



# Optimization for cost-effective design of water distribution networks: a comprehensive learning approach

Indu Bala<sup>1</sup> · Anupam Yadav<sup>2</sup> · Joong Hoon Kim<sup>3</sup>

Received: 10 October 2023 / Revised: 2 February 2024 / Accepted: 14 February 2024 / Published online: 22 March 2024  
© The Author(s) 2024

## Abstract

The Comprehensive Learning Gravitational Search Algorithm (CLGSA) has demonstrated its effectiveness in solving continuous optimization problems. In this research, we extended the CLGSA to tackle NP-hard combinatorial problems and introduced the Discrete Comprehensive Learning Gravitational Search Algorithm (D-CLGSA). The D-CLGSA framework incorporated a refined position and velocity update scheme tailored for discrete problems. To evaluate the algorithm's efficiency, we conducted two sets of experiments. Firstly, we assessed its performance on a diverse range of 24 benchmarks encompassing unimodal, multimodal, composite, and special discrete functions. Secondly, we applied the D-CLGSA to a practical optimization problem involving water distribution network planning and management. The D-CLGSA model was coupled with the hydraulic simulation solver EPANET to identify the optimal design for the water distribution network, aiming for cost-effectiveness. We evaluated the model's performance on six distribution networks, namely Two-loop network, Hanoi network, New-York City network, GoYang network, BakRyun network, and Balerna network. The results of our study were promising, surpassing previous studies in the field. Consequently, the D-CLGSA model holds great potential as an optimizer for economically and reliably planning and managing water networks.

**Keywords** Meta-heuristic algorithm · Comprehensive learning gravitational search algorithm · Binary space · Global optimization benchmarks · Water network system

## 1 Introduction

The landscape of solving optimization problems, especially those encountered in real-world scenarios, has grown increasingly complex and challenging. These problems often exist in high-dimensional spaces where available information might be insufficient for a straightforward mathematical formulation. Traditional methods have frequently struggled to provide effective solutions under these conditions [1]. As a response, a new breed of optimization approaches, known as metaheuristic algorithms, has gained prominence. These

methods utilize heuristic processes to iteratively navigate the problem space, generating high-quality solutions.

Metaheuristic optimization algorithms are typically divided into two main categories: single-point search algorithms and population-based search algorithms. Single-point search algorithms initiate with a set of random solutions, refining them iteratively over a specified number of iterations. In contrast, population-based search algorithms start with multiple random solutions and use an iterative process to converge towards sub-optimal solutions [2]. These approaches involve progressively improving the solutions within the population through nature-inspired probabilistic operators.

A particularly noteworthy population-based metaheuristic algorithm is the Gravitational Search Algorithm (GSA) [3], which has been widely applied across various optimization domains. While the original version of GSA was groundbreaking, it faced challenges like lack of memory, premature convergence, and uncertainty in achieving global optima, along with computational inefficiency [4]. Researchers have thus endeavoured to enhance GSA's capabilities to address

✉ Indu Bala  
indu.bala@adelaide.edu.au; iinduyadav@gmail.com

<sup>1</sup> School of Computer and Mathematical Sciences, The University of Adelaide, Adelaide, SA 5005, Australia

<sup>2</sup> Department of Mathematics and Computing, Dr BR Ambedkar National Institute of Technology, Jalandhar, Jalandhar 144011, India

<sup>3</sup> School of Civil, Environmental and Architectural Engineering, Korea University, Seoul 136-713, South Korea

real-life optimization problems. Enhancements to GSA have included the introduction of a disruption operator to modify gravitational forces [5], the integration of chaotic [6] and crossover operators [7] to improve local search, the incorporation of niching selection operators [8], Kepler operators [9], and niche comprehensive strategy [10] for better exploitation.

In recent years, the field of research has been directed towards adapting GSA for practical optimization challenges. A notable advancement was the incorporation of fuzzy logic into GSA to enhance the accuracy of machine learning models, specifically for the early detection of breast cancer [11]. This improvement significantly increased the precision of predictive models in medical diagnostics. Further development saw the implementation of an adaptive strategy within GSA to refine the velocity equation. This particular enhancement was applied to optimize 2D bi-level thresholding in noisy image segmentation [12], demonstrating GSA's versatility in image processing tasks. In parallel, GSA was improved using aggregating learning for training artificial neural networks, yielding results that surpassed those of traditional back-propagation algorithms [13]. This indicates a substantial step forward in the field of neural network training. Additionally, the algorithm was integrated with machine learning techniques like support vector machines. This combination aimed to balance detection rates while minimizing false alarms and the number of features, addressing key challenges in machine learning [14].

To address the limitations of GSA's memory-less nature, further improvements involved the integration of a memory component. This version of GSA was tested on economic load dispatch problems in micro-grids, focusing on optimizing power generation from multiple sources at minimal costs [15]. Moreover, GSA was hybridized using a Reinforcement Learning-based control approach, incorporating Deep Q-Learning (DQL) for the initialization of weights and biases in Neural Networks [16]. This approach aimed to overcome the instabilities often associated with traditional random initializations in neural networks.

GSA also saw an expansion into multi-objective optimization, incorporating various learning strategies. This version was applied to optimize allocation problems in electric enterprise distribution networks, contributing to more flexible and environmentally friendly power supply strategies [17]. Later, an improved binary version of GSA was employed in an echo state network for enhanced time series forecasting. The performance of this application, tested using the Lorenz and Mackey–Glass benchmark time-series datasets, showed promising results compared to conventional evolutionary methods [18], highlighting GSA's expanding role in diverse problem-solving scenarios.

In conclusion, while GSA has shown effectiveness in various problem-solving contexts, improvements are continually

being made to reduce its inherent limitations. To address these issues, we previously developed the Comprehensive Learning Gravitational Search Algorithm (CLGSA) [19] for optimization problems in continuous search spaces. CLGSA is designed to overcome stagnation in local optima, provide a comprehensive search in continuous search spaces, incorporate external memory for tracking optima, and efficiently locate global optima within a reasonable computational timeframe. The success of CLGSA in real-life continuous optimization problems and our desire to adapt this algorithm for discrete optimization challenges led to the current study. Hence, we introduce a discrete version of CLGSA, named D-CLGSA, specifically designed for computationally expensive optimization problems, such as combinatorial NP-hard problems with discrete parameters. This version includes significant enhancements from the original CLGSA. The study makes the following contributions:

- Develop the Discrete Comprehensive Learning Gravitational Search Algorithm (D-CLGSA) with enhanced position and velocity update mechanisms for effectively solving complex combinatorial optimization challenges.
- Conduct rigorous testing of D-CLGSA against a diverse set of 24 benchmark functions, including unimodal, multimodal, composite, and special discrete types, to comprehensively assess the algorithm's computational efficiency.
- Apply the D-CLGSA to the practical optimization of water distribution networks, aiming to reduce operational costs and demonstrate the algorithm's effectiveness and global applicability across different countries' water network systems.

The proposed D-CLGSA is evaluated in the context of water system optimization, with further details and background provided in the following section.

## 1.1 Literature review of water distribution networks

The field of water distribution network (WDN) design is a classic yet dynamically evolving area within engineering. Traditionally, the primary focus in WDN design has been on selecting appropriate pipe diameters within a given network topology to minimize overall design costs. However, the scope of WDN design has expanded over time. In the early stages, this complex problem was tackled using conventional methods, such as Linear Programming (LP) [20], Linear Programming Gradient (LPG) [21], mixed-discrete nonlinear programming (MDNLP) [22], and nonlinear programming [23, 24], which established the foundation for early WDN optimization.

However, the field experienced a significant shift with the advent and increasing accessibility of stochastic approaches and metaheuristic algorithms. This led to the adoption of various stochastic methodologies, including the Genetic Algorithm (GA) [25], Simulated Annealing (SA) [26], Particle Swarm Optimization (PSO) [27], Hybrid Firefly & Differential Evolution [28], Scattering Search (SS) [29], and Evolutionary Algorithm (EA) [30], each contributing to the cost optimization of WDNs in their unique ways.

In the past few years, the motivation to expand, rehabilitate, or resect water networks has encouraged numerous advancements in WDNs. In 2020, for instance, WDNs in Italy were enhanced to mitigate issues such as pipe bursts or topological changes. This improvement involved the implementation of a multi-scale approach, which partitioned the WDN into district metered areas for better leakage control [31]. Besides an enhanced version of the PSO algorithm, integrated with EPANET hydraulic simulation, was developed to minimize leakages and maintain optimal pressure in pipes, showing superior performance over Evolutionary Algorithm (EA) and Cultural Algorithm (CA) [32].

Further developments include the integration of a binary dragonfly algorithm with EPANET for reducing energy consumption in water networks, demonstrating efficiency and reliability compared to other algorithms [33]. This was extended by a multi-objective model that combined mixed discrete nonlinear programming and PSO to minimize pipeline installation and pumping energy costs [34]. Additionally, the multi-objective Rao algorithm (MORao) was employed for optimizing WDNs using EPANET 2.2 for pressure-driven demand analysis, with its results compared favourably against four other heuristics [35].

Continuing in this vein, a many-objective optimization framework was employed to minimize total design and operational costs, using the EPANET model for pressure constraint analysis and optimizing the Apulian water network using Pareto optimality [36]. To address inconsistent flow and pressure variation issues, hierarchical clustering was used to optimize algorithm parameters, thereby enhancing performance [37]. Recent efforts include implementing a multi-objective Ant Colony Optimization (ACO) algorithm aimed at optimizing reservoir operations and water distribution, along with stormwater network design [38]. A novel hydraulically inspired complex network approach was also used to assess and enhance WDN resilience in case of single-pipe failures, using a method that eschews traditional hydraulic simulations in favour of quantifying failure consequences of pipes based on topological attributes and flow redistribution [39]. Furthermore, a hybrid model integrating Grey Wolf Optimizer and Harris Hawk heuristic algorithms with EPANET was developed to optimize WDN design costs [40], while minimizing installation costs and enhancing water loss monitoring using graph theoretic algorithms

and the NSGA-II multi-objective optimization strategy [41]. Table 1 provides a detailed overview of studies related to Water Distribution Networks.

The field of Water Distribution Network (WDN) remains a critical area of research due to its inherent complexity, combinatorial nature, and nonlinear constraints. These characteristics present ongoing challenges but also opportunities for groundbreaking solutions. This study contributes to this dynamic field by applying the proposed algorithm to WDNs, effectively testing its efficiency in real-life scenarios and demonstrating its compatibility with the inherent features of the algorithm.

This research is structured into two main parts: Initially, the performance of the algorithm was rigorously evaluated using a global benchmark dataset [57]. This was followed by an assessment of the algorithm's effectiveness specifically within the Water Distribution System, utilizing six distinct networks: the Two Loop network, New-York City network, GoYang network, Hanoi network, BakRyun network, and Balerna network [26, 43]. The results from these experiments collectively provide strong evidence supporting the efficacy of the proposed algorithm.

The layout of this paper is organized as follows: Sect. 2 offers a concise overview of CLGSA and introduces the proposed Discrete CLGSA. Section 3 details the experimental setup and shares the results of D-CLGSA across 24 global benchmarks. Section 4 delves into the formulation of the optimization problem in WDS. Section 5 conducts an extensive analysis of WDS performance across the six networks, including detailed simulation results for each. Finally, Sect. 6 concludes the paper and discusses potential avenues for future research.

## 2 Methods and materials

In this section, we will delve into the CLGSA algorithm and elucidate the motivation driving the development of the proposed D-CLGSA. This section provides an in-depth exploration of the proposed algorithm's framework and highlights its key components.

### 2.1 Comprehensive learning gravitational search algorithm (CLGSA)

The concept of CLGSA is inspired by the heuristic algorithm GSA proposed by Rashedi et al. [3]. In CLGSA [19], the particles are termed agents, and the stability of these agents is tabulated from their masses. Due to gravitational force, the objects are attracted to each other, and this force results in the mass movement of all the objects towards the objects with greater mass.

**Table 1** Recent studies solving optimal design problems of WDN

Algorithm	Authors/year of publication	Performance metrics
Genetic algorithm (GA)	Johns et al. [42]	The Adaptive locally constrained genetic algorithm is used to optimize pipe diameter
	Zhang et al. [43]	This approach is based on the non-dominated sorting multi-objective genetic algorithm (NSGA-II) to obtain the optimal schemes of China water networks
	Shao et al. [44]	Three improvements in GA aimed at achieving more efficient and expedited optimization. These enhancements encompass dividing the phase of WNP (Water Network Problem) through preliminary hydraulic analysis, introducing modifications to the crossover mechanism, and incorporating boundary pipe grouping
	Poojitha et al. [45]	A hybrid GA and DE were presented to enhance the exploration stage. It finds an optimal set of diameters
Particle swarm optimization (PSO)	Geem [46]	A hybrid version of PSO and HS is proposed. The approach enhanced the exploration stage and presented good results
	Douglas et al. [27]	PSO applies real and discrete variables for avoiding premature convergence. The approach evaluates an objective function with a penalty factor through EPANET
	Bilal et al. [47]	A hybrid version of the Firefly Algorithm and PSO is used to optimize the diameter of the pipe
	Surco et al. [34]	A multi-objective PSO is implemented for minimizing pumping energy costs and pipeline installation
Harmony search (HS)	Geem et al. [48]	The HS approach integrates a jazz improvisation process to discover enhanced solutions for networks by leveraging EPANET simulations
Simulated annealing (SA)	Costa et al. [30]	Optimize the pipe diameter by incorporating a mutation operator in SA
Cross entropy (CE)	Perelman et al. [49]	The study showcases the application of an adaptive stochastic cross-entropy method on two renowned benchmark examples: single-loading gravitational systems and multiple loadings with pumping and storage
	Shibu et al. [50]	The study employs the theory of fuzzy random variables to model the uncertainty in future water demands. The water demand at each node is represented by a normal distribution with a fuzzy mean, where the standard deviation is 10% (or 20%) of the fuzzy mean. The demand is depicted as a triangular fuzzy number with the random demand as its core
Central force optimization (CFO)	Jabbary et al. [51]	A separate methodology “CFOnet” is created in MATLAB and inferred with the model EPANET to optimize the length of pipes
Decomposition approach (DA)	Lee et al. [52]	The study employed a decomposition approach technique that enables to decompose a real large-scale network. The obtained results were then integrated with an optimization algorithm to achieve an optimal cost design process
Gravitational search algorithm (GSA)	Fallah et al. [53]	Continuous GSA is used and truncates the real value into binary to optimize pipe length
Whale optimization algorithm (WOA)	Riham et al. [54]	The whale bubble strategy was used to search for the best parameters
Ant colony optimization (ACO)	Mehzad et al. [55]	An algorithm having three objectives, named Clustered Non-dominated Archiving ACO, is presented with the objective of minimizing the pipe length
	Bhavya et al. [38]	Multi objective ACO was implemented to optimize the cost of reservoir operations and water network design

**Table 1** (continued)

Algorithm	Authors/year of publication	Performance metrics
Targeted path search approaches	Manolis et al. [56]	The process is driven by two distinct subroutines that systematically and sequentially focus on reducing the diameters of the network pipes. This approach facilitates both exploration and exploitation of the search space, allowing for comprehensive optimization
	Jain et al. [35]	A multi-objective Rao algorithm is developed to optimize the cost of water network using EPANET simulation
	Choi et al. [36]	A many-objective optimization framework was employed to minimize total design and operational costs, using the EPANET model for pressure constraint analysis
	Hajibabaei et al. [39]	A novel hydraulically inspired complex network approach was also used to assess and enhance WDN resilience in case of single-pipe failures
	Pham et al. [40]	A hybrid model integrating Grey Wolf Optimizer and Harris Hawk heuristic algorithms with EPANET was developed to optimize WDN design costs

According to the aspects mentioned above, the state of  $i^{th}$  agent is:

$$X_i = (X_i^1, \dots, X_i^d, \dots, X_i^N), \quad \text{for } i = 1, 2, \dots, N$$

The attractive force experienced by  $i^{th}$  particle towards  $j^{th}$  particle at a given time  $t$  is given as,

$$F_{ijD}^t = G^t \times \frac{m_{pi}^t \times m_{aj}^t}{r_{ij}^t + \epsilon} \times (X_{jd}^t - X_{id}^t); d = 1, 2, \dots, D \quad (1)$$

The value of the gravitational constant  $G^t$  at a given time  $t$  can be computed as follows:

$$G^t = G^{t_0} \times e^{-\alpha \frac{iter}{max\_iter}} \quad (2)$$

where  $\alpha$  is descending coefficient and  $G^t$  is a function of the initial value ( $G^{t_0}$ ),  $iter$  is current iteration, and  $max\_iter$  is the maximum number of predefined iterations. In Eq. (1),  $m_{pi}^t$  is the passive gravitational mass of  $i^{th}$  agent at time  $t$ ,  $m_{aj}^t$  is an active gravitational mass of  $j^{th}$  agent at time  $t$ ,  $\epsilon$  is a small constant,  $r_{ij}^t$  is the Euclidean distance between agents  $i$  and  $j$ ,  $X_{jd}^t$  presents the position of agent  $j$ .

