Outline

1. **Simulations**
   - Rosenbrock Function
   - Himmelblau Function
   - Rastrigin Function
   - Ackley Function
2. **Effect of PSO Parameters**
   - Effect w, c_1 and c_2 values
   - Issues
3. **Algorithmic Implementation of PSO**
   - Data structure
   - Input
   - Random initial swarm
   - Fitness assignment
   - Local and global best updates
   - Velocity and position updates
4. **Closure**

Recap

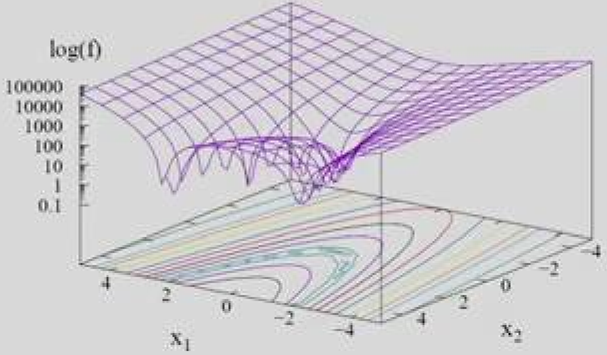
- Introduction of PSO
- Distinct feature of PSO
  - global best, local best, velocity and position updates
- Velocity update
  - Velocity components: Momentum, cognitive and social parts
  - Graphical illustration
- Position update
- Flowchart of PSO
- PSO on the generalized framework
- Working principles of PSO through Rosenbrock function
- Graphical example

Simulations

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 $-5 \le x_i \le 5$ and $i = 1, \ldots, n$.

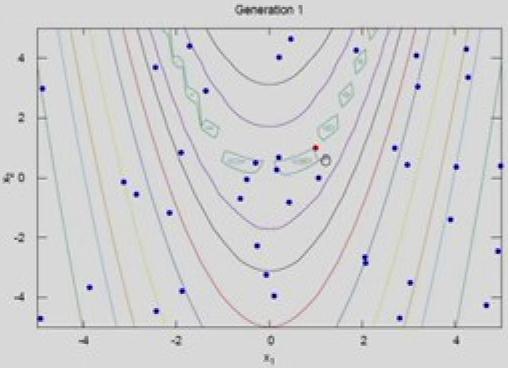


- Optimal solution is $x^* = (1, \ldots, 1)^T$ and $f(x) = 0$

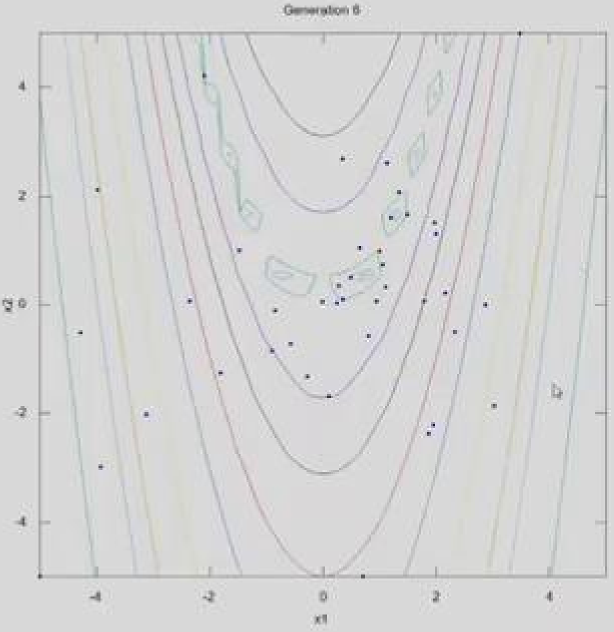
Rosenbrock Function

PSO Parameters

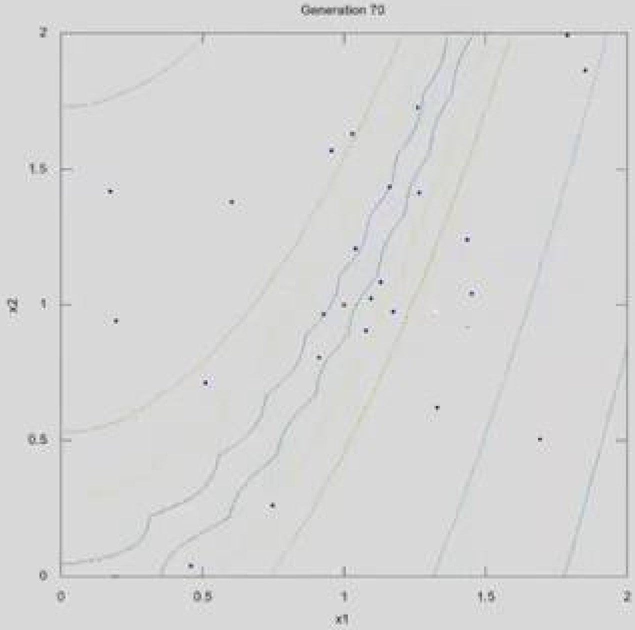
- Number of variables: $n = 2$
- Swarm size: $N = 40$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = c_2 = 1.5$
- Inertia equation $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$

