Engineering with Computers
DOI 10.1007/s00366-016-0457-y

ORIGINAL ARTICLE

Engineering optimization based on ideal gas molecular movement algorithm

Hesam Varaee[1] · **Mohammad Reza Ghasemi**[1]

Received: 24 February 2016 / Accepted: 20 May 2016

Abstract The present work introduces a new metaheuristic optimization method based on the ideal gas molecular movement (IGMM) to solve mathematical and engineering optimization problems. Ideal gas molecules scatter throughout the confined environment quickly. This is embedded in the high speed of molecules, collisions between them and with the surrounding barriers. In IGMM algorithm, the initial population of gas molecules is randomly generated and the governing equations related to the velocity of gas molecules and collisions between those are utilized to accomplish the optimal solutions. To verify the performance of the IGMM algorithm, some mathematical and engineering benchmark optimization problems, commonly used in the literature, are inspected. Comparison of results obtained by IGMM with other optimization algorithms show that the proposed method has a challenging capacity in finding the optimal solutions and exhibits significance both in terms of the accuracy and reduction on the number of function evaluations vital in reaching the global optimum.

Keywords Optimization · Metaheuristic algorithm · Ideal gas molecular movement · Engineering optimization problems · Constrained benchmark functions

1 Introduction

Determination of optimal solutions for engineering design problems in a rational time has always been a major concern to engineers and designers. Hence, enormous complexity and high computational costs of engineering problems has led optimization methods to be constantly promoted.

In general, the optimization techniques are classified into two categories: mathematical programming and metaheuristic methods. The former utilizes gradient information to search the optimal solutions from an initial starting point. These methods are mainly used to continuous problems and may well cease operation after finding a local optimum. On the other hand, major challenges of many engineering optimization problems lay on their side limits, prohibited zones, and non-smooth or non-convex cost functions which need to be looked upon. They cannot, therefore, be solved easily by the traditional mathematical programming methods [1].

Regardless of the complexity of a problem; however, metaheuristic algorithms inspired by nature have been developed to find global optimum solutions. Considering their ability to explore and find promising areas within an affordable time period, they can be used to carry out global searches and reduce the need for continuous cost functions and variables unlike former methods [2, 3]. They are approximate methods that lead to suitable but not necessarily the precise optimum point. Moreover, these methods do not rely on differentiation of the objective function and variables [4].

Some of the novel metaheuristic algorithms are artificial bee colony algorithm (ABC) [5], cuckoo optimization algorithm (CO) [6], ant lion optimizer (ALO) [7], krill herd

✉ Mohammad Reza Ghasemi
mrghasemi@hamoon.usb.ac.ir

Hesam Varaee
hesam.varaee@pgs.usb.ac.ir

[1] Department of Civil Engineering, University of Sistan and Baluchestan, Zahedan, Iran

Published online: 04 June 2016

(KH) [8, 9], earthworm optimization algorithm (EWA) [10], gravitational search algorithm (GSA) [11], monarch butterfly optimization (MBO) [12], elephant herding optimization (EHO) [13, 14], and the grey wolf optimizer (GWO) [15]. Many successful applications of metaheuristic algorithms in engineering optimization problems have been reviewed by researchers [16–22].

According to the “no free lunch theorem” [23], there is no optimization technique better than all the others on all problems. So, there is a need to explore further studies to discover more novel and effective optimization algorithms, not that the authors of the current work claim for such a triumph.

The main goal of this paper is introduce a new optimization algorithm based on the movement of ideal gas molecules and collisions between them, referred to as the ideal gas molecular movement (IGMM) optimization algorithm. Gas molecules have an effective velocity that is related to their mass and temperature. To use the velocity equation in IGMM algorithm, an initial temperature is defined and set for each molecule. Besides, the mass of each molecule is determined in relation to its fitness. In the early stages of optimization procedure, where comprehensive rapid searches in the problem domain occur, the temperature is set to a large value. Gradually, as further calculations take place and optimum regions are exposed, molecules velocities drop with a decrease in their temperatures. Moving gas molecules collide with one another with a certain probability. As a result of the collision caused by distribution of the existing energy, each molecule gains a new velocity and position. The collision probability increases as the optimization procedure continues, molecules accumulate and neighboring points to the global optimum are formed, where it eventually leads to finding the global optimum point.

The paper is organized as follows. Section 2 describes the molecular movement and behavior of ideal gases to build the required background. In Sect. 3, different steps of the IGMM algorithm are described. Some well-known mathematical and engineering benchmark problems are studied in Sects. 4 and 5, respectively. Finally, in Sect. 6, findings of the current research are presented, discussed, and compared where appropriate.

2 Ideal gas

The kinetic theory of gases provides a microscopic model for studying the molecular behavior and thermodynamic properties of gases. In this model, an ideal gas has the following properties [24, 25]:

1. A trivial segregated volume of an ideal gas may contain a large number of gas molecules are spread uniformly inside the container, moving steadily along random paths.
2. No external forces are assumed exerted on the system. Molecules apply no forces on each other unless they collide.
3. The impact between molecules and the isolated container walls is fully elastic and causes no reduction in the kinetic energy of molecules. They may change direction as a result of the collision.
4. The probability of distribution velocity is assumed equal in any direction for all molecules.

The proposed IGMM algorithm is thus based on the above general principles. In the following sub-sections, the required relations to establish such algorithm are extracted by investigating the physical relations and governing equations related to the behavior of gases.

2.1 Ideal gas governing equations

The governing relations of ideal gases show that pressure (P), volume (V), and temperature (T) are the fundamental parameters influencing the behavior of ideal gases [25, 26]. According to the ideal gas principle, these parameters are linked through Eq. (1):

$$PV = nRT = NkT \tag{1}$$

where n is the number of system particles moles, R is the universal gas constant, N denotes the number of gas molecules, and k indicates the Boltzmann constant. To use this relation in the SI system, P is expressed in Pascal, V in cubic meters, and N as a dimensionless number. Moreover, $R = 8.314472\,\mathrm{J\,K^{-1}\,mol^{-1}}$, T is expressed in Kelvin, and $k = R/N_A$ is set to $1.38 \times 10^{-23}\,\mathrm{J\ K^{-1}}$. N_A is the Avogadro constant and is equal to $6.0221 \times 10^{+23}\,\mathrm{mol^{-1}}$ [24].

Assume that a container encloses a large number of molecules of an ideal gas that simultaneously move in different directions. The average velocity of molecules is obtained through vector sum of velocities in different directions.

In the case, gas molecule hits the container wall, its momentum changes based on the elastic collision assumption. Therefore, the total force exerted on the wall can be determined as follows:

$$F_t = N_c \times f_{ave} = N_c \times \frac{2mv_x}{t_c} = \frac{N_c}{t_c} 2mv_x \tag{2}$$

In Eq. (2), N_c stands for the number of particles hitting the wall. f_{ave} denotes the average force exerted on the container wall by a particle. t_c, and m show the time and mass of the particle during the collision, respectively. v_x is the x component of particles’ velocity. According to the fourth law of ideal gases described in Sect. 2, statistics states that at any increment of time, half of the particles move to the right with a positive momentum and the other half move to the left with a negative momentum, inside the container. Thus,

$$N_c = \frac{1}{2}N_m\left(\frac{V_c}{V_b}\right) = \frac{1}{2}N_m\left(\frac{Ad_c}{V_b}\right) = \frac{1}{2}\left(\frac{N_m}{V_b}\right)Av_xt_c \tag{3}$$

where N_m denotes the number of molecules inside the container, V_c shows the volume swept by a molecule during the collision, V_b indicates the volume of the entire container, and A stands for the collision surface. The total force exerted to the collision surface can be calculated by placing the number of collisions computed in Eq. (3) into Eq. (2), the results of which provide that:

$$F_t = \left(\frac{\frac{1}{2}\left(\frac{N_m}{V_b}\right)Av_xt_c}{t_c}\right)2mv_x = \left(\frac{1}{2}\left(\frac{N_m}{V_b}\right)Av_x\right)2mv_x \tag{4}$$

Having simplified the force in Eq. (4) and divided it by the area then, the equation for the pressure to the wall is obtained in terms of the x component of particles' velocities as given by Eq. (5):

$$P = \frac{F}{A} = \left(\frac{N_m}{V_b}\right)mv_x^2. \tag{5}$$

However, since the average velocity of the particles is the vector sum of their components in different directions and since statistically the probability of all particles distribution velocities is the same, the average velocity equation can be written as follows:

$$v_{ave}^2 = 3v_x^2 = v_{rms}^2. \tag{6}$$

Note that in Eq. (6). the average velocity is turned into the root-mean-square term. Therefore,

$$\frac{1}{3}v_{rms}^2 = v_x^2. \tag{7}$$

By substituting v_x^2 from Eq. (7) into Eq. (5) for the pressure, we have:

$$\frac{1}{3}\left(\frac{N_m}{V_b}\right)mv_{rms}^2 = \frac{2}{3}\left(\frac{N_m}{V_b}\right)\left(\frac{1}{2}mv_{rms}^2\right) = \frac{2}{3}\left(\frac{N_m}{V_b}\right)K_E. \tag{8}$$

The ideal gas law can be used to indicate that the average kinetic energy (K_E) can be expressed in terms of the gas temperature. All these equations can be summarized as:

$$\frac{2}{3}\left(\frac{N_m}{V_b}\right)K_E = \frac{nRT}{V}. \tag{9}$$

Thus,

$$K_E = \frac{3}{2}\left(\frac{nR}{N}\right)T = \frac{3}{2}\left(\frac{R}{N_A}\right)T = \frac{3}{2}kT. \tag{10}$$

Finally, a relation is obtained to calculate the effective velocity of each molecule in the container based on its mass and temperature:

$$\frac{1}{2}mv_{rms} = \frac{3}{2}kT. \tag{11}$$

Therefore,

$$v_{rms} = \sqrt{v^2} = \sqrt{\frac{3kT}{m}} \approx 1.7\sqrt{\frac{kT}{m}}. \tag{12}$$

Equation (12) may form the basis for calculating the velocity of molecules in the optimization algorithm.

2.2 Collision

In a set of gas molecules, the interaction between molecules takes place through collisions. Assuming a diluted gas or a gas with few molecules per volume, the concepts required for picturing the collision processes will be simple. Rationally, it is assumed that molecules occupy an insignificant time during collision incidence. In addition, it is assumed that only one collision takes place at a time for a molecule. Based on the above assumptions and definitions, such as the mean free path and the mean successive collision time, the relation for the probability of collisions between gas molecules is introduced as in Eq. (13) followed, where physical relations [27] have been used to derive it.

$$P(t) = 1 - e^{-0.63t} \tag{13}$$

where $P(t)$ shows the collision probability of gas molecules with respect to time. Binary collision theory will be used to determine the collision velocity of gas molecules. Thus, it is assumed that molecules are in the form of rigid spheres that collide fully elastically with each other without any attractive intermolecular forces acting on them. A fully elastic collision is referred to as a remained invariant energy before and after the collision. It only occurs in microscopic mediums, such as ideal gases, where a certain amount of kinetic energy remains invariant after the collision. In a non-elastic collision; however, part of the collision energy is turned into other forms of energy.

In general, the governing relations expressing the momentum conservation may be defined through the following equations:

$$m_1v_1 + m_2v_2 = m_1v_1' + m_2v_2'. \tag{14}$$

However, if assumed that one of the molecules is motionless and is hit with another molecule, the term m_2v_2 is dropped from the above relation.

In general, the invariant total kinetic energy is shown as follows:

$$\frac{1}{2}m_1v_1^2 + \frac{1}{2}m_2v_2^2 = \frac{1}{2}m_1v_1'^2 + \frac{1}{2}m_2v_2'^2 + Q \tag{15}$$

where v_1 and v_2 show the initial velocities of the first and second molecules before the impact, and accordingly, v_1' and v_2' denote their final velocities after the impact, respectively. Moreover, m_1 shows the mass of the first and m_2 the second molecules. In general, the lost amount of energy at the time of the impact is represented by Q [28]. Based on the above relations, it is possible to calculate the velocity of molecules after a one-dimensional collision using the following equations.

$$v_1' = \frac{(m_1 - Em_2)v_1 + (m_2 + Em_2)v_2}{m_1 + m_2} \tag{16}$$

$$v_2' = \frac{(m_2 - Em_1)v_2 + (m_1 + Em_1)v_1}{m_1 + m_2} \tag{17}$$

where E is the modulus of elasticity for the two colliding molecules.

