A comparative analysis between Particle Swarm Optimization and other nature inspired methods

Anda Buinoschi MOC2, Bogdan Luncaşu MOC1

Abstract

Particle Swarm Optimization gains its place between the most used methods for solving optimization problems due to its simplicity and its power to approximate good solutions. Inspired by swarms of biological populations, this method can converge to global optima in numerous problems. This paper proposes to create a comparative analysis between this method, Hill Climbing, a genetic algorithm and a hybrid method created from last two. Our results can prove that this method can find better results than the others using less computations.

1 Introduction

Nature finds its way to amaze us through its simplicity and its power to evolve. There is no surprise that there are optimization methods inspired from it. In this paper we propose to compare four of these methods: Particle Swarm Optimization, Hill Climbing, a genetic algorithm and a hybrid Hill Climbing initialised with genetic algorithm solution. While Hill Climbing is a local iterative search through the neighbourhood of a candidate solution and a genetic algorithm found its source in genetics and biology combining chromosomes (through crossover operator) during a generation and applying mutations to get better results, Particle Swarm Optimization is inspired from flock of birds or colonies of insects.

2 Methods

2.1 Particle Swarm Optimization

2.1.1 Algorithm

While Hill Climbing and genetic algorithms work on binary representations of the candidates, Particle Swarm Optimization works on real numbers in the function definition interval. The idea of the method is to simulate the social interaction between the biological structures in a group of birds or a shoal of fish which are looking for food. In this way, the individuals conserves more energy while they meet their primary needs. Each particle of the algorithm

represents an individual in such a social group being represented by its position at a respective time and its speed. Its velocity changes over time by taking into account the following factors: its previous velocity (since none of the individuals can immediately change its moving direction), its previous known position (since the fish can remember the place where it found food before, tending to go again there) and the position of the entire group (if there may be the most of the individuals in the group, then they might have found food). The pseudocode of the algorithm can be written as:

Algorithm 1: Particle Swarm Optimization algorithm

```
t = 0;
generate P(t) (positions x and velocity arrays v);
initialize g (the swarms best known position);
p = P(t) where p represents the particle's best known position;
while not stopping condition do
    t = t + 1;
    for particle in P(t) do
        v[t + 1] = w_1 · v[t] + w_2 · rand()(p − x) + w_3 · rand()(g − x);
        x[t + 1] = x[t] + v[t + 1];
    evaluate P(t);
    update p,g;
end
```

In the algorithm function $rand()$ is a function which returns a number from a uniform distribution over the interval $(0, 1)$. The parameters w_1, w_2, w_3 have the role of learning factors. While w_1 represents the inertia of an individual, w_2 represents a cognitive factor since it has the tendency to reproduce past actions that proved to be a good result and w_3 is the social factor which drives the particle to the best known solution of the entire swarm.

2.1.2 Parameters

The most important parameters of this algorithm are the inertia w_1, cognitive and social constants w_2 and w_3, maximum iteration number and computed precision ε where the last two can be stopping criterias.
The inertia factor w_1 influences the current velocity from the previous iteration. If this factor would be 0, then the particle will be driven only by its best position and the swarm's best position. But if $w \neq 0$, then the particle will explore new spaces. The higher this value, the bigger will be the particle's space in the search area (it will explore more); but if it's a low value, then the particle will tend to explore local neighbourhood (exploitation). Tipically, inertia factor is chosen to be in the interval $[0.4, 0.9]$.
The constants w_2 and w_3 represents the particle's acceleration rate to its personal best position or to the population's best. If we define $w_2 = 0$ and $w_3 = 0$, the velocity will be based only on particle's inertia, which would mean it would move around with the current speed until the border. If these factors would be low, then the particle will wonder through the search space and skip the optima

and if these values would be too large, the particle would be induced to quickly reach the goal area. If the cognitive factor $w_2 = 0$, the particle will not have memory which means that it is in a social environment and can reach to new search space from the interactions with the other particles. The lack of social factor w_3 can drive the particle based only on its inertia and its cognitive memory into local. This would be equivalent to single particles with no interaction between them, in the end, not in a structured social environment. Usually, the cognitive factor w_2 can take values in the interval $[1.5, 2]$ and the social factor w_3 can be in the interval $[2, 2.5]$.

