Particle Swarm Optimization & Parameter Selection

Christoph Strößner

October 17, 2014

Contents

1 Introduction

All information for this work is taken from M. Jiang, Y. P. Luo and S. Y. Yang's paper "Particle Swarm Optimization - Stochastic Trajectory Analysis and Parameter Selection".
Particle Swarm Optimization(PSO) is a heuristic for finding the location of global extrema in black-box problems. We start with a description of an example for PSO by comparing the algorithm to the search of a mountain top. Then we introduce the formal definition of the PSO algorithm focusing on initialization and particle movement. After these two introductory parts we start the theoretical analysis of the parameters. Firstly we analyse several simpler versions of PSO slowly increasing complexity and finally we concentrate on the algorithm which we described in the beginning using expectation and variance. Afterwards we use the results to figure out good parameter values to find the

extrema. In the end we compare several sets of parameters using benchmark-functions and Matlab for visualization.

2 Allegory & Definition

First of all we want to describe PSO informally with an allegory. The PSO algorithm can be compared to a group of nerds randomly spread in the mountains. They are supposed to get close to the highest point in a limited area. As nerds usually are not used to daylight, their only way to navigate is via the GPS in their mobile phones. Furthermore they are allowed to use facebook to share their position with their friends. Now they walk randomly around, always a bit towards their personal previous best position and a bit towards the best position on facebook. After some time, if you have access to all facebook profiles like the NSA, you can figure out where the highest mountain might be.
The PSO algorithm uses this "concept" for solving black-box problems from $\mathbb{R}^n \to \mathbb{R}$. At first it initializes several particles. Experiments revealed some "good" starting conditions. So the number of particles should be around 20-50 or $10 + 2\sqrt{2 * d}$ where $d =$ dimension of the problem. The particles themselves are uniformly spread at random in the search space. Further adding a starting velocity is beneficial. This can be achieved by using uniformly random vectors from the search space as velocity or calculating it via half of the difference to an other point in the search space. Concerning the neighborhood which equals facebook in the allegory, any graph can be used as representation for different groups of particles. For our analysis we stick to fully connected / complete graphs.
After initialization the PSO algorithm moves the particles in discrete time steps t. It takes each particle i and updates it's velocity V_i and it's position P_i. Further the personal attractor P_i for each particle and the global attractor P_g are necessary to lead the particle towards the positions it should exploit more. This is updated in the end of each time step. The whole algorithm looks as follows:

```
for each particle i(i=1,...,m) do {particle initial.}
    initialize position and velocity
end for
initialize neighborhood structure
while (termination criteria not met)
    for each particle do {particle movement}
        velocity update
        position update
    end for
for each particle do {best update}
    if change then update
end for
end while
```

For the further analysis we concentrate on the equation describing the velocity and position update. It is defined for each dimension $d \in \{1, ..., n\}$ individually:

$$\begin{aligned} V_i^d(t+1) =& \omega \cdot V_i^d(t) + c_1 r_{1,i}^d(t) \cdot (P_i^d(t) - X_i^d(t)) + c_2 r_{2,i}^d(t) \cdot (P_g^d(t) - X_i^d(t)) \\ X_i^d(t+1) =& a \cdot X_i^d(t) + b \cdot V_i^d(t+1) \end{aligned}$$

where (c_1, c_2, ω, a, b) is the parameter tuple on which we focus our analysis. c_1 describes how much a particle trusts its personal attractor. c_2 is analogous for the global attractor. ω is the inertia weight. Both remaining parameters a and b are not necessary and should be set to 1. In addition $r_{1,i}$ and $r_{2,i}$ are independent uniform random numbers in $[0, 1]$. They are necessary for the algorithm to be not exactly predictable. Mainly the advantage is that expectation and variance allow the PSO to explore and exploit better. We will start our theoretical analysis with a simpler model. When analysising the swarm we usually assume it is stagnating which means no updates to personal or global attractor occur. In this situation the particles should converge to an area between both attractors and exploit there.

3 Simpler Versions

To begin the analysis of PSO we start with an extremely basic model slowly adding more and more complexity. First of all we only use a single one-dimensional particle. Therefore we also have only one global attractor $P_g = P_i$. Furthermore we remove the inertia weight $\omega = 1$ and set $a = b = 1$. The latter variables will never be necessary as they do not contribute anything. This will be shown in the stochastic analysis. Right now we also remove the stochastic components and random variables. As this is a big difference to the PSO presented in the previous section we will rename $\phi_1 = c_1, \phi_2 = c_2$. As we will analyse several particles soon which leads to a difference between global and personal attractor. So both ϕ_1 and ϕ_2 remain in the following formula representing the most simplified model:

$$V(t+1) = V(t) + \phi_1 \cdot (P_i(t) - X(t)) + \phi_2 \cdot (P_i(t) - X(t))$$

3.1 Oczan & Mohan

This very simple model can be expanded by adding several particles and allowing multiple dimensions. Oczan & Mohan created the following equation, which is obtained by including the position equation into the simple version of the velocity update, describing the PSO:

$$X_i^d(t) - (2 - \phi_1 - \phi_2) \cdot X_i^d(t-1) + X_i^d(t-2) = \phi_1 P_i^d + \phi_2 P_i^d$$

