Knowledge and Information Systems
https://doi.org/10.1007/s10115-018-1253-3

REGULAR PAPER

Flow Regime Algorithm (FRA): a physics-based meta-heuristics algorithm

Mojtaba Tahani[1] · **Narek Babayan**[1]

Received: 19 September 2017 / Revised: 12 June 2018 / Accepted: 14 July 2018
© Springer-Verlag London Ltd., part of Springer Nature 2018

Abstract
In this research study, a physics-based optimization algorithm, namely Flow Regime Algorithm (FRA) is proposed. The main sources of inspiration are classical fluid mechanics and flow regimes. The flow regime usually is being divided into two categories which are laminar and turbulent flows. Reynolds number is the parameter which defines that the flow regime is laminar or turbulent. In this research study, a similar number to Reynolds has been defined which indicates the search type (global or local) of the algorithm and is called search type factor. For the purpose of developing the local and global searches equations, the concept of boundary layer in fluid mechanics has been used. The performance of the proposed algorithm has been evaluated using 26 benchmark functions and has been compared with seven popular and well-known algorithms which are simulated annealing, particle swarm optimization, firefly algorithm, cuckoo search, flower pollination algorithm, krill herd and monarch butterfly. Finally, the heat wheel optimization problem and horizontal axis marine current turbine (tidal turbine) problem, which are real-case engineering problems, have been solved using FRA. The results indicated that FRA can be a great candidate in solving complex engineering problems.

Keywords Flow Regime Algorithm · Meta-heuristic · Optimization · Heat wheel · Tidal turbine

List of symbols

D_h	Hydraulic diameter (m)
G	Mass flux (kg/s/m^2)
L	Characteristics length (m)
$Levy$	Generated number using Levy distribution
Maxit	Maximum number of iterations
NTU	Number of transfer units
$Rand$	Generated number using Gaussian distribution
Re	Reynolds number

✉ Narek Babayan
n.babayan@ut.ac.ir

[1] Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran

Published online: 31 August 2018

STF	Search type factor
V	Velocity (m/s)
g	Global best solution
h	Channel height (m)
β	Specific area (m^2/m^3)
γ	Scaling factor
δ	Matrix thickness (m)
δ_L	Laminar boundary layer thickness
δ_T	Turbulent boundary layer thickness
ε	Efficiency of heat wheel
μ	Viscosity (Pa s)
ρ	Density (kg/m^3)
σ	Porosity

1 Introduction

In recent years, optimization has become an important subject in almost every scientific and engineering problems. The purpose of each design is obtaining the global optimum combination of variables in order to achieve the best performance. In other words, each system that must be optimized has different variables which affect the system or the objective function itself [33, 52]. In several complex problems, this cannot be achieved by classical methods and meta-heuristic optimization techniques must be utilized. Some examples of application of meta-heuristic techniques are renewable energy systems design [5, 16, 40–43], structures optimization in civil engineering [31], power plant and cogeneration cycles [2, 7], antenna problems [10], etc.

Meta-heuristic optimization algorithms can have different sources of inspiration. Most of them are inspired from nature. For example, Mirjalili [27] proposed the dragon fly algorithm inspired from their static and dynamic swarming behaviors. Li et al. [26] proposed the animal migration algorithm. In this algorithm, two processes are simulated. The first one is the moving behavior of the animal groups from the current position to the new position, and in second one, the algorithm simulates the group leaving characteristics of some animals. Yazdani and Jolai [51] proposed the lion optimization algorithm (LOA) which is based on the cooperation characteristics of lions. Mirjalili [29] proposed the ant lion optimization algorithm which is based on hunting mechanism of ant lions. Whale optimization algorithm was proposed by Mirjalili and Lewis [30] based on the bubble-net hunting strategy. This algorithm mimics the social behavior of whales. Yu and Li [53] proposed the social spider algorithm based on the physics of web vibration for detecting the position of prey. Gandomi and Alavi [14] proposed the krill herd optimization algorithm, based on the herding behavior of krill individuals. Mirjalili [28] proposed the moth-flame algorithm inspired from the navigation method of moths. Artificial algae algorithm was proposed by Uymaz et al. [45] based on the living behavior of photosynthetic species of microalgae, etc. Providing a detail and also comprehensive review of nature inspired optimization algorithms is a difficult task, and it requires a great and deep literature review. All algorithms have their own advantages and also disadvantages but they are common in one property, exploitation and exploration. All algorithms must search the space globally and also locally. Formulation of this process and also creating balance between these two types of searches depends on the source of inspiration and also assumptions which are being used in developing process. In this research study, a

physics-based meta-heuristic optimization algorithm is proposed. Some meta-heuristic optimization algorithms are inspired from a physical or a chemical phenomenon. Here a brief introduction to these algorithms is provided.

1.1 State of art

Table 1 illustrates a brief literature review from physics-based optimization algorithms. As it can be seen, most of the algorithms presented in Table 1 are based on the Newton gravitational law. Also it can be seen that during 2005–2010 period, twelve algorithms are proposed which presents the 54% of the available algorithms. During last years, physics-based algorithms are not proposed, while they can be very suitable for solving nonlinear optimization problems. Flow Regime Algorithm (FRA) is the proposed optimization procedure in this research study. It is inspired from the laminar and turbulence regimes. As it is known in fluid mechanics science, according to the ratio of inertial to viscous forces, the flow can experience two types of regimes: namely laminar and turbulence. In the proposed algorithm, laminar regime is responsible for local search and turbulence is responsible for global search. As in fluid mechanics, the Reynolds number is responsible to identify the regime of the flow, in this research study also a similar number is defined to identify the search type. In this manuscript, first, the basic concepts of laminar and turbulent flows are explained and then the proposed algorithm is presented. After that experimental results of the algorithm are discussed. The performance of the algorithm is compared with other popular optimization algorithms. At the end, two real-case optimization problems, which are the design optimization of heat wheel and horizontal axis marine current turbine, are solved using FRA.

2 Inspiration

2.1 An introduction to laminar and turbulent flow regimes

To those who are familiar with fluid mechanics, indicating the flow regime is one of the most important factors in predicting the flow characteristics such as pressure reduction, boundary layer. Two general flow regimes are defined according to the value of a non-dimensional number called Reynolds number, which is the ratio of inertial forces to viscous forces. The Reynolds number is defined as follows:

$$Re = \frac{\rho V L}{\mu} \tag{1}$$

where ρ is the fluid density, μ presents the dynamic viscosity, V is the free stream velocity and L is the characteristics length. In an internal flow, L is the pipe diameter and in an external flow presents the distance from the starting point of boundary layer. In an internal flow, such as water flow through a pipe, the critical Reynolds number is equal to 2300. Meaning that below 2300 the flow regime is only laminar and above that the flow is becoming turbulence. In an external flow such as the water flow over a flat plate, the critical Reynolds number is being considered between 5×10^5 and 10^7. In a laminar flow, the fluid layers slide over one another in a well-ordered pattern, while in a turbulent flow a mixing of eddies occurs and a complicated phenomenon is being generated. Figure 1 illustrates the difference between laminar and turbulent flows by velocity versus time graph. As it can be seen in a steady laminar flow, the velocity of fluid at a specific position is constant with time, while in a

Table 1 Literature review of physics-based optimization algorithms

No.	Title	Authors	Years	Inspiration	Refs.
1	Ion motion algorithm	Javidy et al.	2015	This algorithm is inspired from the ions motion and the interactions between anion and cations	[20]
2	Ray optimization	Kaveh et al.	2013	This algorithm is inspired from the relation between incidence and fraction angles according to Snell's law	[22]
3	Black hole	Hatamlou	2013	This algorithm is inspired from the black hole concept. Meaning that the best solution is the black hole and other candidate solutions are stars	[17]
4	Water cycle algorithm	Eskandar et al.	2012	This algorithm is inspired from the characteristics of the rivers and streams that are flowing to the seas	[12]
5	Galaxy-based algorithm	Shah Hosseini	2011	This algorithm is inspired from the spiral arms of galaxies	[36]
6	Water flow algorithm	Hieu	2011	This algorithm is based on the spreading characteristics of water droplets (water always moves from higher places to lower ones) and also their erosion capability to overcome obstacles	[18]
7	Charged system search algorithm	Kaveh and Talatahari	2010	This algorithm is inspired from physics and also mechanics. Coulomb and Newton laws are basic principles of this algorithm. All forces are determined using electrostatics laws and the quality of motion is determined using Newton law	[21]
8	Gravitational search algorithm	Rashedi et al.	2009	This algorithm is inspired from the law of gravity and also the interactions between masses	[34]
9	Intelligent water drops algorithm	Shah-Hosseini	2009	This algorithm is inspired from the behavior of rivers in finding the best path	[38]
10	Artificial physics algorithm	Xie et al.	2009	The main inspiration of this algorithm, similar to other gravitation-based algorithms, is the Newton law. The solutions with larger mass attract the solutions with small mass	[47]

Table 1 continued

No.	Title	Authors	Years	Inspiration	Refs.
11	River formation dynamics algorithm	Rabanal et al.	2008	This algorithm is based on river formation by water and also its eroding and sediments depositing behavior	[32]
12	Big Crunch algorithm	Kripka and Kripka	2008	The algorithm is inspired from the cosmological theory known as closed universe. According to this theory, if the generated energy by the Big Bang is smaller than the gravitational energy, then the expansion will be stopped	[25]
13	Magnetic optimization algorithm	Tayarani and Akbarzadeh	2008	This algorithm is inspired from the principles of magnetic field theory	[44]
14	Central force optimization algorithm	Formato	2007	This algorithm is based on the motion of objects or probes which are moving under the influence of gravity	[13]
15	Integrated radiation algorithm	Chuang and Jiang	2007	This algorithm is based on the Einstein's theory of general relativity	[8]
16	Big Bang–Big Crunch algorithm	Erol and Eksin	2006	The energy dissipation (Big Bang) and center of mass (Big Crunch) are the main inspiration sources of this algorithm. In the Big Bang phase, random solutions are generated and in the Big Crunch phase, all solutions are drawn into an order	[11]
17	Space gravitational algorithm	Hsiao et al.	2005	This algorithm uses the Einstein's theory of relativity and the concept of gravitational field for global optimization purpose	[19]
18	Particle collision algorithm	Sacco and de Oliveira	2005	This algorithm is based on the nuclear collision reactions, particularly scattering and absorption	[35]
19	Electromagnetism-like algorithm	Birbil and Fang	2003	The algorithm is based on the attraction–repulsion mechanism of the sample points	[6]
20	Hysteretic optimization	Zarand et al.	2002	This algorithm is based on the demagnetization process of magnetic materials	[54]

Table 1 continued

No.	Title	Authors	Years	Inspiration	Refs.
21	Harmony search	Geem and Kim	2001	This algorithm is based on the improvisation of the music player	[15]
22	Simulated annealing	Kirkpatrick et al.	1983	The algorithm is inspired from the annealing process of molten metals and it uses the Boltzmann's probability function in order to create global search	[24]

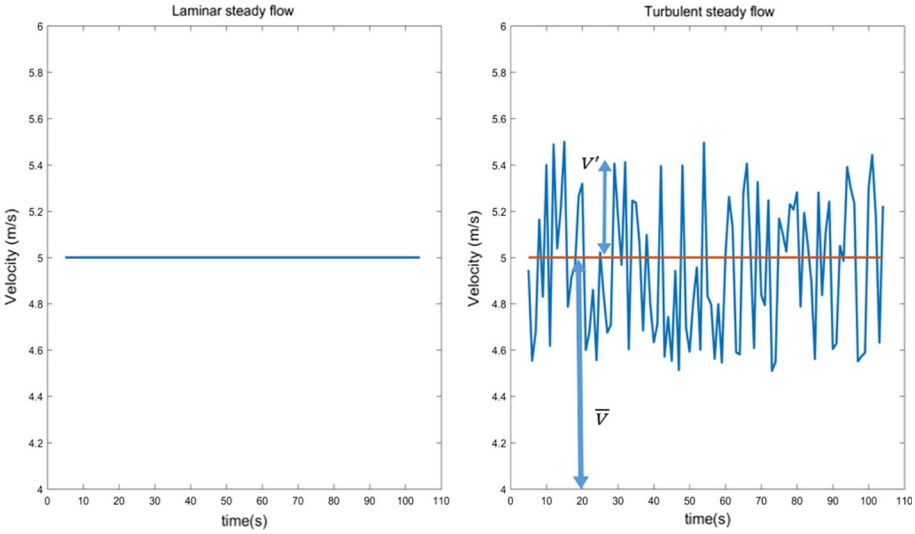

Fig. 1 A comparison between steady laminar and turbulent flows

steady turbulence flow, the mean time average velocity ($\bar{V}$) is constant and the absolute velocity of the fluid is achieved by superposing the fluctuating velocity (V') with $\bar{V}$.

