

Project no. 1 - Piotr Damrych - 04.2023

N queens problem

General info

The n queens problem is the problem of placing n queens on an $n \times n$ chessboard so that no two queens threaten each other, i.e. no two queens can be on the same row, column or diagonal.

There are solutions for all natural numbers n except $n = 2$ and $n = 3$.

Genetic algorithm

A genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection where the fittest individuals are selected for reproduction in order to produce offspring of the next generation.

Particle swarm optimization

Particle swarm optimization (PSO) is one of the bio-inspired algorithms and it is a simple one to search for an optimal solution in the solution space. Each particle's movement is influenced by its local best known position, but is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

Libraries

We use several libraries to solve this problem. We use numpy, pandas, time, matplotlib and libraries to implement the algorithms. For genetic algorithm (GA) we use pygad and for particle swarm optimization (PSO) we use pyswarms.

```
In [2]: import pygad
import numpy as np
import time
import pyswarms as ps
from pyswarms.utils.plotters import plot_cost_history
import matplotlib.pyplot as plt
```

GA

Helper functions no. 1

The first helper function is the createBoard function, which, given an integer n, creates an array of n x n arrays filled with zeros.

```
In [3]: n = 8

def createBoard(n: int) -> list[list[int]]:
    return np.zeros((n, n), dtype=int)

boardToShow = createBoard(n)
print("Board:\n", boardToShow)
```

Board:

```
[[0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0]]
```

Fitness function and helper functions no. 2

The fulfillBoard function is a function that, after receiving the table of solutions with genes (chromosome), inserts ones in the places of zeros where the solution indicates the place of the queen.

The checkCollision function is a function that checks whether there is any collision for the given chessboard and the given column and row values - another queen's attack. The function must check if there is another queen in the same row, column or diagonal.

The fitness function is a function that determines how many attacks occur on a given board. Initially, it receives tables with genes as input (chromosome), then it creates a new board with the help of a helper function and puts queens on the board in places given by genes. Then it iterates through the whole thing and checks the number of attacks. At the end it returns the number of attacks * -1, since our maximum is 0 - when no queen is attacked. This function is crucial to find the best solution for the genetic algorithm and solve our problem.

```
In [4]: def fulfillBoard(solution: list[int]) -> list[list[int]]:
        board = createBoard(n)

        for i in range(len(solution)):
            x = solution[i]
            board[i][x] = 1

        return board

def checkColission(row: int, col: int, board: list[list[int]]) -> bool:
    n = len(board)

    # Check row and column
    if sum(board[row]) > 1 or sum([board[i][col] for i in range(n)]) > 1:
        return True

    # Check diagonal
    for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
        if board[i][j] == 1 and (i, j) != (row, col):
            return True
    for i, j in zip(range(row, n), range(col, -1, -1)):
        if board[i][j] == 1 and (i, j) != (row, col):
            return True

    return False

def fitness_func(solution, solution_idx):
    newSol = solution.astype(int)
    attacks = 0
    board = fulfillBoard(newSol)

    for i in range(len(board)):
        for j in range(len(board[i])):
            if board[i][j] == 1:
                if checkColission(i, j, board):
                    attacks += 1

    board = createBoard(n)
    return (attacks * -1)

fitness_function = fitness_func
```

Helper functions no. 3 and gene space

The functions `printResult` and `printAverages` are functions that allow us to print the output of our results in a nice way.

The `ga_instance_generate` function creates instances of the genetic algorithm. It gives us:

- `gene_space`, which is what our gene looks like. It is a number from 0 to n and determines the index of the column in which our queen will be located. The row index is the index of the number in the array, because queens can't be on the same row anyway, it's just a way to avoid it at the beginning,
- `num_generations`, i.e. the number of generations. For us it is 400,
- `num_parents_mating`, i.e. number of solutions to be selected as parents. We set 150,
- `fitness_func`, which is our previous fitness function,
- `sol_per_pop`, i.e. number of solutions (i.e. chromosomes) within the population. This parameter has no action if `initial_population` parameter exists. For us it is 450,
- `num_genes`, i.e. number of genes in the solution/chromosome. We need n queens, so that's n ,
- `parent_selection_type`, i.e. parent selection type. Supported types are `sss` (for steady-state selection), `rws` (for roulette wheel selection), `sus` (for stochastic universal selection), `rank` (for rank selection), `random` (for random selection), and `tournament` (for tournament selection). We use `sss`,
- `keep_parents`, i.e. number of parents to keep in the current population. For us it's 60,
- `crossover_type`, i.e. type of the crossover operation. We use `single_point`,
- `mutation_type` - type of the mutation operation. We use `random`,
- `mutation_percent_genes`, i.e. percentage of genes to mutate. We use 8 percent,
- `stop_criteria`, i.e. when our algorithm should stop working. Our algorithm terminates when it finds the maximum, i.e. 0, and when it does not improve its solution after 200 generations.

```
In [5]: def printResult(solution, solution_fitness):
    print("Parameters of the best solution : {solution}".format(solution=solution.a
    print("Fitness value of the best solution = {solution_fitness}".format(
        solution_fitness=solution_fitness))

def printAverages(times, generations, solved):
    average_time = 0
    for time in times:
        average_time += time
    average_time = average_time / len(times)
    print("Average time (sec): ", average_time)

    average_gen = 0
    for gen in generations:
        average_gen += gen
    average_gen = average_gen / len(generations)
    print("Average number of generations ", average_gen)

    print(f"GA solved {solved}/100")

def ga_instance_generate(n):
    #sss=steady, rws=roulette, rank, tournament
    parent_selection_type = "sss"
    crossover_type = "single_point"
    mutation_type = "random"
    mutation_percent_genes = 8
    gene_space = np.arange(n)
    sol_per_pop = 450
    num_genes = n
    num_parents_mating = 150
    num_generations = 400
    keep_parents = 60
    return pygad.GA(gene_space=gene_space,
                    num_generations=num_generations,
                    num_parents_mating=num_parents_mating,
                    fitness_func=fitness_function,
                    sol_per_pop=sol_per_pop,
                    num_genes=num_genes,
                    parent_selection_type=parent_selection_type,
                    keep_parents=keep_parents,
                    crossover_type=crossover_type,
                    mutation_type=mutation_type,
                    mutation_percent_genes=mutation_percent_genes,
                    stop_criteria=["reach_0", "saturate_200"])
```

Small input no. 1

We perform our first test on $n = 1$. As you can see, our algorithm found a solution right away without any problems, which is a good sign.

In [6]: `n = 1`

```
ga_instance = ga_instance_generate(n)

ga_instance.run()

solution, solution_fitness, solution_idx = ga_instance.best_solution()
printResult(solution, solution_fitness)

ga_instance.plot_fitness(n)
```

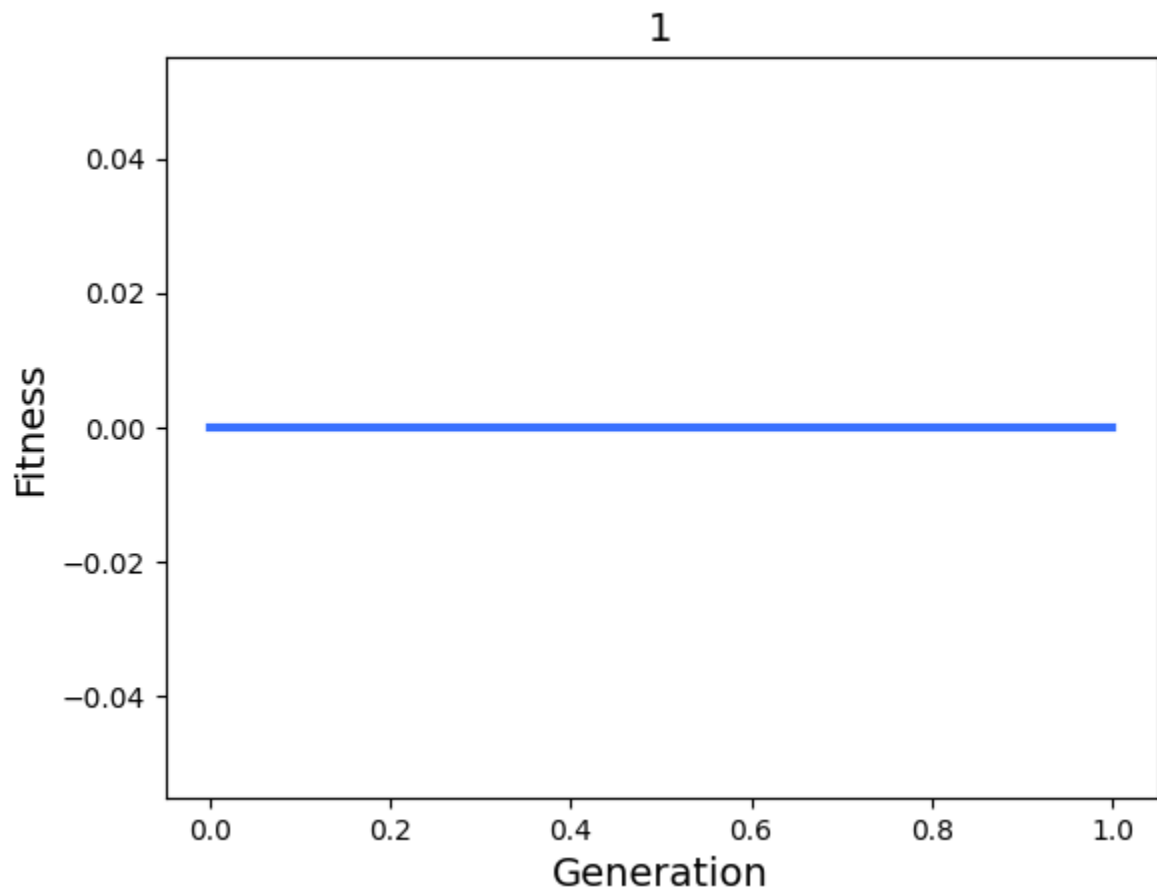
c:\Users\Piotr Damrych\AppData\Local\Programs\Python\Python311\Lib\site-packages\pygad\pygad.py:522: UserWarning: The percentage of genes to mutate (mutation_percent_genes=8) resulted in selecting (0) genes. The number of genes to mutate is set to 1 (mutation_num_genes=1).

If you do not want to mutate any gene, please set mutation_type=None.

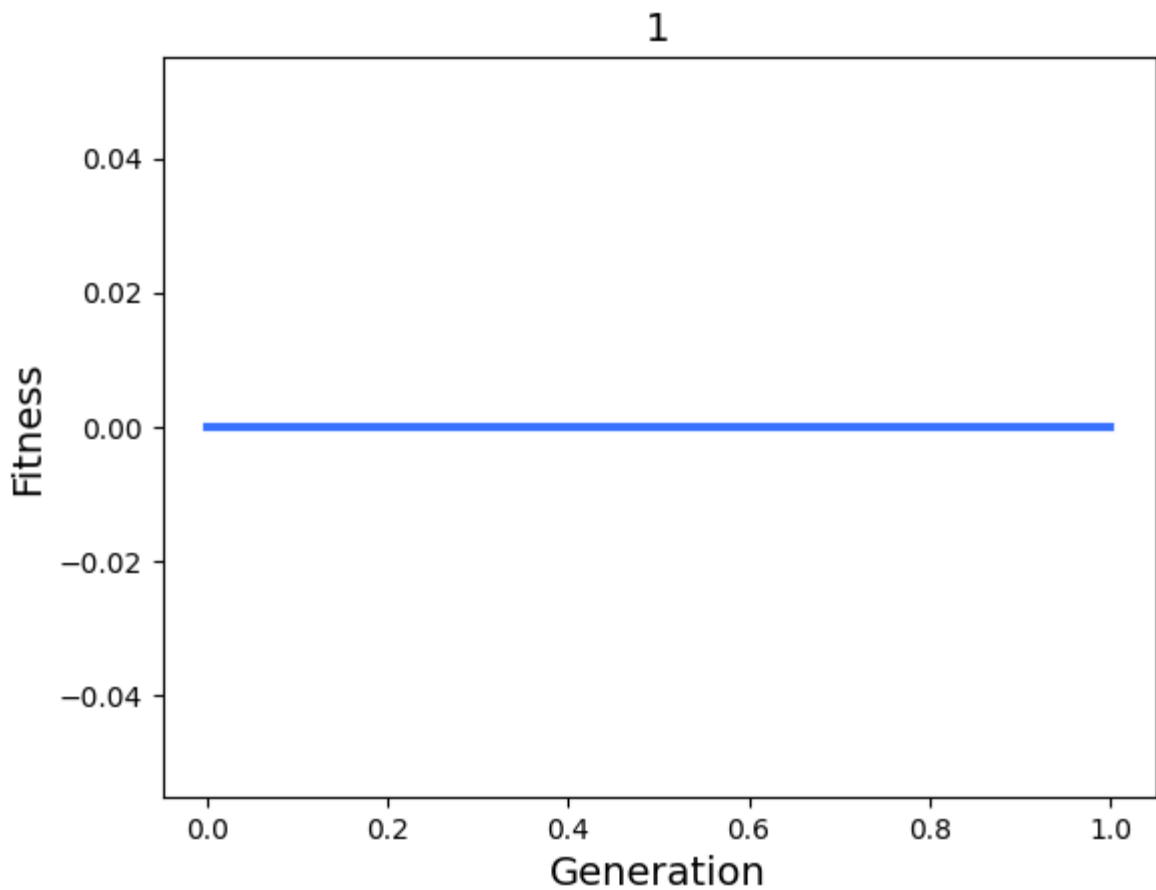
if not self.suppress_warnings: warnings.warn("The percentage of genes to mutate (mutation_percent_genes={mutation_percent}) resulted in selecting ({mutation_num}) genes. The number of genes to mutate is set to 1 (mutation_num_genes=1).\nIf you do not want to mutate any gene, please set mutation_type=None.".format(mutation_percent=mutation_percent_genes, mutation_num=mutation_num_genes))

Parameters of the best solution : [0]

Fitness value of the best solution = 0



Out[6]:



Small input no. 2

The second test was performed on $n = 2$, which is one of the exceptions for which there is no solution. As you can see, our algorithm didn't find it either (which is also a good sign).

```
In [13]: n = 2

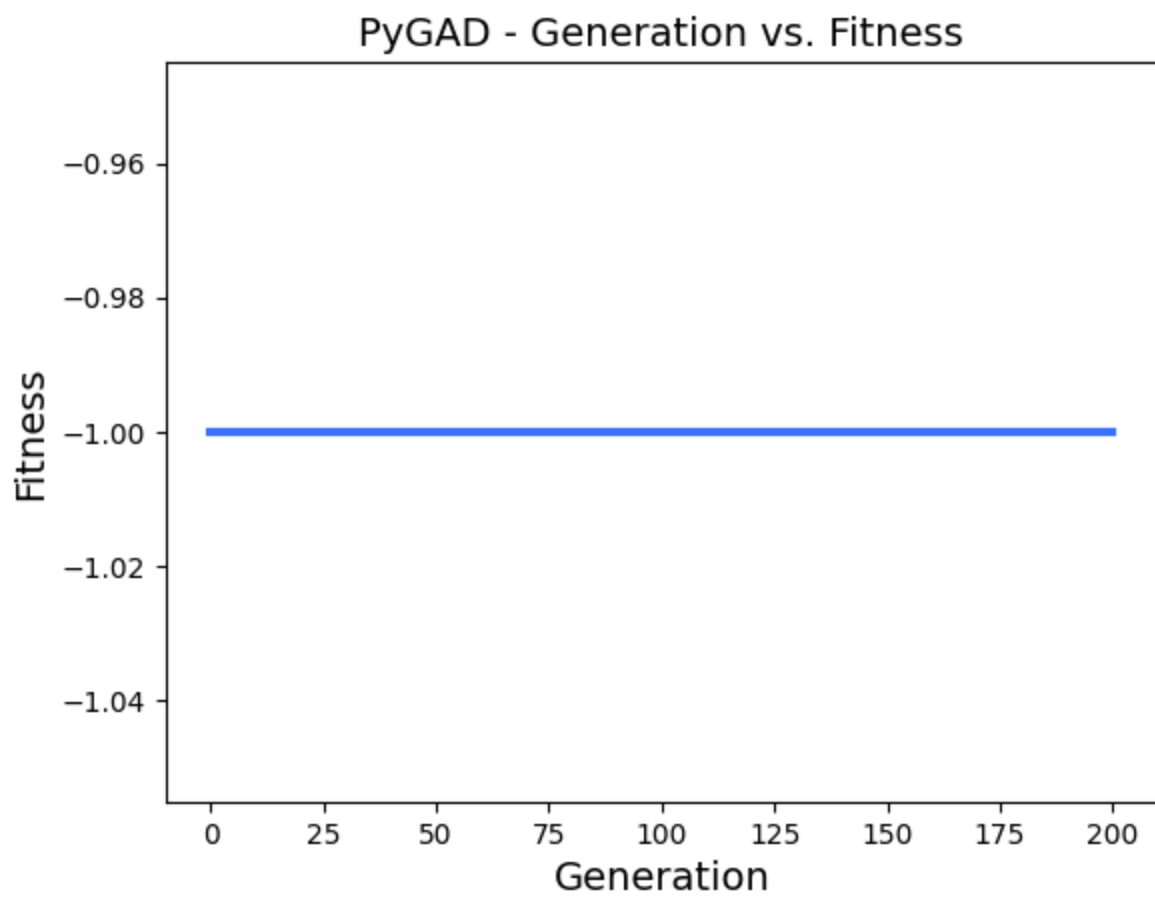
ga_instance = ga_instance_generate(n)

ga_instance.run()

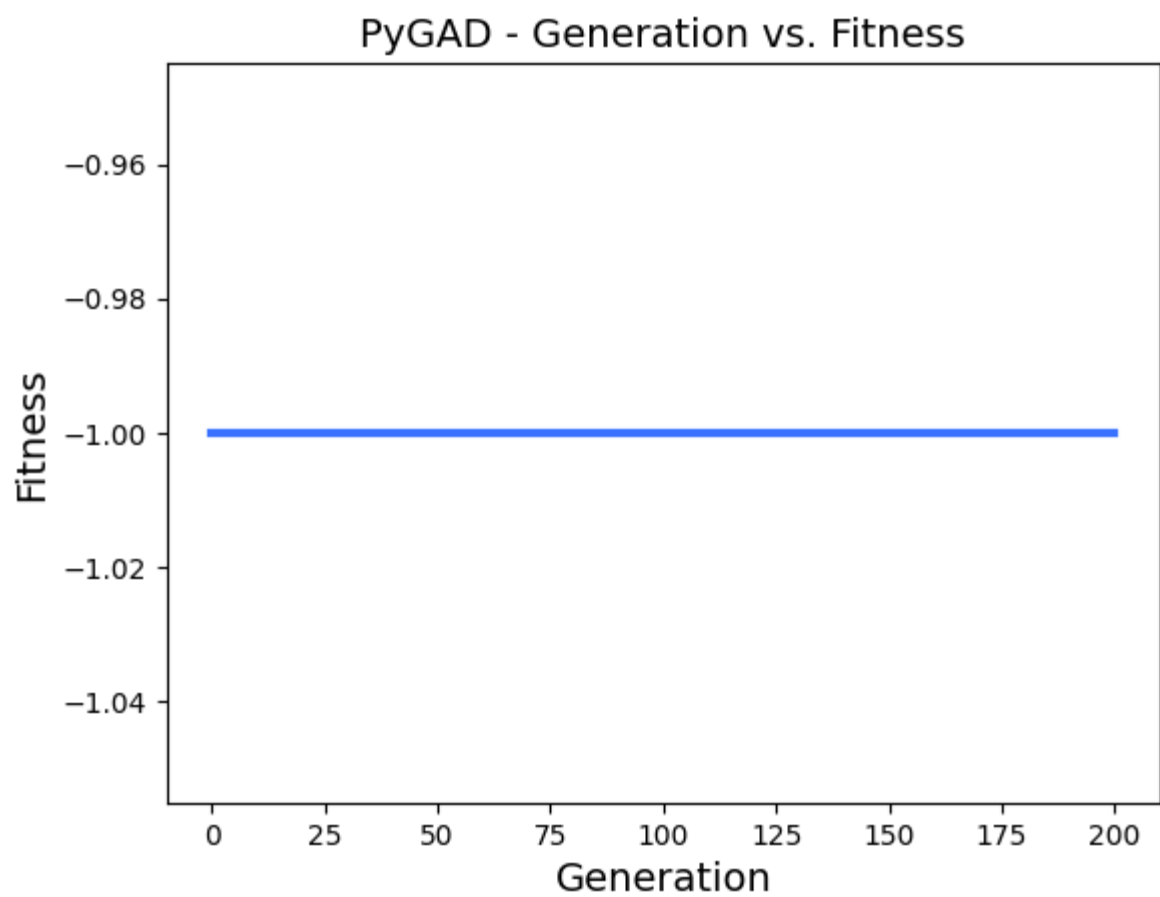
solution, solution_fitness, solution_idx = ga_instance.best_solution()
print("Parameters of the best solution : {solution}".format(solution=solution, astyp=
print("Fitness value of the best solution = {solution_fitness}".format(
    solution_fitness=solution_fitness))

ga_instance.plot_fitness()
```

```
Parameters of the best solution : [0 1]
Fitness value of the best solution = -1
```



Out[13]:



Small input no. 3

We carried out the next, third test on input $n = 5$. Our algorithm also found a solution without any problems.

