Knowledge-Based Systems 163 (2019) 283–304

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Atom search optimization and its application to solve a hydrogeologic parameter estimation problem

Weiguo Zhao [a,b], Liying Wang [a,*], Zhenxing Zhang [b,*]

[a] School of Water Conservancy and Hydropower, Hebei University of Engineering, Handan, Hebei, 056021, China
[b] Illinois State Water Survey, Prairie Research Institute, University of Illinois at Urbana-Champaign, Champaign, IL 61820, USA

HIGHLIGHTS

- A novel optimization algorithm called Atom Search Optimization (ASO) is proposed.
- ASO is benchmarked on 37 well-known test functions.
- The results on test functions show the competitiveness of ASO.
- The results on hydrogeologic parameter estimation confirm the performance of ASO.

ARTICLE INFO

Article history:
Received 26 May 2018
Received in revised form 21 August 2018
Accepted 24 August 2018
Available online 28 August 2018

Keywords:
Optimization algorithm
Heuristic algorithm
Benchmark functions
Atom search optimization
Global optimization
Metaheuristic
Parameter estimation

ABSTRACT

In recent years, various metaheuristic optimization methods have been proposed in scientific and engineering fields. In this study, a novel physics-inspired metaheuristic optimization algorithm, atom search optimization (ASO), inspired by basic molecular dynamics, is developed to address a diverse set of optimization problems. ASO mathematically models and mimics the atomic motion model in nature, where atoms interact through interaction forces resulting from the Lennard-Jones potential and constraint forces resulting from the bond-length potential. The proposed algorithm is simple and easy to implement. ASO is tested on a range of benchmark functions to verify its validity, qualitatively and quantitatively, and then applied to a hydrogeologic parameter estimation problem with success. The results demonstrate that ASO is superior to some classic and newly emerging algorithms in the literature and is a promising solution to real-world engineering problems.

1. Introduction

Metaheuristic optimization algorithms are increasingly popular in intelligent computing and widely applied to a large number of real-world engineering problems. Their popularity derives from the following aspects. Firstly, all of these optimization techniques have some fundamental theories and mathematical models proven to be reasonable, which come from the real world and are inspired by all kinds of physical phenomena or biological behaviors [1,2]. The theories are simple and easy to understand. Secondly, these optimization algorithms can be considered as a black box. It means that given a set of inputs, these algorithms can easily provide a set of outputs for any optimization problem. They are very flexible and versatile since one can change the structures and parameters of algorithms to obtain better solutions. Thirdly, metaheuristic algorithms can effectively avoid local optima, which is very valuable for addressing engineering problems as many engineering problems are considered as multimodal functions. In addition, one can develop their variants by absorbing the merits of other algorithms to improve the accuracy of solutions within a reasonable time. Fourthly, metaheuristic optimization algorithms can tackle different types of problems including, but not limited to, single-objective and multi-objective problems, low-dimensional and high-dimensional problems, unimodal and multimodal problems, and discrete and continuous problems [3–5].

Many metaheuristic algorithms with different inspiration have been proposed and successfully used in a variety of fields, which are roughly classified into three classes [6]: evolution-inspired [7–9], physics-inspired [10], and swarm-inspired [11] methods.

Evolution-inspired algorithms are a stochastic, population-based approach, thus protecting a population's diversity is very important for the sustainable development of the algorithms iteratively. Many evolution-inspired algorithms maintain a population's diversity by mimicking basic genetic rules, including reproduction, mutation, selection, chemotaxis, elimination, and migration [12,13]. These algorithms randomly initialize a population

* Corresponding authors.
E-mail addresses: wgzhao@illinois.edu (W. Zhao), wangliying@hebeu.edu.cn (L. Wang), zhang538@illinois.edu (Z. Zhang).

https://doi.org/10.1016/j.knosys.2018.08.030
0950-7051/

evolved from subsequent iterations and evaluate the individual quality using a fitness function. Genetic algorithm (GA), originally presented by Holland [14], is a well-known classic evolutionary algorithm (EA). As GA can generally obtain high-quality solutions using mutation, crossover, and selection steps, the original version and its variants are widely applied to many real-world problems [15]. Since its emergence, a series of schemes aiming to enhance GA have been developed. With increasing popularity of GA, quite a number of other evolution-based algorithms in the literature, including evolutionary strategies (ES) [16], differential evolution (DE) [17], evolutionary programming (EP) [18], memetic algorithm (MA) [19], and so on, are proposed. Besides, all sorts of new EAs have been proposed recently, such as bacterial foraging optimization (BFO) [12], bat algorithm (BA) [20], fruit fly optimization algorithm (FOA) [21], monkey king evolutionary (MKE) [22], artificial algae algorithm (AAA) [23], biogeography-based optimization (BBO) [24], yin-yang-pair optimization (YYPO) [25], invasive weed optimization (IWO) [26], and dynamic virtual bats algorithm (DVBA) [27].

Physics-inspired algorithms simulate physical laws in the universe, among which, simulated annealing (SA) [1] is one of the most well-known algorithms. SA is inspired from the annealing process used in physical material in which a heated metal cools and freezes into a crystal texture with the minimum energy. Recently, many novel physics-inspired algorithms have been proposed, such as gravitational search algorithm (GSA) [28], electromagnetism-like mechanism (EM) algorithm [29], particle collision algorithm (PCA) [30], vortex search algorithm (VSA) [31], water evaporation optimization (WEO) [32], space gravitational algorithm (SGA) [33], big bang–big crunch algorithm (BB–BC) [34], galaxy based algorithm (GBA) [35], big crunch algorithm (BCA) [36], integrated radiation algorithm (IRA) [37], water drops algorithm (WDA) [38], charged system search (CSS) [39], magnetic optimization algorithm (MOA) [40], gravitation field algorithm (GFA) [41], ions motion algorithm (IMA) [42], water wave optimization (WWO) [43], gravitational interactions optimization (GIO) [44], teaching–learning-based optimization (TLBO) [45], hysteretic optimization (HO) [46], thermal exchange optimization (TEO) [47], light ray optimization (LRO) [48], heat transfer search (HTS) [49], spiral optimization algorithm (SOA) [50], water cycle algorithm (WCA) [51], and curved space optimization (CSO) [52].

Swarm-inspired algorithms mimic the collective behaviors of self-organization and shape-formation, natural or artificial [53]. There are two most classic swarm-inspired algorithms. One is particle swarm optimization (PSO) [2], which mimics bird flocking behaviors. In PSO, every agent moves around the search space to improve its solution, and their personal best positions and the global best position found so far are reserved, by which their positions are updated locally and socially. The other is ant colony optimization (ACO) [54], which follows the foraging process of an ant colony. Essentially, ants communicate with each other by pheromone trails through path formations, which assist them in finding the shortest path between the nest and food source. There are many newly developed swarm-inspired algorithms, such as artificial bee colony (ABC) [55], salp swarm algorithm (SSA) [56], krill herd algorithm (KH) [57], tree-seed algorithm (TSA) [58], social spider optimization (SSO) [59], bird mating optimizer (BMO) [60], cuckoo search (CS) [61], grasshopper optimization algorithm (GOA) [62], sine cosine algorithm (SCA) [63], moth swarm algorithm (MSA) [64], dolphin echolocation (DE) algorithm [65], hunting search (HS) algorithm [66], migrating birds optimization (MBO) [67], firefly algorithm (FA) [68], monkey search (MS) algorithm [69], and squirrel search algorithm (SSA) [70].

Compared with evolution-inspired or physics-inspired algorithms, swarm-inspired algorithms have some distinctive characteristics. On the one hand, part or all of the historical information about the population needs to be preserved, because every agent depends on the information to determine a new position in the search space over subsequent iterations. However, evolution-inspired algorithms require more operators. Swarm-inspired algorithms generally update positions of the population by interaction rules as standard formulas. On the other hand, swarm-inspired algorithms generally have two behaviors: exploration and exploitation [71,72]. Exploration means the ability of the algorithm to search for new solutions far from the current solution in the entire search space. Exploitation means the ability of the algorithm to search for the best solution near a new solution it has already found. In such algorithms, the range of every agent in the search space is scaled to a consensus in its neighborhood, and agents randomly explore the whole search space. If an agent or its neighbors find a good region, this region will be intensively exploited. Otherwise, they still extensively explore other regions, thus indicating their better self-adaptation in searching the global optima. From these perspectives, swarm-inspired algorithms have many advantages over other algorithms. Many evolution-inspired or physics-inspired algorithms have swarm-inspired characteristics, such as PSO, ACO, CS, BFO, GSA, and so on. These algorithms not only reflect the nature of biological phenomena or physical laws, but also share a common characteristic of exploration and exploitation. Thus they are more competitive than those without swarm-inspired characteristics. However, providing a proper balance between exploration and exploitation will lead to an optimal performance of the algorithm, so it is one of the most important tasks in the development of any stochastic optimization algorithm.

With the development of economy, society and technology, a great number of complex and challenging optimization problems have accordingly arisen in different fields. As an illustration, the emergence of ride-sharing companies that offer transportation on demand at a large scale, together with the increasing availability of corresponding demand datasets, develops a new complex optimization problem of effective handling of routing network [73]. Another challenging optimization problem is the identification of pollutant sources for river pollution incidents, which are caused by accident or illegal emissions [74]. Although a number types of optimization algorithms have introduced so far, new optimization algorithms are still being developed to tackle emerging complex optimization problems to obtain a better scheme. Furthermore, according to the No Free Lunch Theorem of Optimization [75], there is no optimization algorithm performing the best over all different types of problems. This theorem keeps this research field active and encourages relevant scholars to develop new algorithms for better optimization. Based on the above, a novel physics-inspired algorithm with swarm-inspired characteristics is proposed for global optimization in this study. The proposed algorithm, named atom search optimization (ASO), is inspired by basic molecular dynamics and is also a population-based heuristic algorithm. As far as the authors know, there are no related studies found in the literature. ASO mimics the atomic motion controlled by interaction and constraint forces to design an effective search mechanism for global optimization problems. The efficiency of ASO proposed is validated on a diverse set of mathematical optimization problems, and the results show its superiority to some classic and emerging algorithms. Additionally, ASO is successfully applied to a hydrogeologic parameter estimation problem, thus demonstrating its feasibility and effectiveness in real-world problems.

This paper is organized as follows. Section 2 provides a brief review of basic molecular dynamics. Section 3 presents the inspiration and the novel ASO algorithm in detail. Section 4 gives a comparative study and discussion on the benchmark functions, and Section 5 describes the application of ASO to a hydrogeologic parameter estimation problem. Finally, Section 6 presents some conclusions and suggests a few future research directions.

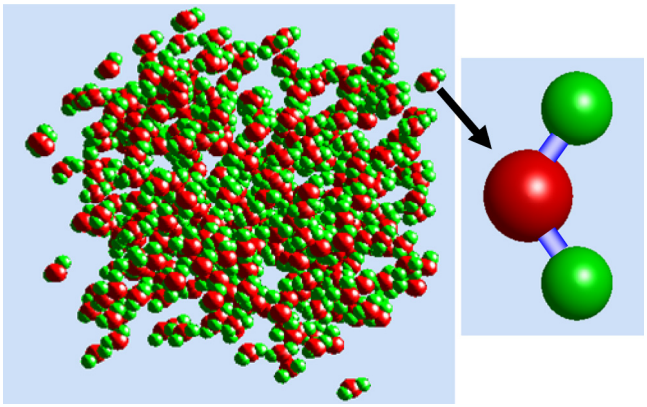

Fig. 1. Water molecules and their composition.

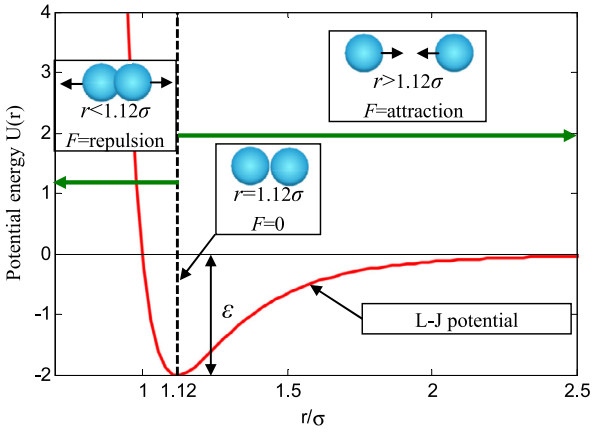

Fig. 2. L-J potential curve.

2. Basic molecular dynamics

ASO is inspired by basic molecular dynamics. From the micro perspective, a definition of "matter", based on its physical and chemical structure, is thus: matter is made up of molecules [76]. A molecule is the smallest unit of a chemical compound, and it exhibits the same chemical properties as those of that specific compound. A molecule is composed of atoms held together by covalent bonds that vary greatly in terms of complexity and size. So all substances are made of atoms and all atoms have mass and volume [77,78]. Fig. 1 shows the composition of water molecules, each of which is made up of two hydrogen atoms and one oxygen atom, jointly held by two covalent bonds. For an atomic system, all the atoms interact and are in constant motion, whether in the state of gas, liquid or solid. They are very complex in terms of their structure and microscopic interactions. Because an atomic system is typically composed of numerous atoms, it is analytically impossible to determine their properties that are affected by factors such as temperature, pressure, and so on. With the development of computer technology, molecular dynamics (MD) has rapidly developed in recent years. It circumvents this problem with the use of a computer simulation method to examine the physical movements of atoms and molecules.

MD was initially conceived in the field of theoretical physics [79,80] but its use has been extended to computational chemistry, materials science, and biology. Atomic motion follows the classical mechanics [81]. The interaction force among the atoms has two principal characteristics in an atom system. The first is the repulsion to compression, which repels at a close range of crowdedness. The second is the attraction that binds atoms together such as in solid and liquid states. Atoms attract each other over a further range of separation [82]. The potential energy of atoms can well account for these two characteristics, and there are a wide variety of pair-wise formulas in the literature used to express the potential energy [83,84]. The Lennard-Jones (L-J) potential [85,86], initially proposed for liquid, is a simple mathematical model that approximates the interaction force between a pair of atoms. The L-J potential between the ith and the jth atoms is commonly expressed as

$$U(r_{ij}) = 4\varepsilon\left[(\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^{6}\right] \tag{1}$$

where ε is the depth of the potential well that represents the strength of the interaction, σ is the length scale that denotes the collision diameter, $\boldsymbol{r}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i$, and $\boldsymbol{x}_i = (x_{i1}, x_{i2}, x_{i3})$ is the position of the ith atom in a 3-D space, so the Euclidian distance between the ith and jth atoms is

$$r_{ij} = ||\boldsymbol{x}_j - \boldsymbol{x}_i|| = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + (x_{i3} - x_{j3})^2} \tag{2}$$

In Eq. (1), $(\sigma/r)^{12}$ and $(\sigma/r)^{6}$ represent the repulsive and attractive interactions, respectively. The L-J potential curve is illustrated in Fig. 2, in which the attraction and repulsion regions are shown. In the repulsion region, the repulsion of the atoms rapidly increases as the distance between two atoms decreases. In the attraction region, as the distance between two atoms increases towards a certain further separation, the attraction gradually drops to zero. When two atoms reach an equilibration distance ($r = 1.12\sigma$), their minimum bonding potential energy is reached. At this point, the interaction force between the atoms is equal to zero.

Having specified the potential energy function, the interaction force that the jth atom exerts on the ith atom is

$$F_{ij} = -\nabla U(r_{ij}) = \frac{24\varepsilon}{\sigma^2}\left[2(\frac{\sigma}{r_{ij}})^{14} - (\frac{\sigma}{r_{ij}})^{8}\right]\boldsymbol{r}_{ij} \tag{3}$$

So the total interaction force exerted on the ith atom is simply given as

$$F_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} F_{ij} \tag{4}$$

where N is the total number of atoms in an atomic system.

To study more complex molecules, a molecular dynamics method with geometric constraints is proposed in [87], in which a combination of geometrical constraints and internal motion of atoms is considered. In polyatomic molecules, the highest-frequency internal vibrations are usually decoupled from rotational and translational motions. Thus a certain number of rigid bonds are introduced in the skeleton of the molecules. Consider the case in which the structure of a molecule is subject to one or more geometries. A constraint needs to be introduced to fix the distance between any two atoms with covalent bonds, and the mode can be expressed as

$$|\boldsymbol{x}_i - \boldsymbol{x}_i|^2 = b_{ij}^2 \tag{5}$$

where b_{ij} is the fixed bond length between the ith and jth atoms. Suppose that there are a total of $l$constraints influencing a particular molecule, and if the kth constraint for a bond works between the ikth and jkth atoms, then the kth constraint is

$$\theta_k = |\boldsymbol{x}_{ik} - \boldsymbol{x}_{ki}|^2 - b_{ij}^2 = 0, \quad k = 1, 2, \ldots, l \tag{6}$$

Hence, the constraint force G_i from the stretch of a covalent bond between two atoms acted on the ith atom can be written as

$$G_i = -\sum_{k=1}^{l} \lambda_k \nabla_i \theta_k = -2\sum_{k=1}^{l} \lambda_k (\boldsymbol{x}_{ik} - \boldsymbol{x}_{jk}) \tag{7}$$

where λ_k is the Lagrangian multiplier associated with θ_k. Hence, the motion equation of atoms with the constraint can be modified as

$$F_i + G_i = m_i a_i \tag{8}$$

For Eq. (8), the forces exerted on the atoms include not only all non-constraint interaction forces among molecules, but also the constraint force(s) within each molecule, thus embodying the essence of atomic motion.

