Ad Hoc Networks 000 (2015) 1–11

Contents lists available at ScienceDirect

Ad Hoc Networks

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

A MapReduce based Parallel Niche Genetic Algorithm for contaminant source identification in water distribution network

Chengyu Hu [a], Jing Zhao [a], Xuesong Yan [a], Deze Zeng [a,*], Song Guo [b]

[a] School of Computer Science, China University of Geosciences, Wuhan, China
[b] School of Computer Engineering, The University of Aizu, Aizuwakamatsu, Japan

ARTICLE INFO

Article history:
Received 17 December 2014
Revised 12 June 2015
Accepted 6 July 2015
Available online xxx

Keywords:
Contaminant source identification
Water distribution network
Multi-model optimization
MapReduce framework
Niche Genetic Algorithm

ABSTRACT

In recent years, water pollution incidents happen frequently, causing serious disasters and society impact. It is advocated that water quality monitoring sensors shall be deployed in water distribution network to enable real-time pollution detection such that we can effectively detect the water pollution event to reduce the risk. Besides event detection, it is also important to identify the contaminant source for depollution actions. But how to use the information derived from the monitoring sensors to identify the contaminant source is a non-trivial task. Contamination source identification problem is characterized by its extremely high computation complexity, uncertainty and non-uniqueness of the solution in a large-scale water distribution network with dynamic water demands. To tackle this issue, we develop a MapReduce based Parallel Niche Genetic Algorithm (MR-PNGA) that is not only able to achieve high identification accuracy but also to explore the cloud resources for performance improvement. The accuracy and efficiency of MR-PNGA is extensively validated on an 8-server cluster.

1. Introduction

Recently, accidental and intentional contamination in a water distribution network occurs frequently, causing significant economic losses and bad social influence. In order to prevent such contamination accidents and alleviate the losses, water quality sensors are installed in water distribution network to monitor contaminant event. According to health supervision work report by Ministry of Public Health of China in 2012, Chinese government has set 28600 monitoring stations including million of water quality sensors all over the country. Many different kinds of contaminant events can be detected by the deployed sensor networks. However, far beyond simple event detection, we all also need to identify the contaminant source for further depollution actions. When a contamination event is detected by sensor networks, a fast and accurate source identification is critical for the municipal authorities to make prompt reactions for improving the safety of citizens. Contamination source identification (CSI) shall be able to expose (1) the location of the contaminant, (2) the type of the contamination, and (3) concentrations of the contamination and its distribution throughout the water distribution network. How to derive these information using the data collected from the water quality sensors has become an important but also a challenging task for many reasons. For example, the source injections can originate at any point across the whole system at any time and the water demand of consumers also vary over time [1].

Much pioneering work has been devoted to addressing the CSI problem by various methods such as particle backtracking [2,3], machine learning [4], data mining [5] and so on. Among them, simulation-optimization [6] that couples simulation (e.g., by EPANET[1]) with heuristic optimization

* Corresponding author. Tel.: +81 15002774210.
E-mail address: deze@cug.edu.cn (D. Zeng).

[1] http://epanet.de/

http://dx.doi.org/10.1016/j.adhoc.2015.07.011
1570-8705/

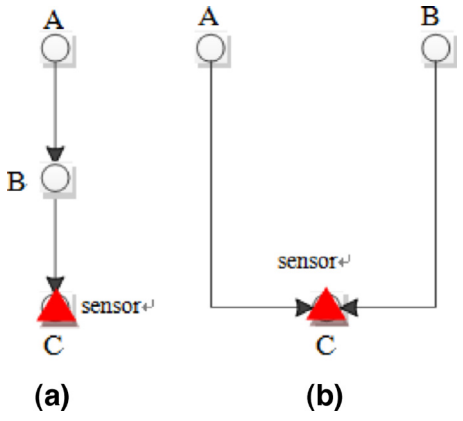


Fig. 1. Existence of multiple possible contaminant sources.

algorithm (e.g., evolutionary algorithms) has been regarded as an efficient and practical way to address the CSI problem for its accuracy and robustness [7]. However, all the existing studies assume that there is a single solution to a CSI problem while we notice that there could be multiple solutions (i.e., non-uniqueness), where the exact contaminant source is included. As shown in Fig. 1, although the sensor at *C* detect a contaminant event in both cases, it can only derive that *A* and *B* are the possible sources but cannot exactly tell which one is the true source. Therefore, searching and listing all the possible sources, i.e., multi-modal optimization (MMO), and filtering them to find out the true one is necessary. Niche Genetic Algorithm that is able to preserve the diversity of population and locate multiple optimal solutions has been regarded as a promising MMO method [8].

On the other hand, although with high accuracy, simulation-optimization methods are not quite efficient as they are often associated with high computation cost, especially when it is applied to solving some typical engineering problems where the function evaluation requires simulation of some complex numerical models. CSI problem falls into such category as it requires computation-intensive function evaluation using hydraulics computation with a big volume of data. For example, the Battle of the Water Sensor Networks (BWSN2) [9] consists of 12,523 nodes. Assume that the sampling interval of each sensor is set as 10 min. If each sampling report has 12 bytes, totally we will get 1.21 gigabytes for 72 h monitoring. Larger network size or higher sampling rate even implies bigger data. Furthermore, it is reported that simulating one scenario by EPANET2.0 requires 4 s on a Pentium 4.3 GHz computer [10]. While a CSI problem demands real-time or near-real-time solution, it is significant to find a way to accelerate simulation-optimization based method to CSI problem. In other words, fast and accurate solution to CSI problem is essentially required.

To evolutionary algorithms based optimization methods, it is possible to explore their parallelism for performance acceleration, e.g., [11,12]. Furthermore, cloud computing emerges as a new and promising parallel computing paradigm with a pool of rich computation resources. To explore the bulk cloud resources, many cloud computing paradigms have been proposed and adopted in practice. One of the most famous ones is known as Hadoop, which enables MapReduce computing paradigm [13]. To improve the efficiency of solving CSI problem, it is natural to wonder whether we can apply MapReduce to simulation-optimization based method. After collecting and storing water quality sensor data in a data center, MapReduce paradigm is promising to explore both the cloud resources in the data center and the parallelism in the simulation-optimization based method.

In this paper, we are motivated to address the non-uniqueness and efficiency of solving CSI problem. The main contributions are as follows.

1. To our best knowledge, we are the first to formulate CSI into an MMO problem and formally prove its non-uniqueness.
2. We propose a MapReduce based Parallel Niche Genetic Algorithm (MR-PNGA), which is able to find out multiple possible contaminant sources fast and accurately.
3. We practically implement MR-PNGA on a 8-server cluster. Experiment results show that MR-PNGA indeed can detect more contaminant sources efficiently.

The remainder of this paper is organized as follows. Section 2 introduces existing related work to CSI problem and multi-modal optimization. Section 3 formulates the CSI problem into an MMO problem and shows its non-uniqueness. Section 4 proposes our MR-PNGA. Section 5 gives the performance evaluation and analysis. Finally, Section 6 concludes this work.

