Scientia Iranica A (2019) 26(5), 2731–2747

Sharif University of Technology
Scientia Iranica
Transactions A: Civil Engineering
http://scientiairanica.sharif.edu

Artificial coronary circulation system: A new bio-inspired metaheuristic algorithm

A. Kaveh[a,*] **and M. Kooshkebaghi**[b]

a. *School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran.*
b. *Department of Civil Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.*

Received 25 January 2019; accepted 6 April 2019

KEYWORDS
Metaheuristic algorithm;
Optimization;
Artificial coronary circulation system;
Coronary arteries growth;
Human heart arterial tree.

Abstract. A new swarm intelligence optimization technique, called Artificial Coronary Circulation System (ACCS), is proposed in this study. This optimization method simulates the growth of coronary arteries (veins) in the human heart. In this algorithm, each capillary is considered a candidate solution. This algorithm starts with a random initial population of candidate solutions and evaluates them by using Coronary Growth Factor (CGF). In each run, the best candidate solution is selected as the main coronary vessel (artery or vein), and the other capillaries are considered as searchers of the search space. Then, the heart decides whether other candidates move toward/away from the main coronary vessels and searches for the optimal solution by using heart memory. Finally, the application of the proposed algorithm is demonstrated with respect to some benchmark functions and some mechanical problems, confirming the potential and capability of the new algorithm.

1. Introduction

Commonly, many optimization methods and algorithms can be categorized into two groups of deterministic and stochastic methods [1,2]. Deterministic techniques are dependent on the mathematical nature of the problems. These techniques are subject to a number of weaknesses such as dependency on gradient, local optimum, and inefficiency in handling large-scale problems [3]. Stochastic techniques are more user-friendly, because they are not dependent on the mathematical properties of a considered function; thus, these are more suitable for finding globally optimal solutions for an arbitrary type of objective function [4].

Optimization represents the process of determining the decision variables of a function such that the corresponding function has its minimum or maximum value [5]. For most of the optimization problems, particularly engineering ones, involved variables should be determined so that the system can operate in its best operation point [6].

As an alternative to the conventional mathematical programming methods, the metaheuristics have been utilized to obtain global or near global optimum solutions. Capable of exploring and finding suitable regions of search space in an inexpensive time period, these methods are quite appropriate for global searches and, thus, lessen the need for continuous cost functions and variables necessary for mathematical optimization methods [4].

To overcome the drawbacks of numerical methods including derivative, complexity, and being trapped in local optimum points, some optimization approaches known as metaheuristic algorithms have been introduced and developed in recent decades [1]. In these

*. *Corresponding author. Tel.: +98 21 44249493; Fax: +98 21 77240398*
E-mail addresses: alikaveh@iust.ac.ir (A. Kaveh); m.kooshkbaghi@hotmail.com (M. Kooshkebaghi)

doi: 10.24200/sci.2019.21366

methods, random operators are used, which are inspired by simple concepts. The metaheuristic methods are simple methods that can be applied to both continuous and discrete functions and require no other complex mathematical operations such as derivative; in addition, they rarely get trapped in local optima and are employed extensively for various optimization problems.

The Artificial Coronary Circulation System (ACCS) algorithm is a new metaheuristic algorithm that mimics the growth of coronary arteries tree of the heart and coronary circulatory system in human beings. This algorithm starts with a randomly generated initial population of candidate solutions for the coronary tree growth; the objective function is computed for them and, then, Coronary Growth Factor (CGF) is calculated based on these values. Based on this CGF, the best candidate solution is found as the main artery, and the others form capillaries. Then, the capillaries search the space and enforce other candidates to grow on the coronary tree and search for the optimal solution. The metaheuristic methods can be classified into six main categories:

1. **Evolutionary algorithms.** These methods simulate the evolution of nature. The first generation is randomly produced and evolved gradually. The best answer forms the best solution among the entire population in the last iteration of the evolution. Genetic Algorithm (Miettinen & Preface By-Neittaanmaki [7]) is the first and most well-known metaheuristic method that simulates Darwin's theory of evolution. Evolution Strategy (ES) [8] and Genetic Programming (GP) [9] are some other evolutionary methods;
2. **Physical-based optimization algorithms.** In these methods, physical rules are utilized to update the solutions in each iteration. Charged system search [10], colliding bodies optimization [11], black hole algorithm [12], and water evaporation optimization algorithm [13] are classified as the physical-based methods;
3. **Behavior of animal-based optimization algorithms.** This type of method mimics the social behavior of animals to enhance the knowledge of their goal such as finding a food source. The most well-studied approach of this group is particle swarm ptimization [14,15]; other approaches of theirs include cuckoo search [16], firefly algorithm [17], Monkey Search Algorithm (MSA) [18], Artificial Bee Colony (ABC) algorithm [19], whale optimization algorithm [20], Krill Herd (KH) [21], and grey wolf optimization algorithm [2];
4. **Behavior of human-based optimization algorithms.** This type of algorithms mimics the social behavior of human beings to increase their knowledge of a goal. The most familiar methods of this group are teaching-learning-based optimization [22], imperialist competitive algorithm [23], and cultural algorithms [24];
5. **Chemical-based optimization algorithms.** In this type of approaches, chemical rules are used for updating the solutions in each iteration. Chemical reaction optimization algorithm [25] and ions motion algorithm [26] are classified as the chemical methods;
6. **Biologically inspired algorithms.** These methods are based on the biological rules of microorganisms. For examples, Neural Network [27] is inspired by the network of interconnected neurons with the aim of imitating neural activities in human brains. Artificial immune systems [28] are inspired by the immune system of human body. Virus optimization algorithm [29] is also a biologically inspired algorithm.

A good optimization algorithm should be able to search all the search space, which is referred to as exploration. The high exploration algorithms are those with a large diversity of solutions in an iteration. Although a large number of optimization algorithms are introduced in the literature [30], it cannot be claimed that an optimization method is capable of solving all types of problems. Thus, new methods are introduced to solve a wider range of problems.

Metaheuristic algorithms are successfully applied to many engineering design problems, examples of which can be found in the studies of Kaveh & Shokohi [31], Kaveh & Talatahari [32], and Kaveh & Ilchi Gazaan [33].

Of note, an optimization algorithm known as Heart, inspired by heart, was previously proposed by Hatamlou [34]. This algorithm employs heart actions and circulatory system of human beings. In addition, it starts with a random population of solutions and an objective function computed for them. The best candidate solution is selected as the heart, and the others form blood molecules. Then, the heart persuades other candidates to move toward/away from the heart and look for an optimal solution. The inspiration of both Heart and the proposed ACCS is the same; however, the view and the equations by which the solutions are updated are completely different from the algorithm presented in this paper. A complete description of the proposed method is presented in Section 2. In order to distinguish the proposed ACCS from other methods, the results of some benchmark functions are compared in Section 3. The conclusion of the paper is presented in Section 4.

2. Artificial Coronary Circulation System (ACCS) algorithm

2.1. Behaviors of coronary circulation system

In this section, the proposed algorithm and the source of its inspiration are described. Coronary circulation is the circulation of blood in the blood vessels of the heart muscle (myocardium). Coronary circulation is part of the systemic circulatory system that supplies blood and provides drainage from the tissues of the heart.

Arterial trees are very important for the transfer of oxygen and nutrients to tissue. Their anatomy has been investigated for a long time period through the dissection of cadavers, inspection of corrosion casts, medical imaging methods, and interesting computational models. It has been shown that individual arterial bifurcations follow optimality principles that lower metabolic demand locally, as exhibited by the scaling laws followed by arterial trees.

Recently, there has been a great interest in models that mimic arterial growth (angiogenesis) by employing physiological principles to simulate vascular anatomy. These models are created based on local optimization principles, where the anatomy of each branch in the arterial tree is governed by a compromise between maximizing fluid dynamical efficiency and minimizing the quantity of blood required. The vessels that deliver oxygen-rich blood to the myocardium are the coronary arteries. The coronary arteries that run deep within the myocardium are referred to as sub endocardial. Further, the vessels that remove deoxygenated blood from the heart muscle are known as cardiac veins. Cardiac veins carry blood with a poor level of oxygen from the myocardium to the right atrium. The anatomy of the veins of the heart is quite variable [35].

