ANALYSIS OF A SIMPLE PARTICLE SWARM OPTIMIZATION SYSTEM

ENDER OZCAN AND CHILUKURI K. MOHAN
eozcan/mohan@top.cis.syr.edu
2-120 CST Department of EECS
Syracuse University, Syracuse, NY 13244-4100, U.S.A.

ABSTRACT:
Particle Swarm Optimization (PSO) is a recently proposed approach, based on social behavior of organisms such as birds and fishes (Kennedy and Eberhart, 1995). Search is conducted by "flying" particles in the space. In research results reported so far, the trajectories of even a simple particle swarm optimization system have not been formally obtained. This paper provides the trajectory equations for various parameter values in closed form.

INTRODUCTION

Birds in a group flock synchronously, but in an unpredictable manner, e.g., scattering suddenly, then regrouping again. Studies to simulate the underlying rules of bird social behavior led to a new approach for optimizing real-valued functions, "Particle Swarm Optimization" (PSO) (Kennedy and Eberhart, 1997, 1995). This new approach combines notions of evolutionary computation and swarm theory (Kennedy, 1997).

PSO provides a population-based search procedure, where each individual is abstracted as a "particle" that flies around in a multidimensional search space. The best positions encountered by a particle and its neighbor determine the particle's trajectory along with other PSO parameters. In other words, a PSO system attempts to balance exploration and exploitation by combining local and global search methods. In this aspect, PSO is similar to modern GAs and memetic algorithms. In addition to providing a tool for optimization, another important aspect of the PSO approach is its applicability in analyzing socio-cognition of human and artificial agents, based on principles of social psychology. In proposing the PSO approach, Kennedy (1998, 1997) suggests that knowledge is optimized by social interaction, and that thought processes are not completely separate for different individuals, they involve interpersonal communication.

Angeline (1998) performed an empirical comparison of PSO with EP, experimenting with different functions, initialization techniques and dimensionality. Eberhart and Shi (1998) show that the behavior of a PSO system falls in between genetic algorithms and evolutionary programming. Experimental results demonstrated that the PSO algorithms were competitive. Both Eberhart and Angeline advocated focusing on hybrid implementations.

To the best of our knowledge, all previous research on PSO systems has provided empirical results and informal analyses. This paper presents the first formal analysis of a traditional simple particle system, crucial to understand the dynamics of how particle behavior depends on parameters for making the right choices of parameter values.

PSO ALGORITHM

Each particle in a PSO system is associated with a point in a multidimensional search space in which we are attempting to find an optimal location with respect to a fitness function f. We will use the following terminology:

- $P(t)$ is a population of m particles in a multidimensional space at time step t, where
$$P(t) = (P_1(t), ..., P_j(t), ..., P_i(t), ..., P_m(t))$$
- $P_i(t)$ is the i^{th} particle in a $d-dimensional$ space at time step t, where
$$P_i(t) = (x_{i,1}(t), x_{i,2}(t), ..., x_{i,d}(t))$$
- The velocity of $P_i(t)$ is represented in a $d-dimensional$ space as
$$V_i(t) = (v_{i,1}(t), v_{i,2}(t), ..., v_{i,d}(t))$$
- $N(P_i)$, the neighborhood of a particle P_i, is defined as all those particles P_k such that P_i and P_k are "near" each other, in the context of a predefined topology.
- Define $P_j^{('')}(t)$ (global best) to be the best position of a particle P_j at time step t, such that $P_j \in N(P_i)$ and $f(P_j(t'')) \geq f(P_k(t-1))$ for all $t'' \leq t-1$ and $P_k \in N(P_i)$.
- Define $P_i^{(*)}(t)$ (local best) to be the best previous position of P_i at time step t, such that $f(P_i^{(*)}(t)) \geq f(P_i(t-1))$, for all $t^* \leq t-1$.
- Using the above notation, each particle P_i moves according to the following equation:
$$\begin{aligned} V_i(t) &= V_i(t-1) + c_1 U_1(0,1)(P_i^{(*)}(t) - P_i(t-1)) + \\ &\quad c_2 U_2(0,1)(P_j^{('')}(t) - P_i(t-1)) \qquad (1) \\ P_i(t) &= P_i(t-1) + V_i(t) \qquad (2) \end{aligned}$$
  where c_1 and c_2 are two positive constants and U(0,1) is a uniformly distributed random number in [0,1].

Shi *et al.* (1998) suggested modifying Equation 1, incorporating an inertia weight w, as follows:

$$\begin{aligned} V_i(t) &= wV_i(t-1) + c_1 U_1(0,1)(P_i^{(*)}(t) - P_i(t-1)) + \\ &\quad c_2 U_2(0,1)(P_j^{('')}(t) - P_i(t-1)) \qquad (3) \end{aligned}$$

Initially, particles are placed in the space randomly. If the topology is defined such that all particles are assumed to be neighbors, then $P_j^{('')}(t)$ becomes the best solution found so far, resembling the approach of KNUX (Maini *et al.*, 1994). Particles start with a constant initial velocity. They update their velocities, then move to their new location.

