Applied Soft Computing 12 (2012) 1158–1167

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

A new hybrid approach for dynamic continuous optimization problems

J. Karimi, H. Nobahari, S.H. Pourtakdoust*

Center for Research and Development in Space Science and Technology, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 21 February 2010
Received in revised form 27 July 2011
Accepted 1 November 2011
Available online 12 November 2011

Keywords:
Dynamic optimization
Particle swarm optimization
Quadratic interpolation
Reproduction

ABSTRACT

A new hybrid approach for dynamic optimization problems with continuous search spaces is presented. The proposed approach hybridizes efficient features of the particle swarm optimization in tracking dynamic changes with a new evolutionary procedure. In the proposed dynamic hybrid PSO (DHPSO) algorithm, the swarm size is varied in a self-regulatory manner. Inspired from the microbial life, the particles can reproduce infants and the old ones die. The infants are especially reproduced by high potential particles and located near the local optimum points, using the quadratic interpolation method. The algorithm is adapted to perform in continuous search spaces, utilizing continuous movement of the particles and using Euclidian norm to define the neighborhood in the reproduction procedure. The performance of the new proposed approach is tested against various benchmark problems and compared with those of some other heuristic optimization algorithms. In this regard, different types of dynamic environments including periodic, linear and random changes are taken with different performance metrics such as real-time error, offline performance and offline error. The results indicate a desirable better efficiency of the new algorithm over the existing ones.

1. Introduction

In most real-world optimization problems, the environment changes along the time. Trajectory optimization of autonomous vehicles through complex environments, constructing financial trading models in changing market conditions, data mining in continuously updating databases and vehicle routing in dynamic traffic networks are examples of dynamic optimization problems (DOPs).

The dynamic nature of problems imposes great challenges to the basic optimization techniques. Some new capabilities are required to be included into the optimization algorithms, i.e. detecting the environment changes, and responding to the changed environment.

In order to mathematically model the dynamic changes, Eberhart and Shi [1] defined three dynamic environment categories. For the first, the optimum location changes in the search space. For the second category, the optimum location remains the same, but the optimum value changes. Finally, for the third category of dynamic environments, both the location and the value of the optimum change. Branke and Schmeck [2] suggest four criteria for dynamic environments; frequency of changes (how often the environment changes), severity of changes (how strongly the environment is changing from small changes to completely new situations), predictability of changes (whether there is a pattern or trend in changes) and the cycle length (how often the optimum returns or appears close to previous locations). Angeline [3] has introduced three different types of dynamic changes: linear (changes happen linearly), circular (changes are periodic), and random dynamic (changes are random).

In the behavioral study of biological swarms, an interesting aspect is the exhibition of complex behaviors despite the simplicity of individual behaviors. Inspired by these organisms, efficient algorithms like ant colony optimization (ACO) and particle swarm optimization (PSO) have been developed. These algorithms have been used successfully to solve difficult and complex real-world problems [4].

ACO was first proposed by Dorigo and colleagues [5,6] as a multi-agent approach to solve difficult combinatorial optimization problems. The main idea utilized in ACO has been adopted from the ant's pheromone trails-laying behavior, which is an indirect form of communication, mediated by modifications of the environment. Several adaptations of ACO to continuous optimization problems have been proposed in the literature [7–14], the last three are directly inspired from the ant's pheromone trails-laying behavior. The continuous interacting ant colony (CIAC) [9] has been extended to a dynamic hybrid CIAC (DHCIAC) [10]; in DHCIAC, ACO is used for global search and dynamic simplex for the local search. Tfaili et al. [11] also used DHCIAC over a set of test functions.

Ramos et al. [15] proposed the self-regulated swarm (SRS) algorithm, within which, ACO is hybridized with a simple evolutionary mechanism. This mechanism is inspired from the microbial life

* Corresponding author. Tel.: +98 21 66164610; fax: +98 21 66022731.
E-mail address: pourtak@sharif.edu (S.H. Pourtakdoust).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.11.005

through a direct reproduction procedure linked to the local environment features. It utilizes the advantageous characteristics of ACO via a collective pheromone laying process in the landscape. In addition, reproduction of new particles by two parents and death of old particles makes SRS able to self-regulate the exploratory swarm population and speed up the convergence to the global optimum.

PSO has also been empowered for DOPs via several improvements. In this regard, Carlisle and Dozier [16–18] proposed three strategies. They used sentry particles to detect the environmental changes. They also proposed that the social-only-model PSO is faster in tracking changing objectives than the full model. In social-only-model, only the social experience of the swarm is used to update the velocity, while in the full model both the social and personal experiences are utilized. The disadvantage of this model is that its reliability deteriorates faster than the full model, for large updating frequencies. When a dynamic change happens, the optimization problem varies in optimum location and/or optimum value. Therefore, they suggested resetting the personal best position of all particles to the current position, when a change is detected. As a result, all particles forget any experience that they have gained about the search space. Eventually they proposed to evaluate the personal best position after the environment change and to have it been reset if it is worse than the current position.