2.2 Boundary layer

When a fluid is flowing on a flat plate, due to viscous effects, a velocity profile is being generated. The velocity value on the surface is equal to zero and at a specific height from the surface which is being called the boundary layer thickness is equal to 99% of the freestream velocity. The thickness of boundary layer varies with distance from the starting point of the boundary layer. Its thickness is equal to zero at the initial point. According to the value of Reynolds number, the boundary layer can be laminar or turbulent. Following equations are being utilized for calculating the boundary layer thickness in laminar and turbulence regions [39]. Here the critical Reynolds number is considered equal to 3.2×10^5. Figure 2 presents a schematic of a boundary layer. As it can be seen first, the flow regime in boundary layer

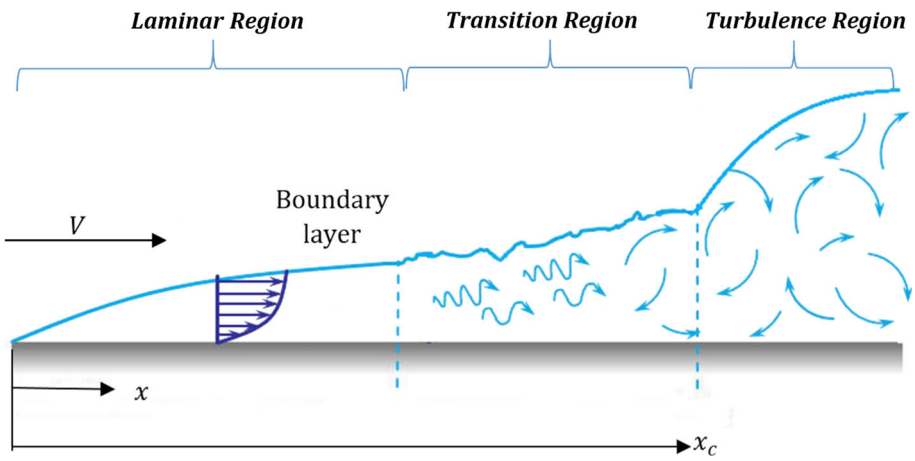

Fig. 2 A schematic of boundary layer

is laminar and then by increasing the distance from the starting point, the inertial forces get dominant and the flow begins to get turbulent.

$$\begin{cases} \delta_{\mathrm{L}} = \frac{4.96x}{\sqrt{Re_x}} & Re_x < 3.2 \times 10^5 \quad \text{(Laminar Flow)} \\ \delta_{\mathrm{T}} = \frac{0.37x}{\sqrt[5]{Re_x}} & Re_x \geq 3.2 \times 10^5 \quad \text{(Turbulent Flow)} \end{cases} \tag{2}$$

3 Flow Regime Algorithm (FRA)

In this research study, a physics-based meta-heuristics algorithm is presented according to the laminar and turbulent flow regimes. As it is known, each meta-heuristic algorithm has three main characteristics which are diversification or exploration, intensification or exploitation and elitism. Creating a great balance between diversification and intensification which are responsible for global and local searches, respectively, has great influence on the algorithm performance. In this research study, a similarity is established between flow regime and local and global searches of the algorithm. Therefore, a similar number to Reynolds number is proposed which selects the type of the search of the algorithm. Following assumptions are considered in developing of the algorithm:

- Laminar and turbulent flows are responsible for local and global searches, respectively.
- The optimum solution is the boundary layer starting point.

In the proposed algorithm, the domain of the fluid flow is being considered the solution space and each fluid particle is being considered as a candidate solution. According to the mentioned points, the FRA is a population-based algorithm and each fluid particle in a N_{var}-dimensional solution space is being represented as follows:

$$\text{fluid particle} = [x_1, x_2, x_3, \ldots, x_{N\mathrm{var}}] \tag{3}$$

As it is known, the first step of each optimization algorithm is generation of the initial population. The initial population consists of N_{pop} fluid particles, and at this stage, the fitness function of each particle must be evaluated. After evaluating each particle cost or fitness

function, the solutions are being sorted and the global best particle is selected. Starting the main loop of the algorithm is the next step.

In the main loop of the algorithm, for a specific number of iterations such as Maxit, new solutions are generated and worst solutions are being replaced with better solutions. For each particle of the fluid, the first step is identification of the required search type. Generally, at first iterations, the algorithm must search the space globally and then by increasing the number of iterations, the local search must be intensified. As was mentioned earlier, for the purpose of creating balance between local and global searches, a similar number to Reynolds has been defined, namely search type factor (STF):

$$\mathrm{STF}^i = 3.2 \times 10^6 \times \frac{\mathrm{Maxit}}{n} \times \frac{\left\| \left(g_n - \mathrm{particle}^i\right) \right\|}{\left\| \left(\mathrm{particle}^j - \mathrm{particle}^k\right) \right\|} \tag{4}$$

where g_n presents the global best solution which has been found till nth iteration, n is the number of current iterations, i, j and k present the number of solutions. STF is being used for identifying the search strategy. Following criteria is being used for this purpose:

$$\begin{cases} \mathrm{STF} \geq 3.2 \times 10^5 \rightarrow \text{Global search} \\ \mathrm{STF} < 3.2 \times 10^5 \rightarrow \text{Local search} \end{cases} \tag{5}$$

The reason of using the number 3.2×10^6 in definition of STF is creating a similarity between the critical Reynolds number and critical STF. As was mentioned earlier, all algorithms, generally, fist must have a global search for the purpose of identifying the search space and then a local search for finding available better solutions around the global optima. This is the reason of utilization the term $\frac{\mathrm{Maxit}}{n}$. According to the presented equation for STF, this term increases the value of STF at the first iterations, and therefore, it increases the probability of the global search, and then, at higher number of iterations, it increases the local search probability by decreasing the value of STF.

The logic behind the definition of the last ratio in equation of STF is as follows. This term represents the ratio of two Euclidean norms. The first is the distance between the ith particle with global best solution, and the second is the distance between two randomly selected particles. This ratio can be equal or greater than one or less than one. If it is greater than one, it means that the distance between the two randomly selected particles (j and k) is less than the distance between the ith particle and global best particle (the ith particle is far from the boundary layer starting point). This means that solutions j and k are moving toward each other or they are near to each other in which it is not desirable in searching the solution space because in this case the probability of exploring new places will decrease. Therefore, a global search (turbulent flow) is required in this case, and because the value of this ratio is greater than one, it will increase the STF. On the other hand, if the ratio of these two norms is less than one, it means that the ith solution (or particle) is in the vicinity of the global best solution (ith solution is near to the boundary layer starting point); therefore, it would be better to search locally for discovering better solutions if they exist. For the purpose of developing the required equations for global and local searches, the turbulence and laminar boundary layer thickness equations are utilized. Figure 3 presents two possible positions for ith particle. If STF is greater than 3.2×10^5, it means that ith particle is far from boundary layer starting point (global best solution); therefore, it is in turbulence region of boundary layer, and therefore, the equations relevant to the turbulence regime will be utilized. On the other hand, if STF is less than 3.2×10^5, it means that ith particle is near to the boundary layer starting point therefore, laminar equations can be utilized. Following equations present

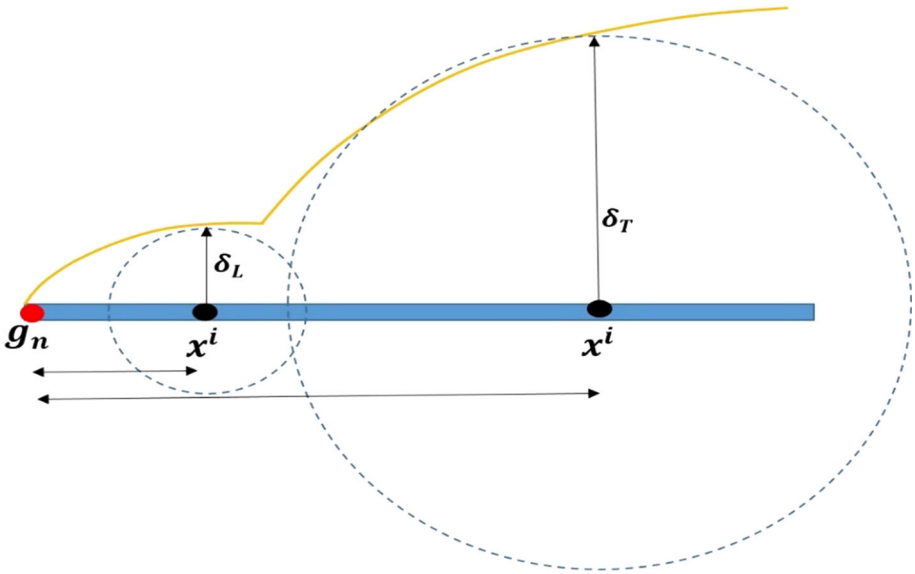

Fig. 3 Two possible positions for *i*th particle

the global and local searches required formulas. Equation (6) is for global search, and Eq. (7) is for local search.

$$x_{n+1}^i = x_n^i + \gamma Levy \times \left(g_n - x_n^i\right) \times \frac{0.37}{\sqrt[5]{\text{STF}}} \tag{6}$$

$$x_{n+1}^i = x_n^i + \gamma Rand \times \left(g_n - x_n^i\right) \times \frac{4.96}{\sqrt{\text{STF}}} \tag{7}$$

In the above equations, $\gamma Levy$ and $\gamma Rand$ are utilized in order to increase the randomness and therefore the global search of the algorithm. $Levy$ is a number generated by Levy distribution function, and $Rand$ is a random number generated by Gaussian distribution. γ is a scaling factor which is considered equal to 30% of the difference between the upper and lower bounds of the variables. Figure 4 illustrates the pseudocode of the proposed algorithm. As it can be seen after initialization, at each iteration, and for each variable, first the STF is being calculated. According to the value of STF, the search type is being selected. According to the pseudocode of the algorithm, simplicity is one of properties of FRA and it can be easily applied for optimization problems.

4 Results and discussion

In this research study, for the purpose of evaluating the performance of the proposed algorithm, 26 popular benchmark functions were utilized. Also a real-case engineering problem, namely heat wheel optimization, has been solved using FRA algorithm. All simulations and experimental test are carried out using MATLAB software and a Lenovo Z51 laptop computer with Intel core i7-5500U CPU at 2.4 GHz and 8 GB memory. In order to compare the FRA performance with existing algorithms, seven well-known meta-heuristic algorithms were selected which are:

Begin

Objective **min** *or* **max** $f(x)$, $x=(x_1, x_2, x_3, \ldots, x_m)$

Initialize a population of $\boldsymbol{m}$ *particles between upper and lower bounds of variables*

Find the best solution $\boldsymbol{g}$ *in the initial population*

Define a stopping criterion (either a fixed number of iterations or accuracy)

While *(n<Maximum Iteration)*

For *i=1:m (all particles)*

Select two random particles (j and k) and calculate the STF using equation (4)

If $STF \geq 3.2\times10^5$

Update the position of particle i using equation (6)

Else

Update the position of particle I using equation (7)

End if

End For

Evaluate the objective function for all solution and updating $\boldsymbol{g}$

End While

Report the best solution

End

Fig. 4 The pseudocode of the algorithm

(a) *Simulated annealing* [24] As shown in Table 1, SA is inspired from the cooling process of the molten metal. In this algorithm, cooling temperature acts as a control parameter. At the first iteration when the global search is more important than local search, temperature is high and the probability of selecting solutions with not much good objective functions is also high. By increasing the number of iterations, the molten metal is being cooled according to a specific temperature decreasing function and the selection probability of the solutions with no desired objective functions decreases.

(b) *Particle Swarm Optimization (PSO)* [23] PSO is one of the most popular meta-heuristic optimization algorithms which is mainly inspired from the bird flocking and also fish schooling behavior. In PSO, each particle has a velocity which defines its displacement direction and is being updated according to its previous value, the best position which has been found so far by the particle and also the global best position which has been found by the entire particles.