In summary, basic molecular dynamics describes the movement principles of atoms, including the characteristics of the potential function, the motion mode of atoms with a non-constraint interaction force, and a geometric constraint force. Despite the simplicity of the analytical model, the physics-based study of molecular dynamics can be used to determine thermodynamic properties of the system, and indeed presents opportunities for many theoretical studies and practical applications [88–91].

3. Atom search optimization (ASO)

In this section, a novel optimization algorithm named atom search optimization (ASO) that is inspired by molecular dynamics is introduced. In ASO, the position of each atom within the search space represents a solution measured by its mass, with a better solution indicating a heavier mass, and vice versa. All atoms in the population will attract or repel each other according to the distance among them, encouraging the lighter atoms to move towards the heavier ones. Heavier atoms have smaller acceleration, which makes them seek intensively for better solutions in local spaces. Lighter atoms have greater acceleration, which makes them search extensively to find new promising regions in the entire search space.

The general unconstrained optimization problems can be defined as

$$\text{Minimize } f(\boldsymbol{x}), \quad \boldsymbol{x} = (x^1, \ldots, x^D) \tag{9}$$

for

$$Lb \le \boldsymbol{x} \le Ub, Lb = [lb^1, \ldots, lb^D], Ub = [ub^1, \ldots, ub^D] \tag{10}$$

where x^d $(d = 1, \ldots, D)$ is the dth component of the search space, lb^D and ub^D are the dth components of the lower and upper limits, respectively, and D is the dimension of the search space.

In order to solve this unconstrained optimization, suppose an atom population with N atoms. The position of the ith atom is expressed as

$$\boldsymbol{x}_i = [x_i^1, \ldots, x_i^D], \qquad i = 1, \ldots, N \tag{11}$$

where x_i^d $(d = 1, \ldots, D)$ is the dth position component of the ith atom in a D-dimension space. In the initial iterations of ASO, each atom interacts with others by the attraction or the repulsion among them, and the repulsion can avoid the over-concentration of atoms and the premature convergence of the algorithm, thus enhancing the exploration ability in the entire search space. As iterations pass, the repulsion gradually weakens and the attraction gradually strengthens, which signifies that the exploration decreases and the exploitation increases. In the final iterations, each atom interacts with others just by the attraction, which ensures that the algorithm has a good exploitation capability.

3.1. Mathematical representation of interaction force

The interaction force resulting from the L-J potential is the priming power of atomic motion. The interaction force acted on the ith atom from the jth atom at the tth iteration in Eq. (3) can be rewritten as

$$F_{ij}(t) = \frac{24\varepsilon(t)}{\sigma(t)}\left[2\left(\frac{\sigma(t)}{r_{ij}(t)}\right)^{13} - \left(\frac{\sigma(t)}{r_{ij}(t)}\right)^{7}\right]\frac{r_{ij}(t)}{\boldsymbol{r}_{ij}^d(t)} \tag{12}$$

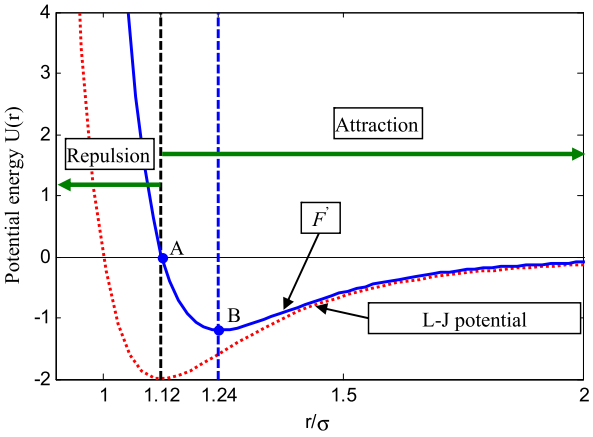

Fig. 3. Force curve of atoms.

and

$$F'_{ij}(t) = \frac{24\varepsilon(t)}{\sigma(t)}\left[2\left(\frac{\sigma(t)}{r_{ij}(t)}\right)^{13} - \left(\frac{\sigma(t)}{r_{ij}(t)}\right)^{7}\right] \tag{13}$$

The force curve of atoms is shown in Fig. 3. As shown, the atoms keep a relative distance, varying in a certain range all the time from the repulsion or attraction, and the change amplitude of the repulsion relative to the equilibration distance ($r = 1.12\sigma$) is much greater than that of the attraction. However, this model cannot be used directly to handle optimization problems, mainly because ASO needs to obtain more positive attraction and less negative repulsion as iterations increase, as shown in Fig. 3, Eq. (13) cannot satisfy this point. Accordingly, a revised version of this equation is developed, as follows, to solve optimization problems

$$F'_{ij}(t) = -\eta(t)\left[2(h_{ij}(t))^{13} - (h_{ij}(t))^{7}\right] \tag{14}$$

where $\eta(t)$ is the depth function to adjust the repulsion region or attraction region, which can be defined as

$$\eta(t) = \alpha\left(1 - \frac{t-1}{T}\right)^3 e^{-\frac{20t}{T}} \tag{15}$$

where a is the depth weight and T is the maximum number of iterations. The function behaviors of F', with different η corresponding to h ranging from 0.9 to 2, are illustrated in Fig. 4. From the figure, the repulsion occurs when h ranges from 0.9 to 1.12, the attraction occurs when h is between 1.12 and 2, and the equilibration occurs when $h = 1.12$. The attraction gradually increases with the increase of h from the equilibration ($h = 1.12$), reaches a maximum ($h = 1.24$) and then begins to decrease. The attraction is approximately equal to zero when h is greater than or equal to 2. Therefore, in ASO, to improve the exploration, a lower limit of the repulsion with a smaller function value is set to $h = 1.1$ and an upper limit of attraction with a larger function value is set to $h = 1.24$. Therefore, h is defined as

$$h_{ij}(t) = \begin{cases} h_{min} & \dfrac{r_{ij}(t)}{\sigma(t)} < h_{min} \\ \dfrac{r_{ij}(t)}{\sigma(t)} & h_{min} \le \dfrac{r_{ij}(t)}{\sigma(t)} \le h_{max} \\ h_{max} & \dfrac{r_{ij}(t)}{\sigma(t)} > h_{max} \end{cases} \tag{16}$$

where h_{min} and h_{max} are the lower and the upper limits of h, respectively, and the length scale $\sigma(t)$ is defined as

$$\sigma(t) = \left\| x_{ij}(t), \frac{\sum_{j \in Kbest} x_{ij}(t)}{K(t)} \right\|_2 \tag{17}$$

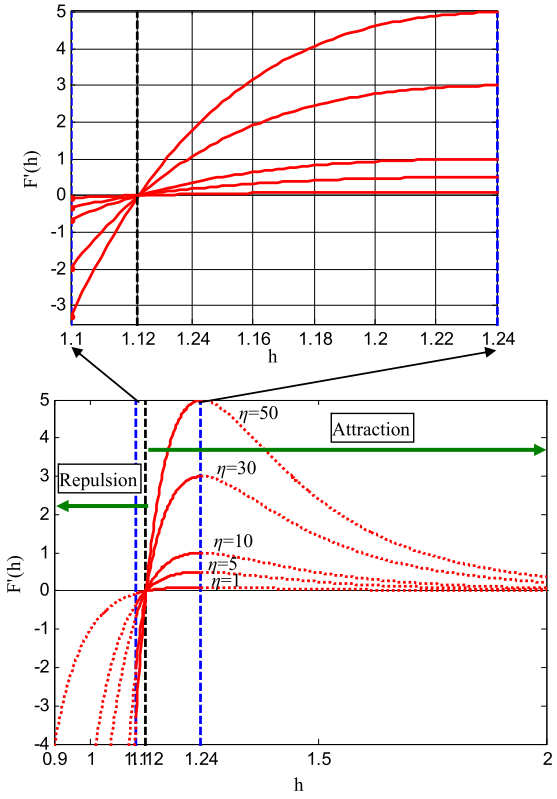

Fig. 4. Function behaviors of F' with different values of η.

and

$$\begin{cases} h_{min} = g_0 + g(t) \\ h_{max} = u \end{cases} \tag{18}$$

where *Kbest*, which is a subset of an atom population, is made up of the first K atoms with the best function fitness values. As a drift factor, g can make the algorithm drift from the exploration to the exploitation and is given as

$$g(t) = 0.1 \times \sin(\frac{\pi}{2} \times \frac{t}{T}) \tag{19}$$

Then the sum of components with random weights acted on the ith atom from the other atoms can be considered a total force, which is expressed as

$$F_i^d(t) = \sum_{j \in Kbest} rand_j F_{ij}^d(t) \tag{20}$$

where $rand_j$ is a random number in [0,1].

3.2. Mathematical representation of geometric constraint

The geometric constraint in molecular dynamics plays an important role in atomic motion. For simplicity, suppose each atom in ASO has a covalence bond with the best atom. Thus each atom is acted on by a constraint force from the best atom, so the constraint of the ith atom can be rewritten as

$$\theta_i(t) = \left[|\boldsymbol{x}_i(t) - \boldsymbol{x}_{best}(t)|^2 - b_{i,best}^2 \right] \tag{21}$$

where $x_{best}(t)$ is the position of the best atom at the tth iteration, and $b_{i,best}$ is a fixed bond length between the ith atom and the best atom. Hence the constraint force can be obtained as

$$G_i^d(t) = -\lambda(t)\nabla\theta_i^d(t) = -2\lambda(t)(x_i^d(t) - x_{best}^d(t)) \tag{22}$$

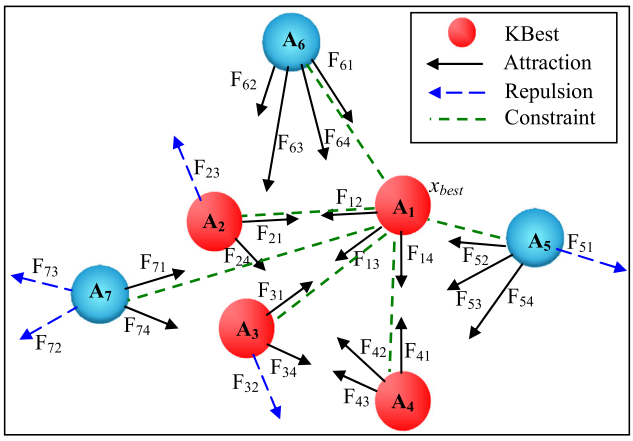

Fig. 5. Forces of an atom system with *KBest* for $K = 5$.

where $\lambda(t)$ is the Lagrangian multiplier. Then, making the substitution of $2\lambda \to \lambda$, the constraint force can be redefined as

$$G_i^d(t) = \lambda(t)(x_{best}^d(t) - x_i^d(t)) \tag{23}$$

The Lagrangian multiplier is defined as

$$\lambda(t) = \beta e^{-\frac{20t}{T}} \tag{24}$$

where β is the multiplier weight.

3.3. Mathematical representation of atomic motion

With the interaction force and the geometric constraint, the acceleration of the ith atom at time t can be written as

$$a_i^d(t) = \frac{F_i^d(t)}{m_i^d(t)} + \frac{G_i^d(t)}{m_i^d(t)} = -\alpha(1 - \frac{t-1}{T})^3 e^{-\frac{20t}{T}} \sum_{j \in Kbest} \frac{rand_j \left[2 \times (h_{ij}(t))^{13} - (h_{ij})^7\right]}{m_i(t)} \frac{(x_j^d(t) - x_i^d(t))}{||\boldsymbol{x}_i(t), \boldsymbol{x}_j(t)||_2} + \beta e^{-\frac{20t}{T}} \frac{x_{best}^d(t) - x_i^d(t)}{m_i(t)} \tag{25}$$

where $m_i(t)$ is the mass of the ith atom at the tth iteration, which can be measured at the simplest level by its function fitness value. The mass of the ith atom can be calculated as

$$M_i(t) = e^{-\frac{Fit_i(t) - Fit_{best}(t)}{Fit_{worst}(t) - Fit_{best}(t)}} \tag{26}$$

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^{N} M_j(t)} \tag{27}$$

where $Fit_{best}(t)$ and $Fit_{worst}(t)$ are the fitness values of the best and worst atoms at the tth iteration, respectively. $Fit_i(t)$ is the function fitness value of the ith atom at the tth iteration. $Fit_{best}(t)$ and $Fit_{worst}(t)$ are expressed as

$$Fit_{best}(t) = \min_{i \in \{1,2,\ldots,N\}} Fit_i(t) \tag{28}$$

$$Fit_{worst}(t) = \max_{i \in \{1,2,\ldots,N\}} Fit_i(t) \tag{29}$$

To simplify the algorithm, the position and velocity of the ith atom at the $(t+1)$th iteration can be denoted as follows

$$v_i^d(t+1) = rand_i^d v_i^d(t) + a_i^d(t) \tag{30}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{31}$$

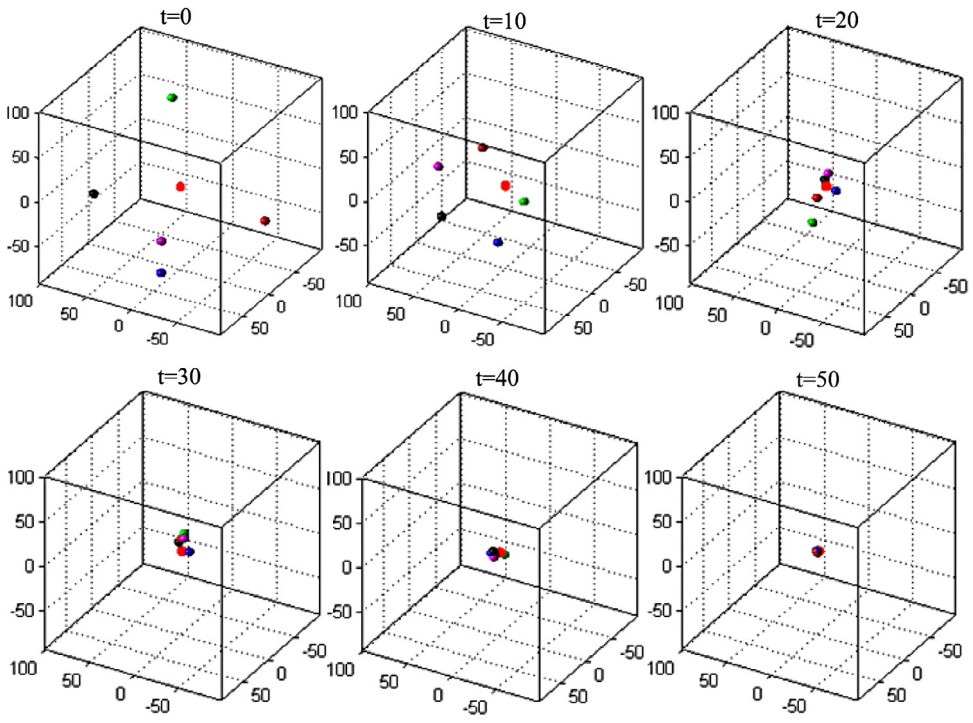

Fig. 6. Swarm motion of 5 atoms around a target in a 3-D space . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In ASO algorithm, to enhance the exploration in the first stage of iterations, each atom needs to interact with as many atoms with better fitness values as its K neighbors. To enhance the exploitation in the final stage of iterations, the atoms need to interact with as few atoms with better fitness values as its K neighbors. Therefore, as a function of time, K gradually decreases with the lapse of iterations. K can be calculated as

$$K(t) = N - (N-2) \times \sqrt{\frac{t}{T}} \tag{32}$$

The forces of an atom population are shown in Fig. 5, in which the first 5 atoms with the best fitness values are regarded as the *KBest*. As shown in the figure, A_1, A_2, A_3 and A_4 compose the *KBest*. A_5, A_6 and A_7 attract or repel each atom in the *KBest,* and A_1, A_2, A_3 and A_4 attract or repel each other. Each atom in the population except for A_1 (x_{best}) has a constraint force from the best atom A_1.

A simulation is conducted to examine how atoms move with this mathematical model. The swarm motion of 5 atoms around a target in a 3-D space is illustrated in Fig. 6, in which 5 different colored balls represent 5 different atoms, and the red point represents the desired target that every atom wants to reach. Initially, the positions of the 5 atoms are randomly generated in the search space. With the lapse of time t, all the atoms gradually approach the target using the mathematical mode and form a swarm. Finally, all the atoms converge to the target. Additionally, it can be found that, although the green atom is far away from the swarm when $t = 20$, the other atoms also pull it back by the attraction in the subsequent iterations, and all the atoms do not become too concentrated because of the repulsion. The motion histories of the 5 atoms during 50 iterations are illustrated in Fig. 7. It is apparent that the atoms grow denser when they are closer to the target, and the distribution of atoms in the search space is sufficient to demonstrate that the model proposed can achieve the transition from the exploration for the entire search space to the exploration for a focused region. It is obvious that this search characteristic can be extended to a n-D space.