The total force of attraction can be computed as follows:

$$F_{id}^t = \sum_{j=1, j \neq i}^{N=P_s} rand_j F_{ijd}^t \quad (3)$$

where  $rand_j$  is a uniform random number generated in  $[0, 1]$ .  $N = P_s$  is particle count. Also, by Newton's 2nd law of motion, the acceleration of the  $i^{th}$  particle is calculated as:

$$ac_{id}^t = \frac{F_{id}^t}{m_{ii}} \quad (4)$$

where  $m_{ii}$  is the initial mass of  $i^{th}$  particle.

The primary objective of developing the CLGSA mechanism was to improve the search process, with the position of each agent playing a crucial role. Specifically, an agent's mass affects the stability of the best and worst values. To ensure efficient agent selection, the following techniques are employed: first, two agents are randomly selected from the swarm (excluding those with fixed velocity). Then, their fitness is compared, and the element with the best fitness quotient is selected. The first constituent of this element is documented as the first constituent of  $X_b$ . This process is repeated recursively, with the  $i^{th}$  constituent of  $X_b$  being replaced by the  $i^{th}$  constituent of the selected agent. Additionally, the CLGSA mechanism incorporates velocity update equations, and position update equations as:

$$Vec_{id}^{t+1} = w \times Vec_{id}^t + ac_{id}^t \quad (5)$$

$$X_{id}^{t+1} = X_{id}^t + Vec_{id}^{t+1} \quad (6)$$

where  $w$  is inertia weight lies between  $[0, 1]$ . It can be a user control parameter. Masses of agents are updated as:

$$m_i^t = \frac{fit_i^t - worst^t}{best^t - worst^t} \quad (7)$$

where  $fit_i^t$  is the fitness value of  $i^{th}$  agent at  $t$ ,  $best^t$  and  $worst^t$  can be defined as (for minimization problem)

$$best^t = \min_{i \in \{1, 2, \dots, N\}} fit_i^t \quad (8)$$

$$worst^t = \max_{i \in \{1, 2, \dots, N\}} fit_i^t \quad (9)$$

The detailed algorithm of CLGSA is presented in Algorithm 1.

**Algorithm 1.** Pseudo-code of CLGSA.

Comprehensive Learning Gravitational Search Algorithm
<p><i>Initialization</i></p> <p>Generate random elements of size <math>N = ps</math> as <math>(X_1^t, X_2^t, \dots, X_{ps}^t)</math> in search range size <math>[X_{min}, X_{max}]</math>.</p> <p>Evaluate the fitness value <math>(fit_1^t, fit_2^t, \dots, fit_{ps}^t)</math></p> <p><b>for</b> <math>i = 1:ps</math></p> <p style="padding-left: 20px;">Calculate <math>G^t, best^t, worst^t</math> and <math>m_i^t</math> for <math>i = 1, 2, \dots, ps</math></p> <p style="padding-left: 20px;">Calculate total force <math>F_i^t</math></p> <p style="padding-left: 20px;">Calculate acceleration and velocity</p> <p><b>for</b> <math>i = 1:D</math></p> <p style="padding-left: 20px;">Set learning probability <math>P_c</math>,</p> <p style="padding-left: 20px;"><b>if</b> <math>rand() &lt; P_c</math></p> <p style="padding-left: 40px;">Choose agents <math>[X_1] = X(ceil(rand * ps), 1:D);</math> &amp;</p> <p style="padding-left: 40px;"><math>[X_2] = X(ceil(rand * ps), 1:D);</math></p> <p style="padding-left: 40px;">Find fitness value of these agents</p> <p style="padding-left: 40px;"><b>if</b> <math>(fit[X_1] &gt; fit[X_2])</math></p> <p style="padding-left: 60px;"><math>X_b = [X_b; X_1(i)];</math></p> <p style="padding-left: 40px;"><b>else</b></p> <p style="padding-left: 60px;"><math>X_b = [X_b; X_2(i)];</math></p> <p style="padding-left: 40px;"><b>end if</b></p> <p style="padding-left: 20px;"><b>else</b></p> <p style="padding-left: 40px;"><math>X_b(i) = X_{best}(i);</math></p> <p style="padding-left: 20px;"><b>end if</b></p> <p><b>end for</b></p> <p style="padding-left: 20px;">Update <math>G^t, best^t, worst^t</math> and <math>m_i^t</math></p> <p style="padding-left: 20px;">Calculate <math>ac_i^t</math> and <math>Vec_i^t</math></p> <p>Back to step 2.</p> <p><b>end for</b></p>

**2.2 Discrete CLGSA****2.2.1 The motivation of designing D-CLGSA**

The key drivers behind the development of D-CLGSA were the need for fast convergence to global solutions, ease of implementation, and minimal parameter tuning. The discrete optimization approaches used by Tolsen et al. [58] and Sadollah et al. [59] to optimize water networks caught our attention due to their robustness. Building on our previous work, where CLGSA outperformed other heuristics, we were motivated to extend the approach to tackle more challenging NP-hard discrete optimization problems [60].

**2.2.2 Framework of D-CLGSA**

Generally, in discrete algorithms, the data is stored in the form of a binary string, and the search process is performed by dividing the string into two halves. The middle element is taken as a reference, and a search key is used for searching the required element. Based on the value of the search key (greater or less than the search value) the required element is selected. Using binary search-based algorithms, the data can be accessed sequentially, whereas, in continuous search algorithms, the data is processed randomly, which requires more computational time. Binary algorithms are advantageous for applications such as cryptographic techniques where there is a huge flow of data, cell formation,



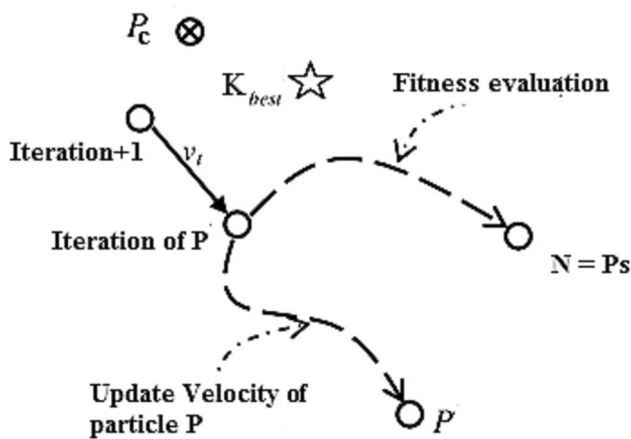


Fig. 1 Dynamics of  $K_{best}$  particle movement

dimension compression with feature selection, unit commitment, opted for binary vectors as their encoding solution. Moreover, the issues encountered in real space can also be addressed in the binary space. One approach is to represent real numbers using binary digits. The binary search space can be conceptualized as a hypercube, allowing agents to navigate between different corners of the hypercube by flipping specific bits. In this section, we introduce a discrete version of the CLGSA algorithm, referred to as D-CLGSA.

In discrete CLGSA, each dimension can take either 0 or 1. The movement of the particle through the dimension refers to the changes in the adjacent variable from 0 to 1 or 1 to 0. In discrete CLGSA, the force, acceleration, and velocity are updated as per the usual CLGSA concept (Acceleration and velocity update is evaluated according to Eqs. 4 and 5). The significant difference between discrete and fundamental CLGSA lies in the position updating procedure. The particle state is updated by switching the states between '0' or '1'. The switching action is done according to the mass velocity and the position is updated so that the present value of the bit is altered with a probability that is evaluated based on the mass velocity. The D-CLGSA upgrades the velocity and contemplates the new state to be either 1 or 0 according to the specified probability.

The proposed D-CLGSA begins with an essential preliminary stage of initialization. During this phase, particles representing potential solutions are randomly generated within the defined search space. This initial population serves as the foundation for the algorithm's search process, where the fitness of each particle is meticulously evaluated to assess its proximity to the optimal solution. The evolutionary process of D-CLGSA is driven by iterative cycles, aiming to reach a predefined total number of generations, denoted as  $N$ . Within each iteration, particles demonstrating higher fitness values exert influence on the velocity of neighbouring particles possessing lower fitness scores. This dynamic is captured in Fig. 1, where the

parameter  $K_{best}$  plays a pivotal role by imparting significant momentum to particles, thereby guiding them towards optimal solutions. The value of  $K_{best}$  is crucial for fostering a balance between exploration of the search space and exploitation of known solutions, is systematically reduced over time. This gradual decrease from the initial population size  $N$  to a singular best solution facilitates enhanced convergence towards the global optimum.

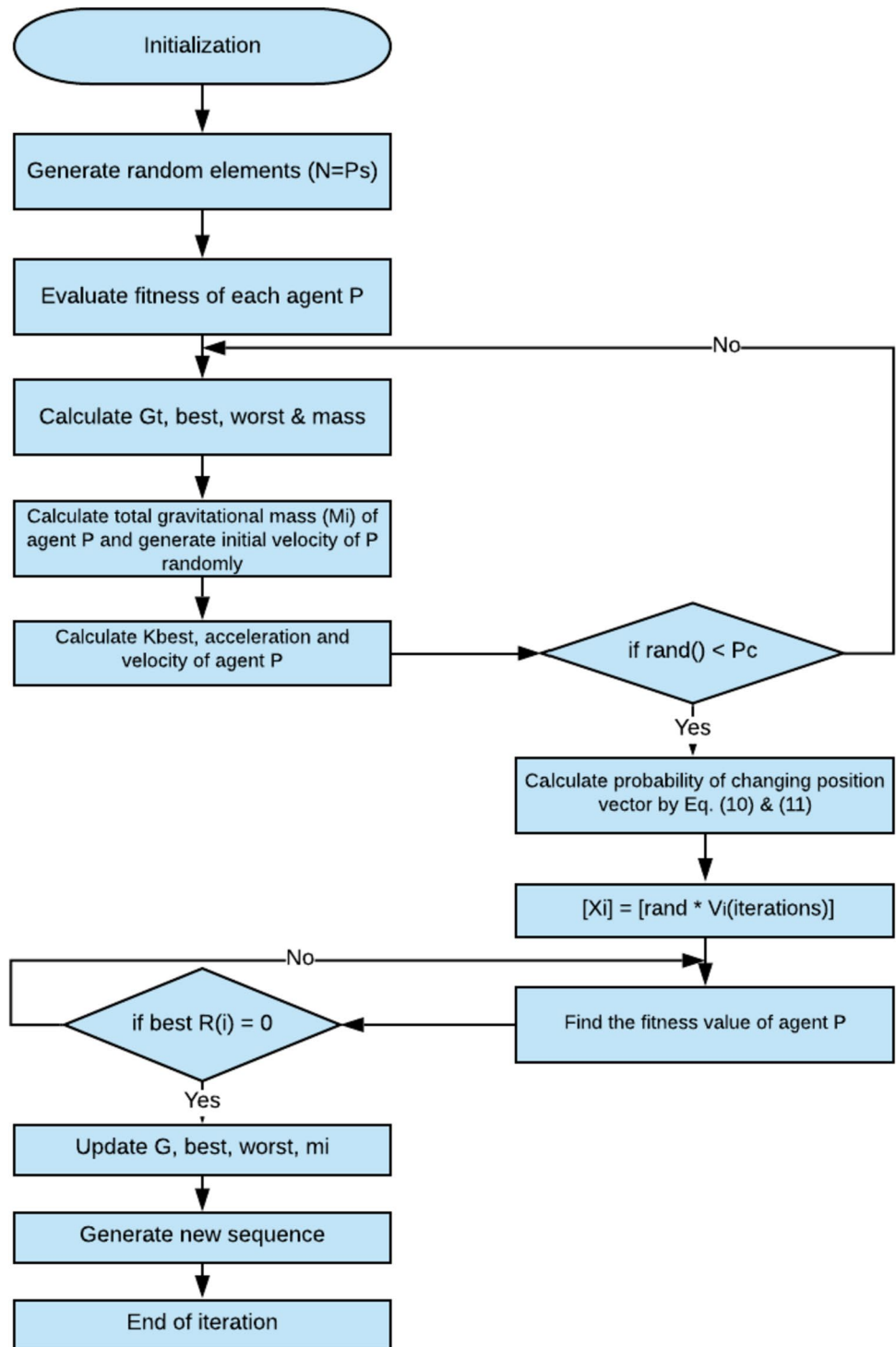
To further refine the search process, D-CLGSA employs the gravitational constant  $G'$ , which decays over time, echoing the algorithm's commitment to balancing exploration with exploitation. Each particle is assigned a fitness value, a measure of its solution quality. High-performing particles are identified and selected, and their gravitational mass is calculated, significantly impacting the algorithm's performance. The movement of each particle, characterized by updates in velocity ( $Vec_i$ ) and acceleration, is meticulously calculated to navigate the particle towards the targeted position within the hypercube search space.

The implementation of D-CLGSA is done following the below-mentioned concepts.

- When a particle exhibits a high velocity, this is interpreted as a strong indication for the need to explore new positions within the search space. Consequently, there is a heightened probability that the particle will switch its state, transitioning from 0 to 1 or vice versa. This mechanism ensures that particles are not stagnant but are actively seeking out new possibilities, especially when far from optimal solutions.
- Conversely, a particle with minimal velocity suggests that its current position is already near an optimal or satisfactory solution. In such cases, the likelihood of altering the particle's state is significantly reduced. This low probability of change means that the algorithm recognizes the particle's current position as being beneficial and sees no immediate need for drastic adjustments.

These concepts form the core of D-CLGSA's strategy for moving through the search space, leveraging velocity as a determinant of a particle's readiness to explore new solutions or maintain its current course. This nuanced approach allows D-CLGSA to balance exploration with exploitation effectively, adapting its search strategy based on the evolving dynamics of the particle swarm.

Depending on the factors mentioned above, a suitable probability function is determined so that for a low value of  $|Vec_i^d|$ , the possibility of switching state of  $X_i^d$  should be near to '0' level and for a high value of  $|Vec_i^d|$ , the probability of switching state of  $X_i^d$  should be very high. The function  $P(Vec_i^d)$  is defined to transfer  $Vec_i^d$  into a probability function. The function  $P(Vec_i^d)$  is defined as:

**Fig. 2** Flow chart of D-CLGSA



$$P(\text{Vec}_i^d(t)) = \left| \tanh(\text{Vec}_i^d(t)) \right| \quad (10)$$

Once  $P(\text{Vec}_i^d(t))$  is evaluated, the movement of the particle will be according to (Eq. 11)

$$\text{if rand} < P(\text{Vec}_i^d(t+1)), \text{ then } X_i^d(t+1) = \text{complement}(X_i^d(t)) \quad \text{else } X_i^d(t+1) = X_i^d(t) \quad (11)$$

The detailed pseudo-code of D-CLGSA is presented in Algorithm 2.

In order to assess the efficiency of D-CLGSA compared to CLGSA, several observations are made:

- Each agent has the ability to gather information from other agents through two means: by studying the historical performance of the best agents and by comparing individual components with other agents in the population. In this context, force acts as a tool for transferring information.
- D-CLGSA efficiently explores the search space, and its sequential behaviors allow for exhaustive discovery of global optima, preventing the algorithm from getting trapped in local optima.
- During the process, if particles become stuck in local optima or if the global optima are located far from the particle's current position, the D-CLGSA strategy assists in guiding the particles to explore other directions, like CLGSA. This cooperative behavior of the swarm enables particles to easily escape local optima. As a result, the strategy effectively avoids being trapped in local optima. The detailed procedure of D-CLGSA is illustrated in the flowchart (Fig. 2).

### 2.2.3 Key points of D-CLGSA

- *Few parameters:* Sensitivity analysis is required to determine the optimal values of parameters for algorithms that have multiple parameters. This process can be time-consuming as it involves finding the parameter values that result in the most optimal solutions. Therefore, it is advantageous to choose metaheuristic algorithms with fewer parameters. In the case of D-CLGSA, only two parameters need to be set: the initial gravitational constant and the decay constant. However, other algorithms

often have more than two parameters that need to be carefully considered before running the optimization process

- *Widespread Participation:* In most metaheuristic algorithms, only a small number of agents contribute to the

solution update process. However, in D-CLGSA, the influence of all agents is taken into account, and the position vectors are updated based on the cumulative force exerted by all agents.

- *Mechanism of exploration and exploitation:* In D-CLGSA, a comprehensive search approach explores the search space and exploited the global solution efficiently. The algorithm is a binary search-based algorithm, where the data is accessed sequentially, and hence requiring reasonable computational time.

### 2.2.4 Potential limitations and challenges of D-CLGSA

The D-CLGSA algorithm may encounter specific limitations and challenges that need addressing. A key challenge is optimizing its performance through the fine-tuning of parameters. Unlike many population-based algorithms that require numerous parameters, D-CLGSA requires setting only two parameters before experimentation that is the initial gravitational constant and the decay constant. However, the fine-tuning of these parameters is critical, especially for certain types of problems. On the upside, the algorithm's comprehensive search capability usually leads to effective convergence, avoiding entrapment in local optima. However, this thorough search process may require considerable memory resources, with its efficiency influenced by the initial parameter settings. Therefore, a deep understanding of the algorithm is crucial to fully harness its capabilities.