2. Related work

2.1. CSI problem

To solve a CSI problem is to derive the locations of the contaminant sources, the contaminant mass loading profile and the start time of injection at each contaminant source, according to the data collected by sensors. CSI is usually regarded as an inverse problem to derive the unknown input (e.g., injections) based on the partially known output (e.g., contaminant information). Due to the high importance of solving CSI problem, many efforts have been devoted to addressing it. Existing methods can be generally classified into three categories.

The first category is particle back-tracing, which directly treats the CSI problem as an "inverse problem" to find out one contaminate source [14]. Shang et al. [14] consider the pollutant as a particle and trace back from the nodes where contamination is detected to the source in reverse. Laird et al. [2] use tracking and simulation algorithm to identify the contaminant source. Sanctis et al. [3] use the particle trace algorithm to identify possible contaminant sources by comparing the consistency of the nodes that are not in conformity by forward and reverse comparison, respectively. Due to its complexity, particle backtracking can be only applied to medium and small size of water distribution network.

The second one is machine learning based to yield a probabilistic contaminant source characterization using both sensor data and system stochastic processes. Huang and McBean [5] propose a data mining based CSI solution. Perelman and Ostfeld [15] divide water distribution network into clusters according to the flow and connectivity, and calculate the probabilities of contaminant sources by Bayesian algorithm.

Please cite this article as: C. Hu et al., A MapReduce based Parallel Niche Genetic Algorithm for contaminant source identification in water distribution network, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.011

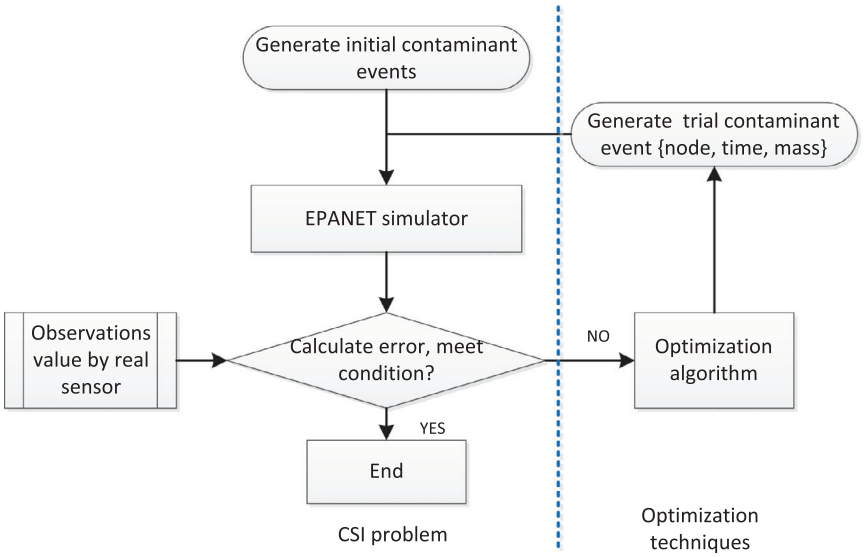


Fig. 2. Workflow of simulation-optimization for CSI problem.

Recently, Wang and Jin [16] propose a random sampling data through the Monte Carlo Markov Chain (MCMC) sampling method and also calculate the probabilities by Bayesian algorithm. Wang and Harrison [4] further combine MCMC and vector regression to improve the algorithm efficiency. Although machine learning is able to list all the contaminant source probabilities, it is with low accuracy but high computation complexity, not applicable to large-scale water distribution network.

The third category includes simulation-optimization based algorithms. Basically, simulation-optimization is a trial-and-error approach. The main procedures are depicted in Fig. 2. As shown in the figure, a searching procedure (i.e., optimization) is coupled with a water distribution simulation (e.g., EPANET based simulation) to evaluate the search results. Optimization technique based searching is to accelerate the searching procedure for fast source identification. Guan et al. [6] first demonstrate simulation-optimization approach's applicability to CSI by incorporating the reduced gradient method. Liu et al. [17] put forward adaptive dynamic optimization technique based on evolutionary algorithm to search the contaminant source. Simulation-optimization based algorithms are with higher accuracy compared to the other two categories. However, to our best knowledge, none of existing studies address the non-uniqueness of CSI problem to give out multiple potential and alternative optimal solutions.

2.2. MMO problem

Knowledge of multiple candidate solutions to many engineering optimization tasks is helpful and MMO to find multiple solutions in one run is thus advocated [18]. Many evolutionary algorithms based methods for MMO problems have been presented in the literature. Li [19] propose a Species-based Particle Swarm Optimization (SPSO) that has a niche radius defined beforehand in order to determine the size of species (i.e., dominant particles). Qu et al. [20] present a Distance-Based Locally Informed Particle Swarm (LIPS), which adopts the local information information from its nearest neighborhood to guide the search of the particles. Qu et al. [21] suggest a differential evolution with neighborhood mutation strategy, where each individual is evolved toward its nearest optimal point. Later, Stoesn et al. [22] differentiate species and save the diversity of population in species conservation version 2 (TSC2) with the consideration of both seed selection and seed conservation.

Evolutionary computing based algorithms to MMO are still notorious for their low efficiency in solving time, it is essential to find a way that can improve their performance for practical engineering adoption.

3. System model and problem statement

3.1. System model

In this paper, we consider a water distribution network as shown in Fig 3. The network is described using a graph $G = (V, E)$, where E is a set of edges representing pipes and V is a set of vertex locating in the intersections of pipes. A vertex could be a source, reservoir, tank or sink (i.e., consumer). A pipe connecting two vertices v_i and v_j indicates certain flow could existing between them. The flow pattern (i.e., flow directions) has significant effect on the contamination diffusion. In this paper, we consider a fixed flow pattern, where the directions of the flows on E are known.

There are r monitoring sensors located at the nodes in the network. Generally, the number of sensors is much smaller than the size of the nodes in a water distribution network, i.e., $r \ll n$. Upon a contaminant event, a sensor can detect both its type and concentration. Further, it is able to store the history data for a few days, i.e., contaminant concentration profile. We assume that the sensors can upload the collected information to the data center as soon as they detect the contaminant if the concentration of contaminant is greater or equal to a threshold. A contaminant injection may happen

Please cite this article as: C. Hu et al., A MapReduce based Parallel Niche Genetic Algorithm for contaminant source identification in water distribution network, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.011

