Expert Systems With Applications 42 (2015) 8881–8895

Contents lists available at ScienceDirect

Expert Systems With Applications

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

PS–ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems

Zhiyong Li*, Weiyou Wang, Yanyan Yan, Zheng Li

College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China

ARTICLE INFO

Keywords:
Particle swarm optimization
Artificial bee colony
Hybrid algorithm
High-dimensional optimization problems

ABSTRACT

Particle swarm optimization (PSO) and artificial bee colony (ABC) are new optimization methods that have attracted increasing research interests because of its simplicity and efficiency. However, when being applied to high-dimensional optimization problems, PSO algorithm may be trapped in the local optimal because of its low global exploration efficiency; ABC algorithm has slower convergence speed in some cases because of the lack of powerful local exploitation capacity. In this paper, we propose a hybrid algorithm called PS–ABC, which combines the local search phase in PSO with two global search phases in ABC for the global optimum. In the iteration process, the algorithm examines the aging degree of *pbest* for each individual to decide which type of search phase (PSO phase, onlooker bee phase, and modified scout bee phase) to adopt. The proposed PS–ABC algorithm is validated on 13 high-dimensional benchmark functions from the IEEE-CEC 2014 competition problems, and it is compared with ABC, PSO, HPA, ABC–PS and OXDE algorithms. Results show that the PS–ABC algorithm is an efficient, fast converging and robust optimization method for solving high-dimensional optimization problems.

1. Introduction

Global optimization can be applied in many areas of science and engineering (Bomze, 1997; Gergel, 1997; Horst & Tuy, 1996; Lin, Ying, Chen, & Lee, 2008). Especially, high-dimensional optimization problem is a branch of global optimization problems that have attracted increasing attention in the past few years. High-dimensional optimization problems can be formulated as a D-dimensional minimization problem as follows (Gergel, 1997; Nguyen, Li, Zhang, & Truong, 2014):

$$\begin{cases} min & \mathbf{f}(\mathbf{x}), \\ s.t. & \mathbf{l} \le \mathbf{x} \le \mathbf{u} \end{cases} \tag{1}$$

where $\mathbf{f}(\mathbf{x})$ is the objective function, $\mathbf{x} = (x_1, x_2, \ldots, x_D)$ is a vector of variables, D corresponds to the problem dimensions, $\mathbf{l} = (l_1, l_2, \ldots, l_D)$ and $\mathbf{u} = (u_1, u_2, \ldots, u_D)$ define the lower and upper limits of the corresponding variables, respectively. In high-dimensional optimization problems, the search space usually becomes more complex with increasing of dimensionality; thus, solving high-dimensional problems is a considerable challenge.

Due to the practical demands, there were some attempts in trying to use different methods for high-dimensional optimization problems in recent years (Grosan & Abraham, 2009). One method is to use parallel optimization algorithm. This approach aims to solve specific standard functions. Höfinger, Schindler, and Aszódi (2002) proposed a parallel global optimization algorithm for typical high-dimensional problems. Schutte, Reinbolt, Fregly, Haftka, and George (2004) introduced a parallel particle swarm algorithm for some standard functions (Griewank and Corona test functions). However, parallel optimization algorithm is limited in some application fields because parallel computing is difficult to implement for high-dimensional optimization problems.

Several metaheuristic algorithms such as Differential Evolution (DE) (Brest, Greiner, Boskovic, Mernik, & Zumer, 2006; Price, Storn, & Lampinen, 2006; Yang, Tang, & Yao, 2007), Genetic Algorithm (GA) (Chelouah & Siarry, 2000; Sánchez, Lozano, Villar, & Herrera, 2009), Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995; Jiang, Hu, Huang, & Wu, 2007; Karaboga, 2005), Artificial Bee Colony (ABC) (Karaboga & Basturk, 2008; Zhang, Ouyang, & Ning, 2010) and so on, have shown considerable successful in solving high-dimensional optimization problems in the past few years. Among the existing metaheuristic for global optimization, the PSO and ABC methods are highly successful and suitable for some classes of high-dimensional optimization problems. However, the main challenge of the PSO algorithm is that it can easily get stuck in a local optima when handling complex high-dimensional problems. Moreover, the