3.2 Clerc & Kennedy

Clerc & Kennedy use the same simple PSO as Oczan & Mohan, but they do not only use the personal attractor P_i, instead they add the global attractor P_g. The second attractor is the reason ϕ_1 and ϕ_2 are different. Within this model particles have a stable point: $P = \frac{c_1 P_i + c_2 P_g}{c_1 + c_2}$. If they are at this point, they will stay there and not move away as long as they do not have any velocity remaining. They will certainly converge to this stable point, if convergence is ensured by the prefactor χ:

$$\chi = \frac{2\kappa}{2 - \phi - \sqrt{\phi^2 - 4\phi}} \text{ with } \phi = \phi_1 + \phi_2 \geq 4 \text{ and } \kappa \in [0, 1]$$

resulting in the following equation describing the particles:

$$V_i^d(t+1) = \chi[V_i^d(t) + \phi_1(P_i^d(t) - X_i^d(t)) + \phi_2(P_g^d(t) - X_i^d(t)]$$

Furthermore it is known that the smaller κ the faster particles converge to the stable point.

3.3 van den Bergh

Van den Bergh again uses the same simple PSO. Additionally he adds inertia weight to scale the influence of the previous velocity. This results in the following recurrence relation:

$$X_{i+1} = (1 + \omega - \phi_1 - \phi_2)X_i - \omega X_{i-1} + \phi_1 P_i + \phi_2 P_g$$

It is easy to obtain the following corresponding characteristic equation with it's two eigenvalues: λ_1 and λ_2

$$\lambda^2 - (1 + \omega - \phi_1 - \phi_2)\lambda + \omega = 0$$

$$\lambda_{1/2} = \frac{1 + \omega - \phi_1 - \phi_2 \pm \sqrt{(1 + \omega - \phi_1 - \phi_2)^2 - 4\omega}}{2}$$

The PSO can now be described via the eigenvalues and certain k_i which are constants depending on $\phi_1, \phi_2, \lambda, \omega$ as long as the swarm stagnates. If a new best position is discovered they can be updated:

$$X_t = k_1 + k_2\lambda_1^t + k_3\lambda_2^t$$

Now the particles converge if the absolute values of their eigenvalues $\lambda_{1/2}$ are < 1. This is what we wanted, because now the swarm will search potentially good areas around the good areas more throughoutly. This ultimately lead to the conditions $(1 + \omega - \phi_1 - \phi_2)^2 < 4\omega$ and $\max\{|\lambda_1|, |\lambda_2|\} < 1$. The first one is necessary as van den Bergh only dealt with complex eigenvalues. As long as these conditions are fulfilled, the position of the particles converges to the stable point:

$$\lim_{t \to +\infty} X_t = \frac{\phi_1 P_i + \phi_2 P_s}{\phi_1 + \phi_2}$$

4 Stochastical Version

In this section we analyse the PSO we introduced in the beginning. To find good parameters for the tuple (c_1, c_2, ω, a, b), the position of the particles is regarded as stochastical variable of which expectation and variance can be calculated. Starting from the particle movement equations with uniform random variables $r_{1,i}$ and $r_{2,i}$:

$$V_i^d(t+1) = \omega \cdot V_i^d(t) + c_1 r_{1,i}^d(t) \cdot (P_i^d(t) - X_i^d(t)) + c_2 r_{2,i}^d(t) \cdot (P_g^d(t) - X_i^d(t))$$
$$X_i^d(t+1) = a \cdot X_i^d(t) + b \cdot V_i^d(t+1)$$

We begin to analyse the expectation of the particle movement during stagnation of the swarm. In that situation it is suifficient to only take a single particle into account as each moves independently. Furthermore, each dimension is independent from the others, so it is enough to work with a single one-dimensional particle:

$$\begin{aligned} V(t+1) =& \omega V(t) + c_1 r_{1,t}(P_i - X(t)) + c_2 r_{2,t}(P_g - X(t)) \\ X(t+1) =& aX(t) + bV(t+1) \end{aligned}$$

First of all we want to remove b from the equation. It is superficial as it can easily be expressed by c_1 and c_2 as those three parameters only occur in the form $c_1 b$ or $c_2 b$ after the position equation is reformed to:

$$X(t+1) = (a + b\omega - c_1 b r_{1,t} - c_2 b r_{2,t})X(t) - a\omega X(t-1) + c_1 b r_{1,t} P_i + c_2 b r_{2,t} P_g$$

So setting b to 1 does not reduce the generality and makes further analysis easier.

