Information Sciences 324 (2015) 217–246

Contents lists available at ScienceDirect

Information Sciences

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

# Heat transfer search (HTS): a novel optimization algorithm

Vivek K. Patel, Vimal J. Savsani*

*Department of Mechanical Engineering, Pandit Deendayal Petroleum University, Gujarat, India*

ARTICLE INFO

*Article history:*
Received 11 November 2014
Revised 17 June 2015
Accepted 28 June 2015
Available online 3 July 2015

*Keywords:*
Heat transfer search (HTS) algorithm
Optimization
Conduction
Convection
Radiation

ABSTRACT

In this paper, a new metaheuristic optimization algorithm based on the law of thermodynamics and heat transfer is introduced. In the proposed algorithm, the search agents are molecules of the system that interact with each other as well as with the surrounding to attain thermal equilibrium state. The interactions of molecules are through various modes of heat transfer (i.e. conduction, convection and radiation). The performance of the proposed algorithm is investigated by implementing it for the parameter optimization of 24 well defined constrained optimization problems of Congress on Evolutionary Computation 2006 (CEC 2006). The results obtained using the proposed algorithm are compared with the results of some well-known metaheuristic search algorithms available in the literature. The statistical analysis of the experimental work has been carried out by conducting Friedman's rank test and Holm post hoc procedure. The comparative results validate the competitive and efficient performance of the proposed algorithm for solving constraint optimization problems.


## 1. Introduction

A majority of the real-life optimization problems in the area of engineering, science, economics, etc. involve different types of constraints. Moreover, these problems are of different characteristics such as linear, nonlinear, quadratic, polynomial, cubic, etc. The classical derivative-based optimization techniques often fail to solve such type of problems. Thus, the difficulties associated with these types of real-life optimization problems motivate to develop alternative and effective methods to solve it.

In recent years, various metaheuristic optimization algorithms have been developed to solve a wide variety of real life problems. All these algorithms are nature inspired and simulate some principle of biology, physics, ethology or swarm intelligence [4]. Many such algorithms are developed during last 2 decades, and few well-known algorithms are listed as follows. Genetic algorithm (GA) is inspired by the Darwinian evolutionary theory of the living beings [15]. Differential evolution (DE) algorithm is similar to GA with specialized crossover and selection method [53]. Artificial immune algorithm (AIA) is inspired by the principle and mechanism of the immune system of living beings [12]. Bacteria foraging optimization (BFO) mimics the social foraging behaviour of *Escherichia coli* [39]. Particle swarm optimization (PSO) simulates the intelligent behaviour of a flock of migrating birds in search of their destination [24]. Ant colony optimization (ACO) works on the principle of foraging behaviour of the ant for the food [10]. Artificial bee colony (ABC) algorithm mimics the foraging behaviour and information sharing ability of honey bee swarm [21,23]. Biogeography based optimization (BBO) simulate the principle of immigration and emigration of the species from one place to the other [51,52]. Gravitational search algorithm (GSA) works on the principle of the law of gravity and mass interaction [50]. Teaching–learning based optimization (TLBO) algorithm mimics the teaching–learning ability of teacher and learners in a class room [45]. Evolutionary membrane algorithm simulates the structure and functioning of a biological living cell [14].

* Corresponding author. Tel.: +91 9825092139, fax.: +917923275484.
*E-mail address:* vimal.savsani@gmail.com, vimal.s@sot.pdpu.ac.in (V.J. Savsani).

http://dx.doi.org/10.1016/j.ins.2015.06.044
0020-0255/

All these algorithms are population-based algorithms where a group of solutions carry out the search process. The search characteristics of different algorithms are based on various natural phenomena as explained above. Researchers reported the successful application of various algorithms for a wide variety of real-life applications [2,11,17,29,30,34,40,56,59]. However, the success of any metaheuristic optimization algorithms depends on how the algorithm balances the exploration and exploitation of the search process [9]. Exploration is the process of abrupt movement in the search space to cover it entirely while exploitation is the process to refine certain areas of explored search space. Pure exploration enhances the capacity of any algorithm to produce new solutions with less precision. Conversely, pure exploitation increases the possibility of trapping at the local optimum solution during the search process. Therefore, a proper balance between exploration and exploitation is always required for a better performance of any search algorithm [6,33].

In most of the metaheuristic algorithms, the trade-off between exploration and exploitation is carried out by proper setting of control parameters of the algorithm. Furthermore, to balance this characteristic and to enhance the performance of the algorithm, modifications in the existing algorithms are reported by various researchers [1,13,18,25,38,41,43,58]. Similarly, hybridization of two or more algorithms also enhances the performance of basic algorithms. Work related to hybridization of algorithms and their performance improvements are also reported by many researchers [3,26,31,36,44,55]. Despite all these investigations, till date, proper trade-off between exploration and exploitation is an unsolved and challenging issue in metaheuristic optimization algorithms. In this paper, efforts have been put to resolve this issue to a certain extent in the proposed algorithm.

The contribution of the present paper is to propose a novel metaheuristic algorithm based on the natural law of thermodynamics and heat transfer, named as Heat Transfer Search (HTS) algorithm. This algorithm is developed to incorporate proper trade-off between exploration and exploitation. The search process of HTS considers three parts namely ‘conduction phase’, ‘convection phase’ and ‘radiation phase’. The HTS algorithm replicates the thermal equilibrium behaviour of any system. The thermal equilibrium can be achieved when molecules of the system transfer heat in the form of conduction, convection and radiation. Each phase of the proposed algorithm is executed with equal probability during an entire search process. The search processes of all three phases are premeditated in such a manner that during the first half (i.e. early generations) each phase explores the search space, while in the second half (i.e. after lapse of predefined number of generations) each phase exploits the search space. In this way, the proposed algorithm balances the exploration and exploitation of an entire search process.

The remainder of this paper is organized as follows. Section 2 describes the thermal equilibrium behaviour of the system and associated heat transfer phenomena. Section 3 explains the proposed HTS algorithm and its characteristics. Section 4 presents experimental investigations of the proposed algorithm and its comparative study with other metaheuristic algorithms. Section 5 deals with the statistical analysis of the comparative results. Finally, the conclusion of the present work is discussed in Section 6.

## 2. Thermal equilibrium and heat transfer

Heat Transfer Search (HTS) algorithm is inspired by the natural law of thermodynamics which states that “Any system always try to achieve equilibrium state with its surroundings” [7,8]. The word ‘equilibrium’ is treated as thermal equilibrium, mechanical equilibrium and chemical equilibrium depending upon the system. The HTS algorithm simulates the thermal equilibrium behaviour of the system. If any system at higher/lower temperature interacts with the surrounding maintained at lower/higher temperature, the system always try to achieve energy level (i.e. temperature level) of the surrounding by heat transfer to establish thermal equilibrium [7,8]. Similarly, if a thermal imbalance exists within any system, it tries to establish thermal equilibrium by the heat transfer within the system itself. A system remains in a stable state if the thermal equilibrium exists within the system as well as between the system and surroundings [7,8].

Fig. 1 shows the model of the thermal equilibrium behaviour of the system. Consider Fig. 1( A), where thermal equilibrium exists within the system as all molecules of the system are at the same temperature (53 °C). But, there is a thermal imbalance between the system and the surrounding as it is maintained at a different temperature (25 °C) than the system. In this situation, the system tries to establish thermal equilibrium with the surrounding by heat transfer. So, after a particular course of time thermal equilibrium exists between the system and the surrounding. Likewise, consider Fig. 1( B) in which thermal imbalance exists within the system (as all the molecules of the system are at different temperature) as well as between system and surrounding (as surrounding is maintained at a different temperature than the system). So, in this situation the system tries to establish a thermal equilibrium by heat transfer within the system and with the surrounding. Similarly, consider Fig. 1( C), where a system is shown without considering the surrounding. Here, a thermal imbalance exists within the system, and it tries to establish thermal equilibrium by conducting intermolecular heat transfer.

It can be observed from the above understanding that the system tries to neutralize thermal imbalance by the process of heat transfer in any conditions. The modes of heat transfer, such as conduction, convection and radiation can take place within the system or between the system and the surrounding if any thermal imbalance exists. Temperature difference (within the system or between the system and the surrounding) is the driving potential for initiating any mode of heat transfer.

Conduction is the mode of heat transfer where energy is transferred from the higher energetic particles of the system to the adjacent lower energetic ones due to interactions between them. Conduction can also take place between the system and the surrounding when both are in direct physical contact with each other. Conduction heat transfer is governed by Fourier’s law of heat conduction and expressed as [7],

$$Q_{\mathrm{cond}} = -kA\frac{dT}{dx} = -\frac{dT}{dx/kA} = -\frac{1}{\mathrm{Resistance}}{}^{*}dT = -\mathrm{Conductance}^{*}dT \tag{1}$$

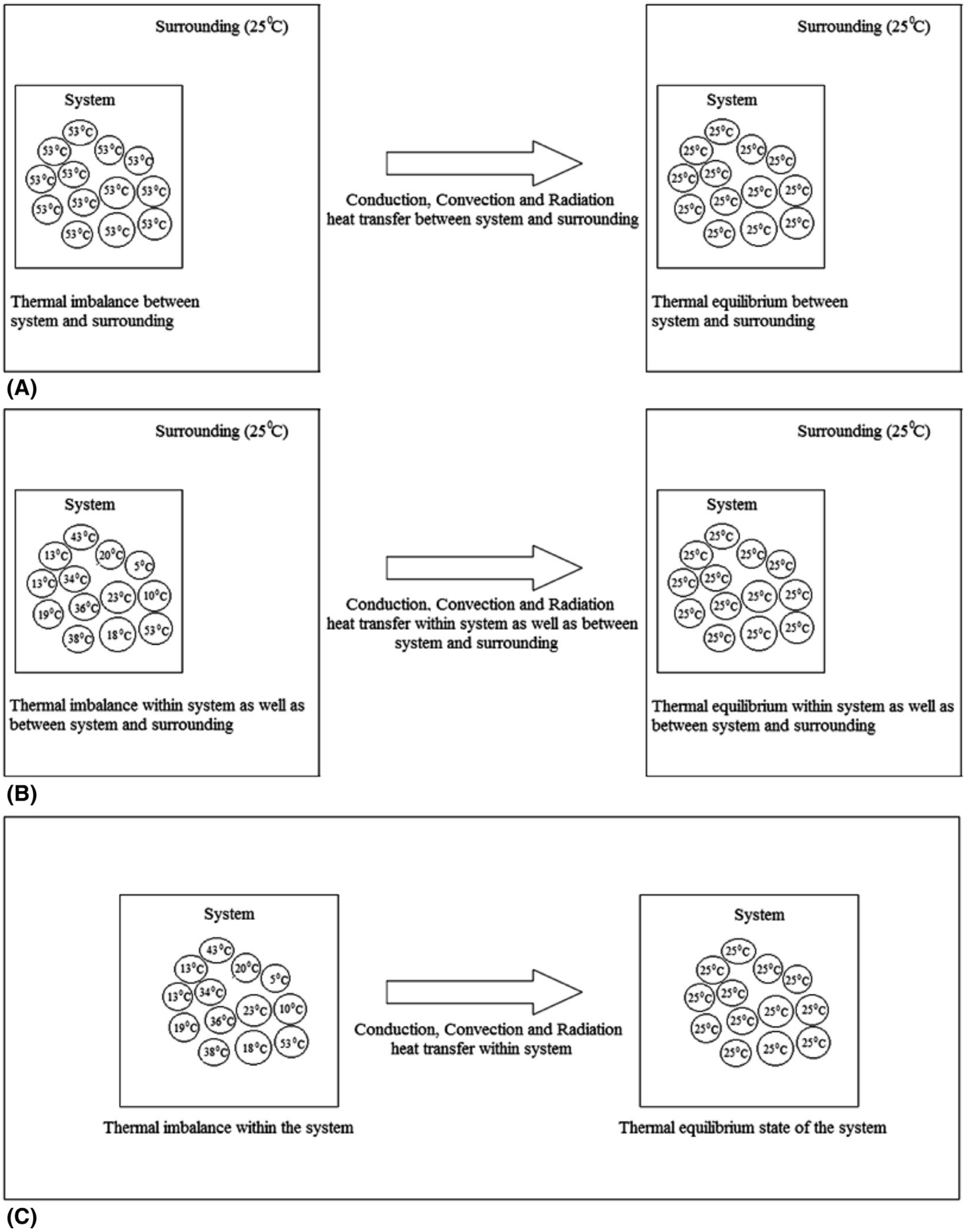


**Fig. 1.** Model of thermal equilibrium behaviour of the system.

where, $k$ is the thermal conductivity of the system, which is a measure of the ability of a system to conduct heat, $A$ is the heat transfer area, $dT$ is the temperature difference and $dT/dx$ is the temperature gradient. The thermal conductivity of a system is a function of temperature, and its value changes during heat transfer.

Convection is the mode of energy transfer between a system and the adjacent fluid in motion, and it involves the combined effects of conduction and fluid motion. The rate of convection heat transfer is observed to be proportional to the temperature difference and is conveniently expressed by Newton's law of cooling as [7],

$$Q_{\mathrm{conv}} = hA(T_{\mathrm{ms}} - T_{\mathrm{surr}}) \tag{2a}$$

To correlate the Newton's law of cooling with the optimization equation used for convection phase in Section 3, the Newton's law of cooling is simplified as below,

$$Q_{\mathrm{conv}} = hA(T_{\mathrm{ms}} - T_{\mathrm{surr}}) = -hA(T_{\mathrm{surr}} - T_{\mathrm{ms}}) = \text{Convection element}(T_{\mathrm{surr}} - T_{\mathrm{ms}}) \tag{2b}$$

Where, $h$ is the convection heat transfer coefficient, $A$ is the heat transfer area, $T_{\mathrm{ms}}$ is the mean temperature of the system, and $T_{\mathrm{surr}}$ is the temperature of the fluid (i.e. surrounding). Convection heat transfer coefficient depends on various parameters like nature of the fluid motion, properties of the fluid, bulk fluid velocity, etc. So, its value is variable during heat transfer.

Radiation is the energy emitted by the system in the form of electromagnetic waves due to its temperature level. All bodies at a temperature above absolute zero emit thermal radiations. The rate of radiation heat transfer depends on the absolute temperature level and is conveniently expressed by Stefan–Boltzmann law as [7],

$$Q_{\mathrm{rad}} = \varepsilon\sigma A\left(T_s^4 - T_{\mathrm{surr}}^4\right) \tag{3a}$$

Where, $\sigma$ is the Stefan–Boltzmann constant, $\varepsilon$ is the emissivity of the system, $A$ is the heat transfer area, $T_{\mathrm{s}}$ is the system temperature, and $T_{\mathrm{surr}}$ is the temperature of the surrounding. To correlate the Stefan–Boltzmann equation with the optimization equation used for radiation phase in Section 3, it is simplified as below,

$$\begin{aligned} Q_{\mathrm{rad}} &= \varepsilon\sigma A(T_{\mathrm{s}}^4 - T_{\mathrm{surr}}^4) \\ &= \varepsilon\sigma A(T_{\mathrm{s}}^2 + T_{\mathrm{surr}}^2)(T_{\mathrm{s}}^2 - T_{\mathrm{surr}}^2) \\ &= \varepsilon\sigma A(T_{\mathrm{s}}^2 + T_{\mathrm{surr}}^2)(T_{\mathrm{s}} + T_{\mathrm{surr}})(T_{\mathrm{s}} - T_{\mathrm{surr}}) \\ &= \text{Radiation element}(T_{\mathrm{s}} - T_{\mathrm{surr}}) \end{aligned} \tag{3b}$$

Thus, the system changes its energy level by the above mentioned heat transfer phenomena within the system or between the system and the surrounding and neutralizes any existing thermal imbalance. In this way, the system achieves a stable state.

Next section describes the proposed heat transfer search (HTS) algorithm.

## 3. Heat transfer search (HTS) algorithm

The HTS algorithm simulates the course of action followed by the system to attain thermal equilibrium. Like other nature-inspired algorithms, HTS is also a population-based method, which uses a set of solutions to attain the global optimum solution. A thermal imbalance exists between the system and the surrounding or within the system itself. Any system always tries to reduce this thermal imbalance to attain a thermal equilibrium state. Similarly, in the HTS algorithm during optimization, if the difference in the solution (i.e. objective function difference) exists within the population, solution tries to improve its value. This improvement can be done by considering the difference between the current solution and either of the best solution, other random solution from the population or the mean value of solution from the population. In HTS, a population is analogous to the clusters of molecules that take part in heat transfer process. The clusters of molecules possess different temperature levels. In HTS, different design variables represent a different temperature of molecules and energy level of the molecules indicates the objective function value of the problem, i.e. fitness value of the objective function. The best solution is treated as the surrounding and rest of the solutions are considered as a system.