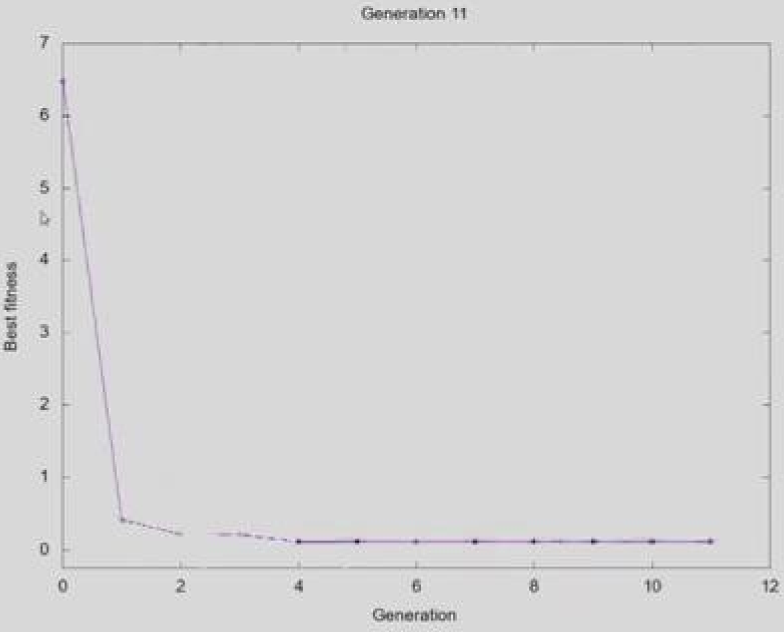


- Simulation Link
- Progress Link



Generation 6
x1
x2
4
2
0
-2
-4
-4
-2
0
2
4

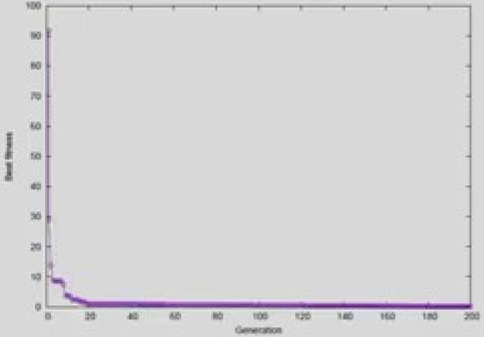

Generation 70
x1
x2
0
0.5
1
1.5
2


Generation 11
Best fitness
Generation
0
1
2
3
4
5
6
7
0
2
4
6
8
10
12

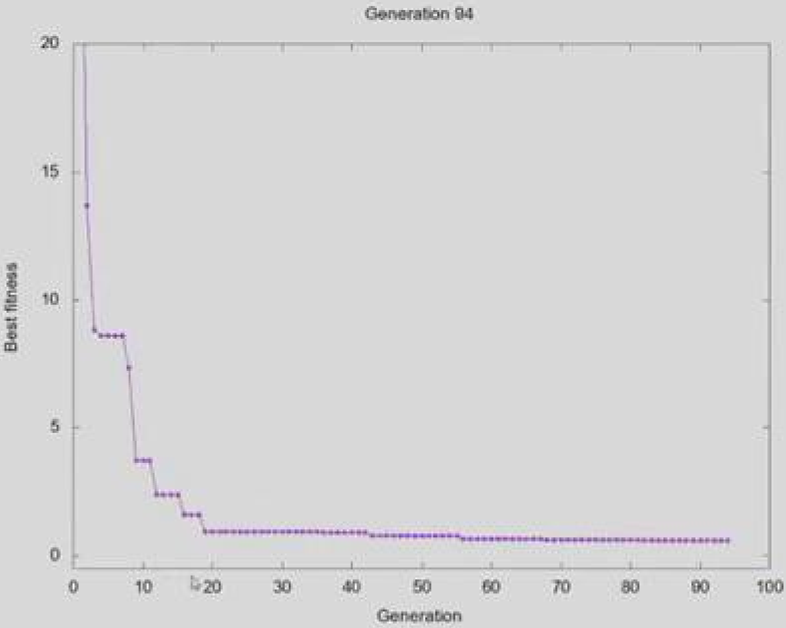
Rosenbrock Function

PSO Parameters

- Number of variables: $n = 4$
- Swarm size: $N = 60$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = c_2 = 1.5$
- Inertia equation $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$



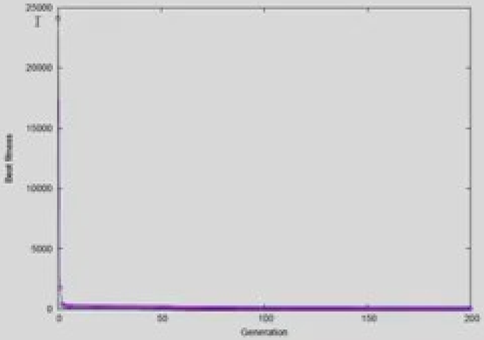
- Progress ▸ Link


Generation 94
Best fitness
Generation
20
15
10
5
0
0
10
20
30
40
50
60
70
80
90
100

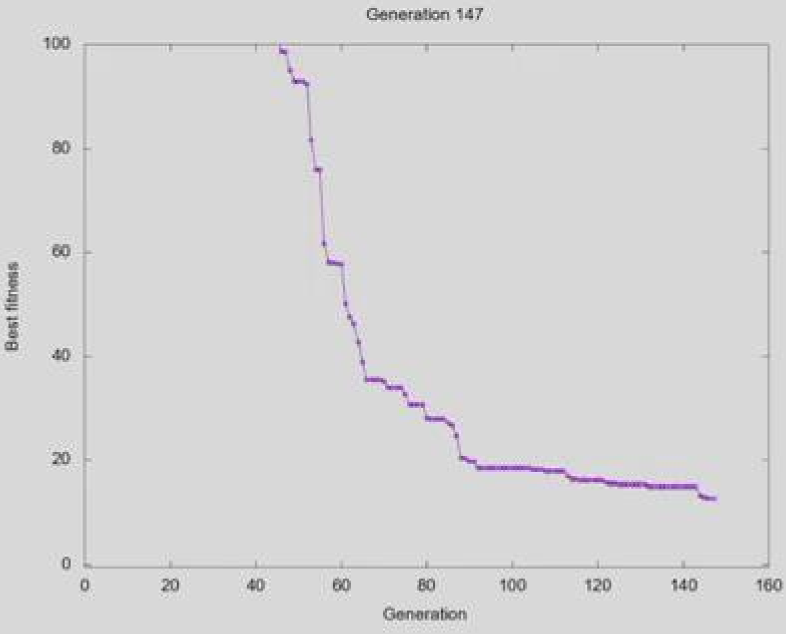
Rosenbrock Function

PSO Parameters

- Number of variables: $n = 10$
- Swarm size: $N = 60$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = c_2 = 1.5$
- Inertia equation $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$



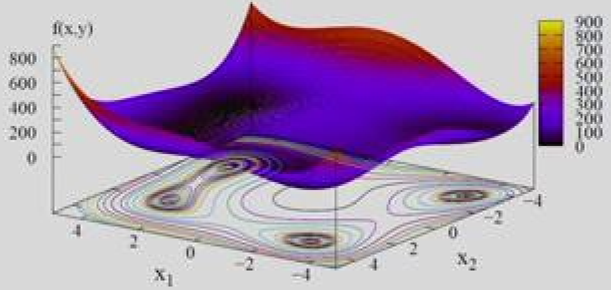
- Progress ▸ Link


Generation 147
Best fitness
Generation
100
80
60
40
20
0
0
20
40
60
80
100
120
140
160

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 $-5 \leq x_2 \leq 5$.