2.2 Testing functions

2.2.1 Griewangk function

The Griewangk function has the following form:

$$f(x) = \sum_{i=1}^{n} \frac{{x_i}^2}{4000} - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, where - 600 \leq x_i \leq 600.$$

It's global minimum optima is: $f(x) = 0, x(i) = 0, i = 1 : n$. When we look into the entire interval of the function, we can suspect that it is easy for an algorithm to find the global optima, but if we would zoom in on a smaller interval, we assume that the algorithm can reach into a local optima.

2.2.2 Rastrigin function

Rastrigin function has the following definition:

$$f(x) = 10 \cdot n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cdot cos(2 \cdot \pi \cdot x_i)\right), -5.12 \leq x_i \leq 5.12$$

having the global optima of: $f(x) = 0, x(i) = 0, i = 1 : n$.

2.2.3 Rosenbrock function

The Rosenbrock function is the following:

$$f(x) = \sum_{i=1}^{n-1} 100 \cdot \left(x_{i+1} - x_i^2\right)^2 + \left(1 - x_i\right)^2, -2.048 \leq x_i \leq 2.048$$

with the global optima $f(x) = 0, x(i) = 1, i = 1 : n$.

2.2.4 Six-hump Camelback

The Six-hump Camelback has the following definition:

$$f(x) = \left(4 - 2.1 \cdot x_1^2 + \frac{x_1^4}{3}\right) \cdot x_1^2 + x_1 \cdot x_2 + \left(-4 + 4 \cdot x_2^2\right) \cdot x_2^2,$$

$$-3 \leq x_1 \leq 3, -2 \leq x_2 \leq 2$$

and its global optima is

$$f(x_1, x_2) = -1.0316, (x_1, x_2) = (-0.0898, 0.7126), (0.0898, -0.7126).$$

3 Experiments

For our experiments we used convergence as stopping criteria (if the value of the best particle g is not changed in the last 5 steps with a precision of 3, then we assume that the algorithm converged to a minimum value). Also, we kept the swarm size of 60 particles, each particle having 30 coordinates, and we ran each parameter configuration for 50 times.

Inertia	Cognitive	Social	mean	std	min	max
0.4	1.5	2	13.1	0.07	13.08	13.25
0.4	1.5	2.2	6.12	0.2	5.84	6.37
0.4	1.5	2.5	90.67	0.12	90.56	90.85
0.4	1.7	2	11.44	0.29	11.28	11.99
0.4	1.7	2.2	3.28	0.01	3.27	3.31
0.4	1.7	2.5	90.94	7e-3	90.93	90.96
0.4	2	2	3.04	0.2	2.89	3.33
0.4	2	2.2	13.99	2.38	6.58	15.34
0.4	2	2.5	2.19	0.01	2.18	2.25
0.6	1.5	2	0.16	2e-3	0.15	0.16
0.6	1.5	2.2	0.07	5e-4	0.07	0.07
0.6	1.5	2.5	0.12	2e-4	0.12	0.12
0.6	1.7	2	90.95	7e-3	90.94	90.96
0.6	1.7	2.2	0.3	0.03	0.29	0.45
0.6	1.7	2.5	180.98	1e-4	180.98	180.98
0.6	2	2	0.12	2e-3	0.12	0.13
0.6	2	2.2	0.31	8e-3	0.3	0.33
0.6	2	2.5	0.019	1e-4	0.019	0.02
0.8	1.5	2	90.40	5e-3	90.4	90.44
0.8	1.5	2.2	0.14	0.38	0.05	2.28
0.8	1.5	2.5	5.06	3.6	2.26	14.57
0.8	1.7	2	1.5e-4	4e-4	3.05e-11	2.2e-3
0.8	1.7	2.2	0.18	0.38	2.4e-6	1.21
0.8	1.7	2.5	91.63	0.15	91.47	92.45
0.8	**2**	**2**	**6.78e-5**	**2.1e-4**	**3.69e-11**	**1.2e-3**
0.8	2	2.2	180.4	0.07	180.3	180.7
0.8	2	2.5	93.34	1.13	92.13	95.11