The search process of HTS considers three parts namely, 'conduction phase', 'radiation phase' and 'convection phase'. The 'conduction phase', 'radiation phase' and 'convection phase' neutralizes thermal imbalance (i.e. change the energy level) of the system by conduction, radiation and convection heat transfer respectively. In, HTS algorithm all three modes of heat transfer take place with equal probability. The equal probability is controlled by the parameter '$R$' in each generation, which is a uniformly distributed random number, varies between 0 and 1. As the value of $R$ varies between 0 and 1, so for equal probability, each phase must share the equal portion of $R$ (i.e. as there are three phases, three equal portions are 0–0.3333, 0.3333–0.6666 and 0.6666–1) during the course of optimization. Depending upon the value of $R$, any one of the three phases can be executed to update the solutions in that generation. Experimentations are required to decide which portion of $R$ will execute any of the phases as mentioned above. The detailed explanation is given in the description of different phases in the section follows.

The HTS algorithm starts with a randomly generated initial population of $n$ solutions, where $n$ denotes the size of the population. Each solution is an $m$-dimensional vector. Here, $m$ is the number of optimization parameters (i.e. design variables). After initialization, the population is updated in each generation $g$ ($g = 1, 2, \ldots, G_{\max}$) by following the search process of conduction, convection or radiation phase. Moreover, the updated solution in HTS algorithm is accepted if it produces better function value, so the greedy selection procedure is employed in the HTS algorithm. After the greedy selection, worst solutions of the population are replaced by the elite solution and finally if any duplicate solutions exist; it is replaced by a randomly generated solution. The working of all three phases is explained below to minimize a function $f(X)$, which denotes objective function for an optimization problem.

### 3.1. Conduction phase

In this part of the algorithm, the system tries to neutralize the thermal imbalance by conduction heat transfer. So, higher energetic molecules transfer their energy to lower energetic molecules. Let, the number of molecules of the system (i.e. population) is $n$, and the different temperature level of the molecules (i.e. design variables) is $m$. During the first part of the conduction phase where generation $g \le G_{max}/\mathrm{CDF}$ (where CDF is the conduction factor), the solutions are updated as below.

$$X^{new}_{j,i} = X^{old}_{k,i} + \mathrm{CDS}_1 \quad \text{If } f\left(X_j\right) > f(X_k) \tag{4}$$

$$X^{new}_{k,i} = X^{old}_{j,i} + \mathrm{CDS}_2 \quad \text{If } f(X_k) > f\left(X_j\right) \tag{5}$$

Where, $j = 1,2,\ldots,n$, $j \neq k$, $k \in (1,2,\ldots,n)$ and $k$ is a randomly selected solution from the population, $i \in (1,2,\ldots,m)$ and $i$ is a randomly selected design variable. Moreover, only one dimension (i.e. design variable) of each solution in each generation is updated in the conduction phase (See Appendix A). $\mathrm{CDS}_1$ and $\mathrm{CDS}_2$ are the conduction steps which are given by,

$$\mathrm{CDS}_1 = -R^2 X^{old}_{k,i} \tag{6}$$

$$\mathrm{CDS}_2 = -R^2 X^{old}_{j,i} \tag{7}$$

Where, $R$ is the variable equal to the value of the probability for the selection of conduction phase. As all three types of heat transfer take place with the equal probability, the value of $R$ can lie between 0 and 0.3333, 0.3333 and 0.6666 or 0.6666 and 1. After conducting many experiments the range, 0–0.3333 is set to the value of $R$ to execute conduction phase.

In Eqs. (6) and (7), $R^2$ is corresponding to the conductance of the Fourier's equation (i.e. Eq. (1)) and $X_{k,i}$ and $X_{j,i}$ are corresponding to the temperature gradient of Eq. (1). The conductance of any system depends on the thermal conductivity (Eq. (1)), and thermal conductivity of any system depends on the temperature. During the heat transfer, the temperature of the system is continuously changing and hence its thermal conductivity and conductance are also changing continuously. Thus, to reflect this temperature dependent behaviour of conductance, it is modelled by variable '$R$' which can attain any value between 0 and 0.3333 at the beginning of each generation in the conduction phase. Moreover, to exploit the search space, this random variable is modelled by squaring its value so that it can follow a fine search.

In the second part of the conduction phase where generation $g \ge G_{max}/\mathrm{CDF}$, the solutions are updated as,

$$X^{new}_{j,i} = X^{old}_{k,i} + \mathrm{CDS}_3 \quad \text{If } f(X_j) > f(X_k) \tag{8}$$

$$X^{new}_{k,i} = X^{old}_{j,i} + \mathrm{CDS}_4 \quad \text{If } f(X_k) > f(X_j) \tag{9}$$

where, $\mathrm{CDS}_3$ and $\mathrm{CDS}_4$ are the conduction steps given by,

$$\mathrm{CDS}_3 = -r_i X^{old}_{k,i} \tag{10}$$

$$\mathrm{CDS}_4 = -r_i X^{old}_{j,i} \tag{11}$$

Where, $r_i$ is a random number in the range [0, 1]. In Eqs. (10) and (11), $r_i$ corresponds to the conductance of the Fourier's equation (i.e. Eq. (1)) and $X_{k,i}$ and $X_{j,i}$ corresponds to the temperature gradient of Eq. (1). As, the value of $r_i$ varies between 0 and 1, it will explore the search space. The conduction factor (CDF) decides the exploration and exploitation tendency of the conduction phase. After conducting many trials, the value of CDF is set as 2 for the conduction phase.

### 3.2. Convection phase

In this part of the algorithm, the system tries to neutralize the thermal imbalance by the convection heat transfer. In this phase, the mean temperature of the system interacts with the surrounding temperature to establish a thermal balance between the system and the surrounding. Surrounding is treated as the best solution in the proposed approach. Let, at any iteration $g$ (where $g < G_{max}/\mathrm{COF}$, and COF is a convection factor), $X_s$ be the temperature of the surrounding and $X_{ms}$ be the mean temperature of the system. The energy possess by the system is higher than the surrounding (i.e. $f(X_s) < f(X_{ms})$). Hence, the solution is updated (i.e. molecules of the system try to reduce their energy) according to the following equation,

$$X^{new}_{j,i} = X^{old}_{j,i} + \mathrm{COS} \tag{12}$$

Where, $j = 1,2,\ldots,n$, $i = 1,2,\ldots,m$. Moreover, each design variable of the population is updated in the convection phase (See Appendix A). COS is the convection step and given by,

$$\mathrm{COS} = R(X_s - X_{ms} * \mathrm{TCF}) \tag{13}$$

where, $R$ is the variable equal to the value of the probability for the selection of convection phase. After conducting many experiments, the range of 0.6666–1 is set to the value of $R$ in this phase. In Eq. (13), $R$ corresponds to the convection element of the Newton law of cooling (i.e. Eq. (2b)). $X_s$ and $X_{ms}$ correspond to the temperature of the surrounding ($T_{surr}$) and the mean temperature of the system ($T_{ms}$). The temperature of the system changes during the heat transfer. The surrounding is treated as a heat sink or heat source, so the temperature of the surrounding remains constant. To take into account this effect, temperature change factor (TCF) is introduced. Thus, TCF is the temperature change factor based on which the mean temperature of the system can be changed. The value of TCF is randomly decided based on the following equations,

$$\mathrm{TCF} = \mathrm{abs}(R - r_i) \quad \text{If } g \le G_{max}/\mathrm{COF} \tag{14}$$

$$\mathrm{TCF} = \mathrm{round}(1 + r_i) \quad \mathrm{If}\, g \geq G_{\max}/\mathrm{COF} \tag{15}$$

Where, $r_i$ is a random number in the range [0, 1]. The value of TCF changes randomly between 0 and 1 in the first part of the convection phase (Eq. (14)). In the second part of the convection phase, value of TCF changes either as 1 or 2 (Eq. (15)). The different value of TCF in the proposed algorithm is to balance the exploration and exploitation. COF is the convection factor, and it decides the tendency of exploration and exploitation of convection phase. After conducting many trials, the value of COF is set as 10 for this phase.

### 3.3. Radiation phase

In this phase of the algorithm, the system tries to neutralize the thermal imbalance by radiation heat transfer. Here, the system interacts with the surrounding (i.e. best solution) or within the system (i.e. other solution) to establish a thermal balance. In the first part of the radiation phase, where generation $g \leq G_{\max}/\mathrm{RDF}$ (where RDF is the radiation factor), the solution updated (i.e. energy reduction of the system) is explained below.

$$X_{j,i}^{new} = X_{j,i}^{old} + \mathrm{RDS}_1 \quad \mathrm{If}\, f\left(X_j\right) > f(X_k) \tag{16}$$

$$X_{j,i}^{new} = X_{j,i}^{old} + \mathrm{RDS}_2 \quad \mathrm{If}\, f(X_k) > f\left(X_j\right) \tag{17}$$

Where, $j = 1,2,\ldots,n, j \neq k, k \in (1,2,\ldots,n)$ and $k$ is a randomly selected solution from the population, $i \in (1,2,\ldots,m)$. Moreover, all design variables of the solution is updated in each generation of the radiation phase (See Appendix A). $\mathrm{RDS}_1$ and $\mathrm{RDS}_2$ are radiation step that is given by,

$$\mathrm{RDS}_1 = R\left(X_{k,i}^{old} - X_{j,i}^{old}\right) \tag{18}$$

$$\mathrm{RDS}_2 = R\left(X_{j,i}^{old} - X_{k,i}^{old}\right) \tag{19}$$

Where, $R$ is the variable equal to the value of the probability for the selection of radiation phase. The range left for radiation phase is 0.3333–0.6666. In Eqs. (18) and (19), $R$ corresponds to the radiation element of the Stefan–Boltzmann law (Eq. (3b)) and $X_k$ and $X_j$ correspond to the system and the surrounding temperature.

In the second part of radiation phase where generation $g \geq G_{\max}/\mathrm{RDF}$, the solution is updated as,

$$X_{j,i}^{new} = X_{j,i}^{old} + \mathrm{RDS}_3 \quad \mathrm{If}\, f(X_j) > f(X_k) \tag{20}$$

$$X_{j,i}^{new} = X_{j,i}^{old} + \mathrm{RDS}_4 \quad \mathrm{If}\, f(X_k) > f(X_j) \tag{21}$$

Where, $\mathrm{RDS}_3$ and $\mathrm{RDS}_4$ are radiation steps and given by,

$$\mathrm{RDS}_3 = r_i\left(X_{k,i}^{old} - X_{j,i}^{old}\right) \tag{22}$$

$$\mathrm{RDS}_4 = r_i\left(X_{j,i}^{old} - X_{k,i}^{old}\right) \tag{23}$$

Where, $r_i$ is a random number in the range [0, 1]. RDF is the radiation factor that decides the exploration and exploitation tendency in this phase. After conducting many trials, the value of RDF is set as 2 for radiation phase.

The search mechanism of the HTS algorithm is composed of three phases. Moreover, each of these phases is sub-divided into two parts, which is controlled by a number of generations (i.e. number of iterations), and it depends on conduction, convection and radiation factor. Thus, depending on the value of these factors, large or small change in the value of design variables can take place in the first or second part of each phase. The large change in design variables corresponds to the exploration of a search space while a small change in design variable corresponds to the exploitation of a search space. Thus, the proper trade-off between exploration and exploitation is carried out in proposed algorithm by incorporating different search mechanisms for exploration and exploitation.

Fig. 2 shows the detailed flow chart of the proposed algorithm. The analogy between the attainments of thermal equilibrium of any system by heat transfer and the course of optimisation is shown in Fig. 3. Moreover, for the deep insight of the proposed algorithm, a stepwise calculation of one constrained optimization problem is given in Appendix A.

The next section describes the experimental investigation of the proposed algorithm.

## 4. Experimental investigation

In this section, the ability of the HTS algorithm is assessed by applying it to optimization of 24 well defined constrained benchmark problems of Congress on Evolutionary Computation 2006 (CEC 2006) [20,22,32]. An objective function and constraints of all benchmark problem of CEC 2006 are of different characteristics such as linear, nonlinear, quadratic, polynomial and cubic. The number of design variables and its ranges are also different for each problem. Similarly, the number of constraints and its type are also different for all problems. The optimization problems of real life applications may consist of any of the above attributes.

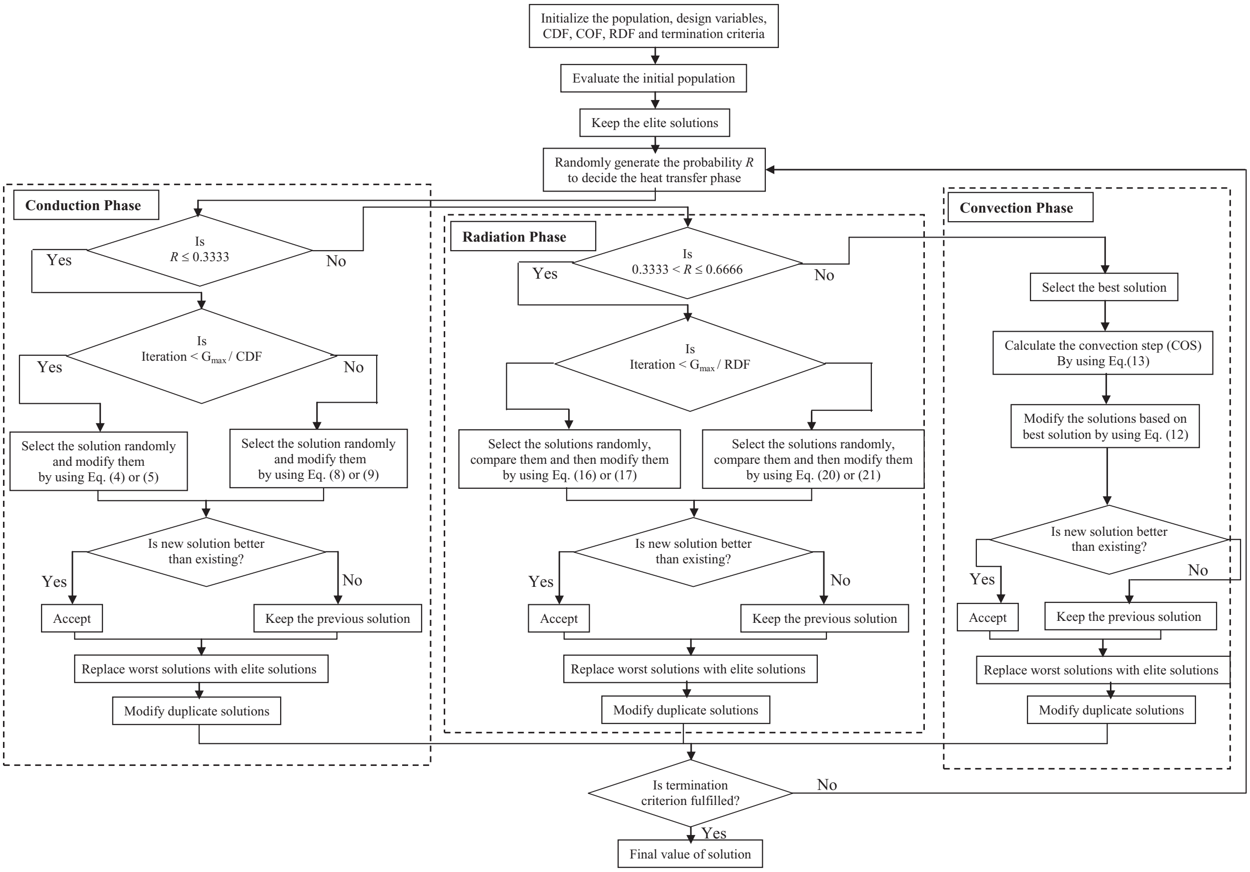


**Fig. 2.** Flow chart of heat transfer search (HTS) algorithm.

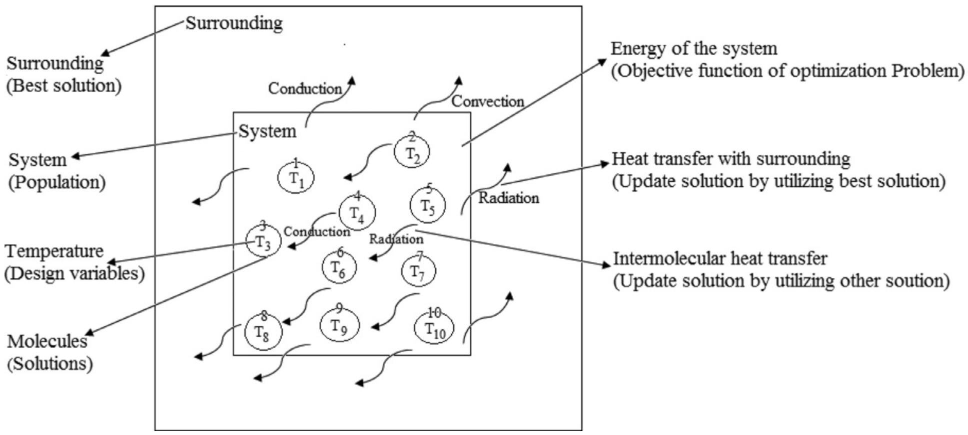


**Fig. 3.** Analogy between attainment of thermal equilibrium and course of optimization.

Moreover, well-known optimization algorithms like GA, PSO, DE, ABC, TLBO, etc., were experimented on the same functions previously and these functions offer challenges to an optimization algorithm. All these properties inspired us to consider CEC 2006 functions for the experimentation. The detailed mathematical formulation of each function and its characteristics are given in Appendix B.

