Outline

1. **Simulations**
   - Rosenbrock Function
   - Himmelblau Function
   - Rastrigin Function
   - Ackley Function
2. **Algorithmic Implementation of DE**
   - Data structure
   - Input
   - Random initial population
   - Fitness assignment
   - Mutant Vector
   - Trial Vector
   - Greedy Selection of DE
3. **Closure**

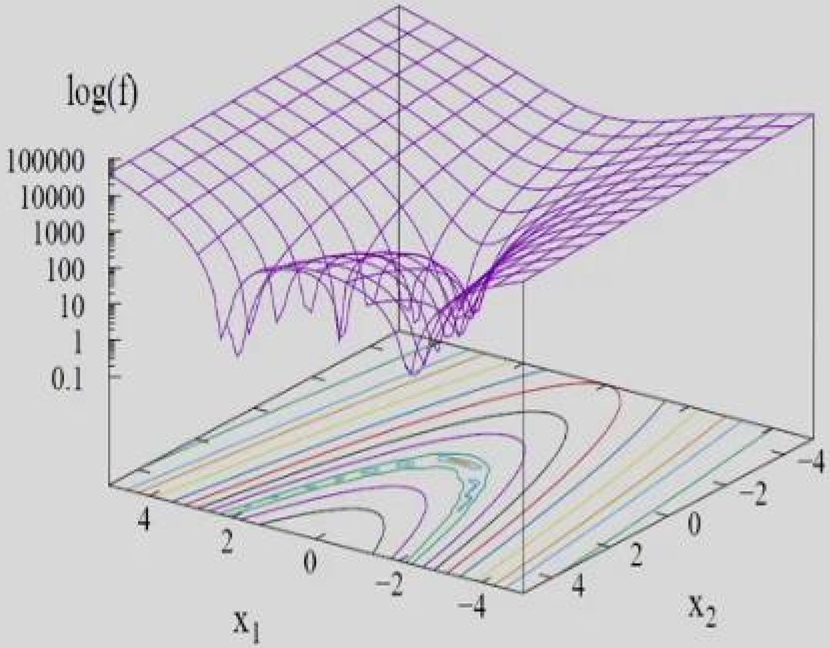
Recap

- Differential evolution (DE)
  - Introduction
  - Mutant vector
  - Trial vector
  - Greedy selection of canonical DE
- Flow chart of DE
- DE on the generalized framework of EC techniques
- Working principles through an example
- Graphical example

Rosenbrock Function

Rosenbrock Function

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

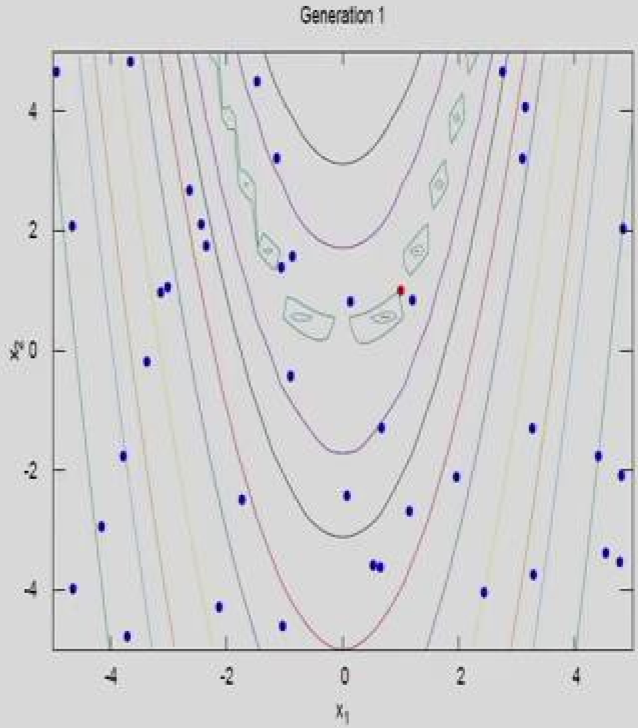


- Optimal solution is $x^* = (1, \ldots, 1)^T$ and $f(x) = 0$

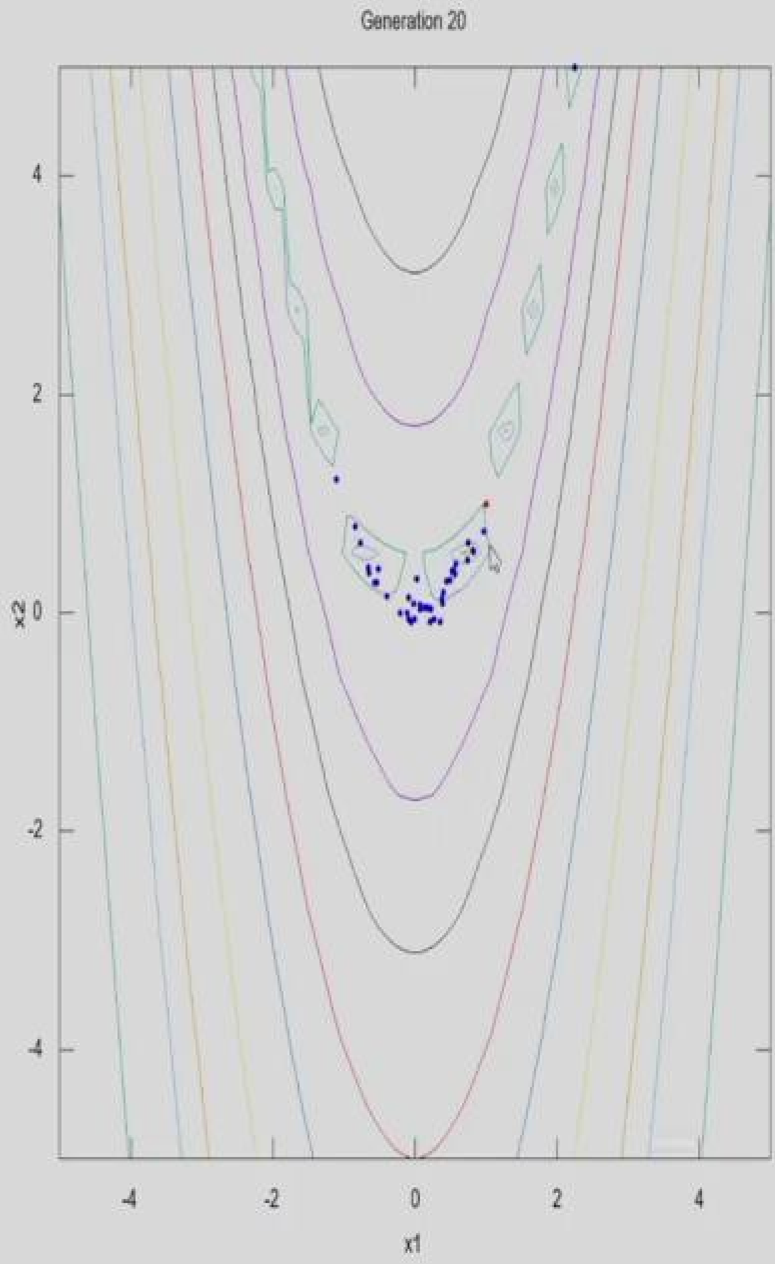
Rosenbrock Function

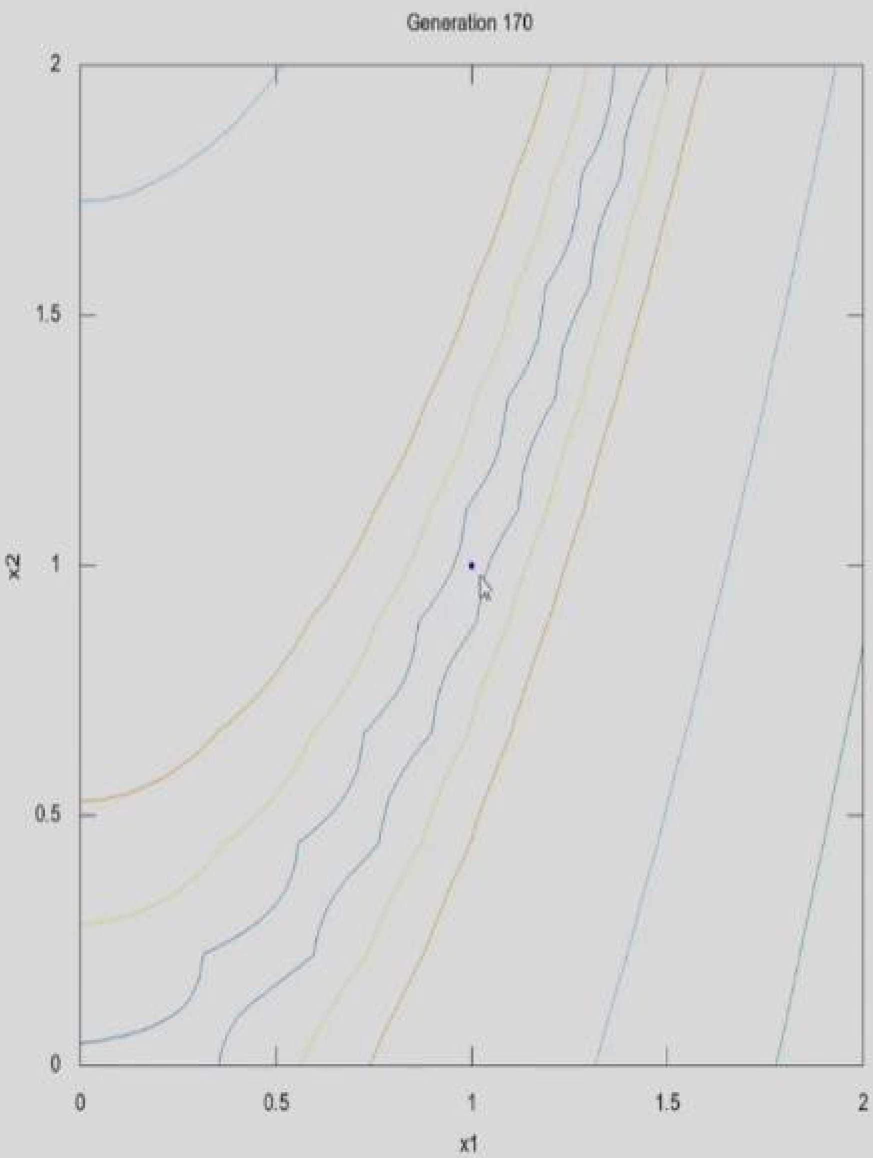
DE Parameters

- Number of variables: $n = 2$
- Population size: $N = 40$
- No. of generations: $T = 200$
- DE/rand/1/bin
- $F = 0.5$
- $p_c = 0.5$

