Stagnation Analysis in Particle Swarm Optimisation or What Happens When Nothing Happens

Maurice Clerc

Stagnation Analysis in Particle Swarm Optimisation

or

What Happens When Nothing Happens

Maurice Clerc
Draft 2005-06-16
Last update 2006-01-19

Introduction

The "classical" PSO version is very simple but the user have to define some parameters (swarm size, neighbourhoods, some coefficients). An adaptive version like TRIBES [1, 2] does not have this drawback. It is also often more effective when the criterion takes into account only the relation "best value vs number of fitness evaluations". However it is more time consuming, for it performs some intermediate computations in order to take advantage of the information collected during the process. This can be quite annoying, or even unacceptabe for some (quasi) real time applications.

That is why it is still interesting to just improve the classical version, without complicating it too much. The idea here is to mathematically analyse the behaviour of the particles when there is no improvement over several time steps. If the system is seen as a black box that displays only the best positions found so far by the particles, it happens nothing seen from outside, as there certainly happens something inside, which is worthly to study. Better parameters can be derived from such an analysis. Five of the suggested PSO variants are tested on five classical functions, just to bring out some that seems interesting.

Reminder of classical PSO

Principles

Particles are agents moving in a D-dimension search space. The set of this particles is the swarm, whose size is *S*. At each time step, each particle is described by four features:

- its position $x = (x_1,...,x_D)$
- its velocity $v = (v_1,...,v_D)$
- the best position found so far $p = (p_1,...,p_D)$
- some information links with some other particles (its "informant group", sometimes called its neighbourhood). These "who informs who" links, are often defined once at the beginning, but not topology has been proved to be globally the best one. So we use here a more robust method (see details in [1]). For each particle *K* links are randomly drawn at initialisation and then before the next iteration if the previous one didn't improve the best position found by the swarm.

Movement equations for a particle and for each dimension *d* are given by

$$\begin{cases} v_d \leftarrow c_1 v_d + c_{\max} alea(0,1)(p_d - x_d) + c_{\max} alea(0,1)(g_d - x_d) \\ x_d \leftarrow x_d + v_d \end{cases} \quad \textbf{(Equ. 1)}$$

where $g = (g_1,...,g_D)$ is the best position of the best informant.

PSO-0

We need a "reference" algorithm, called here PSO-0. It is the one described above, with the following parameters:

$$\begin{cases} S = 30 \\ K = 3 \\ c_1 = 1/(2\ln(2)) \cong 0{,}72 \\ c_{\max} = (c_1 + 1)^2 / 2 \cong 1{,}48 \end{cases}$$

The first three values has been chosen so that they can be kept in all variants described later. The coefficient $c_{\max}$ is derived from c_1 by using the constriction method explained in [3] (see Annexe for a quick explanation).

Stagnation phenomena

In classical PSO as defined by equation 1 each dimension is independent. So the analysis can be performed on just one dimension. In order to simplify notations, we write now *c* instead of $c_{\max}$, and we omit the *d* index. On the other hand, we will need to explicitely indicate the time step. At last, a given realisation of the random uniform variable on $[0, c]$ is now written $\tilde{c}^k$, where *k* is used to distinguish between realisations. Equation 1 is the rewritten as

$$\begin{cases} v(t+1) = c_1 v(t) + \tilde{c}^1 (p - x(t)) + \tilde{c}^2 (g - x(t)) \\ x(t+1) = x(t) + v(t+1) \end{cases} \quad \textit{(Equ. 2)}$$

We say there is stagnation if *p* and *g* (which are now just numerical values) don't change over some time steps. In order to qualitatively observe the behaviour of the particles (along just one dimension, as already noted), we choose some usual values for parameters and initialisations:

$$\begin{cases} c_1 = 0.7 \\ c = 1.4 \\ p = 0 \\ g = 0.5 \\ v(0) = alea(-0.5, 0.5) \\ x(0) = alea(0, 1) \end{cases} \quad \textit{(Equ. 3)}$$

If the particle is the (local) best

In order to study the behaviour ot the best particle, let's temporarily set $p = g$. On figure 1 we can see a typical evolution of the velocity: it tends towards zero, and the particle tends to its best known position, i.e. *g*. If the particle were intelligent, one could say it thinks "Well, I'm already the best, and the more I see the other don't make any improvement, the less I want to spend my energy to look for a better position". In other words, the particle quite rapidly gives up its search. In practice, it means there are some unuseful fitness evaluations, for anyway the particle (almost) doesn't move anymore.

However the behaviour is depending on c_1 and *c*. For example, with the values 0.9 and 2.1, we obtain the figure 2, which seems to be a divergent evolution. Of course such a curve can't be a proof, for the velocity may tends towards zero later, but, at least, it is a strong presumption. The lesson that can be drawn is there probably exists an intermediate parameter set so that the particle keeps some activity but not diverges.

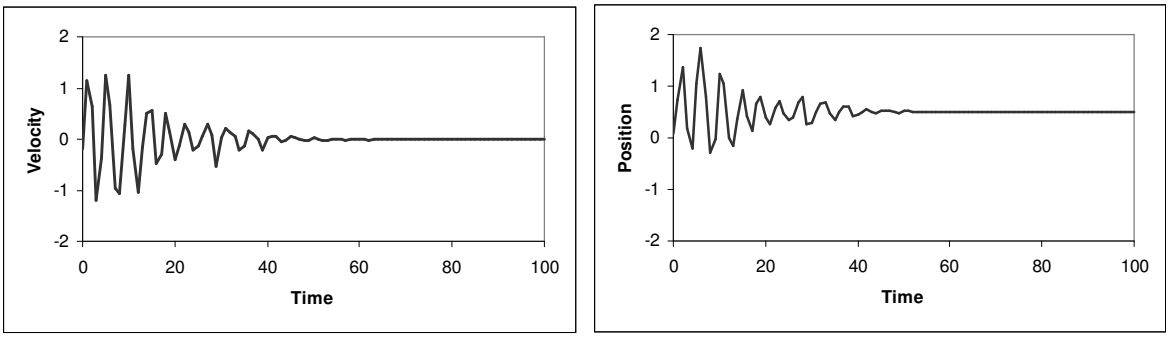

Figure 1. PSO-0. Behaviour of the (local) best particle in case of stagnation, with some usual parameter values

