The mythical balance, or When Particle Swarm Optimisation does not exploit

Maurice Clerc

2008-10-31

1 A ritual claim

In a lot of PSO papers, we can read a claim like "this algorithm ensures a good balance between exploration and exploitation". Sometimes the authors use the words "diversification" and "intensification", but the idea is the same. However a precise definition of "exploitation" is never given. To tell the truth, I have seen once such a definition, in a submitted paper, but it was meaningless, for it was not even sure that the so defined "exploitation area" contains the best position found by now.

So, in this paper, I suggest a rigorous and "reasonable" definition, and, according to it, I show that Standard PSO 2007 (SPSO07), available on the Particle Swarm Central [3], does not really exploit. After that, of course, I also suggest variants that may improve it.

2 Defining exploitation

Roughly speaking, exploitation means "searching around a good position". In PSO, good positions are already defined: each particle memorises the best position it has found. So, it seems natural to define a *local exploitation area (LEA)* around each of these positions. Now, of course, we have to give a mathematical formulation of "around".

Let us consider first a 1D search space $[x_{min}, x_{max}]$, and the following points $(p_0, = x_{min}, p_1, \ldots, p_N, p_{N+1} = x_{max})$, where $(p_1, ..., p_N)$ are the local best positions found by the swarm. If S is the swarm size, we have $N \leq S$. We may have $x_{min} = p1$and/or $p_N = x_{max}$,but it is not a problem. Now we need a parameter $\rho > 0$, which can be called *relative local size*, although its role is a bit more complicated that this name could suggest. Better to carefully examine the following formula. Indeed, we define N local exploitation areas by

$$e_i = [p_i - \rho(p_i - p_{i-1}), p_i + \rho(p_{i+1} - p_i)], i \in \{1, \cdots N\} \tag{1}$$

So, for each p_i, we have now an interval that contains it. Note that, usually, $p_i$is not the centre of this interval. Note also that, as soon as $\rho \geq 0.5$, the union of the local search areas is just one connected domain (here an unique interval). More generally, when the dimension is D, the definition is straightforward. Each local best position is now a vector $p_i = (p_{i,1}, \ldots, p_{i,D})$.On each dimension d we define the intervals

$$e_{i,d} = [p_{i,d} - \rho(p_{i,d} - p_{i-1,d}), p_{i,d} + \rho(p_{i+1,d} - p_{i,d})], i \in \{1, \cdots N\}, d \in \{1, \ldots D\} \tag{2}$$

The local exploitation areas around each best position are now hyperparallelepids, defined by the Cartesian product

