Soft Computing
https://doi.org/10.1007/s00500-021-05734-0

METHODOLOGIES AND APPLICATION

Heat transfer relation-based optimization algorithm (HTOA)

Foad Asef[1] · Vahid Majidnezhad[1] · Mohammad-Reza Feizi-Derakhshi[2] · Saeed Parsa[3]

Accepted: 5 March 2021

Abstract
Novel metaheuristic algorithms are now considered an appealing collection of methods for solving complex optimization problems, in which the challenging objective is to find a better solution in a shorter computation time. Focusing on the same objective, this paper proposes a novel metaheuristic optimization algorithm inspired by heat transfer relationships based on the second law of thermodynamics. Imitating the heat transfer behavior of solid objects, the proposed method is called the heat transfer relation-based optimization algorithm (HTOA). This behavior was modeled on a heat transfer function used to measure temperature differences between the selected solutions and the best solution. This function was employed to determine and add the heat capacity transferred between those solutions. Finally, all the solutions were heat-exchanged with the best solution to select the fittest solution and exclude the rest. This procedure continued until the best solution or solutions were found. The proposed method is challenged by many famous benchmark problems in two categories as well as two real-world problems (PID controller and linear regression). The HTOA was then compared with a number of well-known and state-of-the-art optimization algorithms. Selecting better solutions and requiring shorter computation time, the proposed HTOA outperformed the other algorithms.

Keywords Convection and conduction · Heat transfer · HTOA · Metaheuristic algorithms · Optimization · PID controller · Regression

1 Introduction

Metaheuristic algorithms are now used widely to solve complex optimization problems. Many of these algorithms are based on biological relationships between animals and plants. Metaheuristic optimization algorithms are growing in popularity in engineering applications for various reasons. First, they rely on rather simple concepts and are easy to implement. Second, they require no gradient information. Third, they can bypass local optima. Finally, they can be utilized in a wide variety of problems in different disciplines (Mirjalili and Lewis 2016). Some of the existing optimization algorithms are classified as evolutionary computing and metaheuristic methods. Swarm intelligence (SI) is a motivational area of the population-based metaheuristic algorithms. The concepts of SI were first introduced in 1993 (Beni and Wang 1993). According to Bonabeau et al. (1999), SI is ''the emergent collective intelligence of groups of simple agents''. The SI techniques were inspired mostly by natural colonies, birds, animals, and schools of fishes. Some of the most popular SI techniques are ant colony optimization (ACO) (Dorigo and Birattari 2010), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), and artificial bee colony (ABC) (Basturk 2006). Moreover, the genetic algorithm (GA) (Bonabeau et al. 1999) is the most famous evolutionary algorithm (EA).

Retaining information about the search space in the repetition process, keeping the best solution ever found, and considering a small number of parameters and operators are some advantages of swarm intelligence algorithms over evolutionary techniques. Metaheuristic algorithms are generally classified as single-solution and population-based or multi-solution categories (Mirjalili and Lewis 2016). In

✉ Vahid Majidnezhad
vahidmn@iaushab.ac.ir

[1] Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar, Iran

[2] Department of Computer Engineering, University of Tabriz, Tabriz, Iran

[3] Department of Computer Engineering, Iran Science and Technology University, Tehran, Iran

Published online: 03 April 2021

single-solution algorithms, the search process begins with a candidate solution, whereas it starts with a set of solutions (the initial population) in the population-based model (Dhiman and Kumar 2018). Despite their differences, metaheuristic and evolutionary algorithms share a common feature, which is to divide the search process into exploration and exploitation phases. The exploration phase includes the process of researching and exploring useful areas of the large-scale search space as much as possible, whereas the exploitation phase refers to the local search capability around the appropriate and useful points obtained through the exploration phase. Due to the random nature of optimization algorithms, a major challenge would be to strike a balance between the two phases. Heat exchange occurs in three mechanisms: conduction, convection, and radiation. In solid objects, heat is transferred through conduction, a mechanism which is based on the conductivity of objects. Thanks to temperature differences in molecular structures, the conductivity of solids is higher than that of liquids. Similarly, the conductivity of liquids is higher than that of gases. The heat transfer of energy flow is attributed to temperature differences. In fact, heat transfer occurs when there is a temperature difference in an environment or between objects. This difference can cause the temperature to drop. Heat transfer is also called the net energy transfer caused by the random movement of molecules through energy diffusion. Heat transfer always starts from an object of a higher temperature and continues to an object of a lower temperature, a process which increases the temperature of the secondary object.

This study proposes a novel metaheuristic method based on the heat transfer relation between solid objects through the conduction method. The main contribution of the paper is the use of heat transfer equations between two materials that provide a new relationship for combining random solutions to improve the final solution of optimization problems. This type of heat transfer is common in natural entities such as some animals. It is an intelligent strategy to deal with the extreme coldness in the environment. For instance, it can be seen in the process of dealing with the extreme coldness among the emperor penguins of Antarctica.

This paper consists of the following sections. Section 2 reviews the literature on optimization algorithms, whereas Sect. 3 provides an overview of the proposed HTOA. Sections 4, 5, and 6 address the results and discussions regarding the mathematical functions and real-world functional problems in comparison with the previous works. Finally, Sect. 7 draws the research conclusion and presents recommendations to future studies.

2 Background

As it is shown in Fig. 1, the population-based metaheuristic algorithms are classified as four categories called physics-based, evolutionary-based, bio-inspired, and swarm-based methods (Dhiman and Kumar 2018).

2.1 Evolutionary-based algorithms

Evolutionary algorithms (EAs) are inspired by the theory of natural selection and biological evolution. These algorithms work well to find semi-optimal global solutions because they are not based on the basic fitness prospect. The most popular evolutionary algorithm is the genetic algorithm (Holland 1992). In this algorithm, optimization begins with the evolution of random solutions created from the initial population. Then each solution fits into each generation. Crossover and mutation operators are applied to each solution to produce a new population. The best solutions, based on fitness, can generate new populations in different iterations. Compared with the other random methods, the GA is not trapped in local optima due to the use of operators, especially the mutation operator. Due to the combination of the best solutions, new populations are likely to be better fitted and guarantee the improvements of future populations. Instances of EAs are differential evolution (DE) (Storn and Price 1997), evolutionary programming (EP), and evolution strategy (ES) (Hansen et al. 2003; Rechenberg 1994), genetic programming (GP) (Koza and Rice 1992), and biogeography-based optimizer (BBO) (Simon 2008). For example, the DE algorithm tries to optimize a problem by preserving a candidate solution and creating new candidate solutions through the existing combinations. The best-fitting solution remains optimal. This template applies to nonlinear problems. Few parameters are available to guide minimization problems. In the differential evolution method, a major challenge is to adjust the parameters because similar parameters may not guarantee an optimal solution.

The BBO is also a population-based evolutionary algorithm inspired by the phenomenon of animal migration between islands. The biogeography-based optimization algorithm was first introduced by Don Simon (2008) in the IEEE Transactions. This algorithm presents an interesting pattern obtained from the biogeography debates. Geographical areas that are good shelters for biological species have a high habitat suitability index (HSI). The features associated with the HSI include such factors as rainfall, vegetation diversity, variety of topographic features, land area, and temperature. The variables that determine residence are called the suitability index variables (SIVs). For different animal species, a habitat is superior variable when

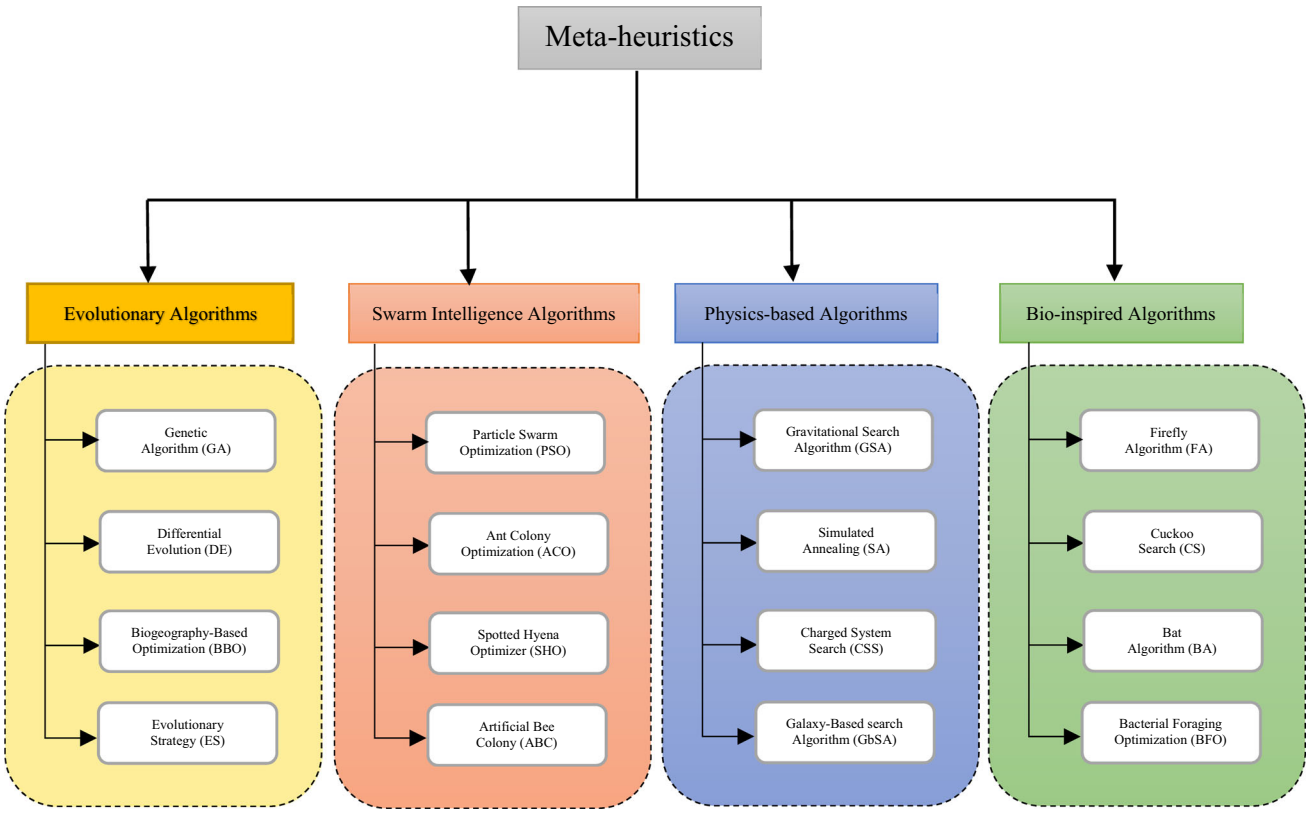

Fig. 1 The classification of the metaheuristic algorithms (Dhiman and Kumar 2018)

it has a smaller population. The BBO algorithm is based on the strategy of species migration and preferences for selecting habitats with a lower population density. The biogeography-based optimization has some features in common with other biological-based algorithms (*e.g.* GA). It also has a way of sharing information between solutions. The GA responses are terminated at the end of each step, whereas BBOs always keep the responses alive. However, their properties change as the optimization process progresses.

The BBO is a population-based optimization algorithm which does not occur in reproduction or generation. This clearly distinguishes the BBO from reproductive strategies such as GA and evolutionary algorithms. The BBO maintains the set of solutions from one iteration to the next, relying on migration for probabilistic adaptation. It has the most in common with DE strategies, in which the solutions are kept from one iteration to the next. In addition, the solutions can be learned from the neighbors and adapted to the algorithm development. Differential evolution changes the solutions directly; however, the changes in a particular DE solution are based on differences in the other DE solutions. Furthermore, DE has no biological stimuli. The BBO differs from the DE because its responses change directly through migration from other (islands) responses. The BBO responses share their attributes (SIVs) directly with other responses (Simon 2008).

2.2 Swarm-based methods

The second category of metaheuristic algorithms includes swarm intelligence (SI) methods. These algorithms are derived from the social behavior of different organisms and are usually inspired by natural patterns such as colonies, clusters of birds, and herds. In this category, PSO is the most popular algorithm proposed by Kennedy and Eberhart (1995). In PSO, particles move around the search space through a combination of the best solutions (Slowik and Kwasnicka 2017). The whole process continues until the termination condition is met. The main advantage of PSO is that it does not overlap computation with the jump operator. The particles move toward their best solutions and the best global solutions ever achieved by the group. Particle swarm optimization presents each solution as a point in space and changes it over time into a velocity vector. However, a PSO solution does not change direction; in fact, it is the change rate that indirectly affects the position (s). The main problem with PSO is its entrapment. The ant colony optimization (ACO) is another popular swarm intelligence algorithm proposed by Dorigo and Birattari (2006). It was mainly inspired by the social

behavior of ants in the ant colony. Their social intelligence helps them find the shortest path between food sources and their nest. ACO can efficiently solve the traveling salesman problem and other similar problems, something which is considered an advantage over other approaches. The theoretical analysis of a problem is very difficult through ACO because the computation cost is high during convergence. Other well-known swarm-based metaheuristic methods are the artificial bee colony algorithm (Karaboga and Basturk 2007), spotted hyena optimizer (SHO) (Dhiman and Kumar 2017), artificial fish-swarm algorithm (AFSA) (Li 2003), fruit fly optimization algorithm (FOA) (He et al. 2006), monkey search (Mucherino and Seref 2007), grey wolf optimizer (GWO) (Mirjalili et al. 2014a), and whale optimization algorithm (Mirjalili and Lewis 2016). All of these algorithms benefit from the same strategy called the group strategy for certain animal species. George and Raimond analyzed several optimization algorithms such as ABC, PSO, ACO, BSO, and GSA. According to their results, the hybrid PSO-ABC algorithm outperformed the other algorithms in the field of numerical benchmark functions and was less likely to be trapped in the local optima. Arora and Singh (2019) proposed a novel nature-inspired algorithm called the butterfly optimization algorithm (BOA). It mimics a butterfly's searching and mating behavior to solve global optimization problems. This framework is mainly based on a butterfly's feeding strategy that uses the sense of smell to determine nectar's position or to find mates. This algorithm performs properly in exploration but does not perform very well in global optimization and operation. Heidari et al. (2019) proposed a novel population-based, nature-inspired optimization paradigm, called Harris hawks optimizer (HHO). The key inspiration of HHO comes from the group behavior and chasing style of Harris's hawks in nature, called the surprise pounce, in which several hawks cooperate to encircle the prey in different directions. Harris's hawks can design a variety of chase patterns based on the dynamic nature of avoidance scenarios. In this framework, the mathematical logic mimics the dynamic behavior to develop an optimization algorithm which performs acceptably in both exploration and exploitation phases.

2.3 Physics-based algorithms

The third branch of meta-algorithms includes physics-based methods. These optimization algorithms usually follow the laws of physics and use such concepts as electromagnetic force and gravitational force to distribute information among search agents. The well-known physics-based metaheuristic algorithms are the simulated annealing (SA) (Kirkpatrick et al. 1983), gravitational search algorithm (GSA) (Rashedi et al. 2009), big bang–big crunch (BB-BC) (Erol and Eksin 2006), artificial chemical reaction optimization algorithm (ACROA) (Alatas 2011), black hole (BH) (Hatamlou 2013), and galaxy-based search algorithm (GBSA) (Shah-Hosseini 2011). These algorithms differ from other approaches in general mechanism due to the strategy adopted by search agents based on physics rules (Dhiman and Kumar 2018). For instance, the GSA is theoretically inspired by Newton's law of universal gravitation, which searches for a series of factors that are mass-proportional to the function value. During the repetition, the masses are absorbed by the gravitational force between them. The larger the mass, the stronger the force of gravity. Other masses are attracted to the heaviest mass near an optimum of the universe.

The CROG is an algorithm that has been proposed based on the CRO framework and a greedy strategy to solve KP01 efficiently. Four problem-specific elementary reactions are carefully designed to implement the local search and global search. A new repair function which helps the algorithm to yield fast convergence and avoids local optima is proposed. The simulation results demonstrate that the proposed algorithm has superior performance when compared with ACO, GA and QEA for all proposed test instances. The new approach obtains better solutions in shorter time (Truong et al. 2013).

