 0

Global Mutation

- Each bit flipped independently with a given probability p_m,
- It is also called the per bit mutation rate, which is often $= 1/n$, where n is the chromosome length.
- 1 0 1 0 1 1 1 0 1 0
- 1 0 0 0 1 1 1 1 1 0
- **Pr**[k bits flipped] $= \binom{n}{k} . p_m^k . (1 - p_m)^{n-k}$

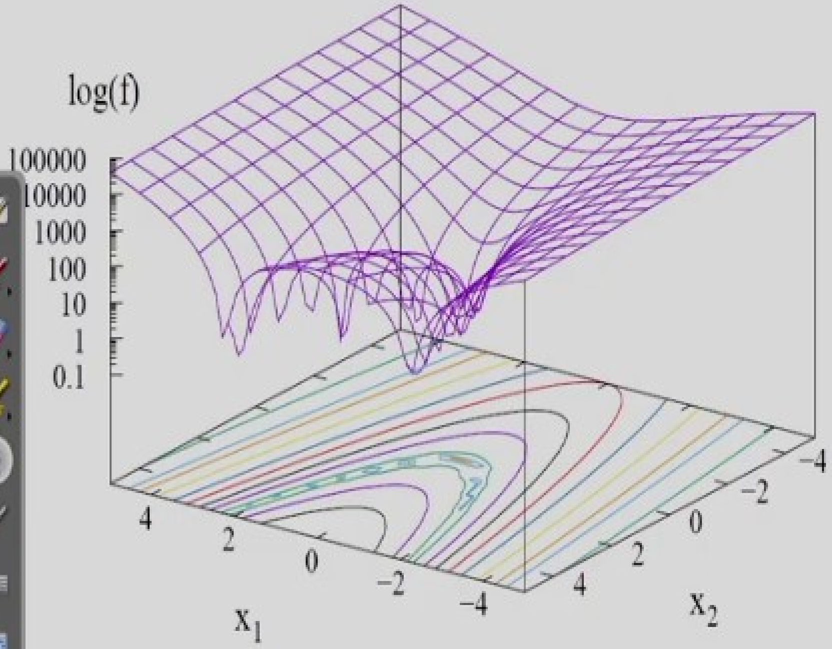
$p_m = \frac{1}{n}$

Simulation and Application

Rosenbrock Function

Rosenbrock Function

Minimize $f(x_1, \ldots, x_N) = \sum_{i=1}^{N-1} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right),$
bounds $-5 \leq x_i \leq 5$ and $i = 1, \ldots, N.$



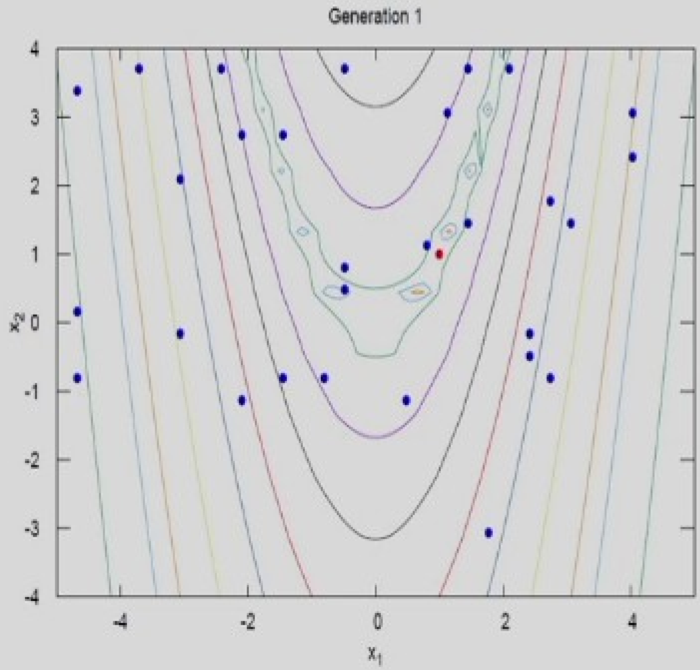
- Optimal solution is $x^* = (1, \ldots, 1)^T$ and $f(x) = 0$