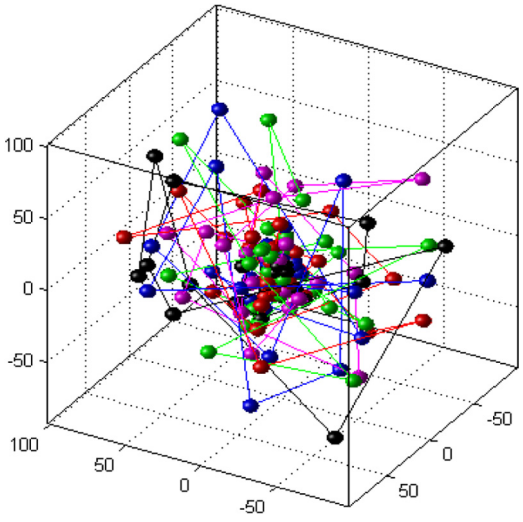

Fig. 7. Motion histories of 5 atoms during 50 iterations.

3.4. Framework of ASO algorithm

ASO starts the optimization by generating a set of random solutions. The atoms update their positions and velocities in each iteration, and the position of the best atom found so far is also updated in each iteration. In addition, the acceleration of atoms comes from two parts. One is the interaction force caused by the L-J potential, which actually is the vector sum of the attraction and the repulsion exerted from other atoms. Another is the constraint force caused by the bond-length potential, which is the weighted position difference between each atom and the best atom. All the updating and the calculation are performed interactively until the stop criterion is satisfied. Finally, the position and the fitness value of the best atom are returned as an approximation to the global optimum. The pseudo code of ASO algorithm is provided in Fig. 8.

```
Randomly initialize a set of atoms X (solutions) and their velocity v, and Fit_Best=Inf.
While the stop criterion is not satisfied do
  For each atom X_i do
    Calculate the fitness value Fit_i;
    If Fit_i< Fit_Best then
      Fit_Best= Fit_i;
      X_Best= X_i;
    End If.
    Calculate the mass using equations (26) and (27);
    Determine its K neighbors using equation (32);
    Calculate the intraction force F_i and the constraint force G_i using equations (20)
    and (23), respectively;
    Calculate the acceleration using equation (25);
    Update the velocity using equation (30);
    Update the position using equation (31);
  End For.
End While.
Find the best solution so far X_Best.
```

Fig. 8. Pseudo code of ASO algorithm.

ASO algorithm is very simple to implement and does not require many parameters except for the maximum number of iterations, the number of the atom population, and the dimension of problems to be solved, which are common parameters to all optimization algorithms. Moreover, the upper limit and the starting point of the lower limit can be selected as fixed values by the analysis of Fig. 4. In Eq. (18), when the starting point of function F' is fixed at $g_0 = 1.1$, ASO algorithm performs well. The upper limit should be set as $u = 1.24$, which is the maximum value of function F'. Therefore, the only parameters to be determined are the depth and multiplier weights. Empirically, it is recommended to set them in the range from 0 to 100 and from 0 to 1, respectively. The values of these parameters can be properly selected by four different benchmark functions, namely the Sphere, Rosenbrock, Ackley, and Griewank functions. For each test function, all combinations of the following sets of parameter values are adopted

$$\alpha = [10;\ 20;\ 30;\ 40;\ 50;\ 60;\ 70;\ 80;\ 90;\ 100]$$
$$\beta = [0.1;\ 0.2;\ 0.3;\ 0.4;\ 0.5;\ 0.6;\ 0.7;\ 0.8;\ 0.9;\ 1].$$

Through testing these functions, it can be found that their valley bottom with the optimum can be obtained for parameter ranges of $40 \le \alpha \le 60$ and $0.1 \le \beta \le 0.3$. Nevertheless, different problems may require a single value for each parameter, so the parameters of ASO are set as $\alpha = 50$ and $\beta = 0.2$ in the following experiments.

With the above formulation of ASO, the following remarks are made:

(1) ASO inherits the innate stochastic motion of atoms in the real world, hence it intrinsically has the high exploration ability in the search space and thus can well avoid being trapped into the local optima compared to its competitors.

(2) ASO is also a population-based optimization algorithm where the interaction forces include attraction and repulsion. The constraint force is an important media for delivering information within the population.

(3) The attraction and repulsion can guarantee the exploration and exploitation, respectively, with the lapse of iterations. The drift factor can enable the interaction forces exerted on the atoms to gradually switch from the combination of attraction and repulsion to the repulsion alone, thus indicating the switch from the exploration to the exploitation.

(4) In the former phase of ASO, whether the interaction forces exerted on the atoms show the attraction or the repulsion depends on the function value of the ratio of $r_{ij}(t)$ to $\sigma_i(t)$, and $\sigma_i(t)$ can adaptively adjust the category (attraction or repulsion) of the interaction forces acted on the atoms.

(5) The atoms with better fitness values have a larger mass, which leads to a smaller acceleration, thus signifying the local search. Atoms with worse fitness values have the lighter mass, thus signifying the global search.

(6) Each atom in the population interacts only with its neighbors *KBest* by the interaction force. The number of *KBest* gradually decreases with the lapse of iterations. Meanwhile, each atom and the best one always generate the constraint force at each iteration.

4. Experimental results

4.1. Benchmark functions

To test the performance of ASO algorithm extensively, it is employed to solve 37 well-known benchmark functions. These functions can be divided into five various types, including unimodal, multimodal, low-dimensional, hybrid, and composite functions, by which the performance of many different optimization algorithms can be efficiently measured. These benchmark functions [92–94] are summarized in Tables 1 and 2. Functions f_1–f_7 are unimodal functions and each has only one global optimum and no local optimum, so the convergence rate and exploitation of algorithms can be verified. Functions f_8–f_{13} are multimodal functions with a considerable number of local optimum. Functions f_{14}–f_{23} are low-dimensional functions, each of which has fewer local optimum. These multimodal and low-dimensional functions with local optima are highly suitable for test avoidance of local optima and exploration capacity of the algorithms. Functions f_{24}–f_{29} are hybrid functions, in which their variables are randomly separated into different subdivisions. These subdivisions are replaced by using either unimodal or multimodal functions. Additionally, functions f_{30}–f_{37} are composition functions, their variables are also randomly separated into different subdivisions. These subdivisions are constructed by using the basic and hybrid functions. Each composition function has different properties for different subdivisions. The global optimum is shifted from a specific position to a random position before each iteration, and occasionally the optimum is relocated on the boundary of search space [95]. Hybrid and composition test functions are more complex and challenging than the basic unimodal and multimodal test functions, thus resulting in more difficult optimization. Accordingly, they are especially suitable for testing the potential performance of the algorithms for solving real-world problems. A detailed description of these functions is available in [28].

4.2. Experimental setup and comparative algorithms

For these test functions, the proposed algorithm is compared with five stochastic optimization algorithms, including three classic and popular algorithms, PSO, GA and SA, and two recently proposed algorithms, GSA and WDO. Although a number of variants based on these algorithms have been developed, the comparisons of standard versions can be used to interpret the results of larger groups.

PSO [2] mimics behaviors of birds flocking in the sky. It updates velocities and positions of a population by using social group learning and individual learning to seek expected goals. PSO has a good local search ability.

GA [14], an evolutionary algorithm, is inspired from the biology evolutionary theory. It uses mutation, crossover and selection operations to generate high-quality solutions. GA tends to be excellent in finding good global solutions.

Table 1
Unimodal, multimodal and low-dimensional test functions.

Name	Function	n	Range	Optimum
Sphere	$f_1(x)=\sum_{i=1}^{n}x_i^2$	30	$[-100,100]^n$	0
Schwefel 2.22	$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	$[-10,10]^n$	0
Schwefel 1.2	$f_3(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2$	30	$[-100,100]^n$	0
Schwefel 2.21	$f_4(x)=\max_i\{\lvert x_i\rvert, 1\le i\le n\}$	30	$[-100,100]^n$	0
Rosenbrock	$f_5(x)=\sum_{i=1}^{n-1}(100(x_{i+1}-x_i)^2+(x_i-1)^2)$	30	$[-30,30]^n$	0
Step	$f_6(x)=\sum_{i=1}^{n}(x_i+0.5)^2$	30	$[-100,100]^n$	0
Quartic	$f_7(x)=\sum_{i=1}^{n}ix_i^4+random[0,1)$	30	$[-1.28,1.28]^n$	0
Schwefel	$f_8(x)=-\sum_{i=1}^{n}(x_i\sin(\sqrt{\lvert x_i\rvert}))$	30	$[-500,500]^n$	−12 569.5
Rastrigin	$f_9(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)^2$	30	$[-5.12,5.12]^n$	0
Ackley	$f_{10}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i)+20+e$	30	$[-32,32]^n$	0
Griewank	$f_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}(x_i-100)^2-\prod_{i=1}^{n}\cos(\frac{x_i-100}{\sqrt{i}})+1$	30	$[-600,600]^n$	0
Penalized	$f_{12}(x)=\frac{\pi}{n}\{10\sin^2(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_i+1)]$ $+(y_n-1)^2\}+\sum_{i=1}^{30}u(x_i,10,100,4)$	30	$[-50,50]^n$	0
Penalized2	$f_{13}(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{29}(x_i-1)^2p[1+\sin^2(3\pi x_{i+1})]$ $+(x_n-1)^2[1+\sin^2(2\pi x_{30})]\}+\sum_{i=1}^{30}u(x_i,5,10,4)$	30	$[-50,50]^n$	0
Foxholes	$f_{14}(x)=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right]^{-1}$	2	$[-65.536,65.536]^n$	0.998
Kowalik	$f_{15}(x)=\sum_{i=1}^{11}\left\lvert a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right\rvert^2$	4	$[-5,5]^n$	3.075×10^{-4}
Six Hump Camel	$f_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	$[-5,5]^n$	−1.0316
Branin	$f_{17}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	2	$[-5,10]\times[0,15]$	0.398
Goldstein-Price	$f_{18}(x)=\left[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\right]$ $\times\left[30+(2x_1+1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\right]$	2	$[-2,2]^n$	3
Hartman 3	$f_{19}(x)=-\sum_{i=1}^{4}\exp\left[-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right]$	3	$[0,1]^n$	−3.86
Hartman 6	$f_{20}(x)=-\sum_{i=1}^{4}\exp\left[-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right]$	6	$[0,1]^n$	−3.322
Shekel 5	$f_{21}(x)=-\sum_{i=1}^{5}\left\lvert(x_i-a_i)(x_i-a_i)^T+c_i\right\rvert^{-1}$	4	$[0,10]^n$	−10.1532
Shekel 7	$f_{22}(x)=-\sum_{i=1}^{7}\left\lvert(x_i-a_i)(x_i-a_i)^T+c_i\right\rvert^{-1}$	4	$[0,10]^n$	−10.4028
Shekel 10	$f_{23}(x)=-\sum_{i=1}^{10}\left\lvert(x_i-a_i)(x_i-a_i)^T+c_i\right\rvert^{-1}$	4	$[0,10]^n$	−10.5363

Table 2
Hybrid and composition test functions.

Function	Name	n	Range	Optimum
$f_{24}(x)$	Hybrid Function 1 ($N=3$)	30	$[-100,100]^n$	1700
$f_{25}(x)$	Hybrid Function 2 ($N=3$)	30	$[-100,100]^n$	1800
$f_{26}(x)$	Hybrid Function 3 ($N=4$)	30	$[-100,100]^n$	1900
$f_{27}(x)$	Hybrid Function 4 ($N=4$)	30	$[-100,100]^n$	2000
$f_{28}(x)$	Hybrid Function 5 ($N=5$)	30	$[-100,100]^n$	2100
$f_{29}(x)$	Hybrid Function 6 ($N=5$)	30	$[-100,100]^n$	2200
$f_{30}(x)$	Composition Function 1 ($N=5$)	30	$[-100,100]^n$	2300
$f_{31}(x)$	Composition Function 2 ($N=3$)	30	$[-100,100]^n$	2400
$f_{132}(x)$	Composition Function 3 ($N=3$)	30	$[-100,100]^n$	2500
$f_{33}(x)$	Composition Function 4 ($N=5$)	30	$[-100,100]^n$	2600
$F_{34}(x)$	Composition Function 5 ($N=5$)	30	$[-100,100]^n$	2700
$f_{35}(x)$	Composition Function 6 ($N=5$)	30	$[-100,100]^n$	2800
$F_{36}(x)$	Composition Function 7 ($N=3$)	30	$[-100,100]^n$	2900
$f_{37}(x)$	Composition Function 8 ($N=3$)	30	$[-100,100]^n$	3000

SA [1] takes inspiration from annealing in metallurgy by heating material and cooling it at a certain rate. It is a probabilistic algorithm for seeking the global optimum in the search space and in a fixed amount of time.

GSA [28], based on the law of gravity, makes agents interact according to the law of motion and is a competitive search algorithm. This interaction can generate an attractive force that facilitates all agents globally moving to the agent with the heavier mass.

WDO [96] is based on the earth's atmosphere motion wherein each small air parcel moves following Newton's second law. It updates the velocity and position of each parcel by the sum of a gradient force, Coriolis force, gravitational force, and friction force to approximate the global optimum. WDO has a good global search ability.

It should be noted that, for each benchmark function, it is time-consuming and difficult for each algorithm to search for a set of appropriate parameters to enhance its optimization performance. Hence, for the sake of fairness, a set of fixed parameters are adopted for each algorithm to evaluate the entire performance of all the test functions. The initial parameters used in all algorithms are provided in Table 3. In this experiment, the size of the population and the maximum number of iterations are set to 50 and 1000, respectively, for all the algorithms. In addition, every algorithm runs 50 times for each function, and the results are based on the average performance of the 50 runs.

4.3. Results and discussion

4.3.1. Qualitative results of ASO

First, five different functions are randomly chosen from five types of benchmark functions and simulated to see the effects of the proposed ASO algorithm qualitatively. The functions chosen are f_5, f_{11}, f_{15}, f_{25} and f_{32} with $D=2$, whose shapes are presented in Figs. 9–13, respectively. ASO is used for their minimization using four atoms during 150 iterations. The diversity of the benchmark functions makes for comprehensive and easy observations of the algorithm performance from different views.

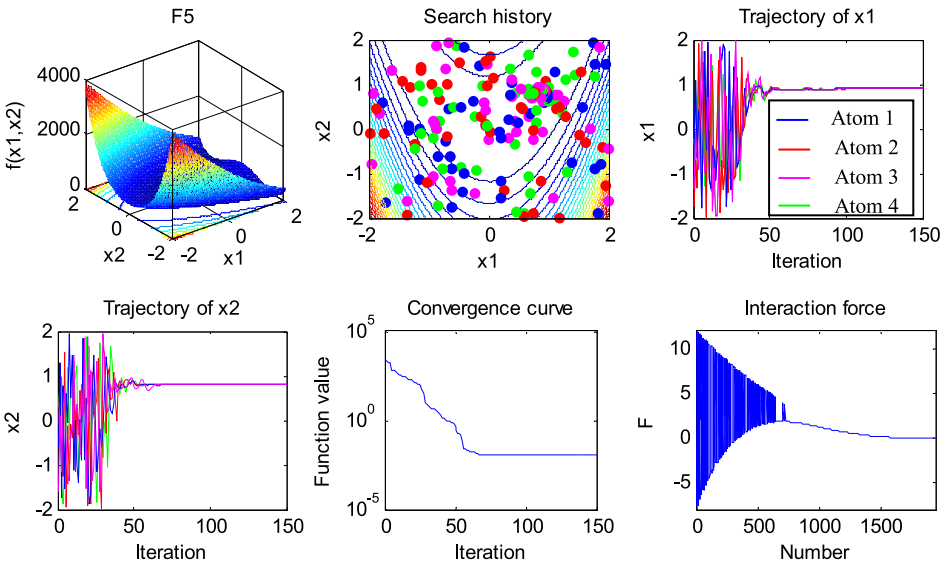

Fig. 9. Qualitative results of f_5.

Table 3
Parameter settings for all algorithms.

Algorithm	Parameter	Value
ASO	Depth weight	50
	Multiplier weight	0.2
WDO	RT coefficient	3
	Gravitational constant	0.2
	Constant in the update equations	0.4
	Coriolis effect	0.4
	Maximum allowable speed	0.3
SA	Initial temperature	0.1
	Temperature reduction rate	0.98
	Mutation rate	0.5
PSO	Cognitive constant	2
	Social constant	2
	Inertia constant	Linearly decrease from 0.8 to 0.2
GSA	Initial gravitational constant	100
	Decreasing coefficient	20
GA	Selection	Roulette wheel
	Crossover	0.8
	Mutation	0.4

Four qualitative metrics used to describe the performance of ASO include search history, trajectory curve, convergence curve, and interaction force. All history positions of the four atoms during the iterations are clearly depicted in the figures, and the distribution density of atoms in the search space can indicate how ASO performs the explorative and exploitative search. Apparently, the low distribution density indicates the explorative search for the global space, and the high distribution density indicates the exploitative search for the local space. It is evident that the distribution of atoms is sparse in the region farther from the global optimum and is dense in the region closer to the global optimum.