SIMPLIFYING THE PSO SYSTEM

In order to understand the behavior of a complex system, it often helps to begin by examining a simpler version of it. For PSO systems, we simplify analyses by considering systems with only one dimension in the search space for the case where $P_j^{('')}(t) = P_i^{(*)}(t)$. Then Equations 1 and 2 simplify to:

$$v_i(t) = v_i(t-1) + (\phi_1 + \phi_2)(x_i^*(t) - x_i(t-1)) \quad (4)$$
$$x_i(t) = x_i(t-1) + v_i(t) \quad (5)$$

where $\phi_1 = c_1 U_1(0,1), \phi_2 = c_2 U_2(0,1)$.

Assuming ϕ_1, ϕ_2 and $x_i^*(t) = p$ are constants and imposing the initial conditions $v(0) = v_0$ and $x(0) = x_0$, we can further simplify the system. A simple particle's behavior is defined by the following equation:

$$v(t) = v(t-1) - \phi x(t-1) + \phi p \quad (6)$$
$$x(t) = x(t-1) + v(t) \quad (7)$$

where $\phi = \phi_1 + \phi_2$.

By substitution, we obtain the following recursion from Equations 6 and 7:

$$x(t) = (2-\phi)x(t-1) - x(t-2) + \phi p \quad (8)$$

with initial conditions, $x(0) = x_0, x(1) = x_0(1-\phi) + v_0 + \phi p$.

The following closed form can be obtained by using generating functions or any method for solving non-homogeneous linear recurrence equations for the displacement.

$$x(t) = \alpha((2-\phi+\delta)/2)^t) + \beta((2-\phi-\delta)/2)^t) + p \quad (9)$$

where

$$\delta = \sqrt{\phi^2 - 4\phi} \quad (10)$$
$$\beta = (x_0 - p)(\delta + \phi)/(2\delta) - v_0/\delta \quad (11)$$
$$\alpha = x_0 - p - \beta \quad (12)$$

TRAJECTORY OF A SIMPLE PARTICLE

The significance of Equation 9 is that at any time of the run we can determine the trajectory of an isolated particle whose personal best is the same as the global best. Given the assumption that a particle has visited a local optimum at

$x = 0$, let us analyze the trajectory of a simple particle, choosing this location as a starting point for our search ($x_0 = p = 0$). The trajectory equation becomes:

$$x(t) = (v_0/\delta)(((2-\phi+\delta)/2)^t - ((2-\phi-\delta)/2)^t) \quad (13)$$

$$\delta = \sqrt{\phi^2 - 4\phi} \quad (14)$$

If $U = \phi^2 - 4\phi > 0$ then δ is a positive real number, otherwise it is a complex number. The roots of parabola U are at 0 and 4. Recall that $\phi \geq 0$. We can analyze the trajectory equation in two domains, $\delta \in \Re$ (Real) and $\delta \in \Im$ (Complex).

Real δ

δ is a real number when $\phi = 0$, $\phi = 4$ and $\phi > 4$.

Case $\phi = 0$: This is a special case, producing the following recursive trajectory equation:

$$x(t) = 2x(t-1) - x(t-2) \quad (15)$$

Solving this equation yields

$$x(t) = (x_0 - p) + v_0 t \quad (16)$$

Note that given $x_0 = p = 0$, the particle will move in the initial direction with the initial velocity forever ($x(t) = v_0 t$).

Case $\phi = 4$: The recursive trajectory equation becomes as follows:

$$x(t) = -2x(t-1) - x(t-2) + 4p \quad (17)$$

The closed form for Equation 17 is:

$$x(t) = ((x_0 - p) + (2(x_0 - p) - v_0)t)(-1)^t + p \quad (18)$$

It is obvious from Equation 18 that given $x_0 = p = 0$, the particle will move in the in opposite directions at consecutive time steps, with an increase in the speed proportionate to the initial velocity ($x(t) = -v_0 t(-1)^t$).

Case $\phi > 4$: The trajectory of a particle, as determined by Equations 13- 14, can be described as an oscillatory graph bounded by exponential functions. The initial velocity determines the steepness of the envelope by which this oscillation is bounded, along with other parameters. As we increase the initial velocity, the envelope will diverge faster and this will cause the step size of particle to increase further, allowing it to search in a larger space. Increasing ϕ may have a similar effect as increasing the velocity with a higher magnitude.

When a particle is in the zone where δ is real, it will increase its step size exponentially at each time step. This will necessitate a control action, since the particle might stray out of the search space and not come back.

Complex δ

From Equations 9- 12, δ is a complex number when $0 < \phi < 4$. We can rewrite the trajectory equations as:

$$x(t) = \alpha z_1^t + \beta z_2^t \quad (19)$$

where

$$z_1 = (2 - \phi + \delta')/2 \tag{20}$$
$$z_2 = (2 - \phi - \delta')/2 \tag{21}$$
$$\beta = (x_0 - p)(\delta' + \phi)/(2\delta') - v_0/\delta' \tag{22}$$
$$\alpha = x_0 - p - \beta \tag{23}$$
$$\delta' = i\sqrt{|\phi^2 - 4\phi|} \tag{24}$$

Assuming $x_0 = p = 0$ the trajectory equation simplifies to:

$$x(t) = (2v_0/||\delta'||)sin(atan(||\delta'||/(2-\phi))t) \tag{25}$$

This equation implies that the trajectory of a simple particle in complex δ zone is a sinusoidal wave and our choice of parameters determines the amplitude and the frequency of the wave. In other words, our choice of parameters determines the direction and step size of a particle. Note that ν in $sin(2\pi\nu)$ is the frequency of the sine wave, with period $T = 1/\nu$. Also recall that $atan$ is an increasing function. Due to the periodic nature of the trajectory equation, a 'good' particle in this zone might get stuck searching the regions that have already been searched, unless another particle in its neighborhood finds a better point.