Thus, adapting D-CLGSA to a variety of optimization problems, without compromising on its efficiency and effectiveness, requires meticulous consideration in the algorithm's parameter settings. Addressing these challenges and limitations is crucial to fully realize the utility and expand the application scope of D-CLGSA in diverse problem-solving contexts.

**Algorithm 2.** Pseudo-code of Discrete CLGSA.

```

Discrete Comprehensive Learning Gravitational Search Algorithm (D-CLGSA)
Initialization
 $N = Ps$  as  $(X_1^t, X_2^t, \dots, X_{Ps}^t)$  in search range size  $[X_{min}, X_{max}]$ .
Set Iteration Condition
while (not terminated) do:
Set Iteration Parameters
Set  $it = \text{current iteration}$ 
Set  $t_o = it / (it + (V_t * n))$ 
Evaluate Fitness
Evaluate the fitness value  $(fit_1^t, fit_2^t, \dots, fit_{Ps}^t)$ 
Particle Loop
for  $i = 1:Ps$ 
  Calculate  $G^t, best^t, worst^t$  and  $m_i^t$  for  $i, j = 1, 2, \dots, Ps$ 
  Calculate gravitational mass for each agent of P:
  Calculate  $M_i^t = (m_i^t) / (\sum_{i,j} m_j^t)$ 
  Generate initial velocity randomly,
  Calculate the  $K_{best} = rand [(2 + (1 - do) + (100 - 2) * i) / 100] \times X_b^*$ 
  Calculate total force  $F_i^t$ 
  Calculate acceleration  $ac_i^t$  and velocity  $Vec_i^{t+1}$ 
Learning Probability Loop
for  $i = 1$  to  $num$  do
  Set learning probability  $P_c$ ,
  if  $rand() < P_c$ 
    Calculate the probability of changing position vectors.
     $P(Vec_i^d(t)) = (Vec_i^d(t))$  by equations (10) and (11)
    Choose agents  $[X_i] = [rand * V_i \text{ (iterations)}]$ 
     $V_i$  is the previous movement length and the  $rand$  is a random number  $[0,1]$ 
    Find fitness value of these agents.
    if  $(fit[X_1] > fit[X_2])$ 
       $X_b^* = [X_b^*; X_1(i)]$ ;
    else
       $X_b^* = [X_b^*, X_2(i)]$ ; end if ;
    end if
  end for
Best agent check
if  $best^t = 0$  do
  Search agents, obtain new sequence  $\pi(i)$ 
  if total time TFT  $(\pi(i)) < \text{TFT}(\pi(i)^*)$  do
     $T(i) = \pi(i)^*$ ;  $\text{TFT}(\pi(i)) = \text{TFT}(\pi(i)^*)$ 
  else if
     $rand < \exp \{(\text{TFT}(\pi(i)) - \text{TFT}(\pi(i)^*)) / T\}$ 
     $\pi(i) = \pi(i)^*$ ;  $\text{TFT}(\pi(i)) = \text{TFT}(\pi(i)^*)$ 
  end if
  else if  $rand < P_c$ ,  $X(t_{\pi}) = X_i(t) \oplus Vec_i(t+1)$ 
  Update  $G^t, best^t, worst^t$  and  $m_i^t$ 
Generate New Sequence
  else generate new sequence.
end
end

```

**Table 2** Parameters setting

Algorithm	Parameter's value	Algorithm	Parameter's value
D-CLGSA/ CLGSA	No. of masses = 30 $G_0 = 1, \alpha = 20, P_c = 0.9$ Maximum iteration = 500 Stopping criteria = max. iter	CLJA	No. of swarm = 30 $k_1, k_2 = 0.9, \theta_1, \theta_2 = 0.4$ to 0.9 Maximum iteration = 500 Stopping criteria = max. iter
BGSA	No. of masses = 30 $G_0 = 1, \alpha = 20$ Maximum iteration = 500 Stopping criteria = max. iter	GA	No. of particles = 30 Selection- Roulette wheel Crossover prob. = one point, i.e. 0.9 Mutation prob. = uniform, i.e. 0.05 Maximum iteration = 500 Stopping criteria = max. iter
HS	No. of HM solutions = 30 PAR = 0.5, HMCR = 0.5 Maximum iteration = 500 Stopping criteria = max. iter	AEFA	No. of masses = 30 $K_0 = 1, \alpha = 20$ Maximum iteration = 500 Stopping criteria = max. iter
GWO	No. of wolf = 30 $a$ = decreasing linearly from 2 to 0 $A = (1, 0), C = (1, 1)$ Maximum iteration = 500 Stopping criteria = max. iter	SPSO/CLPSO/BPSO	No. of swarm = 30 $c_1 = 2, c_2 = 2$ $w$ = decreasing linearly from 0.9 to 0.6 Maximum iteration = 500 Stopping criteria = max. iter
SCA	No. of swarm = 30 $a = 3$ , range of $\sin = (1, 2]$ , range of $\cosine = [-2, -1]$ Maximum iteration = 500 Stopping criteria = max. iter	ALO	No. of swarm = 30 $w = 3$ when $t > 0.5 T$ Maximum iteration = 500 Stopping criteria = max. iter

### 3 Experimental setup and results analysis over global optimization

#### 3.1 Experimental setup

To assess the performance of the proposed algorithm, an experimental evaluation is conducted using the first global experimental set, which consists of 22 minimization and 2 maximization benchmark functions. The 22 minimization functions are categorized into three groups: unimodal, multimodal, and composite functions. Additionally, there are maximization special (discrete) binary functions, namely Max-ones and Royal Road, all are presented in Appendix A. The dimensions, optimum values, and search ranges of these benchmark functions are provided in detail by E. Rashedi in [57].

The performance of the proposed D-CLGSA algorithm is compared with twelve other heuristic algorithms: Comprehensive Learning Gravitational Search Algorithm (CLGSA) [19], Genetic Algorithm (GA) [25], Binary Gravitational Search Algorithm (BGSA) [57], Harmony Search Algorithm (HS) [48], Binary Particle Swarm Optimization (BPSO) [61], Standard Particle Swarm Optimization (SPSO) [62], Comprehensive Learning Particle Swarm Optimization (CLPSO) [63], Artificial Electric Field Algorithm (AEFA) [64], Grey Wolf Optimizer (GWO) [65], Comprehensive Learning Jaya algorithm

(CLJA) [66], Sine Cosine Algorithm (SCA) [67], and Ant Lion Optimizer (ALO) [68].

For representing each variable in binary format, 15 bits are used, with one bit reserved for the sign. Hence, the dimension of the agents for each continuous function is  $N = D \times 15$ , where  $D$  represents the dimension of the function. In this study, the dimension is fixed to 30, and the results are obtained from 500 runs. The algorithms are implemented using Matlab 2019 platform, and the initial values of the basic parameters for all algorithms are provided in Table 2. The numerical results are presented in Table 3, 4, 5, 6, showing the average and standard deviation (St. Dev.) of the best-obtained solutions in the last iterations.

##### 3.1.1 Statistically significance

In this evaluation, we statistically assess the efficiency of the proposed algorithm using the Wilcoxon Signed Rank Test [69]. A pairwise Wilcoxon test is conducted to compare the performance of D-CLGSA with other algorithms, considering a significance level ( $\alpha$ ) of 0.05.

The null hypothesis assumes that the samples being compared are independent samples from identical continuous distributions with higher average values. If the null hypothesis is rejected at the 5% significance level, it is denoted as '+', indicating that the D-CLGSA approach demonstrates superior performance. Conversely, a '-' symbol indicates

**Table 3** Minimization results of unimodal benchmark functions

Algorithm/Function	Parameters/ Rank test	Pair-wise comparison of D-CLGSA with other algorithms (By Signed Rank Test)						
		F1	F2	F3	F4	F5	F6	F7
BPSO	Average	5.4042	0.341	22.412	2.638	148.136	5.501	0.016
	St.Dev	2.819	0.098	0.921	0.921	137.193	3.071	0.007
	Rank Test	+	+	=	+	+	+	+
BGSA	Average	2045.005	1.563	506.034	8.012	2643.454	125.138	0.024
	St.Dev	40.721	0.732	265.421	3.561	1832.347	88.035	0.027
	Rank Test	+	+	+	+	+	+	+
GA	Average	10.121	0.295	549.384	1.555	369.453	6.984	0.047
	St.Dev	24.205	0.249	249.383	1.384	340.884	7.010	0.043
	Rank Test	+	+	+	+	+	+	+
HS	Average	120.12	2.134	89.13	1.222	567.23	6.267	0.123
	St.Dev	34.12	1.123	3.452	1.432	23.56	2.345	0.089
	Rank Test	+	+	+	–	+	+	+
SPSO	Average	22.12	1.963	48.56	3.675	176.34	51.789	1.334
	St.Dev	22.12	1.675	5.456	1.234	45.34	22.345	1.002
	Rank Test	+	+	+	+	+	+	+
CLPSO	Average	12.008	1.897	26.345	1.564	198.342	5.969	2.143
	St.Dev	3.452	1.896	24.356	1.435	23.768	2.350	1.345
	Rank Test	+	+	+	=	+	=	+
AEFA	Average	138.56	1.678	34.234	1.223	147.34	5.456	1.456
	St.Dev	23.569	3.456	14.673	0.345	9.786	7.453	2.356
	Rank Test	+	+	+	–	+	=	+
GWO	Average	12.437	0.123	31.234	4.567	222.564	19.456	2.346
	St.Dev	8.765	0.456	12.78	3.245	127.45	3.679	1.456
	Rank Test	+	=	+	+	+	+	+
CLJA	Average	0.532	0.119	26.143	1.342	210.342	12.765	0.000
	St.Dev	0.573	0.321	34.12	0.564	123.56	5.673	0.000
	Rank Test	–	=	+	=	+	+	–
SCA	Average	18.24	0.122	45.123	4.352	169.340	6.124	0.560
	St.Dev	4.560	0.121	16.342	1.200	26.453	4.231	1.100
	Rank Test	+	=	+	+	+	+	+
ALO	Average	22.72	1.300	19.435	7.298	132.99	18.340	2.453
	St.Dev	18.19	1.23	23.560	4.234	8.345	6.547	1.453
	Rank Test	+	+	–	+	=	+	+
CLGSA	Average	6.167	0.943	29.134	2.880	140.23	6.451	0.781
	St.Dev	8.920	1.172	12.143	3.125	16.340	3.991	1.342
	Rank Test	+	+	+	+	+	+	+
D-CLGSA	Average	0.7428	0.1242	22.486	1.464	134.381	5.461	0.006
	St.Dev	0.7434	0.1314	38.423	1.375	3.984	3.984	0.008

inferior performance, while '=' signifies that the performance difference is not statistically significant.

### 3.1.2 Time and space complexity

The algorithm complexity is a very good quantifier to judge solutions optimality. Usually, there is always more than one algorithm to solve an optimization problem. An algorithm may find a good solution, however, would take significantly more time as well. Furthermore, researchers prefer algorithms that have less computational complexity with a good ability to find the optimal solution. A details description is given as follows:

The D-CLGSA has time complexity to initialize parameters is  $O(1)$ . For evaluations of fitness value  $O(N) \times Totaliterations$ . For calculating masses, it requires  $O(N) \times Totaliterations$ . For computing gravitational constant, it requires  $O(N) \times Totaliterations$ . To calculate acceleration, it at most takes  $2O(N^2) \times Totaliterations$ . For CL strategy, it takes  $O(N^2) \times Totaliterations$ . For updating position, it takes  $O(N) \times Totaliterations$ . For updating velocity, it takes  $O(N) \times Totaliterations$ . For satisfying the boundary condition, it takes  $O(N) \times Totaliterations$ .

The overall time complexity after all above mentioned complexities for D-CLGSA is  $O(N^2)$  which is similar to the swarm-based algorithms taken into account in this paper. The space complexity is also  $O(N^2)$  Fig. 3.

**Table 4** Minimization results of Multimodal benchmark functions

Algorithm/ Function	Parameters/ Rank Test	Pair-wise comparison of D-CLGSA with other algorithms (By Signed Rank Test)								
		F8	F9	F10	F11	F12	F13	F14	F15	F16
BPSO	Average	−789.1	4.977	2.735	0.387	0.621	1.345	−3.74	−0.034	$6 \times 10^{-4}$
	St.Dev	14.87	1.597	0.472	0.130	0.402	1.567	0.429	0.134	0.003
	Rank Test	−	+	+	+	+	+	+	+	+
BGSA	Average	−827.4	5.273	2.937	0.658	12.678	0.782	−3.56	−0.01	$2 \times 10^{-3}$
	St.Dev	42.647	2.974	1.482	0.230	8.775	0.345	1.126	$3.1 \times 10^{-10}$	$9 \times 10^{-4}$
	Rank Test	−	+	+	+	+	+	+	+	+
GA	Average	−90.34	2.184	1.401	0.706	0.291	0.213	−3.81	−0.583	$1.5 \times 10^{-3}$
	St.Dev	27.987	0.833	1.347	0.332	0.244	0.456	0.567	0.724	$8 \times 10^{-4}$
	Rank Test	+	+	+	+	+	+	=	+	+
HS	Average	−11.43	3.001	2.101	1.321	5.342	1.967	0.231	0.004	0.008
	St.Dev	43.12	2.345	1.532	1.000	2.334	1.342	1.675	0.934	1.342
	Rank Test	+	+	+	+	+	+	+	+	+
SPSO	Average	8.342	4.352	1.983	1.002	3.562	1.399	1.321	−0.100	1.001
	St.Dev	32.120	4.123	2.400	1.217	2.145	1.998	0.546	1.345	2.400
	Rank Test	+	+	+	+	+	+	+	+	+
CLPSO	Average	−91.23	1.996	1.269	0.234	0.252	2.430	−1.200	0.001	0.033
	St.Dev	40.12	1.654	2.431	0.176	1.234	1.441	1.184	1.105	0.667
	Rank Test	+	=	+	+	=	+	+	+	+
AEFA	Average	−11.13	3.214	0.932	2.170	1.001	1.000	0.003	−0.543	$8.3 \times 10^{-2}$
	St.Dev	37.12	1.251	1.324	2.100	3.201	1.200	1.892	0.382	0.520
	Rank Test	+	+	+	+	+	+	+	=	+
GWO	Average	12.094	2.001	0.994	0.763	1.497	0.193	0.532	−0.673	0.003
	St.Dev	18.123	4.325	1.100	1.006	0.128	1.002	0.781	1.543	0.234
	Rank Test	+	+	+	+	+	=	+	+	+
CLJA	Average	−93.56	4.025	1.123	1.20	0.199	1.211	−3.00	1.003	0.001
	St.Dev	5.67	3.214	0.453	0.456	0.342	0.342	0.45	0.345	0.008
	Rank Test	=	+	+	+	=	+	=	+	+
SCA	Average	0.000	1.667	1.008	1.067	1.561	2.067	0.00	0.000	0.0004
	St.Dev	1.23	1.324	1.201	1.300	2.345	3.214	0.37	4.321	0.0003
	Rank Test	+	−	+	+	+	+	+	+	=
ALO	Average	0.000	3.178	2.100	1.564	0.000	0.349	0.18	−0.150	0.98
	St.Dev	12.567	1.567	1.745	0.546	0.045	1.251	0.04	0.456	0.003
	Rank Test	+	+	+	+	−	+	+	+	+
CLGSA	Average	−99.15	2.043	0.911	1.213	0.321	0.560	0.000	0.001	0.002
	St.Dev	2.34	1.874	0.973	1.432	0.463	0.675	0.543	0.754	1.459
	Rank Test	+	+	+	+	+	+	+	+	+
D-CLGSA	Average	−103.14	1.992	0.721	0.154	0.213	0.193	−3.843	−0.543	$4 \times 10^{-4}$
	St.Dev	24.134	1.283	0.843	0.092	0.394	0.221	0.473	0.346	$3.9 \times 10^{-4}$

### 3.2 Results and discussion

The results are discussed over four groups of functions such as unimodal, multimodal, composite, and binary maximization functions presented in Table 3, 4, 5, 6 respectively. The corresponding numerical, statistical, and graphical results of each group are discussed as follows:

#### 3.2.1 Unimodal functions

The unimodal functions (F1 to F7) are generally examined for the exploitation stage of the algorithm. The minimization results of these seven unimodal functions are reported

in Table 3. It can be observed that D-CLGSA consistently achieves good results in terms of 'Average' and 'St. Dev' for most unimodal functions, except for BPSO in F3, CLPSO in F4 and F6, AEFA in F6, and GWO and SCA in F2, which perform equally. HS and AEFA perform better than D-CLGSA in F4, CLJA in F1 and F7, and ALO in F3. The convergence behavior of all algorithms can be seen through the convergence curves in Fig. 4. D-CLGSA demonstrates faster convergence and achieves the best position in the initial iterations, indicating its superior exploitation ability and convergence rate. These findings provide evidence that D-CLGSA is an efficient algorithm. The