In [14]: `n = 5`

```
ga_instance = ga_instance_generate(n)
```

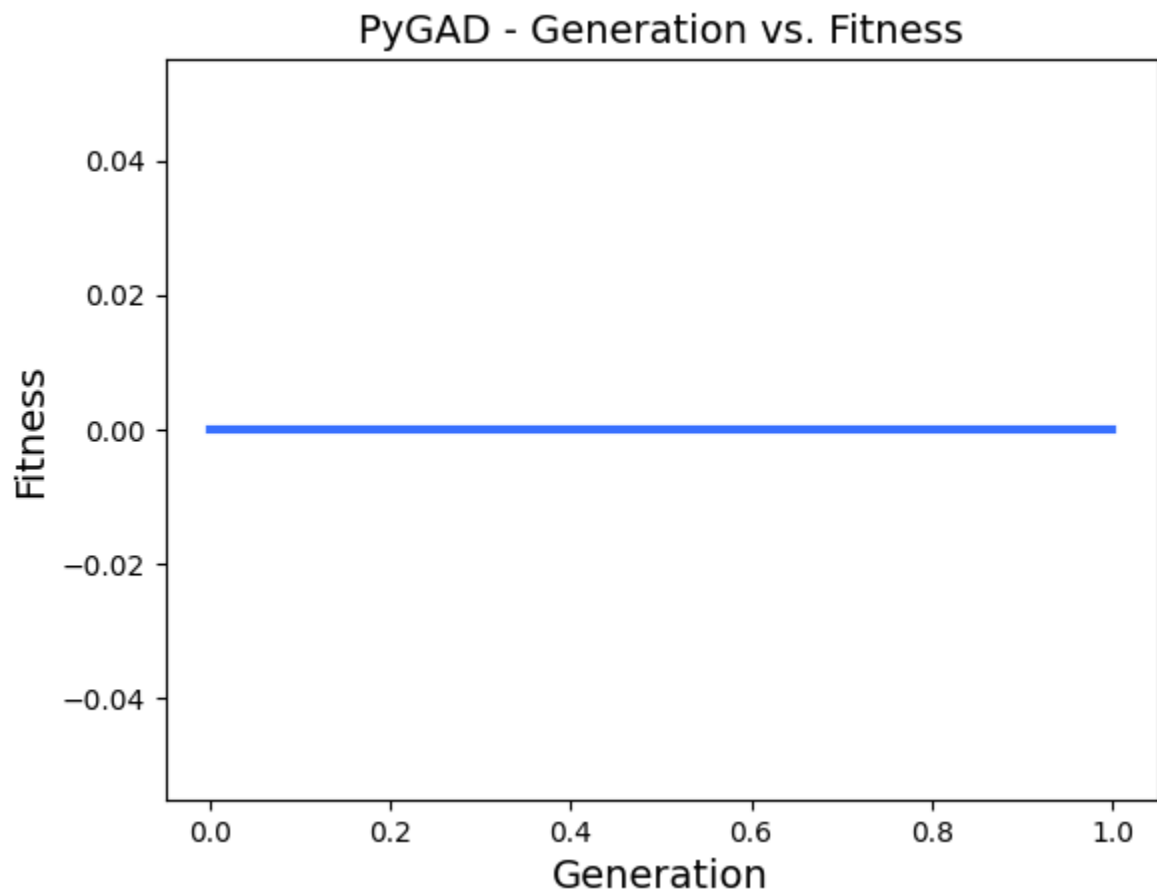
```
ga_instance.run()
```

```
solution, solution_fitness, solution_idx = ga_instance.best_solution()  
printResult(solution, solution_fitness)
```

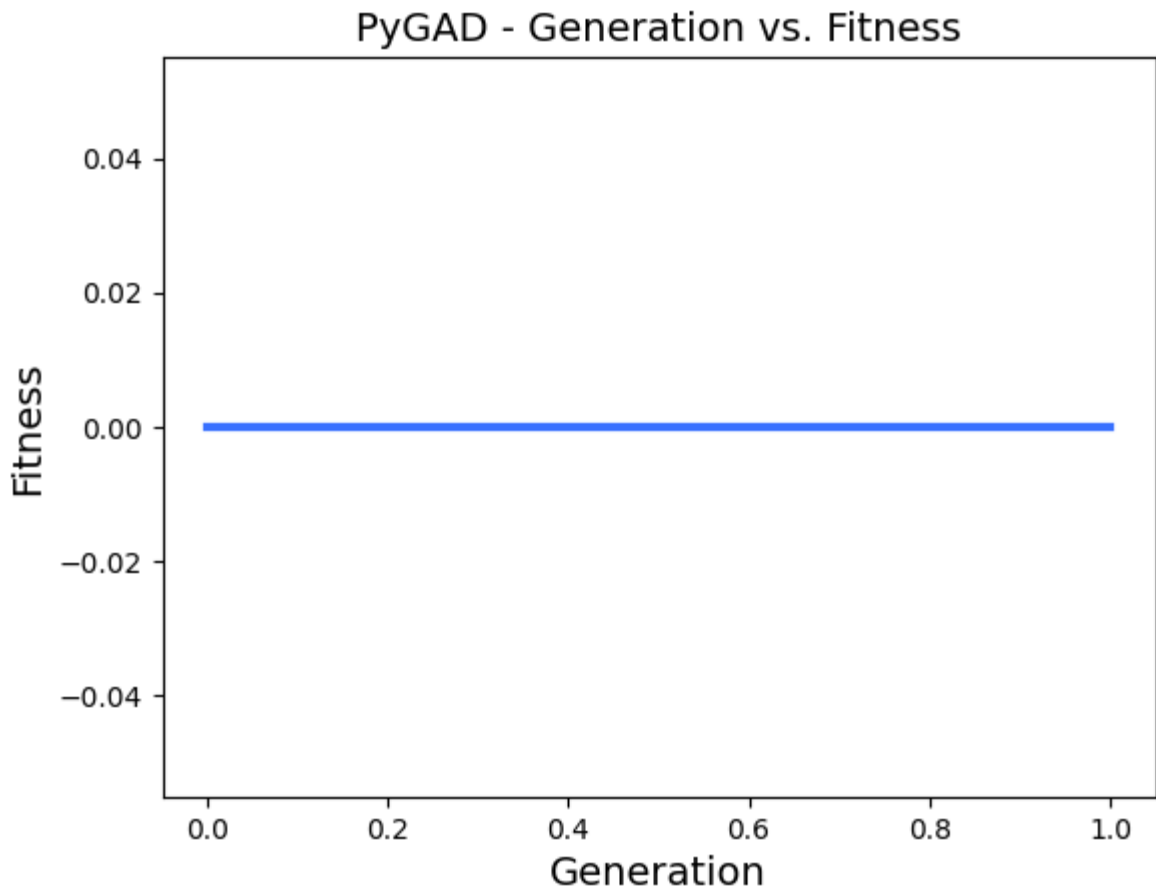
```
ga_instance.plot_fitness()
```

Parameters of the best solution : [2 4 1 3 0]

Fitness value of the best solution = 0



Out[14]:



Medium input no. 1

The next, fourth input, the most popular for the problem, i.e. $n = 8$. Our algorithm found a solution without any problem, already in about 25 generations.

```
In [15]: n = 8

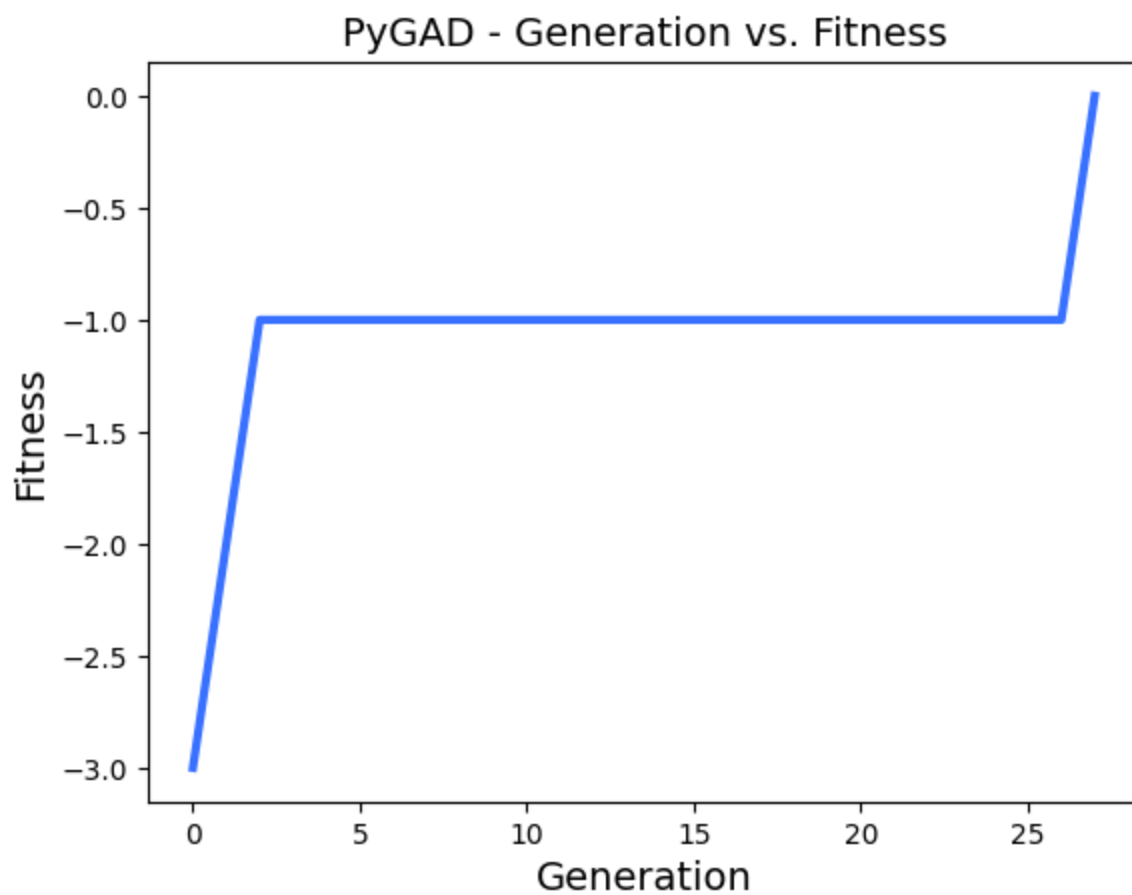
ga_instance = ga_instance_generate(n)

ga_instance.run()

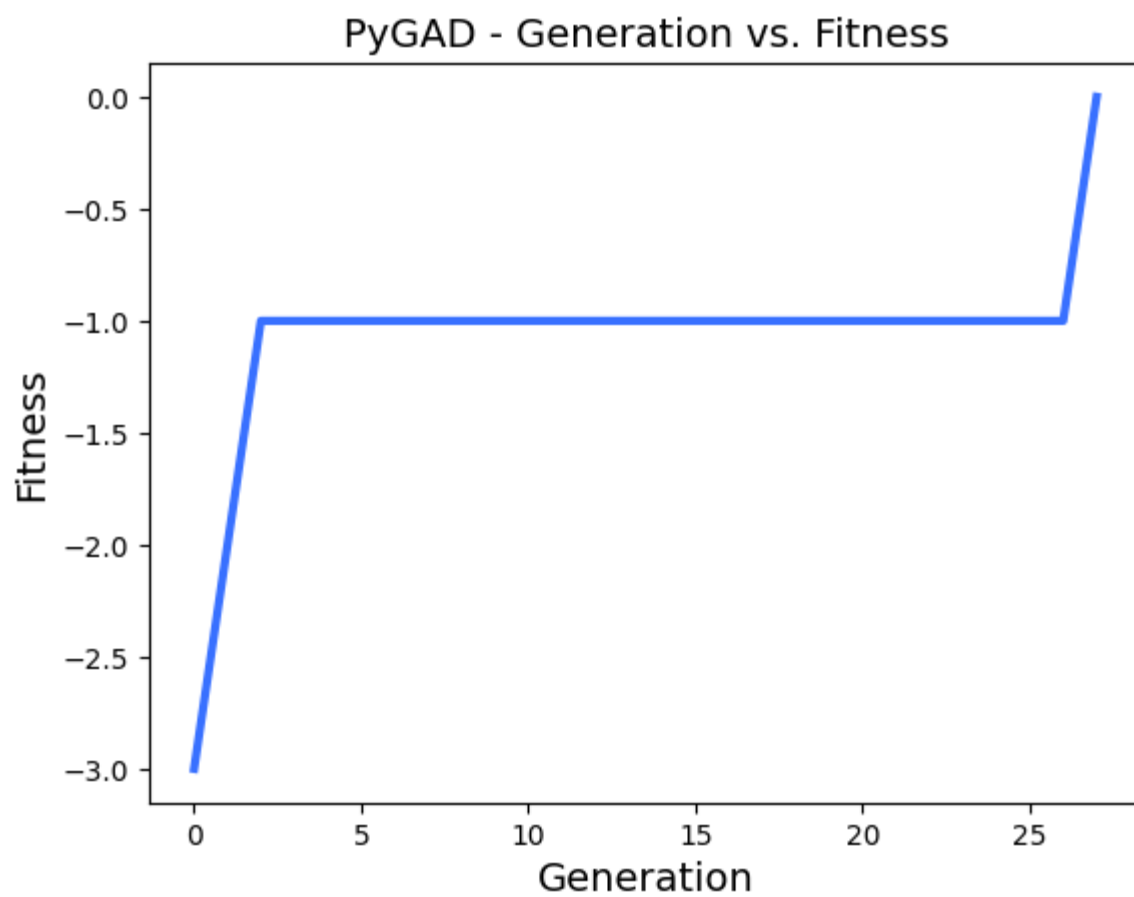
solution, solution_fitness, solution_idx = ga_instance.best_solution()
printResult(solution, solution_fitness)

ga_instance.plot_fitness()
```

Parameters of the best solution : [6 0 2 7 5 3 1 4]
Fitness value of the best solution = 0



Out[15]:



Medium input no. 2

We conducted the fifth test on a slightly larger input, $n = 12$. Our algorithm found a solution, but it required > 100 generations.

In [206...

```
n = 12

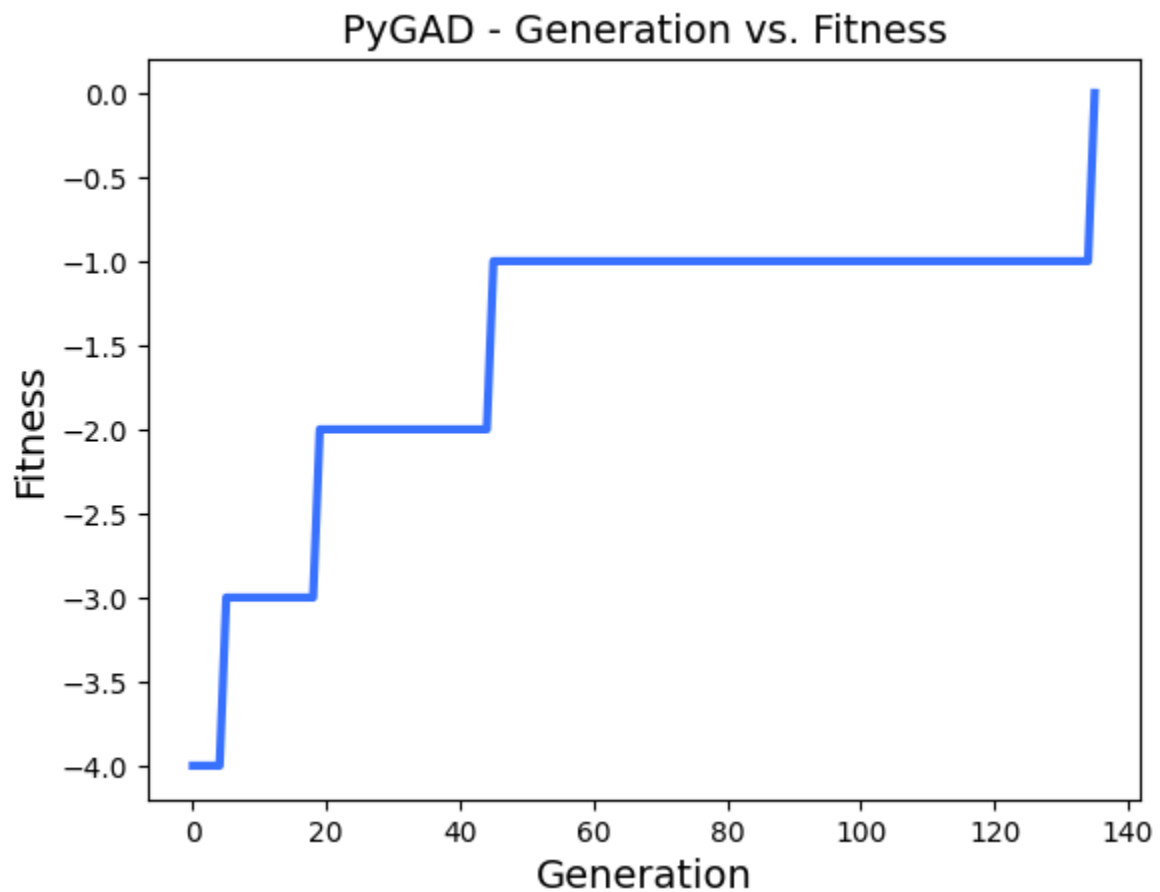
ga_instance = ga_instance_generate(n)

ga_instance.run()

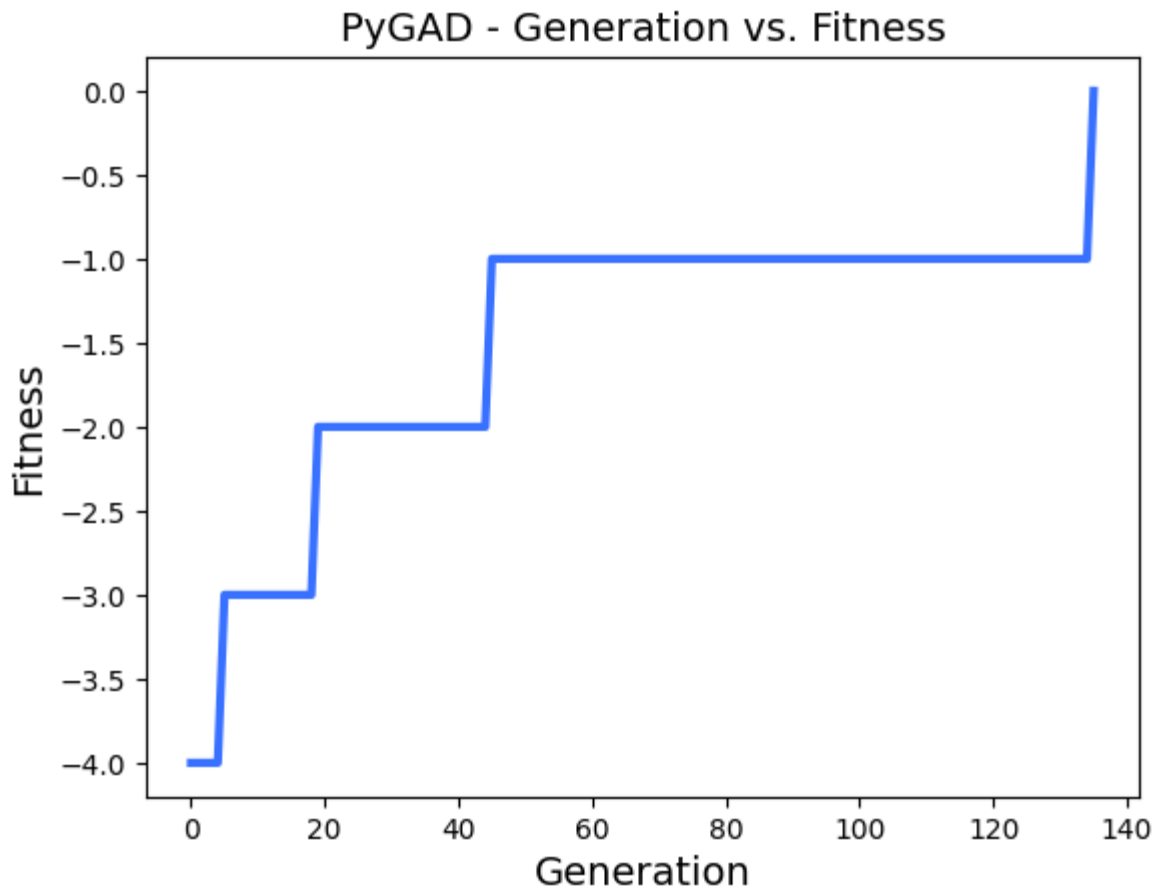
solution, solution_fitness, solution_idx = ga_instance.best_solution()
printResult(solution, solution_fitness)

ga_instance.plot_fitness()
```

Parameters of the best solution : [3 10 8 1 11 7 2 0 5 9 4 6]
Fitness value of the best solution = 0



Out[206]:



Medium input no. 3

We conducted the last test for medium inputs on $n = 14$. Our algorithm surprisingly found a solution faster than for $n = 12$, but this is probably just a coincidence. A good sign, however, is that there is no problem finding a valid solution for an already large enough input.

In [203...

```
n = 14

ga_instance = ga_instance_generate(n)

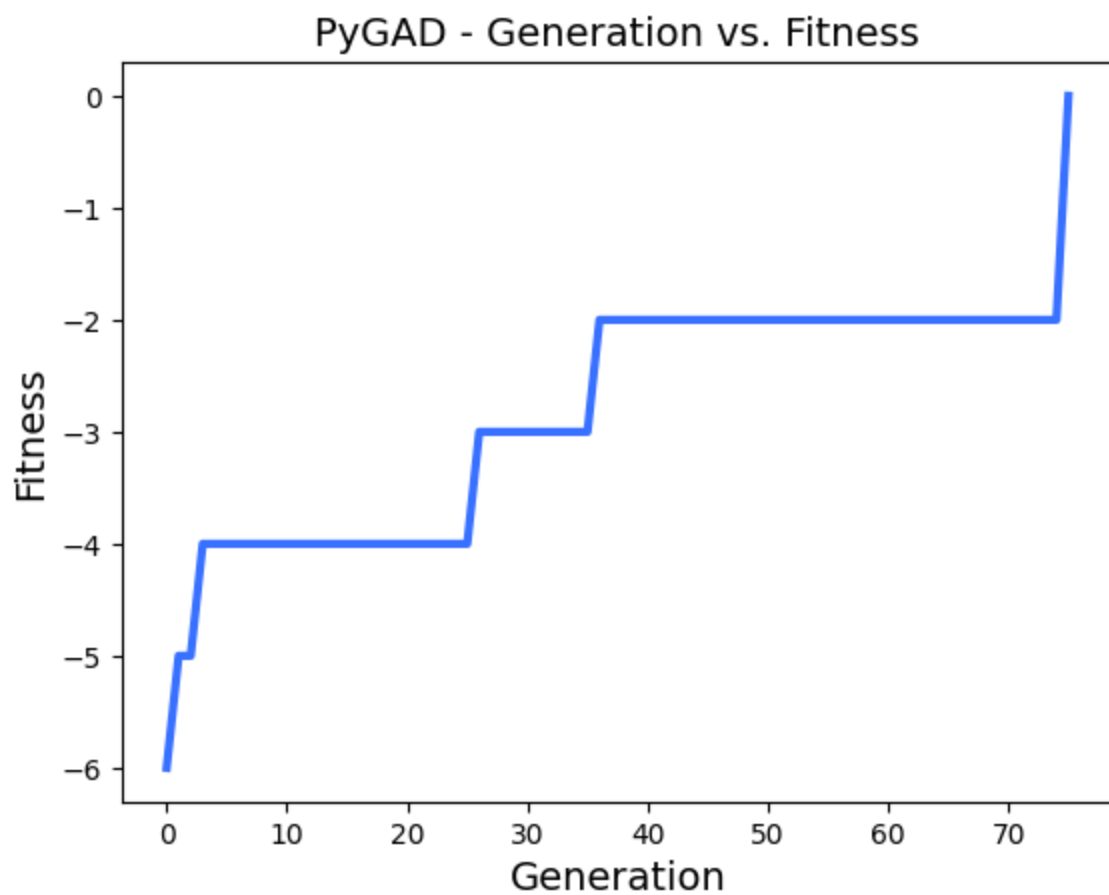
ga_instance.run()

solution, solution_fitness, solution_idx = ga_instance.best_solution()
printResult(solution, solution_fitness)

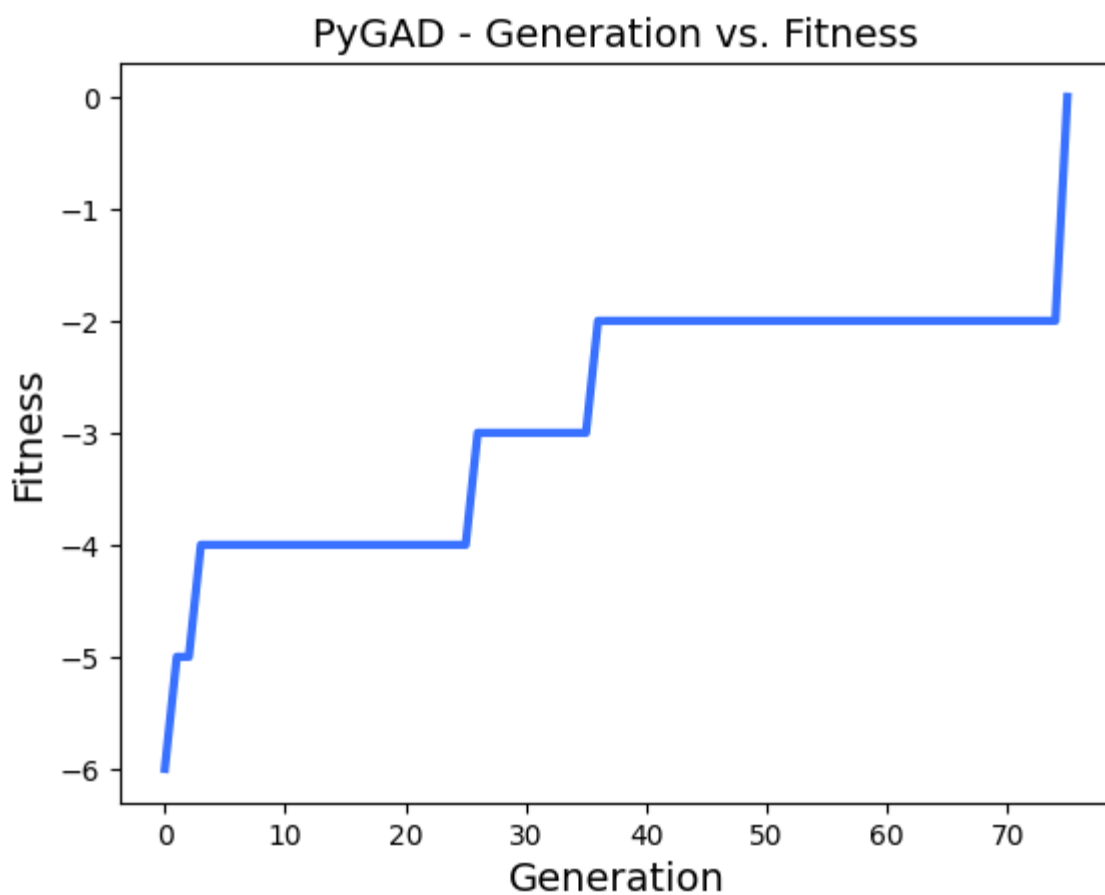
ga_instance.plot_fitness()
```

Parameters of the best solution : [4 2 11 9 1 5 8 12 7 0 13 6 10 3]

Fitness value of the best solution = 0



Out[203]:



Big input no. 1

We conducted our first big input test, the seventh in total, on $n = 18$. This is already quite a large chess board. Our algorithm also dealt with it without any major problems and found a solution after about 100 generations. The graph shows how beautifully our algorithm climbed up.