Since according to the elastic collision hypotheses, kinetic energy is not lost as a result of collision or it is not turned into other forms of energy, we have, therefore, $E = 1$ and $Q = 0$. In a non-elastic collision, however, the modulus of elasticity is no longer equal to 1 ($Q \neq 0$ and $E \leq 1$).

3 Proposed algorithm for ideal gas molecular movement

Gas molecules may be resembled to some potential agents, moving and exploring rapidly all the volume inside the container as an optimization domain, searching all the potential regions for a global optimum solution. They interact and exchange information with one another after each collision, as they travel inside the region. The exchange of information is explored as a change in the velocity of molecules after they collide and move to new positions in different time steps. The main idea of the IGMM algorithm is to find the global optimum during these navigations and collisions. The optimization procedure is described in the following section.

3.1 Progressive steps of the IGMM algorithm

3.1.1 Initialization

The first characteristic of ideal gases, discussed in Sect. 2, plays its role in generating the initial population of the algorithm. As the first characteristic, a number of molecules spread uniformly and move randomly in different paths. This characteristic reinforces the idea of using a uniform distribution for generating the initial population. It is expected to accelerate the search process towards the optimum solution. When the initial population of solutions is selected randomly with a uniform distribution from the allowable range of design variables, the initial temperature of 273 K is set in the velocity equation.

3.1.2 Calculation of molecular masses

In the IGMM algorithm, a comparative measure was implemented to normalize agent's fitness by defining a mass parameter in accordance with the fitness of each gas molecule. Equation related to the velocity of gas molecules reveals the fact that there is an inverse relationship between the velocity of particles and their masses. Thus, accordingly, the molecule with a larger mass has a lower velocity and vice versa. Hence, the mass equation should be defined such that fitter molecules move at a lower velocity. This could be justified to their tendency to remain in the current region which is more likely to be feasible. This is naturally contrary for less fit molecules.

Since this research is focused on minimizing the objective, based on the principle discussed, each solution is assigned a mass according to its fitness using the following relation:

$$m_i = \frac{1}{\mathit{fit}(i)} \bigg/ \frac{1}{\sqrt{\sum \left(\mathit{fit}^2(i)\right)}} \tag{18}$$

where m_i is the mass of the ith molecule and $\mathit{fit}(i)$ reflects the fitness of the i-th molecule with regard to the objective function for the problem. The elite molecule will be found through comparison of molecular masses in every iteration.

3.1.3 Calculating molecules collision probability

Physical governing equations of gas molecules expressed in Sect. 2.2 state that ideal gas molecules collide with one another with a certain specific probability. That is, it increases with an increase in the travelling time and the distance travelled by the molecules. To utilize this characteristic in the optimization process, a new parameter named molecules collision probability (MCP) is defined. At the beginning of the optimization process, when the problem domain is not known and no exchange of information has taken place between the agents, the MCP is at its lowest level. However, as further processed through optimization and the optimum region is found, collision probability grows exponentially according to the following relation and reaches its highest level.

$$\mathrm{MCP} = 1 - \exp\left(-0.63 \times \mathit{iter}\right). \tag{19}$$

3.1.4 Calculating molecules new velocity and position

Every two molecules either collide with one another based on their specific collision probability or keep moving noncollided based on the velocity relation. According to this phenomenon, the following steps will proceed to calculate the new velocity and position of each molecule.

3.1.4.1 Calculating molecular velocity at the collision incidence (MVCI) The second and third characteristics of ideal gases reflect the molecular interactions. The equations (Eqs. (16) and (17)) may be used to determine the new positions after collisions. When two particles collide, their new post-collision velocities are obtained using Eqs. (16) and (17). In using these equations, the molecule with a larger mass is assumed stationary and the lighter molecule moves according to the hypotheses about the elastic collision between molecules. In addition, the velocity of the moving molecule is obtained using relation $\Delta x = v\Delta t$ (for $\Delta t = 1$) by subtracting the positions of the two molecules. Thus,

$$(v_1^d)' = \frac{(m_1 - Em_2)}{m_1 + m_2} \times v_1^d \tag{20}$$

$$(v_2^d)' = \frac{(1 + E)m_1}{m_1 + m_2} \times v_1^d \tag{21}$$

where v_1^d shows the initial velocity of the first molecule before the impact, while assumed $v_2^d = 0$, and accordingly, $(v_1^d)'$ and $(v_2^d)'$ denote final velocities after the impact, respectively, d indicates the dimension of the optimization problem. As stated, in the event of elastic collisions, parameter E is equal to 1, but in the above equation, this parameter is defined as a variable to guarantee the convergence in the algorithm. Therefore, in the first few steps of the optimization process, this variable has a value near 1 but with an increase in the number of optimization cycles its value declines dynamically based on the following linear equation.

$$E = 1 - \left(\frac{iter}{maxIt}\right) \tag{22}$$

where *iter* and *maxIt* indicate current and the maximum iterations of optimization procedure, respectively. Having computed the new velocity of each molecule, its new position can be computed using Eqs. (23) and (24):

$$(x_1^d)' = x_2^d + rand \times (v_1^d)' \tag{23}$$

$$(x_2^d)' = x_2^d + rand \times (v_2^d)' \tag{24}$$

where x_2^d shows the position of stationary molecule before the impact and accordingly, $(x_1^d)'$ and $(x_2^d)'$ indicate the new positions after the impact, respectively. *rand* represents a random normal distributed value in the range [0,1].

3.1.4.2 Calculating no-collision molecular velocity (NCMV) In the isolated media of ideal gases, if assumed no molecular collision occurs, the new velocity of the ith molecule is determined using Eq. (25).

$$(v_i^d)' = 1.7\sqrt{\frac{kT_i}{m_i}} \tag{25}$$

Since the Boltzmann constant has an inverse relationship with the number of molecules, its value in Eq. (25) is set to $k = 1/nVar$, where $nVar$ is the number of molecules in the optimization process.

Velocity of each molecule accords with the mass and temperature of that molecule. Hence, in this phase, it is necessary to calculate the new temperature of each molecule. To do so, a subtractive equation is defined as follows:

$$T_i' = T_i - 1/m_i. \tag{26}$$

After determining the new velocity of each molecule, the new position can be obtained using Eq. (27):

$$(x_i^d)' = x_i^d + rand \times (v_i^d)'. \tag{27}$$

3.1.5 Convergence criteria

In the last step of optimization, the convergence is checked. The optimization procedure is set completed, if the best result in the current iteration hardly shows any changes within an allowable tolerance for some iterations. Another acceptable convergence criterion is where the distance between the best solution and the average penalized objective value of all designs in an iteration reduces to a desired value. Extra commonly measured stopping criterion in metaheuristic algorithms is where the maximum number of iterations is reached without convergence occurs. This is where the optimization procedure may terminate without a rational convergence.

The pseudo code and schematic flowchart of the IGMM algorithm are shown in Figs. 1 and 2, respectively.

To validate the efficiency of the IGMM, some mathematical problems, extracted from literature, were attempted. These problems are investigated in Sect. 4. For further examination, in Sect. 5, some well-known engineering and truss design problems are also studied adapted from the literature.

4 Mathematical optimization problems

To evaluate the performance of the proposed algorithm, the eight standard benchmark functions of Table 1 [29] were examined for optimization. Typically, they can be divided into three categories: (1) unimodal functions, (2) multimodal functions, and (3) multimodal functions with fix

Fig. 1 Pseudo code of the IGMM algorithm

```
Initialize the first population of molecules randomly and set the initial temperature
Calculate the fitness of molecules
Calculate the mass of each molecule using the Eq. (18)
Find the best molecule and assume it as the elite (determined optimum)
while the end criterion is not satisfied
    Pair the molecules randomly (with no repetition)
    for every pair
    Calculate MCP, T and E
        If rand < MCP (molecules will be colliding)
            Set $v_2^d = 0$ and calculate $v_1^d$, $(v_1^d)'$ and $(v_2^d)'$ using Eqs. (20) and (21).
        else (molecules with no collision)
            Set $K = 1/nVar$
            Calculate $(v_i^d)'$ using the Eq. (25)
        end if
    Update the position of molecules using Eqs. (23), (24) or (27)
    Calculate the fitness of molecules
    Calculate the mass of each molecule using the Eq. (18)
    Update elite if a molecule becomes fitter than the elite
    end for
end while
Return elite
```

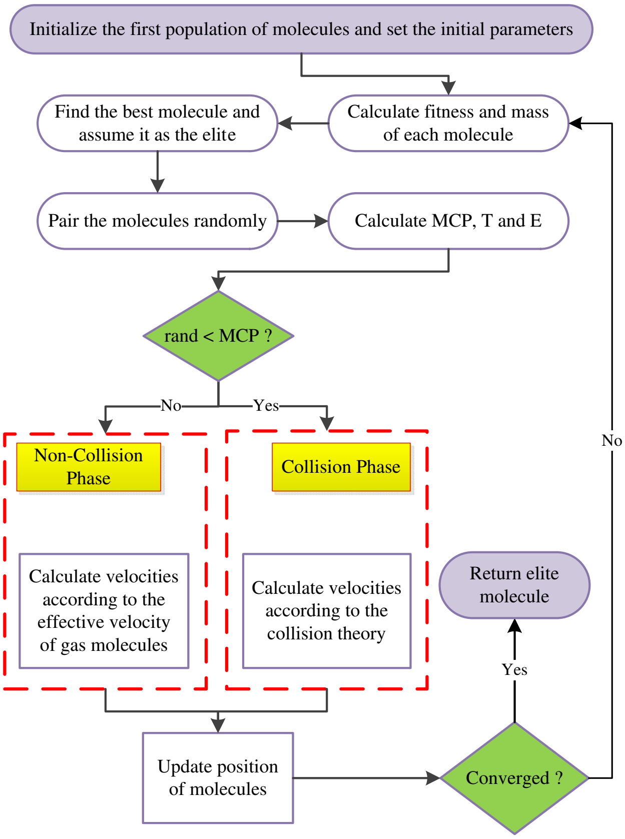
Initialize the first population of molecules and set the initial parameters
Find the best molecule and assume it as the elite
Calculate fitness and mass of each molecule
Pair the molecules randomly
Calculate MCP, T and E
rand < MCP ?
No
Yes
Non-Collision Phase
Collision Phase
Calculate velocities according to the effective velocity of gas molecules
Calculate velocities according to the collision theory
Return elite molecule
Yes
No
Update position of molecules
Converged ?

Fig. 2 Schematic flowchart of IGMM method

dimension. In Table 1, n denotes the function dimension and S is a subset of R^n. In all cases, the population size is kept constant as 50. Besides, the maximum number of iterations for the functions F_1 to F_6 is set to 1000, while the maximum number of iterations for multi modal functions with fix dimension is 500. For providing the stochastic behavior of metaheuristic algorithms, the number of independent runs is chosen as 25. The IGMM algorithm was coded in MATLAB script language using a computer device with an Intel Core i7 CPU, 3.5 GHz processor, and 8.00 GB RAM.

The performance of IGMM on the eight optimization problems attempted was compared with five well-known optimization algorithms, such as genetic algorithm (GA) [30], particle swarm optimization (PSO) [31], harmony search (HS) [32], artificial bee colony (ABC) [5], and biogeography-based optimization (BBO) [33] algorithm. The results of the six algorithms, including IGMM, are recorded in Table 2.

The setting of the parameters in those five methods is exploited from references [34–36], and may be detailed as follows. In GA, the single-point crossover, Gaussian mutation, and Roulette wheel selection techniques are employed. The probabilities of the crossover and mutation are fixed to 1 and 0.01, respectively. In PSO, for swarm interaction [36], an inertial constant of 0.6, cognitive and social constants equal to 2 each were set. In HS, the probability of selecting a value from a total list of corresponding values, stored in the memory as harmony memory consideration rate (HMCR), is set to 0.9, and the probability of transformation of the new value into a neighboring value in the harmony memory or pitch adjustment rate (PAR) is defined as 0.1 [34]. In ABC, the limit is set to the number of generation, and number of employed and onlooker bees is set to 25, as suggested in [35], and for BBO, we used a habitat modification probability = 1, immigration probability bounds per gene = [0, 1], step size for numerical integration of probabilities = 1, maximum immigration and migration rates for each island = 1, and mutation probability = 0.005 [34].