**Table 5** Minimization results of Composite benchmark functions

Algorithm / Function	Parameter/ Rank test	Pair-wise comparison of D-CLGSA with other algorithms (By Signed Rank test)					
		F17	F18	F19	F20	F21	F22
BPSO	Average	196.456	146.300	445.239	462.341	170.110	600.235
	St.Dev	60.435	40.245	50.120	32.453	64.567	138.26
	Rank test	+	+	+	+	+	+
BGSA	Average	252.883	254.786	213.450	252.120	245.367	230.455
	St.Dev	53.456	145.230	68.234	50.382	52.456	25.467
	Rank test	+	+	+	+	+	+
GA	Average	193.279	205.677	384.761	567.677	240.890	956.234
	St.Dev	122.350	160.256	118.452	102.345	126.546	67.677
	Rank test	+	+	+	+	+	+
HS	Average	167.23	178.203	222.178	416.234	134.760	812.002
	St.Dev	102.34	134.287	56.342	98.123	55.34	75.301
	Rank test	+	+	+	+	+	+
SPSO	Average	134.239	167.67	306.002	404.321	189.023	634.78
	St.Dev	68.300	98.342	113.650	103.231	69.120	78.349
	Rank test	+	+	+	+	+	+
CLPSO	Average	119.342	83.234	167.342	189.563	156.342	403.231
	St.Dev	67.342	56.342	56.234	57.780	87.352	198.453
	Rank test	+	–	+	+	+	+
AEFA	Average	145.230	101.3424	136.954	202.341	98.342	359.478
	St.Dev	74.231	89.453	35.462	69.301	89.342	176.453
	Rank test	+	+	=	+	+	+
GWO	Average	203.123	145.362	159.342	171.001	89.453	259.087
	St.Dev	106.784	89.345	65.342	41.672	75.231	167.453
	Rank test	+	+	+	=	+	+
CLJA	Average	189.34	112.65	140.34	159.23	104.32	510.23
	St.Dev	87.45	80.00	56.78	34.12	45.35	167.45
	Rank test	+	+	+	–	+	+
SCA	Average	162.67	88.23	204.00	189.99	72.56	403.67
	St.Dev	102.34	56.12	34.67	56.12	56.89	210.56
	Rank test	+	=	+	+	+	+
ALO	Average	149.34	96.329	135.55	256.23	70.98	333.56
	St.Dev	56.34	53.78	42.17	12.89	74.23	156.45
	Rank test	+	+	=	+	=	+
CLGSA	Average	146.23	99.37	139.23	198.40	89.34	309.12
	St.Dev	68.43	78.34	47.128	61.13	73.56	123.89
	Rank test	+	+	+	+	+	+
D-CLGSA	Average	110.234	87.234	136.876	170.345	70.235	212.342
	St.Dev	62.123	48.671	34.256	40.408	67.123	123.670

Wilcoxon rank results in Table 3 also confirm the better performance of the algorithm.

### 3.2.2 Multimodal functions

The minimization results of the nine multimodal functions (F8 to F16) are presented in Table 4. These multimodal functions are commonly used to evaluate the algorithm's ability to avoid local optima. The original CLGSA algorithm is specifically designed to prevent getting stuck in local optima, and its flying particle strategies aid in quickly reaching the global optima. The proposed D-CLGSA algorithm also possesses these characteristics,

as evident from the results in Table 4. D-CLGSA demonstrates the best minimization results for almost all functions, except for the binary algorithms BPSO and BGSA in F8, SCA in F9, and ALO in F12. The convergence behavior of all algorithms with respect to iterations is shown in Fig. 4. It can be observed from Fig. 4 that for function F15, all algorithms struggle to converge, but D-CLGSA achieves convergence within 100 iterations. Additionally, D-CLGSA exhibits the fastest convergence in seven out of the nine multimodal functions. This evidence confirms that the discrete version of CLGSA can effectively avoid local optima without adversely impacting the convergence speed.

**Table 6** Maximization results of Special Binary functions

Algorithm	Parameter/rank test comparison	Pair-wise comparison of D-CLGSA with other algorithms					
		Max-ones D=32	Max-ones D=64	Max-ones D=80	Royal road D=32	Royal road D=64	Royal road D=80
BPSO	Average	32	59.7	72.1	2.7	3.5	3.4
	St.Dev	0	0.9	0.98	0.23	0.34	0.17
	Rank test	=	+	+	+	=	+
BGSA	Average	32	62.3	78.8	4	7.2	10
	St.Dev	0	0	0.33	0	0.42	0.8
	Rank test	=	+	+	+	+	=
GA	Average	28	48.3	58.1	2.2	4.6	4.7
	St.Dev	2.4	1.78	11.2	1.73	0.48	0.72
	Rank test	+	+	+	+	+	+
HS	Average	28	54.3	52.12	4.3	3.2	5.2
	St.Dev	4	1.2	2.3	1.2	0.35	0.32
	Rank test	+	+	+	+	+	+
SPSO	Average	32	59.2	66.34	4.1	4.2	7.5
	St.Dev	0	1.8	3.2	0.5	1.8	0.5
	Rank test	=	+	+	+	+	+
CLPSO	Average	32	59.7	67	6	8	7.5
	St.Dev	0	0.8	4.5	0	0.8	1.2
	Rank test	=	+	+	=	=	+
AEFA	Average	32	58	72.5	6	6.5	7.1
	St.Dev	0	2.3	0.3	0	1.2	1.9
	Rank test	=	+	+	=	+	+
GWO	Average	32	64	72	6	6.5	6.5
	St.Dev	0	0	0.35	0	2.2	0.3
	Rank test	=	=	+	=	+	+
CLJA	Average	32	64	76	6	8	8
	St.Dev	0	0	0.67	0	0	0
	Rank test	=	=	+	=	=	+
SCA	Average	32	54.5	70.13	6	6.5	7
	St.Dev	0	1.7	0.56	0	0.45	1.2
	Rank test	=	+	+	=	+	+
ALO	Average	32	60	65	6	6.5	7.5
	St.Dev	0	0.8	0.3	0	0.1	0.1
	Rank test	=	+	+	=	+	+
CLGSA	Average	32	64	72.5	4.3	6.5	7
	St.Dev	0	0	0.5	1.5	1.2	0.8
	Rank test	=	=	+	+	+	+
D-CLGSA	Average	32	64	80	6	8	10
	St.Dev	0	0	0.5	0	0.6	0.7

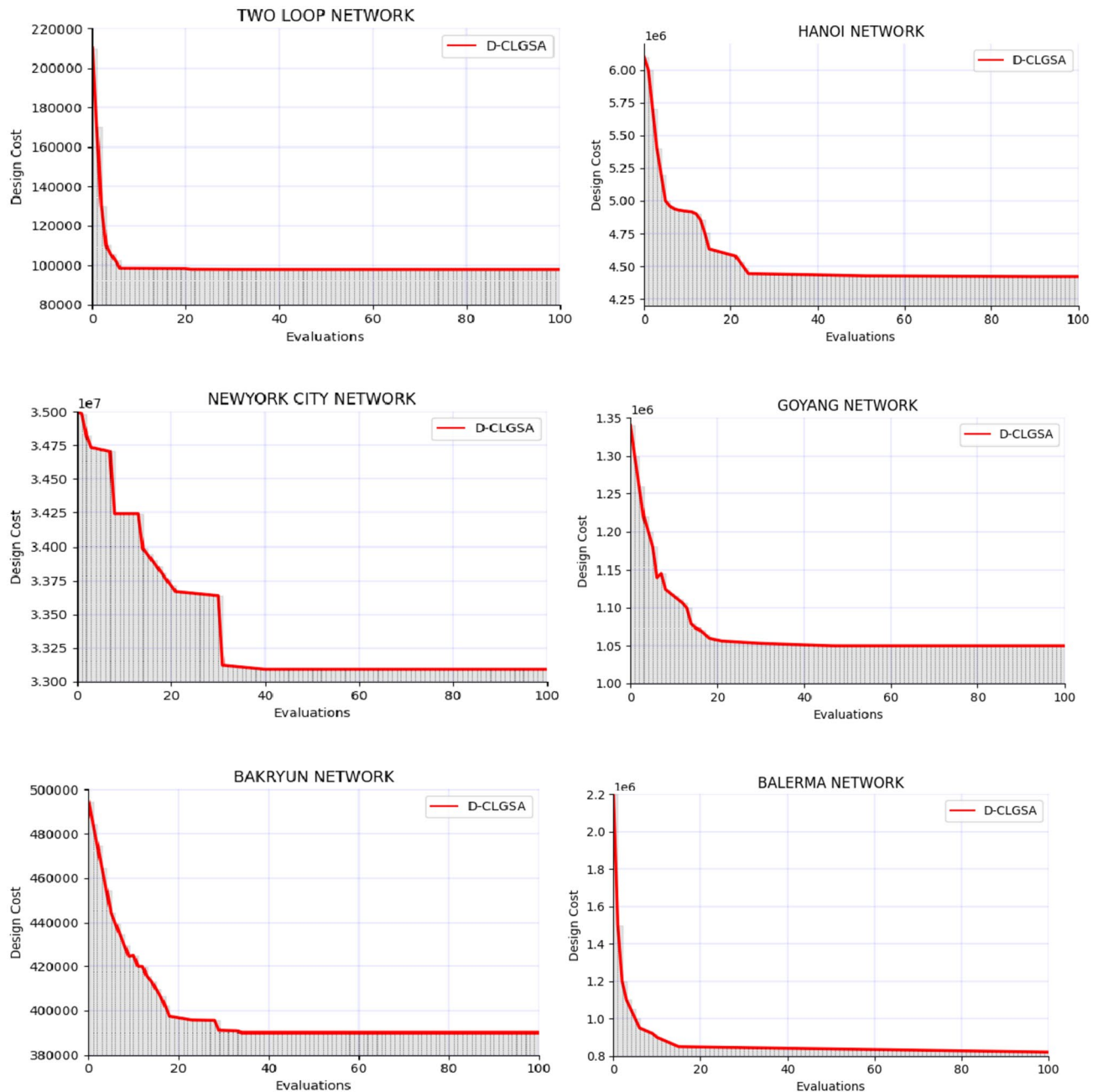
### 3.2.3 Composite functions

The six composite functions (F17 to F22) are the most challenging test functions designed to examine the benchmarking of the exploration and exploitation stage together. The statistical and numerical results in the form of Rank, Average, and Standard deviation are presented in Table 5. D-CLGSA presented good results although CLPSO in F18 and CLJA in F20 out-performed. The convergence of all the algorithms for functions F17 to F22 can be examined in Fig. 4. D-CLGSA converges fast and is stable in the first few iterations. Furthermore, the mean CPU time of all algorithms is presented in Table 7. Among twelve heuristics, D-CLGSA got 3rd position in terms of time computation

which is quite reasonable for the algorithm comprising a comprehensive search strategy. These findings prove that the proposed algorithm efficiently balances exploration and exploitation within a reasonable time.

### 3.2.4 Special discrete functions

The functions (Max-Ones and Royal-Road) are binary in nature; they are presented in Appendix A. These functions should be maximized, so the concept of best and worst in Eq. (8) and Eq. (9) has reversed. For instance, the worst value is evaluated by Eq. (8), and the best value is computed by Eq. (9) for this case. The maximum value of these functions depends on the dimensions. The numerical and

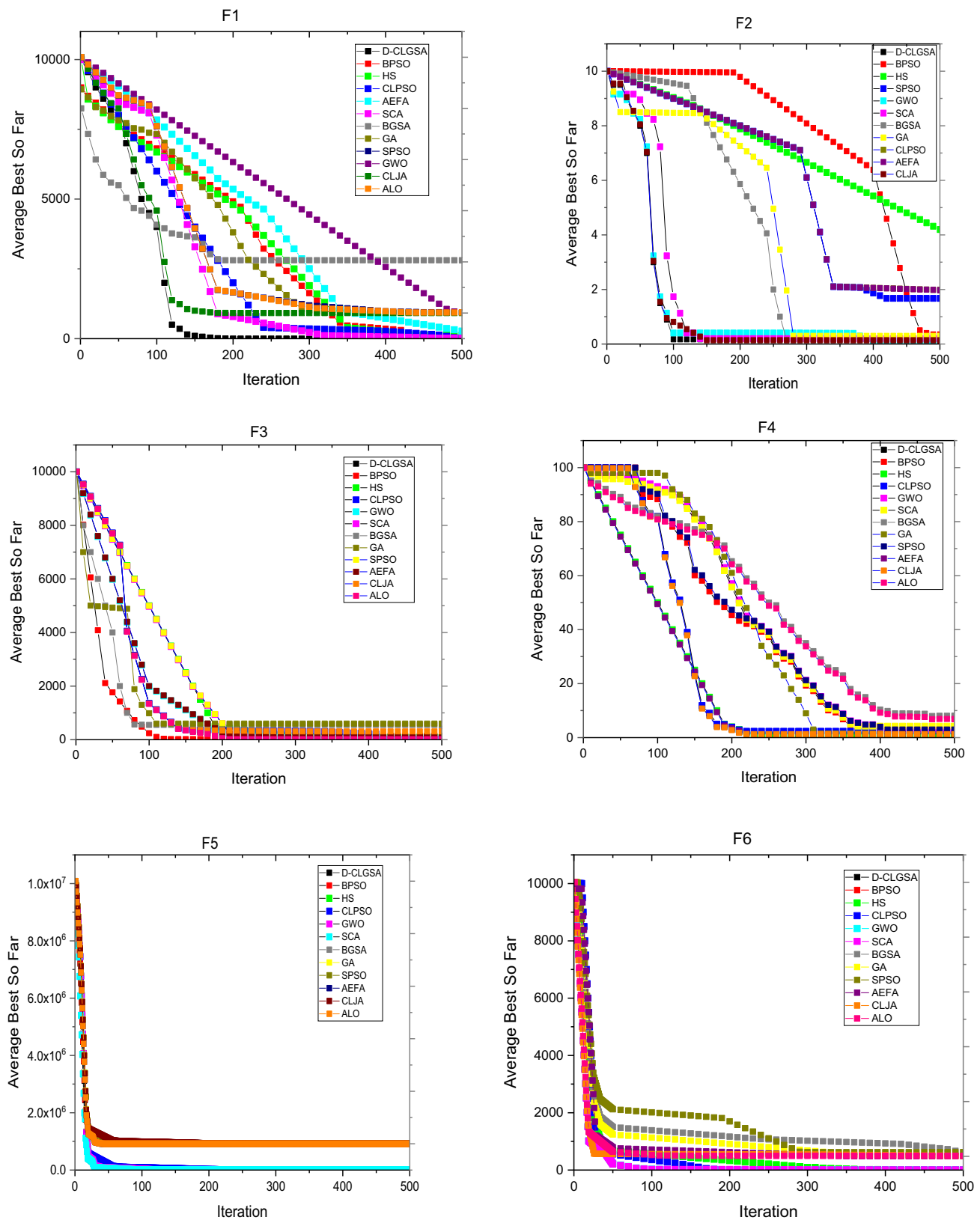


**Fig. 3** Water networks analysis using D-CLGSA

statistical results are presented in Table 6 for the different dimensions such as 32, 64, and 80, and the iterations set for these functions are 1000. From Table 6, D-CLGSA provides the optimum solution for all cases. It can be noted that the most significant difference in the performance of D-CLGSA with other algorithms occurs primarily when

the dimension of the functions increases. In addition, the good convergence rate of D-CLGSA could be concluded in Fig. 5.

Hence, all summarized results prove that D-CLGSA is capable of solving unimodal, multimodal, composite, and discrete functions successfully.