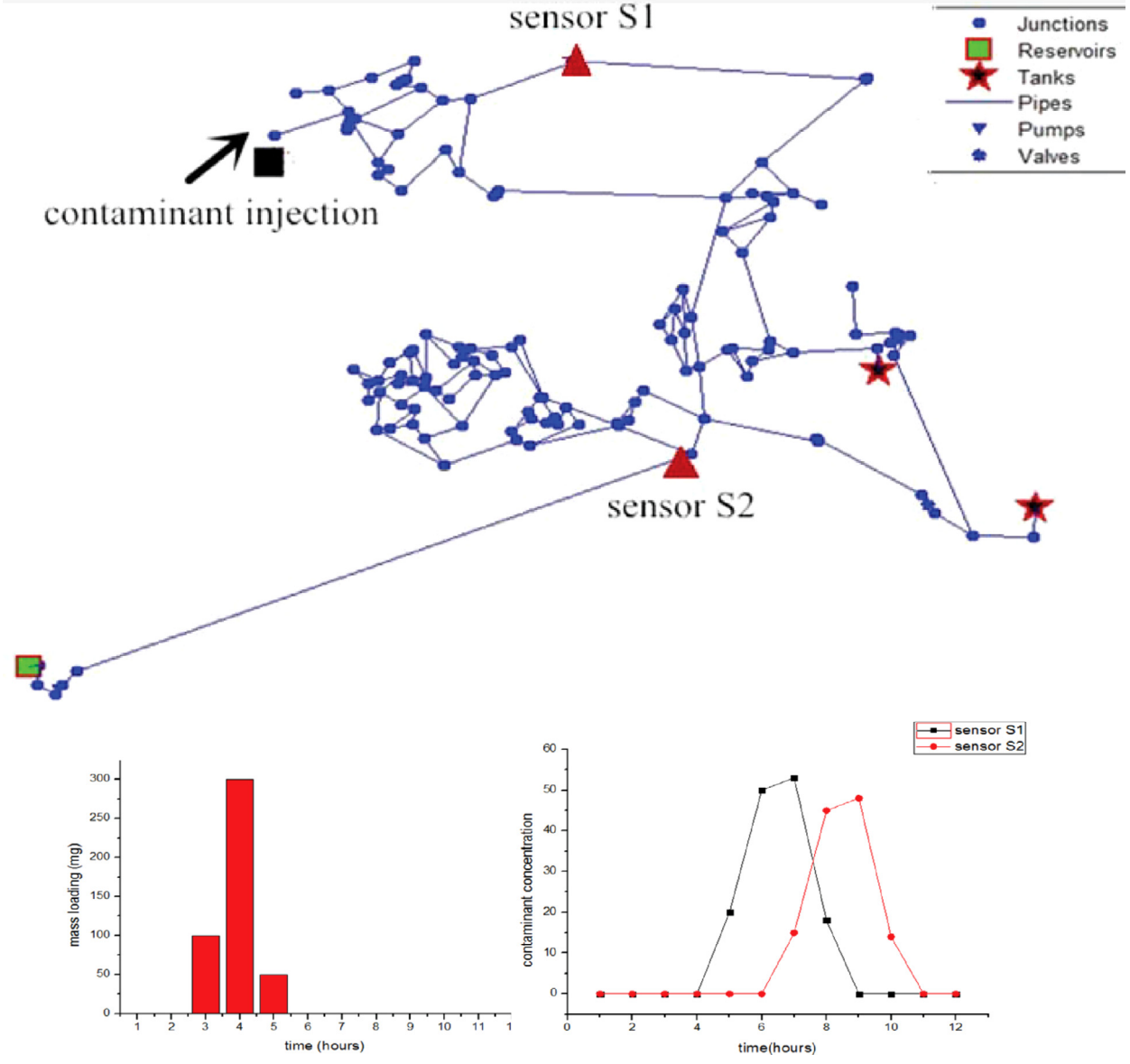


Fig. 3. System model, contaminant mass loading profile and contaminant concentration profile.

at any node in the network. After the injection, the contaminant will gradually dilute with the water flows. In this paper, we assume that there is only one attack. If an attack on node v_i happens, the monitoring sensors at nodes $v_i \neq v_j$, $\forall v_j \in V$ shall be able to detect the contaminant type and its concentration if there is a path from v_i to v_j for which all edges have "positive" flow. For example, in Fig. 3 we deliberately inject contaminant at node 34 with different mass loading during from 3:00 to 5:00. Sensors $S1$ and $S2$ have different concentration profiles due to different contaminant diffusion paths.

3.2. Problem statement

The goal of CSI is to find the locations of the contaminant source and the start time of injection according to the contaminant concentration profiles. For the example shown in Fig. 3, our goal is to find the injection node (i.e., 34) based on the observed concentration profiles of sensors $S1$ and $S2$. If an attack $\delta_i(m, t')$ with mass loading m happens on node i at time t', we can obtain the observed concentration profile of sensor v_j, $\forall v_j \in V$, $v_j \neq v_i$ at the time $t \in [T_0, T_s]$, $t \geq t'$ as $c_j^*(t)$, where T_0 and T_s are the starting and end time of the profile. On the other hand, we can derive the concentration profiles $c_j(t, \delta_{i'}(m, t'))$ by hydraulics computation for a given attack $\delta_{i'}(m, t')$ and water flow pattern. Obviously, different attacks shall have different concentration profiles determined by the corresponding $\delta_i(m, t')$. Therefore, our goal to identify the contaminant source is equivalent to find out $\delta_{i'}(m, t')$ that can lead to concentration profiles $c_j(t, \delta_{i'}(m, t'))$ best matching the observed one $c_j^*(t)$, i.e,

Please cite this article as: C. Hu et al., A MapReduce based Parallel Niche Genetic Algorithm for contaminant source identification in water distribution network, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.011

$$\min: f = \sum_{j=1}^{r} \sum_{t=T_0}^{T_s} |c_j(t, \delta_{i'}(m, t')) - c_j^*(t)|, \quad (1)$$

where f denotes the fitness value and $|\cdot|$ is absolute function.

3.3. Proof of non-uniqueness

Intuitively, due to the ill-posed nature of CSI problems, a set of solutions may match the observed data due to lack of sufficient data or the presence of error in the data. We call this issue as non-uniqueness, which shall be addressed properly. In this section, we formally prove the non-uniqueness of CSI and show that the CSI problem is essentially an MMO problem by linear system theory. To this end, we first make three definitions related to the proof.

Definition 1. (Local minimum) Let $\min f: X \to R$. If $\forall x_l^*, \exists \varepsilon > 0 : \forall x \in X, |x - x_l^*| < \varepsilon$ satisfies $f(x_l^*) \le f(x)$, x_l^* is the local extreme value point and (x_l^*) be local minimum value.

Definition 2. (Global minimum) Let $\min f: X \to R$. If $\forall x \in X$ satisfies $f(x_l^*) \le f(x)$, $f(x_l^*)$ is the global minimum.

Definition 3. (MMO problem) Let $\min f: X \to R$, where X be the set of solutions. If X contains more than one local extremum points or global extreme value point, f is a multimodal function and solving f is an MMO problem.

Theorem 1. *The CSI problem in water distribution networks is an MMO problem.*

Proof. As [23] states, hydraulic water transport process of water distribution network can be described as a linear input/output system, i.e.,

$$c_i(t) = \sum_{j=1}^{n} \sum_{k=1}^{t} \theta_{ij}^k(t) u_j^k, \quad (2)$$

where $c_i(t)$ is the concentration of contaminant on node i at time t, u_j^k is contaminant mass loading on node j at time k and $\theta_{ij}^k(t)$ is the response coefficient of contaminant concentration mass on node i injected from node j at time t.

All the nodes are with the same probability to be attacked, i.e., as contaminant source. Given a set of r monitoring sensors in a water distribution network with n nodes. At time t, the cumulative concentration matrix $C_{r\times 1}$ of all the monitoring sensors can be expressed as

$$C_{r\times 1} = A_{r\times n} X_{n\times 1}, \quad (3)$$

where $A_{r\times n}$ is response coefficients matrix in size $r \times n$.