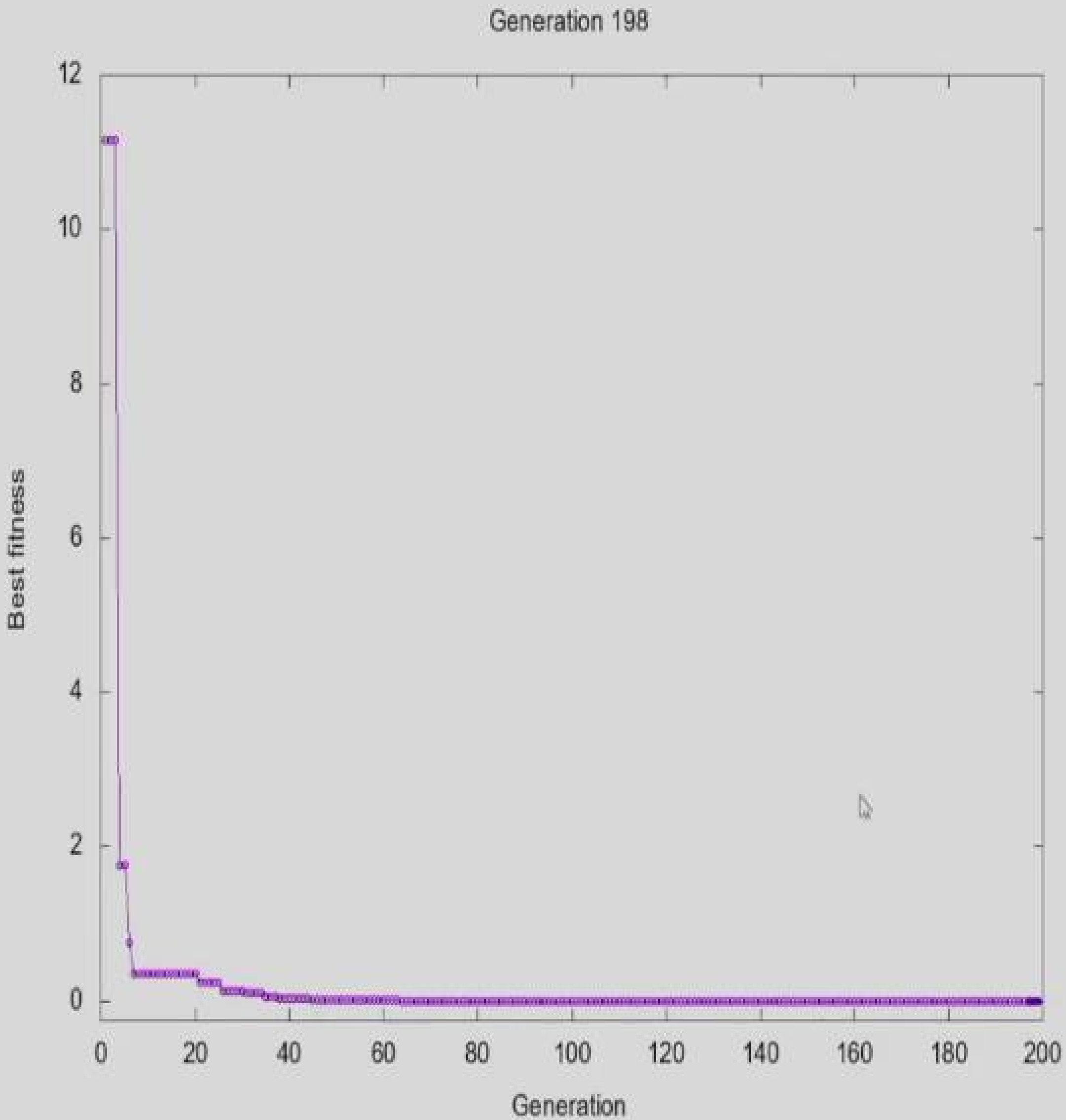


- Simulation Link
- Progress Link


Generation 20
x2
4
2
0
-2
-4
x1
-4
-2
0
2
4



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

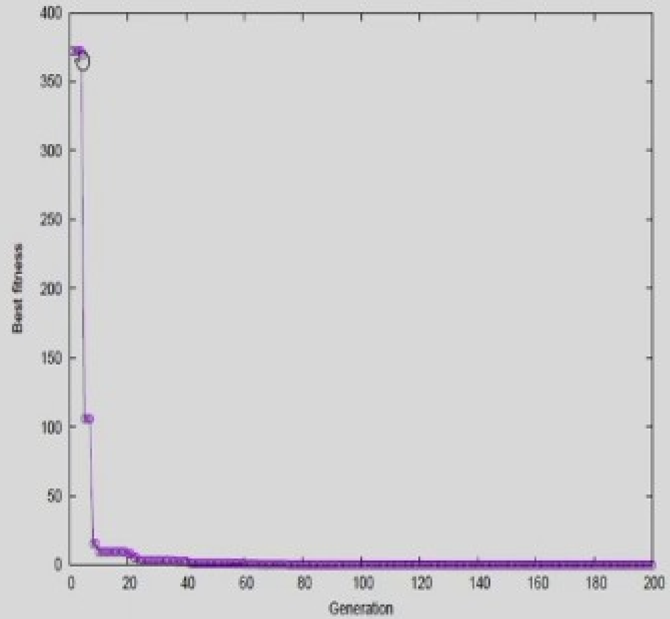


Generation 198
Best fitness
12
10
8
6
4
2
0
0
20
40
60
80
100
120
140
160
180
200
Generation

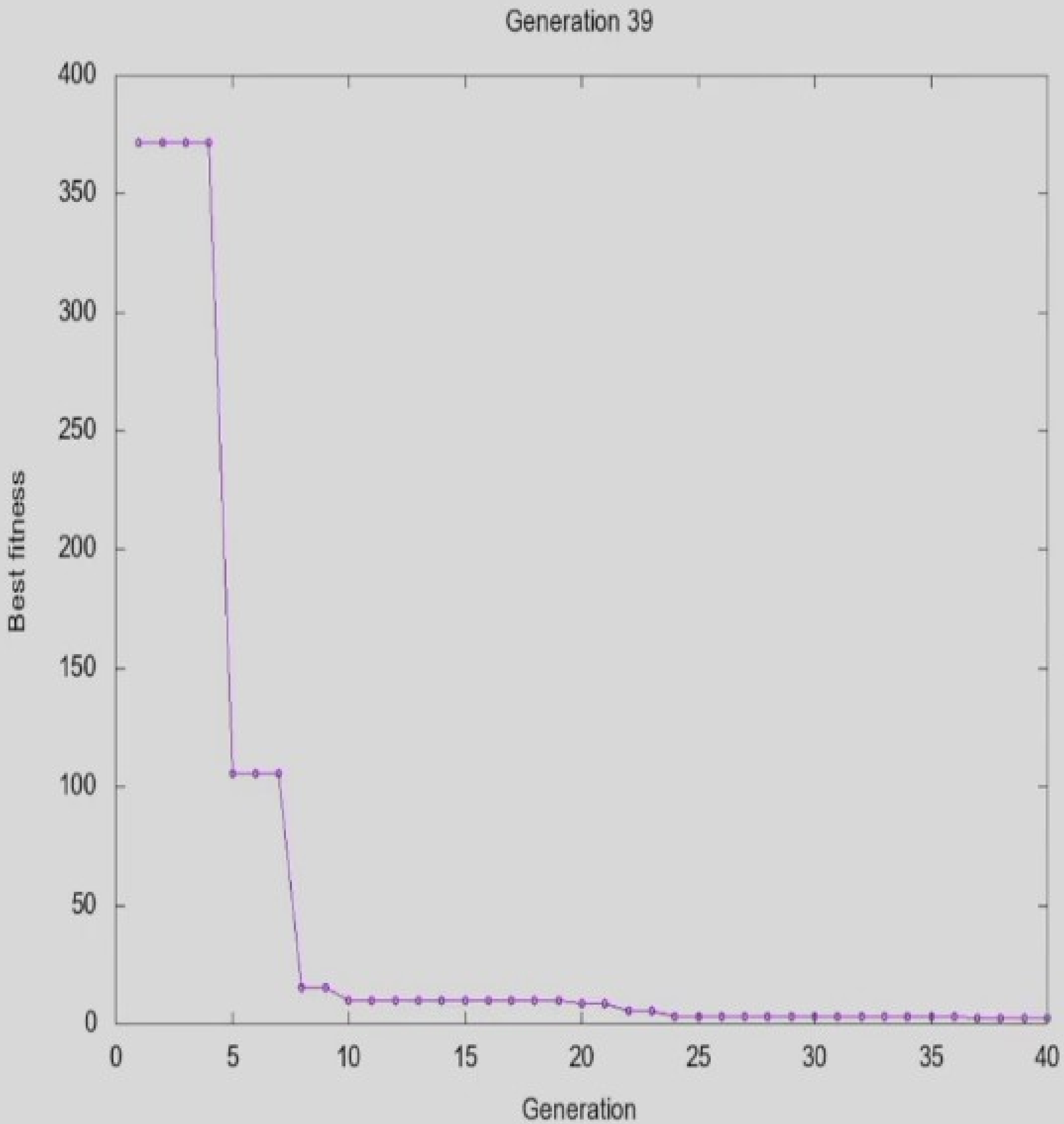
Rosenbrock Function

DE Parameters

- Number of variables: $n = 4$
- Population size: $N = 60$
- No. of generations: $T = 200$
- DE/rand/1/bin
- $F = 0.5$
- $p_c = 0.5$



- Progress Link

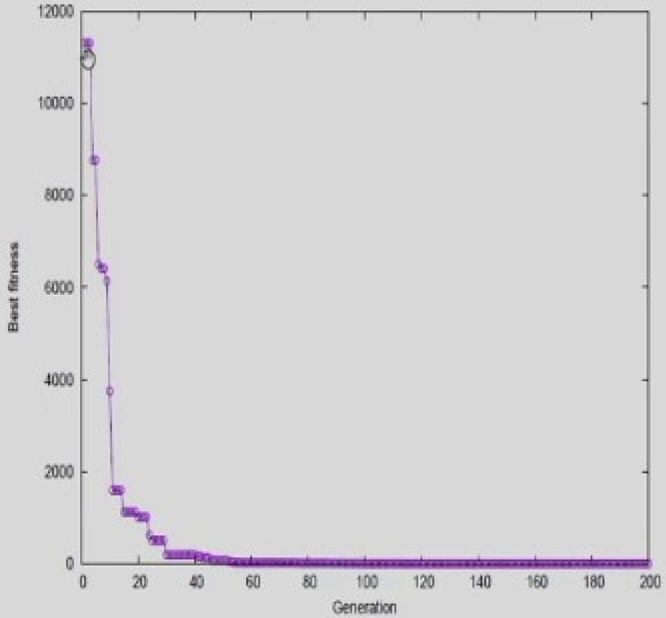


Generation 39
Best fitness
Generation
400
350
300
250
200
150
100
50
0
0
5
10
15
20
25
30
35
40