Table 1: Value results on Griewangk function

The above table containing the results on Griewangk function tells us that the most suited configuration for this function is $w_1 = 0.8, w_2 = 2, w_3 = 2,$

meaning that this function needs a high value for inertia (because it needs exploring), the particles needs a high cognitive factor, to remember a better position in the search space with a lot of local points and a low social factor to prevent them to converge to a local minima.

Inertia	Cognitive	Social	mean	std	min	max
0.4	1.5	2	262.72	6.8e-3	262.71	262.74
0.4	1.5	2.2	140.89	0.019	140.87	140.92
0.4	1.5	2.5	192.51	0.22	192.17	192.68
0.4	1.7	2	184.62	0.25	184.37	184.88
0.4	1.7	2.2	182.65	4.9e-3	182.65	182.66
0.4	1.7	2.5	204.03	2.2e-3	204.03	204.04
0.4	2	2	239.27	0.23	239.14	239.7
0.4	2	2.2	195.07	0.18	194.66	195.16
0.4	2	2.5	209.25	8.2e-3	209.25	209.27
0.6	1.5	2	156.4	2.9	153.29	159.12
0.6	1.5	2.2	171.27	0.013	171.27	171.34
0.6	1.5	2.5	180.3	2.9e-4	180.3	180.301
0.6	1.7	2	242.05	4.7e-4	242.058	242.059
0.6	1.7	2.2	147.39	8.6e-4	147.395	147.398
0.6	1.7	2.5	169.28	1.3e-4	169.284	169.285
0.6	2	2	134.47	0.014	134.44	134.49
0.6	2	2.2	153.29	2.2e-4	153.29	153.29
0.6	2	2.5	185.34	2.3e-4	185.345	185.346
0.8	1.5	2	98.81	0.29	98.71	99.71
0.8	1.5	2.2	142.45	3.91	139.45	155.56
0.8	1.5	2.5	218.42	7.04	210.89	225.01
0.8	**1.7**	**2**	**37.97**	**0.29**	**37.87**	**38.87**
0.8	1.7	2.2	255.21	0.032	255.2	255.3
0.8	1.7	2.5	244.28	10.08	231.034	251.95
0.8	2	2	160.47	4e-4	160.471	160.473
0.8	2	2.2	144.59	9.77	136.45	160.04
0.8	2	2.5	222.87	19.96	200.04	240.33

Table 2: Value results on Rastrigin function

The table 2 showcases the results on Rastrigin function and it tells us that the most suited configuration for this function is $w_1 = 0.8, w_2 = 1.7, w_3 = 2$, meaning that this function needs a cognitive behaviour and a high inertia inside the swarm, rather than social, to find a good optima. Since this function has a lot of local minimas, it is understandable why it gets stuck in such points.

The table 3 containing the results on Rosenbrock function tells us that the most suited configuration for this function is $w_1 = 0.6, w_2 = 1.7, w_3 = 2.2$, meaning that this function needs a social and cognitive behaviour inside the swarm and inertia to find a good optima. The particles exploit the best position known so far but at the same time they explore the search space.

