Cooperative Particle Swarm Optimization Developed with OpenMP

Lan Guoxing

Thursday , September 11, 2014

# 1 Introduction

Particle Swarm Optimization (PSO) is a stochastic and population-based algorithm that can be applied to nonlinear optimization problems. The standard PSO performs well when the problem is low-dimensional. However, when the dimensionality of the search space increases, the performance tends to deteriorate. The algorithm slows down when it generates solutions near surrounding the global optimum. The probability of generating a sample inside the optimality region decreases exponentially as the dimensionality of the search space increases. To solve this problem, an improved PSO called Cooperative PSO (CPSO) was proposed. Its idea is to partition the search space into lower dimensional subspaces, and the simplest way is to divide the particle swarm into small swarms and let each swarm search one dimensionality.

# 2 Standard Particle Swarm Optimization

PSO has two primary operators: Velocity update and Position update. They are expressed as below:



The definitions and variables are as follows:

*  means the current generation step
*  is the velocity of individual 
*  is the position of individual 
*  are random numbers drawn from a uniform distribution between 0.0 and 1.0
*  is the inertia weight
*  is the social parameter 1, usually set to 2.0
*  is the social parameter 2, usually set to 2.0

*  is the i-th individual’s best state (position) found so far
*  is the global best state (position) found so far
*  is a constraint factor, usually set to 1.0

Create and initialize an n-dimensional PSO 

**Repeat:**

**For** each particle:









**End for**

Perform PSO updates on P using eqns (1) and (2)

**Until** stopping criterion is met

Fig.1. Pseudocode for the PSO algorithm

Fig 1 lists the Pseudocode for the standard PSO. For a particle  in a swarm: ,  and  stands for the position, personal best position and global best position.

# 3 Cooperative PSO

Define



Initialize n-dimensional PSOs

**Repeat:**

**For** each swarm:

**For** each particle:









**End for**

Perform PSO updates on  using eqns (1) and (2)

**Until** stopping criterion is met

Fig.2. Pseudocode for the CPSO algorithm

The standard PSO uses a population of n-dimensional vectors. The idea of the Cooperative PSO is to partition these vectors into n swarms of 1-dimensional vectors, each swarm representing one dimensionality of the original problem. Therefore, the n-dimensional problem can be simplified into n 1-dimensional problems, and each swarm attempts to optimize a single component of the problem.

Fig 2 lists the Pseudocode for the Cooperative PSO. Now  refers to the position of particle  in swarm; each of the n swarms has a global best particle; the function  returns an n-dimensional vector formed by concatenating all the global best vectors across all swarms, except for the j-th component, which is replaced with , where  represents the position of any particle from swarm .

# 4 Comparison of Standard PSO and Cooperative PSO.

We choose some high-dimensional problems with different sizes of feasible space to compare the performance of standard PSO and Cooperative PSO in terms of accuracy and simulation time. The Cooperative PSO code is developed using OpenMP on a 4-core computer. For each problem, we test 10 times and the program runs 30 times in each tests in order to avoid accidental jamming.

The problems and results are as follows (SPSO refers to standard PSO while CPSO refers to cooperative PSO):

T1：



Results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Problem 1 | | | | | |
| Methods | Best Result | Mean Result | Worst Result | Optimal | Time（mS） |
| SPSO | 0.000005 | 0.0000123 | 0.000028 | 0 | 3867 |
| CPSO | 0 | 0.0000005 | 0.000003 | 0 | 3659 |

Population size of particles in SPSO: 30 Iterations: 500

Population size of particles in CPSO: 60 Iterations: 500

T2：



Results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Problem 1 | | | | | |
| Methods | Best Result | Mean Result | Worst Result | Optimal | Time（mS） |
| SPSO | 0.001529 | 1.476421 | 1.476421 | 0 | 6242 |
| CPSO | 0.007099 | 0.2394223 | 0.874389 | 0 | 5316 |

Population size of particles in SPSO: 50 Iterations: 500

Population size of particles in CPSO: 90 Iterations: 500

T3：



Results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Problem 1 | | | | | |
| Methods | Best Result | Mean Result | Worst Result | Optimal | Time（mS） |
| SPSO | 1.314288 | 5.6372101 | 12.007295 | 0 | 78526 |
| CPSO | 0.103738 | 1.0954721 | 2.986288 | 0 | 78899 |

Population size of particles in SPSO: 90 Iterations: 1000

Population size of particles in CPSO: 120 Iterations: 1000

# 5 Conclusion

By analyzing the results above, we can draw this conclusion: Cooperative PSO usually has a higher accuracy than standard PSO, especially the dimensionality of the problem is very high and the feasible search space is very large. Since each dimensionality needs one swarm to optimize, the total number of the particles of CPSO is larger than SPSO. However, the simulation time of CPSO developed in OpenMP is often shorter, because it can take advantage of the multi-core processor. When the dimensionality is too large, CPSO might be slower than SPSO because the increase of dimensionality causes the increase of particles which resulting the rapid rise of time cost.