Confinements and Biases in Particle Swarm Optimisation

Maurice Clerc

Confinements and Biases in Particle Swarm Optimisation

Maurice Clerc

Maurice.Clerc@WriteMe.com

ABSTRACT. All PSO versions do present one or more biases, often in favor of the center of the search space. An important factor that induces such biases is the method used to keep particles inside the search space. We compare here nine methods on a few benchmark functions, and the results suggest another one which is less biased. Furthermore this study also suggests how to adaptively modify the search space for each move. Thanks to these two simple modifications the resulting PSO is both more robust and more effective.

KEY WORDS: particle swarm, optimisation, bias, clamping, confinement, adaptive search space

1. Introduction

In Particle Swarm Optimisation sometimes a particle tends to leave the search space. Several methods are used to prevent this, but they all induce a bias, most of the time in favour of the centre of the search space and also quite often in favour of the boundaries. We can see that by progressively "shifting" a few classical benchmark functions and by comparing the results, success rate or best mean value, over a given number of runs. Fortunately not all methods give the same biases, so it is possible two combine two of them in order to obtain a less biased one.

2. Exposing the bias

In [1] C. Monson compare some very different PSO algorithms (i.e. having different velocity update equations) in order to see if they are biased and how much. He finds that they all have an origin-seeking bias, more or less important.

Here the approach is a bit different. We start from the standard PSO algorithm, as it is given on the Particle Swarm Central [2], and we just modify the method that prevents particles to leave the search space, i.e. the confinement method. As we will see, there is usually not just one bias, and not necessarily on the centre of the search space.

Standard PSO

For more details, see the Particle Swarm Central <ref>. In short, this version is very near of the original one, with just a few improvements. So, the main features are the following ones:

- constant swarm size, given by the formula $S=10+2\mathrm{sqrt}(2\mathrm{D})$ where D is the dimension of the search space
- neighbourhood redefined at random after each unsuccessful iteration (i.e. without improvement of the best solution). In order to do that each particle informs itself and choose at random K others that it will inform. In the standard, $K=3$. Note that it does not mean that the neighbourhood of each particle is of size 3. It can take any value between 1 (just itself) and S (the whole swarm).
- For each particle and for each dimension the next move is computed according to the following pseudo-code, with the usual notations:

$$v(t+1) \leftarrow \omega\, v(t) + rand(0..c)(p(t) - x(t)) + rand(0..c)(p_g(t) - x(t))$$
$$x(t+1) \leftarrow x(t) + v(t+1)$$
$$if\ x(t+1) > x_{max}\ \ then\ \ x(t+1) \leftarrow x_{max};\ v(t+1) \leftarrow 0$$
$$if\ x(t+1) < x_{min}\ \ then\ \ x(t+1) \leftarrow x_{min};\ v(t+1) \leftarrow 0$$

The two last instructions define the confinement method. Let us call it Confinement 0 (Standard) and let us define now some others. For simplicity the formulas are given just for dimension 1.

Confinement -2 (No confinement)

No confinement at all. Of course, it is possible only if the function is defined outside the search space., which is the case in the examples below.

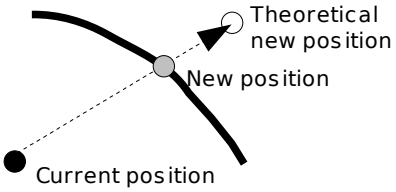
Confinement -1 (No confinement + artificial landscape)

Still no confinement, but outside the search space the fitness is defined by a linear increasing function.

$$if \;\; x(t+1) > x_{max} \;\; then \;\; f(x) \leftarrow f(x_{max}) + x(t+1) - x_{max}$$
$$if \;\; x(t+1) < x_{min} \;\; then \;\; f(x) \leftarrow f(x_{min}) + x_{min} - x(t+1)$$

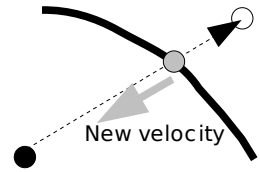
Confinement 0 (Standard)

We have already seen the formulas. Here is the diagram.



