Discrete Evaluation and the Particle Swarm Algorithm.

Tim Hendtlass and Tom Rodgers,
Centre for Intelligent Systems and Complex Processes,
Swinburne University of Technology,
P. O. Box 218 Hawthorn
Australia 3122.
{thendtlass,trodgers@swin.edu.au}

Abstract

We propose that the optimal performance of the PSO algorithm should differ from that of the real life creatures on which PSO is modelled. If a bird finds a good food source, the likely behaviour for a flock is to congregate there, settle and feed. However, once PSO has found an optimum, while some particles should explore in the immediate vicinity for any better optimum present, the rest of the swarm should set out to explore new areas.

The common PSO practice of only evaluating each particle's performance at discrete intervals can, at small computational cost, be used to automatically adjust the PSO behaviour in situations where the swarm is 'settling' so as to encourage part of the swarm to explore further.

1 Introduction

The Particle Swarm Optimisation (PSO) algorithm is based on the movement of real creatures, such as birds, and attempts to model the way they balance their behaviour between exploration and exploitation, thus hoping to replicate the good search behaviour found in nature. As an optimisation tool, PSO has proved extremely effective on a wide range of problems. There are, however, two important differences between the artificial PSO algorithm and the real world behaviour that together lead to differences between their behaviours. These differences are:

- Real creatures continuously evaluate their current situation. Conversely, in PSO we can only evaluate the fitness of the current position at discrete time intervals owing to the computation load of evaluating complex fitness functions.
- It may make sense for real creatures to all settle in one 'spot' if, for example, this 'spot' is especially advantageous in terms of food supply. However, with the PSO, once the fitness of any position visited by any particle is known, it is in fact inefficient for any swarm member to return to that position (at least for episodic problems). In particular, it would be desirable to discourage, at least partially, the tendency for the whole swarm to congregate around any currently 'good' position. Ideally, a part of the swarm should explore in the immediate vicinity, lest there be an even better position close by, while the rest of the swarm should set out to explore new areas.

It is possible to use the discrete evaluation to alter the behaviour of PSO when settling so as to achieve partial settling and partial aggressive further exploration.

2 The Particle Swarm Optimisation Algorithm

The Particle Swarm Optimisation algorithm uses a population of particles (the swarm) each of which describes a possible solution to the process to be optimised. Each particle moves through problem space so that that the set of solutions represented by the particles continuously changes. The movement of each particle is influenced by the current history of all other particles in the swarm together with some past history of the performance of the swarm as a whole.

Formally, the new velocity of a particle at time $T+t$ is $\overline{V}_{T+t}$ and is given by:

$$\overline{V}_{T+t} = \chi(M\overline{V}_T + (1-M)(rand \cdot P(\frac{\overline{X}-\overline{B}}{t}) + rand \cdot G(\frac{\overline{X}-\overline{S}}{t})))$$

where $\overline{V}_T$ is the velocity of this particle at time T, M is the momentum, $\overline{X}$ is the current position of the individual, *rand* returns a random number in the range from 0 to 1, and P and G set the relative attention to be placed on the positions $\overline{B}$ and $\overline{S}$. $\overline{B}$ is the best position found by any individual in the swarm so far, and $\overline{S}$ a position derived by comparing the relative performance of this individual with the performances of a number of other swarm members. Both M and G are bounded to the range (0, 1). The factor χ provides velocity constriction. For more information see [1,2,3,4].

Traditionally finding the position $\overline{S}$ involves defining the neighbourhood of the particle and only considering the effect of other particles within that neighbourhood. An alternate way of calculating the position $\overline{S}$ in an effective (but computationally modest) way that takes account of the performance of all swarm members has been introduced in [5]. All other swarm members are considered to influence the position $\overline{S}$ for a given particle but the magnitude of the influence decreases with both fitness and distance from the given particle for a minimization problem[1].

The parameter t is the time between updates and is required for dimensional consistency. It is usually taken to be one basic time interval and so is often omitted from the equation. This paper is concerned with an effect of the finite value used for t.

3 Introducing a Short-range Force

In order to alter the behaviour of particles when they are close together, it is desirable to introduce a short-range interaction between particles. One possibility is to introduce a gravitational style attraction that produces a force of attraction of particle i towards particle j, whose magnitude is inversely proportional to the square of the distance between them.

This short-range force will produce a velocity component on particle towards the other that can be represented by $v_{ij} = \frac{K}{d_{ij}^2}$, where d_{ij} is the distance between the particles and K is a constant. If evaluation were continuous for PSO this would merely cause close particles to enter stable limit cycles, as opposed to what we wish to achieve. However, for non-continuous evaluation this short-range force can have a useful effect.

---

[1] For a maximization problem it would increase with fitness but decrease with distance.

4 The effect of discrete evaluation

An effect of discrete evaluation on the traditional PSO algorithm is that particles may jump directly over places of interest but not notice them, unless the transit happens to coincide with a fitness evaluation. The actual distance travelled between evaluations depends on the velocity of the particle and it is important that particles slow in regions containing a fitness peak. The exploitation component of basic PSO tends to do this as a consequence of drawing particles that have 'overflown' back towards a good position.