Rosenbrock Function

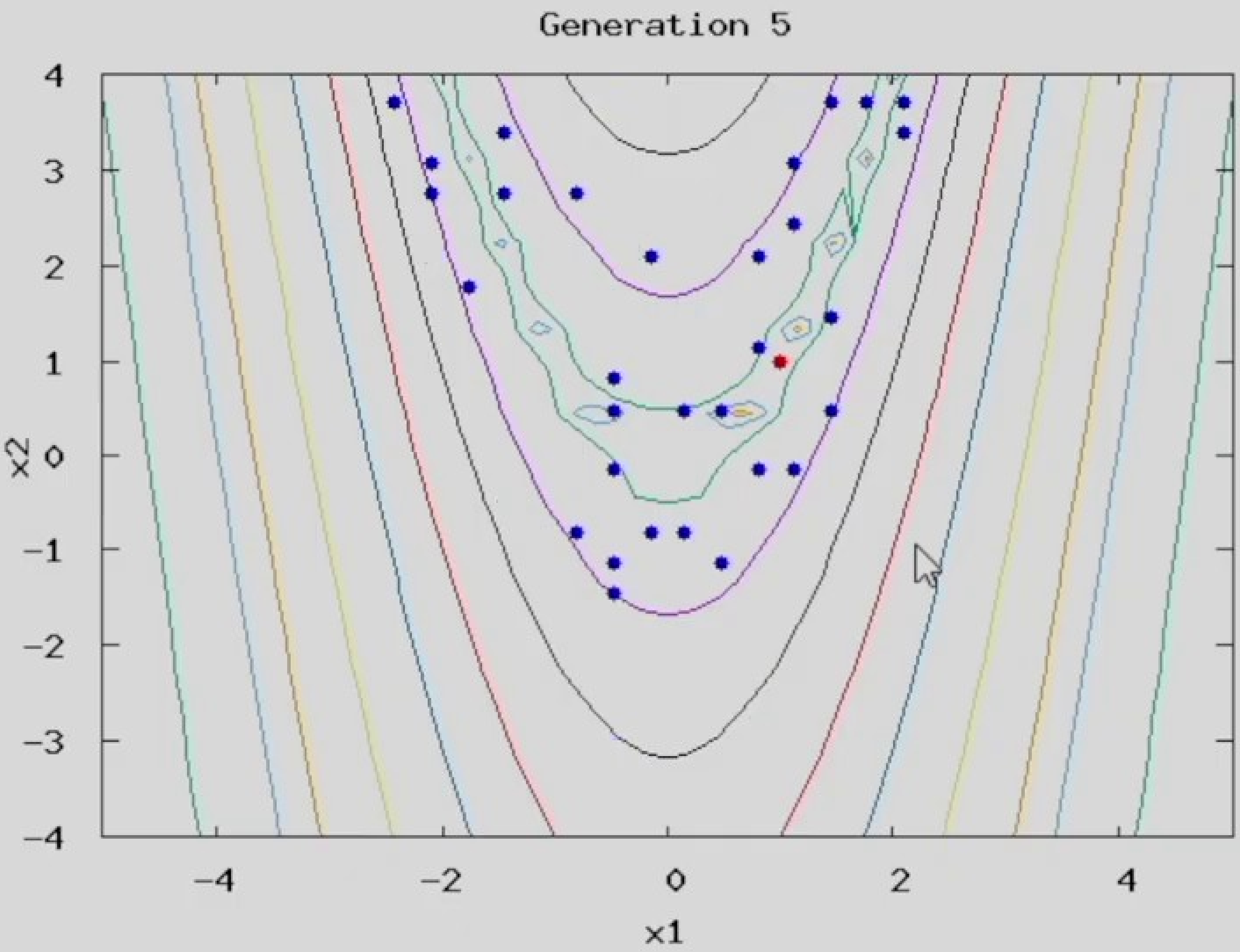
BGA Parameters

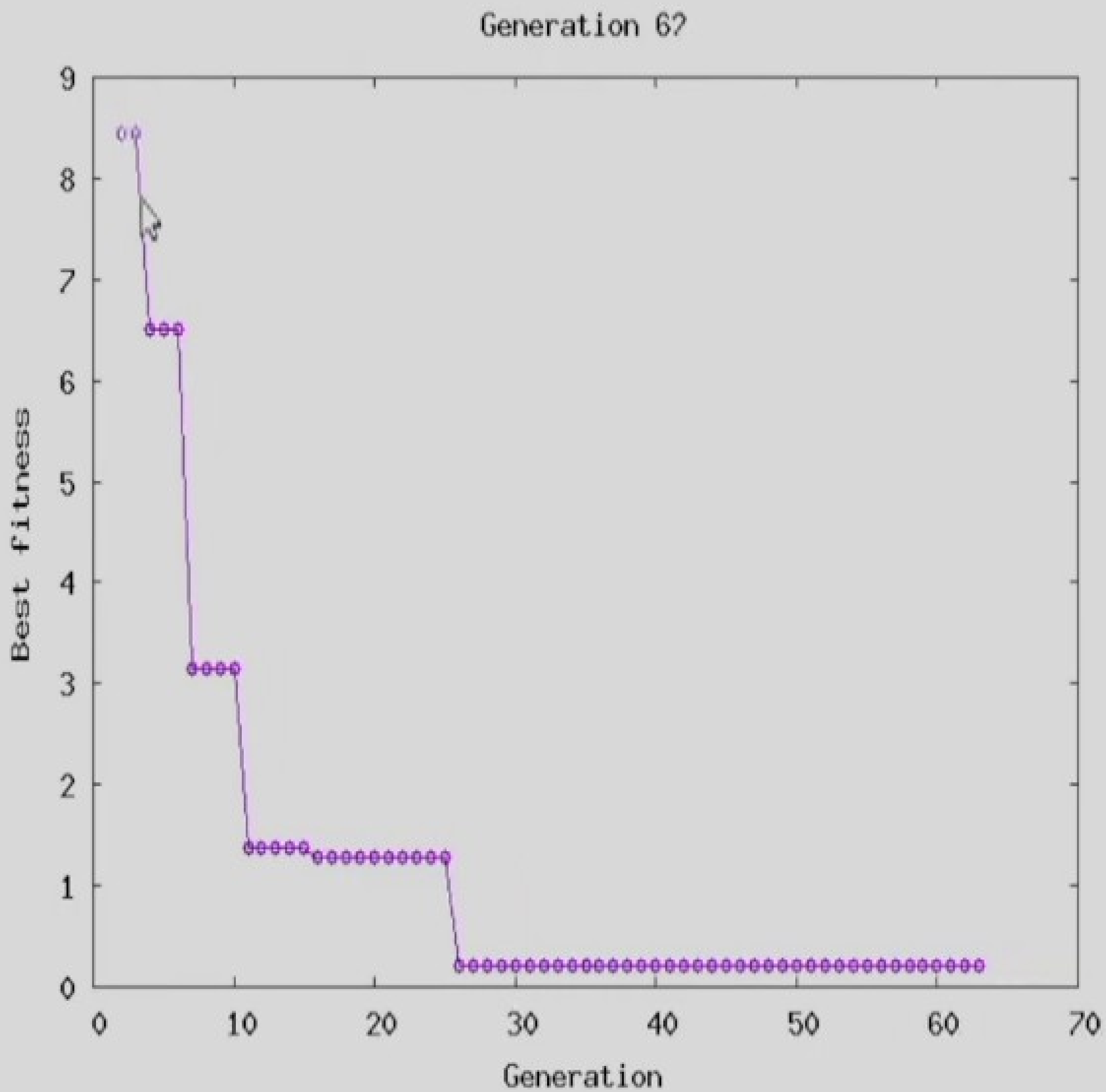
- Number of variables: $N = 2$
- Population size: 40
- No. of generations: 200
- Binary string length of x_1 is $l_1 = 5$
- Binary string length of x_2 is $l_2 = 5$
- Total length of binary string is $l = l_1 + l_2 = 10$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation; $p_m = 1/N = 0.5$
- Binary tournament selection operator
- Single-point crossover operator
- Bit-wise mutation operator

- $(\mu + \lambda)$-strategy



- Simulation ▸ Link
- Progress ▸ Link


Generation 5
4
3
2
1
0
−1
−2
−3
−4
x2
−4
−2
0
2
4
x1

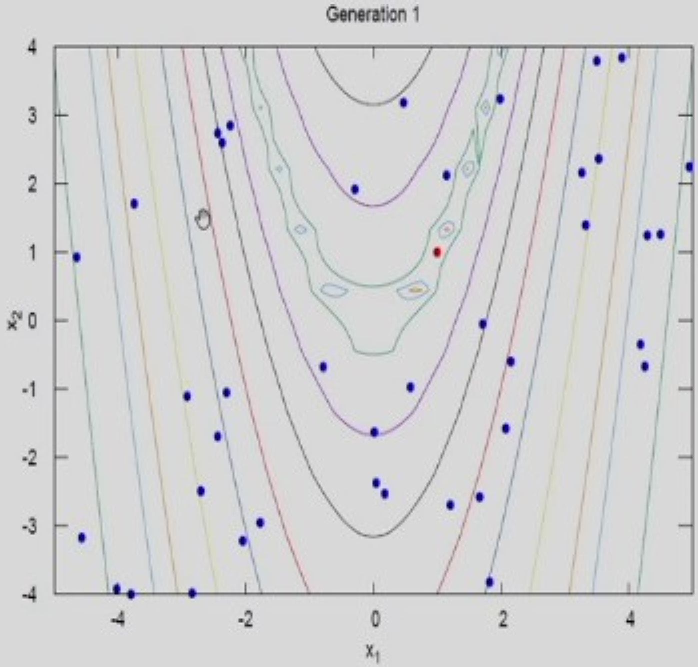

Generation 67
Best fitness
Generation
0
1
2
3
4
5
6
7
8
9
0
10
20
30
40
50
60
70

Rosenbrock Function

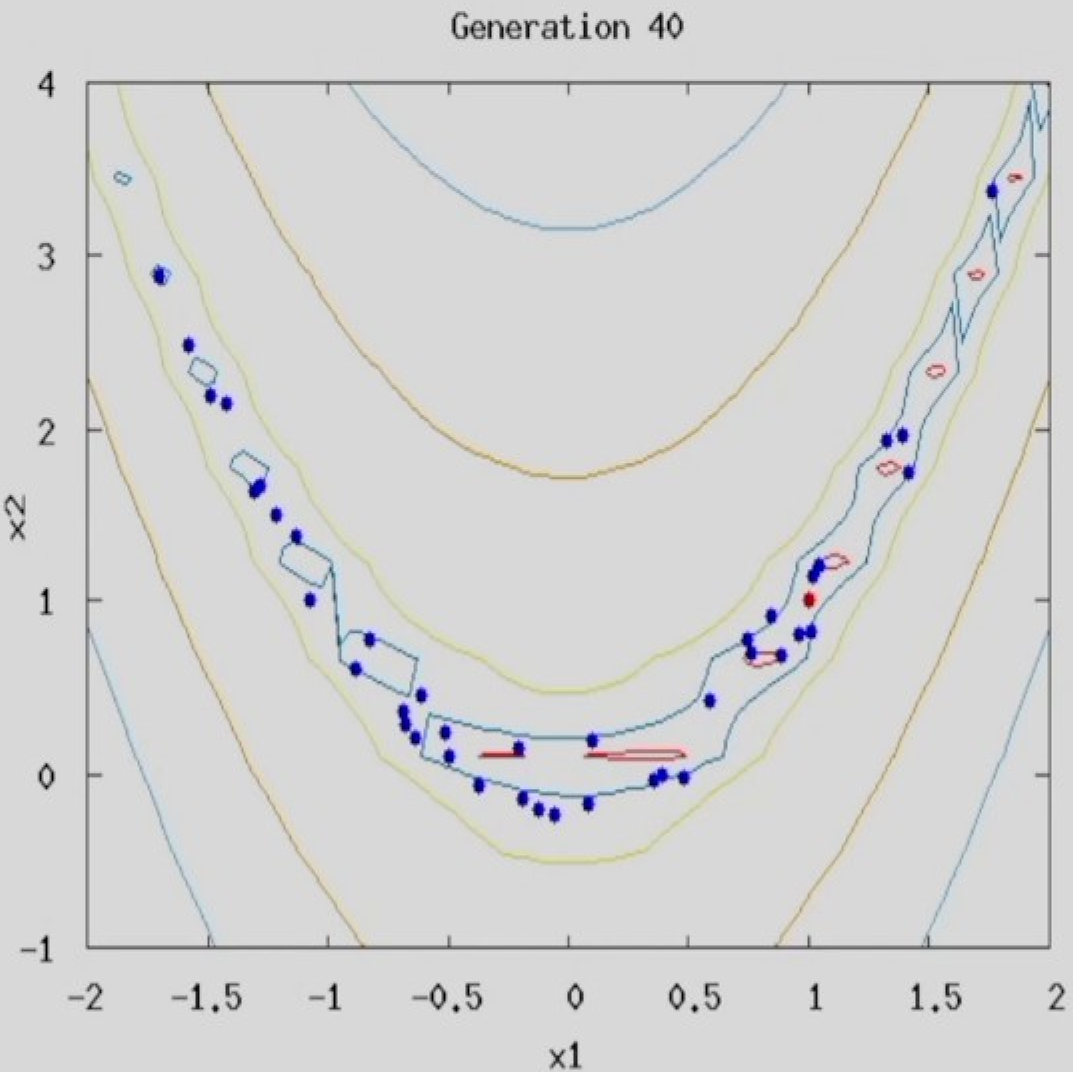
BGA Parameters

- Number of variables: $N = 2$
- Population size: 40
- No. of generations: 200
- Binary string length of x_1 is $l_1 = 10$
- Binary string length of x_2 is $l_2 = 10$
- Total length of binary string is $l = l_1 + l_2 = 20$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation; $p_m = 1/N = 0.5$
- Binary tournament selection operator
- Single-point crossover operator
- Bit-wise mutation operator