4.1 Analysis of Expectation

In the next step we calculate the expectation of the particles' position. By doing so we hope to find some criteria ensuring the convergence of particle for the other parameters. If the particles converge during stagnation, they will exploit potentially good areas more throughoutly. The expectation can be calculated:

$$EX(t+1) = (a + \omega - \frac{c_1 + c_2}{2})EX(t) - a\omega EX_{(t-1)} + \frac{c_1 P_i + c_2 P_g}{2}$$

The corresponding characteristic equation of this recurrence relation leads to two eigenvalues:

$$\lambda^2 - (a + \omega - \frac{c_1 + c_2}{2})\lambda + a\omega$$

Similar to van den Bergh's analysis both eigenvalues' absolute value should be smaller than zero. Thus, the following theorem is achieved: If and only if conditions $-1 < a\omega < 1$ and $2(1-\omega)(a-1) < c_1 + c_2 < 2(1+\omega)(1+a)$ are satisfied together then $\{EX(t)\}$ is guaranteed to converge to

$$EX = \frac{c_1 P_i + c_2 P_g}{2(1-\omega)(1-a) + c_1 + c_2}$$

This equals the stable point in the simple version if $a = 1$.

4.2 Analysis of Variance

Furthermore, the variance has to be analysed. It should not be zero as we want the particles to explore nearby the potentially good areas. The only exception is the one particle having the same global and local attractor. The mathematical analysis of the variance is technically analogous to the one of the expectation:

$$DX(t+2) = (\psi^2 + DR(t) - \varpi)DX(t+1) - \varpi(\psi^2 - DR(t) - \varpi)DX(t) + \varpi^3 DX(t-1) +$$

$$R[(EX(t+1))^2 + \varpi(EX(t)^2] - 2E(R(t)Q(t))(EX(t-1) + \varpi EX(t)) + DQ(t)(1+\varpi)$$

where $\overline{\omega} = a\omega$, $v = (c_1 + c_2)/2$, $\psi = a + \omega - v$, $R(t) = c_1 r_{1,t} + c_2 r_{2,t} - v$, $Q(t) = c_1 r_{1,t} P_i + c_2 r_{2,t} P_g$, $X(t+1) = (\psi - R(t))X(t) - \overline{\omega}X(t-1) + Q(t)$
Once again one can calculate the characteristic equation and its corresponding eigenvalues:

$$\chi(\lambda) = \lambda^3 - (\psi^2 + DR(t) - \overline{\omega})\lambda^2 + \overline{\omega}(\psi^2 - DR(t) - \overline{\omega})\lambda - \overline{\omega}^3$$

As long as $c_1 \neq 0$ or $c_2 \neq 0$ are satisfied, the eigenvalues can be ordered, such that $1 > \lambda_{maxD} = \lambda_{D1} > |\overline{\omega}| \geq |\lambda_{D2}| \geq |\lambda_{D3}| > 0$ and $\lambda_{maxD} \in \mathbb{R}$. This once again leads to a theorem: Given $c_1 \neq 0$ or $c_2 \neq 0$, if and only if $-1 < \overline{\omega} < 1$ and $\chi(1) > 0$ are satisfied together $\{DX_t\}$ is guaranteed to converge to

$$DX = \frac{\frac{1}{12}(1+\overline{\omega})}{\chi(1)[2(1-a)(1-\omega)+c_1+c_2]^2}\{2c_1^2c_2^2(P_g-P_i)^2+$$

$$4(a-1)(1-\omega)c_1c_2(c_2P_g-c_1P_i)(P_g-P_i)+(c_1^2P_i^2+c_2^2P_g^2)[2(1-a)(1-\omega)]^2\}$$

and $\lambda_{maxD} < 1$.
To put it into a nutshell, the following criteria are necessary for the swarm to converge:

$$(b = 1) \tag{1}$$
$$|a\omega| < 1 \tag{2}$$
$$2(1-\omega)(a-1) < c_1 + c_2 < 2(1+\omega)(1+a) \tag{3}$$
$$c_1 \neq 0 \text{ or } c_2 \neq 0, \tag{4}$$
$$\chi(1) > 0 \tag{5}$$

Using these constrictions one can begin to select a parameter tuple which hopefully works rather well.