By (1), we have

$$\min_{x\subseteq X} f = \min_{x\subseteq X} (C^* - C)^2 = \min_{x\subseteq X} \left\{ \sum_{i=1}^{r} [c_i^*(t) - (Ax)_i(t)] \right\}^2 \quad (4)$$

Essentially, solving a CSI problem is to equivalent to finding a set of decision variables $X_{n\times 1}$ that make the observed concentration of r sensors $C_{r\times 1}^*$ as close as possible to the derived concentration $C_{r\times 1}$. Because the optimization function (4) is nonnegative function, it is clear that the optimal point of (1) is the solution to

$$\|C^* - C\| = AX - C^* = 0, \quad (5)$$

which is an MMO problem if it has multiple solutions.

We can divide matrix A and X into basis part and non-basis part with r and $n - r$ vectors, respectively. Thus, (5) can be rewritten as

$$C^* = A_B X_B + A'_B X'_B. \quad (6)$$

Solving (6) to derive X_B, we have

$$X_B = A_B^{-1} C^* - A_B^{-1} A'_B X'_B, \quad (7)$$

which shows that the solution is composed of the base and the non-base vector. As the non-basis vectors are non-uniqueness, (5) shall correspondingly have more than one solutions. Therefore, we can conclude that the CSI problem is an MMO problem. □

4. Algorithm design

4.1. Prerequisite knowledge

According to our problem statement, it is shown that deriving the concentration profiles by hydraulics computation is critical to solving the CSI problem. Fortunately, a set of tools have been proposed to address this issue. Among them, EPANET has been widely adopted. Thus, we can incorporate it into simulation-optimization procedures to evaluate the fitness value. In this section, we briefly introduce the main concept of EPANET based simulation-optimization method for CSI problem as prerequisites. The work flow is as follows. A number of trial solutions to the CSI problem is first randomly generated. Then, EPANET simulation is conducted to evaluate the fitness value of the trial solutions. If the fitness value is not satisfied enough, the trial solutions will be readjusted and the fitness value will be also reevaluated by EPANET. Such routine repeats until a predetermined fitness value is achieved.

4.2. MapReduce based Parallel Niche Genetic Algorithm

In this section, we propose our simulation-optimization algorithm for CSI problem. In particular, we apply niche genetic algorithm as the optimization method. The search process niche genetic algorithm can run in parallel. This motivates us to rewrite the process in MapReduce paradigm to explore the bulk cloud resources to accelerate niche genetic algorithm and maintain the population diversity at the same time. Incorporating with EPANET for simulation, a MapReduce based Parallel Niche Genetic Algorithm (MR-PNGA) is proposed. An overview of MR-PNGA is shown as Fig. 4.

At first, the master node generates an initial population as input. The population is partitioned into n subgroups, i.e., $P(t) = \{P_1(t), P_2(t), \ldots, P_i(t)\}$, each of which comprises of m individuals. An individual represents a trial solution to CSI problem, including (1) node location and (2) injection mass loading. The evolution, i.e., reproduction, mutation, niche and selection, of each subgroup is then conducted on a mapper. In other words, a mapper is responsible for one subgroup and multiple mappers shall be generated for multiple subgroups. Intermediate results generated by the mappers are

Please cite this article as: C. Hu et al., A MapReduce based Parallel Niche Genetic Algorithm for contaminant source identification in water distribution network, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.011

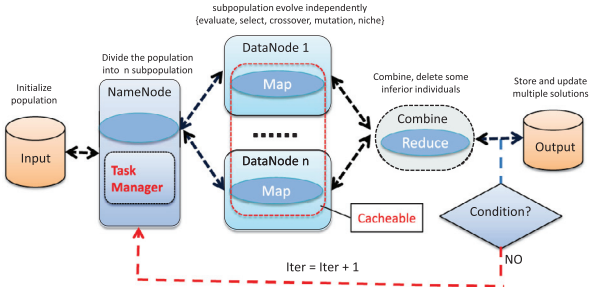


Fig. 4. Overview of MR-PNGA.

then shuffle to the reducers. A reducing task then evaluates the fitness values of the mappers and store the multiple optimal in an archive. Multiple iterations may execute to improves the population diversity and hence the probability to find the best-matching solutions. Finally, we can output the solution with the best fitness value.

The details of the mapper phase and reducer phase are shown in Algorithm 1 and Algorithm 2, respectively. Let us

Algorithm 1 Mapper phase (*Key*, *Vale*).

Require: $Key \leftarrow i; Value \leftarrow P_i(t)$.
Ensure: $Key \leftarrow P_i^n(t).node; Value \leftarrow P_i^n(t).mass$ and $P_i^n(t).fitness$.
1: **while** $t <$ *maximum iteration* **do**
2: $Evaluation : \Phi(f_i(t)) = evaluate[P_i(t)]$
3: $Sort : [P_i(t)] \leftarrow sort[P_i(t)]$
4: $Elite : P_i^k(t) \leftarrow savebeforek\,[P_i(t)]$
5: $Selection : P_i^s(t) \leftarrow select[P_i(t)]$
6: $Mutation : P_i^m(t) \leftarrow mutate[(P_i^s(t)]$
7: $Combine : P_i^*(t) \leftarrow P_i^m(t) + P_i^k(t)$
8: *Niche*
9: **for all** j *from* 0 *to* $m + k$ **do**
10: **if** $P_{ij}^*(t).fitness < \varepsilon$ **then**
11: **if** $P_{ij}^*(t).node$ *not included in* $P_i^n(t)$ **then**
12: $P_i^n(t) \leftarrow P_{ij}^*(t)$
13: **else**
14: **for all** q *from* 0 *to nichenumber* **do**
15: **if** $P_{ij}^*(t).node \equiv P_{iq}^n(t).node$ **then**
16: **if** $P_{ij}^*(t).fitness \leq P_{iq}^n(t).fitness$ **then**
17: $P_{iq}^n(t) \leftarrow P_{ij}^*(t)$
18: **else**
19: $penalty\ P_{ij}^*(t)$
20: **end if**
21: **end if**
22: **end for**
23: **end if**
24: **end if**
25: **end for**
26: **end while**

first check the mapper phase. Each mapper essentially is a local simulation-optimization process under given input (i.e., subgroup). A maximum local searching number, i.e., *maximum iteration*, is set, as shown in line 1. In each iteration, we first evaluate the fitness value of the initial population using EPANET in line 2. According to the niche genetic

Algorithm 2 Reducer phase (*Key*, *Value*).

Require: $Key \leftarrow P_i^n(t).node; Value \leftarrow P_i^n(t).mass$ and $P_i^n(t).fitness$.
Ensure: $Key \leftarrow P^N(t).node; Value \leftarrow P^N(t).mass$ and $P^N(t).fitness$.
1: **for all** *the same Key* **do**
2: **for all** i *from* 1 *to* n **do**
3: $P^N(t) \leftarrow min\{P_i^n(t).fitness\}$
4: **end for**
5: **end for**

Table 1
Cluster configuration.