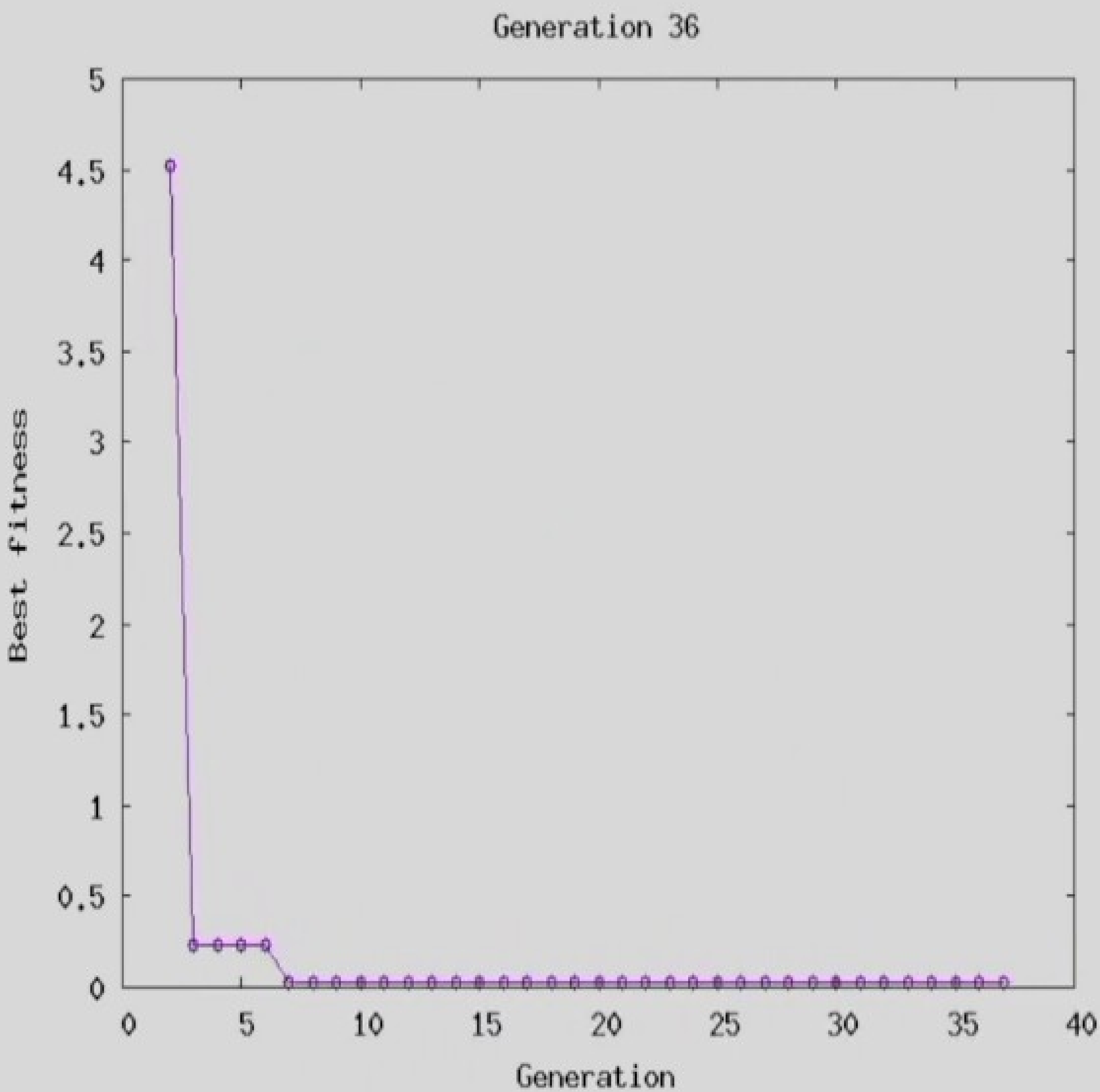
- $(\mu + \lambda)$-strategy



- Simulation ▸ Link
- Progress ▸ Link



Generation 40
x1
x2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
-1
0
1
2
3
4



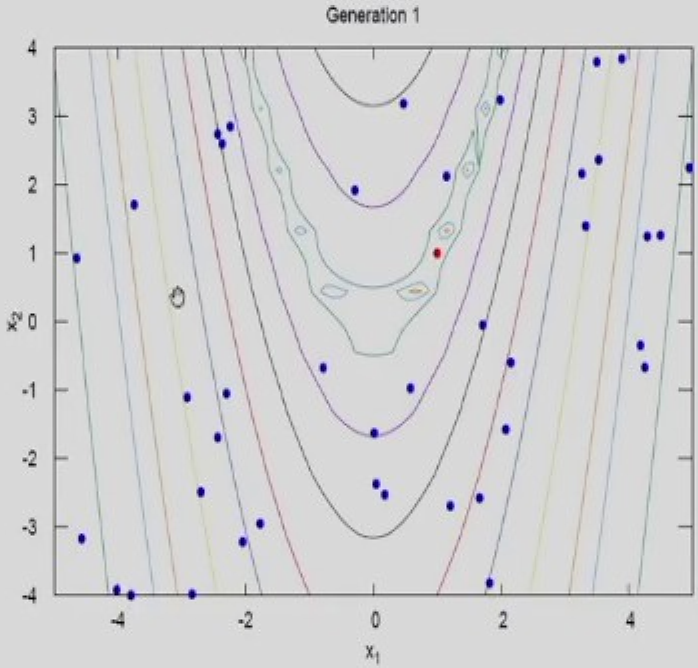
Generation 36
Best fitness
Generation
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
5
10
15
20
25
30
35
40

Rosenbrock Function

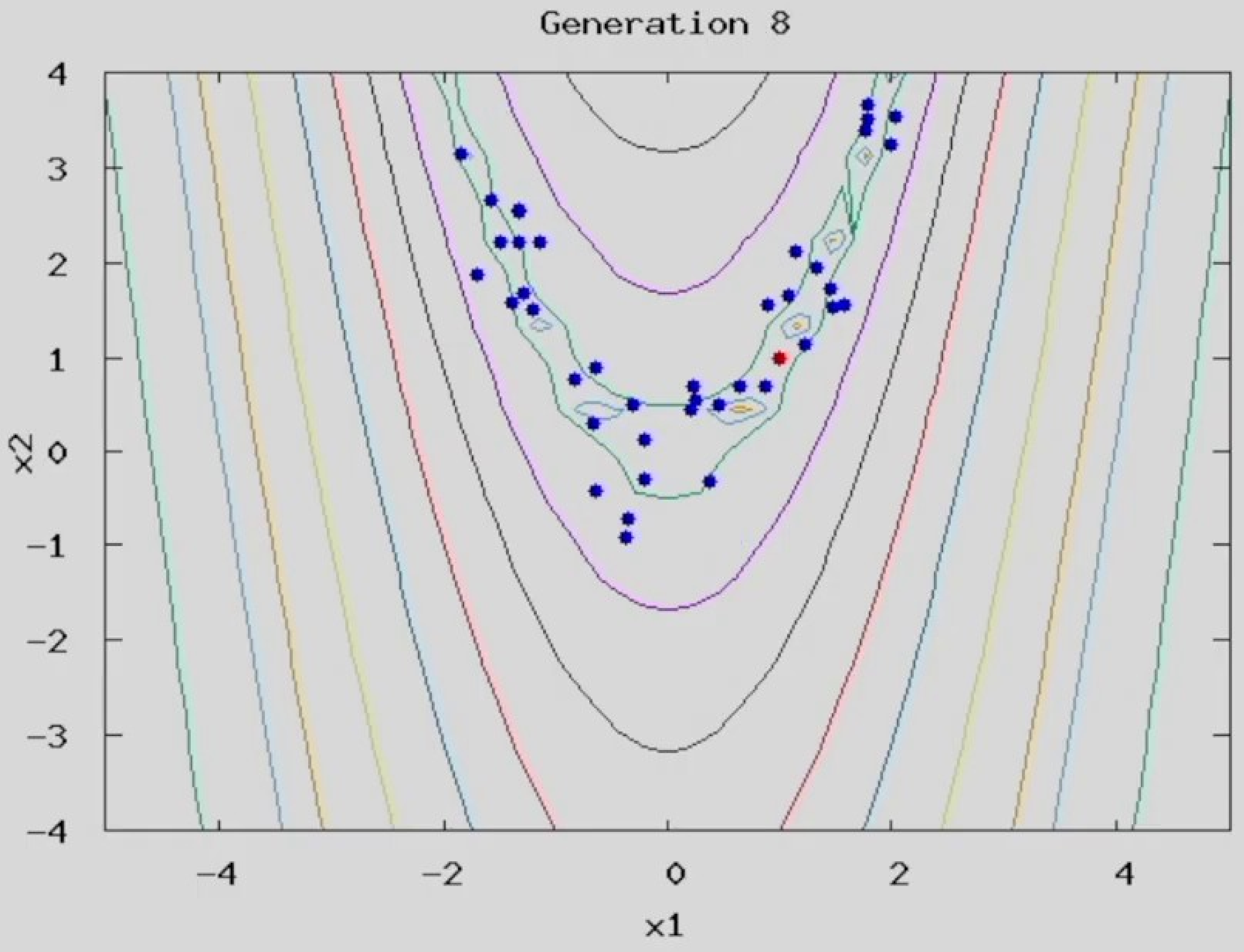
BGA Parameters

- Number of variables: $N = 2$
- Population size: 40
- No. of generations: 200
- Binary string length of x_1 is $l_1 = 20$
- Binary string length of x_2 is $l_2 = 20$
- Total length of binary string is $l = l_1 + l_2 = 40$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation; $p_m = 1/N = 0.5$
- Binary tournament selection operator
- Single-point crossover operator
- Bit-wise mutation operator