**Fig. 4** Convergence graph of unimodal (F1-F7), multimodal (F8-F16), composite (F17-F22) Functions

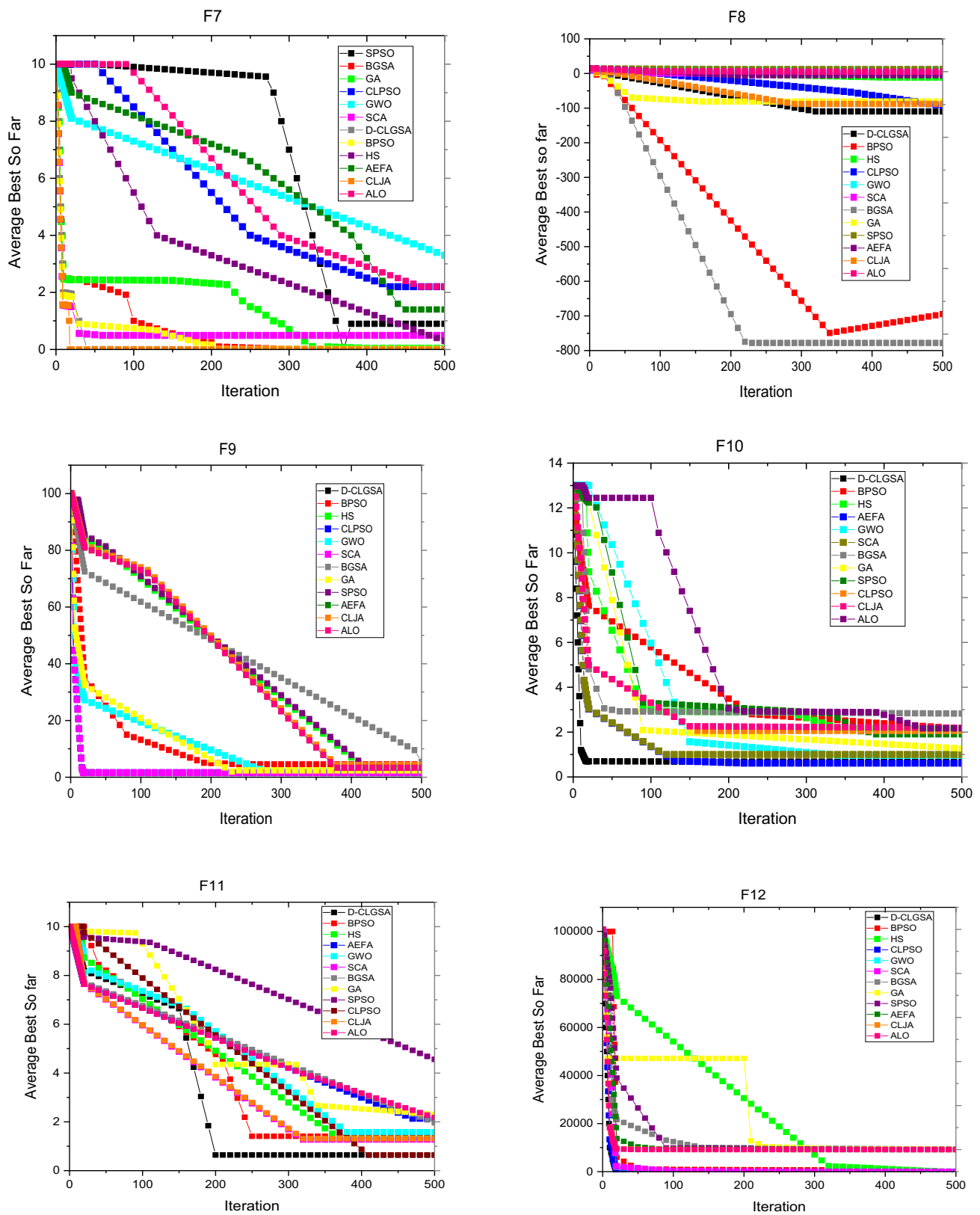


Fig. 4 (continued)



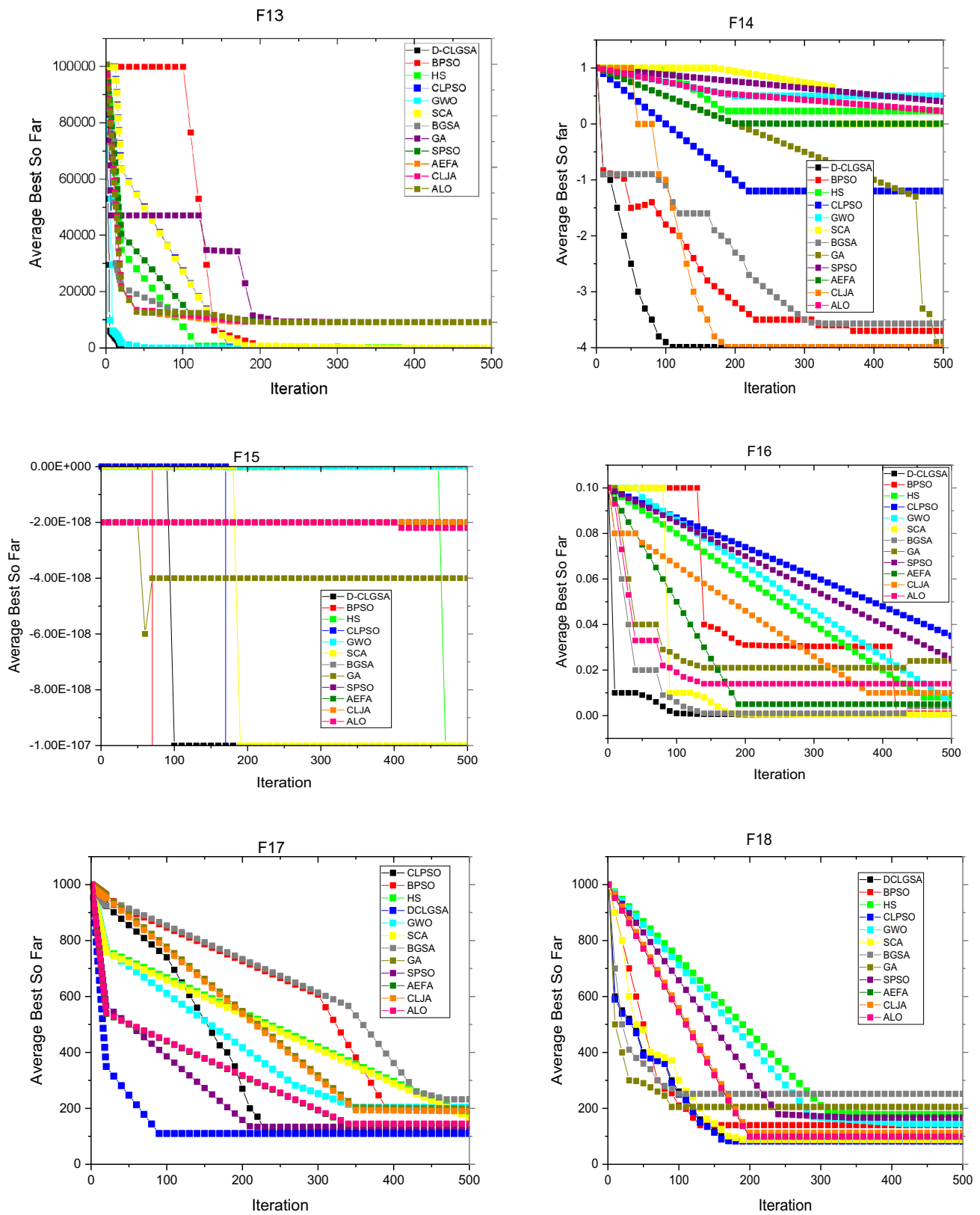


Fig. 4 (continued)

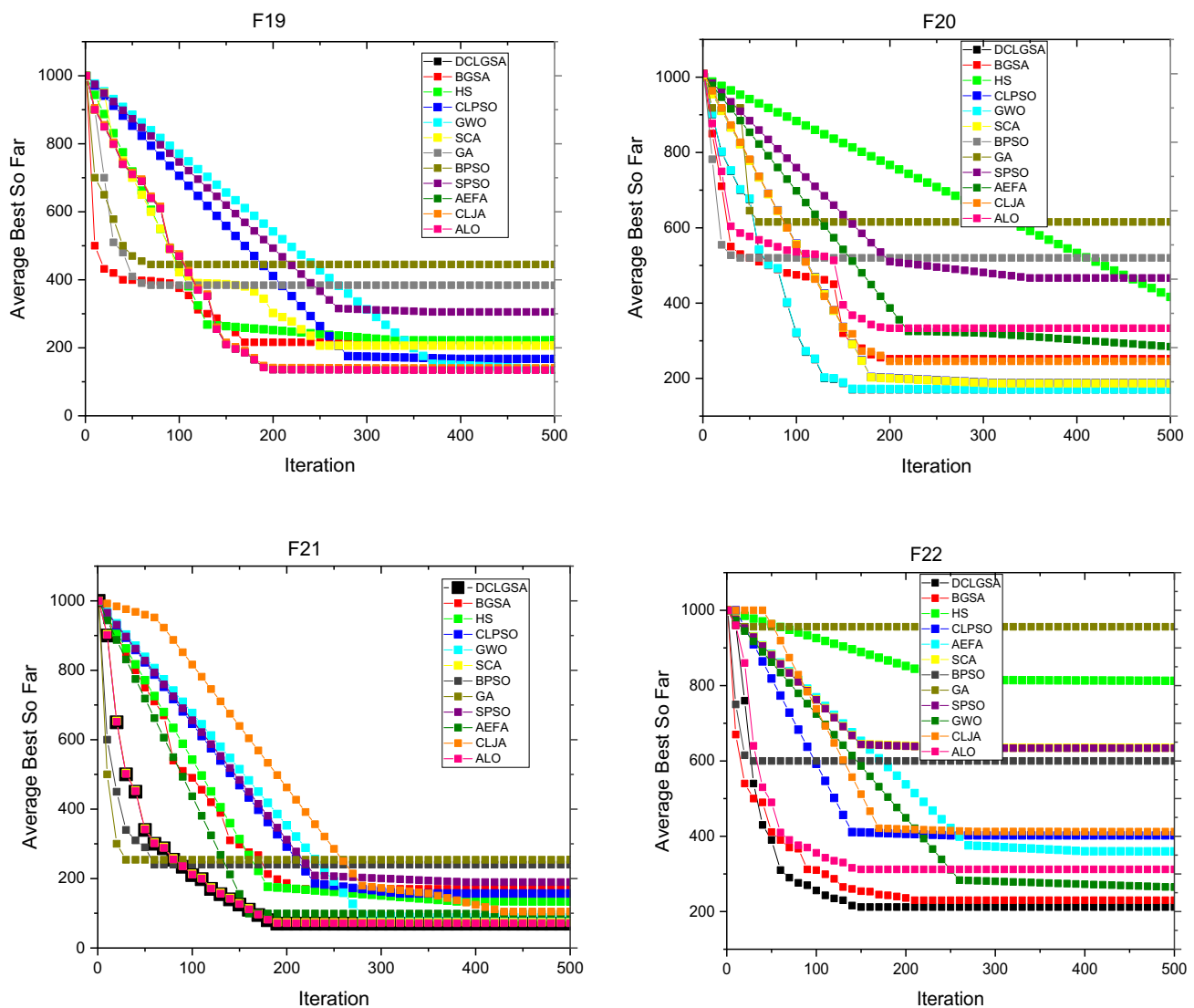


Fig. 4 (continued)

**Table 7** Mean CPU time of all algorithms

Algorithm	BPSO	BGSA	GA	HS	SPSO	CLPSO	AEFA	GWO	CLJA	SCA	ALO	CLGSA	D-CLGSA
Mean CPU time (s)	317	139	512	414	457	402	349	389	367	234	267	408	252

## 4 Water distribution network

The D-CLGSA algorithm has proven its ability to adeptly navigate through discrete decision-making environments, effectively handling various types of optimization problems in previous section. This demonstrated effectiveness forms the basis of our choice to apply D-CLGSA to Water Distribution Network (WDN) optimization challenges. WDNs, defined by their discrete variables like

pipe diameters and valve operations, are complex systems requiring precise configuration for optimal functionality. D-CLGSA's suitability for such discrete variables and its proficiency in addressing complex constraints make it an exemplary choice for optimizing WDNs.

The algorithm's strength lies in its balanced approach to both exploring wide-ranging solution spaces and honing in on the most promising solutions. This balance is crucial for dealing with the diverse constraints that WDNs present. Moreover, the increasing complexity and scale of

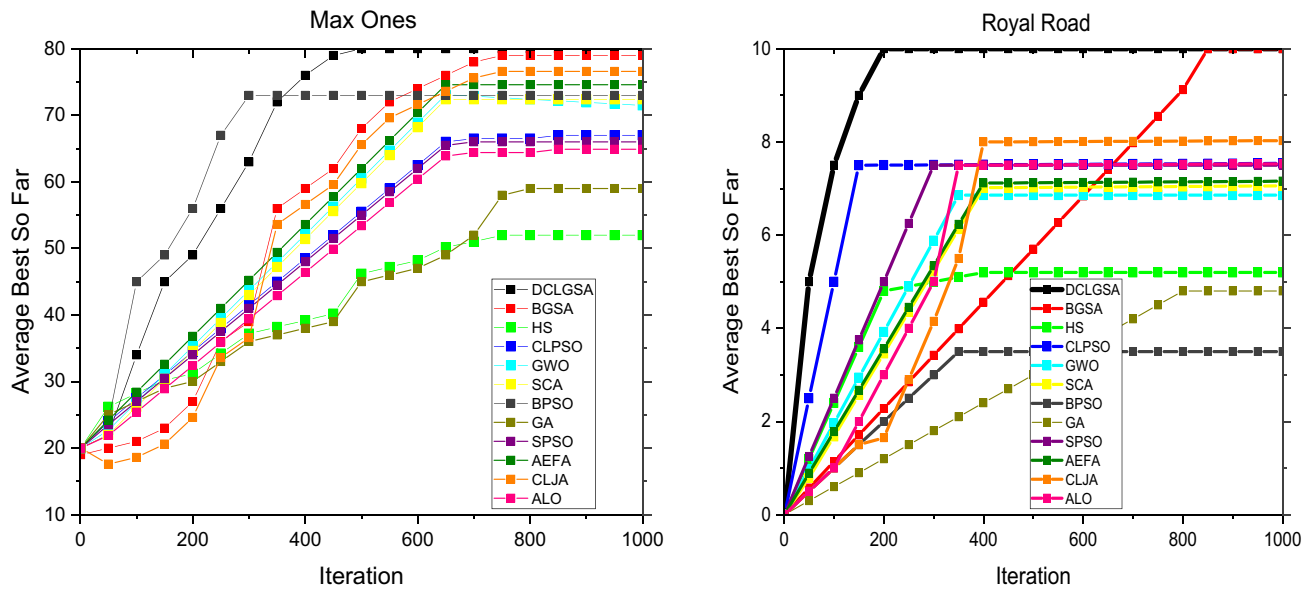


Fig. 5 Convergence graph of binary maximization functions (Max once, Royal Road)

WDNs, with more variables and larger networks, highlight the importance of D-CLGSA's scalability. Its design, which leverages a binary search space and dynamic particle movement, allows D-CLGSA to effectively keep pace with the evolving demands of WDN optimization, affirming its role as a powerful tool in this domain, which is explained in the subsequent discussion.

#### 4.1 Formulation of the water distribution system

The preliminary aim of designing WDS is to obtain the optimal pipe diameters. The parameters of this problem are discrete, and the objective function calculates the minimal constructing cost of the WDS is expressed as:

$$\text{Mincost} = \sum_{i=1}^{n_{\text{pipe}}} C_i(D_i) \times L_i \quad (12)$$

where,  $C_i(D_i)$ =cost per unit length of pipe diameter  $D_i$ ,  $L_i$ =Length of pipe  $i$ ,  $n_{\text{pipe}}$ =Total pipe count in a WDN. The objective function is conditioned by the minimum head requirements in the demand nodes, and conservation law of energy and mass.

The evaluation of residual head at every edge point or node is performed employing hydraulic analysis; EPANET 2.0 adopts the Hazen–Williams (HW) mathematical relation defined as:

$$h_i = 4.727R_i^{-1.852} \times Q_i^{1.852} \times D_i^{-4.871} \times L_i \quad (13)$$

where,  $h_i$ =The loss in reservoir head across the pipe  $i$ , and  $R_i$ =HW roughness coefficient.

$Q_i$ =rate of water flow,  $D_i$ =pipe diameter and  $L_i$ =pipe length.

Certain constraints are involved while determining the connectivity of pipe, layout, nodal demand, and minimal head necessities are discussed as:

##### 4.1.1 Minimum constraints of pressure

The ideal design of WDN should meet the minimum pressure requirement at every node. The minimum pressure constraint is determined as:

$$H_j^{\min} \leq H_j, j = 1, 2, 3, \dots, \text{node} \quad (14)$$

$H_j$  and  $H_j^{\min}$ =pressure and the minimal pressure head at node  $j$ , respectively.

##### 4.1.2 Constraints for energy conservation

In every loop present in the WDN, the constraints for conserving energy are given as:

$$\sum_{k \in NL} \Delta H_k = 0 \quad (15)$$

where,  $\Delta H_k$ =loss in water head in pipe  $k$ ,  $NL$ =number of loops present in the system.

The loss of water head in every pipe is described as the difference between the connected nodes and is evaluated employing Hazen–William's equation as:

$$\Delta H_k = H_{1,k} - H_{2,k} = w \left( \frac{(L_k)}{C_k^\alpha D_k^\beta} \right) Q_k |Q_k|^{\alpha-1}, \forall k \in npipe \quad (16)$$

$C_k$  = coefficient of roughness in pipe  $k$ ,  $Q_k$  = rate of water flow in the pipe,  $\alpha$ , and  $\beta$  = coefficients of regressions. The coefficients of the HW equation,  $w$ ,  $\alpha$ , and  $\beta$  are 10.667, 1.852, and 4.871, respectively.

#### 4.1.3 Constraints for mass conservation

At every node, the conservation constraint for mass must be satisfied.

$$\sum Q_{in} - \sum Q_{out} = Q_c \quad (17)$$

where,  $Q_{in}$  = rate of flow into the node,  $Q_{out}$  = rate of flow out of the node,  $Q_c$  = rate of external inflow at the node.