Therefore, the coronary circulation system constitutes the heart and three types of blood vessels, namely arteries, veins, and capillaries. The heart is the main part of the system and acts as a pump to distribute nutrient- and oxygen-rich blood through the body; it then takes away carbon dioxide and other wastes from the body, which is not required. Therefore, in the coronary circulation system, the growing patterns of arteries and veins and their capillaries are similar.

2.2. Presentation of Artificial Coronary Circulation System (ACCS)

A coronary tree begins from its main arterials (Left Main or Left Coronary Artery (LCA) LAD, RCx, and Right Coronary Artery (RCA)), and the growth of arteries is indispensable for coronary arterial tree growth. All arteries (veins) of a coronary arterial tree can be considered as a system composed of a large number of arterial (veins) stems and crowns. The stem-crown units are defined in Figure 1. A schematic view of the coronary arteries is illustrated in Figure 2. A coronary arterial tree structure consists of many stem-crown units.

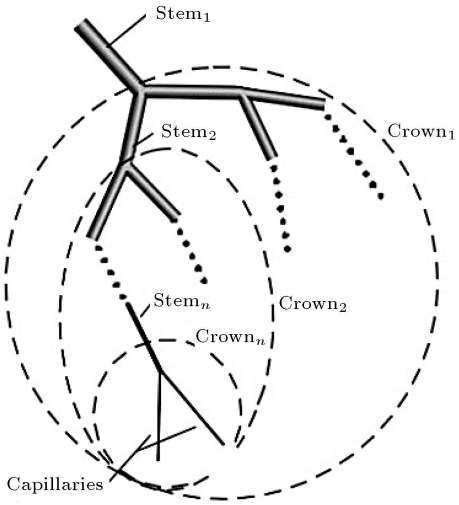

Figure 1. Schematic illustration of the stem-crown unit.

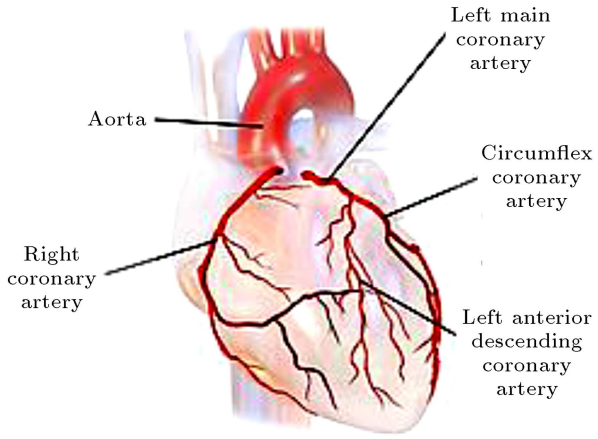

Figure 2. The schematic view of the coronary arteries.

Each capillary searches for a feasible solution to the problem. All arteries try to adjust their growing directions and propagation strategies to search for the optimal growing conditions, providing feedback to improve vessels growth further. In the growing process, all the terminal vessels can select their growth strategies composed of the following three basic actions.

1. Each capillary leader can elongate forward (or sideways) in the search space (***Searching***);
2. Each capillary leader can produce new stem-crown (***Bifurcation***);
3. Each capillary leader can stop and become an ordinary stem of the tree (***Pruning***).

In other words, a capillary may re-grow, produce new stem-crown, or stop growing. According to fitness values, the whole capillaries are divided into three groups. The group with the best fitness values is called

main vessels (stem). The group with the worst fitness values is called ***Pruning*** aging vessels. The rest of vessels are called lateral vessels. In the three groups, except for aging vessels that will stop growing in the next generation, the main vessels and lateral vessels implement different growth strategies.

The L-systems method was first employed to categorize plant growth processes; however, it can be applied to any divaricating system. L-systems are used to describe the growing behavior of the coronary tree.

The main function of an arterial tree is to retain sufficient blood perfusion with the least total metabolic expense. The behavioral tendency of an arterial tree is dominated by two features:

1. Since the blood is viscous, the power required to pump blood through the vasculature should be minimized;
2. Since energy is required to generate and maintain blood, the volume of blood should be reduced to a minimum.

Murray's law provides this for individual bifurcations; however, the optimal organization of a large number of connected bifurcations is far from obvious. The interaction between these tourney concerns for thousands of arterial sectors leads to a complex optimization problem. Of note, in the following, bifurcation points will be referred to as "nodes", arteries as "segments between nodes", and terminal arterioles as "end nodes" [36].

Now, we have obtained a form for all of the relevant costs corresponding to an arbitrary tree configuration that provides an arbitrary tissue shape. Therefore, one can define total cost (C_T), which gives a numerical measure for the fitness of a tree as in the following:

$$C_T = A_{w,v}(C_w + C_v) + C_0, \tag{1}$$

where A_i is the weighting value that scales any relevant cost. There is no way to determine what weights to use analytically, and appropriate weights must be found experimentally. Although C_y is the metabolic cost due to the tree volume and C_w is the total power required to maintain a suitable flow through the tree that is based on Poiseuille's law, $\Delta p = QR$ is also followed in the parts, where Q is the flow and Δp is the pressure drop over the vessel. In addition, C_0 indicates other costs.

In this suggested algorithm, ACCS (Artificial Coronary Circulation System), any branch of the coronary tree is supposed to be a new solution; the total cost of a tree at any ends is considered to be the cost of the objective function in any solution. Accordingly, the Coronary Growth Factor (CGF) is calculated and defined similar to Vascular Endothelial Growth Factor (VEGF) for any solution. Then, the search space of the problem is searched, and a new position is introduced for the growth of the coronary tree as a new solution to problems. This process is repeated to find the optimum solution. The method suggested in this paper is developed based on the biology of the human heart's coronary arterial tree growth. In other words, in the human heart, the coronary vein tree is the same as the coronary arterial tree. This study focuses on coronary arterial tree. Therefore, in an artificial model presented herein, an objective function is considered as the cost function of coronary arteries tree growth; in addition, every young arteries are considered as agents that explore the search space of the problem.

2.3. Mathematical model of the proposed algorithm

In this part, a new effectual optimization algorithm inspirited by the growth coronary arteries tree is presented, which is called Coronary Circulation System (CCS). In the CCS, any solution selects X_i, which includes a number of decision variables (i.e., $X_i = \{x_{i,j}\}$) in a vessel (arteries, veins, and capillaries). The vessels make the coronary arteries tree grow by the Coronary Growth Factor (CGF) of the other arteries (agents). It appears that a good artery (agent) has more capillary leaders than a bad artery; therefore, the amount of the Coronary Growth Factor (CGF) will be considered as the objective function value, fit_i, to describe CCS. The subsequent basic laws are developed:

Law 1: The initial positions of CLs are obtained randomly in the search space as follows:

$$X_0^i, j = L^j_{\text{band}} + \text{rand} \cdot \left(U^j_{\text{band}} - L^j_{\text{band}}\right),$$
$$i = 1, \cdots, N_{pop}, \qquad j = 1, \cdots, N_{var}, \tag{2}$$

where $X_0^{i,j}$ is the initial value of the jth variable for the ith CL. N_{var} is the total number of variables, and L_{band} and U_{band} are the lower and the upper bounds of decision variables, respectively. Here, rand is a random variable selected uniformly in the range of [0,1]. Thus, the Coronary Growth Factor (CGF) of the initial CLs is determined based on the cost function (or fit_i).

Law 2: Multitude of natural evolution algorithms constitutes a population of solutions that are evolved through random alterations and selection. Analogously, ACCS supposes a number of Capillary Leaders (CL), with each CL having a Coronary Growth Factor (CGF).

$$\text{CGF}_i = \frac{fit_i}{\sum fit_i}, \qquad i = 1, \cdots, N_{pop}, \tag{3}$$

where fit_i is the objective function value or the fitness of agent i, $fit_{\max}$ represents the maximum of objective function values, and N_{pop})shows the total number of CLs.