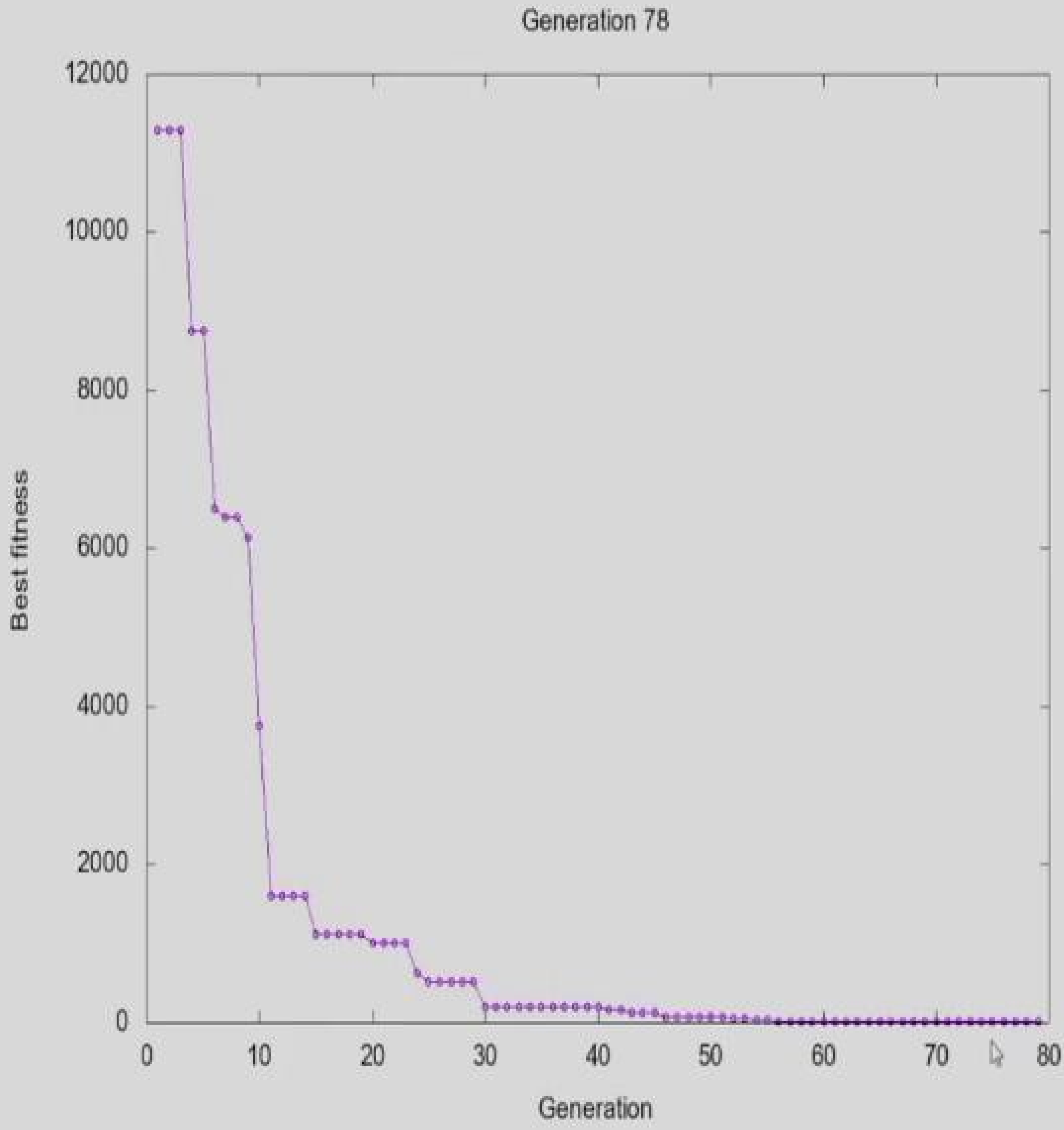
Rosenbrock Function

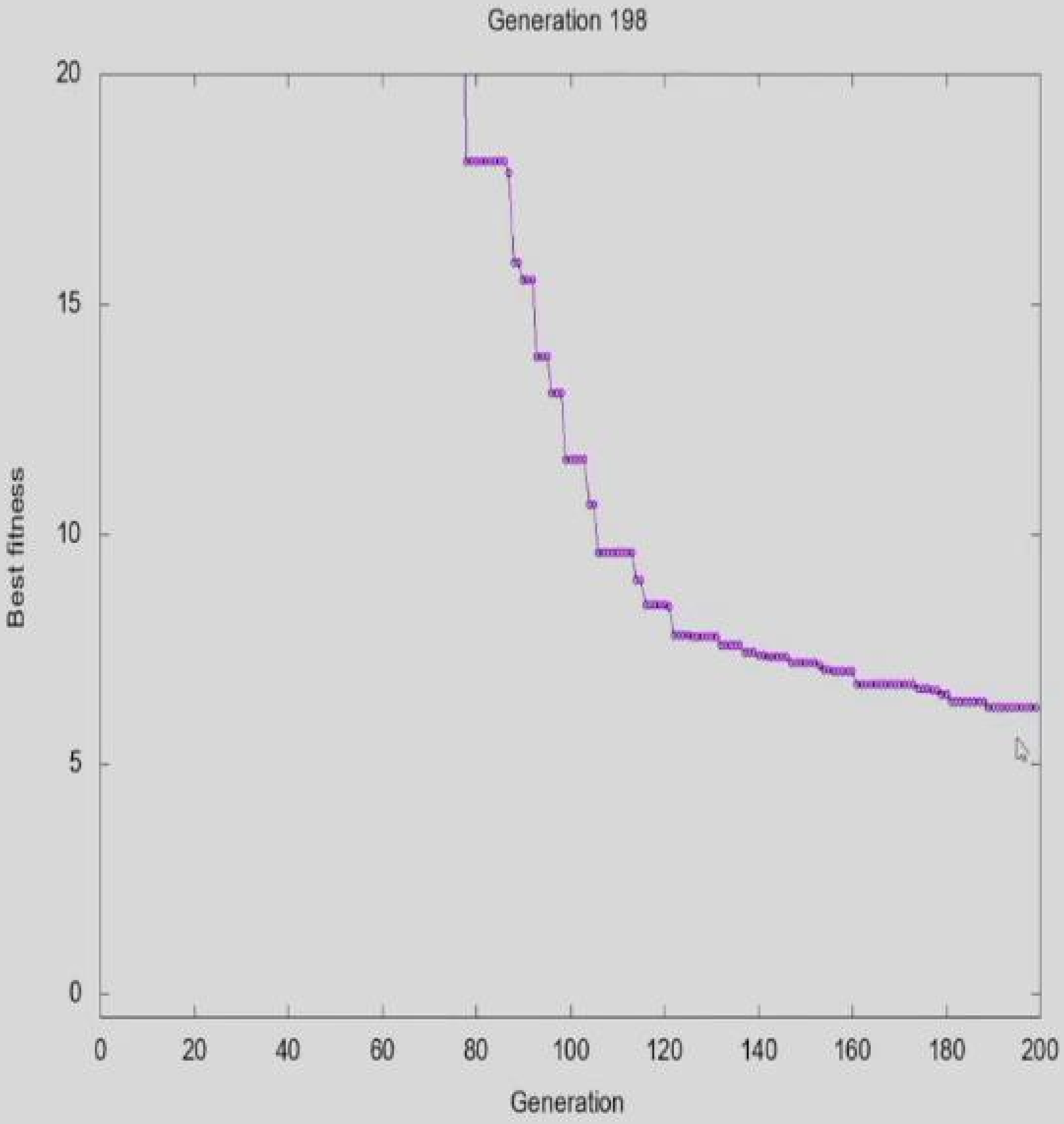
DE Parameters

- Number of variables: $n = 10$
- Population size: $N = 60$
- No. of generations: $T = 200$
- DE/rand/1/bin
- $F = 0.5$
- $p_c = 0.5$



- Progress Link


Generation 78
Best fitness
12000
10000
8000
6000
4000
2000
0
0
10
20
30
40
50
60
70
80
Generation

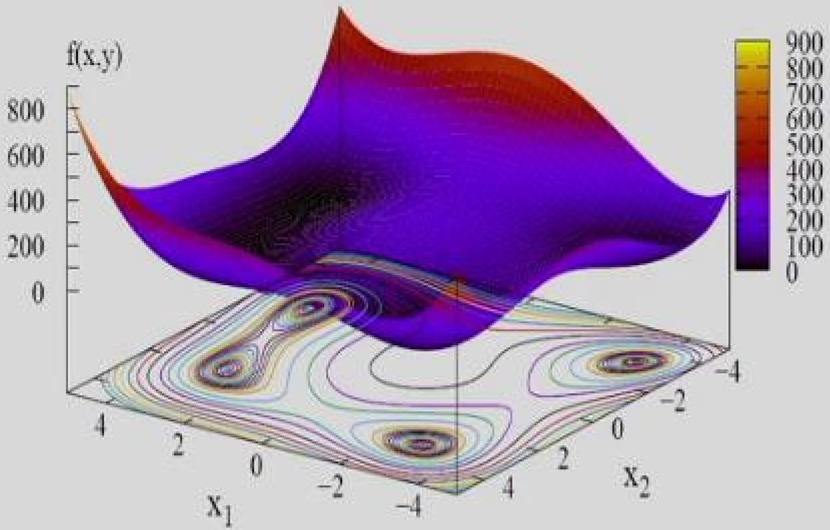

Generation 198
Best fitness
Generation
20
15
10
5
0
0
20
40
60
80
100
120
140
160
180
200

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 $-5 \leq x_1 \leq 5$ and $-5 \leq x_2 \leq 5$.

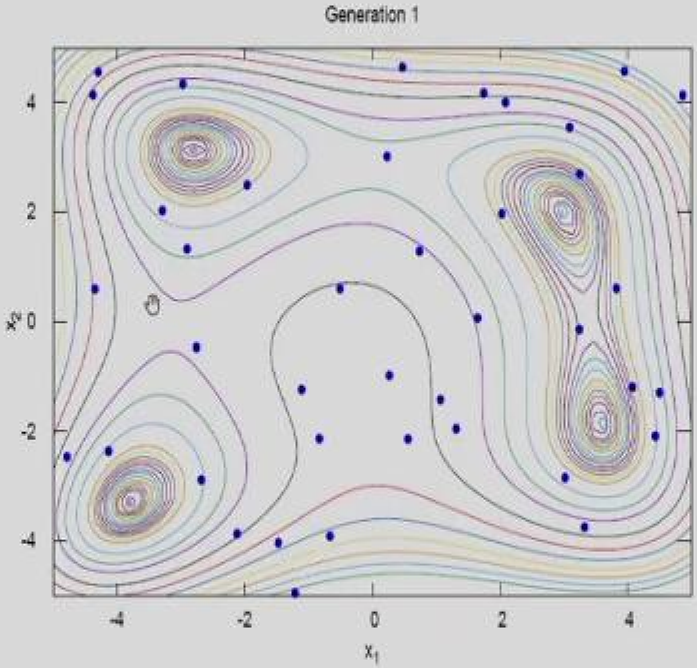


- Multi-modal function: it has 4 minimum points
- First optimal solution is $x^* = (3, 2)^T$ and $f(x) = 0$
- Second optimal solution is $x^* = (-2.805, 3.131)^T$ and $f(x) = 0$
- Third optimal solution is $x^* = (-3.779, -3.283)^T$ and $f(x) = 0$
- Fourth optimal solution is $x^* = (3.584, -1.848)^T$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

