Automatic Partitioning of Water Distribution Networks Using Multiscale Community Detection and Multiobjective Optimization

Qingzhou Zhang[1]; Zheng Yi Wu[2]; Ming Zhao[3]; Jingyao Qi[4];
Yuan Huang[5]; and Hongbin Zhao[6]

Abstract: Partitioning water distribution networks into district metered areas (DMAs) is a challenge for water utilities due to the complexity of network topology, especially for large-scale and highly looped water supply systems. This paper presents a novel approach, using complex network theory and multiobjective optimization, for efficiently partitioning of water distribution networks. In this approach, a water distribution network is first mapped on to a weighted undirected graph, and its strength is measured by an extended modularity by a random walk strategy. Then, the Louvain algorithm is employed for network partitioning based on the modularity. Finally, boundary pipes between DMAs are determined by applying an evolutionary algorithm to minimize the number of boundary pipes and maximize the network pressure and water age uniformity. The approach has been tested on a real-world distribution system to demonstrate the effectiveness of the method. The results indicate that the method is effective at partitioning the network into DMAs with a similar pressure and water age. **DOI: 10.1061/(ASCE)WR.1943-5452.0000819.**

Author keywords: Network partitioning; Community structure; Modularity; Random walk; Water distribution system.

Introduction

The complexity of a water distribution network (WDN) usually grows substantially with the acceleration of urbanization. It consequently increases the difficulty of leakage management and control. A district metered area (DMA) is considered to be an effective way to facilitate the leakage control and management (Farley 2001). With the aid of the graph partition method, a large-scale network can be decomposed into smaller and manageable subsystems called DMAs. Leakage can be reduced effectively through monitoring and pressure control for each DMA.

There has been a wide range of methods developed for addressing the network partition problem. The methods are mostly formulated as two-stage approach, namely: (1) finding initial network partition based on network topology; and (2) identifying which pipes can be closed to further minimize the number of connections between subnetworks by optimizing some objective function(s) and subject to constraints. Grayman et al. (2009) proposed an approach for evaluating DMA designs on the basis of water supply security and reliability, and provided insight into the effects of water supply security criteria on DMA design. Perelman and Ostfeld (2011) developed a topology clustering method that decomposes the WDN into several clusters according to flow directions. Ferrari et al. (2014) presented a method that combines graph traversal algorithm with design criterion for DMA design. Zheng et al. (2013) decomposed an original network into subnetworks based on the connectivity of network and used evolutionary algorithm to optimize boundary pipes. Di Nardo et al. (2015) proposed a new approach for network partitioning on the basis of graph theory and heuristic optimization algorithm to reduce the risk of the effects of water distribution network intentional contamination. Scarpa et al. (2016) proposed a novel approach based on graph theory to create DMA boundaries automatically. Other similar approaches have also been reported in the literature (Deuerlein 2008; Candelieri et al. 2014; Galdiero et al. 2016; Hajebi et al. 2016; Herrera et al. 2016).

The network community as the branch of graph theory has been the focus of research in recent years. Related studies (Girvan and Newman 2002; Fortunato 2010) demonstrate that a complex network is composed of groups, and the nodes within a group are interconnected closely, while the connection between groups is relatively sparse. Such a group is called a community. Girvan and Newman (2002) defined the modularity function as evaluation criteria for detecting complex network community. A water distribution system can be abstracted into a complex network consisting of communities because the topological structure is formed gradually with the evolution of city blocks (Diao et al. 2013). In recent years, most modularity-based community detection algorithms are developed for auto-partitioning in water distribution systems. Giustolisi and Ridolfi (2014) proposed an optimization framework

[1] Ph.D. Student, School of Municipal and Environmental Engineering, Harbin Institute of Technology, Harbin 150090, China. E-mail: wdswater@gmail.com

[2] Bentley Fellow, Bentley Systems, Incorporated, 27 Siemon Co. Dr., Suite 200W, Watertown, CT 06795. E-mail: zheng.wu@bentley.com

[3] Associate Professor, School of Municipal and Environmental Engineering, Harbin Institute of Technology, Harbin 150090, China. E-mail: zhming1188@126.com

[4] Professor, School of Municipal and Environmental Engineering, Harbin Institute of Technology, Harbin 150090, China (corresponding author). E-mail: qjy_hit@hotmail.com

[5] Ph.D. Student, School of Municipal and Environmental Engineering, Harbin Institute of Technology, Harbin 150090, China. E-mail: huangyuanggg@163.com

[6] Professor, School of Municipal and Environmental Engineering, Harbin Institute of Technology, Harbin 150090, China. E-mail: zhb167@126.com

Note. This manuscript was submitted on September 16, 2016; approved on April 14, 2017; published online on July 14, 2017. Discussion period open until December 14, 2017; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Water Resources Planning and Management***, © ASCE, ISSN 0733-9496.

for network partition by optimizing modularity and reconstruction cost. Campbell et al. (2016) presented a flexible methodology for network segmentation based on community theory and multiobjective optimization. Ciaponi et al. (2016) presented a method that combines modularity-based detection with a practical design criterion for DMA design. Similar applications of community detection for network auto-partitioning have been reported by Campbell et al. (2014), Scibetta et al. (2014), Perelman et al. (2014, 2015), and Giustolisi et al. (2015).