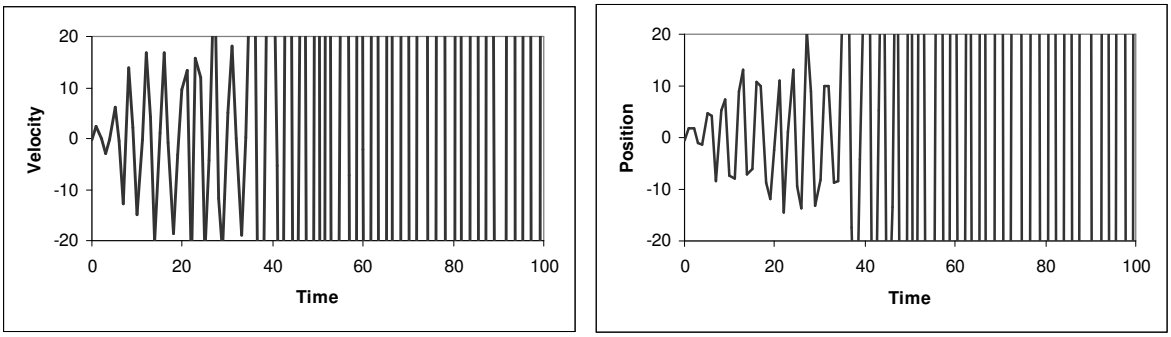

Figure 2. PSO-0. Behaviour of the (local) best particle in case of stagnation, with some unusual parameter values

If the particle is not the best one

Actually, the examination of what happens with a non best particle, i.e. with $p \neq g$, will suggests some possible solutions. The behaviour is indeed very different, even with usual parameter values. As we can see on figure 3 the velocity is constantly modified , without tends to zero and the position is permanently oscillating around a "mean" one.

A deeper examination shows that the module of the velocity follows a statistical law that is exponentially decreasing (cf. figure 4): small oscillations are frequent, more important ones are more rare, big ones are even more rare, etc. In other words, the longer the stagnation, the more probably big moves will occur. This is a quite desirable behaviour for then, even during stagnation, it performs a kind of local search that is usually fruitful: the particle will probably find a better position. So a quite obvious (small) improvement could be to do something so that the best particle has the same behaviour.

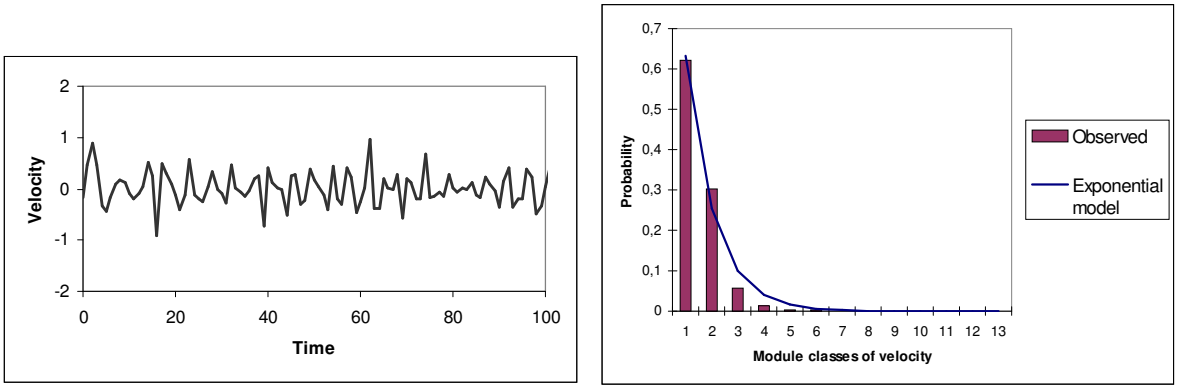

Figure 3. PSO-0. Behaviour of a non best particle in case of stagnation, with some usual parameter values. In case of long stagnation (over 100000 time steps here) a non best particles oscillates according to an exponential decreasing probability distribution: small moves are quite frequent, big ones are rare but do exist

Stagnation analysis

.

General iterative equation and interpretation

In the particular case $c = 0$, we obviously have $v(t+1) = c_1 v(t)$, and the convergence criterion is simply $c_1 < 1$ (we assume it is non negative). In the general case, from equation 2 we can successively write

$$v(t+2) = c_1 v(t) + \tilde{c}^3 (p - x(t+1)) + \tilde{c}^4 (g - x(t+1))$$

$$-x(t+1) = -v(t+1) + x(t)$$

$$-x(t) = \frac{1}{\tilde{c}^1 + \tilde{c}^2} \left(v(t+1) - c_1 v(t) - \tilde{c}^1 p - \tilde{c}^2 g \right)$$

$$v(t+2) = (c_1 - \tilde{s} + \tilde{q}) v(t+1) - c_1 \tilde{q} v(t) - (p - g) \tilde{w} \qquad \textit{(Equ. 4)}$$

with

$$\begin{cases} \tilde{s} = \tilde{c}^3 + \tilde{c}^4 \\ \tilde{q} = \dfrac{\tilde{c}^3 + \tilde{c}^4}{\tilde{c}^1 + \tilde{c}^2} \\ \tilde{w} = \tilde{c}^3 - \tilde{c}^1 \tilde{q} \end{cases} \qquad \textit{(Equ. 5)}$$

Quantities $\tilde{s}$, $\tilde{q}$, et $\tilde{w}$ are realisations of the random variables *S*, *Q* et *W*. Let' s also define $Z = c_1 - S + Q$. The general iterative equation is then based on the three random variables *Z*, *Q* et *W*. It can be rewritten in a summarized form, and by shifting the iteration ranks:

$$v(t+1) = Z v(t) - c_1 Q v(t-1) + (p - g) W \qquad \textit{(Equ. 6)}$$

Probability densities of *S* and *Q* can quite easily be formulated (cf. Annexe). The ones of *Z* and *W* are extremely complicated, but anyway we can have an idea of their "shape" (by simulations), and their main features (minimum, maximum, mean) can be computed (cf. Table 1. Details and figures are in Annexe). Note that *Z* and *Q* have an infinite support, with a density tending towards zero for high values, and that explains the exponential decline that we have seen in figure 3.