Number of servers	8
Processor	2.0 GHZ
Memory	4 GB
Operation system	Ubuntu 12.10
Hadoop	Hadoop-0.21.0.

algorithm concept, the local population evolves by evaluation, selection, mutation and combination operations, as shown from lines 3–6. Lines 8–26 are niche operations. We treat each node as a niche. Within each niche, all the individuals are screened out to select the individual with the best fitness value. Lines 10 and 11 are the niche judgment conditions. If the individual fitness value is less than the threshold ε, then the individual is a possible solution. We define a niche radius, any two possible individuals closer together than the niche radius are considered to be in the same niche and thus share their fitness values. Line 19 use penalty mechanism to eliminate poor individuals.

The mapper phase is mainly responsible for overall evaluation across different subgroups from different mappers. All the individuals from mappers are combined according to the fitness value of each individual. The optimal solutions are stored in an archive.

The proposed MR-PNGA algorithm can be deployed in a data center residing with the data collected from water quality monitoring sensors. Once the contaminant sources are determined, depollution actions can be taken accordingly.

5. Experiment results and analysis

5.1. Experiment settings

To verify the performance of MR-PNGA, we practically implement MR-PNGA in a cluster with 8 servers. Each server equips with a 2.0 GHZ dual-core processor and 4 GB memory, as summarized in Table 1.

The default niche genetic algorithm parameter settings are listed in Table 2.

For the water distribution network, we use widely adopted simulator EPANET and consider a realistic network consisting of 129 nodes, 2 tanks, 2 sources and 170 pipes, as shown in Fig. 3. Hypothetical, but realistic and synthetic, attack scenarios are considered in the simulations. Two kinds of sensor layouts with different sensor deployments (i.e., the number of sensors and the sensor locations) are studied. Sensor data volume varies on different sensor quantities,

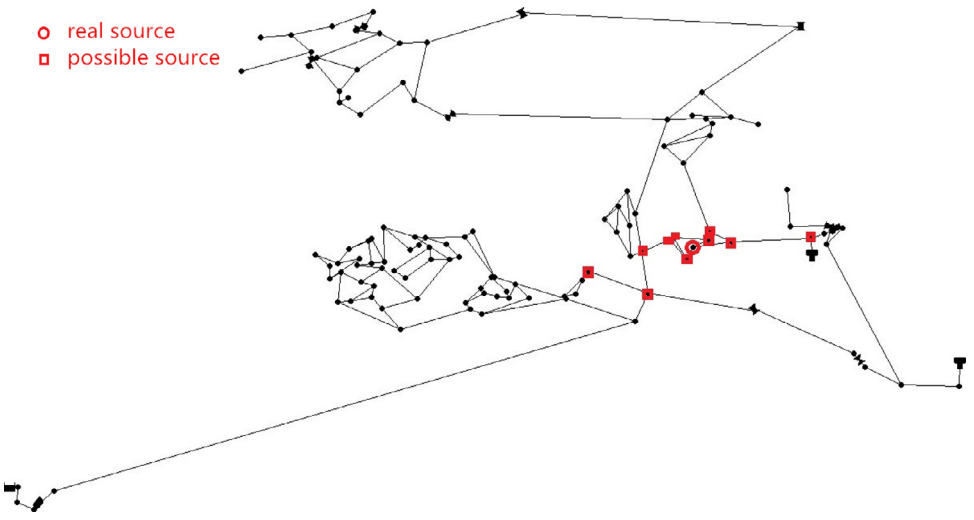


Fig. 5. The possible solutions found out by MR-PNGA under given attack at node 44.

Table 2
Niche genetic algorithm settings.

Crossover probability	0.95
Mutation probability	0.1
Elite strategy	10
Population size	80
Punish coefficient	1.5

Table 3
Multiple solutions by algorithm in one run.

Sensor location	Attack ({node, mg/L})	Results ({node, mg/L})
(10, 83)	(44, 300)	(22, 30.1) (23, 15.2) (24, 80.5) (25, 105.6) (26, 60.3) (27, 115.2) (28, 80.7) (29, 25.8) (43, 70.5) (44, 300) (53, 125.2)

sampling rates and total simulation time. The network is default simulated hourly over 60 h and is assumed to be steady within each hour. The transport of a nonreactive contaminant is simulated in an interval of 10 min, i.e., generating concentration profiles at sensors in a 10 min increment. An attack with contaminant mass loading of 300 mg/L is introduced into the network at node 44 at the beginning of the simulation.

5.2. Accuracy evaluation

To test and evaluate the accuracy of MR-PNGA, three different scenarios in three case studies are considered.

- Scenario 1: the location of sensors and contaminant source are fixed.
- Scenario 2: the location of sensors are fixed but the location of contaminant source changes.
- Scenario 3: the locations of sensors are unfixed but contaminant source location fixed.

We prove the accuracy of our algorithm by showing whether it is able to find a set of possible sources containing the real one. Meanwhile, we will also prove that the CSI problem is an MMO problem by showing that all the found possible sources can generate the same concentration profile on each monitoring sensor.

5.2.1. Evaluation results of scenario 1

In this case study, two sensors are deployed at nodes 10 and 83 and the contaminant injection happens at the node 44, as shown in Fig. 5. Using the sensor profiles from nodes 10 and 83 as input to our MR-PNGA, 11 potential solutions are found. Table 3 summarizes the results including both the contaminant source and its corresponding mass loading. We can see that MR-PNGA finds 11 potential solutions. The real contaminant source (i.e., 44) is included in the solutions found by our MR-PNGA, indicating that we accurately locate the real contaminant source.

Furthermore, we are also interested in whether these possible injection profiles can indeed generate the same concentration profile on each sensor node. We use EPANET to obtain the concentration profile by simulating contaminant event according to each derived result. For example, we simulate attacks at nodes 22 and 23 with loading mass 30 mg/L and 15 mg/L, respectively. The concentration profiles on sensor node 10 from all the 11 different injection profile are illustrated in Fig. 6, from which we can see that all the concentration profile fits well with the real attack at node 44 with the 300 mg/L. This proves that the CSI problem is essentially an MMO problem and exhibits the non-uniqueness feature.

This also proves that the CSI problem is an MMO problem.

5.2.2. Evaluation results of scenario 2

Next, we evaluate the accuracy of our MR-PNGA by varying the contaminant sources as 34, 71, 22, 44 and 28,

Please cite this article as: C. Hu et al., A MapReduce based Parallel Niche Genetic Algorithm for contaminant source identification in water distribution network, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.011

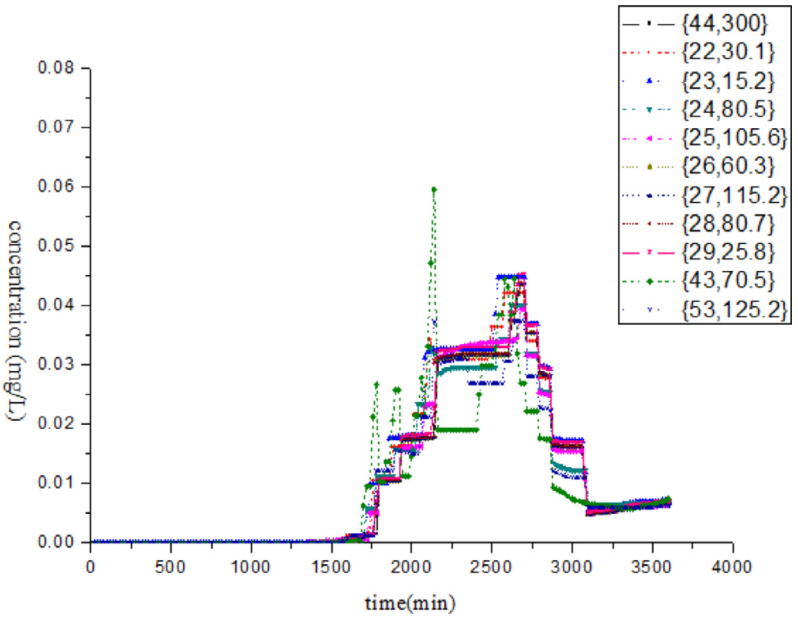


Fig. 6. The contaminant concentration profiles at sensor node 10 under different attacks.

