

<https://doi.org/10.1038/s41545-024-00407-5>

Optimization of valve switch control for contamination detection in water distribution network

Jeng-Shyang Pan^{1,2}, Hao Shu³, Qingyong Yang⁴, Yu-Chung Huang⁵ & Shu-Chuan Chu^{1,4}

As the urban population increases, the consumption of water resources is also increasing. Safely and effectively supplying water to cities has become an issue that urgently needs to be addressed. The purpose of this research is to substantially reduce the number of contaminants in water distribution networks (WDNs) by using valve control, ensuring that the water infrastructure is not impacted by the adverse effects of wastewater. In addition, an improved parallel binary gannet algorithm (IPBGOA) is proposed and combined with this approach to solve the optimization problem of WDN contamination efficiently. The proposed method is compared with the gannet optimization algorithm (GOA), particle swarm optimization (PSO), differential evolution (DE), the grey wolf optimization (GWO), and the genetic algorithm (GA) on synthetic benchmark networks in simulation experiments. The evidence from the study indicates that the algorithm proposed in this paper is significantly more efficient and reliable than the comparison methods.

Water distribution networks (WDNs), a critical type of infrastructure, are responsible for conveying drinking water from purification sites to a wide array of residential and commercial establishments. It plays an integral role in the comprehensive urban water supply process¹. In the context of the Sustainable Development Goals (SDGs), this infrastructure is instrumental in achieving “Clean Water & Sanitation” (i.e., SDG 6), which focuses on ensuring the availability and sustainable management of water and sanitation for all. WDNs are vulnerable to external influences, especially water contamination, which is a pressing issue that not only affects public health but also hinders progress towards SDG 6. Water contamination has become a serious global issue, especially in China, where the economy has been growing rapidly. Water is a scarce resource; a mere 8% of the world’s freshwater is available to satisfy the needs of more than one-fifth of the Earth’s inhabitants. However, 80% of cities in China lack sewage treatment facilities, and 90% of the urban water supply is contaminated². This situation not only results in reduced gross domestic product (GDP) but also poses a serious threat to human health³. Addressing this challenge is crucial for the sustainable development of urban areas and aligns with “Sustainable cities and communities” (i.e., SDG 11), which aims to make cities and human settlements inclusive, safe, resilient, and sustainable. The efforts to reduce the impact of contamination on urban WDNs while maintaining stable water supply capabilities are therefore not only a technical endeavour but

also a step towards fulfilling the global commitment to sustainable development.

For contamination detection in WDNs, targeted high-quality monitoring information is necessary. A list provided in the SDGs outlines substances that are frequently found in water and are harmful to human health and productivity². Additionally, placing appropriate sensors within WDNs to detect contaminants is essential⁴. The optimization problem for WDNs has the objective of minimizing the impact of contaminants within restricted hydraulic and infrastructural limitations. Many researchers have proposed various methods to address such issues. For example, Braunstein et al. used integer linear programming (ILP) to solve the problem of node contamination⁵. Benk et al. used nonlinear programming (NLP) for optimal water allocation to address multiple pollutants in water networks⁶. However, with the increasing complexity of optimization problems, traditional optimization methods are gradually losing their advantages. Metaheuristic optimization algorithms, inspired by natural and biological laws, are designed to effectively address these complex issues⁷. In engineering applications, the popularity of metaheuristic optimization algorithms is increasing because of their simplicity, ease of implementation, absence of the need for gradient information, and capacity to avoid local optima⁸. During the past decade, the advancement and implementation of metaheuristic optimization strategies have been

¹School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing, China. ²Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan. ³School of Computer Science, Nanjing University of Information Science and Technology, Nanjing, China. ⁴College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China. ⁵Taiwan Smarter Water Co., LTD, Taiwan, China.

e-mail: scchu0803@gmail.com

extensively leveraged to solve a comprehensive range of complex problems across a spectrum of disciplines^{9–12}.

In recent years, many researchers have used metaheuristic algorithms such as ant colony optimization (ACO)¹³, the genetic algorithm (GA)¹⁴, particle swarm optimization (PSO)¹⁵, and the nondominated sorting genetic algorithm II (NSGA-II)¹⁶ to solve water network contamination optimization problems. For example, Rath et al. employed a GA to refine the positioning of water quality monitors, aiming to increase the probability of identifying contamination incidents in susceptible zones within a reasonable timeframe, thereby improving network security¹⁷. Hu et al. developed cooperative particle swarm optimization (CPSO) to deploy a limited number of sensors placed in a water network to minimize the average detection time for all contamination events. They demonstrated that the algorithm has ideal capabilities in solving practical problems¹⁸. Razavi et al. employed the NSGA-II to optimize community resilience, aiming to reduce the consumption of polluted water and operational measures, resulting in satisfactory optimization outcomes in three different situations¹⁹. Najafi et al. used ACO to select the optimal combination of hydrants and valves to isolate contaminants, thus reducing the possible risk to public health in the wake of a contamination incident²⁰. Additionally, Afshar, Ehsani and Masoumi et al. conducted multiple studies, employing metaheuristic algorithms for multiobjective optimization to mitigate the impact of contaminants^{21–23}. In the practical domain of engineering, the application of multiobjective optimization faces many complex problems, including nonlinearity and multidimensionality, which make it impractical to traverse all possible situations to find an accurate optimal solution²⁴. Therefore, selecting the appropriate optimization algorithm for evaluation and application to this problem is particularly important.

Since, in this study, valve switching is used to address the optimization problem for contamination in WDNs, the method involves binary issues. In general, many water network optimization problems involve finding solutions within a continuous search space^{25–27}. However, some heuristic algorithms lose their effectiveness in binary problems because they are suitable only for continuous problems. Considering this situation, an optimization algorithm for control valves is proposed in this paper to effectively solve binary optimization problems in WDNs. This optimization algorithm is based on the gannet optimization algorithm (GOA), as previous researchers have demonstrated the effectiveness of the binary GOA (BGOA) in searching for the optimal solution in a binary search space^{28,29}. Specifically, the key contributions presented in this study are outlined below:

- (1) The optimization algorithm based on the GOA is applied for the first time to engineering problems such as WDN contamination optimization.
- (2) An evaluation and comparison are conducted on the basis of specific WDN application scenarios, and the most suitable transfer function is selected to enable the BGOA, with the new transfer function, to be efficiently used for the optimization of WDN contamination.
- (3) An improved parallel BGOA, called the IPBGOA, is proposed as an optimization algorithm for controlling valves. Considering the characteristics of the WDN optimization problem, the method of obtaining the initial solution in the BGOA is modified, and crossover and parallelism rules are added to the update process; this gives the IPBGOA a better exploratory capability and prevents it from falling into local optima.

- (4) The IPBGOA with the new transfer function and updated rules is compared with the GOA, GA, DE, GWO, and PSO, and the experiments prove that the IPBGOA is more effective and stable in solving such WDN optimization problems.

The remainder of this paper is arranged as follows: Section “Results and discussion” presents the experiments and a comparison and analysis of the results. Section “Discussion” provides a summary and suggestions for future work. Section “Methods” presents the optimization model for the contamination issues discussed in this paper and introduces the improved methods, summarizes the overall process, and gives the pseudocode.

Results and discussion

Test scenarios

The test scenario selected for this work is the Hanoi network, which was initially proposed by Fujiwara et al. in 1990. This network water supply originates from a single reservoir at a height of 100 m; all the node ground elevations are set to 0. The overall length of the water network pipes is 39.4 km, and the Hazen–Williams roughness coefficient of each pipe is 130. The total nodal demand is 19,940 m³/h. The network comprises 34 conduits and 31 nodes of demand, with the entire configuration forming three closed circuits³⁰. A map of the Hanoi network and initial data for individual nodes and pipelines in the network are provided in the supplementary material, labeled Supplementary Fig. 1.

Selection of the new transfer function

In this study, the transfer function from the original BGOA is not used; instead, different transfer functions are selected on the basis of the designed algorithm for evaluation in the simulated scenario for the Hanoi network. According to the sensitivity analysis, the number of iterations is 20, and the population size is 100. Then, from the various transfer functions given, the one that best fits the optimization model of this paper is chosen. The implementation results for models with different transfer functions are shown in Table 1, and the convergence graphs of the 9 different transfer functions selected for this study are shown in the supplementary material, labeled Supplementary Fig. 2.

In Supplementary Fig. 2(a), f1 to f4 are S-shaped transfer functions, and f5 to f8 are V-shaped transfer functions. Supplementary Fig. 2(b) shows the linear transfer function. Function f1, in dark blue, converges in Supplementary Fig. 2(a) more quickly than almost all the other functions do. Additionally, in Table 1, the optimal value, mean, standard deviation, and worst value obtained in the 20 evaluations of the 9 transfer functions are displayed. By comparing these values, it can be more accurately verified that TF_1 combined with the IPBGOA optimization model is more effective in finding the optimal solution. After 20 iterations, the optimal value is 0.0128, the mean is 5.9351, the standard deviation is 9.3501, and the worst value is 28.1621.

Experimental setup

Before conducting the environmental simulation, the following parameters are set:

- (1) Sensor layout positions and quantities: The S-Place toolkit is used to optimize the placement of the contaminant sensors. The S-Place toolkit places the sensors by minimizing the impacts of all possible

Table 1 | Statistics of the objective function values achieved by the IPBGOA with different transfer functions

	TF_1	TF_2	TF_3	TF_4	TF_5	TF_6	TF_7	TF_8	TF_9
Best	0.0128	0.1373	1.3303	0.1442	0.3559	0.0348	0.1374	0.1011	0.0219
Mean	5.9351	39.9646	20.6815	23.6000	59.9588	31.3552	11.1214	36.6928	6.6700
Std	9.3501	62.0489	57.2057	63.9664	119.6436	55.0594	15.5797	34.9903	12.3920
Worst	28.1621	272.9346	261.8868	284.5941	332.9178	158.2370	37.2762	82.0780	35.8807

Bold values indicate the four data that perform best when different transfer functions are compared against each other.

- contaminations³¹. In this study, three sensors are set up in the Hanoi network, located at nodes 4, 27, and 30.
- (2) Method of detecting contamination: In addition to the initial setting, the IPBGOA, GOA, GA, DE, GWO, and PSO algorithms are also used for contamination detection.
 - (3) Constraint definition: When employing control valves to direct the contaminant flow towards the designated sensors, pressure constraints must be met. The pressure limits are set to $p_{\min} = 20$ m and $p_{\max} = 150$ m for all nodes.
 - (4) Contamination detection threshold: A sensor detection threshold for the contamination concentration is set to enable effective detection of contamination events. In this work, the concentration threshold is set to $T_{thr} = 7\%$.
 - (5) Simulation duration: The simulation period for this study is set to $T = 24$ h. The time step is set to half an hour, so the total number of time steps is 48.
 - (6) Pump station count and head boundary conditions: The simulation environment is the Hanoi network, so only one pump station (i.e., reservoir) is set up at node 1. Its head boundary conditions are set to an upper limit $Hs_{ub} = 20$ and a lower limit $Hs_{lb} = 6$. In this scenario, the pump station also serves as the only contamination source. With each detection method, the initial settings are used to calculate the impact of contamination in each simulation to avoid unrealistic results.