To sum up, although graph theory-based partitioning method is applied to WDNs, there are still some deficiencies, including:

1. The nodal pressure has not been used adequately. The aforementioned methods mainly decompose the network into subnetworks based on the connection relationship of unweighted pipes, without considering the distribution of node pressure. Network sectorization has two main purposes of both DMA management and pressure management. The key to DMA management is to continuously monitor flow into the DMA and to analyze the night flow to determine whether there is excess flow beyond customer use that would constitute leakage. Pressure management is one of the fundamental elements of a well-developed leakage management strategy. Using a pressure reducing valve (PRV) is a common, cost-effective way for pressure management within a DMA. Therefore, it is desirable to develop a method that combines node pressure with modularity-based community detection, and decomposes the network into DMAs with similar pressure.
2. One drawback of community evaluation criteria is identifying a reasonable community structure. Thus far, Girvan and Newman (GN) modularity (Girvan and Newman 2002) is the most widely used evaluation criteria; however, GN modularity has some resolution limitations. Fortunato and Barthelemy (2007) found that GN modularity-based detection algorithms may fail to identify smaller communities in which the total number of edges is less than $\sqrt{2L}$ (L = total number of edges in the network), such as a DMA in which the total edge number is smaller than $\sqrt{2L}$. Hence, GN modularity is suitable for the evaluation of networks that have a similar size of communities, while for the communities with a greater difference in size, desirable partitioning results may not be achievable.
3. Evolutionary algorithms have limitations in the application of large-scale networks. Multiobjective optimization algorithms such as genetic algorithms (GA), particle swarm optimization (PSO), NSGA-II, etc., are usually used to identify which boundary can be closed by optimizing some objective function(s) and subject to constraints. Reed et al. (2013) and Maier et al. (2014) evaluated the performance of evolutionary algorithms. Their works revealed that when solving such multivariable discrete optimization problems, the ability of these algorithms is restricted with the increase of search space, which leads the search to easily to fall into local optimal solutions.

In this paper, a novel auto-partitioning framework is developed in which the node average pressure is integrated as a weight to represent pressure distribution in a water distribution system, and an extended modularity based on random walk theory is proposed to overcome the resolution limit. Furthermore, a new hyperheuristic algorithm, BORG (Hadka and Reed 2013), is used to improve the calculation precision of the multiobjective optimization problem. The proposed approach will be elaborated in the next section.

Methodology

The approach presented here is applied to auto-partitioning of WDN using multiscale community detection and multiobjective optimization.

Topology Characteristics of WDNs

The topology structure of water distribution networks is mapped into an undirected graph $G(V,E)$, where $V=\{v_1,v_2,\ldots,v_n\}$ is the vertex (junctions, reservoirs, and tanks) set; $E=\{e_1,e_2,\ldots,e_m\}$ is the edge (pipes, valves, and pumps) set; $e_k=(v_i,v_j)$ is an edge from v_i to $v_j (k=1,2,\ldots,m)$; $n=|V|$ is the total number of vertexes; $m=|E|$ is the total number of edges; k_i is the degree of v_i, which is defined as the number of edges attaching to v_i; $\mathbf{d}=[k_1,k_2,\ldots,k_n]^T$ is the $n\times 1$ vector of vertex degree; $\mathbf{D}=\mathrm{diag}(\mathbf{d})$ is the $n\times n$ diagonal matrix of $\mathbf{d}$; $\mathbf{A}$ is the $n\times n$ adjacency matrix ($A_{ij}=1$, if v_i is adjacent with v_j, otherwise, $A_{ij}=0$); $\mathbf{L}=\mathbf{D}-\mathbf{A}$ is the combinatorial graph Laplacian; $\mathbf{W}$ is the $n\times n$ weighted adjacency matrix [W_{ij} = edge weight of (v_i,v_j), if v_i is adjacent with v_j, otherwise, $W_{ij}=0$]; $M=1/2\sum_{ij}W_{ij}$ is the total weight of all the edges in the network; $w_i=\sum_j W_{ij}$ is the sum of edge weight attaching to node i; $\mathbf{w}=[w_1,w_2,\ldots,w_n]^T$ is the $n\times 1$ vector of vertex weights; $\boldsymbol{\psi}=\mathrm{diag}(\mathbf{w})$ is the $n\times n$ diagonal matrix of $\mathbf{w}$; and $\mathbf{Y}=\boldsymbol{\psi}-\mathbf{W}$ is the weighted combinatorial graph Laplacian.

Multiresolution Modularity Based on Random Walk

In this study, a community detection algorithm is used to find the best partition of a WDN. The method uses modularity as the evaluation index of network partitioning. GN modularity (Girvan and Newman 2002) is the most widely used in the related literature, and is defined as

$$Q_{GN}=\sum_{s=1}^{c}\left[\frac{l_s}{m}-\left(\frac{d_s}{2m}\right)^2\right] \tag{1}$$

where c = number of communities; l_s = number of edges inside community s; d_s = sum of degrees of all the vertexes in community s; and m = total number of edges. In general, the bigger value of Q_{GN}, the better result of network partition. Eq. (1) can be written in the form of an adjacency matrix (Newman and Girvan 2004)

$$Q_{GN}=\frac{1}{2m}\sum_{ij}\left[\mathbf{A}_{ij}-\frac{k_ik_j}{2m}\right]\delta(C_i,C_j) \tag{2}$$

where $\delta(C_i,C_j)$ is the Kronecker delta function [$\delta(C_i,C_j)=1$, if node i and j belongs to the same community C, otherwise, $\delta(C_i,C_j)=0$]. As outlined earlier, GN modularity does have a resolution limit. Hence, random walk theory (Lambiotte et al. 2014) is used to overcome this problem. The random walk on the network is a dynamical process: if the network has no significant community structure, the walker would be walking isotropically; conversely, the walker would be transiently trapped in certain communities for some time. The random walk process contains three steps: (1) initially, each node has a walker; (2) the walkers start from the initial nodes and then select a neighboring node randomly as the starting point for the next time step; and (3) repeat step (2) until the setting time is reached. For an unweighted undirected graph, the random walk process (Lambiotte et al. 2014) can be described as