For the best solutions as listed in Table 2, IGMM has relatively the strongest search ability and outperforms the best results in all metrics for F_1 to F_4. For F_5 and F_6, IGMM can find the fittest response, while ABC has shown

Table 1 Mathematical benchmark functions

Test function	n	S	$f_{\min}$	Type
$F_1(X) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$	0	Unimodal
$F_2(X) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]^n$	0	Unimodal
$F_3(X) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	$[-100, 100]^n$	0	Unimodal
$F_4(X) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]^n$	0	Multimodal
$F_5(X) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600, 600]^n$	0	Multimodal
$F_6(X) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right\}$	30	$[-50, 50]^n$	0	Multimodal
$F_7(X) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}, (a_{ij}) = \begin{pmatrix} -32, -16, 0, 16, 32, -32, \ldots, 0, 16, 32 \\ -32, -32, -32, -32, -16, \ldots, 32, 32, 32 \end{pmatrix}$	2	$[-65.53, 65.53]^2$	0.998	Multimodal with fix dimension
$F_8(X) = \left[1 + (x_1 + x_2 + 1)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \times \left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	2	$[-5, 5]^2$	3	Multimodal with fix dimension

a better performance on mean, worst, and std. metrics. For multimodal benchmarks of F_7 and F_8 with fix dimensions, the performance of the six methods (except BBO for F_7) show a little difference. Moreover, the convergence histories of F_2, F_5, and F_6 run by the above six methods are illustrated in Fig. 3. It is clear that IGMM has the better performance than other algorithms during the whole optimization process.

5 Engineering optimization problems

In this section, validity of IGMM in dealing with five real-world constrained benchmark problems and two truss design problems will be investigated, and a comparison of results with those available in the literature will be presented. The convergence measure applied for engineering optimization problems is based on the proximity of the fittest design in the current iteration with that of 20 iterations before. Thus, if the difference between these two values is less than a small allowable tolerance value, it is recorded as converged. The desired value for real-world constrained benchmarks is set to 10^{-5} and for the truss design problems is set to 10^{-3}. If not converged, the algorithm will be terminated by implementing a maximum number of iterations set fixed.

5.1 A general constraint-handling approach introduced

Engineering systems are mostly constrained. Different constraint-handling methods have been utilized to handle linear and non-linear inequality constraints in evolutionary algorithms.

Coello [37] uses a penalty function for penalizing infeasible points. There, the problem is converted into an unconstrained one through penalizing the infeasible points, as a result of which a single objective function will be built and minimized as an unconstrained optimization problem.

Penalty functions can be classified into stationary and non-stationary main groups. A stationary group uses fixed penalty values throughout the minimization, whereas a non-stationary penalty group modifies the penalty values dynamically. As studied by [38, 39], results obtained using non-stationary penalty functions are virtually always superior-to-stationary functions.

Here, however, the IGMM-based technique uses a multi-stage non-stationary penalty functions with a slight modification. Thus, it states that:

$$F(x) = f(x) + h(i)H(x), \quad x \in S \subset R^n \tag{28}$$

where $f(x)$ is the original objective function; $h(i)$ is a dynamically modified penalty value, i is the algorithm current iteration number; and $H(x)$ is a penalty factor, defined as:

Table 2 Results of the benchmark functions over 25 independent runs

Function	Metrics	BBO	HS	PSO	GA	ABC	IGMM
F_1	Best	4.55E−02	84.4662	7.87E−05	0.1566	2.66E−09	*1.75E−12*
	Mean	5.48E−02	109.8388	2.00E+03	0.5373	9.61E−09	*5.87E−12*
	Worst	6.19E−02	131.9033	1.00E+04	0.9917	2.92E−08	*9.77E−12*
	Std.	6.78E−03	17.4986	4.47E+03	0.3335	1.13E−08	*3.01E−12*
F_2	Best	6.83E−02	2.4696	10.0003	0.0928	1.21E−06	*5.38E−09*
	Mean	7.85E−02	2.6843	38.0001	0.2289	2.55E−06	*1.82E−08*
	Worst	9.39E−02	3.0229	80	0.3933	3.46E−06	*2.73E−08*
	Std.	1.04E−02	0.2131	25.8843	0.1104	8.46E−07	*8.49E−09*
F_3	Best	3.12E−02	112.5021	5.86E−05	0.5357	1.00E−09	*1.78E−12*
	Mean	4.12E−02	133.6642	2.20E−04	0.7287	6.20E−09	*2.98E−11*
	Worst	5.56E−02	163.458	5.56E−04	0.8814	1.11E−08	*1.22E−10*
	Std.	1.16E−02	18.7392	2.17E−04	0.1459	4.57E−09	*5.19E−11*
F_4	Best	3.31E−02	3.3704	0.0085	0.2819	8.69E−06	*1.37E−07*
	Mean	5.34E−02	3.644	10.809	0.3274	1.84E−05	*1.11E−06*
	Worst	7.10E−02	4.2429	19.963	0.3928	2.73E−05	*3.15E−06*
	Std.	1.40E−02	0.3449	10.1376	0.0455	8.23E−06	*1.18E−06*
F_5	Best	7.32E−02	1.7265	0.0019	0.2771	3.26E−10	*7.40E−12*
	Mean	1.17E−01	2.0526	0.0126	0.4135	*1.02E−03*	4.93E−03
	Worst	1.91E−01	2.2491	0.0273	0.558	*5.12E−03*	1.23E−02
	Std.	4.42E−02	0.2131	0.0095	0.108	*2.29E−03*	6.75E−03
F_6	Best	9.05E−04	12.4891	0.0177	0.0172	3.00E−09	*9.97E−12*
	Mean	3.56E−03	17.7458	0.2666	0.0357	*1.39E−08*	1.28E−03
	Worst	7.21E−03	22.2295	1.0114	0.0595	*4.10E−08*	6.22E−03
	Std.	2.91E−03	4.0145	0.4228	0.0168	*1.63E−08*	2.76E−03
F_7	Best	6.9033	*0.998*	*0.998*	*0.998*	*0.998*	*0.998*
	Mean	10.9186	*0.998*	*0.998*	*0.998*	*0.998*	*0.998*
	Worst	18.3043	*0.998*	*0.998*	*0.998*	*0.998*	*0.998*
	Std.	4.6759415	7.00E−10	*1.11E−16*	5.20E−13	*1.11E−16*	3.40E−13
F_8	Best	*3*	3	*3*	*3*	3	*3*
	Mean	*3*	3.0001	*3*	*3*	3.0005	*3*
	Worst	*3*	3.0002	*3*	*3*	3.0019	*3*
	Std.	1.27E−06	6.90E−05	4.44E−16	1.49E−15	7.96E−04	*2.22E−16*

The best results are marked in italics

$$H(x) = \sum_{i=1}^{m} \theta(q_i(x)) q_i(x)^{\gamma(q_i(x))}. \tag{29}$$

The function $q_i(x) = \{0, g_i(x)\}, i = 1, 2, 3, \ldots, m$, is a relative violated function of the constraints; $\theta(q_i(x))$ is a multi-segment assignment function; $\gamma(q_i(x))$ is a power of the penalty function; and $g_i(x)$ are the constraints. In this paper, the penalty parameters are applied similar to that of [40].

5.2 Real engineering design problems

In this section, some engineering design problems are presented to show the validity and efficiency of the IGMM. It includes five real engineering design problems, some of which have been recently recorded in the collected works by others using different algorithms.

Twenty independent runs for each problem are carried out. Number of molecules was fixed to 50 for each run along a maximum 200 iterations. The number of maximum function evaluations is, therefore, summed to 10,000.

5.2.1 *Optimum design of welded beam*

The design of the welded beam shown in Fig. 4, as a standard constrained engineering problem, was introduced by Rao [41]. It is a practical problem that has been often used as benchmark to evaluate performance of different optimization techniques [4, 15, 37, 42–47]. The design variables are the thickness of the weld (w), the length of the welded joint (L), the width of the beam (d), and the thickness of the beam (h). The purpose of this example is to find the best set of design variables to minimize the overall construction

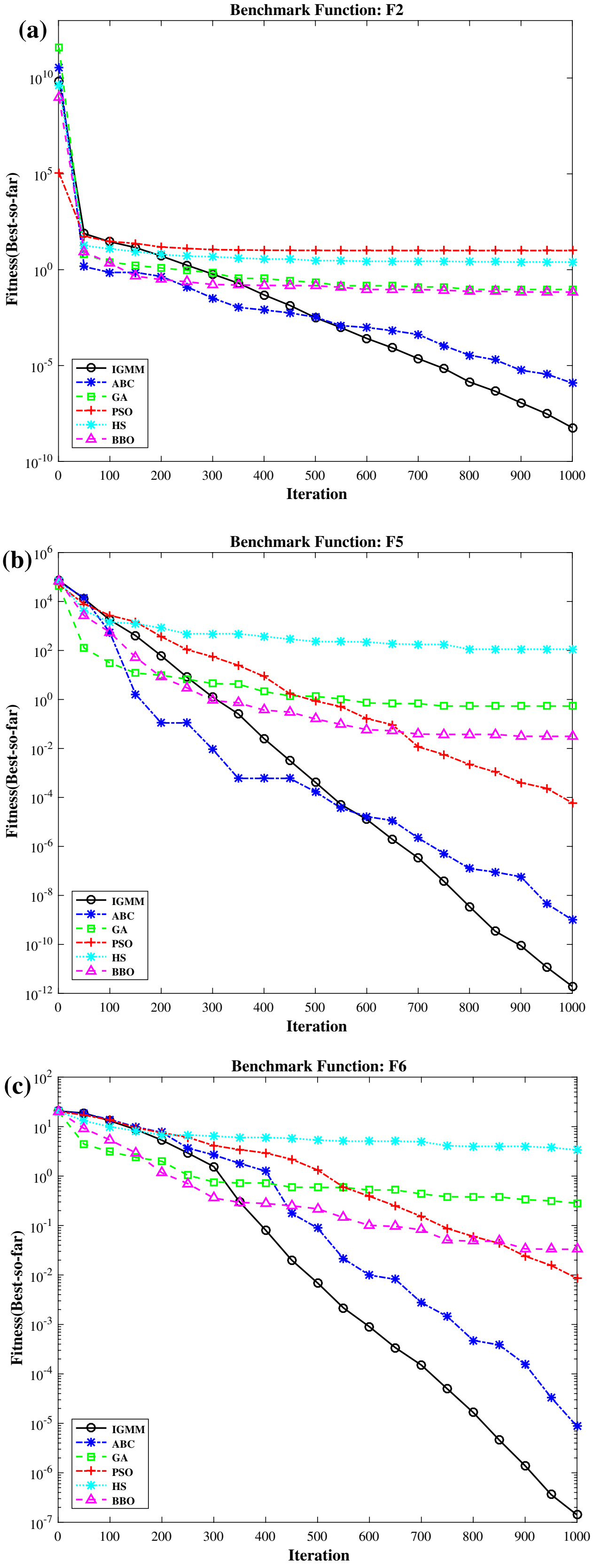

Fig. 3 Comparison of IGMM, GA, PSO, HS, ABC, and BBO for minimization of F_2, F_5 and F_6

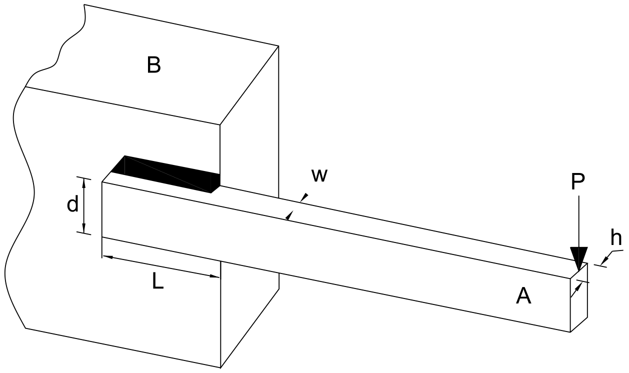

Fig. 4 Welded beam problem with four design variables as shown

cost of the structure under the defined constraints. The constraints are shear stress (τ), bending stress (σ), buckling load (P), and maximum end deflection (δ). The mathematical formulation of the problem is expressed as follows:

$$\min f(w, L, d, h) = 1.1047w^2L + 0.04811dh(14.0 + L). \tag{30}$$

Subject to:

$$\begin{aligned}
g_1 &= \tau - 13600 \le 0,\\
g_2 &= \sigma - 30000 \le 0,\\
g_3 &= w - h \le 0,\\
g_4 &= 0.1047w^2 + 0.04811hd(14 + L) - 0.5 \le 0,\\
g_5 &= 0.125 - w \le 0,\\
g_6 &= \delta - 0.25 \le 0,\\
g_7 &= 6000 - P \le 0,
\end{aligned} \tag{31}$$

where

$$\begin{aligned}
\sigma &= \frac{504000}{hd^2},\\
Q &= 6000\left(14 + \frac{L}{2}\right),\\
D &= \frac{1}{2}\sqrt{(L^2 + (w + d)^2},\\
J &= \sqrt{2}wL\left[\frac{L^2}{6} + \frac{(w + d)^2}{2}\right],\\
\delta &= \frac{65856}{30000hd^3},\\
\beta &= \frac{QD}{J},\\
\alpha &= \frac{6000}{\sqrt{2}wL},\\
\tau &= \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2},\\
P &= 0.61423 \times 10^6\frac{dh^3}{6}\left(1 - \frac{d\sqrt{30/48}}{28}\right).
\end{aligned} \tag{32}$$

The lower and upper bounds on the design variables are:

$$
\begin{aligned}
&0.1 \le L, d \le 10.0,\\
&0.1 \le w, h \le 2.0.
\end{aligned}
\tag{33}
$$

Radgsdell and Phillips [42] compared different optimization techniques which were mainly based on the mathematical optimization methods. Coello [44] and Deb [43] presented an optimum solution for this problem using the GA-based approaches. In addition, He and Wang [46] solved this problem using CPSO, while Montes and Coello [45] used evolution strategies. Kaveh and Talatahari [4] employed an improved ACO and Mirjalili et al. used grey wolf optimizer (GWO) [15] and multi-verse optimizer (MVO) [47].