* Corresponding author. Tel.: +86 13607436411.
E-mail addresses: zhiyong.li@hnu.edu.cn (Z. Li), 1053723695@qq.com (W. Wang), 739954660@qq.com (Y. Yan), 121514277@qq.com (Z. Li).

http://dx.doi.org/10.1016/j.eswa.2015.07.043
0957-4174/

convergence speed of the ABC algorithm is typically lower than other metaheuristic algorithms such as DE and PSO algorithms when solving high-dimensional problems. This is because that PSO has poor exploration ability and ABC has poor exploitation mechanism. Therefore, several modified PSO or ABC algorithms have been proposed to further balance the exploration and exploitation processes, which results in improved convergence speed and avoidance of the local optima. For example, TSai, Pan, Liao, and Chu (2009) improved the exploration ability of ABC by adding the concept of universal gravitation to the onlooker bee phase and applied the interactive ABC (IABC) to five benchmark functions. Jia, Zheng, Qu, and Khan (2011) proposes a novel memetic PSO (CGPSO) algorithm for high-dimensional problems, which combines the canonical PSO with a Chaotic and Gaussian local search procedure. Jamian, Abdullah, Mokhlis, Mustafa, and Bakar (2014) proposed a global PSO (GPSO) algorithm for high-dimensional numerical optimization problems. Imanian, Shiri, and Moradi (2014) proposed a modified ABC (i.e. VABC) for high-dimensional continuous optimization problems.

Metaheuristic algorithms use different exploration and exploitation strategies for high-dimensional optimization problems. In order to overcome the poor exploration ability of PSO and the poor exploitation mechanism of ABC, hybrid metaheuristic algorithm is a new research trend for solving high-dimensional optimization problems, which have attracted considerable attention in recent years. In this paper, hybrid metaheuristic algorithm is a recombination procedure for the hybridization of ABC and PSO. For instance, a novel hybrid swarm intelligent algorithm (IABAP) was developed by Shi et al. (2010) by using information communication between PSO and ABC and the information exchange approach improved the performance of the algorithm. El-Abd (2011) proposed a hybrid approach referred to as ABC–SPSO, which is based on PSO and ABC, for continuous function optimization. Kıran & Gündüz (2013) proposed a hybrid approach (HPA) based on PSO and ABC algorithms for continuous optimization problems. Chun-Feng, Kui, and Pei-Ping (2014) proposed a novel ABC algorithm based on PSO search mechanism (ABC–PS) for global optimization. In these studies, algorithms such as IABAP, ABC–SPSO, HPA and ABC–PS are hybridization of PSO and ABC. Although exploration and exploitation in these algorithms can be balanced to achieve excellent quality results for optimization problems, these techniques cannot solve large-scale global optimization problems that involve high dimensions (Kıran & Gündüz, 2013). Such as in the IABAP, HPA and ABC–PS algorithms, the update rule of the ABC algorithm is executed in each iteration process, thus these three algorithms retain the characteristic of the ABC and have a lower convergence speed on the high-dimensional problems. In addition, IABAP and ABC–SPSO has poor global search ability and poor computing power on the high-dimensional multimodal problems.