We can divide this domain into subregions depending on ϕ, as described below:
Case $0 < \phi \leq 2 - \sqrt{3} \approx 0.268$: The amplitude of the sine wave will increase, as ϕ decreases, since $||\delta'||$ becomes < 1. Meanwhile we will have a wave whose frequency is relatively smaller, making the period larger.
Case $\phi = 2$: The term $x(t-1)$ drops out in Equation 8, the trajectory equation becomes:

$$x(t) = v_0 sin(t\pi/2) \tag{26}$$

Notice that for $\phi < 2$, amplitude of the sine waves will be positive. Whenever $\phi > 2$, the sine waves will be lagging behind by π.
Case $2 - \sqrt{3} < \phi < 2$ and $2 < \phi \leq 2 + \sqrt{3} \approx 3.732$: $||\delta'||$ becomes > 1. Since $||\delta'||$ is also bounded by a maximum of $\sqrt{5}$ in magnitude, amplitude of the sine wave will be approximately v_0.
Case $2 + \sqrt{3} < \phi < 4$: The amplitude of the sine wave will increase with ϕ. Note that similar to the $0 < \phi \leq 2 - \sqrt{3}$ case, $||\delta'||$ becomes < 1, reducing the frequency which is inversely proportional to the period of a sine wave.

CONCLUSIONS

We have formalized the behavior of a simple particle, which gives us a sense of what the trajectories of particles in a PSO system might look like. Based on our results, intelligent parameter adaptation methods can be incorporated into the original PSO equations. The simulations of Kennedy in 1998a correspond exactly to our analytical results, empirically supporting our formal analyses.

For a simple particle, there are four search types, setting the other parameters constant. Using type 1, search is conducted by increasing step sizes in the space.

Due to this behavior, type 1 might not be favored, since it might take a particle out of the search space without letting it return. In other types, a particle may get stuck and repeat exploring a part of the search space already visited. For other types, a particle follows a path on a sinusoidal wave searching a space bounded by the amplitude of the sine wave. In type 3, the amplitude of the sine wave is approximately the initial velocity. In types 2 and 4, the amplitude is magnified.

Our results support the traditional PSO, rather than the modified one with inertia weight (Shi and Eberhart, 1998). Inertia weight arranges the boundaries in between types (cases). Particles "surf" the search space on sine waves. Each particle attempts to "catch" another wave randomly, trying to move to an optimal location. A wave with a high frequency or amplitude is not necessarily a good wave for a particle. Note that limiting the velocity (V_{max}) seems to help the particle to "jump" onto another wave. More details can be found in Ozcan, 1998.

REFERENCES

Angeline, P., (1998). Evolutionary Optimization versus Particle Swarm Optimization: Philosophy and Performance Differences, *The Seventh Annual Conf. on Evolutionary Programming.*

Eberhart, R., and Shi, Y., (1998). Comparison between Genetic Algorithms and Particle Swarm Optimization, *The Seventh Annual Conf. on Evolutionary Programming.*

Kennedy, J., (1998). Thinking is social: Experiments with the adaptive culture model, *Journal of Conflict Resolution*, 42, 56-76.

Kennedy, J., (1998a). The Behavior of Particles, *The Seventh Annual Conf. on Evolutionary Programming.*

Ozcan, E., (1998). Shape Matching Using Genetic Algorithms, *PhD Dissertation.*

Shi, Y., and Eberhart, R., (1998). Parameter Selection in Particle Swarm Optimization. *The Seventh Annual Conf. on Evolutionary Programming.*

Kennedy, J., (1997). The particle swarm: social adaptation of knowledge. *Proc. of IEEE Int. Conf. on Evolutionary Computation (Indianapolis, Indiana)*, pp. 303-308. IEEE Service Center, Piscataway, NJ.

Kennedy, J., and Eberhart, R. C., (1997). A discerete binary version of the particle swarm algorithm. *Proceedings of Conf. on Systems, Man, and Cybernetics*, pp. 4104-4109. IEEE Service Center, Piscataway, NJ.

Kennedy, J., and Eberhart, R. C., (1995). Particle swarm optimization. *Proceedings of IEEE Int. Conf. on Neural Networks*, (Perth, Australia), pp. 1942-1948. IEEE Service Center, Piscataway, NJ.

Maini, H., Mehrotra, K., Mohan, C., and Ranka, S., (1994). Knowledge-based Nonuniform Crossover. *Complex Systems*, Vol. 8, pp. 257-293.