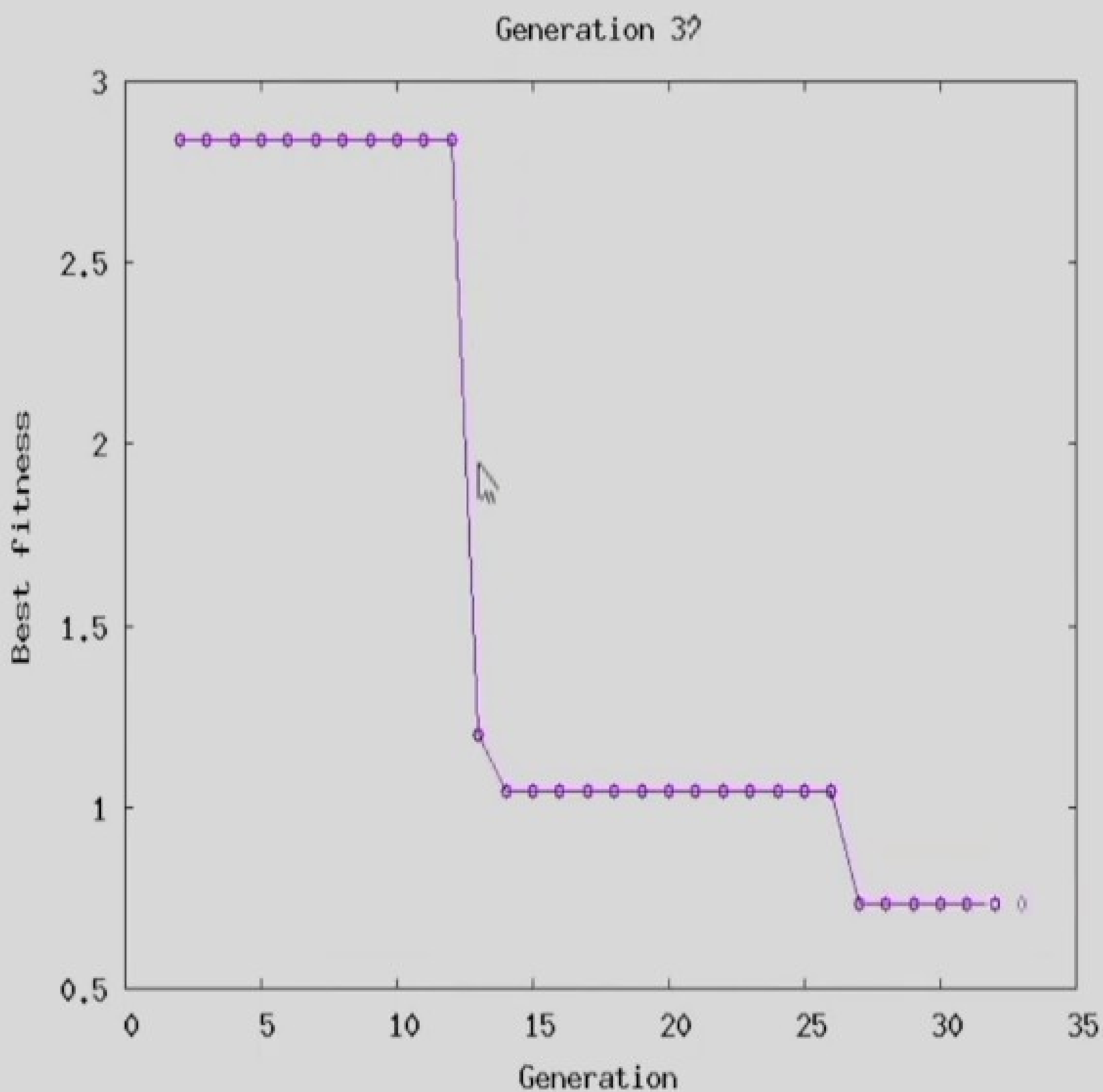
- $(\mu + \lambda)$-strategy



- Simulation Link
- Progress Link



Generation 8
x1
x2
4
3
2
1
0
−1
−2
−3
−4
−4
−2
0
2
4

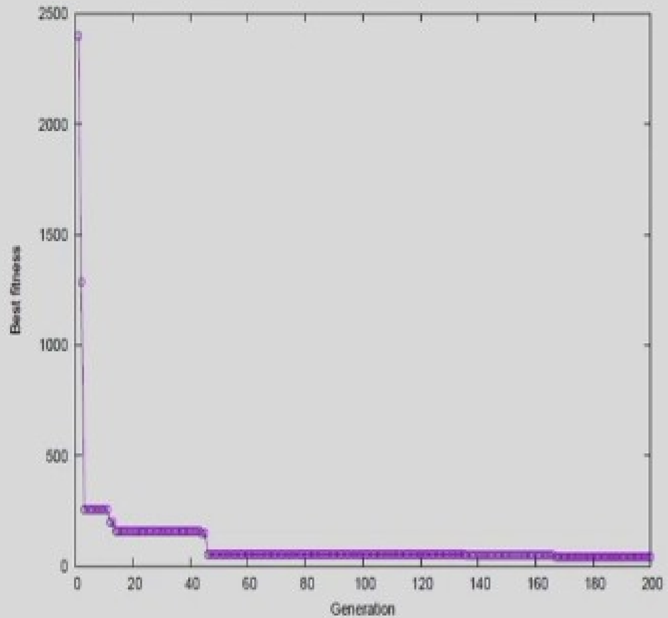

Generation 33
Best fitness
3
2.5
2
1.5
1
0.5
0
5
10
15
20
25
30
35
Generation

Rosenbrock Function

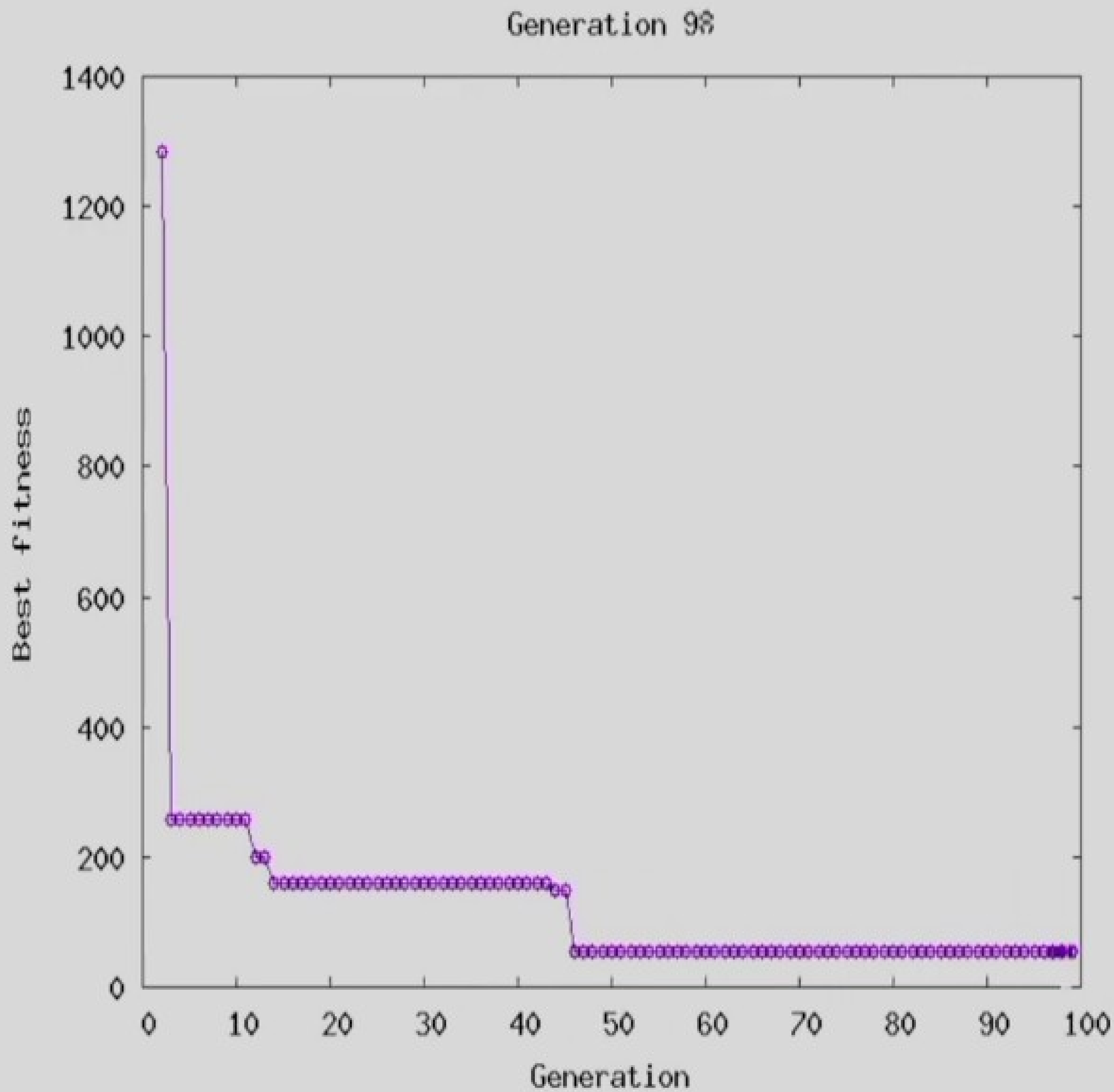
BGA Parameters

- Number of variables: $N = 4$
- Population size: 40
- No. of generations: 200
- Binary string length of x_1 is $l_1 = 20$
- Binary string length of x_2 is $l_2 = 20$
- Total length of binary string is $l = l_1 + l_2 = 40$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation; $p_m = 1/N = 0.5$
- Binary tournament selection operator
- Single-point crossover operator
- Bit-wise mutation operator

- $(\mu + \lambda)$-strategy



- Progress Link

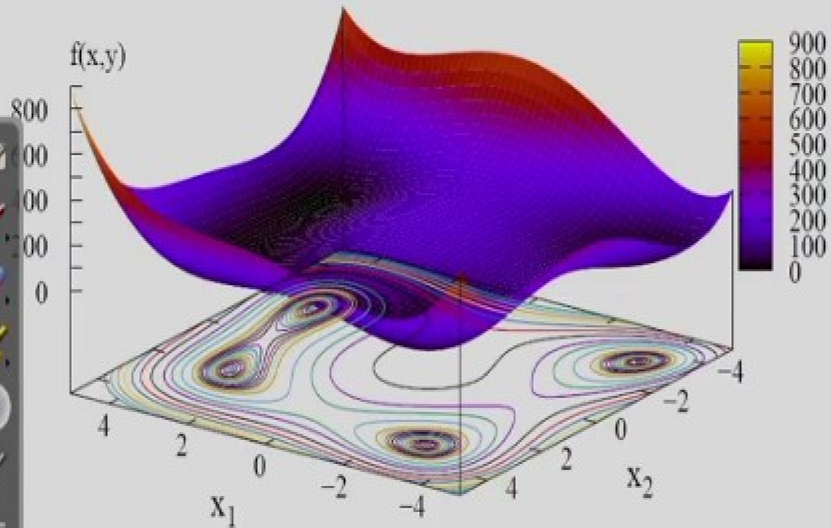


Generation 98
Best fitness
Generation
0
200
400
600
800
1000
1200
1400
0
10
20
30
40
50
60
70
80
90
100

Himmelblau Function

Himmelblau Function

Minimize $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2,$
bounds $-5 \le x_1 \le 5$ and $-5 \le x_2 \le 5.$

