Desalination and Water Treatment
www.deswater.com

doi: 10.5004/dwt.2018.22330

110 (2018) 308–318
April

Pollution source positioning in a water supply network based on expensive optimization

Xuesong Yan[a,b], Kewei Yang[a,*], Chengyu Hu[a], Wenyin Gong[a]

[a]*School of Computer Science, China University of Geosciences, Wuhan 430074, China, email: yanxs1999@126.com (K. Yang)*
[b]*State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China*

Received 7 November 2017; Accepted 4 February 2018

ABSTRACT

In recent years, several drinking water pollution accidents that severely affected social stability and security have occurred. A pollution accident can be effectively prevented by deploying sensors in urban water supply pipes to monitor water quality in real time. However, it is a challenge to back calculate a pollution source from information detected by a water quality sensor. In this paper, characteristics of pollution source positioning are analyzed in detail; pollution source positioning is converted into an expensive optimization problem to find a solution. Additionally, based on the characteristics of the water supply network, a Gaussian agent model is created for each node in the supply network. A Gaussian agent model-based expensive optimization algorithm is proposed to solve the pollution source positioning problem in a water supply network. To verify the effectiveness of the proposed method, data from a water supply network are used for a lab simulation; the predicted results prove the effectiveness and efficiency of the proposed algorithm.

Keywords: Pollution source; Positioning; Expensive optimization algorithm; Gaussian agent model; Water supply network; Simulation–optimization

1. Introduction

In many practical engineering optimization problems, the objective function cannot be clearly expressed; therefore, the optimization model is rather complex. Simulation and evaluation of these problems require simulation software, which is time consuming. Because each calculation is time consuming and has a high economic cost, this type of problem is called an expensive optimization problem.

In recent years, a number of unexpected water pollution accidents in China have occurred. Some of these unexpected drinking water pollution accidents as well as malicious attacks on the water supply networks have caused significant economic loss and severe social impacts in China. To prevent a severe water pollution-induced disaster and a loss of drinking water, a safety real-time monitoring system should be deployed in urban water supply networks. In this system, water quality sensors are deployed at critical nodes or water sources for real-time monitoring. However, when pollution emerges, it is a challenge to identify characteristics of the pollution source via information collected by water quality sensors to predict the pollutant location, injection time, duration and amount.

Recently, numerous researchers have attempted to convert the pollution source positioning problem to an optimization problem using the simulation–optimization model. For example, Ostfeld et al. [1] matched a pollution accident in a random pollution matrix using the pollutant state measured by a water quality sensor and performed a reverse search for the pollution source location and the injection amount. Guan et al. [2] proposed a simulation–optimization method, which continuously read the sensor data to optimize the forecast, corrected the pollution source, and finally identified the pollution source and pollutant discharge history, to solve a non-linear pollution source positioning problem. Preis and

* Corresponding author.

Ostfeld [3] proposed a solution for the pollution source positioning problem using a genetic algorithm and analyzed the sensitivity of the sensor. Zechman and Ranjithan [4] proposed an evolutionary strategy-based method, which identified the best match pollution source based on information from the monitoring point and a global heuristic search algorithm. Mou et al. [5] simulated the pollution source via sodium hypochlorite and compared the results for various input parameters. Liu et al. [6] proposed an evolutionary algorithm based on the adaptive dynamic optimization technology to identify the pollution source pattern (start time, location and discharge history); new sensors were continuously added to gradually assist convergence and obtain a unique optimal solution. Jha and Datta [7] proposed an accurate model to solve the problem of groundwater pollution identification. In the proposed model, the problem is optimized by differential optimization [7]. Hu et al. [8] proposed the MapReduce-based parallel microhabitat genetic algorithm to solve the pollution source positioning problem in which the microhabitat genetic algorithm was the optimizer and EPANET was the simulator. Yan et al. [9] proposed a hybrid encoding-based genetic algorithm to improve the algorithm convergence rate, and they proposed a cultural algorithm for this problem [10]. They also converted the pollution source positioning problem into a multimodal optimization problem and proposed a niching genetic algorithm to solve it [11]. Considering the uncertainty of user water demand, Yan et al. [12,13] applied various models to simulate user water demand and then employed a genetic algorithm to solve a pollution source positioning problem with uncertain water demand. Rasekh and Brumbelow [14] proposed dynamic simulation optimization model taking into account a number of uncertainties that lead to unpredictable time-varying system behaviour in the real world. In the simulation–optimization method, the optimization algorithm is used as the optimizer. In the optimization algorithm, each individual requires EPANET to simulate the pollution event and then calculate the individual fitness. BWSN2 [1] is used as an example (this water supply network contains 12,527 nodes, 2 reservoirs, and 2 water pools with 20 sensors) to simulate a pollution accident. Each fitness calculation takes 1.2 s. When the genetic algorithm (population scale is 100 and generation is 100) is employed to calculate a solution, EPANET is called approximately 16,500 times, which takes 16,500 * 1.2 = 19,800 s or nearly 5.5 h. This example shows that during optimization, the EPANET simulator takes a significant amount of time. To minimize the threat of a pollutant on public health, when a certain amount of water quality information is available, the pollution source should be located as soon as possible. At this moment, more searches are required to find the optimal solution, which consumes more computing resources. Therefore, water supply network pollution source positioning is an expensive optimization problem.

The expensive optimization problem has been the focus of many studies. In 1998, Jones et al. [15] provided the expected value for a non-sampling point using the Gaussian random model in a branch-bound algorithm. They also analyzed the effectiveness of a random model and proved that it was an effective global optimization algorithm for an expensive problem. In 2002, Jin et al. [16] introduced a random model in an evolution algorithm and created a global random model for a global forecast. In 2004, Regis and Shoemaker [17] created a local model from a random model for a local forecast in an evolution algorithm. In 2007, Zhou et al. [18] introduced a random model in an evolution algorithm and created a global model in tandem with a local model to accelerate evolution efficiency. Studies by Paenke et al. [19] and Fieldsend and Everson [20] are papers on a model based on a single-objective evolution algorithm. The study by Liu et al. [21] is a paper on a model based on an expensive multi-objective evolution algorithm. Studies by Jeong and Obayashi [22], Keane [23], Ponweiser et al. [24] and Zhou et al. [25] are papers on a model based on a multi-objective evolution algorithm. The study by Tenne and Goh [26] is the application of an intelligent computing method on an expensive optimization problem. In 2010, Luo et al. [27] embedded a meta-modelling mechanism in a global search algorithm to achieve a balance between the forecast model and global search algorithm. In 2014, Singh et al. [28] applied the Kriging model and a local search in a global search algorithm to create a forecast model. In 2014, Liu et al. [29] combined a Gaussian forecast model and optimization algorithm to solve a high-dimensional global optimization problem. In 2015, Bhattacharjee and Ray [30] embedded a selection evaluation strategy in a support vector machine forecast model to provide a graded forecast for a constrained optimization problem. In 2017, Sun et al. [31] employed a coordinated particle swarm optimization algorithm to solve a high-dimensional expensive optimization problem.