Table 3 compares the results obtained by IGMM with those presented in the literature for this problem. As depicted in Table 3, results exhibit a better search ability of IGMM comparatively in most of the evaluations. The computational cost has also been reduced significantly, a desirable feature in solving engineering and real-world optimization problems.

The statistical data on 20 independent runs presented in Table 4 show that IGMM has the lowest fabrication cost among other optimization algorithms. In addition, the lowest standard deviation achieved by IGMM may indicate that the proposed algorithm is more robust than other techniques.

5.2.2 Optimum design of a pressure vessel

As the second example, the objective is to minimize the total fabrication cost of a cylindrical pressure vessel, as shown in Fig. 5. This problem has also been popular among researchers. Many heuristic techniques have been used to optimize this problem, such as GA [54], CPSO [46], DE [45], PSO [46], and GWO [15]. The problem involves two discrete and two continuous variables and four inequality constraints. The design variables are the shell thickness (T_s), the spherical head thickness (T_h), the radius of cylindrical shell (R), and the shell length (L). The shell and head thicknesses should be multiples of 0.0625 in and within the range 1×0.0625 and 99×0.0625 in. The radius of cylindrical shell and the shell length are limited between 10 and 200 in. The mathematical model for the problem can be stated as follows:

$$
\begin{aligned}
\min f(T_s, T_h, R, L) = {} & 0.6224 T_s R L + 1.7781 T_h R^2 \\
& + 3.1661 T_s^2 L + 19.84 T_s^2 R.
\end{aligned}
\tag{34}
$$

Subject to:

$$
\begin{aligned}
&g_1 = -T_s + 0.0193R \le 0,\\
&g_2 = -T_h + 0.0095R \le 0,\\
&g_3 = -\pi R^2 L - \frac{4}{3}\pi R^3 + 1{,}296{,}000 \le 0,\\
&g_4 = L - 240 \le 0.
\end{aligned}
\tag{35}
$$

The optimal solution and constraint values obtained by IGMM were compared with those in the literature and are presented in Table 5.

As shown in Table 5, IGMM achieved better results for the pressure vessel design problem compared to those based on GA, CPSO, DE, and PSO. From the table, the recorded optimum design based on PSO algorithm is not, however, feasible, where the third constraint has been slightly violated. In addition, it was found that the correct value of the fabrication cost for the recorded best design by GWO is 6061.0135. The statistical data on various independent runs compared with the results by others in the literature are listed in Table 6.

The statistical data presented in Table 6 indicate that the standard deviation of IGMM-based optimum solutions is the lowest among those presented by other researchers, showing its robustness in comparison with other optimization techniques listed in Table 6.

Table 3 Comparison of optimum solutions for welded beam design problem

	GWO [15]	GA [44]	GA [45]	CPSO [46]	MVO [47]	ACO [4]	Present work
w	0.205676	0.2088	0.205986	0.202369	0.205463	0.2057	0.205729
L	3.478377	3.4205	3.471328	3.544214	3.473193	3.471131	3.470496
d	9.03681	8.9975	9.020224	9.04821	9.044502	9.036683	9.036625
h	0.205778	0.21	0.20648	0.205723	0.205695	0.205731	0.205730
g_1	−21.234	−0.3378	−0.1030	−13.6555	NA	−0.0846	−0.00627
g_2	−8.2556	−353.902	−0.2317	−75.8141	NA	−0.5907	−0.01478
g_3	−1.02E−04	−0.0012	−4.94E−04	−0.0034	NA	−3.10E−05	−3.19E−07
g_4	−3.4319	−3.4119	−3.43	−3.4246	NA	−3.4329	−3.43298
g_5	−0.0807	−0.0838	−0.0810	−0.0774	NA	−0.0807	−0.08073
g_6	−0.2355	−2.2355	−0.2355	−0.2356	NA	−0.2355	−0.23554
g_7	−4.3135	−363.2324	−586469	−4.4729	NA	−0.1448	−0.00504
$f_{\min}$	1.72624	1.748309	1.728226	1.728024	1.72645	1.724918	1.724853
NFEs	NA	900,000	80,000	30,000	15,000	17,600	8000

Table 4 Statistical results of different approaches for welded beam design problem

	Best	Mean	Worst	Std.	NFEs
Mirjalili et al. [15]	1.72624	NA	NA	NA	NA
Coello [48]	1.748309	1.771973	1.785835	0.01122	900,000
Radgsdell and Phillips (DAVID) [42]	2.3841	NA	NA	NA	NA
Radgsdell and Phillips (APPROX) [42]	2.3815	NA	NA	NA	NA
Radgsdell and Phillips (SIMPLEX) [42]	2.5307	NA	NA	NA	NA
Radgsdell and Phillips (RANDOM) [42]	4.1185	NA	NA	NA	NA
Coello [44]	1.748309	1.771973	1.785835	0.011220	900,000
Coello and Montes [45]	1.728226	1.792654	1.993408	0.074713	80,000
He and Wang [46]	1.728024	1.748831	1.782143	0.012926	30,000
Kaveh and Talatahari [1]	1.724866	1.739654	1.759479	8.06E−03	4000
Eskandar et al. [49]	1.724856	1.726427	1.744697	4.29E−03	46,450
Deb [43]	2.433116	NA	NA	NA	NA
Coello and Becerra [50]	1.724852	1.971809	3.179707	4.43E−01	50,020
Liu and Cai [51]	1.724852	1.749040	1.724852	6.7E−16	66,600
Coello [44]	1.824500	1.919000	1.995000	5.37E−02	NA
Zahara and Kao [52]	1.724852	1.724852	1.733393	3.50E−03	80,000
Lampinen [53]	1.733461	1.768158	1.824105	2.21E−02	204,800
Montes and Coello [45]	1.737300	1.813290	1.994651	0.070500	200,000
Mirjalili et al. [47]	1.72645	NA	NA	NA	15,000
Kaveh and Talatahari [4]	1.724918	1.729752	1.775961	9.2E−03	17,600
Present work	1.724853	1.732152	1.74769	7.14E−03	8000

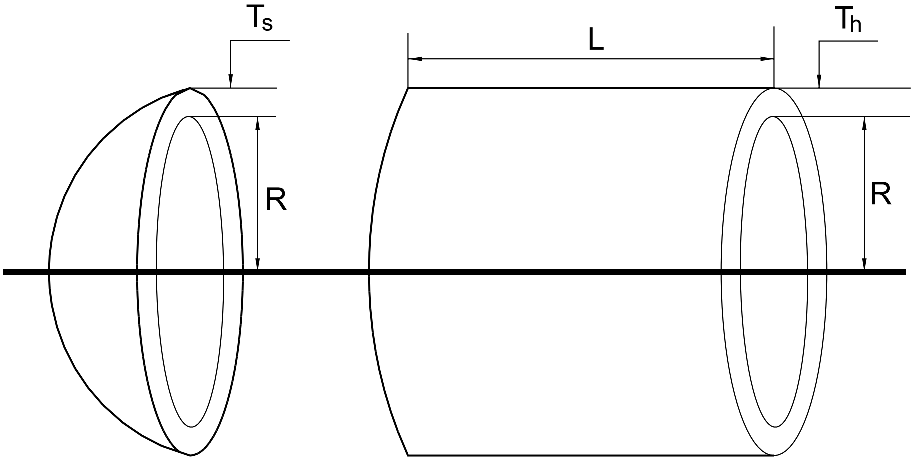

Fig. 5 Pressure vessel design problem with four design variables

Table 5 Comparison of the best solution for pressure vessel design problem

	GA [55]	CPSO [46]	DE [48]	PSO [56]	GWO [15]	Present work
T_s	0.8125	0.8125	0.8125	0.8125	0.8125	0.8125
T_h	0.4375	0.4375	0.4375	0.4375	0.4345	0.4375
R	42.0974	42.0913	40.3239	42.0984	42.0892	42.098445
L	176.6540	176.7465	200.00	176.6366	176.759	176.63659
g_1	−0.00002	−1.37E−06	−0.03425	−8.80E−07	−1.788E−04	−3.1764E−10
g_2	−0.03589	−3.59E−04	−0.05442	−0.0359	−0.0377	−0.03756
g_3	−27.8861	−118.7687	−304.4	*3.1227*[a]	−40.6168	−0.00012
g_4	−63.3459	−63.2535	−40	−63.3634	−63.2413	−63.3634
$f_{\min}$	6059.9463	6061.0777	6288.744	6059.7143	6051.5639[b]	6059.7143
NFEs	80,000	240,000	NA	60,000	NA	8000

[a] Italic value is a violated constraint

[b] True value is 6061.0135

Table 6 Statistical results of different approaches for pressure vessel design problem

	Best	Mean	Worst	Std.	NFEs
Gao and Hailo [56]	6059.7143	6066.0311	NA	12.2718	60,000
Lee and Geem [57]	7198.433	NA	NA	NA	NA
Mirjalili et al. [15]	6051.5939	NA	NA	NA	NA
Coello [48]	6288.745	6293.843	6308.15	7.4133	900,000
Sandgren [58]	7980.894	NA	NA	NA	NA
Kannan and Kramer [59]	7198.042	NA	NA	NA	NA
Akhtar [60]	6171	6335.05	6453.65	NA	NA
Yun [61]	7198.424	NA	NA	NA	NA
Tsai et al. [62]	7079.037	NA	NA	NA	NA
Coelho [63]	6059.721	6440.379	7544.493	448.4711	8000
Cao and Wu [64]	7108.616	NA	NA	NA	NA
Deb [65]	6410.381	NA	NA	NA	NA
Coello [66]	6228.744	NA	NA	NA	NA
Sandgren [67]	8129.104	NA	NA	NA	NA
Coelho [63]	6059.714	NA	NA	NA	8000
Zhang and Wang [68]	7197.7	NA	NA	NA	NA
Coello and Cortes [69]	6061.123	6734.085	7368.06	457.9959	NA
Homaifar et al. [70]	6295.11	8098.03	9528.07	831.69	NA
Gandomi [71]	6059.71	6410.08	7332.84	384.6	5000
Joines and Houck [39]	6273.28	8092.87	10382.1	1017.99	NA
Mirjalili [72]	6059.7143	NA	NA	NA	NA
Michalewicz and Attia [73]	6572.62	8164.56	9580.51	789.65	NA
Gandomi et al. [74]	6059.71	6179.13	6318.95	137.22	375,000
Hadj-Alouane and Bean [75]	6303.5	8065.66	10569.65	821.3	NA
Fu et al. [76]	8048.6	NA	NA	NA	NA
Li and Chou [77]	7127.3	NA	NA	NA	NA
Sadollah et al. [78]	5889.32	6200.64	6392.50	160.35	70,650
Cai and Thierauf [79]	7006.931	NA	NA	NA	NA
He et al. [80]	6059.714	6289.929	NA	305.78	NA
Coello and Montes [55]	6059.946	6177.253	6469.322	130.9267	80,000
Cao and Wu [81]	7108.616	NA	NA	NA	NA
Hu et al. [82]	6059.131	NA	NA	NA	NA
Huang and Wang [83]	6059.734	6085.23	6371.046	43.013	204,800
Zahara and Kao [52]	5930.314	5946.79	5960.056	9.1614	80,000
He and Wang [46]	6061.078	6147.133	6363.804	86.4545	240,000
Litinetski and Amramzon [84]	7197.7	NA	NA	NA	NA
Wu and Chow [85]	7207.494	NA	NA	NA	NA
Lee and Geem [86]	7198.433	NA	NA	NA	NA
Cagnina et al. [87]	6059.714	NA	NA	NA	42,000
Mirjalili et al. [47]	6060.8066	NA	NA	457.9959	15,000
Gandomi et al. [88]	6059.714	6447.736	6495.347	502.693	NA
Coello [37]	6177.253	NA	NA	130.9297	NA
Ray and Liew [89]	6171	6335.05	NA	NA	NA
Montes et al. [90]	6059.702	6059.702	NA	0	24,000
Parsopoulos and Vrahatis [91]	6544.27	9032.55	11638.2	995.573	100,000
Shih and Lai [92]	7462.1	NA	NA	NA	NA
Li and Chang [93]	7127.3	NA	NA	NA	NA
Kaveh and Talatahari [4]	6059.73	6081.78	6150.13	67.2418	NA
Present work	6059.7143	6060.1598	6061.2868	0.5421	8000

