Energy Conversion and Management 140 (2017) 24–35

Contents lists available at ScienceDirect

Energy Conversion and Management

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

Design optimization and analysis of selected thermal devices using self-adaptive Jaya algorithm

R.V. Rao *, K.C. More

Dept. of Mech. Engg., S.V. National Institute of Technology, Surat 395007, India

A R T I C L E   I N F O

Article history:
Received 21 November 2016
Received in revised form 20 February 2017
Accepted 23 February 2017

Keywords:
Thermal devices
Optimization
Jaya algorithm
Self-adaptive Jaya algorithm

A B S T R A C T

The present study explores the use of an improved Jaya algorithm called self-adaptive Jaya algorithm for optimal design of selected thermal devices viz; heat pipe, cooling tower, honeycomb heat sink and thermo-acoustic prime mover. Four different optimization case studies of the selected thermal devices are presented. The researchers had attempted the same design problems in the past using niched pareto genetic algorithm (NPGA), response surface method (RSM), leap-frog optimization program with constraints (LFOPC) algorithm, teaching-learning based optimization (TLBO) algorithm, grenade explosion method (GEM) and multi-objective genetic algorithm (MOGA). The results achieved by using self-adaptive Jaya algorithm are compared with those achieved by using the NPGA, RSM, LFOPC, TLBO, GEM and MOGA algorithms. The self-adaptive Jaya algorithm is proved superior as compared to the other optimization methods in terms of the results, computational effort and function evalutions.

1. Introduction

Thermal system design includes an optimization process in which the designer considers certain objectives such as heat transfer, thermal resistance, total cost, friction factor, pressure amplitude, effectiveness, cooling capacity, and pressure drop depending on the requirements. While achieving the mentioned objectives, it is also desirable to minimize the total cost of the system. However, the design optimization for a whole thermal system assembly may involve complex objective functions with a large number of design variables. So, it is a good practice to apply optimization techniques to individual components or intermediate systems than to a whole system. For example, in a thermal power plant, individual optimization of cooling tower, heat pipe and heat sink is computationally and mathematically simpler than the optimization of whole system. Furthermore, the recent technological trend towards small scale and portable electronic devices has increased the need for micro-scale power sources and small-scale energy storage units like; micro-channel heat sink, MEMS switches and actuators. For example, compared with conventional semiconductor switches, radio frequency microelectromechanical systems (RF MEMS) switches have a better high-frequency characteristics, low power consumption, and high linearity which makes them to become as one of the most attractive devices for RF and microwave applications [1–3]. Power optimization of micro-scale power devices like micro-electro-thermal actuators which have extensive applications in integrated chip (IC) industry, robotics, avionics, and medicine industry have opened the doors for optimization.

The optimization techniques can be classified in two different types as given below.

(a) *Traditional optimization techniques:* These are deterministic algorithms with precise rules for transferring from one solution to the other. These algorithms have been in use for quite some time and have been successfully applied to many engineering design problems. The techniques include linear programming, non-linear programming, quadratic programming, dynamic programming, geometric programming, generalized reduced radient method, etc.
(b) *Advanced optimization techniques:* These techniques are stochastic in nature with probabilistic transition rules. These techniques are comparatively new and gaining popularity due to certain properties which the deterministic algorithms do not have. These techniques can be classified into different groups depending on the criteria being considered. Depending on the nature of phenomenon simulated by these techniques, the advanced optimization techniques have two important groups: evolutionary algorithms (EA) and swarm intelligence based algorithms.

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

http://dx.doi.org/10.1016/j.enconman.2017.02.068
0196-8904/© 2017 Elsevier Ltd. All rights reserved.

Although, traditional optimization techniques had been used to solve optimization problems in thermal system design, these techniques have the following limitations:

- Traditional techniques do not fare well over a broad spectrum of problem domains.
- Traditional techniques are not appropriate for solving multimodal problems as they tend to obtain a local optimal solution.
- Traditional techniques are not ideal for solving multi-objective optimization problems.
- Traditional techniques are not appropriate for solving problems having large number of constraints.

Considering the drawbacks of traditional optimization techniques, attempts are being made to optimize the thermal system optimization problems by using evolutionary and swarm intelligence optimization techniques. Most commonly used evolutionary optimization techniques is genetic algorithm (GA). However, GA gives near to most favorable solution for a complex problem having more number of variables and constraints. Determining the best values of controlling parameters like crossover probability, mutation probability, selection operator, etc. are difficult. Therefore, the effort must be continued to use more recently developed optimization techniques or to modify the existing algorithms to enhance the effectiveness.