To evaluate the performance of HTS algorithm, results obtained using the HTS algorithm is compared with the results of other well-known optimization algorithms (such as GA, HM, ASCHEA, SMES, PSO, BBO, DE and ABC) available in the literature [20,22]. However, the above-mentioned algorithms are experimented on 13 functions only out of 24. Hence, to make the comparison more meaningful and to get the deep insight on the performance of algorithms, optimization algorithms

**Table 1**
Controlled parameters of comparative algorithms.

| |
|---|
| ***PSO algorithm*** |
| Population size: 50, |
| Inertia weight: 0.6 |
| Cognitive parameter: 1.65 |
| Social parameter: 2 |
| ***ABC algorithm*** |
| Number of employed bees: 25 |
| Number of onlooker bees: 25 |
| Limit: number of generations |
| ***BBO algorithm*** |
| Population size: 50 |
| Immigration rate:1 |
| Emigration rate:1 |
| Mutation factor: 0.01 |
| ***DE algorithm*** |
| Population size: 50 |
| Crossover factor: 0.5 |
| Constant factor: 0.5 |
| ***TLBO algorithm*** |
| Population size: 50 |
| ***HTS algorithm*** |
| Population size: 50 |
| Conduction factor: 2 |
| Radiation factor: 2 |
| Convection factor: 10 |

like PSO, BBO, DE, ABC and TLBO are experimented along with HTS on the remaining benchmark problems of CEC 2006. Computation code of all these algorithms (i.e. PSO, BBO, DE, ABC, and TLBO) are taken from the different websites dedicated to these algorithms, where the developers of these algorithms have provided the code. [ABC: http://mf.erciyes.edu.tr/abc/index.htm, BBO: http://embeddedlab.csuohio.edu/BBO, DE: http://www1.icsi.berkeley.edu/~storn/code.html, TLBO: https://sites.google.com/site/tlborao/tlbo-code].

The competitor algorithms (i.e. PSO, BBO, DE, ABC, and TLBO) considered in the present work were experimented on the various constraint optimization problems previously by different researchers [5,20,22,27,35,37,42,46,48,57]. It is observed from their results that these algorithms have produced promising results for the constraint optimization problems. Moreover, during the literature survey it is observed that the considered algorithms were successfully applied to a wide variety of engineering applications and have produced promising results [28,42,46,47,49,54,60]. Based on this history, these algorithms are selected as competitor algorithms for our proposed approach.

### 4.1. Experimental setting

To maintain the consistency in the comparison of competitive algorithms, a common experimental platform is required. In the present work, the common platform is provided by setting a maximum number of function evaluations as 240,000 for each test function. To decide the controlled parameters of each of the competitive algorithm, a number of trials of each algorithm are conducted considering different strategies with a different combination of controlled parameters. Moreover, the research papers are also referred [20,22], where these algorithms (i.e. GA, PSO, ABC, DE, etc.) are experimented on the same functions to get deep insight about the tuning of control parameters for these algorithms. Based on results of trails, control parameters are set for each algorithm for the experimentation that is shown in Table 1. Moreover, to maintain the consistency related to constraint handling techniques, all competitive algorithms are experimented with the 'Static Penalty' method as a constraint handling technique. The HTS algorithm is executed 100 times for each test function and the average results obtained are compared with the other algorithms for same number of runs.

### 4.2. Results and discussion

The comparative result of each competitive algorithm for G01 to G13 benchmark functions are presented in Table 2 in the form of the best solution, worst solution and mean solution. Results of other algorithms are taken from the literature [20,22] except HTS, BBO, and TLBO. These algorithms were experimented on the same functions with a similar experimental platform.

For the G01 test function, HTS, SMES and ABC algorithm have produced better results than the rest of the algorithms. Each of these algorithms has attained global optimum value during every run of the experiment. Thus, HTS, SMES and ABC algorithms have outperformed other comparative algorithms for G01 function in every aspect. The SMES algorithm has produced better

**Table 2**
Comparative results of G01–G13 test functions obtained by different algorithms for 240,000 function evaluations averaged over 100 runs. Result in boldface indicated better performing algorithm. (–) indicate results are not available. NF: means that no feasible solutions were found.

| Function | | HM[a] [20] | ASCHEA[b] [20] | SMES[c] [20] | GA[c] [20] | PSO[c] [20] | DE[c] [20] | ABC[c] [20] | BBO[b] | TLBO[b] | HTS[b] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| G01 | Best | −14.7864 | −15.000 | −15 | 14.44 | −15 | −15 | −15 | −14.977 | −15 | −15 |
| (−15.00) | Worst | ––– | ––– | ––– | ––– | −13 | −11.828 | −15 | −14.5882 | −6 | −15 |
| | Mean | −14.7082 | −14.84 | **−15** | −14.236 | −14.71 | −14.555 | **−15** | −14.7698 | −10.782 | **−15** |
| G02 | Best | −0.79953 | −0.785 | −0.803601 | −0.796231 | −0.669158 | −0.472 | −0.803598 | −0.7821 | −0.7835 | −0.7517 |
| (−0.803619) | Worst | ––– | ––– | ––– | ––– | −0.299426 | −0.472 | −0.749797 | −0.7389 | −0.5518 | −0.5482 |
| | Mean | −0.79671 | −0.59 | −0.785238 | −0.788588 | −0.41996 | −0.665 | **−0.792412** | −0.7642 | −0.6705 | −0.6437 |
| G03 | Best | 0.9997 | 1 | 1 | 0.99 | −1 | −0.99393 | −1 | −1.0005 | −1.0005 | −1.0005 |
| (−1.0005) | Worst | ––– | ––– | ––– | ––– | −0.464 | −1 | −1 | −0.0455 | 0 | 0 |
| | Mean | 0.9989 | 0.99989 | **1** | 0.976 | 0.764813 | **−1** | **−1** | −0.3957 | −0.8 | −0.9004 |
| G04 | Best | −30664.5 | −30665.5 | −30665.539 | −30626.053 | −30665.539 | −30665.539 | −30665.539 | −30665.539 | −30665.5387 | −30665.5387 |
| (−30665.539) | Worst | ––– | ––– | ––– | ––– | −30665.539 | −30665.539 | −30665.539 | −29942.3 | −30665.5387 | −30665.5387 |
| | Mean | −30655.3 | **−30665.5** | **−30665.539** | −30590.455 | **−30665.539** | **−30665.539** | **−30665.539** | −30411.865 | **−30665.5387** | **−30665.5387** |
| G05 | Best | NF | 5126.5 | 5126.599 | NF | 5126.484 | 5126.484 | 5126.484 | 5134.2749 | 5126.486 | 5126.486 |
| (5126.486) | Worst | NF | ––– | ––– | NF | 5249.825 | 5534.61 | 5438.387 | 7899.2756 | 5127.714 | 5126.6831 |
| | Mean | NF | 5141.65 | 5174.492 | NF | 5135.973 | 5264.27 | 5185.714 | 6130.5289 | 5126.6184 | **5126.5152** |
| G06 | Best | −6952.1 | −6961.81 | −6961.814 | −6952.472 | −6961.814 | −6954.434 | −6961.814 | −6961.8139 | −6961.814 | −6961.814 |
| (−6961.814) | Worst | ––– | ––– | ––– | ––– | −6961.814 | −6954.434 | −6961.805 | −5404.4941 | −6961.814 | −6961.814 |
| | Mean | −6342.6 | **−6961.81** | −6961.284 | −6872.204 | **−6961.814** | −6954.434 | **−6961.813** | −6181.7461 | **−6961.814** | **−6961.814** |
| G07 | Best | 24.62 | 24.3323 | 24.327 | 31.097 | 24.37 | 24.306 | 24.33 | 25.6645 | 24.3101 | 24.3104 |
| (24.3062) | Worst | ––– | ––– | ––– | ––– | 56.055 | 24.33 | 25.19 | 37.6912 | 27.6106 | 25.0083 |
| | Mean | 24.826 | 24.6636 | 24.475 | 34.98 | 32.407 | **24.31** | 24.473 | 29.829 | 24.837 | 24.4945 |
| G08 | Best | 0.095825 | 0.095825 | 0.095825 | 0.095825 | −0.095825 | −0.095825 | −0.095825 | −0.095825 | −0.095825 | −0.095825 |
| (−0.095825) | Worst | ––– | ––– | ––– | ––– | −0.095825 | −0.095825 | −0.095825 | −0.095817 | −0.095825 | −0.095825 |
| | Mean | 0.0891568 | **0.095825** | **0.095825** | 0.095799 | **-0.095825** | **-0.095825** | **-0.095825** | -0.095824 | **-0.095825** | **-0.095825** |
| G09 | Best | 680.91 | 680.63 | 680.632 | 685.994 | 680.63 | 680.63 | 680.634 | 680.6301 | 680.6301 | 680.6301 |
| (680.6301) | Worst | ––– | ––– | ––– | ––– | 680.631 | 680.631 | 680.653 | 721.0795 | 680.6456 | 680.644 |
| | Mean | 681.16 | 680.641 | 680.643 | 692.064 | **680.63** | **680.63** | **680.64** | 692.7162 | 680.6336 | **680.6329** |
| G10 | Best | 7147.9 | 7061.13 | 7051.903 | 9079.77 | 7049.481 | 7049.548 | 7053.904 | 7679.0681 | 7250.9704 | 7049.4836 |
| (7049.28) | Worst | ––– | ––– | ––– | ––– | 7894.812 | 9264.886 | 7604.132 | 9570.5714 | 7291.3779 | 7252.0546 |
| | Mean | 8163.6 | 7497.434 | 7253.047 | 10003.225 | 7205.5 | 7147.334 | 7224.407 | 8764.9864 | 7257.0927 | **7119.7015** |
| G11 | Best | 0.75 | 0.75 | 0.75 | 0.75 | 0.749 | 0.752 | 0.75 | 0.7499 | 0.7499 | 0.7499 |
| (0.7499) | Worst | ––– | ––– | ––– | ––– | 0.749 | 1 | 0.75 | 0.92895 | 0.7499 | 0.7499 |
| | Mean | **0.75** | **0.75** | **0.75** | **0.75** | **0.749** | 0.901 | **0.75** | 0.83057 | **0.7499** | **0.7499** |
| G12 | Best | ––– | ––– | −1 | −1 | −1 | −1 | −1 | −1 | −1 | −1 |
| (−1) | Worst | ––– | ––– | ––– | ––– | −0.994 | −1 | −1 | −1 | −1 | −1 |
| | Mean | ––– | ––– | **−1** | **−1** | −0.998875 | **−1** | **−1** | **−1** | **−1** | **−1** |
| | Best | ––– | ––– | 0.054986 | 0.134057 | 0.085655 | 0.385 | 0.76 | 0.62825 | 0.44015 | 0.37319 |
| G13 | Worst | ––– | ––– | ––– | ––– | 1.793361 | 0.99 | 1 | 1.45492 | 0.95605 | 0.79751 |
| (-0.05394) | Mean | ––– | ––– | **0.166385** | ––– | 0.569358 | 0.872 | 0.968 | 1.09289 | 0.69055 | 0.66948 |

The bold values indicate best result.
[a] Used decoder-based penalty method for constraint handling.
[b] Used static penalty method for constraint handling.
[c] Used Deb's method for constraint handling.

results compared to other algorithms for G02 function. However, all the algorithms failed to attain the global optimum value for this test function.

For the G03 test function, SMES and ABC algorithms have produced better results than other algorithms. However, the HTS algorithm has shown a competitive result for G03 test function, and it has achieved the global optimum value for this test function. Except GA, HM, and BBO algorithms, the rest of the other algorithms have obtained competitive performance on the G04 test function. The HTS and TLBO algorithms have produced competitive results on G05 test function. However, the mean solution obtained by HTS algorithm is better than TLBO algorithm. For the G06 test function, HTS, TLBO, ABC, PSO, and ASCHEA have produced equally good results.

HTS, ABC, DE and SMES algorithms have obtained better mean results than other comparative algorithms for the G07 test function. For G08 and G09 test function, except HM and GA all other algorithms have produced equally good results. The HTS algorithm has obtained better mean results than other comparative algorithms for the G10 test function. Thus, HTS algorithm has outperformed other algorithms for the G10 test function. Except BBO and DE algorithms, all other algorithms have produced competitive performance on the G11 test function. Likewise, except PSO algorithm rest of the other algorithms have produced equally good results on the G12 test function. SMES algorithm has obtained better mean results than other algorithms for the G13 test function. However, all the algorithms have failed to obtain a global optimum value for this test function.

The results of the different competitive algorithms available in the literature are up to G13 functions only. To make the comparison more meaningful and to get the deep insight on the performance of the HTS algorithm, optimization algorithms like PSO, BBO, DE, ABC, and TLBO are experimented on the remaining benchmark problems. Comparative results of all the algorithms are presented in Table 3 in the form of the best solution, worst solution, mean solution and standard deviation obtained in 100 independent runs.

The G14 test problem is a difficult function, and all the algorithms have failed to obtain a global optimum value for this test function. HTS algorithm has produced better mean results on G14 test function as compared to other competitive algorithms. The DE and HTS algorithms have shown a better mean result for the G15 test function. For the G16 test function, all the comparative algorithms (except BBO algorithm) have attained a global optimum value and have produced identical results. TLBO, ABC and DE algorithms have shown better mean results than other algorithms for G17, G18, and G19 test functions respectively. However, the HTS algorithm have produced competitive results for G17 and G19 test functions that are close to the results obtained by TLBO and DE algorithms respectively. G20, G22, and G23 test problems are difficult to solve, and all the algorithms have failed to obtain a global optimum value for these test functions. However, the HTS algorithm has produced a better mean result on G20 and G23 functions as compared to other algorithms. The HTS algorithm have produced a better mean result for G21 test function while all the comparative algorithms have produced competitive results and have attained a global optimum value for the G24 test function.

The success rate and computational effort required by optimization algorithms are also important to evaluate the performance. In the present work, experiments have been carried out to obtain the success rate and computational effort required by HTS algorithm for solving all 24 functions. The success rate and computational effort of other competitive algorithms (i.e. PSO, BBO, DE, ABC, and TLBO) for solving considered test functions are not available in the literature. Hence, experiments have been carried out with these algorithms to obtain their success rates and computational effort for solving all 24 functions.

Table 4 shows the success rate of all the algorithms considering G01–G24 functions obtained in 100 independent runs. To find out the success rate of the algorithm, a common error value of 0.01 is taken for all functions except G08, G11, G12, G16, G18, and G24. For these functions, the error value is taken as 0.001 as the optimum value of these functions lies between 0 and 1. During each run, the algorithm terminates when it completes 240,000 function evaluations or when it reaches the global optimum value within the gap of predefined error (i.e. 0.01 or 0.001). It is observed from the results of Table 4 that for six functions (G03, G05, G07, G09, G15, and G21) the HTS algorithm has produced better success rate than other algorithms. All the algorithms (except BBO algorithm) have obtained global optimum value in each run for seven functions (G04, G06, G08, G11, G12, G16, and G24), thus they have produced 100% success rate for these functions. TLBO and ABC algorithms have produced better success rate than other comparative algorithms for two test functions (G17 and G18). For G01 function, ABC and HTS algorithms have performed better than other algorithms and have produced 100% success rate for these functions. Out of twenty-four functions, all algorithms have failed to obtain a global optimum value for eight functions (G02, G10, G13, G14, G19, G20, G22, and G23) within predefined experimental conditions and hence have produced 0% success rate.

To analyse the computational effort required by each competitive algorithm, experimentations are carried out to obtained mean number of function evaluations for each algorithm to solve considered test functions. Here also, the algorithms are terminated when it completes 240,000 function evaluation or when it reaches the global optimum value within the gap of predefined error (i.e. 0.01 or 0.001). Out of 100 independent runs, only successful runs are considered for counting a mean number of function evaluations.

Table 5 shows the comparative results of each algorithm in terms of mean number of function evaluations obtained in 100 independent runs for each benchmark functions. It is observed from the results that for six functions (G03, G05, G07, G09, G17, and G21), HTS algorithm has been computationally more effective (i.e. HTS algorithm has obtained global optimum value with less number of function evaluations) than other comparative algorithms. However, the success rate of TLBO algorithm is better than HTS algorithm for the G17 test function.

For G01, G04, G06, G11 and G18 test functions, the TLBO algorithm has required less computational effort than other algorithms. However, the success rate of ABC and HTS algorithms is better than TLBO algorithm for G01, G18, and G01 test functions respectively.