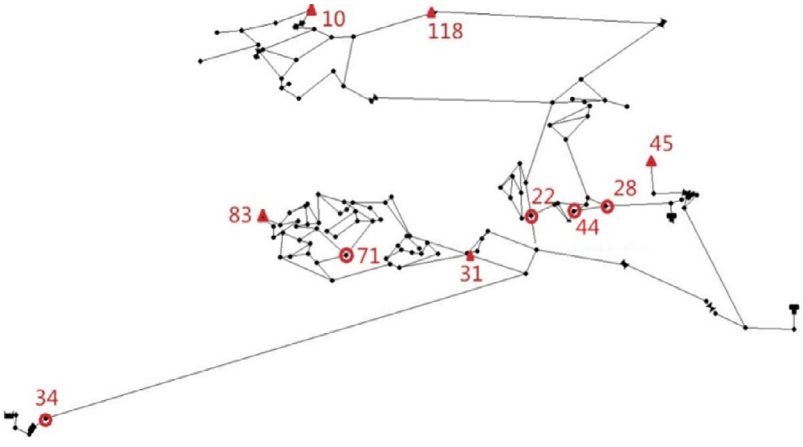


Fig. 7. Testing scenario 2 with different contaminant sources.

Table 4
Success number under different contaminant sources.

	Node 44	Node 71	Node 22	Node 28	Node 34
Success number	29	30	28	30	30

respectively. All the attacks are with contaminant mass loading 300 mg/L. Five sensors are placed at nodes 10, 83, 31, 45 and 118. The testing scenario is shown in Fig. 7, where the source and sensors are represented by red circles and solid triangles, respectively. For each source, 30 different simulation instances with different flow patterns are conducted. We are interested in the ability of our MR-PNGA to find the real contaminate sources in these instances. Table 4 shows the evaluation results. We first see the probabilities of identifying the real source are 96.7%, 100%, 93.3%, 100% and 100%, respectively, for the five attacks, indicating high accuracy of MR-PNGA.

5.2.3. Evaluation results of scenario 3

We finally check the accuracy of MR-PNGA under different sensor deployments under a given attack at node 44 with mass loading 300 mg/L. Three sensors placement strategies, i.e., $A = \{10, 83, 31, 45, 18\}$, $B = \{10, 83, 45\}$ and $C = \{10, 83\}$, are considered. Fig. 8 illustrates strategy *A*. 30 simulation instances with different flow patterns are investigated. We plot in Fig. 9 the average number of possible solutions that can incur the same concentration profile at the sensors. We can see that the number of possible solutions shows as an increasing function of the number of sensors. This is attributed to the intuitive fact that more accurate decision can be made upon

Please cite this article as: C. Hu et al., A MapReduce based Parallel Niche Genetic Algorithm for contaminant source identification in water distribution network, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.011

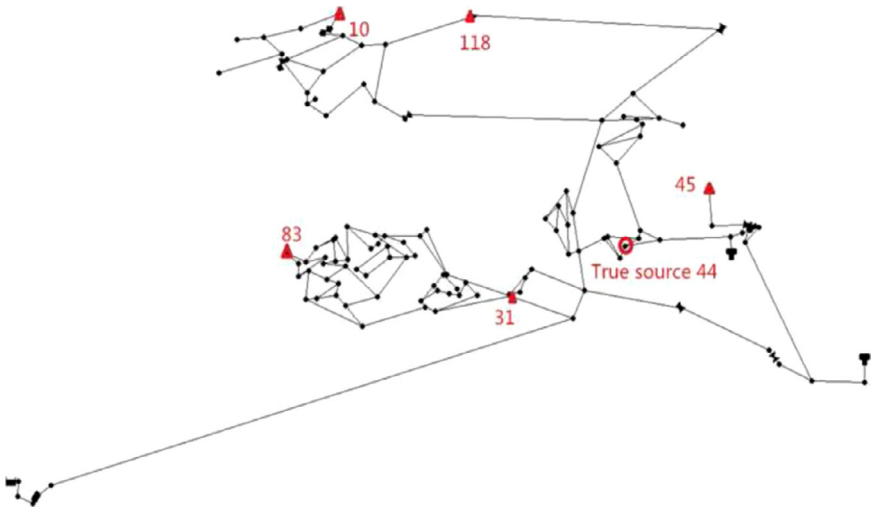


Fig. 8. Sensor placement strategy *A*.

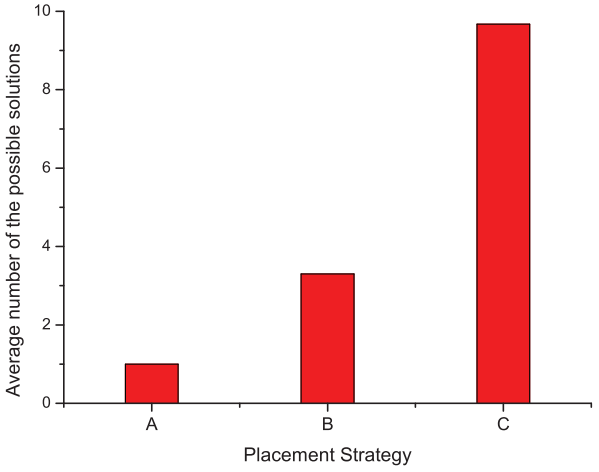


Fig. 9. Number of possible solutions found at different placement strategies.

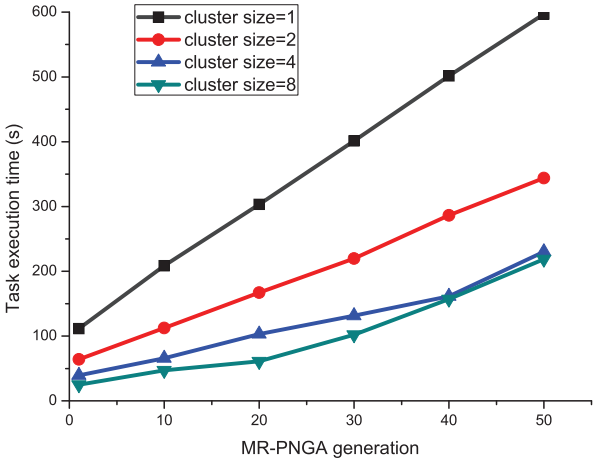


Fig. 10. Task execution time vs. MR-PNGA generation number.

more diverse information. In strategy *A* with 5 sensors, we even can directly locate the real contaminant source, other than finding out a set of possible sources.