The choice of $p_{\min} = 20$ m as the lower limit for pressure is justified by its widespread adoption as a standard in firefighting water systems³². In terms of the upper pressure limit, the Hanoi network is considered a water supply system with large pipes that can withstand pressures up to 150 m.

The parameter T_{thr} defines the lowest concentration level of contaminants that can be detected by the sensor, where the maximum upper limit is 100%, representing the highest threshold of contaminant concentration that the sensor can perceive. Here, contaminants can be defined as water quality indicators that the users care about. Four water quality types are defined in this study: “None”, “Chemical”, “Age” and “Trace”. The “None” type represents normal water quality. The “Chemical” type indicates the presence of chemical components in the water, which may include minerals, organic substances or pollutants. The “Age” type is an indicator of water quality, including the freshness of the water or the efficiency of the water cycle, which may affect the water quality. The “Trace” type refers to the presence of some trace or minimal substances in the water, which may include metals or pesticide residues. Even at low concentrations in water, these substances can affect human health or ecosystems. In this study, “Trace” is chosen as the type of water quality in the water network, and the water quality simulation with the network is performed in EPANET2 software to detect the impacts of pollution. The defined threshold typically has no impact on the functionality of the algorithm; it is determined on the basis of the specific application scenario. When sensors specifically designed to identify a particular type of contaminant are deployed, the detection process does not require additional data to confirm the presence of the contaminant; it only requires an accurate threshold to detect whether the contaminant is present.

Comparisons and analysis

In this section, the benefits of detecting the contamination impact by controlling the valves under the specific arrangement of sensors are investigated. First, the node contamination impact and detection time under the initial settings (i.e., with all valves open) are calculated, and then the results are compared with those obtained via the IPBGOA optimization model proposed in this paper, confirming the effectiveness of the proposed scheme. In addition, the scheme is compared with the original GOA as well as several classic evolutionary and swarm algorithms to demonstrate that the IPBGOA has certain advantages in addressing this type of problem.

The simulation results are given in Figs. 1 and 2, which compare the following seven methods: (a) initial settings, (b) IPBGOA, (c) GOA, (d) GA, (e) DE, (f) GWO, and (g) PSO. In Fig. 1, the yellow and bright purple dots

represent the detected contaminated nodes, the black triangle on a light green background is the pump station, which is the only contamination source, the red dots represent the locations where sensors are placed, and the blue dots represent the nodes that have not been detected. Above each node is the detection time, and below it is the detected contamination impact. Figure 2 presents the contamination impact and detection time for each node under each method in a dual-Y-axis line graph, with time on the left Y-axis and contamination impact on the right Y-axis, allowing a more intuitive view of the data changes under different methods.

Figure 1a shows the contamination of the water network in the initial state. In the state without control valves, the impact of contamination at undetected nodes is very high, and even if some of the detected nodes are contaminated, the situation is very bad. Next, (b) shows the simulation results using the IPBGOA method proposed in this study in terms of the state of the control valve. The method proposed in this study significantly improves the detection of the impact of pollution in comparison with the initial state, except for certain individual nodes. In addition, several classical heuristic algorithms were included in this study for comparison in the simulations. Owing to the placement and number of sensor locations, some nodes in the WDN cannot be monitored. Additionally, there are nodes with data that remain consistent across all methods, so nodes 1, 4, 5, 6, 7, 9, 11, 12, 13, 22, 27, and 30, a total of twelve nodes, are excluded, and the remaining nodes are used for comparison. Supplementary Table 3 in the supplementary material details the possible pollution impact values of the nodes for each algorithm. The details of Fig. 1 are described as follows:

- (1) Figure 1c shows the simulation results under the original GOA. Excluding the nodes with the same pollution impact, better results are obtained for 16 nodes, including nodes 8, 10, and 14, with the IPBGOA than with the original GOA. Only nodes 15 and 18 have better results with the original GOA than with the IPBGOA. Therefore, the optimization rate of the IPBGOA is 88.89% greater than that of the original GOA.
- (2) Figure 1d shows the simulation results under the GA. Excluding the nodes with the same pollution impact, for the IPBGOA, a total of 11 nodes, 8, 10, 14, 15, 16, 19, 20, 21, 24, 25 and 31, have better outcomes than with the GA. For the GA, there are 6 nodes, 2, 17, 18, 21, 23 and 29, that have better outcomes than with the IPBGOA. Therefore, the IPBGOA has 64.7% greater optimization than the GA.
- (3) Figure 1e shows the simulation results under the DE. Only node 2 of DE is better than IPBGOA. Then IPBGOA has a high optimization rate of 94.44% with respect to DE.
- (4) Figure 1f shows the simulation results under GWO. For IPBGOA, there are total 9 nodes of 2, 8, 10, 14, 19, 20, 24, 25 and 28 which are better than GWO. For GWO, there are 6 nodes of 15, 17, 18, 21, 23 and 29 which are better than IPBGOA. The optimization rate of IPBGOA compared to GWO is only 60%.
- (5) Figure 1e shows the simulation results under PSO. Only 4 nodes of 8, 20, 21 and 28 are better than PSO under IPBGOA. On the contrary, 6 nodes are better than IPBGOA under PSO algorithm. This shows that compared to PSO, IPBGOA performs a little bit worse.
- (6) After all the methods have been compared, the nodes that have an advantage under each method (i.e., better than at least one of the methods) are labeled in bright purple in Fig. 1 and are labeled in darker font in Supplementary Table 3. In this study, the proposed IPBGOA has advantages for a total of 11 nodes, specifically 3, 8, 10, 16, 19, 20, 25, 26, 28, 31, and 32. In contrast, the GOA and DE both have only two nodes with advantages, and the GA has seven nodes with advantages. This shows that the improved GOA has better performance than the original GOA and the two classical evolutionary algorithms. Next, 10 nodes have an advantage under GWO, whereas 12 nodes have an advantage under PSO. This suggests that the IPBGOA does not have a large advantage over the classical swarm algorithms and may even be inferior to them. This is also consistent with the results in (4) and (5) above under a single comparison.

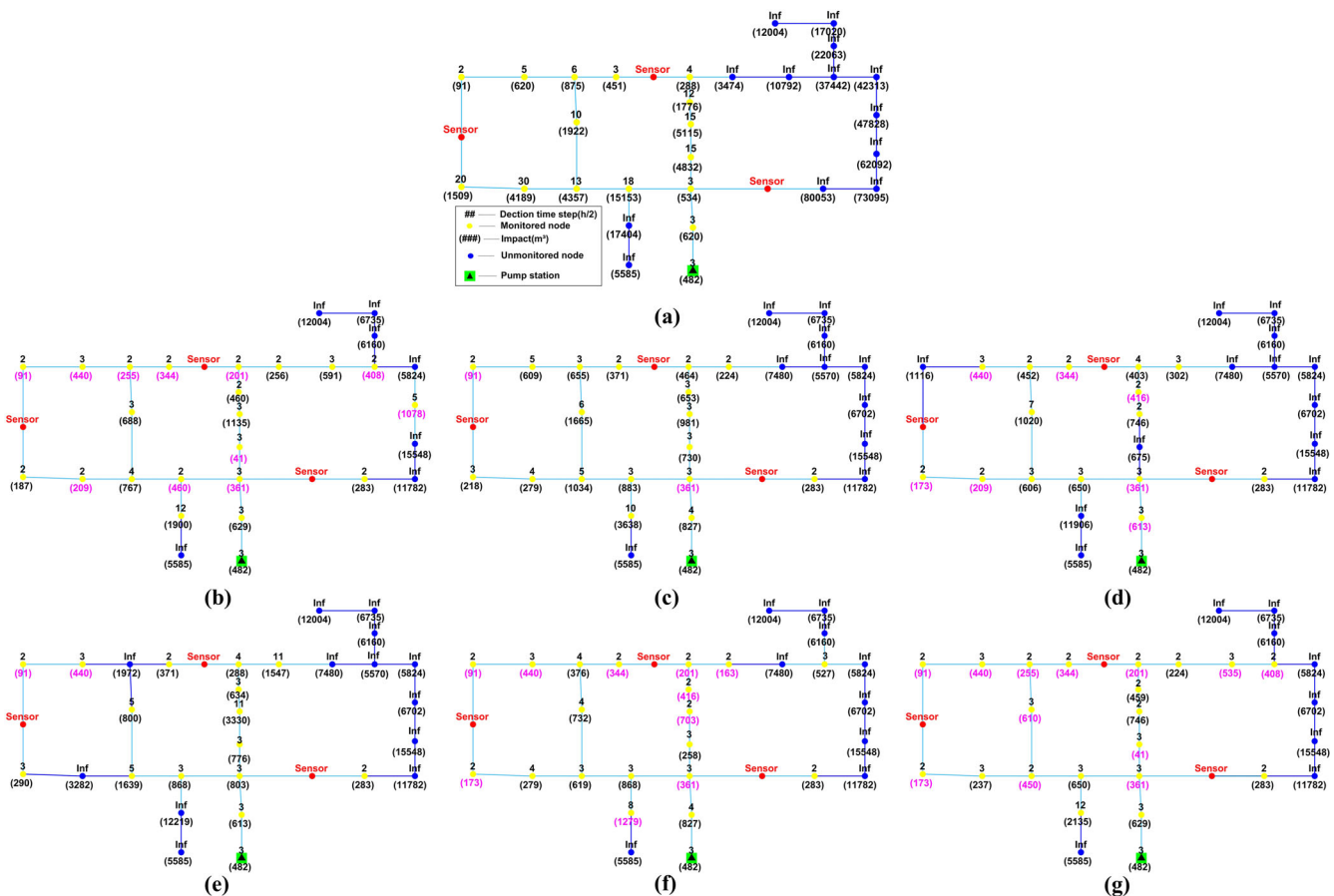


Fig. 1 | Simulation results for the Hanoi network using the initial settings and various algorithms. Calculating the contamination impact and detection time for all nodes that may be contaminated under (a) the initial setup; (b) IPBGOA; (c) GOA; (d) GA; (e) DE; (f) GWO; and (g) PSO.