**Table 3**
Comparative results of G14–G24 test functions obtained by different algorithms for 240,000 function evaluations averaged over 100 runs. Result in boldface indicated better performing algorithm.

| Function | | PSO | BBO | DE | ABC | TLBO | HTS |
|---|---|---|---|---|---|---|---|
| | Best | −44.9343 | 54.6979 | −45.7372 | −44.6431 | −46.5903 | −47.7278 |
| G14 | Worst | −37.5000 | 257.7061 | −12.7618 | −23.3210 | −17.4780 | −45.0648 |
| (−47.764) | Mean | −40.8710 | 175.9832 | −29.2187 | −40.1071 | −39.9725 | **−46.4076** |
| | SD | 2.29E+00 | 7.90E+01 | 1.36E+01 | 7.14E+00 | 1.15E+01 | 8.53E−01 |
| | Best | 961.7150 | 962.6640 | 961.7150 | 961.7568 | 961.7150 | 961.7150 |
| G15 | Worst | 972.3170 | 1087.3557 | 962.1022 | 970.3170 | 964.8922 | 962.0653 |
| (961.715) | Mean | 965.5154 | 1001.4367 | 961.7537 | 966.2868 | 962.8641 | **961.7500** |
| | SD | 3.72E+00 | 4.74E+01 | 1.22E−01 | 3.12E+00 | 1.49E+00 | 1.11E−01 |
| | Best | −1.9052 | −1.9052 | −1.9052 | −1.9052 | −1.9052 | −1.9052 |
| G16 | Worst | −1.9052 | −1.1586 | −1.9052 | −1.9052 | −1.9052 | −1.9052 |
| (−1.9052) | Mean | **−1.9052** | −1.6121 | **−1.9052** | **−1.9052** | **−1.9052** | **−1.9052** |
| | SD | 2.34E−16 | 2.58E−01 | 2.34E−16 | 2.34E−16 | 2.34E−16 | 2.34E−16 |
| | Best | 8857.5140 | 9008.5594 | 8854.6501 | 8859.7130 | 8853.5396 | 8853.5396 |
| G17 | Worst | 8965.4010 | 9916.7742 | 8996.3215 | 8997.1450 | 8919.6595 | 8932.0712 |
| (8853.5396) | Mean | 8899.4721 | 9384.2680 | 8932.0444 | 8941.9245 | **8876.5071** | 8877.9175 |
| | SD | 3.79E+01 | 3.06E+02 | 4.68E+01 | 4.26E+01 | 3.02E+01 | 3.09E+01 |
| | Best | −0.86603 | −0.65734 | −0.86531 | −0.86603 | −0.86603 | −0.86603 |
| G18 | Worst | −0.51085 | −0.38872 | −0.85510 | −0.86521 | −0.86294 | −0.67468 |
| (−0.86603) | Mean | −0.82760 | −0.56817 | −0.86165 | **−0.86587** | −0.86569 | −0.77036 |
| | SD | 1.11E−01 | 8.55E−02 | 3.67E−03 | 3.37E−04 | 9.67E−04 | 1.01E−01 |
| | Best | 33.5358 | 39.1471 | 32.6851 | 33.3325 | 32.7916 | 32.7132 |
| G19 | Worst | 39.8443 | 71.3106 | 32.9078 | 38.5614 | 36.1935 | 33.2140 |
| (32.6555) | Mean | 36.6172 | 51.8769 | **32.7680** | 36.0078 | 34.0792 | 32.7903 |
| | SD | 2.04E+00 | 1.12E+01 | 6.28E−02 | 1.83E+00 | 9.33E−01 | 1.53E−01 |
| | Best | 0.24743 | 1.26181 | 0.24743 | 0.24743 | 0.24743 | 0.24743 |
| G20 | Worst | 1.87320 | 1.98625 | 0.28766 | 1.52017 | 1.84773 | 0.27331 |
| (0.204979) | Mean | 0.97234 | 1.43488 | 0.26165 | 0.80536 | 1.22037 | **0.25519** |
| | SD | 6.34E−01 | 2.20E−01 | 1.91E−02 | 5.93E−01 | 5.89E−01 | 1.25E−02 |
| | Best | 193.7311 | 198.8151 | 193.7346 | 193.7343 | 193.7246 | 193.7264 |
| G21 | Worst | 409.1320 | 581.2178 | 418.4616 | 330.1638 | 393.8295 | 320.2342 |
| (193.274) | Mean | 345.6595 | 367.2513 | 366.9193 | 275.5436 | 264.6092 | **256.6091** |
| | SD | 6.36E+01 | 1.34E+02 | 9.13E+01 | 6.05E+01 | 9.23E+01 | 6.63E+01 |
| | Best | 1.68E+22 | 1.02E+15 | 1.25E+18 | 2.82E+08 | 4.50E+17 | 2.16E+03 |
| G22 | Worst | 3.25E+23 | 6.70E+16 | 2.67E+19 | 1.25E+18 | 4.06E+19 | 1.33E+07 |
| (236.4309) | Mean | 1.63E+23 | 1.41E+16 | 1.78E+19 | 4.10E+17 | 1.61E+19 | 1.36E+06 |
| | SD | 9.17E+22 | 1.96E+16 | 1.17E+19 | 4.72E+17 | 1.51E+19 | 4.20E+06 |
| | Best | −105.9826 | 2.3163 | −72.6420 | −43.2541 | −385.0043 | −390.6472 |
| G23 | Worst | 0 | 74.6089 | 0 | 0 | 0 | 0 |
| (−400.055) | Mean | −25.9179 | 22.1401 | −7.2642 | −4.3254 | −83.7728 | **−131.2522** |
| | SD | 4.30E+01 | 2.51E+01 | 2.30E+01 | 1.37E+01 | 1.59E+02 | 1.67E+02 |
| | Best | −5.5080 | −5.5080 | −5.5080 | −5.5080 | −5.5080 | −5.5080 |
| G24 | Worst | −5.5080 | −5.4857 | −5.5080 | −5.5080 | −5.5080 | −5.5080 |
| (−5.5080) | Mean | **−5.5080** | −5.4982 | **−5.5080** | **−5.5080** | **−5.5080** | **−5.5080** |
| | SD | 9.36E−16 | 6.75E−03 | 9.36E−16 | 9.36E−16 | 9.36E−16 | 9.36E−16 |

The bold values indicate best result.

The DE algorithm has performed computationally better than other algorithms for G12 and G15 test functions. However, the success rate of HTS and TLBO algorithm is better than the DE algorithm for the G15 test function. For G08, G16 and G24 test functions, the PSO algorithm is computationally more efficient than other algorithms.

The results given in Tables 2–5 indicates that the HTS algorithm has performed better than other algorithms in terms of the best result, mean results, success rate and computational efforts for constraint benchmark functions. Thus, the HTS is a promising algorithm for solving constrained optimization problems. Particularly, HTS is very efficient to solve quadratic, cubic and polynomial problems involving linear and nonlinear inequality constraints. But at the same time the HTS is not much efficient to handle linear problems involving linear and nonlinear equality constraints. Moreover, it is observed from the results that because of the proper trade-off between exploration and exploitation, the proposed algorithm can achieve the global optimum value for more benchmark functions as compared to the other algorithms.

The convergence speed of metaheuristic algorithms is also an important factor to evaluate their performance for solving the optimization problems. In the present work, the convergence performance of the HTS algorithm is compared with other algorithms for five test functions (G01, G03, G06, G08, and G24). The considered test functions possess a different characteristic of the objective function (i.e. G01 is quadratic, G03 is polynomial, G06 is cubical, G08 is non-linear and G24 is linear) and have a different number of variables. The algorithms are implemented with 30 independent runs on the considered test functions with 240,000 function evaluations in each run. Based on the results obtained through 30 independent runs, convergence graph is plotted between the average fitness value (i.e. an average function value) and function evaluations. Figs. 4–8 show the convergence

**Table 4**
Success rate of various algorithms for G01–G24 functions over 100 run.

| Function | PSO | BBO | DE | ABC | TLBO | HTS |
|---|---|---|---|---|---|---|
| G01 | 38 | 0 | 94 | 100 | 26 | 100 |
| G02 | 0 | 0 | 0 | 0 | 0 | 0 |
| G03 | 59 | 23 | 41 | 67 | 74 | 86 |
| G04 | 100 | 16 | 100 | 100 | 100 | 100 |
| G05 | 61 | 0 | 93 | 28 | 92 | 95 |
| G06 | 100 | 21 | 100 | 100 | 100 | 100 |
| G07 | 21 | 0 | 26 | 28 | 23 | 37 |
| G08 | 100 | 94 | 100 | 100 | 100 | 100 |
| G09 | 84 | 26 | 95 | 89 | 91 | 96 |
| G10 | 0 | 0 | 0 | 0 | 0 | 0 |
| G11 | 100 | 57 | 19 | 100 | 100 | 100 |
| G12 | 100 | 100 | 100 | 100 | 100 | 100 |
| G13 | 0 | 0 | 0 | 0 | 0 | 0 |
| G14 | 0 | 0 | 0 | 0 | 0 | 0 |
| G15 | 53 | 0 | 73 | 42 | 81 | 83 |
| G16 | 100 | 18 | 100 | 100 | 100 | 100 |
| G17 | 0 | 0 | 0 | 0 | 58 | 26 |
| G18 | 56 | 0 | 61 | 73 | 64 | 47 |
| G19 | 0 | 0 | 0 | 0 | 0 | 0 |
| G20 | 0 | 0 | 0 | 0 | 0 | 0 |
| G21 | 12 | 0 | 24 | 36 | 35 | 48 |
| G22 | 0 | 0 | 0 | 0 | 0 | 0 |
| G23 | 0 | 0 | 0 | 0 | 0 | 0 |
| G24 | 100 | 27 | 100 | 100 | 100 | 100 |

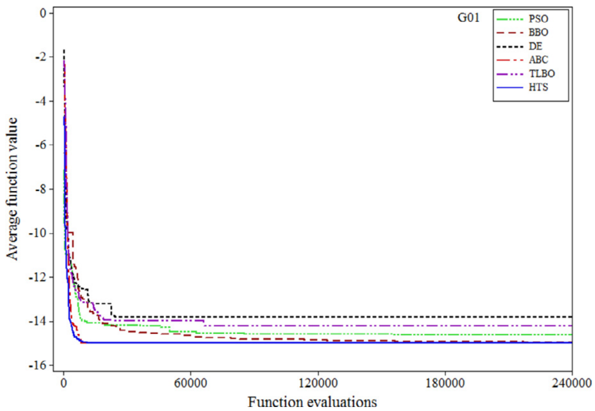


**Fig. 4.** Convergence of comparative algorithms for quadratic function (G01).

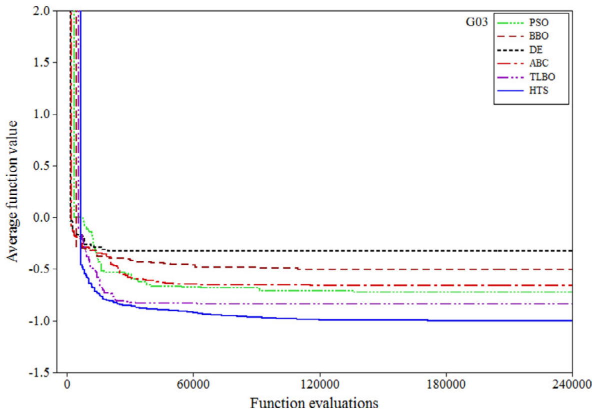


**Fig. 5.** Convergence of comparative algorithms for polynomial function (G03).

**Table 5**
Mean number of function evaluations required to reach global optimum value by comparative algorithms for G01–G24 over 100 independent runs.

| | | PSO | BBO | DE | ABC | TLBO | HTS |
|---|---|---|---|---|---|---|---|
| G01 | Mean_FE | 33750 | —— | 6988.89 | 13200 | 9750 | 12570 |
| | Std_FE | 3872.01 | —— | 157.674 | 708.676 | 4171.93 | 6312.43 |
| G02 | Mean_FE | —— | —— | —— | —— | —— | —— |
| | Std_FE | —— | —— | —— | —— | —— | —— |
| G03 | Mean_FE | 84610 | 157950 | 136350 | 121950 | 178083 | 67400 |
| | Std_FE | 35670 | 5939.7 | 78988.3 | 64296.1 | 43819.3 | 29776.2 |
| G04 | Mean_FE | 14432 | 189475 | 14090 | 29460 | 5470 | 10135 |
| | Std_FE | 309.23 | 35390.7 | 1499.22 | 2619.25 | 804.225 | 1413.63 |
| G05 | Mean_FE | 57921 | —— | 108572 | 197749 | 46888 | 43356 |
| | Std_FE | 14277.4 | —— | 41757.1 | 20576.8 | 19623.2 | 34164.3 |
| G06 | Mean_FE | 14923 | 140150 | 17540 | 69310 | 11600 | 15395 |
| | Std_FE | 1789.32 | 22273.9 | 1214.91 | 3753.65 | 2056.43 | 2566.61 |
| G07 | Mean_FE | 97742 | —— | 147650 | 114351 | 147550 | 92916.7 |
| | Std_FE | 2984.2 | —— | 4737.62 | 11384.4 | 5020.46 | 17237.3 |
| G08 | Mean_FE | 622 | 4290 | 725 | 670 | 680 | 635 |
| | Std_FE | 189.78 | 4418.32 | 259.54 | 249.666 | 181.353 | 171.675 |
| G09 | Mean_FE | 34877 | 194700 | 57205 | 149642 | 37690 | 23235 |
| | Std_FE | 12280.1 | 29557.1 | 10779.1 | 73436.8 | 26350.6 | 10806.2 |
| G10 | Mean_FE | —— | —— | —— | —— | —— | —— |
| | Std_FE | —— | —— | —— | —— | —— | —— |
| G11 | Mean_FE | 23312 | 35490 | 205250 | 29140 | 3000 | 53270 |
| | Std_FE | 1231.41 | 30627.4 | 8273.15 | 12982.5 | 1354.83 | 18215.2 |
| G12 | Mean_FE | 1204 | 1865 | 1150 | 1190 | 2480 | 2190 |
| | Std_FE | 341.3 | 2240.54 | 263.523 | 747.514 | 917.484 | 824.554 |
| G13 | Mean_FE | —— | —— | —— | —— | —— | —— |
| | Std_FE | —— | —— | —— | —— | —— | —— |
| G14 | Mean_FE | —— | —— | —— | —— | —— | —— |
| | Std_FE | —— | —— | —— | —— | —— | —— |
| G15 | Mean_FE | 41972 | —— | 36391.7 | 157800 | 52287.5 | 36756.3 |
| | Std_FE | 4073.9 | —— | 5509.21 | 57558.5 | 47937.1 | 28670.6 |
| G16 | Mean_FE | 7114 | 85200 | 12565 | 19670 | 7840 | 13045 |
| | Std_FE | 643.3 | 16122 | 1155.19 | 714.998 | 2709.74 | 1358.6 |
| G17 | Mean_FE | —— | —— | —— | —— | 126980 | 65600 |
| | Std_FE | —— | —— | —— | —— | 46591.8 | 65053.8 |
| G18 | Mean_FE | 23769 | —— | 170140 | 114120 | 19226 | 35360 |
| | Std_FE | 1009.78 | —— | 20227.7 | 58105.8 | 5762.16 | 7731.14 |
| G19 | Mean_FE | —— | —— | —— | —— | —— | —— |
| | Std_FE | —— | —— | —— | —— | —— | —— |
| G20 | Mean_FE | —— | —— | —— | —— | —— | —— |
| | Std_FE | —— | —— | —— | —— | —— | —— |
| G21 | Mean_FE | 39937 | —— | 89500 | 99150 | 108533 | 28037.5 |
| | Std_FE | 4302.2 | —— | 14283.6 | 3647.94 | 8677.17 | 7032.35 |
| G22 | Mean_FE | —— | —— | —— | —— | —— | —— |
| | Std_FE | —— | —— | —— | —— | —— | —— |
| G23 | Mean_FE | —— | —— | —— | —— | —— | —— |
| | Std_FE | —— | —— | —— | —— | —— | —— |
| G24 | Mean_FE | 2469 | 84625 | 4855 | 5400 | 2710 | 3715 |
| | Std_FE | 245.5 | 2015.25 | 429.761 | 618.241 | 864.677 | 575.929 |

—— indicates that algorithm is failed to obtained a global optimum value for that function,
Mean_FE = Mean number of function evaluations.
Std_FE = Standard deviation of function evaluations.

graphs of different algorithms for considered test functions. It is observed from the convergence graphs that the HTS algorithm has a good convergence performance in comparison with the other algorithms considered in the present work.

To get more insight into the performance of each phase of the HTS algorithm, every phase of the HTS algorithm is experimented individually on the five test functions (G01, G02, G03, G05, and G24). The considered test functions possess different characteristics as explained above. Each phase of the HTS algorithm is executed individually for considered test functions. The average results obtained in 100 independent runs of each phase on considered test functions are presented in Table 6 in the form of the best solution, worst solution, mean solution and mean number of function evaluations to reach the global optimum value. It is observed from results that individually the performance of each phase is inferior as compared to its combination. Results of Table 6 show that for quadratic, cubic and polynomial functions, radiation phase is more competent, while for linear function convection phase is more competent, and for nonlinear function conduction phase is more competent. Hence, for efficient working of the proposed algorithm, all three phases are required to be executed during an optimization process. Furthermore, stepwise calculations of one constraint function with each phase of the proposed algorithm is given in Appendix A.