$$\mathbf{p}_{t+1}=\mathbf{p}_t[\mathbf{D}^{-1}\mathbf{A}] \tag{3}$$

for discrete time or, alternatively, as

$$\frac{d\mathbf{p}}{dt}=-\mathbf{p}[\mathbf{D}^{-1}\mathbf{L}] \tag{4}$$

for the case of continuous time, where $\mathbf{p}$ is the $1 \times n$ probability vector for each walker and n is the total number of nodes. With the increase of time t, $\mathbf{p}$ will gradually converge to a steady solution given by $\boldsymbol{\pi} = \mathbf{d}^T/2m$ (m = total number of edges). Assuming that a network be decomposed into c communities, the partition is encoded in an $n \times c$ index matrix $\mathbf{H}$ ($H_{ij} = 1$ if node i belongs to community j, otherwise, $H_{ij} = 0$). On the basis of the random walk theory, the evaluation index of network partitioning is defined as (Lambiotte et al. 2014)

$$Q_t = \mathrm{Tr}\mathbf{H}^T[\boldsymbol{\Pi}\mathbf{P}(t) - \boldsymbol{\pi}^T\boldsymbol{\pi}]\mathbf{H}\} \tag{5}$$

where $\mathrm{Tr}\{\}$ is the trace of matrix; $\boldsymbol{\Pi} = \mathrm{diag}(\boldsymbol{\pi})$ is the $n \times n$ diagonal matrix of $\boldsymbol{\pi}$; and $\mathbf{P}(t)$ is the transition matrix of random walk process at time t: $\mathbf{P}(t) = (\mathbf{D}^{-1}\mathbf{A})^t$ for discrete time and $\mathbf{P}(t) = \exp(-\mathbf{D}^{-1}\mathbf{L}t)$ for continuous time. It is worth noting that $Q_{t=1}$ for discrete time is

$$Q_{t=1} = \mathrm{Tr}\mathbf{H}^T[\boldsymbol{\Pi}\mathbf{D}^{-1}\mathbf{A} - \boldsymbol{\pi}^T\boldsymbol{\pi}]\mathbf{H} = \frac{1}{2m}\mathrm{Tr}\left\{\mathbf{H}^T\left(\mathbf{A} - \frac{\mathbf{d}\cdot\mathbf{d}^T}{2m}\right)\mathbf{H}\right\} = Q_{GN} \tag{6}$$

GN modularity Q_{GN} is the special case of Q_t for discrete time at time $t = 1$. Due to Q_t, continuous time is approximately equal to discrete time; therefore, Q_t for continuous time can be used for the sake of convenient calculation (Lambiotte et al. 2014)

$$Q_t = \mathrm{Tr}\{\mathbf{H}^T[\boldsymbol{\Pi}e^{-t\mathbf{D}^{-1}\mathbf{L}} - \boldsymbol{\pi}^T\boldsymbol{\pi}]\mathbf{H}\} \tag{7}$$

Q_t is called multiresolution modularity, in which t is the random walk time. Eq. (7) can be employed as the evaluation index of unweighted network partitioning; thus, Eq. (7) needs to be modified for a weighted network. In this paper, the nodal average pressure as the weight is added to Eq. (7) for network partitioning. The element of weighted adjacency matrix $W_{ij} = (H_i^{\mathrm{avg}} + H_j^{\mathrm{avg}})/2$ is the indicator of the strength of connection between node i and node j, where $H_i^{\mathrm{avg}} = \sum_{t=1}^{TD} H_i^t/TD$ is the average pressure at node i for the network without partitioning, H_i^t is the pressure at node i at time t for the network without partitioning, and TD is the total simulation duration. For a weighted network, Eq. (7) is modified as

$$Q_t^P = \mathrm{Tr}\{\mathbf{H}^T[\boldsymbol{\Phi}e^{-t\boldsymbol{\Psi}^{-1}\mathbf{Y}} - \boldsymbol{\eta}^T\boldsymbol{\eta}]\mathbf{H}\} \tag{8}$$

where Q_t^P is the multiresolution modularity based on pressure distribution; t is the time of random walk; $\boldsymbol{\eta} = \mathbf{w}^T/2M$ is the steady solution in weighted random walk process; and $\boldsymbol{\Phi} = \mathrm{diag}(\boldsymbol{\eta})$ is the $n \times n$ diagonal matrix of $\boldsymbol{\eta}$.

Eq. (7) is used as the evaluation index of unweighted network partition, while Eq. (8) contains the information of node average pressure, which evaluates the difference of pressures between communities. Thus, Eq. (8) is used as the multiresolution modularity to evaluate the quantity of network partitioning.

Assuming that node i belongs to community C_j, C_k is one of the neighboring communities of C_j. Assigning i from C_j to C_k increases the system's modularity as follows:

$$\Delta Q_{i\to C_k} = Q_t^p|_{i\in C_k} - Q_t^p|_{i\in C_j} = \mathrm{Tr}\{\mathbf{H}_{i\in C_k}^T\Theta\mathbf{H}_{i\in C_k}\} - \mathrm{Tr}\{\mathbf{H}_{i\in C_j}^T\Theta\mathbf{H}_{i\in C_j}\} \tag{9}$$

where $Q_t^p|_{i\in C_k}$ is the system's modularity when node i is assigned from C_j to C_k; $Q_t^p|_{i\in C_j}$ is the system's modularity when node i belongs to C_j; $\mathbf{H}_{i\in C_k}$ is the index matrix when node i is assigned from C_j to C_k; and $\mathbf{H}_{i\in C_j}$ is the index matrix when node i belongs to C_j; $\Theta = \boldsymbol{\Phi}e^{-t\boldsymbol{\Psi}^{-1}\mathbf{Y}} - \boldsymbol{\eta}^T\boldsymbol{\eta}$.