Law 3: Three conditions could be supposed to be related to the type of the Coronary Growth Factor (CGF):

a) Any CL can grow in any direction;

b) The best CL can grow in its direction if the amount of its Coronary Growth Factor (CGF) (in minimizing problems) is higher than others, and the worst CLs must be pruned (***Selection***).

$$\begin{cases} X_0^i = X_{\text{new}}^i & \text{CGF}_0^i < \text{CGF}_{\text{new}}^i \\ X_0^i = X_0^i & \text{else} \end{cases} \tag{4}$$

c) Any CL grows toward to the best direction. In other words, other CLs grow in a good CL direction if the Coronary Growth Factor (CGF) of the center CL is better (higher) than the random CL (***Searching***).

$$X_c = \text{mean}\left(X_0^{\text{all}}\right), \tag{5}$$

$$fit_c = \text{mean}(fit_0). \tag{6}$$

According to the above conditions, when the best CL grows in the worst direction, the local search capability of the algorithm should be prepared; if an unfavorable CL growth becomes the best CL, the global search is provided. When a CL grows toward the best CL, its performance is improved; thus, the self-adaptation principle is assured. Growth of the best CL toward the worst direction can lead to the loss of the previous favorable solution or, at least, an increase in calculation costs in finding a good solution. To resolve this problem, Heart Memory (HM) is used, which retains the best CLs (favorable solution).

Law 4: The new position of any CL is determined through the equation below:

$$X_{i,j}^{t+1} = X_{r,j}^t + dir.B_f.\left(X_{c,j}^t - \text{rand} \cdot X_{r,j}^t\right), \tag{7}$$

$$\begin{cases} dir = -1 & \text{if CGF}_c^t < \text{CGF}_i^t \\ dir = +1 & \text{else} \end{cases} \tag{8}$$

where $X_{i,j}^{t+1}$ denotes the new position value of the jth variable for the ith CL; dir indicates the growth direction; rand is a different random number, uniformly distributed in the range of (0,1); B_f is a Bifurcation factor, which is equal to CGF_i; $X_{r,j}^t$ denotes a random old position value of the jth variable for the rth CL ($r \in 1, \cdots, N_{pop}$). Note, r in $X_{r,j}^t$ and CGF_r^t and $(r-1)$ is the same. Considering the above equations, any CL searches the space for a new position for the growth of the main arteries (***Global Search***).

Law 5: The new positions of the CLs should be pruned and changed.

When the heart Coronary Growth Factor (CGF) of a new position is better than an old value (in other words, if the fitness function of a new point is more favorable (lower) than the previous point), then the branch changes to the main arteries; otherwise, it prunes and the growing tree stops. The mathematical formulation of this feature is presented in the following:

$$\begin{cases} X_0^i = X_{\text{new}}^i & \text{CGF}_0^i < \text{CGF}_{\text{new}}^i \\ X_0^i = X_0^i & \text{else} \end{cases} \tag{9}$$

This procedure is accomplished for all the old positions, i.e., in this stage, all the new positions are considered as the main artery leaders, and the best CLs in the new position indicate tree growth in the true direction, leading to the extension of the tree. Moreover, the worst positions are replaced by the new positions, meaning that the bad branch is removed and pruning is done.

Law 6: The new position of any CL is obtained as follows.

As mentioned in the last steps, all of the new positions are considered as the main artery leader, and the coronary tree grows. In this stage, all of the new positions are considered as the capillary leader, and the coronary tree grows. In addition, the growth of the capillary leader is based on the Coronary Growth Factor (CGF) of arteries leader along the coronary arteries tree. Thus, angiogenesis factor is defined as follows:

$$\alpha = 0.625^{\sqrt{(itr/itr_{\max})}}, \tag{10}$$

where itr is the number of iterations, and $itr_{\max}$ is the maximum number of iterations. The subsequent positions of the CL have a connection with the Coronary Growth Factor (CGF) of the best and worst CLs in the coronary arteries tree. Thus, the next positions of the CLs are mathematically formulated as follows:

$$X_{i,j}^{t+1} = X_{i,j}^t + \alpha \cdot \text{rand} \cdot \left(X_{b,j}^t - X_{w,j}^t\right), \tag{11}$$

where $X_{i,j}^{t+1}$ indicates the new position value of the jth variable for the ith CL; $X_{i,j}^t$ shows the old position value of the jth variable for the ith CL; α is the angiogenesis index; rand represents random numbers that are uniformly distributed in the range of (0,1); $X_{b,j}^t$ and $X_{w,j}^t$ are the best and worst CLs of the last (old) population, which hark back to the foremost arteries leader and the worst arteries leader. By using Eq. (11), any CL searches the space for a new position for capillary growth (***Local Search***).

Law 7: Considering a memory that saves the foremost CL vectors and their related Coronary Growth Factor (CGF) values can improve the algorithm performance without increasing computational costs. Therefore, Heart Memory (HM) is employed to save a number of the best-so-far solutions. In this paper, the size of the HM (i.e., HMS) is taken as $0.25 \times N_{pop}$. Another profit of the HM results from employing this memory to direct and guide the current CLs. On the other hand, the vectors saved in the HM can be used to generate a new position for the current CLs.

Law 8: The finishing criterion is one of the followings:

"Maximum number of iterations or minimum objective function error or..."

Now, a new optimization algorithm that employs the above laws can be established. Algorithm 1 summarizes the steps of the ACCS.

```
BEGIN
Initialization (set N_pop, N_var, HMS, Iteration_max)
Create the Heart Memory
while Iteration ≤ Iteration_max
    Obtain X_center of the population.
        for i = 1 to N_pop (Global Search)
            Update all variables based on Eq. (7)
        end
    Check the boundaries with Heart Memory.
    Select the best solution based on CGF.
        for i = 1 to N_pop (Local Search)
            Update all variables based on Eq. (11)
        end
    Check the boundaries with Heart Memory.
    Select the best solution based on CGF.
    Update Heart Memory.
    Iteration = Iteration + 1
end
END
```

Algorithm 1. The Artificial Coronary Circulation System (ACCS) algorithm.

3. Evaluation and validation of the ACCS

In this part, the suggested algorithm is appraised by some benchmarks, and the outcomes are compared with the results of some other metaheuristics. In this paper, the first 15 benchmark functions are classified based on modality and separability, and 4 mechanical problems are tested to evaluate the reliability and efficiency of the suggested algorithm. After that, the performance of the ACCS is examined for 3 engineering design problems (widely utilized in the literature), and the optimization outcomes are compared with those of the other optimizers. These instances have been previously studied by various techniques, which are useful to show the validity and effectiveness of the suggested algorithm. To evaluate the effect of the initial population on the final outcome, these instances are independently optimized with different initial populations. The ACCS algorithm is programmed in MATLAB R2016a.

The suggested method is assessed by means of 15 different benchmark mathematical functions, and the results are compared with those of some other metaheuristic methods. Such trial functions are listed in Tables 1-4, where n represents the dimension of the function (i.e., the number of decision variables), Range indicates the search area (i.e., the limitation of variables), and $f_{\min}$ is the optimum value of the test function. By using these optimization problems, the ability of the suggested method to solve high constraints and discrete problems is assessed.

The metaheuristic algorithms with which the suggested method is compared include:

1. Cuckoo Search Algorithm (CS) [37];
2. Firefly Optimization Algorithm (FFA) [3,38];
3. Lightning Search Algorithm (LSA) [39].

The population number is specified for any method so that the evaluation process of the overall objective function in the whole iterations remains the

Table 1. Unimodal and separable test functions.

Function ID	Name	Expression	n	Range	$f_{\min}$
F1	Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	±100	0
F2	Step	$f_2(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	30	±10	0
F3	Quartic with noise	$f_3(x) = \sum_{i=1}^{n} i x_i^4 + \text{rand}$	30	±1.28	0

Table 2. Unimodal and non-separable test functions.

Function ID	Name	Expression	n	Range	$f_{\min}$
F4	Schwefel 1.2	$f_4(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	±100	0
F5	Schwefel 2.21	$f_5(x) = \max(\lvert x_i \rvert, 1 \le i \le n)$	30	±100	0
F6	Schwefel 2.22	$f_6(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	±10	0
F7	Rosenbrock	$f_7(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	±30	0

Table 3. Multimodal and separable test functions.

Function ID	Name	Expression	n	Range	$f_{\min}$
F8	Rastrigin	$f_8(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30	±5.12	0
F9	Branin	$f_9(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	±5	0.398

Table 4. Multimodal and non-separable test functions.