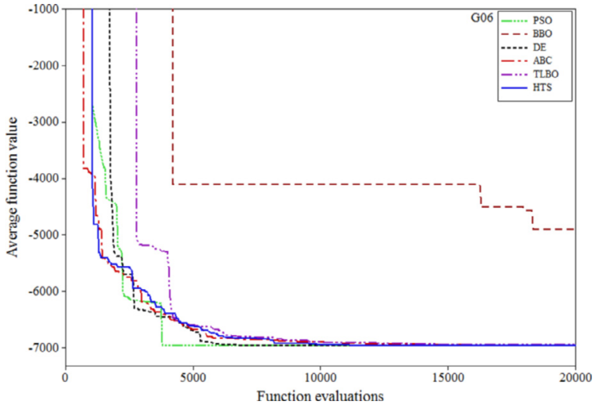


**Fig. 6.** Convergence of comparative algorithms for cubic function (G06).

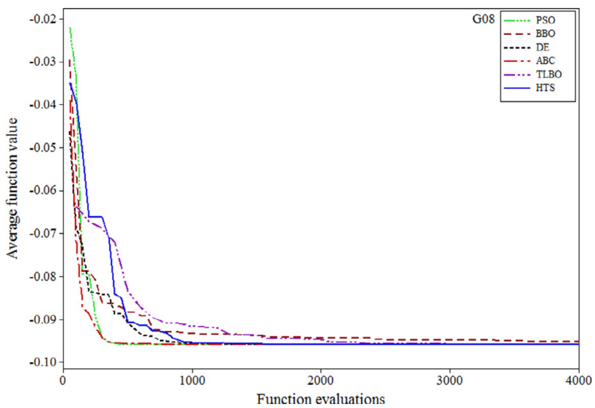


**Fig. 7.** Convergence of comparative algorithms for nonlinear function (G08).

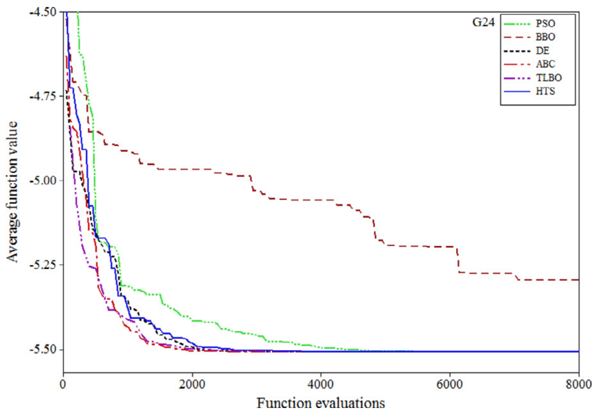


**Fig. 8.** Convergence of comparative algorithms for linear function (G24).

The next section presents the statistical tests conducted on the results obtained by comparative algorithms to identify statistical significance among them.

## 5. Statistical analysis

From results of Tables 2–5 it can be understood that HTS is a better performing algorithm for constraint optimization problems, but at the same time it is also important to find out statistical significance of HTS over other comparative algorithms. Similarly, it is also important to give rank to all the algorithms based on the obtained results. Hence, it is required to follow standard statistical test to rank and to check the statistical significance of the algorithms. To quantify the performance of the

**Table 6**
Comparative results of G01, G02, G03, G05 and G24 test functions obtained by each phase of the HTS algorithm for 240,000 function evaluations averaged over 100 runs.

| Function | | Conduction phase | Radiation phase | Convection phase |
|---|---|---|---|---|
| G01 | Best | −14.1302 | −15 | −14.9834 |
| | Worst | −12.0635 | −14 | −11.9663 |
| | Mean | −13.0284 | −14.7943 | −14.19032 |
| | Mean FE | 240,000 | 109,850 | 240,000 |
| G02 | Best | −0.77622 | −0.77101 | −0.60803 |
| | Worst | −0.75085 | −0.51318 | −0.51377 |
| | Mean | −0.76299 | −0.645912 | −0.581124 |
| | Mean FE | 240,000 | 240,000 | 240,000 |
| G03 | Best | −0.59147 | −0.962071 | −0.088924 |
| | Worst | −0.25811 | −0.000672 | −0.000309 |
| | Mean | −0.365716 | −0.763304 | −0.038169 |
| | Mean FE | 240,000 | 240,000 | 240,000 |
| G05 | Best | 5267.2903 | 5126.486 | 5245.8294 |
| | Worst | 6346.3574 | 5126.5067 | 6030.0885 |
| | Mean | 5654.9045 | 5126.5036 | 5551.7428 |
| | Mean FE | 240,000 | 54,640 | 240,000 |
| G24 | Best | −5.1751 | −5.508 | −5.508 |
| | Worst | −4.5338 | −5.508 | −5.508 |
| | Mean | −4.93268 | −5.508 | −5.508 |
| | Mean FE | 240,000 | 32,230 | 14,980 |

**Table 7**
Friedman Rank test for the ‘Best’ and ‘Mean’ solutions obtained (For G01–G13 Functions).

| Test for best solution | | | | Test for mean solution | | | |
|---|---|---|---|---|---|---|---|
| Algorithms | Friedman value | Normalized value | Rank | Algorithms | Friedman value | Normalized value | Rank |
| HM | 88 | 2.15 | 9 | HM | 76.5 | 2.19 | 8 |
| ASCHEA | 69 | 1.68 | 8 | ASCHEA | 71.5 | 2.04 | 7 |
| SMES | 58 | 1.41 | 6 | SMES | 60 | 1.71 | 6 |
| GA | 95 | 2.32 | 10 | GA | 87.5 | 2.5 | 10 |
| PSO | 41 | 1 | 1 | PSO | 47 | 1.34 | 3 |
| DE | 50.5 | 1.23 | 5 | DE | 53.5 | 1.53 | 5 |
| ABC | 47 | 1.15 | 3 | ABC | 38 | 1.09 | 2 |
| BBO | 63.5 | 1.55 | 7 | BBO | 84 | 2.4 | 9 |
| TLBO | 47.5 | 1.16 | 4 | TLBO | 52 | 1.49 | 4 |
| HTS | 45.5 | 1.11 | 2 | HTS | 35 | 1 | 1 |

**Table 8**
Friedman Rank test for the ‘Best’ and ‘Mean’ solutions obtained (For G14–G24 functions).

| Test for best solution | | | | Test for mean solution | | | |
|---|---|---|---|---|---|---|---|
| Algorithms | Friedman value | Normalized value | Rank | Algorithms | Friedman value | Normalized value | Rank |
| PSO | 40 | 1.7 | 4 | PSO | 41 | 1.95 | 5 |
| BBO | 58 | 2.47 | 6 | BBO | 62 | 2.95 | 6 |
| DE | 38.5 | 1.64 | 3 | DE | 37 | 1.76 | 3 |
| ABC | 42.5 | 1.81 | 5 | ABC | 38 | 1.81 | 4 |
| TLBO | 28.5 | 1.21 | 2 | TLBO | 32 | 1.52 | 2 |
| HTS | 23.5 | 1 | 1 | HTS | 21 | 1 | 1 |

algorithms statistically, Friedman Rank Test [19] and Post Hoc Holm–Sidak test [16] is performed on best and mean solutions obtained by considered algorithms. Student $t$-test can be used to compare only two data sets. If there are more than two groups, say ‘$N$’ groups then using $t$-test can mislead the statistical significance because if $t$-test is performed at fixed $\alpha = 0.05$, there is a probability of $N*0.05$ to find a difference where it actually do not exist. The Holm–Sidak test is a step-down ‘recursive reject’ method, as it applies accept or reject criterion on the sorted set of null-hypothesis. A $p$-value provides information about whether a statistical hypothesis test is significant or not, and it also indicates something about how significant the result is. The smaller $p$-value indicates stronger evidence against null-hypothesis. It does this without committing to a particular level of significance and increases to the acceptance of the null-hypothesis. For each comparison, the alpha value is set according to Sidak correction of Bonferroni inequality [16].

**Table 9**
Friedman Rank test for the 'Success Rate' solutions obtained.

| Algorithms | Friedman value | Normalized value | Rank |
|---|---|---|---|
| PSO | 79 | 1.32 | 5 |
| BBO | 51.5 | 2.02 | 6 |
| DE | 85.5 | 1.22 | 4 |
| ABC | 90.5 | 1.15 | 3 |
| TLBO | 93.5 | 1.11 | 2 |
| HTS | 104 | 1.00 | 1 |

**Table 10**
Holm–Sidak test for the 'Best' and the 'Mean' solutions obtained (For G01–G13 functions).

| Test for best solution | | Test for mean solution | |
|---|---|---|---|
| Algorithm[a] | $p$-value | Algorithm[a] | $p$-value |
| 1–5 | 0.02083 | 1–7 | 0.04131 |
| 1–9 | 0.48556 | 1–5 | 0.36481 |
| 1–10 | 0.82356 | 1–9 | 0.42609 |
| 1–2 | 0.90131 | 1–2 | 0.62039 |
| 1–3 | 0.98783 | 1–3 | 0.9105 |
| 1–7 | 0.99336 | 1–10 | 0.9665 |
| 1–4 | 0.99596 | 1–4 | 0.96797 |
| 1–8 | 0.99612 | 1–8 | 0.97536 |
| 1–6 | 0.99996 | 1–6 | 0.97738 |

[a] 1-HTS, 2-HM, 3-ASCHEA, 4-SMES, 5-GA, 6-PSO, 7-DE, 8-ABC, 9-BBO, 10-TLBO.

**Table 11**
Holm–Sidak test for the 'Best' and the 'Mean' solutions obtained (For G14–G24 functions).

| Test for best solution | | Test for mean solution | |
|---|---|---|---|
| Algorithm[a] | $p$-value | Algorithm[a] | $p$-value |
| 1–3 | 0.013864 | 1–3 | 0.0008 |
| 1–5 | 0.17505 | 1–4 | 0.30255 |
| 1–4 | 0.22207 | 1–2 | 0.43701 |
| 1–2 | 0.26571 | 1–5 | 0.44769 |
| 1–6 | 0.97902 | 1–6 | 0.82863 |

[a] 1-HTS, 2-PSO, 3-BBO, 4-DE, 5-ABC, 6-TLBO.

Friedman rank test is performed on the best and the mean solutions obtained by competitive algorithms. Tables 7 and 8 have presented the Friedman rank test on the best and the mean solutions for G01 to G13 functions (G05, G12 and G13 functions are omitted from the test as the result of these functions are not available for some of the competitive algorithms) and G14–G24 functions (G22 function is omitted from the test as none of the competitive algorithms have produced feasible solution for this function) respectively. To have the more insight and understanding of the results the Friedman value obtained using Friedman test is normalized with respect to the best value obtained. So, '1' indicates the best algorithm and algorithm having value away from '1' shows its inferiority with respect to the best algorithm. Algorithms are ranked based on the normalized value, and it is observed from results that HTS stands first to obtain mean solutions, and PSO stands first to obtain best solutions for G01–G13 functions. The HTS has secured second rank to obtain the best solution for G01–G13 functions. For G14–G24 functions, HTS stands first to obtain best and mean solutions followed by TLBO and DE. However, the difference in the normalized value of TLBO and DE is more than HTS for the mean solutions. Furthermore, it is important to perform Friedman rank test for the success rate of the considered algorithms. Table 9 shows results of Friedman rank test for the success rate of the algorithms. It is observed from the results that HTS has secured first rank followed by TLBO and ABC algorithms.

Friedman rank test is only used to rank the algorithms based on the result data, but it do not specify any statistical difference in the results and hence Holm–Sidak test is used to determine the statistical difference. Tables 10 and 11 have presented the result of Holm–Sidak test on best and mean solutions for G01–G13 functions and G14–G24 functions respectively. The results depict the pairwise $p$-value obtained from the Holm–Sidak test for all algorithms. It is observed from the results that HTS, PSO, ABC and SMES algorithms are statistically similar for obtaining best and mean solutions for function G01–G13. For G14–G24 functions, a significant statistical difference has been observed in the results of the HTS algorithm and other competitive algorithms (except TLBO) for obtaining the best and the mean solutions.

## 6. Conclusions

In this paper, a novel global efficient metaheuristic algorithm called Heat Transfer Search (HTS) has been introduced. The HTS algorithm is inspired by the law of thermodynamics and heat transfer, which states that ‘Any system’ always try to achieve thermal equilibrium state with its surrounding by conducting heat transfer in the form of conduction, convection, and radiation. In HTS, a population is analogous to the clusters of molecules that take part in the heat transfer process. Different design variables are considered as different temperature levels possessed by the molecules and energy level of the molecules represent the objective function value. The best solution is treated as the surrounding and rest of the solutions represent a system. The search process of HTS considers ‘conduction phase’, ‘convection phase’ and ‘radiation phase’. Each phase of the proposed algorithm is executed with equal probability during an entire search process. The proper trade-off between exploration and exploitation during search process of all three phases are controlled by introducing conduction factor, convection factor and radiation factor. The ability of the HTS algorithm is investigated by applying it to 24 well defined constrained benchmark problems of CEC 2006. Results obtained using the proposed algorithm is compared with results of other well-known optimization algorithms. All algorithms are compared by considering best solutions, mean solutions, success rate, computational effort and convergence rate. The statistical analysis of the experimental work is also carried out by conducting Friedman rank test and Holm post hoc procedure. The comparative results show better performance of HTS algorithm over other nature-inspired optimization algorithms for the constraint optimization problems.

## APPENDIX A. Stepwise calculation of one constraint function with each phase of the HTS algorithm

***Step 1: Define optimization problem and initialize the optimization parameters (Input)***

$$\text{Optimization problem}: \textit{Minimize} \left(X_1^2 + X_2 - 11\right)^2 + \left(X_1 + X_2^2 - 7\right)^2$$
$$S/T \quad (X_1 - 5)^2 + X_2^2 - 26 \geq 0$$
$$0 \leq X_1, X_2 \leq 10$$

The following optimization parameters are considered for the above function:

Population size = 10
Number of design variables = 2
Limits of design variables = $0 \leq x_1, x_2 \leq 10$
Elite size = 2
Number of generations = 10

***Step 2: Initialize population***

In the HTS, population is analogous to the cluster of molecules that take part in the heat transfer process. Different design variables are analogous to the different temperature possessed by the molecules whereas the energy level of the molecules represents the fitness value of the objective function.

| $X_1$ | $X_2$ | Value of constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 2.85849 | 7.80315 | 39.47519 | 3245.0353 | 3.25E+03 |
| 8.52872 | 3.45956 | −1.57959 | 4433.0367 | 4.43E+07 |
| 7.87378 | 2.21558 | −12.8326 | 2864.9549 | 2.86E+07 |
| 2.37038 | 9.04479 | 62.72321 | 5969.9682 | 5.97E+03 |
| 7.64172 | 6.29206 | 20.56871 | 4500.9888 | 4.50E+03 |
| 1.65752 | 1.73999 | −11.8003 | 47.773366 | 4.78E + 05 |
| 5.04666 | 6.36876 | 14.56327 | 1924.7619 | 1.92E+03 |
| 2.27991 | 6.75889 | 27.08144 | 1678.8415 | 1.68E+03 |
| 1.93728 | 7.90521 | 45.87266 | 3298.6029 | 3.30E+03 |
| 2.11316 | 4.05676 | −1.20886 | 140.01465 | 1.40E+06 |

In the initial population, the solution generated by population 2, 3, 6 and 10 violate the constraints hence the penalty is added in to the solutions generated by these population. Moreover, the elite size is 2 hence two best solution of the initial population are remembered to replace the 2 worst solutions at the end of first generation. The solution generated by population 7 and 8 are remembered as elite solution.