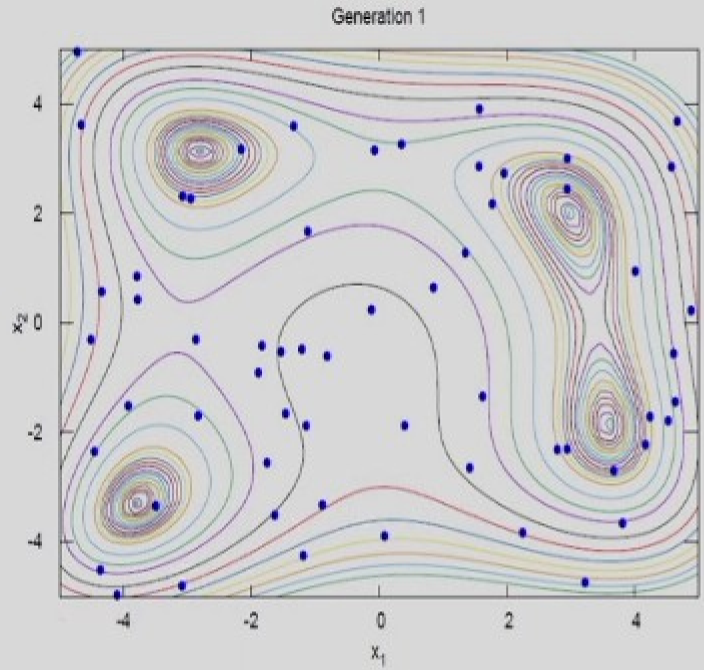


- Multi-modal function: it has 4 minimum points
- First optimal solution is $x^* = (3, 2)^T$ and $f(x) = 0$
- Second optimal solution is $x^* = (-2.805, 3.131)^T$ and $f(x) = 0$
- Third optimal solution is $x^* = (-3.779, -3.283)^T$ and $f(x) = 0$
- Fourth optimal solution is $x^* = (3.584, -1.848)^T$ and $f(x) = 0$

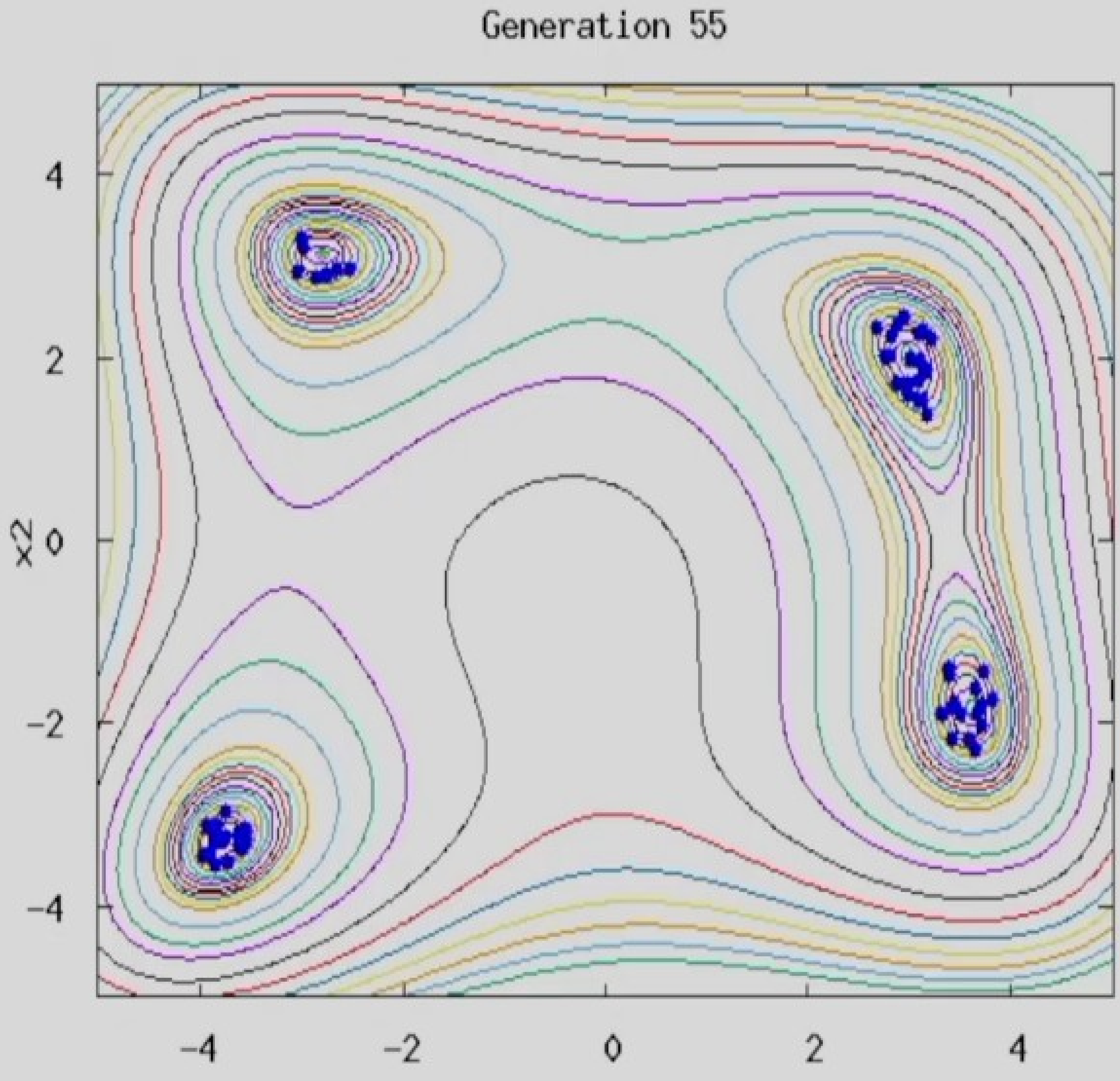
Himmelblau Function

BGA Parameters

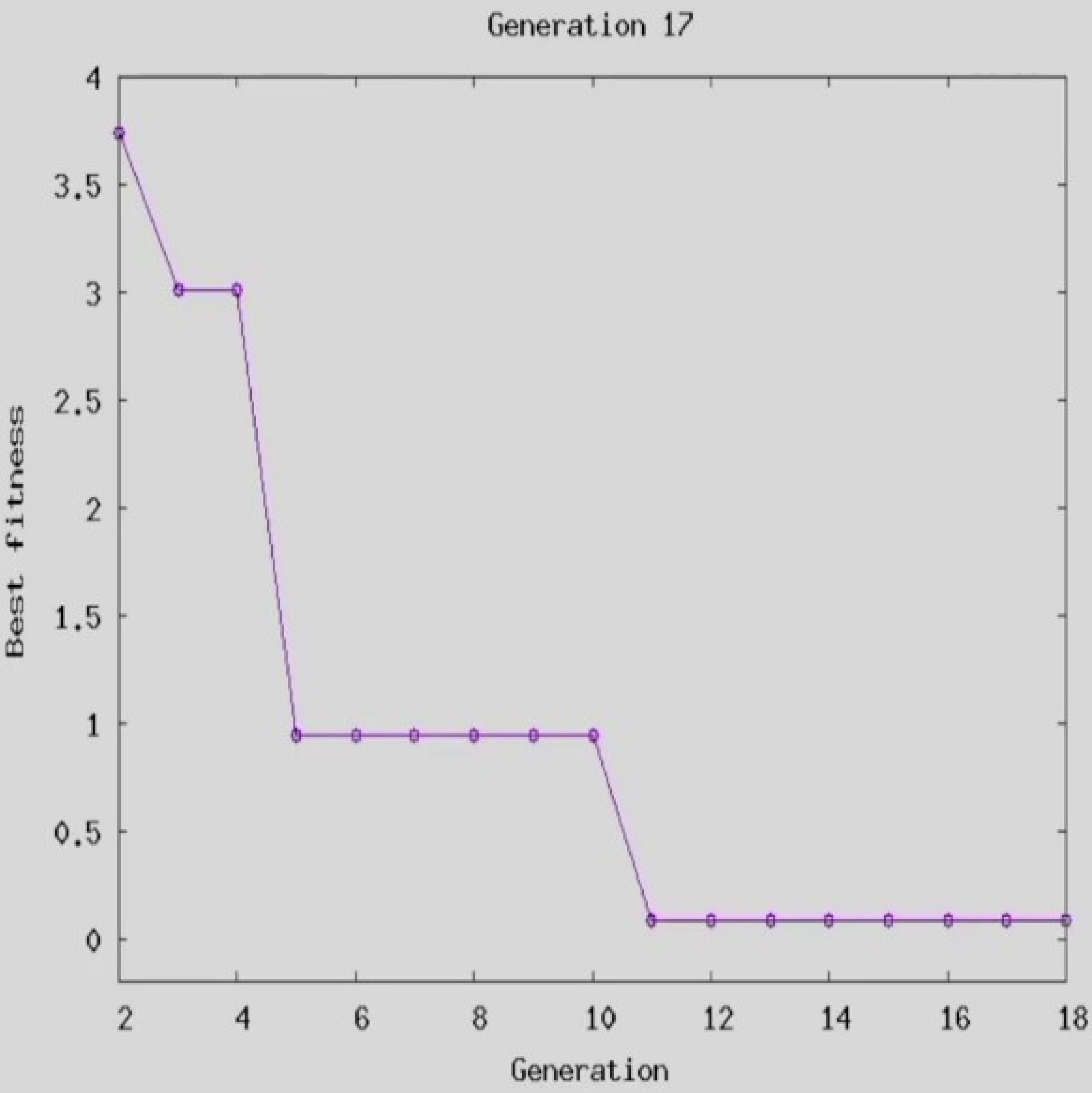
- Population size: 60
- No. of generations: 200
- Binary string length of x_1 is $l_1 = 20$
- Binary string length of x_2 is $l_2 = 20$
- Total length of binary string is $l = l_1 + l_2 = 40$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation; $p_m = 1/N = 0.5$
- Binary tournament selection operator
- Single-point crossover operator
- Bit-wise mutation operator
- $(\mu + \lambda)$–strategy