In the study of contamination events, the detection time is typically also considered. This work calculates not only the impact of contamination at each potential contamination source node but also the corresponding detection time. Figure 2 shows the water network simulation data from Fig. 1 as a line graph. Figure 2a shows the initial state without controlling the valve, so the pollution data have large values, as seen on the right Y-axis. The data of the first 14 nodes in Fig. 2c–e are almost the same, and the data of all nodes in Fig. 2f, g are almost the same. However, the variation in the data in c–e is different from that in f and g. This finding indicates that different types of algorithms yield different results when dealing with such optimization problems, whereas algorithms within the same category tend to produce similar outcomes. This could be due to the characteristics of the algorithms and the specific application scenarios leading to these results. The main purpose of Fig. 2 is to illustrate the relationship between contamination impact and detection time. With the exception of the nodes for which no contamination is detected (i.e., where the detection time step is 48), the variation in contamination impact for the nodes is almost directly proportional to the variation in detection time. This finding indicates that the shorter the detection time is, the less contamination impact is detected, confirming that controlling the valves to ensure that contaminants reach the sensors more quickly can effectively reduce the impact of contaminants on the network.

After each node in the WDN is analyzed, an overall analysis is conducted. Table 2 defines and provides the values of specific indicators:

(1) *Cov*: Monitoring coverage, i.e., the percentage of total nodes that can be detected as potential contamination sources under the simulation scenario with set the sensor locations and quantities. (i.e., the proportion of yellow bright purple dots in Fig. 1).

- (2) \bar{t} : The mean value of the detection times for all nodes in each simulation.
- (3) \bar{I} : The mean value of the impact of contamination across the entire scenario in each simulation.
- (4) \bar{H}_s : The calculation of the pump head based on the situation at each point for each method. The pump head value at one point should be the most suitable for that point, that is, the lowest value. After the whole simulation, each point has the most suitable pump head value. \bar{H}_s represents the average of the most appropriate pump head values at all points under each method, which is an approximate range of settings used to ensure the proper operation of the water network simulation under each method.

According to the table, when the IPBGOA is used, the coverage rate of potential contamination source nodes that the sensors can detect is higher than those of other methods. The detection time is significantly reduced, and the impact of contamination on the WDN is also decreased; the IPBGOA thereby demonstrates better performance than the other methods. These benefits are achieved by controlling the valves, which leads to an increase in WDN pressure. Therefore, pressure constraints cannot only be set for each node; they also need to reduce the pump head as much as possible to alleviate the water pressure throughout the network and reduce energy consumption. Table 2 shows the pump head that is suitable for each detection method.

As shown in Fig. 1, other than in the initial setup, some nodes that are not detected produce contamination impacts several times greater than those of the nodes that are detected. Although these nodes were removed in the previous single-node comparative analysis, there may be instability when performing the overall analysis, which can have an impact on the

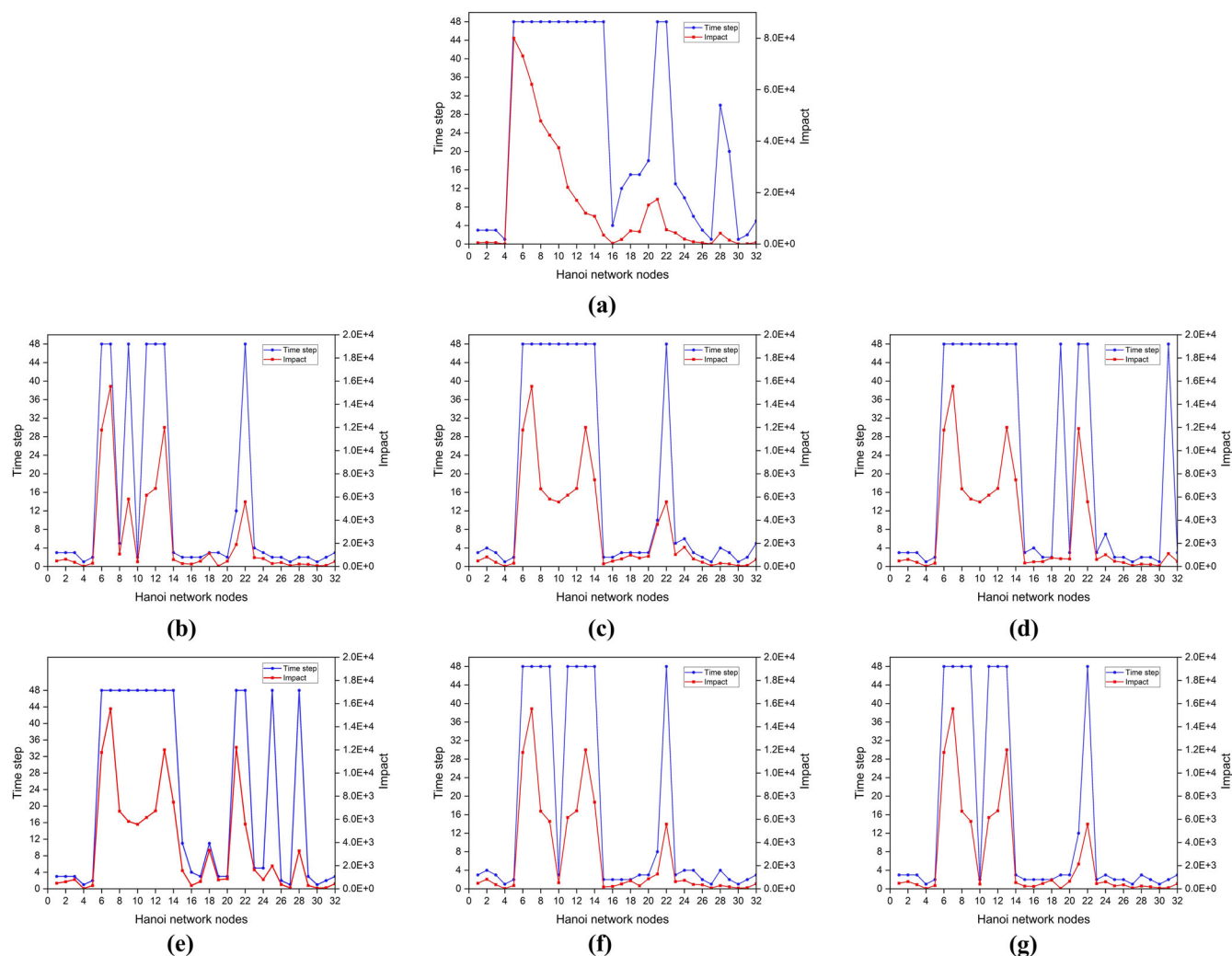


Fig. 2 | Line plots of the contamination impact and detection time of each node under different simulation methods. **a** The initial setup; **(b)** IPBGOA; **(c)** GOA; **(d)** GA; **(e)** DE; **(f)** GWO; and **(g)** PSO.

Table 2 | Metric values resulting from the simulation scenarios

	Initial	IPBGOA	GOA	GA	DE	GWO	PSO
Cov (%)	59.37	78.12	68.75	56.25	59.37	71.87	75
\bar{t} (h)	12.3	6.3	8.6	10.5	10.8	7.75	7
\bar{I} (m ³)	14,816.73	2345.657	3062.364	3273.245	3571.942	2731.242	2507.868
\overline{Hs} (m)		98	109	93	73	79	85

Bold values indicate the values with the highest coverage, the shortest detection time and the least contamination impact for the different simulation methods.

overall metrics of the simulated scenarios. Therefore, six additional simulation experiments are conducted to analyze the stability of the overall metrics under each method. The results of these six experiments are shown in the box plot in Fig. 3. In the figure, the light blue portion is the IPBGOA, the dark blue portion is the original GOA, the light green portion is the GA, the dark green portion is the DE, the pink portion is GWO, and the red portion is PSO. The black horizontal line is the median line of contamination impact under each method, and the diamonds are outliers. An outlier is defined as a value that exceeds the interquartile spacing by a factor of 1.5, so the locations of the diamonds in each method in the figure generally represent the impacts of contamination at nodes that are not detected. The main purpose of constructing a box plot is to clearly observe the parts that affect the stability of the method, and its greatest advantage is that it is unaffected by outliers and accurately depicts the discrete distribution of the

data. The outliers of the six simulation experiments under the IPBGOA method are clearly labeled, and the data of the other nodes are almost all within a certain range, which indicates that the IPBGOA method has obvious stability and better experimental results. The data for the GA and DE methods are nearly identical under the six simulations, which suggests that both methods are more stable. In addition, although there are only one or two outliers, the range of data above the median line is wider, and many undetected nodes are included in the box plot, resulting in a greater overall contamination impact. This suggests that both the GA and DE have poorer results than the IPBGOA does in multiple experiments. The box plots for the GOA and GWO methods show large variations between the six experiments, indicating that these methods are less stable. In addition, they have almost the same data distribution as the GA and DE do, indicating that they produce inferior results to those of the IPBGOA. Finally, the PSO has

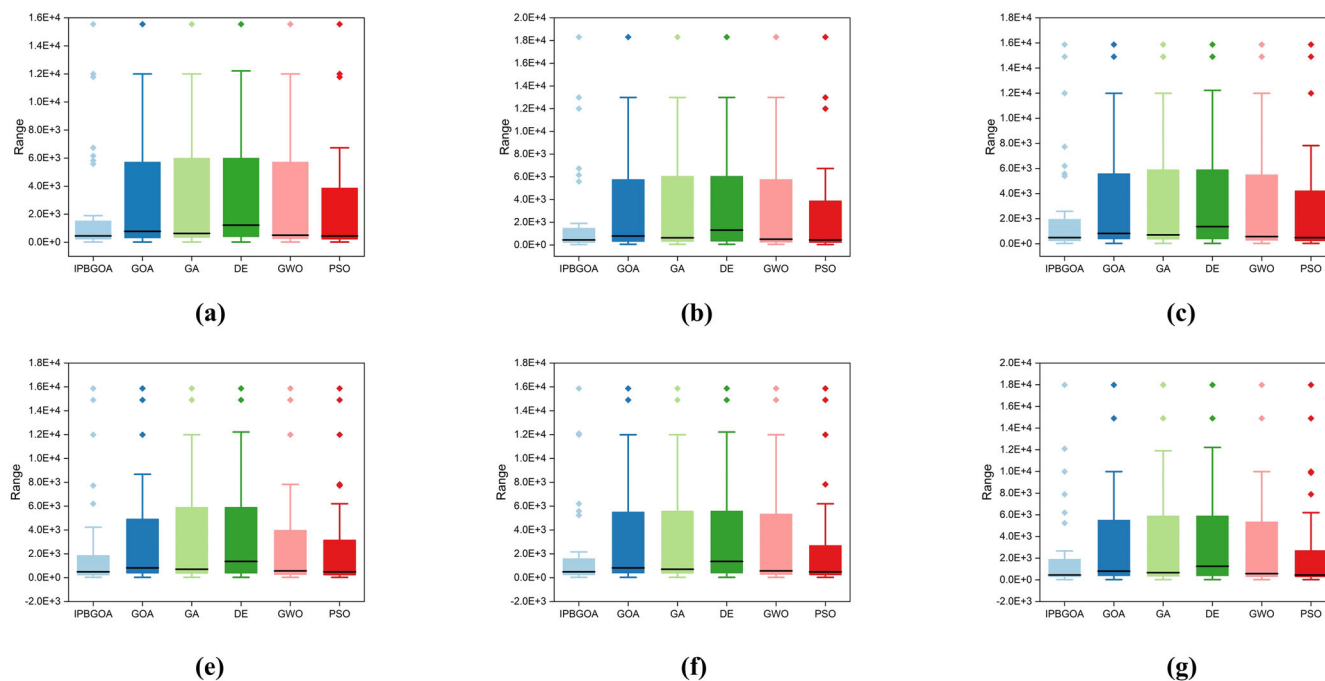


Fig. 3 | Box plots of contamination impact values. Comparison of overall contamination indicators between IPBGOA and GOA, GA, DE, GWO, and PSO methods under (a) the first simulation experiment; (b) the second simulation

experiment; (c) the third simulation experiment; (e) the fourth simulation experiment; (f) the fifth simulation experiment and (g) the sixth simulation experiment.

considerable data fluctuations in the six experiments, and its stability is poor. The data distribution of the PSO is also not as good as that of the IPBGOA, which suggests that it is inferior to the IPBGOA in terms of overall metrics.