***Step 3: First generation***

During the first generation, the above mentioned initial solution updated either by conduction, convection or radiation phase depends on the value of $R$ produced by random number during first generation. Suppose the value of $R$ produced by random number during first generation is 0.3637. Since, this value of $R$ is corresponding to radiation phase hence in the first generation solution is updated by radiation phase using Eqs. (16), (17), (20) and (21). The solution generated by population 4 in step 2 is considered as random solution (i.e. $f(X_k)$) for updating the rest of the solutions. Based on this, the solutions updated in the radiation phase are mention below.

| $X_1$ | $X_2$ | Value of constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 3.03602 | 7.35156 | 31.90269 | 2539.17 | 2.54E+03 |
| 6.28893 | 5.49091 | 5.81142 | 2025.4827 | 2.03E+03 |
| 5.87219 | 4.69936 | −3.15527 | 1233.388 | 1.23E+07 |
| 2.5479 | 8.59321 | 53.85599 | 4831.8151 | 4.83E+03 |
| 9.55891 | 5.29089 | 22.77717 | 8271.7043 | 8.27E+03 |
| 1.91678 | 4.39675 | 2.837597 | 211.59041 | 2.12E+02 |
| 6.02003 | 5.39549 | 4.151712 | 1729.9457 | 1.73E+03 |
| 2.24701 | 5.92751 | 16.71425 | 923.08663 | 9.23E+02 |
| 1.77976 | 7.49075 | 40.48125 | 2590.0194 | 2.59E+03 |
| 2.01961 | 2.24261 | −12.088 | 21.891489 | 2.19E+05 |

***Step 4: Greedy selection***

Apply the greedy selection between the initial solution (step 2) and solution updated in the radiation phase (step 3).

| $X_1$ | $X_2$ | Value of constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 3.03602 | 7.35156 | 31.9027 | 2539.17 | 2539.17 |
| 6.28893 | 5.49091 | 5.81142 | 2025.48 | 2025.483 |
| 5.87219 | 4.69936 | −3.1553 | 1233.39 | 12333880 |
| 2.5479 | 8.59321 | 53.856 | 4831.82 | 4831.815 |
| 7.64172 | 6.29206 | 20.5687 | 4500.99 | 4500.989 |
| 1.91678 | 4.39675 | 2.8376 | 211.59 | 211.5904 |
| 6.02003 | 5.39549 | 4.15171 | 1729.95 | 1729.946 |
| 2.24701 | 5.92751 | 16.7143 | 923.087 | 923.0866 |
| 1.77976 | 7.49075 | 40.4812 | 2590.02 | 2590.019 |
| 2.11316 | 4.05676 | −1.2089 | 140.015 | 1400147 |

***Step 5: Replace the worst solutions of step 4 by elite solution of step 1***

| $X_1$ | $X_2$ | Value of constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 3.03602 | 7.35156 | 31.90269 | 2539.17 | 2539.17 |
| 6.28893 | 5.49091 | 5.81142 | 2025.4827 | 2025.4827 |
| 2.27991 | 6.75889 | 27.08144 | 1678.8415 | 1678.8415 |
| 2.5479 | 8.59321 | 53.85599 | 4831.8151 | 4831.8151 |
| 7.64172 | 6.29206 | 20.56871 | 4500.9888 | 4500.9888 |
| 1.91678 | 4.39675 | 2.837597 | 211.59041 | 211.59041 |
| 6.02003 | 5.39549 | 4.151712 | 1729.9457 | 1729.9457 |
| 2.24701 | 5.92751 | 16.71425 | 923.08663 | 923.08663 |
| 1.77976 | 7.49075 | 40.48125 | 2590.0194 | 2590.0194 |
| 5.04666 | 6.36876 | 14.56327 | 1924.7619 | 1924.7619 |

As no duplicate solutions exists after replacing worst solutions with elite solutions hence no need to modify duplicate solutions. Again 2 best solutions (solution produce by population 6 and 8) of the above table are kept as elite solutions for the next generation.

***Step 6: Second generation (Solution update by convection phase)***

Suppose the value of $R$ produced by random number during second generation is 0.9187. Since, this value of $R$ is corresponding to convection phase hence in the second generation solution is updated by convection phase using Eqs. (12), (13) and (14). Following values are considered for the updating solutions in this phase.

Mean value of design variables $X_1$ and $X_2$ obtained in step 5 = [3.88047 6.40658]
Design variables of best solution (i.e. surrounding) of step 5 = [1.91678 4.39675]
Temperature change factor (TCF) = 0.9106

Based on these, the solutions are updated as below.

| $X_1$ | $X_2$ | Value of constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 1.55069 | 6.0313 | 22.27435 | 963.07263 | 9.63E+02 |
| 4.8036 | 4.17065 | −8.56712 | 494.88388 | 4.95E+06 |
| 0.79458 | 5.43863 | 21.2642 | 570.61463 | 5.71E+02 |
| 1.06257 | 7.27295 | 42.39909 | 2211.8353 | 2.21E+03 |
| 6.15639 | 4.9718 | 0.056041 | 1585.9089 | 1.59E+03 |
| 0.43145 | 3.07649 | 4.336374 | 68.254861 | 6.83E+01 |
| 4.5347 | 4.07523 | −9.17603 | 386.01446 | 3.86E+06 |
| 0.76168 | 4.60725 | 13.19004 | 258.43814 | 2.58E+02 |
| 0.29443 | 6.17049 | 34.21729 | 1006.5317 | 1.01E+03 |
| 3.56133 | 5.0485 | 1.557109 | 531.4581 | 5.31E+02 |

***Step 7: Greedy selection***

Apply the greedy selection between the solutions obtained in first generation (step 5) and solutions updated in the convection phase (step 6).

| $X_1$ | $X_2$ | Value of constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 1.55069 | 6.0313 | 22.2744 | 963.073 | 9.63E+02 |
| 6.28893 | 5.49091 | 5.81142 | 2025.48 | 2.03E+03 |
| 0.79458 | 5.43863 | 21.2642 | 570.615 | 5.71E+02 |
| 1.06257 | 7.27295 | 42.3991 | 2211.84 | 2.21E+03 |
| 6.15639 | 4.9718 | 0.05604 | 1585.91 | 1.59E+03 |
| 0.43145 | 3.07649 | 4.33637 | 68.2549 | 6.83E+01 |
| 6.02003 | 5.39549 | 4.15171 | 1729.95 | 1.73E+03 |
| 0.76168 | 4.60725 | 13.19 | 258.438 | 2.58E+02 |
| 0.29443 | 6.17049 | 34.2173 | 1006.53 | 1.01E+03 |
| 3.56133 | 5.0485 | 1.55711 | 531.458 | 5.31E+02 |

***Step 8: Replace the worst solutions of step 7 by elite solution of step 5.***

| $X_1$ | $X_2$ | Value of constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 1.55069 | 6.0313 | 22.27435 | 963.07263 | 9.63E+02 |
| 2.24701 | 5.92751 | 16.71425 | 923.08663 | 9.23E+02 |
| 0.79458 | 5.43863 | 21.2642 | 570.61463 | 5.71E+02 |
| 1.91678 | 4.39675 | 2.837597 | 211.59041 | 2.12E+02 |
| 6.15639 | 4.9718 | 0.056041 | 1585.9089 | 1.59E+03 |
| 0.43145 | 3.07649 | 4.336374 | 68.254861 | 6.83E+01 |
| 6.02003 | 5.39549 | 4.151712 | 1729.9457 | 1.73E+03 |
| 0.76168 | 4.60725 | 13.19004 | 258.43814 | 2.58E+02 |
| 0.29443 | 6.17049 | 34.21729 | 1006.5317 | 1.01E+03 |
| 3.56133 | 5.0485 | 1.557109 | 531.4581 | 5.31E+02 |

Again no duplicate solutions exist after replacing worst solutions with elite solutions hence no need to modify duplicate solutions. Moreover, 2 best solutions (solution produce by population 4 and 6) of the above table kept as elite solutions for the next generation.

***Step 9: Third generation (Solution update by conduction phase)***

Suppose the value of $R$ produced by random number during third generation is 0.1488. Since, this value of $R$ is corresponding to conduction phase hence in the third generation solution is updated by conduction phase using Eqs. (4)–(7). The solution generated by population 4 in step 8 is considered as random solution (i.e. $f(X_k)$) and its second design variable (i.e. $X_2$) is considered as random variable for updating the rest of the solutions. Based on this, the solutions updated in the conduction phase are mention below.

| $X_1$ | $X_2$ | Value of Constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 1.55069 | 4.2994 | 4.382547 | 188.37956 | 1.88E+02 |
| 2.24701 | 4.2994 | 0.063746 | 191.29053 | 1.91E+02 |
| 0.79458 | 4.2994 | 10.17033 | 187.61919 | 1.88E+02 |
| 1.91678 | 4.50523 | 3.80335 | 239.41973 | 2.39E+02 |
| 6.15639 | 4.2994 | −6.17795 | 1284.6851 | 1.28E+07 |
| 0.43145 | 3.07649 | 4.336374 | 68.254861 | 6.83E+01 |
| 6.02003 | 4.2994 | −6.47474 | 1179.0373 | 1.18E+07 |
| 0.76168 | 4.2994 | 10.44814 | 187.4364 | 1.87E+02 |
| 0.29443 | 4.2994 | 14.62717 | 182.49441 | 1.82E+02 |
| 3.56133 | 4.2994 | −5.44542 | 262.17638 | 2.62E+06 |

***Step 10: Greedy selection***

Apply the greedy selection between the solutions obtained in second generation (step 8) and solutions updated in the conduction phase (step 9).

| $X_1$ | $X_2$ | Value of Constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 1.55069 | 4.2994 | 4.38255 | 188.38 | 188.3796 |
| 2.24701 | 4.2994 | 0.06375 | 191.291 | 191.2905 |
| 0.79458 | 4.2994 | 10.1703 | 187.619 | 187.6192 |
| 1.91678 | 4.39675 | 2.8376 | 211.59 | 211.5904 |
| 6.15639 | 4.9718 | 0.05604 | 1585.91 | 1585.909 |
| 0.43145 | 3.07649 | 4.33637 | 68.2549 | 68.25486 |
| 6.02003 | 5.39549 | 4.15171 | 1729.95 | 1729.946 |
| 0.76168 | 4.2994 | 10.4481 | 187.436 | 187.4364 |
| 0.29443 | 4.2994 | 14.6272 | 182.494 | 182.4944 |
| 3.56133 | 5.0485 | 1.55711 | 531.458 | 531.4581 |

***Step 11: Replace the worst solutions of step 10 by elite solution of step 8***

| $X_1$ | $X_2$ | Value of constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 1.55069 | 4.2994 | 4.382547 | 188.37956 | 188.37956 |
| 2.24701 | 4.2994 | 0.063746 | 191.29053 | 191.29053 |
| 0.79458 | 4.2994 | 10.17033 | 187.61919 | 187.61919 |
| 1.91678 | 4.39675 | 2.837597 | 211.59041 | 211.59041 |
| 0.43145 | 3.07649 | 4.336374 | 68.254861 | 68.254861 |
| 0.43145 | 3.07649 | 4.336374 | 68.254861 | 68.254861 |
| 1.91678 | 4.39675 | 2.837597 | 211.59041 | 2.12E + 02 |
| 0.76168 | 4.2994 | 10.44814 | 187.4364 | 187.4364 |
| 0.29443 | 4.2994 | 14.62717 | 182.49441 | 182.49441 |
| 3.56133 | 5.0485 | 1.557109 | 531.4581 | 531.4581 |

***Step 12: Modified the duplicate solutions (Output)***

Duplicate solutions exist in the results obtained at the end of conduction phase (step 11). So, any one design variable of the duplicate solution is modified using the random number. Here, second design variable (i.e. $X_2$) of the population 6 and 7 are modified using the random number. So, finally following results are obtained at the end of conduction step.

| $X_1$ | $X_2$ | Value of constraint | $f(X)$ | $f(X)$ + Penalty |
|---|---|---|---|---|
| 1.55069 | 4.2994 | 4.382547 | 188.37956 | 188.37956 |
| 2.24701 | 4.2994 | 0.063746 | 191.29053 | 191.29053 |
| 0.79458 | 4.2994 | 10.17033 | 187.61919 | 187.61919 |
| 1.91678 | 4.27979 | 1.822845 | 184.40227 | 184.40227 |
| 0.43145 | 2.99465 | 3.839545 | 66.896906 | 66.896906 |
| 0.43145 | 3.07649 | 4.336374 | 68.254861 | 68.254861 |
| 1.91678 | 4.39675 | 2.837597 | 211.59041 | 211.59041 |
| 0.76168 | 4.2994 | 10.44814 | 187.4364 | 187.4364 |
| 0.29443 | 4.2994 | 14.62717 | 182.49441 | 182.49441 |
| 3.56133 | 5.0485 | 1.557109 | 531.4581 | 531.4581 |

Thus, the best solution obtained at the end of third generation is 66.896906 corresponding to population 5. The algorithm is executed till the maximum generation is reached in the similar manner as mentioned in the above steps.

## APPENDIX B. Characteristic and mathematical formulation of the benchmark function of CEC 2006

### • Characteristics of constrained benchmark functions

| Functions | $n$ | Type of function | $\rho$ | LI | NI | LE | NE | ac | $O$ |
|---|---|---|---|---|---|---|---|---|---|
| G01 | 13 | Quadratic | 0.0111% | 9 | 0 | 0 | 0 | 6 | −15 |
| G02 | 20 | Non-linear | 99.9971% | 0 | 2 | 0 | 0 | 1 | −0.8036 |
| G03 | 10 | Polynomial | 0.0000% | 0 | 0 | 0 | 1 | 1 | −1.0005 |
| G04 | 5 | Quadratic | 52.1230% | 0 | 6 | 0 | 0 | 2 | −30665.5 |
| G05 | 4 | Cubic | 0.0000% | 2 | 0 | 0 | 3 | 3 | 5126.496 |
| G06 | 2 | Cubic | 0.0066% | 0 | 2 | 0 | 0 | 2 | −6961.81 |
| G07 | 10 | Quadratic | 0.0003% | 3 | 5 | 0 | 0 | 6 | 24.3062 |
| G08 | 2 | Non-linear | 0.8560% | 0 | 2 | 0 | 0 | 0 | −0.09582 |
| G09 | 7 | Polynomial | 0.5121% | 0 | 4 | 0 | 0 | 2 | 680.63 |
| G10 | 8 | Linear | 0.0010% | 3 | 3 | 0 | 0 | 6 | 7049.248 |
| G11 | 2 | Quadratic | 0.0000% | 0 | 0 | 0 | 1 | 1 | 0.7499 |
| G12 | 3 | Quadratic | 4.7713% | 0 | 1 | 0 | 0 | 0 | −1 |
| G13 | 5 | Non-linear | 0.0000% | 0 | 0 | 0 | 3 | 3 | 0.0539 |
| G14 | 10 | Non-linear | 0.0000% | 0 | 0 | 3 | 0 | 3 | −47.7649 |
| G15 | 3 | Quadratic | 0.0000% | 0 | 0 | 1 | 1 | 2 | 961.715 |
| G16 | 5 | Non-linear | 0.0204% | 4 | 34 | 0 | 0 | 4 | −1.9051 |
| G17 | 6 | Non-linear | 0.0000% | 0 | 0 | 0 | 4 | 4 | 8853.539 |
| G18 | 9 | Quadratic | 0.0000% | 0 | 13 | 0 | 0 | 6 | −0.86602 |
| G19 | 15 | Non-linear | 33.4761% | 0 | 5 | 0 | 0 | 0 | 32.6556 |
| G20 | 24 | Linear | 0.0000% | 0 | 6 | 2 | 12 | 16 | 0.20498 |
| G21 | 7 | Linear | 0.0000% | 0 | 1 | 0 | 5 | 6 | 193.7245 |
| G22 | 22 | Linear | 0.0000% | 0 | 1 | 8 | 11 | 19 | 236.4309 |
| G23 | 9 | Linear | 0.0000% | 0 | 2 | 3 | 1 | 6 | −400.055 |
| G24 | 2 | Linear | 79.6556% | 0 | 2 | 0 | 0 | 2 | −5.50801 |

***n*** is the number of decision variables, $\boldsymbol{\rho}$ is the estimated ratio between the feasible region and the search space, **LI** is the number of linear inequality constraints, **NI** the number of nonlinear inequality constraints, **LE** is the number of linear equality constraints, **NE** is the number of nonlinear equality constraints, **ac** is the number of active constraints at the optimum solution and **O** is the global optimum result.

