Swarm and Evolutionary Computation 37 (2017) 1–26

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

A self-adaptive multi-population based Jaya algorithm for engineering optimization

R. Venkata Rao*, Ankit Saroj

Mech. Engg. Dept., Saradar Vallabhbhai National Institute of Technology, Surat 395007, India

ARTICLE INFO

Keywords:
Multi-population Jaya algorithm
Engineering optimization
CEC 2015
Multimodal unconstrained benchmark problems
Plate-fin heat exchanger

ABSTRACT

Multi-population algorithms have been widely used for solving the real-world problems. However, it is not easy to get the number of sub-populations to be used for a given problem. This work proposes a self-adaptive multi-population based Jaya (SAMP-Jaya) algorithm for solving the constrained and unconstrained numerical and engineering optimization problems. The Jaya algorithm is a recently proposed advanced optimization algorithm and is not having any algorithmic-specific parameters to be tuned except the common control parameters of population size and the number of iterations. The search mechanism of the Jaya algorithm is upgraded in this paper by using the multi-population search scheme. It uses an adaptive scheme for dividing the population into sub-populations which control the exploration and exploitation rates of the search process based on the problem landscape.

The robustness of the proposed SAMP-Jaya algorithm is tested on 15 CEC 2015 unconstrained benchmark problems in addition to 15 unconstrained and 10 constrained standard benchmark problems taken from the literature. The Friedman rank test is conducted in order to compare the performance of the algorithms. It has obtained first rank among six algorithms for 15 CEC 2015 unconstrained problems with the average scores of 1.4 and 1.9 for 10-dimension and 30-dimension problems respectively. Also, the proposed algorithm has obtained first rank for 15 unimodal and multimodal unconstrained benchmark problems with the average scores of 1.7667 and 2.2667 with 50000 and 200000 function evaluations respectively. The performance of the proposed algorithm is further compared with the other latest algorithms such as across neighborhood search (ANS) optimization algorithm, multi-population ensemble of mutation differential evolution (MEMDE), social learning particle swarm optimization algorithm (SL-PSO), competitive swarm optimizer (CSO) and it is found that the performance of the proposed algorithm is better in more than 65% cases. Furthermore, the proposed algorithm is used for solving a case study of the entropy generation minimization of a plate-fin heat exchanger (PFHE). It is found that the number of entropy generation units is reduced by 12.73%, 3.5% and 9.6% using the proposed algorithm as compared to the designs given by genetic algorithm (GA), particle swarm optimization (PSO) and cuckoo search algorithm (CSA) respectively. Thus the computational experiments have proved the effectiveness of the proposed algorithm for solving engineering optimization problems.

1. Introduction

Solving the complex optimization problems in the limited time is an indispensable issue in the field of engineering optimization. Due to the complexity of the problems the conventional methods become tedious and time consuming and these approaches do not guarantee the achievement of the optimal solution. Therefore, metaheuristic based computational methods are developed. These methods are capable of achieving the global or near global optimum solution with less information about the problems. Some of the well-known metaheuristic optimization algorithms are: genetic algorithm (GA) and its variants (real coded GA, parallel GA, hybrid interval GA, etc.), simulated annealing (SA) algorithm, tabu search (TS), ant colony optimization (ACO), particle swarm optimization (PSO) and its variants(e.g. niching PSO, culture-based PSO, aging theory inspired PSO, etc.), differential evolution (DE) and its variants (e.g. DE with multi-population ensemble, DE with self-adapting control parameter, DE with optimal external archive, etc.), harmony search algorithm (HS), non-dominated sorting genetic algorithm (NSGA-II), artificial bee colony (ABC) algorithm, imperialist competitive algorithm (ICA), biogeography based optimization (BBO), firefly algorithm (FFA), gravitational search algorithm (GSA), bat algorithm(BA), cuckoo

* Corresponding author.
E-mail address: ravipudirao@gmail.com (R. Venkata Rao).

http://dx.doi.org/10.1016/j.swevo.2017.04.008
Received 18 October 2016; Received in revised form 5 March 2017; Accepted 28 April 2017
Available online 04 May 2017
2210-6502/

search (CS) etc. Several metaheuristic algorithms are proposed in the last decade. Some prominent algorithms are: galaxy-based search algorithm, spiral optimization, teaching-learning-based optimization (TLBO), differential search algorithm, cuckoo search algorithm (CSA), colliding bodies optimization algorithm, centripetal accelerated particle swarm optimization algorithm, crisscross optimization algorithm, Lons motion algorithm, ant lion optimization, cat swarm optimization, etc. and hybrid algorithms [1–4].

The advanced optimization algorithms have their own merits but they require tuning of their specific parameters. For example, SA algorithm needs initial annealing temperature and cooling schedule. GA needs proper setting of crossover probability, mutation probability, selection operator, etc.; NSGA-II needs crossover probability, mutation probability, SBX parameter, mutation parameter etc.; PSO needs inertia weight and social and cognitive parameters; HSA needs harmony memory consideration rate, number of improvisations, etc.; BBO algorithm requires immigration rate, emigration rate, etc. Similarly, ICA, DE and other algorithms (except TLBO algorithm) have respective specific parameters to be set for effective execution. These parameters are called algorithm-specific parameters and need to be controlled other than the common control parameters of number of iterations and population size. All population based algorithms need to tune the common control parameters but the algorithm-specific parameters are specific to the particular algorithm and these are also to be tuned as mentioned above.