Inertia	Cognitive	Social	mean	std	min	max
0.4	1.5	2	218.47	0.54	218	219.2
0.4	1.5	2.2	279.34	6e-3	279.33	279.35
0.4	1.5	2.5	31.07	0.059	31.02	31.17
0.4	1.7	2	148.3	0.74	147.75	149.28
0.4	1.7	2.2	93.26	3.19	90.19	101.86
0.4	1.7	2.5	79.60	0.32	79.46	80.45
0.4	2	2	601.72	1.35	601.36	607.10
0.4	2	2.2	136.4	0.03	136.37	136.46
0.4	2	2.5	27.25	0.15	27.018	27.86
0.6	1.5	2	541.41	0.54	541.07	544.03
0.6	1.5	2.2	4.03	0.017	4	4.055
0.6	1.5	2.5	3.98e-2	0.011	0.024	0.064
0.6	1.7	2	25.20	0.012	25.18	25.23
0.6	**1.7**	**2.2**	**3.85e-2**	**0.0184**	**0.022**	**0.091**
0.6	1.7	2.5	814.52	5.8e-4	814.52	814.52
0.6	2	2	209.86	1.78	208.78	214.15
0.6	2	2.2	856.60	2.70	850.53	859.66
0.6	2	2.5	0.051	0.020	0.026	0.1011
0.8	1.5	2	432.31	4.33	425.25	440.70
0.8	1.5	2.2	201.07	0.63	200.48	202.45
0.8	1.5	2.5	302.46	236.91	90.96	615.13
0.8	1.7	2	10.66	22.93	0.42	75.36
0.8	1.7	2.2	76	0.78	75.28	78.82
0.8	1.7	2.5	987.28	45.34	952.99	1220.72
0.8	2	2	4.33	0.592	4.093	7.85
0.8	2	2.2	193.49	0.599	192.62	195.22
0.8	2	2.5	970.46	42.34	935.34	1058.33

Table 3: Value results on Rosenbrock function

For Six Hump Camel Back function, every configuration leads to global optima with the $mean = -1.0316284534898772, std = 2.220446049250313e-16, min = -1.0316284534898774, max = -1.0316284534898774$. This can be explained by the reduced dimensionality of the space (having two variables) and the population can converge to this global value, either in a point or another.

4 Comparative results

The Particle Swarm Optimization has the big advantage that it doesn't need a representation of the space for creating candidate solutions, in contrast to Hill Climbing and genetic algorithms which both works in binary representations. This leads to less evaluations for PSO which means less computational time than in the other two methods.
For Griewangk function the best solution was found using Hill Climbing with

best improvement neighbourhood search ($mean = 2.56 \cdot 10^{-9}, min = 2.56 \cdot 10^{-9}, max = 2.56 \cdot 10^{-9}$.
For Rastrigin function, the best solution was found using the hybrid method (running Hill Climbing with genetic algorithm's solution) with the result $mean = 32.16, min = 24.30, max = 39.43$.
For Rosenbrock function, the most promising solution was found using Particle Swarm Optimization with $mean = 3.85 \cdot 10^{-2}, min = 0.022, max = 0.091$. And each of the methods compared in this paper could find the global minima for Six Hump Camel Back.

5 Conclusion

In conclusion, we can resume that there is no method which fits extremely well on each type of the problem. It depends on search space and finding the good parameters for the method which can be tough work. But using such nature inspired algorithms, you can approximate optimal solutions for various type of problems in a timely manner.

References

[1] Melanie Mitchell, "An introduction to genetic algorithms", 1998
https://www.academia.edu/12824545/An_Introduction_to_Genetic_Algorithms_-_Melanie_Mitchell

[2] Breaban Mihaela, Nature Inspired Methods class course, 2020
https://profs.info.uaic.ro/~pmihaela/MOC/trajectory.html

[3] Wikipedia site for Selection, Mutation and Crossover
https://en.wikipedia.org/wiki/Selection_(genetic_algorithm)
https://en.wikipedia.org/wiki/Mutation_(genetic_algorithm)
https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm

[4] Wikipedia site for Particle Swarm Optimization
https://en.wikipedia.org/wiki/Particle_swarm_optimization

[5] MIT Open Course, "A Basic Introduction to Genetic Algorithms", Prof. Olivier de Weck, 2010
https://ocw.mit.edu/courses/institute-for-data-systems-and-society/ids-338j-multidisciplinary-system-design-optimization-spring-2010/lecture-notes/MITESD_77S10_lec11.pdf
https://ocw.mit.edu/courses/institute-for-data-systems-and-society/ids-338j-multidisciplinary-system-design-optimization-spring-2010/lecture-notes/MITESD_77S10_lec12.pdf