Applied Soft Computing 10 (2010) 119–124

Contents lists available at ScienceDirect

Applied Soft Computing

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

A perturbed particle swarm algorithm for numerical optimization

Zhao Xinchao

School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

ARTICLE INFO

Article history:
Received 13 February 2008
Received in revised form 16 May 2009
Accepted 28 June 2009
Available online 3 July 2009

Keywords:
Particle swarm optimization
Perturbed PSA
Particle updating strategy
Numerical optimization

ABSTRACT

The canonical particle swarm optimization (PSO) has its own disadvantages, such as the high speed of convergence which often implies a rapid loss of diversity during the optimization process, which inevitably leads to undesirable premature convergence. In order to overcome the disadvantage of PSO, a perturbed particle swarm algorithm (pPSA) is presented based on the new particle updating strategy which is based upon the concept of perturbed global best to deal with the problem of premature convergence and diversity maintenance within the swarm. A linear model and a random model together with the initial max–min model are provided to understand and analyze the uncertainty of perturbed particle updating strategy. pPSA is validated using 12 standard test functions. The preliminary results indicate that pPSO performs much better than PSO both in quality of solutions and robustness and comparable with GCPSO. The experiments confirm us that the perturbed particle updating strategy is an encouraging strategy for stochastic heuristic algorithms and the max–min model is a promising model on the concept of possibility measure.

1. Introduction

Particle swarm optimization (PSO) algorithm is a population-based heuristic global optimization technology introduced by Kennedy and Eberhart [1] in 1995. Its basic idea is based on the simulation of simplified animal social behaviors such as fish schooling, bird flocking, etc. Recently, Poli et al. [2] reviewed PSO algorithm, current and ongoing research, applications and open problems. In PSO algorithm, the individual is called particle which has no mass and volume, and the trajectory of each individual in the search space is adjusted by dynamically altering the velocity of each particle, according to its own flying experience and the flying experience of the other particles in the search space. The next iteration takes place after all particles have been moved. Eventually the swarm as a whole, like a flock of birds collectively foraging for food, is likely to move close to an optimum of the fitness function. PSO algorithm is becoming very popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution.

Eberhart and Shi [3] used a fuzzy system to adapt the inertia weight ω to significantly improve PSO performance. A fuzzy variable neighborhood particle swarm optimization [4] is introduced to represent the quadratic assignment problem with a fuzzy matrix. A novel fuzzy adaptive optimization strategy [5] is introduced to avoid falling into local optima based on double-variable and single-dimensional fuzzy control structure. Hu and Li [6] discard the particle velocity and reduces the basic PSO from the second order to the first order difference equation. The evolutionary process is only controlled by the variables of the particles position. Some well known algorithms [7–9] are hybridized with PSO and even better results are reported. Langdon and Poli [10] use evolutionary computation to automatically find problems which demonstrate the strength and weakness of modern search heuristics and illustrate the benefits and drawbacks of different population sizes, velocity limits, and constriction coefficients. A memetic algorithm [11] with a synchronous particle local search and a fuzzy global best for the updating of a particle trajectory is proposed for multi-objective optimization.

The high convergence speed of PSO often results in a rapid loss of diversity during the optimization process, which inevitably leads to undesirable premature convergence. A Guaranteed Convergence PSO (GCPSO) is discussed [12] and a separate velocity update formula is used for the best particle in the swarm. Shelokar et al. [13] proposed an improved PSO hybridized with ant colony approach which applied PSO for global optimization and the idea of ant colony approach to update positions of particles to attain rapidly the feasible solution space. In this paper a perturbed particle swarm algorithm (pPSA) is presented so as to escape from the local optimal trap. The new particle updating strategy is based upon the concept of possibility [14] to deal with the problem of maintaining diversity within the swarm as well as to promote exploration in the search. pPSA with a perturbed particle updating strategy is "possibly at gbest" instead of a crisp location which is different from other fuzzy PSOs. In order to further understand the effects of uncertainty two new models are proposed and compared with each other together with the initial max–min model.

The remainder of this paper is organized as follows. Some background information is provided in Section 2. The details of the

1568-4946/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2009.06.010

perturbed particle updating strategy and pPSA are described in Section 3. The experimental performance, evolutionary behaviors and two new models on measuring the uncertainty on global numerical optimization are shown are proposed in Section 4. Conclusion and future works are drawn in Section 5.

2. Particle swarm algorithm

A standard particle swarm optimizer maintains a swarm of particles and each individual is composed of three D-dimensional vectors, where D is the dimensionality of the search space. These are the current position x_i, the previous best position p_i, and the velocity v_i. The current position $x_i = (x_{i,1}, \ldots, x_{i,D})$ can be considered as a set of coordinates describing a point in space. The best solution found so far is stored in $p_i = (p_{i,1}, \ldots, p_{i,D})$. New points are chosen by adding $v_i = (v_{i,1}, \ldots, v_{i,n})$ coordinates to x_i, and the algorithm operates by adjusting v_i, which can be seen as a step size.