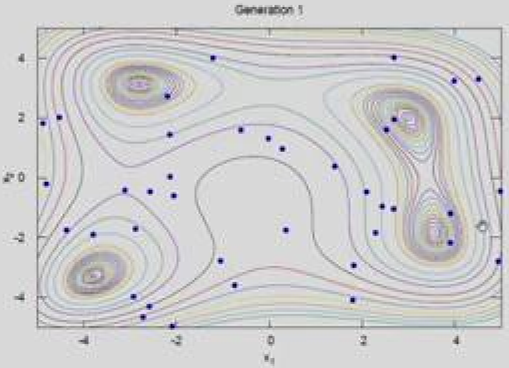


- Multi-modal function: it has 4 minimum points
- First optimal solution is $x^* = (3, 2)^T$ and $f(x) = 0$
- Second optimal solution is $x^* = (-2.805, 3.131)^T$ and $f(x) = 0$
- Third optimal solution is $x^* = (-3.779, -3.283)^T$ and $f(x) = 0$
- Fourth optimal solution is $x^* = (3.584, -1.848)^T$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

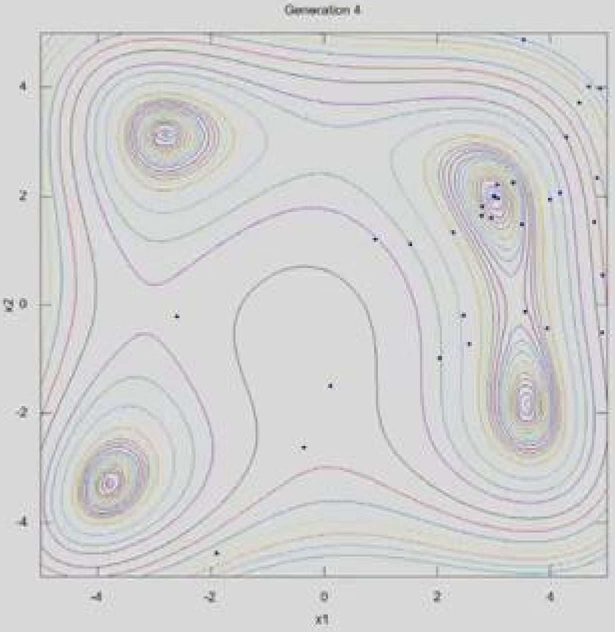
Himmelblau Function

PSO Parameters

- Number of variables: $n = 2$
- Swarm size: $N = 40$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = c_2 = 1.5$
- Inertia equation $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$



- Simulation Link
- Progress Link

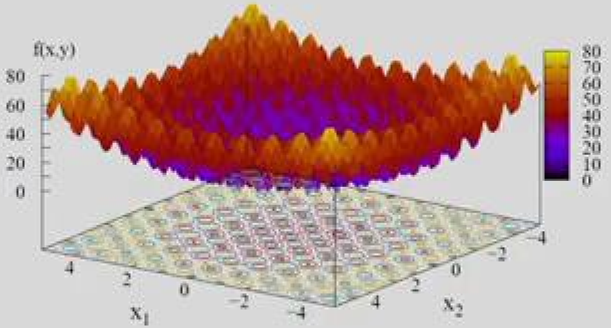

Generation 4
x1
x2
4
2
0
-2
-4

Rastrigin Function

Rastrigin Function

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

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

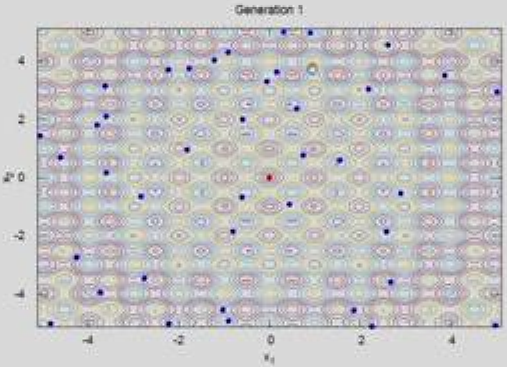


- Optimal solution is $x^* = (0, \ldots, 0)^T$ and $f(x) = 0$

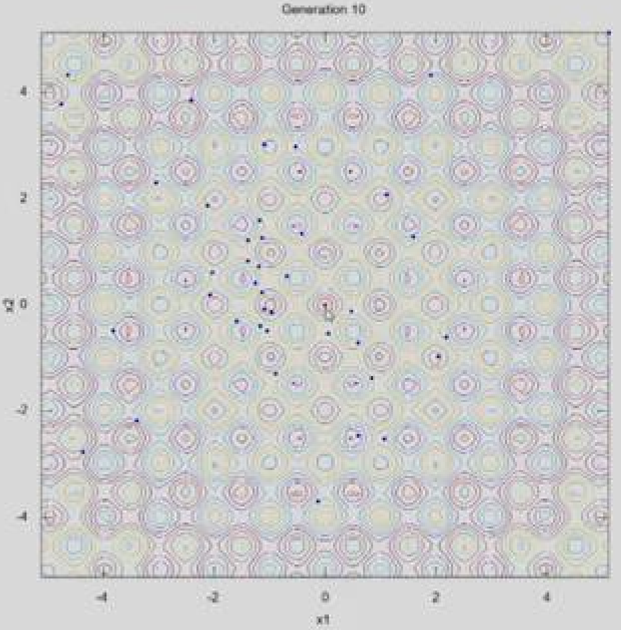
Rastrigin Function

PSO Parameters

- Number of variables: $n = 2$
- Swarm size: $N = 40$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = c_2 = 1.5$
- Inertia equation
  $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where
  $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$