(c) *Firefly algorithm* [48] This algorithm is inspired from the flashing characteristics of fireflies. The brighter fireflies will attract less bright fireflies, and the optimum solution is the brightest firefly.

(d) *Cuckoo search* [50] This algorithm is based on the searching behavior of cuckoos in finding suitable nests where they can lay their eggs, and the host bird cannot figure it out. In other words, the algorithm is inspired form the parasitic behavior of cuckoos.
(e) *Flower pollination algorithm* [49] This algorithm is inspired from the pollination process of flowers. Two types of pollinations, namely biotic and abiotic, are considered as the local and global searches, respectively, in this algorithm.
(f) *Krill herd algorithm* [14] This algorithm is inspired from the herding behavior of krill individuals. In this algorithm, Lagrangian model has been used for simulating their motion and their position is being updated by considering three main actions: (1) the movement that is created by other krill individuals, (2) foraging activity and (3) the created random diffusion.
(g) *Monarch butterfly algorithm* [46] This algorithm is inspired by the migration behavior of monarch butterflies between two lands: northern USA (first land) and Mexico (second land). Two operators, namely migration operator and adjusting operator, are utilized for updating the position of butterflies (solutions) or generating new ones.

In this section, first the experimental studies on the proposed algorithm are provided and then the heat wheel optimization problem is explained and the results are discussed.

4.1 Experimental results

Table 2 indicates the optimization benchmark functions which are utilized for examining the performance of the proposed algorithm. Ten runs have been examined for each function and the number of population for each algorithm is selected to be 30 and the maximum number of iterations is selected equal to 2000.

For the purpose of comparing the performance of FRA with other selected algorithms, following parameters were utilized for each algorithm.

- *Simulated annealing* Inner loop number of iterations ($p_{\max}$) = 200, temperature reduction factor (c) = 0.9.
- *Particle swarm optimization* Inertia weight (ω) = 0.7298, personal and global learning coefficients (c_1 and c_2) = 1.4962.
- *Firefly algorithm* Light absorption coefficient (γ) = 1, power of distance (m) = 2, attraction coefficient base value (β_0) = 1.5, mutation vector coefficient (α) = 0.2.
- *Cuckoo search* Fraction of abandoned nests (p_a) = 0.25, crossover parameter (α) = 0.1.
- *Flower pollination algorithm* Switching probability (p_a) = 0.7, Levy exponent (β) = 1.5.
- *Krill herd* Maximum induced speed ($N_{\max}$) = 0.01, inertia weight (ω_n) = 0.1, singularity preventing parameter (ε) = 0.01, foraging speed (V_f) = 0.02, inertia weight of the foraging motion (ω_f) = 0.1, diffusion maximum amount ($D_{\max}$) = 0.01, diffusion minimum amount ($D_{\min}$) = 0.002, the coefficient of time (C_t) = 0.5.
- *Monarch butterfly algorithm* peri = 1.2, period of migration (p) = 5/12.

Table 3 summarizes the results of the experiment. In the table, Max represents the worst value of the objective function that the algorithm has found, Min is the best value of the objective function, AVG presents the average value of the objective function during 10 runs, and STD is the standard deviation. For each benchmark function, algorithms are sorted according to their AVG value and their rank has been mentioned in the table. As it can be seen in this table, according to the overall rank, the first place is for FRA, the second is for CS, the third is for FPA and other places are for KH, PSO, FA, SA and MBO, respectively.

Table 2 Utilized benchmark functions for validating the proposed algorithm

No.	Name	Equation	Domain	Number of variables
$F1$	Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + \exp(1)$	$[-32.768, 32.768]$	2
$F2$	Beale	$f(x) = (1.5 - x_1 + x_1x_2)^2 + \left(2.25 - x_1 + x_1x_2^2\right)^2 + \left(2.625 - x_1 + x_1x_2^3\right)^2$	$[-4.5, 4.5]$	2
$F3$	Bohachevsky	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	$[-100, 100]$	2
$F4$	Booth	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	$[-10, 10]$	2
$F5$	Colville	$f(x) = 100\left(x_1^2 - x_2\right)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90\left(x_3^2 - x_4\right)^2 + 10.1\left((x_2 - 1)^2 + (x_4 - 1)^2\right) + 19.8(x_2 - 1)(x_4 - 1)$	$[-10, 10]$	4
$F6$	Cross-in-Tray	$f(x) = -0.0001\left(\left\lvert\sin(x_1)\sin(x_2)\exp\left(\left\lvert 100 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert\right)\right\rvert + 1\right)^{0.1}$	$[-10, 10]$	2
$F7$	Drop-Wave	$f(x) = -\frac{1+\cos\left(12\sqrt{x_1^2+x_2^2}\right)}{0.5\left(x_1^2+x_2^2\right)+2}$	$[-5.12, 5.12]$	2
$F8$	Easom	$f(x) = -\cos(x_1)\cos(x_2)\exp\left(-(x_1 - \pi)^2 - (x_2 - \pi)^2\right)$	$[-100, 100]$	2
$F9$	Egg holder	$f(x) = -(x_2 + 47)\sin\left(\sqrt{\left\lvert x_2 + \frac{x_1}{2} + 47\right\rvert}\right) - x_1\sin\left(\sqrt{\lvert x_1 - (x_2 + 47)\rvert}\right)$	$[-512, 512]$	2
$F10$	Goldstein–Price	$f(x) = \left[1 + (x_1 + x_2 + 1)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \times \left[30 + (2x_1 - 3x_2)^2\left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	$[-2, 2]$	2
$F11$	Gramacy and Lee	$f(x) = \frac{\sin(10\pi x)}{2x} + (x - 1)^4$	$[0.5, 2.5]$	1
$F12$	Griewank	$f(x) = \sum_{i=1}^{d}\frac{x_i^2}{4000} - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	2

Table 2 continued

No.	Name	Equation	Domain	Number of variables
$F13$	Hartmann 3-D	$f(x) = -\sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{3} A_{ij}\left(x_j - P_{ij}\right)^2\right)$ $\alpha = \left(1.0,\ 1.2,\ 3.0,\ 3.2\right)^{\mathrm{T}}$ $A = \begin{pmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix}$ $P = 10^{-4}\begin{pmatrix} 3689 & 1170 & 2673 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{pmatrix}$	[0, 1]	3
$F14$	Holder table	$f(x) = -\left\lvert \sin(x_1)\cos(x_2)\exp\left(\left\lvert 1 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert\right)\right\rvert$	$[-10, 10]$	2
$F15$	Langermann	$f(x) = \sum_{i=1}^{5} c_i \exp\left(-\frac{1}{\pi}\sum_{j=1}^{2}\left(x_j - A_{ij}\right)^2\right)\cos\left(\pi \sum_{j=1}^{2}\left(x_j - A_{ij}\right)^2\right)$ $c = (1, 2, 5, 2, 3)$ $A = \begin{pmatrix} 3 & 5 \\ 5 & 2 \\ 2 & 1 \\ 1 & 4 \\ 7 & 9 \end{pmatrix}$	[0, 10]	2
$F16$	Levy N.13	$f(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2\left[1 + \sin^2(3\pi x_2)\right] + (x_2 - 1)^2\left[1 + \sin^2(2\pi x_2)\right]$	$[-10, 10]$	2
$F17$	Michalewicz	$f(x) = -\sum_{i=1}^{2} \sin(x_i)\sin^{20}\left(\frac{ix_i^2}{\pi}\right)$	$[0, \pi]$	2
$F18$	Rastrigin	$f(x) = 20 + \sum_{i=1}^{2}\left[x_i^2 - 10\cos(2\pi x_i)\right]$	$[-5.12, 5.12]$	2

Table 2 continued

No.	Name	Equation	Domain	Number of variables
$F19$	Rosenbrock	$f(x) = 100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2$	$[-5, 10]$	2
$F20$	Schwefel	$f(x) = 837.9658 - \sum_{i=1}^{2} x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-500, 500]$	2
$F21$	Shekel	$f(x) = -\sum_{i=1}^{10}\left(\sum_{j=1}^{4}\left(x_j - C_{ji}\right)^2 + \beta_i\right)^{-1}$ $\beta = \frac{1}{10}(1, 2, 2, 4, 4, 6, 3, 7, 5, 5)^T$ $C = \begin{pmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.0 \end{pmatrix}$	$[0, 10]$	4
$F22$	Shubert	$f(x) = \left(\sum_{i=1}^{5} i cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$	$[-5.12, 5.12]$	2
$F23$	Styblinski–Tang	$f(x) = \frac{1}{2}\sum_{i=1}^{2}\left(x_i^4 - 16x_i^2 + 5x_i\right)$	$[-5, 5]$	2
$F24$	Sum square	$f(x) = \sum_{i=1}^{2} i x_i^2$	$[-10, 10]$	2
$F25$	Trid	$f(x) = \sum_{i=1}^{6}(x_i - 1)^2 - \sum_{i=2}^{6} x_i x_{i-1}$	$[-36, 36]$	6
$F26$	Zakharov	$f(x) = \sum_{i=1}^{2} x_i^2 + \left(\sum_{i=1}^{2} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{2} 0.5 i x_i\right)^4$	$[-5, 10]$	2

Table 3 Summary of the experimental results

No.	Algorithms	SA	PSO	FA	CS	FPA	KH	MBO	FRA
*F*1	Max	9.64E−04	8.88E−16	1.29E−06	8.88E−16	8.88E−16	8.88E−16	4.8841	8.88E−16
	Min	2.77E−05	8.88E−16	2.74E−07	8.88E−16	8.88E−16	8.88E−16	2.122E−08	8.88E−16
	AVG	2.16E−04	8.88E−16	6.20E−07	8.88E−16	8.88E−16	8.88E−16	1.2624	8.88E−16
	STD	2.84E−04	0	2.76E−07	0	0	0	1.6699	0
	Rank	8	1	6	1	1	1	7	1
*F*2	Max	7.88	0	1.47E−14	3.19E−06	1.02E−29	0.76207	1.0953	1.94E−16
	Min	0.01	0	3.28E−16	9.07E−19	0	0	4.367E−16	0
	AVG	2.26	0	5.45E−15	3.19E−07	1.44E−30	0.2286	0.2773	2.53E−17
	STD	2.53	0	4.47E−15	9.57E−07	3.187E−30	0.3492	0.4172	6.05E−17
	Rank	8	1	4	5	2	6	7	3
*F*3	Max	662.23	0	2.99E−10	0	0	0	4.13E−01	0
	Min	0.48	0	2.09E−12	0	0	0	1.2E−12	0
	AVG	117.81	0	4.68E−11	0	0	0	4.13E−02	0
	STD	202.73	0	8.57E−11	0	0	0	0.12388	0
	Rank	8	1	6	1	1	1	7	1
*F*4	Max	2	0	2.58E−13	0	0	0	4.0696	0
	Min	2	0	1.27E−15	0	0	0	2.62E−16	0
	AVG	2	0	4.48E−14	0	0	0	0.4139	0
	STD	0	0	7.19E−14	0	0	0	1.2187	0
	Rank	8	1	6	1	1	1	7	1
*F*5	Max	234,480.8	2.63E−05	1.59E−07	6.011	0.1822	59.3545	120.72	2.62E−04
	Min	92.4	2.81E−07	1.58E−10	0.00037	0.0015	0.27855	2.784E−06	1.08E−07
	AVG	62,620.1	6.83E−06	3.82E−08	0.77754	0.0856	9.33596	33.745	4.72E−05

Table 3 continued

No.	Algorithms	SA	PSO	FA	CS	FPA	KH	MBO	FRA
	STD	68,080.5	9.11E−06	4.37E−08	1.74844	0.0629	16.8274	41.888	8.16E−05
	Rank	8	2	1	5	4	6	7	3
*F*6	Max	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626
	Min	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626
	AVG	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626	−2.0626
	STD	0	0	0	0	0	0	0	0
	Rank	1	1	1	1	1	1	1	1
*F*7	Max	−1	−0.93625	−0.936	−0.93625	−0.93625	−0.93625	−0.78575	−1
	Min	−1	−1	−1	−1	−1	−1	−1	−1
	AVG	−1	−0.9936	−0.981	−0.96812	−0.9554	−0.9554	−0.93395	−1
	STD	0	0.01912	0.0292	0.03188	0.02921	0.02921	0.05543	0
	Rank	1	3	4	5	6	6	8	1
*F*8	Max	−1	−1	−1	−1	−1	−8.1E−05	−1	−1
	Min	−1	−1	−1	−1	−1	−1	−−1	−1
	AVG	−1	−1	−1	−1	−1	−0.9	−1	−1
	STD	0	0	0	0	0	0.299	0	0
	Rank	1	1	1	1	1	8	1	1
*F*9	Max	−734.66	−558.52	−716.165	−786.526	−894.5789	−716.6715	−716.672	−956.92
	Min	−734.66	−894.57	−959.641	−959.641	−959.641	−959.641	−959.641	−959.641
	AVG	−734.66	−791.76	−806.062	−928.754	−940.122	−875.9113	−884.844	−959.34
	STD	0	127.83	82.248	54.4602	29.815	76.1922	68.0521	0.8559
	Rank	8	7	6	3	2	5	4	1
*F*10	Max	136.96	3	3	3	3	3	32.493	3
	Min	4.87	3	3	3	3	3	3	3
	AVG	48.18	3	3	3	3	3	10.6105	3