In this paper (Xu et al. 2014), authors developed a DMSCRO for DAG scheduling on heterogeneous computing systems. As a result, the DMSCRO algorithm can cover a much larger search space than heuristic scheduling approaches. The experiments show that DMSCRO outperforms HEFT_B and HEFT_T and can achieve a higher speedup of task executions.

A hybrid chemical reaction optimization approach is proposed for DAG scheduling on heterogeneous computing systems in (Xu et al. 2013). The algorithm incorporates the CRO technique to search the execution order of tasks while using a heuristic method to map tasks to computing processors. By doing so, the proposed scheduling algorithm can achieve good performance without incurring high scheduling overhead. In the experiments, the HCRO is compared with two heuristic algorithms (HEFT-B and CPOP) and a pure metaheuristic method (DMSCRO). The results show that the proposed HCRO algorithm outperforms HEFT-B and CPOP, and it can achieve a better average performance than DMSCRO with lower overhead.

2.4 Bio-inspired algorithms

Another branch or the fourth category is related to biological algorithms inspired by biological patterns to solve problems. The well-known instances of bio-inspired algorithms are the bat-inspired algorithm (BA) (Yang 2010b), obtained from the echolocation behavior of bats, and the

cuckoo search (CS) (Yang and Deb 2009), inspired by the obligate brood parasitism of cuckoo species. Dhiman and Kumar (2018) proposed a novel swarm-based metaheuristic algorithm called the emperor penguin optimizer (EPO), the fundamental concept of which is the huddling behavior of emperor penguins. The behavior, facts, and relationships existing in nature have obviously been a great source inspiration to the development and implementation of novel algorithms for solving optimization problems, all of which represent a collective or systematic behavior toward a goal.

In this study, solutions were combined through the mathematical modeling of heat transfer relationships in order to propose a novel metaheuristic model based on heat transfer relationships to transfer heat or fit between solutions. This process provided certain solutions for mathematical modeling of behavior among objects at different temperatures. Finally, a new optimal search model was found through heat transfer relationships conductively, and its capabilities were challenged in solving real and standard problems.

2.5 Overview

A major disadvantage of these algorithms is their low speed and low accuracy in convergence on the optimal solution. The need for complex hardware for large problems, such as complex networks, and the long wait to find a solution can also be considered their other major disadvantages. Scalability, high-speed implementation, and early convergence on the optimal solution can be considered important research motivations in this regard. In this study, the authors aimed to develop an accurate and fast-paced algorithm on large-scale environments, especially complex networks, for specific purposes. Similar studies have been conducted on environmental cold control systems of some animals (*e.g.* PEOA); however, this paper aims to integrate the cold control process with the main heat transfer equation. Citing the law of conservation of energy, the paper improves the proposed solutions as presented clearly in the results.

3 Heat transfer relation-based optimization algorithm (HTOA)

The inspired strategy is first discussed in this section, and the related mathematical relationships are then presented. In fact, heat transfer analysis is an extension of thermodynamics that studies the heat transfer rate measurement methods. The section also addresses the heat exchange thermodynamics and its vital role in the first and second laws; however, the heat transfer mechanisms and heat transfer rate measurement methods are not discussed comprehensively. Thermodynamics deals with the equilibrium states of matter and does not prevail in the equilibrium states of temperature. Heat transfer involves performing the tasks that thermodynamics is inherently incapable of. In fact, heat transfer is the energy flow caused by temperature differences. When there is a temperature difference in an environment or between objects, heat transfer must occur. In other words, heat is transferred from one material to another when they have a temperature difference. There will be no heat transfer between an object and an environment, if the object is at the same temperature as its surroundings. Heat is always transferred from a higher temperature to a lower temperature and never in the opposite direction. Thus, heat can be likened to the water flowing from a tank at a higher altitude to a tank at a lower altitude. Heat transfer modes are recognized as different types of heat transfer processes.

3.1 Molecules

The proposed model strategy is partially inspired by the life of organisms living in extremely low temperatures and trying to keep themselves and their chicks warm. Therefore, each member of M_i moves to M_{best} to eschew the cold. By transferring or receiving the heat from M_{best}, selected randomly by one of the high-temperature objects, the heat is increased and transferred back to the previous location. In the propose approach, the low-temperature objects take heat from the high-temperature objects to maintain their shapes and avoid the cold air. This cycle continues until a thermal equilibrium is established. Unlike the heat transfer mechanism in walls, only two members exchange temperatures at any time. Each solution is considered an object that is combined with one of the high-fitness solutions to increase the temperature.

3.2 Selection

This selection is based on the fitness levels of all original members (*i.e.* high-temperature objects), called R, a subset is formed out of (25 to 30) % of the total population as the high fitness members. The remaining members (70% to 75% of the total population including colder objects or members with lower fitness) are combined randomly with this subset. R is updated after each iteration of the algorithm, and the variable must be obtained first from the following equation:

$$\Delta\text{Cost} = \frac{i-1}{10 * nPop}.$$

In this equation, i represents the response rank among the population; thus, the higher this value, the worse the

candidate solution rank and the larger the value of ΔCost. Moreover, *nPop* indicates the initial population. The goal is to leave a greater impact on low-fitness responses.

3.3 Heat (Fitness) transfer methods

The term "guiding" is used to describe the type of heat transfer in the environment if the temperature gradient is present in a static environment which can be solid or fluid. At the same time, "displacement" occurs when there is heat transfer between a surface and a moving fluid. The "heat radiation" is the third heat transfer method. In the form of electromagnetic waves, energy is emitted by all surfaces having specific temperatures. Therefore, there will be different temperatures in the absence of a material mediating pure heat transfer due to radiation between different surfaces. Figure 2 (Lienhard 2019) shows different types of heat transfer processes. This study designs a model of heat transfer between objects based on conduction. The mathematical models and general relations of heat transfer are discussed in the following sections, and the proposed HTOA is then described.

As noted above, conduction is carried out through heat transfer in solid or static fluid environments due to temperature differences. It is generally assumed that heat is not transferred from one point to another due to macroscopic displacement in the environment. However, heat is transferred by such factors as the random movement of gas molecules or the vibration of a solid crystal lattice. Heat transfer modes are classified as one-dimensional, two-dimensional, multi-dimensional, and permanent (Lienhard 2019).

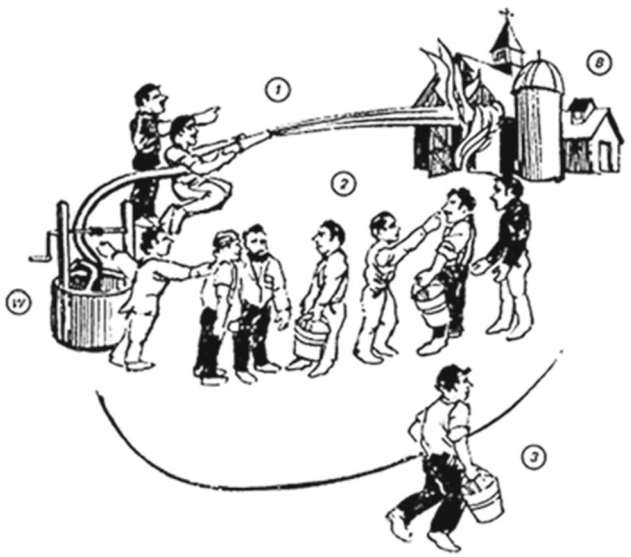

Fig. 2 The barn is on fire (Lienhard 2019). Let the water be akin to the heat, and let the people be akin to the heat transfer medium. Then: *Case 1*: The hose leads water from W to B independently of the medium. This is akin to thermal radiation in a vacuum or in most gases. *Case 2* In the bucket brigade, water goes from W to B through the medium. This is akin to conduction. *Case 3* A single runner, describing the medium, transfers water from W to B. This is akin to convection

3.4 Mathematical model

According to the following law (Lienhard 2019), the heat flux is guided in an environment directly in proportion to the temperature gradient and the heat exchange surface (the temperature gradient is represented by $\frac{\partial T}{\partial x}$ on both sides of the heat exchange surface):

$$\begin{aligned} & q \propto \frac{\Delta T}{\Delta x} A \\ & \Rightarrow q = -kA \frac{\partial T}{\partial x} k \left[\frac{W}{mK}\right] \end{aligned} \tag{1}$$

The one-dimensional permanent constant differential equation is expressed as below:

$$q_x = q_{x+dx} \Rightarrow q_x = q_x + \frac{\partial q_x}{\partial x} dx \tag{2}$$

The cross section is assumed to be the constant A = dx.dy, then:

$$(1) \text{ and } (2) \Rightarrow \frac{\partial}{\partial x}\left(-kA \frac{\partial T}{\partial X}\right) = 0 \Rightarrow k \frac{\partial^2 T}{\partial x^2} = 0.$$

If k is assumed to be constant by integrating the above relation, then:

$$\frac{\partial T}{\partial x} = C_1.$$

$$T = C_1 x + C_2.$$

In Eqs. 1 and 2, q shows the heat flux (W/m^2), whereas T indicates temperature, and $\frac{\partial T}{\partial x}$ shows the temperature gradient (K/m). Moreover, k refers to the conductivity heat transfer coefficient (W/mk). The value of k depends on the environment in which the conductive heat transfer is carried out, and A is the cross-sectional area of the material from which the transfer is conducted.

Accordingly, the one-dimensional temperature distribution in the object will be linear if the thermal conductivity is constant. In metallic environments, k is larger than that of the nonmetallic environments, a fact which is consistent with the hypothesis of heat transfer by free electrons. The gases such as air also have a lower conductivity rate than solids. The negative sign in the above equation means that the temperature decreases to increase the x coordinates. This means that the temperature gradient is lower than $\partial T/\partial x$; however, q is positive. Figure 3 shows the heat transfer from the wall.

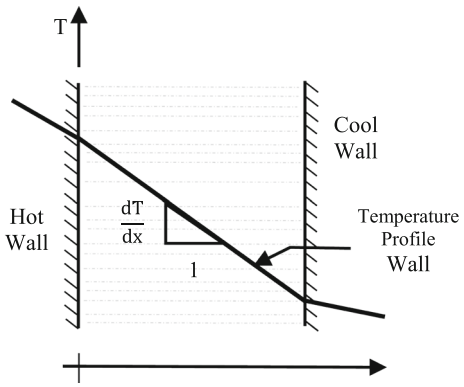

Fig. 3 The heat transfer from the wall

3.5 The *k* variable

The k variable is the heat transfer coefficient of the conduction method. For different materials, the heat transfer rate is obtained from the same surface and temperature conditions which result in different values of q. Thus, a factor is obtained and called the thermal conductivity coefficient. Different materials take different amounts of the coefficient. The k variable depends completely on the properties of materials. Table 1 shows the thermal conductivity of several important materials at 25°C. Fourier's law could be used to conclude thermal conductivity units and dimensions.

$$\text{Thermal Conductivity} = \frac{\text{Heat Flux}}{\text{Temperature Gradient}}$$

The k value in the SI unit is measured through $(W/m^2)/(K/m) = W/(m * K)$, whereas it is obtained from $BTU/(hr \times ft \times F)$ in the British unit. The approximate values of the thermal conductivity of some ordinary materials at 25 °C are given below. Since k is the heat transfer coefficient associated with a particular material, it plays a critical role in the value of q or the transferred amount of heat and can strongly affect the final temperature. The following sections discuss how to select a suitable value for k (Table 1).

Table 1 The thermal conductivity of several important materials at 25 °C (Thermal Conductivity of selected Materials and Gases 2003)

Material/substance	Temperature, 25°C
Glycerol	0.28
Glass, wool insulation	0.04
Glass, window	0.96
Glass, pearls, saturated	0.76
Water	0.606

3.6 Equations governing heat transfer

The equation governing heat transfer by conduction is written in the Cartesian coordinates as follows:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{\dot{q}}{k} = \rho c \frac{\partial T}{\partial t}.$$

In the polar coordinates, the above equation is written as below:

$$\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r} + \frac{1}{r}\frac{\partial T}{\partial \theta^2} + \frac{\dot{q}}{k} = \frac{\rho c}{k}\frac{\partial T}{\partial r}.$$

3.7 Shape factor

The heat transfer between two objects is obtained from the following equation:

$$q = sk\Delta T.$$

In this equation, q refers to the total heat flow rate, whereas S indicates the heat exchange area, and m^2 is the shape factor. Moreover, T shows the temperature, and k is the thermal conductivity. One-dimensional temperature transfer is showed in Fig. 4.

In the above equation, k refers to the coefficient of conduction of the environment between the two bodies through which heat passes, whereas ΔT is the temperature difference between the two bodies. Moreover, S is called the form factor that depends strongly on the geometry, dimensions, and arrangements of the two objects in close proximity. This study benefits from the relationships of S with many arrangements, the wall geometry function, and a large plane wall shape. As it is shown in Figs. 5 and 6, the heat flow through the blade is analyzed below:

(1). Has a direct relation with A, according to which the greater the available area, the greater the amount of heat per unit time.

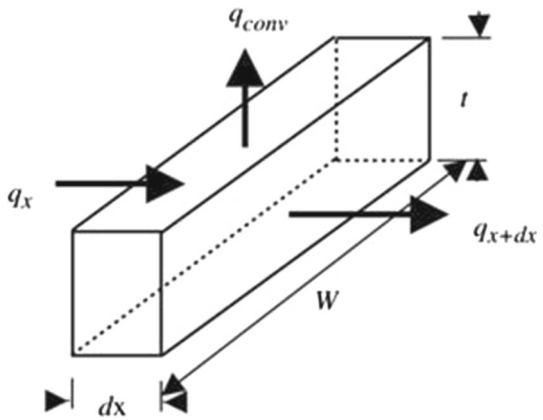

Fig. 4 One-dimensional temperature transfer (Dukhan et al. 2005)

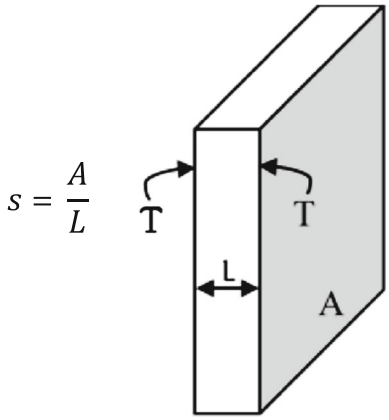

Fig. 5 The large plane wall

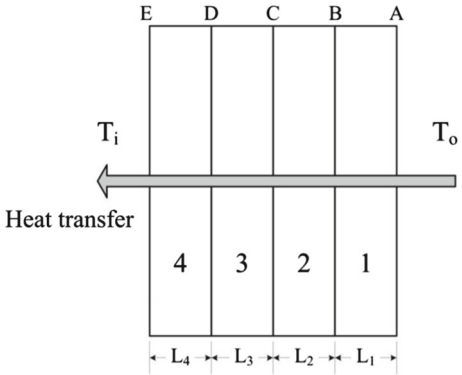

Fig. 6 The wall segmentation (Li et al. 2019)

(2). Has an inverse relation with Δx, according to which the thicker the object, the lower the heat flow per a time unit.
(3). Is directly proportional to ΔT, according to which the larger the temperature difference, the higher the heat flux per a time unit.

The amount of heat passing through an object at t is obtained from the following equation:

$$H = \frac{q}{\Delta t} \propto A\frac{\Delta T}{\Delta x}.$$

The coefficient k can be introduced to modify the above relationship as below:

$$H = \frac{q}{\Delta t} = k.A\frac{\Delta T}{\Delta x}.$$

q: The transferred amount of heat.
k: The heat transfer coefficient.
A: The cross-sectional area of the material from which the transfer is made.
ΔT: The temperature difference between two levels.
Δx : The material thickness.
Assuming the heat transfer per unit time t = 1, we have:

$$H = q = k.A\frac{\Delta T}{\Delta x}.$$

A material with a large k is a good heat conductor if it has a temperature difference (in a large scale) with another object. An object with a small k is a poor conductor or a good insulator.