### • Mathematical formulation of constrained benchmark functions

**(1) G01:**

Minimize:

$$f(X) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$$

Subject to:

$$g_1(X) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \le 0$$
$$g_2(X) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \le 0$$
$$g_3(X) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \le 0$$
$$g_4(X) = -8x_1 + x_{10} \le 0$$
$$g_5(X) = -8x_2 + x_{11} \le 0$$
$$g_6(X) = -8x_3 + x_{12} \le 0$$
$$g_7(X) = -2x_4 - x_5 + x_{10} \le 0$$
$$g_8(X) = -2x_6 - x_7 + x_{11} \le 0$$
$$g_9(X) = -2x_8 - x_9 + x_{12} \le 0$$

where, the bounds are $0 \le x_i \le 1$ $(i = 1, \ldots, 9)$, $0 \le x_i \le 100$ $(i = 10,11,12)$ and $0 \le x_{13} \le 1$

**(2) G02:**

Minimize:

$$f(X) = -\left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$$

Subject to:

$$g_1(X) = 0.75 - \prod_{i=1}^{n} x_i \le 0 \, g_2(X) = \sum_{i=1}^{n} x_i - 7.5n \le 0$$

where, $n = 20$ and the bounds are $0 \le x_i \le 10$ ($i = 1,\ldots,n$)

**(3) G03:**

Minimize:

$$f(X) = -(\sqrt{n})^n \prod_{i=1}^{n} x_i$$

Subject to:

$$h_1(X) = \sum_{i=1}^{n} x_i^2 - 1 = 0$$

where, $n = 10$ and the bounds are $0 \le x_i \le 1$ ($i = 1,\ldots,n$)

**(4) G04:**

Minimize:

$$f(X) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$$

Subject to:

$$g_1(X) = 85.334407 + 0.005685x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \le 0$$
$$g_2(X) = -85.334407 - 0.005685x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \le 0$$
$$g_3(X) = 80.51249 + 0.0071317x_2x_5 + 0.002995x_1x_2 + 0.0021813x_3^2 - 110 \le 0$$
$$g_4(X) = -80.51249 - 0.0071317x_2x_5 - 0.002995x_1x_2 - 0.0021813x_3^2 + 90 \le 0$$
$$g_5(X) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \le 0$$
$$g_6(X) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0$$

where, the bounds are $78 \le x_1 \le 102$, $33 \le x_2 \le 45$, $27 \le x_i \le 45$ ($i = 3,4,5$)

**(5) G05:**

Minimize:

$$f(X) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$$

Subject to:

$$g_1(X) = -x_4 + x_3 - 0.55 \le 0$$
$$g_2(X) = -x_3 + x_4 - 0.55 \le 0$$
$$h_3(X) = 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0$$
$$h_4(X) = 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0$$
$$h_5(X) = 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0$$

where, the bounds are $0 \le x_1 \le 1200$, $0 \le x_2 \le 1200$, $-0.55 \le x_3 \le 0.55$, $-0.55 \le x_4 \le 0.55$

**(6) G06:**

Minimize:

$$f(X) = (x_1 - 10)^3 + (x_2 - 20)^3$$

Subject to:

$$g_1(X) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0$$
$$g_2(X) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0$$

where, the bounds are $13 \le x_1 \le 100$, $0 \le x_2 \le 100$

**(7) G07:**

Minimize:

$$\begin{aligned} f(X) = {} & x_1^2 + x_2^2 - x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 \\ & + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \end{aligned}$$

Subject to:

$$g_1(X) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0$$
$$g_2(X) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0$$
$$g_3(X) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0$$
$$g_4(X) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \le 0$$

$$g_5(X) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0$$
$$g_6(X) = x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0$$
$$g_7(X) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30 \le 0$$
$$g_8(X) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0$$

where, the bounds are $-10 \le x_i \le 10$ ($i = 1,2\ldots,10$)

**(8) G08:**

Minimize:

$$f(X) = -\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$$

Subject to:

$$g_1(X) = x_1^2 - x_2 + 1 \le 0$$
$$g_2(X) = 1 - x_1 + (x_2 - 4)^2 \le 0$$

Where, the bounds are $0 \le x_1 \le 10$, $0 \le x_2 \le 10$

**(9) G09:**

Minimize:

$$f(X) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

Subject to:

$$g_1(X) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0$$
$$g_2(X) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0$$
$$g_3(X) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0$$
$$g_4(X) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0$$

where, the bounds are $-10 \le x_i \le 10$ ($i = 1,2\ldots,7$)

**(10) G10:**

Minimize:

$$f(X) = x_1 + x_2 + x_3$$

Subject to:

$$g_1(X) = -1 + 0.0025(x_4 + x_6) \le 0$$
$$g_2(X) = -1 + 0.0025(x_5 + x_7 - x_4) \le 0$$
$$g_3(X) = -1 + 0.01(x_8 - x_5) \le 0$$
$$g_4(X) = -x_1x_6 + 833.3325x_4 + 100x_1 - 83333.333 \le 0$$
$$g_5(X) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0$$
$$g_6(X) = -x_3x_8 + 125000 + x_3x_5 - 2500x_5 \le 0$$

where, the bounds are $100 \le x_1 \le 10{,}000$, $1000 \le x_i \le 10{,}000$ ($i = 2,3$), $10 \le x_i \le 1000$ ($i = 4, \ldots ,8$)

**(11) G11:**

Minimize:

$$f(X) = x_1^2 + (x_2 - 1)^2$$

Subject to:

$$h(X) = x_2 - x_1^2 = 0$$

where, the bounds are $-1 \le x_1 \le 1$, $-1 \le x_2 \le 1$

**(12) G12:**

Minimize:

$$f(X) = -(100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2)/100$$

Subject to:

$$g(X) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \le 0$$

where, the bounds are $0 \le x_i \le 10$ ($i = 1,2,3$) and $p,q,r = 1,2,\ldots9$.

**(13) G13:**
Minimize:

$$f(X) = e^{x_1 x_2 x_3 x_4 x_5}$$

Subject to:

$$\begin{aligned} h_1(X) &= x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0 \\ h_2(X) &= x_2 x_3 - 5 x_4 x_5 = 0 \\ h_3(X) &= x_1^3 + x_2^3 + 1 = 0 \end{aligned}$$

where, the bounds are $-2.3 \le x_i \le 2.3$ $(i = 1,2)$, $-3.2 \le x_i \le 3.2$ $(i = 3,4,5)$

**(14) G14:**
Minimize:

$$f(X) = \sum_{i=1}^{10} x_i \left( c_i + \ln \frac{x_i}{\sum_{j=i}^{10} x_j} \right)$$

Subject to:

$$\begin{aligned} h_1(X) &= x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0 \\ h_2(X) &= x_4 + 2x_5 + x_6 + x_7 - 1 = 0 \\ h_3(X) &= x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0 \end{aligned}$$

where, the bounds are $0 \le x_i \le 10$ $(i = 1,\ldots,10)$, $c_1 = -6.089$, $c_2 = -17.164$, $c_3 = -34.054$, $c_4 = -5.914$, $c_5 = -24.721$, $c_6 = -14.986$, $c_7 = -24.1$, $c_8 = -10.708$, $c_9 = -26.662$, $c_{10} = -22.179$

**(15) G15:**
Minimize:

$$f(X) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1 x_2 - x_1 x_3$$

Subject to:

$$\begin{aligned} h_1(X) &= x_1^2 + x_2^2 + x_3^2 - 25 = 0 \\ h_2(X) &= 8x_1 + 14x_2 + 7x_3 - 56 = 0 \end{aligned}$$

where, the bounds are $0 \le x_i \le 10$ $(i = 1,2,3)$

**(16) G16:**
Minimize:

$$\begin{aligned} f(X) = {} & 0.000117y_{14} + 0.1365 + 0.00002358y_{13} + 0.000001502y_{16} + 0.0321y_{12} + 0.004324y_5 \\ & + 0.001\frac{c_{15}}{c_{16}} + 37.48\frac{y_2}{c_{12}} - 0.0000005843y_{17} \end{aligned}$$

Subject to:

$$\begin{aligned} g_1(X) &= \frac{0.28}{0.72} y_5 - y_4 \le 0 \\ g_2(X) &= x_3 - 1.5x_2 \le 0 \\ g_3(X) &= 3496\frac{y_2}{c_{12}} - 21 \le 0 \\ g_4(X) &= 110.6 + y_1 - \frac{62212}{c_{17}} \le 0 \\ g_5(X) &= 213.1 - y_1 \le 0 \\ g_6(X) &= y_1 - 405.23 \le 0 \\ g_7(X) &= 17.505 - y_2 \le 0 \\ g_8(X) &= y_2 - 1053.6667 \le 0 \\ g_9(X) &= 11.275 - y_3 \le 0 \\ g_{10}(X) &= y_3 - 35.03 \le 0 \\ g_{11}(X) &= 214.228 - y_4 \le 0 \\ g_{12}(X) &= y_4 - 665.585 \le 0 \\ g_{13}(X) &= 7.458 - y_5 \le 0 \\ g_{14}(X) &= y_5 - 584.463 \le 0 \end{aligned}$$

$$\begin{aligned}
g_{15}(X) &= 0.961 - y_6 \le 0 \\
g_{16}(X) &= y_6 - 265.961 \le 0 \\
g_{17}(X) &= 1.612 - y_7 \le 0 \\
g_{18}(X) &= y_7 - 7.046 \le 0 \\
g_{19}(X) &= 0.146 - y_8 \le 0 \\
g_{20}(X) &= y_8 - 0.222 \le 0 \\
g_{21}(X) &= 107.99 - y_9 \le 0 \\
g_{22}(X) &= y_9 - 273.366 \le 0 \\
g_{23}(X) &= 922.693 - y_{10} \le 0 \\
g_{24}(X) &= y_{10} - 1286.105 \le 0 \\
g_{25}(X) &= 926.832 - y_{11} \le 0 \\
g_{26}(X) &= y_{11} - 1444.046 \le 0 \\
g_{27}(X) &= 18.766 - y_{12} \le 0 \\
g_{28}(X) &= y_{12} - 537.141 \le 0 \\
g_{29}(X) &= 1072.163 - y_{13} \le 0 \\
g_{30}(X) &= y_{13} - 3247.039 \le 0 \\
g_{31}(X) &= 8961.448 - y_{14} \le 0 \\
g_{32}(X) &= y_{14} - 26844.086 \le 0 \\
g_{33}(X) &= 0.063 - y_{15} \le 0 \\
g_{34}(X) &= y_{15} - 0.386 \le 0 \\
g_{35}(X) &= 71084.33 - y_{16} \le 0 \\
g_{36}(X) &= -140000 + y_{16} \le 0 \\
g_{37}(X) &= 2802713 - y_{17} \le 0 \\
g_{38}(X) &= y_{17} - 12146108 \le 0
\end{aligned}$$

where,

$$\begin{aligned}
y_1 &= x_2 + x_3 + 41.6 \\
c_1 &= 0.024x_4 - 4.62 \\
y_2 &= \frac{12.5}{c_1} + 12 \\
c_2 &= 0.0003535x_1^2 + 0.5311x_1 + 0.08705y_2x_1 \\
c_3 &= 0.052x_1 + 78 + 0.002377y_2x_1 \\
y_3 &= \frac{c_2}{c_3} \\
y_4 &= 19y_3 \\
c_4 &= 0.04782(x_1 - y_3) + \frac{0.1956(x_1 - y_3)^2}{x_2} + 0.6376y_4 + 1.594y_3 \\
c_5 &= 100x_2 \\
c_6 &= x_1 - y_3 - y_4 \\
c_7 &= 0.950 - \frac{c_4}{c_5} \\
y_5 &= c_6c_7 \\
y_6 &= x_1 - y_5 - y_4 - y_3 \\
c_8 &= (y_5 + y_4)0.995 \\
y_7 &= \frac{c_8}{y_1} \\
y_8 &= \frac{c_8}{3798} \\
c_9 &= y_7 - \frac{0.0663y_7}{y_8} - 0.3153
\end{aligned}$$

$$
\begin{aligned}
y_9 &= \frac{96.82}{c_9} + 0.321y_1 \\
y_{10} &= 1.29y_5 + 1.258y_4 + 2.29y_3 + 1.71y_6 \\
y_{11} &= 1.71x_1 - 0.452y_4 + 0.580y_3 \\
c_{10} &= \frac{12.3}{752.3} \\
c_{11} &= (1.75y_2)(0.995x_1) \\
c_{12} &= 0.995y_{10} + 1998 \\
y_{12} &= c_{10}x_1 + \frac{c_{11}}{c_{12}} \\
y_{13} &= c_{12} - 1.75y_2 \\
y_{14} &= 3623 + 64.4x_2 + 58.4x_3 + \frac{146312}{y_6 + x_5} \\
c_{13} &= 0.995y_{10} + 60.8x_2 + 48x_4 - 0.1121y_{14} - 5095 \\
y_{15} &= \frac{y_{13}}{c_{13}} \\
y_{16} &= 148000 - 331000y_{15} + 40y_{13} - 61y_{15}y_{13} \\
c_{14} &= 2324y_{10} - 28740000y_2 \\
y_{17} &= 14130000 - 1328y_{10} - 531y_{11} + \frac{c_{14}}{c_{12}} \\
c_{15} &= \frac{y_{13}}{y_{15}} - \frac{y_{13}}{0.52} \\
c_{16} &= 1.104 - 0.72y_{15} \\
c_{17} &= y_9 + x_5
\end{aligned}
$$

where, the bounds are $704.4148 \le x_1 \le 906.3855$, $68.6 \le x_2 \le 288.88$, $0 \le x_3 \le 134.75$, $193 \le x_4 \le 287.0966$, $25 \le x_5 \le 84.1988$

**(17) G17:**

Minimize:

$$f(X) = f(x_1) + f(x_2)$$

where,

$$f_1(x_1) = \begin{cases} 30x_1 & 0 \le x_1 < 300 \\ 31x_1 & 300 \le x_1 < 400 \end{cases}$$

$$f_2(x_2) = \begin{cases} 28x_2 & 0 \le x_2 < 100 \\ 29x_2 & 100 \le x_2 < 200 \\ 30x_2 & 200 \le x_2 < 1000 \end{cases}$$

Subject to:

$$h_1(X) = -x_1 + 300 - \frac{x_3x_4}{131.078}\cos(1.48477 - x_6) + \frac{0.90798x_3^2}{131.078}\cos(1.47588)$$

$$h_2(X) = -x_2 - \frac{x_3x_4}{131.078}\cos(1.48477 + x_6) + \frac{0.90798x_4^2}{131.078}\cos(1.47588)$$

$$h_3(X) = -x_5 - \frac{x_3x_4}{131.078}\sin(1.48477 + x_6) + \frac{0.90798x_4^2}{131.078}\sin(1.47588)$$

$$h_4(X) = 200 - \frac{x_3x_4}{131.078}\sin(1.48477 - x_6) + \frac{0.90798x_3^2}{131.078}\sin(1.47588)$$

where, the bounds are $0 \le x_1 \le 400$, $0 \le x_2 \le 1000$, $340 \le x_3 \le 420$, $340 \le x_4 \le 420$, $-1000 \le x_5 \le 1000$, $0 \le x_6 \le 0.5236$

**(18) G18:**

Minimize:

$$f(X) = -0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7)$$

Subject to:

$$
\begin{aligned}
g_1(X) &= x_3^2 + x_4^2 - 1 \le 0 \\
g_2(X) &= x_9^2 - 1 \le 0 \\
g_3(X) &= x_5^2 + x_6^2 - 1 \le 0
\end{aligned}
$$

$$g_4(X) = x_1^2 + (x_2 - x_9)^2 - 1 \le 0$$
$$g_5(X) = (x_2 - x_5)^2 + (x_2 - x_6)^2 - 1 \le 0$$
$$g_6(X) = (x_2 - x_7)^2 + (x_2 - x_8)^2 - 1 \le 0$$
$$g_7(X) = (x_3 - x_5)^2 + (x_4 - x_6)^2 - 1 \le 0$$
$$g_8(X) = (x_3 - x_7)^2 + (x_4 - x_8)^2 - 1 \le 0$$
$$g_9(X) = x_7^2 + (x_8 - x_9)^2 - 1 \le 0$$
$$g_{10}(X) = x_2x_3 - x_1x_4 - 1 \le 0$$
$$g_{11}(X) = -x_3x_9 \le 0$$
$$g_{12}(X) = x_5x_9 \le 0$$
$$g_{13}(X) = x_6x_7 - x_5x_8 \le 0$$

where, the bounds are $0 \le x_i \le 10$ ($i = 1,2,\ldots,8$), $0 \le x_9 \le 20$

**(19) G19:**

Minimize:

$$f(X) = \sum_{j=1}^{5}\sum_{i=1}^{5} c_{ij}x_{(10+i)}x_{(10+j)} + 2\sum_{j=1}^{5} d_jx_{(10+j)}^3 - \sum_{i=1}^{10} b_ix_i$$

Subject to:

$$g_j(X) = -2\sum_{i=1}^{5} c_{ij}x_{(10+i)} - 3d_jx_{(10+j)}^2 - e_j + \sum_{i=1}^{10} a_{ij}x_i \le 0 \quad j = 1,\ldots,5$$

where, $b = [-40, -2, -0.25, -4, -4, -1, -40, -60, 5, 1]$ and the bounds are $0 \le x_i \le 10$ ($i = 1,2,\ldots,8$). Remaining data is given in the Table below.

| $J$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $e_j$ | −15 | −27 | −36 | −18 | −12 |
| $c_{1j}$ | 30 | −20 | −10 | 32 | −10 |
| $c_{2j}$ | −20 | 39 | −6 | −31 | 32 |
| $c_{3j}$ | −10 | −6 | 10 | −6 | −10 |
| $c_{4j}$ | 32 | −31 | −6 | 39 | −20 |
| $c_{5j}$ | −10 | 32 | −10 | −20 | 30 |
| $d_j$ | 4 | 8 | 10 | 6 | 2 |
| $a_{1j}$ | −16 | 2 | 0 | 1 | 0 |
| $a_{2j}$ | 0 | −2 | 0 | 0.4 | 2 |
| $a_{3j}$ | −3.5 | 0 | 2 | 0 | 0 |
| $a_{4j}$ | 0 | −2 | 0 | −4 | −1 |
| $a_{5j}$ | 0 | −9 | −2 | 1 | −2.8 |
| $a_{6j}$ | 2 | 0 | −4 | 0 | 0 |
| $a_{7j}$ | −1 | −1 | −1 | −1 | −1 |
| $a_{8j}$ | −1 | −2 | −3 | −2 | −1 |
| $a_{9j}$ | 1 | 2 | 3 | 4 | 5 |
| $a_{10j}$ | 1 | 1 | 1 | 1 | 1 |