Table 3 continued

No.	Algorithms	SA	PSO	FA	CS	FPA	KH	MBO	FRA
	STD	39.09	0	0	0	0	0	11.1992	0
	Rank	8	1	1	1	1	1	7	1
*F*11	Max	−0.869	−0.6632	−0.8690	−0.8690	−0.66328	−0.8690	−0.43446	−0.8690
	Min	−0.869	−0.8690	−0.8690	−0.8690	−0.8690	−0.8690	−0.86901	−0.8690
	AVG	−0.869	−0.7867	−0.8690	−0.8690	−0.8484	−0.8690	−0.78441	−0.8690
	STD	0	0.1008	0	0	0.06172	0	0.14209	0
	Rank	1	7	1	1	8	1	6	1
*F*12	Max	9.59E−06	0.0197	0.0024662	0.02219	0.088743	0.08878	0.55982	0.0074
	Min	2.34E−09	0	0.0024662	0	0	0	1.243E−14	0
	AVG	2.25E−06	0.0042	0.0024662	0.007149	0.02769	0.01233	0.1625	0.0016
	STD	3.29E−06	0.0061	0	0.005669	0.02604	0.0262	0.1832	0.0031
	Rank	1	4	3	5	6	7	8	2
*F*13	Max	−1.8996	−3.8628	−3.8628	−3.8628	−3.8628	−3.8628	−3.8441	−3.8628
	Min	−1.8996	−3.8628	−3.8628	−3.8628	−3.8628	−3.8628	−3.8628	−3.8628
	AVG	−1.8996	−3.8628	−3.8628	−3.8628	−3.8628	−3.8628	−3.85696	−3.8628
	STD	0	0	0	0	0	0	0.00689	0
	Rank	8	1	1	1	1	1	7	1
*F*14	Max	−15.1402	−9.5047	−19.2085	−19.2085	−19.2085	−19.2085	−17.1919	−19.2085
	Min	−15.1402	−19.2085	−19.2085	−19.2085	−19.2085	−19.2085	−19.2085	−19.2085
	AVG	−15.1402	−18.2381	−19.2085	−19.2085	−19.2085	−19.2085	−19.0068	−19.2085
	STD	0	2.9111	0	0	0	0	0.6049	0
	Rank	8	7	1	1	1	1	6	1
*F*15	Max	−4.127	−1.8216	−3.863	−4.1244	−3.6959	−2.1933	−2.1933	−4.0827
	Min	−4.127	−4.1558	−4.156	−4.1558	−4.1558	−4.1558	−4.1276	−4.1558
	AVG	−4.127	−3.5187	−4.097	−4.1386	−3.9520	−3.5045	−3.7212	−4.1416

Table 3 continued

No.	Algorithms	SA	PSO	FA	CS	FPA	KH	MBO	FRA
	STD	0	0.9534	0.081	0.01410	0.16795	0.8677	0.6457	0.027
	Rank	3	7	4	2	5	8	6	1
*F*16	Max	2.93E−07	1.35E−31	1.61E−12	1.35E−31	1.35E−31	1.35E−31	1.09	1.35E−31
	Min	4.45E−11	1.35E−31	2.85E−15	1.35E−31	1.35E−31	1.35E−31	4.69E−18	1.35E−31
	AVG	4.87E−08	1.35E−31	3.41E−13	1.35E−31	1.35E−31	1.35E−31	1.31E−01	1.35E−31
	STD	8.81E−08	0	4.47E−13	0	0	0	0.3223	0
	Rank	7	1	6	1	1	1	8	1
*F*17	Max	− 1.001	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.8013
	Min	− 1.001	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.8013
	AVG	− 1.001	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.8013	− 1.8013
	STD	0	0	0	0	0	0	0	0
	Rank	8	1	1	1	1	1	1	1
*F*18	Max	7.62E−07	1.98	9.95E−01	0	1.9899	0.99496	0.995	0
	Min	3.85E−11	0	7.11E−15	0	0	0	0	0
	AVG	9.19E−08	0.4965	2.98E−01	0	0.4975	0.0995	0.497	0
	STD	2.37E−07	0.6652	0.0456	0	0.6674	0.2985	0.4975	0
	Rank	3	6	5	1	8	4	7	1
*F*19	Max	1129.76	0	7.87E−12	0.6481	0.02489	0.34253	7.4478	1.38E−04
	Min	2.29	0	3.26E−14	2.49E−08	1.203E−08	0	0.0058	5.36E−12
	AVG	343.49	0	1.25E−12	0.08152	0.004538	0.05857	1.4042	1.54E−05
	STD	445.13	0	2.28E−12	0.1937	0.007954	0.10285	2.1622	4.34E−05
	Rank	8	1	2	6	4	5	7	3
*F*20	Max	2.37E+02	3.36E+02	2.17E+02	1.18E+02	1.18E+02	1.18E+02	1.18E+02	2.55E−05
	Min	2.55E−05	2.55E−05	2.55E−05	2.55E−05	2.55E−05	2.55E−05	2.55E−05	2.55E−05
	AVG	5.26E+01	1.62E+02	8.09E+01	1.18E+01	5.92E+01	2.37E+01	1.18E+01	2.55E−05

Table 3 continued

No.	Algorithms	SA	PSO	FA	CS	FPA	KH	MBO	FRA
	STD	98.47	104.005	71.963	35.53	59.2192	47.375	35.53	0
	Rank	5	8	7	2	6	4	2	1
*F*21	Max	− 10.5319	− 1.8535	− 2.6	− 4.079	− 4.079	− 1.8535	− 2.2182	− 10.5319
	Min	− 10.5319	− 10.5321	− 10.5	− 10.5321	− 10.5321	− 10.5321	− 10.5321	− 10.5319
	AVG	− 10.5319	− 4.7510	− 7.03	− 8.8103	− 8.70531	− 3.9201	− 3.7114	− 10.5319
	STD	0	2.9825	3.53	2.6445	2.8047	2.3859	2.3839	0
	Rank	1	6	5	3	4	7	8	1
*F*22	Max	7.05E−09	− 186.731	− 186.731	− 186.731	− 180.974	− 186.731	− 124	− 186.731
	Min	3.81E−11	− 186.731	− 186.731	− 186.731	− 186.731	− 186.731	− 186.731	− 186.731
	AVG	1.4E−09	− 186.731	− 186.731	− 186.731	− 185.754	− 186.731	− 180	− 186.731
	STD	2.11E−09	0	0	0	1.9918	0	18.9462	0
	Rank	8	1	1	1	6	1	7	1
*F*23	Max	− 78.33	− 78.3323	− 78.3323	− 78.3323	− 78.3323	− 78.3323	− 77.8264	− 78.3323
	Min	− 78.33	− 78.3323	− 78.3323	− 78.3323	− 78.3323	− 78.3323	− 78.3323	− 78.3323
	AVG	− 78.33	− 78.3323	− 78.3323	− 78.3323	− 78.3323	− 78.3323	− 78.2817	− 78.3323
	STD	0	0	0	0	0	0	0.15177	0
	Rank	1	1	1	1	1	1	8	1
*F*24	Max	2.18E−09	3.51E−163	5.36E−14	6.78E−114	0	0	8.21E−02	0
	Min	6.36E−14	8.56E−171	2.83E−15	5.52E−124	0	0	8.12E−19	0
	AVG	5.13E−10	6.82E−164	1.17E−14	7.03E−115	0	0	1.23E−02	0
	STD	6.45E−10	0	1.448E−14	2.03E−114	0	0	0.02629	0
	Rank	7	4	6	5	1	1	8	1

Table 3 continued

No.	Algorithms	SA	PSO	FA	CS	FPA	KH	MBO	FRA
*F*25	Max	−30	−2	−50	−50	−50	−36.7372	14.16	−50
	Min	−30	−2	−50	−50	−50	−49.9996	−50	−50
	AVG	−30	−2	−50	−50	−50	−46.9569	−36.812	−50
	STD	0	0	0	0	0	4.86292	21.3182	0
	Rank	7	8	1	1	1	5	6	1
*F*26	Max	2.44E−09	1.89E−164	1.22E−14	2.23E−108	0	0	6.53E−01	0
	Min	1.40E−11	4.67E−170	6.03E−16	6.11E−122	0	0	3.77E−18	0
	AVG	6.63E−10	2.18E−165	3.87E−15	2.25E−109	0	0	8.83E−02	0
	STD	7.51E−10	0	3.839E−15	6.69E−109	0	0	0.20027	0
	Rank	7	4	6	5	1	1	8	1
Average rank		5.462	3.308	3.346	2.346	2.885	3.269	6.115	1.269
Overall rank		7	5	6	2	3	4	8	1

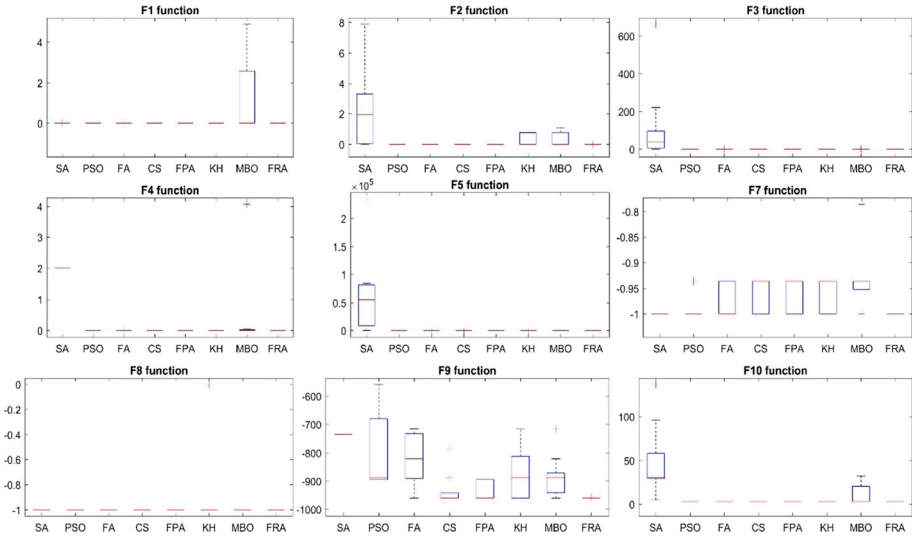

Fig. 5 Boxplots for $F1$ to $F10$ benchmark functions (except F6 function)

Figures 5, 6 and 7 show the boxplots for each benchmark function except $F6$, because all algorithms were converged to the global optimum for this benchmark function. Boxplots are very suitable for the purpose of investigating the scattering of data. In other words, it indicates the data density. Also the median, first and third quartiles and the minimum and maximum values of data can be achieved using this type of plot. As it can be seen for $F1$, $F3$, $F4$, $F7$, $F8$, $F10$, $F11$, $F13$, $F14$, $F16$, $F18$, $F20$, $F21$, $F22$, $F24$, $F25$ and $F26$ benchmark functions, the length of the achieved box by FRA is equal to zero. Therefore, for all entire runs, the algorithm has converged to the global best solution. Also it can be seen that for $F9$ and $F15$ benchmark functions the box length is the smallest for FRA and also the achieved results by this algorithm are more accurate. Boxplots are very suitable for comparing the distributions visually; however, there are statistical methods such as Mann–Whitney U test, which has been utilized in several studies such as the study conducted by De Paz et al. [9] that provide more detail analysis. In this research study, Wilcoxon rank-sum test method has been used for pairwise comparison. Hypothesis test has been utilized for making inferences about two algorithms. According to the mentioned point, H0 indicates that there is no significant difference between the results of two algorithms and H1 indicates the opposite. The level of significance has been selected equal to 0.05. Table 4 illustrates the calculated p values. Small p value is against the H0. For each benchmark function, the pairwise comparison has been done against the best algorithm. The negative sign indicates that both algorithms have the same distributions, and N/A illustrates the algorithm that other methods are compared to it. As shown in Table 4, FRA outperforms MBO and SA most of the times. For $F2$ benchmark function, the FRA has achieved the highest p value and it can be assumed that it has the same distribution as PSO. For $F5$, $F12$ *and* $F19$ functions, FRA has failed because of the small p value. In case of $F5$, FRA has the highest p value in comparison with other algorithms; therefore, it can be said that it outperforms other algorithms (except FA which is the best one for $F5$). For $F12$ function, the achieved p value for FRA is 0.0437, which is close to the selected significant level (0.05). According to the p values reported in Table 4, it can be concluded that FRA outperforms other algorithms most of the times. Figures 8, 9, 10, 11,