.

	Minimun	Maximum	Mean
Z	$c_1 + 1 - 2c$	∞	$c_1 - c + 2\ln(2)$
Q	0	∞	$2\ln(2)$
W	$-c$	c	0 (std. dev. $c\sqrt{(11 - 14\ln(2))/12}$)

Table 1. Main features of the three random variables of the general iterative equation

Let us try now to give the "meaning" of the three terms. The last one (*W*) can be seen as a "noise". The bigger $p - g$ the more important the noise. This is intuitively quite sastifying: the more the particle is far from the best position it knows, the more it tends to try big moves. Note that there is a finite maximum possible value (and a finite minimum one). It might be better to replace this noise by a classical Gaussian one.

As $-c_1 Q$ is always negative and as Q is parameter free, this term can be seen as a "back force", whose intensity is only depending on c_1.

At last Z, the first one, is usually sometimes negative (back force) sometimes positive (forth force). However we can note that the back force may exist only if $c_1 + 1 - 2c$ is negative. Again this suggests a possible relationship, that is to say

$$c = \frac{c_1 + 1}{2} \qquad \textit{(Equ. 7)}$$

This would clarify the role of each random variable: noise for W, back force for Q, and only forth force for Z. However, in order to find some other constraints on parameters we have to go further in our analysis.

If the particle is the best one

For the best particle we have $p = g$ and we clearly see from equation 6 that there is then no noise component. In passing, it explains why the particle may converge for some parameter values. If we consider two consecutive time steps (don't forget we are in stagnation, so p is constant), the equation can be rewritten in a matrix form

$$\begin{bmatrix} v(t+1) \\ v(t) \end{bmatrix} = \begin{bmatrix} \tilde{z}^t & -c_1 \tilde{q}^t \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ v(t-1) \end{bmatrix} = \tilde{M}^t \begin{bmatrix} v(t) \\ v(t-1) \end{bmatrix} \qquad \textit{(Equ. 8)}$$

Then the behaviour is defined by $\left\| \prod_{k=0}^{t} \tilde{M}^k \right\|$. To study it is quite difficult (see Annexe), but the final result is a relation between c_1 and c that we can see on figure 4. Any pair of parameters in the convergence domain is usable. However, in iterative optimisation, it is often a good thing to work "on the edge of chaos". So points on the frontier curve should be better. Moreover, in order to enlarge the local search, c_1 and c should be "both as big as possible". There are several way to formulate this, for example by maximizing the quantity $\sqrt{c_1 c}$. Finally, the most promising parameters are on the curve and "around" the point

$$\begin{cases} c_1 = \dfrac{1}{2\ln(2)} \cong 0.72 \\ c = 1 + 2\ln(2) \cong 2.39 \end{cases} \qquad \textit{(Equ. 9)}$$

Remember it is only for stagnation phases. If we want to use just one set of parameters, no matter there is stagnation or not, it may be a good idea to choose say c_1 according to the equation 9, and c according to equation 7, for example (see variant 3PD-PSO-0 below).

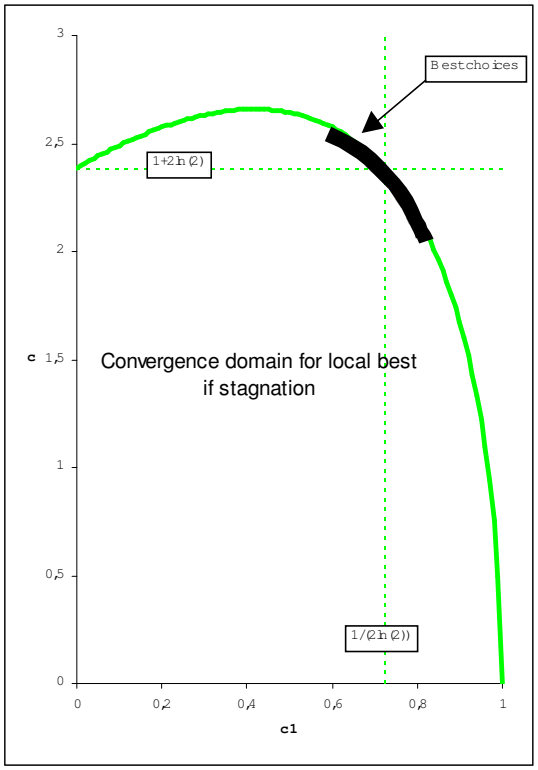

Figure 4. In case of stagnation best parameters are on the frontier of the convergence domain and, on the other hand, both "as big as possible"

If the particle is not the best one

Let' s remind that our hypothesis is stagnation: neither p nor g are modified. Then, when the particle is not the best one, that is to say when $p \neq g$, it can' t converge, because of the "noise" component W. It is indeed what we observe, no matter how small c_1 is. Actually, even for $c_1 = 0$ the particle is oscillating. However it can diverge it this coefficient is too big.

More precisely, for the mean of W is zero, the conditions that we have found for the best particle are still valid, except that they don' t imply convergence but non divergent oscillations. This is quite a good thing for this behaviour performs a kind of local search..

