Engineering Applications of Artificial Intelligence 71 (2018) 275–292

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Interactive search algorithm: A new hybrid metaheuristic optimization algorithm

Ali Mortazavi [a], Vedat Toğan [b,*], Ayhan Nuhoğlu [c]

[a] Civil Engineering Department, Uşak University, Uşak, Turkey
[b] Civil Engineering Department, Karadeniz Technical University, Trabzon, Turkey
[c] Civil Engineering Department, Ege University, Izmir, Turkey

ARTICLE INFO

Keywords:
Hybrid optimization algorithm
Metaheuristics
Integrated particle swarm optimization
Teaching and learning based optimization
Numerical functions
Mechanical design problem

ABSTRACT

In this paper, a new hybrid optimization algorithm, called "Interactive Search Algorithm (ISA)" is proposed for the solution of the optimization problems. This algorithm modifies and combines affirmative features of two developed metaheuristic methods called Integrated Particle Swarm Optimization (iPSO) and Teaching and Learning Based Optimization (TLBO). ISA consists of two separate paradigms: (i) *Tracking* and (ii) *Interacting*. Tracking paradigm utilizes the information stored in the current agent's memory and two other important agents, the weighted and best agents, to guide the colony. On the other hand, interacting paradigm provides a pairwise interaction between agents to share their knowledge with each other. Each agent based on its tendency factor employs one of these two paradigms in each cycle of ISA to explore the search space. Additionally, rather than conventional penalty approach, ISA utilizes the improved fly-back approach to handle problem constraints. The search capability of the proposed method is tested on the number benchmark mathematical functions and constrained mechanical design problems as the real-world examples. Consequently, the achieved numerical results demonstrate that the proposed method is competitive with other well-established metaheuristic methods.

1. Introduction

Metaheuristic algorithms are non-gradient based methods which are generally deal with mathematical models inspired by a physical or social law or from a natural phenomenon (Cheng et al., 2016; Ding et al., 2016; Rao et al., 2011; Sönmez, 2011; Tang et al., 2015; Tejani et al., 2016). These methods are applicable for solving complex problems with continuous, discrete or even mixed search spaces (Daloğlu et al., 2016; Mortazavi et al., 2015; Stolpe, 2016).

One of the most practical and efficient methods belonged to this class of optimizers is the particle swarm optimizer (PSO) which is introduced by Kennedy and Eberhart (1995). This method mimics the collaborative behavior of the birds and fish colonies on finding food sources and avoiding from enemies. PSO is a population-based method which starts with a random initial swarm so that each agent (i.e. particle) of the swarm is a potential solution to the problem. Simplicity and efficiency of PSO have turned it into an overwhelming optimization tool in different fields of engineering (Gholizadeh, 2013; Kaveh and Javadi, 2014; Mortazavi and Togan, 2014). In spite of all affirmative specifications of PSO, it has own drawbacks confining its search capability. As reported in several research works, one of the most important shortcomings of this method is its premature convergence (Gholizadeh, 2013; El-Maleh et al., 2013; Nickabadi et al., 2011; Zhu et al., 2011). Another drawback of this method is its limitation to establish a proper balance between exploration and exploitation abilities of the algorithm (Shi and Eberhart, 1998). To relieve these drawbacks, several researchers try to modify its parameters or to combine this method with other techniques and approaches.

He and Wang (2007) presented a co-evolutionary particle swarm optimizer and tested it on solving some engineering problems. A particle swarm optimizer with adaptive population size was introduced by Chen and Zhao (2009). In this method, the number of population is gradually reduced to prevent non-essential objective function evaluations (OFEs) in the vicinity of the optimal point. Kaveh and Talatahari (2009) combined the particle swarm optimizer with ant colony optimizer to improve the performance of standard PSO. They tested their combined method on some structural optimization problems. An adaptive inertia weight strategy for PSO is provided by Nickabadi et al. (2011). In this method, the coefficient of prior velocity was adaptively adjusted in accordance with the progress of the algorithm. Gholizadeh (2013) tested the affirmative characteristics of cellular automata (CA) approach in

* Corresponding author.
E-mail address: togan@ktu.edu.tr (V. Toğan).