5.3. Efficiency evaluation

After showing the accuracy of our MR-PNGA, we further evaluate its efficiency in the metric of task execution time as well as the derived fitness value.

We first vary the generation number (i.e., *maximum iteration* in Algorithm 1) under cluster size (i.e., the number of servers) 1, 2, 4 and 8 to check how it affects the task execution time in the metric of seconds. Fig. 10 shows the task execution time under different generation numbers ranging from 1 to 50. It can be first seen that the task execution time is proportional to the generation number in MR-PNGA. This is simply because larger generation number requires longer evolution time, which further incurs longer task execution time. On the other hand, we can also see that larger cluster size also indicates shorter task execution time. This implies that MR-PNGA efficiently explore the available cloud resources.

We next investigate how the generation number affects the optimization objective, i.e., fitness value, and plot the results in Fig. 11. On the contrary, we can see that increasing generation number is beneficial to the fitness value. However, the benefit becomes marginal when the generation number is large enough. This is the nature of niche genetic algorithm as the quality of the generation finally may converge.

Next, we check how the cluster size influences the task execution time in the metric of speedup ratio, i.e, the ratio between the task execution time of cluster and one server. Fig. 12 presents evaluation results under the cluster size ranging from 1 to 8. We note that the speedup scales well with the cluster size when the cluster size is less than 4, for any generation number. The speedup then increases slowly because larger communication overhead among the servers raises.

In order to further analyze our MR-PNGA, we check the time spent on the three main phases, i.e., (1) the HDFS setup

Please cite this article as: C. Hu et al., A MapReduce based Parallel Niche Genetic Algorithm for contaminant source identification in water distribution network, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.011

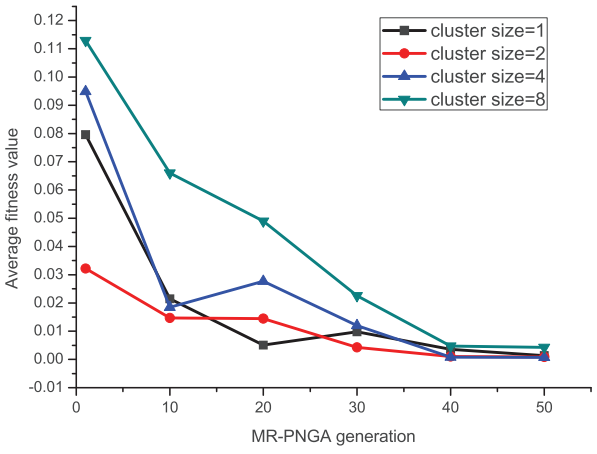


Fig. 11. The fitness value vs. evolution generation.

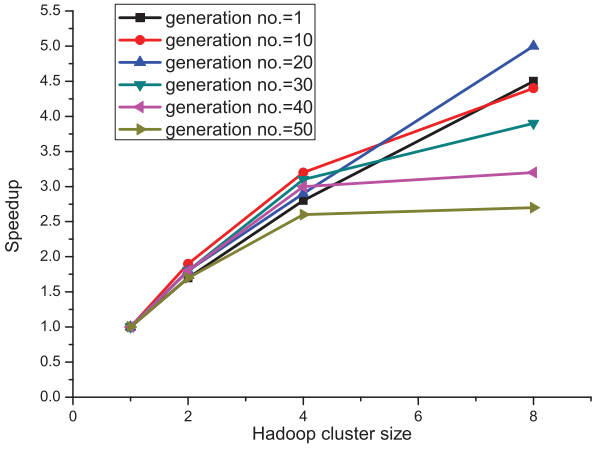


Fig. 12. Speedup vs. cluster size.

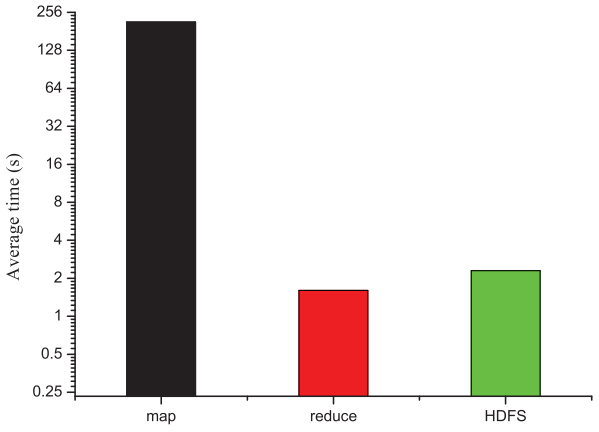


Fig. 13. Average time spent in three main phases.

phase, (2) mapper phase, (3) reducer phase. Average time of 10 runs is logged and plotted in Fig. 13. We observe that the time at HDFS setup and reducer phases is much smaller to the mapper phase. This is because the mapper phase is with intensive hydraulics computation using EPANET. Fortunately, this part is embarrassingly parallel and therefore it is possible to greedily explore the bulk cloud resources to improve the performance. It is expected that MR-PNGA can be significantly acerbated if applied to large-scale cluster or data center.

6. Conclusion and future work

CSI, as an interdisciplinary science of environment and computer, has been regarded as a critical issue to prevent malicious attacks. In this paper, we study a CSI in a water distribution network with limited number of monitoring sensors. We first formally prove that CSI is essentially an MMO problem and show its non-uniqueness feature. To address this issue, an accurate and efficient MapReduce based Parallel Niche Genetic Algorithm (MR-PNGA) is proposed. We also successfully implement MR-PNGA in an 8-server cluster. By incorporating practical contaminant attack scenarios on EPANET, the accuracy and efficiency of MR-PNGA is extensively validated on the cluster.

Acknowledgment

This research was supported in part by the NSF of China (Grant No. 61305087, 61402425, 61272470, 61440060), the China Postdoctoral Science Foundation funded project (Grant No. 2014M562086), the Fundamental Research Funds for National University, China University of Geosciences, Wuhan (Grant No. CUG14065, CUGL150829), the Provincial Natural Science Foundation of Hubei (Grant No. 2015CFA065).