In [178...

```
n = 18

ga_instance = ga_instance_generate(n)

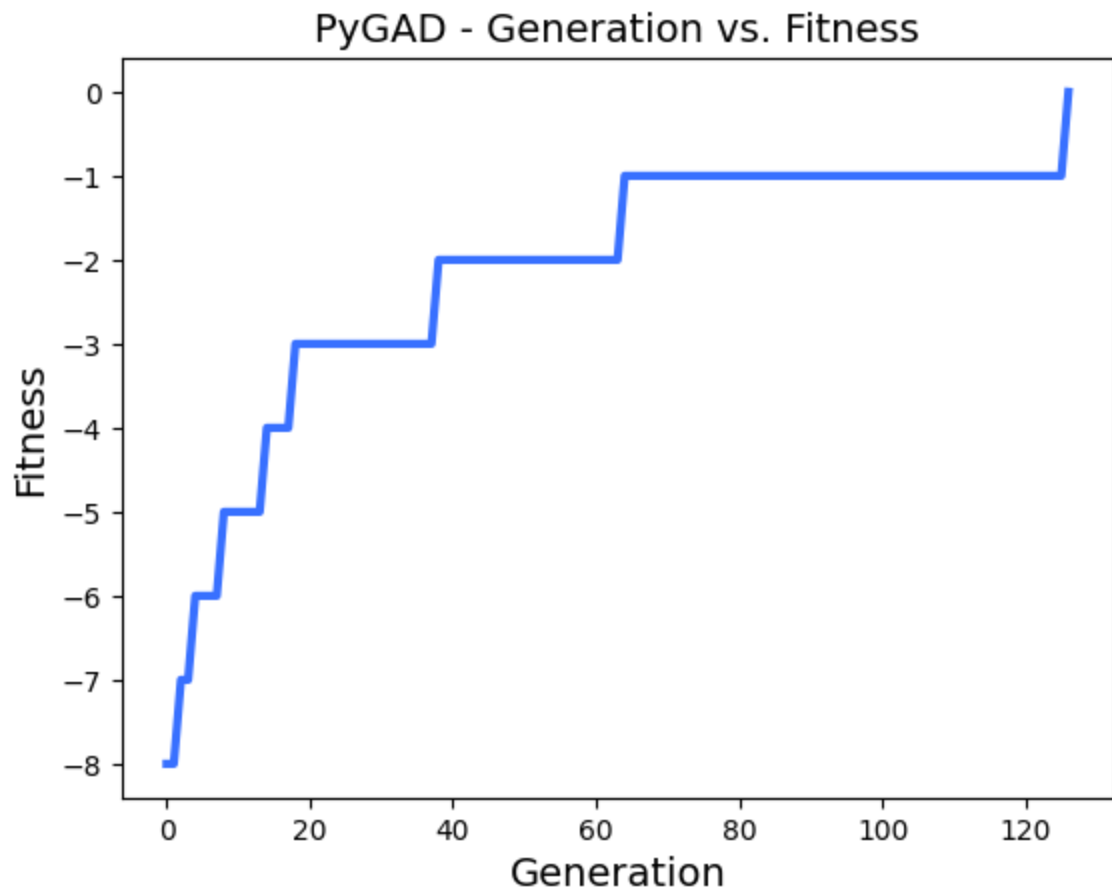
ga_instance.run()

solution, solution_fitness, solution_idx = ga_instance.best_solution()
printResult(solution, solution_fitness)

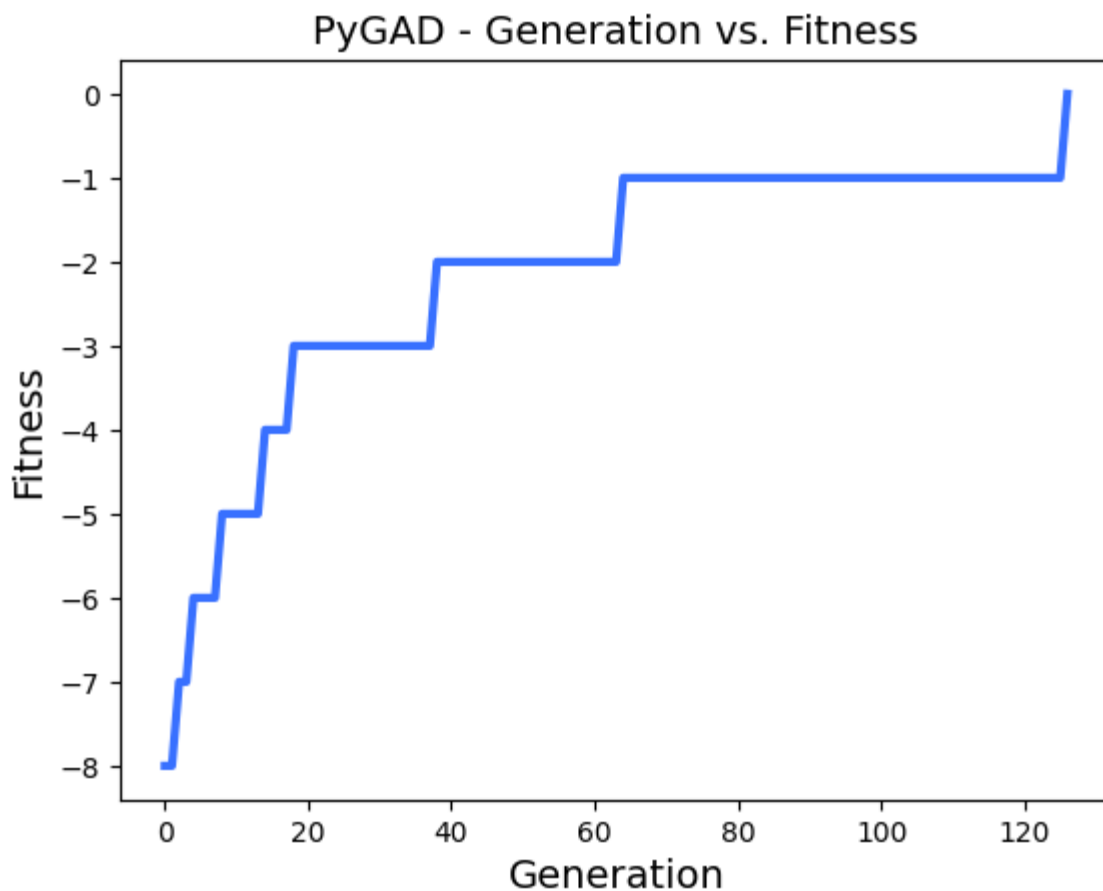
ga_instance.plot_fitness()
```

Parameters of the best solution : [10 13 6 1 11 7 3 0 16 9 15 12 8 4 17 14 5 2]

Fitness value of the best solution = 0



Out[178]:



Big input no. 2

The eighth test for $n = 20$. Here our algorithm starts to have more frequent problems and no longer finds the maximum in most trials. However, we found a solution after 80 generations.

In [135...

```
n = 20

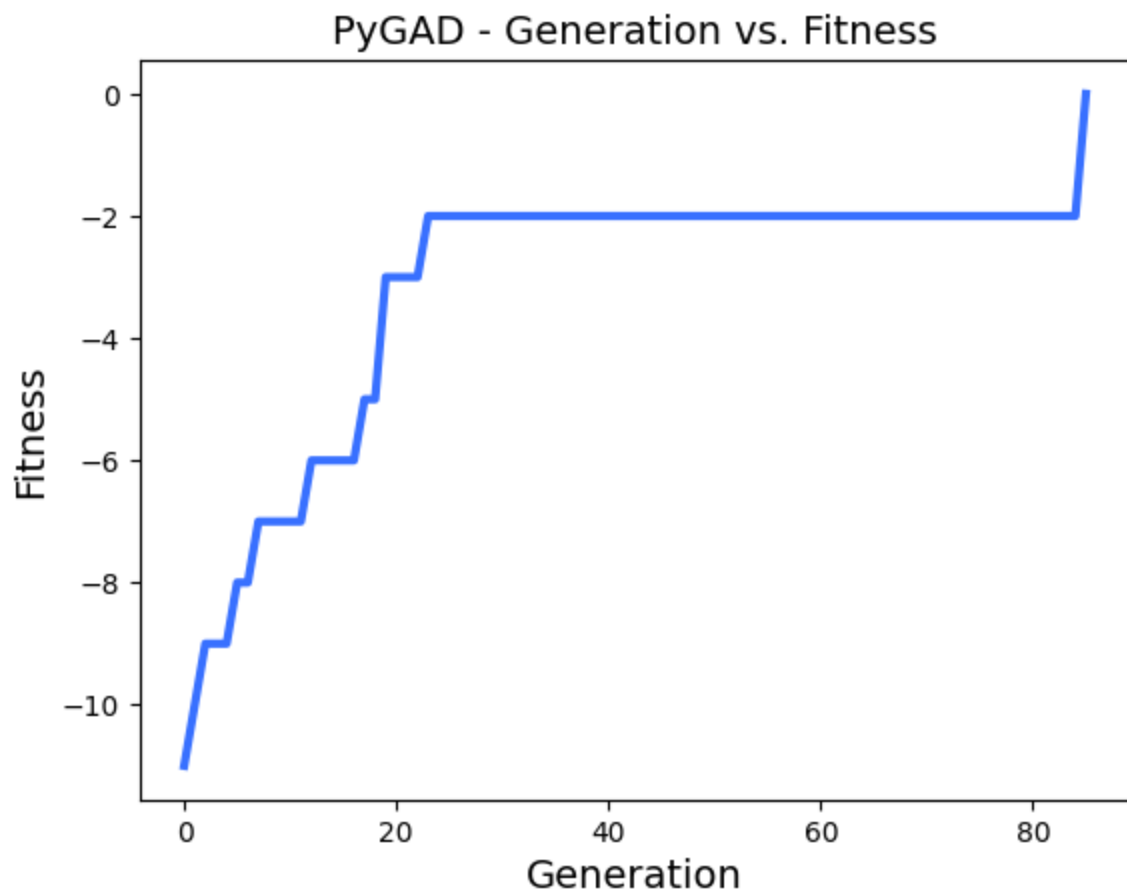
ga_instance = ga_instance_generate(n)

ga_instance.run()

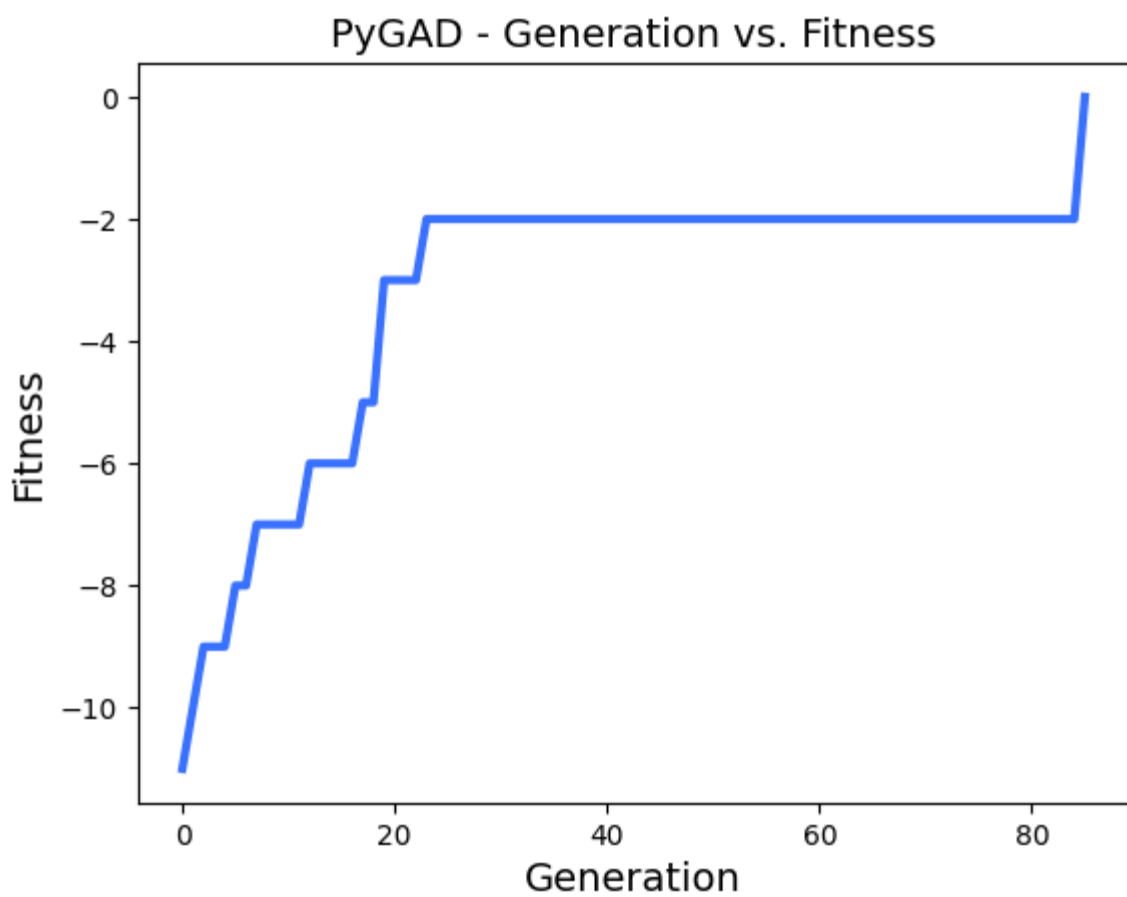
solution, solution_fitness, solution_idx = ga_instance.best_solution()
printResult(solution, solution_fitness)

ga_instance.plot_fitness()
```

Parameters of the best solution : [0 18 12 5 9 11 19 2 4 13 17 10 3 7 15 8 1
4 1 6 16]
Fitness value of the best solution = 0



Out[135]:



Big input no. 3

The last test for $n = 21$. Here, unfortunately, our algorithm did not find good solutions in most of the tests, similar to $n = 20$, but this is a really large input. However, we found a solution after about 80 generations. The graph shows how beautifully our algorithm searched for new maxima of the function.

In [173...

```
n = 21
```

```
ga_instance = ga_instance_generate(n)
```

```
ga_instance.run()
```

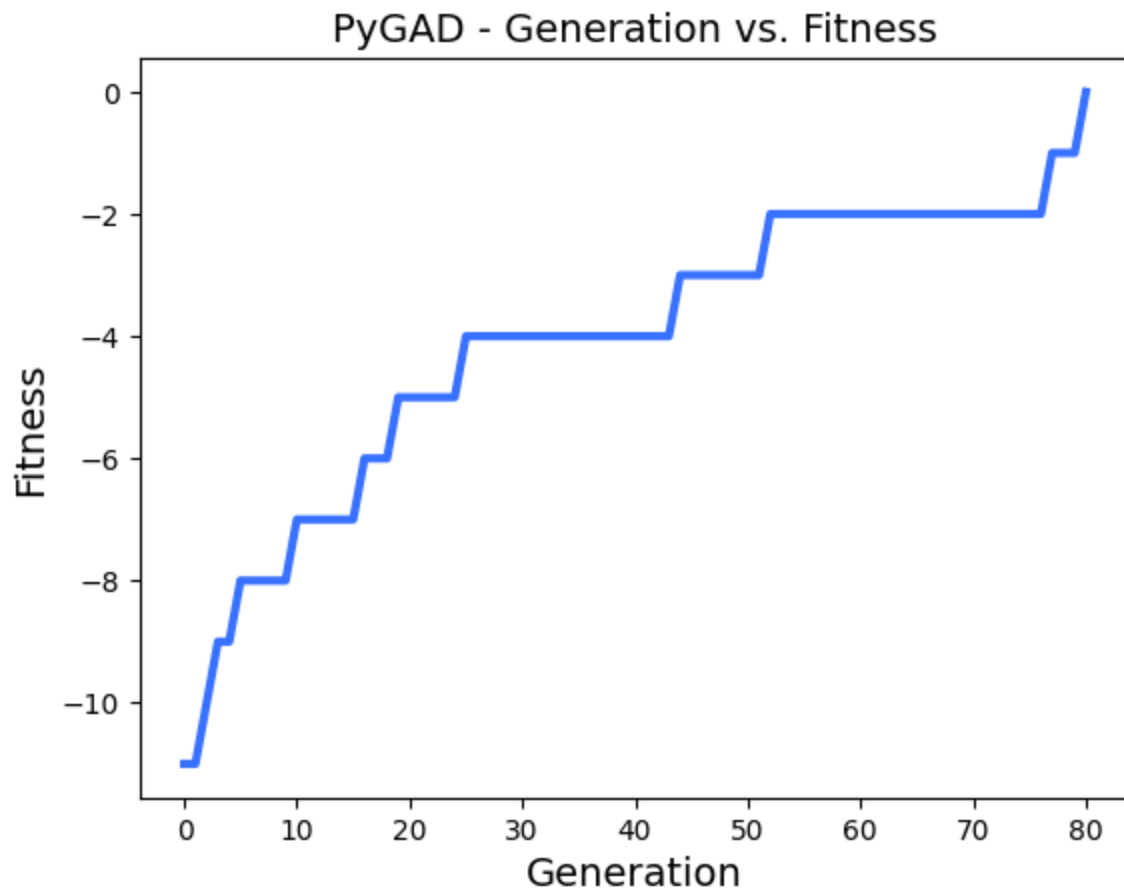
```
solution, solution_fitness, solution_idx = ga_instance.best_solution()
```

```
printResult(solution, solution_fitness)
```

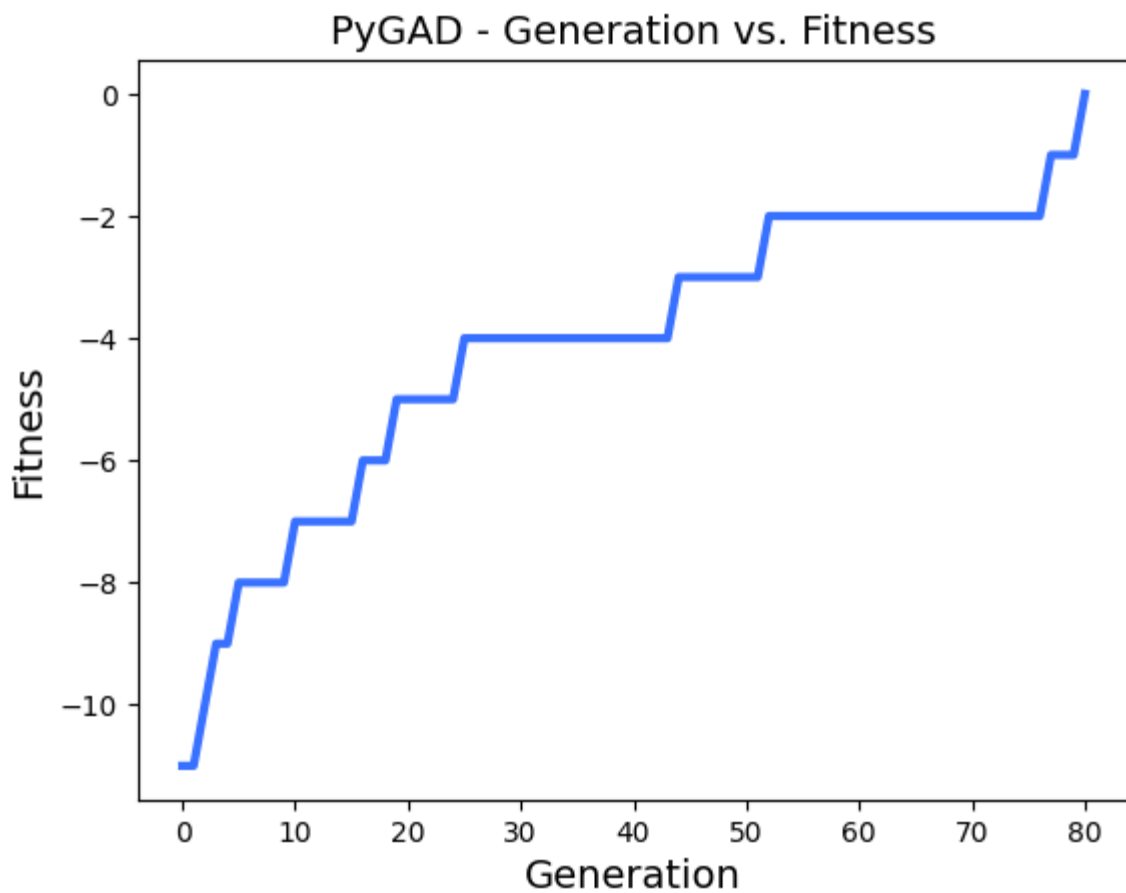
```
ga_instance.plot_fitness()
```

Parameters of the best solution : [13 16 0 12 7 17 6 11 1 14 4 18 9 3 20 10 1
5 19 2 5 8]

Fitness value of the best solution = 0



Out[173]:



Tests

100-sample tests for our algorithm. They check how many percent our algorithm solves well and the average search time.

Small input test

We conducted the first test for $n = 5$. Our algorithm finds 100% of solutions in an average time of approximately 0.06 seconds. He succeeds in an average of about 1 generation.

```
In [16]: n = 5

times = []
generations = []
solved = 0
for i in range(100):
    start = time.time()
    ga_instance = ga_instance_generate(n)

    ga_instance.run()
    end = time.time()
    times.append(end - start)
    generations.append(ga_instance.generations_completed)
    solution, solution_fitness, solution_idx = ga_instance.best_solution()

    if solution_fitness == 0:
        solved += 1

printAverages(times, generations, solved)
```

Average time (sec): 0.06435400247573853

Average number of generations 1.04

GA solved 100/100

Medium input test

We conducted another test for $n = 10$. Our algorithm finds 86% of solutions in an average time of approximately 7 seconds. It manages to do so on average 100 generations.

```
In [207... n = 10

times = []
generations = []
solved = 0
for i in range(100):
    start = time.time()
    ga_instance = ga_instance_generate(n)

    ga_instance.run()
    end = time.time()
    times.append(end - start)
    generations.append(ga_instance.generations_completed)
    solution, solution_fitness, solution_idx = ga_instance.best_solution()

    if solution_fitness == 0:
        solved += 1

printAverages(times, generations, solved)
```

Average time (sec): 7.108049712181091

Average number of generations 101.08

GA solved 86/100

Big input test

We conducted the last test for $n = 16$. Our algorithm finds 53% of solutions in an average time of approximately 28 seconds. He succeeds in an average of about 188 generations.

In [208...

```
n = 16

times = []
generations = []
solved = 0
for i in range(100):
    start = time.time()
    ga_instance = ga_instance_generate(n)

    ga_instance.run()
    end = time.time()
    times.append(end - start)
    generations.append(ga_instance.generations_completed)
    solution, solution_fitness, solution_idx = ga_instance.best_solution()

    if solution_fitness == 0:
        solved += 1

printAverages(times, generations, solved)
```

Average time (sec): 28.87527069091797

Average number of generations 188.19

GA solved 53/100

PSO

Fitness and helper functions

The fitness function is exactly the same function that was used for the genetic algorithm (it uses the same helper functions), but it does not accept 2 parameters, but only a solution table, which we convert to integer at the very beginning, because the pyswarms package uses floating point numbers.

The function `f` is a helper function that inserts only one table of solutions from the table of solutions into the fitness function. We use "-" before the fitness function to find the maximum.