- Simulation Link
- Progress Link

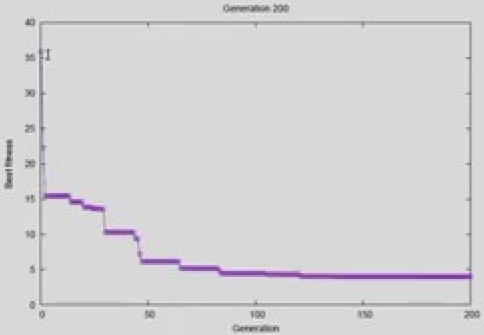


Generation 10
4
2
x2 0
-2
-4
-4
-2
0
2
4
x1

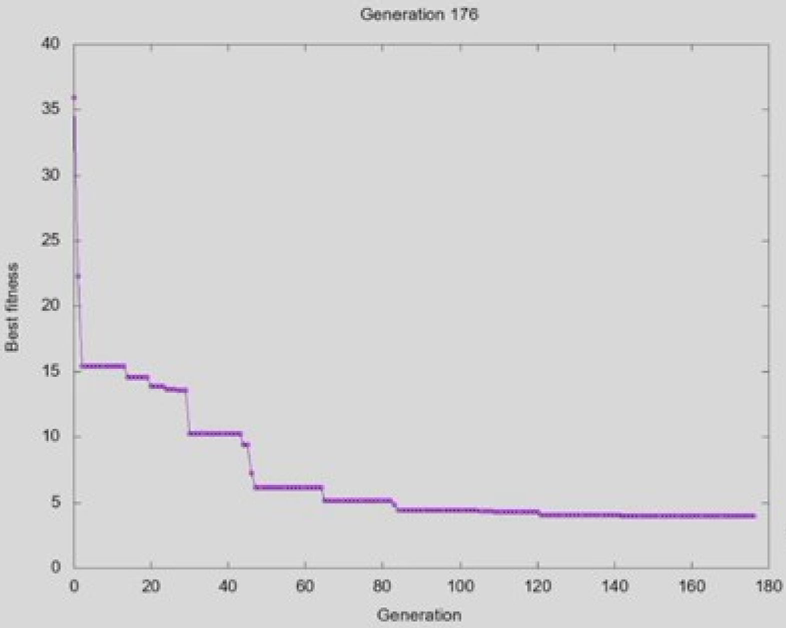
Rastrigin Function

PSO Parameters

- Number of variables: $n = 4$
- Swarm size: $N = 60$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = c_2 = 1.5$
- Inertia equation
  $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where
  $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$



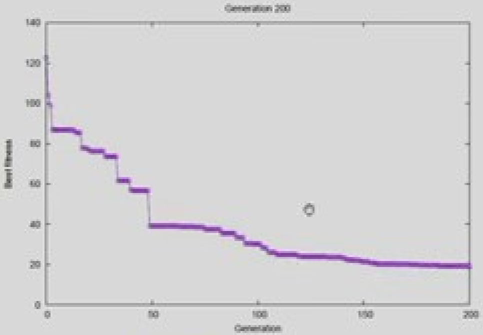
- Simulation Link


Generation 176
Best fitness
Generation
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100
120
140
160
180

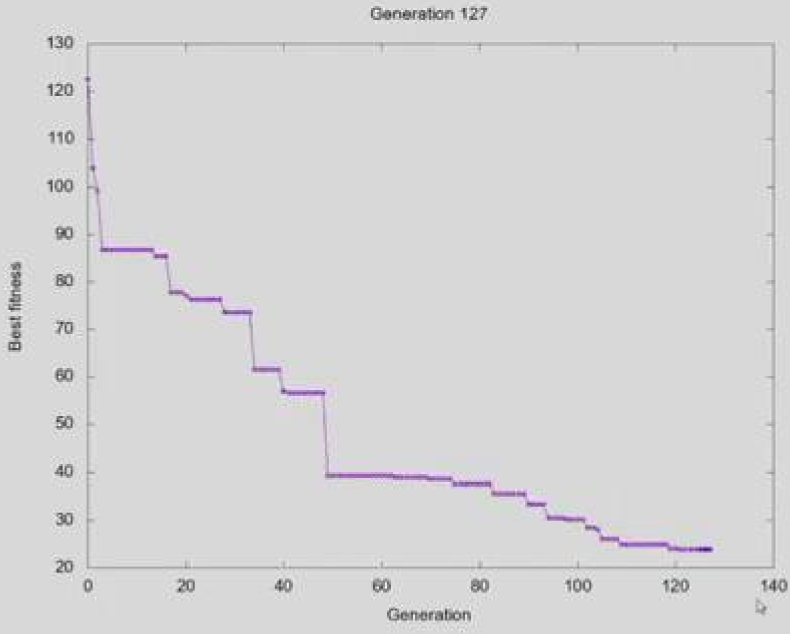
Rastrigin Function

PSO Parameters

- Number of variables: $n = 10$
- Swarm size: $N = 60$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = c_2 = 1.5$
- Inertia equation
  $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$



- Simulation ▸ Link

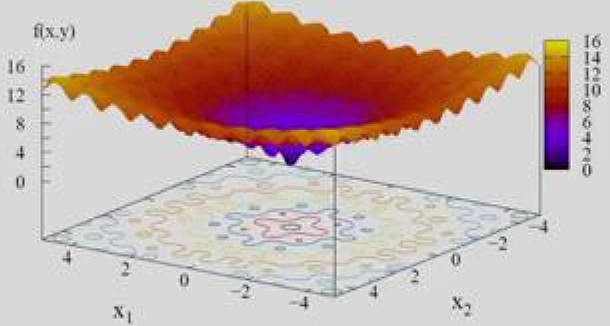

Generation 127
Best fitness
Generation
130
120
110
100
90
80
70
60
50
40
30
20
0
20
40
60
80
100
120
140