Comparison with other studies

Leakage in water networks was addressed by Jowitt et al.³³ by controlling valves. Vrachimis et al.³⁴ demonstrated that control valves can effectively reduce the impact of pollution by comparing passive and active contamination detection, and Yan et al.³⁵ reduced the pollution impact in a water network by controlling valves and hydrants. The method of controlling the valves used in this study is consistent with the methods used by these previous researchers, and each node in the water network is monitored by the sensors as well as possible via this method. In addition, the appropriate pump head is calculated for each node to prevent pressure loading in the water network, which is different from previous studies.

There are also other ways of detecting the effects of pollution that may be present in a water network. The main way of detecting water quality, used by M. Aral et al.⁴, HU et al.¹⁸ and Ahmadabadi et al.²⁶, is to find the best sensor location so that the node status of the water network are detected as well as possible. A shortcoming of this approach is that several randomized experiments may be needed to determine the optimal locations of the sensors in a water network, which can lead to long detection times and continued expansion of contamination. In addition, the effectiveness of this approach depends on the number of sensors, and too many or too few sensors can affect detection. If there are too few sensors, they may be unable to be fully utilized, whereas having too many sensors may increase installation and maintenance costs, thus exceeding the budget. However, in this study, the number of sensors can be determined on the basis of the actual cost, and then the sensors can be placed at the correct locations in the water network via the S-Place toolkit, which addresses the problem of increasing cost owing to the uncertainty in the number of sensors. Since the locations and number of sensors are set directly, the node water quality is detected by controlling the valves. This allows detection to be carried out directly without the need to spend time determining the sensor positions prior to inspection, as was previously the case. In addition, M. Aral et al. proposed that the number of sensors determines the reliability of the system and that

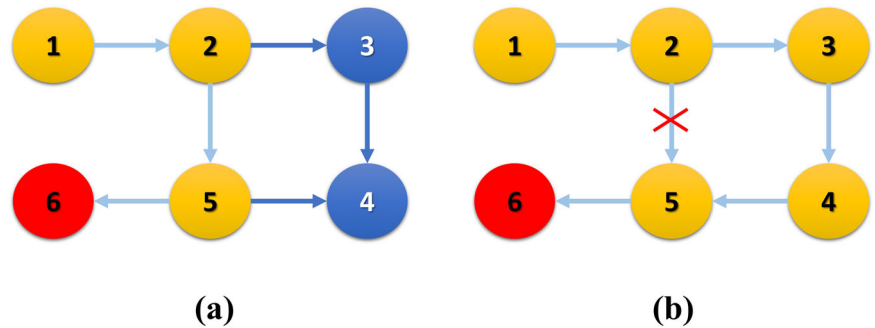
reliability is determined by the sensor's usefulness in the water network, with more coverage indicating greater reliability. Reliability is dependent on the number of sensors, and sensors will inevitably fail; reliability will be drastically reduced if a failure occurs. The method used in this paper, on the other hand, has the relative advantage that even if the sensors fail, more coverage can be ensured with the control valves, as was verified in the experiments of Vrachimis et al.³⁴.

Products are becoming ever more intelligent, with many industries no longer relying on staff to manually operate their systems. Contemporary urban water systems no longer require staff to make adjustments and rather tend towards intelligent control and automatic supply. The research of this paper aligns well with contemporary science and technology; if a water supply company can use an intelligent control valve system that is appropriate for the detection method in this paper, it may be able to reduce the number of valves to significantly improve the coverage, increase the detection efficiency and reduce the cost of detection.

Discussion

This work aims to solve the contamination optimization issue for modern WDNs. To achieve this, sensors and valves are set up in the network to monitor unknown contamination source nodes, preventing the occurrence of serious events in which undetected contamination spreads. An improved parallel binary gannet optimization algorithm is proposed and tested in specific application scenarios, and the objectives are achieved. Additionally, compared with the original gannet optimization algorithm, the genetic algorithm, differential evolution, and grey wolf optimization, the proposed algorithm demonstrates superior performance in minimizing the objective function. However, compared with particle swarm optimization, there is no advantage in terms of the comparison of individual nodes. Furthermore, the line charts of contamination impact and detection time also confirm that controlling the valves so that contaminants can be detected by sensors earlier can effectively reduce the impact of contamination. The IPBGOA shows a clear advantage over all other algorithms in terms of monitoring coverage, overall metrics, and data stability. Despite the advantages of the IPBGOA as demonstrated in this study, several additional areas could be studied in depth. First, from the perspective of optimization algorithms, this algorithm

Fig. 4 | Six-node network. **a** Sensor detection range when all valves are open; and **(b)** sensor detection range when one of the valves is closed.



demonstrates better performance than the GA, DE, and the GOA. However, in terms of the comparison of individual nodes, it does not have a large advantage over swarm intelligence algorithms. Further improvements to the algorithm could be made, and it could be compared with multiple swarm intelligence algorithms with the goal of outperforming them³⁶. Second, regarding application scenarios, the simulation scenario in this paper is the Hanoi network, which is a small network. The method could also be applied to large networks with multiple pump stations^{37,38}, and new optimization strategies could be designed according to the size of the network.

Methods

Problem formulation

The water distribution network includes multiple components, such as pipes, pumps, and other hydraulic equipment, as well as reservoirs. From the fundamental examination of daily water consumption to the final phase of construction, each stage requires a deep level of professional knowledge^{39,40}. In this study, the focus is solely on network optimization. The aim is to minimize the risk of pollution and the energy consumption of reservoirs. Assuming that the layout and connectivity of the pipes, the node demands, and the minimum head requirements are known, the network needs to be optimized under the following conditions⁴¹:

- (1) Conservation of mass: The inflow and outflow at each node in the WDN must be balanced.
- (2) Conservation of energy: The cumulative head loss along a path must equal the difference between the head at the starting node and the head at the ending node.
- (3) Pressure constraints: For a given set of demands, the pressure at each node must be maintained within the specified pressure range.
- (4) The head loss of each pipe should follow the basic principles of fluid mechanics.

These conditions can be mathematically expressed with the following formulas^{41,42}:

$$Q_i^{\text{ext}} + \sum_{j=1}^{Nn} Q_{j,i}^{\text{in}} = Q_i^n + \sum_{k=1}^{Nn} Q_{i,k}^{\text{out}}, i = 1, \dots, Nn \quad (1)$$

$$\begin{cases} \sum_{i \in \mathbf{P}} \Delta H_i = H^s - H^e, & \mathbf{P} \in \mathbf{SP} \\ \Delta H_i = H_i^s - H_i^e, & i = 1, \dots, Np \end{cases} \quad (2)$$

$$H_{i,\min} \leq H_i \leq H_{i,\max}, i = 1, \dots, Nn \quad (3)$$

Equation (1) represents the principle of mass conservation, where Nn indicates the number of nodes in the network. Q_i^{ext} is the velocity of external inflow into the i th node, $Q_{j,i}^{\text{in}}$ is the velocity of inflow from node j to i , Q_i^n is the velocity of water consumption at the i th node, and $Q_{i,k}^{\text{out}}$ is the velocity of outflow from node i to k . Equation (2) represents the principle of conservation of energy, where ΔH_i is the head loss of the i th pipe. The head loss in individual pipes can be estimated with the aid of the Hazen-Williams equation. Np represents the total number of pipes, \mathbf{P} is a path consisting of a

series of consecutive pipes in the network, and \mathbf{SP} is the completion set of \mathbf{P} . H^s and H^e are the head pressures at the starting and ending nodes of \mathbf{P} , respectively. H_i^s and H_i^e are the head pressures at the starting and ending nodes of the i th pipe, respectively. H_i is the actual water pressure at the i th node, and $H_{i,\min}$ and $H_{i,\max}$ are the minimum and maximum pressures at the i th node, respectively.

Among these three conditions, the simulation software EPANET2⁴³ can address the constraints of mass and energy. The pressure constraint is processed through the optimization algorithm and encoding scheme presented in this paper. This is not a mathematical programming model; instead, it represents constraints as seen through the lens of the simulation software EPANET2.

Active contamination detection scheme

The contamination detection method used in this study is the active contamination detection (ACD) scheme proposed by Vrachimis et al.³⁴, which involves closing and opening valves to drive the flow in specific parts of the network. With a certain number of sensors in place, it becomes possible to monitor water quality at locations that were previously unobservable. The concept of this detection method is simply illustrated through the six-node network in Fig. 4. The red node at position 6 represents the sensor. The yellow nodes denote areas under sensor surveillance, whereas the dark blue nodes signify regions outside the sensor's coverage; the arrows depict the flow direction. Figure 4a shows the case of monitoring a contaminated network with the sensor located at node 6. According to the direction of flow, nodes 1, 2, and 5 can be monitored, but nodes 3 and 4 cannot be monitored. This indicates that the sensor's location does not allow it to cover the entire network. On the other hand, as shown in Fig. 4b, by controlling the pipeline valves to close the connection between nodes 2 and 5, the flow is directed along a continuous path that includes all system nodes. This ensures that the contamination of any node will be detected by the sensor. If one of the nodes is contaminated, the contaminant will definitely reach the sensor at some time. Because this method may increase the range of contamination, a better indicator for measuring the impact of contamination is the amount of contaminated water consumed (where the consumption of contaminated water is equivalent to the extent of contamination). If the alternative method for confirming the presence of contaminants in Fig. 4a is manual sampling at nodes 3 and 4, which could be very time-consuming, then the impact of contamination at nodes 3 and 4 will become very large. However, if control valves are used, it will be possible for the water quality at all nodes to be detected within just a few hours, significantly reducing the impact of contamination on the entire network. The simulation data for this six-node network are given in the supplementary material, labeled Supplementary Fig. 4 and Supplementary Table 4, where Supplementary Fig. 4a corresponds to Fig. 4a and Supplementary Fig. 4b corresponds to Fig. 4b.