**(20) G20:**

Minimize:

$$f(X) = \sum_{i=1}^{24} a_ix_i$$

Subject to:

$$g_i(X) = \frac{(x_i + x_{(i+12)})}{\sum_{j=1}^{24} x_j + e_i} \le 0 \quad i = 1,2,3$$

$$g_i(X) = \frac{(x_{(i+3)} + x_{(i+15)})}{\sum_{j=1}^{24} x_j + e_i} \le 0 \quad i = 4,5,6$$

$$h_i(X) = \frac{x_{(i+12)}}{b_{(i+12)}\sum_{j=13}^{24}\frac{x_j}{b_j}} - \frac{c_ix_i}{40b_i\sum_{j=1}^{12}\frac{x_j}{b_j}} = 0 \quad i = 1,\ldots,12$$

$$h_{13}(X) = \sum_{i=1}^{24} x_i - 1 = 0$$

$$h_{14}(X) = \sum_{i=1}^{12} \frac{x_i}{d_i} + k\sum_{i=13}^{24} \frac{x_i}{b_i} - 1.671 = 0$$

where, the bounds are $0 \le x_i \le 10$ ($i$ = 1,2,...,24) and $k = (0.7302)(530)(14.7/40)$. Remaining data set is detailed in Table given below.

| $i$ | $a_i$ | $b_i$ | $c_i$ | $d_i$ | $e_i$ |
|---|---|---|---|---|---|
| 1 | 0.0693 | 44.094 | 123.7 | 31.244 | 0.1 |
| 2 | 0.0577 | 58.12 | 31.7 | 36.12 | 0.3 |
| 3 | 0.05 | 58.12 | 45.7 | 34.784 | 0.4 |
| 4 | 0.2 | 137.4 | 14.7 | 92.7 | 0.3 |
| 5 | 0.26 | 120.9 | 84.7 | 82.7 | 0.6 |
| 6 | 0.55 | 170.9 | 27.7 | 91.6 | 0.3 |
| 7 | 0.06 | 62.501 | 49.7 | 56.708 | – |
| 8 | 0.1 | 84.94 | 7.1 | 82.7 | – |
| 9 | 0.12 | 133.425 | 2.1 | 80.8 | – |
| 10 | 0.18 | 82.507 | 17.7 | 64.517 | – |
| 11 | 0.1 | 46.07 | 0.85 | 49.4 | – |
| 12 | 0.09 | 60.097 | 0.64 | 49.1 | – |
| 13 | 0.0693 | 44.094 | – | – | – |
| 14 | 0.0577 | 58.12 | – | – | – |
| 15 | 0.05 | 58.12 | – | – | – |
| 16 | 0.2 | 137.4 | – | – | – |
| 17 | 0.26 | 120.9 | – | – | – |
| 18 | 0.55 | 170.9 | – | – | – |
| 19 | 0.06 | 62.501 | – | – | – |
| 20 | 0.1 | 84.94 | – | – | – |
| 21 | 0.12 | 133.425 | – | – | – |
| 22 | 0.18 | 82.507 | – | – | – |
| 23 | 0.1 | 46.07 | – | – | – |
| 24 | 0.09 | 60.097 | – | – | – |

**(21) G21:**

Minimize:

$$f(X) = x_1$$

Subject to:

$$g_1(X) = -x_1 + 35x_2^{0.6} + 35x_3^{0.6} \le 0$$
$$h_1(X) = -300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 = 0$$
$$h_2(X) = 100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 25x_4x_7 - 15536.5 = 0$$
$$h_3(X) = -x_5 + \ln(-x_4 + 900) = 0$$
$$h_4(X) = -x_6 + \ln(x_4 + 300) = 0$$
$$h_5(X) = -x_7 + \ln(-2x_4 + 700) = 0$$

where, the bounds are $0 \le x_1 \le 1000$, $0 \le x_2, x_3 \le 1000$, $100 \le x_4 \le 300$, $6.3 \le x_5 \le 6.7$, $5.9 \le x_6 \le 6.4$, $4.5 \le x_7 \le 6.25$

**(22) G22:**

Minimize:

$$f(X) = x_1$$

Subject to:

$$g_1(X) = -x_1 + x_2^{0.6} + x_3^{0.6} + x_4^{0.6} \le 0$$
$$h_1(X) = x_5 - 100000x_8 + 1 \times 10^7 = 0$$
$$h_2(X) = x_6 + 100000x_8 - 100000x_9 = 0$$
$$h_3(X) = x_7 + 100000x_9 - 5 \times 10^7 = 0$$
$$h_4(X) = x_5 + 100000x_{10} - 3.3 \times 10^7 = 0$$
$$h_5(X) = x_6 + 100000x_{11} - 4.4 \times 10^7 = 0$$
$$h_6(X) = x_7 + 100000x_{12} - 6.6 \times 10^7 = 0$$
$$h_7(X) = x_5 - 120x_2x_{13} = 0$$
$$h_8(X) = x_6 - 80x_3x_{14} = 0$$
$$h_9(X) = x_7 - 40x_4x_{15} = 0$$
$$h_{10}(X) = x_8 - x_{11} + x_{16} = 0$$

$$h_{11}(X) = x_9 - x_{12} + x_{17} = 0$$
$$h_{12}(X) = -x_{18} + \ln(x_{10} - 100) = 0$$
$$h_{13}(X) = -x_{19} + \ln(-x_8 + 300) = 0$$
$$h_{14}(X) = -x_{20} + \ln(x_{16}) = 0$$
$$h_{15}(X) = -x_{21} + \ln(-x_9 + 400) = 0$$
$$h_{16}(X) = -x_{22} + \ln(x_{17}) = 0$$
$$h_{17}(X) = -x_8 - x_{10} + x_{13}x_{18} - x_{13}x_{19} + 400 = 0$$
$$h_{18}(X) = x_8 - x_{10} - x_{11} + x_{14}x_{20} - x_{14}x_{21} + 400 = 0$$
$$h_{19}(X) = x_9 - x_{12} - 4.6051x_{15} + x_{15}x_{22} + 100 = 0$$

where, the bounds are $0 \le x_1 \le 1000$, $0 \le x_2, x_3, x_4 \le 1 * 10^6$, $0 \le x_5, x_6, x_7 \le 4 * 10^7$, $100 \le x_8 \le 299.99$, $100 \le x_9 \le 399.99$, $100.01 \le x_{10} \le 300$, $100 \le x_{11} \le 400$, $100 \le x_{12} \le 600$, $0 \le x_{13}, x_{14}, x_{15} \le 500$, $0.01 \le x_{16} \le 300$, $0.01 \le x_{17} \le 400$, $0 \le x_{18}, x_{19}, x_{20}, x_{21}, x_{22} \le 500$

**(23) G23:**

Minimize:

$$f(X) = -9x_5 - 15x_8 + 6x_1 + 16x_2 + 10(x_6 + x_7)$$

Subject to:

$$g_1(X) = x_9x_3 + 0.02x_6 - 0.025x_5 \le 0$$
$$g_2(X) = x_9x_4 + 0.02x_7 - 0.015x_8 \le 0$$
$$h_1(X) = x_1 + x_2 - x_3 - x_4 = 0$$
$$h_2(X) = 0.03x_1 + 0.01x_2 - x_9(x_3 + x_4) = 0$$
$$h_3(X) = x_3 + x_6 - x_5 = 0$$
$$h_4(X) = x_4 + x_7 - x_8 = 0$$

where, the bounds are $0 \le x_1, x_2, x_6 \le 300$, $0 \le x_3, x_4, x_5 \le 100$, $0 \le x_4, x_8 \le 200$, $0.01 \le x_9 \le 0.03$

**(24) G24:**

Minimize:

$$f(X) = -x_1 - x_2$$

Subject to:

$$g_1(X) = -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \le 0$$
$$g_2(X) = -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36 \le 0$$

where, the bounds are $0 \le x_1 \le 3$, $0 \le x_2 \le 4$

## References


[1] B. Akay, D. Karaboga, A modified Artificial Bee Colony algorithm for real-parameter optimization, Inform Sci. 192 (2012) 120–142.
[2] M. Ali, C.W. Ahn, M. Pant, Multi-level image thresholding by synergetic differential evolution, Appl. Soft Comput. 17 (2014) 1–11.
[3] M.Z. Ali, N.H. Awad, A novel class of niche hybrid cultural algorithms for continuous engineering optimization, Inform. Sci. 267 (2014) 158–190.
[4] I. Boussad, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Inform. Sci. 237 (2013) 82–117.
[5] J.C.F. Cabrera, C.A.C. Coello, Handling Constrained in Particle Swarm Optimization Using a Small Population Size, LNAI, Vol. 4827, Springer-Verlag, 2007, pp. 41–51.
[6] R. Caponetto, L. Fortuna, S. Fazzino, M. Xibilia, Chaotic sequences to improve the performance of evolutionary algorithms, IEEE Trans. Evol. Comput. 7 (2003) 289–304.
[7] Y.A. Cengel, Heat Transfer, McGraw-Hill, 2003.
[8] Y.A. Cengel, M.A. Boles, Thermodynamics: An Engineering Approach, McGraw-Hill, 2005.
[9] E. Cuevas, A. Echavarría, M.A. Ramírez-Ortegón, An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation, Appl. Intell. 40 (2) (2014) 256–272.
[10] M. Dorigo, Optimization, Learning and Natural Algorithms, Politecnico di Milano, Italy, 1992 Ph.D. Thesis.
[11] R. Esmaeili, M.R. Dashtbayazi, Modelling and optimization for micro structural properties of Al/SiC nanocomposite by artificial neural network and genetic algorithm, Expert Syst. Appl. 41 (2014) 5817–5831.
[12] J.D. Farmer, N. Packard, A. Perelson, The immune system, adaptation and machine learning, Physica D 22 (1986) 187–204.
[13] W.F. Gao, S.Y. Liu, L.L. Huang, Enhancing artificial bee colony algorithm using more information-based search equations, Inform. Sci. 270 (2014) 112–133.
[14] M. Han, C Liu, J. Xing, An evolutionary membrane algorithm for global numerical optimization problems, Inform. Sci. 276 (2014) 219–241.
[15] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
[16] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (2) (1979) 65–70.
[17] M. Imran, M. Kowsalya, Optimal size and siting of multiple distributed generators in distribution system using bacterial foraging optimization, Swarm Evol. Comput. 15 (2014) 58–65.
[18] G. Jia, Y. Wang, Z. Cai, Y. Jin, An improved $(\mu + \lambda)$-constrained differential evolution for constrained optimization, Inform. Sci. 222 (2013) 302–322.
[19] D. Joaquín, G. Salvador, M. Daniel, H. Francisco, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[20] D. Karaboga, B. Akay, A modified Artificial Bee Colony (ABC) algorithm for constrained optimization problems, Appl. Soft Comput. 11 (2011) 3021–3031.

[21] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (2007) 459–471.
[22] D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems, LNAI, Vol. 4529, Springer-Verlag,, Berlin, 2007, pp. 789–798.
[23] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. 8 (2008) 687–697.
[24] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942–1948.
[25] M. Khajehzadeh, M.R. Taha, A. El-Shafie, M. Eslami, A modified gravitational search algorithm for slope stability analysis, Eng. Appl. Artif. Intell. 25 (8) (2012) 1589–1597.
[26] D.H. Kim, A. Abraham, J.H. Cho, A hybrid genetic algorithm and bacterial foraging approach for global optimization, Inform Sci. 177 (2008) 2937–3918.
[27] H.K. Kim, J.K. Chong, K.Y. Park, D.A. Lowther, Differential evolution strategy for constrained global optimization and application to practical engineering problems, IEEE T Magn. 43 (2007) 1565–1568.
[28] S.S. Kim, J.H. Byeon, H. Yu, H. Liu, Biogeography-based optimization for optimal job scheduling in cloud computing, Appl. Math. Comput. 247 (2014) 266–280.
[29] V. Kumar, J.K. Chhabra, D. Kumar, Automatic cluster evolution using gravitational search algorithm and its application on image segmentation, Eng. Appl. Artif. Intell. 29 (2014) 93–103.
[30] R.J. Kuo, S.Y. Hung, W.C. Cheng, Application of an optimization artificial immune network and particle swarm optimization-based fuzzy neural network to an RFID-based positioning system, Inform. Sci. 262 (2014) 78–98.
[31] X.L. Li, X.D. He, A hybrid particle swarm optimization method for structure learning of probabilistic relational models, Inform Sci. 283 (2014) 258–266.
[32] J.J. Liang, T.P. Runarsson, E. Mezura-Montes, M. Clerc, P.N. Suganthan, C.A.C. Coello, K. Deb, Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session On Constrained Real-Parameter Optimization, Technical Report, Nanyang Technological University, Singapore, 2006, http://www.ntu.edu.sg/home/EPNSugan.
[33] S.H. Liu, M. Mernik, B. Bryant, To explore or to exploit: an entropy-driven approach for evolutionary algorithms, Int. J. Knowl. Base. Intell. Eng. Syst. 13 (3) (2009) 185–206.
[34] F.S. Lobato, M.N. Sousa, M.A. Silva, A.R. Machado, Multi-objective optimization and bio-inspired methods applied to machinability of stainless steel, Appl. Soft Comput 22 (2014) 261–271.
[35] H. Ma, D. Simon, Blended biogeography-based optimization for constrained optimization, Eng. Appl. Artif. Intell. 24 (3) (2011) 517–525.
[36] H. Ma, D. Simon, M. Fei, X. Shu, Z Chen, Hybrid biogeography-based evolutionary algorithms, Eng. Appl. Artif. Intel. 30 (2014) 213–224.
[37] A. Mũnoz-Zavala, A.A. Hernández, E.R.V. Diharce, Constrained optimization via particle evolutionary swarm optimization algorithm, in: Proceedings of the Genetic and Evolutionary Computation Conference, vol. 1, ACM Press, New York, 2005, pp. 209–216.
[38] Q.K. Pan, Y. Dong, An improved migrating birds optimisation for a hybrid flow shop scheduling with total flow time minimisation, Inform. Sci. 277 (2014) 643–655.
[39] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. Mag. 22 (2002) 52–67.
[40] V.K. Patel, V.J. Savsani, Optimization of a plate-fin heat exchanger design through an improved multi-objective teaching-learning based optimization (MO-ITLBO) algorithm, Chem. Eng. Res. Des 92 (2014) 2371–2382.
[41] V.K. Patel, V.J. Savsani, A multi-objective improved teaching–learning based optimization algorithm (MO-ITLBO), Inform. Sci. DOI: 10.1016/j.ins.2014.05.049
[42] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag, Berlin, 2005.
[43] B.Y. Qu, J.J. Liang, P.N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization, Inform. Sci. 197 (2012) 131–143.
[44] A. Rahmani, S.A. MirHassani, A hybrid Firefly-Genetic Algorithm for the capacitated facility location problem, Inform. Sci 283 (2014) 70–78.
[45] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems, Inform. Sci. 183 (1) (2011) 1–15.
[46] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. Aid. Des 43 (3) (2011) 303–315.
[47] R.V. Rao, V.K Patel, Optimization of rotary regenerator using artificial bee colony algorithm, J. Power Energy 225 (6) (2011) 1088–1098.
[48] R.V. Rao, V. Patel, An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems, Int. J. Ind. Eng. Comput. 3 (4) (2012) 535–560.
[49] R.V. Rao, V.K Patel, Multi-objective optimization of combined Brayton and inverse Brayton cycle using advanced optimization algorithms, Eng. Optim 44 (8) (2012) 965–983.
[50] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA, a gravitational search algorithm, Inform. Sci. 179 (2009) 2232–2248.
[51] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (2008) 702–713.
[52] D. Simon, A probabilistic analysis of a simplified biogeography-based optimization algorithm, Evol. Comput. 19 (2011) 167–188.
[53] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[54] R. Ugolotti, Y.S.G. Nashed, P. Mesejo, Š. Ivekovič, L. Mussi, S. Cagnoni, Particle swarm optimization and differential evolution for model-based object detection, Appl. Soft Comput. 13 (6) (2013) 3092–3105.
[55] J. Xiao, L.P. Li, A hybrid ant colony optimization for continuous domains, Exp. Syst. Appl 38 (2011) 11072–11077.
[56] A.R. Yildiz, Optimization of cutting parameters in multi-pass turning using artificial bee colony-based approach, Inform. Sci 220 (2013) 399–407.
[57] M. Zhang, W. Luo, X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, Inform. Sci 178 (2008) 3043–3074.
[58] Z.H. Zhan, J. Zhang, Y. Li, Y.H. Shi, Orthogonal learning particle swarm optimization, IEEE Trans Evol. Comput. 15 (2011) 832–847.
[59] Z. Zhang, N. Zhang, Z. Feng, Multi-satellite control resource scheduling based on ant colony optimization, Expert Syst. Appl 41 (2014) 2816–2823.
[60] H. Zhu, Y. Wang, K. Wang, Y. Chen, Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem, Expert Syst. Appl 38 (8) (2011) 10161–10169.