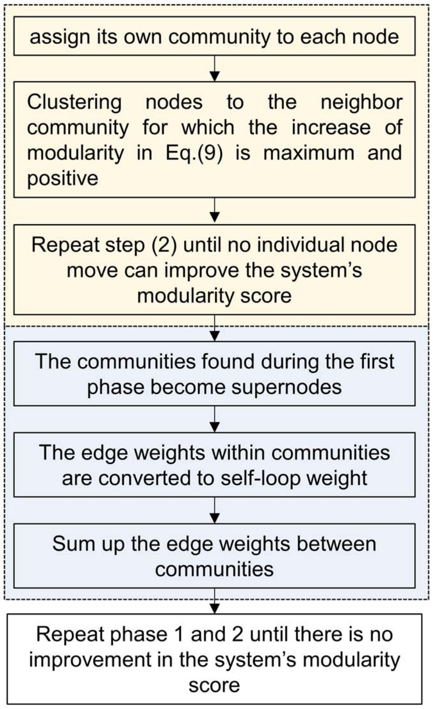
assign its own community to each node
Clustering nodes to the neighbor community for which the increase of modularity in Eq.(9) is maximum and positive
Repeat step (2) until no individual node move can improve the system's modularity score
Phase 1
Find a local maximum
The communities found during the first phase become supernodes
The edge weights within communities are converted to self-loop weight
Sum up the edge weights between communities
Phase 2
Reconstruct a new network
Repeat phase 1 and 2 until there is no improvement in the system's modularity score

Fig. 1. Flowchart of Louvain algorithm

Community Detection Algorithm

This paper uses a Louvain algorithm (Blondel et al. 2008) for community detection. The algorithm is a heuristic greedy method based on modularity optimization, given a network where there are n nodes. The flowchart of the algorithm is shown in Fig. 1.

The algorithm mainly consists of two phases that are repeated iteratively. The first phase is to find local communities by optimizing modularity in multiple steps: (1) initially, each node is considered as an independent community, and the number of communities is equal to the number of nodes; (2) traverse all the nodes in an ordered way (from node 1 to node n). For each node i, try to assign node i to the neighboring communities and calculate the increase of the system's modularity in Eq. (9), respectively. Then, select node k and assign it to the community for which the selected node contributes the maximum and positive modularity increase; next, (3) repeat step (2) until no individual node move can improve the system's modularity score.

Once local communities have been attained, the second phase is to reconstruct a new network by compressing the communities found during the first phase. All the nodes within the same community are compressed into a supernode. The sum of edge weights within the same community is converted to the supernode's self-loop weight, and the weights of the edges between the supernodes are the sum of edge weights between nodes in the corresponding two communities. The two phases are repeated iteratively until there is no improvement in the system's modularity score.

A simple weighted network that has 12 nodes and 17 edges is presented here to illustrate the entire calculation process of the Louvain algorithm. The layout of the network as shown in Fig. 2(a). Thiss example assumes that random walk time t in Eq. (8) is equal to 1.5. The detailed calculation process is as follows:

1. The first phase is to find local communities by optimizing modularity.
   a. Initialization, each node is treated as an independent community, namely

$$\begin{aligned} &C_1=\{1\}; \quad C_2=\{2\}; \quad C_3=\{3\}; \quad C_4=\{4\}; \quad C_5=\{5\}; \quad C_6=\{6\}; \\ &C_7=\{7\}; \quad C_8=\{8\}; \quad C_9=\{9\}; \quad C_{10}=\{10\}; \quad C_{11}=\{11\}; \quad C_{12}=\{12\} \end{aligned}$$

For the original network, the weighted adjacency matrix $\mathbf{W}$ is

$$\mathbf{W}=\begin{bmatrix} 0 & 14.55 & 0 & 14.70 & 14.70 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 14.55 & 0 & 15.55 & 0 & 14.85 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15.55 & 0 & 0 & 0 & 16.45 & 0 & 0 & 0 & 0 & 0 & 0 \\ 14.70 & 0 & 0 & 0 & 15 & 0 & 15.15 & 0 & 0 & 0 & 0 & 0 \\ 14.70 & 14.85 & 0 & 15 & 0 & 15.75 & 0 & 15.30 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16.45 & 0 & 15.75 & 0 & 0 & 0 & 16.50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 15.15 & 0 & 0 & 0 & 15.45 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 15.30 & 0 & 15.45 & 0 & 16.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 16.50 & 0 & 16.05 & 0 & 17.55 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 17.55 & 0 & 18.75 & 18.60 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18.75 & 0 & 18.75 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18.60 & 18.75 & 0 \end{bmatrix}$$

where the element of weighted adjacency matrix $W_{ij}=(H_i+H_j)/2$ is calculated as $W_{12}=(H_1+H_2)/2=(14.4+14.7)/2=14.55$

$$\begin{aligned} M&=\frac{1}{2}\sum_{ij}W_{ij}=273.65 \\ \mathbf{w}&=[43.95, 44.95, 32, 44.85, 75.6, 48.7, 30.6, 46.8, 50.1, 54.9, 37.5, 37.35]^T \\ \boldsymbol{\eta}&=\mathbf{w}^T/2M=[0.0803, 0.0821, 0.0585, 0.0819, 0.1381, 0.089, 0.0559, 0.0855, 0.0915, 0.1003, 0.0685, 0.0682], \\ \boldsymbol{\psi}&=\operatorname{diag}(\mathbf{w}), \qquad \boldsymbol{\Phi}=\operatorname{diag}(\boldsymbol{\eta}) \end{aligned}$$