Tip 1: The coefficient k could be assumed to be constant within the range of temperatures that we normally encounter.

As it is shown in Fig. 7. The heat transfer to the exterior walls of buildings is a classic case of directional use. The exterior wall temperature of the building is assumed to be T_0 with the interior wall temperature being considered T_i in constant conditions. Existing walls usually consist of several layers of different materials. This section focuses on heat transfer through these walls, which is the main source of inspiration for integrating solutions in the proposed algorithm.

$$q = kA\frac{T_1 - T_2}{\Delta x_1}$$

$$\left.\begin{aligned} q &= kA\frac{T_{2-}T_3}{\Delta x_2} \\ q &= kA\frac{T_{3-}T_4}{\Delta x_3} \end{aligned}\right\} T - T_{i+1} = \frac{q_i \Delta x_1}{k_i A}.$$

In constant and steady conditions $\Rightarrow q = q_1 = q_2 = q_3$

$$\Rightarrow q = \frac{T_1 - T_4}{\frac{\Delta x_1}{k_1 A} + \frac{\Delta x_2}{k_2 A} + \frac{\Delta x_3}{k_3 A}}.$$

As discussed above, the heat transfer rate between two objects in physics is finally expressed through the following equations:

$$H = q = kA\frac{\Delta T}{\Delta x}.$$

3.8 Configuration of parameters

The cross section of all objects in a set is considered to be a constant value to prevent interference with the degree of the algorithm penetration. Based on the experimental results, its value ranges from 1 to 1.22 for different applications. It is 1 for the problems with high complexity or skewness, and it is considered 1.22 in other problems.

The k parameter Heat transfer coefficient (FIT): It might be considered the same for everything, based on Tip 1, as the same heat source is used by everything. Nevertheless, for the objects with a higher temperature difference, this value might be slightly higher. Based on the experimental results, the value of k is obtained from the following equation:

$$k = 0.9 + \Delta\text{Cost}.$$

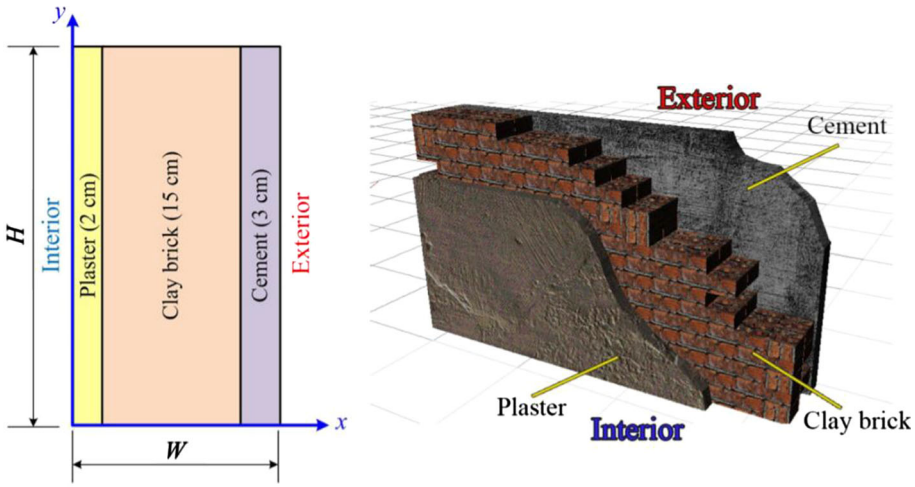

Fig. 7 The simulated walls (No-PCM) (Li et al. 2019)

Thus, k will be a number between 0.9 and 1. In fact, it is 1 (0.999) for the worst member of the nearest integer, whereas k for the members with better-fitness is as follows: $k > 0.9$.

Parameter k helps the HTOA display random behavior across the global optimum and discover and avoid local optima. Adjusting the value of this parameter is very important to strike a balance between exploration and exploitation.

3.9 Exploitation phase

ΔT or Δθ: According to this variable, that the greater the temperature difference between the two objects, the higher the heat transfer between them. This factor is attributed to the temperature difference between dissimilar members of a set when the algorithm is concerned. It is also related to the difference between the decision variables (genes); therefore, as the difference increases, more changes are made to Mi (M_i.Position(j)), which in turn causes a change in the cost or fitness function of Mi.

$$\Delta\theta = M_{\text{Best}}(R).\text{Position}(j) - M_i.\text{Position}(j).$$

The following set contains all the solutions:

$$M = (M_1, M_2, M_3, \ldots, M_n)$$

$\text{M}_{\text{Best}} = \text{M}_{\text{Globalbest}}$.

The following equation is used to measure the temperature difference or the fitness difference between the two solutions:

$$\Delta\theta = M_{\text{Best}}(R) - M_i$$

This parameter is used to maximize the exploitation of the discovered map. It can help the algorithm search for the global optimum. As stated above, the values of two solutions and their convergence rates are controlled by parameter k. The following step is taken to calculate the transferred amount of heat based on the temperature difference between the two objects:

$$q = k.A.\frac{\Delta\theta}{\Delta x}.$$

In some specific applications, $\sin(\Delta x)$ can be used instead of Δx to reduce the over-convergence of solutions on each other and to achieve better exploration.

$$q = k.A.\frac{\Delta\theta}{\sin(\Delta x)}.$$

The value of q is added to the relevant variable as the difference between the two solutions, which represent the distance from the optimal value. This is repeated for all decision variables of low-fitness or out-of-order (R) solutions.

$$M_i^{\text{new}}.\text{Position}(j) = M_i^{\text{old}}.\text{Position}(j) + q.$$

In the above relation, $\sin(\Delta x)$ can be used as another algorithm adjustment screw to enhance the operation. Finally, after each iteration of compound cycle, some of the solutions or those which have not met the minimum fitness or maximum cost can be removed from the system and be replaced with other random solutions. To replace the deleted solutions, new purposive solutions are created and produced around the global optimum.

3.10 Heat balance

It is the temperature or point in which the two objects reach the same temperature after an interval.

$$\theta e = Q1 + Q2 = 0.$$

As the temperature of the object with high-fitness is not supposed to change, the elemental factors or good solutions are changed dramatically because the equality and the

efficiency of the algorithm is reduced by applying this relationship; therefore, this rule is ignored.

Colder temperature received = Colder temperature lost.

3.11 Algorithm

Figure 8 shows the schematic view of the proposed algorithm. The general process of the HTOA starts with generating a random population of materials. The initial population is sorted and 30% of the population with the most fitness is considered the elements of the set R. Depending on the problem complexity, as stated earlier in *Configuration of Parameters*, elements A and Δx are quantified. In the process of producing a new population, 20% of the old population is always added to the final population. In other words, it indicates the retention rate of the old population in the new one. The best solution in the population is kept as global optimization. In the synthesis process, the distances of all decision variables of low-fitness solutions (outside the set R) are randomly measured by one of the elements of the set R. The differences are included in $\Delta\theta$. The method of compensating for this difference, temperature, or fitness is calculated by q which corresponds to the transferred amount of temperature to the colder object. It is finally added to the relevant decision variable in the candidate solution. Nevertheless, to reduce the convergence of solutions, the method of adding a temperature difference or fitness is controlled through the following equation:

$$G = 2/n\mathrm{Var}$$
$$\mathrm{Newpop}(i) = G \times (\mathrm{pop}(i) + q)\,.$$

In the next phase, the mutation factor can also be used to prevent entrapment in local optima; however, to increase the algorithm search speed, it may be neglected and assigned to other variables. It should be noted that the need for a mutation factor is felt in a number of applications and issues, something which is discussed in *Results*. Consequently, the newly generated and sorted population and the R set are updated. Some of the solutions (20% of the population) or those which have not met the minimum fitness or maximum cost were removed from the system and were replaced by other random solutions. This is completely a targeted replacement and done around the optimal global point.

This process is continued until the final cycle of the algorithm. Figure 9 presents the corresponding pseudocode. Some points can be raised to solve the HTOA problems theoretically.

- In some applications, the initial population created for the HTOA may help with better algorithm process and affect the final results to some extent across different iterations. To avoid this problem, it is recommended to run the algorithm more than 5 times in an iteration.
- The exploration is guaranteed by k.
- The exploitation is guaranteed by $\Delta\theta$ and Δx, which make the system easy-to-optimize for the global optimum.
- Adaptive values of k and $\Delta\theta$ in the HTOA allow for a smooth transition between exploration and the exploitation.
- In many cases, the HTOA only has a parameter of composition or measure of discrepancy and has no parameter to modify, something which increases the search speed.

The HTOA speed is rather lower due to the measurement factor of the difference between all decision variables to achieve the global optimum. However, due to the algorithm accuracy in finding the global optimum, this slight deceleration is insignificant. Attempts will be made to apply other forms of factor configurations in heat

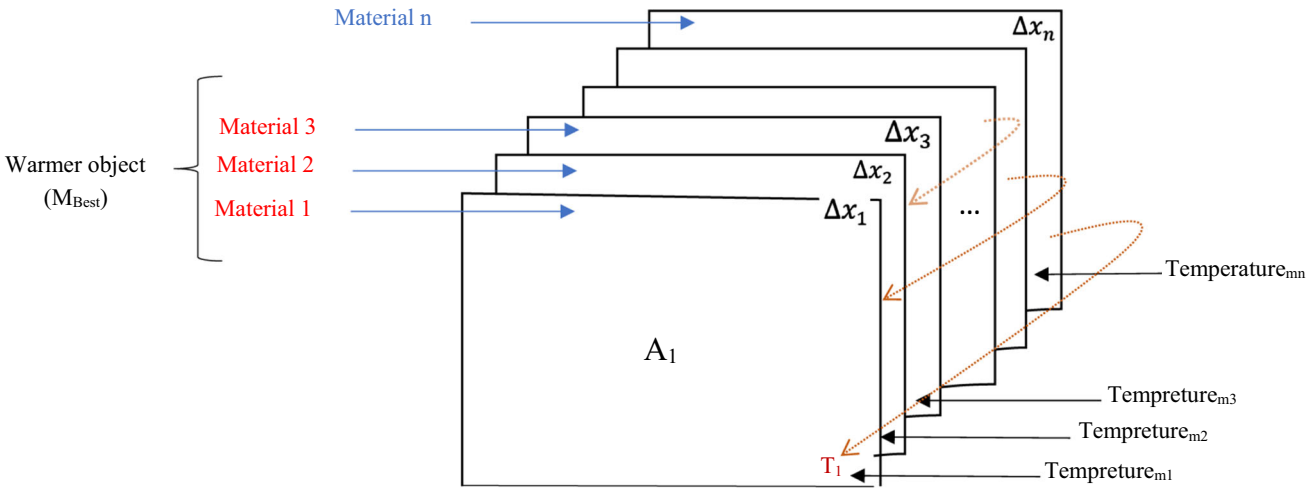

Fig. 8 The schematic of the proposed pattern

Population ⟵ InitializePpulation (Populationsize, Problemsize)
EvaluatePopulation (Population)
Population ⟵ SortPopulation ()
Globalbest ⟵GetBestSolution (Population)
While ¬ StopCondition () do
For (1) each $M_i \in$ Population do
For (2) each search agent
R⟵randi (30%Populationsize)
ΔCost⟵((i-1)/(10×Populationsize))
k⟵0.9+ΔCost
q⟵k.A.Δθ/ sin(Δx)
Δθ⟵$M_{(R)}$.Position-M_i.Position
$M_{(i)}$.Position^{new}⟵(2/Problemsize)×($M_{(i)}$.Position^{old}+q)
Endfor2
EPi⟵ Mutate (M_i^{new}, $P_{mutation}$)
EvaluatePopulation (Population)
Population ⟵ SortPopulation ()
Globalbest ⟵ GetBestSolution (Population)
Endfor1
Vorst⟵ Population − (0.2× Population)
For (3) each $M_{(worst)}$ to Populationsize
Δθ⟵M_{Best}.Position-$M_{\_worst}$. Position
M_i ⟵ Replace (M_{worst}+ (rand (1)× Δθ))
Endfor3
Sbest ⟵ Globalbest
endwhile
return Sbest;

Fig. 9 The pseudo code of the HTOA algorithm

transfer in future studies. Table 2 presents each symbol and notation used in the algorithm pseudocode.

4 Results and discussion

In this section, the HTOA algorithm is evaluated on 26 benchmark functions. The first 20 benchmark functions are the classical functions utilized by many researchers (Abbass 2001; Alatas 2011; Askarzadeh and Rezazadeh 2013; Digalakis and Margaritis 2001; Du et al. 2006; Erol and Eksin 2006; Formato 2007; Gandomi and Alavi 2012; Hansen et al. 2003; Hatamlou 2013; Heidari et al. 2019; Kaveh and Khayatazad 2012; Kaveh and Talatahari 2010; Koza and Koza 1992; Li 2003; Lu and Zhou 2008; Mech 1999; Mirjalili and Lewis 2013; Mirjalili and Lewis 2016; Mirjalili et al. 2014a; Mirjalili et al. 2014b; Moghaddam et al. 2012; Mucherino and Seref 2007; Muro et al. 2011; Pan 2012; Pinto et al. 2007; Rashedi et al. 2009; Rechenberg 1994; Roth and Stephen 2005; Shah-Hosseini 2011; Shiqin et al. 2009; Simon 2008; Webster and Bernhard 2003; Yang 2010a; Yang 2010b; Yang and Deb 2009; Yao et al. 1999). Despite simplicity, these test functions have been selected to compare our results with those of the existing metaheuristic methods. The HTOA is compared with the state-of-the-art and the newest swarm-based optimization algorithms. Table 3 shows the benchmark functions. On this table, dim refers to the function dimensions, whereas range shows the boundary of a functions search space, and Fmin indicates the global optimum. The other testbeds which have been selected were six composite benchmark functions from a CEC 2005 special session (Suganthan et al. 2005). These benchmark functions are the shifted, expanded, and amalgamated variants of the classical functions which offer the greatest complexity amongst the existing benchmark functions (Suganthan et al. 2005). Table 4 lists the CEC 2005 test functions. On this table, dim refers to the function dimensions, whereas range shows the boundary of functions search space, and Fmin indicates the global optimum. Figures 10 and 11 illustrate the 2D versions of the benchmark functions. In general, the benchmark functions are minimization functions which could be divided into four groups called unimodal, multimodal, fixed-dimension multimodal, and composite functions. It is worth mentioning that the detailed definition of the composite benchmark function is accessible in the CEC 2005 technical report (Suganthan et al. 2005). The HTOA was executed 30 times on each benchmark function.