$$E_i = e_{i,1} \otimes \ldots \otimes e_{i,D}, i \in \{1, \cdots N\} \tag{3}$$

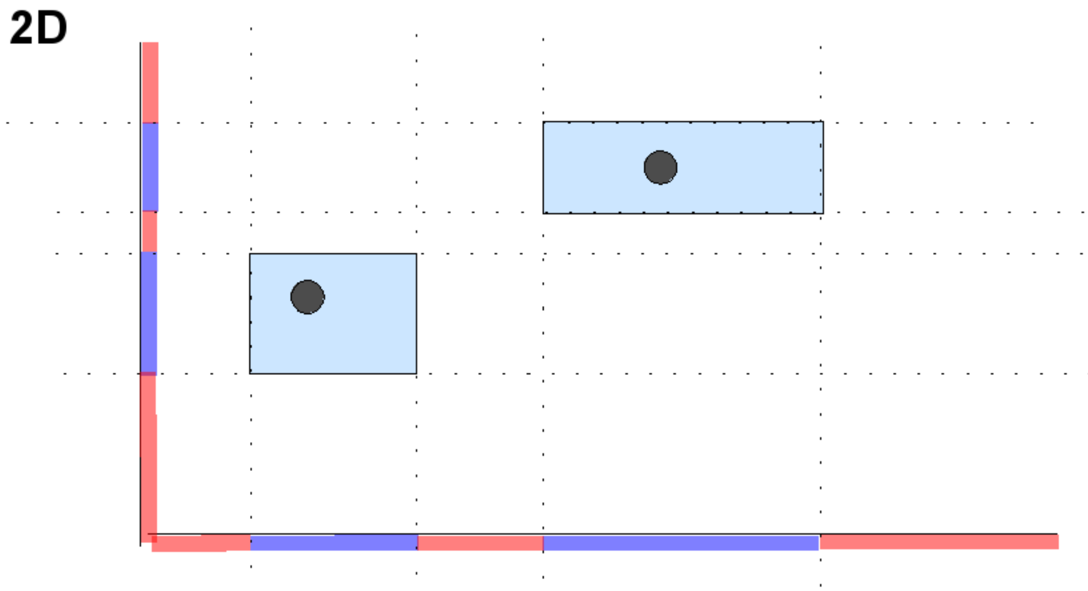

Figure 1: 2D example of Local exploitation areas, with $\rho = 1/3$

Here, if $\rho > 0.5$, all that can we say about the union of the E_i is that each projection on any dimension is just one connected domain. The figure 1 shows an example for a two dimensional search space, and two known best positions.

3 Checking the exploitation rate

At each iteration t, let $S_E(t)$ be the number of particles that are inside a local exploitation area. We can define the *exploitation rate* by

$$r(t) = \frac{S_E(t)}{S} \tag{4}$$

It is now easy to observe, from this point of view, the behaviour of an algorithm like SPSO07, i.e. the evolution of the exploitation rate, on some test functions. On the figure 2 one can see a typical example: there is no clear tendency, except maybe, and not surprisingly a very slight increase when the swarm has converged. Also, and more important, the mean value is quickly decreasing when the dimension increases, as we can better see on the figure 3. In practise, as soon as the dimension is greater than 6, there is no exploitation at all. And this is true no matter what the value of ρ. That is why the “good balance” often claimed for PSO is just a myth (at least for SPSO07).

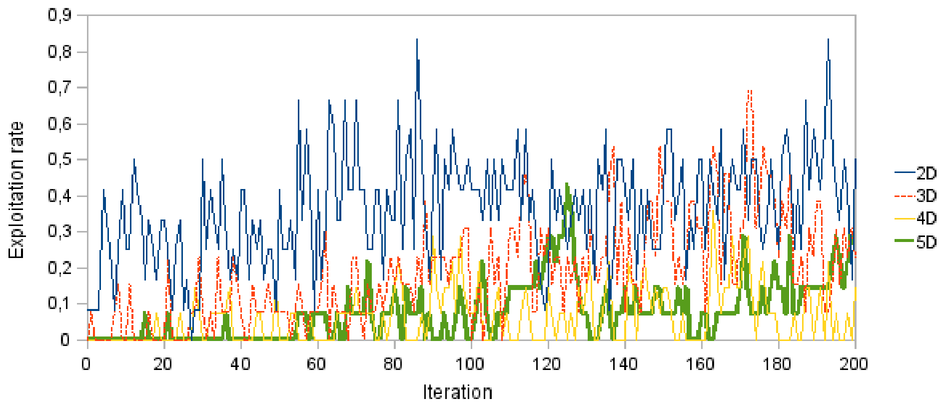

Figure 2: Rosenbrock. Evolution of the exploitation rate for different dimensions, for $\rho = 0.5$

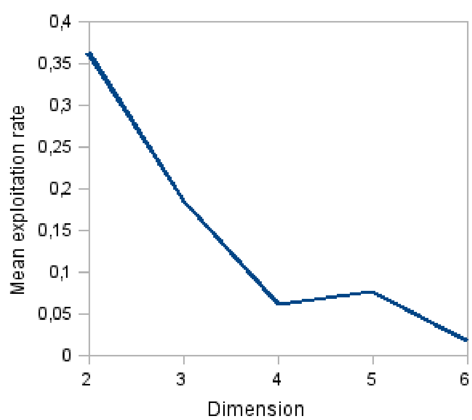

Figure 3: Rosenbrock. Mean exploitation rate for different dimensions, for $\rho = 0.5$

4 Manipulating the exploitation rate

4.1 Balanced PSO

So, it is tempting to slightly modify the algorithm, in order to ensure a better exploitation rate, and to define a "balanced" PSO, in which it is kept more or less constant, or, at least, significantly greater than zero for any dimension of the search space. The basic idea here is to "force" some particles to move to a given LEA. It can be seen as a cheap kind of local search. I tried a lot of combinations induced by the following definitions of the three "parameters":

- *some particles*
  - according to a given probability α.
- *move to*
  - to the middle of the LEA
  - at random (uniform distribution)
  - at random (Gaussian distribution). Can be seen as a partial Bare Bones PSO [2]
- a given LEA
  - the LEA of the particle
  - the best LEA (i.e. around the global best position)

More are obviously possible. A first interesting result is that the idea that a good balance always implies good results is simply wrong. However, and not surprisingly, it already appears that for each problem (function), there is a set of these parameters that leads to improved results, sometimes even excellent. Nevertheless, this is not that interesting, for there is no clear guidelines. So I was mainly looking for a compromise. In order to do that, I tried to automatically modify ρ during the process, after each iteration. The tested methods, combined with the three above parameters, are summarised below. Each time that a parameter $\rho_0$is used, several values were tested.