https://doi.org/10.1016/j.engappai.2018.03.003
Received 12 October 2017; Received in revised form 1 March 2018; Accepted 4 March 2018

combination with PSO to show its performance in solving the structural optimization problems. In this method, considering the CA mechanism, the velocity updating process of standard PSO is modified. Elsayed et al. (2014) proposed an adaptive particle swarm optimizer to provide a better balance between global and local search capabilities of standard PSO. A self-adaptive particle swarm optimizer was presented by Fan and Yan (2014). In this approach, a multiple velocity control scheme is introduced to improve the global search ability of PSO. Kaveh et al. (2014) introduced a chaotic version of particle swarm optimizer. In this method, the chaos theory is combined with the standard particle swarm optimizer to provide a new multi-phase PSO algorithm. All of these modifications through different search scenarios try to put forward an improved version of the standard PSO with higher search capability and efficiency.

Yet, in the most of these methods, the particles of the colony have not a considerable level of interaction with each other to perform an efficient local search. Also, in most of them, the main guidance task is on two main spots of the search space, they are the best prior location saved in the memory of each particle ($\mathbf{X}_i^P$), and the global best particle ($\mathbf{X}^G$). To provide a different optimizer tool, Rao et al. (2011) presented the teaching and learning based optimizer (TLBO). This method mathematically models the knowledge transfer process between teacher and students in a class. Lim and Mat Isa (2014a) combined TLBO with standard PSO to enhancing the search capability of the algorithm. They reported that this combined method shows better performance than the standard PSO. This hybrid method is mostly based on the TLBO approach and consequently evaluates each particle twice per each iteration. However, in the several fields of science and engineering objective function evaluations are the most time-consuming steps of the optimization process (e.g. performing analyses based on the finite element method). Therefore, for this class of problems, the number of objective function evaluations (OFEs) plays a very important role in the selection priority of an optimization algorithm compared to the others (Mortazavi et al., 2016b). Thus, such a multi-phase optimizer might not show adequate efficiency on this class of complex problems in comparison with the single-phase optimizers (Mortazavi et al., 2016b).

However, generally, a metaheuristic method should be efficiently applicable as an optimization tool for solving different type of problems available in the engineering and mathematical fields. In this respect, Mortazavi and Toğan (2016) proposed an integrated particle swarm optimization (iPSO) and tested it on complex structural optimization problems. iPSO was equipped with an improved fly-back approach to be able to handle both non-constrained and constrained optimization problems. This method utilizes the weighted particle in both constraint handling and velocity updating steps. It shows acceptable global search capability (Mortazavi and Toğan, 2017). However, the local search ability of this method still can be improved via generating more interaction among the agents in the colony especially in the vicinity of local optima.

To meet this aim, in the current study a new optimization technique so-called interactive search algorithm (ISA) is presented. Based on the numerical implementations carried out in the current work, ISA provides more efficient search strategy in comparison with its parental methods. This new method modifies two existing optimization methods, iPSO and TLBO, and hybridizes them to put forward a new efficient search scheme. Similar to iPSO, ISA uses weighted particle as a specific efficient particle (i.e. in addition to $\mathbf{X}_i^P$ and $\mathbf{X}^G$ particles) to guide the swarm. But, in contrast with two phases analyses (e.g. like TLBO and its variants), ISA still is a single phase approach applying two different paradigms to update agents' location. It utilizes a tendency factor to provide a balance between local and global search abilities.

ISA uses a distinct coefficient related to each variable (i.e. using the vector of random coefficients). This implementation gives individual random decision factor for each component of the agent, and consequently, the agents can move inside the search space in more democratic fashion. In addition, ISA utilizes the improved fly-back (IFB) technique to handle any available constraints and variables' boundaries. Thus, ISA can handle both constrained and non-constrained problems.

The remind of the current article is arranged as follows: Section 2 provides the formulation of interactive search algorithm (ISA) and its basic methods as integrated particle swarm optimization (iPSO) and teaching and learning based optimizer (TLBO). In Section 3, the performance of the proposed ISA is evaluated through several benchmark problems. Section 4 is devoted to a discussion about ISA and selected optimization methods. Finally, a conclusion about the current study is provided in the last section.

2. Formulation of optimization techniques

Since iPSO and TLBO are two basic methods for the proposed ISA, in this section primarily a brief explanation of them is provided. Subsequently, the proposed approach is described in detail.

