ARTICLE IN PRESS

Engineering Science and Technology, an International Journal xxx (2017) xxx–xxx

HOSTED BY ELSEVIER

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

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

Full Length Article

A new adaptive differential evolution optimization algorithm based on fuzzy inference system

M. Salehpour, A. Jamali *, A. Bagheri, N. Nariman-zadeh

Department of Mechanical Engineering, Faculty of Engineering, University of Guilan, Rasht 3756, Iran

ARTICLE INFO

Article history:
Received 17 October 2016
Revised 8 January 2017
Accepted 16 January 2017
Available online xxxx

Keywords:
Differential evolution
Fuzzy logic
Mutation factor
Population diversity
Optimization
Vehicle vibration model

ABSTRACT

In this paper, a new version of differential evolution (DE) with adaptive mutation factor has been proposed for solving complex optimization problems. The proposed algorithm uses fuzzy logic inference system to dynamically tune the mutation factor of DE and improve its exploration and exploitation. In this way, two factors, named, the number of generation and population diversity are considered as inputs and, one factor, named, the mutation factor as output of the fuzzy logic inference system. The performance of the suggested approach has been tested firstly by using some popular single objective test functions. It has been shown that the proposed method finds better solutions than the classical differential evolution and also the convergence rate of that is really fast. Secondly, a five degree of freedom vehicle vibration model is chosen to be optimally designed by the aforesaid proposed approach. Comparison of the obtained results with those in the literature demonstrates the superiority of the results of this work.

1. Introduction

Evolutionary algorithms (EAs), motivated by the natural evolution of species [1], are popular for their ability to handle nonlinear and complex optimization problems [2]. EAs are often called meta-heuristic approaches because the structure of such optimization process is based on the discovering issues from the experiences of real life. Most of EAs use random components during the search process, therefore they belong to the category of the stochastic optimization approaches [3–4]. As long as meta-heuristic algorithms are intrinsically non-deterministic and not sensitive to the continuity and differentiability of the objective functions, use of such methods contains broad range of complex optimization problems [4]. In addition, the stochastic global optimizations can discover global minimum without trapping in the local minima [3].

One of the recently developed meta-heuristic methods is differential evolution (DE) presented by Storn and Price [5,6] is a fast and robust [7,8] stochastic metaheuristic algorithm which needs not any gradient-based data. In addition, it is a population-based and derivative-free method which can be applied for solving non-convex, nonlinear, non-differentiable and multimodal problems [7]. Besides, real numbers are applied in DE as solution strings, so no encoding and decoding is required [9]. Empirical results have shown that DE has good convergence characteristics and overcomes other popular EAs [10]. DE uses three main operators, namely, mutation, crossover and selection, respectively [5,6]. Due to its simple structure, simple implementation, fast convergence and robustness, DE has been widely applied to the optimization problems arising in some fields of science and engineering, such as robot control [11], controller design [12], data clustering [13], optimal design [14], microbiology [15], image processing [16] and so forth.

It is very important to notice that the behavior of DE largely depends on the two parameters named mutation and crossover [9,17–19]. As widely discussed in the literature, a larger mutation factor (F) can be effectual in global search; on the other hand, a smaller one can hasten the convergence rate. In addition, the larger crossover probability (C_r) leads to the higher diversity of the population but, a smaller one causes local exploitation [20]. Consequently, it could be readily observed that selecting a proper control parameter is considerably an important issue. The mutation factor is the most sensitive one. $F \in [0,2]$ is allowable in theory [9,17,21] but $F \in (0,1)$ is more effectual in reality. As a matter of fact, $F \in [0.4, 0.95]$ seems a proper range while a good first choice can be $F \in [0.7, 0.9]$ [9]. The crossover probability as $Cr \in [0,1]$ is acceptable in theory [17], but $Cr \in [0.1, 0.8]$ sounds a proper range, and the first choice which can be convenient to be used is $Cr = 0.5$ [9].

Even though DE is a good and fast algorithm, but it has some deficiencies [22]. Global exploration ability of DE seems proper

* Corresponding author.
E-mail address: ali.jamali@guilan.ac.ir (A. Jamali).
Peer review under responsibility of Karabuk University.

http://dx.doi.org/10.1016/j.jestch.2017.01.004
2215-0986/© 2017 Karabuk University. Publishing services by Elsevier B.V.

enough which it can recognize the feasible region of the global optimum, but its local exploitation one is considered slow at fine-tuning the solution [1,18,23–25]. In addition, DE suffers from loss of diversity which happens while the population stagnation or premature convergence occurs [22]. Besides, DE is a parameter dependent algorithm, and, therefore it is a difficult task to adapt its control parameter for various problems [23,24]. Furthermore, by increasing the dimensionality of the optimization problem, the efficacy of the algorithm debases [23,24,26]. Consequently, the aforementioned drawbacks make the scholars find methods to improve performance and increase the effectiveness of DE. Such modifications, not only for DE but also for other EAs, can generally classified into two main categories. The first one are based on the tuning or controlling the control parameters [27] of DE, and the second one concentrates on the hybridization of DE with other optimization methods such as particle swarm optimization [28] or so forth.

In terms of tuning the control parameters of DE (as done in this work), recently, some methods, based on the dynamical adjustment of DE have been revealed. Fuzzy logic plays a pivotal role in this category. As a matter of fact, fuzzy logic is a knowledge-based system considering a set of fuzzy rules proposed by Zadeh [29] that shows the relationship between the input(s) and output (s) of the system. Some hybridization of differential evolution and fuzzy logic is reviewed here.