- Options for ρ
  - constant ρ_0
  - adaptive. After each iteration t, the exploitation rate $r(t)$ is calculated, and ρ is modified according to a formula (if $r(t)$ increases then ρ decreases, and vice versa). Several formulae were tested
  - alternative $\{1/S, \rho_0\}$
  - random, uniform distribution in $[0, \rho_0]$
  - random, Gaussian distribution $(1/S, \rho_0)$

Finally, the following set of parameters seems to be always better than SPSO07, although it is certainly possible to find even better ones, for they are here purely empirical:

- $\alpha = 0.5$, i.e. about half of the particles are forced to a LEA
- "move to" = at random (uniform distribution in the LEA)
- "a given LEA" = the best LEA
- alternative ρ value, chosen randomly in $\{1/S, \rho_0\}$, with the same probability 0.5 for each value, and with $\rho_0 = 0.5$. Note that in SPSO07 the swarm size S is automatically computed as a function of the dimension D by $S = 10 + 2\sqrt{D}$.

4.2 Comparisons

I do not give here all the results, just a small comparison between SPSO07 and the suggested Balanced PSO, with the above set of parameters. As we can see on the table 1, BPSO seems to be better than SPSO07 as long as the success rate is smaller than 100%. Even in that case, it is not necessarily worse, and anyway very slightly, with a mean number of evaluations just a bit higher. The three last functions are coming from the CEC 2005 benchmark suite [1].

5 More comments

With the suggested set of parameters, Balanced PSO actually performs a local search, inside in the best local exploitation area, by sampling at random about $S/2$ positions, according to an uniform distribution. Without that, and as soon as dimension is high, the exploitation rate would be zero. On the contrary, by doing that, it is kept more or less constant, about equal to 0.5. Such a fifty-fifty experimental result is intuitively satisfying. The bad news is that this method needs a additional parameter, but as soon as the dimension is greater than one, its value $\rho_0 = 0.5$ is also quite satisfying. This is the highest one for which the union of the local exploitation areas has the two following properties:

- no overlapping
- its projection on each dimension is a simple unique interval

There is probably a theoretical reason behind that. An interesting challenge!

References

[1] CEC. Congress on evolutionary computation benchmarks,http://www3.ntu.edu.sg/home/epnsugan/. 2005.

[2] J. Kennedy. Bare Bones Particle Swarms. In *IEEE Swarm Intelligence Symposium*, pages 80–87, 2003. Fully informed PSO.

[3] PSC. Particle Swarm Central, http://www.particleswarm.info.

Table 1: On classical test functions, keeping more or less constant the exploitation rate may improve the performance. Mean results are over 100 runs

	Search space	Acceptable error	Max. number of fitness evaluations	Standard PSO 2007	Balanced PSO
				Mean best value Success rate Mean number of fitness evaluations	
Sphere (Parabola)	$[-100, 100]^{30}$	0.0001	6000	1.46	1.01
Griewank	$[-100, 100]^{30}$	0.0001	9000	0.017 6%	0.017 6%
Rosenbrock	$[-10, 10]^{30}$	0.0001	40000	40.2	34.7
Rastrigin	$[-10, 10]^{30}$	0.0001	40000	60.34	55.9
Tripod	$[-100, 100]^{2}$	0.0001	10000	0.63 40%	0.78 43%
Ackley	$[-32, 32]^{30}$	0.0001	40000	1.19 26%	0.48 53%
CEC 2005 F1 (shifted Sphere)	$[-100, 100]^{30}$	0.000001	300000	0.0000009 100% 14150	0.0000009 100% 14502
CEC 2005 F6 (shifted Rosenbrock)	$[-100, 100]^{30}$	0.01	100000	0.0046 100% 4092	0.0053 100% 3924
CEC 2005 F9 (shifted Rastrigin)	$[-100, 100]^{30}$	0.01	300000	6.7	3.8