An Introduction to Particle Swarm Optimization

Matthew Settles
Department of Computer Science, University of Idaho,
Moscow, Idaho U.S.A 83844

November 7, 2005

1 Introduction

When the search space is too large to search exhaustively, population based searches may be a good alternative, however, population based search techniques cannot guarantee you the optimal (best) solution.

I will discuss a population based search technique, Particle Swarm Optimization (PSO). The PSO Algorithm shares similar characteristics to Genetic Algorithm, however, the manner in which the two algorithms traverse the search space is fundamentally different.

Both Genetic Algorithms and Paticle Swarm Optimizers share common elements:

1. Both initialize a population in a similar manner.
2. Both use an evaluation function to determine how fit (good) a potential solution is.
3. Both are generational, that is both repeat the same set of processes for a predetermined amount of time.

Algorithm 1 Population Based Searches

```
 1: procedure PBS
 2:     Initialize the population
 3:     repeat
 4:         for $i = 1$ to number of individuals do
 5:             $G(\vec{x}_i)$                                  ▷ $G()$ evaluates goodness
 6:         end for
 7:         for $i = 1$ to number of individuals do
 8:             $P(\vec{x}_i, \theta)$          ▷ Modify each individual using parameters $\theta$
 9:         end for
10:     until stopping criteria
11: end procedure
```

2 Particle Swarm Optimization

Particle Swarm Optimization was first introduced by Dr. Russell C. Eberhart [1] and Dr. James Kennedy [2] in 1995.

As described by Eberhart and Kennedy, the PSO algorithm is an adaptive algorithm based on a social-psychological metaphor; a population of individuals (referred to as particles) adapts by returning stochastically toward previously successful regions[1].

Particle Swarm has two primary operators: Velocity update and Position update. During each generation each particle is accelerated toward the particles previous best position and the global best position. At each iteration a new velocity value for each particle is calculated based on its current velocity, the distance from its previous best position, and the distance from the global best position. The new velocity value is then used to calculate the next position of the particle in the search space. This process is then iterated a set number of times, or until a minimum error is achieved.

Algorithm 2 Particle Swarm algorithm

```
 1: procedure PSO
 2:     repeat
 3:         for $i = 1$ to number of individuals do
 4:             if $G(\vec{x}_i) > G(\vec{p}_i)$ then                  ▷ $G()$ evaluates goodness
 5:                 for $d = 1$ to dimensions do
 6:                     $p_{id} = x_{id}$                  ▷ $p_{id}$ is the best state found so far
 7:                 end for
 8:             end if

 9:             $g = i$                                                    ▷ arbitrary
10:             for $j$ = indexes of neighbors do
11:                 if $G(\vec{p}_j) > G(\vec{p}_g)$ then
12:                     $g = j$      ▷ $g$ is the index of the best performer in the neighborhood
13:                 end if
14:             end for

15:             for $d = 1$ to number of dimensions do
16:                 $v_{id}(t) = f(x_{id}(t-1), v_{id}(t-1), p_{id}, p_{gd})$      ▷ update velocity
17:                 $v_{id} \in (-V_{max}, +V_{max})$
18:                 $x_{id}(t) = f(v_{id}(t), x_{id}(t-1))$                        ▷ update position
19:             end for
20:         end for
21:     until stopping criteria
22: end procedure
```

[1]Dr. Russell C. Eberhart is the Chair of the Department of Electrical and Computer Engineering, Professor of Electrical and Computer Engineering, and Adjunct Professor of Biomedical Engineering at the Purdue School of Engineering and Technology, Indiana University Purdue University, Indianapolis (IUPUI).

[2]Dr. James Kennedy is a research psychologist, at the Bureau of Labor and Statistics in Washington, DC.

3 Definitions and Variables Used

PSO Particle Swarm Optimizer.

t means the current time step, $t-1$ means the previous time step.

T_{max} the maximum number of time step the swarm is allowed to search.

$P(x_{id}(t)=1)$ is the probability that individual i will choose 1 for the bit at the dth site on the bitstring.

$x_{id}(t)$ is the current state (position) at site d of individual i.

$v_{id}(t)$ is the current velocity at site d of individual i.

$\pm V_{max}$ is the upper/lower bound placed on v_{id}.

p_{id} is the individual's i best state (position) found so far at site d.

p_{gd} is the neighborhood best state found so far at site d.

c_1 social parameter 1, a positive constant, ususally set to 2.0.

c_2 social parameter 2, a positive constant, usually set to 2.0.

φ_1 is a positive random number drawn form a uniform distribution between 0.0 and 1.0.

φ_2 is a positive random number drawn from a uniform distribution between 0.0 and 1.0.