To obtain an optimal solution for a pollution source positioning problem, a large number of iterative calculations and thousands of evaluations are required. If the solution is based on a normal optimization algorithm, a large number of iterations are required to find an optimal solution using the optimization algorithm, which results in frequent use of the EPANET simulator and severely affects the algorithm performance and efficiency. The key to solving this problem is to minimize the EPANET simulator usage without affecting the algorithm positioning precision. Therefore, a proper agent model is introduced in the expensive optimization algorithm to replace the EPANET simulator for the individual fitness calculation. There are two major challenges in solving a pollution source positioning problem using an expensive optimization algorithm: one is how to create an agent model with a high forecast precision based on a sampling point; the other is how to balance the usage of the agent model and expensive evaluation function so that the algorithm can find the optimal solution in a fast and accurate manner.

In this paper, the pollution source positioning problem is converted into an expensive optimization problem to find a solution. First, a problem model for the pollution source positioning is provided; next, based on the problem model, an expensive optimization problem-based solution framework is proposed; then, an agent model is created using the Gaussian random process. Due to characteristics of the Gaussian random process and the water supply network, a sub-model for each node in the supply network is created and the effectiveness is verified. Then, a Gaussian agent model-based expensive optimization algorithm is proposed to solve the pollution source positioning problem. Finally, the model's effectiveness and efficiency are verified.

2. Materials and methods

2.1. Modelling the expensive optimization-based pollution source positioning problem

2.1.1. Model for the pollution positioning problem

The simulation–optimization method converts the pollution source positioning problem into an optimization problem. Then, they identify the pollution source location by calculating the optimal solution using an evolution method. When calculating a solution for the pollution source positioning problem via a simulation–optimization model, an optimization algorithm is used as the optimizer to generate the pollution event, and EPANET is used as the simulator to simulate the pollution event and generate the predicted pollutant concentration at each node. The EPANET simulator generates the forward waterpower and water quality state, which is compared with the actual water quality measured by a sensor. From the perspective of optimization, when the minimum variance between the simulated accumulated concentration for a pollution event at the sensor and the measured accumulated concentration is 0 or less than a threshold, the injection node of this pollution event is treated as the actual pollution source. The optimization problem is described as shown in Eq. (1).

$$\begin{aligned}
&\underset{\{M,n,t_I\}}{\text{Minimize}}\, f=\sum_{j=1}^{N_s}\sum_{t=1}^{T_s}(c_j(t)-c_j^*(t))^2\\
&\text{S.T.}\, M=\{m_1,m_2,...,m_\mathrm{k}\}; m_\mathrm{i}\geq 0\\
&n\in\{1,N\}\\
&t_I\leq T_s
\end{aligned}\tag{1}$$

where N is the total number of nodes in the supply network; N_s represents the number of sensors; T_s represents the simulation cycle; M represents the pollutant injection vector; n represents the sequence number of the node in the supply network with a pollution source injection; t_I represents the start time of the pollutant injection; $c_j(t)$ represents the pollutant concentration at time t at sensor j, which is a function of (M,n,t_I); and $c_j*(t)$ represents the measured pollutant concentration at time t at sensor j. The optimization objective is to calculate the (M,n,t_I) that minimizes the variance.

2.1.2. Expensive optimization-based solution model

When the simulation–optimization model is employed to solve a pollution source positioning problem, EPANET is used as the simulator, and the optimization algorithm is used as the optimizer. Differing from a normal simulation–optimization model, when calculating the individual fitness, either the EPANET simulator or Gaussian agent model can be used. Introducing the Gaussian agent model in the optimization algorithm reduces the usage of the EPANET simulator and improves the algorithm efficiency. An expensive optimization algorithm-based solution framework is shown in Fig. 1.

The expensive optimization algorithm employs the Gaussian random process for modelling and a genetic algorithm for the optimization algorithm. Each individual in the population represents a pollution event. The pollution event is simulated using the EPANET simulator to obtain the pollutant concentration at a node in the supply network. The predicted pollutant concentration is compared with the actual measurement of a sensor to calculate the individual fitness. The individual fitness value can also be predicted using the Gaussian agent model. A proper balance of usage of EPANET and the Gaussian agent model minimizes the usage of the EPANET simulator to reduce the algorithm time cost while ensuring positioning precision. Therefore, the algorithm has two major challenges: one is how to create the proper Gaussian agent model, and the other is how to balance the usage of the Gaussian agent model and EPANET simulator.

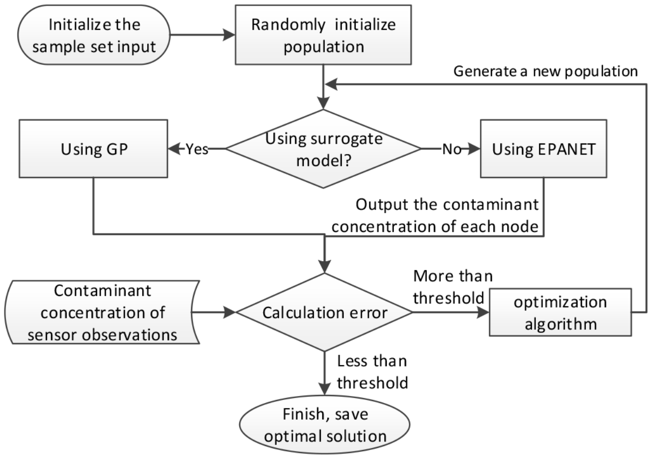

Initialize the sample set input
Randomly initialize population
Generate a new population
Using GP
Yes
Using surrogate model?
No
Using EPANET
Output the contaminant concentration of each node
Contaminant concentration of sensor observations
Calculation error
More than threshold
optimization algorithm
Less than threshold
Finish, save optimal solution

Fig. 1. Expensive optimization algorithm-based solution framework.

2.2. Pollution source positioning algorithm based on expensive optimization

The expensive optimization-based pollution source positioning algorithm proposed in this paper has two major steps. The first step is to create a proper agent model using the Gaussian random process; and the second step is to apply the Gaussian agent model in the pollution source positioning algorithm to minimize the usage of the EPANET simulator while ensuring the algorithm positioning precision.

2.2.1. Modelling based on the Gaussian random process

The forecast simulation has the most direct impact on the individual evaluation. Therefore, a proper forecast model is the key to an expensive optimization problem. The Gaussian random process [32–34] model is a method to create an agent model method. The Gaussian random process has a limited number of simulation parameters and facilitates finding a solution by employing the maximum likelihood probability and optimization algorithm. The Gaussian random process is employed as an agent model because (1) the Gaussian random process can easily overcome over fitting, (2) the Gaussian random process model has a limited number of adjustable parameters and (3) after modelling, a sample can be added to the Gaussian random process model in real time to update the model, which helps improve the model precision.

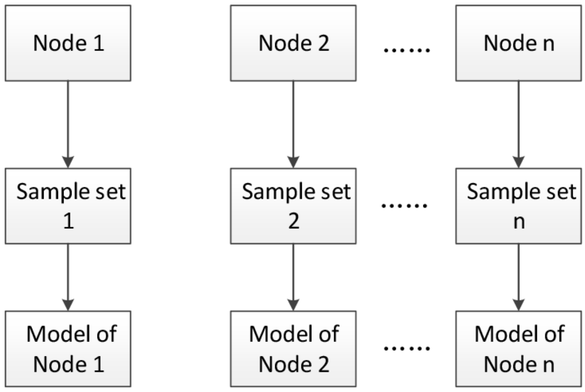


Fig. 2. Sub-model creation.

In this paper, a sub-Gaussian model based on the inter-node pattern in a water supply network is created for each node in the network, as shown in Fig. 2.