The `generate_optimizer` function is the generate function of the pso optimizer. We set in it:

- `options`, that is, a dictionary containing the parameters for the specific optimization technique. We use the default
- `bounds`, i.e., a tuple of size 2 where the first entry is the minimum bound while the second entry is the maximum bound, for us minimum bound is an array of `n` lengths of all zeros, and maximum bound is an array of length `n` with only `n` values,
- `n_particles`, i.e. the number of particles in the swarm. We use 20,
- `dimensions`, i.e. number of dimensions in the space. We use `n`.

```
In [8]: def fitness_func(solution):
    newSol = solution.astype(int)
    attacks = 0
    board = fulfillBoard(newSol)

    for i in range(len(board)):
        for j in range(len(board[i])):
            if board[i][j] == 1:
                if checkColission(i, j, board):
                    attacks += 1

    board = createBoard(n)
    return (attacks * -1)

def f(x):
    n_particles = x.shape[0]
    j = [-fitness_func(x[i]) for i in range(n_particles)]
    return np.array(j).astype(int)

def generate_optimizer(n):
    options = {'c1': 0.5, 'c2': 0.3, 'w': 0.9}

    x_max = np.full((n,), n)
    x_min = np.zeros(n)
    my_bounds = (x_min, x_max)

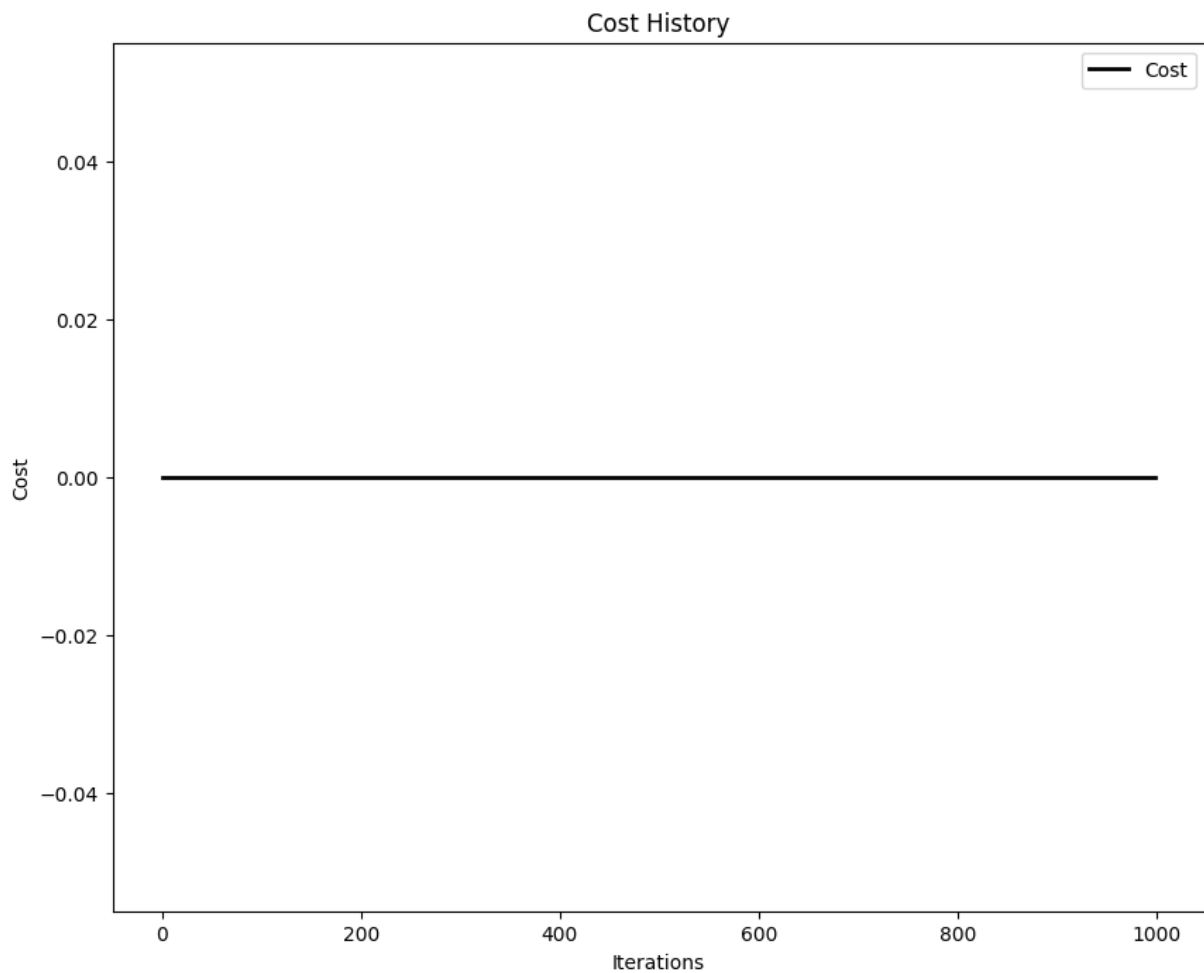
    return ps.single.GlobalBestPSO(n_particles=20, dimensions=n,
    options=options, bounds=my_bounds)
```

Small input no. 1

The first input is $n = 1$. Our algorithm handled it without any problem.

```
In [9]: n = 1
optimizer = generate_optimizer(n)
optimizer.optimize(f, iters=1000)
cost_history = optimizer.cost_history
plot_cost_history(cost_history)
plt.show()
```

```
2023-04-07 18:59:32,361 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 18:59:34,060 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.46826359]
```

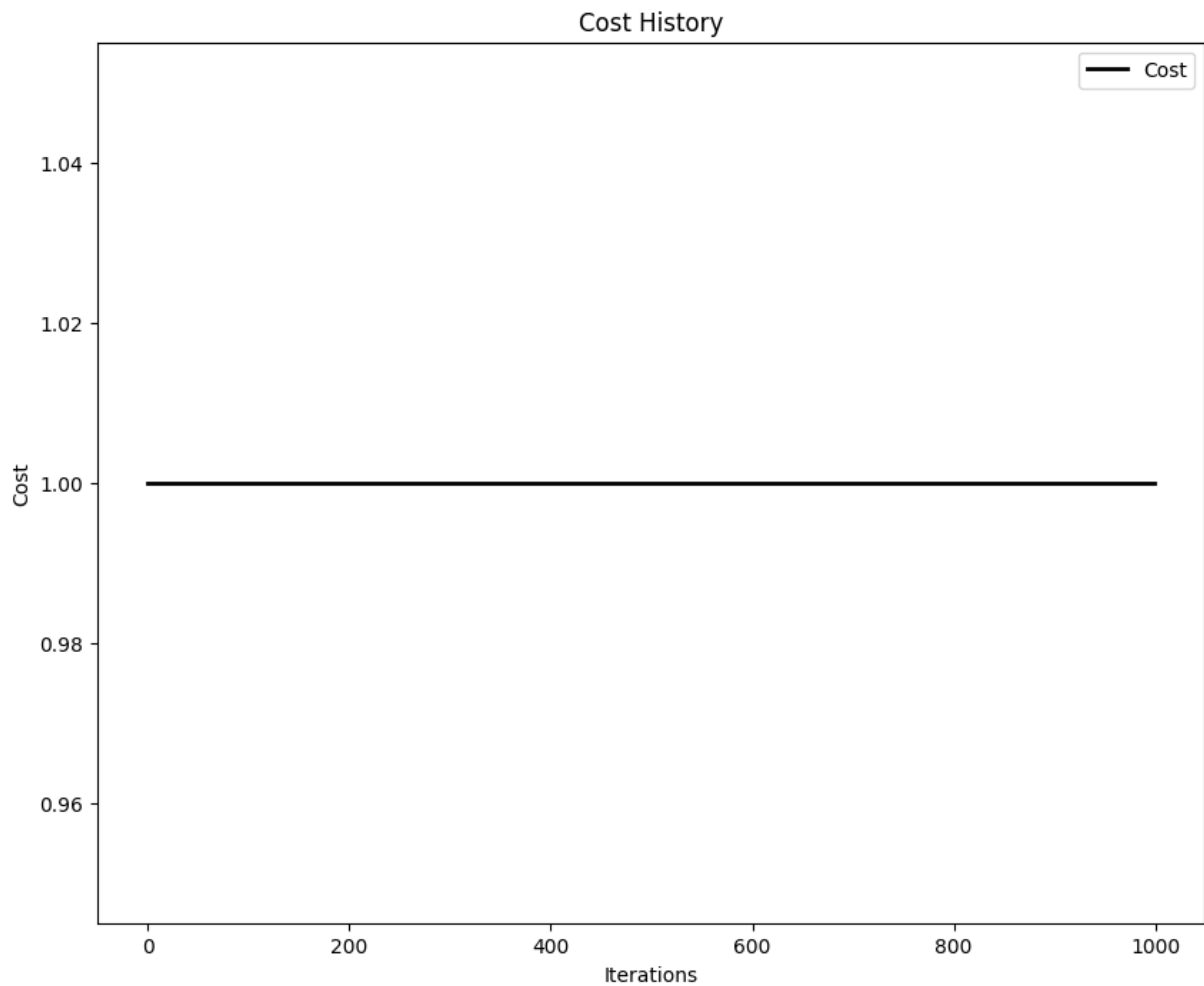


Small input no. 2

The next input is $n = 2$, which is an exception. Our algorithm didn't solve it, which is a good sign.

```
In [10]: n = 2
optimizer = generate_optimizer(n)
optimizer.optimize(f, iters=1000)
cost_history = optimizer.cost_history
plot_cost_history(cost_history)
plt.show()
```

```
2023-04-07 18:59:37,798 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 18:59:39,493 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.53409238 0.07243081]
```

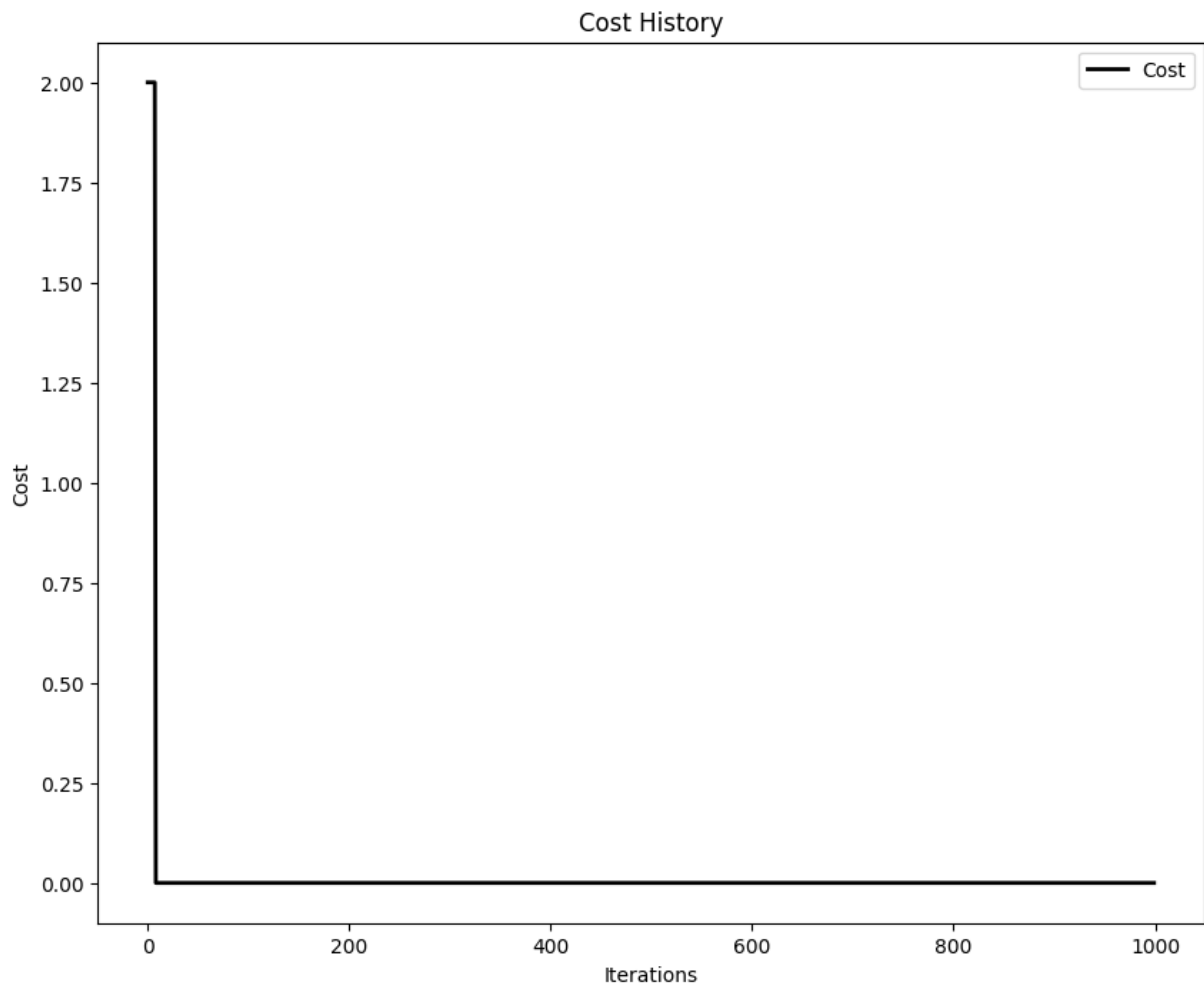



Small input no. 3

The last small input is $n = 5$. Our algorithm quickly found a good solution.

```
In [11]: n = 5
optimizer = generate_optimizer(n)
optimizer.optimize(f, iters=1000)
cost_history = optimizer.cost_history
plot_cost_history(cost_history)
plt.show()
```

```
2023-04-07 18:59:40,385 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 18:59:42,620 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.76119335 3.11512474 1.00035776 4.79005471 2.83803338]
```

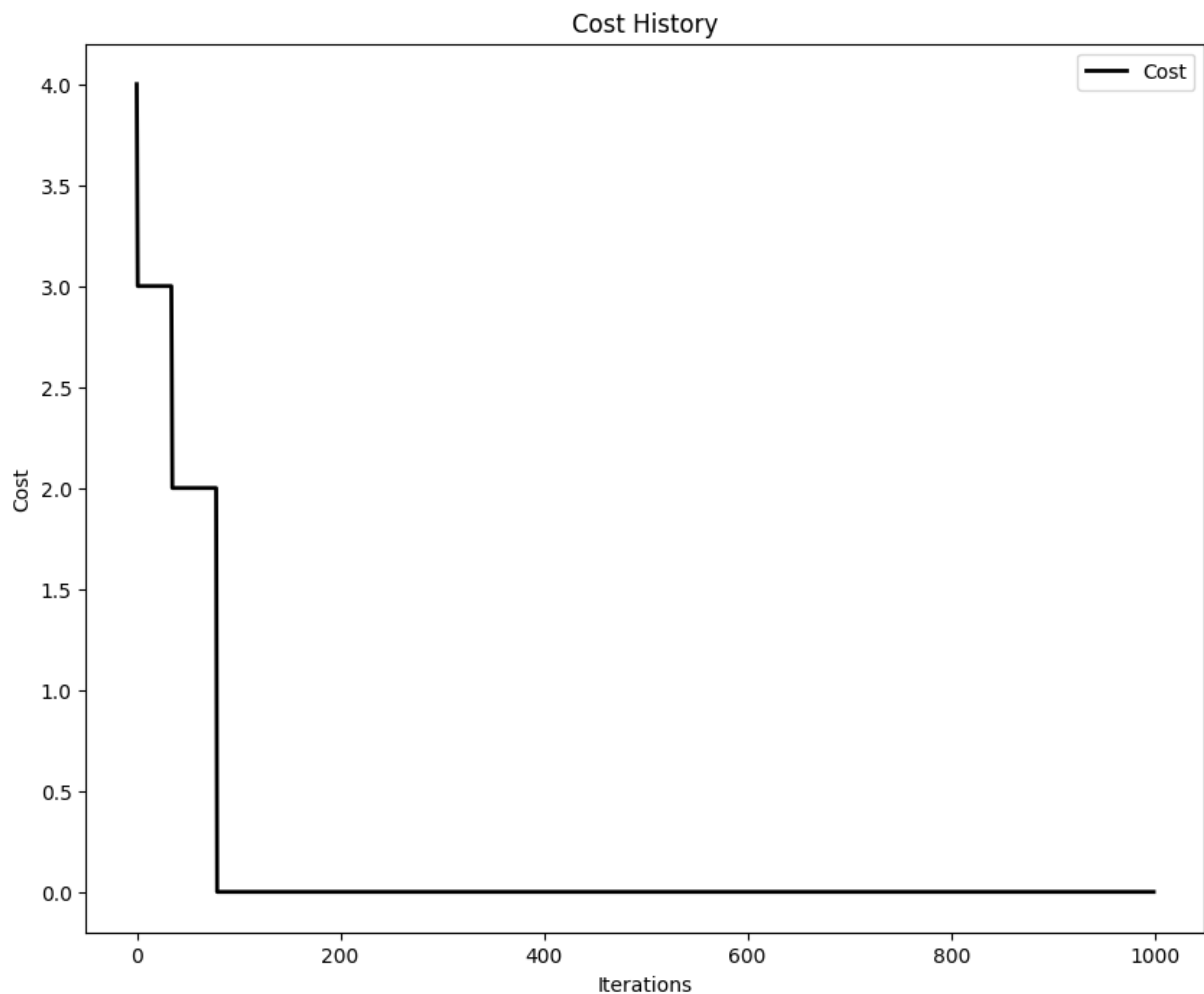


Medium input no. 1

The fourth input is $n = 7$. Our algorithm still works and will find a solution after about 100 iterations.

```
In [25]: n = 7
optimizer = generate_optimizer(n)
optimizer.optimize(f, iters=1000)
cost_history = optimizer.cost_history
plot_cost_history(cost_history)
plt.show()
```

```
2023-04-07 19:00:46,626 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:00:49,254 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.895432  5.02931994 1.2107059  4.83211181 0.85698802
3.59071043
6.20367047]
```

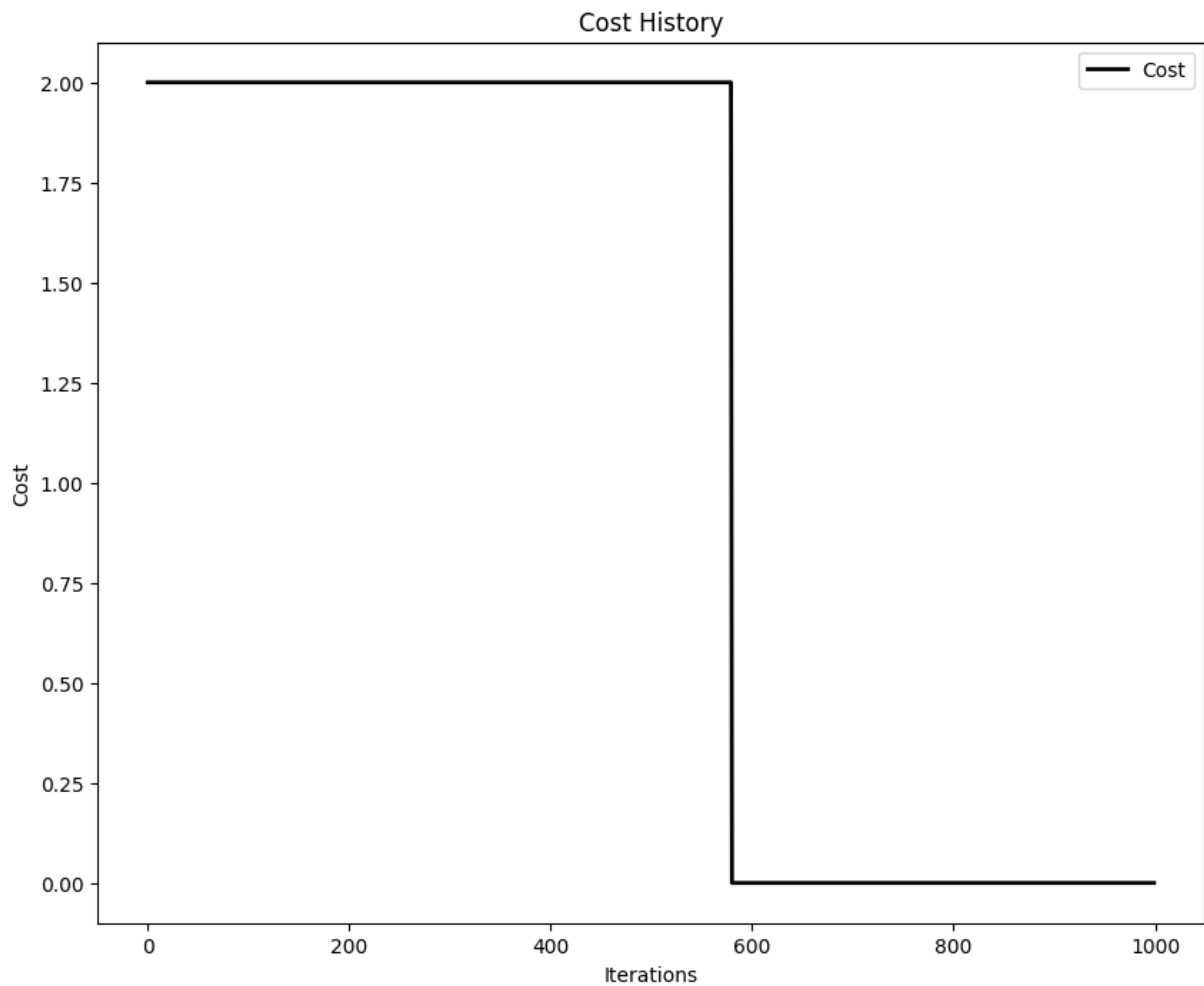


Medium input no. 2

The next input is $n = 8$. Our algorithm still finds a solution, but the graph shows that it takes many more iterations.

```
In [24]: n = 8
optimizer = generate_optimizer(n)
optimizer.optimize(f, iters=1000)
cost_history = optimizer.cost_history
plot_cost_history(cost_history)
plt.show()
```

```
2023-04-07 19:00:39,749 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:00:42,703 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [6.77255121 2.86051894 7.09846247 1.58970394 4.1967417
0.24649502
5.4041038 3.08400947]
```