Confinement 1 (Deterministic Back)

$$if\, x(t+1) > x_{max} \;\; then \;\; x(t+1) \leftarrow x_{max};\, v(t+1) \leftarrow -\gamma\, v(t+1)$$
$$if\, x(t+1) < x_{min} \;\; then \;\; x(t+1) \leftarrow x_{min};\, v(t+1) \leftarrow -\gamma\, v(t+1)$$

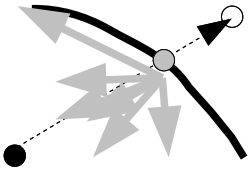


In the following tests we choose $\gamma = 0.5$.

Confinement 2 (Random Back)

$$if\, x(t+1) > x_{max} \;\; then \;\; x(t+1) \leftarrow x_{max};\, v(t+1) \leftarrow -rand(0..1)\, v(t+1)$$
$$if\, x(t+1) < x_{min} \;\; then \;\; x(t+1) \leftarrow x_{min};\, v(t+1) \leftarrow -rand(0..1)\, v(t+1)$$

Note that the random coefficient is different for each dimension.



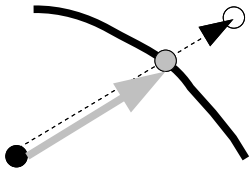
Confinement 3 (Consistent)

Just like Standard, but the velocity is modified so that after clamping we still have

$$x(t+1) \leftarrow x(t) + v(t+1)$$

The confinement is then given by:

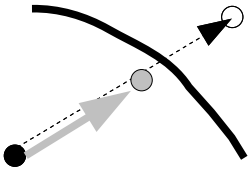
$$if\ x(t+1) > x_{max}\ \ then\ \ x(t+1) \leftarrow x_{max};\ v(t+1) \leftarrow x_{max} - x(t)$$
$$if\ x(t+1) < x_{min}\ \ then\ \ x(t+1) \leftarrow x_{min};\ v(t+1) \leftarrow x_{min} - x(t)$$



For the following confinements (4, 5, and 6) the velocity is modified so that when applied to the position we are sure that the new position is inside the search space.

Confinement 4 (Hyperbolic)

$$if\ v(t+1) > 0\ then\ v(t+1) \leftarrow \frac{v(t+1)}{1 + \left| \frac{v(t+1)}{x_{max} - x(t)} \right|}$$
$$else\ \ v(t+1) \leftarrow \frac{v(t+1)}{1 + \left| \frac{v(t+1)}{x(t) - x_{min}} \right|}$$



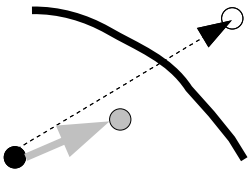
Confinement 5 (Relativity Like)

Just for fun, and based on the famous formula that adds two velocities according to a maximum velocity v_{max} :

$$add(v_1, v_1) = \frac{v_1 + v_2}{1 + \frac{v_1 v_2}{v_{max}^2}}$$

$$v(t+1) \leftarrow add(\omega\, v(t), add(rand(0..c)(p(t) - x(t)), rand(0..c)(p_g(t) - x(t))))$$

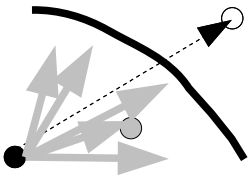
Note that this operation is not commutative. However preliminary tests seem to show that the six possibilities give equivalent results.



Confinement 6 (Random Forth)

This one is coming from [3]

$$if \quad v(t+1)+x(t)>x_{max} \quad then \quad v(t+1)\leftarrow rand(0..(x_{max}-x(t)))$$
$$if \quad v(t+1)+x(t)<x_{min} \quad then \quad v(t+1)\leftarrow rand(0..(x(t)-x_{min}))$$



Results

We now try these methods on classical functions:

	Search space	Max. number of eval.
Parabola/Sphere	$[-100,100]^{30}$	9000
Griewank	$[-100,100]^{30}$	9000
Rosenbrock	$[-9,11]^{30}$	40000
Rastrigin	$[-10,10]^{30}$	40000

Table 1: Benchmark functions

All search spaces are defined so that the solution is right on the centre, even for Rosenbrock. When the success rate (over 100 runs) is not null, it is given. If it is null, the mean best value is given: the smaller the better.