Function ID	Name	Expression	n	Range	$f_{\min}$
F10	Ackley	$f_{10}(x) = 20 - 20\exp\left(-0.2\sum_{i=1}^{n}\frac{x_i^2}{n}\right) - \exp\left(\sum_{i=1}^{n}\frac{\cos(2\pi x_i)}{n}\right) + e$	30	±32	0
F11	Griewank	$f_{11}(x) = 1 + \sum_{i=1}^{n}\frac{x_i^2}{4000} - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	30	±600	0
F12	Penalized no.1	$f_{12}(x) = \frac{\pi}{n}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{n}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})] + (y_n-1)^2\right\} + \sum_{i=1}^{n}u_i(x_i,10,100,4)$, $y_i = 1 + \frac{x_i+1}{4}$, $u_i(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i-a)^m & x_i < -a \end{cases}$	30	±50	0
F13	Penalized no.2	$f_{13}(x) = 10\left\{\sin^2(3\pi x_1) + (x_n-1)^2 + \sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]\right\} + \sum_{i=1}^{n}u_i(x_i,5,100,4)$, $u_i(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i-a)^m & x_i < -a \end{cases}$	30	±50	0
F14	6-Hump Camel Back	$f_{14}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	±5	-1.0316
F15	Goldstein-Price	$f_{15}(x) = [1 + (x_1+x_2+1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	±2	3.0

Table 5. Parameter settings used in LSA, CS, and FFA.

Algorithm	Function evaluation in each iteration	Populations size	Other parameter
CS	2	25	Discovery rate (Pa)=0.25
LSA	2	25	Maximum channel time=10
FFA	1	50	$\alpha = 0.25$, $\beta = 1$, $\gamma = 1$
ACCS	2	25	—

Table 6. Test 1 global optimization results for benchmark functions in Table 1.

Function ID	Statistics	ACCS	LSA	CS	Firefly
F1	Best	$\mathbf{1.138 \times 10^{-55}}$	8.951×10^{-7}	1.638	2.024×10^{-7}
	Average	$\mathbf{1.211 \times 10^{-49}}$	0.00398	4.634	2.388×10^{-7}
	Worst	$\mathbf{2.040 \times 10^{-48}}$	0.0283	8.792	2.739×10^{-7}
	Standard deviation	$\mathbf{4.141 \times 10^{-49}}$	0.00702	1.867	2.131×10^{-8}
F2	Best	**0**	2.0	6.0	20
	Average	**0**	19.7	15.533	185.3
	Worst	**0**	110.0	36.0	875.0
	Standard deviation	**0**	22.085	7.147	187.684
F3	Best	**0.00612**	5.60947	0.42498	0.96311
	Average	**0.01964**	16.0165	1.07974	1.75927
	Worst	**0.05142**	48.6114	2.19394	3.56069
	Standard deviation	**0.01195**	9.47034	0.42321	0.61097

same for all methods. For example, the evaluation function is assessed two times in any iteration in the suggested ACCS algorithm, while this is done only once in any iteration for the FFA.

Therefore, the population of ACCS is half of the population of the FFA. The algorithm-dependent parameter settings for any algorithm in comparison are given in Table 5, as mentioned in the literature.

The maximum cycle number for all the algorithms is 500 for the mathematical functions and 200 for the engineering design problems. In addition, the MATLAB programs of the compared algorithms can be taken from the Appendix.

Since the metaheuristic methods are based on random movement of their particles, statistical analysis is required. In order to validate the suggested algorithm, the efficiency of the ACCS is assessed by using ten real-world structural optimization problems.

3.1. Experimental results and discussion of unconstrained functions

3.1.1. Unimodal and separable functions

This trial is used to estimate the reliability and performance of the ACCS in exploring the global minimum value when it is due to benchmark functions with unimodal and separable characteristics. Table 1 provides details of these functions. This trial also compares the ACCS with three other methods (namely LSA, CS, and FFA) for validation. Each benchmark function is tested 30 times. The outcomes consist of the best, worst, average, and standard deviation of the objective function, as illustrated in Table 6. The best efficiency for each function is boldfaced.

The ACCS reaches the best global minimum or near global minimum for F1, F2, and F3, and its efficiency is acceptable because it has the lowest standard deviation and average global minimum value. Figure 3(c) clearly shows this which has been obtained from the data determined in 30 runs.

Therefore, the convergence characteristic curves illustrated in Figure 3 are used for efficiency comparison. According to this figure, it should be noted that the ACCS converges faster than the other methods and, thus, possesses superior convergence characteristics for this kind of function optimization.

3.1.2. Unimodal and non-separable functions

To appraise the efficiency and consistency of the ACCS in solving unimodal and non-separable functions, four benchmark functions represented in Table 2 are ex-

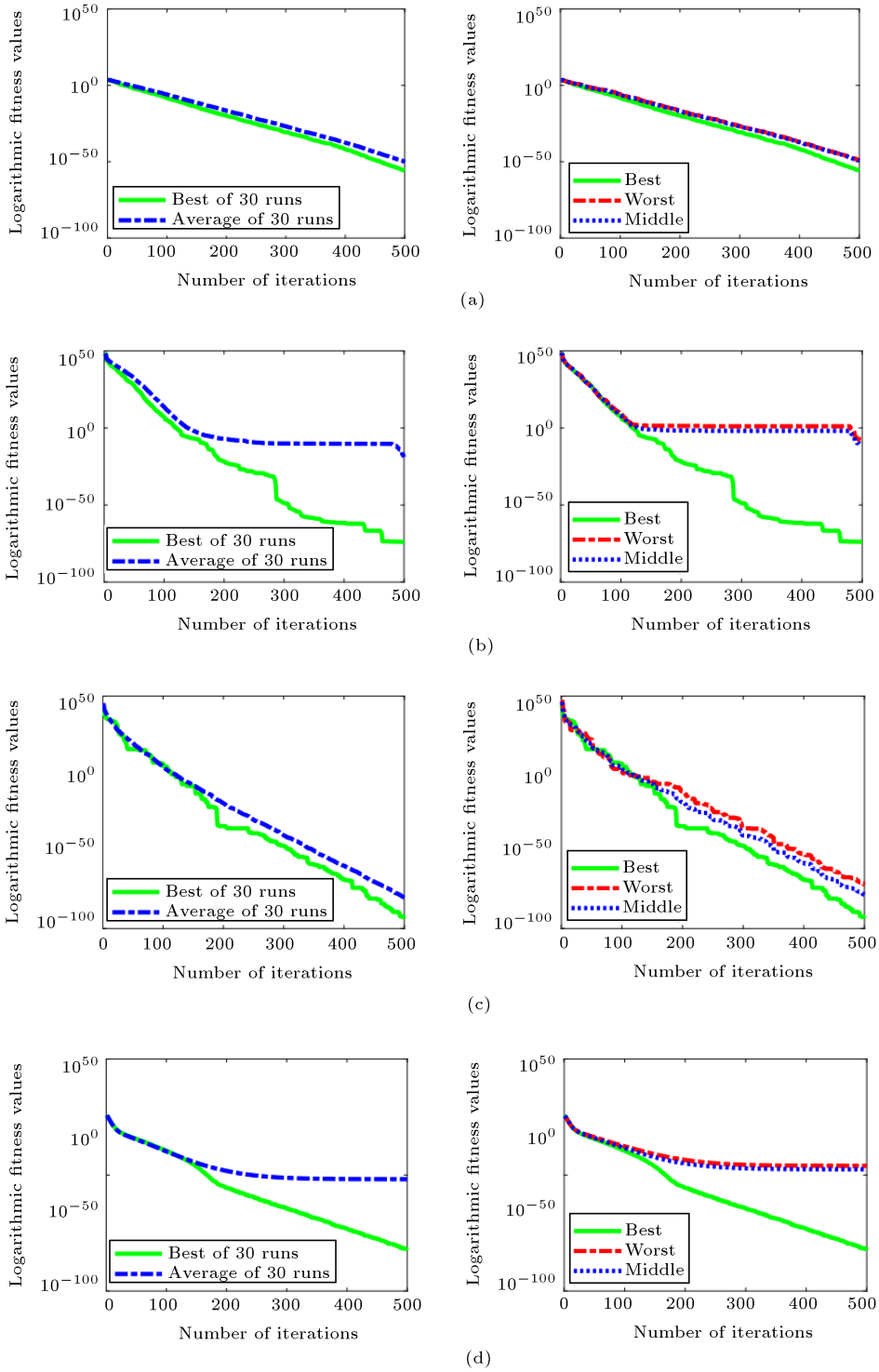

Figure 3. Comparison of convergence curves relative to the best design and average optimization run and global search speed and local search speed of all algorithms for the sphere function: (a) ACCS, (b) LSA, (c) CS, and (d) firefly.

amined in this test: Schwefel 1.2 (F4), Schwefel 2.21 (F5), Schwefel 2.22 (F6), and Rosenbrock (F7). These functions have the same dimension size ($n = 30$) as those used in Test 1; however, the obstacle level is higher because these functions are non-separable. The test outcomes achieved by the ACCS are compared with those calculated by the four optimization methods (Table 7). The best efficiency for any function is boldfaced. Table 7 shows that the ACCS can find the best near-optimum solution for functions F4, F5, and F6. For F7, the ACCS fails to achieve the best solution, and the LSA finds a better outcome. In any case, its

Table 7. Test 2 global optimization results for benchmark functions in Table 2.