The trajectory curve is one of the most important metrics that can effectively depict exploration and exploitation of the algorithm. It can trace positions of all atoms in different dimensions during iterations. Figs. 9–13 illustrate the trajectory curves when the five functions are optimized using the ASO algorithm. In the figures, the trajectory curves of four atoms in the first and the second dimensions are depicted. All trajectory curves show frequent large-scale fluctuations in the early iterations. With the lapse of iterations, such variation decreases in amplitude or frequency, and the positions of atoms become monotonous and gradually tend to stabilize to the global optimum in the later iterations. Evidently, the large-scale fluctuations in the former iterations indicate the explorative search for the global space, and the small-scale fluctuations in the latter iterations indicate the exploitative search for the local space. These two behaviors are consistent with what is shown in the search histories. What is more, it can be observed that ASO performs the exploration before the exploitation in the trajectory curves. Therefore, the trajectory curves together with the search histories reveal how well ASO conducts the explorative and exploitative search.

For all optimization algorithms, their final goal is to find the approximation to the global optimum quickly and precisely, so how to show this behavior is of great importance. But there is nothing the trajectory curves and search histories can do about it. The convergence curve is the most often used qualitative metric in evaluating the convergence performance of the algorithms. It is apparent from the figures that the convergence curves can clearly depict the convergence rate and the approximation to the global optimum. As shown in the figures, the convergence curve of function f_5 is very smooth and drops rapidly, demonstrating that the exploitation contributes more to ASO than the exploration. In contrast, for functions f_{11}, f_{15}, f_{25} and f_{32}, their convergence curves are very rough and drop slowly, which indicates that the exploration contributes more to ASO than the exploitation. Finally, their convergence curves can all accurately approximate the global optimum in the final iterations.

A large number of real-world problems can be considered multimodal functions with a considerable number of local optima [97]. For solving these problems, the exploration is very important to avoid the local optima and discover a promising region with the global optimum. In Figs. 9–13, the interaction forces between every two atoms during the whole iterations are provided. The interaction forces are shown as either the repulsion (negative) or the attraction (positive) in the former stage and only as the attraction in the latter stage. Besides, the interaction forces gradually decrease with the lapse of iterations. The attraction in the latter stage can prevent the atom swarm from prematurely concentrating in a local region, which can contribute significantly to the exploration of ASO.

Based on the above results and discussion, the four qualitative metrics can show how well the exploration and exploitation perform, the convergence performance, and the accuracy of the final approximation to the global optimum. The results indicate

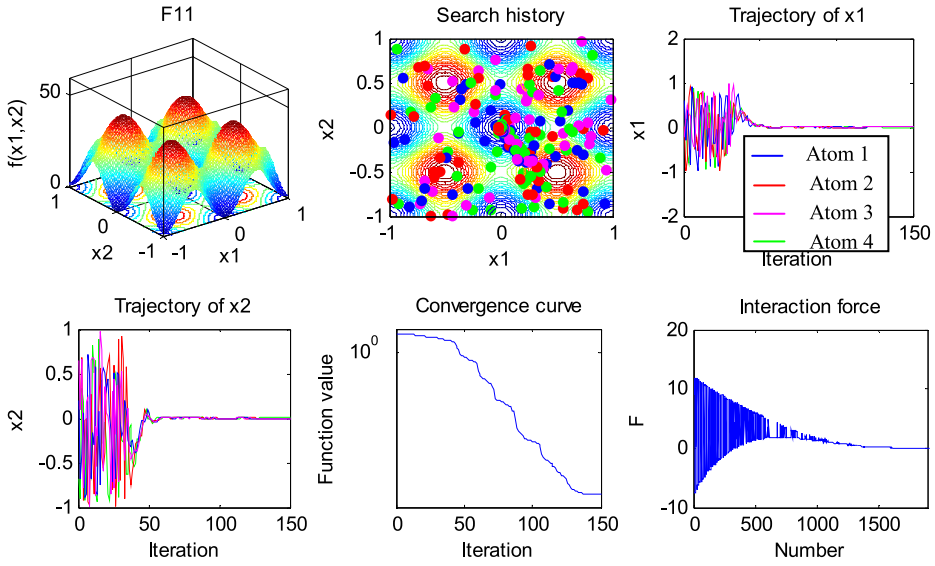

Fig. 10. Qualitative results of f_{11}.

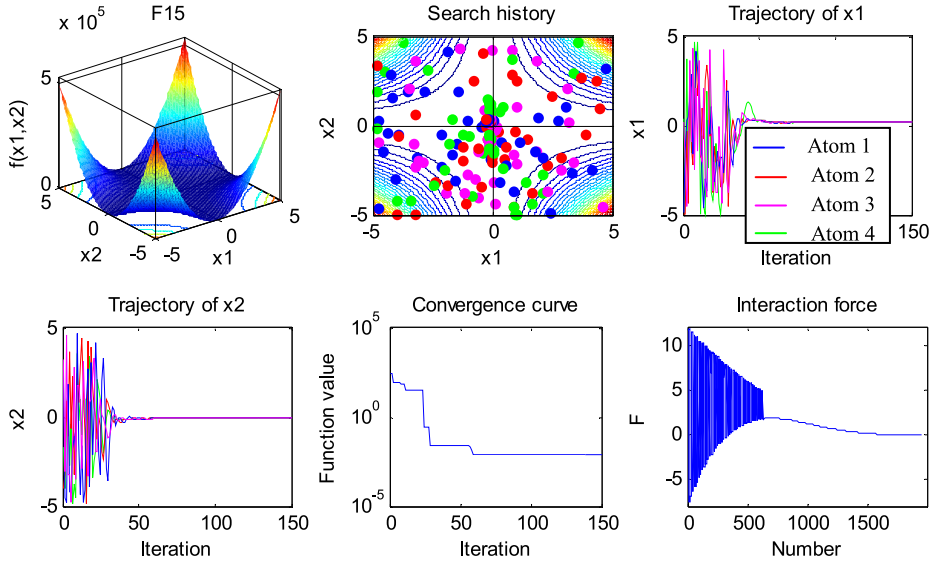

Fig. 11. Qualitative results of f_{15}.

that ASO establishes a proper balance between the exploration and exploitation. In addition, the search performance for the global optimum is also satisfactory.

4.3.2. Convergence preference of the algorithms

Although the qualitative metric results reveal that ASO has a good optimization performance, they cannot indicate how well the proposed ASO performs for all benchmark functions. In this subsection, ASO is evaluated using some performance evaluation indexes, which are compared to those of the other algorithms. To obtain statistical solutions and better evaluate the algorithms, there are three performance evaluation indexes used to quantitatively compare all the algorithms: the average, standard deviation and minimum of the best-so-far solution. Apparently, among all the 50 runs, the lower the average of the best-so-far solution is, the greater the algorithm's ability to avoid local optima and approximate a global optimum is. Also, the lower the standard deviation of the best-so-far solution is, the closer the solution is to the average. Further, the more stable the algorithm is, the better the minimum of the best-so-far solution is, and the more accurate the approximation is to the global optimum. The average of the best-so-far solution among the comparative algorithms on each function is shown in bold.

Tables 4–8 show the comparisons of optimization results obtained for different types of functions given in Tables 1–2, and the convergence processes of all comparative algorithms are shown in Figs. 14–18. As described previously, there is no optimization algorithm that can best perform for all different types of problems, hence the performance evaluation of the proposed algorithm should come from different perspectives by using multifarious benchmark functions [98].

For the unimodal functions f_1–f_7, from Fig. 14 and Table 4, ASO performs better than the others, except for WDO on functions f_1–f_4, and performs the best for function f_5. Although ASO fails to achieve the best convergence performance for functions f_6 and f_7, it performs better than PSO, GA and SA. Specifically, the convergence rate of ASO on some functions is slower than that of the other algorithms in the early iterations because of the repulsion among atoms, which makes them perform a global search in the entire search space to prevent the optimization from prematurity. In

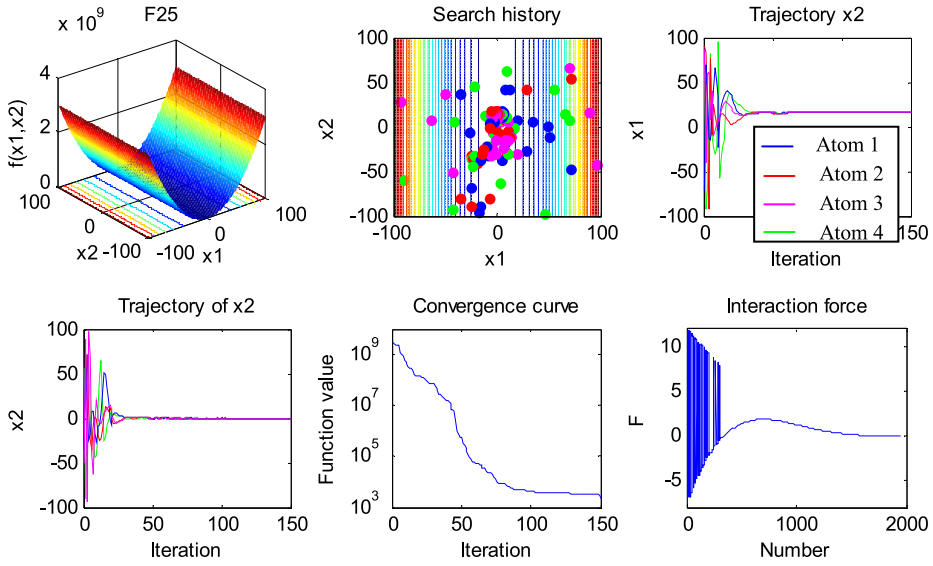

Fig. 12. Qualitative results of f_{25}.

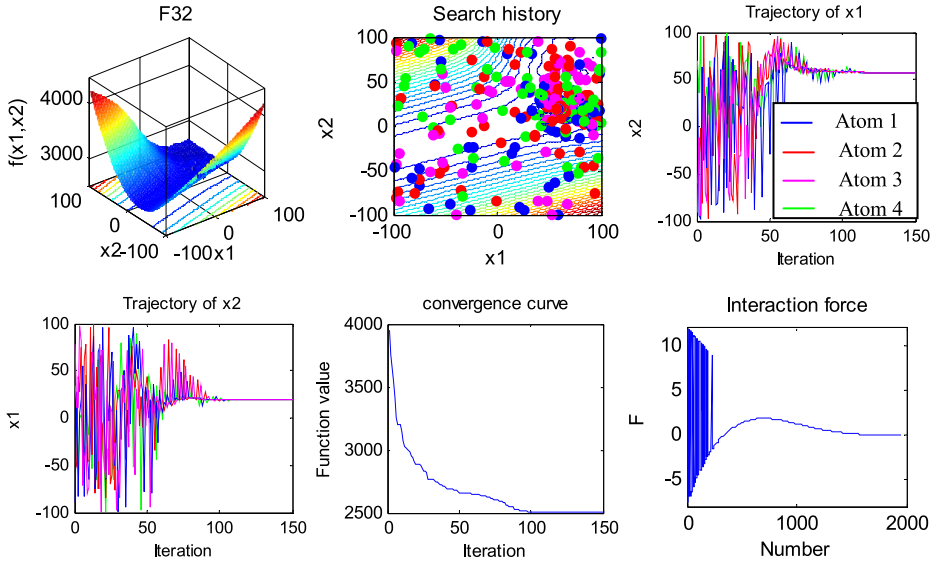

Fig. 13. Qualitative results of f_{32}.

addition, WDO and GSA are also excellent and novel stochastic algorithms developed recently that have a certain advantage in terms of the convergence rate. Even so, ASO is still competitive in terms of the convergence rate for unimodal functions. The final solutions in Table 4 show that ASO achieves better results than the others on functions f_5 and has the same results as WDO, GSA and GA on function f_6. It ranks behind only WDO on functions f_1–f_4. Therefore, ASO can achieve a good precision for the unimodal functions.

For the multimodal functions f_8–f_{13}, from Fig. 15 and Table 5, ASO performs much better than the others, except for functions f_8 and f_{10}. ASO is inferior to SA on function f_8 and is outperformed by WDO on function f_{10}. However, WDO shows the worst results for functions f_9 and f_{13}, and GSA shows the worst results for functions f_8 and f_{11}. Although SA has the best results on function f_8, the algorithm shows worse results than the others on functions f_{10} and f_{12}, indicating that the three algorithms all fail to step out of the local optimum in some multimodal functions. For ASO, the repulsion can effectively avoid prematurity and greatly improve the exploration in the early stage. In the later stage, the stronger attraction force and the constraint force jointly contribute to the fine search for the global optimum.

For low-dimensional functions f_{14}–f_{23}, which are multimodal functions with low-dimension and a few local optima. Fig. 16 and Table 6 show that all the algorithms provide nearly the same performance for functions f_{16}, f_{17} and f_{19}. They also perform similarly well except for GA for function f_{18}, and ASO obtains the same good results as SA and GSA on functions f_{22} and f_{23}. ASO also performs the best for functions f_{14} and f_{20}, ranking behind only SA for function f_{21}. Even for function f_{15}, ASO performs better than WDO, GSA and GA. Obviously, ASO is highly competitive for these low-dimensional functions with local optima.

For hybrid functions f_{24}–f_{29}, Fig. 17 and Table 7 show that obviously, ASO performs the best on functions f_{24}, f_{26} and f_{28}. For functions f_{25}, f_{27} and f_{29}, ASO ranks only second to GSA, WDO and SA, respectively. In addition, GSA shows the worst convergence performance and results on functions f_{26}–f_{29}, and PSO performs the worst on functions f_{24} and f_{25}. It should be noted that the variables of the hybrid functions are randomly separated into different subdivisions, which are replaced by using unimodal and multimodal

Table 4
Comparisons of results for unimodal functions.

Function	Index	WDO	SA	PSO	GSA	GA	ASO
$f_1(x)$	Mean	**0**	2.04E−13	0.000146	2.11E−17	0.010212	2.68E−21
	Std	0	6.14E−14	0.000119	6.67E−18	0.005073	3.65E−21
	Best	0	7.76E−14	9.56E−06	1.06E−17	0.002975	3.52E−22
$f_2(x)$	Mean	**0**	9.73102	0.000198	2.37E−08	0.020789	3.33E−10
	Std	0	7.34169	0.000188	2.60E−09	0.005434	1.89E−10
	Best	0	0.003379	5.45E−05	1.71E−08	0.012651	5.34E−11
$f_3(x)$	Mean	**6.73E−29**	4554.414	2659.282	251.2258	1115.564	197.5452
	Std	3.69E−28	2329.664	1194.893	41.96643	493.2294	79.7024
	Best	0	1527.615	760.8198	179.2156	401.8981	23.1228
$f_4(x)$	Mean	**0**	6.399475	18.26166	3.35E−09	0.994831	3.24E−09
	Std	0	3.284424	4.247802	7.53E−9	0.354854	6.14E−09
	Best	0	1.338689	10.65944	2.28E−09	0.547858	2.13E−10
$f_5(x)$	Mean	28.19528	1059.101	134.3922	28.1473	97.11795	**24.8388**
	Std	0.168679	2050.492	128.5852	11.29836	128.7488	0.515853
	Best	27.95513	23.1826	26.23857	25.75286	9.696092	16.58185
$f_6(x)$	Mean	**0**	0.566667	0.133333	**0**	**0**	**0**
	Std	0	0.727932	0.345746	0	0	0
	Best	0	0	0	0	0	0
$f_7(x)$	Mean	**6.10E−05**	0.123989	0.066316	0.02076	0.0505	0.035641
	Std	4.49E−05	0.039739	0.019519	0.007716	0.021781	0.019498
	Best	1.43E−06	0.062475	0.029819	0.007006	0.0132	0.03615

Table 5
Comparisons of results for multimodal functions.

Function	Index	WDO	SA	PSO	GSA	GA	ASO
$f_8(x)$	Mean	−5840.86	**−9289.96**	−5197.85	−2653.67	−6795.16	−7428.17
	Std	857.0229	402.1469	529.3583	341.6006	632.7758	422.3977
	Best	−7594.51	−10 055.4	−6787.39	−3488.44	−8147.59	−5561.44
$f_9(x)$	Mean	57.69854	54.39099	29.31833	15.05704	12.52966	**0**
	Std	21.2409	13.82194	6.879953	4.439862	2.915823	0
	Best	15.56306	26.86388	17.34745	7.959667	6.094298	0
$f_{10}(x)$	Mean	**8.88E−16**	0.343794	0.007432	3.69E−09	0.021188	3.00E−11
	Std	0	0.447427	0.014159	3.96E−10	0.004548	2.15E−11
	Best	8.88E−16	8.29E−08	0.000549	2.96E−09	0.010696	1.13E−11
$f_{11}(x)$	Mean	0.009891	0.012439	0.022795	4.472721	0.018359	**0**
	Std	0.021887	0.010314	0.02737	2.048563	0.009772	0
	Best	0	3.16E−06	5.88E−05	1.944629	0.006121	0
$f_{12}(x)$	Mean	0.03139	0.698388	0.337564	0.020732	2.80E−05	**4.51E−23**
	Std	0.087872	0.647321	0.429656	0.057102	1.89E−05	1.88E−23
	Best	0.000355	0.046437	2.54E−05	7.00E−20	7.49E−06	8.69E−25
$f_{13}(x)$	Mean	0.548891	0.514392	0.232059	0.00155	0.000452	**1.91E−23**
	Std	0.931434	1.324152	0.71351	0.003696	0.00069	3.12E−22
	Best	0.010295	2.77E−05	0.000487	1.31E−18	0.000119	2.31E−24

functions, directly affecting the optimization performance. The results show that the optimization performance of GSA and PSO is not satisfactory for many hybrid functions, while ASO achieves a good optimization performance on them.