Gannet optimization algorithm

The foundational idea behind the GOA was introduced by Pan et al.⁴⁴. This algorithm was inspired by the foraging movements of gannets. In the GOA, there are two stages that simulate the predatory behavior of gannets: exploration and exploitation. These stages incorporate a repertoire of

foraging tactics, including the U-shaped diving pattern, the V-shaped diving pattern, sudden rotation, and random wandering.

The exploration stage is the hunting stage. When the gannets hunt, they observe the distance from their prey from the surface to determine the diving pattern (i.e., foraging movement) to use. There are two types of diving patterns: the first is a U-shaped pattern for diving into deeper water to hunt, whereas the second is a V-shaped pattern used to hunt for prey in shallow waters. Equation (4) is the U-shaped coefficient equation, and Eq. (5) is the V-shaped coefficient equation.

$$a = 2 * \cos(2 * \pi * r_1) * t \quad (4)$$

$$b = 2 * V(2 * \pi * r_2) * t \quad (5)$$

$$t = 1 - (T_{\text{cur_iter}} / T_{\text{max_iter}}) \quad (6)$$

$$V(x) = \begin{cases} -(1/\pi) * x + 1, & x \in (0, \pi) \\ -(1/\pi) * x - 1, & x \in (\pi, 2\pi) \end{cases} \quad (7)$$

where r_1 and r_2 are random values uniformly distributed from 0 to 1, $T_{\text{cur_iter}}$ represents the current iteration count, and $T_{\text{max_iter}}$ represents the number of iterations.

After the U-shaped and V-shaped coefficients are calculated, the next step uses these two diving patterns to update the position. Gannets have an equal likelihood of choosing between the two feeding strategies, so a random value uniformly distributed from 0 to 1 is used to randomly decide between the two diving patterns. Equation (8) is the formula used to update the positions of the gannets.

$$MX_i(t+1) = \begin{cases} X_i(t) + u1 + u2, & q \geq 0.5 \text{ (a)} \\ X_i(t) + v1 + v2, & q < 0.5 \text{ (b)} \end{cases} \quad (8)$$

$$u1 \in (-a, a), v1 \in (-b, b) \quad (9)$$

$$u2 = A * (X_i(t) - X_r(t)) \quad (10)$$

$$v2 = B * (X_i(t) - X_{\text{avg}}(t)) \quad (11)$$

$$A = (2 * r_3 - 1) * a \quad (12)$$

$$B = (2 * r_4 - 1) * b \quad (13)$$

$$X_{\text{avg}}(t) = (1/N) * \sum_{i=1}^N X_i(t) \quad (14)$$

where $u1$ is a random value uniformly distributed from $-a$ to a , $v1$ is a random value uniformly distributed from $-b$ to b , and r_3 and r_4 are random values uniformly distributed from 0 to 1. In the context of the current iterative population, $X_i(t)$ denotes the i th individual, $X_r(t)$ denotes a randomly selected individual from the population, and $X_{\text{avg}}(t)$ denotes the average position. $X_{\text{avg}}(t)$ can be calculated via Eq. (14), where N is the population size.

Next is the Exploitation stage. After the gannets enter the water according to the aforementioned methods, the fish often perform sudden turns in the water to evade the pursuit of the gannets, so the gannets also expend a great deal of energy to catch the fleeing fish. Here, the capture capability is defined by Eq. (15).

$$\text{Capturability} = 1/(R * t2) \quad (15)$$

Table 3 | S-shaped, V-shaped and linear transfer functions

Name	Equation	Type
TF_1	$TF_1(X) = 1/(1 + \exp(-2X))$	S-shaped
TF_2	$TF_2(X) = 1/(1 + \exp(-X))$	S-shaped
TF_3	$TF_3(X) = 1/(1 + \exp(-X/2))$	S-shaped
TF_4	$TF_4(X) = 1/(1 + \exp(-X/3))$	S-shaped
TF_5	$TF_5(X) = \tanh(X/2) $	V-shaped
TF_6	$TF_6(X) = \tanh(X) $	V-shaped
TF_7	$TF_7(X) = X/\sqrt{X^2 + 1} $	V-shaped
TF_8	$TF_8(X) = (2/\pi) \arctan((\pi/2)X) $	V-shaped
TF_9	$TF_9(X) = (X - R_{\min})/(R_{\max} - R_{\min})$	Linear

$$t2 = 1 + (T_{\text{cur_iter}} / T_{\text{max_iter}}), \quad R = (M * vel^2)/L, \quad L = 0.2 + (2 - 0.2) * r_5 \quad (16)$$

where r_5 is a random value uniformly distributed from 0 to 1. $M = 2.5 \text{ Kg}$ represents the weight of the gannet, and $vel = 1.5 \text{ m/s}$ represents the pre-dation speed when water resistance is ignored.

When the gannet has sufficient energy, that is, when the capture capability is high, the gannet will catch the fish. As time progresses, the energy level of the gannet decreases, leading to an eventual inability to execute the capture manoeuvre. The capture equation is shown in Eq. (17).

$$MX_i(t+1) = \begin{cases} t * \text{delta} * (X_i(t) - X_{\text{Best}}(t)) + X_i(t), & \text{Capturability} \geq c \text{ (a)} \\ X_{\text{Best}}(t) - (X_i(t) - X_{\text{Best}}(t)) * P * t, & \text{Capturability} < c \text{ (b)} \end{cases} \quad (17)$$

$$\text{delta} = \text{Capturability} * |X_i(t) - X_{\text{Best}}(t)| \quad (18)$$

$$P = \text{Levy}(\text{Dim}) \quad (19)$$

where the constant c has been established to be 0.2 through extensive experimental validation. The individual exhibiting the highest performance within the current population iteration is denoted by $X_{\text{Best}}(t)$. The Levy flight, which is a stochastic process characterized by long-range jumps, is derived from Eq. (20).

$$\text{Levy} = (1/100) * ((\sigma * \varepsilon)/|v|^{(1/\psi)}) \quad (20)$$

$$\sigma = \left(\frac{\Gamma(1 + \psi) * \sin(\pi\psi/2)}{\Gamma((1 + \psi)/2) * \alpha * 2^{((\psi-1)/2)}} \right)^{(1/\psi)} \quad (21)$$

where ε and σ are random values uniformly distributed from 0 to 1 and where $\psi = 1.5$ is a constant determined by the researchers in previous studies.

Binary Gannet Optimization Algorithm

The GOA was originally designed to address complex engineering optimization problems. However, the GOA cannot effectively solve binary optimization problems, so a binary version of the GOA needs to be designed. Given the inherent differences between optimization in binary and continuous search spaces, a straightforward approach to adapting a continuous optimization algorithm for binary use without changing its fundamental structure is to use a transfer function^{7,45,46}. There are various types of transfer functions, and this paper presents several S-shaped, V-shaped and linear transfer functions, as shown in Table 3.

The transfer function is used to insert a solution from a general continuous search space into the function and return a number within the range

of [0 or 1], as typically represented by Eq. (22).

$$P(t) = \begin{cases} 1 & \text{if } TF(x) > \text{rand} \\ 0 & \text{else} \end{cases} \quad (22)$$

The main process of the BGOA involves substituting MX , the position of which has been updated through two stages, into TF to obtain $TF(x)$. This value is then compared with a random value rand , which is uniformly distributed from 0 to 1. $P(t)$ refers to the updated binary value.

Decision variables

Since the aim of this study is to optimize the control of valve opening and closing, which is essentially a binary problem represented by values of 0 and 1 (where an open valve is denoted as 1 and a closed valve as 0), the decision variables are binary integer sequences. Additionally, the variation of the pump head in the WDN is of interest under these circumstances. The aim is to minimize the impact of contaminants while also reducing the energy consumption of the pump stations. Therefore, the decision variables are not only binary integers; they also need to include variables representing the heads of the pump stations. The decision variables are described below:

$$X = \begin{bmatrix} x_1^1, x_1^2, \dots, x_1^{Np}, x_h^{Hs} \\ x_2^1, x_2^2, \dots, x_2^{Np}, x_h^{Hs} \\ \vdots \\ x_{Nt}^1, x_{Nt}^2, \dots, x_{Nt}^{Np}, x_h^{Hs} \end{bmatrix}_{Nt \times (Np+h)}$$

$$s.t. \quad Hs_{lb} < x_h^{Hs} < Hs_{ub}$$

where X represents the matrix of decision variables, Nt represents the simulated time, Np represents the total number of pipes, and h represents the number of pump stations. x_h^{Hs} represents the head of the h th pump station. The maximum and minimum limits for the pump head are designated Hs_{ub} and Hs_{lb} , respectively.

Constraints

The contamination detection scheme in this study requires the closure of a certain number of valves, and the closure of valves has a certain relationship with the pressure in the pipes. If too many valves are closed, the water flowing through the other pipes with open valves will be excessive, which will lead to excessive water pressure in some of these pipes. In this case, it is necessary to impose constraints on the pressure of the pipes. If the pressure exceeds the constraint range, there will be a penalty for this pressure deviation. The penalty function P is shown in Eq. (23):

$$P = \max_{i \in NP} (p_{\min} - p_i(t), 0) + \max_{i \in NP} (p_i(t) - p_{\max}, 0) \quad (23)$$

where $p_i(t)$ is the pressure in pipe i at time t and the pressure limits within the water distribution network are defined by p_{\min} and p_{\max} . If the nodal pressures meet the pressure limits, then the penalty function P takes a value of 0; otherwise, a deviation penalty is calculated.

Fitness function

In this work, node water quality is determined by controlling the valves in the WDN under a fixed sensor setup. The fitness function is designed to minimize the amount of contamination in the WDN. Minimizing the heads of the pumps in the pumping stations is also considered in reducing the water pressure and energy consumption.

A contamination impact function f_{con_imp} is then designed to optimize the primary objective. Additionally, the impact of contamination is represented as the amount of polluted water that has been consumed, which is the actual water demand. The amount of impact at node i at time step t is

represented by Eq. (24):

$$Imp_i(t) = f_{con_imp}(k_i^d, w_j(t); Tr(i), T_{thr}) = \begin{cases} \sum_{t=1}^{k_i^d} w_j(t), & k_i^d < T \\ \sum_{t=1}^T w_j(t), & k_i^d \geq T \end{cases} \quad (24)$$

$$k_i^d = \min(Tr(i) > T_{thr})$$

where k_i^d is the contamination detection time at node i , $Tr(i)$ is the water quality condition at node i , T_{thr} is the contamination threshold, $w_j(t)$ is the water demand of pipe j at time step t , and T is the maximum detection time.