ρ_{id} is a positive random number, drawn from a uniform distribution between 0.0 and 1.0 (Binary Particle Swarm).

$w(t)$ is the inertia weight (Inertia Particle Swarm).

w_{start} is the starting inertia weight ($w(0)=w_{start}$). (Inertia Particle Swarm)

w_{end} is the ending inertia weight ($w(T_{max})=w_{end}$). (Inertia Particle Swarm)

χ is the constriction coefficient (Constriction Coefficient Particle Swarm).

4 Binary Particle Swarm Optimizer

Model of Binary Decision [2]. The probability that an individual's decision will be yes or no, true or false, or some other binary decision is a function of personal and social factors.

$$\begin{aligned} P(x_{id}(t)=1) &= f(x_{id}(t-1), v_{id}(t-1), p_{id}, p_{gd}) \\ P(x_{id}(t)=0) &= 1-P(x_{id}(t)=1) \end{aligned} \tag{4.1}$$

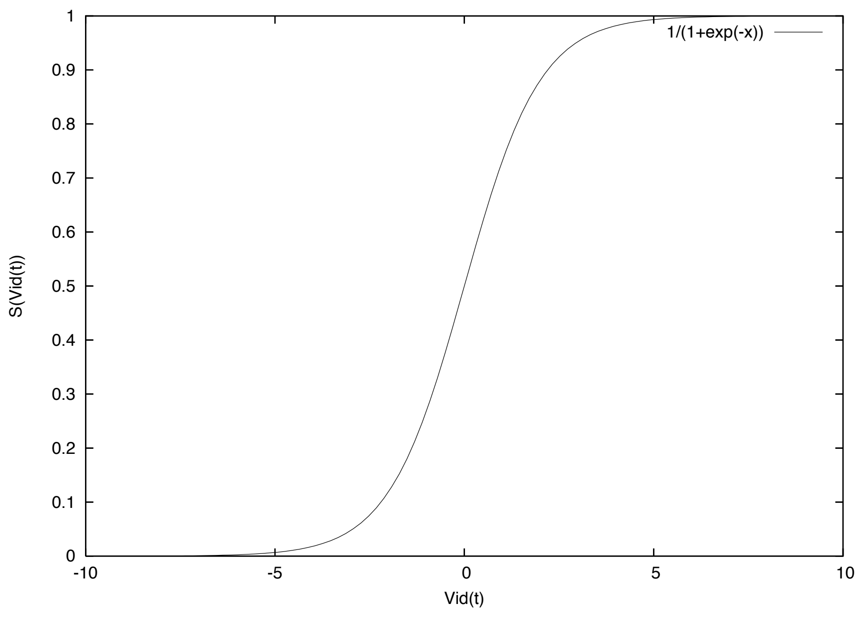

Figure 1: Sigmoidal Function

The parameter $v_{id}(t)$, an individuals predisposition to make one or the other choice, will determine a probability threshold. If $v_{id}(t)$ is higher, the individual is more likely to choose 1, and lower values favor the 0 choice. Such a threshold needs to stay in the range [0.0,1.0]. The sigmoidal function is a logical choice to do this. The sigmoidal function squashes the range of v_{id} to a range of [0.0,1.0].

$$s(v_{id}) = \frac{1}{1 + \exp(-v_{id})} \tag{4.2}$$

Finally a formula for modeling binary decision making is as follows.

$$\begin{aligned} v_{id}(t) = v_{id}(t-1) + c_1\varphi_1(p_{id} - x_{id}(t-1)) + c_2\varphi_2(p_{gd} - x_{id}(t-1)) \\ \text{if } \rho_{id} < s(v_{id}(t)) \text{ then } x_{id}(t) = 1; \text{ else } x_{id}(t) = 0 \end{aligned} \tag{4.3}$$

Further more we can limit v_{id} so that $s(v_{id})$ does not approach too closely to 0.0 or 1.0. This ensures that there is always some chance of a bit flipping. A constant parameter V_{max} can be set at the start of a trial to limit the range of vid is often set at ± 4.0, so that there is always at lease a chance of $s(v_{max}) \approx 0.0180$ that a bit will change state. In this binary model, V_{max} functions similarly to mutation rate in genetic algorithms.