As shown in the diagram, each node in the water supply network has individual sampling and a sub-Gaussian agent model. There are two reasons for this configuration:

1. In the Gaussian random process, modelling is primarily based on the correlation coefficient matrix, which consists of a correlation coefficient between each sample. Because of the complexity of a water supply network, the correlation between most nodes is insignificant. Therefore, a model of the entire supply network has a low forecast precision for a node outside the sample.
2. Based on the Gaussian random process modelling procedure described above, the time complexity of the Gaussian agent model creation is $O(N_{it}K^3d)$ [19]. In the expression, N_{it} represents the number of iterations, K represents the size of the sample set and d represents the number of variables in the model. The expression shows that, as the size of the sample set increases, the modelling calculation time increases by a power of 3. For a large scale water supply network, when a sample set does not cover all nodes, the forecast precision for a node outside the sample is low; when all nodes are covered, the sample set increases and the time cost increases significantly (e.g., BWSN2 in Fig. 4 [this supply network contains 12,527 nodes, 2 reservoirs, 2 water pools and 20 sensors] has a sample size of 640 and a modelling time of 1,030 s ≈ 17 min).

Based on Fig. 2, the Gaussian agent model is created in two major steps:

Step 1: Sampling. A model sample is collected at each node. Each sample contains individual and corresponding fitness. Individual fitness contains four variables including the pollution source position, start time, duration and injection quantity vector. In this paper, 10 samples are collected randomly for a point at a different node (to balance the time cost and precision, and 10 samples are collected for each model).

Step 2: Modelling. For each node, based on the 10 samples collected in Step 1, a Gaussian agent model is created using the Gaussian random process. For a detailed Gaussian random process modelling procedure [32].

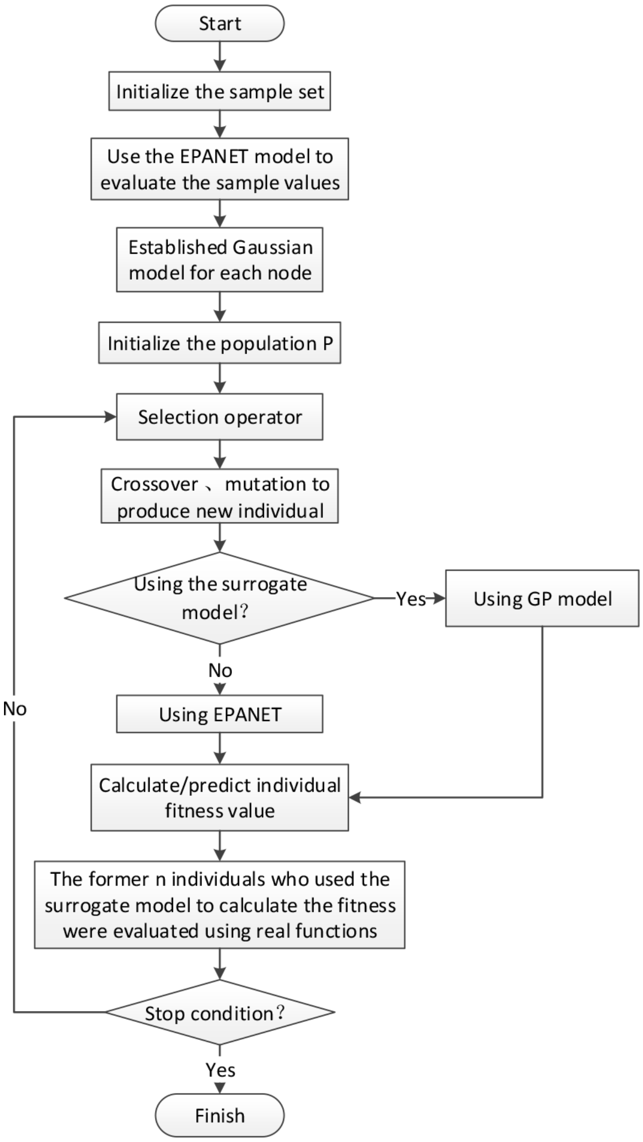
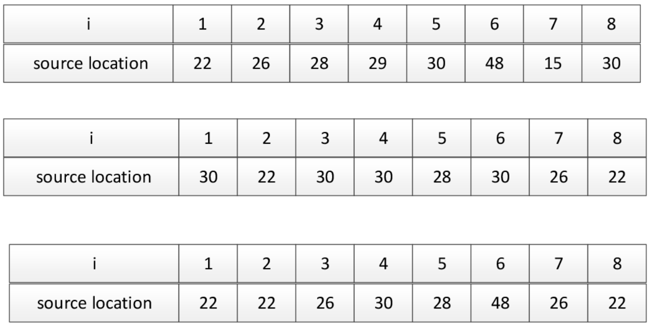

Start
Initialize the sample set
Use the EPANET model to evaluate the sample values
Established Gaussian model for each node
Initialize the population P
Selection operator
Crossover 、 mutation to produce new individual
Using the surrogate model?
Yes
Using GP model
No
Using EPANET
Calculate/predict individual fitness value
The former n individuals who used the surrogate model to calculate the fitness were evaluated using real functions
Stop condition?
No
Yes
Finish

Fig. 3. Flow chart of the proposed expensive optimization algorithm.



i	1	2	3	4	5	6	7	8
source location	22	26	28	29	30	48	15	30

i	1	2	3	4	5	6	7	8
source location	30	22	30	30	28	30	26	22

i	1	2	3	4	5	6	7	8
source location	22	22	26	30	28	48	26	22

Fig. 4. Improved individual selection strategy.

2.2.2. Design of the expensive optimization algorithm

Based on the classical expensive optimize solution, the Gaussian agent model is applied to the optimization algorithm convergence process to reduce the usage of the actual evaluation function. In the optimization process, a strategy is adjusted continuously to balance the usage of the Gaussian forecast model and the EPANET simulator so that the algorithm meets the required precision and the usage of the EPANET simulator is minimized. Based on the expensive optimization algorithm-based pollution source positioning solution framework in Fig. 1, a Gaussian random process-based expensive optimization algorithm is proposed, which is shown in a flow chart in Fig. 3.

Based on the algorithm flow chart, a detailed procedure of the algorithm is as follows:

Step 1: Initialize the population. To minimize the EPANET usage, the population is initialized directly from the sample set.

Step 2: Improve the roulette selection.

Step 3: The crossover operator is based on a dual-point crossover and real number crossover; the mutation operator is based on a single point mutation and Gaussian mutation [9].

Step 3.1: For a new individual generated from the crossover and mutation, the individual fitness μ and error σ are forecasted by the Gaussian agent model. If the trigger coefficient satisfies $3\sigma/\mu < 0.2$ (0.2 is obtained based on the test and analysis), the forecast is used as the new individual fitness; otherwise, go to Step 3.2.

Step 3.2: A probability $P_*$is generated randomly. If $P_* < P, P = t/x$, the fitness is calculated by EPANET. In the expression, t represents iterations, and x is the base; otherwise, the individual fitness is calculated by the Gaussian agent model.

Step 4: After the completion of each iteration, the population is sorted based on fitness. For the first N individuals whose fitness values are calculated by the Gaussian agent model, the fitness is recalculated by the EPANET model for correction.

Step 5: The first M individuals with superior fitness values are reserved by an elite strategy and directly selected as the next generation.