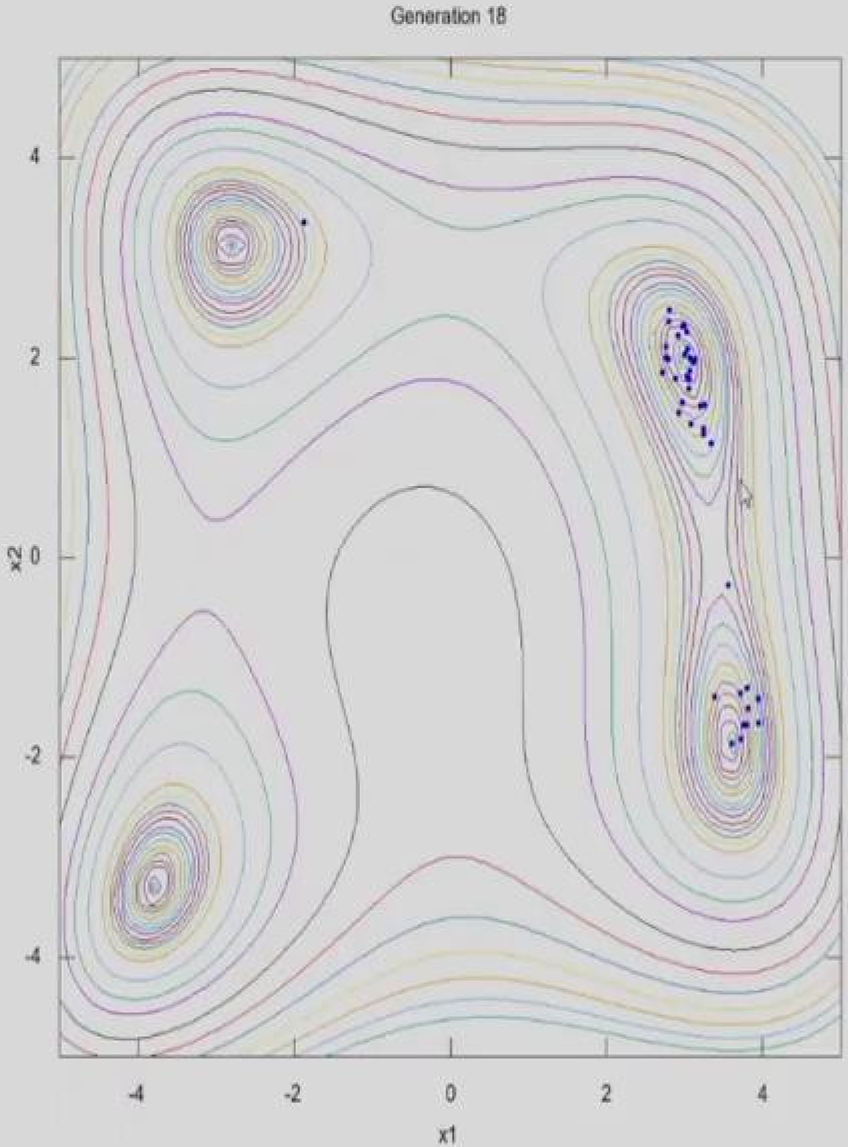
Himmelblau Function

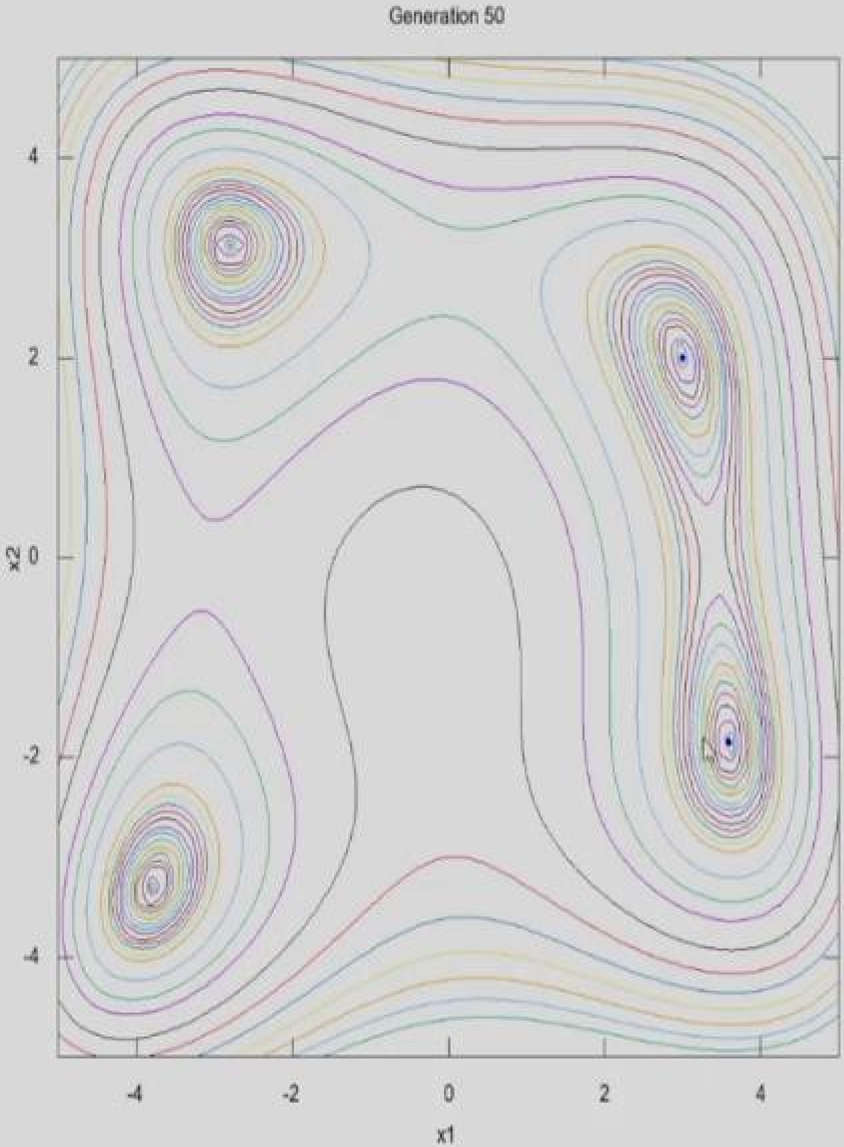
DE Parameters

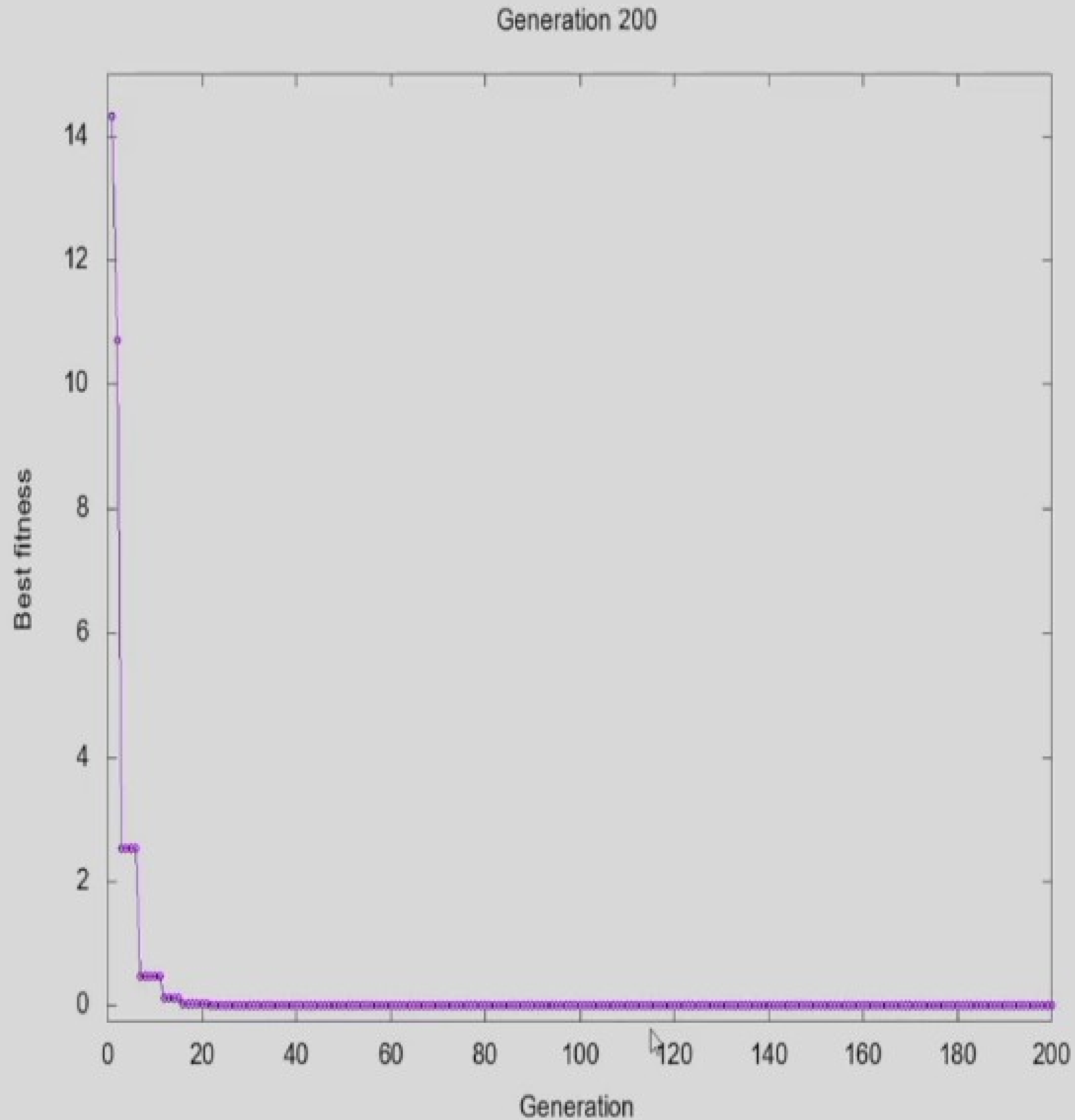
- Number of variables: $n = 2$
- Population size: $N = 40$
- No. of generations: $T = 200$
- DE/rand/1/bin
- $F = 0.5$
- $p_c = 0.5$



- Simulation Link
- Progress Link


Generation 18
4
2
x2 0
-2
-4
-4
-2
0
2
4
x1


Generation 50
x2
x1
4
2
0
-2
-4
-4
-2
0
2
4

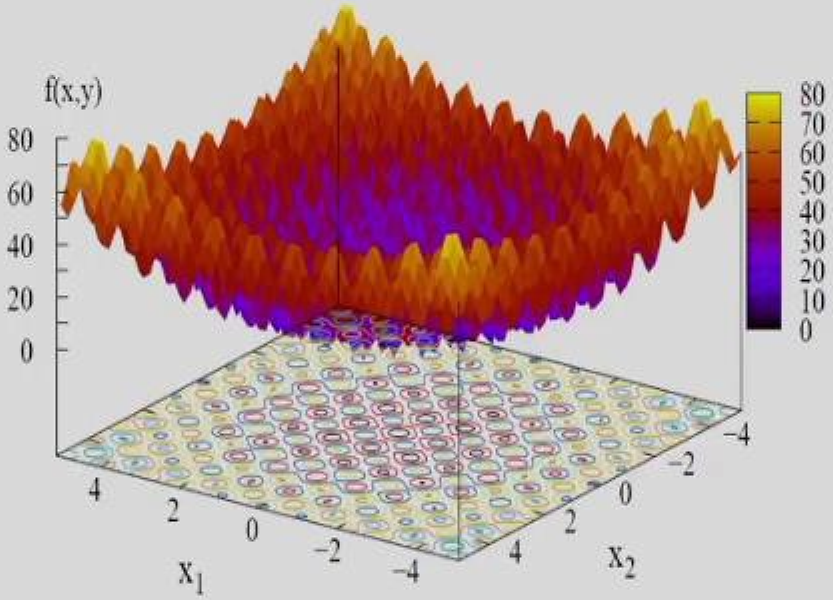

Generation 200
Best fitness
0
2
4
6
8
10
12
14
0
20
40
60
80
100
120
140
160
180
200
Generation