#### 4.1.4 Implementation procedure of D-CLGSA on WDN

In this study, we utilize a combined simulation optimization computer model that integrates the D-CLGSA algorithm and EPANET, a hydraulic simulator tool. EPANET is powerful software developed by the United States Environmental Protection Agency, capable of conducting comprehensive hydraulic and water quality simulations for pressurized water distribution networks (WDN). It provides a dynamic link library (DLL) of functions that can be customized to meet specific user requirements. Additionally, the EPANET-MATLAB toolkit was developed, enabling users to access EPANET through a programming interface in external software. In this integrated model, the D-CLGSA algorithm serves as the external driver model, while EPANET functions as the internal model. The algorithm imports the layout and data of the water network from EPANET, including design parameters such as roughness coefficient, available pipe sizes, associated costs, required pressures, and flow rates for the network. The D-CLGSA algorithm requires certain parameters to be set by the user, such as the number of agents, population size, and dimensions. During the hydraulic simulation, each data point is tested against pressure constraints at different nodes using the EPANET toolkit [70]. The number of nodes that fall below the minimum required pressure is counted, and this count is used as a penalty multiplier ( $\mu$ ). The fitness index of each data point is then computed as the sum of the cost and the penalty (if applicable). This process helps evaluate the performance of different data points and guide the optimization process towards finding better solutions for the water distribution network design. Mathematically, the fitness index is expressed as:

$$Fitnessindex = Cost + (\mu \times penalty) \quad (18)$$

The value of  $\mu$  become zero if the pressures at all nodes are at least the minimum pressure required. The penalty approach is inducted to convert the constrained into unconstrained problems. Due to the conversion, the solution may fall outside the feasible region penalized, and be forced to fall into the feasible region after a few iterations. During this process, proper parameter tuning of parameters is required. Because, when the penalty parameters are large, the penalty function tends to be ill-conditioned near the boundary of the feasible domain and that may tend the process to local optima. If this situation occurs, D-CLGSA performs the comprehensive search described in algorithm 2 and flew out toward the feasible region. Some repeated runs may require getting satisfactory results (with no constraints violation). The stepwise procedure of implementation is described below:

1. *Generation of initial population*: First, we produce a set of initial solutions through random generation. For instance, each of the 24 bits in a string can be set to either 0 or 1. In the scenario of a pipe network, a binary value is assigned to each discrete pipe diameter. Additional bits are utilized to denote various choices for each pipe being evaluated. Consequently, each string signifies a distinct arrangement of pipe sizes and corresponds to a unique configuration of the pipe network. Find the fitness value of each particle and choose the local best and global best solution by CLGSA (algorithm 1).
2. *Computation of network cost*: In the algorithm, each string in the population is processed one by one. The decoding process involves determining the pipe size based on the substrings, and then the cost is calculated using Eq. (12).
3. *Hydraulic analysis of each network*: The steady-state hydraulic analysis is performed for each network in the population, which calculates the heads and discharges based on the given demand patterns. The computed pressures are then compared with the minimum allowable pressure heads to identify any discrepancies.
4. *Computation of penalty cost*: Each particle undergoes pressure constraint testing on various nodes using the EPANET toolkit. The number of nodes that have pressure below the minimum required value is determined, and this count is used as a penalty multiplier ( $\mu$ ) to calculate the number of violations.
5. *Computation of total network cost*: The fitness index of each network is computed by calculating the total cost, which is the sum of the network cost and the penalty, using Eq. (18).
6. *Update Fitness, Local, and Global Best*: The initial velocity vector of each individual is determined using Eq. (10). The particles' new positions are then calculated using Eq. (11). The fitness index of each particle is com-

puted based on its new position, and the local and global best positions are updated accordingly.

7. *Update Position and Velocity*: The velocities of the particles are updated using Eq. (10), and subsequently, the new positions of the particles are determined based on Eq. (11).
8. *Termination*: In case the termination criterion is not met, Steps 2 to 7 will be repeated.

## 5 Water network applications and results discussion

In this section, we will discuss six water networks of different countries as Two-Loop Network, Hanoi Network, New-York City Network, GoYang Network, BakRyun Network, and Balerna Water Network. The motive of the objective function (Eq. 12) is to minimize the cost function, which depends on the pipe length and diameter. So, the motive can be achieved by reducing the length and diameter of the pipe, along with satisfying all hydraulic constraints. The optimal route for WDN was designed using MATLAB 2019. Furthermore, the performance evaluation of the pipe network is classified into two parts. In the first part, the proposed algorithm D-CLGSA is compared with the base approaches, which were specially designed to solve the water networks. These approaches provide a set of pipe lengths, which helps to minimize the cost function. In the 2nd approach, the proposed algorithm is compared statistically with recent heuristics which are multi-tasking in nature and can be able to solve other real-life optimization problems as well. Here, the results are presented in the form of the Minimum Cost, Maximum Cost, and Average Cost of the network. These metrics evaluate the performance of the proposed algorithm in comparison to base methods and recent heuristics.

### 5.1 Two-loop network

The Two-loop network was first proposed by Alperovits et al. [21]. This network consists of two loops with seven nodes and eight pipes, and the gravity-fed water reservoir has a fixed head of 210 m. The height of the pipes is 1000 m with a Hazen-Williams coefficient ( $HW = 130$ ). The elevation corresponding to each node is {210, 150, 160, 155, 150, 165, 160 m}, and the demand at each node is {−1120, 100, 100, 120, 270, 330, and  $200 \text{ m}^3/\text{hr}$ }. The unit cost of for fourteen different pipe diameters pipe and D-CLGSA parameter values are listed in Table 8. In the optimization problem, engineering expertise could be utilized to choose more effective initial feasible solutions. However, in this study, the initial solutions are generated randomly, and only feasible solutions are used as starting points. In cases where infeasible solutions are encountered, a penalty value of 1.05 million is applied in addition to the penalty multiplier  $\mu$  (as detailed in Eq. (18)). The 1.05 million penalty value corresponds to the cost of implementing an 18-inch pipe network.

Table 9 presents the diameters of the pipes for the Two-loop network obtained using different approaches, along with the obtained pressure head values: {32.2, 32.4, 31.3, 35.5, 37.5, 40, and 35.6 m}. This table compares the results obtained using the D-CLGSA-based model with those obtained using other methods, including Alperovits et al. [21], Kessler et al. [22], Goulter et al. [24], Savic et al. [71], and the proposed D-CLGSA. Among all, D-CLGSA solves the Two-loop network for \$406,489, which is the least cost among all methods. However, the solutions of all models except D-CLGSA deliver two segments of diameters with different discrete sizes. According to Savic et al., a split-pipe design should be more realistic if one diameter is chosen from each pipe. Hence,

**Table 8** Diameter of the pipes and value of D-CLGSA parameters

Network	Pipe diameter	Unit cost	No of Var's	D-CLGSA Max. Evaluations (NI)
Two-loop	{1, 2, 3, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24} in inches	{2,5,8,11,16,23,32,50,60,90,130,170,300,550} in dollar/ meter	8	5000
Hanoi	{12, 16, 20, 24, 30, 40} in inches	{45.73, 70.4, 98.38, 129.3, 180.8, 278.3} in dollar/meter	34	12,010
New York	{36, 48, 60, 72, 84, 96, 108, 120, 132, 144, 156, 168, 180, 192, 204} in inches	{93.5, 134, 176, 221, 267, 316, 365, 417, 469, 522, 577, 632, 689, 746, 804} in dollar/foot	21	2100
GoYang	{80, 100, 125, 150, 200, 250, 300, 350} in millimeters	{37,890, 38,933, 40,563, 42,554, 47,624, 54,125, 62,109, 71,524}, in won/meter	30	10,000
BakRyun	{300, 350, 400, 450, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1350, 1500, 1600, 2000} in millimeters	{118,000, 129,000, 145,000, 160,000, 181,000, 214,000, 242,000, 285,000, 325,000, 370,000,434,000, 434,000, 434,000, 434,000, 4,340,000}, in won/meter	9 pipes (1–9)	5000
Balerna	{113, 125, 142.6, 160.2, 175.34, 224.2, 285, 361.8, 452.2, 581.8}, in millimeters	{7.22, 8.5, 10.53, 13.5, 17.12, 26.75, 44.12, 76.32, 124.64, 215.85}, in EUR/meter	10	25,000



**Table 9** Comparative data for different diameters of pipes for two-loop network

Pipe no	1	2	3	4	5	6	7	8	Cost (\$)
Alperovits et al [21]	20	8	18	8	16	12	6	6	497,525
	18			6	16	10	6	4	
Goulter et al [24]	20	10	16	6	16	12	10	2	435,015
	18			4	14	10	8	1	
Kessler et al [22]	18	12	12	10	3	16	12	10	417,500
	18			16	2	14	10	8	
D-CLGSA	18	10	16	4	16	10	10	1	406,489
	18			4	16	10	10	1	

D-CLGSA considered the best possible diameter for each pipe and delivered the least cost water network.

Alperovits et al. model is based on a non-split-pipe solution using up to 25,000 cost function evaluations (NI) and 10 runs. Goulter et al. used 70,000 evaluations and 1000 runs, and Kessler et al. used 15,000 evaluations and 1000 runs. However, D-CLGSA produced better results with only 5000 evaluations (NI) and five runs (Table 8). This method uses a less approving hydraulic conversion constant ( $\omega = 10.5879$ ) than the genetic algorithm ( $\omega = 10.5088$ ).

Furthermore, Table 10 presents a statistical analysis of cost optimization using different recent stochastic algorithms such as GA, BGSA, HS, BPSO, AEFA, GWO, CLJA, SCA, and ALO. The results of D-CLGSA are presented in the “Minimum,” “Maximum,” and “Average Network Cost” columns. Table 10 shows that D-CLGSA solves networks at the least cost. Figure 3 presents the performance of D-CLGSA in graphical form.

## 5.2 Hanoi network

The Hanoi WDN was first proposed by Fujiwara & Khang in 1990 [72] in Vietnam. The network is made up of thirty-two nodes, thirty-four pipes, three loops, a reservoir with 100 m, and HW constant is 130. The basic permissive level for the head of the reservoir is 30 m. Moreover, the Hanoi network is first compared by Fujiwara et al. [72] and Savic et al. [71] model. The data for the optimal length of the pipe are tabulated in Table 11. The demand ( $m^3/hr$ ) for 32 nodes are {−19,940, 890, 850, 130, 725, 1005, 1350, 550,525, 525, 500,560, 940, 615, 280, 310, 865, 1345, 60, 1275, 930, 485, 1045, 820, 170, 900, 370,290, 360,360, 105,805} respectively. A penalty worth \$10.5 million is applied, which is

equivalent to the expense of constructing the network using pipes with a diameter of 24 inches.

Fujiwara modal solves this problem by a non-linear programming method with  $\omega = 10.5088$ . They obtained \$6,320,000 an optimal cost after 1,000,000 evaluations presented in Table 11. Savic et al. solve the same problem in 1,000,000 evaluations with  $\omega = 10.5088$ , and the optimal cost of the network was \$6,073,000. However, the proposed D-CLGSA solves the same problem with  $\omega = 10.5088$ , and the optimal cost is \$6,056,000 using 12,010 function evaluation, using an Intel(R) 1.8-GHz processor. The pressure obtained by D-CLGSA corresponding to 32 nodes lies between 30.5 and 42.8 m.

The statistical results of the Hanoi network are compared with recent studies presented in Table 12, which includes the minimum, average, and maximum costs reported by the considered optimizers. It can be observed that the ALO algorithm obtained the least cost-optimal network, while D-CLGSA attained the 2nd position. ALO required 31,122 function evaluations, whereas D-CLGSA achieved the optimal cost in just 12,010 function evaluations, which is the least among all. Figure 3 illustrates the performance of the algorithm in optimizing WDN, highlighting that the D-CLGSA algorithm quickly reduces the Hanoi Network’s cost in the initial evaluations and demonstrates swift convergence to an optimal solution after roughly 20 evaluations.

## 5.3 New-York City WDN

The New York City water network model is constructed with twenty nodes, twenty-one pipes, one loop, and the gravity of the reservoir with a 300-ft (fixed head) [48]. The main aim of this model is to include additional pipes analogous to the

**Table 10** Statistical comparison of cost-optimization in two-loop network

Serial no	1	2	3	4	5	6	7	8	9	10
Methods	GA	BGSA	HS	BPSO	AEFA	GWO	CLJA	SCA	ALO	D-CLGSA
Minimum cost (\$ Million)	0.419	0.432	0.414	0.465	0.422	0.499	0.513	0.567	0.487	0.406
Maximum cost (\$ Million)	0.463	0.498	0.482	0.479	0.478	0.501	0.561	0.592	0.512	0.411
Average cost (\$ Million)	0.442	0.486	0.458	0.472	0.458	0.493	0.537	0.579	0.491	0.408

**Table 11** Comparative data for different diameters of pipes for the Hanoi model

Pipe number	Length of pipe (m)	Fujiwara et al	Savic et al	D-CLGSA	Pipe number	Length of pipe (m)	Fujiwara et al	Savic et al	D-CLGSA
1	100	40	40	40	18	800	29.01	20	20
2	1350	40	40	40	19	400	29.28	20	20
3	900	40	40	40	20	2200	38.58	40	40
4	1150	40	40	40	21	1500	17.36	20	20
5	1450	40	40	40	22	500	12.65	12	12
6	450	40	40	40	23	2650	32.59	40	40
7	850	38.16	40	40	24	1230	22.06	30	30
8	850	36.74	40	40	25	1300	18.34	30	30
9	800	35.33	40	40	26	850	12.00	20	20
10	950	29.13	30	30	27	300	22.27	12	12
11	1200	26.45	24	24	28	750	24.57	12	12
12	3500	23.25	24	24	29	1500	21.29	16	16
13	800	19.57	20	20	30	2000	19.34	16	12
14	500	15.62	16	16	31	1600	16.52	12	12
15	550	12.00	12	12	32	150	12.00	12	16
16	2730	22.50	12	12	33	860	12.00	16	16
17	1750	25.24	16	16	34	950	22.43	20	24
Cost (\$)					–	6,320,000	6,073,000	6,056,000	

**Table 12** Statistical optimization results obtained by several optimizers for the Hanoi network

SL No	Methods	Minimum cost (\$ million)	Maximum cost (\$ million)	Average cost (\$ million)	Maximum no. of evaluations
1	GA	6.173	6.956	6.573	26,450
2	HS	6.081	6.319	6.213	27,720
3	BPSO	6.134	6.333	6.301	39,510
4	BGSA	6.350	6.451	6.444	27,410
5	GWO	6.111	6.675	6.575	34,120
6	AEFA	6.453	6.516	6.404	35,450
7	CLJA	6.555	6.900	6.726	16,000
8	SCA	6.212	6.512	6.363	25,620
9	ALO	6.012	6.316	6.164	31,122
10	D-CLGSA	6.056	6.1078	6.094	12,010

already inducted pipes since the actual model can't satisfy the requirement of pressure at a few nodes (nodes 16–20). The different lengths of pipe are tabulated in Table 13. The model has an H-W constant of 100. Table 8 displays the candidate diameters and their corresponding cost values, while the demand values and settings remain consistent with those mentioned by Geem et al. [48]. The penalty value is set at \$90.3 million which is equivalent to the expense of constructing the network using pipes with a diameter of 180 inches.

The comparison results derived using the proposed algorithm and other mechanisms are given in Table 13; the third column tabulates the data obtained from (Schaake & Lai, 1969) [20], data from the fourth column is taken from

(Savic & Walters, 1997) [71], and fifth one is from Cunha et al. [26]. (Schaake & Lai, 1969) Used linear programming methodology and got \$78,090,000 as the optimal cost. The same model solved by (Savic & Walters, 1997) got \$37,130,000 with  $\omega = 10.5088$  using a GA model after ten lakh evaluations. The D-CLGSA model solved the problem using  $\omega = 10.5088$  and got an optimal cost of \$36,660,000 after six thousand evaluations, and this takes up to 20 min using an Intel(R) 1.8-GHz. Further, Statistical results of the New-York city network are compared with metaheuristic and presented in Table 14. It shows the statistical optimization results, including the best, average, and worst costs reported by considered approaches. In terms of the minimum optimized cost, D-CLGSA is superior to all other methods.