Ackley Function

Ackley Function

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

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

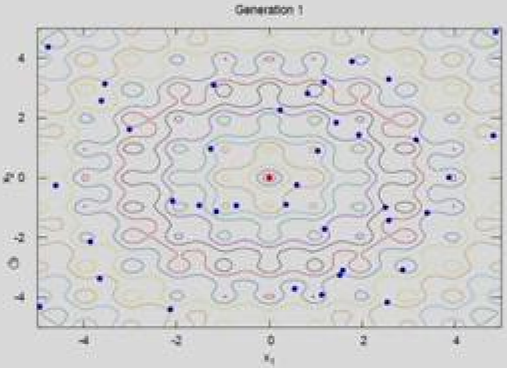


- Optimal solution is $x^* = (0, 0)^T$ and $f(x) = 0$

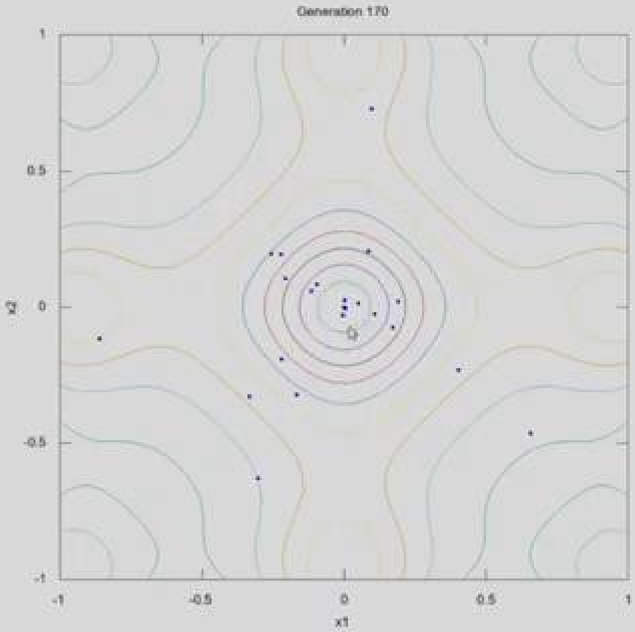
Ackley Function

PSO Parameters

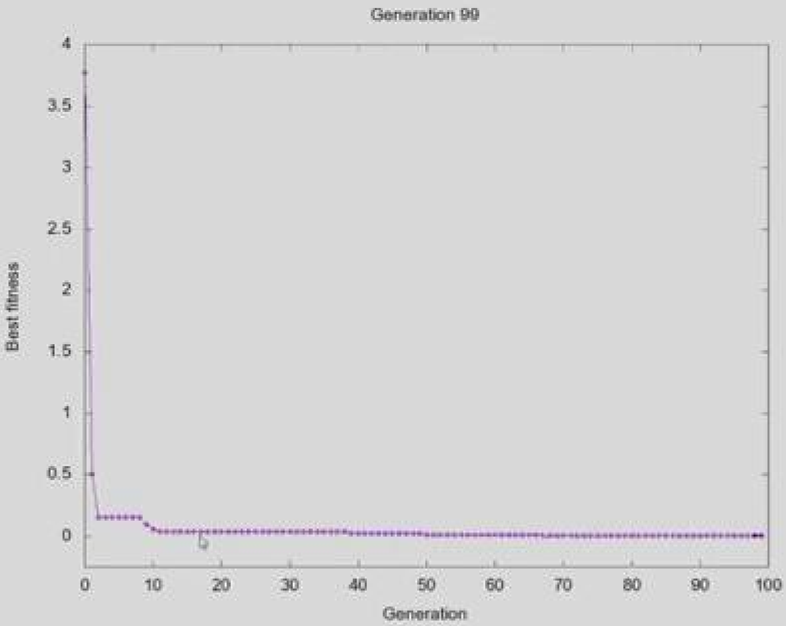
- Number of variables: $n = 2$
- Swarm size: $N = 40$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = c_2 = 1.5$
- Inertia equation $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$



- Simulation ▸ Link
- Progress ▸ Link



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



Generation 99
Best fitness
Generation
4
3.5
3
2.5
2
1.5
1
0.5
0
0
10
20
30
40
50
60
70
80
90
100