2.1. Integrated particle swarm optimization (iPSO)

In the standard PSO, the current particle (e.g. ith particle) navigation is upon two significant locations of the search space specified by $\mathbf{X}_i^P$ and $\mathbf{X}^G$. However, if the current particle lies so close to any of these two significant particles, their guidance roles are highly decreased or even vanished. To resolve this problem, iPSO incorporates a third particle, so-called weighted particle ($\mathbf{X}^W$), into the velocity updating formulation (Mortazavi and Toğan, 2016). The weighted particle is the weighted average of all available particles in the colony. Based on this definition, $\mathbf{X}^W$ is formulated as below:

$$\mathbf{X}^W = \sum_{i=1}^{M} \bar{c}_i^w \mathbf{X}_i^P \tag{1.1}$$

$$\bar{c}_i^w = \left(\hat{c}_i^w \Big/ \sum_{i=1}^{M} \hat{c}_i^w \right)$$

in which

$$\hat{c}_i^w = \frac{\max_{1\le f\le M}\left(f(\mathbf{X}_{kw}^P)\right) - f(\mathbf{X}_i^P) + \varepsilon}{\max_{1\le kw\le M}\left(f(\mathbf{X}_{kw}^P)\right) - \min_{1\le kw\le M}\left(f(\mathbf{X}_{kw}^P)\right) + \varepsilon}, \quad i = 1, 2, \ldots, M \tag{1.2}$$

where M is the number of all particles, $\mathbf{X}^W$ denotes the position vector of the weighted particle, $\hat{c}_i^w$ is the coefficient shows influence of each particle on the weighted particle, $f(.)$ is the objective function of the optimization problem, $\max_{1\le kw\le M}\left(f(\mathbf{X}_{kw}^P)\right)$ and $\min_{1\le kw\le M}\left(f(\mathbf{X}_{kw}^P)\right)$, respectively, are the worst and best objective values among all particles, ε is a small positive number to avoid zero division failure, which is set as 0.0001.

iPSO tries to provide a new spot to navigate the swarm via applying the weighted particle. It is remarkable that the weighted particle is the weighted gravity of the colony. So, when $\mathbf{X}^G$ is trapped into local minima, $\mathbf{X}^W$ as a significant particle prevents other particles to fly just toward the grabbed local optima. Subsequently, iPSO algorithm is mathematically formulated as follows:

$$^{t+1}\mathbf{v}_i = \varphi_{4i}\left({}^t\mathbf{X}_w - {}^t\mathbf{X}_i\right) \qquad \text{if } \mathrm{rand}_{0i} \le \alpha \tag{2.1}$$

$$^{t+1}\mathbf{v}_i = w_i \times {}^t\mathbf{v}_i + \left(\varphi_{1i} + \varphi_{2i} + \varphi_{3i}\right)\left({}^t\mathbf{X}_j^P - {}^t\mathbf{X}_i\right) + \varphi_{2i}\left({}^t\mathbf{X}^G - {}^t\mathbf{X}_j^P\right) + \varphi_{3i}\left({}^t\mathbf{X}^w - {}^t\mathbf{X}_j^P\right) \qquad \text{if } \mathrm{rand}_{0i} > \alpha \tag{2.2}$$

$$^{t+1}\mathbf{X}_i = {}^t\mathbf{X}_i + {}^{t+1}\mathbf{v}_i \tag{2.3}$$

where superscripts 't' and '$t+1$' indicate the current and following iterations, respectively, $^{t+1}\mathbf{v}_i$ is the updated velocity, w_i is the factor of inertia. Also, for ith particle $\varphi_{ki} = C_k \times \mathrm{rand}_{ki}$ and $C_1 = -(\varphi_{2i} + \varphi_{3i})$, $C_2 = 2$, $C_3 = 1$, and $C_4 = 2$ are acceleration coefficients, and rand_{ki}, $k \in 0, 1, 2, 3, 4$, is the random number selected from the interval [0, 1], $^t\mathbf{X}_j^P$ is the randomly selected particle from the current $\mathbf{X}^P$ matrix. Also, $^t\mathbf{X}^G$ is the global best particle, $^{t+1}\mathbf{X}_i$ and $^t\mathbf{X}_i$ are the updated and current positions of the ith particle, respectively. $^t\mathbf{X}_W$ is the weighted particle calculated for current step.