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 $-5.12 \leq x_i \leq 5.12$ and $i \in (1, \ldots, n).$

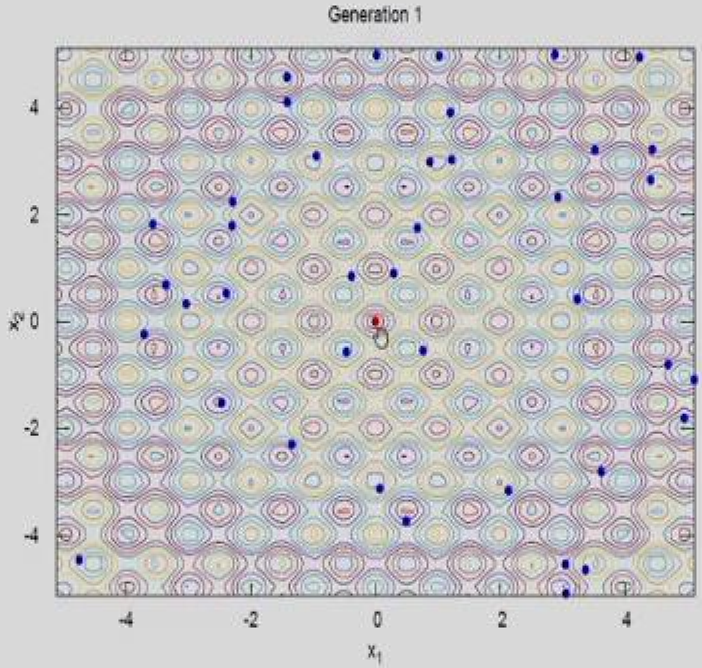


- Optimal solution is $x^* = (0, \ldots, 0)^T$ and $f(x) = 0$

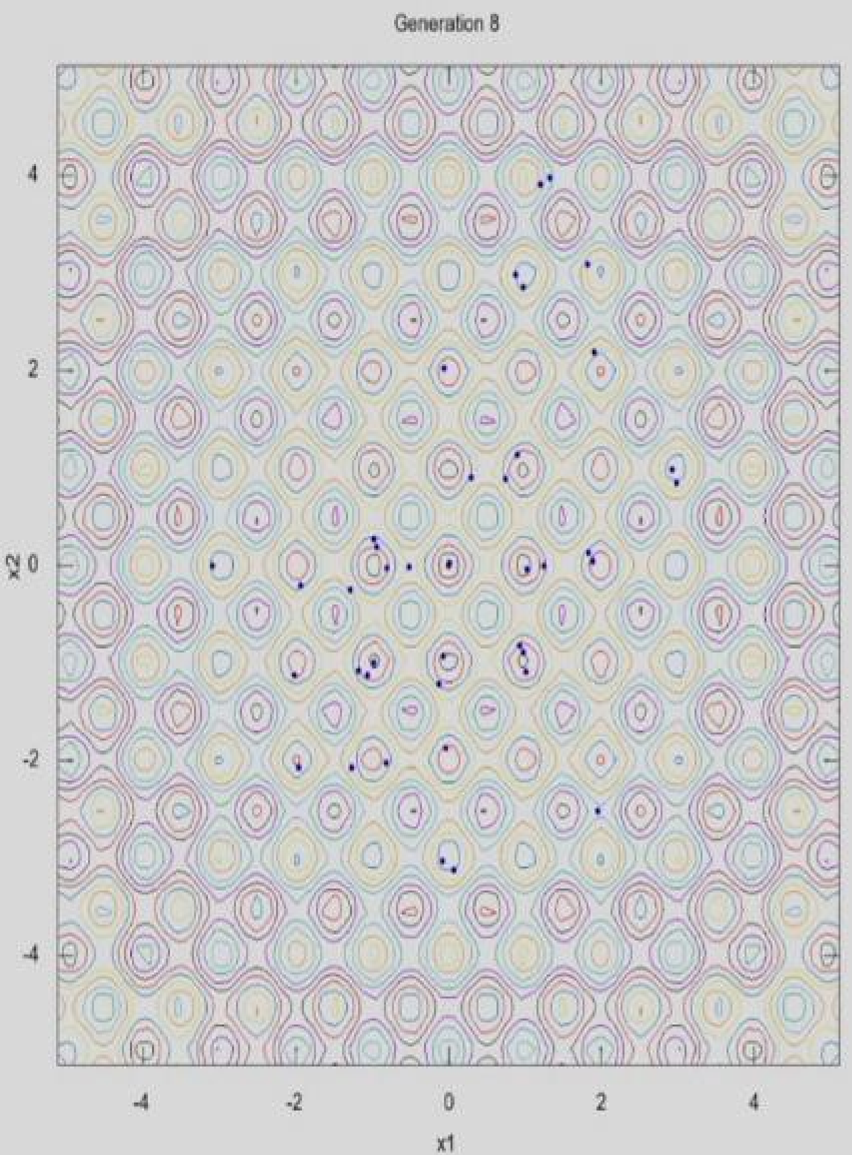
Rastrigin Function

DE Parameters

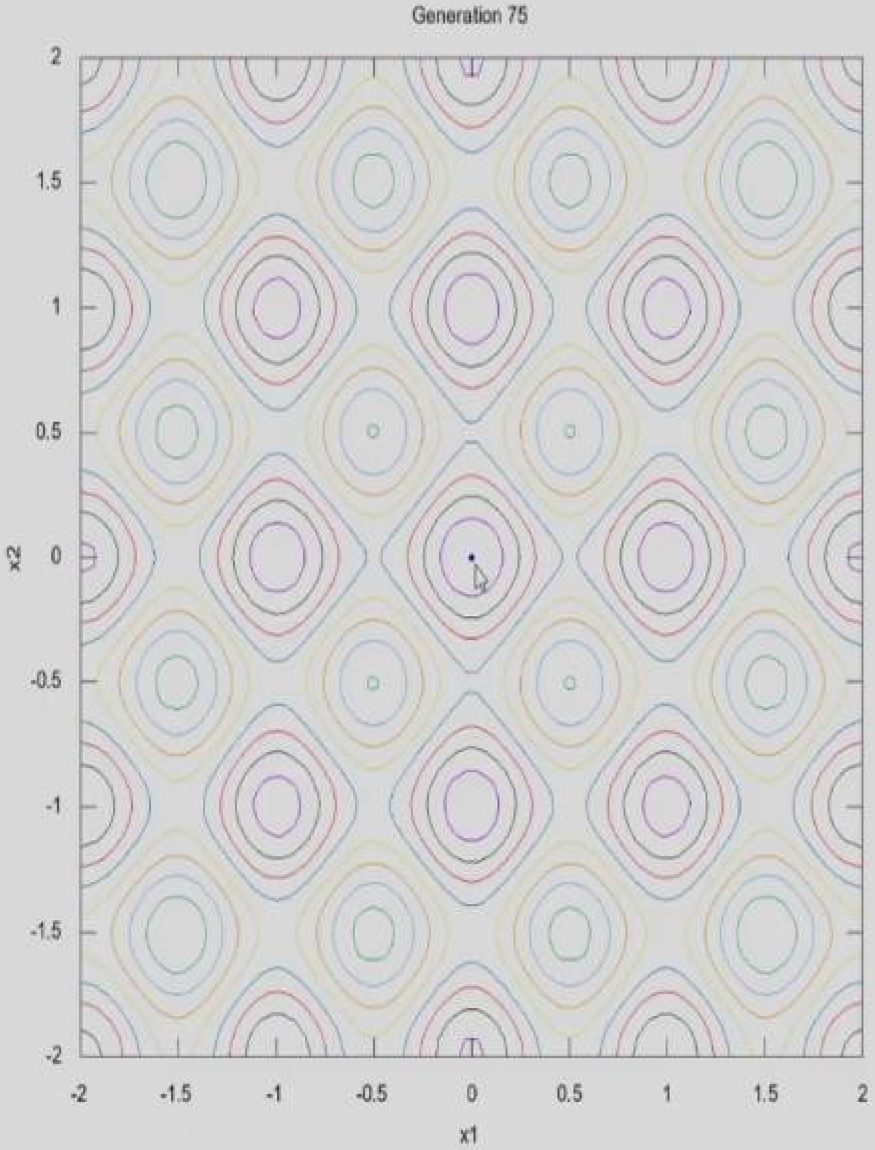
- Number of variables: $n = 2$
- Population size: $N = 40$
- No. of generations: $T = 200$
- DE/rand/1/bin
- $F = 0.5$
- $p_c = 0.5$



