TJHSST Senior Research Project
Particle Swarm Optimization and Social Interaction Between Agents
2007-2008

Kenneth Lee

January 24, 2008

Abstract

Particle Swarm Optimization is a method of optimization for n-dimensional infinte search spaces. This project aims to test different social influences, the way in which the particles communicate with eachother in order to find a global minimum, on the particles and their ability to converge on a correct solution. The different versions of the social interactions are tested using various benchmark functions and then the different methods are compared to eachother.

Keywords: Particle Swarm Optimization, Fully Informed Particle Swarm, Social Interaction.

1 Introduction - Purpose and Scope

Particle Swarm Optimization(PSO) is a technique used to optimize n-dimensional infinite search space problems. A large amount of particles exist in the search space and "fly" though it searching for the global minimum. Particles are influenced by both cognitive and social interactions changing their course of "flight". This project aims to alter the social interactions in order to increase the effeciency of the swarm.

If a way of social interaction is proven to be more effecient than the others, then it can replace the older method, and thus make the algorithm more powerful. This could also lead to other improvements increasing the quality of algorithm.

Anyone who has a problem in which a optimal result needs to be found quickly can use this algorithm. It has seen extensive use in neural networks and finds potential in time-critical optimization problems where the constraints quickly change.

This project will deal only with the social interactions between agents. It will not deal with inertia or cognitive influences and they will remain constant thoughout the program. It should also be noted that thusfar in the project, the only form of comparing the different influences will be by iteration count not by actual time to run the program.

2 Background

PSO is a relatively new swarm intelligence technique. It was first created in 1995, inspired from flocks of birds and schools of fish.[5] As the technique progressed, however, various modifications were made in order to improve both reliability and time cost. It was not until the work of Clerc that the technique was truely analyzed from a algebric point of view. In his work, Clerc describes a simple one-dimensional, one-particle, Particle Swarm and then proceeded to rebuild the swarm back to its originial form.[1] PSO is used for n-dimensional optimization problems, because it is relatively easy to implement. A set of particles is randomly created in the search space. Each particle is given a random velocity to move about the search space. Its velocity can be adjusted during the run by both cognitive and social interactions. The congnitive interactions involve the particle remembering where it had the lowest fitness value, the particle's velocity is adjusted to move towards this point. The social influences adjust the particle's velocity based upon other particles (one or more). [2]

Though different types of social interactions have been tested in the past, the conculsions have not been conclusive.[3] This could be in part due to the so-called No Free-Lunch Theorem, which states that because there are so many numerous testing types that if all possible tests were performed over any algorithm, they would all be equivalent.[4] However, since not all functions are being tested, but rather only a small subset wherein PSO has application, it is unclear as to whether the No Free-Lunch theorem will hold or not.

3 Social Interactions

The social interactions currently covered by this project include:

1. Non-Informed Particle Swarm(NIPS)
2. Singly Informed Particle Swarm(SIPS)
3. Fully Informed Particle Swarm(FIPS)

The section below will describe the different forms of social interaction and how they are implemented and act on the swarm.

3.1 Non-Informed Particle Swarm

NIPS works on a very basic principle that the particles do not in any way associate with each other. The only method by which the particles velocites are adjusted in any meaninful way is by congnitive means. The particles velocity is updated by the following method:

$$\overrightarrow{v_{t+1}} = \alpha \overrightarrow{v_t} + \varphi(\overrightarrow{P_i} - \overrightarrow{x_t}) \tag{1}$$

$$\overrightarrow{x_{t+1}} = \overrightarrow{x_t} + \overrightarrow{v_{t+1}} \tag{2}$$

For each particle, the velocity, $\overrightarrow{v_{t+1}}$, and position, $\overrightarrow{x_{t+1}}$ vectors are updated. $\overrightarrow{P_i}$ is representive of the particles best previous value, the pbest. φ is a random number between 0 and 1.0.

