Engineering Applications of Artificial Intelligence 39 (2015) 109–119

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Roach infestation optimization with friendship centers

Hsing-Chih Tsai [a,b,*]

[a] Department of Civil and Construction Engineering, National Taiwan University of Science and Technology, Taiwan
[b] Ecological and Hazard Mitigation Engineering Researching Center, National Taiwan University of Science and Technology, Taiwan

ARTICLE INFO

Article history:
Received 3 June 2014
Received in revised form
27 October 2014
Accepted 3 December 2014
Available online 24 December 2014

Keywords:
Particle swarm optimization
Roach infestation optimization
Friendship centers

ABSTRACT

Roach infestation optimization (RIO) is a new adaption of particle swarm optimization (PSO) that significantly improves algorithm effectiveness in finding the global optima. This paper assesses the effectiveness of using swarm centers to further improve RIO convergence performance. Swarm centers have previously been applied in PSO as the center PSO. This paper introduces two RIO variants using one center agent and individual friendship center agents. In the first, the center agent has no explicit velocity and is positioned in each iteration at the center of the swarm. In the second, each individual friendship center adopts a position located at the center of its friends. This paper conducts experiments on 13 benchmark function optimization problems, 2 neural network learning problems, and 2 engineering design problems. Experimental results show that the RIO with a swarm center did not perform as well as the center particle in improving PSO. The behavior of Find_Friends in RIO requires each roach agent to move toward its friendship center rather than oscillate around the swarm center. The friendship centers significantly improved RIO in terms of convergence speed and stability with a minor 37.47% additional time cost.

1. Introduction

Swarm intelligence (SI), the collective behavior that models natural systems of ant colonies, bird flocks, animal herds, bacterial growth, and fish schools, has attracted significant attention in recent years. There are many SI algorithms, including particle swarm optimization (PSO) (He and Wang, 2007; Tsai, 2010), fish swarm algorithm (Tsai and Lin, 2011), artificial bee colony (Tsai, 2014), gravitational search algorithm (Rashedi et al., 2009; Tsai et al., 2013), cuckoo search (Agrawal et al., 2013), and firefly algorithm (Bojic et al., 2012), among others. PSO, inspired by the social behavior of birds and fish, has received significant positive attention in recent years due to its ease of implementation and rapid convergence on optimum solutions (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995). Its ability to solve numerous scientific and engineering problems efficiently and effectively has given PSO increasing support and acceptance among researchers (Reche López et al., 2008; Liu et al., 2008; Anghinolfi and Paolucci, 2009; Wang and Singh, 2009; Ibrahim et al., 2012; Kwok et al., 2013; Mazhoud et al., 2013; Shao et al., 2013; Tsai et al., 2014). Susceptibility of the original PSO to entrapment by local minima led to the development of a revised version of PSO (Parsopoulos and Vrahatis, 2005; Liu et al., 2007; Tsai, 2010; Tsai et al., 2012). Angeline (1998) highlighted poor local search ability as a drawback of the original version of PSO. To overcome this disadvantage, Shi and Eberhart (1998) proposed introducing a linearly decreasing inertia factor into the original PSO velocity update equation in order to linearly decrease inertia weight of particle swarm optimization. Clerc and Kennedy (2002) introduced a constriction factor into PSO to restrict particle velocities. Kennedy and Mendes (2002), Mohais et al. (2004), and Suganthan (1999) investigated PSO neighborhood effects. Parsopoulos and Vrahatis (2005) developed a unified PSO with both local and global variants. Liu et al. (2007) proposed a center PSO (CPSO), which added a center particle that assumed the average features of the entire swarm. All of the above have improved the efficacy of PSO as an optimization tool.