The weighted combinatorial graph Laplacian $\mathbf{Y}$ is

$$\mathbf{Y}=\boldsymbol{\psi}-\mathbf{W}=\begin{bmatrix} 43.95 & -14.55 & 0 & -14.70 & -14.70 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -14.55 & 44.95 & -15.55 & 0 & -14.85 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -15.55 & 32 & 0 & 0 & -16.45 & 0 & 0 & 0 & 0 & 0 & 0 \\ -14.70 & 0 & 0 & 44.85 & -15 & 0 & -15.15 & 0 & 0 & 0 & 0 & 0 \\ -14.70 & -14.85 & 0 & -15 & 75.60 & -15.75 & 0 & -15.30 & 0 & 0 & 0 & 0 \\ 0 & 0 & -16.45 & 0 & -15.75 & 48.70 & 0 & 0 & -16.50 & 0 & 0 & 0 \\ 0 & 0 & 0 & -15.15 & 0 & 0 & 30.60 & -15.45 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -15.30 & 0 & -15.45 & 46.80 & -16.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -16.50 & 0 & -16.05 & 50.10 & -17.55 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -17.55 & 54.90 & -18.75 & -18.60 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18.75 & 37.5 & -18.75 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -18.60 & -18.75 & 37.35 \end{bmatrix}$$

   b. Traverse all the nodes in an ordered way and cluster each node to the neighbor community for which the increase of modularity in Eq. (9) is maximum and positive. Take node 1 during the first iteration; for example, assigning node 1 from C_1 to C_2 increases the system's modularity, which is calculated as follows.

The index matrix when node 1 belongs to C_1 is

$$\mathbf{H}_{1\in C_1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The index matrix when node 1 is assigned from C_1 to C_2 is

$$\mathbf{H}_{1\in C_2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \Theta &= \Phi e^{-t\boldsymbol{\psi}^{-1}\mathbf{Y}} - \boldsymbol{\eta}^T\boldsymbol{\eta} \\ &= 10^{-2} \times \begin{bmatrix} 1.88 & 0.61 & -0.14 & 0.62 & 0.53 & -0.42 & -0.13 & -0.40 & -0.65 & -0.80 & -0.55 & -0.55 \\ 0.61 & 2.00 & 0.70 & -0.20 & 0.38 & -0.11 & -0.35 & -0.48 & -0.62 & -0.81 & -0.56 & -0.56 \\ -0.14 & 0.70 & 1.52 & -0.37 & -0.20 & 0.73 & -0.30 & -0.39 & -0.23 & -0.53 & -0.39 & -0.39 \\ 0.62 & -0.20 & -0.37 & 2.00 & 0.39 & -0.50 & 0.69 & -0.10 & -0.62 & -0.81 & -0.56 & -0.56 \\ 0.53 & 0.38 & -0.20 & 0.39 & 2.68 & 0.11 & -0.18 & 0.12 & -0.68 & -1.29 & -0.93 & -0.93 \\ -0.42 & -0.11 & 0.73 & -0.50 & 0.11 & 2.09 & -0.39 & -0.29 & 0.46 & -0.57 & -0.55 & -0.55 \\ -0.13 & -0.35 & -0.30 & 0.69 & -0.18 & -0.39 & 1.46 & 0.69 & -0.22 & -0.51 & -0.38 & -0.37 \\ -0.40 & -0.48 & -0.39 & -0.10 & 0.12 & -0.29 & 0.69 & 2.01 & 0.45 & -0.55 & -0.53 & -0.52 \\ -0.65 & -0.62 & -0.23 & -0.62 & -0.68 & 0.46 & -0.22 & 0.45 & 2.07 & 0.48 & -0.22 & -0.22 \\ -0.80 & -0.81 & -0.53 & -0.81 & -1.29 & -0.57 & -0.51 & -0.55 & 0.48 & 2.77 & 1.31 & 1.30 \\ -0.55 & -0.56 & -0.39 & -0.56 & -0.93 & -0.55 & -0.38 & -0.53 & -0.22 & 1.31 & 2.04 & 1.32 \\ -0.55 & -0.56 & -0.39 & -0.56 & -0.93 & -0.55 & -0.37 & -0.52 & -0.22 & 1.30 & 1.32 & 2.03 \end{bmatrix} \end{aligned}$$

Thus, the increase of the system's modularity when node 1 is assigned from C_1 to C_2 is

$$\Delta Q_{1\to C_2} = \mathrm{Tr}\{\mathbf{H}_{1\in C_2}^T \Theta \mathbf{H}_{1\in C_2}\} - \mathrm{Tr}\{\mathbf{H}_{1\in C_1}^T \Theta \mathbf{H}_{1\in C_1}\} = 0.258 - 0.246 = 0.0122$$