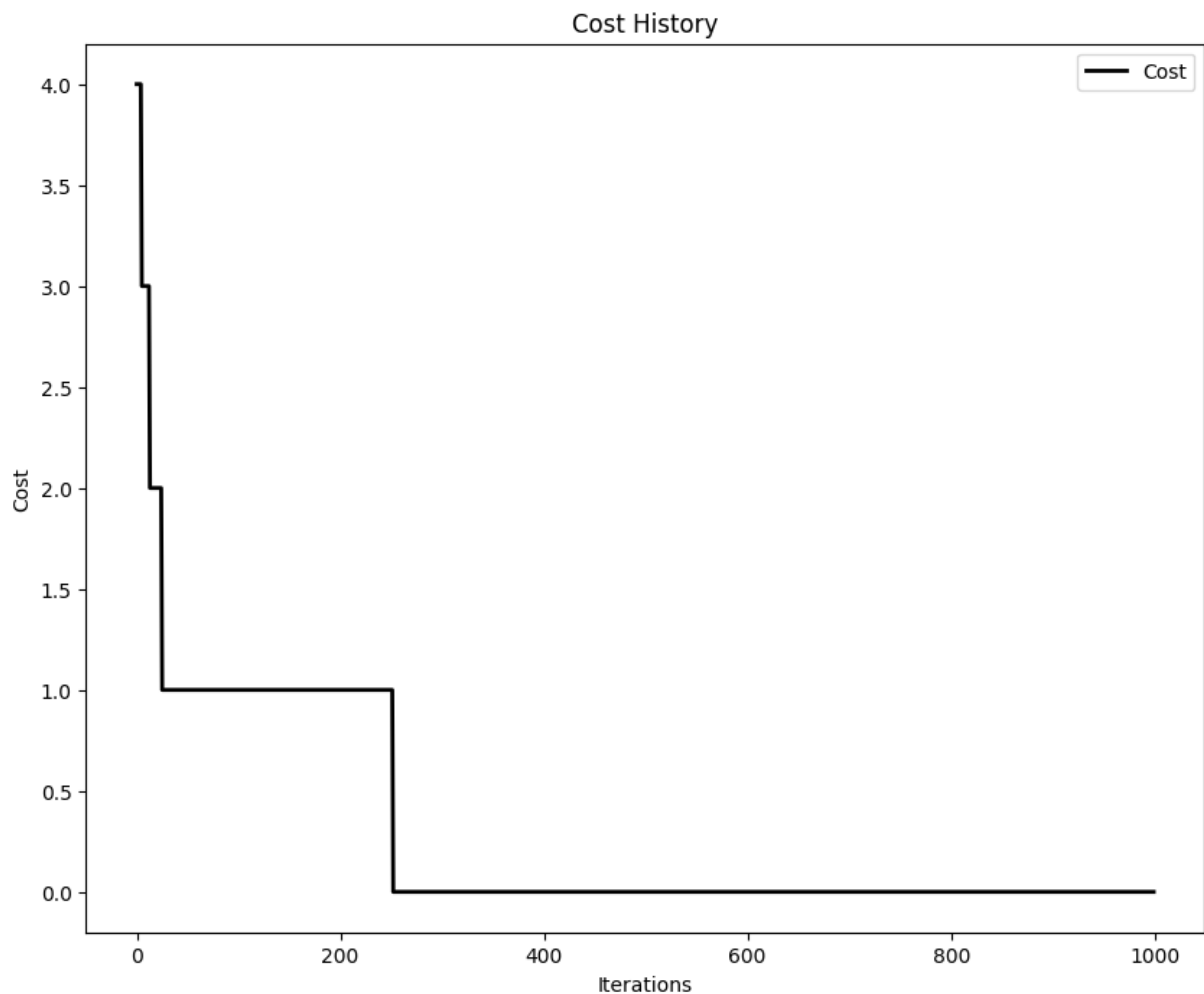


Medium input no. 3

The last medium input is $n = 9$. The algorithm still finds a solution, even faster than for $n = 8$, which is of course random. However, you can see the differences and the first suspicions that pso will not solve our problem as well as ag.

```
In [35]: n = 9
optimizer = generate_optimizer(n)
optimizer.optimize(f, iters=1000)
cost_history = optimizer.cost_history
plot_cost_history(cost_history)
plt.show()
```

```
2023-04-07 19:01:36,892 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:01:41,341 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.53697515 5.67814771 0.43172223 2.71117228 6.58373559
8.32451062
3.63290102 7.12193594 4.45218577]
```

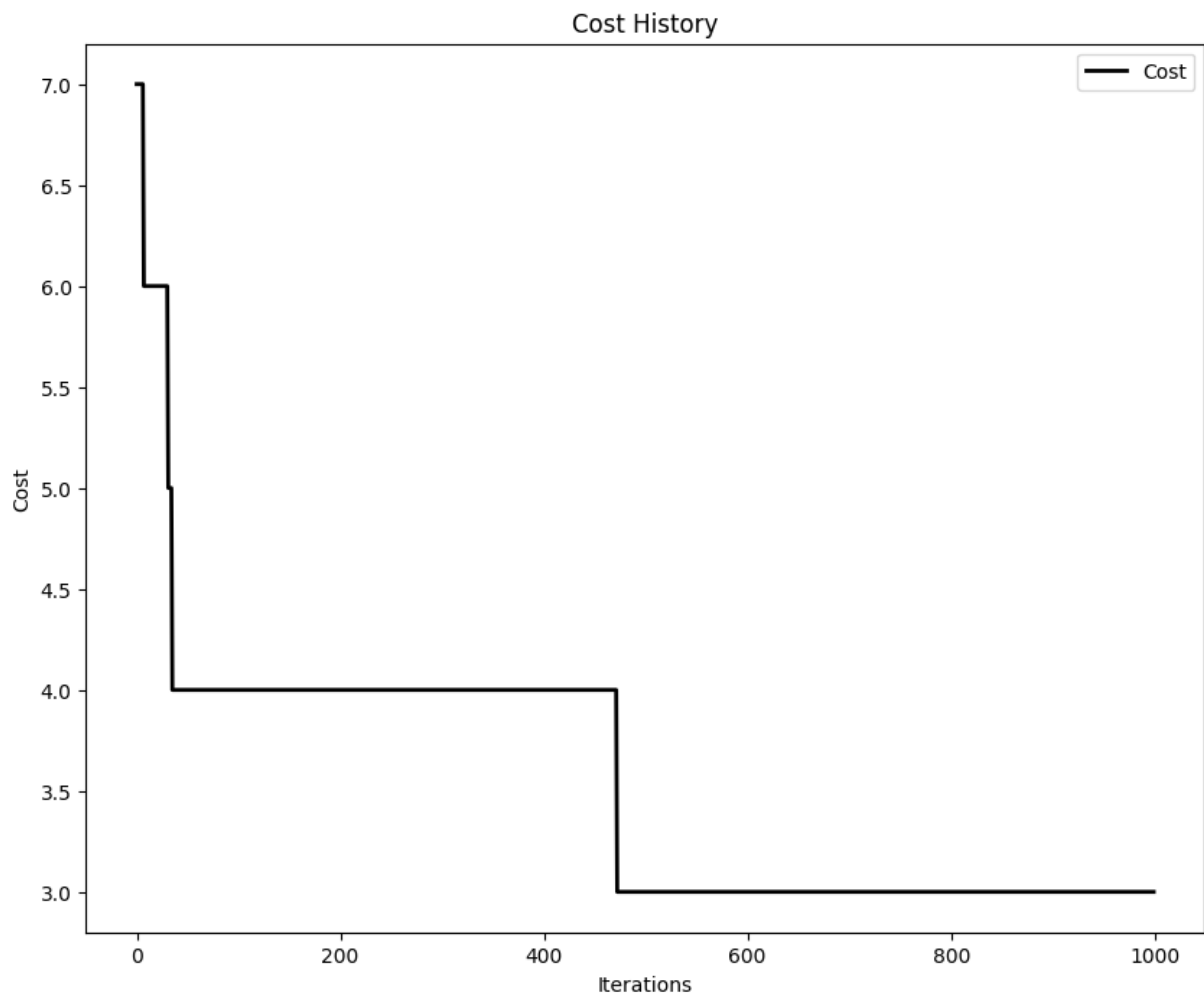


Big input no. 1, 2, 3

Unfortunately, our algorithm cannot cope with any of the larger inputs, i.e. $n = 12$, $n = 13$, $n = 15$. You can see that it is much weaker than our genetic algorithm.

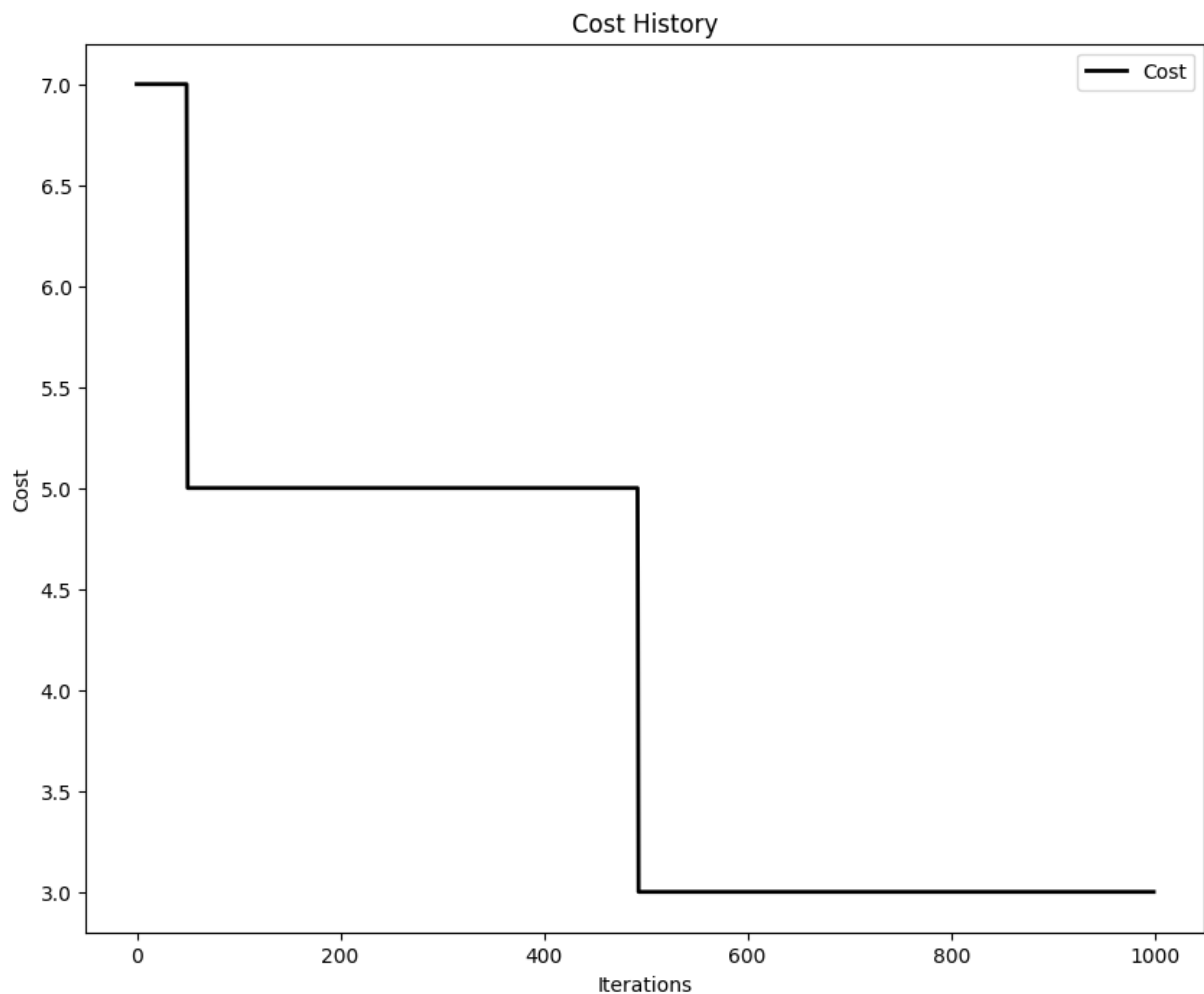
```
In [36]: n = 12
optimizer = generate_optimizer(n)
optimizer.optimize(f, iters=1000)
cost_history = optimizer.cost_history
plot_cost_history(cost_history)
plt.show()
```

```
2023-04-07 19:01:44,901 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:01:49,947 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 3.57774921  1.26970425  4.87955127  2.41673486  9.8226
4245  6.67858761
      8.04597854  5.09612814  8.30725383  0.64397957  7.28277332 11.4139461 ]
```



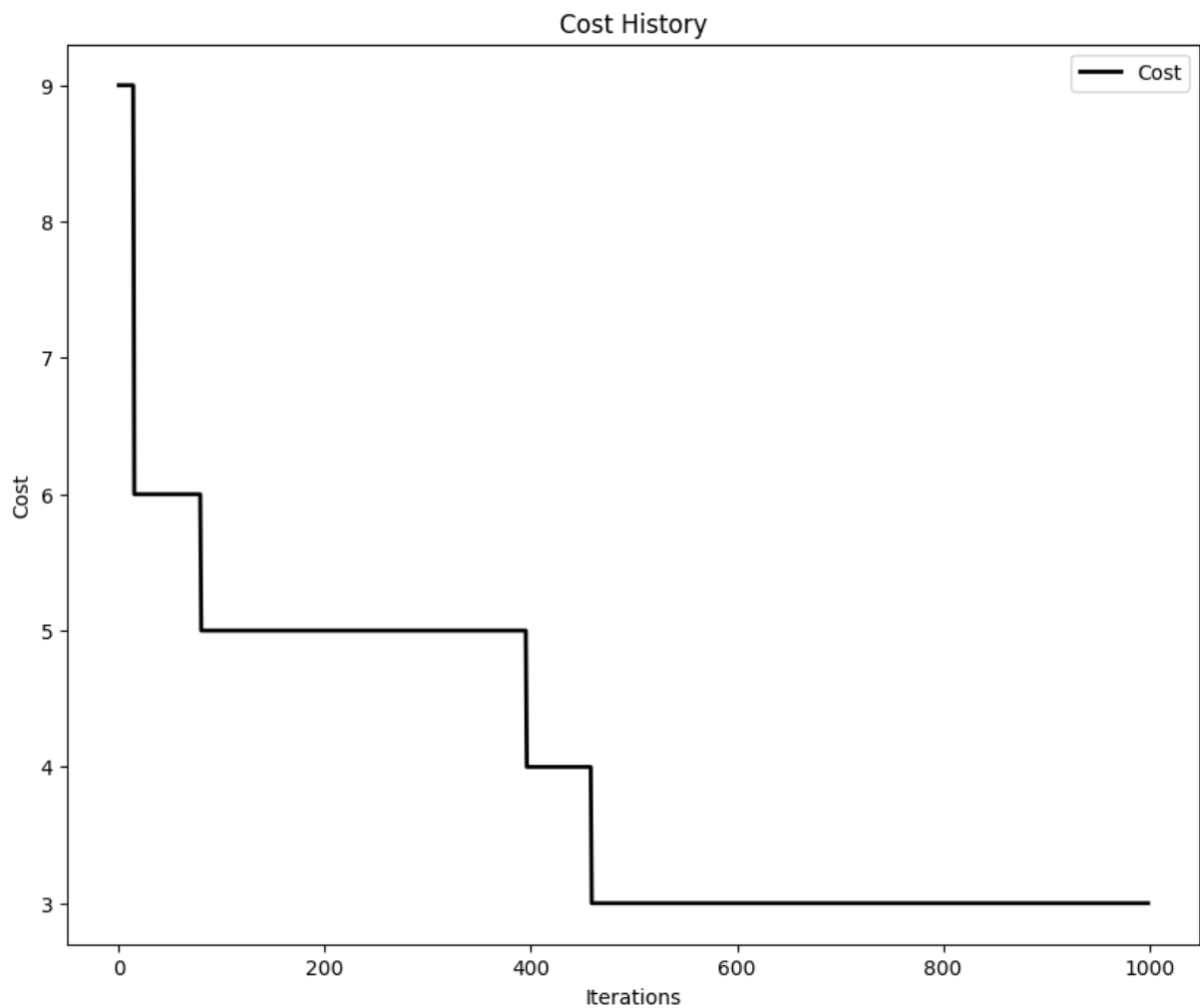
```
In [37]: n = 13
optimizer = generate_optimizer(n)
optimizer.optimize(f, iters=1000)
cost_history = optimizer.cost_history
plot_cost_history(cost_history)
plt.show()
```

```
2023-04-07 19:01:54,214 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:02:00,041 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 0.69465295  9.5742524  1.53805862  8.52594985 11.0208
2886  3.52890511
 6.87060737  6.36828665  6.79102563  5.61232019  7.85129201 12.15078854
 4.38851284]
```



```
In [38]: n = 15
optimizer = generate_optimizer(n)
optimizer.optimize(f, iters=1000)
cost_history = optimizer.cost_history
plot_cost_history(cost_history)
plt.show()
```

```
2023-04-07 19:02:01,071 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:02:08,319 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 6.49131674  1.83190743  7.81132278  5.10537628 11.4901
7428  8.44109654
  4.74530325 12.08987684  4.91689559  3.24225179  9.98265369  0.69216227
13.1155534 10.05134555  2.33323626]
```



Helper functions

The `printAverages` function is a function that presents the results in a nice way.

```
In [39]: def printAverages(times, solved):  
         average_time = 0  
         for time in times:  
             average_time += time  
         average_time = average_time / len(times)  
         print("Average time (sec): ", average_time)  
  
         print(f"PSO solved {solved}/100")
```

Tests

100-sample tests for our algorithm. They check how many percent our algorithm solves well and the average search time.

Small input test

We conducted the first test for $n = 5$. Our algorithm finds 100% of solutions in an average time of approximately 0.6 seconds.

```
In [40]: n = 5

times = []
generations = []
solved = 0
for i in range(100):
    start = time.time()
    optimizer = generate_optimizer(n)

    cost, _ = optimizer.optimize(f, iters=200)
    end = time.time()
    times.append(end - start)

    if cost == 0.0:
        solved += 1

printAverages(times, solved)
```

```
2023-04-07 19:02:17,164 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:17,624 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [0.85236647 3.85141042 1.29134466 4.03404249 2.74879426]
2023-04-07 19:02:17,634 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:18,106 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [3.96252073 0.63449856 2.76579545 4.10722106 1.29739636]
2023-04-07 19:02:18,113 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:18,570 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [2.88995472 0.42300005 3.13663859 1.20927655 4.06587103]
2023-04-07 19:02:18,580 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:19,080 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [2.21276326 4.10702545 1.35366121 3.64176932 0.29932118]
2023-04-07 19:02:19,093 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:19,565 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [1.17723345 4.12323593 2.29477356 0.67358865 3.0847707 ]
2023-04-07 19:02:19,576 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:20,012 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [4.71952158 2.88535787 0.93120876 3.38480521 1.89549359]
2023-04-07 19:02:20,020 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:20,619 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [1.9032963 4.96523137 2.52061642 0.87711525 3.32460981]
2023-04-07 19:02:20,628 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:21,238 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [2.16758759 0.97378848 3.68016853 1.51218649 4.05907852]
2023-04-07 19:02:21,252 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:21,846 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [2.99917544 4.31077751 1.41920817 3.46256836 0.55714906]
2023-04-07 19:02:21,861 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:22,432 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [4.17173304 2.76692574 0.43649765 3.03056071 1.63751153]
2023-04-07 19:02:22,441 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:23,054 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [0.92412885 2.12028962 4.62214242 1.6639588 3.79561906]
2023-04-07 19:02:23,065 - pyswarms.single.global_best - INFO - Optimize for 200 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
```

```
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:23,631 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.95417626 2.48894981 4.16354812 1.94875825 3.28902833]
2023-04-07 19:02:23,642 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:24,224 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.25958631 1.69178409 3.93038304 0.39672956 2.21473983]
2023-04-07 19:02:24,235 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:24,863 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.98653328 4.26281206 2.8860963 0.56940308 3.34065065]
2023-04-07 19:02:24,875 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:25,464 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.972503 4.14897257 2.77418724 0.86814696 3.85875377]
2023-04-07 19:02:25,475 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:26,085 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.59676468 3.25787994 1.52949122 4.52003914 2.31291127]
2023-04-07 19:02:26,100 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:26,694 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.26816601 2.49859054 4.29971036 1.97298731 3.53041292]
2023-04-07 19:02:26,708 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:27,301 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.0254491 1.48481218 3.435301 0.54416874 2.56953184]
2023-04-07 19:02:27,310 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:27,876 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.70847478 0.84110983 2.81447828 4.28106677 1.03245444]
2023-04-07 19:02:27,886 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:28,440 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.77973544 3.04920281 0.51356306 2.09356815 4.2835203 ]
2023-04-07 19:02:28,450 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:29,023 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.8830628 3.70149566 0.16489859 2.26844206 4.04662831]
2023-04-07 19:02:29,034 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:29,639 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.59867519 1.83267565 3.88184376 0.98968033 2.46211615]
2023-04-07 19:02:29,651 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
```

```
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:30,284 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.72270022 1.64653656 4.61758516 2.95361087 0.25290249]
2023-04-07 19:02:30,296 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:30,864 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.29142603 0.8677796 2.15289679 4.94444775 1.01984754]
2023-04-07 19:02:30,875 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:31,507 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.05408957 4.24106964 1.02257016 3.97738758 0.86337523]
2023-04-07 19:02:31,516 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:32,089 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.04958311 4.06660249 1.80497761 3.5589957 0.63154216]
2023-04-07 19:02:32,103 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:32,656 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.4047077 0.90677996 2.59285413 4.45559941 1.6150751 ]
2023-04-07 19:02:32,665 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:33,258 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.6916108 4.13420136 2.10487151 0.91856441 3.19408378]
2023-04-07 19:02:33,267 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:33,839 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.48332651 3.45209731 0.66216969 2.96089659 4.29579692]
2023-04-07 19:02:33,850 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:34,434 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.18896655 1.19246242 3.17799614 0.50137444 2.45679356]
2023-04-07 19:02:34,447 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:35,095 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.30246462 1.63611548 4.93431037 2.4172171 0.27463446]
2023-04-07 19:02:35,108 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:35,665 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.68677431 3.22249162 0.15742079 2.21836239 4.71823713]
2023-04-07 19:02:35,674 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:36,288 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.18086169 3.04331564 0.91344314 2.15147564 4.20257734]
2023-04-07 19:02:36,299 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
```

```
2023-04-07 19:02:36,867 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.7091874 0.82852766 2.04406071 4.87886118 1.39610071]
2023-04-07 19:02:36,877 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:37,444 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.27832732 2.27448886 0.90221748 3.85577342 1.74844106]
2023-04-07 19:02:37,454 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:38,052 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.66044326 0.70336066 3.81522016 1.00020994 4.11348239]
2023-04-07 19:02:38,065 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:38,653 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.81938099 0.85117655 3.67184457 1.87445198 4.02847582]
2023-04-07 19:02:38,664 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:39,226 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.07766783 1.64587711 3.85385384 0.97136712 2.17954396]
2023-04-07 19:02:39,239 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:39,896 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.6516907 3.00663232 1.73521982 4.44057797 2.75582437]
2023-04-07 19:02:39,908 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:40,487 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.94223783 2.04998336 4.53041497 1.67317576 3.26424149]
2023-04-07 19:02:40,497 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:41,093 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.07631884 1.86823141 4.40197089 2.0722508 0.60165268]
2023-04-07 19:02:41,102 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:41,703 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.56679437 1.372647 3.66528307 0.07619857 2.21826412]
2023-04-07 19:02:41,714 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:42,249 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.95602352 0.65571954 2.72597236 4.74627344 1.82701972]
2023-04-07 19:02:42,263 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:42,905 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.49449806 2.00743925 0.5973036 3.64201506 1.43728329]
2023-04-07 19:02:42,914 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:43,494 - pyswarms.single.global_best - INFO - Optimization finished
```

```
| best cost: 0.0, best pos: [4.68703044 2.48862894 0.33410352 3.78930876 1.74892153]
2023-04-07 19:02:43,513 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:44,175 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.46335216 1.41948522 3.58880897 0.51578029 2.92317319]
2023-04-07 19:02:44,188 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:44,886 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.30422475 0.84937392 3.25136805 1.26107454 4.18102939]
2023-04-07 19:02:44,897 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:45,517 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.17459598 1.69857002 4.05992177 2.92360426 0.96972884]
2023-04-07 19:02:45,536 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:46,188 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.96992965 3.59974628 1.79474468 4.22033731 2.94367768]
2023-04-07 19:02:46,197 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:46,836 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.77888441 0.58828494 3.76801845 1.73255997 4.72206981]
2023-04-07 19:02:46,849 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:47,487 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.81397839 4.47189551 1.74841898 3.89992771 0.11630359]
2023-04-07 19:02:47,498 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:48,062 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.64884816 1.62345308 3.54943318 0.37738782 2.54430025]
2023-04-07 19:02:48,071 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:48,648 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.8841396 3.98318521 0.04607079 2.75906111 4.85819741]
2023-04-07 19:02:48,659 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:49,269 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.7179182 1.51410672 3.55566711 0.23541472 2.99100955]
2023-04-07 19:02:49,279 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:49,922 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.2493845 4.34233042 2.79732014 0.87331562 3.28910409]
2023-04-07 19:02:49,936 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:50,533 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.77053956 4.37445441 2.90236391 0.72118256 3.08844413]
```

```
2023-04-07 19:02:50,546 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:51,179 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.40718849 0.93128212 2.48830904 4.85901464 1.97179309]
2023-04-07 19:02:51,190 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:51,761 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.83303791 3.26227304 1.58756146 4.64600545 2.14428932]
2023-04-07 19:02:51,771 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:52,373 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.6985507 0.918249 3.30947208 1.28802584 4.37030391]
2023-04-07 19:02:52,383 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:52,962 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.75397409 3.51513844 1.06651449 4.64930943 2.11363937]
2023-04-07 19:02:52,971 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:53,584 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.27782366 4.43582174 2.90020861 0.74352186 3.34115122]
2023-04-07 19:02:53,596 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:54,194 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.02244829 1.62249622 3.82978081 0.80046288 2.41502934]
2023-04-07 19:02:54,205 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:54,871 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.39771742 0.93305733 3.62715054 1.07240566 4.29688896]
2023-04-07 19:02:54,889 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:55,511 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.22807368 4.14664129 2.68392022 0.58883317 3.48097835]
2023-04-07 19:02:55,523 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:56,145 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.46549821 3.02484665 1.2145438 4.63101693 2.07238842]
2023-04-07 19:02:56,160 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:56,805 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.92370967 0.67068632 2.79402214 4.23437639 1.74516374]
2023-04-07 19:02:56,815 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:57,412 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.00526019 2.53233628 0.69168232 3.44854706 1.96472624]
2023-04-07 19:02:57,425 - pyswarms.single.global_best - INFO - Optimize for 200 iter
```

```
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:58,051 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.45162599 3.56709121 0.21875094 2.16415482 4.10882605]
2023-04-07 19:02:58,063 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:58,622 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.11512359 1.72365391 4.48573435 2.99682744 0.89564708]
2023-04-07 19:02:58,636 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:59,242 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.96518135 3.93650526 1.75946575 4.41338501 2.56071039]
2023-04-07 19:02:59,256 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:02:59,914 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.59112015 0.90404985 2.79869025 4.60664219 1.48032614]
2023-04-07 19:02:59,934 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:00,543 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.91648386 0.97867106 3.09637385 1.79704328 4.31546055]
2023-04-07 19:03:00,554 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:01,159 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.93040678 2.07220453 4.69408207 1.82855083 3.35140168]
2023-04-07 19:03:01,171 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:01,802 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.03511365 3.86460943 1.86459411 4.61278605 2.47708731]
2023-04-07 19:03:01,818 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:02,414 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.55472127 1.38423469 4.39399635 2.39224068 0.07575977]
2023-04-07 19:03:02,424 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:03,081 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.7365704 0.00868468 3.40166569 1.30210295 4.00970081]
2023-04-07 19:03:03,097 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:03,662 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.93318906 0.43498016 3.18857316 1.57008899 4.11064277]
2023-04-07 19:03:03,676 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:04,287 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.88536379 3.27266771 1.74738163 4.71321198 2.86282695]
2023-04-07 19:03:04,296 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
```



```
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:04,996 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.56728766 2.69064633 0.49700316 3.67056487 1.04393834]
2023-04-07 19:03:05,007 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:05,636 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.44013819 3.44691621 0.57030532 2.74366327 4.87709368]
2023-04-07 19:03:05,648 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:06,320 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.62520628 4.25923232 1.04290744 3.25386234 0.67463293]
2023-04-07 19:03:06,331 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:06,911 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [0.47312124 2.57640184 4.59996611 1.00239719 3.91917921]
2023-04-07 19:03:06,925 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:07,530 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.30646413 0.16026324 3.32754549 1.25901656 4.80882986]
2023-04-07 19:03:07,541 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:08,104 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.40542832 0.61375439 3.46879738 1.84073195 4.52062041]
2023-04-07 19:03:08,114 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:08,718 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.25553248 1.31445841 4.45233945 2.99666078 0.66190571]
2023-04-07 19:03:08,730 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:09,348 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.02630323 1.00244015 3.68621634 0.84842395 2.90785471]
2023-04-07 19:03:09,360 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:10,088 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.899053 1.85073084 3.98279648 0.66971683 2.92373828]
2023-04-07 19:03:10,101 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:10,694 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.47411028 2.82440481 0.6816633 3.93145425 1.80711301]
2023-04-07 19:03:10,704 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:11,356 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.29328662 0.92935778 2.57326857 4.85196041 1.69746049]
2023-04-07 19:03:11,366 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
```

```
2023-04-07 19:03:11,976 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.77322532 0.94081799 2.84716556 4.30442491 1.00348874]
2023-04-07 19:03:11,989 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:12,571 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.848942 3.94530313 0.32841258 2.80917226 4.80579179]
2023-04-07 19:03:12,583 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:13,192 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.50600468 4.24318846 2.3678587 0.50060038 3.4903343 ]
2023-04-07 19:03:13,204 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:13,789 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.0695199 1.64218927 3.08775316 0.65401137 2.17554199]
2023-04-07 19:03:13,802 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:14,457 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.31454023 2.25818475 0.56125591 3.90170637 1.48774813]
2023-04-07 19:03:14,468 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:15,182 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.34572402 4.1592138 2.11204108 0.5513207 3.21139568]
2023-04-07 19:03:15,195 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:15,800 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.14112214 1.0113389 4.05840512 2.47675813 0.55272783]
2023-04-07 19:03:15,813 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:16,468 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.65625124 3.07556455 0.63388934 2.07404786 4.12909681]
2023-04-07 19:03:16,478 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:17,084 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.7255997 2.08064379 0.86680991 3.56093771 1.91701245]
2023-04-07 19:03:17,098 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:17,725 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.24630634 4.19440607 1.49660366 3.27828842 0.98075344]
2023-04-07 19:03:17,736 - pyswarms.single.global_best - INFO - Optimize for 200 iter
s with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|200/200, best_cost=0
2023-04-07 19:03:18,368 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.85591534 3.95695614 0.95109768 2.75439816 4.22447162]
Average time (sec): 0.612137746810913
PSO solved 100/100
```

