Applied Soft Computing 48 (2016) 584–596

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Particle swarm optimization using dynamic tournament topology

Lin Wang [a,b], Bo Yang [a,*], Jeff Orchard [b]

[a] Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, Jinan 250022, China
[b] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

ARTICLE INFO

Article history:
Received 19 July 2014
Received in revised form 1 July 2016
Accepted 25 July 2016
Available online 1 August 2016

Keywords:
Particle swarm optimization
Dynamic tournament topology
Evolutionary computation
Artificial neural network

ABSTRACT

Particle swarm optimization (PSO) is a nature-inspired global optimization method that uses interaction between particles to find the optimal solution in a complex search space. The swarm's evolving solution is represented by the *best* solution found by any particle. However, using this *best* solution often limits the search area. In this paper, we propose a dynamic tournament topology strategy to improve PSO. In our method, each particle is guided by several *better* solutions, chosen from the entire population. The selection of the *better* particles is stochastic, but still favors particles with better solutions. Experimental results on benchmark functions indicate that the proposed method is promising. Furthermore, the application of our dynamic tournament topology strategy in optimization of artificial neural networks indicates that this method has favorable performance.

1. Introduction

Optimization is a fundamental challenge in many real-world problems. Several optimization algorithms, such as Powell's method [1], have been examined to solve optimization problems. However, traditional algorithms have difficulty determining a satisfactory solution when dealing with highly complex real-world problems (multimodal, high dimensional, and noisy). Nature-inspired methods have been proposed to solve problems that are hard for traditional methods [2–6]. As one example (inspired by the foraging of birds), particle swarm optimization (PSO) [2] yields impressive results. The algorithm has been proven to be a practical optimization tool with several success stories in various applications, such as communication, finance, energy, medicine, materials science, and remote sensing [7–13].

This study proposes a dynamic tournament topology strategy to improve PSO (DTT-PSO). Instead of the *gbest*, *lbest*, or other *bests*, a tournament strategy is introduced to choose several above-average (*better*) guides from the population. Each potential individual has a chance to inform the others. The vast majority of particles has the opportunity to be picked as a guide to inform others. The method also incorporates merits from random topology and fully informed strategy. Each particle simultaneously receives information from all *better* guides. Furthermore, particle guides continuously change with evolution. The method increases population diversity and decreases the probability of dropping into local optima.

The rest of this paper is organized as follows. Section 2 reviews the basics, improvements, applications, and challenges of PSO. Section 3 defines the benchmark functions. Section 4 describes the details of the dynamic tournament topology strategy. Section 5 outlines and discusses the experimental results and shows its application to optimizing neural networks. Finally, Section 6 provides the conclusion.

2. Motivation

The solution of a problem in PSO is represented as a *particle* in the solution space. Each particle moves throughout the solution space following *two bests*: one that records the particle's personal best solution (denoted *pbest*), and another that records the global best solution (denoted *gbest*). However, particle or swarm historical memory in traditional algorithms is represented by the "*best*" position found by the particle itself or a group of particles. Suppose a particle is in the basin of attraction of a local optimum that is not the global optimum. If this particle happens to exhibit the best fitness of all the points, then it will enlist the others to follow it. However, this will mislead the swarm into a local optimum, and possibly away from the global optimum.

This problem gives rise to the following question: *can we replace the best position with better positions to further improve the performance of PSO by thoroughly searching the potential optimal regions?*

This study proposes a dynamic tournament topology strategy. No concept of *gbest* or *lbest* exists in DTT-PSO. Instead, we propose

* Corresponding author.
E-mail address: yangbo@ujn.edu.cn (B. Yang).

http://dx.doi.org/10.1016/j.asoc.2016.07.041
1568-4946/

the use of a tournament strategy to distribute the guidance of the swarm more evenly. Every particle (except for the worst ones) can potentially become a *better* guide. The method also adopts advantages from random topology [14] and the fully informed strategy [15]. Each particle simultaneously receives information from all *better* guides and changes its guides frequently during evolution.

On one hand, this strategy avoids the myopic tendency to follow a single global best particle [2]. Rather, it selects several better guides randomly to enable a more diversified search. The randomness of the tournament topology ensures that the best particle *tends* to inform the others, but not always. On the other hand, our method adopts a dynamic topology in which the neighborhood of a particle frequently changes according to the tournament strategy, similar to local PSO [16]. This dynamic random topology ensures a wide exchange rather than a directional flow of information between particles. The DTT strategy helps pull the PSO out of local optima, and toward the global optimum. The next subsection describes the DTT-PSO in detail.