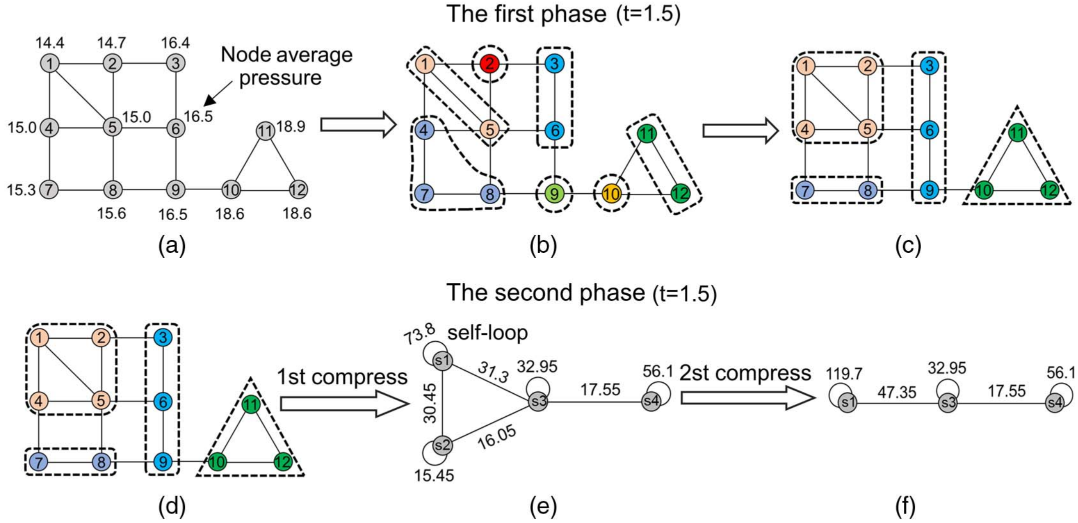

Fig. 2. Calculation process of Louvain algorithm for a simple network ($t = 1.5$): (a) initially, each node belongs to its own community; (b) after each node has been iterated one time; (c) after each node has been iterated four times; (d) local communities attained by first phase; (e) new network of four supernodes; (f) new network of three supernodes (convergence)

similarly

$$\Delta Q_{1\to C_4} = \mathrm{Tr}\{\mathbf{H}^T_{1\in C_4}\Theta\mathbf{H}_{1\in C_4}\} - \mathrm{Tr}\{\mathbf{H}^T_{1\in C_1}\Theta\mathbf{H}_{1\in C_1}\}$$
$$= 0.2580 - 0.2455 = 0.0125$$
$$\Delta Q_{1\to C_5} = \mathrm{Tr}\{\mathbf{H}^T_{1\in C_5}\Theta\mathbf{H}_{1\in C_5}\} - \mathrm{Tr}\{\mathbf{H}^T_{1\in C_1}\Theta\mathbf{H}_{1\in C_1}\}$$
$$= 0.2561 - 0.2455 = 0.0106$$

since

$$\max\Delta Q = \max\{\Delta Q_{1\to C_2}, \Delta Q_{1\to C_4}, \Delta Q_{1\to C_5}\} = \Delta Q_{1\to C_4}$$

hence, node 1 is assigned to C_4. For node 2 to node 12, the calculation process is similar to node 1. The calculation result of the first iteration is shown in Table 1 and Fig. 2(b).

c. Repeat step b. In the example, after each node has been iterated four times, there is no individual node move that can improve the system's modularity score. As shown in Fig. 2(c), the first phase has found four local communities by optimizing the modularity.

2. The second phase is to reconstruct a new network by compressing the communities found during the first phase. As shown in Fig. 2(e), the self-loop weight of each supernode is calculated as follows:

$$W_{s_1s_1} = W_{12} + W_{14} + W_{15} + W_{25} + W_{45}$$
$$= 14.55 + 14.7 + 14.7 + 14.85 + 15 = 73.8$$
$$W_{s_2s_2} = W_{78} = 15.45$$
$$W_{s_3s_3} = W_{36} + W_{69} = 16.45 + 16.50 = 32.95$$
$$W_{s_4s_4} = W_{10,11} + W_{10,12} + W_{11,12}$$
$$= 18.75 + 18.60 + 18.75 = 56.10$$

Table 1. Calculation Result of the First Iteration ($t = 1.5$)