Medium input test

We conducted another test for $n = 8$. Our algorithm finds only 21% of solutions in an average time of approximately 4.3 seconds.

```
In [41]: n = 8

times = []
generations = []
solved = 0
for i in range(100):
    start = time.time()
    optimizer = generate_optimizer(n)

    cost, _ = optimizer.optimize(f, iters=1000)
    end = time.time()
    times.append(end - start)

    if cost == 0.0:
        solved += 1

printAverages(times, solved)
```

```
2023-04-07 19:04:19,801 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:04:22,773 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.81532313 7.2643764 5.24512018 1.14813623 6.10212341
4.33596176
0.74134511 3.84174222]
2023-04-07 19:04:22,781 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:04:25,728 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.21599974 5.40675675 7.62135484 4.27740569 0.52283499
3.80740864
6.61998492 1.99046952]
2023-04-07 19:04:25,741 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:04:29,098 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [1.05302063 4.10640642 7.78079751 3.8417478 6.19656487
6.06651209
2.29664402 5.03650249]
2023-04-07 19:04:29,108 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:04:33,516 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [0.71837822 5.09107863 3.25646768 6.02603186 2.29904757
5.45275216
1.15711177 4.31149373]
2023-04-07 19:04:33,526 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:04:37,914 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [7.32832002 2.65695762 4.45013652 1.22358925 6.66237015
0.35462512
5.00605969 3.65633344]
2023-04-07 19:04:37,924 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:04:42,295 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.74508544 7.30023512 5.1818035 2.46095951 6.61328633
3.05199026
0.22612542 4.2222247 ]
2023-04-07 19:04:42,309 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:04:46,665 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [5.17207612 0.02678724 4.98495921 1.77219156 3.88576466
6.77866051
2.43047041 7.46461899]
2023-04-07 19:04:46,676 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:04:51,067 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [4.93376323 2.24850099 7.17769443 3.81795525 6.23820793
1.9515911
5.78484056 0.87831128]
```

```
2023-04-07 19:04:51,081 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:04:55,347 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [0.26972288 7.12601674 4.54862542 1.50524136 6.482643
2.4001408
5.10519548 3.94433072]
2023-04-07 19:04:55,361 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:04:59,553 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.64242337 5.91828446 3.20161741 0.42740105 4.47884665
7.63330181
1.40970911 6.65839013]
2023-04-07 19:04:59,567 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:05:03,505 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [4.08483186 1.77310426 5.87327611 0.95029483 2.3388522
6.82022497
4.66590911 3.05279866]
2023-04-07 19:05:03,515 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:05:07,464 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [5.75919153 2.81485319 6.38565173 1.94307054 7.61163194
4.78381475
0.96378517 3.86703849]
2023-04-07 19:05:07,473 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:05:11,526 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [4.39339647 0.80635601 5.12365124 3.19744217 1.55573512
7.90739287
2.92178148 6.29381226]
2023-04-07 19:05:11,537 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:05:15,544 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.18772638 7.12279753 1.76338764 3.59808654 0.77924869
6.81854521
4.47098401 5.38571337]
2023-04-07 19:05:15,554 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:05:19,340 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.27218177 6.46231558 1.31864784 5.00602698 2.60533887
0.31501825
3.36857399 7.98525098]
2023-04-07 19:05:19,354 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:05:23,316 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.84409418 7.33094686 5.22840604 1.3925001 6.22993167
4.31272454
0.74244444 3.04972592]
```

```
2023-04-07 19:05:23,327 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:05:27,263 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [1.17457904 4.49366373 6.52451605 6.81940804 2.48964054
5.64082218
3.63754017 0.06668992]
2023-04-07 19:05:27,273 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:05:31,256 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.37208666 3.20752877 6.11297269 2.80378793 7.42056365
5.99759985
0.39586595 4.26990095]
2023-04-07 19:05:31,267 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:05:35,318 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [6.5227066 0.48508468 5.34426285 1.35909821 4.17852682
2.78912361
7.31196142 3.19006694]
2023-04-07 19:05:35,329 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:05:39,201 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [1.73799297 6.67643857 3.84340925 7.3988147 0.02244586
3.28158038
5.08308049 2.80296407]
2023-04-07 19:05:39,209 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:05:43,222 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [4.75529478 2.36447846 5.34184995 7.89730748 1.56204441
3.20614288
0.04075977 6.45161424]
2023-04-07 19:05:43,233 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:05:47,287 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [3.69237559 5.00513518 7.00005768 4.37647081 2.87403474
0.75182069
6.00855998 1.33128405]
2023-04-07 19:05:47,300 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:05:51,361 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.92999448 5.1733561 0.8356408 4.54228307 1.18369726
7.06393033
2.26973764 6.20424441]
2023-04-07 19:05:51,370 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:05:55,401 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [3.9330665 6.80096581 2.85236323 5.28460408 1.88140003
7.88293538
4.86063188 7.13387782]
```

```
2023-04-07 19:05:55,410 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:05:59,364 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.32481899 0.69500528 4.48440601 7.48218073 5.08616964
2.13795289
6.16646234 1.17904406]
2023-04-07 19:05:59,374 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:06:03,716 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [5.66170693 7.22470862 6.10904312 0.99839026 2.07529564
4.36151438
1.47334707 3.25606368]
2023-04-07 19:06:03,728 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:06:08,360 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [0.5409554 4.62004886 6.50168446 1.69500611 5.85137204
2.70491294
7.37187983 3.45040049]
2023-04-07 19:06:08,371 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:06:12,860 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [5.46086111 2.27692387 6.01018509 1.5871055 7.49288501
4.18623234
0.58308774 3.80380313]
2023-04-07 19:06:12,871 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:06:17,242 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.20278594 5.30692474 7.01454641 6.8772174 3.80077969
0.28776093
4.69557444 1.29319539]
2023-04-07 19:06:17,254 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:06:21,212 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [0.84373616 6.6900723 3.71570735 7.79461444 1.54999725
4.00821839
2.86800121 5.8158861 ]
2023-04-07 19:06:21,221 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:06:25,342 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.91978169 4.54567502 6.42970202 0.49437276 3.30787681
1.51030785
7.52485268 5.52727292]
2023-04-07 19:06:25,353 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:06:29,337 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [5.27829281 0.71463105 2.68449758 4.18236842 7.68142995
3.91821285
3.21812635 1.49476099]
```

```
2023-04-07 19:06:29,347 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:06:33,444 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [5.55227666 2.80519777 6.15569535 3.59558446 0.61929914
7.8196073
1.8027788 4.12407881]
2023-04-07 19:06:33,453 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:06:37,543 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [2.93311687 5.84418892 6.12393507 6.6755269 3.1998076
0.28226894
4.23315819 1.69262237]
2023-04-07 19:06:37,553 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:06:41,692 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [5.9087442 0.17410989 2.72658948 6.38181133 3.46761297
7.27714598
4.50477005 1.45125208]
2023-04-07 19:06:41,702 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:06:45,926 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [7.16907821 3.70987075 0.58729709 2.15859136 5.47334645
1.96264581
6.35686775 4.17143741]
2023-04-07 19:06:45,941 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:06:50,028 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [0.50929463 3.32312006 6.07946812 7.41457534 5.18612024
2.76481604
4.86786611 1.61125501]
2023-04-07 19:06:50,040 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:06:54,091 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.82928649 0.27638975 3.92271498 5.23073334 7.14489974
1.69002982
6.01179344 2.02356408]
2023-04-07 19:06:54,102 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:06:58,246 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [5.56375484 7.17185114 1.63822326 3.71468508 0.82051988
6.12378385
4.80451521 2.6805594 ]
2023-04-07 19:06:58,256 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:07:02,485 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [7.29934424 1.2181267 4.68432002 2.32369081 0.39085067
6.38889336
3.48216993 5.90505363]
```



```
2023-04-07 19:07:02,494 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:07:06,851 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [6.3842689 4.67001352 2.88453122 0.89328081 5.47960601
7.59523543
1.90799091 3.76423522]
2023-04-07 19:07:06,861 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:07:11,184 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [6.70586559 3.19087447 0.97270799 4.75546229 1.5369982
5.72159643
7.34319916 2.81325188]
2023-04-07 19:07:11,195 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:07:15,438 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [6.23631848 3.27091122 5.39015417 7.03722659 4.111795
0.28010101
2.83286949 4.26861008]
2023-04-07 19:07:15,449 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:07:19,457 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [0.91333236 3.81346157 7.0523646 4.65314426 1.38364973
5.06575305
2.29907932 6.70784199]
2023-04-07 19:07:19,467 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:07:23,807 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.81948259 5.24422274 2.54859116 6.58103797 3.38893654
7.49649399
4.48222142 0.51533857]
2023-04-07 19:07:23,820 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:07:28,106 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.86244336 7.5451818 4.67340617 1.40646383 5.09235113
0.37611441
6.62817597 3.68058085]
2023-04-07 19:07:28,117 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:07:32,357 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.65687018 4.21394608 7.21281379 0.86422488 3.54235912
6.29233282
2.14946666 5.66033196]
2023-04-07 19:07:32,367 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:07:36,615 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [5.38447923 2.5054267 4.02731979 6.56518563 3.39902293
1.66198829
4.7435675 7.37666461]
```

```
2023-04-07 19:07:36,626 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:07:40,762 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.17606916 1.63297427 7.47629041 5.08207173 0.86505428
2.76207117
4.12240633 6.84612337]
2023-04-07 19:07:40,775 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:07:44,934 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [3.99800255 6.44062399 4.14287605 7.7028808 2.17366
0.52767275
5.66839537 1.8441765 ]
2023-04-07 19:07:44,944 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:07:49,118 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [3.74853962 7.77666761 4.98990109 1.09025692 5.75695629
2.04281429
6.69478252 0.83184317]
2023-04-07 19:07:49,130 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:07:53,298 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [6.14510528 1.16873737 4.29466848 7.23795554 3.68570025
0.4464226
2.37990889 5.55820264]
2023-04-07 19:07:53,307 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:07:57,452 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [3.53905043 5.01991978 0.29008647 2.48180892 6.7963797
4.64303712
1.21865962 4.70075607]
2023-04-07 19:07:57,462 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:08:02,204 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.22612109 4.18377975 5.24229315 1.46129248 3.68069072
6.01744298
0.51398277 7.38032051]
2023-04-07 19:08:02,214 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:08:06,365 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [7.69255019 3.01647994 1.79701946 5.58296056 2.09579753
0.27601859
6.26609307 4.86974328]
2023-04-07 19:08:06,376 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:08:10,570 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [2.19423777 5.59995331 7.84560132 0.73496646 3.88802606
6.65553462
4.26365785 5.49336164]
```

```
2023-04-07 19:08:10,581 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:08:15,085 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [5.91768765 3.26420878 6.98037925 0.45949694 4.27304738
1.76231429
7.68278948 2.42435137]
2023-04-07 19:08:15,097 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:08:19,439 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [4.86621653 5.04029378 7.72167227 0.44611005 2.64586618
6.1181229
1.30359212 3.55506205]
2023-04-07 19:08:19,452 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:08:25,242 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.79881096 1.81896438 5.50366253 0.51161364 6.71147821
3.35619704
7.63886227 2.88599794]
2023-04-07 19:08:25,256 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:08:30,063 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [3.74343048 6.01503636 2.87659546 7.35556675 5.77896572
0.29918457
4.38622415 1.8214577 ]
2023-04-07 19:08:30,074 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:08:34,904 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [6.60973686 3.36433465 5.6038246 0.59828184 2.66990015
4.20412268
7.53323735 1.4963738 ]
2023-04-07 19:08:34,917 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:08:39,521 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [5.02257125 3.80510687 4.07478245 0.99300465 2.8506659
6.60552689
1.70226155 7.17008216]
2023-04-07 19:08:39,538 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:08:43,854 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [6.79038896 4.56649655 1.58921723 5.9763084 0.4059426
2.58906881
7.42779883 3.23340643]
2023-04-07 19:08:43,865 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:08:48,123 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.11835292 5.38947196 3.56547266 0.35322722 7.97083907
1.48453819
4.99095829 6.29165276]
```

```
2023-04-07 19:08:48,135 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:08:52,446 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.76072533 4.59277972 5.78216799 7.19332744 0.99407894
3.00533377
6.68353627 2.11037248]
2023-04-07 19:08:52,457 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:08:56,694 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [1.49721162 3.35151631 5.26417941 6.75453559 6.6964306
4.81235672
2.35705556 7.93597647]
2023-04-07 19:08:56,705 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:09:01,084 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [6.95926929 2.18986749 5.57603599 1.80716965 4.56282956
7.28213162
0.60365149 3.94940915]
2023-04-07 19:09:01,095 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:09:06,581 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [5.9606062 2.94391763 4.64010955 7.39733291 3.24706835
0.66055678
6.85183928 1.67692555]
2023-04-07 19:09:06,594 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:09:11,383 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [7.15197679 3.47790248 6.95532385 2.63641205 4.18289241
1.79814913
6.19474795 5.82320862]
2023-04-07 19:09:11,398 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:09:16,642 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [7.26747561 4.38941801 6.11513542 1.36616789 3.05099233
5.96697982
0.64507257 2.32317998]
2023-04-07 19:09:16,658 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:09:21,717 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [4.11086935 7.83988231 1.589088 3.33462921 7.51677219
6.76898179
2.88026745 0.78693989]
2023-04-07 19:09:21,728 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:09:25,912 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [7.32846977 6.07858287 3.27695757 0.53526741 4.78672172
1.18063653
5.26421747 2.67430693]
```

```
2023-04-07 19:09:25,925 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:09:30,222 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [1.37414191 6.84988207 4.37549373 7.7080173 0.81096429
3.57776929
5.29207853 2.75585238]
2023-04-07 19:09:30,232 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:09:34,541 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.29962452 7.63212531 4.57128902 3.41949568 0.17259036
6.44233428
1.6004209 5.08936852]
2023-04-07 19:09:34,553 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:09:38,814 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [5.70284284 3.38256097 7.72422217 0.22995075 2.54461652
4.94793413
6.0323334 1.77363901]
2023-04-07 19:09:38,824 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:09:43,359 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [5.28870868 3.05219329 1.71258194 4.11466705 2.16662622
7.50534267
6.52843761 0.71325936]
2023-04-07 19:09:43,369 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:09:47,812 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [7.4105395 1.70442947 4.36535805 2.75762449 0.34691618
3.41934692
6.26838699 6.61775367]
2023-04-07 19:09:47,823 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:09:52,359 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [5.34678583 3.26965648 1.09262477 7.01764398 2.21202465
0.8884508
6.03453434 4.82552051]
2023-04-07 19:09:52,371 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:09:56,631 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.59794737 5.38257053 7.41279704 3.19496721 0.35836097
2.96140876
4.74007248 6.2636723 ]
2023-04-07 19:09:56,640 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:10:01,011 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [5.32934406 2.40333514 0.8045594 6.1066058 4.2484005
7.40486633
1.4832407 3.40513627]
```

```
2023-04-07 19:10:01,022 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:10:05,435 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [1.53271992 7.61757287 5.89187101 3.66144699 0.58204553
4.61577447
4.13653535 2.85676549]
2023-04-07 19:10:05,445 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:10:09,818 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [5.27256841 2.82537882 4.00654485 1.69656067 7.52067913
4.33334151
6.02913704 3.05366487]
2023-04-07 19:10:09,830 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:10:14,494 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [2.64214888 5.65263142 3.72765084 1.37976581 7.15356114
4.19968785
6.35068494 0.96819995]
2023-04-07 19:10:14,511 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:10:18,976 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.83611588 7.21701778 0.9176059 4.69868124 6.69226197
1.14741128
5.07500836 2.86186846]
2023-04-07 19:10:18,991 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:10:23,529 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [5.42107501 1.12466727 4.57778592 0.76648018 3.81821917
6.78744282
3.88778804 2.48527934]
2023-04-07 19:10:23,545 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:10:28,419 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [6.09893606 3.73280626 5.14420088 0.09915747 1.11836389
4.66010739
2.88991416 7.16726806]
2023-04-07 19:10:28,431 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:10:32,953 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [2.41992165 5.40769593 1.86234048 6.66244747 6.22428595
3.27999587
7.53089995 4.02815087]
2023-04-07 19:10:32,966 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:10:37,493 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.11534089 5.48575718 7.73627799 1.87900178 6.24772942
0.07363395
2.71137319 4.37138185]
```

```
2023-04-07 19:10:37,506 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:10:42,062 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [4.61563157 1.5492353 5.41701894 0.98635979 2.32488056
7.55353909
3.71314324 6.22523527]
2023-04-07 19:10:42,072 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:10:46,552 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.82303692 6.65564969 3.54467739 0.31933375 7.18665654
4.77069997
1.99296369 5.91962135]
2023-04-07 19:10:46,564 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:10:51,143 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [4.26819933 7.47778427 3.27426306 0.34226329 6.29846623
1.90618333
5.33039958 2.19077474]
2023-04-07 19:10:51,158 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:10:55,579 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [0.77475585 2.5388627 4.25655998 5.68393527 7.05277085
5.48452066
1.77568099 3.01979684]
2023-04-07 19:10:55,589 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:11:00,039 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.83891898 0.86241455 5.09276852 3.71807635 7.62467202
6.81665636
4.17853289 1.23356678]
2023-04-07 19:11:00,052 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:11:04,514 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.51852588 5.26484927 0.74394459 2.74176808 4.09695593
7.19839433
3.75250384 6.13348274]
2023-04-07 19:11:04,528 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:11:09,380 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [7.91144533 4.56455778 6.33607413 1.42229132 5.59288268
2.08483724
0.94847198 3.64632677]
2023-04-07 19:11:09,390 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:11:13,895 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [1.18862731 5.51532552 2.31533692 6.19787106 3.11874291
0.04593845
2.4225983 4.95303076]
```

```

2023-04-07 19:11:13,905 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:11:18,483 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 1.0, best pos: [7.48035267 6.56238754 1.45909858 3.51414487 5.15720353 0.80909823 2.19163838 4.65289999]
2023-04-07 19:11:18,495 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:11:23,095 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 0.0, best pos: [3.93828031 7.62063192 0.08541896 2.49502597 5.26841414 1.38759045 6.00766036 4.8752687 ]
2023-04-07 19:11:23,106 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:11:27,357 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 2.0, best pos: [6.5838047 2.42885249 6.01820147 5.2551426 3.25826889 0.78896556 4.17429114 7.3793009 ]
2023-04-07 19:11:27,368 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:11:32,069 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 1.0, best pos: [7.2348719 2.3222711 4.07595579 6.72816087 1.60512897 3.63355731 5.52213313 0.90001996]
Average time (sec): 4.322767145633698
PSO solved 21/100