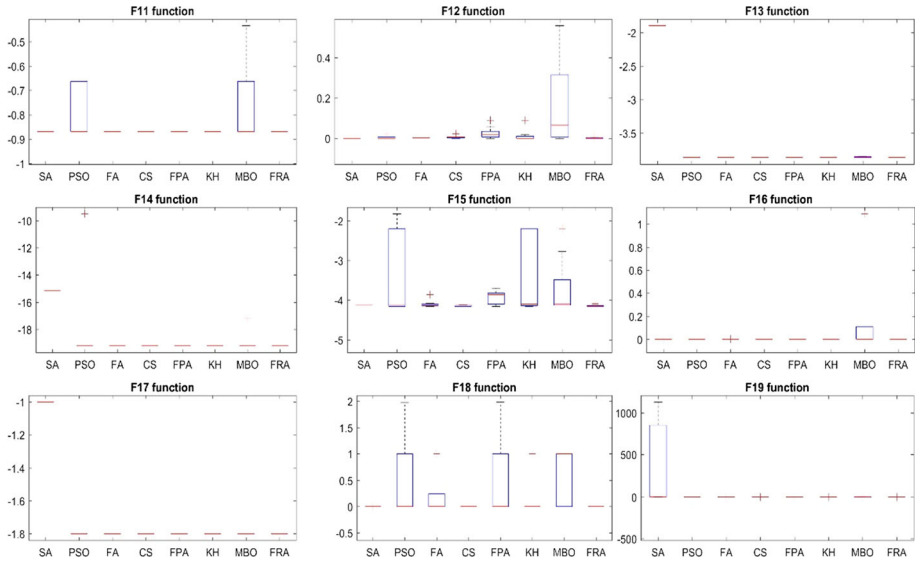

Fig. 6 Boxplots for $F11$ to $F19$ benchmark functions

12, 13 and 14 present the convergence of the algorithms. According to these figures, FRA convergence is faster and also more accurate for $F5$ and $F10$ functions in comparison with other algorithms. Also it can be seen that most of the time the proposed algorithm takes the second, third and fourth place in convergence. This is because a significant global search takes place in this algorithm by utilizing the equation of turbulence boundary layer thickness and this is why most of the times FRA can obtain the global optimum solution. Table 5 illustrates the average run time of a single iteration for different algorithms. According to the presented results, the run time of FRA is less than FPA, KH, FA and SA. According to the results presented in Table 3 and also Figs. 5, 6, 7, 8, 9, 10, 11, 12, 13 and 14, it can be concluded that FRA can compete with other popular and well-known heuristic algorithms such as krill herd algorithm and flower pollination algorithm, and therefore, it is trustable algorithm in solving optimization problems.

4.2 Real-case engineering problem 1: heat wheel

Intelligent-based algorithms are being widely utilized in the field of heat exchanger problems [1]. In this research study, heat wheel optimization problem has been considered as a case study. Heat wheels are rotary type regenerators that are being utilized in power plants and also air handling units for HVAC applications. This type of regenerators has two sections, namely process section and regenerator section. The hot and cold air pass from the process and regeneration sections, respectively. Meaning that by passing the hot air from the process section, the matrix of the wheel captures the heat of the process air and it gets heated. Because the wheel rotates with a specific speed, therefore the heated part of the matrix comes into the cold air path, and therefore, it gets cold and gets ready for another cycle. The matrix of these wheels usually contains channels with sinusoidal or honeycomb cross sections. The matrix sheet thickness and also the channel height have influence on the porosity and therefore on the performance of the wheel. Figure 15 illustrates a typical heat wheel. In the presented

Table 4 p value of Wilcoxon rank-sum test

	SA	PSO	FA	CS	FPA	KH	MBO	FRA
$F1$	4.11E−05	–	2.17E−05	–	–	–	2.17E−05	N/A
$F2$	2.17E−05	N/A	6.39E−05	6.39E−05	0.168	0.076	6.39E−05	0.173
$F3$	4.11E−05	–	2.17E−05	–	–	–	2.17E−05	N/A
$F4$	4.11E−05	–	2.17E−05	–	–	–	2.17E−05	N/A
$F5$	2.17E−05	1.82E−04	N/A	1.82E−04	1.82E−04	1.82E−04	1.82E−04	2.44E−04
$F6$	–	–	–	–	–	–	–	–
$F7$	–	–	0.2477	0.0433	0.0047	0.0047	0.0011	N/A
$F8$	–	–	–	–	–	–	–	–
$F9$	4.11E−05	2.17E−05	2.16E−04	0.2476	0.2476	0.0047	0.0011	N/A
$F10$	4.11E−05	–	–	–	–	–	0.1084	N/A
$F11$	–	0.1084	–	–	–	–	0.2477	N/A
$F12$	N/A	0.4775	2.17E−05	0.00158	0.02632	0.4749	0.0279	0.0437
$F13$	4.11E−05	–	–	–	–	–	0.0047	N/A
$F14$	4.11E−05	–	–	–	–	–	–	N/A
$F15$	0.0265	0.0687	0.0724	0.3298	0.0021	0.01678	0.00173	N/A
$F16$	4.11E−05	–	2.17E−05	–	–	–	2.17E−05	N/A
$F17$	4.11E−05	–	–	–	–	–	–	N/A
$F18$	4.11E−05	0.1084	2.17E−05	–	0.1084	–	0.0012	N/A
$F19$	2.17E−05	N/A	6.39E−05	6.39E−05	6.39E−05	0.0022	6.39E−05	6.34E−05

Table 4 continued

	SA	PSO	FA	CS	FPA	KH	MBO	FRA
*F*20	0.2058	0.0012	0.01548	–	0.0433	0.5263	–	N/A
*F*21	4.11E−05	0.0012	0.0433	0.2477	0.2477	0.00021	0.00021	N/A
*F*22	4.11E−05	–	–	–	0.5263	–	–	N/A
*F*23	–	–	–	–	–	–	–	–
*F*24	4.11E−05	2.17E−05	2.17E−05	2.17E−05	–	–	2.17E−05	N/A
*F*25	4.11E−05	2.17E−05	–	–	–	2.17E−05	0.01548	N/A
*F*26	4.11E−05	2.17E−05	2.17E−05	2.17E−05	–	–	2.17E−05	N/A

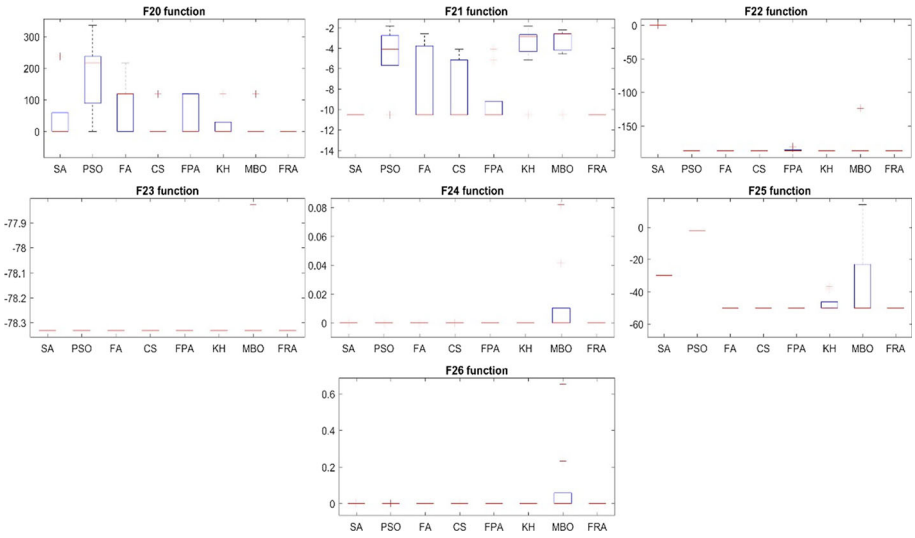

Fig. 7 Boxplots for $F20$ to $F26$ benchmark functions

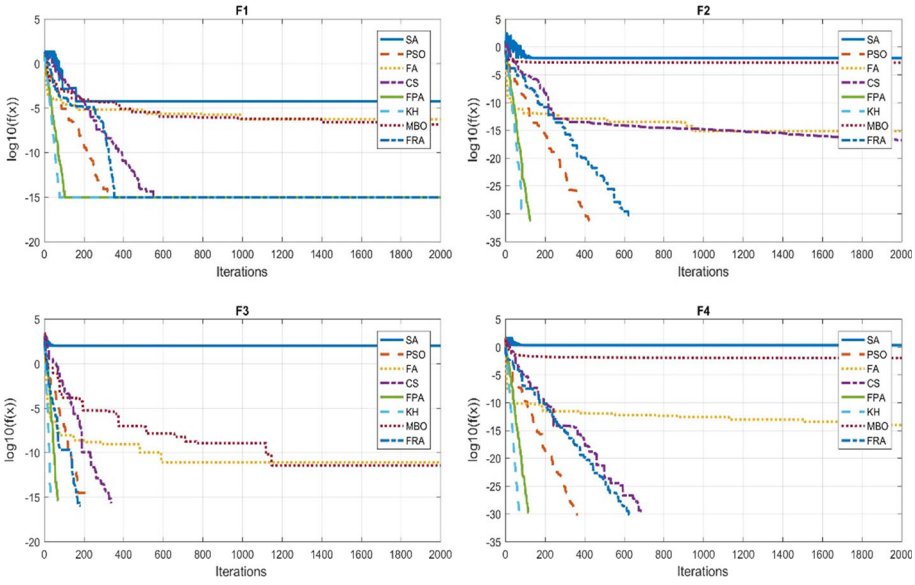

Fig. 8 The convergence rate for $F1$ to $F4$ functions

figure, the passing area of both cold and hot flows is equal and the wheel channels cross section is equilateral triangle.

For the purpose of calculating the regenerator efficiency, the equations presented in Table 6 are utilized. The heat transfer coefficients are also calculated using the equations presented in Shah and Shekulic [37].