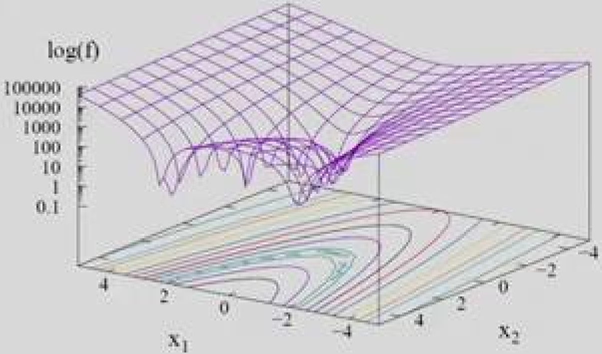
Effect of PSO Parameters

Rosenbrock Function

Rosenbrock Function

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

bounds $\quad -5 \leq x_i \leq 5$ and $i = 1, \ldots, n.$

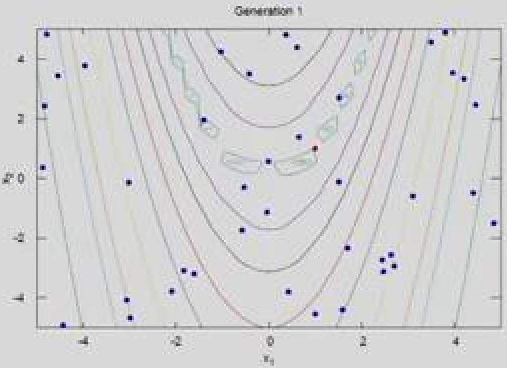


- Optimal solution is $x^* = (1, \ldots, 1)^T$ and $f(x) = 0$

Large w

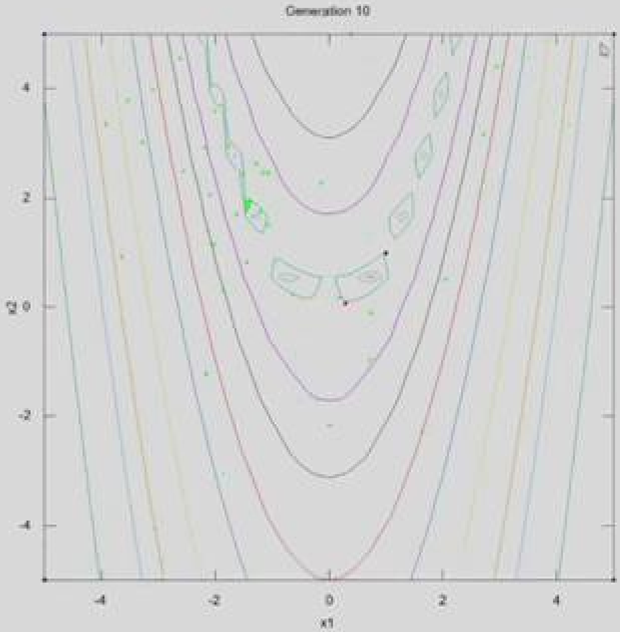
PSO Parameters

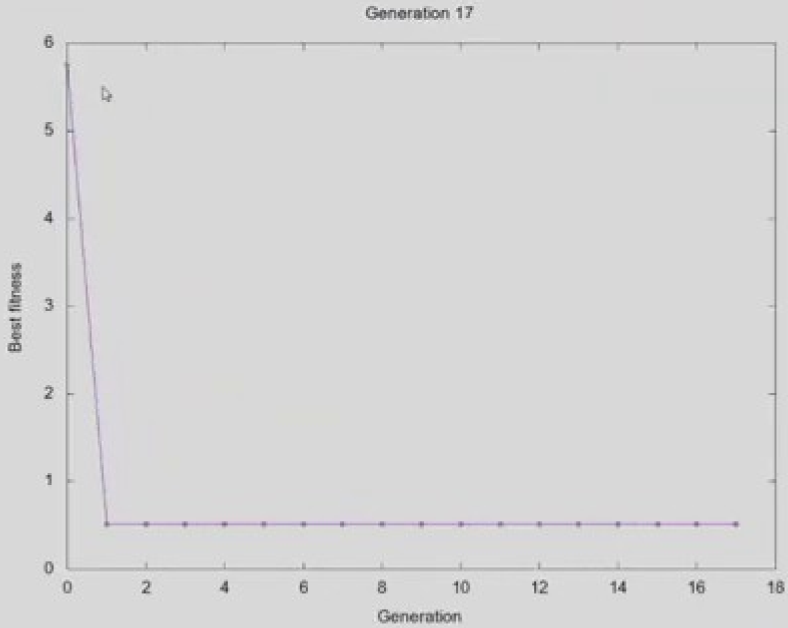
- Number of variables: $n = 2$
- Swarm size: $N = 40$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = c_2 = 1.5$
- Inertia equation $w = 5.0$.
- Initial velocity of each particle: $v^{(i)} = 0$



- Simulation Link
- Progress Link

- Velocities increase over time
- Swarm diverges
- Particles sometimes fail to change direction toward more promising regions


Generation 10
4
2
0
-2
-4
x2
-4
-2
0
2
4
x1

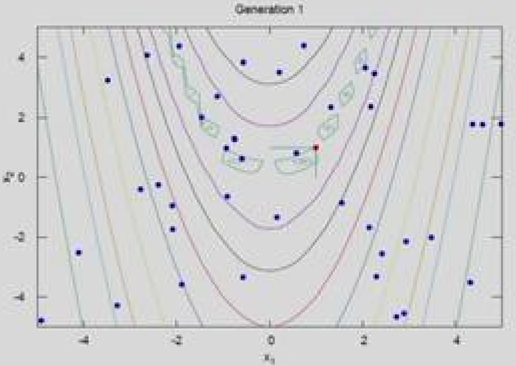


Generation 17
Best fitness
Generation
0
1
2
3
4
5
6
0
2
4
6
8
10
12
14
16
18

Effect of $c_1 > c_2$

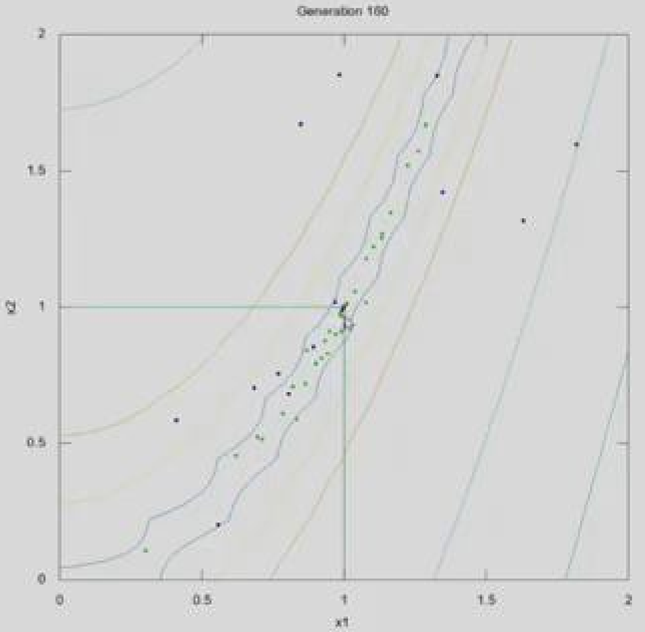
PSO Parameters

- Number of variables: $n = 2$
- Swarm size: $N = 40$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = 2.5, c_2 = 0.5$
- $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$



- Simulation Link
- Progress Link

- It can be useful for multi-modal optimization problems.

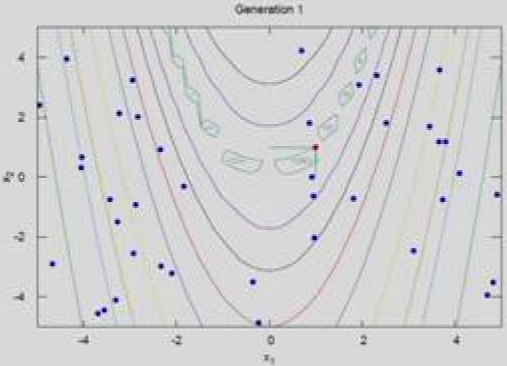


Generation 180
2
1.5
1
0.5
0
0
0.5
1
1.5
2
x1
x2

Effect of $c_1 < c_2$

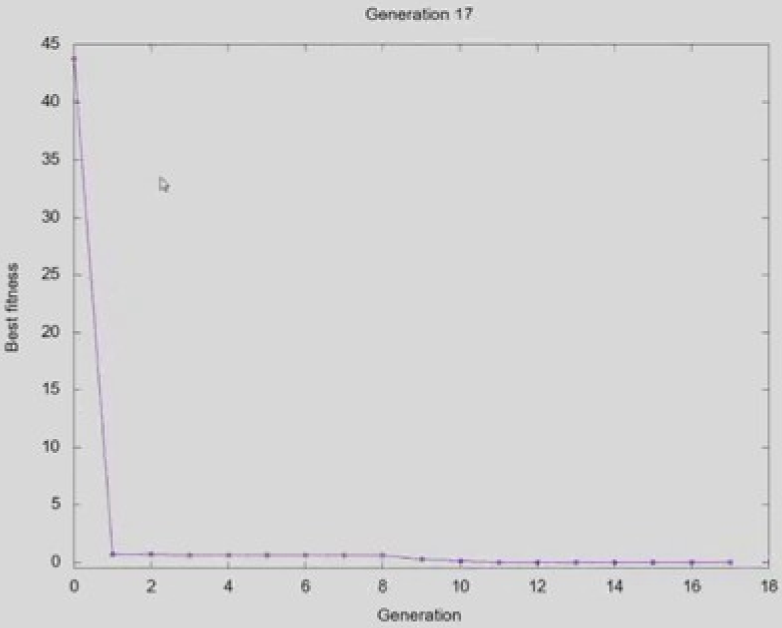
PSO Parameters

- Number of variables: $n = 2$
- Swarm size: $N = 40$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = 0.5, c_2 = 2.5$
- $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$



- Simulation ▸ Link
- Progress ▸ Link

- It can be useful for unimodal optimization problems.