```

Medium input test no. 2

We conducted another medium input test for $n = 10$. Our algorithm finds only 2% of solutions in an average time of approximately 5.2 seconds. For comparison, the genetic algorithm for the same input found 86% of correct solutions.


```
In [42]: n = 10

times = []
generations = []
solved = 0
for i in range(100):
    start = time.time()
    optimizer = generate_optimizer(n)

    cost, _ = optimizer.optimize(f, iters=1000)
    end = time.time()
    times.append(end - start)

    if cost == 0.0:
        solved += 1

printAverages(times, solved)
```

```
2023-04-07 19:12:51,562 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:12:55,005 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [7.45959248 5.51686193 8.76244498 2.59259731 7.94219293
3.28875843
6.06205755 4.33649436 9.90337751 0.65360247]
2023-04-07 19:12:55,011 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:12:58,414 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [1.23597675 4.86155189 9.38815499 3.0082694 4.5391849
4.41101632
8.02359968 0.05415364 2.41866462 6.27493602]
2023-04-07 19:12:58,423 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:13:02,306 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [9.5895058 0.87421085 3.76847582 5.2560091 7.23812214
2.42347761
6.92195387 1.2365421 5.46862533 4.59165694]
2023-04-07 19:13:02,317 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:13:07,092 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [3.78940699 8.61363045 5.33372853 1.8026068 9.79794614
6.51287582
2.69911057 0.84454314 7.68382927 4.26561073]
2023-04-07 19:13:07,101 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:13:11,940 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [9.31539263 5.03112302 3.27024387 8.25530297 0.07227619
2.79127032
6.16331344 1.22512586 7.46273581 4.69457259]
2023-04-07 19:13:11,949 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:13:16,820 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [5.19792292 6.47473925 2.34247487 7.54233854 3.50735678
8.35751129
0.08934702 4.98220278 6.62934797 9.71085875]
2023-04-07 19:13:16,829 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:13:21,666 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [5.0132892 9.51722701 2.60558115 4.83772974 5.69283795
1.90742667
8.88630692 6.69605773 3.34561759 7.69388993]
2023-04-07 19:13:21,677 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:13:26,570 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [5.16833112 8.77859602 1.27703919 3.73147462 6.79509545
2.43426516
9.11400778 5.77679656 4.74538875 7.39860582]
```

```
2023-04-07 19:13:26,580 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:13:31,720 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [8.20227286 0.76731857 9.367799 4.86777693 2.21559284
4.05220522
1.36333884 7.55265369 5.82995035 3.37996286]
2023-04-07 19:13:31,731 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:13:36,677 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [4.37238992 5.78496004 7.20834327 2.97117641 1.8835618
8.63330569
5.97140799 3.9629344 6.85872027 9.15160248]
2023-04-07 19:13:36,686 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:13:41,765 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [8.25400876 6.67386756 3.80740296 1.3508325 5.10908044
5.76733835
0.42426994 2.58048627 4.91278715 7.7884623 ]
2023-04-07 19:13:41,778 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:13:47,176 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [9.15059225 2.55472989 5.50429412 8.71742014 4.8800853
7.74072386
0.33367901 3.57695679 6.82462825 8.78559344]
2023-04-07 19:13:47,187 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:13:52,260 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [5.41386552 1.19241471 8.36584518 5.1259529 7.31604135
4.23279178
2.64383328 0.61730967 9.44152001 6.28503686]
2023-04-07 19:13:52,270 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:13:57,590 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [2.69124217 6.48225254 5.91062662 9.17288476 7.94659008
3.89815869
0.89699834 4.8465266 8.9326611 1.78761319]
2023-04-07 19:13:57,600 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:14:02,827 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [8.23598156 6.40321597 2.61758626 0.82702621 7.67942319
9.93772377
4.03045393 5.55639027 1.37151222 3.15429508]
2023-04-07 19:14:02,837 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:14:08,126 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [0.49233964 7.3399605 4.52377156 1.73075476 6.98396722
2.01338008
9.05340456 5.28529751 6.10327842 3.17464763]
```

```
2023-04-07 19:14:08,138 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:14:13,334 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.62515961 8.49569856 4.88435375 0.1810789 7.05450981
5.9829561
2.3622795 6.76374323 9.54577737 3.40763472]
2023-04-07 19:14:13,346 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:14:18,344 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [2.50124426 7.97853993 1.84114337 8.39525715 5.60176417
0.25033097
4.90644874 6.25728828 6.68177635 3.95329706]
2023-04-07 19:14:18,356 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:14:23,394 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [6.15811118 0.90706582 5.77894136 8.293719 1.72770191
3.64655742
7.73161673 9.83662758 7.25773943 4.35489779]
2023-04-07 19:14:23,405 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:14:28,433 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.63285093 5.08881992 7.28342525 9.35879892 3.28935366
8.41446876
2.54559032 4.26583745 6.23260339 0.24853598]
2023-04-07 19:14:28,443 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:14:33,563 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [6.32960826 1.87160997 5.29041257 4.26750522 5.59083753
0.74364919
8.67937015 3.90962936 7.81446695 2.53118656]
2023-04-07 19:14:33,577 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:14:38,636 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [2.84662916 4.32596044 9.57207565 3.32713901 0.78923891
5.197098
7.72784759 1.31223079 8.51112093 6.43069061]
2023-04-07 19:14:38,646 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:14:43,636 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [8.36986444 5.66757418 2.04933819 4.11958331 7.43121571
0.20023175
7.15919146 6.52271651 1.50202205 3.44260647]
2023-04-07 19:14:43,646 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:14:48,902 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [7.02554091 9.34266944 6.13390197 1.20793307 4.88542927
0.75188134
8.08130717 6.09282023 5.45153074 2.89437494]
```

```
2023-04-07 19:14:48,913 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:14:54,120 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [1.32570061 5.29993215 7.42648346 9.11462516 6.40018133
2.85386034
2.7946531 4.56095203 3.91888643 8.01315902]
2023-04-07 19:14:54,132 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:14:59,085 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [3.29962984 9.56029439 0.85182732 6.04082859 4.1890203
1.95378292
6.1284871 2.94357382 5.31725232 6.52844324]
2023-04-07 19:14:59,098 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:15:04,240 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [2.8712136 7.45559644 5.38701453 1.80002051 6.8288455
0.48446564
4.82452612 3.7116112 4.61760202 9.00929768]
2023-04-07 19:15:04,252 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:15:09,385 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [7.22509901 4.79284057 1.19830512 3.14590386 8.23390152
6.17168229
3.82510409 2.45753288 0.56170822 5.5272449 ]
2023-04-07 19:15:09,400 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:15:14,667 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [6.7102689 0.2804837 5.33273423 8.71604424 1.7010389
3.29668984
2.80001416 7.00971104 9.32814329 4.47083162]
2023-04-07 19:15:14,682 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:15:19,755 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [7.08549517 5.9281645 8.0182857 2.92588693 9.84454967
3.60916587
6.31780559 4.54623975 1.57122023 6.59419056]
2023-04-07 19:15:19,766 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:15:24,912 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [3.84646479 1.30490013 6.75514874 8.56477052 2.76571495
7.22013666
5.6285321 0.06062943 9.14149957 4.86903948]
2023-04-07 19:15:24,923 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:15:30,270 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [7.88486701 5.3531503 0.35838306 9.13288677 1.86208025
6.27175265
8.53034396 3.37850186 2.59053478 4.22993906]
```

```
2023-04-07 19:15:30,280 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:15:35,451 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [9.20138297 5.4605259 8.27134937 4.30211659 7.04884711
0.900799
2.12587192 6.14060801 1.32426875 3.32819489]
2023-04-07 19:15:35,463 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:15:40,592 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [9.18074504 6.09154264 3.5933397 5.17204372 0.45678026
7.22027195
5.36589296 1.21761359 8.0061606 4.01395511]
2023-04-07 19:15:40,602 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:15:45,873 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [7.42417084 4.75632833 9.13358734 5.87721897 2.45899927
6.88369636
1.4322438 9.3032793 8.64754268 3.48436473]
2023-04-07 19:15:45,885 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:15:51,290 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [2.42340699 3.90019416 9.20542965 0.71269769 4.55941322
8.15613518
3.04535222 3.65381581 6.87096552 1.45829864]
2023-04-07 19:15:51,301 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:15:56,670 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [9.51343714 5.56750308 4.49408038 1.65601241 3.09274724
0.05042806
5.6051393 7.59453071 2.61033778 6.67730565]
2023-04-07 19:15:56,679 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:16:01,941 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [7.2821185 9.70241956 2.95477069 6.13267936 1.48667432
3.70792148
5.14245381 0.45229149 4.81826525 7.28568851]
2023-04-07 19:16:01,950 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:16:07,135 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [9.50032362 1.63025033 6.83171644 0.16469279 3.0082108
7.36705476
5.1048192 8.38239114 2.53602526 4.36701371]
2023-04-07 19:16:07,146 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:16:12,339 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [4.48665855 0.96640373 9.9601359 5.67794944 1.90695935
8.01462852
6.38792265 3.34174205 2.15204113 7.25455976]
```

```
2023-04-07 19:16:12,349 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:16:17,564 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [4.1250844 8.63191972 5.47147035 0.24588606 6.40278037
1.82456142
7.73410102 2.2603501 6.10610923 3.23414625]
2023-04-07 19:16:17,575 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:16:23,136 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [4.81915516 9.7448655 6.10412631 2.57068934 7.45017635
3.56744842
8.60963978 0.86578718 5.64931113 6.61472658]
2023-04-07 19:16:23,151 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:16:28,409 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [1.74571312 5.38435675 6.31949354 9.13232935 7.27385781
2.61645799
8.30528703 3.44609812 0.50241164 4.20572791]
2023-04-07 19:16:28,425 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:16:34,069 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [6.33076476 0.82076762 1.53932015 5.945956 7.09925702
9.2313511
3.08113192 8.33167875 2.86950078 4.04718912]
2023-04-07 19:16:34,081 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:16:39,733 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [9.02010895 6.51947811 0.76204064 2.12087834 4.85775843
8.34432244
7.43867195 3.46930959 5.22125534 8.48663596]
2023-04-07 19:16:39,746 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:16:45,467 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [2.53502751 5.2176363 8.04998144 4.20558455 0.32872372
3.67851346
7.88663425 7.43013128 1.56276444 6.87842656]
2023-04-07 19:16:45,478 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:16:51,109 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [6.52208353 1.5296548 5.81978777 2.89060116 0.48889888
4.27075341
8.83332568 4.24840588 9.30545525 7.22150092]
2023-04-07 19:16:51,118 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:16:56,726 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [0.69715903 4.46560859 7.39425685 1.9614647 8.57349971
6.89042773
2.07452283 3.57990602 5.50183511 9.25374905]
```

```
2023-04-07 19:16:56,741 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:17:02,630 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [4.95534219 2.05588473 0.94154553 8.30581213 1.77141277
3.04517949
6.86711594 9.6453326 7.06813014 5.50001491]
2023-04-07 19:17:02,641 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:17:08,312 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [4.61416081 1.6906015 3.77737804 9.35254374 6.61493075
2.62504797
5.67971661 8.38933876 0.48219116 7.51132875]
2023-04-07 19:17:08,323 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:17:13,938 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [6.65391452 3.68416009 1.63492193 7.84492469 9.3426728
4.3572375
5.51616709 8.69980598 2.80224864 0.61500477]
2023-04-07 19:17:13,949 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:17:20,193 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [3.28811724 1.42833531 4.87314974 8.16367265 6.78625868
9.05607782
7.71592513 2.13250259 0.18456643 5.22613754]
2023-04-07 19:17:20,209 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:17:25,995 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [7.50153549 6.3129409 3.94269272 8.42634707 2.55069313
5.62692156
9.74509725 1.89273745 4.26235265 0.81347419]
2023-04-07 19:17:26,010 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:17:31,766 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [4.78815373 1.66821663 5.7205485 7.49531489 9.30836211
3.05910794
0.37439096 2.88104579 7.44383038 6.75025093]
2023-04-07 19:17:31,778 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:17:37,513 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [8.43792861 2.07557217 5.440296 9.97287041 6.24900493
3.41169197
0.56915047 7.57276731 1.6806104 4.40619678]
2023-04-07 19:17:37,524 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:17:43,218 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [7.10804354 1.13647646 6.81044078 8.75888347 0.87219238
9.44130992
3.65766454 5.10143459 2.69268566 4.91756929]
```



```
2023-04-07 19:17:43,229 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=0
2023-04-07 19:17:48,852 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 0.0, best pos: [3.29754622 7.50347221 2.53392681 4.29418424 8.53136598
0.92640068
5.5786431 9.68311414 1.85743387 6.27852593]
2023-04-07 19:17:48,864 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:17:54,688 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [5.17454732 1.20804623 7.83102055 9.19540701 3.06787337
8.39210733
2.58530411 4.66013582 6.28332641 0.91254606]
2023-04-07 19:17:54,702 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:18:00,376 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [8.56370628 5.26600416 2.31142351 6.63836629 3.42257472
0.73885418
9.70123504 1.6421578 4.81505783 7.31586259]
2023-04-07 19:18:00,396 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:18:06,375 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [5.74248904 8.25137088 6.04598528 0.27215988 3.59465334
7.68670109
4.54587233 1.11657924 9.51451981 2.39669164]
2023-04-07 19:18:06,388 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:18:12,375 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [2.67883817 8.85488747 6.76090783 0.69427815 9.23644783
1.15502236
4.79295979 5.22757747 7.59564051 3.43623879]
2023-04-07 19:18:12,388 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:18:18,450 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [8.86109996 4.32450327 5.29503821 1.63996812 9.21028088
2.30157514
6.75637757 3.55337095 7.51995772 0.15987666]
2023-04-07 19:18:18,462 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:18:24,647 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [1.79872964 5.2846803 7.09955298 4.97516461 0.57302181
3.4001666
6.49012871 4.9209948 2.56864888 8.51204658]
2023-04-07 19:18:24,662 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:18:30,521 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [8.72591709 4.37714513 1.7794793 6.58769314 5.51332106
7.09258564
0.99962831 6.07953701 3.41454198 9.47743254]
```

```
2023-04-07 19:18:30,534 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:18:36,796 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [4.42908735 1.36198268 8.24074111 5.49785467 2.42020654
7.91937394
7.38231469 0.23249829 3.15475049 6.19506974]
2023-04-07 19:18:36,809 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:18:42,912 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [6.09225708 3.04636785 2.87356513 8.79244565 7.92035576
0.58754898
4.64013963 1.42281152 9.32447907 5.14852318]
2023-04-07 19:18:42,924 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:18:49,091 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [9.41426067 0.32384333 3.83389132 8.29064636 4.10621483
7.32413684
1.84423499 6.66694427 2.05984277 5.53222981]
2023-04-07 19:18:49,104 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:18:55,745 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [0.60959088 3.55325216 5.85643696 7.21737555 2.18858808
6.41813847
9.93992679 1.64931291 4.25681359 8.33611789]
2023-04-07 19:18:55,758 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:19:02,231 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [9.13412182 6.77001007 2.11790889 5.26379108 1.877829
8.49985266
4.87347141 0.36932895 7.67011122 3.57985929]
2023-04-07 19:19:02,243 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:19:08,552 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [4.67169757 2.83789222 8.78073474 6.93136242 9.30157712
7.51593355
3.04138375 0.91496029 6.67673306 5.39496871]
2023-04-07 19:19:08,563 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:19:15,197 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [6.55655571 0.13777015 7.22042423 4.52963452 1.26060761
8.02524424
2.70968597 9.30219659 3.16905138 5.38536115]
2023-04-07 19:19:15,210 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:19:21,868 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [3.30226729 7.89501131 4.14315229 4.91034804 8.42822725
1.90848919
5.75687836 2.3011486 6.43963708 9.83212831]
```

```
2023-04-07 19:19:21,881 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:19:28,079 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [6.98778035 2.29067993 0.6406253 5.82451774 1.909425
4.79068133
7.81922532 7.08234514 3.11446725 7.21779296]
2023-04-07 19:19:28,092 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:19:34,446 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [9.24347722 6.85720708 0.68119679 2.5158091 7.75254847
5.06706751
3.49602335 8.27208249 4.85738871 1.23916627]
2023-04-07 19:19:34,459 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:19:40,619 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [2.67176404 6.75096085 3.9405788 0.83853091 9.92110232
8.1750882
4.49029684 9.00559027 1.0834203 5.60818152]
2023-04-07 19:19:40,630 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:19:46,494 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [1.6804342 4.91349091 8.47942652 0.2210192 7.19705891
3.62308083
6.71347703 2.18391511 7.91848816 5.85819708]
2023-04-07 19:19:46,505 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:19:52,508 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [9.50371035 0.50872543 3.77061184 5.76432569 2.92768174
8.04292089
6.11983995 4.09733502 7.48145766 1.9942727 ]
2023-04-07 19:19:52,521 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:19:58,360 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [0.84226822 9.69754011 5.58157499 1.81442071 7.23859865
6.19074068
7.06878018 2.86260071 4.29974664 8.01066371]
2023-04-07 19:19:58,371 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:20:04,374 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [5.93190966 3.41257607 1.50520297 4.12889658 7.02592101
9.08487305
2.27572099 5.41576404 8.21288395 6.74204213]
2023-04-07 19:20:04,387 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:20:10,395 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [3.83114611 1.25044068 5.6487459 8.08242601 0.63065928
5.63360359
4.32331781 2.97433665 7.4014001 9.09922359]
```

```
2023-04-07 19:20:10,406 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:20:16,444 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [0.88391389 1.51742698 4.17866644 2.39644132 8.46749007
6.19698582
9.14731084 3.83803852 5.01123715 7.82499261]
2023-04-07 19:20:16,469 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:20:22,869 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [8.09609187 0.06860412 6.7110105 3.21124134 1.84502785
7.42410168
9.22451225 4.76877018 2.48188863 5.32880883]
2023-04-07 19:20:22,880 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:20:28,862 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [3.6312408 7.77962423 2.67344955 4.38473588 9.10549918
6.21912031
0.19767479 5.23262489 1.89335606 8.26451191]
2023-04-07 19:20:28,878 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:20:34,629 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [6.55737933 3.19523938 8.06414285 4.48632288 8.03196101
0.66340857
2.92364325 7.74017788 5.55270244 1.15603362]
2023-04-07 19:20:34,646 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:20:40,467 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [4.59633534 6.13309842 3.46081907 7.12305659 7.09933335
1.38716406
8.87566748 5.72215856 0.51478364 2.30902917]
2023-04-07 19:20:40,483 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:20:46,580 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [3.50301025 9.94945287 4.30575507 5.19653547 7.72568135
2.66217433
6.27253686 7.76855668 0.95352181 8.32526212]
2023-04-07 19:20:46,592 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:20:52,789 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [1.89287955 4.00112227 0.04125989 6.22242741 2.26004156
7.67794252
3.55762836 6.25447949 8.66542744 5.43369663]
2023-04-07 19:20:52,803 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:20:59,417 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [7.39903854 4.61599577 8.37085778 3.23887144 9.67902618
1.57979265
5.50267017 2.93595788 6.68593105 3.13688596]
```

```
2023-04-07 19:20:59,431 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:21:05,879 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [7.03282073 1.61886971 4.7537029 9.65999764 6.52706856
3.01245559
8.94374895 8.14669594 2.06328103 5.86298575]
2023-04-07 19:21:05,892 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:21:12,200 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [4.90934902 9.20596341 5.49411023 0.94032112 2.78604798
6.67997134
8.24690945 7.33321922 1.50944765 3.12803164]
2023-04-07 19:21:12,219 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:21:18,535 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [8.80134037 5.66244842 5.94157955 0.52080133 3.27033136
1.72761263
4.11422642 7.23135078 9.11503761 2.61922676]
2023-04-07 19:21:18,547 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:21:24,993 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [7.79224081 0.66311315 8.41157216 1.67849836 4.8571873
4.7277263
2.90591019 9.9702314 5.66775357 3.91713127]
2023-04-07 19:21:25,005 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:21:31,449 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [0.61052406 2.62913307 4.44499422 9.15639067 7.82597296
5.05801349
1.83555631 3.57364452 6.68413607 8.85350177]
2023-04-07 19:21:31,461 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:21:37,839 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [6.51704816 4.45815435 0.30907327 8.24740212 5.2617976
7.04929018
3.8397481 6.88868228 2.54664398 9.1073398 ]
2023-04-07 19:21:37,851 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:21:44,321 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [4.92548005 0.96897116 9.0458229 2.21217751 5.77683185
7.24754207
1.08390426 3.62183969 8.25328663 6.28502179]
2023-04-07 19:21:44,332 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:21:50,542 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [5.99748131 8.88351916 4.49894828 0.8466758 7.11846112
9.19645076
2.27488654 6.06202859 4.41725565 1.07453241]
```

```

2023-04-07 19:21:50,555 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:21:56,434 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 2.0, best pos: [5.37381529 0.97244086 2.47161224 6.54486674 4.24069743 7.21419381 4.46285269 8.73736616 3.43397704 1.95175799]
2023-04-07 19:21:56,446 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:22:02,567 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 2.0, best pos: [3.44089216 0.44377199 2.73236269 5.89052715 9.16844147 1.50565469 4.22144506 7.21071136 8.16042662 6.24275299]
2023-04-07 19:22:02,577 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:22:08,581 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 1.0, best pos: [7.40657999 2.10228891 4.79357075 9.92989657 8.31261601 0.8436875 5.64364012 3.30972825 1.89657893 6.70392851]
2023-04-07 19:22:08,591 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:22:14,331 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 1.0, best pos: [7.56382515 3.09136688 8.77610845 0.26575378 4.47231655 1.94409554 5.71319014 2.96957481 6.16152562 9.10219749]
Average time (sec): 5.627781248092651
PSO solved 2/100