	No confinement	No confinement + artificial landscape	Standard	Deterministic back γ=0.5	Random back
Parabola/Sphere	10%	4%	26%	6%	16%
Griewank	30%	20%	34%	44%	36%
Rosenbrock	34,2	36,2	35,0	34,8	30,8
Rastrigin	59,6	60,1	58,4	54,5	59,4

	Consistent	Hyperbolic	Relativity like	Random forth	Hybrid Hyperbolic+ Random back
Parabola/Sphere	16%	95%	72%	12%	86%
Griewank	38%	53%	40%	38%	42%
Rosenbrock	39,6	30,2	24,9	32,1	32,7
Rastrigin	56,9	39,7	40,7	56,7	46,8

Table 2: Offset = 0%. Results with ten confinements methods

Table 2 show the result (also for a hybrid method, which is explained below). As we can see "Hyperbolic" confinement is extremely good. Too good to be honest, in fact, so we will now try with a quite big offset of 90%. It means, for example, that the solution for Griewank is on (90, 90,...,90).

	No confinement	No confinement + artificial landscape	Standard	Deterministic back γ=0.5	Random back
Parabola/Sphere	0,01	0,02	25%	4%	25%
Griewank	4%	12%	15%	48%	50%
Rosenbrock	28,6	34,9	25,7	23,0	23,9
Rastrigin	81,7	64,7	28,0	38,2	37,8

	Consistent	Hyperbolic	Relativity like	Random forth	Hybrid Hyperbolic+ Random back
Parabola/Sphere	32%	59%	50%	32%	74%
Griewank	42%	27%	35%	18%	36%
Rosenbrock	24,0	23,1	22,0	23,7	23,4
Rastrigin	34,7	302,1	235,1	30,1	96,5

Table 3: Offset=90%. Results with ten confinements methods

Now results are quite different. In particular "no confinement" with or without artificial landscape is not a good choice. The two best ones are now "Random back" confinement, and "Consistent" confinement. We can also already see that the bias is not always in favour of the centre of the search space. We now more carefully study what happens when progressively modifying the offset, from 0% to 100%. We have do that for all quite "contrasted" pairs of confinement methods, but (Hyperbolic, Random back) is the most interesting one. Results are on figures 1 to 4.

These figures also show the results obtained with a "hybrid" method, i.e. a combination of the two other ones, and with an adaptive one, as it is explained below.

3. Reducing the bias

Principle of the hybrid method

Each particle, at each time step chooses at random (50/50) between two confinement methods. When using Random Back and Hyperbolic, we find the curves shown on figures 1 to 4.

Comments on results

All confinements methods have the same kind of bias for Rosenbrock: worse near of the centre of the search space. So, of course, the hybrid one does the same. However for the other problems, it does better. For example Random Back is good for Rastrigin and bad for Parabola/Sphere, as it is the contrary for Hyperbolic. So the hybrid gives acceptable results for all functions. It is far more robust than the other "single" methods.

Conclusion and future work

Studying in detail some biases of several confinement methods in Particle Swarm Optimisation suggested us to use a hybrid one, more robust. This modification of the standard PSO give really better results, particularly for difficult functions. However standard PSO by itself is not the best on the market and, anyway, it is still a parametric version. So the idea is now to include this modification in the parameter free PSO called TRIBES [4]. As TRIBES already uses several strategies it should be easy to add new ones, or to replace some of them by new ones.

References

[1] C. K. Monson and K. D. Seppi, "Exposing Origin-Seeking Bias in PSO," presented at GECCO'05, Washington, DC, USA, 2005, pp. 241-248.
[2] Particle Swarm Central, http://www.particleswarm.info
[3] X.-F. Xie, W.-J. Zhang, and Z.-L. Yang, "Varying the Topology and the Probability of Adding Perturbations in Dissipative Particle Swarm Optimisation," submitted at GECCO 2006.
[4] M. Clerc, Particle Swarm Optimization: ISTE (International Scientific and Technical Encyclopedia), 2006, translated from: M. Clerc, L'optimisation par essaims particulaires. Versions paramétriques et adaptatives: Hermès Science, 2005.