5 Standard Particle Swarm Optimizer

In real number space, the parameters of a function can be conceptualized as a point in space. Furthermore the space in which the particles move is heterogeneous with respect to fitness; that is some regions are better than others. A number of particles can be evaluated and there is presumed to be some kind of preference or attraction for better regions of the search space.

$$x_{id}(t) = f(x_{id}(t-1), v_{id}(t-1), p_{id}, p_{gd}) \tag{5.1}$$

$$\begin{aligned} v_{id}(t) &= v_{id}(t-1) + c_1\varphi_1(p_{id} - x_{id}(t-1)) + c_2\varphi_2(p_{gd} - x_{id}(t-1)) \\ x_{id}(t) &= x_{id}(t-1) + v_{id}(t) \end{aligned} \tag{5.2}$$

The standard version of the PSO has a tendancy to explode as oscillations become wider and wider, unless some method is applied for damping the velocity. The usual method for preventing explosion is simply to define a parameter V_{max} and prevent the velocity from exceeding it on each dimension d for individual i. Typically V_{max} is set to X_{max}, the maximum initalization range of x_{id}.

$$\begin{aligned} &\text{if } v_{id} > V_{max} \text{ then } v_{id} = V_{max} \\ &\text{else if } v_{id} < -V_{max} \text{ then } v_{id} = -V_{max} \qquad (5.3) \\ &\qquad (5.4) \end{aligned}$$

Other methods have also been introduced that deal with controlling the explosion of v_{id}, the two most notable are Eberhart and Shi's PSO with inertia and Clerc's PSO with Constriction.

6 Particle Swarm Optimizer with Inertia

In 1998 Shi and Eberhart came up with what they called PSO with inertia. The inertia weight is multiplied by the previous velocity in the standard velocity equation and is linearally decreased throughtout the run. A nonzero inertia weight introduces a preference for the particle to continue moving in the same direction it was going on the previous iteration. Decreasing the inertia over time introduces a shift from the exploratory (global search) to the exploitative (local search) mode.

$$x_{id}(t) = f(w(t), x_{id}(t-1), v_{id}(t-1), p_{id}, p_{gd}) \tag{6.1}$$

$$\begin{aligned} v_{id}(t) &= w(t) * v_{id}(t-1) + c_1\varphi_1(p_{id} - x_{id}(t-1)) + c_2\varphi_2(p_{gd} - x_{id}(t-1)) \\ x_{id}(t) &= x_{id}(t-1) + v_{id}(t) \end{aligned} \tag{6.2}$$

Typically $w(t)$ is reduced linearly, from w_{start} to w_{end}, each iteration, a good starting point is to set w_{start} to 0.9 and w_{end} to 0.4.

$$w(t) = \frac{(T_{max} - t) * (w_{start} - w_{end})}{T_{max}} + w_{end} \tag{6.3}$$

Thought V_{max} has been found not to be necessary in the PSO with inertia version, however it can be useful and is suggested that a $V_{max} = X_{max}$ be used.

7 Particle Swarm Optimizer with Constriction Coefficient

Another PSO implementation dubbed PSO Constriction Coefficient was developed by Clerc [3] in 2000. Clerc modeled the Particle Swarms interactions using a set of complicated linear equations. Using a constriction coefficient results in particle convergence over time. That is the amplitude of the particle's oscillatoins decreases as it focuses on the local and neighborhood previous best points. Though the particle converges to a point over time, the contriction coefficient also prevents colapse if the right social conditions are in place. The particle will oscillate around the weighted mean of p_{id} and p_{gd}, if the previous best position and the neighborhood best position are near each other the particle will perform a local search. If the previous best position and the neighborhood best position are far apart from each other the particle will perform a more exploratory search (global search). During the search the neighborhood best position and previous best position will change and the particle will shift from local search back to global search. The constriction coefficient method therefor balances the need for local and global search depending on what social conditions are in place.

$$x_{id}(t) = f(\chi, x_{id}(t-1), v_{id}(t-1), p_{id}, p_{gd}) \tag{7.1}$$

$$\chi = \frac{2k}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, where \varphi = c_1 + c_2, \varphi > 4 \tag{7.2}$$

$$\begin{aligned} v_{id}(t) = \chi[v_{id}(t-1) + c_1\varphi_1(p_{id} - x_{id}(t-1)) + c_2\varphi_2(p_{gd} - x_{id}(t-1))] \\ x_{id}(t) = x_{id}(t-1) + v_{id}(t) \end{aligned} \tag{7.3}$$

Clerc, et al., found that by modifying φ, the convergence characteristics of the system can be controlled. Typically $k = 1$ and $c_1 = c_2 = 2$, and φ is set to 4.1, thus $K = 0.73$.