Table 2 The symbols and notations used in the algorithm

Symbol	Describe
A	The cross-sectional area of the material from which the transfer is made
k	The heat transfer coefficient
q	The amount of the heat transferred
ΔT or $\Delta\theta$	The temperature difference between two levels
M_i	Every element of the population
i	Every iteration

Table 3 The benchmark functions (unimodal, multimodal and fixed-dimension) (Jamil and Yang 2013)

Item	Function name	Mathematical expression	Range	Type	Dim	f_{min}
F1	Sphere	$f(x) = \sum_{i=1}^{n}(x_i^2)$	[− 100,100]	Unimodal	30	0
F2	Schwefel 2.22	$f(x) = \sum_{i=1}^{n}\|x_i\| + \prod_{i=1}^{n}\|x_i\|$	[− 10,10]	Unimodal	30	0
F3	Schwefel 1.2	$f(x) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	[− 100,100]	Unimodal	30	0
F4	Schwefel 2.21	$f(x) = \max_{i=1,\ldots,n}\|x_i\|$	[− 100,100]	Unimodal	30	0
F5	Brown	$f(x) = \sum_{i=1}^{n-1}(x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}$	[− 1,4]	Unimodal	30	0
F6	Alpine	$f(x) = \sum_{i=1}^{n} ix_i^4 + \text{random}[0,1]$	[0,10]	Unimodal	30	0
F7	Quartic	$f(x) = 418.9828\ \text{n} - \sum_{i=1}^{n}(x_i \sin\sqrt{\|x_i\|})$	[− 1.28,1.28]	Unimodal	30	0
F8	Schwefel	$f(x) = 418.9828\ \text{n} - \sum_{i=1}^{n}(x_i \sin\sqrt{\|x_i\|})$	[− 500,500]	Multimodal	30	0
F9	Rastrigin	$f(x) = 10\text{n} + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$	[− 5.12,5.12]	Multimodal	30	0
F10	Ackley	$f(x) = -a\exp - \left(b\sqrt{\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(cx_i)\right) + a + \exp(1)$	[− 32,32]	Multimodal	30	0
F11	GRIEWANK	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[− 600,600]	Multimodal	30	0
F12	Salomon	$f(x) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\left(\sqrt{\sum_{i=1}^{D} x_i^2}\right)$	[− 100,100]	Multimodal	30	0
F13	Xin-She Yang	$f(x) = \sum_{i=1}^{n} \epsilon_i\|x_i\|^i$	[− 5,5]	Multimodal	30	0
F14	Ackley N. 2	$f(x) = -200e^{-0.2\sqrt{x^2+y^2}}$	[− 32,32]	Fixed-dimension	2	− 200
F15	Kowalik	$f(x) = \sum_{i=0}^{10}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]$	[− 5,5]	Fixed-dimension	4	0.0003
F16	SIX-HUMP CAMEL	$f(x) = 4x^2(1) - 2.1x^4(1) + \frac{1}{3}x^6(1) + x(1)\times x(2) - 4x^2(2) + 4x^4(2)$	[− 5,5]	Fixed-dimension	2	− 1.0316
F17	Branin's RCOS No.01	$f(x) = \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + \left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	[− 5, 0], [10,15]	Fixed-dimension	2	0.398
F18	Goldstein Price	$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]\times[30 + (2x_1 - 3x_2)^2\times(18 - 32x_1 - 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[− 2,2]	Fixed-dimension	2	3
F19	Hartman 3	$f(x) = -\sum_{i=1}^{4} c_i e^{\left[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right]}$	[0,1]	Fixed-dimension	3	− 3.86
F20	Drop Wave	$f(x,y) = -\frac{1+\cos\left(12\sqrt{x^2+y^2}\right)}{(0.5(x^2+y^2)+2)}$	[− 5.2,5.2]	Fixed-dimension	2	− 1

Table 4 The composite benchmark functions (Jamil and Yang 2013)

Function	Dim	Range	f_{min}
$F_{21}(CF_1)$:	10	[− 5,5]	0
$f_1, f_2, f_3,\ldots, f_{10}$ = Sphere Function			
$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1,1,1,\ldots,1]$			
$[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/100,5/100,5/100,\ldots,5/100]$			
$F_{22}(CF_2)$:	10	[− 5,5]	0
$f_1, f_2, f_3,\ldots, f_{10}$ = Griewank's Function			
$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1,1,1,\ldots,1]$			
$[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/100,5/100,5/100,\ldots,5/100]$			
$F_{23}(CF_2)$:	10	[− 5,5]	0
$f_1, f_2, f_3,\ldots, f_{10}$ = Griewank's Function			
$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1,1,1,\ldots,1]$			
$[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [1,1,1,\ldots,1]$			
$F_{24}(CF_4)$:	10	[− 5,5]	0
f_1, f_2 = Ackley's Function 10 [5,5] 0			
f_3, f_4 = Rastrigin's Function			
f_5, f_6 = Weierstrass Function			
f_7, f_8 = Griewank's Function			
f_9, f_{10} = Sphere Function			
$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1,1,1,\ldots,1]$			
$[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/32,5/32,1,1,5/0.5, 5/0.5,5/100,5/100,5/100,5/100]$			
$F_{25}(CF_5)$:	10	[− 5,5]	0
f_1, f_2 = Rastrigin's Function 10 [5,5] 0			
f_3, f_4 = Weierstrass Function			
f_5, f_6 = Griewank's Function			
f_7, f_8 = Ackley's Function			
f_9, f_{10} = Sphere Function			
$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1,1,1,\ldots,1]$			
$[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [1/5,1/5,5/0.5, 5/0.5,5/100,5/100,5/32,5/32,5/100,5/100]$			
$F_{26}(CF_6)$:	10	[− 5,5]	0
f_1, f_2 = Rastrigin's Function			
f_3, f_4 = Weierstrass Function			
f_5, f_6 = Griewank's Function			
f_7, f_8 = Ackley's Function			
f_9, f_{10} = Sphere Function			
$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [0.1,0.2,0.3, 0.4, 0.5,0.6,0.7,0.8,0.9,1]$			
$[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [0.1 * 1/5, 0.2 * 1/5, 0.3 * 5/0.5, 0.4 * 5/0.5, 0.5 * 5/100, 0.6 * 5/100, 0.7 * 5/32, 0.8 * 5/32, 0.9 * 5/100, 1 * 5/100]$			

Tables 5 and 6 report the analytical results (mean and standard deviation). To verify the results, the HTOA was compared with the PSO (Kennedy and Eberhart 1995), GWO (Mirjalili et al. 2014a) and WOA (Mirjalili and Lewis 2016) regarded as the SI-based techniques. It was also compared with the HHO (Heidari et al. 2019) known as a population-based, gradient-free optimization algorithm.

The HTOA was also compared with three EAs: DE (Storn and Price 1997), GA (Bonabeau et al. 1999), and the culture algorithm (CA) (Omran 2016). In the end, it was compared with the butterfly optimization algorithm, a new comer in the category of nature-inspired metaheuristic algorithms, inspired by the food foraging behavior of butterflies (Arora and Singh 2019). Table 7 shows the scores and ranks of the HTOA algorithm and other algorithms for the benchmark problems.

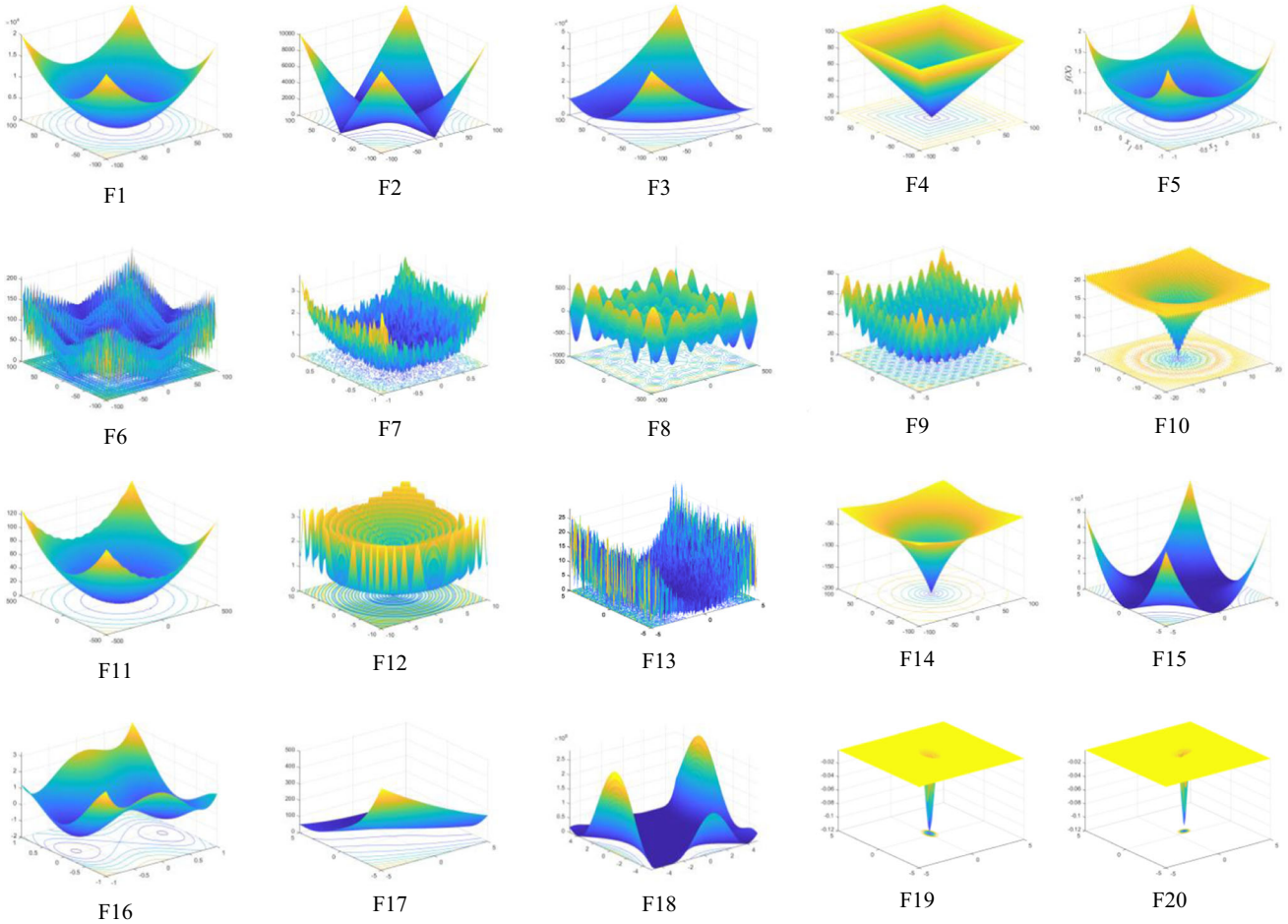

Fig. 10 The 2-D versions of benchmark functions (unimodal, multimodal and fixed-dimension) (Jamil and Yang 2013)

4.1 System hardware

Table 8 presents the detailed specifications of the computer used for implementation, testing, and evaluation steps. Normal hardware requirements were used for this purpose:

1. Multicore Intel(R) Core i7 (TM) 4800MQ CPU @ 2.70 GHZ & 2.70 GHZ.
3. VGA card with (sufficient) capabilities for running HMD-enabled application.
4. Hard disk space and RAM sufficient to run Windows.

4.2 Exploitation analysis

As it is shown in Fig. 5 the HTOA can provide better results than the other algorithms as well as the global minimum fmin. This algorithm outperformed all the other methods in F1–F2-F3-F4-F5-F7-F8-F12-F13. Accordingly, the unimodal functions are appropriate for benchmarking exploitation. Consequently, these results reveal the superior performance of HTOA in exploiting the optimum. This is due to the proposed exploitation operators discussed previously.

4.3 Exploration analysis

In contrast with the unimodal functions, multimodal functions have various local optima, the number of which increases exponentially with the number of dimensions. This makes them appropriate for benchmarking the exploration ability of the associated algorithm. The HTOA is also able to provide very competitive results on the multimodal benchmark functions. This algorithm outperformed other algorithms on most of the multimodal functions (F9-F13). Regarding F13–F20, the HTOA showed very competitive results and outperformed them occasionally. Therefore, the HTOA has the merit of exploration.

4.4 Ability to escape from local minima

The second class of benchmark functions includes composite functions. In general, they are very challenging

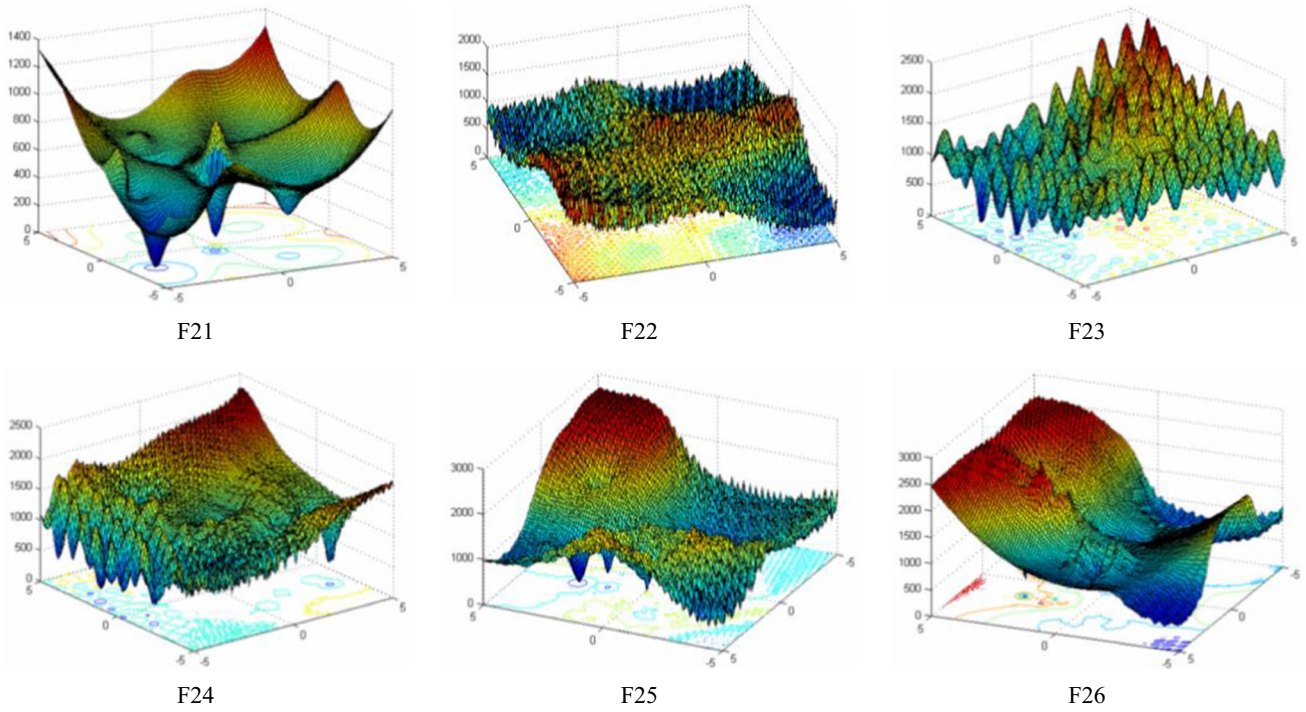

Fig. 11 The 2-D versions of composite benchmark functions (Suganthan et al. 2005)

testbeds for metaheuristic algorithms. Therefore, exploration and exploitation could be benchmarked simultaneously through the composite functions. Furthermore, the local optimum avoidance of an algorithm could also be examined owning to the enormous number of local optima in such test functions. According to Table 9, the HTOA outperformed all the other methods on half of the composite benchmark functions. Table 10 reports the analytical results (standard deviation). The HTOA algorithm also provided very competitive results on the remaining composite benchmark functions. Therefore, the HTOA showed a very good balance between exploration and exploitation, something which resulted in high local optimum avoidance. This superior capability is due to the adaptive values of k and ΔT. As mentioned earlier, half of the iterations were devoted to exploration (k) and the rest to exploitation (ΔT). This mechanism helps the HTOA provide appropriate exploration, avoid local minima, and result in proper exploitation simultaneously. Table 11 shows the scores and ranks of the HTOA algorithm and other algorithms for the composite benchmark functions.

4.5 Analysis of convergence behavior

This section analyzes the convergence behavior of the HTOA. According to Van den Bergh and Engelbrecht (2006), there should be abrupt variations in the movement of search agents over the initial optimization steps. It helps a metaheuristic explore the search space extensively. The results obtained from the HTOA and other algorithms were similar in some of the problems. This is backed by Fig. 12 showing the convergence curves of the HTOA, WOA, HHO and GWO. Accordingly, the HTOA outperformed the other state-of-the-art metaheuristic algorithms.