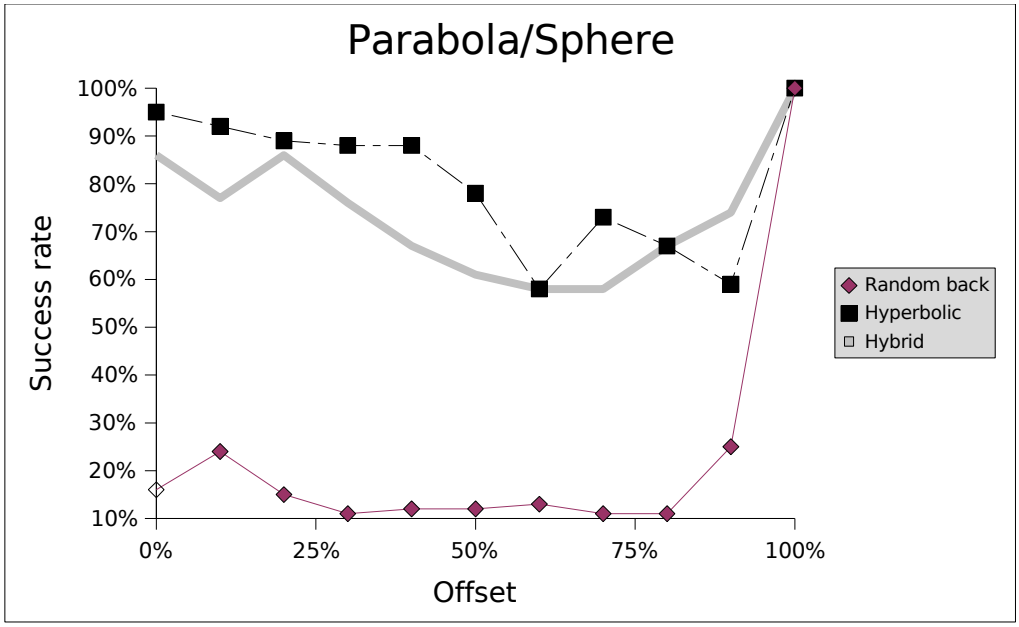


Illustration 1: Parabola/Sphere - Success rate versus (diagonal) offset

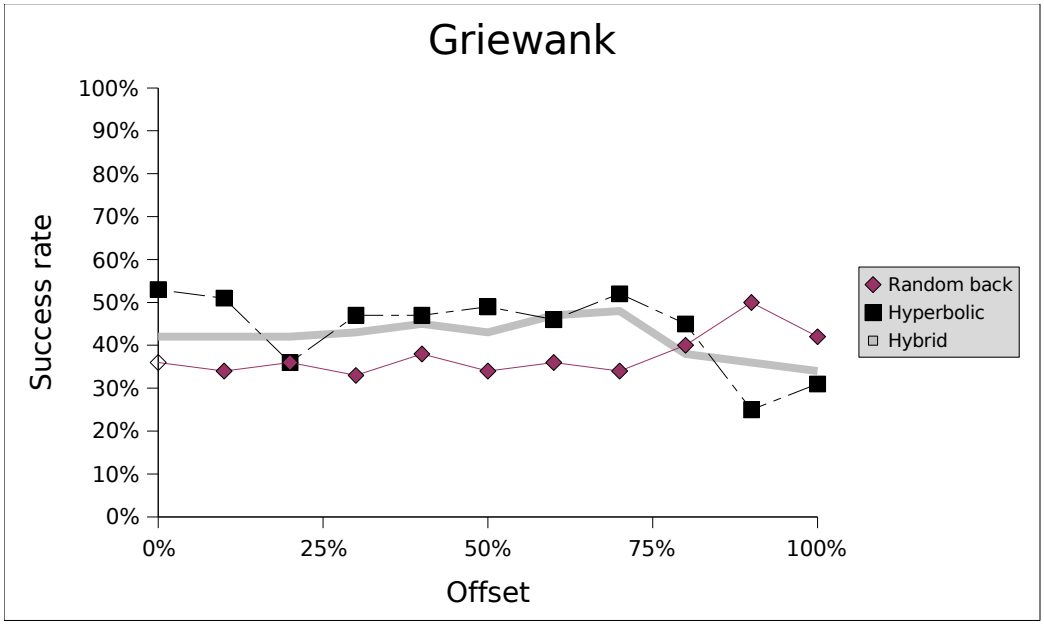


Illustration 2: Griewank - Success rate versus offset