In essence, the trajectory of each particle is updated according to its own flying experience as well as to that of the best particle in the swarm. Shi and Eberhart [15] introduced the concept of inertia weights to control the relationship between exploration and exploitation. A new variation of PSO model [16] with a nonlinear variation of inertia weight is proposed along with a particle's old velocity to improve the speed of convergence as well as fine tune the search in the multidimensional space. Experimental results suggest that it is preferable to initialize the inertia weight to a large value, giving priority to global exploration of the search space, and gradually decreasing so as to obtain refined solutions [15,17].

$$v_{i,d}^{k+1} = \omega \times v_{i,d}^{k} + c_1 \times r_1^k \times (p_{i,d}^k - x_{i,d}^k) + c_2 \times r_2^k \times (p_{g,d}^k - x_{i,d}^k) \tag{1}$$

$$x_{i,d}^{k+1} = x_{i,d}^{k} + v_{i,d}^{k+1} \tag{2}$$

where $v_{i,d}^k$ is the d-th dimension velocity of particle i in cycle k; $x_{i,d}^k$ is the d-th dimension position of particle i in cycle k; $p_{i,d}^k$ is the d-th dimension of personal best (pbest) of particle i in cycle k; $p_{g,d}^k$ is the d-th dimension of the gbest in cycle k; ω is the inertia weight; c_1 is the cognition weight and c_2 is the social weight; and r_1 and r_2 are two random values uniformly distributed in the range of [0, 1].

Clerc [18] proposed the constriction factor in his "swarm and queen" approach.

$$v_{i,d}^{k+1} = \mathcal{X}[v_{i,d}^{k} + c_1 \times r_1^k \times (p_{i,d}^k - x_{i,d}^k) + c_2 \times r_2^k \times (p_{g,d}^k - x_{i,d}^k)] \tag{3}$$

Kennedy [19] noted that the trajectories of non-stochastic one-dimensional particles contained interesting regularities when $r_1 + r_2$ was between 0.0 and 4.0.

3. Perturbed particle swarm algorithm (pPSA)

3.1. Perturbed particle updating strategy

In PSO, the swarm converges rapidly within the intermediate vicinity of the gbest. However, such a high convergence speed often results in: (1) the lost of diversity and (2) premature convergence if the gbest corresponds to a local optima. This motivates the development of pPSA-a perturbed particle swarm algorithm based on the perturbed gbest updating strategy, which is based on the concept of possibility measure [14] to model the lack of information about the true optimality of the gbest. In contrast to conventional approaches, the gbest in pPSA is denoted as "possibly at gbest (p-gbest) $= (p_{g,1} \times p_{g,2} \times \cdots \times p_{g,D})$", instead of a crisp location. Consequently, the calculation of particle velocity can be rewritten as

$$p_{g,d}^{k'} = N(p_{g,d}^k, \sigma) \tag{4}$$

$$\sigma = p(k) \tag{5}$$

$$v_{i,d}^{k+1} = \omega \times v_{i,d}^{k} + c_1 \times r_1^k \times (p_{i,d}^k - x_{i,d}^k) + c_2 \times r_2^k \times (p_{g,d}^{k'} - x_{i,d}^k) \tag{6}$$

where $p_{g,d}^{k'}$ is the d-th dimension of p-gbest in cycle k. From (4), it can be observed that the p-gbest is characterized by a normal distribution $N(p_{g,d}^k, \sigma)$, where σ represents the degree of uncertainty about the optimality of the gbest. In order to account for the information received over time that reduces uncertainty about the gbest position, σ is modeled as some non-increasing function of the number of cycles as Eq. (7). For simplicity, $p(k)$ is defined as

$$p(k) = \begin{cases} \sigma_{max}, & \text{cycles} < \alpha \times \text{max\_cycles} \\ \sigma_{min}, & \text{otherwise} \end{cases} \tag{7}$$

where σ_{max}, σ_{min}, and α are manually set parameters.

The perturbed global best updating strategy Eqs. (4)–(6) should be distinguished from conventional mutation operator (1) and (2), which applies a random perturbation to the particles. The function of p-gbest is to encourage the particles to explore a region beyond that defined by the search trajectory. By considering the uncertainty associated with each gbest as a function of time, p-gbest provides a simple and efficient exploration at the early stage when σ is large and encourages local fine-tuning at the latter stage when σ is small. Subsequently, this approach helps to reduce the likelihood of premature convergence and guides the search toward the promising search area.

3.2. The perturbed particle swarm algorithm

Procedure of pPSO algorithm

1: Initialize a population array of particles with random positions and velocities on D dimensions in the search space.

2: loop

3: For each particle, evaluate the desired optimization fitness function in D variables.

4: Compare particle's fitness evaluation with its $pbest_i$. If current value is better than $pbest_i$, then let $pbest_i$ be the current value, and p_i be the current location x_i in D-dimensional space.

5: Identify the particle in the neighborhood with the best success so far, and assign its index to the variable g.

6: Change the velocity and position of the particle according to the following equations: $p_{g,d}^{k'} = N(p_{g,d}^k, \sigma)$, for $d = 1, \ldots, D$, where σ is decided by Eq. (5). $\vec{v}_i^{\,k} \leftarrow \omega \times \vec{v}_i^{\,k} + c_1 \times \vec{U}(0,1) \otimes (\vec{p}_i^{\,k} - \vec{x}_i^{\,k}) + c_2 \times \vec{U}(0,1) \otimes (\vec{p}_g^{\,k'} - \vec{x}_i^{\,k}) . \vec{x}_i^{\,k} \leftarrow \vec{x}_i^{\,k} + \vec{v}_i^{\,k}$

7. If a criterion is met (usually a satisfied fitness or a maximal number of iterations), exit loop.

8: End loop

End of pPSO algorithm

Notes:

- $\vec{U}(0,1)$ represents a vector of random numbers uniformly distributed in [0, 1] which is randomly generated at each iteration and for each particle.
- $\otimes$ is component-wise multiplication.

4. Experiments on numerical optimization

In order to validate the necessity and good performance of the proposed perturbed particle updating strategy and pPSA, 12 benchmark [20] functions are adopted, and the results are compared with PSO. Two new models are proposed to control the uncertainty of perturbed particle updating strategy.