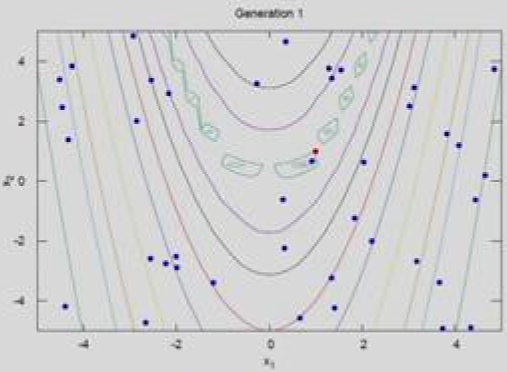


Generation 17
Best fitness
Generation
0
5
10
15
20
25
30
35
40
45
0
2
4
6
8
10
12
14
16
18

Low c_1 and c_2 values

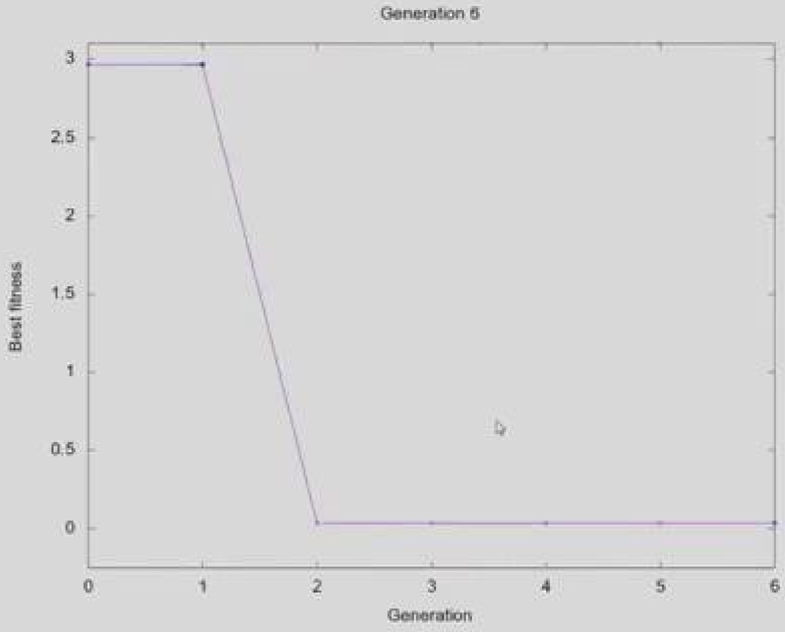
PSO Parameters

- Number of variables: $n = 2$
- Swarm size: $N = 40$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = 0.5, c_2 = 0.5$
- $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$



- Simulation Link
- Progress Link

- Smooth particles trajectories can be observed.

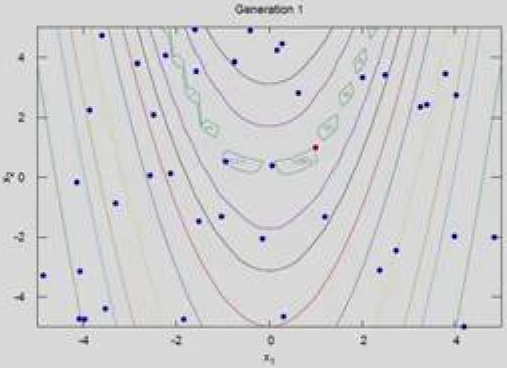


Generation 6
Best fitness
Generation
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
6

Large c_1 and c_2 values

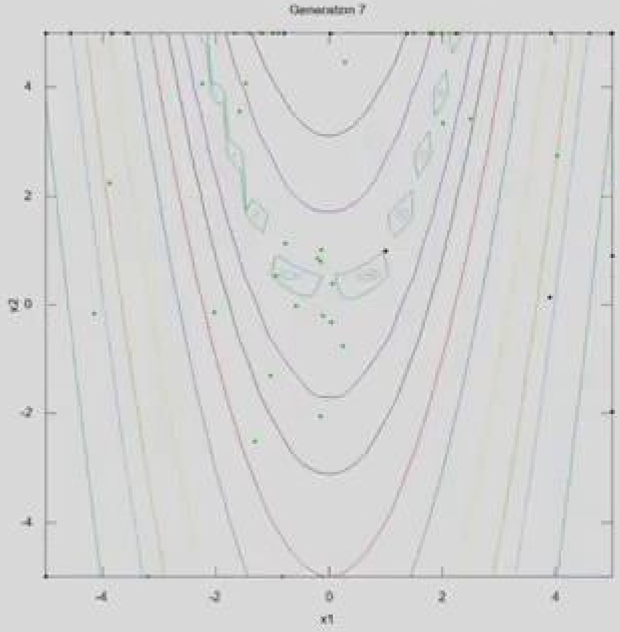
PSO Parameters

- Number of variables: $n = 2$
- Swarm size: $N = 40$
- No. of generations: $T = 200$
- Velocity coefficients: $c_1 = 5.0, c_2 = 5.0$
- $w = w_{max} - \frac{t}{T}(w_{max} - w_{min})$, where $w_{max} = 0.9$ and $w_{min} = 0.1$.
- Initial velocity of each particle: $v^{(i)} = 0$

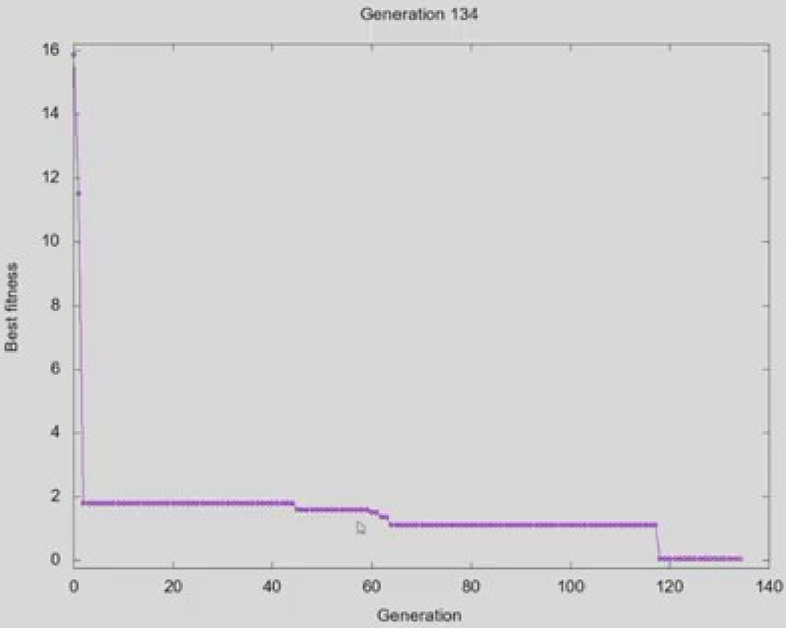


- Simulation Link
- Progress Link

- It supports large acceleration to particles but with abrupt movement.