The short-range force introduced in 3 above will have little effect while the swarm is dispersed for either continuous or non-continuous evaluation. However, as particles approach each other the magnitude of the short-range force will climb significantly, producing a substantial increase in the velocity of the particles towards each other. For discrete evaluation, *by the time of the next evaluation***,** particles may have passed each other and be at such a distance from each other that the short-range attraction that could bring them back together is far too weak to do this. As a result, the particles will continue to move rapidly apart with almost undiminished velocity, exploring beyond their previous positions. This process is shown diagrammatically in figure 1.

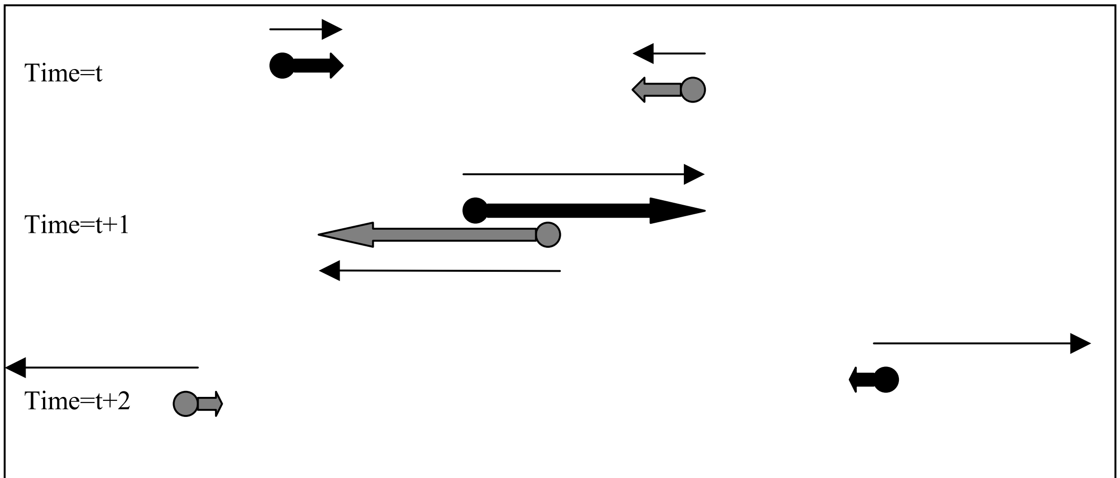

Figure 1. The short-range forces (block arrows) and the velocities (normal arrows) these forces induce on two approaching particles. The length of each arrow indicates the magnitude of the corresponding force or velocity. Note that by time *t+2* the short-range force is again small but the particles have kept much of the velocity calculated at time *t+1*.

At time *t,* the separation between the particles is moving into the region in which the magnitude of the short-range attraction (shown by broad arrows) is becoming significant. This augments the effect of their velocities (shown by normal arrows) so that the particles move close together. By time *t+1* the particles are close and the short-range effect is large. As a result, the velocity of the particles increases substantially, almost entirely as a consequence of the short-range attraction. Shortly after time *t+1* the particle will pass each other, but it is not until time *t+2* that the next evaluation is made. By this time the particles are far apart and the short-range force is weak. Consequently, the particles continue to diverge, retaining at *t+2* much of the velocity obtained as a result of the short-range forces acting at time *t+1*. The short-range forces will continues to decrease as the particles move apart, leaving only the normal swarm factors to influence their future movement in the absence of other close encounters.

If continuous[2] evaluation was used, as soon as the particles pass each other the direction of the short-range force would reverse and the still high values of short-range attraction would rapidly nullify the effects of the acceleration as the particles converged. As far as the effect of the short-range force is concerned, after the particles have passed and separated they would be first slowed, then come to rest before once again converging. The net effect of the short-range force under continuous evaluation would be for the particles to eventually enter a stable limit cycle.