- Simulation Link
- Progress Link



Generation 55
4
2
0
−2
−4
x2
−4
−2
0
2
4
x1

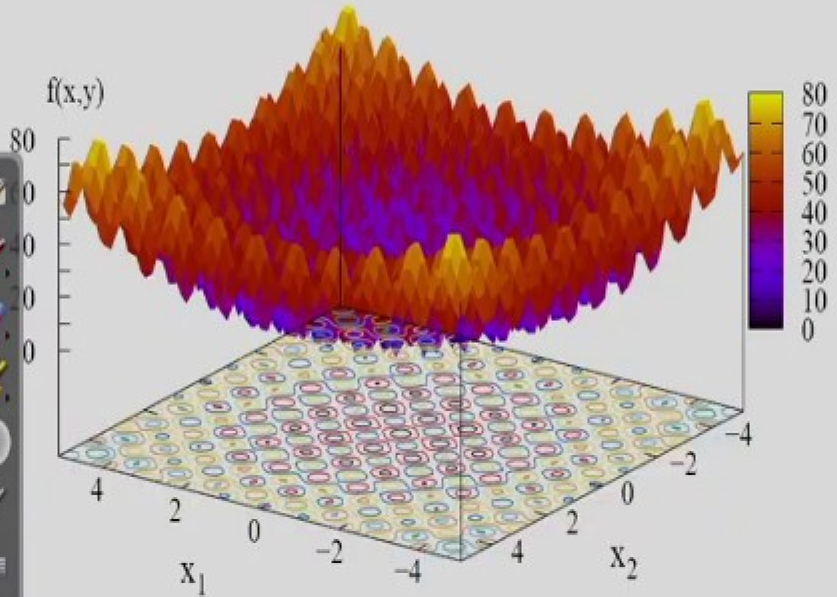

Generation 17
Best fitness
Generation
4
3.5
3
2.5
2
1.5
1
0.5
0
2
4
6
8
10
12
14
16
18

Rastrigin Function

Rastrigin Function

Minimize $f(x_1, \ldots, x_N) = 10N + \sum_{i=1}^{N}(x_i^2 - 10\cos(2 * \pi x_i)),$

bounds $-5.12 \leq x_i \leq 5.12$



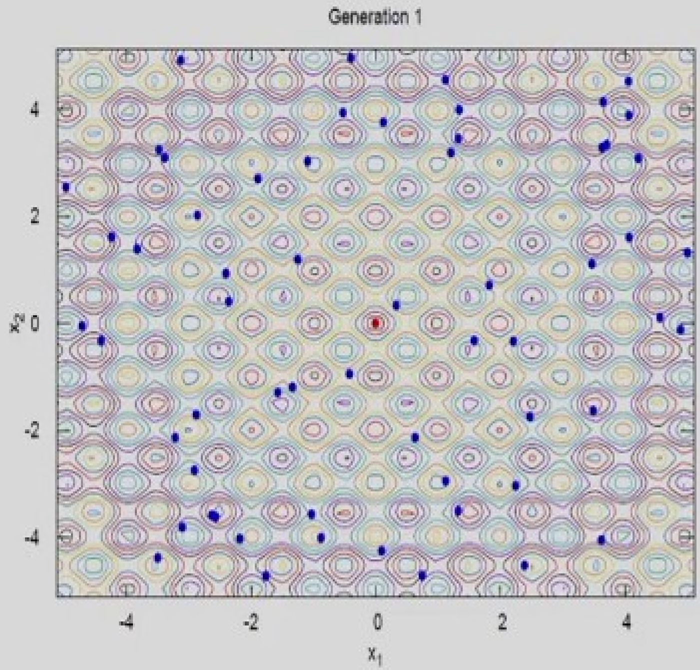
- Optimal solution is $x^* = (0, \ldots, 0)^T$ and $f(x) = 0$

Rastrigin Function

BGA Parameters

- Number of variable: $N = 2$
- Population size: 60
- No. of generations: 200
- Binary string length of x_1 is $l_1 = 20$
- Binary string length of x_2 is $l_2 = 20$
- Total length of binary string is $l = l_1 + l_2 = 40$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation; $p_m = 1/N = 0.5$
- Binary tournament selection operator
- Single-point crossover operator
- Bit-wise mutation operator