3. Review of particle swarm optimization

3.1. Basics of PSO

PSO is a nature-inspired global optimization method designed by Kennedy et al. [2] in 1995. The method can determine the optimum area in a complex solution space by the interaction between particles. Compared with the genetic algorithm (a classical nature-inspired optimization algorithm), no selection, recombination, and mutation occurs in the PSO frame. This method views an individual in a population as a particle in the solution space, with zero mass and volume. These particles fly over the solution space at a given speed, updated according to the experience of the particles and the entire population. In traditional PSO, the term "experience" refers to the best position found by the particle (*pbest*) and by the entire population (*gbest*). All particles follow two *bests* to update their velocities and positions. Therefore, as a differential system [17,18], PSO involves two types of behaviors: cognitive and social. The formula to update a particle's velocity and position is

$$\begin{cases} v_{id}^d = v_{id}^d + \varphi_1 r(pbest_{id}^d - x_{id}^d) + \varphi_2 r(gbest^d - x_{id}^d), \\ x_{id}^d = x_{id}^d + v_{id}^d, \end{cases} \tag{1}$$

where v is the velocity vector, x is the particle's position in the solution space, d represents the dth-dimension, $pbest_{id}$ represents the best position found by the particle id, and $gbest$ represents the best position found by the entire population. The variable r is a random positive number drawn from the uniform distribution [0, 1]. φ_1 and φ_2 represent the acceleration constants. The PSO algorithm is shown in Algorithm 1.

Algorithm 1. Algorithm of particle swarm optimization

Input: Population size *PopSize*, and acceleration constants φ_1, φ_2
Output: The best solution.
Initialization;
while *termination condition has not been met* **do**
  Evaluate the fitness for each particle according to its position vector x;
  Update the best position *pbest* for each particle;
  Update the best position *gbest* for the whole population;
  for *id=1* ***to*** *PopSize* **do**
    for *d=1* ***to*** *Dim* **do**
      $v_{id}^d = v_{id}^d + \varphi_1 r(pbest_{id}^d - x_{id}^d) + \varphi_2 r(gbest_{id}^d - x_{id}^d)$;
      $v_{id}^d = \min(VMAX^d, \max(-VMAX^d, v_{id}^d))$;
      $x_{id}^d = x_{id}^d + v_{id}^d$;
    end
  end
end
Return the best position found by all of particles.

3.2. Applications and improvements

PSO is a practical approach with several success stories in many applications. Zhang et al. [19] proposed an improved adaptive particle swarm optimization to solve the reservoir operation problem. Bin et al. [20] proposed a novel binary particle swarm optimization to solve the haplotype inference by pure parsimony problem (HIPP). Davoodi et al. [21] proposed a new optimization approach, based on a hybrid algorithm combining improved quantum-behaved PSO and simplex algorithms for solving the power system load flow problem. Cervantes et al. [22] proposed an adaptive Michigan PSO and examined its application in nearest neighborhood classification. Wang et al. [23] proposed the use of nearest neighbor classification in conjunction with a neural-network classifier and uses PSO as the optimizer. Li et al. [24] suggested a novel fuzzy neural network, which adopts PSO to facilitate parameter estimation and improve accuracy, for system state forecasting. Chen et al. [25] proposed an online modeling algorithm for nonlinear and nonstationary systems with RBF neural network, in which the quantum PSO algorithm is introduced to optimize the tunable center vector and diagonal covariance matrix. Zhan et al. [26] used PSO for each population and developed a coevolutionary multiswarm PSO to solve multi-objective optimization problems. Lu et al. [27] investigated decision making and finite-time motion control for a group of robots, which introduced PSO to determine the probable position of the odor source. Wang et al. [28] distilled the middle-age hydration kinetics for cement hydration using phased hybrid evolution method and adopted PSO to search for the best value of coefficients for a given form of kinetics.

Despite the success of PSO in certain applications, traditional PSO still has room for improvements in updating velocity and topology structure. Several studies have focused on further improving the performance of PSO by integrating mechanisms from other algorithms, adding new strategies or techniques, and designing topologies. Mendes et al. [15] proposed a novel fully informed particle swarm (FIPS) that adopts a fully informed mechanism. The method allows every neighbor of a particle to contribute information for updating the velocity. Updating the velocity can be considered a cumulative effect of the information contributed by neighbors. This method increases the source of information for each particle, so it avoids dropping into the local optimum during evolution, and it improves the ability of converging and global searching for PSO. The traditional topologies in PSO are static; therefore, Liang and Siganthan [14] proposed a PSO method that adopts the dynamic topological strategy. This method divides the original population into several small swarms that regroup after every fixed number of iterations during the evolution process. On one hand, the division strategy ensures the diversity of the entire population. On the other hand, the regrouping strategy enables the exchange of information between small swarms. Liang et al. [29] also developed a comprehensive learning particle swarm optimizer (CLPSO), which has the distinction of independently optimizing the vector for each dimension. A particle receives information from the others with a certain probability and updates its velocity accordingly.

Leu et al. [30] proposed a grey evolutionary analysis to analyze the population distribution of particle swarm optimization during the evolutionary process. Lim and Isa [31] developed a teaching and peer-learning PSO algorithm by improving the cutting edge teaching-learning-based optimization algorithm and adapting the enhanced framework into the PSO. Calazan et al. [32] presented a novel massively parallel coprocessor for PSO's implementation using reconfigurable hardware. Zhan et al. [33] presented an adaptive PSO that consists of a real-time evolutionary state estimation and an elitist learning strategy. Valdez et al. [34] proposed a new