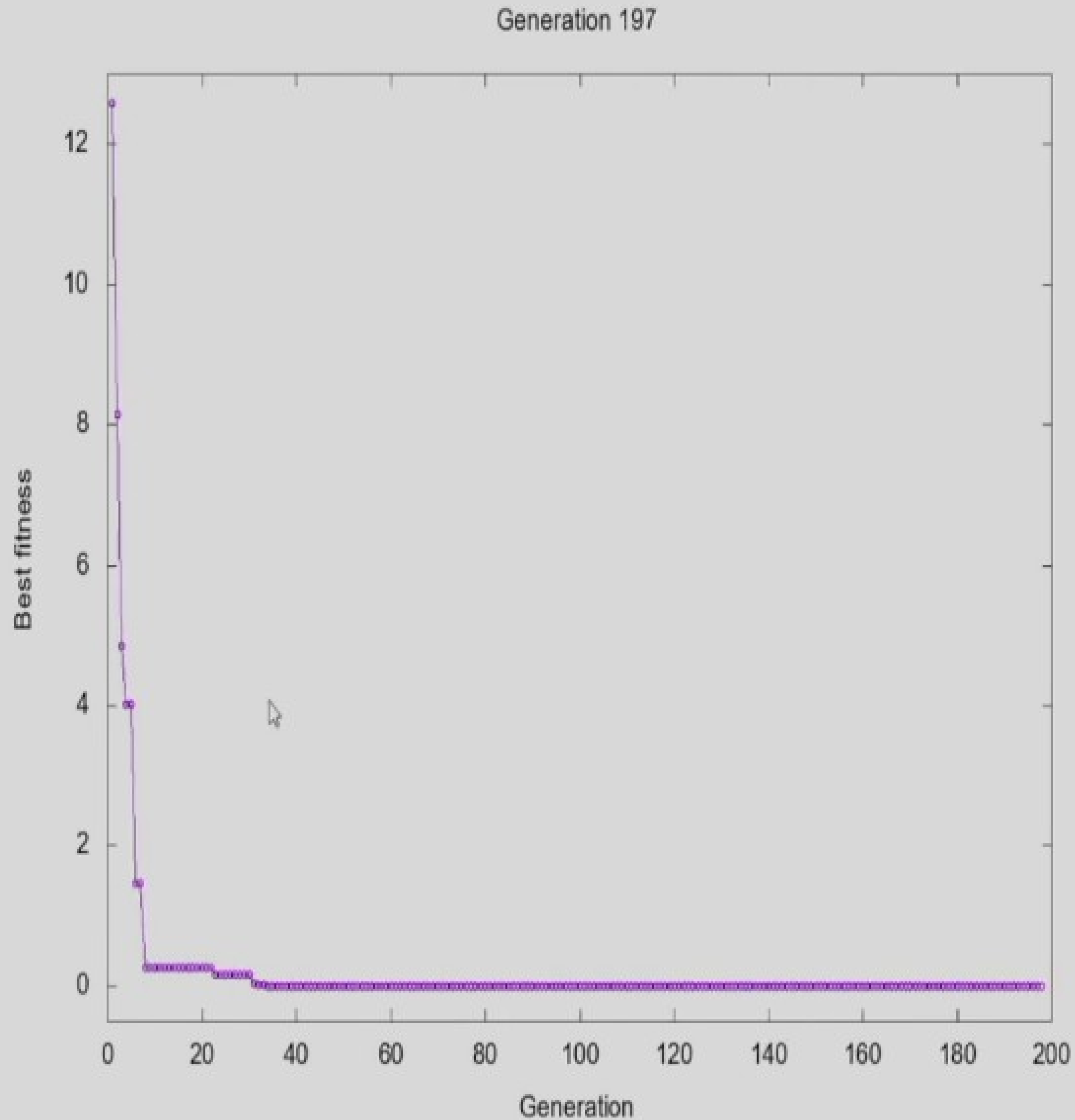
- Simulation ▸ Link
- Progress ▸ Link



Generation 8
4
2
0
-2
-4
x2
-4
-2
0
2
4
x1



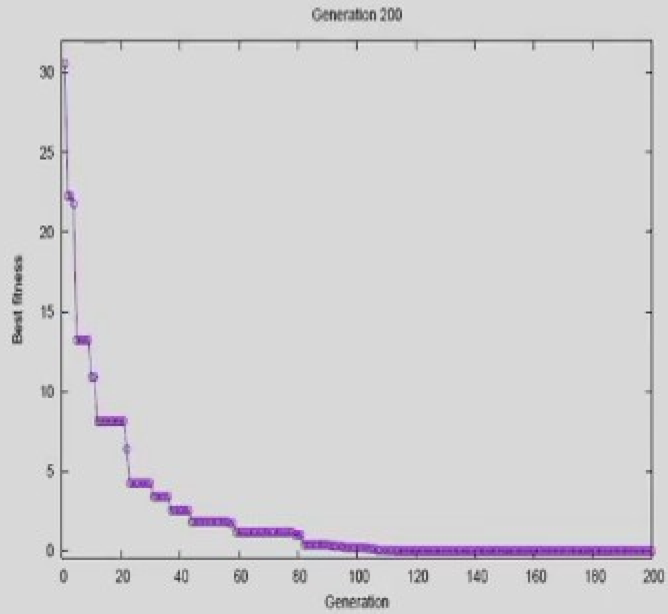
Generation 75
x1
x2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


Generation 197
Best fitness
12
10
8
6
4
2
0
0
20
40
60
80
100
120
140
160
180
200
Generation

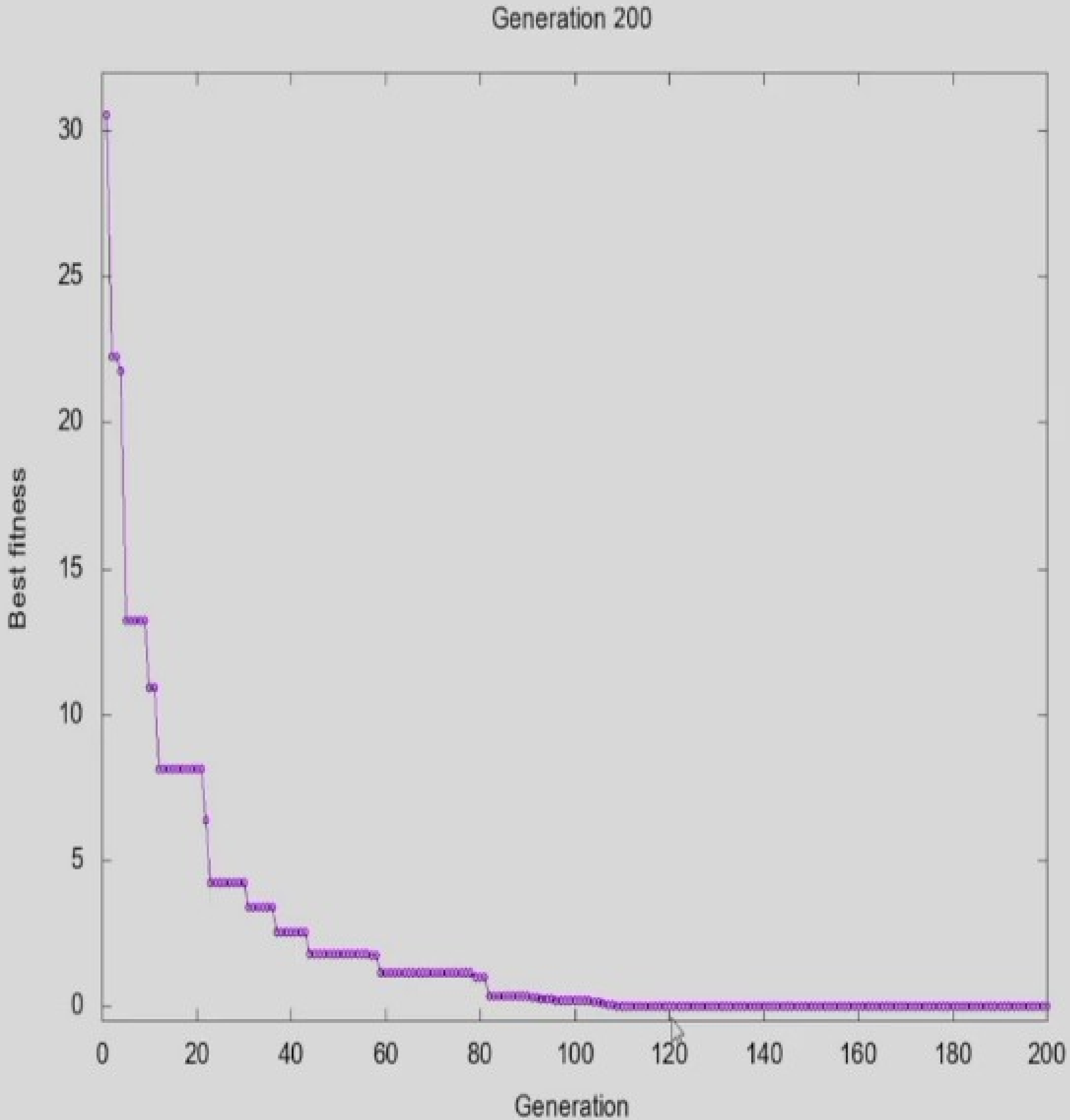
Rastrigin Function

DE Parameters

- Number of variables: $n = 4$
- Population size: $N = 60$
- No. of generations: $T = 200$
- DE/rand/1/bin
- $F = 0.5$
- $p_c = 0.5$



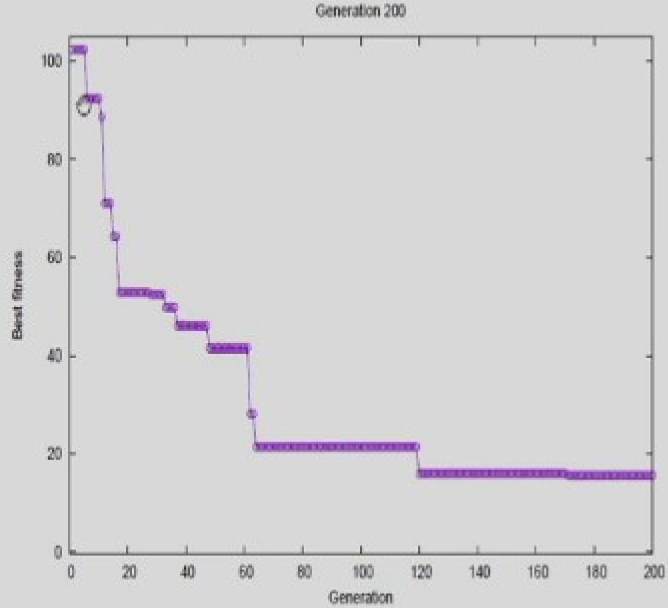
- Simulation Link


Generation 200
Best fitness
30
25
20
15
10
5
0
0
20
40
60
80
100
120
140
160
180
200
Generation