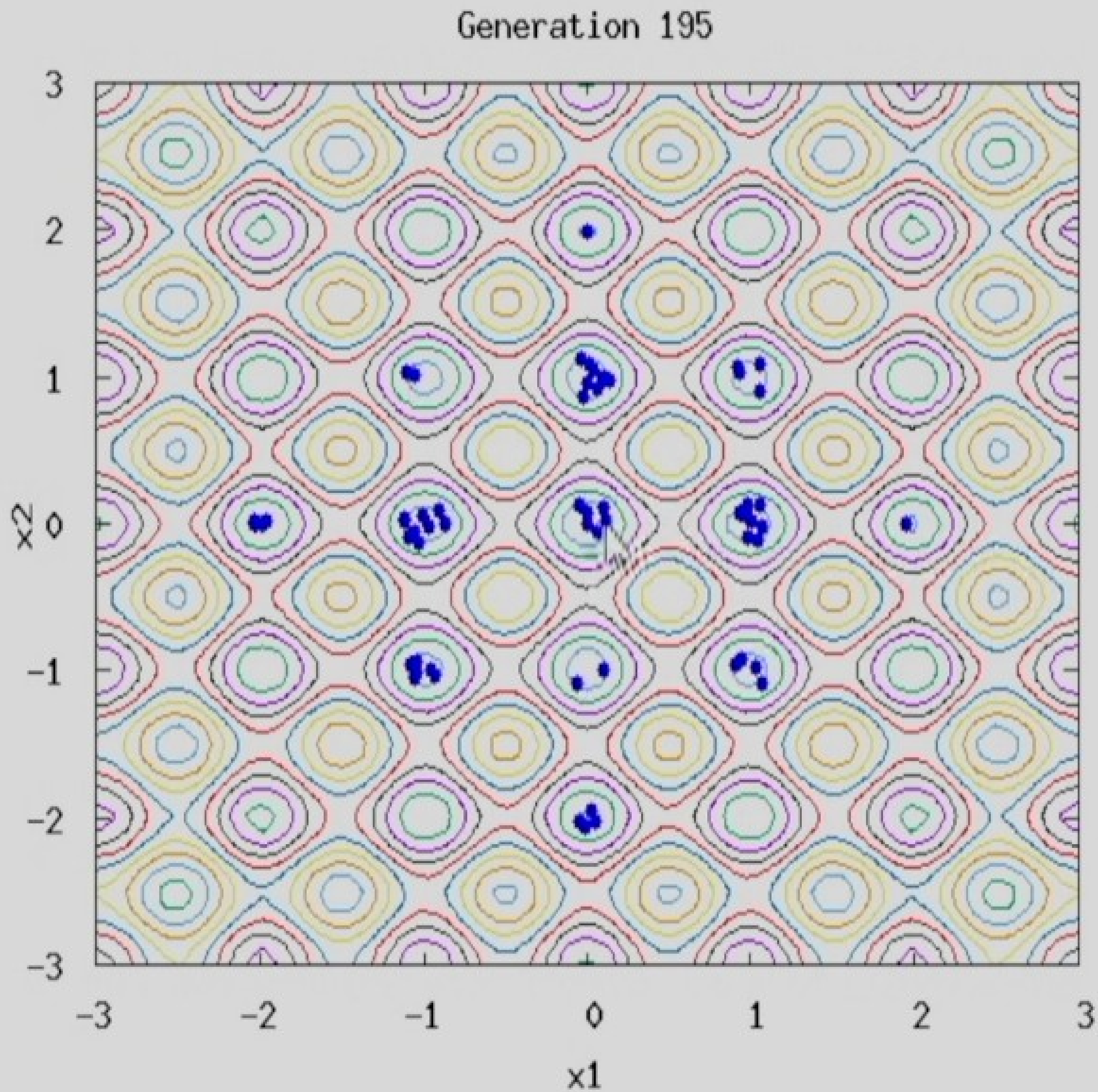
- $(\mu + \lambda)$-strategy

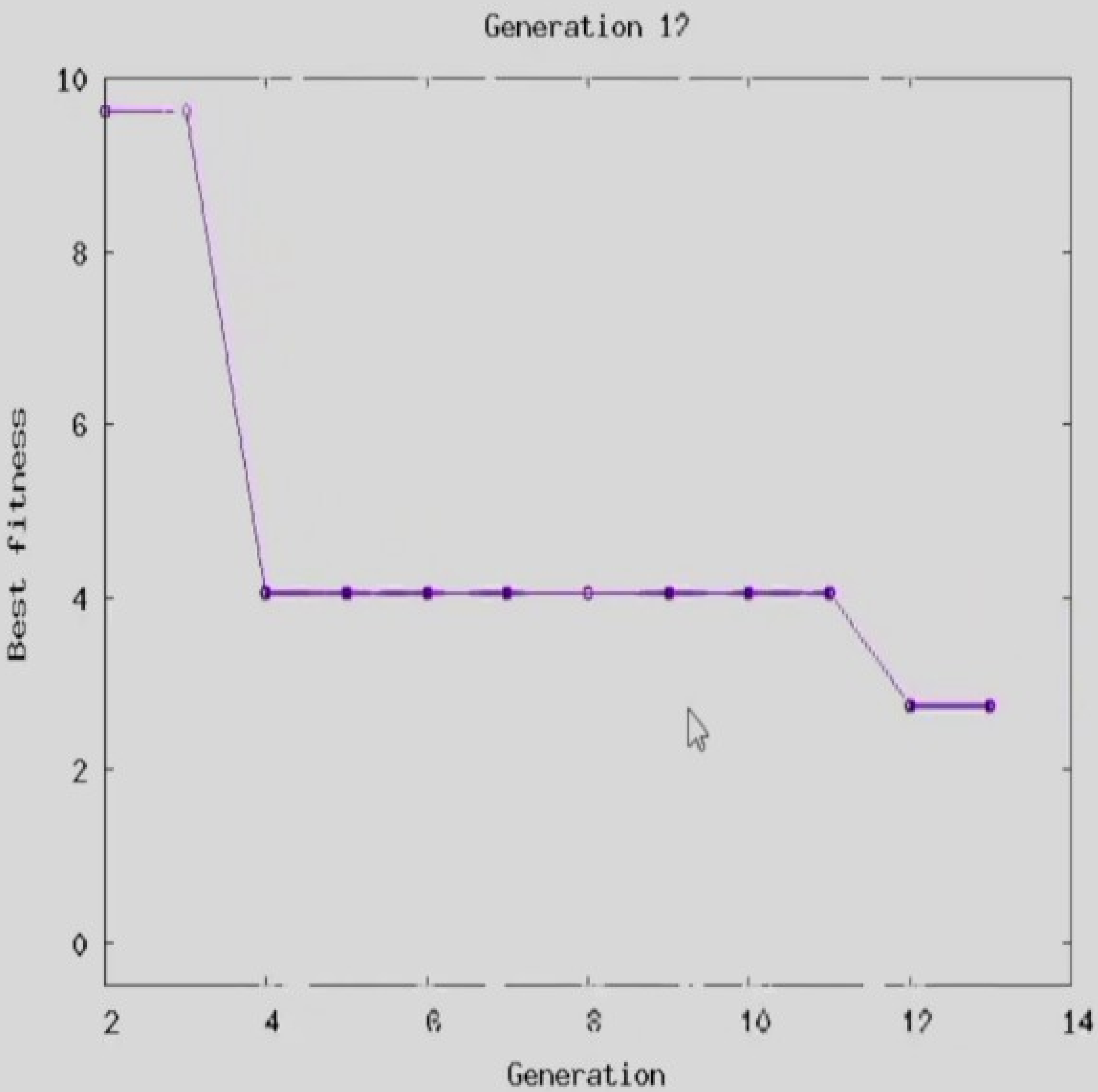


- Simulation Link

file:///C|/Users/CET-IITG/Desktop/MOOCs%20%20January%202021/Deepak%20Sharma/rastrigin_bga/Rastrigin_simulation_bga.gif

- Progress


Generation 195
3
2
1
0
−1
−2
−3
x2
−3
−2
−1
0
1
2
3
x1



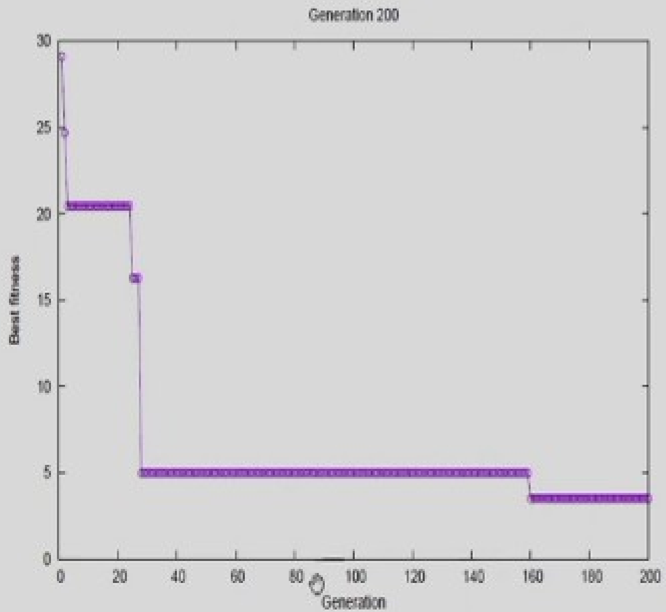
Generation 12
Best fitness
Generation
10
8
6
4
2
0
2
4
6
8
10
12
14

Rastrigin Function

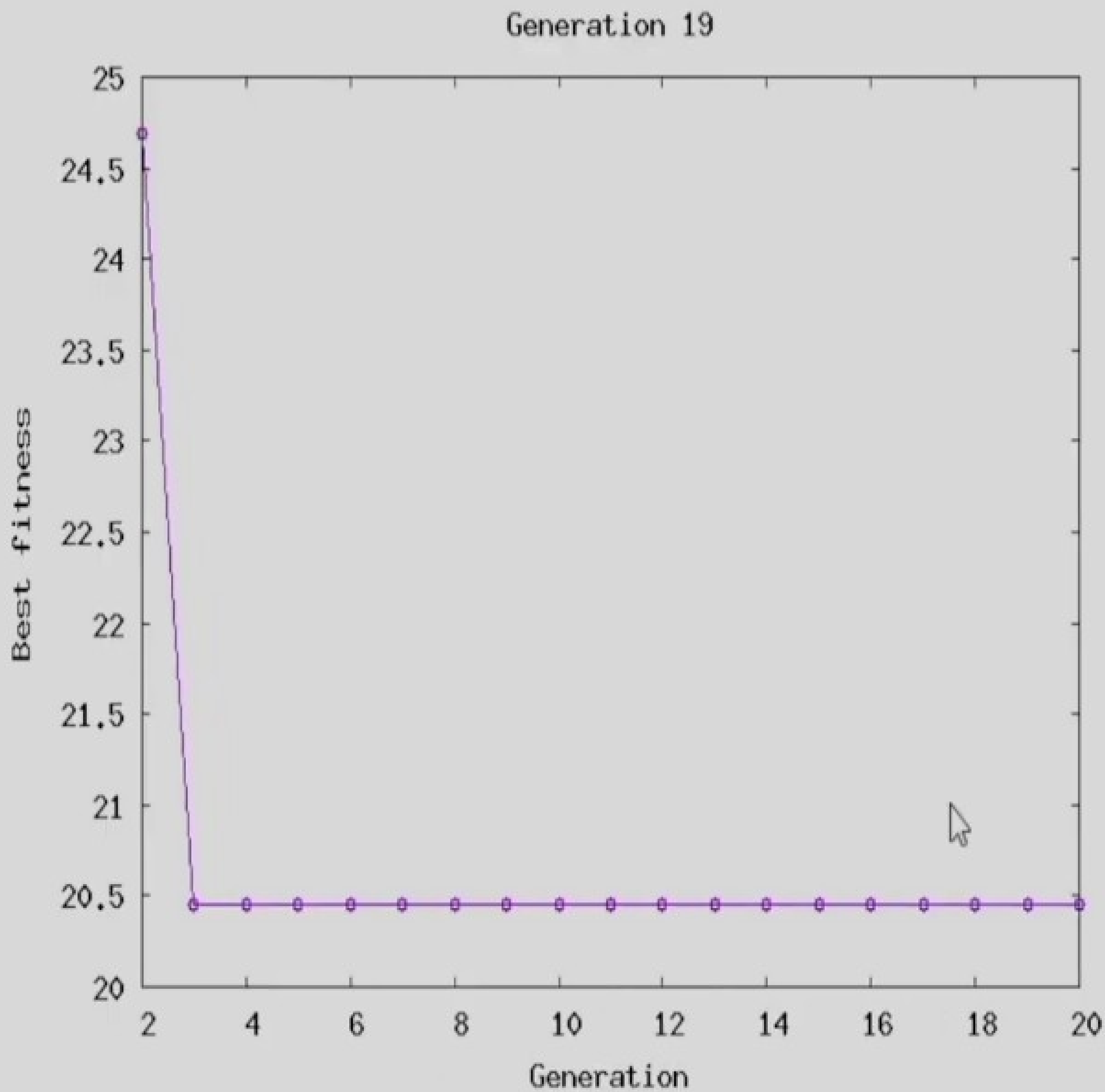
BGA Parameters

- Number of variable: $N = 4$
- Population size: 60
- No. of generations: 200
- Binary string length of x_1 is $l_1 = 20$
- Binary string length of x_2 is $l_2 = 20$
- Total length of binary string is $l = l_1 + l_2 = 40$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation; $p_m = 1/N = 0.5$
- Binary tournament selection operator
- Single-point crossover operator
- Bit-wise mutation operator

- $(\mu + \lambda)$-strategy



- Progress ▸ Link



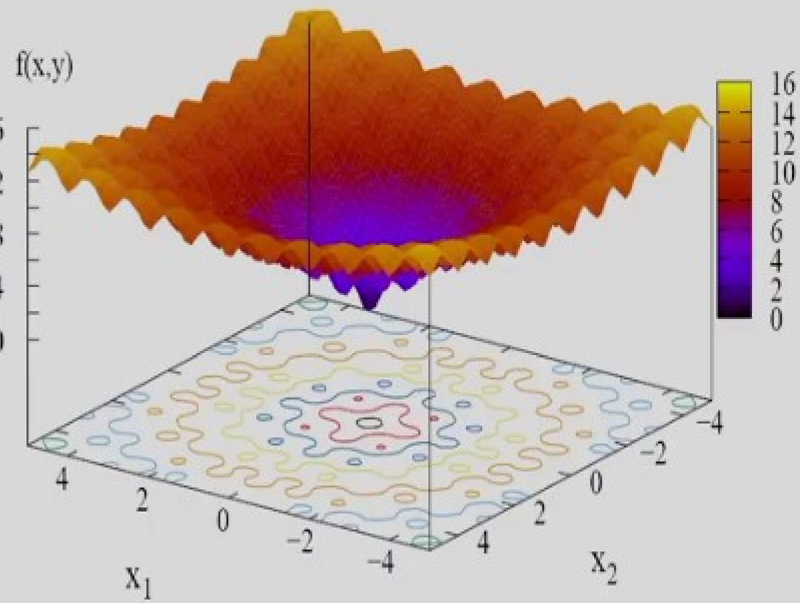
Generation 19
Best fitness
Generation
25
24.5
24
23.5
23
22.5
22
21.5
21
20.5
20
2
4
6
8
10
12
14
16
18
20