Function ID	Statistics	ACCS	LSA	CS	Firefly
F4	Best	$\mathbf{5.611 \times 10^{-12}}$	16.674	435.93	2136.13
	Average	$\mathbf{1.037 \times 10^{-6}}$	117.655	1232.33	6522.55
	Worst	$\mathbf{2.307 \times 10^{-5}}$	354.017	2060.69	16004.4
	Standard deviation	$\mathbf{4.214 \times 10^{-6}}$	74.345	418.89	3304.3
F5	Best	$\mathbf{5.3136 \times 10^{-16}}$	6.795	4.904	5.9682
	Average	$\mathbf{3.7424 \times 10^{-14}}$	27.088	8.423	8.3848
	Worst	$\mathbf{1.9806 \times 10^{-13}}$	65.486	11.818	11.6706
	Standard deviation	$\mathbf{5.2905 \times 10^{-14}}$	15.125	1.814	1.70011
F6	Best	$\mathbf{2.662 \times 10^{-33}}$	9.974×10^{-3}	1.009	12.375
	Average	$\mathbf{1.643 \times 10^{-31}}$	1.522	1.606	23.005
	Worst	$\mathbf{1.888 \times 10^{-30}}$	10.093	2.792	42.143
	Standard deviation	$\mathbf{3.532 \times 10^{-31}}$	2.412	0.408	7.622
F7	Best	21.5317	**12.0247**	40.724	50.995
	Average	**22.6195**	56.1413	78.252	147.232
	Worst	**24.0443**	182.264	176.352	314.053
	Standard deviation	**0.6932**	40.671	31.609	74.1709

Table 8. Test 3 global optimization results for benchmark functions in Table 3.

Function ID	Statistics	ACCS	LSA	CS	Firefly
F8	Best	**0**	56.7125	66.1701	69.6471
	Average	**1.88747**	109.142	83.4389	114.022
	Worst	**11.56313**	186.305	100.363	174.117
	Standard deviation	**3.10235**	33.3461	9.8677	24.948
F9	Best	**0.39788**	**0.39788**	**0.39788**	**0.39788**
	Average	**0.39788**	**0.39788**	**0.39788**	**0.39788**
	Worst	**0.39788**	**0.39788**	**0.39788**	**0.39788**
	Standard deviation	**0**	5.5501×10^{-6}	**0**	1.48131×10^{-14}

efficiency is admissible since it has the lowest standard deviation and average global minimum value.

3.1.3. Multimodal and separable functions

In this test function, the obstacle level of the optimization problem becomes higher by multimodal and separable functions (Table 3).

F8 and F9 are associated with high- and low-dimensional problems, respectively. Each benchmark function is run again 30 times. The results considering the best, worst, average, and standard deviation of the objective functions are illustrated in Table 8. The best efficiency for any function is boldfaced. All the algorithms achieve the best global minimum or near global minimum for F9. For F8, only the ACCS algorithm finds the best solution. The comparative outcomes illustrate better efficiency of the ACCS than other algorithms.

3.1.4. Multimodal and non-separable functions

The global and local search capabilities of the suggested ACCS are examined with six multimodal and non-separable high- and low-dimensional benchmark functions. Table 4 provides the details of these functions. Similar to the previous test function analysis, this test also compares the ACCS with three other methods, namely LSA, CS, and FFA, for validation using the same statistical indices (Table 9). The best efficiency for each function is boldfaced. The ACCS reaches the best global minimum or near global minimum for all the tested functions. These outcomes confirm the exploration and exploitation capabilities of the suggested ACCS.

Table 9. Test 4 global optimization results for benchmark functions in Table 4.

Function ID	Statistics	ACCS	LSA	CS	Firefly
F10	Best	$\mathbf{8.8817 \times 10^{-16}}$	2.01374	0.11665	1.15514
	Average	$\mathbf{1.0066 \times 10^{-15}}$	3.674301	0.33521	1.83652
	Worst	$\mathbf{4.4408 \times 10^{-15}}$	5.05808	0.89566	2.53809
	Standard deviation	$\mathbf{6.4863 \times 10^{-16}}$	0.78941	0.22182	0.28382
F11	Best	**0.000000**	5.60705×10^{-6}	0.847211	2.7863×10^{-7}
	Average	**0.006555**	0.007863	1.005346	0.334255
	Worst	**0.041631**	0.04462	1.066508	1.857301
	Standard deviation	**0.010733**	0.011281	0.046827	0.589796
F12	Best	$\mathbf{1.42603 \times 10^{-6}}$	4.29435×10^{-6}	1.20636	1.50081
	Average	**0.006926**	3.23765	2.75402	2.97896
	Worst	**0.103681**	17.16439	4.30311	4.33802
	Standard deviation	**0.0263**	4.521759	0.872652	0.781504
F13	Best	$\mathbf{5.59887 \times 10^{-6}}$	3.81127×10^{-5}	1.09591	0.63735
	Average	**0.030906**	1.74476	6.49742	31.9031
	Worst	**0.073753**	18.3532	20.8961	66.4301
	Standard deviation	**0.041681**	4.23541	4.175104	16.39227
F14	Best	**-1.03163**	**-1.03163**	**-1.03163**	**-1.03163**
	Average	**-1.03163**	**-1.03163**	**-1.03163**	**-1.03163**
	Worst	**-1.03163**	**-1.03163**	**-1.03163**	**-1.03163**
	Standard deviation	5.215×10^{-16}	6.5194×10^{-16}	$\mathbf{4.8781 \times 10^{-16}}$	2.6225×10^{-14}
F15	Best	**2.99999**	**2.99999**	**2.99999**	**2.99999**
	Average	**2.99999**	**2.99999**	**2.99999**	3.00000
	Worst	**2.99999**	**2.99999**	**2.99999**	3.00000
	Standard deviation	6.2744×10^{-15}	$\mathbf{2.9836 \times 10^{-15}}$	3.1811×10^{-15}	2.1989×10^{-13}

Finally, Figure 4 shows the convergence history of the ACCS for the best-viewed outcomes and the average of 30 independent runs for many functions.

3.2. Experimental results and discussion on engineering problems

Herein, the suggested ACCS is utilized for the solution of 3 classic engineering optimization problems consisting of a tension/compression spring, a welded beam, and a pressure vessel. For these problems, there are some equality and inequality constraints that must be satisfied during the solution of the problem. In this paper, the constraint management is carried out by utilizing penalty factors, i.e., whenever the constraints are violated for a solution, a great fitness function is assigned to the solution.

3.2.1. The tension/compression spring design problem

In this problem, the objective is to minimize the weight of a tension/compression spring with a certain number of constraints such as the shear stress, the surge frequency, and minimum deflection, as shown in Figure 5. The design variables include the mean coil diameter $D(= x_1)$, the wire diameter $d(= x_2)$, and the number of active coils $N(= x_3)$. The problem can be expressed better with the cost function as:

$$f_{\text{cost}}(\mathbf{X}) = (x_3 + 2)x_2 x_1^2,$$

that has to be minimized by considering the following constraints:

$$g_1(\mathbf{X}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,$$

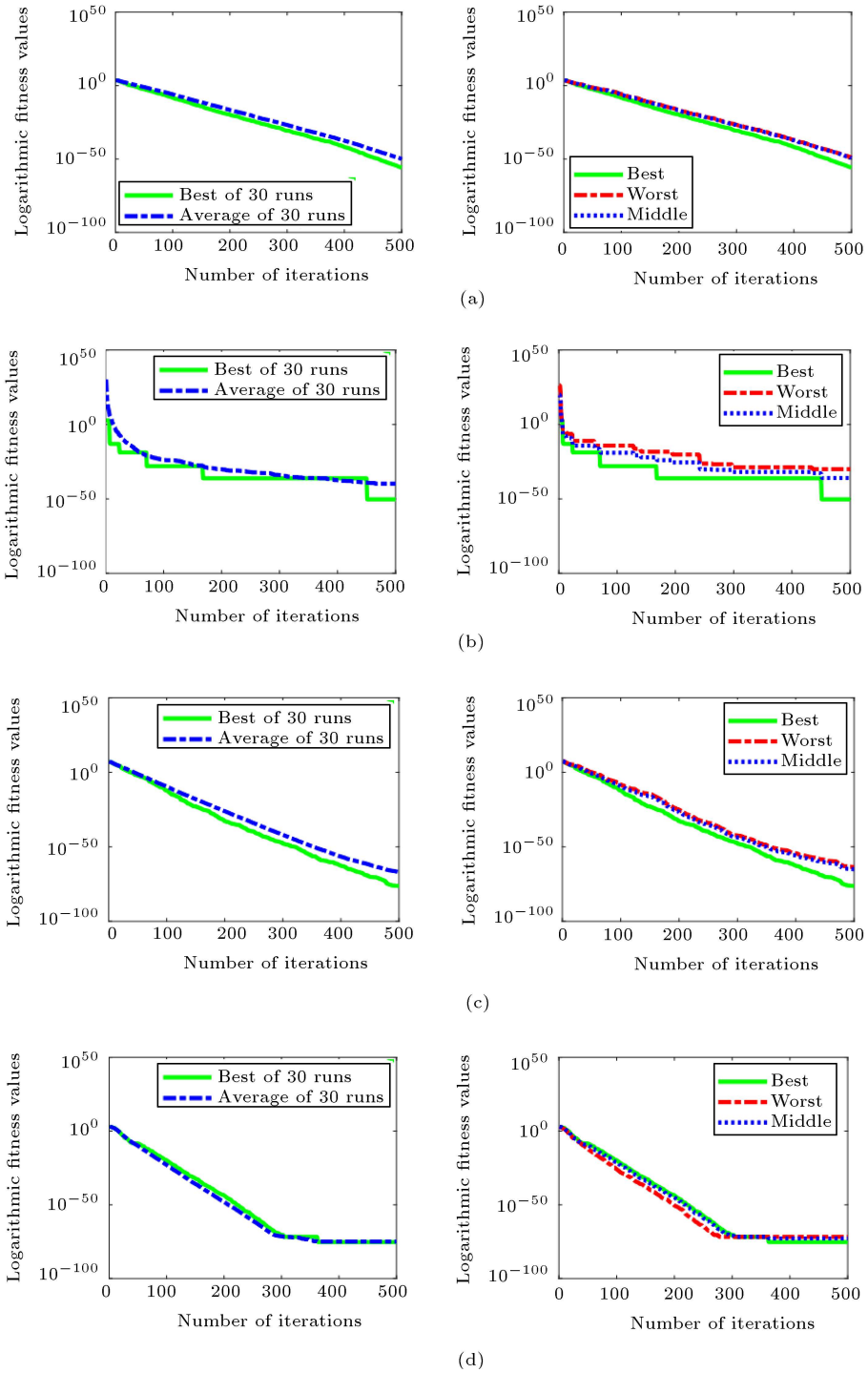

Figure 4. Comparison of convergence curves relative to the best design and average optimization run and global search speed and local search speed of the ACCS: (a) f_1, (b) f_3, (c) f_7, and (d) f_{10}.

$$g_2(\mathbf{X}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0,$$

$$g_3(\mathbf{X}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0,$$

$$g_4(\mathbf{X}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,$$

$$0.05 \leq x_1 \leq 2.0 \qquad 0.25 \leq x_2 \leq 1.3 \qquad 2.0 \leq x_3 \leq 15.$$

This problem is tackled by various methods consisting of some metaheuristic methods and mathematical programming approaches. The best outcomes

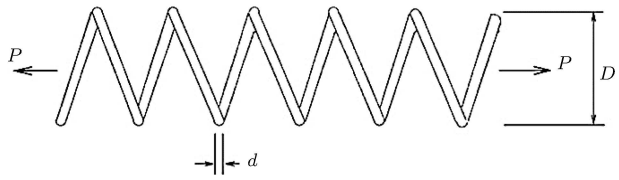

Figure 5. A tension/compression spring.

for this problem, according to the author's knowledge, are obtained by metaheuristic algorithms. In this regard, GA-based method [40], Co-evolutionary Particle Swarm Optimization (CPSO) [41], Evolution Strategies (ESs) [42], Charged System Search (CSS) algorithm [10], and Bat Algorithm (BA) [37], Water Evaporation Optimization (WEO) [13] are used successfully for this problem.

Table 10 provides the statistical results obtained by ACCS and other metaheuristics for the present problem. The best result is obtained by the CSS algorithm. As is clear, the ACCS is competitively better than other algorithms in terms of accuracy, demonstrating the most robust results. Table 11 shows the optimum design achieved by ACCS and some other metaheuristic algorithms. It can be observed that the best outcomes are achieved by the presented ACCS algorithm.

3.2.2. The welded beam design

In this problem, the aim is to minimize the fabrication cost of a welded beam, as shown in Figure 6. The considered variables contain the thickness of weld ($x_1 = h$), the length of attached part of bar ($x_2 = l$), the height of the bar ($x_3 = t$), and the thickness of the bar ($x_4 = b$). The constraints include shear stress (s), bending stress in beams (h), and buckling load on the bar (P_c), end deflection of the beam (d), and the side constraints. The mathematical formulation of this problem is as follows:

Minimize:

$$f(X) = 1.10471 x_2 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2),$$

Subject to:

$$g_1(X) = \tau(x) - \tau_{\max} \leq 0,$$

$$g_2(X) = \sigma(x) - \sigma_{\max} \leq 0,$$

$$g_3(X) = \delta(x) - \delta_{\max} \leq 0,$$

$$g_4(X) = x_1 - x_4 \leq 0,$$

$$g_5(X) = P - P_c(x) \leq 0,$$

Table 10. Comparison of accuracy, robustness, and reliability of ACCS and other metaheuristics regarding the spring design problem.

Algorithm	Best	Mean	Worst	SD
GA	0.012681	0.012742	0.012973	5.90e-05
CPSO	0.012675	0.01273	0.012924	5.20e-05
ESs	0.012698	0.013461	0.16485	9.66e-04
IACO	0.012643	0.01272	0.012884	3.49e-05
CSS	0.012638	0.012852	0.013626	8.36e-05
BA	0.012665	0.013501	0.016895	0.00142
WEO	0.012665	0.012669	0.012697	0.000006
ACCS	0.012665	0.012728	0.012845	4.90e-05

Table 11. Optimum designs obtained by ACCS and various metaheuristic algorithms for the spring design problem.

Methods	Design variables			$f_{\min}$
optimal	$x_1(d)$	$x_2(D)$	$x_3(N)$	
GA	0.051989	0.363965	10.89052	0.012681
CPSO	0.051728	0.357644	11.24454	0.012675
ESs	0.051643	0.35536	11.39793	0.012698
IACO	0.051865	0.3615	11.0000	0.012643
CSS	0.051744	0.358532	11.1657	0.012638
BA	0.05169	0.35673	11.2885	0.012665
WEO	0.051685	0.35663	11.2941	0.012665
ACCS	0.051758	0.35837	11.19226	0.012665

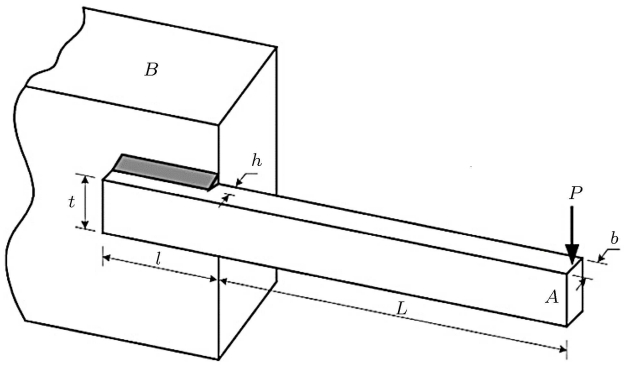

Figure 6. A welded beam system.

$$g_6(X) = 0.125 - x_1 \leq 0,$$

$$g_7(X) = 1.10471 x_2 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0,$$

$$0.1 \leq x_1 \leq 2.0, \qquad 0.1 \leq x_2 \leq 10.0,$$

$$0.1 \leq x_3 \leq 10.0, \qquad 0.1 \leq x_4 \leq 2,$$

where:

$$\tau(x) = \sqrt{(\tau')^2 + (\tau'')^2 + 2\tau'\tau''\frac{x_2}{2R}},$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \qquad \tau'' = \frac{MR}{J},$$

$$M = P\left(L + \frac{x_2}{2}\right), \qquad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

$$J = 2\left[\sqrt{2}x_1x_2\left(\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right)\right],$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}, \qquad \delta(x) = \frac{6PL^3}{Ex_4x_3^2},$$

$$P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

$P = 6000$ lb, $L = 14$ in,

$\tau_{\max} = 13600$ psi, $\sigma_{\max} = 30000$ psi,

$\delta_{\max} = 0.25$ in, $E = 30 \times 10^6$ Psi,

$G = 12 \times 10^6$ Psi.