8 Neighborhood Topologies

There are 3 main neighborhood topologies used in PSO: circle, wheel, and star. The choice for neighborhood topology determines which individual to use for p_{gd}. In the circle toplogy (See Figure 2), each individual in socially conneted to its k nearest topological neighbors ($p_{gd} =$ best individual result among its k nearest neighbors, k typically equal 2

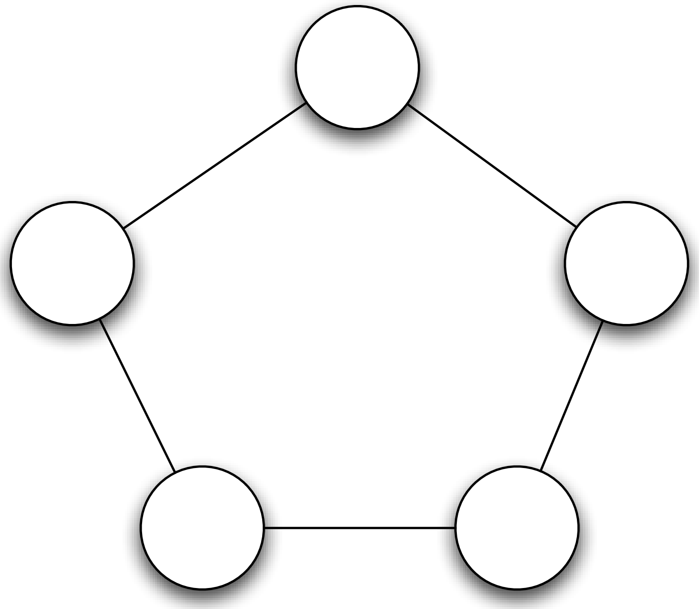

Figure 2: Circle Topology

). The wheel topology (See Figure 3), effectively isolates individuals from one another, as information has to be communicated through a focal individual, p_{fd} ($p_{gd} = best\{p_{fd}, p_{id}\}$). The star topology (See Figure 4) is better known as the global best topology, here every individual is connected to every other individual (p_{gd} = best individual results in the population).

References

[1] J. Kennedy and R. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2001.

[2] J. Kennedy and R. C. Eberhart. A discreet binary version of the particle swarm algorithm, 1997.

[3] Clerc M. The swarm and the queen: Towards a determininistic and adaptive particle swarm optimization. In *Congress on Evolutionary Computation (CEC99)*, pages 1951–1957, 1999.

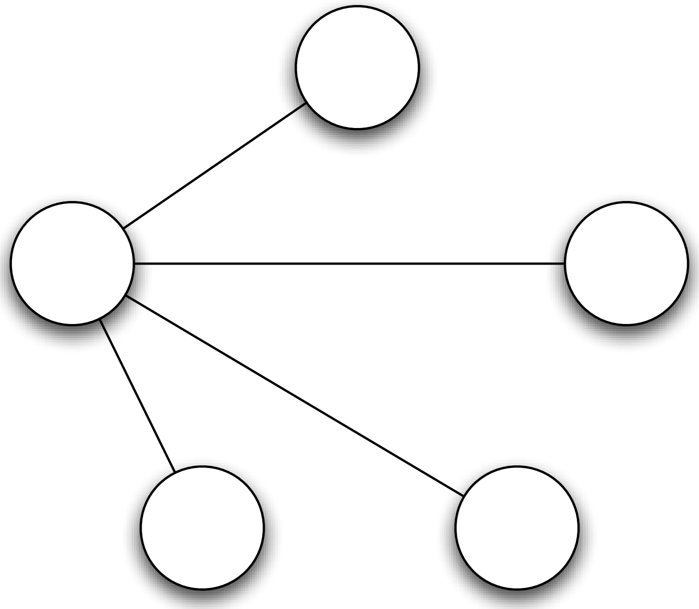

Figure 3: Wheel Topology

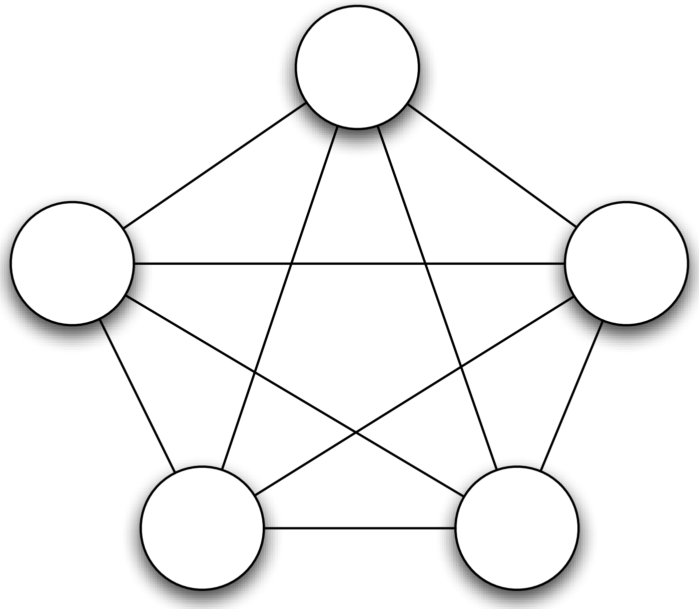

Figure 4: Star Topology