**Table 13** Comparative data for different pipe diameters for the New-York City network

Pipe no	Length of pipe (ft)[27]	Schaake et al	Savic et al	Cunha et al	D-CLGSA	Pipe no	Length of pipe (ft)	Schaake et al	Savic et al	Cunha et al	D-CLGSA
1	11,600	52.02	0	0	0	12	12,200	125.25	0	0	0
2	19,800	49.90	0	0	0	13	24,100	126.87	0	0	0
3	7300	63.41	0	0	0	14	21,100	133.07	0	0	0
4	8300	55.59	0	0	0	15	15,500	126.52	0	0	0
5	8600	57.25	0	0	0	16	26,400	19.52	96	96	96
6	19,100	59.19	0	0	0	17	31,200	91.83	96	96	96
7	9600	59.06	108	108	96	18	24,000	72.76	84	84	84
8	12,500	54.95	0	0	0	19	14,400	72.61	72	72	72
9	9600	0.0	0	0	0	20	38,400	0.0	0	0	0
10	11,200	0.0	0	0	0	21	26,400	54.82	72	72	72
11	14,500	116.21	0	0	0	Cost (\$1000)	–	78,090	37,130	37,130	36,660

Figure 3 graphically depicts the convergence of the D-CLGSA algorithm applied to the New York City Network, demonstrating a significant reduction in design costs initially. This pattern suggests the algorithm's proficiency in quickly approaching an optimal solution, with subsequent improvements becoming more incremental.

## 5.4 GoYang WDN

The GoYang network, which was proposed by Kim et al. in 1994 [23] in South Korea, consists of twenty nodes, thirty pipes, and nine loops. The power of the pump used in the network is 4.52 kW with a fixed head of 71 m. The pipe lengths for the network are provided in Table 15. The network has an HW coefficient is 100. Table 8 provides values for eight commercial diameters along with cost values, and the specification that the minimum height from the ground level should be 15 m. The demand values and pipe length remain consistent with those mentioned by Geem et al. [48]. The minimum pressure values for nodes 1–20 are listed as {300, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 260, 272, 255, 255, and 255 m} in the given context. The penalty value is set at 350,000,000 won which is equivalent to the expense of constructing the network using pipes with a diameter of 150 mm.

The above-mentioned Table 15 gives the comparative analysis for the diameter evaluated using the proposed model and other techniques. The 3rd column provides the analysis from the actual design, the 4th column gives the data from (Kim et al.) [23] and the 5th column provides results from the current analysis. This model solved the problem and got the cost of 176,100,800 won after ten thousand mathematical iterations and 25 steps, which takes 11 min on an Intel(R) 1.8-GHz processor. The pressure values obtained for each node using D-CLGSA are as follows: {301.1, 290.3, 285.2, 280, 280.3, 280.3, 275.7, 275.1, 273.8, 275.1, 272.8, 273.4, 275.7, 270.8, 273.7, 275, 273.4, 275.0, 272.3, and 282.3 m}.

Further, the statistical results of metaheuristic are compared with the proposed D-CLGSA are presented in Table 16. The D-CLGSA algorithm reached the lowest cost for the Goyang Network faster than any competing algorithms, within just 10,000 evaluations. Figure 3 illustrates this efficiency, showing a quick drop to the best solution and then little change in cost thereafter, confirming D-CLGSA's superiority in achieving cost-effective solutions efficiently.

## 5.5 BakRyun WDN

The BakRyun network model was first utilized by (Lee et al.) [52] in South Korea. The model is constructed using thirty-five nodes, fifty-eight pipes, and seventeen loops with a water reservoir of a fixed length of 58 m. The purpose of this model is to calculate the sizes of the new pipes (pipes 1–3) and

**Table 14** Statistical analysis of cost-optimization for New-York City network

SI No	Method	Minimum cost (\$ million)	Maximum cost (\$ million)	Average cost (\$ million)	Maximum no. of evaluations
1	GA	38.64	39	38.54	49,950
2	BGSA	37.12	40.67	39.94	6600
3	HS	38.64	40.21	38.12	3373
4	AEFA	37.04	40.40	39.12	17,000
5	GWO	39.14	39.99	39.82	10,000
6	BPSO	40.94	40.94	40.94	10,000
7	CLJA	37.21	39.12	38.17	19,800
8	SCA	39.67	40.12	39.99	45,870
9	ALO	37.78	42.18	39.98	37,000
10	D-CLGSA	36.66	38.36	37.65	2100

**Table 15** Comparative data for diameters of pipes for the GoYang network

Pipe no	Pipe length (m)[48]	Diameter original (mm) [48]	Diameter (NLP) (mm) [23]	Diameter D-CLGSA (mm)	Pipe no	Pipe length (m)	Diameter (original) (mm)	Diameter (NLP) (mm) [23]	Diameter (D-CLGSA) (mm)
1	165.0	200	200	150	16	261.0	80	80	80
2	124.0	200	200	125	17	72.0	80	80	80
3	118.0	150	125	125	18	373.0	80	100	80
4	81.0	150	125	125	19	98.0	80	125	80
5	134.0	150	100	100	20	110.0	80	80	80
6	135.0	100	100	80	21	98.0	80	80	80
7	202.0	80	80	80	22	246.0	80	80	80
8	135.0	100	80	80	23	174.0	80	80	80
9	170.0	80	80	80	24	102.0	80	80	80
10	113.0	80	80	80	25	92.0	80	80	80
11	335.0	80	80	80	26	100.0	80	80	80
12	115.0	80	80	80	27	130.0	80	80	80
13	345.0	80	80	80	28	90.0	80	80	80
14	114.0	80	80	80	29	185.0	80	100	80
15	103.0	100	80	80	30	90.0	80	80	80
Cost (won)						–	179,428,600	179,142,700	176,100,800

the parallel pipes (pipes 4–9). For all the pipes in the network, the coefficient  $C$  assigned is 100. The base value for the limitation of the water head is 15 m above the ground level. Table 8 displays a list of sixteen commercial diameters and their corresponding cost values, while the demand values and pipe parameters remain the same as those stated in Geem et al. [48]. The original pressure head on these nodes are {32.76, 26.95, 25.33, 15.13, 20.88, 27.08, 31.68, 21.01, and 23.44 m}. The penalty value is set at 1,050,000,000 won which is equivalent to the expense of constructing the network using pipes with a diameter of 800 mm.

The comparative analysis of the pipe diameter is tabulated in Table 17. The pressure head obtained for each node using D-CLGSA are as follows: {32.76, 26.96, 25.40, 15.13, 20.88, 27.10, 31.68, 21.05, and 23.45 m}. The data was

compared using the proposed methodology with other methods. The 3rd column consists of data derived using actual design, the 4th column data is taken from the literature [52] and the 5th column data is from the current research. The network solved using GA resulted in an absolute cost of 903,620,000won and the model based on D-CLGSA obtained the same minimal cost but after five thousand functional evaluations, and this requires five minutes using an Intel(R) 1.8-GHz processor.

Table 18 also presents the statistical results of all algorithms in terms of the minimum, maximum, and average cost of the water network. The minimum cost of GA and D-CLGSA are equal, but D-CLGSA attains the cost value in 5000 iterations and the same cost value obtained by GA in 10,000 iterations. The performance of D-CLGSA using

**Table 16** Statistical analysis of cost-optimization in GoYang network

Sl No	Method	Minimum cost (\$ million)	Maximum cost (\$ million)	Average cost (\$ million)	Maximum evaluations
1	GA	1.774	1.82	179.5	60,000
2	BGSA	1.790	1.83	1.82	20,000
3	HS	1.772	1.79	1.78	40,000
4	BPSO	1.81	1.81	1.81	20,000
5	AEFA	1.84	1.84	1.84	20,000
6	GWO	1.78	1.79	1.79	20,000
7	CAJA	1.77	1.89	1.83	22,800
8	SCA	1.92	NA	NA	56,500
9	ALO	1.771	1.775	1.773	15,350
10	D-CLGSA	1.76	1.765	1.76	10,000

1000 won  $\approx$  1 US dollar

Bakryun and its graphical analysis is given in Fig. 3. It displays the D-CLGSA algorithm's application to the Bakryun Network, showing a substantial initial drop in design cost followed by stabilization. This suggests that D-CLGSA rapidly achieves an efficient solution, after which it consistently upholds performance levels.

## 5.6 Balerma WDN

The Balerma WDN was initially proposed by Reza et al. in 2006 [73]. The network consists of four reservoirs, eight loops, four hundred fifty-four pipes, and four hundred forty-three demand nodes. The absolute roughness coefficient,  $R$ , is 0.0025 mm for each pipe, and the minimum head requirement for each node is 20 m.

Table 8 provides the cost data for 10 different pipe diameters within the range of 113.0–581.8 mm, while considering the demand values and pipe parameters suggested by

Geem [48]. There is an overall  $10^{454}$  feasible solution for the absolute cost designing of the Balerma WDN. Pipe diameters are extracted from an auxiliary materials site of the American Geophysical Union (<ftp://ftp.agu.org/wr/2005wr004383/>).

The penalty cost is set at 5.2 million euros for this network. The pressure head obtained by D-CLGSA lies between 24.3 and 38.9. Table 19 presents a comparison of the optimization results with those obtained by the proposed D-CLGSA. The D-CLGSA algorithm achieves the second-lowest cost, taking 25,000 iterations and 18 min on an Intel i3 processor. The BGSA algorithm achieves the lowest cost; however, it requires 50,000 function evaluations to do so. Figure 3 presents the D-CLGSA's performance on the Balerma network through graphical analysis, confirms that the algorithm efficiently reaches an optimal solution and then maintains consistent performance.

**Table 18** Statistical analysis of cost-optimization in the BakRyun network

Sl No	Method	Minimum cost (\$ million)	Maximum cost (\$million)	Average cost (\$million)	Maximum evaluations
1	HS	0.91	0.93	0.926	20,000
2	GA	0.903	0.94	0.927	10,000
3	BGSA	1.02	1.34	1.30	10,000
4	BPSO	1.56	1.67	1.60	20,000
5	AEFA	1.99	1.99	1.99	20,000
6	GWO	0.99	0.99	0.99	20,000
7	CLJA	0.911	NA	NA	22,000
8	SCA	1.20	1.23	1.22	32,500
9	ALO	0.910	1.13	1.02	7200
10	D-CLGSA	0.903	0.912	0.907	5000

**Table 17** Comparative data for diameters of pipes for the BakRyun network

Pipe Number	Pipe Length (m) [48]	Diameter (original) (mm) [48]	Diameter (GA) (mm) [52]	Dia. (D-CLGSA) (mm)
1	200.0	1100	1100	1100
2	470.0	1100	1100	1000
3	80.0	1100	1000	1000
4	370.0	900	900	1000
5	410.0	900	900	900
6	540.0	800	700	700
7	530.0	600	700	700
8	130.0	500	300	300
9	470.0	500	300	300
Cost (won)	-	954,920,000	903,620,000	903,620,000

**Table 19** Statistical analysis of cost-optimization in the Balerna network

Sl No	Method	Minimum cost (€ million)	Maximum cost (€ million)	Average cost (€ million)	Maximum Evaluations
1	GA	3.738	3.843	3.754	100,000
2	BGSA	1.777	2.675	1.977	50,000
3	HS	2.201	2.721	2.653	100,000
4	BPSO	4.777	4.777	4.777	50,000
5	AEFA	4.329	4.678	4.456	25,000
6	GWO	4.342	4.342	4.342	50,000
7	CLJA	4.122	NA	NA	72,000
8	SCA	3.120	3.981	3.552	42,700
9	ALO	3.864	3.952	3.910	37,420
10	D-CLGSA	2.0189	2.0196	2.0191	25,000

## 6 Conclusion

This study introduces a discrete version of the CLGSA algorithm for solving optimization problems. The experiment is divided into two parts. In the first part, the proposed algorithm is evaluated on a standard benchmark consisting of unimodal, multimodal, composite, and special binary functions. The robustness of the algorithm is tested by evaluating the results on higher dimensions. The numerical, statistical, and graphical ways are used to demonstrate the results and to compare them with 12 other heuristic algorithms. The results show that the proposed algorithm is more efficient than others and can achieve the solution in less computational time. In the second set of experimentation, the algorithm is applied to minimize the cost of six different water distribution networks. These networks include Two-loop, Hanoi, New-York City, GoYang, BakRyun, and Balerna. The results are compared with metaheuristic algorithms using a performance indicator that uses minimum function evaluation and cost value of the network. The D-CLGSA algorithm was able to solve the Two-loop network with a cost of \$0.406 million in 5000 evaluations. Similarly, the

Hanoi network was solved with a cost of \$6.05 million in 12,010 evaluations, the New York City network was solved with a cost of \$36.6 million in 2100 evaluations, the GoYang network was solved with a cost of 176,100,800 won in 10,000 evaluations, the BakRyun network was solved with a cost of 903,620,000 won in 5000 evaluations, and the Balerna network was solved with a cost of €2.01 million in 25,000 evaluations. The proposed D-CLGSA approach is shown to be capable of achieving the minimum cost unit with the minimum number of function evaluations. Among the six networks, the D-CLGSA approach provides the least cost for five networks when compared to other algorithms.

In summary, the proposed D-CLGSA algorithm exhibits the capacity to explore optimal solutions, particularly in the context of water network design. While demonstrating notable performance in the assessed scenarios, it holds potential as a valuable tool for optimizing water distribution networks, and with minor adjustments, it could be applied to various other network designs. Nonetheless, its efficiency may vary across different optimization problem types, prompting the need for future investigations into its adaptability to a broader array of problem domains. It is worth noting that the performance of the D-CLGSA approach may be influenced by parameter choices, since our evaluation employed finely tuned parameters. Subsequent research could delve into a comparative analysis of alternative penalty function approaches, addressing parameter sensitivity for a more robust application. Additionally, future developments for D-CLGSA include its combination with real-time data integration and multi-objective optimization to broaden its utility in various scientific and engineering domains. Furthermore, merging D-CLGSA with deep learning techniques could revolutionize its capability for intricate pattern detection and prediction in complex optimization scenarios. This advancement is anticipated to elevate the algorithm's functionality, facilitating smarter and self-reliant optimization solutions in a spectrum of high-tech arenas.



## Appendix A

Functions: Uni-modal (F1- F7), Multimodal (F8-F16), Composite (F17-F22).

Functions	Range	$F_{min}$
$F1(X) = \sum_{i=1}^n X_i^2$	$[-100, 100]$	0
$F2(X) = \sum_{i=1}^n  X_i  + \prod_{i=1}^n  X_i $	$[-10, 10]$	0
$F3(X) = \sum_{i=1}^n \left( \sum_{j=1}^i X_j \right)^2$	$[-100, 100]$	0
$F4(X) = \max\{ X_j , j = 1, 2, 3 \dots, n\}$	$[-100, 100]$	0
$F5(X) = \sum_{j=1}^{n-1} \left[ 100(X_{j+1} - X_j)^2 + (X_j - 1)^2 \right]$	$[-30, 30]$	0
$F6(X) = \sum_{j=1}^n [X_j + 0.5]^2$	$[-100, 100]$	0
$F7(X) = \sum_{j=1}^n jX_j^4 + \text{random}[0, 1)$	$[-1.28, 1.28]$	0
$F8(X) = \sum_{j=1}^n -X_j \sin\left(\sqrt{ X_j }\right)$	$[-500, 500]$	$-418.992 \times 5$
$F9(X) = \sum_{j=1}^n X_j^2 - 10 \cos(2\pi X_j) + 10$	$[-5.12, 5.12]$	0
$F10(X) = -20 \exp\left(-0.2\sqrt{\frac{1}{n} \sum_{j=1}^n X_j^2}\right) - \exp\left(\frac{1}{n} \sum_{j=1}^n \cos(2\pi X_j)\right) + 20 + e$	$[-32, 32]$	0
$F11(X) = \frac{1}{4000} \left[ \sum_{i=1}^n X_i^2 - \prod_{i=1}^n \cos\left(\frac{X_i}{\sqrt{i}}\right) + 1 \right]$	$[-600, 600]$	0
$F12(X) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{j=1}^{n-1} (y_j - 1)^2 [1 + 10 \sin^2(\pi y_{j+1})] + (y_n - 1)^2 \right\}$ $+ \sum_{j=1}^n u(X_j, 10, 100, 4); y_j = 1 + \frac{X_j + 1}{4}, u(X, b, k, m) = \begin{cases} k(X_j - b)^m, X_j > b \\ 0 - b < X_j < b \\ k(-X_j - b)^m, X_j < -b \end{cases}$	$[-50, 50]$	0
$F13(X) = 0.1 \left\{ \sin^2(3\pi X_1) + \sum_{j=1}^n (X_j - 1)^2 [1 + \sin^2(3\pi X_{j+1})] + (X_{n-1})^2 [1 + \sin^2(2\pi X_n)] \right\} + \sum_{j=1}^n u(X_j, 5, 100, 4);$	$[-50, 50]$	0
$F14(X) = -\sum_{j=1}^n \sin(X_j) \cdot \left( \sin\left(\frac{jX_j^2}{\pi}\right) \right)^{2m}; m = 10$	$[0, \pi]$	-4.687
$F15(X) = \left[ e^{-\sum_{j=1}^n \left(\frac{X_j}{\beta}\right)^{2m}} - 2e^{-\sum_{j=1}^n X_j^2} \right] \cdot \prod_{j=1}^n \cos^2 X_j; m = 5$	$[-20, 20]$	-1
$F16(X) = \left\{ \left[ \sum_{j=1}^n \sin^2(X_j) \right] - \exp\left(-\sum_{j=1}^n X_j^2\right) \right\} \cdot \exp\left[-\sum_{j=1}^n \sin^2\left(\sqrt{ X_j }\right)\right]$	$[-10, 10]$	-1
$F17(CF1) : F1, F2, \dots, F10 = \text{Spherefunction},$ $[\beta_1, \beta_2, \dots, \beta_{10}] = [1, 1, \dots, 1],$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = \left[ \frac{5}{100}, \frac{5}{100}, \dots, \frac{5}{100} \right]$	$[-5, 5]$	0
$F18(CF2) : F1, F2, \dots, F10 = \text{Griewankfunction},$ $[\beta_1, \beta_2, \dots, \beta_{10}] = [1, 1, \dots, 1],$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = \left[ \frac{5}{100}, \frac{5}{100}, \dots, \frac{5}{100} \right]$	$[-5, 5]$	0
$F19(CF3) : F1, F2, \dots, F10 = \text{Griewankfunction},$ $[\beta_1, \beta_2, \dots, \beta_{10}] = [1, 1, \dots, 1],$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = [1, 1, \dots, 1]$	$[-5, 5]$	0
$F20(CF4) : F1, F2 = \text{Ackleytsfunction},$ $F3, F4 = \text{Rastrigintsfuction}$ $F5, F6 = \text{Weierstrassfunction}$ $F7, F8 = \text{Griewanktsfunction}$ $F9, F10 = \text{Spherefunction}$ $[\beta_1, \beta_2, \dots, \beta_{10}] = [1, 1, \dots, 1],$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = \left[ \frac{5}{32}, \frac{5}{32}, 1, 1, \frac{5}{0.5}, \frac{5}{0.5}, \frac{5}{100}, \frac{5}{100}, \frac{5}{100}, \frac{5}{100} \right]$	$[-5, 5]$	0