According to Fig. 12, the HTOA tilts and converges on the optimal solution very quickly, whereas the other algorithms are far behind the HTOA, the search agents of which tend to widely explore the promising areas of search space and make the most of them. The main reason for early convergence might be the relationship used in the composition of solutions which would lead to the early elimination of low-fit solutions and help the algorithm reach the optimal inclination point. Another reason might be the use of several new random solutions to replace the worst solutions in each iteration, which would increase the population diversity. If the solutions are trapped in local optima, they can properly lead the algorithm to better points or areas. However, this finding may not work well in certain problems, although the early convergence on local optima is seen in most optimization problems. Table 5 shows the acceptable performance of this algorithm in optimization problems. The convergence on local optima happened at a very high speed. Accordingly, the algorithm quickly tends to the desired optima in many cases. As it is shown in Table 12 the algorithms presented in this paper showed better results than those of the GWO and HHO

Table 5 The mean results on the unimodal, multimodal and fixed-dimension benchmark test functions in 30 iterations

F	HTOA	GWO	PSO	BBO	WOA	GA	CA	DE	BOA	HHO
F1	**0**	1.45e-27	2.90e-04	21.02	6.48e-74	0.92	2.35e + 03	5.00e-04	1.31e-11	1.85e-88
F2	**4.72e-203**	1.15e-16	0.01	1.64	2.42e-51	0.14	35.11	0.0025	4.72e-09	1.58e-48
F3	**0**	9.73e-06	1.93e + 03	508.21	4.30e + 04	8.89 + 03	1.96e + 04	3.63e + 04	1.25e-11	1.34e-62
F4	**1.17e-200**	7.10e-07	0.24	2.66	456.15	0.82	35.44	12.85	6.11e-09	8.72e-49
F5	**0**	1.65e-30	0.07	0.02	2.81e-76	0.0043	74.97	2.93e-06	9.60e-12	1.50e-98
F6	**2.51e-205**	1.03e-19	8.54	1.99	**0**	0.24	18.29	0.0247	**0**	**0**
F7	**1.64e-04**	0.0021	0.1203	0.0142	0.0047	0.76	0.99	0.055	0.0014	1.243–04
F8	**2.06e + 03**	− 5.81e + 03	− 6.48e + 03	− 8.44e + 03	− 1.019e + 04	− 4.11 + 03	− 7.54e + 03	− 3.06e + 09	− 3.73e + 03	− 1.25e + 04
F9	**0**	1.82	53.44	59.72	3.78e-15	136.55	133.07	85.51	13.71	**0**
F10	**8.88e-16**	1.01e-13	3.93	2.11	3.96e-15	11.71	10.34	0.0059	5.93e-09	**8.88e-16**
F11	**0**	0.0047	1.09	1.19	**0**	32.02	22.96	0.0093	4.66e-12	**0**
F12	**0**	0.19	3.03	0.82	0.13	6.70	6.48	1.07	0.33	3.49e-49
F13	**8.12e-53**	2.85e-18	1.67	0.04	0.0016	115.68	9.93e + 03	0.45	1.17e-04	8.41e-15
F14	**− 200**	**− 200**	**− 200**	**− 200**	**− 200**	− 199.21	**− 200**	**− 200**	**− 200**	**− 200**
F15	0.0012	0.0045	0.0016	8.13e-04	8.12e-04	0.011	**0.0039**	0.001	3.77e-04	4.56e-14
F16	**− 1.0316**	**− 1.0316**	**− 1.0316**	− 1.0044	**− 1.0316**	− 1.011	**− 1.0316**	**− 1.0316**	− 1.34e + 02	**− 1.0316**
F17	**0.398**	0.397	0.397	0.397	0.397	0.397	0.397	0.397	− 4.477	0.397
F18	**3**	**3**	**3**	7.5	3.0001	9.56	**3**	**3**	3.0736	**3**
F19	− 3.6462	− 3.8613	− 3.8370	− 3.8628	− 3.8506	− 3.8078	− 3.8370	− 3.8628	− 3.9983	**− 3.8599**
F20	**− 1.8698**	− 3.2703	− 3.2903	− 3.2705	− 3.2232	− 3.0730	− 3.2641	− 3.3220	− 3.4157	− 3.1305

Table 6 The standard deviation on the unimodal, multimodal and fixed-dimension benchmark test functions in 30 iterations

F	HTOA	GWO	PSO	BBO	WOA	GA	CA	DE	BOA	HHO
F1	0	2.95e-28	2.90e-04	2.7194	9.61e-75	0.8702	1.24e + 03	2.45e-04	8.86e-13	1.93e-102
F2	0	2.40e-16	0.0106	0.3314	2.5193e-51	0.1621	3.0141	3.3890e-04	2.01e-09	2.71e-53
F3	0	2.20e-06	36.3161	167.8321	9.5828e + 03	3.2925 + 03	5.9983e + 03	1.0002e + 04	5.84e-13	1.20e-76
F4	0	5.03e-07	0.2119	0.1479	32.8733	0.7232	5.4623	0.5156	3.58e-10	1.61e-52
F5	0	1.09e-29	8.42e-04	0.0058	3.9586e-78	0.0043	55.5259	1.83e-06	9.30e-13	2.44e-99
F6	0	1.97e-19	0.0396	0.4633	0	0.2054	3.7260	0.0039	0	0
F7	9.07e-05	0.0018	0.0607	0.0053	0.0012	0.0279	0.1762	0.0106	8.23e-04	1.18e-04
F8	382.1235	181.3645	716.4183	762.0621	1.7318e + 03	322.9540	1.0583e + 03	2.3526e + 09	375.5847	0.2274
F9	0	3.8375	9.5678	19.6430	84.5781	2.2506	23.0531	8.2596	6.3448	0
F10	0	9.53e-19	0.7712	0.1233	2.97e-15	0.3750	2.2513	0.0012	5.49e-10	0
F11	0	0	0.0066	0.0286	0.0871	0.0108	4.5403	0.0057	3.4741e-12	0
F12	0.1698	4.95e-10	0.0548	0.0878	0.0999	0.3564	0.7925	0.0999	0.0319	1.05e-48
F13	0	1.83e-30	0.0986	0.0065	0.1893	0.0181	6.26e + 03	0.3830	6.64e-05	3.13e-10
F14	2.00e-14	0	0	2.89e-08	2e-14	9.60e-05	0	0	8.03e-06	0
F15	3.02e-04	0.0107	1.40e-04	0.0040	4.48e-04	0.0012	0.0026	2.22e-04	5.60e-05	5.07e-04
F16	0	1.54e-08	0	0	2.60e-09	3.50e-07	1.92e-16	0	4.7521	2.26e-11
F17	5.18e-08	3.74e-06	0	0	3.74e-06	5.67e-05	0	0	24.0032	3.89e-06
F18	0.0018	3.41e-05	1.19e-15	2.54e-14	3.41e-05	12.0748	4.83e-15	4.44e-16	0.4197	2.12e-08
F19	0.4800	8.98e-04	3.14e-16	2.22e-13	8.9878e-04	1.24e-07	3.1402e-16	0	0.2058	0.0015
F20	0.7530	0.0808	0.0532	7.53e-08	0.0808	2.64e-06	0.0739	1.50e-10	0.7039	0.1280

Table 7 The obtained scores and ranks of the HTOA, GWO, PSO, BBO, WOA, GA, CA, DE, BOA, HHO on the benchmark test functions

F	HTOA	GWO	PSO	BBO	WOA	GA	CA	DE	BOA	HHO
F1	**10**	7	5	2	8	3	1	4	6	9
F2	**10**	7	4	2	8	3	1	5	6	9
F3	**10**	7	6	5	1	4	3	2	8	9
F4	**10**	7	6	5	1	4	2	3	8	9
F5	**10**	7	4	2	8	3	1	5	6	9
F6	**10**	9	8	5	**10**	6	4	7	**10**	**10**
F7	**10**	7	2	5	6	3	1	4	8	9
F8	**10**	7	9	5	3	4	6	4	8	2
F9	**10**	9	6	5	**10**	8	3	4	7	**10**
F10	**10**	8	5	3	9	4	2	6	7	**10**
F11	**10**	6	8	4	**10**	5	3	7	9	**10**
F12	**10**	7	5	4	8	2	1	3	6	9
F13	**10**	9	2	4	7	5	1	3	6	8
F14	8	**10**	**10**	**10**	**10**	9	**10**	**10**	**10**	**10**
F15	3	2	6	7	9	4	1	5	10	8
F16	**10**	**10**	**10**	9	**10**	**10**	**10**	**10**	8	**10**
F17	**10**	9	9	8	9	8	9	9	7	9
F18	**10**	9	**10**	**10**	9	**10**	**10**	**10**	8	**10**
F19	5	9	8	8	7	8	8	8	6	**10**
F20	9	**10**	**10**	**10**	9	**10**	9	10	8	**10**
$\sum$	**185**	156	133	113	152	113	86	119	152	180
Rank	**1**	3	5	7	4	7	8	6	4	2

Table 8 System details

CPU	Intel(R) Core (TM) i7-4800MQ CPU @ 2.70GHZ 2.70 GHZ
RAM	DDR3 1600 MHz, 8 Gb
GPU	Nvidia Quadro K100M, 2 Gb 256bit
SSD	SAMSUNG SSD PM851 mSATA, 128 GB
OS	64-bit Operating System, × 64-based processor
Software	MATLAB R2019a (9.6.0.1072779), License Number: 968398

Table 9 The obtained results on the composite benchmark test functions

ID	HTOA	PSO	WOA	GA	CA	BOA	HHO
CF_1	40.0016	140	185.3050	40.0180	114.4564	394.5764	153.3965
CF_2	80.0014	20	48.1670	60.0065	175.7005	367.3461	57.3045
CF_3	120.0008	60	46.2117	60.0049	141.8486	349.8980	159.4132
CF_4	62.6588	140	106.7404	100.0103	159.4140	451.1071	138.6715
CF_5	20.0014	80	125.0077	80.0189	128.3652	470.4742	77.5940
CF_6	517.6638	40	162.8656	60.0037	285.1967	458.8754	78.0856

algorithms. This comparison is based on the worst and best solutions. According to the standard deviation table, the proposed algorithm had a very low standard deviation; therefore, there was very little difference between solutions. This guarantees and proves the early convergence of the algorithm on the optimal solution. This type of exploration results in very accurate solutions and explores better areas in the search space. However, according to Table 13 it slows down the algorithm in some cases. Also, Table 14 reports CPU time for HTOA and some of algorithms.

Table 10 The calculated standard deviation on the composite benchmark test functions

ID	*HTOA*	*PSO*	*WOA*	*GA*	*CA*	*BOA*	*HHO*
CF_1	54.7718	134.1640	191.5335	54.7636	69.5152	81.3525	128.8373
CF_2	83.6673	44.7213	86.70406	89.4420	89.9597	80.0802	51.2943
CF_3	83.6662	89.4427	52.6417	89.4404	88.9671	97.7832	202.5912
CF_4	87.3873	54.7722	97.8570	122.4743	129.4898	143.1960	109.5118
CF_5	44.7214	83.6660	175.4535	130.3757	116.0120	155.3036	83.3793
CF_6	25.5756	54.7722	113.0414	54.7715	110.3974	137.1245	84.4277

Table 11 The obtained scores and ranks of the HTOA, PSO, WOA, GA, CA, BOA, HHO on the composite benchmark functions

ID	HTOA	PSO	WOA	GA	CA	BOA	HHO
CF1	7	4	2	6	5	1	3
CF2	3	7	6	4	2	1	5
CF3	4	6	7	5	3	1	2
CF4	7	3	5	6	2	1	4
CF5	7	5	3	4	2	1	6
CF6	1	7	4	6	3	2	5
$\sum$	29	31	25	30	16	6	25
Rank	1	2	4	3	5	6	4

Table 15 draws a comparison between the proposed algorithm and the two algorithms ranked as the 1st and the 2nd at CEC 2019 (Price et al. 2019) as well as the CRO algorithm. Since the main codes of most algorithms in this competition are unavailable, an attempt was made to draw a fair comparison with those top two algorithms. In future works, especially regarding large-scale issues, these comparisons will be more and more complete. According to Table 15, the proposed algorithm showed higher accuracy than the top two algorithms of CEC 2019 and the CRO algorithm. Although the execution speeds of these algorithms are higher, accuracy in large-scale problems might be considered more important than speed.

In conclusion, the results verified the HTOA efficiency in solving several benchmark functions compared to the well-known metaheuristic methods. Two real-world problems are discussed in the following sections to further analyze the performance of the proposed algorithm. The

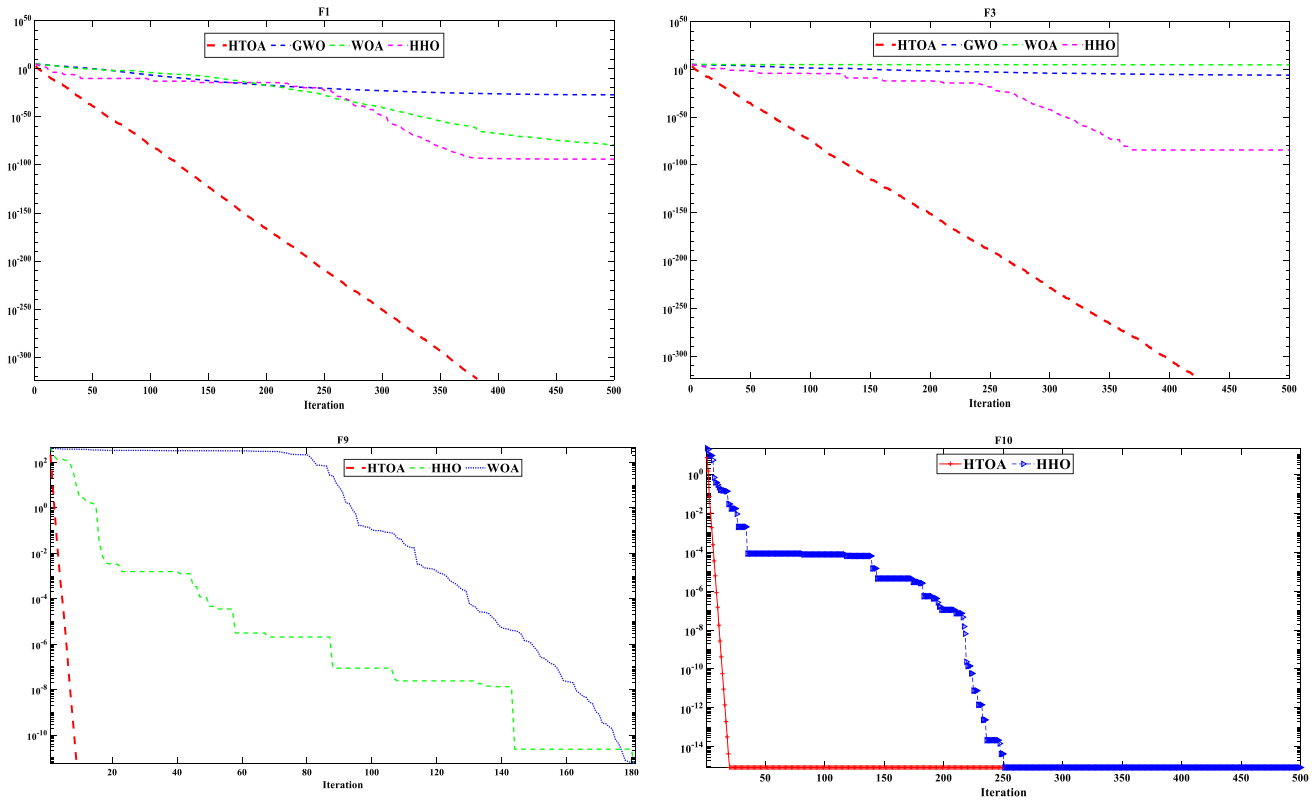

Fig. 12 The comparison of convergence curves of HTOA and some of the algorithms obtained in some of the benchmark problems

Table 12 The worst & best results for some of the top ranked algorithms in 30 iterations

Algorithm	F1		F2		F3		F4		F5		F6	
	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best	Worst	Best
HTOA	0	0	1.3e-202	8.0e-212	0	0	5.83e-198	5.37e-205	0	0	3.78e-204	5.14e-213
GWO	1.38e-27	9.71e-28	1.49e-16	9.04e-17	3.56e-05	9.57e-09	1.19e-06	8.43e-07	1.68e-29	8.28e-31	1.90e-19	6.68e-21
HHO	3.10e-94	5.2e-116	1.97e-48	9.43e-57	1.14e-76	6.49e-102	1.86e-46	5.90e-55	4.72e-99	5.58e-114	0	0