Table 12 contains the statistical results attained by ACCS and the GA-based method for this problem [40], Co-evolutionary Particle Swarm Optimization (CPSO) [41], Evolution Strategies (ESs) [42], Charged System Search (CSS) algorithm [10], Bat Algorithm (BA) [16], and Water Evaporation Optimization (WEO) [13]. The best result is obtained by the ACCS algorithm. As is clear, ACCS results are the best, and ACCS is competitively superior to others in terms of robustness. Table 13 presents the details of the optimum design obtained by the ACCS and some other considered metaheuristic algorithms.

Table 12. Comparison of accuracy, robustness, and reliability of ACCS and other metaheuristics regarding the welded beam design problem.

Algorithm	Best	Mean	Worst	SD
GA	1.728226	1.792654	1.993408	0.074713
CPSO	1.728024	1.748831	1.782143	0.012926
ESs	1.7373	1.81329	1.994651	0.0705
IACO	1.724918	1.729752	1.775961	0.0092
CSS	1.724866	1.739654	1.759479	0.008064
WEO	1.724852	1.739437	1.818454	0.02314
ACCS	1.724852	1.724852	1.724853	1.486e-07

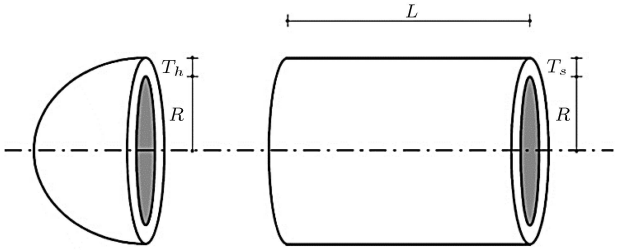

Figure 7. A pressure vessel.

3.2.3. The pressure vessel design problem

This main objective of this problem is to minimize the total cost of the materials used and the cost of the welding of the cylindrical vessel, as shown in Figure 7. The two ends of the vessel are hemispherical shaped caps. The variables of this problem consist of the thickness of the shell ($x_1 = T_s$), thickness of the head ($x_2 = T_h$), inner radius ($x_3 = R$), and length of the cylindrical section without considering the head ($x_4 = L$).

This problem is formulated as follows:

Minimize:

$$f(X) = 0.6224x_1x_3x_4 + 1.7781x_3^2x_2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3,$$

Subject to:

$$g_1(X) = 0.0193x_3 - x_1 \leq 0,$$

$$g_2(X) = 0.0095x_3 - x_3 \leq 0,$$

$$g_3(X) = 1296000 - \pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 \leq 0,$$

$$g_4(X) = x_4 - 240 \leq 0,$$

$$0.0 \leq x_1, x_2 \leq 99.0,$$

$$10.0 \leq x_3, x_4 \leq 200.$$

Many investigators optimized the present problem by different mathematical, numerical, and metaheuristic optimization approaches. The best outcomes of this problem, based on the author's knowledge, are obtained almost by metaheuristic algorithms. Accordingly, GA-based method [40], Co-evolutionary Particle Swarm Optimization (CPSO) [41], Evolution Strategies (ESs) [42], Charged System Search (CSS) algorithm [10], Bat Algorithm (BA) [16], and Water Evaporation Optimization (WEO) [13] were used successfully for solving this problem.

Table 14 shows the statistical results yielded by ACCS and other metaheuristics for this problem. The

Table 13. Optimum designs obtained by ACCS and various metaheuristic algorithms for the welded beam design problem.

Methods optimal	Design variables				$f_{\min}$
	$x_1(h)$	$x_2(l)$	$x_3(t)$	$x_4(b)$	
GA	0.205986	3.471328	9.020224	0.20648	1.728226
CPSO	0.202369	3.544214	9.04821	0.205723	1.728024
ESs	0.199742	3.61206	9.0375	0.206082	1.7373
IACO	0.2057	3.471131	9.036683	0.205731	1.724918
CSS	0.20582	3.468109	9.038024	0.205723	1.724866
WEO	0.20573	3.470489	9.036624	0.20573	1.724852
ACCS	0.205729	3.470488	9.036623	0.205729	1.7248523

Table 14. Comparison of accuracy, robustness, and reliability of ACCS and other metaheuristics regarding the pressure vessel design problem.

Algorithm	Best	Mean	Worst	SD
GA	6059.95	6177.25	6469.32	130.9297
CPSO	6061.08	6147.13	6363.8	86.4545
ESs	6059.75	6850	7332.88	426
IACO	6059.73	6081.78	6150.13	67.2418
CSS	6059.09	6067.91	6085.48	10.2564
BA	6059.71	6179.13	6318.95	137.223
WEO	6059.71	6138.61	6410.19	129.9033
ACCS	5885.71	5904.01	5967.45	20.8378

best results are obtained firstly by the CSS and, then, by BA algorithm. In addition, ACCS was able to find the same best results as those obtained by the BA, and the difference between the results obtained by the two mentioned methods was found negligible. Robustness of the ACCS is competitively better than that of other algorithms, except the CSS. Table 15 presents the optimum designs obtained by ACCS and other metaheuristic algorithms.

4. Conclusion

Nature-inspired algorithms that mimic the specific phenomena or behaviors of nature have become quite popular for use in computational optimization in recent years. Further, modeling the behavior of natural phenomena for the purpose of search and problem solving is a dynamic research area. In this article, a novel optimization algorithm was developed to solve engineering problems that are inspired by the growth of coronary arteries tree on the heart and coronary circulation system.

In other words, this article proposed a new optimization algorithm called artificial coronary circulation system algorithm, which simulates the arteries growth and the capillaries search strategies for growth in/on the heart. The present method utilizes three main behaviors: exploring capillaries (searching), creating stem-crown (bifurcation), and removing bad capillaries (pruning).

In the first phase, each capillary by the normal random walk method generated a new capillary around its current position. The second phase simulated the bifurcation procedure. By evaluating some of the worst capillaries first, the last phase mimicked the pruning process of worse capillaries based on the CGF.

The simulation and statistical outcomes of the

Table 15. Optimum designs obtained by ACCS and various metaheuristic algorithms for the pressure vessel design problem.

Methods optimal	Design variables				$f_{\min}$
	$x_1(T_s)$	$x_2(T_h)$	$x_3(R)$	$x_4(L)$	
GA	0.8125	0.4375	42.0974	176.6541	6059.946
CPSO	0.8125	0.4375	42.09127	176.7465	6061.078
ESs	0.8125	0.4375	42.09809	176.6405	6059.746
IACO	0.8125	0.4375	42.09835	176.6378	6059.726
CSS	0.8125	0.4375	42.10362	176.5727	6059.089
BA	0.8125	0.4375	42.09845	176.6366	6059.714
WEO	0.8125	0.4375	42.09844	176.6366	6059.71
ACCS	0.7783	0.38474	40.32626	199.9076	5885.715

constraint optimization problems illustrate that the suggested ACCS algorithm can be equal to or outperform other nature-inspired algorithms used in this paper. In order to evaluate the performance of the proposed algorithm, it was compared with the performance results of other popular algorithms using many benchmark engineering problems. The experimental outcomes indicate the high potential and efficiency of the ACCS algorithm. This research was conducted based on different dimensions of the research directions that could be combined with local search strategy or hybridized with other metaheuristic algorithms. The application of the suggested algorithm can be extended to other optimization problems.