Step 6: Determine if the termination condition is met. If it is met, the algorithm terminates; otherwise, go to Step 2.

As shown in the figure, the expensive optimization algorithm proposed in this paper has two major improvements: an improved selection strategy and an improved Gaussian agent model usage strategy.

2.2.3. Improved selection strategy

Because of the complexity of a water supply network, when a supply network is large, different pollution source positions have significantly different individual fitness; therefore, a classical roulette selection operator can easily trap in a local optimal situation. The improved roulette selection method is used in this paper to avoid this issue. Details of the improved roulette selection method [35] are as follows:

Step 1: For population P, a new population np is selected using classical roulette.

Step 2: Occurrences of individuals with an identical pollution source position in the new population np are counted.

Step 3: For an individual with an identical pollution source position and whose occurrence is equal to or exceeds n, the individual pi with the best fitness is replicated to population P; the remaining n–1 individuals are not replicated, and the corresponding positions are reserved for population P. If the pollution source position occurrence is less than n, the individual in np is directly replicated to the corresponding position in population P (n is obtained based on an empirical value from the test). For example, population P is the first figure, after classical roulette selection, np is obtained as the second figure, then assume that $n = 4$, the pollution source position occurrence is counted. There are four occurrences of the pollution source position at 30. Assume that position $i = 4$ has the best fitness; therefore, the updated population P is shown as the Fig. 4.

2.2.4. Gaussian agent model usage strategy

For the expensive optimization algorithm, the usage of the Gaussian model has a direct impact on the algorithm efficiency. If the agent model is used excessively, the algorithm may not converge and the pollution source cannot be identified. On the other hand, underuse does not reduce the time cost. The Gaussian agent model usage strategy maximizes the usage of the Gaussian agent model while ensuring identification of the pollution source. In the pollution source positioning problem, which is limited by the number of sensors in a supply network (sensor deployment cannot cover an entire network), pollutant detection requires time; due to the characteristics of a supply network (a portion of pipes only support unidirectional flow), some nodes cannot be detected by a sensor or can only be detected by a small portion of the sensors. The difference between the sensor measurement and actual data at the pollution source (i.e., the fitness in this paper) has a small range of fluctuation. At other nodes, especially those close to the pollution source, the sensor measurement data are sufficient, and the fitness has a wide range of fluctuation.

In the pollution source positioning problem, the Gaussian agent model is applied to forecast two types of points. One is a relatively sensitive point (a slight change in the decision variable results in a significant change in the fitness) with inferior forecast stability and significant forecast error. The other is relatively insensitive point with a high forecast precision and stability. For these points, the agent model usage should be maximized to reduce EPANET usage and save time cost.

First, a trigger coefficient is proposed to determine insensitivity. When a point trigger coefficient forecast is below a threshold, this point is classified as an insensitive point. The formula for the trigger coefficient is as shown in Eq. (2).

$$3\sigma/\mu<\varepsilon \tag{2}$$

The Gaussian agent model is applied to each newly generated point to create a Gaussian model $N(\mu,\sigma)$for each forecast point. In the expression, μ represents the forecast expectation (point forecast value) and σ represents the forecast error or model forecast stability. A smaller σ value means a forecast model is more stable and the forecast has higher credibility. In a Gaussian distribution, the probability of forecast within the range of [μ–3σ,μ+3σ] is 0.9973; the probability outside this range is rare and is not considered in this paper.

In the expensive optimization algorithm, the individual forecast is based on the Gaussian agent model, and individual fitness is μ. |μ–(μ–3σ)| or 3σ is the maximum forecast deviation. Therefore, the trigger coefficient 3σ/μ is proposed as a criterion to measure the error in this paper. Due to the complexity of a supply network, a different individual in the algorithm has a significantly different fitness; the difference between individuals can even reach to the millions. Some sensitive points have a large forecast error σ. Therefore, 3σ/μ is large and unsuitable for sensitive point determination. In this paper, the trigger coefficient 3σ/μ is used as a criterion to determine an insensitive point. When this value is less than a threshold, the model forecast precision meets the requirement and the Gaussian agent model is applied directly. However, when the trigger coefficient is used for determination, numerous points still cannot be determined. To maximize the agent model usage without affecting the algorithm convergence, the agent model is applied in a certain probability. Because a sensor is more sensitive at a point close to the pollution source and sensitivity at the forecast point is high (large forecast error), when an algorithm converges, the population becomes closer to the pollution source. Therefore, a linear probability formula is proposed in this paper as shown in Eq. (3).

$$P = t/x \tag{3}$$

where t represents the algorithm iterations and x represents the base, which are defined based on the test and analysis.

3. Results

3.1. Water supply network parameters and algorithm parameters setup

To prove the necessity of the agent model, a large-scale supply network, BWSN2 [1], is used in this paper for testing. As shown in Fig. 5, the water supply network contains 12,527 nodes, 2 reservoirs and 2 water pools. In this water supply network, 20 sensors are deployed (7626, 8912, 5363, 6632, 6725, 4889, 10861, 2372, 8820, 3070, 6840, 11550, 3430, 7959, 6744, 9488, 11330, 7211, 6006, 5890). The total simulation time for the water supply network is 48 h. In the simulation, the water power time step is 1 h, and the water quality time step is 5 min. The actual pollution scenario is a pollutant is injected at node 4528 continuously for 2 h after 2 h of simulation. Parameters of the Gaussian agent model-based expensive optimization algorithm are listed in Table 1.

Test platform specifications are an Intel Core i5-6500 @ 3.20GHZ processor, 8.0GB of memory and a Windows 7 Professional 64-bit operating system.