For composite functions f_{30}–f_{37}, Fig. 18 and Table 8 show that ASO performs a better convergence and obtains better statistical results than the others on functions f_{33}, f_{34}, f_{36} and f_{37}. For functions f_{30} and f_{35}, ASO ranks only second to GSA and SA, respectively. In addition, ASO ranks third to WDO and GSA on function f_{31}. Even for function f_{32}, ASO is also superior to GA and PSO. Because the subfunctions from the basic functions contribute to too many local optima, it is very difficult for the algorithms to find the global optimum. It can clearly be shown that all the algorithms except for ASO are trapped into local optima at different levels on functions f_{33}, f_{34}, f_{36} and f_{37}. This is due to the fact that the repulsion of ASO contributes significantly to the exploration and avoids the local optima efficiently.

Tables 4–8 and Figs. 14–18 show that from the perspective of the convergence performance, ASO is highly competitive compared to its competitors on the benchmark functions, including unimodal, multimodal, low-dimensional, hybrid and composition functions.

4.3.3. Overall performance of the algorithms

To draw a statistical conclusion, a pair-wise statistical test named Wilcoxon Signed-Rank Test (WSRT) is used to better compare the overall performance of the algorithms. WSRT is a nonparametric test that can be used to check for the statistical significance difference between two algorithms. The null hypothesis H_0 for a two-sided test is: "there is no difference between the median of the solutions produced by algorithm A and the median of the solutions produced by algorithm B for the same benchmark function" [99]. To determine whether algorithm A achieves a statistically better solution than algorithm B, or if not, whether the alternative hypothesis is valid, the sizes of the ranks provided by WSRT are examined. When using WSRT, the R+ and R− related to the comparisons between two algorithms can be calculated and their p-values can be obtained. In this part, WSRT is used for single-problem-based statistical analysis and multi-problem-based statistical analysis, with the significance level $a = 0.05$.

Table 6
Comparisons of results for low-dimensional functions.

Function	Index	WDO	SA	PSO	GSA	GA	ASO
$f_{14}(x)$	Mean	5.311273	1.395219	**0.998004**	4.215728	3.92122	**0.998004**
	Std	5.651641	0.669666	3.41e−13	3.636458	2.697616	7.40e−17
	Best	0.998004	0.998004	0.998004	0.998021	0.998004	0.998004
$f_{15}(x)$	Mean	0.001089	7.67E−04	**3.20E−04**	0.002059	0.001259	9.47E−04
	Std	0.003656	2.19E−04	3.64E−05	8.64E−04	0.001909	2.27E−04
	Best	3.08E−04	4.78E−04	3.07E−04	9.98E−04	3.25E−04	2.79E−04
$f_{16}(x)$	Mean	−1.03162	**−1.03163**	**−1.03163**	**−1.03163**	**−1.03163**	**−1.03163**
	Std	1.00E−05	6.39E−16	6.78E−16	5.68E−16	7.56E−10	0
	Best	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163	−1.03163
$f_{17}(x)$	Mean	0.39791	**0.397887**	**0.397887**	**0.397887**	**0.397887**	**0.397887**
	Std	1.90E−05	0	6.09E−08	0	3.95E−08	0
	Best	0.397888	0.397887	0.397887	0.397887	0.397887	0.397887
$f_{18}(x)$	Mean	3.000224	**3**	**3**	**3**	3.9	**3**
	Std	0.000385	1.28E−15	1.32E−15	1.99E−15	4.929503	1.51E−15
	Best	3	3	3	3	3	3
$f_{19}(x)$	Mean	−3.86273	**−3.86278**	**−3.86278**	**−3.86278**	**−3.86278**	**−3.86278**
	Std	0.00011	2.71E−15	2.70E−15	2.46E−15	2.72E−09	2.68E−15
	Best	−3.86278	−3.86278	−3.86278	−3.86278	−3.86278	−3.86278
$f_{20}(x)$	Mean	−3.26356	−3.27717	−3.2702	**−3.322**	−3.30614	**−3.322**
	Std	0.05942	0.056126	0.060244	1.36E−15	0.041107	1.12E−15
	Best	−3.32082	−3.322	−3.322	−3.322	−3.322	−3.322
$f_{21}(x)$	Mean	−6.89682	**−9.61699**	−5.53835	−7.18959	−8.73973	−8.774464
	Std	3.606535	1.642171	2.55709	3.354687	2.904334	2.186718
	Best	−10.1532	−10.1532	−10.1532	−10.1532	−10.1532	−10.1532
$f_{22}(x)$	Mean	−7.16466	**−10.4029**	−8.23528	**−10.4029**	−8.68393	**−10.4029**
	Std	3.580271	3.30E−16	3.17987	4.66E−16	3.176461	1.84E−15
	Best	−10.4029	−10.4029	−10.4029	−10.4029	−10.4029	−10.4029
$f_{23}(x)$	Mean	−6.34454	**−10.5364**	−8.63826	**−10.5364**	−8.81243	**−10.5364**
	Std	4.002617	2.91E−15	3.015894	1.75E−15	3.212375	1.54E−15
	Best	−10.5363	−10.5364	−10.5364	−10.5364	−10.5364	−10.5364

Table 7
Comparisons of results for hybrid functions.

Function	Index	WDO	SA	PSO	GSA	GA	ASO
$f_{24}(x)$	Mean	636 845.7	826 767.3	3 778 947	4 974 916	2 123 123	**452 502.66**
	Std	400 326.4	859 291.4	3 856 484	2 230 865	1 146 082	538 536.4
	Best	114 642.2	53 802.62	413 267.5	2 197 117	606 506.3	10 509.77
$f_{25}(x)$	Mean	473 538.8	9897.176	1 892 268	**2360.049**	2922.662	2521.198
	Std	167 046.5	8871.518	9 041 101	402.0756	884.0251	1001.254
	Best	231 041.9	2319.467	1880.679	2051.498	1899.538	1973.163
$f_{26}(x)$	Mean	1932.082	1912.644	1946.653	2048.729	1927.013	**1911.15**
	Std	30.64479	2.039958	33.06431	40.27749	24.00536	0.742535
	Best	1907.461	1907.34	1907.134	1929.327	1910.308	1909.686
$f_{27}(x)$	Mean	**17 305.09**	57 189.46	28 438.85	146 057.1	39 156.38	21 999.82
	Std	6298.762	25 860.57	11 237.46	43 914.36	14 032.56	8835.043
	Best	4085.151	15 530.32	11 915.96	82 249.38	13 558.32	8941.514
$f_{28}(x)$	Mean	222 605.1	200 164	793 350.6	1 330 821	460 277.3	**22 083.121**
	Std	106 098.9	193 452.9	761 226.8	1 065 693	504 584.7	16 323.2
	Best	57 142.06	33 146.05	**14 201.94**	346 135.4	67 981.71	16 885.179
$f_{29}(x)$	Mean	2872.024	**2463.856**	2831.22	3267.385	2825.28	2756.953
	Std	183.498	116.4591	265.9089	293.3663	203.9491	204.2695
	Best	2431.908	2282.106	2370.504	2628.743	2348.308	2360.424

For the single-problem-based statistical analysis, the best solutions of 50 runs for every benchmark function are used for their pair-wise comparisons. The single-problem-based statistical comparisons of the algorithms by WSRT are shown in Tables 9 and 10. In these tables, '+' indicates the case in which the null hypothesis is rejected and ASO shows a better performance in the single-problem-based statistical comparison tests at 95% significance level ($a = 0.05$); '−' indicates the case in which the null hypothesis is rejected and ASO shows a worse performance; and '=' indicates a case in which no statistically significant difference between ASO and the other algorithms exists. The corresponding statistical results for each function in 50 runs are listed in Table 11. This table shows that for unimodal functions, ASO provides better results than the others, except for WDO. For multimodal functions, ASO performs significantly better than the others. For low-dimension functions, the performance of ASO is not inferior to that of the others, and it seems that GSA is just as competitive as ASO. For hybrid functions, ASO performs significantly better than the others. For composition functions, ASO significantly outperforms the other algorithms. It can be seen that from Table 11, ASO is

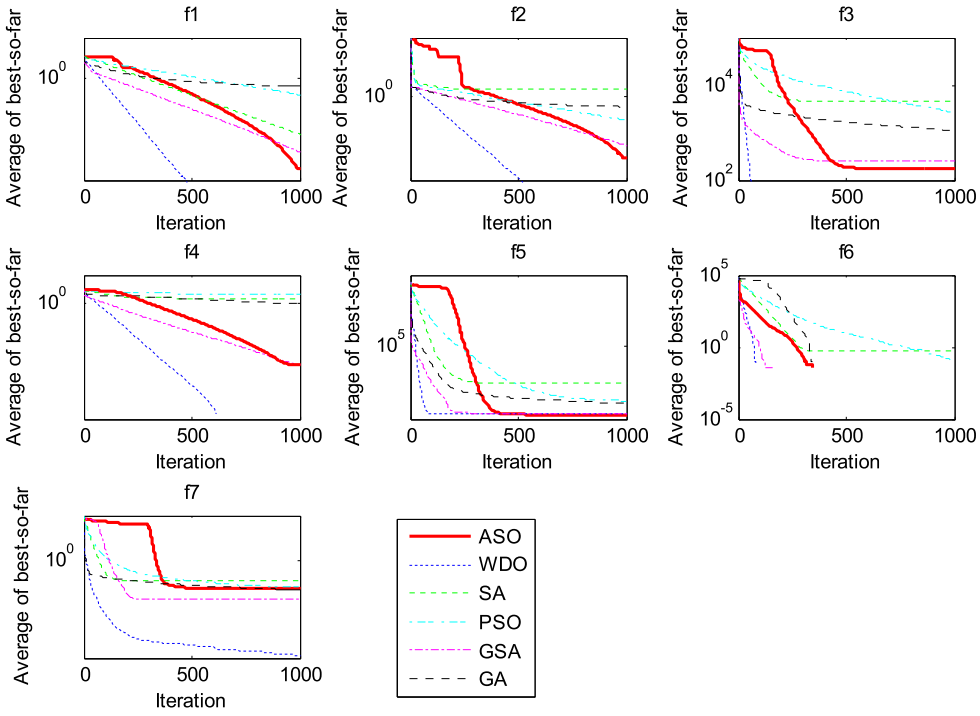

Fig. 14. Performance comparisons of algorithms for unimodal functions.

Table 8
Comparisons of results for composition functions.

Function	Index	WDO	SA	PSO	GSA	GA	ASO
$f_{30}(x)$	Mean	2624.533	2632.702	2645.122	**2534.215**	2616.423	2616.1100
	Std	2.286212	10.07164	11.87371	63.42403	0.438999	1.941234
	Best	2620.749	2620.702	2628.129	2500	2615.828	2615.248
$f_{31}(x)$	Mean	**2600**	2637.749	2632.222	2603.002	2630.226	2624.429
	Std	4.73E−05	5.807932	7.351382	6.04137	1.804637	4.75412
	Best	2600	2624.82	2625.289	2600.112	2627.5	2600.568
$f_{32}(x)$	Mean	**2700**	2710.652	2722.703	2701.248	2713.621	2710.987
	Std	0	2.170978	7.350302	2.489568	2.764908	2.35979
	Best	2700	2707.026	2710.674	2700	2708.848	2700
$f_{33}(x)$	Mean	2766.93	2706.882	2704.985	2793.516	2793.486	**2700.254**
	Std	4.169525	25.60925	18.07296	16.97263	25.30876	0.04997
	Best	2700.599	2700.073	2700.389	2747.814	2700.357	2700.133
$f_{34}(x)$	Mean	3606.247	3206.937	3384.102	4372.258	3468.123	**3126.185**
	Std	309.7327	80.83431	208.4262	411.4156	345.4187	35.12669
	Best	3113.73	3110.855	3151.307	3054.283	3104.8	3102.133
$f_{35}(x)$	Mean	7136.114	**3844.078**	7047.004	5169.593	6658.137	4795.1736
	Std	967.6085	162.4164	698.6452	884.0748	541.6948	286.4972
	Best	4984.829	3636.991	5780.443	3946.267	5431.646	3894.87
$f_{36}(x)$	Mean	1 006 116	1 002 510	13 894 512	2 023 296	8380.851	**7366.303**
	Std	4 951 051	3 802 991	19 870 463	11 065 067	1290.275	1360.375
	Best	30 874.43	7012.566	5756.596	5100.102	5285.181	5024.042
$f_{37}(x)$	Mean	62 783.62	29 698.5	148 713.6	195 076.3	13 699.9	**12 028.283**
	Std	22 308.01	11 159.28	174 091.8	88 263.52	3855.628	2022.0797
	Best	7525.18	12 019.36	16 603.94	8200.015	8526.658	7651.182

evidently superior to all other algorithms in five different types of benchmark functions. For the multi-problem-based statistical analysis, the mean values of the best solutions of 50 runs for every benchmark function are used for the pair-wise comparisons. Thirty-seven mean values of all the functions in Table 12 are used for the input data of WSRT. For ASO, the sum of R+ is more than the sum of R− in each pair of comparisons. The statistical analysis shows that the optimization performance of ASO is far superior to that of its counterparts quantitatively.

5. Application to hydrogeologic parameter estimation

In this part, a hydrogeologic parameter estimation problem is considered. Without considering the effects of aquitard storage, the drawdown data of leaky aquifers is from the Hantush model, which is expressed as [100–103]

$$\lg s = \lg W\left(u, \frac{r}{B}\right) + \lg \frac{Q}{4\pi T} \tag{33}$$

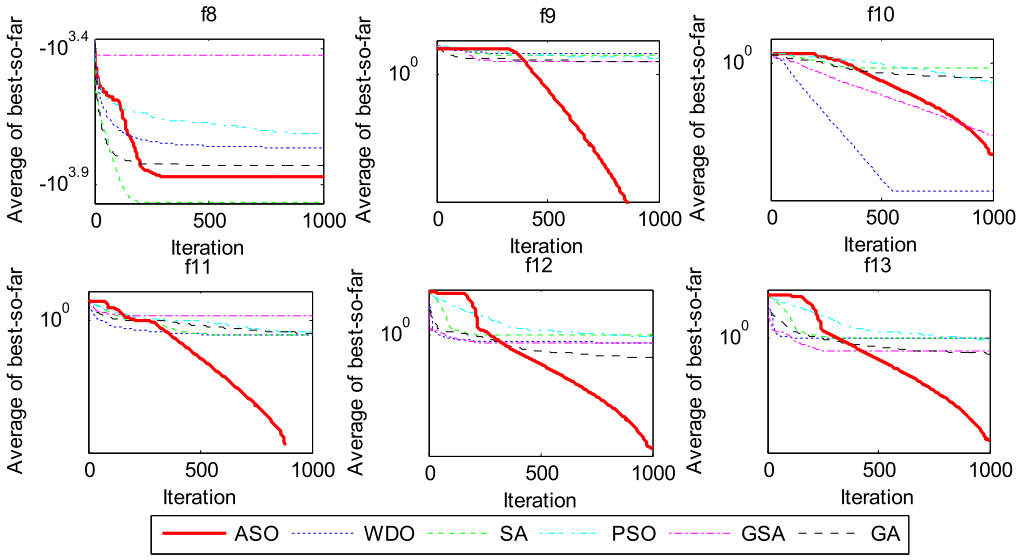

Fig. 15. Performance comparisons of algorithms for multimodal functions.