Minimizing the pump heads of the pumping stations is taken as a secondary objective. The pump head is already defined in the decision variables as x_h^{Hs} , where $Hs_{lb} < x_h^{Hs} < Hs_{ub}$.

Since this work involves multiobjective optimization, the fitness function to be designed is generally a nonlinear function, which complicates computation. The multiobjective optimization process can be simplified into a single-objective linear function by linearly combining $Imp_i(t)$ with x_h^{Hs} . To simplify, these two quantities need to be normalized and assigned weights. $Imp_{tot}(t)$ is set to represent the total contamination impact at time step t , which is the total water demand, and this is used to normalize $Imp_i(t)$. Hs_{ub} is used to normalize $x_h^{Hs}(t)$. Then, α is used as the weight coefficient for the two. The resulting linear combination is shown in Eq. (25):

$$Obj(t) = \alpha(Imp_i(t)/Imp_{tot}(t)) + (1 - \alpha)(x_h^{Hs}(t)/Hs_{ub}) \quad (25)$$

Although transforming multiobjective optimization problems into single-objective optimization problems through standardization and weighting reduces the computational burden, it may inadvertently result in the exclusion of certain solutions. Such solutions, which could be optimal or nearly so, typically appear on the Pareto front. The set of these solutions largely depends on the construction scale of the WDNs and the feasibility of the optimization problem. In the future, an effective Pareto front can be created specifically on the basis of the particular circumstances of the network.

In addition to the necessary optimization objectives, there are also constraints to consider. Here, the pressure constraint penalty function P from Section “Constraints” needs to be utilized, and the expression for the fitness function F is presented in Eq. (26):

$$F(X) = P + \alpha(Imp_i(t)/Imp_{tot}(t)) + (1 - \alpha)(x_h^{Hs}(t)/Hs_{ub}) \quad (26)$$

$$s.t. \quad \begin{cases} P = \sum_{t=1}^T \sum_{j=1}^{NP} \xi(t, j) + \xi(t, j) \cdot ((p_{\min} - p_j(t)) + (p_j(t) - p_{\max})) \\ \xi(t, j) = \begin{cases} 1, & \text{if } p_{\min} > p_j(t) || p_{\max} < p_j(t) \\ 0, & \text{otherwise.} \end{cases} \end{cases}$$

The final fitness function has two components: the objective function part Obj and the constraint penalty part P . Both optimization objectives in the objective function have been standardized, so their values are within the range of 0 to 1. Additionally, a weight $\alpha \in [0, 1]$ is added to linearly combine the two, so the objective function part is always less than or equal to 1. The pressure constraint penalty part is used to determine the feasibility of the solution. If the solution is feasible, then the penalty part will be equal to 0, and when these two parts are added together, the fitness value will always be less than or equal to 1. If the solution is not feasible, it indicates that at least one pipeline's pressure does not meet the pressure limit, and the penalty part will be greater than 1, which will always make the fitness value greater than 1.

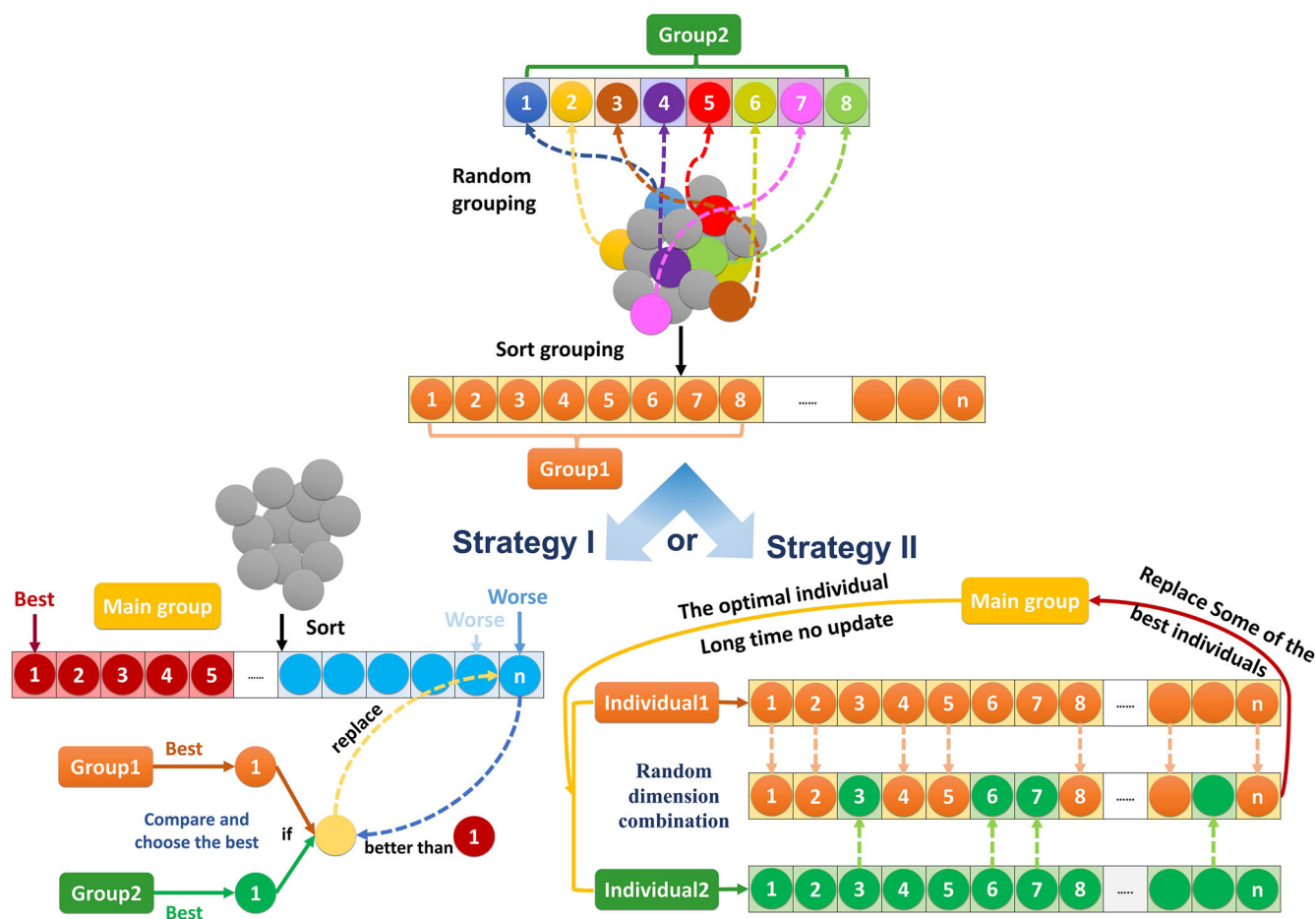


Fig. 5 | Parallel communication strategy. The main population in the algorithm is divided into two subpopulations with two methods, then Strategy I or Strategy II is used for parallel communication between groups.

Random flipping strategy

It is widely acknowledged that before an algorithm is implemented, it requires initialization. Typically, initialization is performed randomly, but this could lead to some initial solutions being infeasible in practical application scenarios. Consequently, the algorithm might have to search for a feasible solution within a larger search space, which will increase resource consumption. Moreover, it could be difficult for the algorithm to converge to an optimal or effective solution. To address this issue, an improvement is proposed to preprocess the random initialization. The decision variable X defined in Section “Decision variables” is randomly initialized before the algorithm is implemented. After initialization, X is input into the fitness function for a feasibility check. If $F(X)$ is greater than 1, the binary integer sequence within X is randomly flipped, turning it into a new set of solutions, and then it is re-entered into the fitness function for judgement. If it is less than or equal to 1, this indicates that the initial solution is feasible; otherwise, flipping continues until a feasible solution is obtained. A schematic of the proposed random flip strategy can be found in the supplementary material, labeled Supplementary Fig. 5.

Parallel strategy

In this study, a parallel strategy is employed in the algorithm. Parallel strategies are commonly used for optimizing algorithms. By employing parallel processing techniques, better solutions can be found, and the convergence process can be expedited. The parallel strategy is executed specifically by partitioning the search space into distinct groups, each of which functions autonomously to identify the optimal solution. When specific criteria are met, intergroup communication is initiated to facilitate the sharing of information among the groups⁴⁷.

The parallel strategy used in the algorithm of this study is shown in Fig. 5. As shown in the upper part of the figure, in the initialization phase, the main population in the algorithm is divided into two subpopulations, which indicates that there are two strategies for population grouping. The first grouping strategy is sort grouping. In this strategy, the fitness value of each individual in the main population is calculated; then, the fitness values are compared and ranked to determine the global optimal individual. The first subpopulation (i.e., Group 1) retains the global optimal individual and also retains the better individuals in the higher ranks of the main population since new superior individuals are more likely to be found in the higher-ranked population. The second grouping strategy is random grouping. The second subpopulation (i.e., Group 2) directly selects individuals at random from the main population to avoid falling into local optima.

The next step is to perform intergroup communication. Adding communication strategies to the algorithm has a great impact on it, and a good strategy can lead to faster convergence of the algorithm. Two communication strategies are proposed, as shown in the lower part of Fig. 5. Strategy I is the primary communication strategy. In each iteration, according to the given algorithm, individuals in the main population are compared and sorted to update the population. In addition, the individuals in the two subpopulations are updated during the iterations. An individual in the population is randomly selected for updating, and if the updated fitness value is better, it replaces the poorer individual. In Strategy I, at a certain interval in terms of the number of iterations, the optimal individuals in the two subpopulations are compared to identify the better individuals (i.e., the yellow circle in the lower left part of Fig. 5, which is obtained by comparing the orange circle with the green circle). This individual is then compared with the optimal individual in the main population. If the former