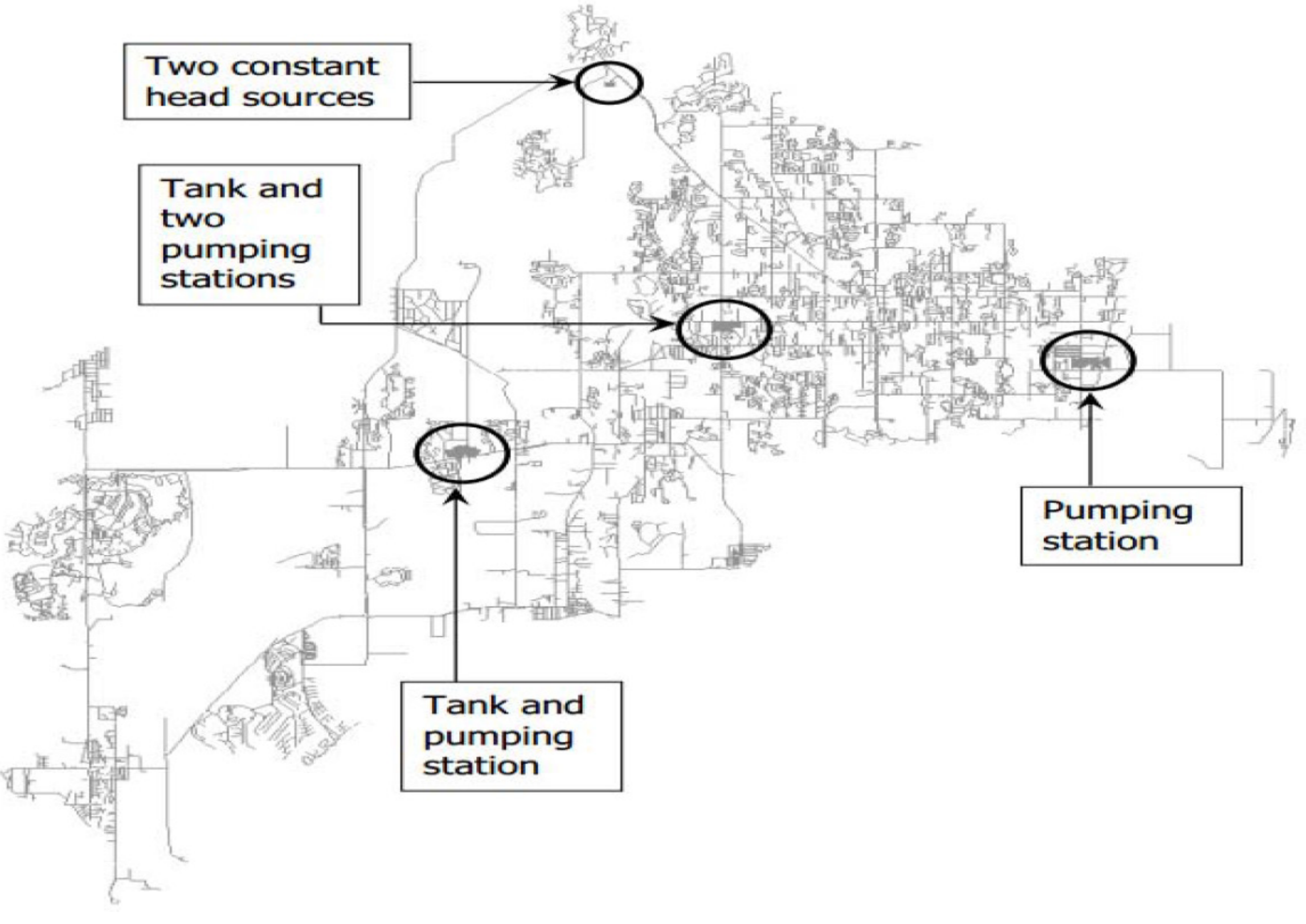


Fig. 5. BWSN2 network.

Table 1
Algorithm parameters setup

Parameter	Description	Value
POP_SIZE	Population size	100
NUM_ITRE	Number of iterations	100
Pc	Crossover probability	95%
P_m	Mutation probability	70%
M	Individual selected by the elite strategy	5
n	Improved roulette parameter	6
N	Number of updated individuals	10

In this paper, the test is divided into two parts. Part 1 is the algorithm parameter analysis. Parameters that directly affect the usage of EPANET are analyzed and proper values for the parameters are provided. Part 2 is the algorithm performance analysis. The algorithm with agent model usage and the algorithm without agent model usage are compared and any difference is identified. The number of EPANET evaluations and algorithm time cost are analyzed to verify effectiveness and efficiency of the Gaussian agent model-based expensive optimization algorithm.

3.2. Algorithm parameter analysis

3.2.1. Trigger coefficient 3σ/μ threshold analysis

The trigger coefficient 3σ/μ is a criterion to measure the forecast error of the Gaussian agent model. A threshold less than this value is used to determine a relatively insensitive point and it is a factor that directly affects actual EPANET usage. When the threshold is oversized, the Gaussian agent model is applied more frequently, which continuously reduces the time cost, but the algorithm positioning precision is affected. On the other hand, when the threshold is undersized, the algorithm time cost increases. Therefore, this threshold is analyzed via a test in this paper to identify the proper threshold that minimizes the usage of the EPANET simulator and maximizes the usage of the Gaussian agent model while ensuring algorithm positioning precision.

As shown in Fig. 6, each threshold is tested for 15 times. Fig. 6(a) represents the average usage of the EPANET simulator for different thresholds. As the threshold increases, the usage of EPANET gradually decreases. Correspondingly, as shown in Fig. 6(b), the average time cost of the algorithm gradually decreases as the threshold increases. In contrast, as shown in Fig. 6(c), as the threshold increases, the average fitness of the algorithm continuously increases (objective function in this paper is to identify the minimal value). This means the algorithm accuracy continuously deteriorates as the threshold increases. When the threshold is 0.1 or 0.2, the fitness is similar. However, when the threshold is 0.2, it takes less time. Moreover, as shown in Fig. 6(d), when the threshold is 0.1 or 0.2, the algorithm positioning accuracy is similar. Therefore, the test setup in this paper is 3σ/μ<0.2.

3.2.2. Determination of *x* for the linear probability analysis *P* = *t*/*x*

Because there is no explicit method to determine a relatively sensitive point, this paper proposes a linear

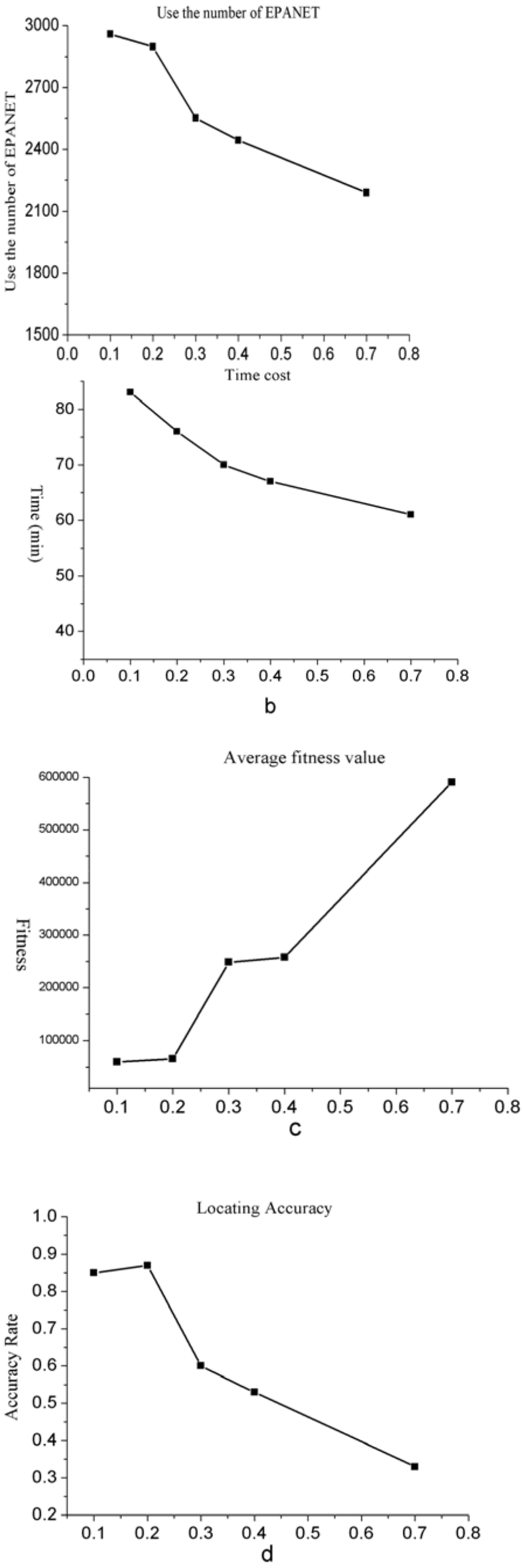


Fig. 6. Trigger coefficient 3σ/μ threshold analysis.

probability-based EPANET simulator usage based on the characteristics of pollution source positioning problem convergence. This linearity factor is another direct factor that affects EPANET simulator usage. When *x* is larger, the EPANET simulator usage probability is lower, and the time cost is lower. On the other hand, when *x* is smaller, the EPANET simulator is used more frequently, and the algorithm is more accurate, but it takes a longer time. Therefore, it is critical for the algorithm to select a proper value for *x* that minimizes the usage of the EPANET simulator while ensuring algorithm precision.