Table 13 The Running times (second)

F	HTOA	GWO	PSO	BBO	WOA	GA	CA	DE	BOA	HHO
F1	1.28	0.15	2.23	1.83	0.20	0.62	2.92	3.59	0.17	0.20
F2	1.38	0.14	2.48	1.64	0.17	0.59	2.73	3.39	0.15	0.19
F3	1.49	0.26	2.93	1.77	0.29	0.76	2.81	359	0.39	0.56
F4	1.28	0.14	2.32	1.68	0.17	0.59	2.82	3.49	0.15	0.24
F5	1.42	0.19	280	1.98	0.23	0.73	3.08	3.86	0.42	0.33
F6	1.51	0.14	2.82	1.67	0.17	0.61	2.72	3.37	0.17	0.34
F7	1.82	0.27	1.53	2.12	0.18	0.80	3.42	0.76	0.37	0.48
F8	1.69	0.21	1.43	2.05	0.11	0.74	3.32	0.73	0.41	0.40
F9	1.63	0.19	1.35	20.04	0.09	0.69	3.30	0.67	0.23	0.35
F10	1.68	0.19	1.41	2.05	0.09	0.72	3.35	0.71	0.22	0.35
F11	1.68	0.21	1.48	2.02	0.11	0.76	3.38	0.76	0.27	0.41
F12	1.83	0.18	1.48	2.09	0.09	0.71	3.29	0.69	0.20	0.26
F13	1.97	0.31	1.72	2.17	0.22	0.82	3.44	0.79	0.46	0.55
F14	0.38	0.08	1.01	0.26	0.07	0.66	0.34	0.62	0.24	0.28
F15	0.47	0.01	1.16	0.41	0.08	0.69	0.56	0.63	0.20	0.30
F16	0.38	0.09	1.03	0.27	0.08	0.68	0.33	0.63	0.26	0.29
F17	0.29	0.15	2.09	0.38	0.12	0.91	0.52	0.68	0.49	0.28
F18	0.35	0.07	0.97	0.20	0.06	0.52	0.26	0.45	0.14	0.22
F19	0.44	0.11	1.07	0.28	0.09	0.57	0.37	0.50	0.32	0.29
F20	0.58	0.15	1.16	0.42	0.10	0.63	0.73	0.58	0.35	0.31

Table 14 CPU time for some of algorithms

Algorithm	F1	F2
HTOA	3.81	3.81
GWO	0.59	0.48
PSO	3.93	4.31
BBO	2.40	2.54
WOA	0.34	0.35
GA	1.25	1.14
CA	3.87	3.85
DE	1.03	0.95
BOA	0.39	0.34
HHO	0.84	0.68

HTOA was also compared with other well-known techniques for the confirmation of results.

5 HTOA in real problems

In this section, the HTOA is tested in two challenging fields of control and data mining, named the PID controller and regression. Since these problems have different constraints, it is necessary to utilize a constraint handling method. Two types of penalty and objective functions are employed to minimize errors on each problem. The most widely used control algorithm in the industry is the proportional integral derivative (PID) controller. The popularity of the PID controller popularity can be attributed to its efficacy in a wide range of operation conditions, its functional

Table 15 Comparison of the HTOA with the state-of-the-art algorithms

Algorithm	Ackley	Rastrigin	Griewank	Schwefel	Schwefel 1.2
HTOA	**8.88e-16**	**0**	**0**	2.06e + 03	**0**
SOMA T3A	2.99e-12	2.9849	3.33e-16	**0.00038**	–
jDE100 (Maučec et al. 2018)	1.09e + 00	1.31e + 03	–	9.40e + 09	–
CRO (Lam and Li 2009)	8.44e-02	6.92e + 00	3.02e-03	–	**0**

simplicity, and its ease of engineering implementation through modern computer technology. The PID controller consists of three parameters that should be tuned such as the proportional term (K_p), the integral term (T_i) and the differential term (T_d). Tuning the parameters might be a challenge for the optimization algorithm to decrease the closed-loop-system error. This study also addresses how to deal with this challenge through the HTOA algorithm. The parameter estimation or prediction is the process of ascertaining an unknown variable from the measurement group through a special statistical method.

The estimation methods should possess certain features called consistence, unbiasedness, minimum variance, efficiency, and sufficiency (Arora and Singh 2019; Van den Bergh and Engelbrecht 2006) so that the estimation values can characterize the statistics of the main set. In statistical modeling, regression analysis is a series of statistical processes implemented for quantifying the relations among variables (Table 5).

As the emphasis is on the relationship between a dependent variable and one or more independent variables (or 'predictors'), it includes many techniques for modeling and analyzing several variables. More specifically, regression analysis assists one understand how the typical value of the dependent variable (or 'criterion variable') alters when any one of the independent variables is diverse, while the other independent variables are held fixed. Calculating a line or a curve to separate or create a boundary between different data could be the purpose of the regression. The main challenge of the algorithm is to estimate this line or curve. Simplicity and low computational cost are the two merits of this method. Nevertheless, the information of infeasible solutions which might be helpful when solving problems with dominated infeasible regions is not utilized in this method. Thanks to simplicity, the HTOA algorithm is equipped with a death penalty function to handle constraints in this section.

6 PID controller

Because of their outstanding effectiveness, implementation simplicity, and broad applicability, the proportional–integral–derivative (PID) controllers are among the most popular controllers used in the industry. The PID controller is a generic control loop feedback mechanism that is used widely in industrial control systems, especially for systems with accurate mathematical models. The PID controller calculation consists of three separate parameters called proportional, integral, and derivative values. Based on the degree at which the error is changed, the proportional value estimates the current error, the integral value determining the sum of recent errors, and the derivative value determining the reactions. The weighted sum of these three actions is imported into the controlled system. Regarding the PID controllers, the major challenge is to tune the parameters precisely and effectively.

In practice, the controlled systems are usually characterized by such aspects as nonlinearity, time variability, and time delay, which make controller parameters tuning more complicated. Furthermore, system parameters and even system structures can vary in time and environment in some cases (Zhang et al. 2009). Hence, the goal of PID controller tuning is to determine the parameters that meet the closed-loop system performance specifications over a wide range of operating conditions. Therefore, amongst the conventional PID tuning methods, the Ziegler–Nichols (ZN) approach is the most celebrated technique. This tuning approach works perfectly for a wide range of practical processes. Nonetheless, it sometimes fails to provide acceptable tuning results and tends to produce a substantial overshoot. To enhance the capabilities of traditional PID parameter tuning techniques, several intelligent approaches have been proposed such as those based on the GA, PSO, and ACO. The HTOA is employed for tuning three parameters of PID controllers in this section.

6.1 Mathematical form

The overall control function could be expressed mathematically as below:

$$u(t) = K_p e(t) + K_i \int_0^t e(t')\mathrm{d}t' + K_d \frac{\mathrm{d}e(t)}{\mathrm{d}t}$$

In this function, K_p, Ki and K_d are nonnegative parameters denoting the coefficients for proportional, integral, and derivative terms, respectively. In the standard form, K_i and K_d are replaced by $\frac{K_p}{T_i}$ and $K_p T_d$, respectively. As a result, T_i and T_d have understandable physical meanings,

for they represent the integration time and the derivative, respectively.

$$\text{Output}(t) = K_p\left(e(t) + \frac{1}{T_i}\int_0^t e(t)\mathrm{d}t + T_d\frac{\mathrm{d}e}{\mathrm{d}t}\right)$$

In this function, k shows discrete time (0, 1, 2, …), whereas u_k is the manipulated variable (MV) at time k, and e_k is equal to PV − SP or the error rate at time k. Moreover, SP refers to the set point at time k, whereas PV indicates the process variable at time k, and Kp shows the proportional gain. Finally, Ti refers to the integral time, whereas Td shows the derivative time, and Ts refers to the sample time.

Evidently, the fine-tuning of these parameters may increase the controller accuracy.

It is nearly impossible to manually adjust these parameters. As noted previously, although there are common methods which provide relatively good accuracy, they certainly lack the convergence speed of evolutionary approaches toward the optimum. However, the HTOA can largely rise up to this challenge and concurrently achieve unerring accuracy.

- *Proportional Control* The control signal is partly commensurate with the tracking error; however, it has limited performance. It enhances the system speed and increases its overshoot. As a result, the persistent error is reduced; however, it cannot be completely eliminated. Furthermore, the rise time is reduced.
- *Integral Control* A portion of the control signal is produced in proportion to the integral of the tracking error, whereas the steady-state error is completely eliminated, and the overshoot is similarly increased by this controller. Therefore, the relative stability of the system and the rise time are reduced, although the settling time increases.
- *Derivative Control* A portion of the control signal is produced in proportion to the derivation of the tracking error. This controller responds to the signal change rate and can cause noteworthy correction before the error becomes substantial and increases the relative stability of the system. It also reduces the speed and the overshoot similarly (improving the transient response).

The open loop and closed loop methods (Ziegler and Nichols) are two techniques for tuning a PID controller. Tuning the PID controller parameters has widely been studied and discussed over the years. The tuning process usually needs a great deal of human power, energy, and time. In the worst-case scenario, the bad tuning parameters may result in poor controlling performance or even crash in the control system. As discussed earlier, the PID tuning process aims to meet the closed-loop system performance specifications such as peak overshoot, settling time, rising time, and the robust performance of the control loop in different situations. Nevertheless, it is often difficult to concurrently achieve all of these desirable qualities practically. For instance, if the PID controller is adjusted to provide a good transient response to a set point change, it usually results in a bad steady state characteristic. At the same time, if the control system is forced to generate a small steady state error by selecting conservative values for the PID controller, it may result in a slow closed-loop response to a set point change. To solve this problem, optimization methods have been used to tune PID parameters. Comparing with the conventional Ziegler–Nichols (ZN) method, the optimization methods such as GA, PSO and ACO have proved their excellence in yielding better results by improving the steady state characteristics and performance metrics. The GA was proposed as a method of auto-tuning PID controllers by Jones and Oliveira (1995). The continuous GA was then proposed to improve the operation time and efficiency for PID adjustment (Chen 2007; Park and Choi 1996). The GA uses the chromosome fitness value to create a new population consisting of the fittest members, and each chromosome consists of three separate strings constituting a P, I, or D term. Figure 13 shows the diagram of a closed-loop PID system. In tuning PID controllers, the main objective is to adjust the reactions of PID controllers to set point changes and unmeasured disturbances; thus, the control error variability is minimized. The PID controllers are implemented primarily for the purpose of holding measured process values at a set point or a desired value. Figure 14 illustrates the effects of adjustment parameters on the final outputs of the controller.

6.2 Simulation results

6.2.1 Typical objects in simulation experiments:

Comparison experiments were conducted between the HTOA and the other optimization algorithms such as the PSO, HHO, GA, DE, and GWO to verify the efficiency of

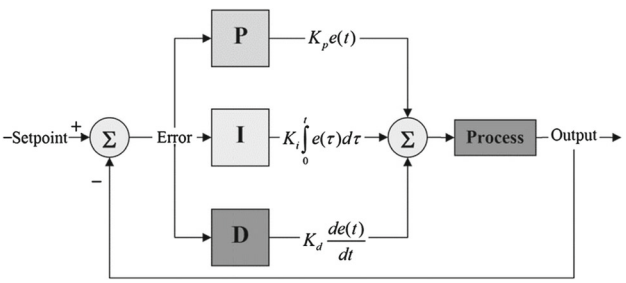

Fig. 13 The diagram of a closed-loop PID system (Wikipedia Contributors https://en.wikipedia.org/w/index.php?title=PID_controller&oldid=925590136)

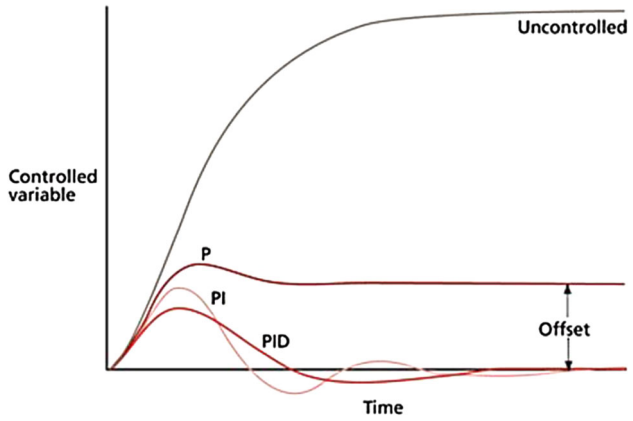

Fig. 14 The effect of the adjustment parameters on the final output of the controller (Murrill 1991)

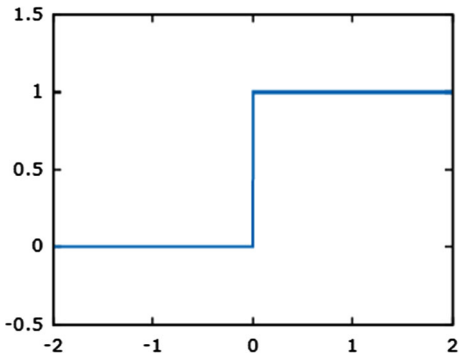

Fig. 15 The step function (input function)

the proposed algorithm in tuning the PID controller. The tuning methods are tested through different kinds of control systems such as the second-order system, which is showed in equation below. Figure 15 indicates the input function.

$$G = \frac{1}{(s+1)^2}$$

6.3 Initialization of HTOA parameters

The following parameters were employed to verify the performance of the HTOA-PID controller on different experiments. The tuning parameters were k_p, k_i, and k_d, whereas the population size was nPop = 50. The number of iterations and the sampling time are shown by MaxIt = 100 and t_s = 0.01 s, respectively. The other HTOA parameters are initialized as below:

$$\text{Dim} = 3,\ \text{keep rate} = 0.2, A = 1, \Delta x = 0.65,\ \text{without mutation},\ K = 0.9 + \Delta\text{Cost},\ q = (KA)\frac{\Delta\theta}{\Delta x}$$

This problem is intended to tune the values of *Kp, Ki,* and *Kd* to minimize the system error.

6.4 Comparison of the HTOA and other algorithms

Table 16 presents the simulation results showing the optimal solutions. The HTOA results were better optimized, and the algorithm managed to minimize the cost function. As it is shown in Fig. 16 the HTOA controller was able to create very perfect step response, indicating that the HTOA outperformed the other algorithms in parameter optimization for the PID controller. In addition, the HTOA obtained the best value after the 10th iteration, whereas the PSO fell into the local minimum until the 50th iteration and obtained the best value after the 55th iteration. Consequently, the HTOA controller can search the optimal parameters more quickly and efficiently than the other algorithms. Hence, the final results of evolution with HTOA were better than the other algorithms. It is very important that the HTOA obtained the best value from the first iteration.