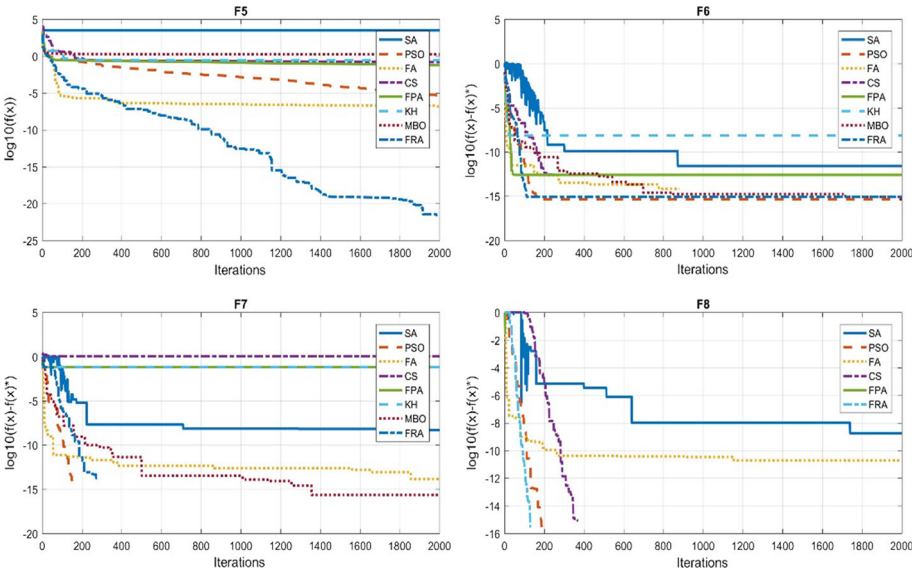

Fig. 9 The convergence rate for $F5$ to $F8$ functions

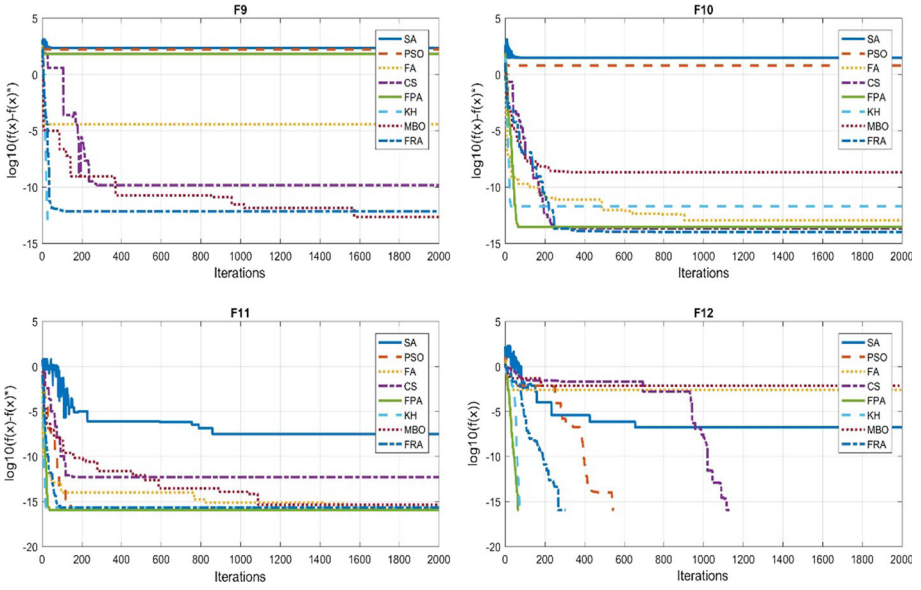

Fig. 10 The convergence rate for $F9$ to $F12$ functions

For the purpose of calculating the pressure drop of the regenerator, following equation has been utilized [37]:

$$\Delta p = \frac{G^2}{2\rho_{\mathrm{i}}}\left[1-\sigma^2+K_{\mathrm{c}}+2\left(\frac{\rho_{\mathrm{i}}}{\rho_{\mathrm{o}}}-1\right)+f\frac{L}{r_{\mathrm{h}}}\rho_{\mathrm{i}}\left(\frac{1}{\rho}\right)_m-\left(1-\sigma^2-K_{\mathrm{e}}\right)\frac{\rho_{\mathrm{i}}}{\rho_{\mathrm{o}}}\right] \quad (8)$$

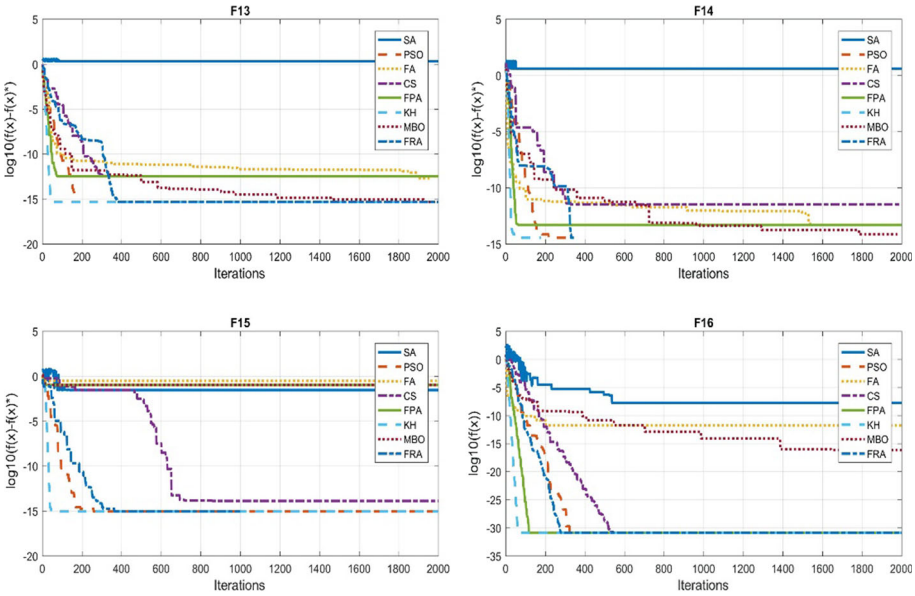

Fig. 11 The convergence rate for $F13$ to $F16$ functions

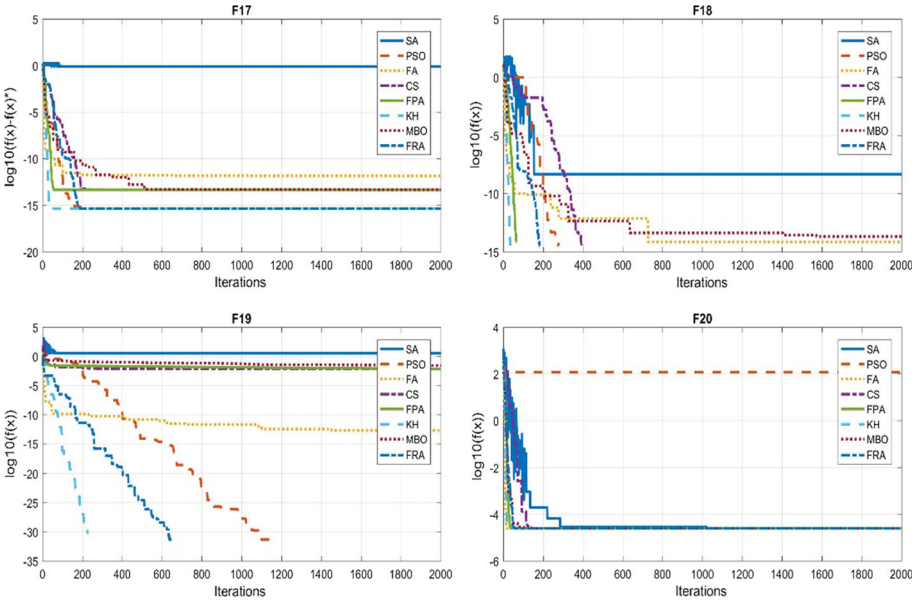

Fig. 12 The convergence rate for $F17$ to $F20$ functions

In the above equation, G presents the mass flux, ρ_i is the inlet density, σ is the porosity, K_c is the contraction coefficient, ρ_o is the outlet density, f is the Manning factor, L presents the regenerator length, r_h is the hydraulic radius, K_e is the expansion coefficient at the outlet,

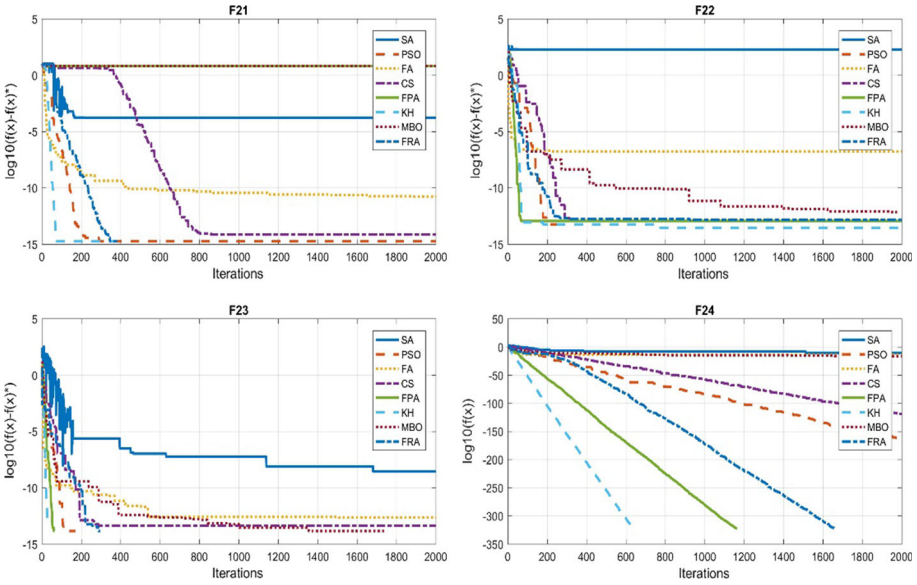

Fig. 13 The convergence rate for $F21$ to $F24$ functions

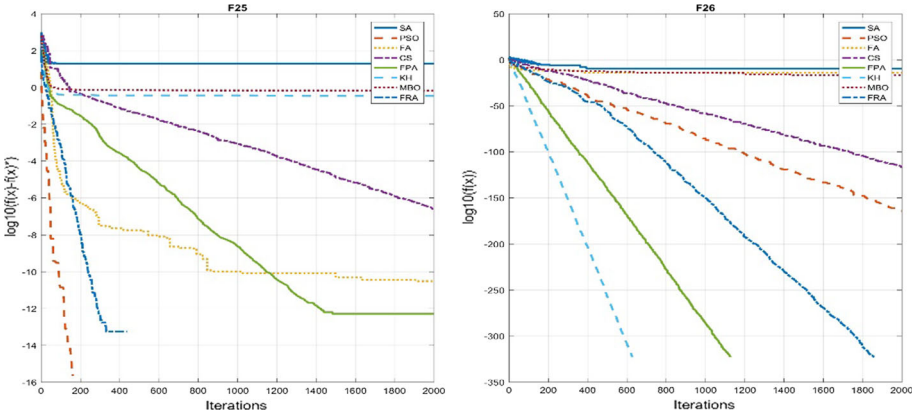

Fig. 14 The convergence rate for $F25$ and $F26$

and $\left(\frac{1}{\rho}\right)_m$ presents the inverse of the average density which can be calculated according to the following formula [37]:

$$\left(\frac{1}{\rho}\right)_m = \frac{1}{2}\left(\frac{1}{\rho_{\mathrm{i}}} + \frac{1}{\rho_{\mathrm{o}}}\right) \tag{9}$$

In this research study, the outlet temperature of the hot flow (process flow) has been considered as the objective function. Meaning that the algorithm tries to find the best values of the variables in order to minimize the outlet temperature of the process flow. The amount of pressure drop has been considered as the constraint of the problem. Therefore, a constraint optimization problem has been solved using FRA. Four decision variables are selected in order to obtain the optimum design which are the regeneration to process flow rates ratio, the

Table 5 Required average run time for single iteration of different algorithms

	SA	PSO	FA	CS	FPA	KH	MBO	FRA
Run time (s)	0.02316	0.004	0.0294	0.0049	0.0066	0.0184	0.0051	0.0057

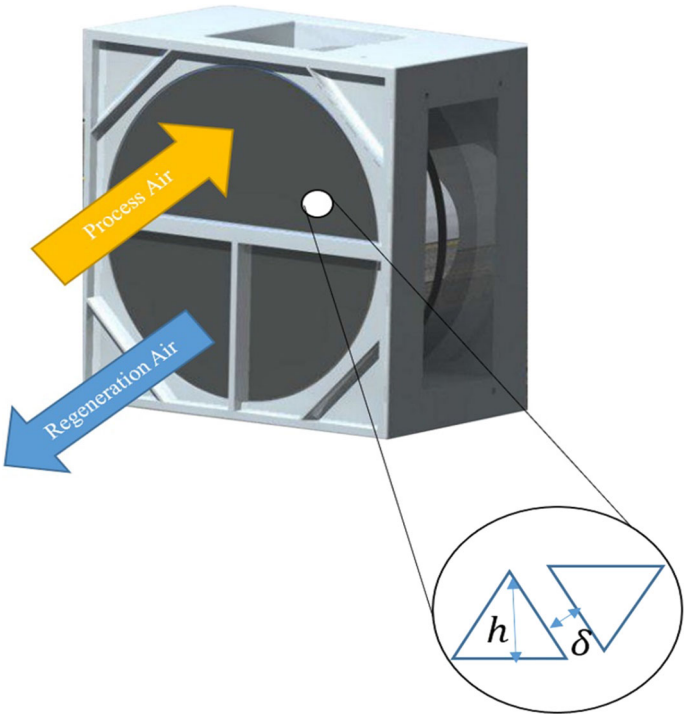

Fig. 15 Schematic of the heat wheel

matrix thickness (δ), the channel height (h) and the wheel diameter (D). Following equation presents the variables and also the objective function:

$$\begin{cases} \text{objective function: Minimum}(T_{\text{out}}) + \text{penalty}(Pressure) \\ \text{Variables:} \begin{cases} 0.1 \leq \frac{\text{reg}}{\text{process}} \leq 1 \\ 0.5 \leq \text{Diameter} \leq 1 \\ 0.25 \times 10^{-3} \leq \delta \leq 1 \times 10^{-3} \\ 2 \times 10^{-3} \leq h \leq 1 \times 10^{-2} \end{cases} \end{cases} \tag{10}$$