References

1. Pardalos, P.M. and Floudas, C.A., *Deterministic Global Optimization: Theory, Algorithms and Applications*, Kluwer Academic Publishers, USA (2000).
2. Mirjalili, S., Mirjalili, S.M., and Lewis, A. "Grey wolf optimizer", *Adv. Eng. Softw.*, **69**, pp. 46-61 (2014).
3. Spall, J.C., *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, **65**, John Wiley & Sons (2005).
4. Kaveh, A., *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, 2nd Edn., Springer, Switzerland (2017).
5. Boussaï, D.I., Lepagnot, J., and Siarry, P. "A survey on optimization metaheuristics", *Inform. Sci.*, **237**, pp. 82-117 (2013).
6. Gogna, A. and Tayal, A. "Metaheuristics: review and application", *J. Experim. Theor. Artific. Intell.*, **25**(4), pp. 503-526 (2013).
7. Miettinen, K., *Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming*, John Wiley & Sons, Inc, New York (1999).
8. Knowles, J. and Corne, D. "The Pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimization", *CEC 99. Proc. Congress. Evol. Comput.* (1999).
9. Koza, J.R., *Genetic Programming II, Automatic Discovery of Reusable Subprograms*, MIT Press, Cambridge, MA (1992).
10. Kaveh, A. and Talatahari, S. "A novel heuristic optimization method: charged system search", *Acta Mech.*, **213**(3-4), pp. 267-289 (2010).
11. Kaveh, A., and Mahdavi, V.R. "Colliding bodies optimization: a novel meta-heuristic method", *Comput. Struct.*, **139**, pp. 18-27 (2014).
12. Hatamlou, A. "Black hole: A new heuristic optimization approach for data clustering", *Inform. Sci.*, **222**, pp. 175-184 (2013).
13. Kaveh, A. and Bakhshpoori, T. "Water evaporation optimization: A novel physically inspired optimization algorithm", *Comput. Struct.*, **167**, pp. 69-85 (2016).
14. Kennedy, J., *Particle Swarm Optimization Encyclopedia of Machine Learning*, Springer, pp. 760-766 (2010).
15. Naka, S., Genji, T., Yura, T., et al. "Hybrid particle swarm optimization based distribution state estimation using constriction factor approach", *Proc. Int. Conf. SCIS and ISIS* (2002).
16. Gandomi, A.H., Yang, X.-S., and Alavi, A.H. "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems", *Eng. Comput.*, **29**(1), pp. 17-35 (2013).
17. Yang, X.-S. "Firefly algorithms for multimodal optimization", *Int. Symp. Stochastic Algorithms* (2009).
18. Mucherino, A. and Seref, O. "Monkey search: A novel metaheuristic search for global optimization", *AIP Conf. Proc.* (2007).
19. Karaboga, D. and Basturk, B. "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", *J. Global Optimiz.*, **39**(3), pp. 459-471 (2007).
20. Mirjalili, S. and Lewis, A. "The whale optimization algorithm", *Adv. Eng. Softw.*, **95**, pp. 51-67 (2016).
21. Gandomi, A.H. and Alavi, A.H. "Krill herd: A new bio-inspired optimization algorithm", *Commun. Nonlinear Sci. Numer. Simul.*, **17**(12), pp. 4831-4845 (2012).
22. Rao, R.V., Savsani, V.J., and Vakharia, D. "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems", *Comput.-Aided Des.*, **43**(3), pp. 303-315 (2011).
23. Talatahari, S., Azar, B.F., Sheikholeslami, R., et al. "Imperialist competitive algorithm combined with chaos for global optimization", *Commun. Nonlinear Sci. Numer. Simul.*, **17**(3), pp. 1312-1319. (2012).
24. Omran, M.G., Alsharhan, S., and Clerc, M. "A modified intellects-masses optimizer for solving real-world optimization problems", *Swarm Evol. Comput.*, **41**, pp. 159-166 (2018).
25. Alatas, B. "A novel chemistry based metaheuristic optimization method for mining of classification rules", *Expert Syst. Appl.*, **39**(12), pp. 11080-11088 (2012).
26. Javidy, B., Hatamlou, A., and Mirjalili, S. "Ions motion algorithm for solving optimization problems", *Appl. Soft Comput.*, **32**, pp. 72-79 (2015).
27. Schalkoff, R.J., *Artificial Neural Networks*, **1**, McGraw-Hill, New York (1997).
28. Timmis, J., *Artificial Immune Systems: A Novel Data Analysis Technique Inspired by the Immune Network Theory*, Department of Computer Science, University of Wales, Aberystwyth (2000).
29. Liang, Y.-C. and Cuevas Juarez, J.R. "A novel metaheuristic for continuous optimization problems: Virus optimization algorithm", *Eng. Optimiz.*, **48**(1), pp. 73-93 (2016).

30. Wolpert, D.H. and Macready, W.G. "No free lunch theorems for optimization", *IEEE Trans. Evol. Comput.*, **1**(1), pp. 67-82 (1997).

31. Kaveh, A. and Shokohi, F. "A hybrid optimization algorithm for the optimal design of laterally-supported castellated beams", *Scientia Iranica, Trans Civil Eng.*, **23**(2), pp. 508-519 (2016).

32. Kaveh, A. and Talatahari, S. "Hybrid charged system search and particle swarm optimization for engineering design problems", *Eng Comput*, **28**(4), pp. 423-440 (2011).

33. Kaveh, A. and Ilchi Ghazaan, M. "A new meta-heuristic algorithm: vibrating particles system", *Scientia Iranica, Trans Civil Eng.*, **24**(2), pp. 551-566 (2017).

34. Hatamlou, A. "Heart: A novel optimization algorithm for cluster analysis", *Prog. Artific. Intell.*, **2**(2-3), pp. 167-173 (2014).

35. Keelan, J., Chung, E.M., and Hague, J.P. "Simulated annealing approach to vascular structure with application to the coronary arteries", *Royal Soc. Open Sci.*, **3**(2), p. 150431 (2016).

36. Chen, X., Niu, P., Niu, X., et al. "Growth, ageing and scaling laws of coronary arterial trees", *J. Royal Soc. Interface*, **12**(113), p. 20150830 (2015).

37. Gandomi, A.H., Yang, X.-S., Alavi, A.H. et al. "Bat algorithm for constrained optimization tasks", *Neural Comput. Appl.*, **22**(6), pp. 1239-1255 (2013).

38. Łukasik, S. and Żak, S. "Firefly algorithm for continuous constrained optimization tasks", *Lecture Notes in Computer Science*, **5796**, pp. 97-106 (2009).

39. Shareef, H., Ibrahim, A.A., and Mutlag, A.H. "Lightning search algorithm", *Appl. Soft Comput.*, **36**, pp. 315-333 (2015).

40. Coello, C.A. and Mezura Montes, E. "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection", *Adv. Eng. Inform.*, **16**(3), pp. 193-203 (2002).

41. He, Q. and Wang, L. "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", *Eng. Appl. Artific. Intell*, **20**(1), pp. 89-99 (2007).

42. Mezura-Montes, E. and Coello, C.A.C. "An empirical study about the usefulness of evolution strategies to solve constrained optimization problems", *Int. J. General Syst.*, **37**(4), pp. 443-473 (2008).

Appendix

CS:
http://www.mathworks.com/matlabcentral/fileexchange/29809-cuckoo-search-cs-algorithm

LSA:
https://www.mathworks.com/matlabcentral/fileexchange/ 54181-lightning-search-algorithm-lsa-

Firefly:
https://www.mathworks.com/matlabcentral/fileexchange/29693-firefly-algorithm

Biographies

Ali Kaveh was born in 1948 in Tabriz, Iran. After graduating from the Department of Civil Engineering at the University of Tabriz in 1969, he continued his studies on Structures at Imperial College of Science and Technology at London University and received his MSc, DIC, and PhD degrees in 1970 and 1974, respectively. He then joined the Iran University of Science and Technology. Professor Kaveh is the author of 625 papers published in international journals and 165 papers presented at national and international conferences. He has authored 23 books in Farsi and 10 books in English published by Wiley, RSP, American Mechanical Society, and Springer.

Mohsen Kooshkebaghi is a PhD candidate at the Civil Engineering Department of Science and Research Branch, Islamic Azad University, Tehran, Iran. He received the BSc degree in 2009 and MSc degree in 2012. His main expertise and experience lie in the field of structural engineering and optimal design of structures. He is interested in metaheuristic algorithms and their applications.