Expert Systems with Applications 41 (2014) 3460–3476

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Hierarchical Particle Swarm Optimization with Ortho-Cyclic Circles

Kirupa Ganapathy *, V. Vaidehi, Bhairavi Kannan, Harini Murugan

Department of Information Technology, Madras Institute of Technology, Anna University, Chennai, India

ARTICLE INFO

Keywords:
Particle Swarm Optimization
Cloud computing
Remote health monitoring
Dynamic environment
Orthogonality
Dynamic Round Robin Scheduling

ABSTRACT

Cloud computing is an emerging technology which deals with real world problems that changes dynamically. The users of dynamically changing applications in cloud demand for rapid and efficient service at any instance of time. To deal with this paper proposes a new modified Particle Swarm Optimization (PSO) algorithm that work efficiently in dynamic environments. The proposed Hierarchical Particle Swarm Optimization with Ortho Cyclic Circles (HPSO-OCC) receives the request in cloud from various resources, employs multiple swarm interaction and implements cyclic and orthogonal properties in a hierarchical manner to provide the near optimal solution. HPSO-OCC is tested and analysed in both static and dynamic environments using seven benchmark optimization functions. The proposed algorithm gives the best solution and outperforms in terms of accuracy and convergence speed when compared with the performance of existing PSO algorithms in dynamic scenarios. As a case study, HPSO-OCC is implemented in remote health monitoring application for optimal service scheduling in cloud. The near optimal solution from HPSO-OCC and Dynamic Round Robin Scheduling algorithm is implemented to schedule the services in healthcare.

1. Introduction

Optimization problem refers to the process of minimizing or maximizing the value of an optimization function subject to constraints. It is the process of evaluating the optimization function with selected values of input from a confined set. Evolutionary Computation (EC) is an optimization method with stochastic behavior. ECs work on a set of inputs called population with an iterative approach. Swarm Intelligence (SI) (Poli, Kennedy, & Blackwell, 2007) is the process of achieving desirable results by a swarm as a whole. This is facilitated by local interactions within the swarm and communication with the environment. Members of the swarm learn from others by synergy and move towards the goal, thereby exhibiting a social behavior.

Particle Swarm Optimization (PSO) is one of the Evolutionary Algorithm (EA), which have become an active branch of SI, simulating swarm behavior like fish schooling, wasp swarming and bird flocking. PSO is found to be different from other evolutionary optimization algorithms, for not employing conventional operators like crossover and mutation. PSO solves the optimization problem by augmenting candidate solutions iteration by iteration. The solutions are represented as particles, the collection of which constitutes the swarm. The particles have distinct properties like velocity and position that define their state in the search space. These particles move in steps as defined by their velocity, which is determined by their local best known position and the global best position of the swarm. This way, the swarm is expected to converge at an optimum point. In order to avoid premature convergence, the particles may use the best position of sub-swarm that is formed within the neighborhood of the particle. Particle neighborhood depends on the scheme of the swarm's population topology.

PSO is found to be useful to copious applications deployed in dynamic backdrop. Applications are said to be dynamic when their environment consists of continuous changes like advent of a new process, machine failure, unanticipated downtime, network failure, unstable network connectivity and others. Using classic PSO in dynamic environment is a critical issue where the particles converge to local or global optima over some successive iterations and the arrival of new particle makes the converged particles to start from the scratch for tracking the new optima. Classic PSO exhibits major drawback due to loss of diversity. The probability of premature convergence is high when the dimensionality of the search space is large. Another reason for loss of diversity is that the particles move to a single point which is determined by the gbest and pbest. But this point is not guaranteed to be a local optimum. Drawbacks of classic PSO is more in static environment and it is eventually more severe in dynamic problems. Therefore, classic PSO needs to be modified to deal loss of diversity in real world problems. Several mechanisms are adopted to improve the performance of classic PSO in dynamic environment. These mechanisms are dynamic parameter tuning, dynamic network topology, hybridizing PSO

* Corresponding author.
E-mail addresses: kirupa1980@gmail.com (K. Ganapathy), vaidehi@annauniv.edu (V. Vaidehi), bhairavi_812@yahoo.com (B. Kannan), harinim02@gmail.com (H. Murugan).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.10.050

with genetic algorithm, multi swarm approach, multi dimensionality, dynamically changing neighborhood structures etc. However, there are a number of problems in such Dynamic Optimization Problems (DOP) which remain unsolved.

This paper suggests a two level novel PSO optimization technique as a solution to DOPs, where a dynamic change in one level is overcome by the other. Contribution of this paper for improving the adaptation of PSO in dynamic environment are described as follows.

1. Multiple swarms are constructed based on the particles similarities in the form of circles/swarms.
2. The particles share the information within the circles undergoes conventional PSO for convergence.
3. Selection of similarly converging circle for information sharing between the swarms employs a special orthogonal array analysis.
4. Hierarchical PSO (second level PSO) is employed where the velocity of the gbest of the selected ortho cyclic circle is used to update the velocity of the competing circle particles and refine the position.

Brief discussion on proposed Hierarchical Particle Swarm Optimization with Ortho-Cyclic Circles (HPSO-OCC) is as follows. The algorithm aims to improve the performance of dynamic PSO by offering accurate best solution and faster convergence speed. In the first level, the swarms are grouped based on the similar properties and allowed to undergo conventional PSO. In a topological structure, every swarm discovers the similarly converging neighbor swarms using cyclic property and selects the best neighbor swarm using orthogonal analysis. The information from the personal best fitness (pbest) of thus discovered ortho-cyclic circle is used along with the pbest of the circle and gbest of all circles to define the velocity and refine the position. Second level PSO is performed in the current swarm with the updated velocity equation.

