Evaporation as a self-adaptation mechanism for PSO

Jose Luis Fernandez-Marquez and Josep Lluis Arcos
IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council
Campus UAB, Bellaterra, E-08193 (SPAIN)
{fernandez,arcos}@iiia.csic.es

Abstract

This work extends the Particle Swarm Optimization (PSO) algorithm for working on dynamic environments. We propose an evaporation mechanism to solve the outdated memory problem. We empirically show that our evaporation mechanism is able to achieve self-adaption without any knowledge on when changes occur.

1 Introduction

The original version of Particle Swarm Optimization (PSO) was proposed by J. Kennedy and R. Eberhart in 1995. PSO was aimed to produce a collaborative intelligence behavior by borrowing the analogy of social interaction. In PSO a number of particles is placed in a search space, each particle trying to reach the optimal position. The movements of each particle are based on the combination of a cognitive and a social model. The cognitive model drives each particle to its best found position. The social model drives each particle to the best position found by particles belonging to its neighborhood.

Since 2001, when Eberhart and Shi proposed the original PSO for solving dynamic optimization problems, different authors have proposed extensions to the original PSO algorithm, such as reseting the particles position frequently or using a multi-swarm model [9], for improving its adaptiveness in dynamic environments.

In order to adapt the PSO for working on dynamic environments, two main problems have been identified: diversity loss (due to particles' convergence) and outdated memory (due to the environment dynamism)[1]. Diversity loss has been addressed either by introducing randomization, repulsion, dynamic networks, or multi-populations [1]. Repulsion leads the particles to spread over the search space allowing a broader exploration of the search space. Nevertheless, the time to reach the optimum increases, and sometimes is not reached, due to the excess of exploration. For instance, RPSO [8] proposes a repulsive social model as a way to find the optimum in very complex *static* optimization problems.

The outdated memory problem has been tackled by setting current particle positions as their best positions or by re-evaluating best positions to detect the changes (increasing the computation cost). Most of these existing approaches assume that either the changes are known in advance by the algorithm or that they can be easily detected (hypothesis not feasible in many real problems).

In this paper we propose a new method, inspired on the ant pheromone evaporation mechanism [6], able to work on dynamic environments without any knowledge about when changes will occur. Using the Moving Peaks Benchmark (MPB) [4], we will show how our approach, together with a diversity lost mechanism, presents good performance results.

2 The evaporation mechanism

We propose a hybrid PSO-RPSO algorithm to deal with the diversity lost problem and an evaporation mechanism to deal with the outdated memory problem. The main motivation for the hybrid algorithm is to better control the distribution of the particles: RPSO particles will mainly present an exploration behavior whereas PSO particles will be more focused on an exploitation behavior. Using this hybrid approach, some particles are collaborating each other to reach the optimal value whereas other particles are mainly exploring possible new local optimums (i.e. maintaining the diversity). A similar behavior has been achieved by Charge Particle Swarm Optimization (CPSO) [2] and Quantum Swarm Optimization QSO [3]. Nevertheless, CPSO presents a quadratic complexity and QSO uses a randomization operator to spread explorer particles around the peaks.

For solving the outdated memory problem when the changes are not known in advance, we hypothesize that providing a mechanism for continuously forgetting is better than a periodical resetting approach. Moreover, a continuous mechanism avoids the assumption that changes can be predicted in some way.

We propose an evaporation mechanism for reducing the fitness value of the best position found by each particle along time. This mechanism will penalize optimums that were visited a long time ago. Thus, evaporation provides an automatic dissipation mechanism over the information taking into account the adquisition time. The idea of evaporation is not new. Ant Systems use evaporation in pheromone trails as a mechanism to achieve a signal degradation [6].

We incorporated an evaporation factor ν on the best position found by each particle. Specifically, at each particle iteration, when the fitness value of the current position is not better than the fitness value of the best position stored in its memory, the fitness value of the memory best position is decreased by the evaporation factor ν. This evaporation factor will affect the cognitive model of each particle, providing a dynamic tradeoff between the cognitive and the social information.

3 Experiments

We used Moving Peaks Benchmark (MPB) [4] for comparing our evaporation mechanism with an *informed* mechanism where the memory of the particles is reseted to their current position whenever a change occurred. MPB is a benchmark to compare dynamic optimization algorithms by modeling problems less complex than the real world but more complex that a simple simulation. MPB allows to generate search spaces that change over time (in the height, width and location of peaks).

For experimentation, a 4-dimensional search space with dimension ranges from 0 to 100 was generated. Since we were interested in analyzing the outdated memory problem, the experiments were performed with a single peak. A run consisted of 100 peak changes. The peak changed its height (height $\in$ [30, 70]), width (width $\in$ [0.001, 0.1]), and position every 5000 steps. Results are based on averages over 30 runs with uncorrelated peak changes at different distances (peak shift $\in$ [1, 6]). The evaporation factor selected in the experiments was $\nu = 0.7$.

We focused the analysis of the results on the *offline error* (see Table 1). The offline error measures the performance of an algorithm when tracking environment changes (where an error equal to zero indicates the perfect tracking). Offline error is calculated as the error average of all the best solutions found since the last peak change.

Peak shift	Evaporation	Informed
1	0.1682 ± 0.0180	0.1865 ± 0.0141
2	0.2878 ± 0.0189	0.3316 ± 0.0167
3	0.4258 ± 0.0228	0.5000 ± 0.0247
4	0.5996 ± 0.0299	0.6666 ± 0.0314
5	0.8246 ± 0.0387	0.8686 ± 0.0386
6	1.0937 ± 0.0442	1.0938 ± 0.0493

Table 1. Offline Error ± Std. Dev.

Resuts in Table 1 show that the evaporation mechanism was able to maintain a performance equivalent to the informed mechanism (with a memory initialization after each peak change). Thus, the evaporation factor provides a self-adaptation mechanism that allow particles to autonomously track the changes in the environment.

From these initial encouraging results, we are now designing more exhaustive experiments for analyzing the impact of different evaporation values and for comparing our hybrid PSO-RPSO algorithm with the existing literature.

References

[1] T. Blackwell. Particle swarm optimization in dynamic environments. *Evolutionary computation in dynamic environments*, pages 29–49, 2007.

[2] T. Blackwell and P. Bentley. Dynamic search with charged swarms. *Proc. Genetic Evol. Comput.*, pages 19–26, 2002.

[3] T. Blackwell and J. Branke. Multiswarm, exclusion, and anti-convergence in dynamic environments. *IEEE Transactions on Evolutionary Comp.*, 10(4), 2005.

[4] J. Branke. The moving peaks benchmark website. www.aifb.uni-karlsruhe.de/ jbr/movpeaks/.

[5] R. C. Eberhart and Y. Shi. *Swarm Optimization*. Morgan Kaufmann, 2001.

[6] S. Garnier, J. Gautrais, and G. Theraulaz. The biological principles of swarm intelligence. *Swarm Intelligence*, 1:3–31, 2007.

[7] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. IEEE Int. Conf. Neural Networks*, pages 1942–1948, 1995.

[8] S. K. Mishra. Repulsive particle swarm method on some difficult test problems of flobal optimization. 2006.

[9] R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. an overview. *Swarm Intelligence*, 1:33–57, 2007.