Engineering Applications of Artificial Intelligence 26 (2013) 1291–1300

Contents lists available at SciVerse ScienceDirect

Engineering Applications of Artificial Intelligence

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

Multi-objective optimization using teaching-learning-based optimization algorithm

Feng Zou [a,b], Lei Wang [a,*], Xinhong Hei [a], Debao Chen [b], Bin Wang [a]

[a] School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China
[b] School of Physics and Electronic Information, Huaibei Normal University, Huaibei 235000, China

A R T I C L E I N F O

Article history:
Received 8 May 2012
Received in revised form
30 October 2012
Accepted 12 November 2012
Available online 21 December 2012

Keywords:
Teaching-learning-based optimization
Multi-objective optimization
Nondominated sorting
Crowding distance

A B S T R A C T

Two major goals in multi-objective optimization are to obtain a set of nondominated solutions as closely as possible to the true Pareto front (PF) and maintain a well-distributed solution set along the Pareto front. In this paper, we propose a teaching-learning-based optimization (TLBO) algorithm for multi-objective optimization problems (MOPs). In our algorithm, we adopt the nondominated sorting concept and the mechanism of crowding distance computation. The teacher of the learners is selected from among current nondominated solutions with the highest crowding distance values and the centroid of the nondominated solutions from current archive is selected as the Mean of the learners. The performance of proposed algorithm is investigated on a set of some benchmark problems and real life application problems and the results show that the proposed algorithm is a challenging method for multi-objective algorithms.

1. Introduction

In many cases, most engineering design problems, such as investment decision, city programming, program management, university timetable, control system design, the objectives present some degree of conflict among them in nature. That is to say, one objective cannot be improved without deterioration of at least another objective. These problems are called multi-objective optimization problems (MOPs), which have a set of several optimal solutions known as Pareto optimal solutions (Deb, 2001). Therefore, multi-objective optimization also differs from single-objective optimization in that the former is composed of two different tasks to solve the problem: a searching task whose goal is to find Pareto optimal solutions, and a decision making task in which a most preferred solution is chosen from the set of Pareto optimal solutions. In other words, two major tasks in multi-objective optimization are to obtain a set of nondominated solutions as closely as possible to the true Pareto front (PF) and maintain a well-distributed solution set along the Pareto front. Hence, the goal of multi-objective optimization methods is to find a set of good trade-off solutions from which the decision maker want to select one.

In order to solve multi-objective problem, V. Pareto offers the most common definition of optimum in multi-objective optimization in 1896. In 1984, Schaffer (Schaffer, 1985) proposed the first actual implementation of evolutionary algorithms to solve multi-objective problems, which it is now called multi-objective evolutionary algorithm (MOEA). Evolutionary computation techniques are suitable for multi-objective optimizations because of the fact that Evolutionary Algorithm (EA) deals with a set of solutions which help in the generation of well distributed Pareto optimal front more quickly and efficiently in comparison to the classical techniques. Since 1984, many researchers have proposed their own multi-objective evolutionary algorithms (MOEAs). Representative multi-objective evolutionary methods, such as NPGA (Horn et al., 1994), NPGA2 (Erickson and Mayer, 2001), NSGA (Srinivas and Deb, 1994), NSGA-II (Deb et al., 2000), SPEA (Zitzler and Thiele, 1999), SPEA2 (Knowles and Corne, 2000), MOPSO (Coello Coello et al., 2004), MODE (Xue and Sanderson, 2003), MOSaDE (Huang et al., 2009), VEDA (Larrañaga and Lozano, 2001), MOHBOA (Pelikan et al., 2005), RM-MEDA (Qingfu, 2008), and MOEA-D (Zhang and Li,. 2007), are utilized to optimize several objectives simultaneously and some efficient results are derived.

In this paper, we propose a teaching-learning-based optimization (TLBO) algorithm based on the nondominated sorting and crowding distance sorting for MOPs. In our algorithm, we adopt the nondominated sorting concept used in NSGA-II, where the entire population is sorted into various non-domination levels. This provides the means for selecting the individuals in the better fronts, hence providing the necessary selection pressure to push the population towards PF. To maintain the diversity of the current best solutions in the external archive, the mechanism of crowding distance computation used in NSGA-II is adopted.

* Corresponding author. Tel.: +86 29 8231 2087.
E-mail address: wangleei@hotmail.com (L. Wang).

0952-1976/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2012.11.006

The teacher of the learners is selected from among current nondominated solutions with the highest crowding distance values and the centroid of the nondominated solutions from current archive is selected as the Mean of the learns. The performance of proposed algorithm is investigated on a set of some unconstrained and constrained benchmark problems and the results show that the proposed algorithm is a challenging method for multi-objective algorithms.