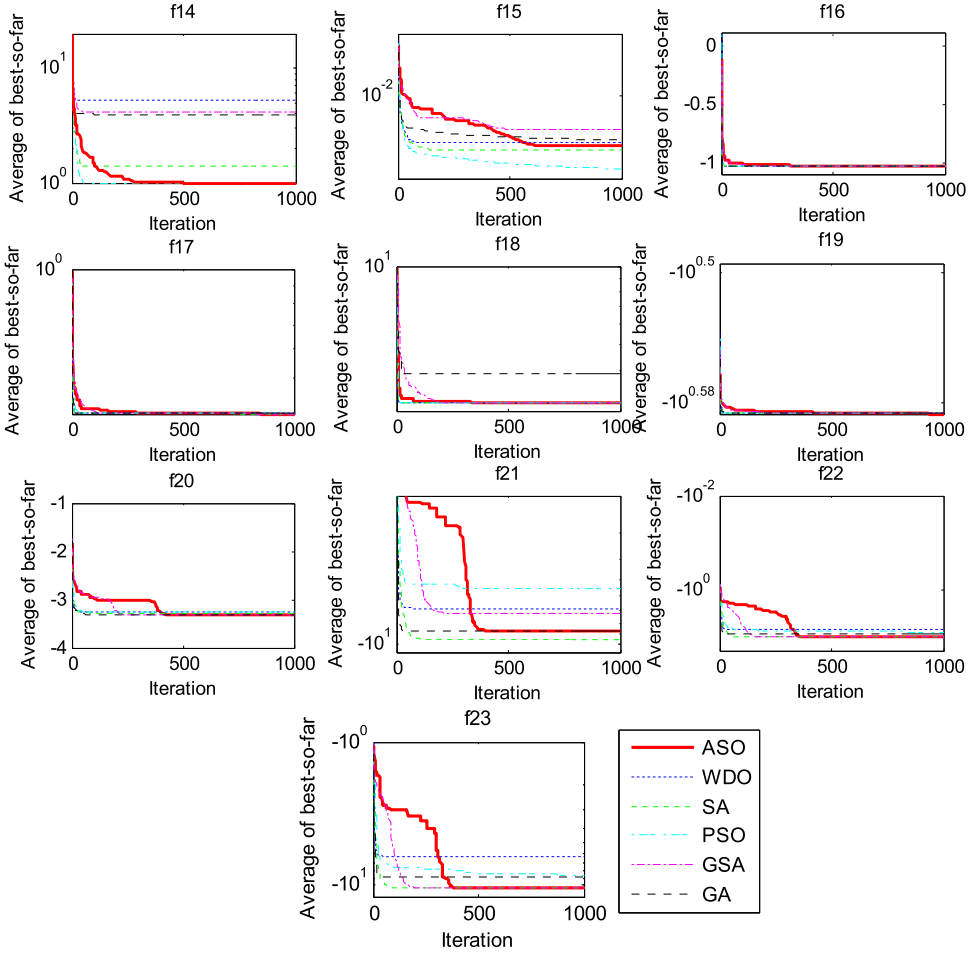

Fig. 16. Performance comparisons of algorithms for low-dimensional functions.

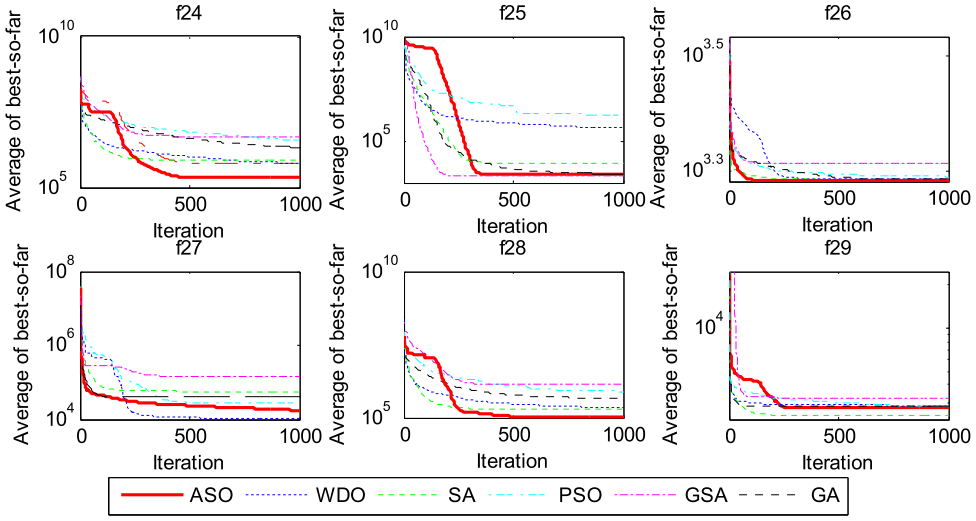

Fig. 17. Performance comparisons of algorithms for hybrid functions.

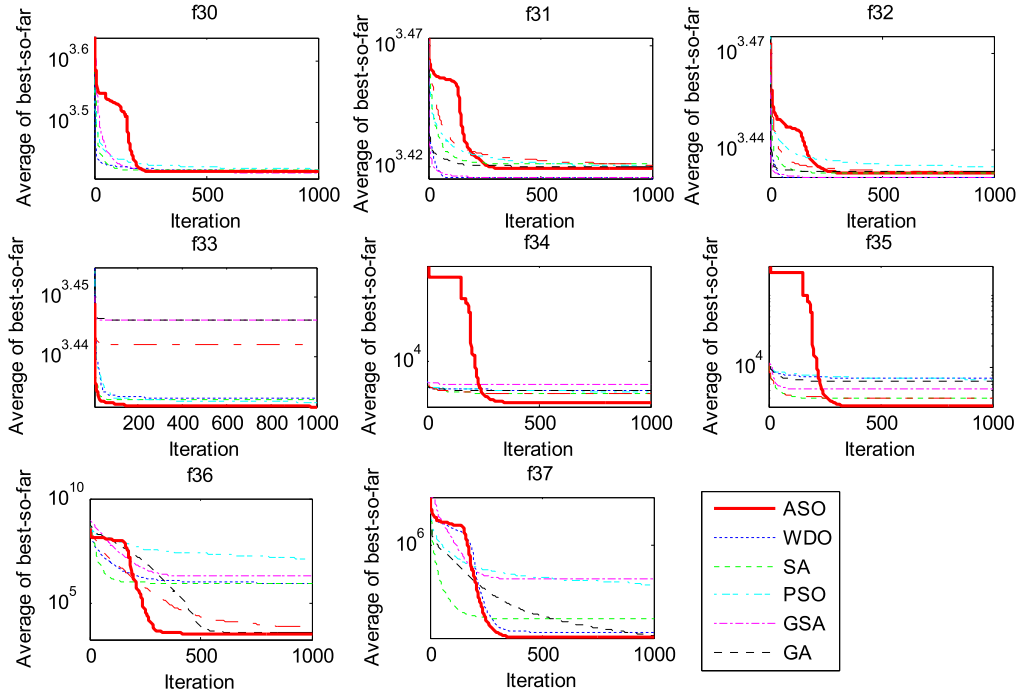

Fig. 18. Performance comparisons of algorithms for composition functions.

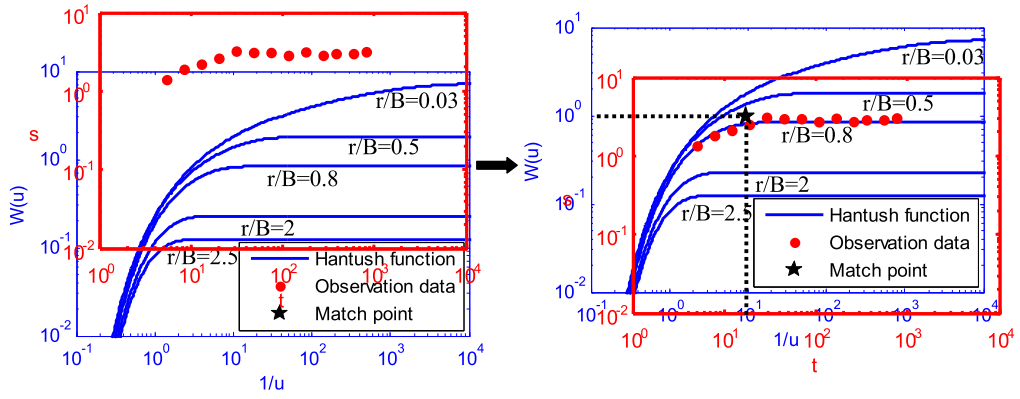

Fig. 19. Curve fitting procedure of leaky aquifer without aquitard storage.

Table 9
Statistical comparisons of single-problem-based WSRT for ASO vs. WDO, SA and PSO.

Function	ASO vs. WDO				ASO vs. SA				ASO vs. PSO			
	p-value	T−	T+	Winner	*p*-value	T−	T+	Winner	p-value	T−	T+	Winner
$f_1(x)$	7.52E−10	0	1275	−	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_2(x)$	7.52E−10	0	1275	−	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_3(x)$	7.52E−10	0	1275	−	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_4(x)$	7.52E−10	0	1275	−	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_5(x)$	7.52E−10	1275	0	+	9.03E−10	1272	3	+	7.52E−10	1275	0	+
$f_6(x)$	1	0	0	=	2.18E−05	231	1044	−	0.007813	1239	36	+
$f_7(x)$	7.52E−10	0	1275	−	2.50E−09	1255	20	+	0.988446	636	639	=
$F_8(x)$	9.60E−08	85	1190	−	7.52E−10	0	1275	−	0.001467	308	967	−
$F_9(x)$	9.14E−07	1146	129	+	1.13E−07	1187	88	+	0.003834	338	937	−
$f_{10}(x)$	7.52E−10	0	1275	−	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_{11}(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_{12}(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_{13}(x)$	7.52E−10	1275	0	+	3.35E−09	1250	25	+	7.52E−10	1275	0	+
$f_{14}(x)$	1.47E−07	1182	93	+	1.23E−06	1140	135	+	1	0	0	=
$f_{15}(x)$	1.23E−06	1140	135	+	0.000311	264	1011	−	7.52E−10	0	1275	−
$f_{16}(x)$	7.52E−10	1275	0	+	0.109375	44	11	=	0.125	0	10	=
$f_{17}(x)$	7.52E−10	1275	0	+	1	0	0	=	0.5	3	0	=
$f_{18}(x)$	7.52E−10	1275	0	+	8.84e−07	1263.5	11.5	+	0.098	660	615	=
$f_{19}(x)$	7.52E−10	1275	0	+	0.25	0	6	=	1	6	9	=
$f_{20}(x)$	7.52E−10	1275	0	+	5.75E−06	902	373	+	5.90E−05	1044	231	+
$f_{21}(x)$	0.67451	681	594	=	4.19E−06	110	1165	−	0.498087	654	621	=
$f_{22}(x)$	2.92E−08	1212	63	+	0.671875	12	16	=	0.000135	833	442	+
$f_{23}(x)$	1.49E−08	1224	51	+	0.475681	149.5	201.5	=	0.218917	389	886	=
$f_{24}(x)$	0.073311	823	452	=	0.160106	783	492	=	1.77E−08	1221	54	+
$f_{25}(x)$	7.52E−10	1275	0	+	2.09E−08	1218	57	+	7.70E−05	1047	228	+
$f_{26}(x)$	0.225655	512	763	=	6.73E−09	1238	37	+	0.174958	778	497	=
$f_{27}(x)$	1.71E−07	96	1179	−	2.22E−07	1174	101	+	0.01037	372	903	−
$f_{28}(x)$	0.057816	441	834	=	0.046192	431	844	−	9.16E−06	1097	178	+
$f_{29}(x)$	0.005675	924	351	+	2.22E−09	18	1257	−	0.252604	756	519	=
$f_{30}(x)$	7.52E−10	1275	0	+	7.99E−10	1274	1	+	7.52E−10	1275	0	+
$f_{31}(x)$	7.52E−10	0	1275	−	7.99E−10	1274	1	+	4.00E−09	1247	28	+
$f_{32}(x)$	7.52E−10	0	1275	−	0.154439	490	785	=	9.03E−10	1272	3	+
$f_{33}(x)$	1.55E−09	1263	12	+	2.19E−06	1128	147	+	7.52E−10	1275	0	+
$f_{34}(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_{35}(x)$	7.52E−10	1275	0	+	1.33E−08	49	1226	−	7.52E−10	1275	0	+
$f_{36}(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+	8.50E−10	1273	2	+
$f_{37}(x)$	8.50E−10	1273	2	+	2.81E−09	1253	22	+	0.000268	1015	260	+

$$\lg t = \lg \frac{1}{u} + \lg \frac{r^2 \mu^*}{4T} \tag{34}$$

$$W\left(u, \frac{r}{B}\right) = \begin{cases} W\left(0, \frac{r}{B}\right) - W\left(\frac{r^2}{4B^2 u}, 0\right) & u < u_{\min} \\ 0.5 \cdot W(0,u) \cdot erfc(\alpha X + \beta X^3) & u_{\min} \le u \le u_{\max} \\ W(u,0) & u > u_{\max} \end{cases} \tag{35}$$

$$\begin{cases} X = \lg \frac{2uB}{r} \\ \alpha = 0.7708 + 0.3457 lg(\frac{r}{B}) + 0.09128 lg^2(\frac{r}{B}) \\ \quad +0.09937 lg^3(\frac{r}{B}) \\ \beta = 0.02796 + 0.01023 lg(\frac{r}{B}) \\ umin = \begin{cases} max(0.06541 v^{0.2763}, 0.02985) & \frac{r}{B} < 0.5 \\ 0.1192 v^{1.142} & \frac{r}{B} \ge 0.5 \end{cases} \\ umax = max(39.93 v^{2.391}, 0.02985) \end{cases} \tag{36}$$

$$W(0,v) = \begin{cases} (1 + 0.2062 v^2) \cdot lg(-2lg(0.5772)/v)^2 \\ \quad +0.5579 v^2 & \mathrm{v} < 1 \\ 1/(v \cdot \exp(v)) \cdot \sqrt{2\pi / v} \cdot (v - 0.09173) & \mathrm{v} \ge 1 \end{cases} \tag{37}$$

$$W(u,0) = \begin{cases} -\ln u + \sum_{i=0}^{5} a_i u^i & u \le 1 \\ \frac{1}{ue^u} \frac{\sum_{i=0}^{4} b_i u^i}{\sum_{i=0}^{4} c_i u^i} & u \ge 1 \end{cases} \tag{38}$$

where,
s: drawdown in an observation well,
T: transmissivity of the aquifer,
r: distance from discharging well to observation well,
μ^*: storage coefficient of the aquifer,
t: time from the beginning of discharge to observation,
Q: discharge rate.

The rounded-off values are given by $a_0 = -0.57722$, $a_1 = 0.99999$, $a_2 = -0.24991$, $a_3 = 0.05519$, $a_4 = -0.00976$, $a_5 = 0.00108$, $b_0 = 0.26777$, $b_1 = 8.63476$, $b_2 = 18.05902$, $b_3 = 8.57333$, $b_4 = 1$, $c_0 = 3.95850$, $c_1 = 21.09965$, $c_2 = 25.63296$, $c_3 = 9.57332$ and $c_4 = 1$.

If the discharge is kept constant, the second terms on the right of Eqs. (33) and (34) are both constant, and the relationship between s and t is the same as the relationship between $W(u, r/B)$ and $1/u$. If the values of $W(u, r/B)$ and $1/u$ are shown on a log–log paper, then a type curve for the relationship between s and t is produced. The values of s versus t can be shown on a transparent log–log paper using the same scale as the type curve, and its curve may be similar

Table 10
Statistical comparisons of single-problem-based WSRT for ASO vs. GSA and GA.

Function	ASO vs. GSA				ASO vs. GA			
	p-value	T−	T+	Winner	p-value	T−	T+	Winner
$f_1(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_2(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_3(x)$	3.84E−08	1207	68	+	1.26E−08	1227	48	+
$f_4(x)$	5.27E−01	703	572	=	7.52E−10	1275	0	+
$f_5(x)$	4.76E−09	1244	31	+	1.01E−07	1189	86	+
$f_6(x)$	1	0	0	=	1.00E+00	0	0	=
$f_7(x)$	7.52E−10	0	1275	−	1.24E−04	240	1035	−
$F_8(x)$	7.52E−10	1275	0	+	7.52E−10	0	1275	−
$F_9(x)$	7.52E−10	0	1275	−	7.52E−10	0	1275	−
$f_{10}(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_{11}(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_{12}(x)$	7.51E−10	1275	0	+	7.52E−10	1275	0	+
$f_{13}(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_{14}(x)$	8.18E−08	1193	82	+	6.60E−08	1197	78	+
$f_{15}(x)$	2.50E−09	1255	20	+	0.233136	514	761	=
$f_{16}(x)$	9.58E−06	350	925	−	0.484375	18	10	=
$f_{17}(x)$	1.00E+00	0	0	=	0.000122	946	329	+
$f_{18}(x)$	4.79E−05	983.5	291.5	+	7.08E−02	734.5	540.5	=
$f_{19}(x)$	3.41E−07	351	924	−	7.52E−10	1275	0	+
$f_{20}(x)$	5.00E−01	3	0	=	7.52E−10	1275	0	+
$f_{21}(x)$	0.141593	334	569	=	5.51E−03	350	925	−
$f_{22}(x)$	6.25E−01	6	9	=	5.62E−08	1200	75	+
$f_{23}(x)$	2.65E−02	312	963	−	7.50E−07	1150	125	+
$f_{24}(x)$	7.52E−10	1275	0	+	1.30E−09	1266	9	+
$f_{25}(x)$	2.02E−02	397	878	−	2.90E−01	747	528	=
$f_{26}(x)$	7.52E−10	1275	0	+	1.72E−01	496	779	=
$f_{27}(x)$	7.52E−10	1275	0	+	2.14E−04	1021	254	+
$f_{28}(x)$	4.24E−09	1246	29	+	0.003954	936	339	+
$f_{29}(x)$	5.04E−09	1243	32	+	8.84E−02	814	461	=
$f_{30}(x)$	1.06E−08	45	1230	−	5.99E−04	282	993	−
$f_{31}(x)$	7.52E−10	0	1275	−	7.52E−10	1275	0	+
$f_{32}(x)$	9.03E−10	3	1272	−	1.64E−06	1134	141	+
$f_{33}(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_{34}(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_{35}(x)$	7.52E−10	1275	0	+	7.52E−10	1275	0	+
$f_{36}(x)$	2.00E−07	1176	99	+	7.52E−10	1275	0	+
$f_{37}(x)$	4.49E−09	1245	30	+	7.52E−10	1275	0	+

Table 11
Statistical results of single-problem-based WSRT of ASO.