Generalisation and variants

The three distributions

We can now rewrite equation 6 in a more general way

$$v(t+1) = R_1 v(t) - R_2 v(t-1) + (p-g) R_3 \qquad \textit{(Equ. 10)}$$

where the three probability distributions have the following features

R_1 is positive unimodal on a finite support

R_2 is positive unimodal on a finite support

R_3 is symmetrical and unimodal with a mean equal to zero

We have seen that we can choose $R_3 = W$, $R_2 = c_1 Q$, and $R_1 = Z$, with the condition $c \leq (c_1 + 1)/2$, but some others are clearly possible. In particular, as R_1 is for the "forth force" and $-R_2$ for the "back force", we can manipulate c and c_1 so that they are quite similar. We will see a simple example below. Starting from PSO-0,

we will now progressively add just some small modifications. Don' t forget that although we are looking for improvements we also want to keep the algorithm as fast as possible.

There will be two kinds of modification that takes advantage of our stagnation analysis:

- the ones that just define some new parameter sets, that are used no matter there is no stagnation or not (see below variants 0, 0' and 1)
- the ones that add a rule like "if there is stagnation, then ..." (see below variants 2 and 3)

Theoretically, stagnation occurs as soon as *p* and *g*don' t change for two time steps. However, as we have seen, it is not a good idea to do something special immediately, at least as long as the best particle is moving "enough". So we have first to define how long should be the stagnation before to modify the moving strategy.

Defining a stagnation threshold

The underlying idea is to say there is indeed time do try something different if:

- the local best particle has been enough time to be informed (either directly or undirectly) by a possible better particle
- it has nevertheless not improved its best known position

Note that this criterion is not pertinent for the global best, but in order to keep the algorithm simple, we neglect this particular case.

The probability that a given particle *A* is still not informed after *t* time steps that there exists a better particle *G* is given by

$$p_t(G\neg\rightarrow A)=\left(1-\frac{1}{S}\right)^{Kt} \qquad \textit{(Equ. 11)}$$

where *S* is the swarm size and *K* the number of information links drawn at random (at each time step, for there is no improvement. For detailed formulas of "temporal connexity", see [1]) . If we want that this probability is smaller than a threshold ε the number of time steps should be at least equal to

$$t_{stag}=\frac{\ln(\varepsilon)}{K\ln\left(1-\frac{1}{S}\right)} \qquad \textit{(Equ. 12)}$$

For example for $S=30$, $K=3$ and $\varepsilon=0.0001$, we would now say there is stagnation if the local best particle didn' t improve its position from at least 91 time steps. This is the value used in some variants below. The additional parameter ε is unfortunately empirical: this is the price to pay to be sure the algorithme remains fast.

Test bed

We have now in hand all what we need to test some variants on a some functions, in order to compare them. The five functions used here are very classical, and the minimum is always zero. However, the dimension (30), the search spaces and the required accuracies are chosen so that our "reference" algorithm PSO-0 has a success rate never equal to zero nor to 100% (over 100 runs with at most 40000 fitness evaluations). This success rate can therefore be chosen as a comparison criterion, for each function. For a quick comparison the mean success rate is even enough. In case of a tile (or almost), we will prefer the variant that gives the smaller standard deviation, for it is then probably more robust.

Function	Search space	Accuracy	Success rate with PSO-0
Parabola (Sphere)	$[-20,20]^{30}$	10^{-9}	32%
Griewank	$[-300,300]^{30}$	0.0001	44%
Rosenbrock	$[-10,10]^{30}$	25	21%
Rastrigin	$[-5{,}12,5{,}12]^{30}$	35	12%

Ackley	$[-32,32]^{30}$	0.0002	5%

Table 2. Test bed and success rate with PSO-0 (100 runs for each function, and at most 40000 fitness evaluations for each run)

Six variants

3PD-PSO is for "Three Probability Distributions Particle Swarm Optimisation". The following variants are just a sampling of the most obvious ones suggested by stagnation analysis. In particular we don' t modify at all distribution R_2, which is still equal to $c_1 Q$, and R_1 is also equal to *Z*, just possibly translated. Table 3 summarizes the results.

Function	3PD-PSO-0	3PD-PSO-0'	3PD-PSO-1	3PD-PSO-2	3PD-PSO-2'	3PD-PSO-3
Parabola (Sphere)	100%	100%	100%	100%	100%	100%
Griewank	35%	49%	52%	48%	57%	53%
Rosenbrock	74%	74%	65%	77%	72%	72%
Rastrigin	58%	35%	24%	34%	39%	33%
Ackley	5%	66%	86%	70%	62%	64%
Mean	*54%*	*65%*	*65%*	*66%*	*66%*	*64%*
Std. dev.	0.33	0.22	0.27	0.23	0.20	0.22

Table 3. Results with some variants. When the success rate is equal to 100%, the mean number of fitness evaluations is about 18000

3PD-PSO-0

This variant is simply PSO-0 with some different coefficients. That is to say stagnation is not checked. So we need the find a "compromise", some parameters that are not bad if there is stagnation, and not bad if there is no stagnation. As already suggested, according to the above analysis, a promising choice is

$$\begin{cases} c_1 = \dfrac{1}{2\ln(2)} \cong 0.72 \\ c = \dfrac{c_1 + 1}{2} \cong 0.86 \end{cases}$$

so that the random variable *Z* (the "forth force") never gives negative values. The movement equation is then directly derived from equation 1 and can be written for dimension *d*

$$v_d \leftarrow 0.72 v_d + 0.86 alea(0,1)(p_d - x_d) + 0.86 alea(0,1)(g_d - x_d) \qquad \textbf{(Equ. 13)}$$

Other parameters are the same (30 particles, 3 informations drawn at random for each particles). Effectiveness is significantly improved.

3PD-PSO-0'

Here we keep $c_1 = 1/2\ln(2) \cong 0{,}72$, but we choose *c* so that the mean of *Z* (for the "forth force") is equal to the one of $c_1 Q$ (for the "back force"). It gives $c = 1/2\ln(2) + 2\ln(2) - 1 \cong 1.108$. The velocity is then updated by

$$v_d \leftarrow 0.72 v_d + 1.108 alea(0,1)(p_d - x_d) + 01.108 alea(0,1)(g_d - x_d) \qquad \textbf{(Equ. 14)}$$

Effectiveness is more improved than with 3PD-PSO-0, and the standard deviation is also reduced.