Generation 7
x1
x2


Generation 134
Best fitness
Generation
0
2
4
6
8
10
12
14
16
0
20
40
60
80
100
120
140

Issues

- We can observe a potential dangerous property of PSO
  - when $x_i^{(t)} = p_{(i,lb)}^{(t)} = p_{gb}^{(t)}$
  - The velocity update depends only on $wv_i^{(t)}$
  - If this condition persists for a number of generations, $wv_i^{(t)} \to 0$
- There are certain potential problems with PSO
  - **Infeasible solutions:** Particles leave the search boundaries frequently.
  - **Wasted search effort:** Particles are pulled back into the feasible space or on the boundary.
  - **Incorrect swarm diversity calculations:** As particles move outside of the search boundaries, diversity increases.

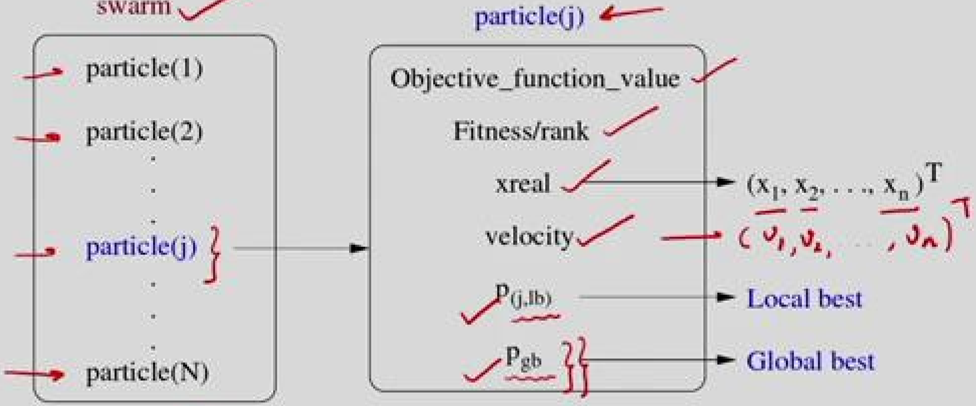
Algorithmic Implementation of PSO

Algorithm 1 Generalized Framework

Solution representation — % Genetics
Input: $t := 1$ (Generation counter), Maximum allowed generation $= T$, etc.
Initialize random swarm $(P(t))$; — % Swarm
Evaluate $(P(t))$; — % Evaluate objective, constraints and assign fitness
while $t \leq T$ **do**
    Update $p_{(i,lb)}^{(t)}$ of each particle (i) and find $p_{gb}^{(t)}$; — % New step
    for $(i = 1; i \leq N, i++)$ **do** — % For each particle i
        Update velocity $(v_i^{(t+1)})$;
        Update position $(x_i^{(t+1)})$; — % Variation
        Evaluate $(x_i^{(t+1)})$;
    end for
    $t := t + 1$;
end while

Data Structure for PSO

- Data Structure for swarm



- datatype **swarm**;
  - swarm.particle(j).objective_function_value;

Input to PSO

Algorithm 2 Input

```
Swarm size: N
Number of generations: T
Number of real variables: n
for (j = 1; j ≤ 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: w, c_1, c_2
```

Initialize random swarm

Algorithm 3 Initialize random population

```
Input: N: population size, n: number of variables
for (i = 1; i ≤ N; i++) do                                        %For each particle in the swarm
    for (j = 1; j ≤ n; j++) do                                    %For each variable of a particle
        x_j = Generate real number randomly between x_j^(L) and x_j^(U)
        v_j = 0                                                   %Initial velocity
    end for
end for
```

Evaluate Particle

Algorithm 4 Evaluate Population

1: **Input**: particle (j)
2: Evaluate $f(x_j)$ %Extract $x_j = (x_1, \dots, x_n)^T$ from the data structure of a particle(j)

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

Local best update of particle (j)

Algorithm 5 Local best update of particle (j)

```
Input: particle (j)
if ( t == 1) then                          %Only for the first generation
    p_(j,lb) = x_j;
else
    if (f(x_j) < f(p_(j,lb))) then          %Update when the fitness of particle is better
        p_(j,lb) = x_j;
    end if
end if
```

Global best of swarm

Algorithm 6 Global best of swarm

```
Input: P(t): swarm, N: size of swarm
p_gb = p_{1,lb}                          %Initialization of the global best with the local best of the first particle
for (j = 2; j ≤ N; j++) do
    if (f(p_{(j,lb)}) < f(p_gb)) then    %Update when the fitness of the local best of particle is better
        p_gb = p_{j,lb}
    end if
end for
```

Velocity and Position Updates

Algorithm 7 Velocity and position updates of each particle (j)

1: **Input**: particle (j) and constant parameters

2: Update velocity of particle (j) using

$$v_j = wv_j + c_1 r_1 (p_{(j,lb)} - x_j) + c_2 r_2 (p_{gb} - x_j)$$

3: Update position of particle (i) using

$$x_j = x_j + v_j$$

Copy Particle

Algorithm 8 Copy Particle

Input: particle 1, particle 2
Copy objective function value of particle 1 to particle 2
Copy fitness/rank of particle 1 to particle 2
Copy x_j of particle 1 to x_j of particle 2
Copy v_j of particle 1 to v_j of particle 2
Copy $p_{(1,lb)}$ of particle 1 to $p_{(2,lb)}$ of particle 2
Copy p_{gb} of particle 1 to p_{gb} of particle 2

- Copy the complete data structure

Closure

Closure

- Simulations of PSO on various functions
  - Rosenbrock function with $n = 2, 4, 10$ variables
  - Rastrigin function with $n = 2, 4, 10$ variables
  - Himmelblau multi-modal function
  - Ackley function
- Effect of PSO parameters
  - Effect of large w
  - Effect of setting $c_1 > c_2$
  - Effect of setting $c_1 < c_2$
  - Effect of low values of c_1 and c_2
  - Effect of large values of c_1 and c_2