5.2.3 *Optimum design of a tension/compression spring*

The tension/compression spring problem was first introduced by Belegundu [94] and Arora [95] as shown in Fig. 6. The aim for this problem is to minimize the weight of the tension/compression spring subject to surge frequency, shear stress, and minimum deflection. There are three design variables in this problem: wire diameter (d), mean coil diameter (w), and the number of active coils (N). The problem can be stated as follows:

$$\min f(w, d, N) = (N + 2)w^2 d. \tag{36}$$

Subject to:

$$\begin{aligned} g_1 &= 1 - \frac{d^3 N}{71785 w^4} \le 0, \\ g_2 &= \frac{d(4d - w)}{12566 w^3 (d - w)} + \frac{1}{5108 w^2} - 1 \le 0, \\ g_3 &= 1 - \frac{140.45 w}{d^2 N} \le 0, \\ g_4 &= \frac{2(w + d)}{3} - 1 \le 0. \end{aligned} \tag{37}$$

The bounds on the design variables are:

$$0.05 \le w \le 2.0, 0.25 \le d \le 1.3, 2.0 \le N \le 15.0. \tag{38}$$

The problem has been solved by Belegundu [94] using eight different mathematical optimization methods. Arora [95] also found the optimum results using a numerical optimization approach with a constraint correction at the constant cost. Coello and Montes [37] utilized a GA-based method. In addition, He and Wang [46] solved this problem using a co-evolutionary particle swarm optimization (CPSO). Recently, Eskandar et al. [49] and Kaveh and Talatahari [1] used water cycle algorithm (WCA) and the charged system search (CSS) to find the optimum results, respectively. Tables 7 and 8 compare the best results obtained by IGMM with those recorded by other researches.

The standard deviations of optimum cost reported in Table 8 prove the consistency of IGMM with those in the literature.

5.2.4 *Cantilever beam design problem*

Figure 7 shows a cantilever beam consisting of five hollow square blocks. There is also a vertical load applied to the free end of the beam while the other side of the beam is rigidly supported. The aim is to minimize the weight of the beam, while the vertical displacement is defined as a constraint that should not be violated by the final optimal design. The design variables are the heights (or widths) of the different hollow blocks with fixed thickness ($t = 2/3$) [105].

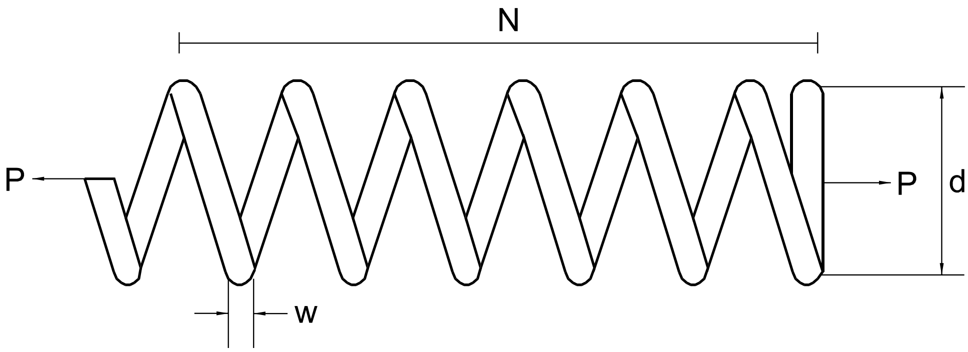

Fig. 6 Tension/compression spring design problem indicating also the design variables

Table 7 Comparing the best solutions for the tension/compression spring design problem using different approaches

	MPM [94]	GA [37]	WCA [49]	CC [95]	CPSO [46]	Present work
w	0.50	0.051989	0.051689	0.053396	0.051728	0.051718
d	0.3159	0.363965	0.356522	0.399180	0.357644	0.357415
L	14.25	10.890522	11.30041	9.185400	11.244543	11.2482
g_1	−0.000014	−0.000013	−1.65E−13	−0.053396	−0.000845	−7.87E−08
g_2	−0.003782	−0.000021	−7.90E−14	−0.000018	−1.26E−05	−3.77E−08
g_3	−3.938302	−1.061338	−4.053399	−4.123832	−4.051300	−4.05516
g_4	−0.756067	−0.722698	−0.727864	−0.698283	−0.727090	−1.09087
$f_{\min}$	0.0128334	0.0126810	0.012665	0.0127303	0.0126747	0.01266525
NFEs	NA	80,000	11,750	NA	NA	4000

Table 8 Statistical results of different methods for tension/compression spring optimum design problem

	Best	Mean	Worst	Std.	NFEs
Belegundu and Arora [94]	0.0128334	NA	NA	NA	NA
Mirjalili et al. [15]	0.012666	NA	NA	NA	NA
Coello and Montes [37]	0.012681	0.012742	0.012973	5.90E−05	80,000
Krohling et al. [96]	0.012665	0.012746	0.012952	8.06E−06	240,000
Arora [95]	0.0127303	NA	NA	NA	NA
Kaveh and Talatahari [1]	0.0126384	0.012852	0.013626	8.3564E−05	4000
Mahdavi et al. [97]	0.0126706	NA	NA	NA	NA
Kaveh and Mahdavi [98]	0.0126697	0.0127296	0.128808	5.0038E−05	4000
He and Wang [46]	0.0126747	0.01273	0.012924	5.20E−05	200,000
Mirjalili [72]	0.0126669	NA	NA	NA	30,000
Coello [48]	0.012704	0.012769	0.012822	3.94E−05	900,000
Coello and Becerra [50]	0.012721	0.013568	0.015116	8.42E−04	50,020
He and Wang [99]	0.012665	0.012707	0.012719	1.58E−05	81,000
Zahara and Kao [52]	0.01263	0.012631	0.012633	8.47E−07	80,000
Coelho [63]	0.012665	0.013524	0.017759	0.001268	2000
Liu and Cai [51]	0.012857	0.019555	0.071802	0.011662	2000
Wang and Li [100]	0.012665	0.012665	0.012665	1.3E−07	20,000
Lampinen [53]	0.01267	0.012703	0.01279	2.7E−05	204,800
Zhang and Luo [101]	0.012665	0.012669	0.012738	1.3E−05	24,000
Wang et al. [102]	0.012665	0.012665	0.012665	1.4E−09	24,000
Ray and Liew [89]	0.012669	0.012922	0.016717	5.9E−04	25,167
Liu and Cai [51]	0.012665	0.012665	0.012665	1.2E−08	24,950
Karaboga and Basturk [103]	0.012665	0.012709	NA	0.012813	30,000
Eskandar et al. [49]	0.012665	0.012746	0.012952	8.06E−05	11,750
Montes and Coello [104]	0.012689	0.013165	NA	3.9E−04	30,000
Present work	0.0126653	0.0128657	0.0135125	2.56E−04	4000

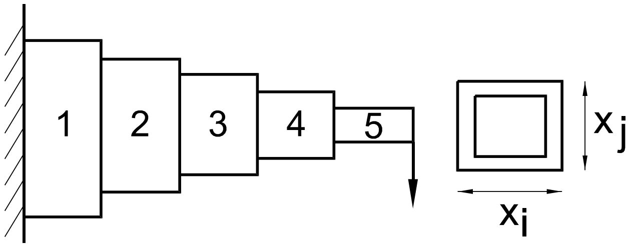

Fig. 7 Cantilever beam design problem also indicating the design variables

Based on the discretization of five elements, the optimization problem was formulated by Svanberg [106] in a closed form:

$$\min f(x) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5). \tag{39}$$

Subject to:

$$g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \le 1. \tag{40}$$

The bounds on the design variables are:

$$0.01 \le x_1, x_2, x_3, x_4, x_5 \le 100. \tag{41}$$

Having solved the problem using IGMM, the results were recorded in Table 9, where a comparison is made between method of moving asymptotes (MMA) [105], generalized convex approximation (GCA_I) [105], GCA_II [105], moth-flame optimization (MFO) algorithm [72], and symbiotic organisms search (SOS) [107]. It shows that the proposed algorithm exhibits a better performance compared to other algorithms. The statistical data on various independent runs compared with the results by others in the literature are listed in Table 10.

5.2.5 Gear train design problem

The next problem as shown in Fig. 8 is a discrete case study with four parameters. The objective is to find the optimal number of tooth for four gears of a train for minimizing the gear ratio [108]. The mathematical formulation of this problem is as follows:

$$\min f(n_A, n_B, n_C, n_D) = \left(\frac{1}{6.931} - \frac{n_A n_B}{n_C n_D}\right)^2. \tag{42}$$

Table 9 Comparison results for cantilever design problem

	SOS [107]	MMA [105]	GCA-I [105]	GCA-II [105]	MFO [72]	Present work
x_1	6.01878	6.0100	6.0100	6.0100	5.98487	6.026098781
x_2	5.30344	5.3000	5.3040	5.3000	5.31673	5.29213389
x_3	4.49587	4.4900	4.4900	4.4900	4.49733	4.495163027
x_4	3.49896	3.4900	3.4980	3.4900	3.51362	3.505163246
x_5	2.15564	2.1500	2.1500	2.1500	2.16162	2.155279835
g	*0.000139*	NA	NA	NA	NA	−2.9447E−06
$f_{\min}$	1.33996	1.3400	1.3400	1.3400	1.339988	1.3399675
NFEs	15000	NA	NA	NA	30,000	7000

Italic sets are violated constraint

Table 10 Statistical results using different approaches for cantilever design problem

	Best	Mean	Worst	Std.	NFEs
Cheng and Prayogo [107]	1.33996	NA	NA	NA	15,000
Chickermane and Gea [105]	1.34	NA	NA	NA	NA
Chickermane and Gea [105]	1.34	NA	NA	NA	NA
Chickermane and Gea [105]	1.34	NA	NA	NA	NA
Mirjalili et al. [72]	1.33998	NA	NA	NA	30,000
Mirjalili et al. [47]	1.33996	NA	NA	NA	15,000
Present work	1.339967	3.3401	3.3404	3.48E−06	7000

Subject to:

$$12 \leq n_A, n_B, n_C, n_D \leq 60. \tag{43}$$

The best optimal design recorded by IGMM and other algorithms available in the literature are presented in Table 11.