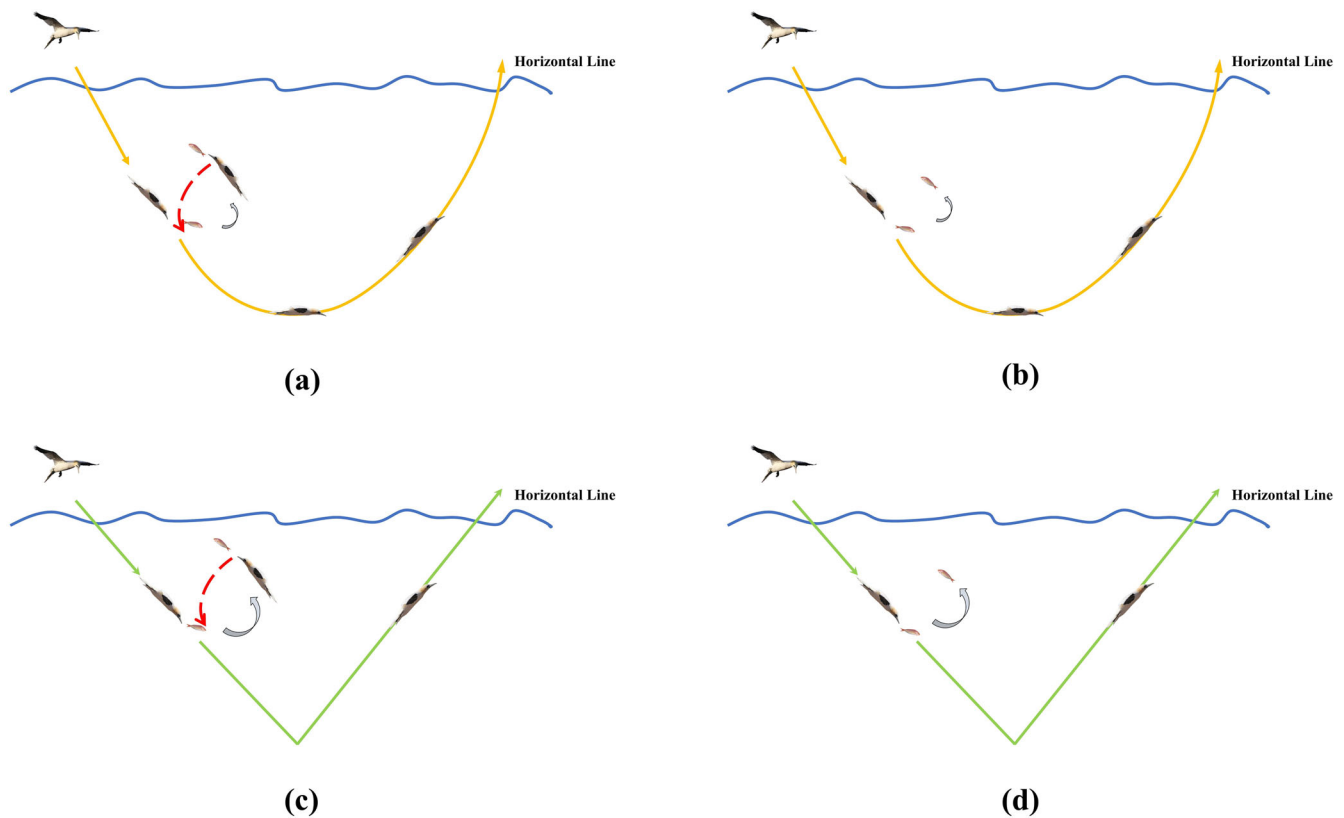


Fig. 6 | The feeding scenarios when the gannets choose the U-shaped and V-shaped methods of entering and exiting the water. a, c They return to the original route after completing fishing when they have the ability to hunt. **b, d** They do not take any action when they cannot hunt.

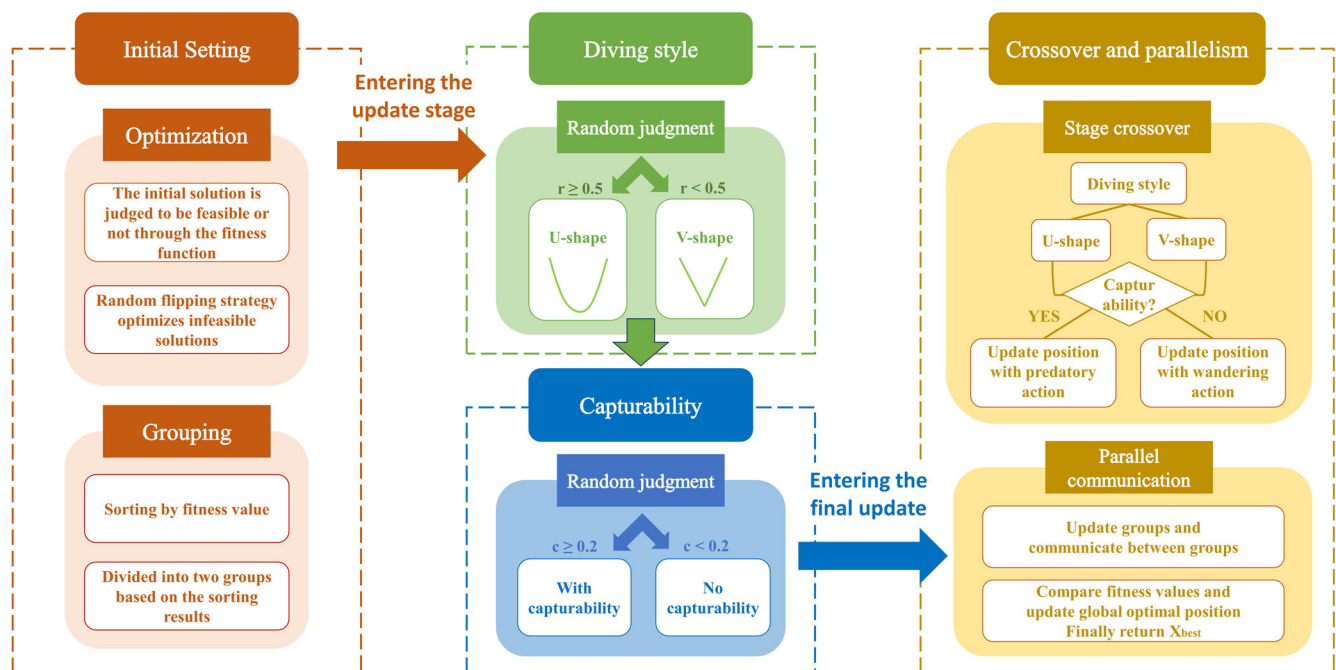


Fig. 7 | Flowchart of the IPBGOA. First optimize the initial value and set the population state, then determine the update method, and finally optimize by crossover and parallelism.

is superior, it replaces the worst individual in the main population (the optimal individual in the main population, the red circle, is compared to the yellow circle; if yellow is better, it replaces the last blue circle in the main population). This strategy of optimizing and updating from the weakest end

allows the algorithm to converge faster. When the optimal individuals in the main population are not updated for a long period, the algorithm is considered to be stuck in a local optimum; Strategy II is used to prevent this. The two previous best individuals are selected randomly, and their different

dimensions are combined to form a new individual (in the lower right part of Fig. 5, the different dimensions of Individual 1 with an orange background and Individual 2 with a green background are randomly combined to form a new individual). This new individual is used to replace some of the best individuals in the main population to prevent the algorithm from falling into local optima.

Stage crossover strategy

As described in the introduction to the GOA in Section “Gannet Optimization Algorithm”, the GOA uses two stages to update positions. In the exploration stage, the gannet dives to catch its prey in a V-shaped or U-shaped motion and then resurfaces. In the exploitation stage, the gannet determines whether to hunt on the basis of its catching ability. If it has the ability to catch, it can rotate quickly to capture fish that are trying to escape. If it cannot catch, it will allow the fish to escape and roam randomly. The algorithm randomly determines which stage's method to use for updating positions. However, these two stages do not fully reflect the way gannets hunt. For example, during the exploration stage, the U-shaped diving approach is employed, but the actual fishing actions in the water are not considered. Similarly, during the exploitation stage, the sudden rotation to catch fish is considered, but the diving and resurfacing approaches are not considered. Therefore, this paper presents a dual-stage crossover strategy that can address the shortcomings of the representation of actual fishing while diversifying the individuals considered by the algorithm to prevent convergence to local optima. Figure 6 shows the feeding diagrams of gannets according to the stage crossover strategy.

Overall optimization procedure

The IPBGOA is described by the pseudocode in Algorithm 1. Unlike in the GOA, optimization adjustments are made during the initial random solution initialization. The random flipping strategy used in Section “Random flipping strategy” enables the IPBGOA to generate feasible solutions that conform to practical applications as well as possible (lines 3–9). At the start of each iteration, the individuals are arranged according to their fitness scores and then categorized. Subsequently, each individual randomly selects a stage and then updates their final position according to their predatory behavior (lines 11–32). Since this paper discusses 0-1 problems, a transfer function is used to translate the continuous search space into a binary space (lines 33–35). After binarization, the groups are updated, and intergroup communication is performed (lines 36–41). When the stopping criteria are met, the IPBGOA returns the best solution, namely, X_{best} . The flowchart is shown in Fig. 7.

Algorithm 1. IPBGOA

Input: population size N , problem dimension Dim , number of iterations T , fitness function $F(X)$, transfer function $TF(X)$, crossover function $Crossover(X)$;

Output: Best solution X_{best} ;

```

1: Initialize the population with  $N$  individuals  $\{X_1, \dots, X_N\}$  and the best
   solution  $X_{best}$ , where  $r$  and  $q$  are random values between 0 and 1 and  $c$ 
   is a designed predation factor;
2: Create storage matrices  $SX$ ,  $SX1$ ,  $SX2$ , and  $B\_SX$ .
3: Perform feasibility judgement on the initialized individuals and
   randomly flip them.
4: while  $F(X_i) > 1$  do
5:   Update  $x_i^{Dim}$ 
6:   if  $r < 0.5$  then
7:      $\{x_i^1, \dots, x_i^{Dim-1}\} = 1 - \{x_i^1, \dots, x_i^{Dim-1}\}$ ;
8:   end if
9: end while
10: Sort the individuals based on their fitness values to obtain  $X_{best}$  and
    group them.
11: while the condition for stopping is not met do
12:   if  $r \geq 0.5$  then
13:     for  $\{SX1, SX2\}$  do

```

```

14:       Update  $SX1_i$  using Eq.(8a);
15:       if  $c \geq 0.2$  then
16:         Update  $SX2_i$  using Eq.(17a);
17:       else
18:         Update  $SX2_i$  using Eq.(17b);
19:       end if
20:        $SX_i = Crossover(SX1, SX2)$ ;
21:     end for
22:   else
23:     for  $\{SX1, SX2\}$  do
24:       Update  $SX1_i$  using Eq.(8b);
25:       if  $c \geq 0.2$  then
26:         Update  $SX2_i$  using Eq.(17a);
27:       else
28:         Update  $SX2_i$  using Eq.(17b);
29:       end if
30:        $SX_i = Crossover(SX1, SX2)$ ;
31:     end for
32:   end if
33:   for  $i = 1 \rightarrow N$  do
34:      $B\_SX = TF(SX_i)$ ;
35:   end for
36: Sort the updated individuals based on their fitness values to obtain  $X_{best}$ 
   and update the groups;
37: Intergroup communication.
38: if the set number of iterations is reached then
39:   Compare the optimal individuals of group1 and group2 with  $X_{best}$ 
   and perform a swap;
40:   If there has been no update for a long time, perform random
   dimension combination;
41: end if
42: end while
43: return  $X_{best}$ ;

```

Data availability

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Code availability

The underlying code for this study [and training/validation datasets] is not publicly available but may be made available to qualified researchers on reasonable request from the corresponding author.