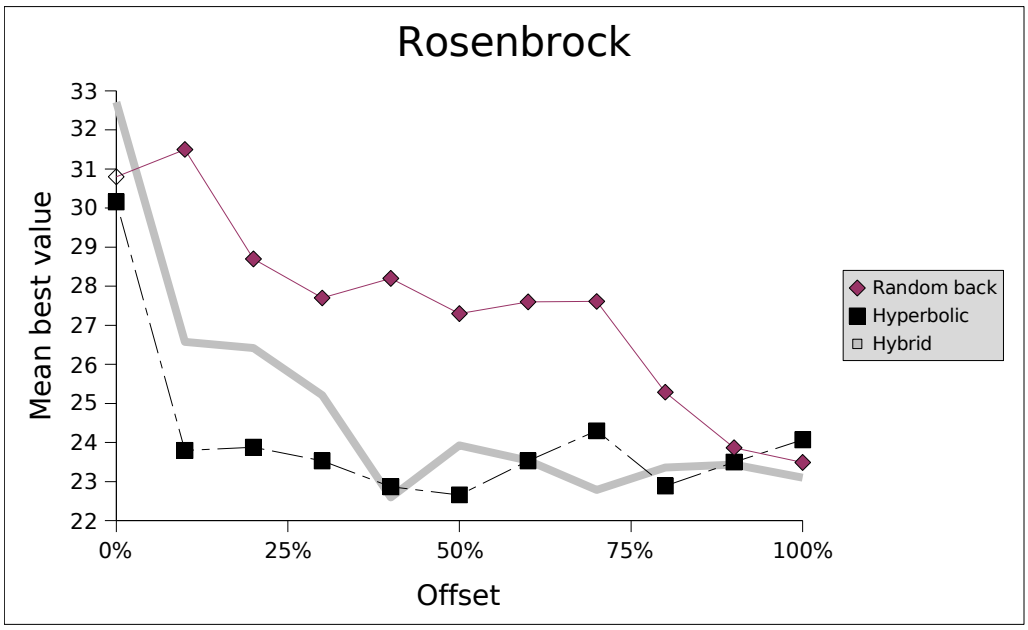


Illustration 3: Rosenbrock - Best mean valus vs offset (the smaller the better)

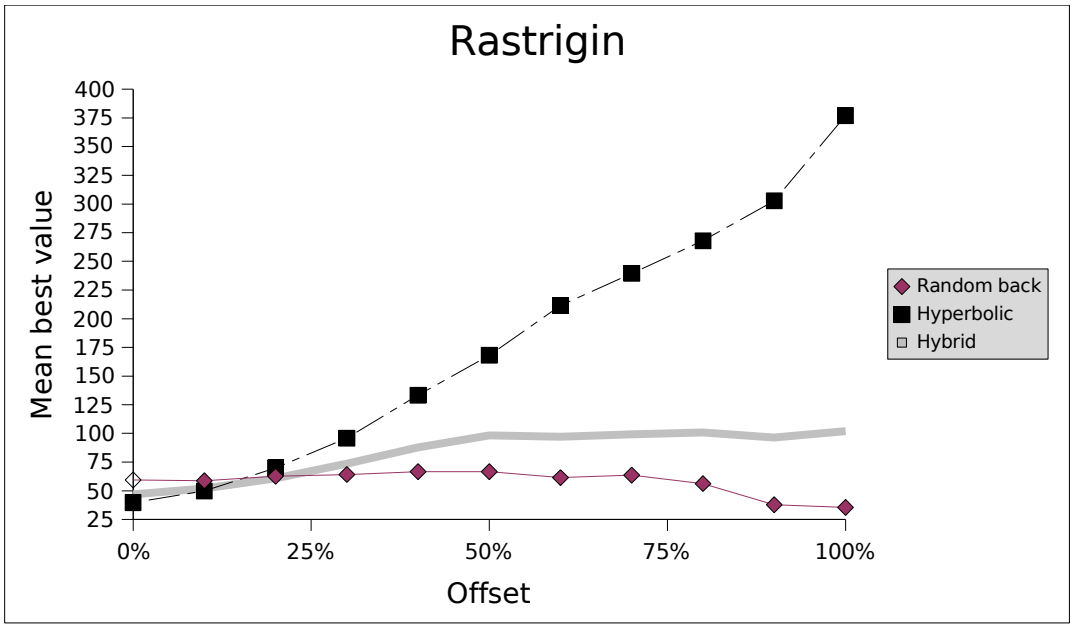


Illustration 4: Rastrigin - Best mean value vs offset (the smaller the better)