Although PSO currently ranks amongst the most popular SI algorithms, it is only one of many SI algorithms in circulation inspired by natural phenomena, especially phenomena observed in biological systems. Recent research has shown that cockroaches have complex social behavior (Jeanson et al., 2005). Cockroaches prefer to concurrently optimize numbers of friends and level of shelter darkness (Halloy et al., 2007). Groups of robots may simulate the collective behavior of cockroaches, with each robot programmed to act according to a simple set of behaviors. Perceiving cockroaches as a good model for swarm intelligence algorithms, Havens et al. (2008) proposed a roach infestation optimization (RIO) algorithm based partly on PSO equations. Major differences between PSO and RIO are: (1) RIO agents are designed to congregate under dark shelters, whereas PSO particles are designed to gather food; (2) RIO uses

* Corresponding author at: #43, Sec. 4, Keeling Rd., Taipei, Taiwan, R.O.C. 106. Tel.: +886 2 27301277, +886 2 27376663; fax: +886 2 27301074.
E-mail address: tsaihsingchih@gmail.com

http://dx.doi.org/10.1016/j.engappai.2014.12.003
0952-1976/

individual local best positions to replace the global best position of PSO; and (3) RIO adds a random search behavior to prevent convergence on local minima. Havens et al. (2008) conducted function optimization problem experiments and concluded that RIO finds global optima more effectively than PSO. Therefore, RIO may be considered a practical improvement on the PSO.

Based on the aforementioned contributions of the CPSO and RIO (Liu et al., 2007; Havens et al., 2008), this paper studies the concept of swarm centers and investigates the effectiveness of using a center agent and/or friendship centers to improve RIO. The remainder of this paper is organized as follows: Section 2 introduces PSO, CPSO, RIO, and RIO variants with a center agent or friendship agents; Section 3 assesses the comparative performance of PSO, CPSO, RIO, and the RIO variants using benchmarks of 13 function optimization problems, 2 neural network learning problems, and 2 engineering design problems as well as makes suggestions for future research in this area; and Section 4 presents conclusions.

2. Swarm intelligent algorithms

2.1. Particle swarm optimization

PSO solves a D-dimensional optimization problem with N particles using a randomly initialized population in which each particle i moves at a certain velocity at iteration t, denoted as $\vec{V}_i(t)=(V_{i,1}(t),...,V_{i,d}(t),...,V_{i,D}(t))$. $\vec{V}_i(t)$ is a D-dimensional vector, numbered by index d to give velocity momentum to the particle. Momentum is updated using 2 behavior variables: memory of the entire swarm (social behavior) and current perception of each particle (cognitive behavior). This fundamental PSO behavior can be expressed as

$$\vec{V}_i(t+1)=C_0(t)\vec{V}_i(t)+C_{max}\vec{R}_1\otimes(\vec{P}_i(t)-\vec{X}_i(t))+C_{max}\vec{R}_2\otimes(\vec{G}(t)-\vec{X}_i(t)) \quad (1)$$

$$\vec{X}_i(t+1)=\vec{X}_i(t)+\vec{V}_i(t+1) \quad (2)$$

where $\otimes$ is the component-wise multiplication; $\vec{R}_1$ and $\vec{R}_2$ are the D-dimensional vectors that create a rotationally variant PSO (Havens et al., 2008; Janson and Middendorf, 2007). Reflecting the argument of Spears et al. (2010) that ambiguity unfortunately continues to propagate through the literature, current references offer two interpretations of R_1 and R_2 as either scalars or vectors. James Kennedy, one of the creators of PSO, stated a preference for the rotationally variant PSO due to its explorative quality. Therefore, Spears et al. (2010) further compared the two versions of scalar and vector R_1 and R_2 in detail and finally concluded that the rotationally variant PSO was the most commonly used. C_0 and C_{max} are tuning parameters for momentum, cognitive, and social terms. A positive number less than 1 is suggested for C_0 to create a decreasing momentum. Values among 0 to 4 are usually adopted for C_{max} and 2 is the original suggestion. t is the iteration index and the D-dimensional position of the ith particle is $\vec{X}_i(t)=(X_{i,1}(t),...,X_{i,D}(t))$ updated by D-dimensional momentum $\vec{V}_i(t)$. The particle best position $\vec{P}_i(t)=(P_{i,1}(t),...,P_{i,D}(t))$ influences the ith particle and the global best position $\vec{G}(t)=(G_1(t),...,G_D(t))$ impacts the entire swarm. Both parameters of $\vec{P}_i(t)$ and $\vec{G}(t)$ are determined by the value of fitness functions (F) by the lapse of iterations.