Table 11 shows that the proposed IGMM algorithm finds an optimal design identical to those of ant lion optimization (ALO) [7], artificial bee colony (ABC) algorithm [109], MBA [78], and CS [88]. However, the maximum numbers of function evaluations presented in Table 11 shows that the IGMM requires much less computational cost to determine the global optimum against GA, ABC, MBA, and CS. Besides, these results prove that the proposed IGMM algorithm can also solve discrete real problems efficiently

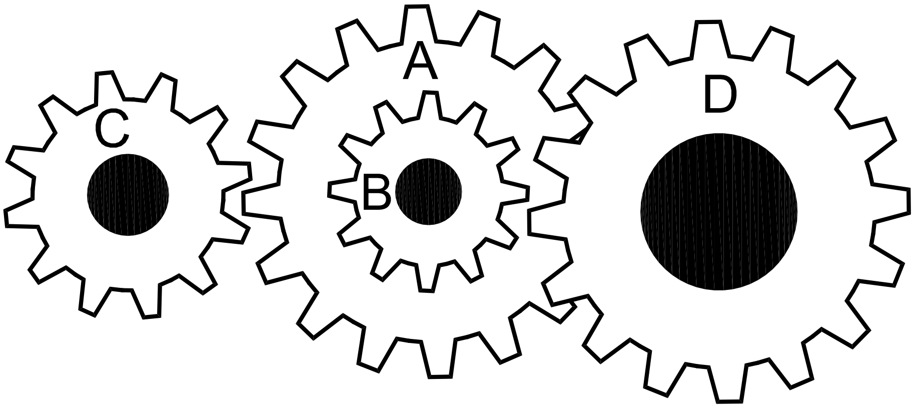

Fig. 8 Gear train design problem

Table 11 Comparison results for gear train design problem

	GA [85]	GA [108]	ABC [109]	MBA [78]	CS [88]	Present work
x_1	NA	33	19	43	43	43
x_2	NA	14	16	16	16	16
x_3	NA	17	44	19	19	19
x_4	NA	50	49	49	49	49
$f_{\min}$	2.33E−07	1.362E−09	2.78E−11	2.7009E−12	2.7009E−12	2.7009E−12
NFEs	10,000	NA	40,000	10,000	5000	480

Table 12 Statistical results of different approaches for gear train design problem

	Best	Mean	Worst	Std.	NFEs
Deb and Goyal [108]	1.362E−09	NA	NA	NA	NA
Sadollah et al. [78]	2.7009E−12	NA	NA	NA	10,000
Mirjalili [7]	2.7009E−12	NA	NA	NA	120
Sharma et al. [109]	2.78E−11	NA	NA	NA	40,000
Sandgran [58]	5.712E−05	NA	NA	NA	NA
Kannan and Kramer [59]	2.146E−08	NA	NA	NA	NA
Parsopoulos and Vrahatis [91]	2.7009E−12	NA	NA	NA	100,000
Hossein et al. [88]	2.7009E−12	NA	NA	NA	5000
Mirjalili et al. [72]	2.7009E−12	NA	NA	NA	30,000
Kannan and Kramer [59]	2.1469e−08	NA	NA	NA	NA
Wu and Chow [85]	2.33E−07	NA	NA	NA	10,000
Gandomi [71]	2.7009E−12	8.050E−08	8.95E−07	2.00E−07	200
Litinetski and Abramzon [84]	2.7009E−12	NA	NA	NA	5100
Present work	2.7009E−12	6.25E−10	2.36E−09	8.61E−10	480

with low computational cost. The statistical data of different approaches for gear train design problem are listed in Table 12.

The results recorded for these three engineering problems may prove the suitability of IGMM algorithm in solving problems with unknown search spaces. The results also show that the proposed algorithm is appropriate for constrained and discrete problems.

5.3 Truss design problems

Truss optimization is a challenging area of structural optimization. For most truss optimization problems, the objective function can be expressed as:

$$\min W(A) = \sum_{i=1}^{NM} \gamma_i A_i L_i \tag{44}$$

where $W(A)$ indicates the weight of the structure; NM is the number of members; γ_i shows the material density of member i; L_i indicate the length of member i; and A_i is the cross-sectional area of member i chosen between $A_{\min}$ and $A_{\max}$ (the lower and upper bounds, respectively) for the structure.

In the truss design problems, 20 independent runs are attempted, each with 40 molecules, where a maximum of 500 iterations were allowed.

Usually adjusting parameters in optimization algorithms have a significant impact on their performance and convergence process. As the appropriate values of these parameters can be various for different problems, sensitivity analysis and trial and error process should necessarily apply to set them. Since this paper is not seeking to determine the best values for the adjusting parameters, the sensitivity analysis will be ignored and the physical-based relations are used instead.

5.3.1 *10-bar planar truss problem*

The 10-bar truss of Fig. 9 has been investigated by many researchers. In this example, the stress limits are $\sigma_0 = \pm 25\,\text{ksi}$ (± 172 MPa) and the cross-sectional area for each element is limited to a minimum value of 0.1 in.^{2} (0.6451 cm^2). The material density is 0.1 lb/in.^{3} (2767.990 kg/m^3) and the modulus of elasticity is 10 Msi (68.950 GPa). Vertical displacement limits of $\pm 2.0\,\text{in}$. (± 5.08 cm) are imposed to all joints. Two distinct loading cases are considered for this example. In the first case, the structure is under a single loading condition with $P_1 = 100\text{kips}$ (444.82 kN), $P_2 = 0$. In the second case, a single loading condition is applied as $P_1 = 150\,\text{kips}$ (667.233 kN), $P_2 = 50\,\text{kips}$ (222.411 kN).

In Tables 13 and 14, $\delta_{\max}$ and $\sigma_{\max}$ are the maximum deflection of nodes and maximum stress in elements, respectively. Although the result reported in [57] for this example is lighter than other designs, the displacement

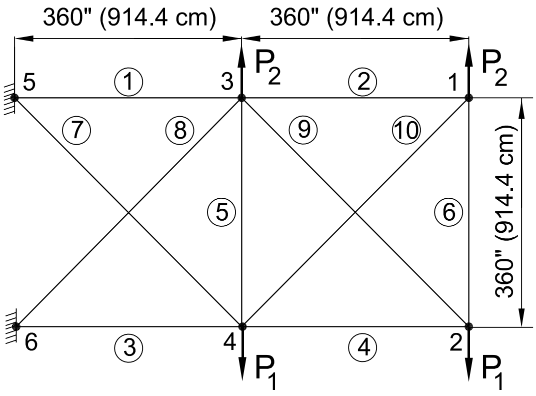

Fig. 9 Ten-bar planer truss structure

limit is exceeded at node 2 in both cases, and the stress limit is exceeded in case 2.

The optimization results reported in Table 13 show that IGMM has best results among those that satisfy constraints. There are some smaller weights among reported results that violated in some constraints and are not acceptable.

As reported in Table 14, in case 2, IGMM has best result among algorithms which satisfy all constraints. In this problem, Farshi and Alinia [115] have also found good results. Figure 10 illustrates the convergence trend of IGMM algorithm for 10-bar truss example.

As shown in Fig. 10, in case 1, the IGMM has been able to reach the 99 % of the global optimum point through 2440 structural analyses in 61 iterations, and finally, after 345 iterations and with 13,800 analyses, the best solution is found. In case 2, after 109 iterations and 4360 structural analyses, the IGMM could reach the 99 % of the best solution, and finally, after 341 iterations and with 13,640 analyses, the global solution is found.

Table 13 Comparison result of the 10-bar truss (case 1)

Member number	Schmit and Farshi [110]	Schmit and Miura [111]	Lee and Geem [57]	Li et al. [112]	Kaveh and Rahami [113]	Li et al. [112]	Camp et al. [114]	Farshi and Alinia [115]	This work
1	33.43	30.67	30.15	30.704	30.6677	30.569	28.92	30.5208	30.50694
2	0.1	0.1	0.102	0.1	0.1	0.1	0.1	0.1	0.1
3	24.26	23.76	22.71	23.167	22.8722	22.974	24.07	23.2040	23.302
4	14.26	14.59	15.27	15.183	15.3445	15.148	13.96	15.2232	15.19534
5	0.1	0.1	0.102	0.1	0.1	0.1	0.1	0.1	0.1
6	0.1	0.1	0.544	0.551	0.4635	0.547	0.56	0.5515	0.543684
7	8.388	8.578	7.541	7.460	7.4796	7.493	7.69	7.669	7.461249
8	20.74	21.07	21.56	20.978	20.9651	21.159	21.95	21.0342	21.11301
9	19.69	20.96	21.45	21.508	21.7026	21.556	22.09	21.5294	21.41383
10	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Weight (lb)	5089.0	5076.85	5057.88	5060.9	5061.90	5061.03	5076.31	5061.4	5060.99
δ_{max}	1.9998	1.9999	*2.0018*	2	2	2	*2.0007*	1.9976	2
σ_{max}	21.192	20.349	24.997	25	24.995	24.998	23.866	24.046	24.9903
NFEs	NA	NA	400,000	150,000	NA	150,000	NA	NA	13,800

1 in.^{2} = 6.452 cm^2; 1 lb. = 4.448 N; *italic* sets are violated constraint

Table 14 Comparison of results for the 10-bar truss (Case 2)

Member number	Schmit and Farshi [110]	Schmit and Miura [111]	Lee and Geem [57]	Rizzi [116]	John et al. [117]	Li et al. [112]	Kaveh and Talatahari [118]	Farshi and Alinia [115]	This work
1	24.29	23.55	23.25	23.53	23.59	23.473	23.194	23.5270	23.41074
2	0.1	0.1	0.102	0.1	0.1	0.101	0.1	0.1	0.100005
3	23.35	25.29	25.73	25.29	25.25	25.287	24.585	25.2941	25.30489
4	13.66	14.36	14.51	14.37	14.37	14.413	14.221	14.3760	14.35426
5	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.100082
6	1.969	1.97	1.977	1.97	1.97	1.969	1.969	1.9698	1.970133
7	12.67	12.39	12.21	12.39	12.39	12.362	12.489	12.4041	12.46945
8	12.54	12.81	12.61	12.83	12.80	12.694	12.925	12.8245	13.1059
9	21.97	20.34	20.36	20.33	20.37	20.323	20.952	20.3304	20.06795
10	0.1	0.1	0.1	0.1	0.1	0.103	0.101	0.1	0.100027
Weight (lb)	4691.84	4676.96	4668.8	4677.1	4676.93	4677.70	4675.78	4677.8	4677.5138
δ_{max}	1.9999	*2.0001*	*2.0039*	1.9999	1.9993	2	*2.0016*	1.9999	2
σ_{max}	24.995	24.99	*25.041*	*25.002*	*25.005*	*25.001*	25	25	24.998
NFEs	NA	NA	400,000	NA	NA	150,000	NA	NA	13,640

1 in.^{2} = 6.452 cm^2; 1 lb. = 4.448 N; *italic* sets are violated constraint

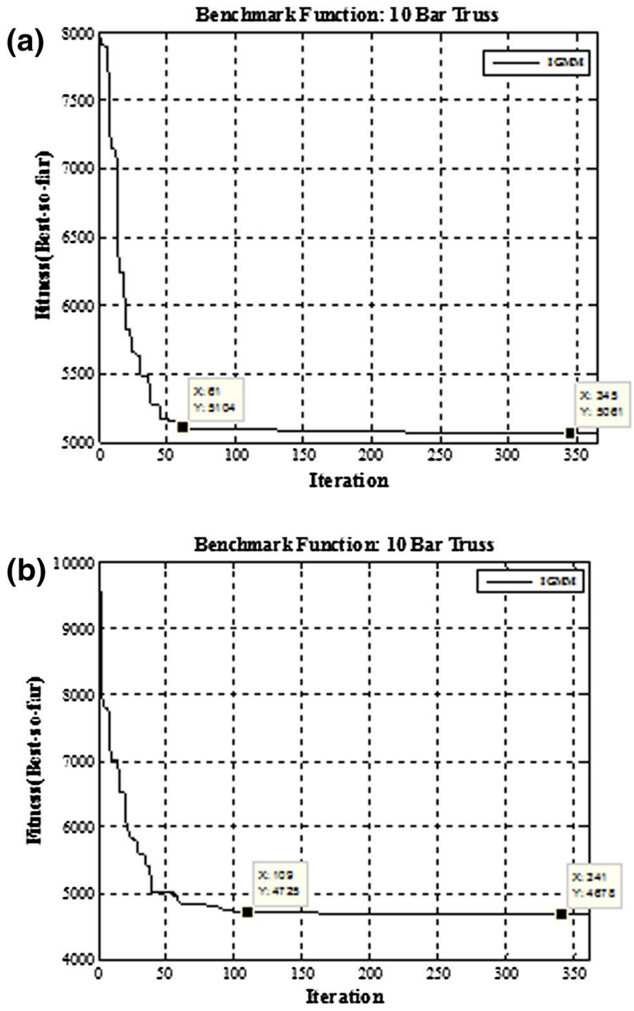

Fig. 10 Convergence history for the 10-bar truss example for **a** case 1, **b** case 2

5.3.2 *Spatial 25-bar truss problem*

The 25-bar spatial truss problem is shown in Fig. 11. The material density is 0.1 lb/in.^{3} (2767.990 kg/m^3), and the modulus of elasticity is 10 Msi (68.950 GPa). Elements can be set into eight distinct groups, as: (1) A1, (2) A2–5, (3) A6–9, (4) A10–11, (5) A12–13, (6) A14–17, (7) A18–21, and (8) A22–25.

