G Model
ASOC 3409 1–22
ARTICLE IN PRESS

Applied Soft Computing xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

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

A new Reinforcement Learning-based Memetic Particle Swarm Optimizer

Hussein Samma [a,b], Chee Peng Lim [c,*], Junita Mohamad Saleh [a]

[a] School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Malaysia
[b] Faculty of Education, University of Aden, Shabwah, Yemen
[c] Centre for Intelligent Systems Research, Deakin University, Australia

ARTICLE INFO

Article history:
Received 1 August 2015
Received in revised form 2 January 2016
Accepted 5 January 2016
Available online xxx

Keywords:
Memetic algorithm
Particle Swarm Optimization
Reinforcement learning
Local search

ABSTRACT

Developing an effective memetic algorithm that integrates the Particle Swarm Optimization (PSO) algorithm and a local search method is a difficult task. The challenging issues include when the local search method should be called, the frequency of calling the local search method, as well as which particle should undergo the local search operations. Motivated by this challenge, we introduce a new Reinforcement Learning-based Memetic Particle Swarm Optimization (RLMPSO) model. Each particle is subject to five operations under the control of the Reinforcement Learning (RL) algorithm, i.e. exploration, convergence, high-jump, low-jump, and fine-tuning. These operations are executed by the particle according to the action generated by the RL algorithm. The proposed RLMPSO model is evaluated using four uni-modal and multi-modal benchmark problems, six composite benchmark problems, five shifted and rotated benchmark problems, as well as two benchmark application problems. The experimental results show that RLMPSO is useful, and it outperforms a number of state-of-the-art PSO-based algorithms.

1. Introduction

Memetic-based optimization algorithms have been used successfully in many applications, e.g. DNA sequence compression [1], flow shop scheduling [2], multi-robot path planning [3], wireless sensor networks [4], finance applications [5], image segmentation [6], and radar applications [7]. The main objective of developing memetic-based algorithms is to exploit the benefits of both global and local search methods and combine them into a single model. As an example, the Particle Swarm Optimization (PSO) algorithm is an effective global optimizer, and has been integrated with different local search methods to produce a number of memetic PSO-based models [1,2,8–11]. The resulting models combine the global search strength of PSO and the refinement capability of local search methods into a unified framework.

In the literature, many successful applications of memetic PSO-based models have been reported. In [1], a memetic model integrating PSO and an Intelligent Single Particle Optimizer (ISPO) [12] to solve the DNA sequence compression problem was presented. In [11], an adaptive memetic algorithm with PSO was developed and applied to the Latin hypercube design problem. Specifically, the standard PSO algorithm was adopted to perform the global search operations. It was integrated with a Lamarckian algorithm to perform the refinement operations. A hybrid model of PSO and a pattern-based local search method was studied in [10]. The resulting model was useful for parameter tuning of the Support Vector Machine (SVM). On the other hand, some studies indicate that PSO can be used for performing the local search operations in memetic models [5,13,14]. In [5], a hybrid model of PSO and genetic algorithm was introduced, whereby the PSO algorithm acted as a local search method. A hybrid shuffled frog-leaping algorithm and modified quantum-based PSO local search method was described in [13]. Recently, a hybrid model combining the differential evaluation algorithm and PSO was introduced. Again, PSO functioned as a local search method [14].

There are a lot of challenges in developing an effective memetic-based PSO model. The key challenges include when the local search method should be called, the frequency of calling the local search method, and which particle should undergo the local search operations. Indeed, the findings in [1] indicate that efficient management of the local search method in terms of time and frequency of calling has a significant impact on the performance. Besides these challenges, the standard PSO algorithm also suffers from several weaknesses, primarily the premature convergence and high computational cost problems. The first weakness is related to its fast premature convergence condition [15,16]. As pointed in [15,16], PSO can be trapped quickly in local optima at the beginning of the search process. The second limitation of PSO comes from its high

* Corresponding author. Tel.: +61 352273307.
E-mail addresses: chee.lim@deakin.edu.au, cplim123@yahoo.com (C.P. Lim).

http://dx.doi.org/10.1016/j.asoc.2016.01.006
1568-4946/

Please cite this article in press as: H. Samma, et al., A new Reinforcement Learning-based Memetic Particle Swarm Optimizer, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.006

computational cost. While a large particle population size gives a better swarm diversity capability, the computational cost becomes intensive too, e.g. each particle needs to undergo the fitness evaluation in every search cycle. This limitation of PSO has been reported in [17,18].

To mitigate the aforementioned problems, this study introduces a new reinforcement learning-based memetic PSO (RLMPSO) model. RL has been employed with standard PSO and other evolutionary algorithms [3,19]. An integration of RL and PSO was proposed by Grigoris [19]. Another recent study [3] employed RL for parameter tuning a differential evolution algorithm. On the other hand, RL worked independently from PSO in [6], whereby it was adopted to enhance the estimation of the objective function in noisy problems.