Table 16 The simulation results for PID controller

Algorithm	Optimum Variables						Optimum cost
	K_p	K_i	K_d	MaxRealPole	SI	MP	
HTOA	7.6072	3.6994	3.5470	0.9093	1.0998	9.6367e-12	1.6825
PSO	7.8329	3.7361	3.7332	0.8898	1.1238	0	1.6888
HHO	100	− 0.158	− 1.490	0	-inf	86.5300	-inf
GA	8.2447	4.6029	4.3821	0.7938	1.2597	0.4908	2.2356
DE	8.0472	4.7756	5.1538	0.6664	1.5006	0.4750	3.3298
GWO	7.6310	3.7049	3.5306	0.8294	1.2057	0.1227	1.9110

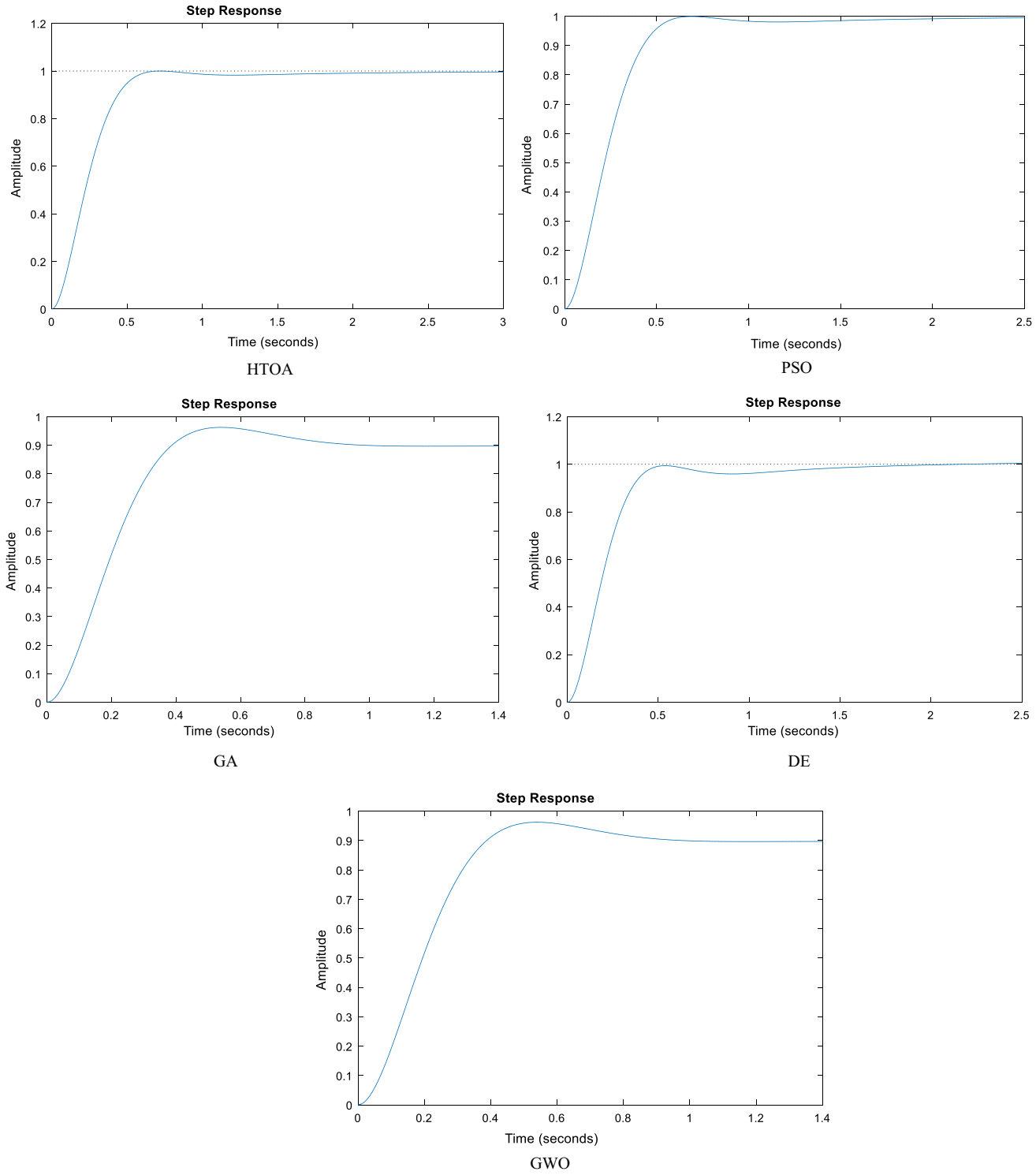

Fig. 16 The step responses of the system with optimal PID controllers using HTOA, PSO, GA, GWO

7 Linear regression

The regression problem was first introduced by Legendre and Gauss in the eighteenth century. The methodology had been used nearly exclusively in the physical sciences until the late nineteenth century. The relation function, which is obtained from the regression analysis, explains the relationship that is the closest to the real one, although the real relationship between the dependent and independent variables could not be detected completely with regression

analysis. The regression function determination consisted of two stages. The type of relationship between the variables was determined in the first stage. Then, the regression functions were determined through a calculation method that was selected according to the relationship type in the second stage. The relationships between the independent and dependent variables might be of the linear, curvilinear, polynomial, quadratic, or logarithmic type (Bárdossy 1990).

The regression analysis can be defined as a statistical procedure explaining the relationship between two or more variables. The relationship between two kinds of measurements is described by the regression analysis as the independent or predictor measurements denoted by $x = (\times 1, \times 2, \ldots, xk)$ and the dependent or response measurements, denoted by y. The general form of a regression model is $y = (x, \beta) + \varepsilon$. The response y consists of two parts called the systematic part $f(x)$ that depends on x and the random part ε that is independent from predictors. In a linear regression model, the regression function is a linear function of the unknown parameters; however, in a nonlinear regression model, the regression function is not a linear function of the unknown parameters. Nonlinear regression models are formally defined as the models in which at least one of the model parameters occurs nonlinearly in the model expression. They are employed to model complex interrelationships between variables and play a key role in various scientific disciplines and engineering fields. The growth, the yield density, and the dose–response models with other similar models, used to describe the physical, biological, industrial, and econometric processes are the prevalent examples of nonlinear models (Nash and Walker-Smith 1987; Watts 1984).

A linear approach was defined as the linear regression to model the relationship between a scalar response (or a dependent variable) and one or more explanatory variables (or independent variables). One explanatory variable was termed as the simple linear regression, and the process was defined as the multiple linear regression for more than one explanatory variable. This term is dissimilar to the multivariate linear regression in which multiple correlated dependent variables are predicted rather than a single scalar variable.

The relations are modeled and named linear models in the linear regression through the linear predictor functions whose unknown model parameters are estimated from the data (Seal, 1967). Most widely, the conditional mean of the response given to the values of explanatory variables (or predictors) is presumed to be an affine function of those values. Less widely, the conditional median or some other quantile is then utilized. Akin to all forms of the regression analysis, linear regression emphasizes the conditional probability distribution of the response given to the values of predictors rather than the joint probability distribution of all of these variables, something which is the domain of multivariate analysis. The linear regression is the first type of regression analysis which has been studied thoroughly and utilized widely in practical applications (Yan and Su 2009). This is because of the models which depend linearly on their unknown parameters and are easier to fit in comparison to those models which are nonlinearly related to their parameters and determine the statistical properties of the resultant estimators.

The linear regression models frequently fitted through the least squares approach; however, they may also be fitted in other ways such as minimizing the "lack of fit" in some other norms (with least absolute deviations regression), minimizing a penalized version of the least squares cost function such as ridge regression (L2-norm penalty) and lasso (L1-norm penalty). Conversely, the least square approach can be employed to fit the models which are not linear. The terms "least squares" and "linear model" are closely linked; however, they are not identical. Figure 17 shows the model of regression.

7.1 Implementation of HTOA for parameter estimation in linear regression model

Since the sum of the squared error function estimation was used in the present study, the sum of the squared error function (β) should be minimized to provide a solution in the real parameter neighborhood. Therefore, the cost (fitness) function in the HTOA search engine is selected as β:

$$S(\beta) = \sum_{i=1}^{n} (y_i - f(x_i, \beta))^2$$

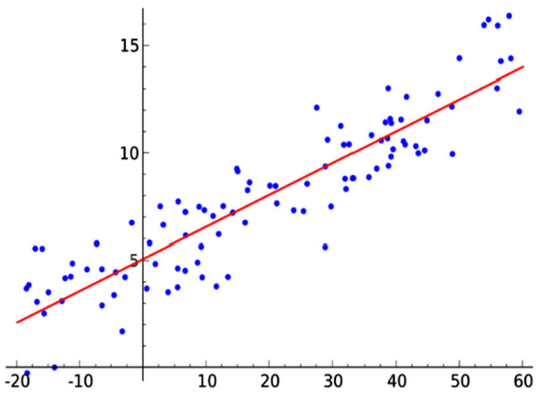

Fig. 17 The model of regression (Wikimedia Commons Contributors https://commons.wikimedia.org/w/index.php?title=File:Linear_regression.svg&oldid=343115066)

Table 17 The Measurement values

NN	x	y
1	0	1
2	1	1
3	2	2
4	4	2

7.2 Simulation results

The comparison experiments between the HTOA and the other optimization algorithms such as the PSO, HHO, GA, DE, and GWO were provided to verify the efficiency of the proposed algorithm in estimating the linear regression parameters. The tuning method was tested through two kinds of dimensions and polynomials. For instance, the second-order system is as follows:

$$\text{polynomial } 1 = a_0(x_0)^0 + a1(x1)^1$$

Table 17 shows the values and the general distributions of measurements:

7.3 Initialization of HTOA parameters

The following parameters were employed to verify the performance of the HTOA regression on experiments. The tuning parameter was S(β), and the population size was nPop = 100. The number of iterations was set MaxIt = 500. The other HTOA parameters were initialized as below:

$$\text{Dim} = 2 \text{ and } 4,\ \text{keep rate } = 0.2,\ A = 1, \Delta x = 0.655,$$
$$\text{mutation} = 0.02,\ K = 0.9 + \Delta\text{cost}, q = (KA)\sin\left(\frac{\Delta\theta}{\Delta x}\right)$$

Notice that this problem aims to minimize the system error.

Table 18 indicates the mean and standard deviation of the best optimal solution. For each polynomial test function, the algorithms utilized 30 independent runs, each of which used 500 iterations. To demonstrate the effectiveness of the proposed algorithm, it was compared with four acclaimed optimization algorithms. According to Table 18, the HTOA and other algorithms yielded similar, and the algorithm presented in this particular application provided highly competitive results. Figure 18 shows the algorithm convergence on the optimal solution in comparison with the other algorithms, whereas Fig. 19 illustrates the boundary line created by the proposed algorithm on test points.

Table 18 The obtained results for the linear regression

Algorithm	Average (mean) optimum cost	Standard deviation
HTOA	1.5429	2.2204e-16
HHO	1.5429	3.9250e-07
GA	1.5429	0
DE	1.5429	0
GWO	1.5429	2.1718e-08

7.4 Estimating nonlinear regression parameters

The nonlinear regression models are used widely to model the stochastic phenomena and to estimate the parameters that have a key role in the inference of models. The regression function is a linear function with unknown parameters in a linear regression model. By contrast, in a nonlinear regression model, the regression function is not a linear function of unknown parameters. At least one of the model parameters arises nonlinearly in the nonlinear regression models.

Nonlinear models are employed to model complex reciprocal relationships between variables and play a significant role in various scientific disciplines and engineering fields. Figure 20 shows the cubic polynomial regression.

As noted earlier, the tuning method was tested in two kinds of dimensions and polynomials. The third-order system is shown as the following equation:

$$\text{Polynomial } 1 = a_0(x_0)^0 + a_1(x_1)^1 + a_2(x_2)^2 + a_3(x_3)^3$$

Table 19 indicates the mean and standard deviation of the best optimal solution. It can be claimed that the HTOA had better exploitation capabilities and managed to find the best optimal solution very efficiently. Figure 21 reports the created boundary line among distribution of the measurement values by HTOA and Fig. 22 shows the comparison of the convergence curves of the HTOA and some of the algorithms.

8 Conclusion

This study proposed a novel metaheuristic optimization algorithm inspired by the heat transfer relations. The proposed method consists of two parameters for controlling and balancing the exploration and exploitation phases. Twenty-six test functions were employed to benchmark the performance of the proposed algorithm in terms of exploration, exploitation, local optimum avoidance, and convergence. In comparison with some of the acclaimed metaheuristic methods such as the PSO, HHO, GA, BBO, GWO, WOA, BOA, DE and CA, the findings indicated that the HTOA was able to provide highly competitive results. First, the results of the unimodal functions showed the

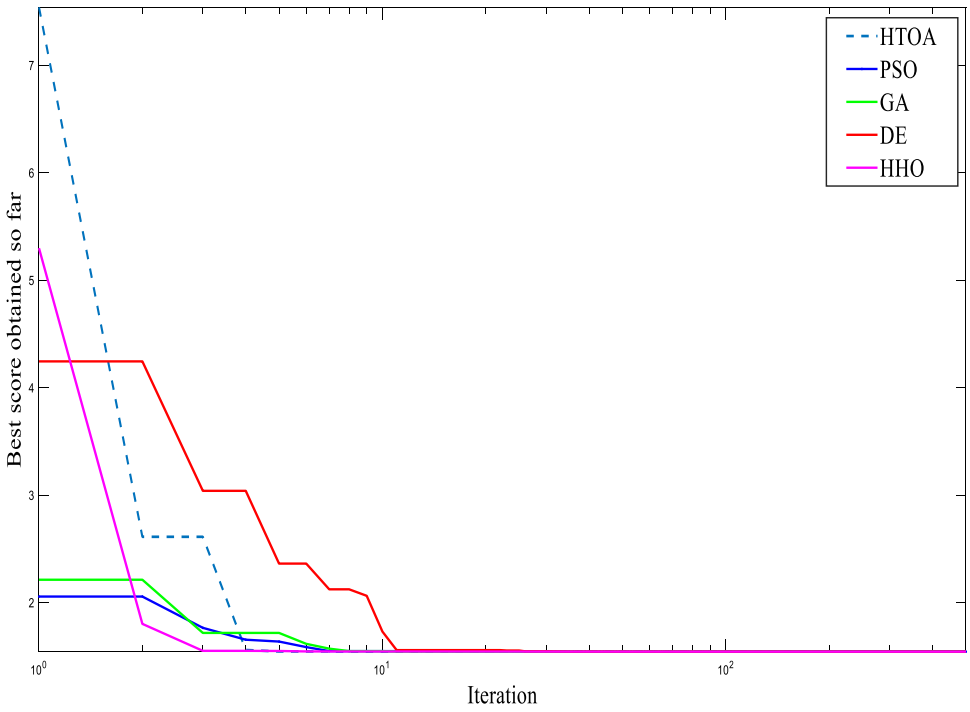

Fig. 18 The comparison of the convergence curves of the HTOA and some other algorithms

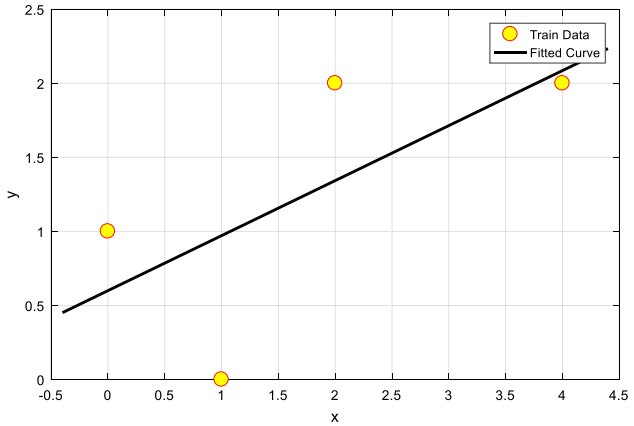

Fig. 19 The created boundary line among distribution of the measurement values by HTOA

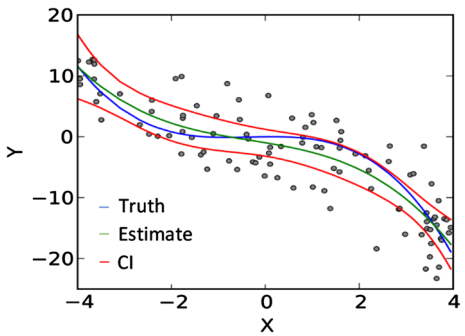

Fig. 20 The cubic polynomial regression (contributors 2016)

superior exploitation capability of the HTOA algorithm. Second, the exploration capability of the HTOA was confirmed by the results of multimodal functions. Third, the results of the composite functions indicated high local optimum avoidance rate. The algorithm convergence was also confirmed by the convergence analysis of the HTOA. The HTOA was finally used to solve real-world problems in controlling and data mining fields named PID controller and regression. Compared with the existing approaches, the results of these problems showed a substantial improvement in the minimized error of the PID controller and regression. This finding reveals the applicability of the proposed algorithm in solving real-world problems. The results of semi-real and real problems also validated that the HTOA showed high performance not only in unconstrained problems but also in constrained problems. The HTOA handled various kinds of constraints more effectively and provided better solutions than the other optimization approaches. There are some research directions that might be proposed for forthcoming studies. The binary version of the HTOA can be perceived as a future