3PD-PSO-1

For $c_1 = 1/2\ln(2) \cong 0{,}72$, stagnation analysis suggested a possible *c* (i.e. $c_{\max}$ in equation 1) equal to 0.86. The idea is here to define another plausible value, and to choose *c* at random between them for each move. For example value 1.48 used in PSO-0 is not arbitrary: it comes from a deterministic convergence analysis (cf. Annexe). We can then try the variation interval

$$[a,c]=\left[\frac{c_1+1}{2},\frac{(c_1+1)^2}{2}\right] \qquad \textbf{(Equ. 15)}$$

In practice, to compute the new velocity of a particle, $c_{\max}$ is first chosen at random

$$c_{\max}=alea(a,c)$$

and then for each dimension *d* equation 1 is applied. Formally, it is equivalent to apply equation 21

$$v_d \leftarrow c_1 v_d+\tilde{b}^1(p_d-x_d)+\tilde{b}^2(g_d-x_d) \qquad \textbf{(Equ. 16)}$$

where $\tilde{b}^1$ et $\tilde{b}^2$ are two realisations of the random variable *B*, defined by

$$B=(a+(c-a)U_1)U_2 \qquad \textbf{(Equ. 17)}$$

U_1 and U_2 being two uniform random variables on $[0,1]$. This distribution is obviously not uniform anymore, as we can see on figure 10. Intuitively it is more satisfying, for big moves are now less probable. The improvement is equivalent to the one given by 3PD-PSO-0 , although the standard deviation is slightly higher.

3PD-PSO-2

In this variant, we define two sets of parameters, one for "normal" case, and one for stagnation. In the first case, we use the parameter set of 3PD-PSO-0' , and in the second case, we choose the "central point" of the promising area that we have seen on figure 4:

$$\begin{cases} c_1=\dfrac{1}{2\ln(2)}\cong 0.72 \\ c=\dfrac{1}{2\ln(2)}+2\ln(2)-1\cong 1.108 \end{cases} \quad \text{and} \quad \begin{cases} c_1=\dfrac{1}{2\ln(2)}\cong 0.72 \\ c=1+\ln(2)\cong 2.39 \end{cases}$$

This variant is again slightly better both for the mean and the standard deviation.

3PD-PSO-2'

We have seen that in case of stagnation the local best particle has a special behaviour that rapidly induces some unuseful fitness evaluations. Using two set of coefficients is already a way to change this, but can also modify the movement equation so that the "noise" term is not null.

Let *G* be the local best particle. In order to compute its new velocity, we can consider all best known positions in the swarm (in short, all "*p*"), and to keep the ones that are better that the one of *G* (i.e. *g*). Then we choose one at random, say g_2, and in the movement equation we replace *g* by g_2. So, except for the global best, the movement equation is now similar to the one of a non best particle.

By adding this modification to 3PD-PSO-2, the effectiveness it more or less the same and the standard deviation is a bit smaller.

3PD-PSO-3

Finally, we have seen that the "noise" *W* is vaguely similar to a truncated Gaussian curve. Therefore, in case of stagnation, it is tempting to replace it by a true Gaussian distribution. In order to not have to define a new parameter, we can choose this Gaussian distribution with the same standard deviation that *W*, i.e. $\sigma=c\sqrt{(11-14\ln(2))/12}$ (almost $c/3$).

The algorithm is then a bit more complicated. We need to memorize velocities at time *t* and *t*-1, and also the random values that are used. Then, in case of stagnation velocity is directly computed by equation 6. In practice, it gives

$$v_d(t+1)=(c_1-\tilde{s}+\tilde{q})v_d(t)-c_1\tilde{q}v(t-1)-(p-g)alea\_normal(0,\sigma) \qquad \textbf{(Equ. 18)}$$

The process is not really slowed down. However applying this method to 3PD-PSO-0' doesn' t improve it (the result is the same if applied to 3PD-PSO-2).

Conclusion

There is a slight advantage to variant 3PD-PSO-2' (best mean and seems a bit more robust). Let' s summarize the differences compared to PSO-0:

- two sets of coefficients, one when everything goes well, and one where there is stagnation
- for the local best the set of informants is the whole swarm, and if possible a better than itself it chosen (at random in case of a tile)

Other variants based on stagnation analysis are of course possible. In particular we could more completely manipulate the "forth force" and "back force" components, for example by using Beta distributions. And more tests are needed to choose the best variant.

Annexe

Mathematical developments below have been simplified. In particular the details of some tedious probability densities are not given, for they are not really more helpul than their graphical representations to understand this study.

Relationship between coefficients according to deterministic analysis

In [3] several constriction methods are defined, that guarantee the non-divergence of the swarm. For historical reasons, the most commonly used gives the two coefficient c_1 and $c_{\max}$ as depending on an unique parameter φ:

$$\begin{cases} c_1 = \dfrac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}} \\ c_{\max} = c_1 \dfrac{\varphi}{2} \end{cases}$$

However this can be easily rewritten as a relationship between them

$$c_{\max} = \frac{(c_1 + 1)^2}{2}$$

S study

Let X_1 and X_2 be two uniform random variables on $[0, c]$. The density of $S = X_1 + X_2$ is given by

$$s(u) = \int_{-\infty}^{+\infty} g(x, u - x)\,dx$$

with $g(x, u - x) = (1/c)(1/c)$ if x and $u - x$ are both in $[0, c]$, and 0 else. One can distinguish two cases (we do suppose here c is not null):

$$\begin{cases} 0 < u < c \Rightarrow s(u) = s_1(u) = \dfrac{1}{c^2} \displaystyle\int_0^u dx = \frac{u}{c^2} \\ c \le u \le 2c \Rightarrow s(u) = s_2(u) = \dfrac{1}{c^2} \displaystyle\int_{u-c}^{c} dx = \frac{2c - u}{c^2} \end{cases}$$

S support is $[0, 2c]$ and the mean is the sum of the means of X_1 and X_2, i.e. $\hat{S} = c$. Actually S can be rewritten $S = cU$ where U is a uniform random variable on $[0,1]$.