```

Big input test

I carried out the test for large inputs for $n = 12$. It is actually a medium-sized input, but I suspected that the algorithm would not be able to handle it anymore. Unfortunately, I was right and our algorithm found 0% correct answers. The genetic algorithm is much better for this problem.

```
In [43]: n = 12

times = []
generations = []
solved = 0
for i in range(100):
    start = time.time()
    optimizer = generate_optimizer(n)

    cost, _ = optimizer.optimize(f, iters=1000)
    end = time.time()
    times.append(end - start)

    if cost == 0.0:
        solved += 1

printAverages(times, solved)
```

```
2023-04-07 19:24:08,157 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:24:12,629 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [ 7.39805672  3.32364829 11.45932027  6.61684286  4.9686
2463  1.43015777
5.64575341  0.67519456  7.94540542 10.55648699  2.79168896  9.03283262]
2023-04-07 19:24:12,637 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:24:17,162 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 6.22226497  4.32197245  2.17637188  8.39619253  3.6632
6128  3.29709483
7.61274886  0.43123483 10.33558001  1.68067127  9.06447979  5.59326313]
2023-04-07 19:24:17,169 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:24:22,080 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [ 7.5909763  8.99923741  4.19351443  9.55623258  1.1829
6895  5.7422536
11.156827  2.51648767  6.82860236 10.20186937  3.77467605  0.49353504]
2023-04-07 19:24:22,087 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:24:26,645 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 9.54210715  1.99132533  8.42046924  2.92819249 11.3980
2367  9.22906543
7.29968198  3.95587582 10.10535745  0.34678886  9.16319592  5.19050742]
2023-04-07 19:24:26,653 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:24:31,526 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 8.02082701  5.43607279  2.4983126  0.56904912 11.9117
0973  4.38044565
6.87240858  9.6495368  6.52799978  3.21801148  4.25121014  7.29734308]
2023-04-07 19:24:31,535 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:24:36,284 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [10.78159659  1.31167116  4.37020429  9.84532495  3.6684
7982  9.4920831
7.89982556  9.73973736  6.61750348  2.62453469  5.69208349  8.57496524]
2023-04-07 19:24:36,293 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:24:41,488 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 2.58876857 10.53508602  9.1728898  3.60558184  5.8634
7647 11.61082909
4.52933302  5.56158362  1.99345221  6.40705872  8.84582999  7.96415617]
2023-04-07 19:24:41,497 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:24:46,853 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 1.90410972 10.78973155  0.88820777  7.06079054  4.7305
1395  2.9813582
9.23210237  5.71097982  6.61760328  5.36957853  6.07740167  3.48250509]
```



```
2023-04-07 19:24:46,862 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:24:52,281 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 3.79671611  5.91223725 11.34509618  7.18993338  6.49246188  4.49801865
 2.35551881  1.60886781  9.3081596  7.98191392 10.24101326  8.75969623]
2023-04-07 19:24:52,291 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:24:57,829 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 4.90373774  7.97949054 11.72130186  8.85200173  8.01585494  1.50160533
 6.00063028  0.91116566  2.65701094 10.26908451  8.15645576  3.583959 ]
2023-04-07 19:24:57,840 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:25:03,702 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 2.0, best pos: [ 0.06911986  5.75043354  8.45547784  2.36274733  7.15200008 11.07958289
 1.65408168  9.09316963  6.19931595  3.44313695 10.47932168  4.26302621]
2023-04-07 19:25:03,712 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:25:09,592 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 7.7353005  2.54517531  6.48537716  9.42444574  5.44965891  0.36417953
 5.41571166 10.62590286  5.82955961  1.832524  4.71805016 11.66713276]
2023-04-07 19:25:09,607 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:25:15,432 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [10.11452078  5.72142527  7.7654503  1.87554239  4.50451756  8.95590684
 3.43028271  7.60260397  9.127749  2.01289872  6.01828436  7.20384518]
2023-04-07 19:25:15,441 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:25:21,420 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 6.90317488  2.85366649  7.97672253 10.48774449  1.83623056  5.54925271
 8.20919523  4.36218236 11.97256281  3.9335238  6.32967312  9.36805617]
2023-04-07 19:25:21,431 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:25:27,364 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 8.44100726 10.6280093  1.89796348  9.2630618  6.25260689  2.27401915
11.28385541  5.32013891  7.80501957  0.14022247  9.20555544  3.11949726]
2023-04-07 19:25:27,375 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:25:33,438 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 4.0, best pos: [ 6.15904086  3.89478111  7.46522168  5.63161939  8.31681498  0.82340639
 9.43826607  4.82061558  2.35012059  9.50861633  5.951485  10.61646378]
```

```
2023-04-07 19:25:33,448 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:25:39,555 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 0.28927581  7.85174299  8.69588625  1.87584282  8.5517
0647  6.49249119
 9.44990934  3.08640784  6.93504662 11.77892746  2.39064697  5.48303617]
2023-04-07 19:25:39,565 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=5
2023-04-07 19:25:45,676 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 5.0, best pos: [ 0.7465729  5.58070852  5.38670094  9.91600777  6.2853
1194  4.75866261
 1.9414407  3.80014448  7.94086212  7.11575838 11.00714846  8.97881608]
2023-04-07 19:25:45,687 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:25:52,013 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 1.20694005  5.57437131  7.21071581  2.06454469 11.7317
4328  6.23949026
 4.40990389 10.04004889  3.44966559  0.95699801  6.8999448  8.26751904]
2023-04-07 19:25:52,023 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=5
2023-04-07 19:25:57,983 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 5.0, best pos: [ 5.68567855 10.52242958  8.80338687  1.72751069  2.5379
0191  9.18283023
11.59081806  5.95940295  0.2427754  3.55870989  5.57318002  4.09894211]
2023-04-07 19:25:57,996 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:26:04,021 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [ 0.08307022  7.4429594  5.68703284  6.39573078  8.1893
1992  6.35948979
 3.39361301  9.1331982  2.37373753  4.33388344  1.86874439 10.1895681 ]
2023-04-07 19:26:04,034 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:26:10,206 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [ 3.08201722  5.50335731  0.49885062 11.41153763  9.3523
5008  6.30933938
 4.42117917  2.94549186  8.18334945 10.52090681  7.9369021  1.41928187]
2023-04-07 19:26:10,216 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:26:16,480 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 3.81617759  1.76043164  5.78914849 10.12644477  8.4644
5208  2.14679704
 4.8887252 11.10530209  0.18747628  5.33830103  9.54559629  6.19202948]
2023-04-07 19:26:16,489 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:26:22,744 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 6.06113668  3.07272645  1.47942346  8.16359303  4.8412
9681  4.39348264
 9.8520963  0.13394415  2.2612794 10.67671124  7.06537056  4.40064797]
```

```
2023-04-07 19:26:22,754 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:26:28,725 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [8.53792173 6.52276803 3.530046 7.0444705 7.74984407
5.33637747
0.81676398 2.55468183 4.85899797 4.238805 1.65043464 9.44480455]
2023-04-07 19:26:28,736 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:26:34,936 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [11.78389171 3.47491831 10.7618746 8.12804196 1.9245
052 6.19258609
4.30566121 0.01217164 7.61356931 9.22691473 5.22795972 2.80920045]
2023-04-07 19:26:34,945 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:26:41,285 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [10.40062156 3.16701284 5.56087917 2.61073299 9.4281
4034 11.5888061
8.05946714 1.71679554 6.64842053 8.04937874 7.27680442 4.54923226]
2023-04-07 19:26:41,295 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:26:47,438 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [9.14540927 3.48440168 1.66383704 8.36448223 4.2893
3156 0.75827571
7.79670203 10.02531087 5.2351276 7.23037629 6.83589804 11.67987919]
2023-04-07 19:26:47,449 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:26:53,664 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [0.75090864 4.10691423 7.23912938 1.59731399 10.8249
6629 2.64211407
8.33393282 6.57759516 9.59769357 7.14754122 5.46977957 7.38548694]
2023-04-07 19:26:53,675 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:27:00,092 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [6.94922295 3.46715962 10.91915866 8.85789144 1.9221
7599 9.2164401
7.07561861 5.3181775 11.05249572 0.82796155 2.4998791 4.19779315]
2023-04-07 19:27:00,105 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:27:06,252 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [7.13983145 0.70735779 3.87733723 6.62542736 8.46507117
6.7193871
4.53733444 9.44023093 1.80005528 6.42901165 5.14613208 2.7377626 ]
2023-04-07 19:27:06,260 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:27:12,303 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [0.8857507 7.76786548 3.8476872 8.17332409 6.4781
0034 2.97878577
4.83406952 6.61954436 11.42666512 6.40999946 5.71341731 1.81201516]
```

```
2023-04-07 19:27:12,313 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:27:18,538 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 3.8390903  5.54890192  7.76585779  4.84053252  2.8609
9046  0.68428896
 8.72193144  5.80641374  1.32610977  9.33923576  6.371657  10.55759962]
2023-04-07 19:27:18,549 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:27:24,792 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 3.45303842  6.34789426  9.32704436  7.93496922 10.5380
6139  7.43305913
 7.4954152  1.81953197  4.21427026  8.33979906  5.79231642  2.38524985]
2023-04-07 19:27:24,807 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:27:31,430 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 7.60931668 10.36015495  8.26300633  0.17055981  4.9821
7315  1.32737287
 9.59205628  6.62393847  9.40678636 11.4015143  9.79127151  5.38570927]
2023-04-07 19:27:31,442 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:27:38,664 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [ 3.7285963  10.15533556  7.15296823  4.98604256  2.6748
5223 11.26869691
 6.23952248  1.01018519  5.84371169  8.89373724  9.88470809  9.51844161]
2023-04-07 19:27:38,675 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:27:44,901 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 6.50963995 10.12407788  6.47270673  6.85940996  9.0705
6008  0.89876966
 3.91600257  7.14827205  2.73353936  5.70945939  8.29212382  4.57172312]
2023-04-07 19:27:44,913 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:27:51,205 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 1.23929067  5.29379618  8.44667428 11.68741878  4.3704
859  2.78066199
 3.78422427 10.21958378  3.16130862  7.57349089  6.34538581  7.57889539]
2023-04-07 19:27:51,217 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:27:57,430 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [ 8.26077238  5.27000435  8.47332306  6.48892246  0.2274
6102  2.65252283
 4.58834566  7.22373494  9.33539359 11.45265455  3.83910747  1.46578992]
2023-04-07 19:27:57,439 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:28:04,017 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 9.40483904  4.38188706  2.83184783 10.35019636  5.8464
4679 11.17492843
 8.35945638  1.5086574  3.45703286  6.67974858  7.79515346  7.37810071]
```

```
2023-04-07 19:28:04,030 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:28:10,677 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 1.82149354  3.70612848  8.25183453 10.69360497  4.7672
1745  2.25827272
 0.77481788  5.45667468  7.99360856  5.23868258  9.9839723  6.06455149]
2023-04-07 19:28:10,688 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:28:17,291 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [10.46411285  5.72685759  4.15495174  5.26403212  8.4084
5802  6.94332801
 3.2468648  0.58614789  8.58224722  7.98796932  9.41337084  2.45505922]
2023-04-07 19:28:17,302 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:28:24,002 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [ 1.52338453  9.27214217  7.2113429  5.72647251  3.5154
9478 11.02996162
 6.38987277 10.24129964 10.29804728  2.47476702  4.89683126  8.82132367]
2023-04-07 19:28:24,014 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:28:30,388 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [ 7.13434232 11.0791051  4.50021624  5.72286443  0.1078
2014  9.64466927
 3.08397606  8.69060566  5.58534072  2.68776616 10.32358825  6.0414414 ]
2023-04-07 19:28:30,399 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:28:36,607 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 3.27360925  9.21553839  8.73610693  0.93764653  2.5801
6276  4.19752228
 1.65688543 11.38791635  6.82558611 10.55604164  7.08461919  5.77206763]
2023-04-07 19:28:36,617 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:28:42,762 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 6.18280932  1.40148676  5.52084148  8.02946043  9.4873
8457  0.03235895
 2.33655013  4.84171545  7.37346533  8.62171853 11.0397698  3.55206921]
2023-04-07 19:28:42,772 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=5
2023-04-07 19:28:48,881 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 5.0, best pos: [ 2.59598693  8.34495015  1.94032539 10.19223699  6.7924
1872  6.82153834
 4.13501593 11.71396725  5.93908452  3.22954669  7.57885017  6.76121032]
2023-04-07 19:28:48,890 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:28:55,171 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 8.98223406  5.61205038  2.4565867  4.41959022  7.3988
5302  1.6705333
 3.13315093  5.52184435 10.08930029  5.09883524  9.23026225  6.4901108 ]
```

```
2023-04-07 19:28:55,182 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:29:01,577 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [ 3.76018837  8.82054778  0.76634141 10.72156134 10.4025
2277  5.60542546
 1.6707909  6.65578254  9.82991357  2.88643302  4.70379281  7.89191205]
2023-04-07 19:29:01,589 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:29:08,090 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 0.85687321  2.81047895  6.48404341 11.97193872 10.1464
9133  8.51477156
 5.66190693  3.60134053  8.38392153  4.57926781  7.35787309  9.19879921]
2023-04-07 19:29:08,101 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:29:14,605 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 7.88757027  8.42971901  9.20182322  1.77754452  5.6634
0509  3.76745958
 2.91263766  3.28187666 11.03304837  6.06608684  4.69073587 10.05459639]
2023-04-07 19:29:14,619 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:29:21,046 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 2.28035283  6.25995007  7.78590827  7.6582377  4.6358
7395  0.50736163
 5.42087786  8.05850754 11.79626894  1.48863077  7.22346179  9.90906614]
2023-04-07 19:29:21,057 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:29:27,451 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 0.91297814  9.25970475  6.62223247  2.59403987  5.7808
2475 10.92415904
 8.45481329  4.07063439 11.41899092  7.48098918  3.96412212  5.95750283]
2023-04-07 19:29:27,462 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:29:33,852 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [10.24942298  8.27107762  1.82818501  9.47230214  7.3917
0249  6.53116197
 7.83930021 11.60834055  9.52396668  3.65587614  5.00925547  2.78868084]
2023-04-07 19:29:33,862 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=1
2023-04-07 19:29:40,406 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 1.0, best pos: [ 6.60062103 11.00811252  0.66559603  8.8011195  3.2730
4098  5.2390744
 9.86063698  1.29593125 10.47778786  4.21403232  7.04606001  2.20074646]
2023-04-07 19:29:40,418 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:29:46,704 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 4.4635379  8.2898949  5.99405391  2.39430419  2.5975
3135  1.51169821
10.25648558  9.52423925  3.37141975  6.06376531  7.34319187 11.86372201]
```

```
2023-04-07 19:29:46,714 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:29:52,993 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [10.37081053  8.81185052  2.77084179  9.04421783  6.0469
3471  1.46639673
 10.13123725  0.76682862  3.73920138  7.86170964 11.09391906  4.18832477]
2023-04-07 19:29:53,003 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:29:59,378 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 6.75011199  8.80323351  3.98707593  4.20800222  0.4263
0837  7.04661015
 9.3319921  1.60872423  5.41892449  2.9860807  7.36419512 10.85405841]
2023-04-07 19:29:59,388 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:30:05,940 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [ 1.91870294  3.48644812  4.67258258  7.41331889 11.3859
4588 10.08029472
 2.70809904  5.20727351  8.099086  0.7947292  9.29320593  6.89032299]
2023-04-07 19:30:05,953 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:30:12,511 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 6.02953  1.22370259  5.64029191  2.34685223  8.8800
1192 10.16885011
 3.0802406 11.83912498  9.50738996  7.56495363  4.95834088 10.15921724]
2023-04-07 19:30:12,522 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:30:19,214 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [11.1956848  3.41114686  5.50792588  8.6501261 11.2333
4874 1.22313129
10.04478904  7.766268  9.3721382  6.75778823  0.54199051  2.73654662]
2023-04-07 19:30:19,224 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:30:25,608 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 9.49376963  3.28356615  1.27990713  4.26169203  8.2785
6279  7.58768339
 2.09844111  5.06979014 11.14225466  7.9872181 10.40451721  6.83099345]
2023-04-07 19:30:25,620 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:30:32,030 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 8.39089974  3.67629513  1.95566311  6.83837903  9.6661
6762  7.40063013
 4.261582 11.83101396  6.30305309  2.84790207 10.81101901  5.32933908]
2023-04-07 19:30:32,042 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:30:38,490 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [10.53675638  6.62191621  4.04455223 11.79447584  1.1499
5783  7.05558002
 9.11340498 11.59843334  3.36932688  5.82289128  8.66071689  2.412817 ]
```

```
2023-04-07 19:30:38,504 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:30:44,987 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 7.32822749  1.58739736  8.7519296  11.55793201  5.3689
2413  3.97589131
  9.30697413  5.83420755  5.00279138  2.96361573  6.05780007 10.38109235]
2023-04-07 19:30:44,999 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:30:52,155 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 4.72734612  9.19520816  7.36311919 10.3240072  5.2793
27  3.46039493
  8.00847037  0.98058322  7.27799284  6.61961061  7.18986994  1.67730425]
2023-04-07 19:30:52,165 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:30:59,130 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 4.78892527  2.74084559  9.25394102  6.75069717 10.7031
7267  1.66401405
  7.8814348  9.34535581  0.68606778  8.45259669  3.41819956  5.23379502]
2023-04-07 19:30:59,141 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:31:07,112 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 9.50145981  7.75150442  5.99644797  1.67088332 11.5613
5858  4.29355118
  8.0616701  0.21540738  5.16188707 10.21587419  2.11363152  6.77746139]
2023-04-07 19:31:07,125 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:31:14,780 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 2.0, best pos: [10.02715764  6.31125111  7.72998438  1.65786426  5.4307
2907  9.01848039
  0.99221506  2.42526745  4.21183266 11.17638415  3.68375495  8.3620821 ]
2023-04-07 19:31:14,794 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:31:22,572 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 8.09649303 10.51653994  3.14749591  2.5040618  6.7077
0059  1.594862
  9.83196078  5.28599252  6.76802364  4.74555855  7.53680994 11.04386408]
2023-04-07 19:31:22,585 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:31:29,477 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 6.86075695 11.39619935  9.73127668  1.9410739  4.5817
1172  5.32726997
  8.4766435  7.7299592  2.76077996 10.01962441  8.59230824  3.2896888 ]
2023-04-07 19:31:29,490 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:31:36,608 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [ 8.14588271  8.04259745  5.54460153  0.02702026 10.0062
8249  8.72600981
  7.04827595 11.17326991  2.17417255  3.58872294  6.27803292  9.35797312]
```



```
2023-04-07 19:31:36,620 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:31:43,594 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 4.0, best pos: [11.4506948 1.71225051 4.64855086 7.88354004 0.66452471 9.93576968
7.14807319 5.9092116 2.56629585 8.68161597 3.48646948 6.12656268]
2023-04-07 19:31:43,606 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:31:50,786 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 2.0, best pos: [ 5.83964737 7.50447405 1.68470928 4.26189696 10.52287303 8.23874232
6.17897172 2.90832526 9.32102127 11.30196944 0.81191102 3.03779624]
2023-04-07 19:31:50,797 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:31:58,020 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 6.23376714 4.3681353 2.00022653 7.08246859 3.80124201 9.47447349
8.00978672 10.05349153 5.81989707 0.04335711 10.00296656 1.88277076]
2023-04-07 19:31:58,030 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:32:05,206 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 4.0, best pos: [ 8.41565428 1.68887265 7.64499593 0.19809087 3.24855085 6.07010946
10.7749636 6.65419502 9.76700746 4.13593982 2.32377873 6.76090004]
2023-04-07 19:32:05,219 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=5
2023-04-07 19:32:12,218 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 5.0, best pos: [ 2.15021131 5.97480249 6.72366781 7.14842417 0.63819333 8.27198266
4.60009272 10.03625872 1.33377677 3.7349827 6.53026804 7.19861137]
2023-04-07 19:32:12,229 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:32:19,458 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 6.32062421 2.9002006 7.07536565 5.61759132 3.58608468 7.8385444
10.44217858 4.87859735 1.51998992 8.67221197 0.15986892 9.50072692]
2023-04-07 19:32:19,477 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:32:27,166 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 4.0, best pos: [ 9.61571929 7.87639495 4.56783022 9.37164715 3.39042307 9.35691066
6.30919084 2.64163562 5.77591796 9.33874867 11.54009725 0.8007403 ]
2023-04-07 19:32:27,176 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:32:34,118 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 4.0, best pos: [ 6.94287884 10.24310135 4.05851924 3.42908564 11.07381208 4.8550644
7.77685861 2.39959304 0.680493 5.06446709 1.20018589 8.36320462]
```

```
2023-04-07 19:32:34,144 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:32:41,581 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 7.31370197  6.48848536 11.89241578  2.81269325  5.7000
4579  9.02104092
 1.32629345  4.98362719  8.48295849  3.93552405  0.36626885  6.20239195]
2023-04-07 19:32:41,594 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:32:47,833 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [8.5673757  3.75592402 6.07320862 0.41356272 9.28314488
9.11524401
 1.6585688  5.72252162 8.02631692 2.10699127 4.55648899 7.16401395]
2023-04-07 19:32:47,843 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:32:54,171 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 6.05001409  3.65787199  1.86541404 11.69137793  9.4389
9937  7.91674282
 2.36089411 10.04975953 5.00920328 4.10234801 8.59036307 0.74439746]
2023-04-07 19:32:54,189 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:33:01,606 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 3.50465202  7.77715196 11.89540273  8.31624028  8.1062
1377  5.87218221
 1.96430041  8.78098191 10.45937139 0.31293334 6.70474576 4.73018674]
2023-04-07 19:33:01,621 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:33:08,767 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 4.887731   6.88507769  8.40724457 10.22545616  7.3739
5975  2.89232207
 6.20390977  9.04609123 1.24359417 6.71051375 5.98922387 3.1312312 ]
2023-04-07 19:33:08,782 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:33:16,048 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 4.0, best pos: [10.69946115  4.24504859 0.35163172  9.93040922  6.7918
4211  8.36959446
 11.40105544 1.8281516  3.89295682 5.598126  3.26819631 2.74409322]
2023-04-07 19:33:16,061 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:33:23,316 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 1.97365978 10.29659815 4.85612339 7.58783646 9.6804
0003  3.96495885
 0.63326328 2.30150309 5.50497262 8.65953946 6.41003067 4.97180067]
2023-04-07 19:33:23,329 - pyswarms.single.global_best - INFO - Optimize for 1000 ite
rs with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:33:30,810 - pyswarms.single.global_best - INFO - Optimization finished
| best cost: 3.0, best pos: [ 3.58338667  6.30777108 10.32003294 7.85514724 7.5786
9552  1.73074236
 4.55988815  9.22481999 5.44796025 2.14625626 7.02831287 11.72950579]
```

```
2023-04-07 19:33:30,827 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:33:38,210 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 2.0, best pos: [ 9.93759118  3.90237267 10.30124253  8.8405913   6.22410898  1.16313336
 4.80649022  0.97876562 11.91226221  7.40129714  5.64741917  2.84340866]
2023-04-07 19:33:38,222 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:33:45,287 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 4.98327915 11.45444304  1.45902094  8.87973464  6.34833717  0.89241046
 2.48130535 10.46079899  9.07862472  7.34527663  3.506026   8.22468866]
2023-04-07 19:33:45,300 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=5
2023-04-07 19:33:52,337 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 5.0, best pos: [ 7.71429612  5.20506621 10.08573462  5.95979042  5.29948386  4.76934676
 2.61366633  8.67968127 11.51446422  6.51535447  1.28386117  5.90804678]
2023-04-07 19:33:52,349 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:33:59,442 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 2.0, best pos: [ 4.96475265  2.8544172  11.6180026  5.64396798  8.56076477 10.19464044
 0.72511325  6.48461031  3.32555269  1.90579345  7.27920729  9.5828457 ]
2023-04-07 19:33:59,455 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:34:06,758 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 4.0, best pos: [ 4.01782084  0.39851948  3.90591405  7.70665615  7.97182231 10.61421116
 6.08878355  7.86888704  5.02375472 11.82977292  8.13343285  2.3493408 ]
2023-04-07 19:34:06,769 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:34:13,991 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 4.0, best pos: [ 2.42645195  5.74214878  3.70350874  9.11663012  1.74584883  3.51277439
 10.41654226  6.02545803  8.82449595 10.36544294  0.7873595  4.77894654]
2023-04-07 19:34:14,003 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:34:21,344 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 2.26678858  6.07517809 10.53224093  1.98482009  7.21995629  3.41544839
 9.32718028  9.25736541  5.98039112  8.69191363  4.2146193  11.02775329]
2023-04-07 19:34:21,355 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:34:28,563 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 7.99401191  5.86924812 10.18224091  1.61910365  4.12100161  4.10242244
 9.08945025  2.25989449  8.17635457 11.55676131  6.26751032  3.15159023]
```

```

2023-04-07 19:34:28,574 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=4
2023-04-07 19:34:36,283 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 4.0, best pos: [ 6.58798205  1.78526722  7.63383257  8.03276333  3.87920058  6.36420748
      8.30112491 10.91101165  0.80338969  2.53161777  5.62015824  9.20241106]
2023-04-07 19:34:36,295 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=2
2023-04-07 19:34:43,736 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 2.0, best pos: [ 7.30906197  2.35374354  4.60069112  1.06080515  9.96362762  5.17876273
      3.29046497 11.09914245  0.09822819  8.29451329  6.08670816  3.53173044]
2023-04-07 19:34:43,747 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:34:51,076 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 2.78001349  6.79865655  9.39856714  1.84555903  8.34670409  5.04818489
      3.09296408 11.283457   7.23075275 10.54732147  4.9878586   6.10255759]
2023-04-07 19:34:51,090 - pyswarms.single.global_best - INFO - Optimize for 1000 iterations with {'c1': 0.5, 'c2': 0.3, 'w': 0.9}
pyswarms.single.global_best: 100%|██████████|1000/1000, best_cost=3
2023-04-07 19:34:58,285 - pyswarms.single.global_best - INFO - Optimization finished | best cost: 3.0, best pos: [ 8.42238038  9.33875212 11.6401842   4.81125228  0.47694913  9.39092606
      9.48771578  7.54300109 10.2144953   6.86526021  2.79200006  5.85089916]
Average time (sec):  6.50140148639679
PSO solved 0/100

```

Summary

Summing up the tests, the genetic algorithm fits the n queens problem much better than PSO. GA found really good solutions in a fairly short time and even for large inputs it kept about 50% of the correctness of the solutions, while PSO could not cope with them at all.

Bibliography

- <https://pyswarms.readthedocs.io/en/latest/api/pyswarms.single.html>
- https://en.wikipedia.org/wiki/Particle_swarm_optimization
- https://en.wikipedia.org/wiki/Eight_queens_puzzle
- https://pygad.readthedocs.io/en/latest/README_pygad_ReadTheDocs.html#pygad-ga-class
- https://en.wikipedia.org/wiki/Genetic_algorithm
- <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>
- <https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/>