Therefore, we propose a new hybrid procedure (PS–ABC) for the hybridization of PSO and ABC by using the exploitation ability of PSO and the exploration ability of ABC. This method use the exploration ability of the ABC based on PSO in the algorithm process. Traditional PSO has great exploitation ability and fast convergence speed (Jia et al., 2011). By contrast, basic ABC has effective exploration ability (Zhu & Kwong, 2010). Thus, the proposed method has fast convergence speed and excellent computing performance for high-dimensional optimization problems. The update status of *pbest* in the PS–ABC algorithm is characterized by three states: active, aged, and dying states. The proposed method determines the optimal solution in the three corresponding phases. An active individual will perform PSO phase to exploit a new solution along the direction of *pbest* and *gbest*. The onlooker bee phase in the aged state has the most outstanding *pbest* for exploring additional possible solutions in the new search space to escape from the search space of the PSO phase. An optimal solution that cannot be updated indicates that the process is in a dying state, and the modified scout bee phase is used to explore the whole search spaces. The performance of PS–ABC is compared with ABC, PSO, HPA, ABC–PS and OXDE algorithms. The experimental results show that the proposed PS–ABC algorithm is more effective on the high-dimensional optimization problems.

The rest of the paper is organized as follows: Section 2 presents the instructions for the PSO and ABC algorithms. Section 3 briefly presents the PS–ABC algorithm, including the algorithm detail, algorithm search ability analysis, algorithm complexity analysis, and algorithm convergence analysis. Section 4 describes the test problems and parameter settings. Section 5 discusses the simulation results over 13 high-dimensional benchmark functions and PS–ABC control parameters. Finally, the conclusion is drawn in Section 6.

2. Related work

2.1. PSO algorithm

PSO, which was proposed by Kennedy and Eberhart in Eberhart and Kennedy (1995), is one of the most recent evolutionary algorithms based on the searching behavior of animals such as fish schooling and bird flocking. In PSO model, each individual is composed of three vectors: the velocity $\mathbf{v}_i$, the current position $\mathbf{x}_i$, and the previous best position $\mathbf{pbest}_i$. Suppose that the objective function is D-dimensional, then the velocity and position of the ith particle are represented as $\mathbf{v}_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iD})$ and $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD})$, respectively, while its previous best position is stored in $\mathbf{pbest}_i = (pbest_{i1}, pbest_{i2}, pbest_{i3}, \ldots pbest_{iD})$. In each generation, the best position discovered from all *pbest* positions is known as the global best position $\mathbf{gbest} = (gbest_1, gbest_2, gbest_3, \ldots gbest_D)$.

The process of PSO is presented below (Eberhart & Shi, 2001):

***Step 1:* Initialization**
Assign parameters and create populations.
Set $iter = 0$.
***Step 2*: Reproduction and updating loop**
for $i = 1, 2, \ldots, N$ **do**
Update the velocity v_i of particle x_i by using (2)
$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (pbest_i^t - x_i^t) + c_2 \cdot r_2 \cdot (gbest^t - x_i^t)$. (2)
Update the position of particle x_i by using (3)
$x_i^{t+1} = x_i^t + v_i^{t+1}$. (3)
Evaluate the fitness value of the new particle x_i.
if x_i is better than $pbest_i$ **then**
Set x_i to be $pbest_i$.
end if
end for
Set the particle with best fitness value to be *gbest*.
$iter = iter + 1$.
***Step 3*: If the stop criterion is satisfied, the process is terminated. Otherwise, return to Step 2.**

The PSO algorithm has three stages: initialization, iteration and termination criterion. In initialization stage, the population is initialized and randomly distributed in the search space. In the iteration stage, the velocities and positions of the particles are updated by Eqs. (2) and (3), respectively. The velocity equation in PSO is

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (pbest_i^t - x_i^t) + c_2 \cdot r_2 \cdot (gbest^t - x_i^t) \tag{2}$$

and the position equation is

$$x_i^{t+1} = x_i^t + v_i^{t+1}. \tag{3}$$

where c_1 and c_2 are two positive constants that indicate the relative influence of the cognition and social components, respectively; w is inertia weight that provides a balance between local exploitation and global exploration; r_1 and r_2 are random real values in interval [0, 1]. The velocity of the particles on each dimension is clamped to the range $[-V_{\max}, V_{max}]$.

If the terminate criterion is satisfied, the algorithm produce the best solution (*gbest*). Otherwise, the iteration stage is repeated.