The performance of the optimization algorithms is much affected by the algorithm-specific parameters. Increase in the computational cost or tending towards the local optimal solution is caused by the improper tuning of these parameters. Hence, to overcome the problem of tuning of algorithm-specific parameters, TLBO algorithm was proposed which is an algorithm-specific parameter less algorithm [2,4]. Keeping in the view of the good performance of the TLBO algorithm, another algorithm-specific parameter less algorithm has been recently proposed and it is named as Jaya algorithm [5].

Multi-population based advanced optimization methods are used for improving the diversity of search by splitting the entire population into groups (sub-populations) and allocating these throughout the search space so that the problem changes can be detected effectively. This basic idea is used for keeping the diversity of the search process by allocating different sub-populations to different areas. Each population is subjected to either diversifying or intensifying the search processes of the algorithm [6,7]. The interaction between the sub-populations takes place by means of a merge and divide process whenever there is a change in the solution is observed. The multi-population approaches are found effective while dealing with various problems and these have outperformed the existing fixed population size methods for different problems.

A self-organizing scout's multi-population evolutionary algorithm was proposed for the dynamic optimization problems [8]. A multi-swarm PSO algorithm was proposed by Li and Yang [9]. A clustering-based PSO was proposed by Yang and Li [10]. A multi-population HSA was proposed by Turky and Abdullah [11]. A multiple teacher based TLBO was proposed by Rao and Patel [12] for the optimization of heat exchanger. Nseef et al. [13] proposed a multi-population ABC algorithm for the optimization dynamic optimization problems [13].

The multi-population approaches are useful for maintaining the population diversity. The characteristics of the multi population optimization approaches are useful because [14]:

a. Overall diversity of the search process can be maintained by allocating the entire population into groups, because various sub-populations can be situated in different regions of the problem search space.
b. These are having the ability of search in various regions simultaneously.
c. Population based optimization methods can be easily integrated within this method.

The selection of number of sub-populations is a critical issue in algorithm's performance. It is related with the complexity of the problem. The size of sub-populations continuously changes during the search process. The solutions in the sub-populations may also not be enough for enough diversity. In order to address these issues, the present work proposes a self-adaptive multi-population (SAMP) Jaya algorithm for the engineering optimization problems. In order to effectively monitor the problem landscape changes, the SAMP-Jaya algorithm adaptively changes the number of sub-populations based on the change strength of the problem landscape.

The basic objectives of this study are:

a. To propose a SAMP-Jaya algorithm that adapts the number of sub-populations based on the change strength of the problem.
b. To investigate the performance of the proposed algorithm on standard benchmark problems.
c. To investigate the performance of the proposed algorithm for an engineering application of design of a plate-fin-heat exchanger (PFHE) for minimum entropy generation rate.

The optimization studies in the present work have shown that SAMP-Jaya algorithm is capable of producing highly competitive results in comparison to the latest optimization methods reported. The design of PFHE suggested by the present approach reduces the entropy generation rate in comparison to the other algorithms considered.

The next section describes the working of the proposed SAMP Jaya algorithm.

2. SAMP Jaya algorithm

The Jaya algorithm is based on the concept that the solution obtained for a given problem should move towards the best solution and avoid the worst solution. Let $O(y)$ is an objective which is being optimized. Assume that at any iteration i, number of design variables is 'd' (i.e. $q=1, 2\ldots d$) and population size 'P' (i.e. $r=1, 2,\ldots,P$). If $Y_{q,r,i}$ is the value of the qth variable for the rth candidate during the i^{th} iteration, then this value is modified as per the following Eq. (2.1).

$$Y'_{q,r,i} = Y_{q,r,i} + r_1(Y_{q,best,i} - |Y_{q,r,i}|) - r_2(Y_{q,worst,i} - |Y_{q,r,i}|) \tag{2.1}$$

where $Y_{q,best,i}$ is the value of the qth parameter for the best solution and $Y_{q,worst,i}$ is the value of the qth parameter for the worst solution in the population. $Y'_{q,r,i}$ is the new value of $Y_{q,r,i}$ and r_1, r_2 are random numbers having the range of [0,1]. The term "$r_1(Y_{q,best,i}- |Y_{q,r,i}|)$" indicates that the solution tries to approach the best solution and the term "$- r_2(Y_{q,worst,i}- |Y_{q,r,i}|)$" shows that the solution tries to escape from the worst solution. $Y'_{q,r,i}$ is accepted if function value produced by it is better [5].

Comparison of the solutions for a particular candidate is based on comparison of the modified and the old solution and the "best" out of these is considered as best solution for that particular candidate. After the modification of the old solution (from the previous iteration), the algorithm compares the modified solution (of current iteration) with its corresponding old solution of the candidate. The modified solution is considered *if and only if* its fitness value (of the objective function) is better than the old solution; otherwise the old solution is considered. The same procedure is followed for all the candidates in the population. Thus, it is clear that only the best solutions will be forwarded as input to the next iteration. The algorithm always tries to get closer to success (i.e. reaching the best solution) and tries to avoid failure (i.e. moving away from the worst solution). The algorithm strives to become victorious by reaching the best solution and hence it is named as **Jaya** (a Sanskrit word meaning **victory**).

The basic difference between island-model GA and the proposed SAMP-Jaya algorithm is that the island-model GA uses only two groups