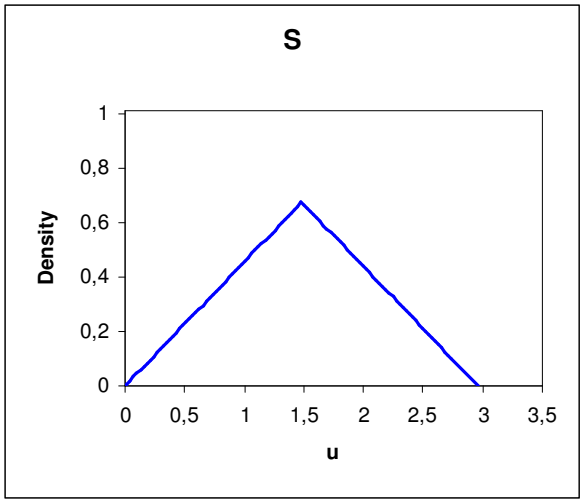

Figure 5. S density for c=1.48

Q study

Let S_1 and S_2 be two random variables like *S*. Their quotient is $Q = S_1/S_2$. The corresponding probability density is given by

$$q(u) = \int_{-\infty}^{+\infty} |x| s(ux) s(x) dx$$

To more easily compute this definite integral one can consider four cases.

$0 < u \le 1/2$

$$\begin{cases} q(u) = \int_0^c x s_1(ux) s_1(x) dx + \int_c^{2c} x s_1(ux) s_2(x) dx \\ \quad = \frac{7}{6} u \end{cases}$$

$1/2 < u \le 1$

$$\begin{cases} q(u) = \int_0^c x s_1(ux) s_1(x) dx + \int_c^{c/u} x s_1(ux) s_2(x) dx + \int_{c/u}^{2c} x s_2(ux) s_2(x) dx \\ \quad = \frac{1}{6u^3} - \frac{2}{3u^2} + \frac{8}{3} - \frac{3}{2} u \end{cases}$$

$1 < u \le 2$

$$\begin{cases} q(u) = \int_0^{c/u} x s_1(ux) s_1(x) dx + \int_{c/u}^{c} x s_2(ux) s_1(x) dx + \int_c^{2c/u} x s_2(ux) s_2(x) dx \\ \quad = -\frac{3}{2u^3} + \frac{9}{3u^2} - \frac{2}{3} - \frac{u}{6} \end{cases}$$

$2 < u$

$$\begin{cases} q(u) = \int_0^{c/u} x s_1(ux) s_1(x) dx + \int_{c/u}^{2c/u} x s_2(ux) s_1(x) dx \\ \quad = \dfrac{7}{6u^3} \end{cases}$$

Note that Q is parameter free. The mean can be computed by noting that $Q = S_1(1/S_2)$, as a product of two independent random variables. This mean is then the one of $1/(U_1 + U_2)$. The density of this last random variable is

$$\begin{cases} \dfrac{2}{u^2} - \dfrac{1}{u^3} \text{ if } u \in [1/2,1] \\ \dfrac{1}{u^3} \text{ if } u > 1 \end{cases}$$

Its mean is then

$$\int_{1/2}^{1} \left(\frac{2}{u} - \frac{1}{u^2} \right) du + \int_1^{+\infty} \frac{1}{u^2} du = 2\ln(2)$$

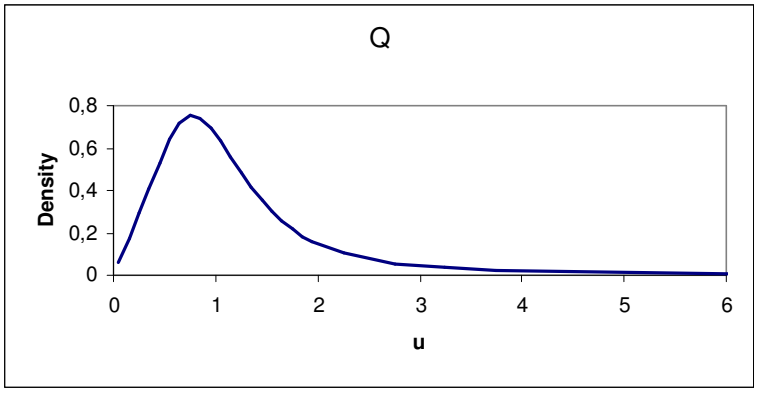

Figure 6. Q density

W study

The random variable *W* can be rewritten

$$\begin{cases} W = X_3 - X_1 \dfrac{X_3 + X_4}{X_1 + X_2} \\ \quad = \dfrac{X_3 X_2 - X_1 X_4}{X_1 + X_2} \end{cases}$$

So we immediately see that it is symmetrical and that its mean is zero. In order to compute the maximum value, one can note that it is reached when X_1 is null and that in this case *W* is equal to X_3. This maximum is then equal to *c*. The formulas that give the density are quite complicated, for they make use of the order 2 Jonquières function (dilogarithm) $\mathrm{Li}_2(y) = \sum_{k=1}^{\infty} \frac{y^k}{k^2}$. It is enough here the visualize it and to compute the standard deviation

σ . By using the fundamental theorem about the variance of a combination of random variables, and by noting that each X_i is equal to cU_i, where density of U_i is 1 on $[0,1]$, we can write

$$\sigma^2 = \int_{-\infty}^{+\infty} t.density(W^2)dt =$$

$$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} W^2 density(X_1X_2X_3X_4)dx_1dx_2dx_3dx_4$$

$$= c^2 \int_0^1\int_0^1\int_0^1\int_0^1 \left(\frac{u_3u_2 - u_1u_4}{u_1+u_2}\right)^2 du_1du_2du_3du_4$$

$$= c^2 \frac{11-14\ln(2)}{12}$$

that is to say $\sigma = c\sqrt{(11-14\ln(2))/12}$, almost equal to $c/3$. Thanks to this information, we can define the best Gaussian approximation of *W*.