As shown in Fig.7, *x* is set to different values and each value is tested for 20 times. Test diagrams show the average EPANET simulator usage, average time cost and average fitness of the algorithm with different *x* values. Fig. 7(a) represents the average EPANET simulator usage. As *x* increases, the EPANET simulator usage decreases, and the time cost correspondingly decreases. On the other hand, as shown in Fig. 7(b), the fitness increases gradually. When *x* is 100 or 200, the fitness is similar. To minimize the EPANET simulator usage while ensuring the algorithm accuracy, *x* is set to 200 in this paper. When *x* = 200, the fitness is significantly less than when *x* = 300.

To verify the effectiveness of the setup for two parameters, the pollutant concentration curve measured by water quality sensors corresponding to the algorithm solution when 3σ/μ<0.2, *x* = 200 is shown in Fig. 8. The curve measured by the water quality sensor at position 7626 in the solution that essentially matches the pollution source concentration curve measured by the actual water quality sensor. This means that the solution is feasible and effective.

3.3. Algorithm performance analysis

The expensive optimization algorithm replaces the original time-consuming model with an agent model to reduce the time cost. In this paper, the Gaussian agent model-based microhabitat genetic algorithm is employed to solve the expensive optimization problem. The Gaussian agent model replaces the EPANET simulator in the fitness calculation. To maintain the pollution source positioning accuracy and minimize the time cost, a Gaussian agent model usage strategy is proposed in this paper. In this section, a comparative analysis of the created Gaussian agent model versus EPANET is conducted using a simulation test. Next, numerous tests are performed to prove the effectiveness of the Gaussian agent model-based expensive optimization algorithm proposed in this paper.

During algorithm convergence, frequent Gaussian agent model usage significantly reduces the time cost. Fig. 9 shows

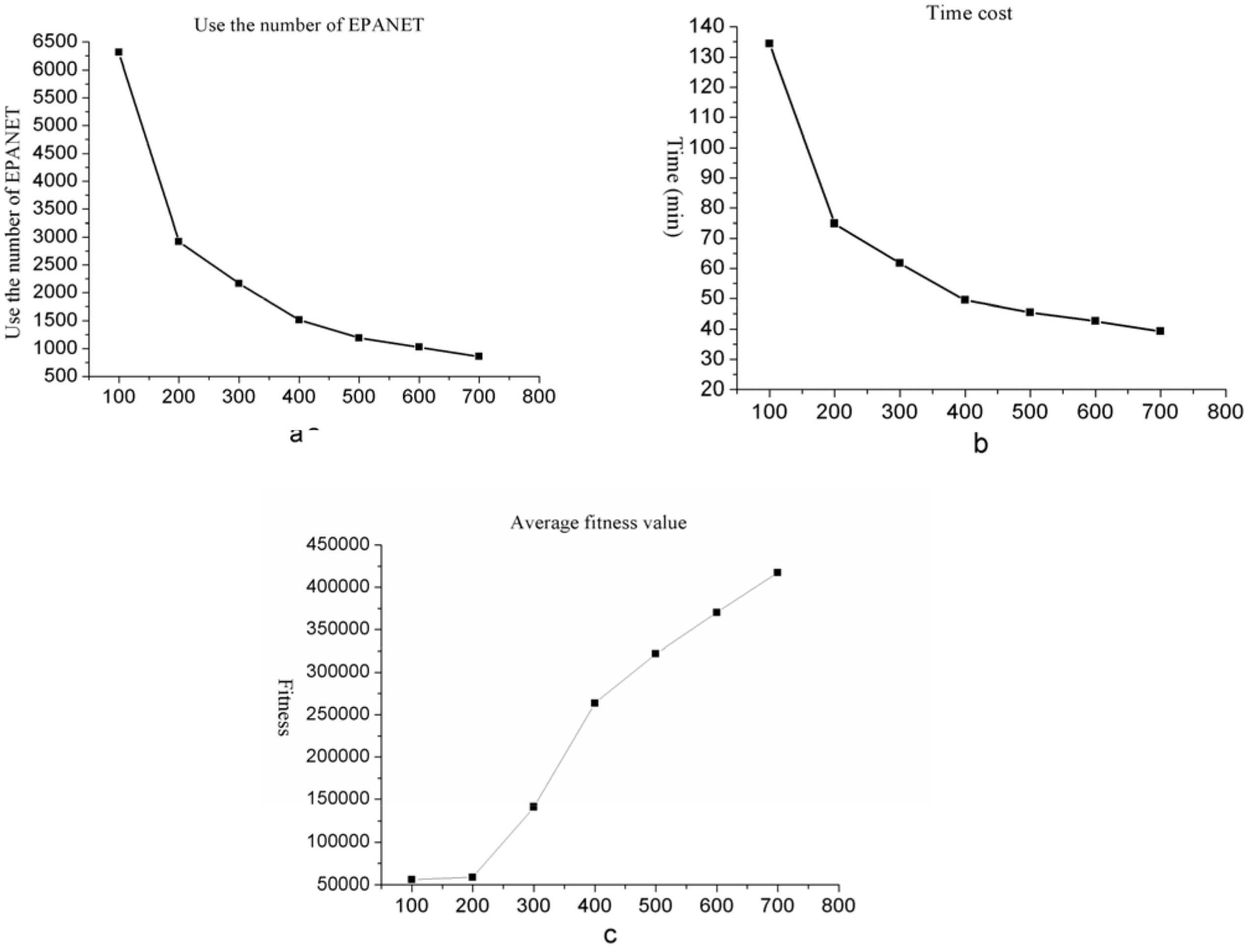


Fig. 7. Analysis diagrams for various values of *x*.

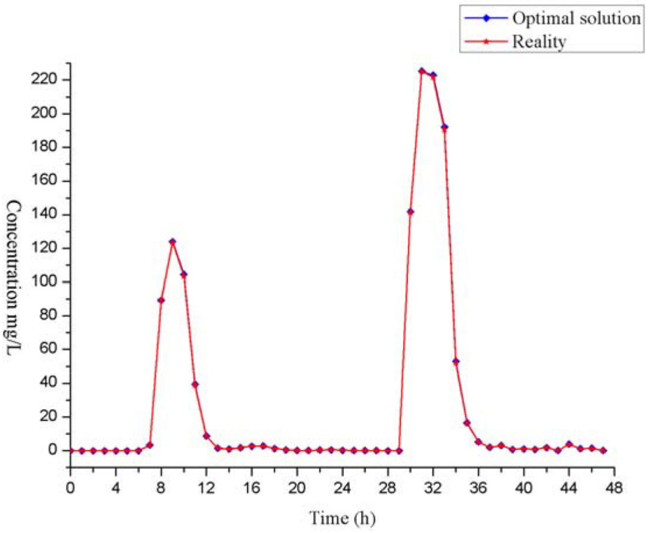


Fig. 8. Concentration measured by sensor 7626.