Node	ΔQ	Update community
Node 1	$\Delta Q_{1\to C_2} = 0.0122$, $\Delta Q_{1\to C_4} = 0.0125$, $\Delta \mathrm{Q}_{1\to C_5} = 0.0106$, $\max\Delta Q = \Delta Q_{1\to C_4}$	$C_2 = \{2\}$, $C_3 = \{3\}$, $C_4 = \{1,4\}$, $C_5 = \{5\}$, $C_6 = \{6\}$, $C_7 = \{7\}$, $C_8 = \{8\}$, $C_9 = \{9\}$, $C_{10} = \{10\}$, $C_{11} = \{11\}$, $C_{12} = \{12\}$
Node 2	$\Delta Q_{2\to C_4} = 0.008$, $\Delta Q_{2\to C_3} = 0.014$, $\Delta \mathrm{Q}_{2\to C_5} = 0.007$, $\max\Delta Q = \Delta Q_{2\to C_3}$	$C_3 = \{2,3\}$, $C_4 = \{1,4\}$, $C_5 = \{5\}$, $C_6 = \{6\}$, $C_7 = \{7\}$, $C_8 = \{8\}$, $C_9 = \{9\}$, $C_{10} = \{10\}$, $C_{11} = \{11\}$, $C_{12} = \{12\}$
Node 3	$\Delta Q_{3\to C_6} = 0.0005$, $\max\Delta Q = \Delta Q_{3\to C_6}$	$C_3 = \{2\}$, $C_4 = \{1,4\}$, $C_5 = \{5\}$, $C_6 = \{3,6\}$, $C_7 = \{7\}$, $C_8 = \{8\}$, $C_9 = \{9\}$, $C_{10} = \{10\}$, $C_{11} = \{11\}$, $C_{12} = \{12\}$
Node 4	$\Delta Q_{4\to C_5} = -0.005$, $\Delta Q_{4\to C_7} = 0.0014$, $\max\Delta Q = \Delta Q_{4\to C_7}$	$C_3 = \{2\}$, $C_4 = \{1\}$, $C_5 = \{5\}$, $C_6 = \{3,6\}$, $C_7 = \{4,7\}$, $C_8 = \{8\}$, $C_9 = \{9\}$, $C_{10} = \{10\}$, $C_{11} = \{11\}$, $C_{12} = \{12\}$
Node 5	$\Delta Q_{5\to C_4} = 0.011$, $\Delta Q_{5\to C_3} = 0.008$, $\Delta Q_{5\to C_7} = 0.004$, $\Delta Q_{5\to C_6} = -0.002$, $\Delta Q_{5\to C_8} = 0.002$, $\max\Delta Q = \Delta Q_{5\to C_4}$	$C_3 = \{2\}$, $C_4 = \{1,5\}$, $C_6 = \{3,6\}$, $C_7 = \{4,7\}$, $C_8 = \{8\}$, $C_9 = \{9\}$, $C_{10} = \{10\}$, $C_{11} = \{11\}$, $C_{12} = \{12\}$
Node 6	$\Delta Q_{6\to C_4} = -0.023$, $\Delta Q_{6\to C_9} = -0.005$, $\max\Delta Q = \Delta Q_{6\to C_9} < 0$	Since $\max\Delta Q < 0$, the community does not change
Node 7	$\Delta Q_{7\to C_8} = -0.016$, $\max\Delta Q = \Delta Q_{7\to C_8} < 0$	Since $\max\Delta Q < 0$, the community does not change
Node 8	$\Delta Q_{8\to C_7} = 0.012$, $\Delta Q_{8\to C_4} = -0.006$, $\Delta \mathrm{Q}_{8\to C_9} = 0.009$, $\max\Delta Q = \Delta Q_{8\to C_7}$	$C_3 = \{2\}$, $C_4 = \{1,5\}$, $C_6 = \{3,6\}$, $C_7 = \{4,7,8\}$, $C_9 = \{9\}$, $C_{10} = \{10\}$, $C_{11} = \{11\}$, $C_{12} = \{12\}$
Node 9	$\Delta Q_{9\to C_6} = 0.005$, $\Delta Q_{9\to C_7} = -0.008$, $\Delta \mathrm{Q}_{9\to C_{10}} = 0.010$, $\max\Delta Q = \Delta Q_{9\to C_{10}}$	$C_3 = \{2\}$, $C_4 = \{1,5\}$, $C_6 = \{3,6\}$, $C_7 = \{4,7,8\}$, $C_{10} = \{9,10\}$, $C_{11} = \{11\}$, $C_{12} = \{12\}$
Node 10	$\Delta Q_{10\to C_{11}} = 0.0166$, $\Delta Q_{10\to C_{12}} = 0.0165, \max\Delta Q = \Delta Q_{10\to 11}$	$C_3 = \{2\}$, $C_4 = \{1,5\}$, $C_6 = \{3,6\}$, $C_7 = \{4,7,8\}$, $C_{10} = \{9\}$, $C_{11} = \{10,11\}$, $C_{12} = \{12\}$
Node 11	$\Delta Q_{11\to C_{12}} = 0.0002$, $\max\Delta Q = \Delta Q_{11\to C_{12}}$	$C_3 = \{2\}$, $C_4 = \{1,5\}$, $C_6 = \{3,6\}$, $C_7 = \{4,7,8\}$, $C_{10} = \{9\}$, $C_{11} = \{10\}$, $C_{12} = \{11,12\}$
Node 12	$\Delta Q_{12\to C_{11}} = -0.0003$, $\max\Delta Q = \Delta Q_{12\to C_{11}} < 0$	Since $\max\Delta Q < 0$, the community does not change

The weights of the edges between the supernodes are calculated as follows:

$$W_{s_1 s_2} = W_{47} + W_{58} = 15.15 + 15.30 = 30.45$$
$$W_{s_1 s_3} = W_{23} + W_{56} = 15.55 + 15.75 = 31.3$$
$$W_{s_2 s_3} = W_{89} = 16.05$$
$$W_{s_3 s_4} = W_{9,10} = 17.55$$

Hence, for the compressed network, the weighted adjacency matrix $\mathbf{W}$ is

$$\mathbf{W} = \begin{bmatrix} 73.8 & 30.45 & 31.3 & 0 \\ 30.45 & 15.45 & 16.05 & 0 \\ 31.3 & 16.05 & 32.95 & 17.55 \\ 0 & 0 & 17.55 & 56.1 \end{bmatrix}$$

Then, the generation of the new network continues to repeat the first phase. The two phases are repeated iteratively. In this example, after the two phases have been looped twice, there is no improvement in the system's modularity score. As shown in Fig. 2(f), the network has been divided into the following three communities eventually: $C_1 = \{1,2,4,5,7,8\}$, $C_2 = \{3,6,9\}$, and $C_3 = \{10,11,12\}$.

Multiscale Network Partitioning

Multiscale network partitioning, as shown in Fig. 3, can be performed in the following six steps: (1) input the water distribution network model and set the required input parameters of the range of random walk time $[t_{\min}, t_{\max}]$ and the random walk time increment Δt; (2) run the hydraulic model and calculate the average pressure of each node; (3) initialize the random walk time $t = t_{\min}$ in Eq. (8); (4) use the Louvain algorithm for network partitioning of the corresponding time t; (5) save the partition results of corresponding time t; and (6) update the parameter $t = t + \Delta t$ and check whether $t \geq t_{\max}$. If not, repeat step (4) until the termination condition is met.