Received: 26 June 2024; Accepted: 3 November 2024;

Published online: 12 November 2024

References

1. Zhao, W., Beach, T. H. & Rezgui, Y. Optimization of Potable Water Distribution and Wastewater Collection Networks: A Systematic Review and Future Research Directions. *IEEE Trans. Syst. Man Cybern. Syst.* **46**, 659–681 (2016).
2. Geissen, V. et al. Emerging pollutants in the environment: A challenge for water resource management. *Int. Soil Water Conserv. Res.* **3**, 57–65 (2015).
3. Nabeela, F. et al. Microbial contamination of drinking water in Pakistan —a review. *Environ. Sci. Pollut. Res.* **21**, 13929–13942 (2014).
4. Aral, M. M., Guan, J. & Maslia, M. L. Optimal Design of Sensor Placement in Water Distribution Networks. *J. Water Resour. Plann. Manag.* **136**, 5–18 (2010).
5. Braunstein, A., Lage-Castellanos, A. & Ortega, E. Contamination source inference in water distribution networks. Preprint at <http://arxiv.org/abs/1712.00486> (2017).
6. BENK[Otilde], N., Rev, E. & Fonyo, Z. The use of nonlinear programming to optimal water allocation. *Chem. Eng. Commun.* **178**, 67–101 (2000).

7. Pan, J.-S., Hu, P., Snášel, V. & Chu, S.-C. A survey on binary metaheuristic algorithms and their engineering applications. *Artif. Intell. Rev.* **56**, 6101–6167 (2023).
8. Mirjalili, S. & Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **95**, 51–67 (2016).
9. Negm, N. et al. Intracranial Haemorrhage Diagnosis Using Willow Catkin Optimization With Voting Ensemble Deep Learning on CT Brain Imaging. *IEEE Access* **11**, 75474–75483 (2023).
10. Liu, N., Liu, S., Chai, Q.-W. & Zheng, W.-M. A method for analyzing Stackelberg attack–defense game model in 5G by tCPSO. *Expert Syst. Appl.* **228**, 120386 (2023).
11. Chu, S.-C., Xu, X.-W., Yang, S.-Y. & Pan, J.-S. Parallel fish migration optimization with compact technology based on memory principle for wireless sensor networks. *Knowl. Based Syst.* **241**, 108124 (2022).
12. Li, L., Pan, T.-S., Sun, X.-X., Chu, S.-C. & Pan, J.-S. A Novel Binary Slime Mould Algorithm with AU Strategy for Cognitive Radio Spectrum Allocation. *Int. J. Comput. Intell. Syst.* **14**, 161 (2021).
13. Dorigo, M., Birattari, M. & Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **1**, 28–39 (2006).
14. Katoch, S., Chauhan, S. S. & Kumar, V. A review on genetic algorithm: past, present, and future. *Multimed. Tools Appl.* **80**, 8091–8126 (2021).
15. Kennedy, J. & Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* vol. 4 1942–1948 (IEEE, 1995).
16. Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**, 182–197 (2002).
17. Rath, S., Gupta, R., Kamble, S. & Sargaonkar, A. Risk Based Analysis for Contamination Event Selection and Optimal Sensor Placement for Intermittent Water Distribution Network Security. *Water Resour. Manag.* **30**, 2671–2685 (2016).
18. Hu, C., Tian, D., Liu, C. & Yan, X. Sensors Placement in Water Distribution Systems Based on Co-evolutionary Optimization Algorithm. In *Proceedings of the 1st International Conference on Industrial Networks and Intelligent Systems (ICIST, 2015)*. <https://doi.org/10.4108/icist.iniscom.2015.258402>.
19. Razavi, S. G., Nazif, S. & Ghorbani, M. Community Resilience and Consequence Management of Pollution Intrusion Into Water Distribution Network: A Case Study. *Soc. Nat. Resour.* **36**, 821–839 (2023).
20. Afshar, A. & Najafi, E. Consequence management of chemical intrusion in water distribution networks under inexact scenarios. *J. Hydroinformatics* **16**, 178–188 (2014).
21. Bashi-Azghadi, S. N., Afshar, M. H. & Afshar, A. Multi-objective optimization response modeling to contaminated water distribution networks: Pressure driven versus demand driven analysis. *KSCE J. Civ. Eng.* **21**, 2085–2096 (2017).
22. Ehsani, N. & Afshar, A. Optimization of Contaminant Sensor Placement in Water Distribution Networks: Multi-Objective Approach. 338–346, [https://doi.org/10.1061/41203\(425\)32](https://doi.org/10.1061/41203(425)32) (2012).
23. Masoumi, F., Bashi-Azghadi, S. N. & Afshar, A. Application of Achieve-Based Genetic Algorithm for Consequence Management of Contaminant Entering in Water Distribution Networks. *Amirka-bir J. Civ. Eng.* **53**, 3593–3604 (2021).
24. Tao, Y., Yan, D., Yang, H., Ma, L. & Kou, C. Multi-objective optimization of water distribution networks based on non-dominated sequencing genetic algorithm. *PLoS ONE* **17**, e0277954 (2022).
25. Tahani, M., Yousefi, H., Noorollahi, Y. & Fahimi, R. Application of nature inspired optimization algorithms in optimum positioning of pump-as-turbines in water distribution networks. *Neural Comput. Appl.* **31**, 7489–7499 (2019).
26. Afzali Ahmadi, S., Jafari-Asl, J., Banifakhr, E., Houssein, E. H. & Ben Seghier, M. E. A. Risk-Based Design Optimization of Contamination Detection Sensors in Water Distribution Systems: Application of an Improved Whale Optimization Algorithm. *Water* **15**, 2217 (2023).
27. Du, K. et al. Auto-enhanced population diversity and ranking selection-based differential evolutionary algorithm applied to the optimal design of water distribution system. *AQUA Water Infrastruct., Ecosyst. Soc.* **72**, 1553–1565 (2023).
28. Pan, J.-S., Yue, L. & Chu, S.-C. Binary Gannet Optimization Algorithm for Feature Selection Problem. In *The Eleventh International Conference on Machine Intelligence Theory and Applications*.
29. Xu, L., Geng, F.-D., Hu, R.-B. & Wang, R.-B. *Binary Gannet Optimization Algorithm for Feature Selection Using Time-Varying Transfer Function*. <https://www.researchsquare.com/article/rs-3111122/v1> (2023).
30. Bilal & Pant, M. Parameter Optimization of Water Distribution Network – A Hybrid Metaheuristic Approach. *Mater. Manuf. Process.* **35**, 737–749 (2020).
31. Eliades, D. G., Kyriakou, M. & Polycarpou, M. M. Sensor Placement in Water Distribution Systems Using the S-PLACE Toolkit. *Procedia Eng.* **70**, 602–611 (2014).
32. Ghorbani, V., Karney, B. & Guo, Y. Pressure Standards in Water Distribution Systems: Reflection on Current Practice with Consideration of Some Unresolved Issues. *J. Water Resour. Plan. Manag.* **142**, 04016023 (2016).
33. Jowitt, P. W. & Xu, C. Optimal Valve Control in Water-Distribution Networks. *J. Water Resour. Plan. Manag.* **116**, 455–472 (1990).
34. Vrachimis, S. G., Lifshitz, R., Eliades, D. G., Polycarpou, M. M. & Ostfeld, A. Active Contamination Detection in Water-Distribution Systems. *J. Water Resour. Plan. Manag.* **146**, 04020014 (2020).
35. Hu, C. et al. Multi-objective based scheduling algorithm for sudden drinking water contamination incident. *Swarm Evolut. Comput.* **55**, 100674 (2020).
36. Piotrowski, A. P., Napiorkowski, M. J., Napiorkowski, J. J. & Rowinski, P. M. Swarm Intelligence and Evolutionary Algorithms: Performance versus speed. *Inf. Sci.* **384**, 34–85 (2017).
37. Wang, Y. et al. Minimizing Pumping Energy Cost in Real-Time Operations of Water Distribution Systems Using Economic Model Predictive Control. *J. Water Resour. Plan. Manag.* **147**, 04021042 (2021).
38. Sadatiyan Abkenar, S. M., Stanley, S. D., Miller, C. J., Chase, D. V. & McElmurry, S. P. Evaluation of genetic algorithms using discrete and continuous methods for pump optimization of water distribution systems. *Sustain. Comput.: Inform. Syst.* **8**, 18–23 (2015).
39. Geem, Z. W. Particle-swarm harmony search for water network design. *Eng. Optim.* **41**, 297–311 (2009).
40. Ratnayaka, D. D., Brandt, M. J. & Johnson, M. *Water Supply* (Butterworth-Heinemann, 2009).
41. Eusuff, M. M. & Lansey, K. E. Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm. *J. Water Resour. Plan. Manag.* **129**, 210–225 (2003).
42. Samani, H. M. V. & Mottaghi, A. Optimization of Water Distribution Networks Using Integer Linear Programming. *J. Hydraul. Eng.* **132**, 501–509 (2006).
43. Rossman, L. A. EPANET 2: users manual. (2000)
44. Pan, J.-S., Zhang, L.-G., Wang, R.-B., Snášel, V. & Chu, S.-C. Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems. *Math. Comput. Simul.* **202**, 343–373 (2022).
45. Mirjalili, S. & Lewis, A. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm Evolut. Comput.* **9**, 1–14 (2013).
46. Beheshti, Z. UTF: Upgrade transfer function for binary meta-heuristic algorithms. *Appl. Soft Comput.* **106**, 107346 (2021).
47. Pan, J.-S., Shi, H.-J., Chu, S.-C., Hu, P. & Shehadeh, H. A. Parallel Binary Rafflesia Optimization Algorithm and Its Application in Feature Selection Problem. *Symmetry* **15**, 1073 (2023).

Acknowledgements

This work is supported by the National Natural Science Foundation of China (61872085).

Author contributions

Conceptualization, Jeng-Shyang Pan and Hao Shu; methodology, Jeng-Shyang Pan, Hao Shu; validation, Hao Shu and Qingyong Yang; formal analysis, Hao Shu; investigation, Hao Shu and Shu-Chuan Chu; investigation, Hao Shu and Shu-Chuan Chu; resources, Hao Shu and Qingyong Yang; data curation, Shu-Chuan Chu and Yu-Chung Huang; writing—original draft, Hao Shu; writing—review and editing, Jeng-Shyang Pan and Shu-Chuan Chu. All authors have read and agreed to the published version of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41545-024-00407-5>.

Correspondence and requests for materials should be addressed to Shu-Chuan Chu.

Reprints and permissions information is available at <http://www.nature.com/reprints>

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

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, 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 you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. 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-nc-nd/4.0/>.

© The Author(s) 2024