5 Parameter Selection

We found criteria necessary for convergence of the particles in the previous section. Now it is time to select them as good as possible. Therefore two search components - exploration and exploitation - have to be balanced. Exploration is needed for the swarm to search the whole space roughly whereas exploitation is the focusing on certain potentially good areas. λ_{maxD} is the defining factor for the swarm's exploration ability. The larger λ_{maxD}, the more space is scanned. This ability is the main strength of PSO whereas exploitation is more of its weakness. Nevertheless λ_{maxD} should be roughly around 0.9.
One parameter was not analysed before. When selecting a it is possible to enhance the algorithm by ensuring that the particle which found the global attractor converges to that attractor with zero variance. As $P_i = P_g$ for this particle the way to achieve this is to set a to 1. That way the expectation of the variance DX becomes 0. The other criteria to be satisfied are $-1 < \omega < 1$, $c_1 + c_2 < 4(1+\omega)$, $\chi(1) > 0$. Literature now suggests several different values for (c_1, c_2, ω) which we compare in the next section:

	ω	c_1	c_2	λ_{maxD}
Clerc & Kennedy	0.729	1.494	1.494	0.942
Trelea	0.6	1.7	1.7	0.889
Carlisle & Dozier	0.729	2.041	0.948	0.975
Jiang & Luo & Yang	0.715	1.7	1.7	0.995

6 Comparison

We obtained several suggestions for the parameters as shown in the table at the end of the previous section. The next step is to compare these parameters for the PSO heuristic by testing them on different test functions. There are a few benchmark functions often used to compare optimization algorithms:

function	formula	optimal position
sphere	$\Sigma_{i=1}^d x_i^2$	0
griewank	$\frac{1}{4000}\Sigma_{i=1}^d x_i^2 - \prod_{i=1}^d \cos(\frac{x_i}{\sqrt{i}}) + 1$	0
rastrigin	$\Sigma_{i=i}^d (x_i^2 - 10\cos(2\pi x_i) + 10)$	0
rosenbrock	$\Sigma_{i=i}^{d-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	(1,...,1)

The following data is from the paper of Jiang, Luo and Yang. They tested each of the parameter options on each of the functions with 20, 40 and 100 particles for dimensions 10, 30 and 50. The results of 100 runs for each combination are shown in the following tables. The success rate is the percentage of the particle swarm finding a certain ϵ-sphere around the maximum within the boundaries of the search space. Success iterations is the median of the number of iterations required to find the sphere. After 10000 iterations without success the algorithm stopped:
dimension 10:

		Clerc	Carlisle	Trelea	Jiang
Succes Rate	Ackley	0.92	0.90	0.86	0.96
	Griewank	0.64	0.90	0.67	0.72
	Rastrigin	0.68	1.00	0.96	0.88
	Rosenbrock	1.00	0.99	1.00	1.00
	Sphere	1.00	1.00	1.00	1.00
Success Iteration	Ackley	84	72.5	63	108
	Griewank	265	192	160.5	426
	Rastrigin	179	161	141	201
	Rosenbrock	333.5	319.5	280.5	538.5
	Sphere	186	162	139	259

dimension 50:

		Clerc	Carlisle	Trelea	Jiang
Succes Rate	Ackley	0.77	0.60	0.82	1.00
	Griewank	0.94	0.94	0.93	0.98
	Rastrigin	0.25	0.80	0.27	0.97
	Rosenbrock	0.80	0.60	0.91	0.61
	Sphere	1.00	1.00	1.00	1.00
Success Iteration	Ackley	257	214	214	471
	Griewank	383	214	338	787
	Rastrigin	∞	262	∞	597
	Rosenbrock	2188	3981	2205	5474
	Sphere	603	332	531	1281

This shows that PSO is just an heuristic which has a problem with the curse of dimension. So the larger the dimension the weaker the swarm's ability to find the global optima.

7 Conclusion

PSO is a viable method to find extrema of black-box functions. Using the right parameters for the swarm, it is possible to exploit the area around the attractors and to slowly converge closer to the optimum. Currently swarm behaviour is mainly analysed by testing the swarm many times, but with stochastical analysis it is possible to figure out how to choose the swarm's parameters for optimal results. Nevertheless, it is not possible to ensure that the swarm finds a good local or even the global maximum as PSO is only an heuristic approaching the solution sometimes.

8 Sources

M. Jiang, Y. P. Luo and S. Y. Yang (2007). Particle Swarm Optimization - Stochastic Trajectory Analysis and Parameter Selection, Swarm Intelligence, Focus on Ant and Particle Swarm Optimization, Felix T. S. Chan and Manoj Kumar Tiwari (Ed.), ISBN: 978-3-902613-09-7, InTech, Available from: `http://www.intechopen.com/books/swarm_intelligence_focus_on_ant_and_particle_swarm_optimization_stochastic_trajectory_analysis_and_parameter_selection`

S. Helwig (2010): Particle Swarms for Constrained Optimization, Partikelschwärme für Optimierungsprobleme mit Nebenbedingungen. Universität Erlangen-Nürnberg.