It is observed from the literature survey on the design optimization of thermal system that certain advanced optimization techniques have been applied such as GA, particle swarm optimization (PSO), artificial bee colony (ABC), differential evolution (DE), niched Pareto genetic algorithm (NPGA), multi-objective genetic algorithm (MOGA) and teaching-learning-based optimization (TLBO) algorithm for the optimization of various objectives. These algorithms have showed their good performance in certain design optimization problems. However, these algorithms suffer from the problem of tuning of algorithm-specific parameters (except TLBO algorithm). For example, GA requires crossover probability, mutation probability, selection operator, etc.; NSGA-II requires crossover probability, mutation probability, real-parameter SBX parameter, and real parameter mutation parameter; PSO requires inertia weight, and social and cognitive parameters; and DE requires scaling factor and crossover constant. Recently another algorithm-specific parameter-less algorithm called Jaya algorithm has been developed by Rao [4]. The Jaya algorithm is simple in concept and is reported to give better results as compared to the other optimization algorithms [4,5]. In the present paper an improved version of Jaya algorithm is developed and is applied for design optimization of selected thermal devices. The selected thermal devices included are heat pipe, cooling tower, honeycomb heat sink and thermo-acoustic prime mover. The key feature is that the self-adaptive Jaya Algorithm determines the population size automatically. Hence, the user is not required to concentrate on choosing the population size. The improved Jaya algorithm (i.e. self-adaptive Jaya algorithm) is selected for this study due to its simplicity, robustness, algorithm-specific parameter free nature and its ability to get optimal solutions with less number of function evaluations and less memory requirement.

This research work is carried out with the following objectives:

1. To apply the recently developed Jaya algorithm to the design optimization of selected thermal devices such as heat pipe, cooling tower, honeycomb heat sink and thermo-acoustic prime mover.
2. To modify the existing Jaya algorithm so as to improve its performance by introducing the self-adaptive concept.
3. To apply the self-adaptive Jaya algorithm to the design optimization of selected thermal devices such as, heat pipe, cooling tower, honeycomb heat sink and thermo-acoustic prime mover and to compare the results.
4. The next section presents a brief introduction of Jaya algorithm and the proposed improved version of Jaya algorithm.

2. Proposed optimization algorithms

2.1. Jaya algorithm

Let the objective function is $Z(x)$ which is to be maximized or minimized and at any generation i, let 'l' is the number of design variables and 'n' is the population size. Let the best and worst values of the objective function during a generation are denoted by $Z(x)_{best,i}$ and $Z(x)_{worst,i}$ respectively. If $X_{l,n,i}$ is the lth design variable value for the nth population in the ith generation, then the value is updated as per Eq. (1) [4,5].

$$X'_{l,n,i} = X_{l,n,i} + r_{1l,i}(X_{l,best,i} - |X_{l,n,i}|) - r_{2,l,i}(X_{l,worst,i} - |X_{l,n,i}|) \tag{1}$$

where $X_{l,best,i}$ is the lth variable value for best population and $X_{l,worst,i}$ is the lth variable value for the worst population. $X'_{l,n,i}$ is the modified value of $X_{l,n,i}$ and the two random numbers are $r_{1l,i}$ and $r_{2,l,i}$ are in between [0,1]. The term "$r_{1l,i}\ (X_{l,best,i} - |X_{l,n,i}|)$" shows the ability of solution to go nearer to the best solution and the term "$- r_{2,l,i}\ (X_{l,worst,i} - |X_{l,n,i}|)$" shows the ability of the solution to shun the worst solution. $X'_{l,n,i}$ is accepted if it gives superior function value. At the end of each generation all accepted function values are kept and considered as the input to the next generation. The flowchart of the basic Jaya algorithm is shown in Fig. 1. This algorithm tries to obtain a best optimal solution and rejects the worst solution. The absolute value of $X_{l,n,i}$ (i.e. $|X_{l,n,i}|$), instead of $X_{l,n,i}$, is subtracted from $X_{l,best,i}$ and $X_{l,worst,i}$ in Eq. (1) for better exploration of the search space.

2.2. Self-adaptive Jaya algorithm

In this paper, an improved version of Jaya algorithm is proposed. Just like all population-based algorithms, the basic Jaya algorithm requires the common control parameters of population size and no. of iterations (but it does not require the algorithm-specific parameters). The choice of a particular population size for appropriate case studies is a difficult task [6]. However, not much work has been conducted yet on self-adaptive population sizes; hence this aspect is taken up in this work and the proposed algorithm is named as self-adaptive Jaya algorithm. The key feature is that the self-adaptive Jaya algorithm determines the population size automatically. Hence, the user need not concentrate on choosing the population size. Let the random initial population is $(10 * l)$, where l is the number of design variables then the new population is formulated as,

$$n_{new} = \mathrm{round}(n_{old} + r * n_{old}) \tag{2}$$

where r is a random value between [−0.5, 0.5]; it acts as a comparative population development rate. The population size may decrease or increase due to negative or positive random value of r. The flowchart of the proposed self-adaptive Jaya algorithm is presented in Fig. 2. Elitism is implemented when the population size of the next iterations is larger than the population size of the present generation ($n_{\text{new}} > n_{\text{old}}$). Then all of the existing population will go into the next generation and the best optimal solutions in the current population are assigned to the remaining $n_{\text{new}} - n_{\text{old}}$ solutions. When the population size of the next generation is less than the population size of the current generation ($n_{\text{new}} < n_{\text{old}}$), then, only