Functions	Range	$F_{min}$
$F21(CF5) : F1, F2 = \text{Rastrigin's function},$ $F3, F4 = \text{Weierstrass function}$ $F5, F6 = \text{Griewank's function}$ $F7, F8 = \text{Ackley's function}$ $F9, F10 = \text{Sphere function}$ $[\beta_1, \beta_2, \dots, \beta_{10}] = [1, 1, \dots, 1],$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = \left[ \frac{1}{5}, \frac{1}{5}, \frac{5}{0.5}, \frac{5}{0.5}, \frac{5}{100}, \frac{5}{100}, \frac{5}{32}, \frac{5}{32}, \frac{5}{100}, \frac{5}{100} \right]$	$[-5, 5]$	0
$F22(CF6) : F1, F2 = \text{Rastrigin's function},$ $F3, F4 = \text{Weierstrass function}$ $F5, F6 = \text{Griewank's function}$ $F7, F8 = \text{Ackley's function}$ $F9, F10 = \text{Sphere function}$ $[\beta_1, \beta_2, \dots, \beta_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, \dots, 1],$ $[\lambda_1, \lambda_2, \dots, \lambda_{10}] = \left[ 0.1 * \frac{1}{5}, 0.2 * \frac{1}{5}, 0.3 * \frac{5}{0.5}, 0.4 * \frac{5}{0.5}, 0.5 * \frac{5}{100}, 0.6 * \frac{5}{100}, 0.7 * \frac{5}{32}, 0.8 * \frac{5}{32}, 0.9 * \frac{5}{100}, 1 * \frac{5}{100} \right]$	$[-5, 5]$	0
$\text{Max-Ones} = \sum_{j=1}^D X_j$	$\{0, 1\}^D$	-
$\text{Royal Road} = \sum_{i=1}^D \left( \prod_{j=8(i-1)+1}^{8i} X_j \right)$	$\{0, 1\}^D$	-

**Acknowledgements** I extend my sincere gratitude to Prof. Lewis Mitchell from The University of Adelaide for their invaluable support throughout this endeavour.

**Author contributions** I.B.: conceptualization, data curation, formal analysis, investigation, methodology, validation, visualization, writing—original draft; A.Y.: formal analysis, investigation, methodology, supervision, validation, visualization, review, JH.K. review, editing, supervision. All authors reviewed the manuscript.

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions.

## Declarations

**Conflict of interest** We declare we have no competing interests.

**Ethical approval** This work did not require ethical approval from a human subject or animal welfare committee.

**Supplementary material** Additional Material: The code is available at the GitHub repository: <https://github.com/InduBala-Y/Discrete-CLGSA>

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will

need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Powell WB (2019) A unified framework for stochastic optimization. *Eur J Op Res* 275(3):795–821
- Roewa O, Slavov T, and Fidanova S (2014). Population-based vs. single point search meta-heuristics for a pid controller tuning. In *Handbook of research on novel soft computing intelligent algorithms: theory and practical applications*, pp. 200–233. IGI Global.
- Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. *Inf Sci* 179(13):2232–2248
- Bala I, Yadav A (2018) Gravitational search algorithm: a state-of-the-art review. *Harmon Search Nat Inspir Optim Algorithms: Theory Appl ICHSA* 2019:27–37
- Sarafrazi S, Nezamabadi-pour H, Saryazdi S (2011) Disruption: a new operator in gravitational search algorithm. *Sci Iranica* 18(3):539–548
- Gao S, Vairappan C, Wang Y, Cao Q, Tang Z (2014) Gravitational search algorithm combined with chaos for unconstrained numerical optimization. *Appl Math Comput* 231:48–62
- Güvenc U, Katircioğlu F (2019) Escape velocity: a new operator for gravitational search algorithm. *Neural Comput Appl* 31:27–42
- Haghighyan P, Nezamabadi-Pour H, Kamyab S (2017) A niche GSA method with nearest neighbor scheme for multimodal optimization. *Swarm Evol Comput* 35:78–92
- Sarafrazi S, Nezamabadi-Pour H, Seydnejad SR (2015) A novel hybrid algorithm of GSA with Kepler algorithm for numerical optimization. *J King Saud Univ-Comput Inf Sci* 27(3):288–296

10. Bala I, Yadav A (2022) Niching comprehensive learning gravitational search algorithm for multimodal optimization problems. *Evol Intel* 15(1):695–721
11. Bala I, Malhotra A (2019) Fuzzy classification with comprehensive learning gravitational search algorithm in breast tumor detection. *Int J Recent Technol Eng* 8(2):2688–2694
12. Tan Z, Zhang D (2020) A fuzzy adaptive gravitational search algorithm for two-dimensional multilevel thresholding image segmentation. *J Ambient Intell Humaniz Comput* 11(11):4983–4994
13. Lei Z, Gao S, Gupta S, Cheng J, Yang G (2020) An aggregative learning gravitational search algorithm with self-adaptive gravitational constants. *Expert Syst Appl* 152:113396
14. Gauthama Raman MR, Somu N, Jagarapu S, Manghnani T, Selvam T, Krithivasan K, Shankar Sriram VS (2020) An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm. *Artif Intell Rev* 53:3255–3286
15. Younes Z, Alhamrouni I, Mekhilef S, Reyesudin MJ (2021) A memory-based gravitational search algorithm for solving economic dispatch problem in micro-grid. *Ain Shams Eng J* 12(2):1985–1994
16. Zamfirache IA, Precup R-E, Roman R-C, Petriu EM (2022) Reinforcement Learning-based control using Q-learning and gravitational search algorithm with experimental validation on a nonlinear servo system. *Inf Sci* 583:99–120
17. Qian J, Wang P, Chenggen Pu, Peng X, Chen G (2023) Application of effective gravitational search algorithm with constraint priority and expert experience in optimal allocation problems of distribution network. *Eng Appl Artif Intell* 117:105533
18. Ahmad Z, Mahmood T, Rehman A, Saba T, and Alamri FS (2023). Enhancing time series forecasting with an optimized binary gravitational search algorithm for echo state networks. *IEEE Access*
19. Bala I, Yadav A (2020) Comprehensive learning gravitational search algorithm for global optimization of multimodal functions. *Neural Comput Appl* 32:7347–7382
20. Schaake Jr, JC., and Lai D (1969). Linear programming and dynamic programming application to water distribution network design.
21. Alperovits E, Shamir U (1977) Design of optimal water distribution systems. *Water Resour Res* 13(6):885–900
22. Kessler A, Shamir U (1989) Analysis of the linear programming gradient method for optimal design of water supply networks. *Water Resour Res* 25(7):1469–1480
23. Kim JH, Kim TG, Kim JH, Yoon YN (1994) A study on the pipe network system design using non-linear programming. *J Korean Water Resour Assoc* 27(4):59–67
24. Goulter IC, Lussier BM, Morgan DR (1986) Implications of head loss path choice in the optimization of water distribution networks. *Water Resour Res* 22(5):819–822
25. Simpson AR, Dandy GC, Murphy LJ (1994) Genetic algorithms compared to other techniques for pipe optimization. *J Water Resour Plan Manag* 120(4):423–443
26. Cunha MDC, Sousa J (1999) Water distribution network design optimization: simulated annealing approach. *J Water Resour Plan Manag* 125(4):215–221
27. Surco DF, Vecchi TPB, Ravagnani MA (2018) Optimization of water distribution networks using a modified particle swarm optimization algorithm. *Water Sci Technol: Water Supply* 18(2):660–678
28. Sarbazfard S, Jafarian A (2017) A hybrid algorithm based on firefly algorithm and differential evolution for global optimization. *J Adv Comput Res* 8(2):21–38
29. Lin M-D, Liu Y-H, Liu G-F, Chu C-W (2007) Scatter search heuristic for least-cost design of water distribution networks. *Eng Optim* 39(7):857–876
30. Costa ALH, De Medeiros JL, Pessoa FLP (2000) Optimization of pipe networks including pumps by simulated annealing. *Braz J Chem Eng* 17:887–896
31. Giudicianni C, Herrera M, Di Nardo A, Adeyeye K (2020) Automatic multiscale approach for water networks partitioning into dynamic district metered areas. *Water Resour Manag* 34:835–848
32. Jafari-Asl J, Sami Kashkooli B, Bahrami M (2020) Using particle swarm optimization algorithm to optimally locating and controlling of pressure reducing valves for leakage minimization in water distribution systems. *Sustain Water Resour Manag* 6:1–11
33. Jafari-Asl J, Azizyan G, Monfared SAH, Rashki M, Andrade-Campos AG (2021) An enhanced binary dragonfly algorithm based on a V-shaped transfer function for optimization of pump scheduling program in water supply systems (case study of Iran). *Eng Fail Anal* 123:105323
34. Surco DF, Macowski DH, Cardoso FAR, Vecchi TPB, Ravagnani MA (2021) Multi-objective optimization of water distribution networks using particle swarm optimization. *Desalin Water Treat* 218:18–31
35. Jain P, Khare R (2022) Multi-objective Rao algorithm in resilience-based optimal design of water distribution networks. *Water Supply* 22(4):4346–4360
36. Choi YH (2022) Development of optimal water distribution system design and operation approach considering hydraulic and water quality criteria in many-objective optimization framework. *J Comput Des Eng* 9(2):507–518
37. Xia W, Wang S, Shi M, Xia Q, Jin W (2022) Research on partition strategy of an urban water supply network based on optimized hierarchical clustering algorithm. *Water Supply* 22(4):4387–4399
38. Bhavya R, Elango L (2023) Ant-inspired metaheuristic algorithms for combinatorial optimization problems in water resources management. *Water* 15(9):1712
39. Hajibabaei M, Yousefi A, Hesarkazzazi S, Minaei A, Jenewein O, Shahandashti M, Sitzenfrie R (2023) Resilience enhancement of water distribution networks under pipe failures a hydraulically inspired complex network approach. *Aqua—Water Infrastruct Ecosyst Soc* 72(12):2358–2376
40. Vu Hong PS, Thanh VN (2023) Application of artificial intelligence algorithm to optimize the design of water distribution system. *Int J Constr Manag* 23(16):2830–2840
41. Shekofteh MR, Yousefi-Khoshqalb E, Piratla KR (2023) An efficient approach for partitioning water distribution networks using multi-objective optimization and graph theory. *Water Resour Manag* 37(13):5007–5022
42. Johns MB, Keedwell E, Savic D (2014) Adaptive locally constrained genetic algorithm for least-cost water distribution network design. *J Hydroinf* 16(2):288–301
43. Zhang K, Yan H, Zeng H, Xin K, Tao T (2019) A practical multi-objective optimization sectorization method for water distribution network. *Sci Total Environ* 656:1401–1412
44. Shao Yu, Yao H, Zhang T, Chu S, Liu X (2019) An improved genetic algorithm for optimal layout of flow meters and valves in water network partitioning. *Water* 11(5):1087
45. Poojitha SN, Singh G, Jothiprakash V (2020) Improving the optimal solution of GoYang network—using genetic algorithm and differential evolution. *Water Supply* 20(1):95–102
46. Geem ZW (2009) Particle-swarm harmony search for water network design. *Eng Optim* 41(4):297–311

47. Bilal, Pant M (2020) Parameter optimization of water distribution network—a hybrid metaheuristic approach. *Mater Manuf Process* 35(6):737–749
48. Geem ZW (2006) Optimal cost design of water distribution networks using harmony search. *Eng Optim* 38(03):259–277
49. Perelman L, Ostfeld A (2007) An adaptive heuristic cross-entropy algorithm for optimal design of water distribution systems. *Eng Optim* 39(4):413–428
50. Shibu A, Reddy MJ (2014) Optimal design of water distribution networks considering fuzzy randomness of demands using cross entropy optimization. *Water Resour Manag* 28:4075–4094
51. Jabbary A, Podeh HT, Younesi H, Haghiabi AH (2016) Development of central force optimization for pipe-sizing of water distribution networks. *Water Sci Technol: Water Supply* 16(5):1398–1409
52. Lee HM, Yoo DG, Sadollah A, Kim JH (2016) Optimal cost design of water distribution networks using a decomposition approach. *Eng Optim* 48(12):2141–2156
53. Fallah H, Ghazanfari S, Suribabu CR, Rashedi E (2021) Optimal pipe dimensioning in water distribution networks using gravitational search algorithm. *ISH J Hydraul Eng* 27:242–255
54. Ezzeldin RM, Djebedjian B (2020) Optimal design of water distribution networks using whale optimization algorithm. *Urb Water J* 17(1):14–22
55. Mehzad N, Asghari K, Chamani MR (2020) Application of clustered-NA-ACO in three-objective optimization of water distribution networks. *Urb Water J* 17(1):1–13
56. Manolis A, Sidiropoulos E, Evangelides C (2021) Targeted path search algorithm for optimization of water distribution networks. *Urb Water J* 18(3):195–207
57. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2010) BGSA: binary gravitational search algorithm. *Nat Comput* 9:727–745
58. Tolson BA, Asadzadeh M, Maier HR, Zecchin A (2009) Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization. *Water Resour Res* 45(12):1–15, W12416. <https://doi.org/10.1029/2008WR007673>
59. Sadollah A, Yoo DG, Kim JH (2015) Improved mine blast algorithm for optimal cost design of water distribution systems. *Eng Optim* 47(12):1602–1618
60. Bala I (2021) A comprehensive learning gravitational search algorithm and its applications' PhD Thesis, The Northcap University (Formerly ITM University, Gurgaon)
61. Ji B, Xiaozheng Lu, Sun G, Zhang W, Li J, Xiao Y (2020) Bio-inspired feature selection: an improved binary particle swarm optimization approach. *IEEE Access* 8:85989–86002
62. Qian W, Li M (2018) Convergence analysis of standard particle swarm optimization algorithm and its improvement. *Soft Comput* 22:4047–4070
63. Cao Y, Zhang H, Li W, Zhou M, Zhang Y, Chaovalitwongse WA (2018) Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions. *IEEE Trans Evolut Comput* 23(4):718–731
64. Yadav A, Kumar N (2020) Artificial electric field algorithm for engineering optimization problems. *Expert Syst Appl* 149:113308
65. Nadimi-Shahraki MH, Taghian S, Mirjalili S (2021) An improved grey wolf optimizer for solving engineering problems. *Expert Syst Appl* 166:113917
66. Zhang Y, Jin Z (2022) Comprehensive learning Jaya algorithm for engineering design optimization problems. *J Intell Manuf* 33(5):1229–1253
67. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. *Knowl-Based Syst* 96:120–133
68. Mirjalili S (2015) The ant lion optimizer. *Adv Eng Softw* 83:80–98
69. Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18
70. Eliades, DG., Kyriakou M, Vrachimis S, and Polycarpou MM (2016). EPANET-MATLAB toolkit: An open-source software for interfacing EPANET with MATLAB. In *Proc. 14th international conference on computing and control for the water industry (ccwi)*, vol. 8.
71. Savic DA, Walters GA (1997) Genetic algorithms for least-cost design of water distribution networks. *J Water Resour Plan Manag* 123(2):67–77
72. Fujiwara O, Khang DB (1990) A two-phase decomposition method for optimal design of looped water distribution networks. *Water Resour Res* 26(4):539–549
73. Reza J, Martínez J (2006) Genetic algorithms for the design of looped irrigation water distribution networks. *Water Resour Res* 42(5):1–9, W05416. <https://doi.org/10.1029/2005WR004383>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.