2.2. Center PSO

The CPSO proposed by Liu et al. incorporates an additional center particle to improve PSO performance (Liu et al., 2007). The entire swarm is used to calculate the current position of the center particle (X_c). The center particle has no velocity momentum and, therefore, does not participate in Eqs. (1) and (2) operations. However, the center particle is used to replace the $\vec{G}$, if competent.

$$\vec{X}_c(t+1)=\frac{1}{N}\sum_{i=1}^{N}\vec{X}_i(t+1) \quad (3)$$

Therefore, at each iteration t, Eqs. (1) and (2) update particle velocities and positions, and then each $\vec{P}_i(t+1)$ and the $\vec{G}(t+1)$ are updated for the next iteration. If its position provides better fitness, the center particle participates in the $\vec{G}(t+1)$ competition in order to lead the swarm toward this center position. The CPSO algorithm is outlined in Algorithm 1. The PSO algorithm can be also obtained from Algorithm 1 by removing the pseudocodes of the swarm center.

Algorithm 1. Center particle swarm optimization

Inputs: fitness function $F(\vec{X}_i)\in R^D$
Parameters: Number of particles N, maximum iterations *Max_ite*, and swarm parameters $C_0=0.7$, $C_{max}=1.43$, termination criteria.
Initialization: Random population $\vec{X}_i$, zero velocities $\vec{V}_i=\vec{0}$, $\vec{P}_i=\vec{X}_i$, $\vec{G}=$best of $\vec{X}_i$
for $t=1$ to *Max_ite* **do**
  for $i=1$ to N **do**
    Calculate particle positions $\vec{X}_i$ using Eqs. (1) and (2)
  Calculate particle fitness values $F(\vec{X}_i)$
  Update $\vec{P}_i$ and $\vec{G}$ based on particle fitness values
  Calculate $\vec{X}_c$using Eq. (3)//*perform CPSO*
  if $F(\vec{X}_c)<F(\vec{G})$ **then**//for minimization problems
    $\vec{G}=\vec{X}_c$//*center particle*
    end if
  end for
  Check termination criteria
end for

2.3. Roach infestation optimization

RIO, a revised version of PSO based on the social behavior of cockroaches proposed by Havens et al. (2008), implements optimization using the following behaviors:

1. Find_Darkness: A roach moves at velocity $\vec{V}_i$ at position $\vec{X}_i$ with a recognized personal best (darkest) position $\vec{P}_i$ in search of a comfortable (dark) position.
2. Find_Friends: A roach uses strong ties of friendship and communication with roaches near its current position, depending on group parameters (A_1, A_2 and A_3), to attain a local best position $\vec{L}_i$ and search the hyper-rectangle formed by $\vec{P}_i$ and $\vec{L}_i$ in search of an optimally comfortable position.
3. Find_Food: Each roach grows increasingly hungry over time and will eventually leaves its comfortable position and seeks a new position ($\vec{b}$) to satisfy its hunger.

Therefore, each roach particle updates it position using either its new velocity or a random position determined by hunger.

$$\vec{X}_i(t+1)=\begin{cases}\vec{X}_i(t)+\vec{V}_i(t+1), & hunger_i<T_{hunger}\\ \vec{b}, & hunger_i=T_{hunger}\end{cases} \quad (4)$$

where $hunger_i$ is an incremental hunger counter initially determined at random from $[0,T_{hunger}]$ that identifies a roach's current