The simulations are carried out for two values of process air flow rates which are 1000 m^3/h and 2000 m^3/h. The number of iterations is set equal to 1000, and number of population is selected equal to 30 and 50. The process air inlet temperature is considered equal to 50 °C, and the regeneration air temperature is set equal to 20 °C. The rotational speed of the wheel is set equal to 10 RPM. Table 7 summarizes the results of optimization using three different algorithms. According to the presented results, by increasing the number of population from 30 to 50, the performance of the FPA and KH algorithms has been improved. On the other hand, the FRA has obtained the optimum result at both population numbers. This indicates

Table 6 The equations utilized for calculating the regenerator efficiency [37]

No.	Equation	Definition of the parameter
1	$\sigma = \frac{4h^2}{(2h+3\delta)^2}$	Porosity of a wheel with equilateral triangle cross sections
2	$\beta = \frac{24h}{(2h+3\delta)^2}$	The specific area of a wheel with equilateral triangle cross sections
3	$D_{\mathrm{h}} = \frac{4\sigma}{\beta}$	The channel hydraulic diameter
4	$\mathrm{NTU} = \frac{1}{C_{\min}}\left[\frac{1}{1/(hA)_h+1/(hA)_c}\right]$	Number of transfer units, where $C_{\min}$ presents the minimum heat capacity between the two flows, h presents the convection coefficient and A is the heat transfer area
5	$C^* = \frac{C_{\min}}{C_{\max}}$	The ratio of minimum to maximum heat capacity
6	$\varepsilon_{cf} = \frac{1-\exp[-NTU(1-C^*)]}{1-C^*\exp[-NTU(1-C^*)]}$	The efficiency of the exchanger without considering the effect of rotation and also the conduction in the axial direction
7	$C_r^* = \frac{M_{\mathrm{w}}c_{\mathrm{w}}(N/60)}{C_{\min}}$	Where M_{w} presents the wheel mass, c_{w} represents the specific heat of wheel, and N is the rotational speed
8	$\varphi = 1 - \frac{1}{9\times C_r^{*1.93}}$	The correction factor for efficiency
9	$A_k = A_{fr}(1-\sigma)$	The conduction heat transfer area
10	$\lambda = \frac{k_{\mathrm{w}}A_k}{LC_{\min}}$	Where k_w presents the conduction coefficient of the matrix and L is the length of regenerator
11	$\Phi = \left(\frac{\lambda\mathrm{NTU}}{1+\lambda\mathrm{NTU}}\right)^{\frac{1}{2}}\tanh\left\{\frac{\mathrm{NTU}}{\left[\frac{\lambda\mathrm{NTU}}{1+\lambda\mathrm{NTU}}\right]^{\frac{1}{2}}}\right\}$	Coefficient for modifying the axial conduction effect
12	$C_\lambda = \frac{1}{1+\frac{\mathrm{NTU}(1+\lambda\Phi)}{1+\lambda\mathrm{NTU}}} - \frac{1}{1+\mathrm{NTU}}$	Coefficient for modifying the axial conduction effect
13	$\varepsilon = \varepsilon_{cf}\varphi\left(1-\frac{C_\lambda}{2-C^*}\right)$	The efficiency of the heat wheel

the robustness of the algorithm. It can be seen that all algorithms have satisfied the required constraint; therefore, FRA can also be used for constrained optimization problems. Regarding the physical results, it can be seen that regeneration to process air flow ratio is always equal to 1 and the matrix thickness is equal to 0.25. Both values are the lower bounds of these variables. According to the equations presented in Table 6, this achievement can be explained. If C^* is considered to be equal to 1, then the value of ε_{cf} would also be equal to 1 and therefore the overall efficiency of the heat wheel would be maximum in this case. On the other hand, for the purpose of controlling the pressure drop, the porosity must be reduced, and also by

reducing the porosity, the conduction effect in the axial direction would be minimized and the efficiency would be maximized. By decreasing the required pressure drop, it can be seen that the channel height and also wheel diameter have been increased. This is because of decreasing the velocity of process air flow in the channels.

Table 8 illustrates the required process time for 150 iterations in solving heat wheel optimization problem and also the number of iterations when the convergence has been started. The number of population has been considered equal to 50 for all algorithms, and the pressure drop is considered equal to 100. As it can be seen, the quickest algorithm is the FRA. Therefore, it can be said the proposed algorithm has a great performance and can be considered as a candidate algorithm in solving complex optimization problems.

4.3 Real-case engineering problem 2: horizontal axis marine current turbine

In recent years, global warming, air pollution, fossil fuel restrictions, etc., have attracted the attention of governments and investors to renewable energy sources. Solar, wind, geothermal, hydro, ocean, etc., are different kinds of renewable energies. Horizontal axis marine current turbines are one of the novel technologies, which are being used for harnessing the kinetic energy of marine currents or tidal currents. The idea of extracting the tidal current energy is adapted from wind turbines. Meaning that similar turbines can be used for this purpose except that the fluid in this case will be water, which is almost 1000 times denser than air. Another difference is the possible occurrence of cavitation in tidal turbines. However, the design procedure and also performance prediction of wind and tidal turbines are similar. One of the most popular methods for this purpose is the blade element momentum (BEM) theory. This theory combines two approaches for calculating the power coefficient of the turbine. The first one is the momentum theory, which derives the equations of axial force and the torque using the Newton's second law and continuity equations. The second one extracts the torque and axial force using external flow analysis. In this case, lift and drag forces are converted to tangential and axial forces. Since all axial and angular momentums of the flow cannot be transferred to the turbine, therefore, an axial and angular induction factors are being defined in this theory in which results are calculated by combining the explained approaches. The detailed formulation and physical explanation of the BEM theory can be found in several research studies [41, 43]. Here, only the procedure of calculating the power coefficient of a specific geometry using BEM theory will be explained. At the first stage, the blade of the turbine must be divided into a finite number of sections (10–20 sections). After defining the number of sections, the geometrical specifications such as the chord and twist values and also type of the hydrofoil for each section must be defined. Therefore, geometrical specifications are required inputs for BEM code. After this stage, an initial value is being considered for axial and tangential induction factors for the first section. Using the defined twist angle and also considered values for induction factors, the angle of attack and inlet angle will be calculated. According to the calculated value of angle of attack and also defined hydrofoil type, the lift and drag coefficients will be extracted from the available data base or using other methods. After this, tangential and axial induction factors will be calculated using equations which are a function of chord and twist values and will be compared with their initial considered values. If the difference is smaller than the predefined value, the same procedure will be repeated for the next section; otherwise, the calculated values of induction factors will be used as new options for repeating the procedure for the same section. At the end (after calculating the induction factors for all sections), by integrating along the blade span, the power coefficient will be calculated. Therefore, according to the explained procedure, if

Table 7 The results of heat wheel optimization using three different algorithms

Algorithm	nPop	Constraint (pressure Pa)	Flow rate (m^3/h)	Output temperature (°C)	Pressure drop (Pa)	Regen/process flow	Diameter (m)	δ (mm)	h (mm)
FPA	30	100	1000	26.8182	100	1	0.5883	0.25	2.5
	30	80	1000	26.9311	80	1	0.6148	0.25	2.7
	30	60	1000	27.078	60	1	0.652	0.25	2.9
	50	100	1000	26.8181	100	1	0.5891	0.25	2.5
	50	80	1000	26.9311	80	1	0.6156	0.25	2.7
	50	60	1000	27.0779	60	1	0.6504	0.25	2.9
	50	100	2000	26.8181	100	1	0.8337	0.25	2.5
	50	80	2000	26.9311	80	1	0.87	0.25	2.7
	50	60	2000	27.0779	60	1	0.92	0.25	2.9
KH	30	100	1000	26.8186	100	1	0.5865	0.25	2.5
	30	80	1000	26.9312	80	1	0.6158	0.25	2.7
	30	60	1000	27.0779	60	1	0.6505	0.25	2.9
	50	100	1000	26.8181	100	1	0.5895	0.25	2.5
	50	80	1000	26.9311	80	1	0.615	0.25	2.7
	50	60	1000	27.0779	60	1	0.6505	0.25	2.9
	50	100	2000	26.8181	100	1	0.8335	0.25	2.5
	50	80	2000	26.9311	80	1	0.87	0.25	2.7
	50	60	2000	27.0779	60	1	0.92	0.25	2.9

Table 7 continued

Algorithm	nPop	Constraint (pressure Pa)	Flow rate (m^3/h)	Output temperature (°C)	Pressure drop (Pa)	Regen/process flow	Diameter (m)	δ (mm)	h (mm)
FRA	30	100	1000	26.8181	100	1	0.5896	0.25	2.5
	30	80	1000	26.9311	80	1	0.615	0.25	2.7
	30	60	1000	27.0779	60	1	0.6501	0.25	2.9
	50	100	1000	26.8181	100	1	0.5892	0.25	2.5
	50	80	1000	26.9311	80	1	0.615	0.25	2.7
	50	60	1000	27.0779	60	1	0.6501	0.25	2.9
	50	100	2000	26.8181	100	1	0.8335	0.25	2.5
	50	80	2000	26.9311	80	1	0.8706	0.25	2.7
	50	60	2000	27.0779	60	1	0.9208	0.25	2.9

Table 8 The comparison between the required process time of different algorithms in heat wheel optimization problem

	Output temperature (°C)	Process time for 150 iterations (s)	Iteration that convergence starts
FPA	26.8181	65.945	44
KH	26.8181	425.460	19
FRA	26.8181	62.196	32

the BEM is considered as a black box, the input to this box will be geometrical specifications and the output will be power coefficient. For the purpose of optimization of marine current turbines, the chord and twist values of each section can be considered as decision variables. By combining BEM theory with a heuristic optimization algorithm, first an initial population of chords and twists will be generated and then power coefficient of each set of chords and twists will be calculated using BEM theory. After this stage, new solutions (chords and twists) will be generated using the algorithm equations and then they will be used as inputs for BEM theory. The new solutions will be sorted according to their generated power coefficient, and the best one will be selected as the global best solution. This procedure will be repeated for a specific number of iterations or until a stopping criterion is satisfied.

Design optimization of marine current turbine can be considered as both constraint and unconstraint problem. The constraint that can be considered is the occurrence of cavitation. Here, the experimental scale turbine proposed by Bahaj et al. [3] has been considered for optimization. The selected turbine has three blades with a radius equal to 400 mm. NACA 63-812, NACA 63-815, NACA 63-818, NACA 63-821 and NACA 63-824 are used across the blade span. The geometrical specification of the blade can be found in Bahaj et al. [3]. The decision variables, which are considered for optimization, are chord and twist distributions. Because the specifications of 17 sections are provided in Bahaj et al. [3], therefore, 17 chords and 17 twists would be the decision variables and overall the problem would have 34 variables. Equation (11) illustrates the objective function and variables. The upper and lower bounds of chord and twist are selected according to the values reported in Bahaj et al. [3] for the chord and twist of hub and tip sections.

$$\begin{cases} \text{objective function: Maximum}(C_P) + \text{penalty(Cavitation)} \\ \text{Variables:} \begin{cases} 0.02 \leq \text{chord}_i \leq 0.05 \quad i = 1, 2, \ldots, 17 \\ 0 \leq \text{twist}_i \leq 20 \quad i = 1, 2, \ldots, 17 \end{cases} \end{cases} \tag{11}$$

For predicting the cavitation occurrence, the cavitation-free bucket diagram developed by Batten et al. [4] has been used. According to this diagram, if the available cavitation number and the lift coefficient are known, then the occurrence of cavitation can be predicted. The free stream velocity is considered equal to 1.73 m/s, and the tip speed ratio is selected equal to 5.8.