3.2 Singly-Informed Particle Swarm

SIPS is the basic version of PSO. Simply put, the swarm finds the particle with the lowest fitness value, and all the other particles are drawn towards it. The mathematical view is very similar, but only slightly differs in the adjustment of the velocity.

$$\overrightarrow{v_{t+1}} = \alpha \overrightarrow{v_t} + \varphi_1(\overrightarrow{P_i} - \overrightarrow{x_t}) + \varphi_2(\overrightarrow{P_g} - \overrightarrow{x_t}) \tag{3}$$

$$\overrightarrow{x_{t+1}} = \overrightarrow{x_t} + \overrightarrow{v_{t+1}} \tag{4}$$

In this method of swarm, ϕ_1 and ϕ_2 are between 0 and 1.0. The particle has an equal amount of influence between the swarm's best and the particle's own best position.

3.3 Fully-Informed Particle Swarm

The last swarm being tested by this project thus far is the FIPS. This particle swarm is slightly different than the others. Instead of the cannonical version of velocity testing (as seen in SIPS). The particles find the best point, collectively and then move towards that point.

$$\overrightarrow{P_m} = \frac{\sum_{k \in N} W(k)\varphi \bigotimes P_k}{\sum_{k \in N} W(k)\varphi} \tag{5}$$

$$\overrightarrow{v_{t+1}} = \alpha \overrightarrow{v_t} + \varphi(\overrightarrow{P_m} - \overrightarrow{x_t}) \tag{6}$$

The best point found by the swarm, $\overrightarrow{P_m}$, is found by adding all particle's position vector multiplied by W(k), a weighting factor of some sort. What is actually contained in W(k) does not matter to a large degree as it is averaged out over the entire swarm. The weighting factor only determines to what extent the particle is influential in the swarm [3]

4 Procedures

Procedure and Methodology

The first step for this project was to correctly recreate the basic PSO for a simple situation. This basic PSO had included a method for the social interaction between agents, more specifically a Singly Influenced Particle Swarm(SIPS). After the cannonical method was produced and tested to some extent, and the social interaction of the particles was made modular, other methods of social interaction were introduced into the program. More specifically, those interactions are the Fully Informed Particle Swarm(FIPS) and the No Influence Particle Swarm(NIPS).

4.1 Testing

For this project, it would not be very possible to use a mathematical formula to judge performance for the swarm, due to the fact that a great part of the algorithm(including starting position and velocity) are derived randomly. Therefore, the program will be tested by running the program multiple times for each social interaction and function tested, and from it determining the average running time and number of time steps needed for the swarm to converge on the correct answer, if the swarm indeeds converges.

4.2 Software

For this project C was used for coding purposes. In addition, the OpenGL library was used in order to graphically depict 2 dimensions of the benchmark function tested.

5 Expected Results

Expected Results and Value to Others

I expect that the FIPS will fare better than some of the other types of interaction tested, but it will not be the most effecient in terms of time due to the n^2 addition required for that method. Overall, I believe that the Single Influenced Particle Swarm will do the best overall because of its simplicity and robustness. The NIPS will do the worst due to the tendancy of its particles to reach and maintain at local extrema.

References

[1] Maurice Clerc and James Kennedy. The particle swarm: Explosion stability, and convergence in a multi-dimensional complex space. *IEEE Trans. Evolutionary Computation*, 6:58–73, 2002.

[2] Hirotaka Yoshida et al. A particle swarm optimization for reactive power and voltage control considering voltage sability. 1999.

[3] Rui Mendes, James Kennedy, and José Neves. The fully informed particle swarm: Simpler, maybe better. *IEEE Trans. Evolutionary Computation*, 8(3):204–210, 2004.

[4] Frans van den Bergh. *An Analysis of Particle Swarm Optimizers*. PhD thesis, University of Pretoria, November 2001.

[5] Zhi-Lian Yang Xiao-Feng Xie, Wen Jun Zhang. A dissipative particle swarm optimization. Hawaii, USA, 2002. Institute of Microelectronics, Tsinghua University, Beijing.