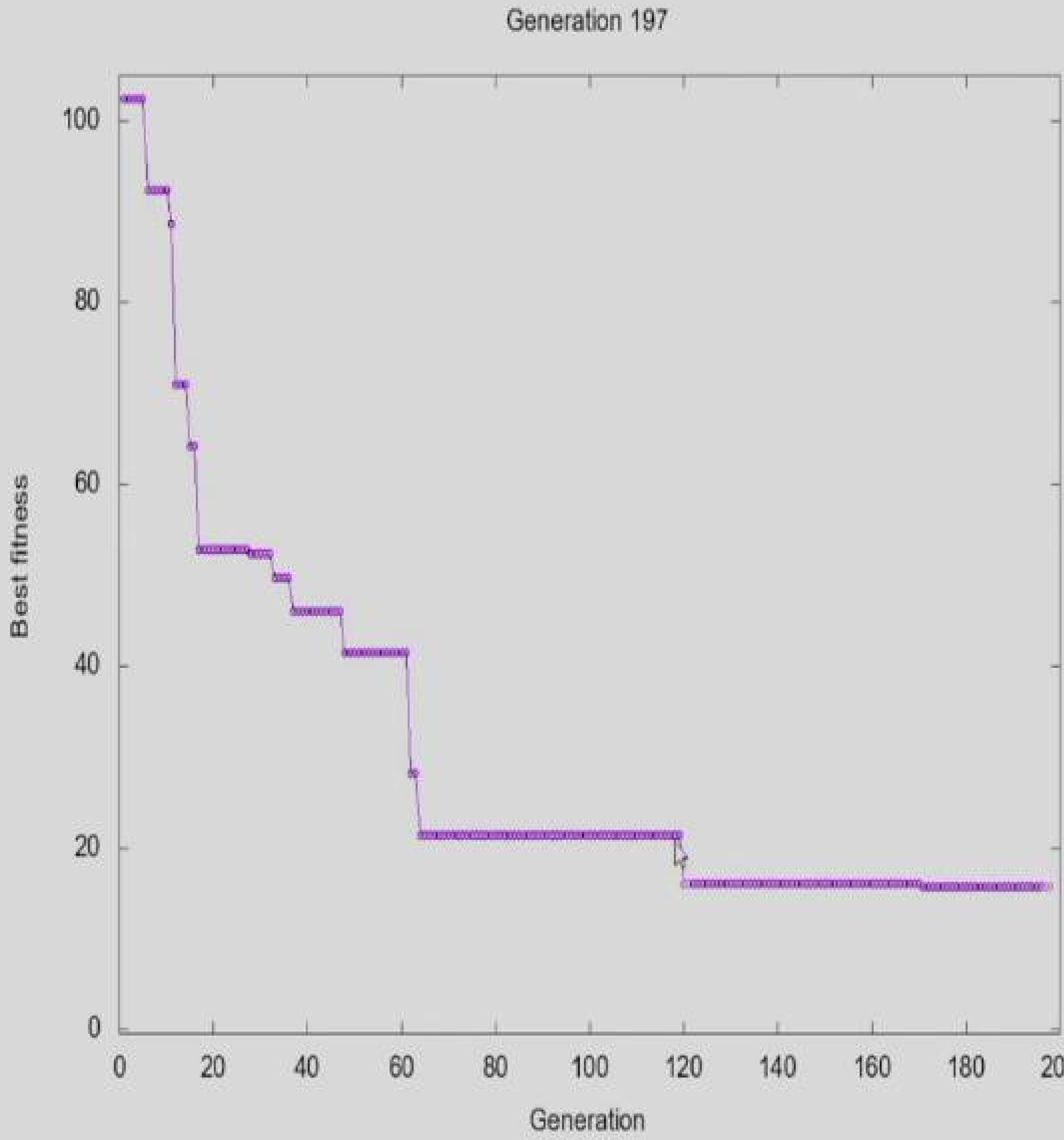
Rastrigin Function

DE Parameters

- Number of variables: $n = 10$
- Population size: $N = 60$
- No. of generations: $T = 200$
- DE/rand/1/bin
- $F = 0.5$
- $p_c = 0.5$



- Simulation ▸ Link



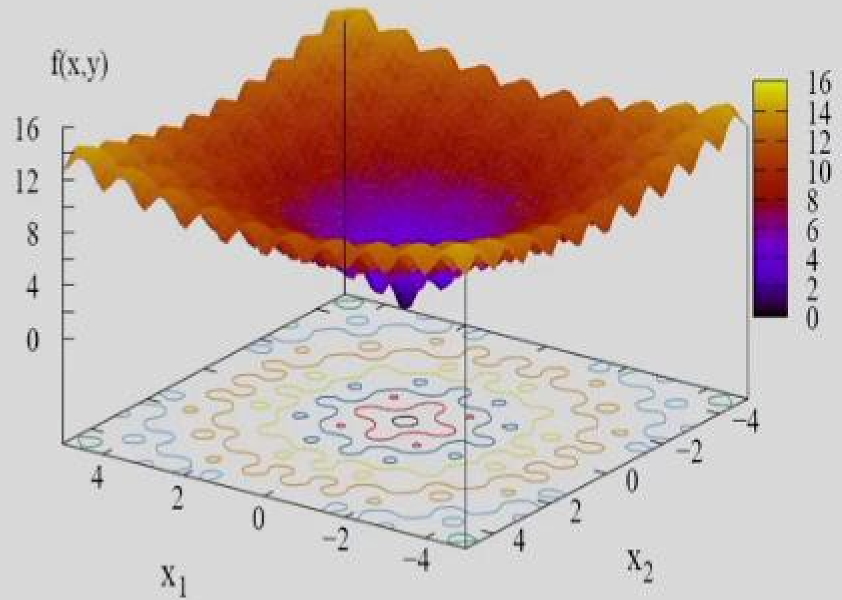
Generation 197
Best fitness
Generation
100
80
60
40
20
0
0
20
40
60
80
100
120
140
160
180
200

Ackley Function

Ackley Function

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

bounds $-5 \leq x_1 \leq 5$ and $-5 \leq x_2 \leq 5$.

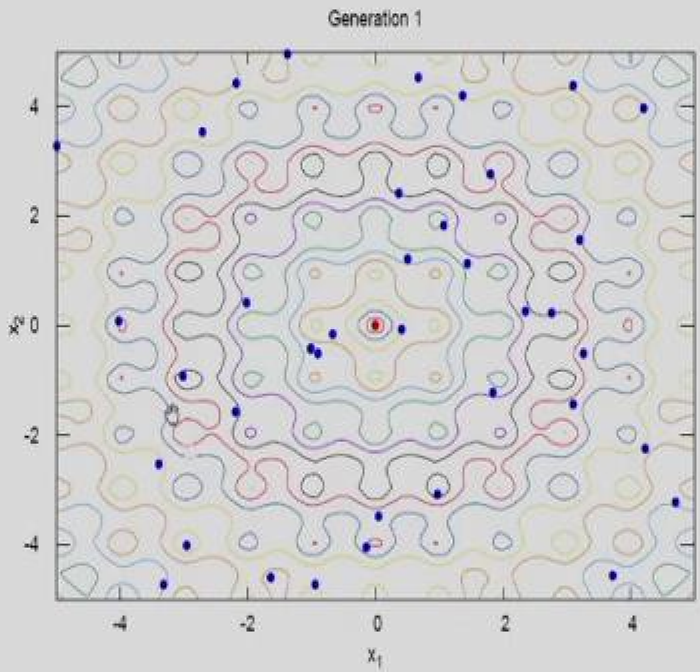


- Optimal solution is $x^* = (0, 0)^T$ and $f(x) = 0$

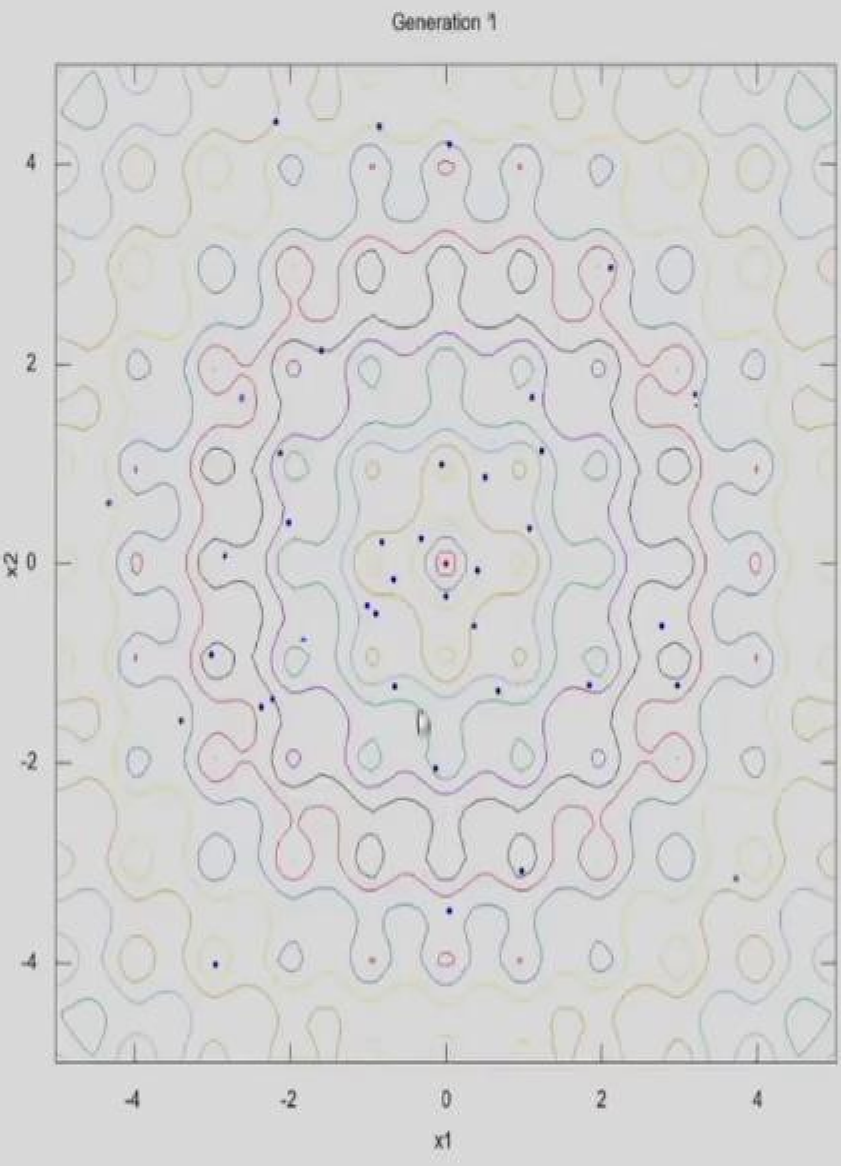
Ackley Function

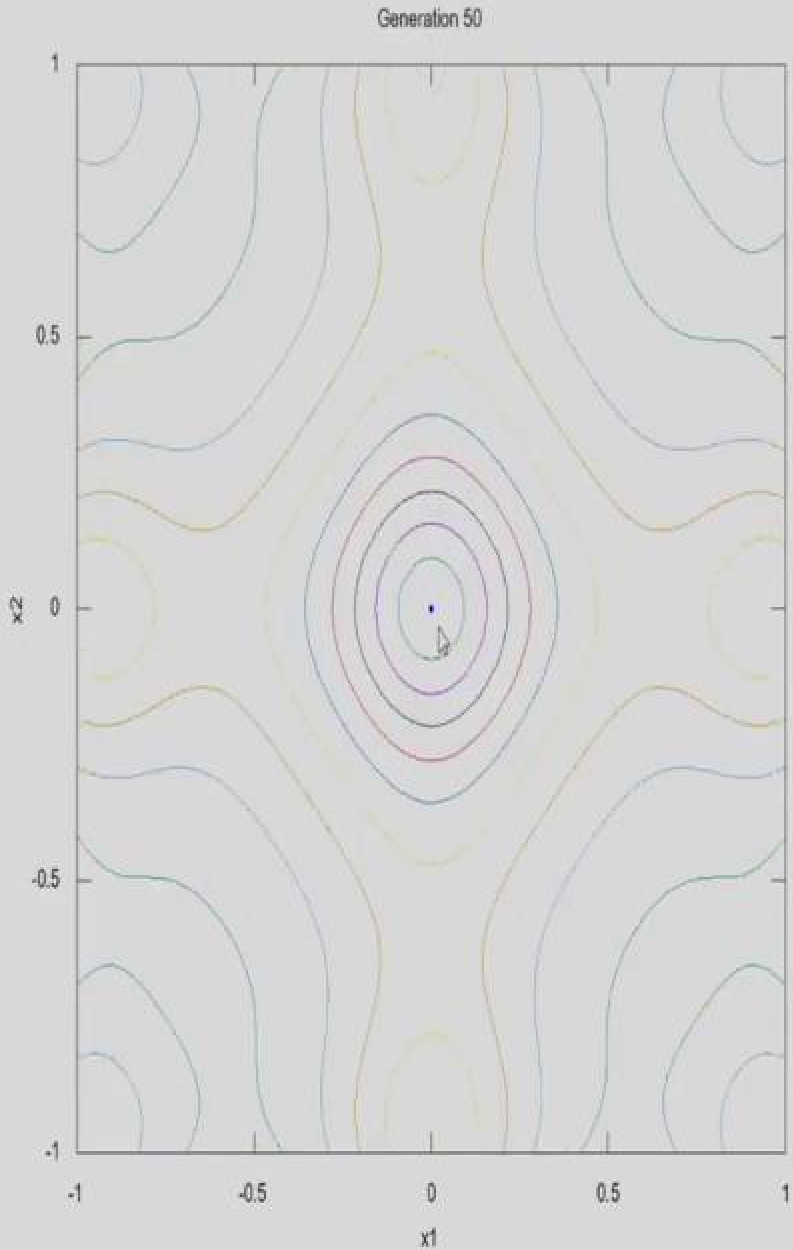
DE Parameters

- Number of variables: $n = 2$
- Population size: $N = 40$
- No. of generations: $T = 200$
- DE/rand/1/bin
- $F = 0.5$
- $p_c = 0.5$

