Particle Swarm Optimization and Social Interactions Between Agents

TJHSST Systems Lab 2007-2008
Kenneth Lee

Abstract

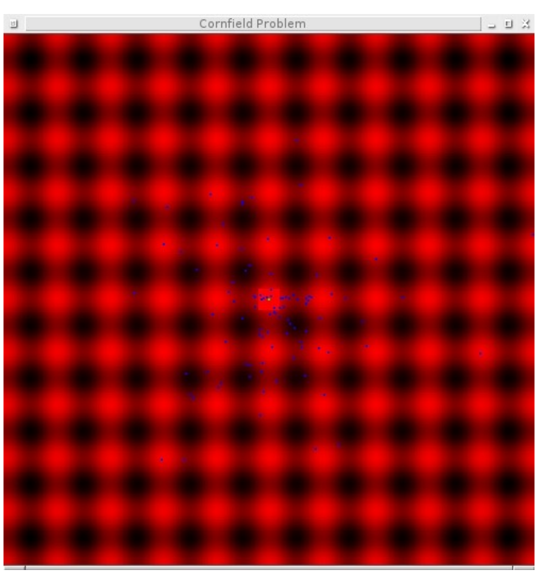
Particle Swarm Optimization is a method of optimization used in n-dimensional infinite search space problems. This project aims to test different social influences and topologies, the way in which the particles communicate with each other in order to find a global minimum, of the particles and determine their ability to converge on a correct solution as opposed to the more common social interaction seen in PSO. The different versions of the social interactions are tested against each other using various benchmark functions based upon iterative cost to run the swarm.

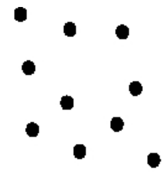
Background

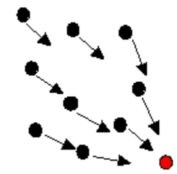
PSO is a relatively new swarm intelligence technique. It was first created in 1995, inspired from flocks of birds and schools of fish. It is considered a good technique because it is both inexpensive in time and in memory.
PSO is used for n-dimensional optimization problems, because it is relatively easy to implement.

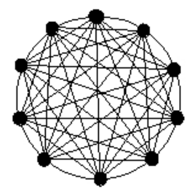
A set of particles is randomly created in the search space. Each particle is given a random velocity to move about the search space. Its velocity can be adjusted during the run by both cognitive and social interactions. The cognitive interactions involve the particle remembering where it had the highest fitness value, and wanting to return there.

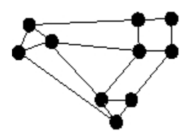
The social influences are where the particle in influenced based on the other particles, either by their current position and fitness value or their personal best (pbest) fitness value.

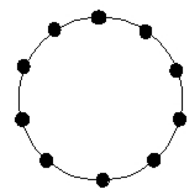



No influence


Single influence


Fully Informed Particle Swarm


Neighborhood influence


Circular Influence

Results and Conclusions

The results of the experiment so far are that the effectiveness of a swarm is very dependent on the topology of the swarm as they relate to the Rastrigin function. For instance, though the iterative cost of FIPS is relatively low compared to the iterative cost of RIPS. On the other hand, RIPS is substantially more accurate in determining the correct solution. The answer, as it stands now, seems to be an attempt to bridge the gap between RIPS and FIPS, the low k and high k values. This is seen in DIPS, which while being extremely accurate has one-forth the iterative cost of RIPS.

Though research is not yet complete in this field, it does seem to show promise in continuing to optimize PSO for all real optimization problems. With the addition of more benchmark functions in the proceeding paper, hopefully the results will be more conclusive then at the current junction.