Patricia Ochoa et al. [30] proposed a method based on the combination of fuzzy logic and DE to dynamical adjustment of the mutation parameter. In this case, fuzzy logic provides optimal parameters for improving the efficiency of DE. It has been shown that the differential evolution algorithm with Fuzzy F (mutation factor) Decrease performs better than the differential evolution algorithm with F increase. Liu and Lampinen [31] presented an approach based on the hybrid application of the differential evolution algorithm and fuzzy logic. The aim of this methodology is to dynamically adapt the population size of the search process. The obtained results have shown that the adaptive population size might lead to the higher convergence velocity and, of course, decrease the number of the model assessments. After that, Liu and Lampinen [32], suggested a fuzzy adaptive differential evolution algorithm to adjust the mutation and crossover parameters using a set of standard test functions. It has been shown that the proposed method works better than the original DE when the dimensionality of the problem is high. Furthermore, this method was applied to hasten the convergence rate of DE by the use of adaptive parameters.

More description of the hybrid usage of EAs and fuzzy logic can be seen in [33–38].

In this paper, fuzzy logic inference system is used to dynamically adapt the mutation factor of conventional differential evolution. In this way, two main factors, namely, number of generation and population diversity of each generation which may affect the exploration and exploitation ability of the algorithm are selected as inputs and mutation factor as output of the fuzzy logic inference system. The ability of the proposed algorithm for resolving single optimization problems is appraised by using six well-known benchmark functions. Afterwards, the proposed method has been used for the single optimization of the five degree of freedom vehicle vibration model for analyzing the performance of the proposed method on the engineering problems. The obtained results show the very good behavior of the proposed method, and also, comparison with the ones reported in the literature (two categories of previous works used here which contains one work related to benchmark functions [30] and two works related to the vehicle vibration model [39,40]) demonstrates the superiority of the suggested method of this work.

2. Differential evolution

Like all other evolutionary algorithms, DE uses a population of potential solutions and genetic operators to seek for the optimums through feasible search space. For each solution vector indicated by x_i, at any generation G, x_i can be shown as:

$$\boldsymbol{x_i^G} = (x_{1,i}^G, x_{2,i}^G, \ldots, x_{d,i}^G), \quad i = 1, 2, \ldots, n \tag{1}$$

in which, n indicates the number of population which is composed of d-elements. This vector is called chromosome or genome.

Differential evolution comprises three major operators, namely, mutation, crossover and selection. Initially a population of n solutions is randomly generated using uniform distribution, and then the aforesaid operators are applied to the population to produce next generation. In this way, for each vector x_i, mutation scheme is carried out firstly. For each vector x_i at any generation, three distinct vectors x_{r_1}, x_{r_2}, and x_{r_3} are randomly selected, and then a so-called mutant vector (perturbed or donor vector) is generated by applying the mutation scheme:

$$v_i^G = x_{r_1}^G + F(x_{r_2}^G - x_{r_3}^G), \quad r_1 \neq r_2 \neq r_3 \neq i \tag{2}$$

The constant $F \in [0, 2]$ [9,17,21] in the previously mentioned equation, is a mutation factor (scale factor or differential weight) which affects the diversity of the set of mutant vectors and helps to manage the trade-off between exploration and exploitation of the search process [21]. Essentially, in theory $F \in [0, 2]$, but in practice, a scheme with $F \in [0, 1]$ is more efficient and stable, and it seems that it is used by almost all the studies in the literature.

It is easily seen that the perturbation term indicated by $\delta = F(x_{r_2} - x_{r_3})$ is added to the base vector indicated by x_{r1} to generate a mutant vector v_i, and as a result, such perturbation defines the direction and length of the search space [21].

Secondly, the crossover operator amalgamates the mutant vector (v_i^G) with the parent vector (target vector) (x_i^G) to create a so-called trial vector (u_i^G). The crossover scheme is classified into two forms, namely, binomial and exponential. In the binomial scheme, the trial vector is generated according to the next probabilistic formula:

$$u_{j,i}^G = \begin{cases} v_{j,i}^G & if r_i \leqslant C_r \; or \; j = J_r, \\ x_{j,i}^G & Otherwise. \end{cases} \quad j = 1, 2, \ldots, d \tag{3}$$

in which $\mathrm{r_i}$ is a random number extracted from the interval $[0, 1]$ [17], $\mathrm{J_r}$ is used to guarantee that $\mathrm{u}_i^G \neq x_i^G$, which may improve the efficiency of the searching ability of the algorithm. In addition, $Cr \in [0, 1]$ [17] is the crossover probability (crossover rate) as mentioned earlier.

In the exponential scheme, a section of the mutant vector is chosen, and this section commences with an integer k and length L randomly selected from the intervals $\{1, 2, \ldots, \mathrm{n}\}$, and the trial vector is created according to the formula below:

$$u_{j,i}^G = \begin{cases} v_{j,i}^G & if \; j\{k, <k+1>_n, \ldots, <k+L-1>_n\} \quad , \\ x_{j,i}^G & Otherwise. \end{cases} \quad j = \{1, 2, \ldots, n\} \tag{4}$$

The main difference between binomial and exponential crossover is the fact that while in the binomial case the components inherited from the mutant vector are arbitrarily selected, in the case of exponential crossover they form one or two compact subsequences. The influence of this difference on the performance of differential evolution is not fully understood yet. Choosing

Please cite this article in press as: M. Salehpour et al., A new adaptive differential evolution optimization algorithm based on fuzzy inference system, Eng. Sci. Tech., Int. J. (2017), http://dx.doi.org/10.1016/j.jestch.2017.01.004