References

[1] A. Ostfeld, l. e. a. Uber, The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms, J. Water Res. Plan. Manag. 134 (6) (2008) 556–568, doi:10.1061/(ASCE)0733-9496(2008)134:6(556).
[2] C. Laird, L. Biegler, B. van Bloemen Waanders, R. Bartlett, Contamination source determination for water networks, J. Water Res. Plan. Manag. 131 (2) (2005) 125–134, doi:10.1061/(ASCE)0733-9496(2005)131:2(125).
[3] A. De Sanctis, F. Shang, J. Uber, Real-time identification of possible contamination sources using network backtracking methods, J. Water Res. Plan. Manag. 136 (4) (2010) 444–453, doi:10.1061/(ASCE)WR.1943-5452.0000050.
[4] H. Wang, K. Harrison, Improving efficiency of the bayesian approach to water distribution contaminant source characterization with support vector regression, J. Water Res. Plan. Manag. 140 (1) (2014) 3–11, doi:10.1061/(ASCE)WR.1943-5452.0000323.
[5] J. Huang, E. McBean, Data mining to identify contaminant event locations in water distribution systems, J. Water Res. Plan. Manag. 135 (6) (2009) 466–474, doi:10.1061/(ASCE)0733-9496(2009)135:6(466).
[6] J. Guan, M. Aral, M. Maslia, W. Grayman, Identification of contaminant sources in water distribution systems using simulation optimization method: Case study, J. Water Res. Plan. Manag. 132 (4) (2006) 252–262, doi:10.1061/(ASCE)0733-9496(2006)132:4(252).
[7] E.M. Zechman, S.R. Ranjithan, Evolutionary computation-based methods for characterizing contaminant sources in a water distribution system, J. Water Res. Plan. Manag. 135 (5) (2009) 334–343.
[8] C.-Y. Lin, W.-H. Wu, Niche identification techniques in multimodal genetic search with sharing scheme, Adv. Eng. Softw. 33 (11-12) (2002) 779–791.
[9] A. Ostfeld, J.G. Uber, E. Salomons, J.W. Berry, W.E. Hart, C.A. Phillips, J.-P. Watson, G. Dorini, P. Jonkergouw, Z. Kapelan, et al., The battle of the water sensor networks (bwsn): A design challenge for engineers and algorithms, J. Water Res. Plan. Manag. 134 (6) (2008) 556–568.
[10] A. Krause, J. Leskovec, C. Guestrin, J. VanBriesen, C. Faloutsos, Efficient sensor placement optimization for securing large water distribution networks, J. Water Res. Plan. Manag. 134 (6) (2008) 516–526.

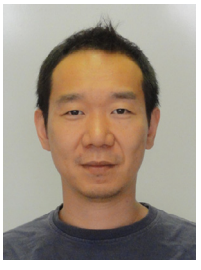
[11] J. Kumar, G. Adviser-Mahinthakumar, S.R. Adviser-Ranjithan, Asynchronous hierarchical parallel evolutionary algorithm-based framework for water distribution systems analysis, North Carolina State University, 2010.
[12] L. Wang, D. Chen, W. Liu, Y. Ma, Y. Wu, Z. Deng, Dddas-based parallel simulation of threat management for urban water distribution systems, Comput. Sci. Eng. 16 (1) (2014) 8–17, doi:10.1109/MCSE.2012.89.
[13] X.Y. Yang, Z. Liu, Y. Fu, Mapreduce as a programming model for association rules algorithm on hadoop, in: Information Sciences and Interaction Sciences (ICIS), 2010 3rd International Conference on, 2010, pp. 99–102, doi:10.1109/ICICIS.2010.5534718.
[14] F. Shang, J. Uber, M. Polycarpou, Particle backtracking algorithm for water distribution system analysis, J. Environ. Eng. 128 (5) (2002) 441–450, doi:10.1061/(ASCE)0733-9372(2002)128:5(441).
[15] L. Perelman, A. Ostfeld, Bayesian networks for source intrusion detection, J. Water Res. Plan. Manag. 139 (4) (2013) 426–432, doi:10.1061/(ASCE)WR.1943-5452.0000288.
[16] H. Wang, X. Jin, Characterization of groundwater contaminant source using bayesian method, Stoch. Environ. Res. Risk Assess. 27 (4) (2013) 867–876, doi:10.1007/s00477-012-0622-9.
[17] L. Liu, S. Ranjithan, G. Mahinthakumar, Contamination source identification in water distribution systems using an adaptive dynamic optimization procedure, J. Water Res. Plan. Manag. 137 (2) (2011) 183–192, doi:10.1061/(ASCE)WR.1943-5452.0000104.
[18] K.-C. Wong, C.-H. Wu, R.K. Mok, C. Peng, Z. Zhang, Evolutionary multimodal optimization using the principle of locality, Inf. Sci. 194 (2012) 138–170.
[19] X. Li, Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization, in: K. Deb (Ed.), Proceedings of the Genetic and Evolutionary Computation GECCO 2004, vol. 3102, Lecture Notes in Computer Science Springer Berlin, Heidelberg, 2004, pp. 105–116.
[20] B. Qu, P. Suganthan, S. Das, A distance-based locally informed particle swarm model for multimodal optimization, Evol. Comput. IEEE Trans. 17 (3) (2013) 387–402, doi:10.1109/TEVC.2012.2203138.
[21] B.Y. Qu, P. Suganthan, J. Liang, Differential evolution with neighborhood mutation for multimodal optimization, Evol. Comput. IEEE Trans. 16 (5) (2012) 601–614, doi:10.1109/TEVC.2011.2161873.
[22] C. Stoean, M. Preuss, R. Stoean, D. Dumitrescu, Multimodal optimization by means of a topological species conservation algorithm, Evol. Comput. IEEE Trans. 14 (6) (2010) 842–864, doi:10.1109/TEVC.2010.2041668.
[23] O. Piller, M. Propato, M.E. Tryby, Linear algebra analysis for contaminant source identification in water distribution systems, Proceedings of the Conference on World Environmental and Water Resources Congress 2007, pp. 1–10 (Chapter 513).

Chengyu Hu received his M.S. degree in automation and control from Wuhan University of Technology in 2003 and obtained his Ph.D. in automation control from Huazhong University of Science and Technology in 2010. He is currently an associate professor and head of Department of Computer Science, China University of Geosciences, Wuhan, China. His current research interests include: evolutionary algorithm, swarm intelligence and cloud computing.

Jing Zhao is a master student at the School of Computer Science, China University of Geosciences, Wuhan, China. Her research interests include: parallel computing and evolutionary computation.

Xuesong Yan received him B.E. degree in Computer Science and Technology in 2000 and M.E. degree in Computer Application from China University of Geosciences in 2003. He received his Ph.D. degree in Computer Software and Theory from Wuhan University in 2006. He is currently with School of Computer Science, China University of Geosciences, Wuhan, China and was as a visiting scholar with Department of Computer Science, University of Central Arkansas, Conway, USA. His research interests include evolutionary computation, data mining and computer application.

Deze Zeng received his Ph.D. and M.S. degrees in computer science from University of Aizu, Aizu-Wakamatsu, Japan, in 2013 and 2009, respectively. He received his B.S. degree from School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2007. He is currently an associate professor and head of Department of Network Engineering in School of Computer Science, China University of Geosciences, Wuhan, China. His current research interests include: cloud computing, software-defined sensor networks, data center networking, networking protocol design and analysis. He is a member of IEEE.



Song Guo received the Ph.D. degree in computer science from the University of Ottawa, Canada in 2005. He is currently a Full Professor at School of Computer Science and Engineering, the University of Aizu, Japan. His research interests are mainly in the areas of protocol design and performance analysis for reliable, energy-efficient, and cost effective communications in wireless networks. He has published over 250 papers in refereed journals and conferences in these areas and received three IEEE/ACM best paper awards. Dr. Guo currently serves as Associate Editor of IEEE Transactions on Parallel and Distributed Systems, Associate Editor of IEEE Transactions on Emerging Topics in Computing with duties on emerging paradigms in computational communication systems, and on editorial boards of many others. He has also been in organizing and technical committees of numerous international conferences. Dr. Guo is a senior member of the IEEE and the ACM.

Please cite this article as: C. Hu et al., A MapReduce based Parallel Niche Genetic Algorithm for contaminant source identification in water distribution network, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.07.011