HPSO-OCC algorithm is found to be suitable for numerous applications such as surveillance, military, habitat monitoring, sensor selection etc. As a case study, the proposed HPSO-OCC with Dynamic Round Robin scheduler is implemented in remote health monitoring application for optimal service scheduling. Physiological sensors worn over the body of the patient's sends the vital sign measurement to the remote cloud server. Web enabled sensor data stream are categorised as normal and abnormal class. Based on the abnormality percentage, the particles enter the swarms and HPSO-OCC identifies the optimal patient (particles) to be served with minimum waiting time and schedules them using dynamic round robin scheduler.

2. Particle Swarm Optimization Preliminaries

PSO employs population-based stochastic search (Engelbrecht, 2006; Kennedy & Mendes, 2002) with an iterative learning approach, exploring optima in a multidimensional search space. The algorithm is initialized with a population of particles, where each particle betokens a plausible solution in a δ-dimensional space, which is found by utilizing personal memories of the particles and shared information within a specific neighborhood. Each particle P_i has position vector $\varkappa_i = [\varkappa_{i1,}\varkappa_{i2},\ldots,\varkappa_{id}]$ and velocity vector $\upsilon_i = [\upsilon_{i1,}\upsilon_{i2},\ldots,\upsilon_{id}]$ where $\varkappa_{id} \in \{-100, 100\}$; $i = 1, 2,\ldots,N$ (total number of particles); and $d = 1, 2,\ldots,\delta$ (Kennedy & Eberhart, 1997). The movements of the particles are guided by their own best known position in the search-space, called pbest (μ_i), as well as the entire swarm's best known position, called gbest (ε). This is called Global PSO (GPSO). A second version of PSO exists, called Local PSO (LPSO) that considers best known position in a particle's neighborhood (lbest) instead of gbest. Neighborhood is defined based on the topology used as shown in Fig. 1. The vectors υ_i and $\varkappa_i$ are randomly initialized and are revised based on Eqs. (1) and (2).

$$\upsilon_{id} \leftarrow \omega\upsilon_{id} + \varphi_p r_p(\mu_{id} - {}_{id}) + \varphi_g r_g(\upsilon_{id} - \chi_{id}) \tag{1}$$

$$\chi_{id} = \chi_{id} + \upsilon_{id} \tag{2}$$

The parameters ω, φ_p, and φ_g which are selected by the practitioner regulates the delivery and potency of the PSO algorithm. Coefficients φ_p and φ_g are cognitive and social acceleration factors (Zhan, Zhang, Li, & Chung, 2009) that can take values in the range [0,4] (1.49 and 2.00 are the most commonly used values), where φ_p is the personal accelerator and φ_g is the global accelerator. High value of ω, the inertia weight (Shi & Eberhart, 1998) advocates exploration, and a small value patronizes exploitation. ω is often linearly decreased from a high value (0.90) to a low value (0.40) along the generations of PSO.The search space of the flying particles is limited in the range $[\varkappa_{min}, \varkappa_{max}]$. Their velocities are regulated within a reasonable limit, which is taken care by the parameter $\upsilon_{max.}$ $\upsilon_{max.}$ is a positive integer that determines the maximum step one particle can take during one iteration, and is generally *set* to a value$\varkappa_{max}$- $\varkappa_{min.}$ The fitness of each particle is calculated in every generation using a fitness function *f*. The functioning of classical PSO is shown in Algorithm 1.

Algorithm 1 PSO

BEGIN
(a) UNTIL a termination criterion is m*et* i.e., number of iterations performed, REPEAT
(b) FOR EACH particle $\boldsymbol{i} \leftarrow \boldsymbol{0, 1, \ldots, N}$ **DO**
(i) FOR EACH dimension $\mathbf{d} \leftarrow \mathbf{1, \ldots,}\ \delta$ DO
(1)Pick random numbers: $\boldsymbol{r_p, r_g} \sim \boldsymbol{U(0,1)}$
(2) Update the particle's velocity:
$\upsilon_{\boldsymbol{id}} \leftarrow \boldsymbol{\omega}\ \upsilon_{\boldsymbol{id}} + \boldsymbol{\varphi_p r_p (\mu_{i,d}\text{-}\varkappa_{id}) + \varphi_g r_g (\varepsilon_d\text{-}\varkappa_{id})}$
(ii) Update the particle's position: $\boldsymbol{\varkappa_{id}} \leftarrow \boldsymbol{\varkappa_{id}} + \upsilon_{\boldsymbol{id}}$
(iii) IF ($f(\varkappa_i) < f(\mu_i)$) THEN
(1) Update the particle's best known position: $\boldsymbol{\mu_i} \leftarrow \boldsymbol{\varkappa_i}$
(2) IF ***(f(μ) < f(ε))*** THEN update the swarm's best known position: $\boldsymbol{\varepsilon} \leftarrow \boldsymbol{\mu_i}$
RETURN ε
END

The remainder of this paper is organised as follows. Section 3 describes the related works in PSO algorithm. Section 4 elaborates the proposed HPSO-OCC algorithm. Section 5 describes the application of HPSO-OCC in remote healthcare application. Section 6 presents the result and discussions with experimental set-up, performance analysis of HPSO-OCC with existing PSO algorithms and performance of HPSO-OCC with optimization functions respectively. Section 7 discusses the conclusions.

3. Related works

PSO algorithm is one of the evolutionary algorithm that solves optimization problems. Recently, much of the real time problems are solved using PSO due to its simplicity. Many researchers have modified classic PSO to solve the problems such as loss of diversity, outdated memory, reliability, convergence speed etc. Several techniques are applied to improve traditional PSO search mechanism in static and dynamic environments. In recent years, investigation of PSO in changing environment problems has become one of the most important issue for real time applications. To solve the issues,