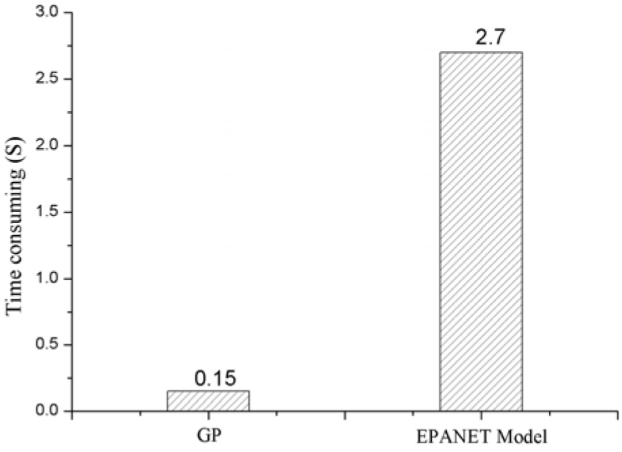


Fig. 9. Time consumption of a single evaluation.

a comparison of the individual fitness calculation time for a single usage of the Gaussian agent model forecast versus the case with a single usage of the EPANET simulator. A single usage of the Gaussian agent model saves a significant amount of time compared with a single usage of EPANET simulator. This proves that frequent Gaussian agent model usage significantly reduces the algorithm time cost.

The algorithm proposed in this paper is compared with the algorithm without using the Gaussian agent model. Table 2 lists the calculation results of the algorithm proposed in this paper versus the result of the algorithm without the Gaussian agent model. Although the optimal solution of the algorithm without the agent model has a lower fitness, this will not affect the pollution source positioning accuracy. The only error in the positioning result is the pollutant injection quantity. Both algorithms located the pollution source at 4528 and the pollutant injection quantity vectors are similar. On the other hand, Fig. 10 shows the concentration curve measured by the sensor at position 7626. It shows that the pollutant concentration curve predicted by the optimal solution matches the curve measured by the actual sensor 7626. This means the solutions of both algorithms are feasible and effective because they can accurately locate the actual pollution source.

Table 2
Test results

	Time cost	EPANET usage	Optimal solution fitness
Algorithm proposed in this paper	88 min	3,550 times	17.81
Algorithm without the Gaussian agent model	319 min	16,555 times	7.8

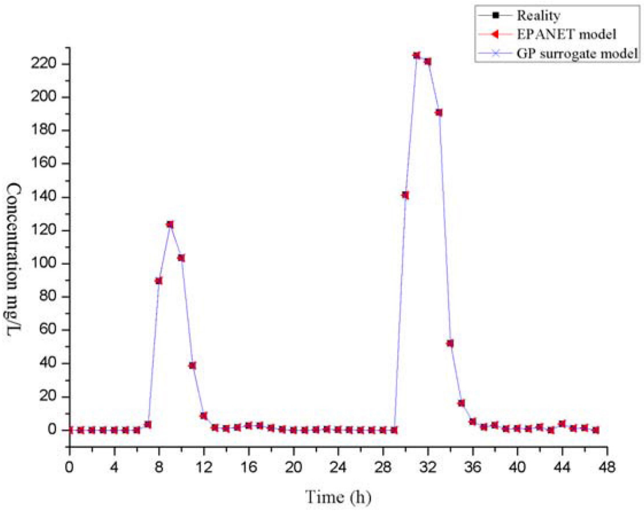


Fig. 10. Concentration curve measured by sensor 7626.

When both algorithms are capable of accurate positioning, the EPANET usage and time consumption of the Gaussian agent model-based algorithm versus the algorithm without the Gaussian agent model are shown in Figs. 11 and 12. To achieve an identical result, the usage of EPANET in the Gaussian agent model-based algorithm is reduced significantly by 2/3, which significantly reduces the algorithm time consumption and improves the algorithm efficiency. This proves the effectiveness and efficiency of the algorithm proposed in this paper.

4. Conclusions

The water supply network pollution source positioning problem is a cross discipline problem that involves environmental science and computer science. In this paper, the pollution source positioning problem is converted into a function optimization problem using the simulation–optimization model. Additionally, the pollution source positioning problem is further converted to an expensive optimization problem to reduce the computing cost to find a solution. To solve this problem, this paper employs the Gaussian agent model-based expensive optimization algorithm. Based on the characteristics of the water supply network, a sub-model is created for each node via the Gaussian random process. A proper Gaussian agent model usage strategy is proposed to maximize the usage of the Gaussian agent model while

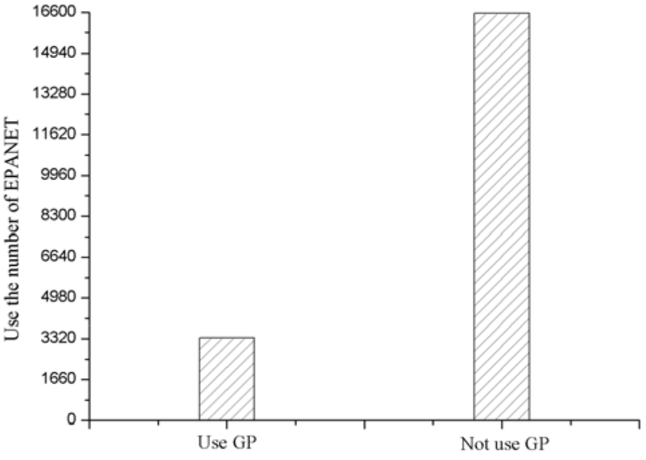


Fig. 11. EPANET usage for the Gaussian model-based algorithm and the algorithm without the Gaussian model to achieve an identical result.

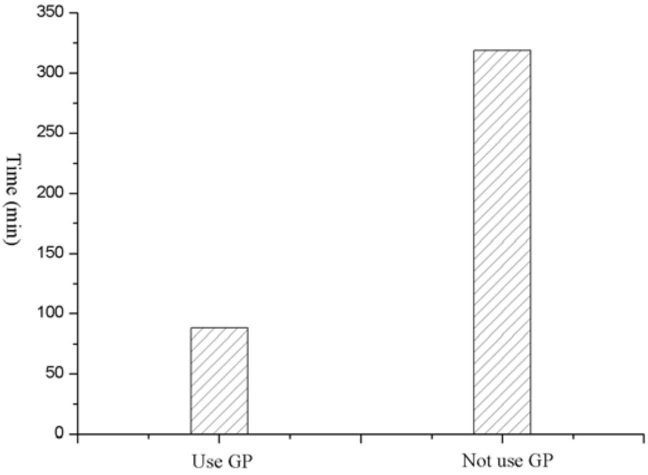


Fig. 12. Time consumption of the Gaussian agent model-based algorithm and the algorithm without the Gaussian agent model to achieve an identical result.

ensuring the algorithm positioning precision. Finally, the simulation test analysis proves the effectiveness and efficiency of the algorithm proposed in this paper.

When investigating a pollution source positioning problem, if the number of nodes in an urban water supply network exceeds 1,000, the user water demand varies in real time and this problem is abstracted as a dynamic, expensive and multimodal function optimization problem. This requires a solution for a dynamic multimodal expensive optimization problem, which is the plan for subsequent research work.

Acknowledgments

This paper is supported by Natural Science Foundation of China (No. 61673354, 61573324) and the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) and the State Key Lab of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology (DMETKF2018020).