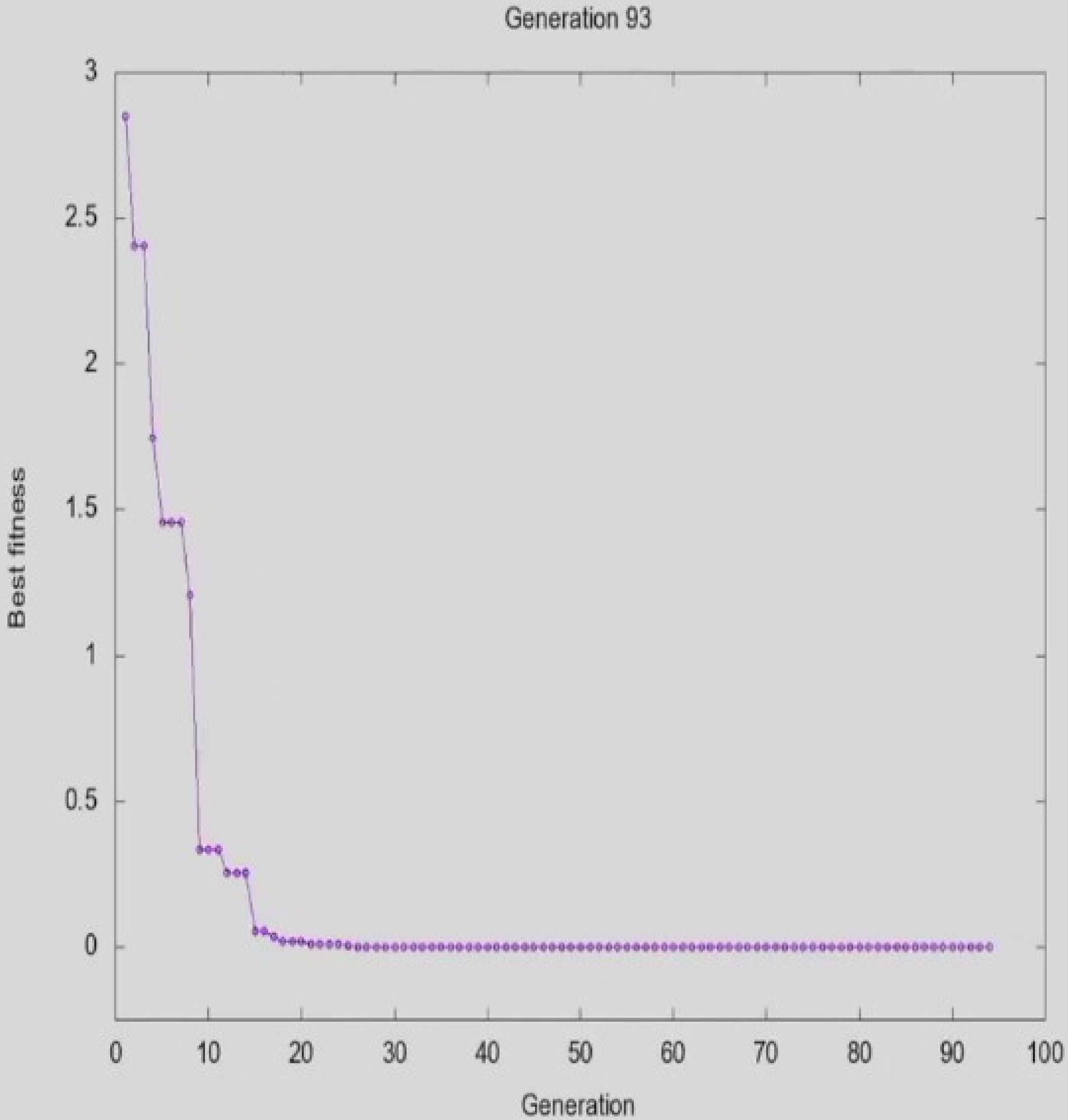


- Simulation Link
- Progress Link


Generation 1
x2
x1
4
2
0
-2
-4
-4
-2
0
2
4



Generation 50
1
0.5
0
-0.5
-1
x2
-1
-0.5
0
0.5
1
x1


Generation 93
Best fitness
Generation
3
2.5
2
1.5
1
0.5
0
0
10
20
30
40
50
60
70
80
90
100

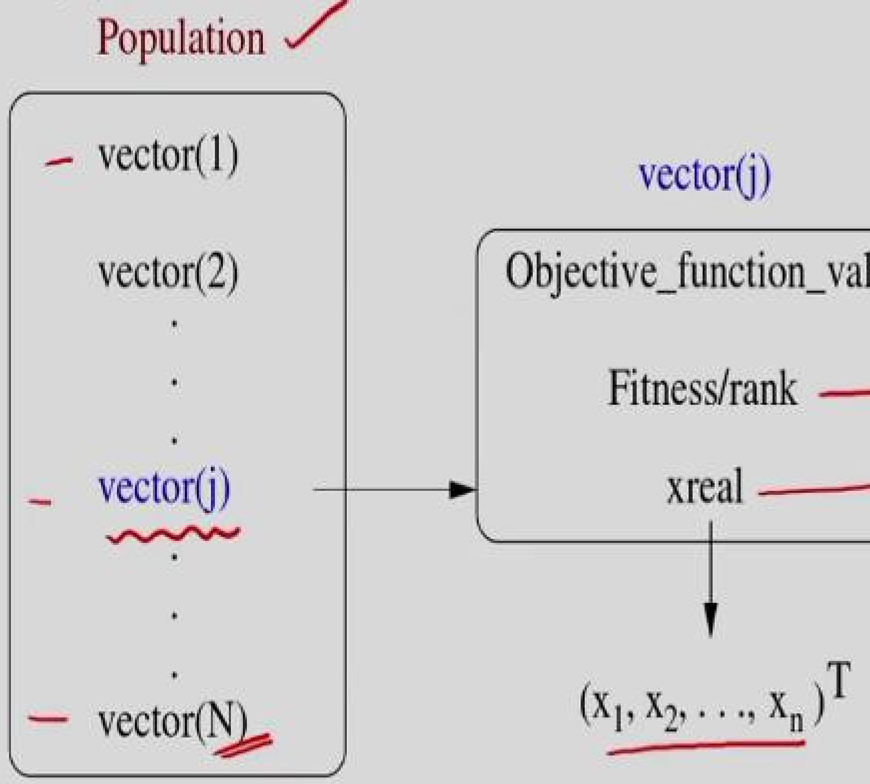
Generalized Framework of EC Techniques

Algorithm 1 Generalized Framework for DE

Solution representation — % Genetics
Input: $t := 1$ (Generation counter), Maximum allowed generation $= T$
Initialize random population $(P(t))$; — % Population
Evaluate $(P(t))$; — % Evaluate objective, constraints and assign fitness
while $t \leq T$ **do**
  for $(i = 1; i \leq N; i++)$ **do**
    Find the mutant vector $(v_i^{(t+1)})$ for target vector (i); — % Mutation
    Find the trail vector $(u_i^{(t+1)})$ for target vector (i); — % Crossover
    Evaluate $(u_i^{(t+1)})$;
    $x_i^{(t+1)} :=$ Survivor$(x_i^{(t)}, u_i^{(t+1)})$; — % Selection
  end for
  $t := t + 1$;
end while

Data Structure for DE

- Data Structure for population



- datatype **Population** target_vectors, mutant_vectors, trial_vectors;
  - target_vectors(j).objective_function_value;

Input to DE

Algorithm 2 Input

Population size: N
Number of generations: T
Number of real variables: n
for $(j = 1; j \leq n; j++)$ **do** %For each variable
    Lower and upper bounds on x_j that are $x_j^{(L)}$ and $x_j^{(U)}$
end for
Other parameters: F, p_c
Variant of DE

Initialize random population

Algorithm 3 Initialize random population

Input: N: population size, n: number of variables

for $(i = 1; i \leq N; i++)$ **do** %For each vector in the population

  for $(j = 1; j \leq n; j++)$ **do** %For each variable of a vector

    x_j =Generate real number randomly between $x_j^{(L)}$ and $x_j^{(U)}$

  end for

end for

Evaluate Particle

Algorithm 4 Evaluate Population

1: **Input**: vector(j)
2: Evaluate $f(x^{(j)})$ %Extract $x^{(j)} = (x_1, \ldots, x_n)^T$ from the data structure of a vector(j)

- Assign fitness same as the function value
- target_vectors(j).objective_function_value $= f(x_1, \ldots, x_n)$;
- target_vectors(j).fitness = target_vectors(j).objective_function_value;

Mutant Vector for each target vector(j)

Algorithm 5 Mutant Vector for each target vector(j)

1: **Input**: Three random vectors $(r1, r2, r3)$ from the population such that $(r_1 \neq r_2 \neq r_3 \neq j)$
2: Generate mutant vector (v_j) for the target vector (x_j) using
$v_j = x_{r1} + F \times (x_{r2} - x_{r3})$ %Vector operations

---

Trial Vector

Algorithm 6 Trial Vector(j)

```
Input: target vector (x_j), mutant vector (v_j), n: number of variables
for (i = 1; i ≤ n; i++) do                      %For each variable (i)
    if (if (rand_no ≤ p_c) or i = rnbr(j)) then
        u_{j_i} = v_{j_i}
    end if
    if (if (rand_no > p_c) and i ≠ rnbr(j)) then
        u_{j_i} = x_{j_i}
    end if
end for
```

u_{j_i}

j-th vector

i-th variable

Greedy Selection of DE

Algorithm 7 Trial Vector(j)

Input: target vector (x_j), trial vector (u_j)
if ($f(u_j) < f(x_j)$) **then** %Comparison of fitness
    $x_j = u_j$ %Adding trial vector
end if

Copy Vector

Algorithm 8 Copy Vectors

1: **Input**: vector 1, vector 2
2: Copy objective function value of vector 1 to vector 2
3: Copy fitness/rank of vector 1 to vector 2
4: Copy x_j of vector 1 to x_j of vector 2

- Copy the complete data structure

Closure

- Simulations
  - Rosenbrock function with $n = 2, 4, 10$ variables
  - Rastrigin function with $n = 2, 4, 10$ variables
  - Himmelblau multi-modal function
  - Ackley function

- Algorithmic Implementation of DE
  - Data structure for DE
  - Input to DE
  - Random initial population
  - Fitness evaluation
  - Mutant vector
  - Trial vector
  - Greedy selection of canonical DE
  - copy ~~particle~~ vector