Table 19 The obtained results for the nonlinear regression

Algorithm	Average (mean) optimum cost	Standard deviation
HTOA	5.3367e-04	0.0012
PSO	0.0415	0.0387
HHO	0.3177	0.1281
GA	0.3654	0.2655
DE	0.1326	0.0970
GWO	0.2297	0.5135

contribution. Similarly, it is also recommended to generalize this algorithm to solve the real-world multi-objective and many-objective optimization problems in future works. Moreover, future tasks include the multi-objective optimization of this algorithm in addition to providing a general framework and a fixed relationship to adjust the variable parameters in the algorithm. Manual adjustment in the form of trial and error (*e.g.* adjusting parameter k) is a disadvantage of this algorithm, which can be considered an

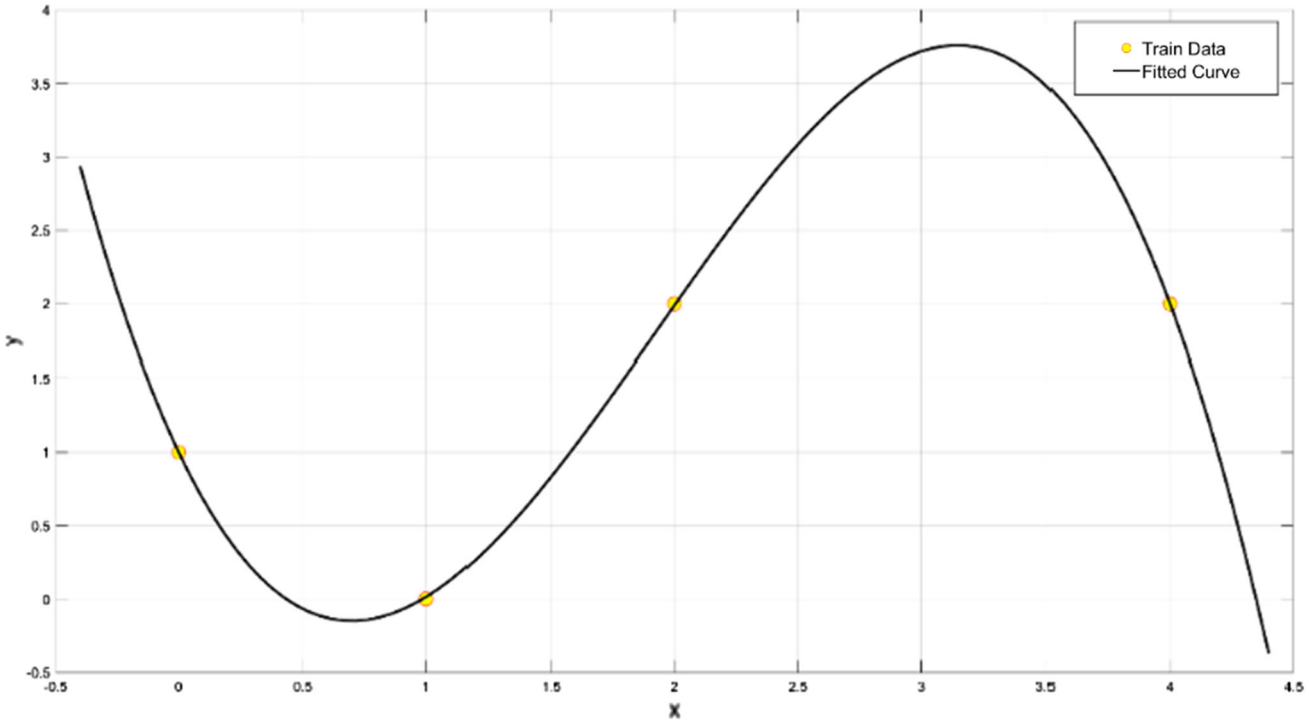

Fig. 21 The created boundary line among distribution of the measurement values by HTOA

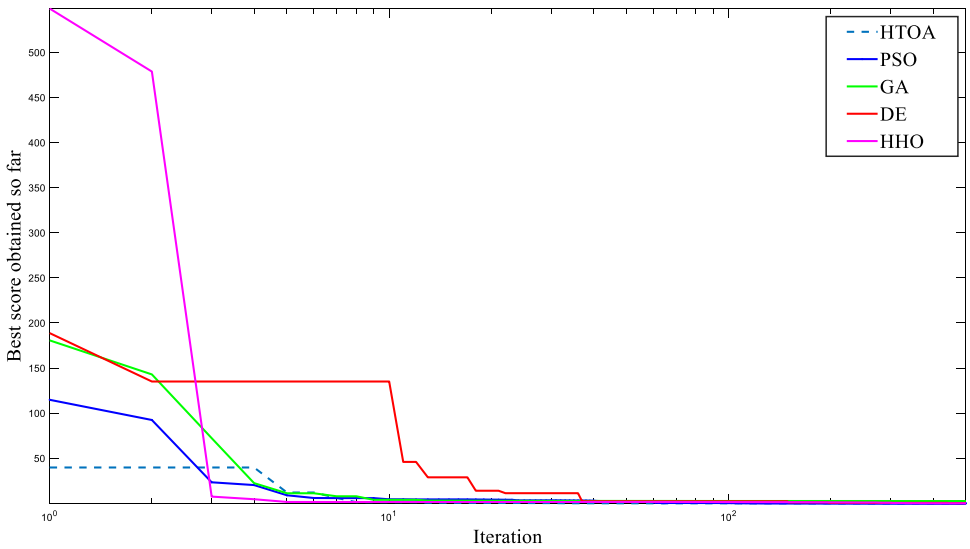

Fig. 22 The comparison of the convergence curves of the HTOA and some other algorithms

area of future research; however, it does not pose much trouble for the operator.

Funding No funding was received for this work.

Declarations

Conflict of interest The authors declare that they have no competing interests

Ethical approval We further confirm that any aspect of the work covered in this manuscript that has involved human patients has been conducted with the ethical approval of all relevant bodies and that such approvals are acknowledged within the manuscript.
IRB approval was obtained (required for studies and series of 3 or more cases)
Written consent to publish potentially identifying information, such as details or the case and photographs, was obtained from the patient(s) or their legal guardian(s).

Human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study

References

Abbass HA MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach. In: Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546), 2001. IEEE, pp 207–214

Alatas B (2011) ACROA: artificial chemical reaction optimization algorithm for global optimization. Expert Syst Appl 38:13170–13180

Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput 23:715–734

Askarzadeh A, Rezazadeh A (2013) A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: bird mating optimizer. Int J Energy Res 37:1196–1204

Bárdossy A (1990) Note on fuzzy regression. Fuzzy Sets Syst 37:65–75

Basturk B (2006) An artificial bee colony (ABC) algorithm for numeric function optimization. In: IEEE swarm intelligence symposium, Indianapolis, IN, USA.

Beni G, Wang J (1993) Swarm intelligence in cellular robotic systems. In: Robots and biological systems: towards a new bionics? Springer, pp 703–712

Bonabeau E, Marco DdRDF, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems, vol 1. Oxford University Press, Oxford

Chen S-F (2007) Particle swarm optimization for PID controllers with robust testing. In: 2007 international conference on machine learning and cybernetics, IEEE, pp 956–961

Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Softw 114:48–70

Dhiman G, Kumar V (2018) Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. Knowl-Based Syst 159:20–50

Digalakis JG, Margaritis KG (2001) On benchmarking functions for genetic algorithms. Int J Comput Math 77:481–506

Dorigo AM, Birattari M (2006) Thomas St utzle, Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique IEEE CIM

Dorigo M, Birattari M (2010) Ant colony optimization. Springer

Du H, Wu X, Zhuang J (2006) Small-world optimization algorithm for function optimization. In: International Conference on Natural Computation, Springer, pp 264-273

Dukhan N, Quinones-Ramos PD, Cruz-Ruiz E, Vélez-Reyes M, Scott EP (2005) One-dimensional heat transfer analysis in open-cell 10-ppi metal foam. Int J Heat Mass Transf 48:5112–5120

Erol OK, Eksin I (2006) A new optimization method: big bang–big crunch. Adv Eng Softw 37:106–111

Formato RA (2007) Central force optimization. Prog Electromagn Res 77(1):425–491

Gandomi AH, Alavi HA (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845

Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Comput 11:1–18

Hatamlou A (2013) Black hole: A new heuristic optimization approach for data clustering. Inf Sci 222:175–184

He S, Wu Q, Saunders J A novel group search optimizer inspired by animal behavioural ecology. In: 2006 IEEE international conference on evolutionary computation, 2006. IEEE, pp 1272–1278

Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: Algorithm and applications. Futur Gener Comput Syst 97:849–872

Holland J (1992) Genetic Algorithms Scientific American 267:66–72. https://doi.org/10.1038/scientificamerican0792-66

Jamil M, Yang X-S (2013) A literature survey of benchmark functions for global optimization problems arXiv preprint https://arxiv.org/abs/1308.4008

Jones A, Oliveira P (1995) "Genetic auto-tuning of PID controllers". In: Proceedings of international conference on genetic algorithms in engineering systems: innovations and applications, pp 141–145

Karaboga D, Basturk B (2007) Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. International fuzzy systems association world congress. Springer, pp 789–798

Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. Comput Struct 112:283–294

Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213:267–289

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, Piscataway NJ, IEEE Service Center, pp 1942–1948

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680

Koza JR, Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection, vol 1. MIT press, Cambridge

Koza JR, Rice JP (1992) Automatic programming of robots using genetic programming. In: AAAI, Citeseer, pp 194–207

Lam AYS, Li VOK (2009) Chemical-reaction-inspired metaheuristic for optimization. IEEE Trans Evol Comput 14:381–399

Li X (2003) A new intelligent optimization-artificial fish swarm algorithm Doctor thesis. Zhejiang University of Zhejiang, China

Li Z, Al-Rashed AA, Rostamzadeh M, Kalbasi R, Shahsavar A, Afrand M (2019) Heat transfer reduction in buildings by embedding phase change material in multi-layer walls: Effects

of repositioning, thermophysical properties and thickness of PCM. Energy Conv Manag 195:43–56
Lienhard JH (2019) A heat transfer textbook. Dover Publications
Lu X, Zhou Y (2008) A novel global convergence algorithm: bee collecting pollen algorithm. In: International conference on intelligent computing, Springer, pp 518-525
Maučec MS, Brest J, Bošković B, Kačič Z (2018) Improved differential evolution for large-scale black-box optimization IEEE. Access 6:29516–29531. https://doi.org/10.1109/ACCESS.2018.2842114
Mech LD (1999) Alpha status, dominance, and division of labor in wolf packs. Canadian J Zool 77:1196–1203
Mirjalili S, Lewis A (2013) S-shaped versus V-shaped transfer functions for binary particle swarm optimization. Swarm Evolut Comput 9:1–14
Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
Mirjalili S, Mirjalili SM, Lewis A (2014a) Grey wolf optimizer. Adv Eng Softw 69:46–61
Mirjalili S, Mirjalili SM, Yang X-S (2014b) Binary bat algorithm. Neural Comput Appl 25:663–681
Moghaddam FF, Moghaddam RF, Cheriet M (2012) Curved space optimization: a random search based on general relativity theory arXiv preprint https://arxiv.org/abs/1208.2214
Mucherino A, Seref O (2007) Monkey search: a novel metaheuristic search for global optimization. In: AIP conference proceedings, vol 1. AIP, pp 162–173
Muro C, Escobedo R, Spector L, Coppinger R (2011) Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. Behav Process 88:192–197
Murrill PW (1991) Fundamentals of process control theory. Instrument Society of America Research Triangle Park, NC
Nash JC, Walker-Smith M (1987) Nonlinear parameter estimation Dekker. New York 5:5014–5019
Omran MG (2016) A novel cultural algorithm for real-parameter optimization. Int J Comput Math 93:1541–1563
Pan W-T (2012) A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowl-Based Syst 26:69–74
Park J-H, Choi Y-K (1996) An on-line PID control scheme for unknown nonlinear dynamic systems using evolution strategy. In: Proceedings of IEEE international conference on evolutionary computation. IEEE, pp 759–763
Pinto PC, Runkler TA, Sousa JM (2007) Wasp swarm algorithm for dynamic MAX-SAT problems. In: International conference on adaptive and natural computing algorithms, Springer, pp 350-357
Price K, Awad NH, Ali MZ, Suganthan P (2019) The 2019 100-Digit Challenge on Real-Parameter, Single Objective Optimization: Analysis of Results; Technical Report 2019. https://www.ntu.edu.sg/home/epnsugan/index_files/CEC2019/CEC2019.htm. Accessed 12 April 2020
Rashedi E, Hossein N-P, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf sci 179(13):2232–2248
Rechenberg I (1994) Evolutionsstrategie'94, volume 1 of Werkstatt Bionik und Evolutionstechnik Frommann Holzboog, Stuttgart
Roth M, Stephen W (2005) Termite: A swarm intelligent routing algorithm for mobile wireless ad-hoc networks. In: Stigmergic Optimization, vol 31. Springer, Berlin, Heidelberg, pp 155–184. doi:https://doi.org/10.1007/978-3-540-34690-6_7
Shah-Hosseini H (2011) Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. Int J Comput Sci Eng 6:132–140
Shiqin Y, Jianjun J, Guangxing Y (2009) A dolphin partner optimization. In: 2009 WRI Global Congress on Intelligent Systems, IEEE, pp 124–128
Simon D (2008) Biogeography-based optimization. IEEE Trans Evolut Comput 126:702–713
Slowik A, Kwasnicka H (2017) Nature inspired methods and their industry applications—Swarm intelligence algorithms. IEEE Trans Ind Inf 14:1004–1015
Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11:341–359
Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization KanGAL report 2005005:2005
Thermal Conductivity of selected Materials and Gases (2003) [online] Available at: https://www.engineeringtoolbox.com/thermal-conductivity-d_429.html. 2019
Truong Tung Khac, Li Kenli, Yuming Xu (2013) Chemical reaction optimization with greedy strategy for the 0–1 knapsack problem. Appl Soft Comput 13(4):1774–1780
Van den Bergh F, Engelbrecht AP (2006) A study of particle swarm optimization particle trajectories. Inf Sci 176:937–971
Watts DG (1984) Nonlinear regression modeling: a unified practical approach (Statistics: Textbooks and Monographs Series), vol 48. Taylor & Francis Group, UK
Webster B, Bernhard PJ (2003) A local search optimization algorithm based on natural principles of gravitation.
Wikipedia Contributors PID controller. Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=PID_controller&oldid=925590136. Accessed 11 November 2019 02:10 UTC
Wikimedia Commons Contributors Linear regression.svg. Wikimedia Commons, the free media repository. https://commons.wikimedia.org/w/index.php?title=File:Linear_regression.svg&oldid=343115066. Accessed 26 November 2019 20:16 UTC
Xu Yuming et al (2013) A DAG scheduling scheme on heterogeneous computing systems using double molecular structure-based chemical reaction optimization. J Parallel Distrib Comput 73(9):1306–1322
Xu Yuming et al (2014) A hybrid chemical reaction optimization scheme for task scheduling on heterogeneous computing systems. IEEE Trans Parallel and Distributed Syst 26(12):3208–3222
Yan X, Su X (2009) Linear regression analysis: theory and computing. World Scientific
Yang X-S (2010a) Firefly algorithm, stochastic test functions and design optimisation arXiv preprint https://arxiv.org/abs/1003.1409
Yang X-S (2010b) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp 65–74
Yang X-S, Deb S (2009) Cuckoo search via Lévy flights. In: 2009 world congress on nature & biologically inspired computing (NaBIC), IEEE, pp 210–214
Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster IEEE Transactions on. Evol Comput 3:82–102
Zhang J, Zhuang J, Du H (2009) Self-organizing genetic algorithm based tuning of PID controllers. Inf Sci 179:1007–1018