Function type	ASO vs. WDO (+/= /−)	ASO vs. SA (+/= /−)	ASO vs. PSO (+/= /−)	ASO vs. GSA (+/= /−)	ASO vs. GA (+/= /−)
Unimodal	1/1/5	6/0/1	6/1/0	4/2/1	5/1/1
Multimodal	4/0/2	5/0/1	4/0/2	5/0/1	4/0/2
Low-dimension	9/1/0	3/5/2	2/7/1	3/4/3	6/3/1
Hybrid	2/3/1	3/1/2	3/2/1	5/0/1	3/3/0
Composition	6/0/2	6/1/1	8/0/0	5/0/3	7/0/1
Total	22/5/10	23/7/7	23/10/4	22/6/9	25/7/5

to the type curve, but would be displaced by the term $Q/(4\pi T)$ on the s and $W(u, r/B)$ axes and by the term $r^2\mu^*/(4T)$ on the t and $1/u$ axes. The plot of s versus t could be carefully slid on the type curve, keeping the axes parallel, until it matches a segment of the type curve.

When the match is finished, the match point is any intersecting line set on the overlay curve. Choosing different match points will produce the similar results. A match point is frequently chosen at $W(u, r/B) = 1$ and $1/u = 10$ in $W(u, r/B)$−$1/u$ coordinate systems, and the values of s and t are read in s-t coordinate systems. Fig. 19 shows the curve fitting procedure, the blue plot is the $W(u, r/B)$−$1/u$ coordinate including the type curve and the red is the s-t coordinate. The curve fitting procedure includes two aspects: one is the motion of the s-t coordinate relative to the $W(u, r/B)$−$1/u$ coordinate horizontally and vertically, and the other is the change of the Hantush function curve. After the match point is determined, the transmissivity and storage coefficients can be calculated as

$$T = \frac{Q}{4\pi s} W\left(u, \frac{r}{B}\right) \tag{39}$$

$$\mu^* = \frac{4Ttu}{r^2} \tag{40}$$

In short, such a problem indicates how to find a suit of appropriate parameters to achieve the best match between $W(u, r/B)$-$1/u$ coordinate systems and s-t coordinate systems, such that the objective function of the problem can be expressed as

$$f(a, b, \beta) = \sum_{i=1}^{n} (\log_{10}(s_{obs,i} \cdot 10^b) - \log_{10} W(\frac{1}{t_i \cdot 10^a}, \beta))^2 \tag{41}$$

for

$$\begin{cases} -1 - \log_{10}(t_1) \le a \le 4 - \log_{10}(t_n) \\ -2 - \log_{10}(s_1) \le b \le 1 - \log_{10}(s_n) \\ 0 \le \beta \le 10 \end{cases} \tag{42}$$

where, $t_i (i = 1, \ldots, n)$ is the ith time from the beginning of discharge to observation, n is the size of the drawdown data, $s_{obs,i}$ is the ith drawdown in an observation well, a and b respectively represent the displacement of the observation data with respect to

Table 12
Statistical comparisons of multi-problem-based WSRT of ASO.

Comparison	p-value	T−	T+	Winner
ASO vs. WDO	1.03E−03	542	124	–
ASO vs. SA	1.60E−03	409	87	–
ASO vs. PSO	1.06E−06	525	3	–
ASO vs. GSA	1.65E−03	363	72	–
ASO vs. GA	1.47E−06	550	11	–

the Hantush function curve in a log–log coordinate both horizontally and vertically, and β represents the Hantush function which is equal to r/B.

The pumping test dataset taken from [104] is shown in Table 13. The distance r between the observation well and pumping well is 29.0 m, the water inflow of the pumping well Q is 69.1 m^3/h and the total pumping time is 540 min.

All the algorithms described above are used to solve this problem, whose parameters are the same as those in the previous section. The size of the population and the maximum number of iterations are set to 30 and 300, respectively. Every algorithm runs 50 times and the average performance among 50 runs is summarized. Fig. 20 shows the convergence comparisons of algorithms for the objective value of the hydrogeologic parameter identification. It is clear that ASO has a better performance compared to the other algorithms in terms of the convergence rate. The performance comparisons of algorithms for the hydrogeologic parameter identification are illustrated in Table 14. From Table 14, it is evident that ASO significantly outperforms the other algorithms in terms of the average of the best-so-far solution, the standard deviation of the best-so-far solution, and the minimum of the best-so-far solution. Although SA and PSO maintain some levels of competitiveness, the superiority of the standard deviation and the convergence performance of ASO shows that it performs a more stable optimization and has more accurate results than them. Accordingly, Fig. 20 and Table 14 show the prominent advantages. A certain repulsion among atoms contributes effectively to the exploration in the entire search space in the initial iteration. With the lapse of iterations, the exploration fades out and a growing attraction contributes significantly to the exploitation in promising local regions. ASO initially focuses on a global search to avoid the local optima and then focuses on a local search to improve the accuracy of the final solutions, thus greatly enhancing a smooth transition from the explorative search to the exploitative search.

Fig. 21 shows the fitting graphs between the Hantush function curves and the observation data corresponding to the average of the best-so-far solution using the algorithms, and Fig. 22 also shows the fitting graphs in [104]. Meanwhile, the Hantush function curves corresponding to the third variable obtained using all the algorithms and the traditional fitting method are described in Fig. 23. It can be seen that the fitting effects of ASO, SA and PSO are obviously better than those of the others and the fitting effect of the traditional fitting method is the worst. The visual effects of these fitting curves are identical with the above results. However, it is difficult to visually identify the fitting differences among ASO, SA and PSO owing to the similar precision. But as shown in Table 15 and Fig. 20, ASO is more competitive than SA and PSO. These figures and tables reveal that the time-drawdown obtained from ASO is in excellent agreement with the observation data, as compared to those obtained from its competitors. These more accurate hydrogeological parameters are essential in the context of proper estimation and management of groundwater resources.

6. Conclusions

In this study, a novel physics-inspired metaheuristic algorithm, namely atom search optimization (ASO) algorithm, has been developed for global optimization problems. ASO is inspired by the basic molecular dynamics to mathematically establish the atomic motion model, which is based on the interaction and constraint forces. In ASO, each atom is affected by the attractive force or repulsive force from its neighbors and the constraint force from the atom with the best fitness value. The atomic motion follows Newton's second law. The attractive force encourages atoms to explore the entire search space extensively, and the repulsive force enables them to exploit the promising regions intensively.

To test the effectiveness of ASO, five comparable algorithms are used on diverse benchmark sets including unimodal, multimodal, low-dimensional, hybrid, and composition functions at the same time for comparison. The results are verified regarding qualitative results, convergence preference and overall preference. Furthermore, ASO is also applied to a hydrogeologic parameter estimation problem, which demonstrates its availability and effectiveness in solving real-world problems. Based on the above results, analysis and discussion of the experiments and the application, the following conclusions can be drawn.

ASO performs a better self-adaptive convergence based on different types of benchmark functions. The algorithm can successfully find the local regions around the promising solution.

ASO performs well for unimodal functions and shows a strong competitiveness for multimodal functions.

ASO can adeptly balance the explorative and exploitative search in dealing with hybrid and composition functions.

The simplicity of ASO and its few control parameters make it easy to implement.

The experiment results of the benchmark functions and the application to a hydrogeologic parameter estimation problem show its potential in solving other real-world problems such as vehicle path planning, job shop scheduling, and structural optimization.

Table 13
Pumping test data.

Time (min)	Drawdown (m)	Time (min)	Drawdown (m)	Time (min)	Drawdown (m)
1	0.05	75	0.62	360	0.772
4	0.054	90	0.64	390	0.785
7	0.12	120	0.685	420	0.79
10	0.175	150	0.725	450	0.792
15	0.26	180	0.735	480	0.794
20	0.33	210	0.755	510	0.795
25	0.383	240	0.76	540	0.796
30	0.425	270	0.76	570	0.797
45	0.52	300	0.763	600	0.798
60	0.575	330	0.77	660	0.80

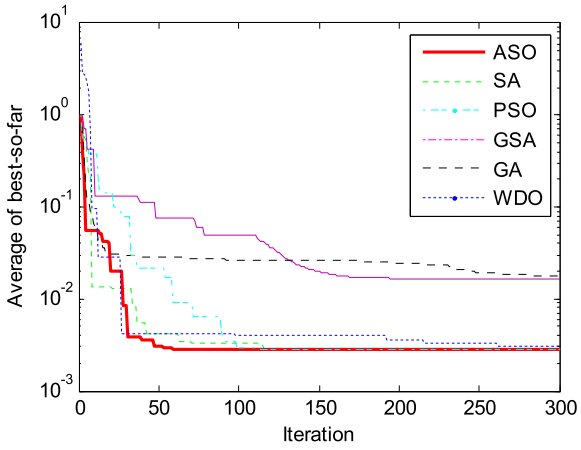

Fig. 20. Convergence comparisons of algorithms for hydrogeologic parameter identification.

Table 14
Performance comparisons of algorithms for hydrogeologic parameter identification.

Index	SA	PSO	GSA	GA	WDO	ASO
Mean	2.7769E−03	2.7770E−03	1.6689E−02	1.6691E−02	3.0660E−03	2.7768E−03
Std	5.5879E−08	5.3861E−05	1.3479E−02	1.9422E−02	1.4192E−05	2.5748E−18
Best	2.7768E−03	2.7768E−03	6.1622E−03	5.9201E−03	2.7768E−03	2.7768E−03

Table 15
Result comparisons of different methods for hydrogeologic parameter identification.

Algorithm	Solution (x, y, β)	Match point $(W, 1/u, s, t)$	Hydrogeologic parameters (T, μ^*, B)	Fitting error (f)
ASO	(−0.7695, 0.4113, 0.4347)	(1, 10, 0.3879, 58.8152)	(340.2486, 1.4324E−04, 453.1502)	2.7768E−03
SA	(−0.7696, 0.4113, 0.4348)	(1, 10, 0.3879, 58.6262)	(340.2676, 1.4327E−04, 453.1293)	2.7769E−03
PSO	(−0.7695, 0.4113, 0.4351)	(1, 10, 0.3879, 58.8153)	(340.2486, 1.4324E−04, 452.7695)	2.7770E−03
GSA	(−0.7946, 0.3308, 0.5477)	(1, 10, 0.4669, 62.3120)	(282.6632, 1.2607E−04, 359.6588)	1.6689E−02
GA	(−0.7974, 0.3272, 0.5502)	(1, 10, 0.4708, 62.7193)	(280.3185, 1.2584E−04, 358.0227)	1.6691E−02
WDO	(−0.7703, 0.4193, 0.4255)	(1, 10, 0.3808, 58.9245)	(346.5939, 1.4618E−04, 453.1673)	3.0660E−03
[104]	(−0.6946, 0.4971, 0.3500)	(2.45, 101, 0.78, 500)	(414.7, 1.469E−04, 562.86)	3.4900E−02

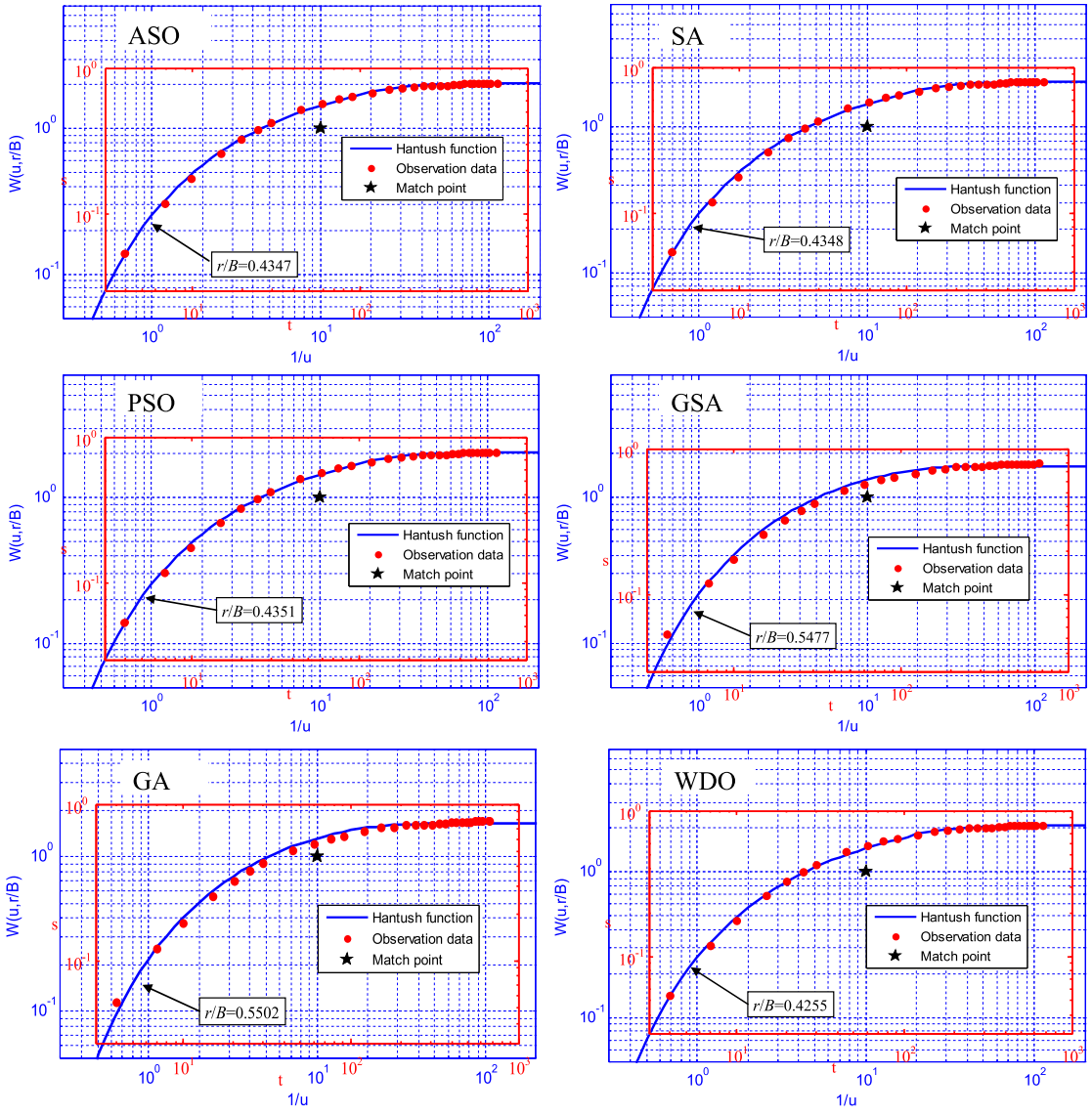

Fig. 21. Fitting graphs between Hantush function curves and observation data using algorithms.

For future work, the binary version of ASO can be presented to deal with discrete problems, and its multi-objective version can also be proposed to solve multi-objective problems. ASO proposed here can be applied to other real-world problems in hydrology involving the calibration of surface runoff simulation, parameter estimation of a watershed model, and dispersion coefficients estimation.

Molecular dynamics is a complicated subject, if possible, other physical characteristics of atoms can be discovered and unitized to enrich ASO and make it become much more competitive than

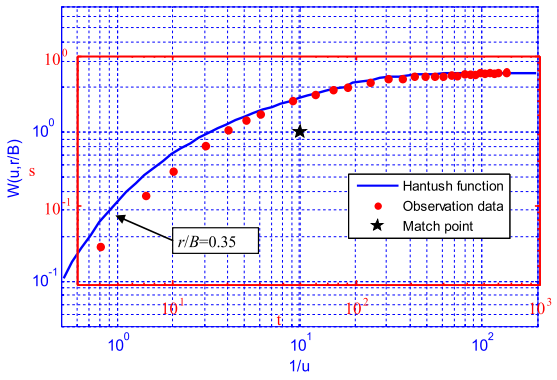

Fig. 22. Fitting graphs between Hantush function curves and observation data using traditional fitting method [104].