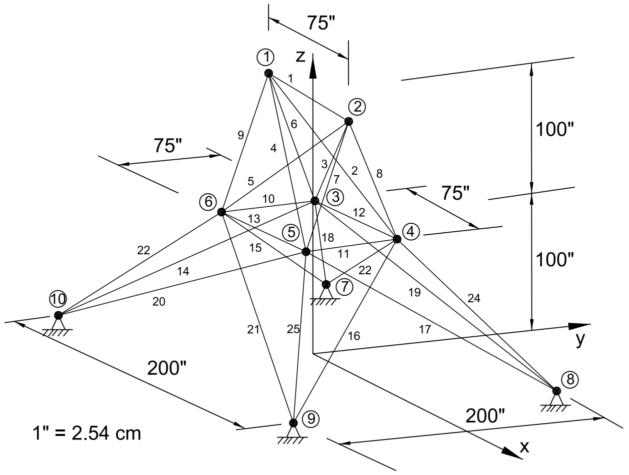

Fig. 11 Graphic view of the 25-bar spatial truss

As shown in Table 15, the truss structure is subjected to two loading cases. The allowable nodal displacement is limited to a maximum value of ±0.35 in. (±8.89 mm) in all coordinate directions. Table 16 lists all allowable compressive and tensile stresses for each element group. The cross-sectional areas are allowed to vary from 0.01 in.^{2} (0.0645 cm^2) to 3.4 in.^{2} (21.94 cm^2).

In Table 17, the optimum cross-sectional areas and the statistical information of solutions found by IGMM are compared with the other techniques available in the literature, including ACO [119], BB-BC [120], PSO [121], HPSACO [118], CSS [122], and CSP [123]. The best weight obtained by the IGMM is 545.186 lb.

Apparently, ACO performed better results and in less number of total analyses required. However, as shown in Table 18, in case 2, where the truss elements 19 and 20 are found to be the active stress constraints using any of the techniques listed, results show that IGMM has found the optimal solution without any violation, while all others are violated either with respect to stress or displacement constraints.

Figure 12 illustrated the convergence history for the 25-bar truss problem using IGMM. As is clear from Fig. 12, IGMM was able to reach 99 % of the global optimum solution after 1840 analyses at iteration 46 and the global solution after 5240 analyses at iteration 131.

Table 15 Loading conditions for spatial 25-bar truss structure

Node	Case 1			Case 2		
	P_x Kips (kN)	P_y Kips (kN)	P_z Kips (kN)	P_x Kips (kN)	P_y Kips (kN)	P_z Kips (kN)
1	0.0	20.0 (89)	−5.0 (22.25)	1.0 (4.45)	10.0 (44.5)	−5.0 (22.25)
2	0.0	−20.0 (89)	−5.0 (22.25)	0.0	10.0 (44.5)	−5.0 (22.25)
3	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0
6	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0

Table 16 Stress limits for the spatial 25-bar truss structure

Element group		Compressive stress limits ksi (MPa)	Tensile stress limits ksi (MPa)
1	A_1	35.092 (241.96)	40.0 (275.80)
2	A_{2-5}	11.590 (79.913)	40.0 (275.80)
3	A_{6-9}	17.305 (119.31)	40.0 (275.80)
4	A_{10-11}	35.092 (241.96)	40.0 (275.80)
5	A_{12-13}	35.092 (241.96)	40.0 (275.80)
6	A_{14-17}	6.759 (46.603)	40.0 (275.80)
7	A_{18-21}	6.959 (47.982)	40.0 (275.80)
8	A_{22-25}	11.082 (76.410)	40.0 (275.80)

Table 17 Comparison of optimized designs for the 25-bar truss problem under multiple load cases

Element group		Optimal cross-sectional areas (in.^{2})						
		ACO [119]	BB-BC [120]	PSO [121]	HPSACO [118]	CSS [122]	CSP [123]	This work
1	A_1	0.010	0.010	0.010	0.010	0.010	0.010	0.01
2	A_{2-5}	2.042	1.993	2.052	2.054	2.003	1.910	2.005954809
3	A_{6-9}	3.001	3.056	3.001	3.008	3.007	2.798	2.961418134
4	A_{10-11}	0.010	0.010	0.010	0.010	0.010	0.010	0.01
5	A_{12-13}	0.010	0.010	0.010	0.010	0.010	0.010	0.01
6	A_{14-17}	0.684	0.665	0.684	0.679	0.687	0.708	0.686620167
7	A_{18-21}	1.625	1.642	1.616	1.611	1.655	1.836	1.675840145
8	A_{22-25}	2.672	2.679	2.673	2.678	2.660	2.645	2.667559246
Best (lb)		545.03	545.16	545.21	544.99	545.10	545.09	545.186
Average (lb)		545.74	545.66	546.84	545.52	545.58	545.20	545.86
Std. (lb)		0.940	0.491	1.478	0.315	0.412	0.487	0.307
NFEs		3520	12,500	9596	9875	7000	17,500	5240

Table 18 Comparison of active constraint's values for the 25-bar truss problem

	Active constraints	Allowable values	ACO [119]	BB-BC [120]	PSO [121]	HPSACO [118]	CSS [122]	CSP [123]	This Work
Load case 1	Y direction displacement in node 1	0.35	0.34994	0.34998	0.34999	*0.35004*	0.34998	*0.35123*	0.35
	X direction displacement in node 3	0.35	0.34994	0.34998	0.34999	*0.35004*	0.34998	*0.35123*	0.35
Load case 2	Stress in element 19	−6.959	*−7.1497*	*−7.1026*	*−7.1837*	*−7.2042*	*−7.0428*	−6.3869	−6.9517
	Stress in element 20	−6.959	*−7.1497*	*−7.1026*	*−7.1837*	*−7.2042*	*−7.0428*	−6.3869	−6.9517
	Y direction displacement in node 1	0.35	0.34996	0.35002	0.35	*0.35003*	*0.35002*	*0.35229*	0.35
	X direction displacement in node 3	−0.35	−0.34996	−0.35002	−0.35	*−0.35003*	*−0.35002*	*−0.35229*	−0.35

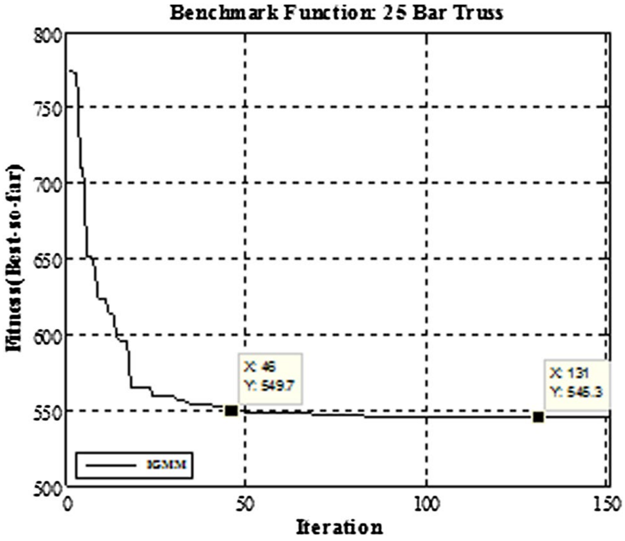

Fig. 12 Convergence history for the 25-bar truss example

6 Conclusions

In recent years, various heuristic optimization methods have been developed. Some of them are inspired by natural processes. In this paper, based on the characteristics of ideal gases and movement and collision of their molecules, an optimization algorithm named Ideal Gas Molecular Movement (IGMM) algorithm was proposed. For the optimization process, IGMM randomly generates a set of molecules in the problem domain. Each molecule acts as a potential response to the optimization problem moves, collides, and exchanges information with other molecules throughout the problem domain. Since gas molecules have an effective velocity that relates with their mass and temperature, to use the velocity equation, an initial temperature is defined for each molecule. Moreover, the mass of each molecule is determined in relation to its fitness. In the early stages of optimization, where a comprehensive rapid search takes place, the temperature is set to a large value. Gradually, as more regions are discovered with local optima, the velocity of molecules declines with a decrease in their temperatures. Moving gas molecules collide with one another with a certain probability. As a result of the collision caused by distribution of the existing energy, each molecule gains a new velocity and position. The collision probability increases as further involved into optimization process and molecules more accumulatively unite around the global optimum point led to the final optimum solution. The evaluation assessment of the proposed algorithm took place against various benchmark mathematical and engineering optimization problems. The results were compared with those based on several well-known optimization algorithms which showed the significance of the proposed IGMM algorithm both in terms of the accuracy and the number of function evaluations towards the optimal solutions.

Compliance with ethical standards

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

References

1. Kaveh A, Talatahari S (2010) A novel heuristic optimization method: charged system search. Acta Mech 213:267–289. doi:10.1007/s00707-009-0270-4
2. Alberdi R, Khandelwal K (2015) Comparison of robustness of metaheuristic algorithms for steel frame optimization. Eng Struct 102:40–60. doi:10.1016/j.engstruct.2015.08.012
3. Mirjalili S (2015) Dragonfly algorithm : a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl. doi:10.1007/s00521-015-1920-1
4. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput 27:155–182. doi:10.1108/02644401011008577
5. Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Tech Rep TR06, Erciyes Univ 10. doi:citeulike. Article-id: 6592152
6. Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In: 2009 World Congr. Nat. Biol. Inspired Comput. NABIC 2009—Proc, pp 210–214
7. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98. doi:10.1016/j.advengsoft.2015.01.010
8. Wang G-G, Guo L, Gandomi AH et al (2014) Chaotic Krill Herd algorithm. Inf Sci (Ny) 274:17–34. doi:10.1016/j.ins.2014.02.123
9. Guo L, Wang G-G, Gandomi AH et al (2014) A new improved krill herd algorithm for global numerical optimization. Neurocomputing 138:392–402. doi:10.1016/j.neucom.2014.01.023
10. Wang G-G, Deb S, Coelho LDS (2015) Earthworm optimization algorithm: a bioinspired metaheuristic algorithm for global optimization problems. Int J Bio Inspir Comput (**in press**)
11. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci (NY) 179:2232–2248. doi:10.1016/j.ins.2009.03.004
12. Wang G-G, Deb S, Cui Z (2015) Monarch butterfly optimization. Neural Comput Appl. doi:10.1007/s00521-015-1923-y
13. Wang G-G, Deb S, Coelho LDS (2015) Elephant herding optimization. 3rd Int Symp Comput Bus Intell. doi:10.1109/ISCBI.2015.8
14. Wang G-G, Deb S, Gao X-Z, Coelho LDS (2016) A new metaheuristic optimization algorithm motivated by elephant herding behavior. Int J Bio Inspir Comput (**in press**)
15. Mirjalili SM, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. doi:10.1016/j.advengsoft.2013.12.007
16. Dizangian B, Ghasemi MR (2015) A fast decoupled reliability-based design optimization of structures using B-spline interpolation curves. J Brazilian Soc Mech Sci Eng 1–13. doi:10.1007/s40430-015-0423-4
17. Dizangian B, Ghasemi MR (2015) Ranked-based sensitivity analysis for size optimization of structures. J Mech Des 137:121402-1–121402-10. doi:10.1115/1.4031295