References

[1] A. Ostfeld, J.G. Uber, E. Salomons, J.W. Berry, W.E. Hart, The battle of the water sensor networks (BWSN): a design challenge for engineers and algorithms, J. Water Resour. Plann. Manage., 134 (2008) 556–568.

[2] J.B. Guan, M.M. Aral, M.L. Maslia, W.M. Grayman, Identification of contaminant sources in water distribution systems using simulation-optimization method: case study, J. Water Resour. Plann. Manage., 132 (2006) 252–262.

[3] A. Preis, A. Ostfeld, Genetic algorithm for contaminant source characterization using imperfect sensors, Civil Eng. Environ. Syst., 25 (2008) 29–39.

[4] E.M. Zechman, S.R. Ranjithan, Evolutionary computation-based methods for characterizing contaminant sources in a water distribution system, J. Water Resour. Plann. Manage., 135 (2009) 334–343.

[5] L. Mou, W. Menglin, L. Jie, D. Shen, Notice of Retraction Investigation on Backward Tracking of Contamination Sources in Water Supply Systems—Case Study, 2nd Conference on Environmental Science and Information Application Technology, Wuhan, 2010, pp. 484–487.

[6] L. Liu, S.R. Ranjithan, G. Mahinthakumar, Contamination source identification in water distribution systems using an adaptive dynamic optimization procedure, J. Water Resour. Plann. Manage., 137 (2010) 183–192.

[7] M. Jha, B. Datta, Application of dedicated monitoring—network design for unknown pollutant-source identification based on dynamic time warping, J. Water Resour. Plann. Manage., 141 (2015) 04015022.

[8] C.Y. Hu, J. Zhao, X.S. Yan, D.Z. Zeng, A MapReduce based Parallel Niche Genetic Algorithm for contaminant source identification in water distribution network, Ad Hoc Networks, 35 (2015) 116–126.

[9] X.S. Yan, J. Zhao, C.Y. Hu, Q.H. Wu, Contaminant source identification in water distribution network based on hybrid encoding, J. Comput. Methods Sci. Eng., 16 (2016) 379–390.

[10] X.S. Yan, W.Y. Gong, Q.H. Wu, Contaminant source identification of water distribution networks using cultural algorithm, Concurrency Comput. Pract. Exper., 29 (2017) e4230. doi: 10.1002/cpe.4230.

[11] X.S. Yan, J. Zhao, C.Y. Hu, D.Z. Zeng, Multimodal optimization problem in contamination source determination of water supply networks, Swarm Evol. Comput., (2017). doi: doi.org/10.1016/j.swevo.2017.05.010.

[12] X.S. Yan, Z.X. Zhu, T. Li, Pollution source localization in an urban water supply network based on dynamic water demand, Environ. Sci. Pollut. Res., (2017). doi: https://doi.org/10.1007/s11356-017-0516-y.

[13] X.S. Yan, J. Sun, C.Y. Hu, Research on contaminant sources identification of uncertainty water demand using genetic algorithm, Cluster Comput., 20 (2017) 1007–1016.

[14] A. Rasekh, K. Brumbelow, A dynamic simulation–optimization model for adaptive management of urban water distribution system contamination threats, Appl. Soft Comput., 32 (2015) 59–71.

[15] D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions, J. Global Optim., 13 (1998) 455–492.

[16] Y.C. Jin, M. Olhofer, B. Sendhoff, A framework for evolutionary optimization with approximate fitness functions, IEEE Trans. Evol. Comput., 6 (2002) 481–494.

[17] R.G. Regis, C.A. Shoemaker, Local function approximation in evolutionary algorithms for the optimization of costly functions, IEEE Trans. Evol. Comput., 8 (2004) 490–505.

[18] Z.Z. Zhou, Y.S. Ong, P.B. Nair, A.J. Keane, K.Y. Lum, Combining global and local surrogate models to accelerate evolutionary optimization, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., 37 (2007) 66–76.

[19] I. Paenke, J. Branke, Y.C. Jin, Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation, IEEE Trans. Evol. Comput., 10 (2006) 405–420.

[20] J.E. Fieldsend, R.M. Everson, On the Efficient Use of Uncertainty when Performing Expensive ROC Optimization, IEEE Congress on Evolutionary Computation, Hong Kong, 2008, pp. 3984–3991.

[21] W.D. Liu, Q.F. Zhang, E. Tsang, Fuzzy Clustering Based Gaussian Process Model for Large Training Set and Its Application in Expensive Evolutionary Optimization, IEEE Congress on Evolutionary Computation, Trondheim, Norway, 2009, pp. 2411–2415.

[22] S. Jeong, S. Obayashi, Efficient Global Optimization (EGO) for Multi-Objective Problem and Data Mining, IEEE Congress on Evolutionary Computation, Edinburgh, 2005, pp. 2138–2145.

[23] A.J. Keane, Statistical improvement criteria for use in multiobjective design optimization, AIAA J., 44 (2006), 879–891.

[24] W. Ponweiser, T. Wagner, D. Biermann, M. Vincze, Multiobjective Optimization on a Limited Budget of Evaluations Using Model-Assisted, International Conference on Parallel Problem Solving from Nature PPSN X, Dortmund, 2008, pp. 784–794.

[25] A. Zhou A, Q. F. Zhang, Y. C. Jin, Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm, IEEE Trans. Evol. Comput., 13 (2009) 1167–1189.

[26] Y. Tenne, C.K. Goh, Computational Intelligence in Expensive Optimization Problems, Springer, Berlin, Heidelberg, 2012, p. 5.

[27] C.T. Luo, S.L. Zhang, C. Wang, Z.L. Jiang, A meta model-assisted evolutionary algorithm for expensive optimization, J. Comput. Appl. Math., 236 (2011) 759–764.

[28] H.K. Singh, A. Isaacs, T. Ray, A Hybrid Surrogate Based Algorithm (HSBA) to Solve Computationally Expensive Optimization Problems, IEEE Congress on Evolutionary Computation, Wuhan, 2014, pp. 1069–1075.

[29] B. Liu, Q.F. Zhang, G. Gielen, A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems, IEEE Trans. Evol. Comput., 18 (2014) 180–192.

[30] K.S. Bhattacharjee, T. Ray, An Evolutionary Algorithm with Classifier Guided Constraint Evaluation Strategy for Computationally Expensive Optimization Problems, Australasian Joint Conference on Artificial Intelligence, Springer International Publishing, 2015, pp. 49–62.

[31] C.L. Sun, Y.C. Jin, R. Cheng, J.L. Ding, J.C. Zeng, Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems, IEEE Trans. Evol. Comput., 21 (2017) 644–660.

[32] Z.L. Liu, China's strategy for the development of renewable energies, Energy Sources Part B, 12 (2017) 971–975.

[33] H.L. Fu, X.J. Liu, Research on the phenomenon of Chinese residents' spiritual contagion for the reuse of recycled water based on SC-IAT, Water, 9 (2017) 846.

[34] Z.W. Feng, Q.B. Zhang, Q.F. Zhang, Q.G. Tang, T. Yang, Y. Ma, A multiobjective optimization based framework to balance the global exploration and local exploitation in expensive optimization, J. Global Optim., 61 (2015) 677–694.

[35] Q.X. Wei, X.F. Liu, Q. Huang, G.Z. Cheng, The comparison of selection methods in different genetic algorithms, J. Commun. Comput., (2008) 61–65.