Figure 16 illustrates the convergence curves for FPA, KH and FRA. In this case, no constraints are considered. The number of population is set equal to 150, and the number of iterations is selected equal to 2000. As it can be seen, the obtained power coefficients by FRA, FPA and KH are 0.487, 0.485 and 0.483, respectively. The maximum power coefficient of the geometry presented in Bahaj et al. [3] is almost equal to 0.46; therefore, all algorithms have optimized the turbine performance but FRA has reached the optimum solution. In Fig. 16, the values of twist and chord of different sections, at initial iterations and final iterations, can be observed. Red airfoils are the tip and hub sections.

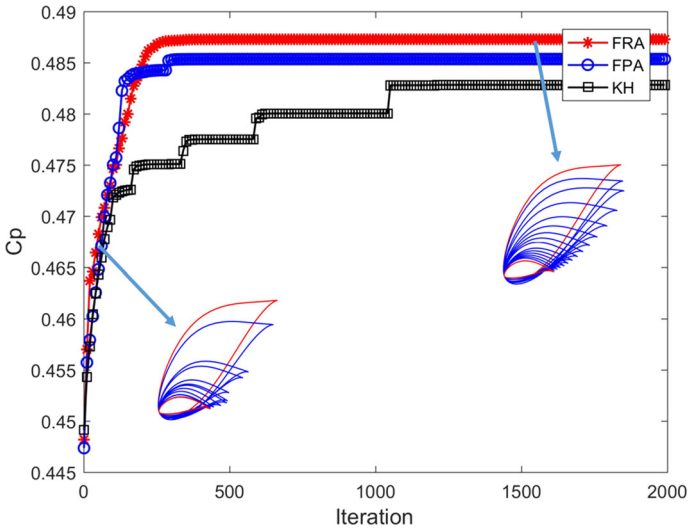

Fig. 16 The convergence of FPA, KH and FRA in horizontal axis marine current optimization problem (with no cavitation constraint)

Table 9 Obtained power coefficient by different algorithms at two cavitation numbers

	nPop	Cavitation number	C_P (power coefficient)
FPA	100	4	0.4800
	150	4	0.4858
	100	2	0.3281
	150	2	0.3328
KH	100	4	0.4736
	150	4	0.4769
	100	2	0.3239
	150	2	0.3291
FRA	100	4	0.4858
	150	4	0.4873
	100	2	0.3281
	150	2	0.3341

Table 9 indicates the obtained power coefficients by three algorithms and two population numbers. As it can be seen by increasing the number of population from 100 to 150, the performance of algorithms gets better. If the available cavitation number is 4, then according to the obtained result by FRA, the maximum power coefficient would be 0.487 and it is the same as the case where the cavitation was not considered as constraint. In other words, if cavitation number is selected equal to 4, there would be no risks for cavitation. On the other hand, if the available cavitation number is 2, then the design must be changed in order to prevent the cavitation. In this case, the maximum power coefficient would be equal to 0.3341 which is obtained by FRA. As it can be seen, in the second engineering case study, the performance of FRA and FPA are similar, while both of them outperform KH algorithm.

Therefore, it can be said that FRA has a great performance in solving complex engineering problems.

5 Conclusion

In this research study, it was aimed to develop a novel meta-heuristic optimization algorithm which was mainly inspired from the concepts of fluid mechanics. The algorithm was named Flow Regime Algorithm (FRA) because the laminar and turbulent flows were used for the purpose of local and global searches, respectively. A search type factor (STF) was defined for the purpose of creating balance between exploration and exploitation. This idea was adapted from the definition of Reynolds number. The performance of the proposed algorithm was evaluated using 26 popular benchmark functions. After comparing its performance to seven other well-known optimization algorithms, namely SA, PSO, FA, CS, FPA, KH and MBO, the results indicated that the FRA overall rank was equal to 1. In addition, Wilcoxon rank-sum test results illustrated that FRA outperformed other algorithms for most of the benchmark functions. The FRA performance was also investigated in two real-case engineering problems. The design of a heat wheel and the design of a horizontal axis marine current turbine were optimized using FRA, FPA and KH. The results also indicated that FRA can obtain the optimum design and also its required process time is less than two other algorithms. For a conclusion, it can be said that FRA has great performance and it can be definitely a candidate algorithm in solving complex optimization problems.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ahilan C, Dhas JER, Somasundaram K et al (2015) Performance assessment of heat exchanger using intelligent decision making tools. Appl Soft Comput 26:474–482
2. Arora R, Kaushik SC, Kumar R et al (2016) Soft computing based multi-objective optimization of Brayton cycle power plant with isothermal heat addition using evolutionary algorithm and decision making. Appl Soft Comput 46:267–283
3. Bahaj AS, Molland AF, Chaplin JR et al (2007) Power and thrust measurements of marine current turbines under various hydrodynamic flow conditions in a cavitation tunnel and towing tank. Renew Energy 32:407–426
4. Batten WMJ, Bahaj AS, Molland AF et al (2006) Hydrodynamics of marine current turbines. Renew Energy 31:249–256
5. Benyoucef AS, Chounder A, Kara K et al (2015) Artificial bee colony based algorithm for maximum power point tracking (MPPT) for PV systems operating under partial shaded conditions. Appl Soft Comput 32:38–48
6. Birbil SI, Fang SC (2003) An electromagnetism-like mechanism for global optimization. J Glob Optim 25:263–282
7. Braun MA, Seijo S, Echanobe J et al (2016) A neuro-genetic approach for modeling and optimizing a complex cogeneration process. Appl Soft Comput 48:30–43
8. Chuang CC, Jiang JA (2007) Integrated radiation optimization: inspired by the gravitational radiation in the curvature of space–time. In: IEEE Congress on evolutionary computation, pp 3157–3164
9. De Paz JF, Tapia DI, Alonso RS et al (2013) Mitigation of the ground reflection effect in real-time locating systems based on wireless sensor networks by using artificial neural networks. Knowl Inf Syst 34:193–217
10. El Moiz Dahi ZA, Mezioud C, Draa A (2016) On the efficiency of the binary flower pollination algorithm: application on the antenna positioning problem. Appl Soft Comput 47:395–414

11. Erol OK, Eksin I (2006) A new optimization method: big bang–big crunch. Adv Eng Softw 37:106–111
12. Eskandar H, Sadollah A, Bahreininejad A et al (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110–111:151–166
13. Formato RA (2007) Central force optimization: a new metaheuristic with applications in applied electromagnetics. Progr Electromagn Res 77:425–491
14. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17:4831–4845
15. Geem ZW, Kim JH (2001) A new heuristic optimization algorithm: harmony search. Simulation 76:60–68
16. Hashemi Y, Shayeghi H, Moradzadeh M (2017) Design of dual-dimensional controller based on multi-objective gravitational search optimization algorithm for amelioration of impact of oscillation in power generated by large-scale wind farms. Appl Soft Comput 53:236–261
17. Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. Inf Sci 222:175–184
18. Hieu TT (2011) A water flow algorithm for optimization problems. Ph.D. thesis, National University of Singapore
19. Hsiao YT, Chuang CL, Jiang JA et al (2005) A novel optimization algorithm: space gravitational optimization. In: IEEE international conference on systems, man and cybernetics, pp 2323–2328
20. Javidy B, Hatamlou A, Mirjalili S (2015) Ions motion algorithm for solving optimization problems. Appl Soft Comput 32:72–79
21. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213:267–289
22. Kaveh A, Ghazaan MI, Bakhshpoori T (2013) An improved ray optimization algorithm for design of truss structures. Period Polytech Civ Eng 57:97–112
23. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the 1995 IEEE international conference on neural networks, vol 4, pp 1942–1948
24. Kirkpatrick S, Gelatt Jr CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680
25. Kripka M, Kripka RML (2008) "Big crunch" optimization method. In: International conference on engineering optimization
26. Li X, Zhang J, Yin M (2014) Animal migration optimization: an optimization algorithm inspired by animal migration behavior. Neural Comput Appl 24:1867–1877
27. Mirjalili S (2015) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27:1053–1073
28. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
29. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
30. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
31. Mortazavi A, Togan V (2017) Sizing and layout design of truss structures under dynamic and static constraints with an integrated particle swarm optimization algorithm. Appl Soft Comput 51:239–252
32. Rabanal P, Rodriguez I and Rubio F (2008) Solving dynamic TSP by using river formation dynamics. In: Fourth international conference on natural computation. IEEE computer society, pp 246–250
33. Rao SS (2009) Engineering optimization theory and practice. Wiley, New Jersey, pp 1–6
34. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitation search algorithm. Inf Sci 179:2232–2248
35. Sacco WF, de Oliveira CRE (2005) A new stochastic optimization algorithm based on a particle collision meta heuristic. In: 6th World Congresses of structural and multidisciplinary optimization
36. Shah Hosseini H (2011) Otsu's criterion-based multilevel thresholding by a nature-inspired metaheuristic called galaxy-based search algorithm. In: Third World Congress on nature and biologically inspired computing (NaBIC), pp 383–388
37. Shah RK, Shekulic DP (2003) Fundamentals of heat exchanger design. Wiley, NewYork, pp 308–326
38. Shah-Hosseini H (2009) The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. Int J Bio-Inspired Comput 1:71–79
39. Shames IH (2003) Mechanics of fluids. McGraw Hill, New York, pp 637–649
40. Tahani M, Babayan N, Mehrnia S et al (2016) A novel heuristic method for optimization of straight blade vertical axis wind turbine. Energy Convers Manag 127:461–476
41. Tahani M, Babayan N (2017) Optimum section selection procedure for horizontal axis tidal stream turbines. Neural Comput Appl. https://doi.org/10.1007/s00521-017-3079-4
42. Tahani M, Babayan N, Pouyaei A (2015) Optimization of PV/Wind/Battery stand-alone system, using hybrid FPA/SA algorithm and CFD simulation, case study: Tehran. Energy Convers Manag 106:644–659

43. Tahani M, Babayan N, Astaraei FR et al (2015) Multi objective optimization of horizontal axis tidal current turbines, using Meta heuristics algorithms. Energy Convers Manag 103:487–498
44. Tayarani MH, Akbarzadeh MR (2008) Magnetic optimization algorithms a new synthesis. In: IEEE Congress on evolutionary computation, pp 2659–2664
45. Uymaz SA, Tezel G, Yel E (2015) Artificial algae algorithm (AAA) for nonlinear global optimization. Appl Soft Comput 31:153–171
46. Wang GG, Deb S, Cui Z (2015) Monarch butterfly optimization. Neural Comput Appl. https://doi.org/10.1007/s00521-015-1923-y
47. Xie L, Zeng J and Cui Z (2009) General framework of artificial physics optimization algorithm. In: World Congress on nature and biologically inspired computing, pp 1321–1326
48. Yang XS (2010) Firefly algorithm, stochastic test functions and design optimisation. Int J Bio-Inspired Comput 2:78–84
49. Yang X.S (2012) Flower pollination algorithm for global optimization. In: Unconventional computation and natural computation, pp 240–249
50. Yang XS, Deb S (2009) Cuckoo search via Levy flights. In: Proceedings of World Congress on nature and biologically inspired computing (NaBIC), pp 210–214
51. Yazdani M, Jolai F (2016) Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J Comput Des Eng 3:24–36
52. Yilmaz S, Kucuksille EU (2015) A new modification approach on bat algorithm for solving optimization problems. Appl Soft Comput 28:259–275
53. Yu JJQ, Li VOK (2015) A social spider algorithm for global optimization. Appl Soft Comput 30:614–627
54. Zarand G, Pazmandi F, Pal KF et al (2002) Using Hysteresis for Optimization. Phys Rev Lett 89:150201-1–150201-4

Mojtaba Tahani is currently an associate professor at University of Tehran. He received his Ph.D. degree in Mechanical Engineering from Iran University of Science and Technology in 2011. His research interest includes developing of optimization algorithms, simulation methods and turbulence modeling in engineering applications such as renewable energy systems, power plants and hydro-aerodynamic phenomena.

Narek Babayan received his M.Sc. degree in Renewable Energies Engineering from University of Tehran in 2015 and his B.Sc. degree in Mechanical Engineering from K.N. Toosi University of Technology in 2013. His research interests are nature inspired optimization algorithms, optimization of energy systems including different types of heat exchangers, energy convertors and, etc., energy modeling and simulation, renewable energies, HVAC systems, fluid mechanics and heat transfer.