Comparing with the existing work in the literature, this study differs in the aspect that RL is embedded in RLMPSO to control the operation of each particle during the search process. Each particle, under the control of RL, performs one of the five possible operations [20], i.e. exploration, convergence, high-jump, low-jump, and fine-tuning. Moreover, each operation is given a reward or penalty according to its achievement. The proposed RMLPSO model has the following advantages:

(1) RLMPSO works with a small population size (typically 3 particles). It utilizes the ISPO (i.e. Intelligent Single Particle Optimizer) algorithm [12]. Additionally, it is enhanced with a total of five operations, i.e. exploration, convergence, high-jump, low-jump, and fine-tuning.
(2) The RL algorithm is embedded into RLMPSO to control the operation of each individual particle in the swarm. Specifically, RL adaptively switches the particle from one operation to another in accordance with the particle's achievement. Positive rewards are given to particles that have performed well, while penalties are imposed to non-performing particles.
(3) Each particle in RLMPSO evolves independently, e.g. one particle executes exploration, while others perform their respective operations.
(4) To minimize the computational cost of fine-tuning, two parameters are introduced i.e. delay (D) and cost (C). The delay parameter prevents fine-tuning (i.e., for local search) to be initiated at the beginning of the search process. The cost parameter controls the duration between each consecutive call of the fine-tuning operation.

Similar to RL, the idea of selecting the best performing operators from a set of alternatives has been comprehensively studied in the literature [21–24]. As an example, four PSO velocity updating strategies were used in [21]. A probability execution variable was assigned for each strategy, and the best operation was given a higher probability of selection. An evolutionary-based optimization algorithm with an ensemble of mutation operators was introduced in [22]. Each individual in the population would select a mutation strategy according to a probability distribution. Improved results were achieved with the ensemble strategy as compared with the single mutation strategy [25].

Differential Evolution (DE)-based methods with ensemble strategies were studied in [23,24,26]. In [23], an evolving DE model with an ensemble mutation strategy was presented. During the search process, DE randomly selected a mutation strategy with a random set of parameters to generate a new offspring. If the produced vector was better than the parent, the strategy would be retained; otherwise a new random mutation strategy with a new set of parameters would be generated [23]. The multi-objective DE algorithm with a pool of Neighbourhood Size (NS) parameter was presented in [24]. In particular, DE was developed using k NS candidates. The best NS value was adaptively selected from k candidates according to their historical performances. Improvements were achieved using k NS candidates as compared with only one candidate. Another DE-based model with an ensemble mutation strategy was presented in [26]. In particular, the population was randomly divided into three small sub-populations and one large sub-population. The three small sub-populations were executed for a specific number of Fitness Evaluations (FEs). Each sub-population was executed with a different mutation strategy, i.e. "current-to-pbest/1" and "current-to-rand/1", and "rand/1" [26]. A reward was computed as the ratio of fitness improvement to the total number of fitness calls consumed by each sub-population. After that, the large sub-population was executed with the setting of the best performing small sub-population. This process was repeated until the maximum number of FEs is met. In this case, the best mutation strategy could be selected dynamically during run time. The proposed model was able to outperform other DE variants.

The rest of this paper is organized as follows. In Section 2, an overview of PSO and its variants is given. The proposed RLMPSO model is explained in Section 3. In Section 4, a series of experiments to evaluate the effectiveness of RLMPSO using benchmark optimization problems is described. A summary of the research findings is presented in Section 5.

2. Particle Swarm Optimization and its variants

PSO was introduced by Kennedy and Eberhart about two decades ago [27]. The motivation of PSO is to mimic social interaction and search behaviours of animals, such as bird flocking and fish schooling. In general, PSO is represented by a swarm of N particles. Each particle in the swarm is associated with two vectors, i.e., the velocity (V) and position (X) vectors, as follows:

$$X_i = \left[d_i^1, d_i^2, d_i^3, \ldots, x_i^D\right] \tag{1}$$

$$V_i = \left[v_i^1, v_i^2, v_i^3, \ldots, v_i^D\right] \tag{2}$$

where D represents the dimension of the optimization problem and i denotes the particle number in the swarm. During the search process, the velocity and position vectors are updated as follows:

$$\begin{aligned} V_{i+1} = {} & w * V_i + c_1 * rand_{uniform}(pBest - X_i) \\ & + c_2 * rand_{uniform}(gBest - X_i) \end{aligned} \tag{3}$$

$$X_{i+1} = X_i + V_{i+1} \tag{4}$$

where w is the inertia weight, c_1 is the cognitive acceleration coefficient, c_2 is the social acceleration coefficient, $rand_{uniform}$ is a uniformly distributed random number within [0, 1], $pBest$ is the local best position achieved by a particular particle, and $gBest$ is the global best position achieved by the whole swarm.

As can be seen in Eq. (3), each particle's movement is affected by three components, namely its particle velocity (V_i), the distance from its local best ($pBest - X_i$), and the distance from the global best ($gBest - X_i$) in the swarm. Therefore, to control each component in Eq. (3), three parameters are used, i.e., w, c_1, and c_2. The suggested range of the inertia weight is $w \in [0.4, \ \ 0.9]$ [27]. It has been pointed out that w must be high in the exploration stage and low in the convergence stage [20]. On the other hand, the settings of c_1 and c_2 need to strike a balance between $pBest$ and $gBest$. As suggested in [20,21], c_1 must be higher than c_2 in the exploration stage, and the opposite in the convergence stage.

Since the introduction of the original PSO algorithm, many PSO variants have been put forward to improve its performance [1,2,4–11,13,14,17,18,28–48]. The main PSO-based algorithms available in the literature can be divided into five categories i.e.

Please cite this article in press as: H. Samma, et al., A new Reinforcement Learning-based Memetic Particle Swarm Optimizer, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.006