The remainder of this paper is organized as follows. The description of teaching-learning-based optimization algorithm is introduced in Sections 2 and 3. describes the proposed algorithm. Comparison and analysis of experimental results of some unconstrained test problems are shown in Section 4. Some constrained optimization examples are shown in Section 5 and some conclusions are given in Section 6.

2. Teaching-learning-based optimization

Rao et al. (2011, 2012) first proposed a novel teaching-learning-based optimization (TLBO) inspired from the philosophy of teaching and learning. TLBO has emerged as one of the simple and efficient techniques for solving single-objective benchmark problems and real life application problems in which it has been empirically shown to perform well on many optimization problems (Rao et al., 2011a, 2011b, 2012; Rao, 2012; Rao and Patel, 2011; Rao and Kalyankar, 2012; Togan, 2012). These are precisely the characteristics of TLBO that make it attractive to extend it to solve MOPs (Rao and Patel, 2012a, 2012b; Niknam and Golestaneh, 2012; Niknam et al., 2012; Satapathy et al., 2012).

The TLBO method is based on the effect of the influence of a teacher on the output of learners in a class which is considered in terms of results or grades. The teacher is generally considered as a highly learned person who shares his or her knowledge with the learners. The quality of a teacher affects the outcome of learners. It is obvious that a good teacher trains learners such that they can have better results in terms of their marks or grades. Moreover, learners also learn from interaction between themselves, which also helps in their results. Like other nature-inspired algorithms, TLBO is also a population based method which uses a population of solutions to proceed to the global solution. For TLBO, the population is considered as a group of learners or a class of learners. In optimization algorithms, the population consists of different design variables. In TLBO, different design variables will be analogous to different subjects offered to learners and the learners' result is analogous to the "fitness", as in other population based optimization techniques. The teacher is considered as the best solution obtained so far.

The process of working of TLBO is divided into two parts. The first part consists of "Teacher Phase" and the second part consists of "Learner Phase". The "Teacher Phase" means learning from the teacher and the "Learner Phase" means learning through the interaction between learners.

2.1. Teaching phase

A good teacher is one who brings his or her learners up to his or her level in terms of knowledge. But in practice this is not possible and a teacher can only move the mean of a class up to some extent depending on the capability of the class. This follows a random process depending on many factors.

Let M_i be the mean and T_i be the teacher at any iteration *i*. T_i will try to move mean M_i towards its own level, so now the new mean will be T_i designated as M_{new}. The solution is updated according to the difference between the existing and the new mean given by

$$\text{Difference\_Mean}_i = r_i(M_{new} - T_F M_i) \tag{1}$$

where T_F is a teaching factor that decides the value of mean to be changed, and r_i is a random number in the range [0, 1]. The value of T_F can be either 1 or 2, which is again a heuristic step and decided randomly with equal probability as

$$\text{TF} = \text{round}[1 + \text{rand}(0,1)] \tag{2}$$

This difference modifies the existing solution according to the following expression

$$X_{new,i} = X_i + \text{Difference\_Mean}_i \tag{3}$$

Learner modification is expressed as (*Pn* is the number of learners),

T_F=round[1+rand(0,1)]
for p=1:P_n
    $\text{Difference\_Mean}_i = r_i \times (M_{new} - T_F \times M\text{i})$
    $X_{new,p} = X_p + \text{Difference\_Mean}_i$
endfor
Accept X_{new} if it gives a better function value

2.2. Learning phase

Learners increase their knowledge by two different means: one through input from the teacher and other through interaction between themselves. A learner interacts randomly with other learners with the help of group discussions, presentations, formal communications, etc. A learner learns something new if the other learner has more knowledge than him or her. Learner modification is expressed as (*Pn* is the number of learners),

for I =1:P_n
    Randomly select one learner X_j, such that $i \neq j$
    if $f(X_i) < f(X_j)$
        $X_{new,i} = X_{old,i} + r_i \times (X_i - X_j)$
    else
        $X_{new,i} = X_{old,i} + r_i \times (X_j - X_i)$
    endif
endfor
Accept X_{new} if it gives a better functions value

2.3. The sketch of TLBO algorithm

As explained above, the step-wise procedure for the implementation of TLBO can be summarized as follows.

Step 1: Define the optimization problem and initialize the optimization parameters.
Step 2: Initialize the population.
Step 3: Teacher phase. Learners is learning from the teacher.
Step 4: Learner phase. Learners increase their knowledge with the help of their mutual interaction.
Step 5: Termination criterion. Stop if the maximum generation number is achieved; otherwise repeat from Step 3.

3. Description of the proposed algorithm

In the current study, we have concentrated our work on teaching-learning-based optimization (TLBO) for solving MOPs. In this paper, we proposed an improved TLBO called multi-objective teaching-learning-based optimization (MOTLBO). In our MOTLBO, we use an external archive to keep the best solutions obtained so far. We adopt the nondominated sorting concept used in NSGA-II