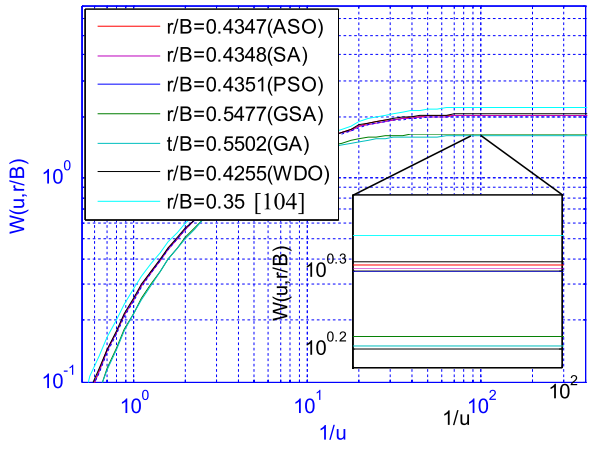

Fig. 23. Hantush function curves corresponding to the third variable obtained from all methods.

ever. Many aspects are still left unknown about ASO, and there are many interesting avenues for future research, for example, the leapfrog-type method can be used to develop the mathematical representation of atomic motion.

Acknowledgments

The authors would like to thank Lisa Sheppard of University of Illinois at Urbana–Champaign for editorial review. This work is supported in part by Natural Science Foundation of Hebei Province of China (E2018402092, F2017402142), High-Level Talent Funding Project of Hebei Province of China (B2017003014), Engineering Technology Research Centre of Hebei Province of China for High Efficient Utilization of Water Resources (18965307H), and Scientific Research Key Project of University of Hebei Province of China (ZD2017017).

References

[1] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (1983) 671–680.
[2] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the 1995 IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.
[3] J.Q. Li, H.Y. Sang, Y.Y. Han, C.G. Wang, K.Z. Gao, Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions, J. Cleaner Prod. 181 (2018) 584–598.
[4] Y. Liu, D. Gong, J. Sun, Y. Jin, A many-objective evolutionary algorithm using a one-by-one selection strategy, IEEE Trans. Cybern. 47 (9) (2017) 2689–2702.
[5] P.Y. Duan, J.Q. Li, Y. Wang, H.Y. Sang, B.X. Jia, Solving chiller loading optimization problems using an improved teaching-learning-based optimization algorithm, Optim. Control Appl. Methods 39 (1) (2018) 65–77.
[6] W. Hare, J. Nutini, S. Tesfamariam, A survey of non-gradient optimization methods in structural engineering, Adv. Eng. Softw. 59 (2013) 19–28.
[7] H. Mühlenbein, M. Gorges-Schleuter, O. Krämer, Evolution algorithms in combinatorial optimization, Parallel Comput. 7 (1) (1988) 65–85.
[8] D. Gong, J. Sun, X. Ji, Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems, Inf. Sci. 233 (2013) 141–161.
[9] D. Gong, X. Ji, J. Sun, X. Sun, Interactive evolutionary algorithms with decision-maker's preferences for solving interval multi-objective optimization problems, Neurocomputing 137 (2014) 241–251.
[10] Z.W. Geem, J. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Trans. Simul. 76 (2) (2001) 60–68.
[11] J. Krause, J. Cordeiro, R.S. Parpinelli, H.S. Lopes, A survey of swarm algorithms applied to discrete optimization problems, in: Swarm Intelligence and Bio-inspired Computation: Theory and Applications, Elsevier Sci. Technol. Books, 2013, pp. 169–191.
[12] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. 22 (3) (2002) 52–67.
[13] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, E. Tarantino, Biological invasion-inspired migration in distributed evolutionary algorithms, Inf. Sci. 207 (2012) 50–65.
[14] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, USA, 1975.
[15] D. Gong, J. Sun, Z. Miao, A set-based genetic algorithm for interval many-objective optimization problems, IEEE Trans. Evol. Comput. 22 (1) (2018) 47–60.
[16] H.G. Beyer, H.P. Schwefel, Evolution strategies-a comprehensive introduction, Nat. Comput. 1 (1) (2002) 3–52.
[17] P. Rocca, G. Oliveri, A. Massa, Differential evolution as applied to electromagnetics, IEEE Antennas Propag. Mag. 53 (1) (2011) 38–49.
[18] K.A. Juste, H. Kita, E. Tanaka, J. Hasegawa, An evolutionary programming solution to the unit commitment problem, IEEE Trans. Power Syst. 14 (4) (1999) 1452–1459.
[19] P. Moscato, A. Mendes, R. Berretta, Benchmarking a memetic algorithm for ordering microarray data, Biosystems 88 (1) (2007) 56–75.
[20] X.S. Yang, A. Hossein Gandomi, Bat algorithm: a novel approach for global engineering optimization, Eng. Comput. 29 (5) (2012) 464–483.
[21] WT. Pan, A new fruit fly optimization algorithm: taking the financial distress model as an example, Knowl.-Based Syst. 26 (2012) 69–74.
[22] Z. Meng, J.S. Pan, Monkey King Evolution: a new memetic evolutionary algorithm and its application in vehicle fuel consumption optimization, Knowl.-Based Syst. 97 (2016) 144–157.
[23] S.A. Uymaz, G. Tezel, E. Yel, Artificial algae algorithm (AAA) for nonlinear global optimization, Appl. Soft Comput. 31 (2015) 153–171.
[24] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (6) (2009) 702–713.
[25] V. Punnathanam, P. Kotecha, Yin-Yang-pair Optimization: a novel lightweight optimization algorithm, Eng. Appl. Artif. Intell. 54 (2016) 62–79.
[26] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecol. Inf. 1 (4) (2006) 355–366.
[27] A.O. Topal, O. Altun, A novel meta-heuristic algorithm: Dynamic Virtual Bats Algorithm, Inf. Sci. 354 (2016) 222–235.
[28] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (13) (2009) 2232–2248.
[29] Ş.I. Birbil, S.C. Fang, An electromagnetism-like mechanism for global optimization, J. Global Optim. 25 (3) (2003) 263–282.
[30] W.F. Sacco, C.R.E. De Oliveira, A new stochastic optimization algorithm based on a particle collision metaheuristic, in: Proceedings of 6th WCSMO, 2005.
[31] B. Doğan, T. Ölmez, A new metaheuristic for numerical function optimization: Vortex Search algorithm, Inf. Sci. 293 (2015) 125–145.
[32] A. Kaveh, T. Bakhshpoori, Water evaporation optimization: a novel physically inspired optimization algorithm, Comput. Struct. 167 (2016) 69–85.
[33] Y.T. Hsiao, C.L. Chuang, J.A. Jiang, C.C. Chien, A novel optimization algorithm: space gravitational optimization, in: 2005 IEEE International Conference on Systems, Man and Cybernetics, Vol. 3, IEEE, 2005, pp. 2323–2328.
[34] H.M. Genç, I. Eksin, O.K. Erol, Big Bang-Big Crunch optimization algorithm hybridized with local directional moves and application to target motion analysis problem, in: 2010 IEEE International Conference on Systems Man and Cybernetics, SMC, IEEE, 2010, pp. 881–887.
[35] H. Shah-Hosseini, Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, Int. J. Comput. Sci. Eng. 6 (1–2) (2011) 132–140.
[36] M. Kripka, R.M.L. Kripka, Big crunch optimization method, in: International Conference on Engineering Optimization, Brazil, 2008, pp. 1–5.
[37] C.L. Chuang, J.A. Jiang, Integrated radiation optimization: inspired by the gravitational radiation in the curvature of space–time, in: IEEE Congress on Evolutionary Computation, CEC 2007, IEEE, 2007, pp. 3157–3164.

[38] H. Shah-Hosseini, The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm, Int. J. Bio-Inspired Comput. 1 (1–2) (2009) 71–79.
[39] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (3) (2010) 267–289.
[40] S. Mirjalili, S.Z.M. Hashim, BMOA: binary magnetic optimization algorithm, Int. J. Mach. Learn. Comput. 2 (3) (2012) 204.
[41] M. Zheng, G. Liu, C. Zhou, Y. Liang, Y. Wang, Gravitation field algorithm and its application in gene cluster, Algorithms Mol. Biol. 5 (1) (2010) 32.
[42] B. Javidy, A. Hatamlou, S. Mirjalili, Ions motion algorithm for solving optimization problems, Appl. Soft Comput. 32 (2015) 72–79.
[43] Y.J. Zheng, Water wave optimization: a new nature-inspired metaheuristic, Comput. Oper. Res. 55 (2015) 1–11.
[44] J.J. Flores, R. López, J. Barrera, Gravitational interactions optimization, in: International Conference on Learning and Intelligent Optimization, Springer, Berlin, Heidelberg, 2011, pp. 226–237.
[45] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems, Inf. Sci. 183 (1) (2012) 1–15.
[46] G. Zarand, F. Pazmandi, K.F. Pál, G.T. Zimányi, Using hysteresis for optimization, Phys. Rev. Lett. 89 (15) (2002) 150–201.
[47] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84.
[48] J. Shen, Y. Li, Light ray optimization and its parameter analysis, in: International Joint Conference on Computational Sciences and Optimization, Vol. 2, CSO 2009, IEEE, 2009, pp. 918–922.
[49] V.K. Patel, V.J. Savsani, Heat transfer search (HTS): a novel optimization algorithm, Inf. Sci. 324 (2015) 217–246.
[50] K. Tamura, K. Yasuda, Primary study of spiral dynamics inspired optimization, IEEE Trans. Electr. Electron. Eng. 6 (S1) (2011) S98–S100.
[51] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm-a novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110 (2012) 151–166.
[52] F.F. Moghaddam, R.F. Moghaddam, M. Cheriet, Curved space optimization: a random search based on general relativity theory, 2012. arXiv preprint arXiv:1208-2214.
[53] G. Beni, J. Wang, Swarm intelligence in cellular robotic systems, in: Robots and Biological Systems: Towards a New Bionics?, Springer, Berlin, Heidelberg, 1993, pp. 703–712.
[54] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. Part B 26 (1) (1996) 29–41.
[55] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Inf. Sci. 192 (2012) 120–142.
[56] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.
[57] A.H. Gandomi, A.H. Alavi, Krill herd: a new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul. 17 (12) (2012) 4831–4845.
[58] M.S. Kiran, TSA: Tree-seed algorithm for continuous optimization, Expert Syst. Appl. 42 (19) (2015) 6686–6698.
[59] E. Cuevas, M. Cienfuegos, D. Zaldívar, M. Pérez-Cisneros, A swarm optimization algorithm inspired in the behavior of the social-spider, Expert Syst. Appl. 40 (16) (2013) 6374–6384.
[60] A. Askarzadeh, Bird mating optimizer: an optimization algorithm inspired by bird mating strategies, Commun. Nonlinear Sci. Numer. Simul. 19 (4) (2014) 1213–1228.
[61] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, nature & biologically inspired computing, in: NaBIC 2009, World Congress on IEEE, 2009, pp. 210–214.
[62] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[63] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[64] A.A.A. Mohamed, Y.S. Mohamed, A.A.M. El-Gaafary, A.M. Hemeida, Optimal power flow using moth swarm algorithm, Electr. Power Syst. Res. 142 (2017) 190–206.
[65] A. Kaveh, N. Farhoudi, A new optimization method: dolphin echolocation, Adv. Eng. Softw. 59 (2013) 53–70.
[66] R. Oftadeh, M.J. Mahjoob, M. Shariatpanahi, A novel meta-heuristic optimization algorithm inspired by group hunting of animals: hunting search, Comput. Math. Appl. 60 (7) (2010) 2087–2098.
[67] E. Duman, M. Uysal, A.F. Alkaya, Migrating Birds Optimization: A new meta-heuristic approach and its performance on quadratic assignment problem, Inf. Sci. 217 (2012) 65–77.
[68] X.S. Yang, Firefly algorithm, stochastic test functions and design optimization, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84.
[69] A. Mucherino, O. Seref, Monkey search: a novel metaheuristic search for global optimization, in: AIP conference proceedings, Vol. 935, AIP, 2007, pp. 162–173, (1).
[70] M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, Swarm Evol. Comput. (2018) http://dx.doi.org/10.1016/j.swevo.2018.02.013.
[71] E. Alba, B. Dorronsoro, The exploration/exploitation tradeoff in dynamic cellular genetic algorithms, IEEE Trans. Evol. Comput. 9 (2) (2005) 126–142.
[72] N. Lynn, P.N. Suganthan, Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation, Swarm Evol. Comput. 24 (2015) 11–24.
[73] D. Bertsimas, P. Jaillet, S. Martin, Online Vehicle routing: The edge of optimization in large-scale applications, Oper. Res. (2018).
[74] S.P. Zhang, X.K. Xin, Pollutant source identification model for water pollution incidents in small straight rivers based on genetic algorithm, Appl. Water Sci. 7 (4) (2017) 1955–1963.
[75] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[76] G.F. Barker, Divisions of matter, in: A Text-Book of Elementary Chemistry: Theoretical and Inorganic, John F Morton & Co., 1870.
[77] S.M. Walker, A. King, What is Matter?, Lerner Publications, Minneapolis U.S.A., 2005.
[78] J. Kenkel, P.B. Kelter, D.S. Hage, Chemistry: An Industry-Based Introduction With CD-ROM, CRC Press, 2000.
[79] B.J. Alder, T.E. Wainwright, Studies in molecular dynamics, I. general method, J. Chem. Phys. 31 (2) (1959) 459–466.
[80] A. Rahman, Correlations in the motion of atoms in liquid argon, Phys. Rev. 136 (2A) (1964) A405–A411.
[81] H. Goldstein, C.P. Poole, J.L. Safko, Classical Mechanics, Addison-Wesley, 2001.
[82] D.C. Rapaport, R.L. Blumberg, S.R. McKay, W. Christian, The art of molecular dynamics simulation, Comput. Phys. 10 (5) (1996) 456-456.
[83] G.C. Maitland, M. Rigby, E.B. Smith, W.A. Wakeham, Intermolecular Forces: Their Origin and Determination, Clarendon Press, Oxford, 1981.
[84] C.G. Gray, K.E. Gubbins, Theory of Molecular Fluids, Volume 1: Fundamentals, Clarendon Press, Oxford, 1984.
[85] A.J. Stone, The Theory of Intermolecular Forces, Clarendon Press, Oxford, 1996.
[86] J.E. Lennard-Jones, On the determination of molecular fields, Proc. R. Soc. Lond. A: Math. Phys. Eng. Sci. R. Soc. 106 (738) (1924) 463–477.
[87] J.P. Ryckaert, G. Ciccotti, H.J.C. Berendsen, Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes, J. Comput. Phys. 23 (3) (1977) 327–341.
[88] M.E. Tuckerman, B.J. Berne, G.J. Martyna, M.L. Klein, Efficient molecular dynamics and hybrid Monte Carlo algorithms for path integrals, J. Chem. Phys. 99 (4) (1993) 2796–2808.
[89] H.N. Pishkenari, E. Mohagheghian, A. Rasouli, Molecular dynamics study of the thermal expansion coefficient of silicon, Phys. Lett. A 380 (48) (2016) 4039–4043.
[90] F. Hasheminasab, N. Mehdipour, Molecular dynamics simulation of fluid sodium, Fluid Phase Equilib. 427 (2016) 161–165.
[91] D.A. Kilymis, J.M. Delaye, S. Ispas, Density effects on the structure of irradiated sodium borosilicate glass: a molecular dynamics study, J. Non-cryst. Solids 432 (2016) 354–360.
[92] W. Zhao, L. Wang, An effective bacterial foraging optimizer for global optimization, Inf. Sci. 329 (2016) 719–735.
[93] L. Wang, W. Zhao, Y. Tian, G. Pan, A bare bones bacterial foraging optimization algorithm, Cognit. Syst. Res. 52 (2018) 301–311.
[94] J.J. Liang, B.Y. Qu, P.N. Suganthan, Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, 2013.
[95] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[96] Z. Bayraktar, M. Komurcu, J.A. Bossard, D.H. Werner, The wind driven optimization technique and its application in electromagnetics, IEEE Trans. Antennas Propag. 61 (5) (2013) 2745–2757.
[97] E. Cuevas, M. González, An optimization algorithm for multimodal functions inspired by collective animal behavior, Soft Comput. 17 (3) (2013) 489–502.
[98] P. Civicioglu, Backtracking search optimization algorithm for numerical optimization problems, Appl. Math. Comput. 219 (15) (2013) 8121–8144.
[99] N. Zhang, C. Li, R. Li, X. Lai, Y. Zhang, A mixed-strategy based gravitational search algorithm for parameter identification of hydraulic turbine governing system, Knowl.-Based Syst. 109 (2016) 218–237.
[100] R. Srivastava, A. Guzman-Guzman, Practical approximations of the well function, Groundwater 36 (5) (1998) 844–848.
[101] M.S. Hantush, C.E. Jacob, Non-steady radial flow in an infinite leaky aquifer, Trans. Amer. Geophys. Union 36 (1) (1955) 95–100.
[102] M.P. Samuel, M.K. Jha, Estimation of aquifer parameters from pumping test data by genetic algorithm optimization technique, J. Irrig. Drain. Div. 129 (5) (2003) 348–359.
[103] H.D. Yeh, Y.C. Lin, Y.C. Huang, Parameter identification for leaky aquifers using global optimization methods, Hydrol. Process. 21 (7) (2007) 862–872.
[104] J. Hui, C. Bo, P. Hongyu, Groundwater Dynamics, Geological Publishing House, Beijing, China, 2009, pp. 107–115.