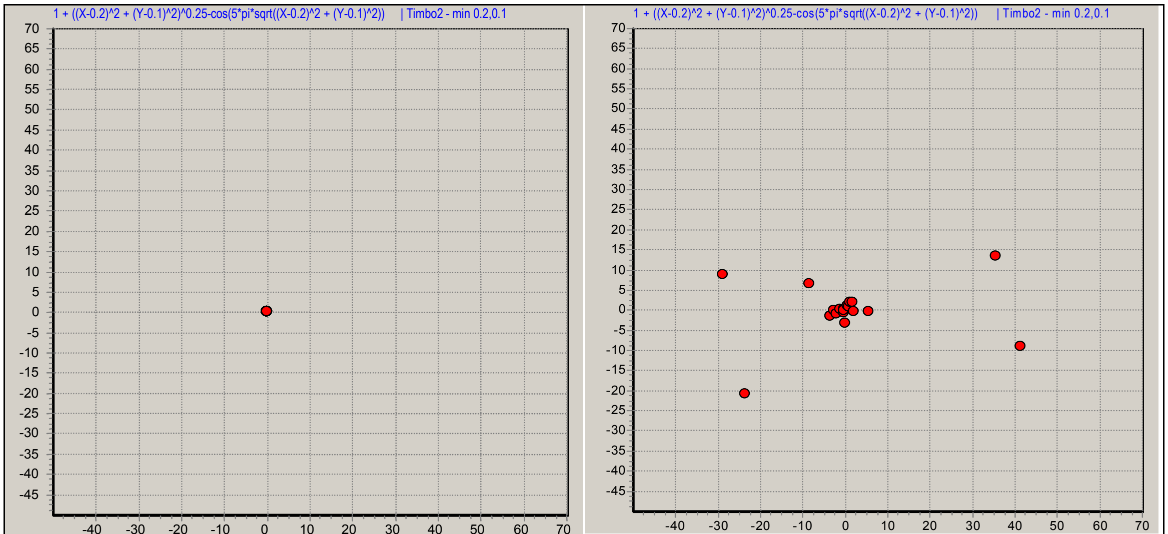

Figure 2. The distribution of twenty particles after approximately the same number of evaluations. At the left PSO without a short-range force, at the right PSO with a short-range force.

Under discrete evaluation conditions, we are able to take advantage of an aliasing effect in which conservation of energy appears to have been violated with a convenient spontaneous increase in the net energy of the two particles[3]. This increase in energy of the two particles may or may not be a good thing as far as any two particles are concerned. They might just be happening to be passing each other closely in region of poor fitness rather than converging to a fit point. In this case the short-range force may just act as a local perturbation of unpredictable benefit. But when a significant part of a swarm is tending to settle on a point of good fitness, the short-range force will have the desirable effect that some of the members so engaged will be ejected with significant velocities. As members get ejected the number of those left to cluster is decreased with a decrease in the probability of further ejections. Hence the effect of the short-range force

[2] Truly continuous evaluation is obviously impossible for any computer based PSO algorithm. Here the meaning of the word 'continuous' has to be expanded to include any situation in which the interval between evaluations is sufficiently short so that no particle has moved a significant distance between evaluations.

[3] Both authors, having a physics background, first found this strangely disconcerting but now realise that, if you adapt a real life algorithm by the addition of one or more non-physical features (in this case discrete evaluation) it is unreasonable to expect that the behaviour of the algorithm will continue to match real life behaviour.

on the normal settling behaviour of PSO is self-limiting and some swarm particles are left to continue exploration in the local vicinity.

Figure 2 shows the results of using a particle swarm optimisation algorithm to find the minimum of a function. This minimum occurs at the position (0.2,0.1). In the left hand picture, without a short-range force, all the swarm individuals have converged very close to the minimum. In the right hand picture, with a short-range-force and after approximately the same number of evaluations, the minimum has again been found. This time, while at any moment a number of swarm members remain searching in the immediate vicinity of the minimum, others are being ejected to continue searching elsewhere. In this case, as there is no better minimum to be found, these ejected particles subsequently return to the known minimum, increasing the probability of others being ejected. This production of successive waves of explorer particles continues indefinitely.

5 Conclusion

It must be stressed that this short-range force only has a scattering effect when particles are very close. The normal swarm algorithm is simultaneously operating with this short-range effect only having significant effect during the settling stage of a swarm. At other times the short-range force only adds what is effectively a small amount of noise to the normal behaviour of the swarm.

Preliminary experiments so far have used the simple formula for the short-range force given in this paper, but the power to which the particle separation distance is raised could be changed so as to make this attraction even more short-ranged. In these experiments, the short-range force between each particle and all others in the swarm is calculated - again in order to negate the need for a user defined neighbourhood - although in practice the number of occasions on which there is more than one significant contribution are few. These experiments have not shown any observable change in the behaviour of the PSO except for the modification to behaviour during severe convergence phases. When applied to problems with multiple local maxima, the addition of the short-range force improves the probability of the swarm finding the global maximum rather than converging on a good but suboptimal position. The additional computational cost of the behaviour modification is small when compared to the overall computation involved, especially for problems with complex fitness functions.

6 References

1 Kennedy, J and Eberhart R.C. *"Particle Swarm Optimization"*, Proc. IEEE International Conference on Neural Networks, Perth Australia, IEEE Service Centre, Piscataway NJ USA IV:1942-1948. (1995)

2 Eberhart. R. C., Dobbins, P., Simpson, P. *Computational Intelligence PC Tools,* Academic Press, Boston. (1996).

3 Corne D, Dorigo M and Glover F (Editors), *The Particle Swarm: Social Adaptation in Information-Processing Systems.* Chapter 25 in *New Ideas in Optimization.* McGraw-Hill Publishing Company, England ISBN 007 709506 5 (1999).

4 Eberhart, R. and Shi, Y. "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimisation", Proceedings of the 2000 Congress on Evolutionary Computation, pp. 84-88. (2000)

5 Hendtlass T "A combined swarm differential evolution algorithm for optimization problems". Lecture Notes in Artificial Intelligence, Vol 2070 pages 374-382 Springer, Berlin. (2001)