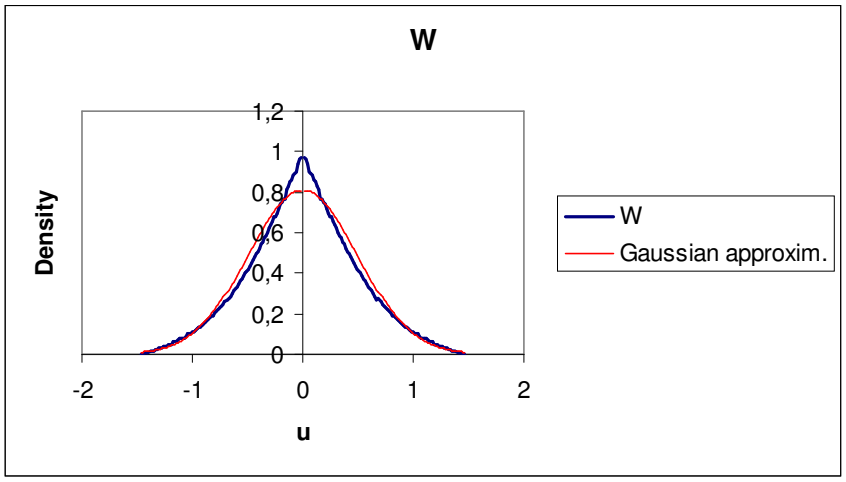

Figure 7. W density for c=1.48, and best Gaussian approximation

Z study

The random variable $Z = c_1 - S + Q$ can be rewritten

$$Z = c_1 - (U_3 + U_4)\left(c - \frac{1}{U_1+U_2}\right)$$

So we can see that the support is $[c_1 - c + 1, +\infty[$. We have seen that the mean of $1/(U_1+U_2)$ is $2\ln(2)$. Therefore the mean of Z is $\hat{Z} = c_1 - c + 2\ln(2)$. Note that $\hat{Z} = c_1 - \hat{S} + \hat{Q}$, but this formula couldn' t have been directly written, for random variables S and Q are not independent. Here again, we just visualize the density curve, without giving the too complicated formulas.

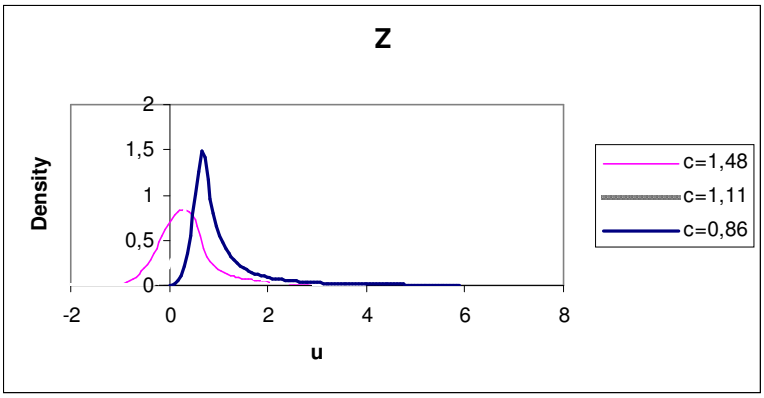

Figure 8. Some Z densities for c_1=0.72

B study

The random variable *B* can be rewritten $B=(a+(c-a)U_1)U_2$. Its support is obviously $[a,c]$ It can be seen as the product of the two independent random variables $X_1=(a+(c-a)U_1)$ and U_2. Its mean is the product of ones of X_1 and U_2, respectively $(a+c)/2$ and $1/2$. The mean value is then $\hat{B}=(a+c)/4$.

The probability density is given by

$$\begin{cases} b(u)=\dfrac{1}{c-a}\displaystyle\int_a^c \frac{1}{x}dx=\frac{1}{c-a}\ln\left(\frac{c}{a}\right) \text{if } u\in[0,a] \\ \quad =\dfrac{1}{c-a}\displaystyle\int_c^u \frac{1}{x}dx=\frac{1}{c-a}\ln\left(\frac{c}{u}\right) \text{if } u\in[a,c] \end{cases}$$

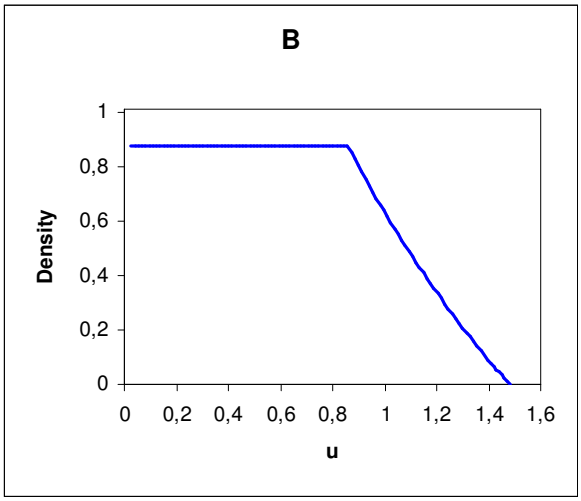

Figure 9. B density for c=1.48

Convergence of an iterative random sequence

What follows is just a sketch of the proof.