Eberhart and Shi [1] used the basic PSO but with a dynamic randomly selected inertia weight. This approach however, is shown not to be as efficient for high severity environments. Thus for increasing diversity, they suggested to reinitialize the swarm through resetting all particle positions to new random positions and recalculating the personal best and the neighborhood best positions. Hu and Eberhart [19] also suggested resetting the personal best positions combined with the partial reinitializing of the swarm. Reinitializing of the swarm increases diversity while resetting the personal best positions prevents return to the out-of-date positions.

A number of variable size versions of PSO have been developed for DOPs. Clerc [20] introduced the Cheap-PSO algorithm, which is based on environment changes; if there is no sufficient improvement in a particle's neighborhood, the particle reproduces a new particle within its neighborhood. On the other hand, if sufficient improvement is observed, the worst particle of that neighborhood is killed. According to this strategy the decrease of swarm size increases the probability of reproducing new particles. Coelho et al. [21] also utilized the Cheap-PSO in order to improve the efficiency of their algorithm. Koay and Srinivasan [22] developed a similar approach to dynamically change the swarm size by an analogy with the natural adaptation of amoeba to the environment. According to this strategy, when a particle finds itself in a potential optimum, the number of particles increases in that area.

Some researchers have divided the swarm into subpopulations instead of a single population. Parrott and Li [23] constructed multiple parallel subpopulations by a form of speciation and encouraged to simultaneously track multiple peaks by preventing overcrowding at peaks. Wang et al. [24] divided the whole population into three parts: explore-population, memory-population and exploit-population. Explore and exploit subpopulations perform the exploration and exploitation, respectively and good particles are stored in memory-population. Du and Li [25] proposed a new multi-strategy ensemble PSO (MEPSO) for dynamic optimization. In MEPSO, the particles are divided into two groups. Gaussian local search and differential mutation strategies are utilized over these two parts, respectively. Yan et al. have proposed a multi-population and diffusion univariate marginal distribution algorithm (MDUMDA) which in the multi-population approach is used to locate multiple local optima and the diffusion model is used to increase the diversity [26].

In the current study, a new hybrid PSO-based algorithm called dynamic hybrid PSO (DHPSO) is developed for continuous dynamic optimization problems. Inspiring from the microbial life, a new variable population size variant of PSO is introduced. It allows the new particles to be reproduced and the old ones to die. In this approach, three neighboring particles can probably reproduce new infants especially in high potential regions of the continuous search space, that are located near the local optimum points utilizing the quadratic interpolation method. An initial age, equal to zero, is assigned to all particles in the initialization. New born particles also posses this value of age. The age of the particles increase at a constant rate and a particle dies when its age reaches to a predetermined value. The adaptive population size and the use of quadratic interpolation to distribute the new born particles results in a rapid convergence of the algorithm towards the global optima. In DHPSO the population is divided into two subpopulations to perform exploitation and exploration, respectively. The first subpopulation includes a predetermined number of the best particles. The position updating of the second subpopulation is done using a differential mutation strategy, proposed in [25]. There is a good balance between the exploration and exploitation features of DHPSO and the experimental analyses show that DHPSO is efficient for various types of dynamic optimization problems with different types of dynamic environments such as linear, periodic and random.

The arrangement of this paper is as follows. In Section 2, the basic PSO algorithm is described. Section 3 is devoted to explanation of the new proposed approach. In Section 4, DHPSO results are compared with those of existing dynamic optimization algorithms. Conclusions are drawn in Section 5.

2. Basic PSO

PSO algorithm is inspired from the collective behavior of social animals like flock of birds, school of fishes and herd of sheep that have not a hierarchy system [4]. PSO was introduced by Kennedy and Eberhart [27] in 1995 as a stochastic population based search technique for solving optimization problems. In basic PSO the position of particles is updated according to their social and cognitive knowledge.

Let $\mathbf{x}_i(t)$ and $\mathbf{v}_i(t)$ denote the position and the velocity vector of particle i in the search space at time/iteration t, the position is updated in the current movement by:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t) \tag{1}$$

The velocity vector that reflects both the experimental knowledge of a particle and the socially exchanged information from other particles, is also updated as:

$$\mathbf{v}_i(t+1) = \omega\mathbf{v}_i(t) + c_1\, r_1(\mathbf{y}_i(t) - \mathbf{x}_i(t)) + c_2\, r_2(\hat{\mathbf{y}}_i(t) - \mathbf{x}_i(t)) \tag{2}$$

where c_1 and c_2 are the constant parameters, r_1 and r_2 are the random numbers between [0,1], ω is the inertia weight that controls the exploration/exploitation abilities of the swarm, $\mathbf{y}_i(t)$ is the personal best position of particle i and $\hat{\mathbf{y}}_i(t)$ represents the best position found by neighbors (other particles). Three terms of Eq. (2) are referred to as the inertia component, the cognitive component and the social component, respectively.

In PSO, each particle follows a leader; this feature is realized through the social term of Eq. (2). A leader can be global to all particles (global-best PSO) or local to a particle's neighborhood (local-best PSO) [4]. Due to larger interconnectivity, the global-best PSO converges faster than the local-best PSO. On the other hand the local-best PSO has larger diversity and it is less susceptible to being trapped in local minima.