18. Shobeiri V (2016) The optimal design of structures using ACO and EFG. Eng Comput. doi:10.1007/s00366-016-0443-4
19. Sadollah A, Eskandar H, Bahreininejad A, Kim JH (2015) Water cycle, mine blast and improved mine blast algorithms for discrete sizing optimization of truss structures. Comput Struct 149:1–16. doi:10.1016/j.compstruc.2014.12.003
20. Safaeian Hamzehkolaei N, Miri M, Rashki M (2015) An enhanced simulation-based design method coupled with meta-heuristic search algorithm for accurate reliability-based design optimization. Eng Comput 1–19. doi:10.1007/s00366-015-0427-9
21. Fabritius B, Tabor G (2015) Improving the quality of finite volume meshes through genetic optimisation. Eng Comput. doi:10.1007/s00366-015-0423-0
22. Ghasemi MR, Dizangian B (2010) Size, shape and topology optimization of composite steel box girders using PSO method. ASIAN J Civ Eng (BUILDING HOUSING) 11:699–715
23. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82. doi:10.1109/4235.585893
24. Sonntag RE, Borgnakke C, Van Wylen GJ (2009) Fundamentals of thermodynamics 7th edn. University of Michigan, Willey
25. Van Wylen GJ, Sonntag RE, Dybbs A (1975) Fundamentals of classical thermodynamics (2nd edition). J Appl Mech 42:522. doi:10.1115/1.3423632
26. Laurendeau NM (2005) Statistical thermodynamics fundamentals and applications. Cambridge University Press, Cambridge
27. Harsha PS (2006) Principles of vapor deposition of thin films. The Boulevard, Oxford
28. Moore FL (1963) Kinetic theory of gases. Am J Phys 31:213. doi:10.1119/1.1969378
29. Mirjalili S, Zaiton S, Hashim M et al (2010) A new hybrid PSOGSA algorithm for function optimization. Proc ICCIA 2010 Int Conf Comput Inf Appl 374–377. doi:10.1109/ICCIA.2010.6141614
30. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison Wesley, Boston. doi:10.1007/s10589-009-9261-6
31. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Neural Networks, 1995. Proceedings, IEEE Int. Conf, vol 4, pp 1942–1948
32. Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76:60–68. doi:10.1177/003754970107600201
33. Simon D, Member S (2008) Biogeography-based optimization. Evol Comput IEEE Trans 12:702–713. doi:10.1109/TEVC.2008.919004
34. Wang G-G, Guo L, Wang H et al (2014) Incorporating mutation scheme into krill herd algorithm for global numerical optimization. Neural Comput Appl 24:853–871. doi:10.1007/s00521-012-1304-8
35. Patel VK, Savsani VJ (2015) Heat transfer search (HTS): a novel optimization algorithm. Inf Sci (NY) 324:217–246. doi:10.1016/j.ins.2015.06.044
36. Clerc M, Kennedy J (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space. Evol Comput IEEE Trans 6:58–73
37. Coello CAC, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inform 16:193–203
38. Parsopoulos KE, Vrahatis MN et al (2002) Particle swarm optimization method for constrained optimization problems. Intell Technol Appl New Trends Intell Technol 76:214–220
39. Joines J, Houck CR, others (1994) On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's. In: Evol. Comput. 1994. IEEE World Congr. Comput. Intell. Proc. First IEEE Conf., pp 579–584
40. Ahmadi-Nedushan B, Varaee H (2009) Optimal design of reinforced concrete retaining walls using a swarm intelligence technique. In: first Int. Conf. Soft Comput. Technol. Civil, Struct. Environ. Eng. UK, pp 1–12
41. Rao SS (2009) Engineering optimization: theory and practice. Wiley, New York. doi:10.1002/9780470549124
42. Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming. J Eng Ind 98:1021–1025
43. Deb K (1991) Optimal design of a welded beam via genetic algorithms. AIAA J 29:2013–2015. doi:10.2514/3.10834
44. Coello CAC (2000) Constraint-handling using an evolutionary multiobjective optimization technique. Civ Eng Syst 17:319–346
45. Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 37:443–473
46. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99
47. Mirjalili SM, Mirjalili SM, Hatamlou A (2015) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 1–19. doi:10.1007/s00521-015-1870-7
48. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41:113–127
49. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110:151–166
50. Coello CAC, Becerra RL (2004) Efficient evolutionary optimization through the use of a cultural algorithm. Eng Optim 36:219–236
51. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput 10:629–640
52. Zahara E, Kao Y-T (2009) Hybrid Nelder-Mead simplex search and particle swarm optimization for constrained engineering design problems. Expert Syst Appl 36:3880–3886
53. Lampinen J (2002) A constraint handling approach for the differential evolution algorithm. In: wcci, pp 1468–1473
54. Coello CAC (2012) Constraint-handling techniques used with evolutionary algorithms. In: Proc. 14th Annu. Conf. companion Genet. Evol. Comput, pp 849–872
55. Coello CAC, Montes EM (2001) Use of dominance-based tournament selection to handle constraints in genetic algorithms. Intell Eng Syst through Artif Neural Netw 11:177–182
56. Gao L, Hailu A (2010) Comprehensive learning particle swarm optimizer for constrained mixed-variable optimization problems. Int J Comput Intell Syst 3:832–842
57. Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. Comput Struct 82:781–798. doi:10.1016/j.compstruc.2004.01.002
58. Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 112:223–229
59. Kannan BK, Kramer SN (1994) An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116:405–411
60. Akhtar S, Tai K, Ray T (2002) A socio-behavioural simulation model for engineering design optimization. Eng Optim 34:341–354. doi:10.1080/03052150212723
61. Yun YS (2005) Study on Adaptive hybrid genetic algorithm and its applications to engineering design problems. MSc Thesis, Waseda University

62. Tsai J-F, Li H-L, Hu N-Z (2002) Global optimization for signomial discrete programming problems in engineering design. Eng Optim 34:613–622
63. Coelho LDS (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 37:1676–1683
64. Cao YJ, Wu QH (1997) Evolutionary programming. In: Evol. Comput. 1997., IEEE Int. Conf., pp 443–446
65. Deb K (1997) Geneas: A robust optimal design technique for mechanical component design. In: Evol. algorithms Eng. Appl. Springer, Berlin, pp 497–514
66. Coello CAC (1999) Self-adaptive penalties for GA-based optimization. Evol. Comput. 1999. CEC 99. Proc. 1999 Congr., p 1
67. Sandgren E (1988) Nonlinear integer and discrete programming in mechanical design. In: Proc. ASME Des. Technol. Conf., pp 95–105
68. Zhang C, Wang H-P (1993) Mixed-discrete nonlinear optimization with simulated annealing. Eng Optim 21:277–291
69. Coello CAC, Cortés NC (2004) Hybridizing a genetic algorithm with an artificial immune system for global optimization. Eng Optim 36:607–634
70. Homaifar A, Qi CX, Lai SH (1994) Constrained optimization via genetic algorithms. Simulation 62:242–253
71. Gandomi AH (2014) Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans 53:1168–1183
72. Mirjalili S (2015) Moth-flame optimization algorithm : a novel nature-inspired heuristic paradigm. Knowl Based Syst. doi:10.1016/j.knosys.2015.07.006
73. Michalewicz Z, Attia N (1994) Evolutionary optimization of constrained problems. In: Proc. 3rd Annu. Conf. Evol. Program, pp 98–108
74. Gandomi AH, Yang X-S, Alavi AH, Talatahari S (2013) Bat algorithm for constrained optimization tasks. Neural Comput Appl 22:1239–1255
75. Ben Hadj-Alouane A, Bean JC (1997) A genetic algorithm for the multiple-choice integer program. Oper Res 45:92–101
76. Fu J-F, Fenton RG, Cleghorn WL (1991) A mixed integer-discrete-continuous programming method and its application to engineering design optimization. Eng Optim 17:263–280
77. Li H-L, Chou C-T (1993) A global approach for nonlinear mixed discrete programming in design optimization. Eng Optim 22:109–122
78. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput J 13:2592–2612. doi:10.1016/j.asoc.2012.11.026
79. Cai J, Thierauf G (1997) Evolution strategies in engineering optimization. Eng Optim 29:177–199
80. He S, Prempain E, Wu QH (2004) An improved particle swarm optimizer for mechanical design optimization problems. Eng Optim 36:585–605
81. Cao YJ, Wu QH (1999) A mixed variable evolutionary programming for optimisation of mechanical design. Eng Intell Syst Electr Eng Commun 7:77–82
82. Hu X, Eberhart RC, Shi Y (2003) Engineering optimization with particle swarm. In: Swarm Intell. Symp. 2003. SIS'03. Proc. 2003 IEEE, pp 53–57
83. Huang F, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186:340–356
84. Litinetski VV, Abramzon BM (1998) MARS-A multistart adaptive random search method for global constrained optimization in engineering applications. Eng Optim 30:125–154
85. Wu S-J, Chow P-T (1995) Genetic algorithms for nonlinear mixed discrete-integer optimization problems via meta-genetic parameter optimization. Eng Optim A35(24):137–159
86. Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933. doi:10.1016/j.cma.2004.09.007
87. Cagnina LC, Esquivel SC, Coello CAC (2008) Solving engineering optimization problems with the simple constrained particle swarm optimizer. Informatica 32:319–326
88. Hossein A, Yang GX, Gandomi AH et al (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35. doi:10.1007/s00366-011-0241-y
89. Ray T, Liew KM (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. Evol Comput IEEE Trans 7:386–396
90. Mezura-Montes E, Coello CAC, Velázquez-Reyes J, Muñoz-Dávila L (2007) Multiple trial vectors in differential evolution for engineering design. Eng Optim 39:567–589
91. Parsopoulos KE, Vrahatis MN (2005) Unified particle swarm optimization for solving constrained engineering optimization problems. In: Adv. Nat. Comput. Springer, Berlin, pp 582–591
92. Shih CJ, Lai TK (1995) Mixed-discrete fuzzy programming for nonlinear engineering optimization. Eng Optim 23:187–199
93. Li H-L, Chang C-T (1998) An approximate approach of global optimization for polynomial programming problems. Eur J Oper Res 107:625–632
94. Belegundu AD, Arora JS (1985) A study of mathematical programming methods for structural optimization. Part I: theory. Int J Numer Methods Eng 21:1583–1599
95. Arora J (2004) Introduction to optimum design. Academic Press, Cambridge
96. Krohling R, dos Santos Coelho LDS et al (2006) Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems. Syst Man Cybern Part B Cybern IEEE Trans 36:1407–1416
97. Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188:1567–1579. doi:10.1016/j.amc.2006.11.033
98. Kaveh A, Mahdavi VRR (2014) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27. doi:10.1016/j.compstruc.2014.04.005
99. He Q, Wang L (2007) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 186:1407–1422
100. Wang L, Li L (2010) An effective differential evolution with level comparison for constrained engineering design. Struct Multidiscip Optim 41:947–963
101. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci (NY) 178:3043–3074
102. Wang Y, Cai Z, Zhou Y, Fan Z (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct Multidiscip Optim 37:395–413
103. Karaboga D, Basturk B (2007) Artificial bee colony (ABC) optimization algorithm for solving constrained optimization. In: Probl. LNCS Adv. Soft Comput. Found. Fuzzy Log. Soft Comput. Springer, IFSA (2007). Citeseer, pp 789–798
104. Mezura-Montes E, Coello CAC (2005) Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: MICAI 2005 Adv. Artif. Intell. Springer, Berlin, pp 652–662
105. Chickermane H, Gea HC (1996) Structural optimization using a new local approximation method. Int J Numer Methods Eng 39:829–846
106. Svanberg K (1987) The method of moving asymptotes- a new method for structural optimization. Int J Numer Methods Eng 24:359–373

107. Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112
108. Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. Comput Sci Inform 26:30–45
109. Sharma TK, Pant M, Singh VP (2012) Improved local search in artificial bee colony using golden section search. arXiv:1210.6128 (arXiv Prepr)
110. Schmit LA, Farshi B (1974) Some approximation concepts for structural synthesis. AIAA J 12:692–699
111. Schmit LA, Miura H (1976) Approximation concepts for efficient structural synthesis. US National Aeronautics and Space Administration
112. Li LJ, Huang ZB, Liu F (2009) A heuristic particle swarm optimization method for truss structures with discrete variables. Comput Struct 87:435–443. doi:10.1016/j.compstruc.2009.01.004
113. Kaveh A, Rahami H (2006) Analysis, design and optimization of structures using force method and genetic algorithm. Int J Numer Methods Eng 65:1570–1584
114. Camp C, Pezeshk S, Cao G (1998) Optimized design of two-dimensional structures using a genetic algorithm. J Struct Eng 124:551–559. doi:10.1061/(ASCE)0733-9445(1998)124:5(551)
115. Farshi B, Alinia-Ziazi A (2010) Sizing optimization of truss structures by method of centers and force formulation. Int J Solids Struct 47:2508–2524. doi:10.1016/j.ijsolstr.2010.05.009
116. Rizzi P (1976) Optimization of multi-constrained structures based on optimality criteria. Conf. AIAA/ASME/SAE 17th Struct. Struct. Dyn. Mater., King Prussia, PA
117. John KV, Ramakrishnan CV, Sharma KG (1987) Minimum weight design of trusses using improved move limit method of sequential linear programming. Comput Struct 27:583–591
118. Kaveh A, Talatahari S (2009) Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 87:267–283. doi:10.1016/j.compstruc.2009.01.003
119. Camp CV, Bichon BJ (2004) Design of space trusses using ant colony optimization. J Struct Eng 130:741–751
120. Kaveh A, Talatahari S (2009) Size optimization of space trusses using Big Bang-Big Crunch algorithm. Comput Struct 87:1129–1140. doi:10.1016/j.compstruc.2009.04.011
121. Schutte JF, Groenwold AA (2003) Sizing design of truss structures using particle swarms. Struct Multidiscip Optim 25:261–269
122. Kaveh A, Talatahari S (2010) Optimal design of skeletal structures via the charged system search algorithm. Struct Multidiscip Optim 41:893–911
123. Kaveh A, Sheikholeslami R, Talatahari S, Keshvari-Ilkhichi M (2014) Chaotic swarming of particles: a new method for size optimization of truss structures. Adv Eng Softw 67:136–147. doi:10.1016/j.advengsoft.2013.09.006