A and *B* are two random variables. $\tilde{a}_t$ and $\tilde{b}_t$ are two realisations of *A* and *B* at time *t*. We consider the *y* function iteratively defined by $y(0)$, $y(1)$, and

$$y(t+1) = \tilde{a}_t\, y(t) + \tilde{b}_t\, y(t-1)$$

This can be rewritten

$$\begin{bmatrix} y(t+1) \\ y(t) \end{bmatrix} = \begin{bmatrix} \tilde{a}_t & \tilde{b}_t \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y(t) \\ y(t-1) \end{bmatrix}$$

We want to study the convergence of *y* to zero. Let' s define $\tilde{m}_t = \begin{bmatrix} \tilde{a}_t & \tilde{b}_t \\ 1 & 0 \end{bmatrix}$, a realisation of the random matrix $M = \begin{bmatrix} A & B \\ 1 & 0 \end{bmatrix}$ at time *t*. As the quadratic convergence implies the convergence in probability, we can study the norm of the product

$$\left\| \prod_0^t \tilde{m}_k \right\|^2 = \prod_0^t \|\tilde{m}_k\|^2$$

We can suppose all $\tilde{m}_k$ are non null (if one is null, then *y* is obviously convergent). Then we can study

$$\tilde{h}_t = 2\sum_0^t \ln(\|\tilde{m}_k\|)$$

This is a realisation of a random variable H_t. The key point is that if its mean $\hat{H}_t$ tends to $-\infty$, then *y* tends to zero. By applying the central limit theorem to $\tilde{h}_t / t$, we find $\hat{H}_t \approx t\hat{G}$, where $\hat{G}$ is the mean of the random variable $\mathrm{G} = \ln(\|M\|)$. So the condition is simply $\hat{G} < 0$. We have

$$\hat{G} = \int_{G\,\text{support}} \ln(\|M\|)\, dA\, dB$$

In the particular case of stagnation analysis, we have

$$\begin{cases} A = Z = c_1 - X_3 - X_4 + \dfrac{X_1 + X_2}{X_3 + X_4} \\ B = -c_1 \dfrac{X_1 + X_2}{X_3 + X_4} \end{cases}$$

so the condition is

$$\int_0^c \int_0^c \int_0^c \int_0^c \ln(\|M\|)\, dx_1 dx_2 dx_3 dx_4 < 0$$

Note that there is trick here, for $\tilde{m}_k$ and $\tilde{m}_{k+1}$ are not independent. However $\tilde{m}_k$ and $\tilde{m}_{k+2}$ do are, so the condition is still valid. Unfortunately the result is quite complicated. A good quadratic approximation can be found by saying that the frontier of this convergence domain contains the three points $(0, 1+2\ln(2))$, $(1/2\ln(2), 1+2\ln(2))$, and $(1, 0)$ (see figure 4), and by empirically estimating two parameters α and β :

$$c < \frac{\beta c_1 + \delta}{2} + \sqrt{\alpha c_1^2 + \gamma c_1 + \eta + \frac{(\beta c_1 + \delta)^2}{4}}$$

with

$$\begin{cases} \alpha = -36\ln(2) \\ \beta = 15\ln(2) \\ \gamma = -\dfrac{\alpha}{2\ln(2)} - \beta\left(1 + 2\ln(2)\right) \\ \delta = \dfrac{\alpha + \gamma}{1 + 2\ln(2)} + 1 + 2 * \ln(2) \\ \eta = -\alpha - \gamma \end{cases}$$

Now, it is easy to prove (by recurrence) that $\det\left(\prod_{0}^{t} \tilde{m}_k\right) = c_1^{t-1}\tilde{q}$. So, as soon as we have $c_1 < \overline{q}^{\frac{1}{t-1}} = \dfrac{1}{(2\ln(2))^{\frac{1}{t-1}}}$ the system begins to converge. That is why the value $\chi_1 = \dfrac{1}{2\ln(2)}$ is special. When c_1 is smaller than χ_1, convergence begins immediately. If it it bigger than χ_1 (but still smaller than 1), convergence may begin only after $\dfrac{\ln(\chi_1)}{\ln(c_1)}$ time steps.

Stagnation in FIPS (Fully Informed Particle Swarm)

When more than just two particles inform a given one, the motion equations can be written

$$\begin{cases} v(t+1) = c_0 v(t) + \sum_{k=1}^{K} \tilde{c}_k^1 \left(p_k - x(t)\right) \\ x(t+1) = x(t) + v(t+1) \end{cases}$$

where $\tilde{c}_k^1$ is a realisation of a random uniform variable on $[0, c]$.

If no p_k is modified , we can deduce

$$v(t+2) = \tilde{z}\, v(t+1) - c_0 \tilde{q}\, v(t) + \tilde{\omega}$$

with

$$\begin{cases} \tilde{q} = \dfrac{\sum_{k=1}^{K} \tilde{c}_k^2}{\sum_{k=1}^{K} \tilde{c}_k^1} \\ \tilde{z} = c_0 - \sum_{k=1}^{K} \tilde{c}_k^2 + \tilde{q} \\ \tilde{\omega} = \sum_{k=1}^{K} \tilde{\alpha}_k p_k \end{cases}$$

where

$$\tilde{\alpha}_k = \tilde{c}_k^2 - \frac{\sum_{j=1}^{K} \tilde{c}_j^2}{\sum_{j=1}^{K} \tilde{c}_j^1} \tilde{c}_k^1$$

As $\tilde{c}_k^1$, $\tilde{c}_k^2$ is a realisation of a random uniform variable on $[0, c]$.

When $K = 2$, we retrieve equation 4, with $p = p_1$, and $g = p_2$, and for $K > 2$ the formulas are on the whole quite similar. The formula for $\tilde{\omega}$ is of course more complicated, but we can note that if all p_k are equal then we still have $\tilde{\omega} = 0$. In particular it would be true for the global best: the "noise" component would again disappear. However, in FIPS each particle is not informed by itself, and we see here why the probability there is no noise component is then almost null.

Acknowledgements

Jean-Pierre Levrel (newsgroup fr.sci.maths)

Ray Koopman (newsgroup alt.sci.math.probability)

References

[1] M. Clerc, *L'optimisation par essaims particulaires. Versions paramétriques et adaptatives*: Hermès Science, 2005.

[2] G. C. Onwubolu, "TRIBES application to the flow shop scheduling problem," in *New Optimization Techniques in Engineering*. Heidelberg, Germany: Springer, 2004, pp. 517-536.

[3] M. Clerc and J. Kennedy, "The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 58-73, 2002.