Engineering Applications of Artificial Intelligence 26 (2013) 2327–2348

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Two-layer particle swarm optimization with intelligent division of labor

Wei Hong Lim, Nor Ashidi Mat Isa *

Imaging and Intelligent System Research Team (ISRT), School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia

ARTICLE INFO

Article history:
Received 12 December 2012
Received in revised form
15 May 2013
Accepted 26 June 2013
Available online 17 July 2013

Keywords:
Particles swarm optimization (PSO)
Intelligent division of labor (IDL)
Two-layer particle swarm optimization with intelligent division of labor (TLPSO-IDL)

ABSTRACT

Early studies in particle swarm optimization (PSO) algorithm reveal that the social and cognitive components of swarm, i.e. memory swarm, tend to distribute around the problem's optima. Motivated by these findings, we propose a two-layer PSO with intelligent division of labor (TLPSO-IDL) that aims to improve the search capabilities of PSO through the evolution memory swarm. The evolution in TLPSO-IDL is performed sequentially on both the current swarm and the memory swarm. A new learning mechanism is proposed in the former to enhance the swarm's exploration capability, whilst an intelligent division of labor (IDL) module is developed in the latter to adaptively divide the swarm into the exploration and exploitation sections. The proposed TLPSO-IDOL algorithm is thoroughly compared with nine well-establish PSO variants on 16 unimodal and multimodal benchmark problems with or without rotation property. Simulation results indicate that the searching capabilities and the convergence speed of TLPSO-IDL are superior to the state-of-art PSO variants.

1. Introduction

In recent years, there has been a growing interest in utilizing the stochastic search algorithms to solve the global optimization problems, due to the emergence of high complexities problems such as non-differentiable rotated shifted multimodal and hybrid composition functions (Suganthan et al., 2005). Backed by their heuristic and random natures, the stochastic search algorithms are independent on the differentiability of the fitness function and thus have better searching capability than the classical analytical approach (Nasir et al., 2012). Most of these stochastic search algorithms are nature and biological inspired and some of them that are widely used nowadays are Genetic Algorithm (GA) (Goldberg and Holland, 1988; Melanie, 1999), Differential Evolution (DE) (Das and Suganthan, 2011; Storn and Price, 1997; Zhang and Sanderson, 2009), Harmony Search (HS) (Alia and Mandava, 2011; Ashrafi and Dariane; Geem et al., 2001; Lee and Geem, 2005), Group Search Optimizer (GSO) (He et al., 2006, 2009), and Particle Swarm Optimization (PSO) (Banks et al., 2007, 2008; del Valle et al., 2008; Eberhart and Shi, 2001; Kennedy and Eberhart, 1995; Kennedy et al., 2001).

PSO algorithm, proposed by Kennedy and Eberhart (1995), is inspired from the collaboration behavior of bird flocking and fish schooling in searching for food (Banks et al., 2007; del Valle et al., 2008; Eberhart and Shi, 2001; Kennedy and Eberhart, 1995; Kennedy et al., 2001). Each individual (i.e. particle) in the PSO population represents the potential solution of the optimization problem, whilst the location of food source is the global optimum solution of the corresponding problem. As a population-based stochastic algorithm, PSO simultaneously evaluates many points in the search space, in order to probe the promising regions of the search space effectively. One key feature that distinguishes the PSO from other stochastic search algorithms is the capability of the PSO particle to remember its current position in the search space, as well as its personal best position, i.e. the best position that a particle has ever attained. In other words, the PSO particles are capable to roam around the search space through the current swarm and simultaneously memorize the best positions found so far in the memory swarm. While searching in the multi-dimensional search space, each particle's movement is adjusted through the information sharing and collaboration mechanisms between individuals, in order to lead the swarm to the global optimum solution. Specifically, each particle's searching trajectory is stochastically manipulated according to the memory swarm members, i.e. the personal best position of particle (i.e. cognitive experience) and the position attained by the best particle in its society (i.e. social experience) (Banks et al., 2007; Eberhart and Shi, 2001; Kennedy and Eberhart, 1995; Kennedy et al., 2001). Due to its effectiveness and robustness in optimization problems, the application of PSO in various engineering application has become an active research trend (AlRashidi and El-Hawary, 2009; Banks et al., 2008; del Valle et al., 2008; Gao et al., 2010; Juang et al., 2010; Song et al., 2007).

* Corresponding author. Tel.: +60 45996093; fax: +60 45941023.
E-mail addresses: limweihong87@yahoo.com (W.H. Lim), ashidi@eng.usm.my (N.A. Mat Isa).

0952-1976/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2013.06.014

Despite the competitive performance of PSO, researchers have noted the tendency of PSO swarm to converge prematurely in the local optima, due to its rapid convergence on the best position found so far at the early stage of optimization (Liang et al., 2006; van den Bergh and Engelbrecht, 2004). Most of the time, the locality of this best-so-far position may be the local optimum that is located potentially distant from the global optimum. Once the swarm congregates at such position, little opportunity is afforded for the population to explore for other solution possibilities. This leads to the entrapment of the swarm within the local optima of search space and thus premature convergence occurs. Another challenging issue that needs to be addressed is the proper control on the exploration and exploitation searching of PSO swarm, in order to locate the optimum solution efficiently (Angeline, 1998; Shi and Eberhart, 1998). Neither the exploration nor the exploitation searching should be over-emphasized as the former tends to prevent the swarm convergence, whilst the latter could raise the risk of premature convergence issue (Shi and Eberhart, 1998).