Ackley Function

Ackley Function

$$\text{Minimize} \quad f(x_1, x_2) = -20\exp\left(-0.2\sqrt{(0.5(x_1^2 + x_2^2))}\right) - \exp\left(0.5(\cos(2\pi x_1) + \cos(2\pi x_2))\right) + \exp(1) + 20,$$

$$\text{bounds} \quad -5 \le x_1 \le 5 \text{ and } -5 \le x_2 \le 5.$$

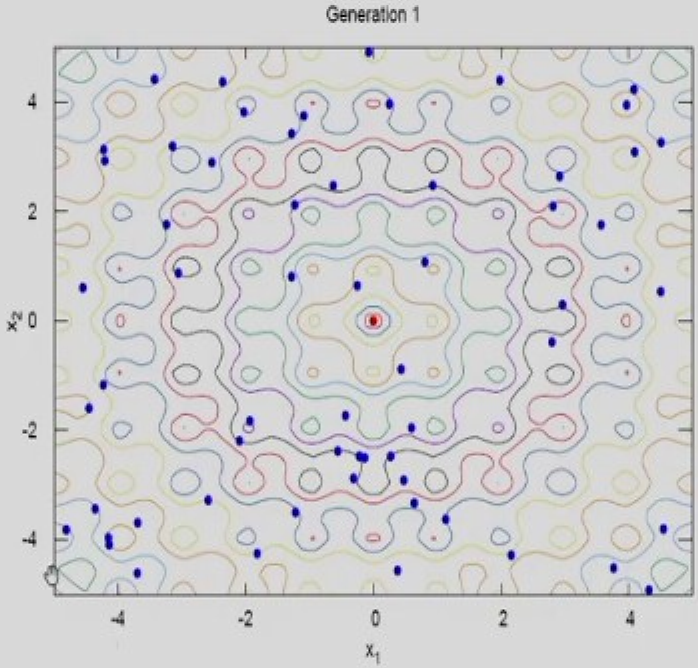


- Optimal solution is $x^* = (0,0)^T$ and $f(x) = 0$

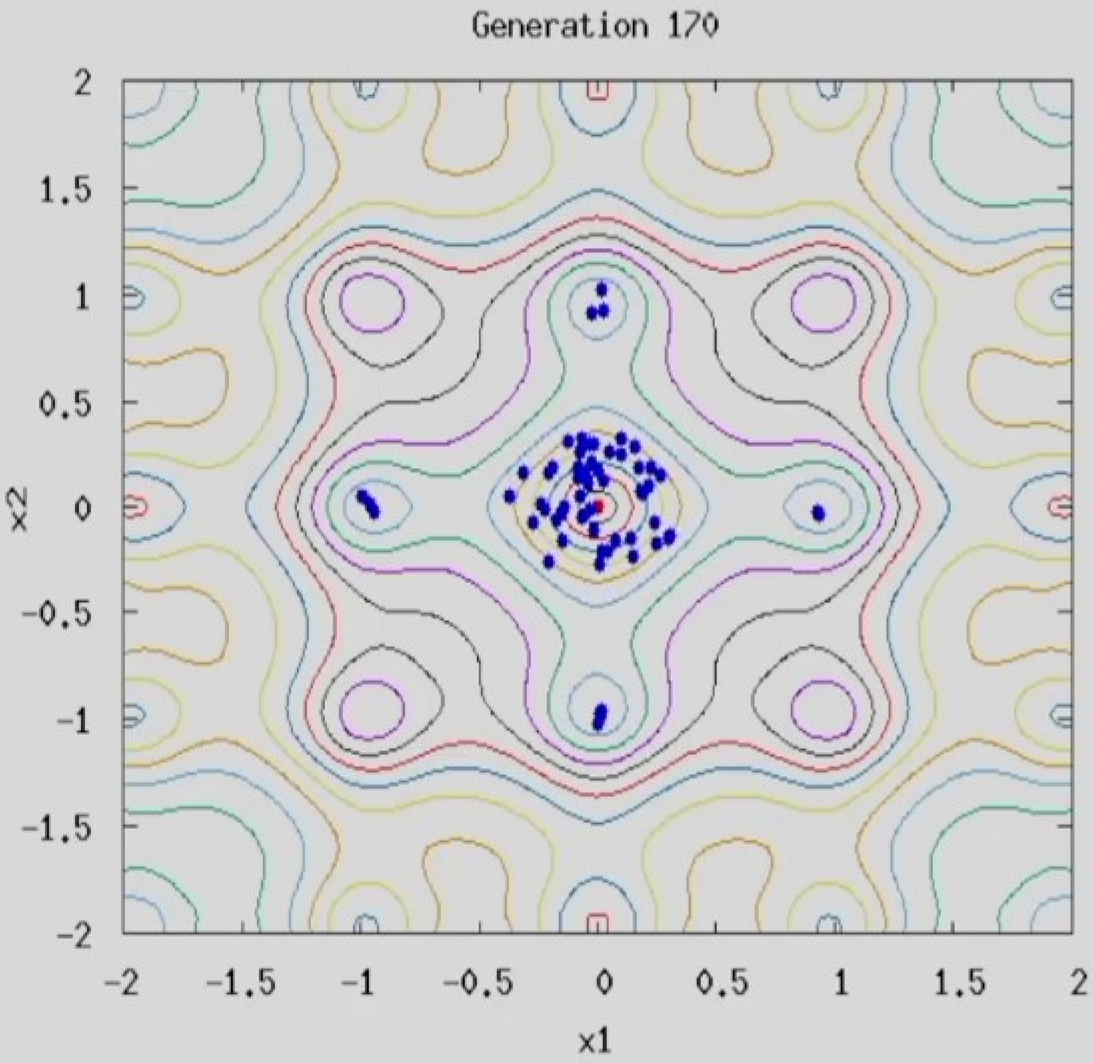
Ackley Function

BGA Parameters

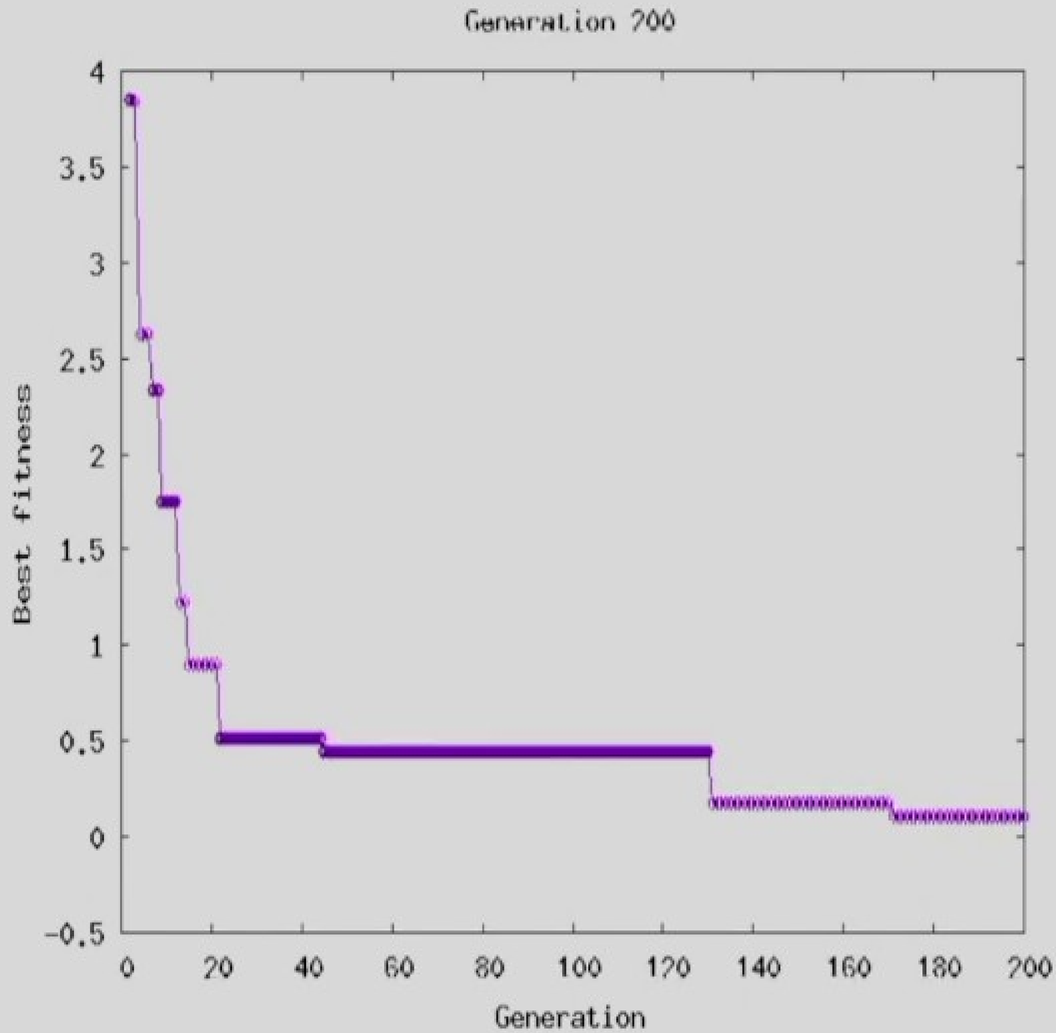
- Population size: 60
- No. of generations: 200
- Binary string length of x_1 is $l_1 = 20$
- Binary string length of x_2 is $l_2 = 20$
- Total length of binary string is $l = l_1 + l_2 = 40$
- Probability of crossover: $p_c = 1.0$
- Probability of mutation; $p_m = 1/N = 0.5$
- Binary tournament selection operator
- Single-point crossover operator
- Bit-wise mutation operator
- $(\mu + \lambda)$–strategy



- Simulation Link
- Progress Link



Generation 170
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
x2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
x1



Generation 200
Best fitness
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
20
40
60
80
100
120
140
160
180
200
Generation

Closure

- Binary-coded genetic algorithm
  - Solution representation
  - Working principles through an example: selection operator, crossover and mutation operators, survivor operator
  - Graphical example
- More operators
  - selection operator, crossover and mutation operators, survivor operator
- Simulation and application of BGA on four mathematical problems