Extensive experimental studies have verified that PSO's current swarm tends to explore the search space before congregate around their attractors, i.e. the personal and best neighborhood positions, whilst the memory swarm is distributed around the vicinity of problem's minima to exploit these regions (Epitropakis et al., 2012). Motivated by these findings, we aim to improve the best knowledge of each particle through the evolution of both the cognitive and social components in memory swarm. Specifically, after the evolution of current swarm, an intelligent optimization procedure is proposed to provide additional evolution on the memory swarm, so that it can either locate more promising regions around the minimizer or to fine-tune the regions of the already found minima. To this end, we propose a two-layer PSO with intelligent division of labor (TLPSO-IDL), which is capable to perform the evolution on both the current swarm and memory swarms sequentially. For current swarm, a new learning mechanism is proposed to enhance the swarm's exploration capability. Meanwhile, an intelligent division of labor (IDL) module is developed to evolve the memory swarm by adaptively allocate different search tasks (i.e. exploitation and exploration) to each swarm member, without compromising the algorithm's search dynamic. To resolve the premature convergence issue, an elitist-based perturbation (EBP) module is employed to prevent the stagnation of PSO swarm in local optima for m successive function evaluations (FEs).

The remainder of this paper is organized as follows. Section 2 briefly discusses some related works. Section 3 details the methodologies of the proposed TLPSO-IDL algorithm. Section 4 describes the experimental settings, whilst Section 5 presents the experimental results. Finally, Section 6 concludes the paper.

2. Related works

For self-completeness purpose, we briefly discuss the basic mechanism and some insight about the main characteristic of the basic PSO algorithm. This section ends with a review of several state-of-the-art PSO variants.

2.1. Basic PSO algorithm

In basic PSO, each particle that is roaming through the D-dimensional problem hyperspace represents the potential solution for a specific problem. For particle i, two vectors, i.e. position vector $X_i = [X_{i1}, X_{i2}, ..., X_{iD}]$ and velocity vector $V_i = [V_{i1}, V_{i2}, ..., V_{iD}]$, are used to represent its current state. Additionally, each particle i can memorize its personal best experience ever encountered (i.e. cognitive experience), represented by the personal best position vector $P_i = [P_{i1}, P_{i2}, ..., P_{iD}]$. The position attained by the best particle in the society (i.e. social experience) is represented as $P_n = [P_{n1}, P_{n2}, ...P_{nD}]$. In the literature, P_n is represented as P_g and P_l for global and local versions of PSO respectively (del Valle et al., 2008; Kennedy et al., 2001). More specifically, P_g is propagated to the entire swarm, whilst P_l is communicated between the neighboring particles with pre-specified topology. Without loss of generality, the global version PSO is employed in the rest of this paper.

During the execution, the trajectory of each particle in the search space is manipulated by dynamically updating the particle's velocity according to its cognitive experience, P_i and the group's social experience, P_g. Mathematically, at iteration (t+1) of the searching process, the dth dimension of particle i's velocity, $V_{i,d}$ (t +1) and position $X_{i,d}$ (t +1) are updated as follows:

$$V_{i,d}(t+1) = V_{i,d}(t) + c_1 r_1 (P_{i,d}(t) - X_{i,d}(t)) + c_2 r_2 (P_{g,d}(t) - X_{i,d}(t)) \quad (1)$$

$$X_{i,d}(t+1) = X_{i,d}(t) + V_{i,d}(t+1) \quad (2)$$

where c_1 and c_2 are the acceleration coefficients; r_1 and r_2 are two random numbers generated from a uniform distribution with the range of [0, 1]. Particle's velocity is clamped to a maximum magnitude of V_{max} to prevent swarm explosion (Angeline, 1998; Eberhart and Shi, 2001). If the absolute velocity of particle i in d-th dimension,$|V_{i,d}|$, exceeds the limit of $|V_{\max,d}|$, $V_{i,d}$ is then confined to $sign(V_{i,d}) * V_{\max,d}$.

Without loss of generality, the minimization problem is considered in the rest of this paper. When minimizing the fitness function f in D-dimensional search space, particle i's P_i position in iteration (t+1) is updated as follows:

$$P_i(t+1) = \begin{cases} X_i(t+1) & \text{if } f(X_i(t+1)) < f(P_i(t)) \\ P_i(t) & \text{otherwise} \end{cases} \quad (3)$$

Meanwhile, the P_g position is identified from the P_i position if all particles and P_i with lowest fitness is assigned as P_g. For particle swarm with the population size of S, the P_g position at iteration t is updated as follows:

$$P_g(t) = \text{argmin}(f(P_i(t))) \quad \forall i \in [1, S] \quad (4)$$

The implementation of basic PSO algorithm is illustrated in Fig. 1. From Eq. (1), it is evident that each PSO particle is strongly influenced by their attractors, i.e. P_i and P_g and this leads to the swarm hastily congregate to the best position found so far. Nevertheless, such position is never guaranteed to be the global optimum or even the local optimum. As the swarm converges, it is unlikely for particles to explore for other solution possibilities due to the population's diversity loss. Henceforth, the population is trapped within the narrow region of the search space and premature convergence occurs (Liang et al., 2006; van den Bergh and Engelbrecht, 2004).

```
Generate initial swarm and set up parameters for each particle;
Reset t = 0;
while t < max_generation do
    for each particle i do
        Update the velocity V_i and position X_i using Equations (1) and (2);
        Fitness evaluation is performed on the updated X_i of particle i;
        if f(X_i) < f(P_i) then
            P_i = X_i;  f(P_i) = f(X_i);
            if f(X_i) < f(P_g) then
                P_g = X_i;  f(P_g) = f(X_i);
            end if
        end if
    end for
    t = t + 1;
end while
```

Fig. 1. Basic PSO algorithm.