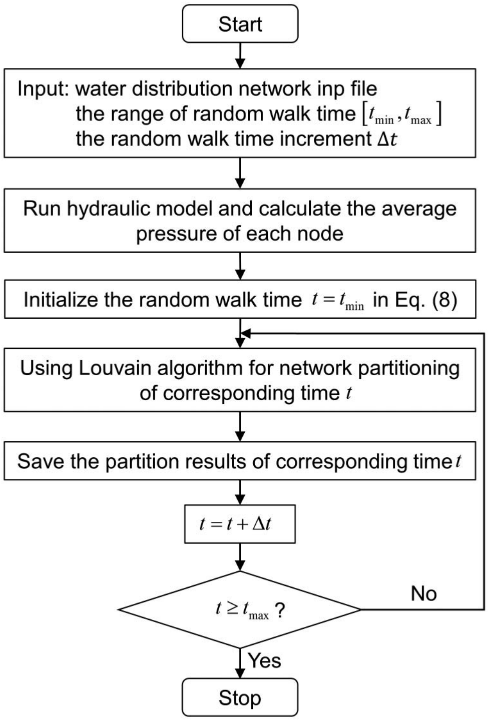
Start
Input: water distribution network inp file
the range of random walk time $[t_{\min}, t_{\max}]$
the random walk time increment Δt
Run hydraulic model and calculate the average pressure of each node
Initialize the random walk time $t = t_{\min}$ in Eq. (8)
Using Louvain algorithm for network partitioning of corresponding time t
Save the partition results of corresponding time t
$t = t + \Delta t$
$t \geq t_{\max}$?
No
Yes
Stop

Fig. 3. Flowchart of multiscale network partitioning

In the previous section, a simple WDN as illustrated in Fig. 2(a) was used to illustrate the calculation process of Louvain algorithm in the case of the random walk time of $t = 1.5$. In this section, the same network is used to illustrate the approach of multiscale network partitioning. There are no existing criteria to set the range of random walk time and the random walk time increment. The corresponding parameters are set depending on the scale and complexity of water distribution systems. For large-scale and highly looped systems, the range and the increment of random walk time must be greater than small systems. In this example, the range of random walk time is set to $[t_{\min}, t_{\max}] = [0.5, 5]$ and the random walk time increment is set to $\Delta t = 0.5$. Table 2 shows the partition results at different random walk times. As shown in Table 2, the network was decomposed into four communities at random walk time $t = 0.5$; three communities at $t = 1.0$, 1.5; two communities at $t = 2.0$; 2.5, 3.0, 3.5, 4.0; and one community at $t = 4.5$, 5.0.

In order to better illustrate how the Louvain algorithm is affected by the random walk time, the network partition was recalculated based on smaller time increment $\Delta t = 0.05$. Fig. 4 shows the partition results at different time periods. With the increase of t, the network is decomposed into different scales of community structures (from finer to coarser). As shown in Fig. 4, according to random walk theory (Lambiotte et al. 2014), using the Louvain algorithm usually results in different partition schemes at different random walk time periods. In general, the scale and complexity of water distribution systems vary greatly in different cities. Hence, it would be reasonable for engineers to select the appropriate partition scheme from these schemes according to the local conditions of the cities.

The GN modularity-based community detection method (Diao et al. 2013) was applied and compared here to better illustrate the superiority of the new approach proposed in this paper. The method described by Diao et al. (2013) decomposed a network into subnetworks based on the connection relationship of unweighted pipes, without considering the node pressure. Fig. 5 shows the partition results using the method proposed by Diao et al. (2013). As shown in Fig. 5, using top-down search can result in different partitioning schemes. The network can be decomposed into 2, 3, and 4 communities using the both methods. Since variance is a measurement

Table 2. Network Partition Results at Different Random Walk Times

Random walk time t	Partition results	Systems' modularity
$t = 0.5$	$C_1 = \{1,2,4,5\}$, $C_2 = \{7,8\}$, $C_3 = \{3,6,9\}$, $C_4 = \{10,11,12\}$	0.5764
$t = 1.0$	$C_1 = \{1,2,4,5,7,8\}$, $C_2 = \{3,6,9\}$, $C_3 = \{10,11,12\}$	0.4410
$t = 1.5$	$C_1 = \{1,2,4,5,7,8\}$, $C_2 = \{3,6,9\}$, $C_3 = \{10,11,12\}$	0.3869
$t = 2.0$	$C_1 = \{1,2,3,4,5,6,7,8,9\}$, $C_2 = \{10,11,12\}$	0.2779
$t = 2.5$	$C_1 = \{1,2,3,4,5,6,7,8,9\}$, $C_2 = \{10,11,12\}$	0.2663
$t = 3.0$	$C_1 = \{1,2,3,4,5,6,7,8,9\}$, $C_2 = \{10,11,12\}$	0.2499
$t = 3.5$	$C_1 = \{1,2,3,4,5,6,7,8,9\}$, $C_2 = \{10,11,12\}$	0.2374
$t = 4.0$	$C_1 = \{1,2,3,4,5,6,7,8,9\}$, $C_2 = \{10,11,12\}$	0.2258
$t = 4.5$	$C_1 = \{1,2,3,4,5,6,7,8,9,10,11,12\}$	0
$t = 5.0$	$C_1 = \{1,2,3,4,5,6,7,8,9,10,11,12\}$	0

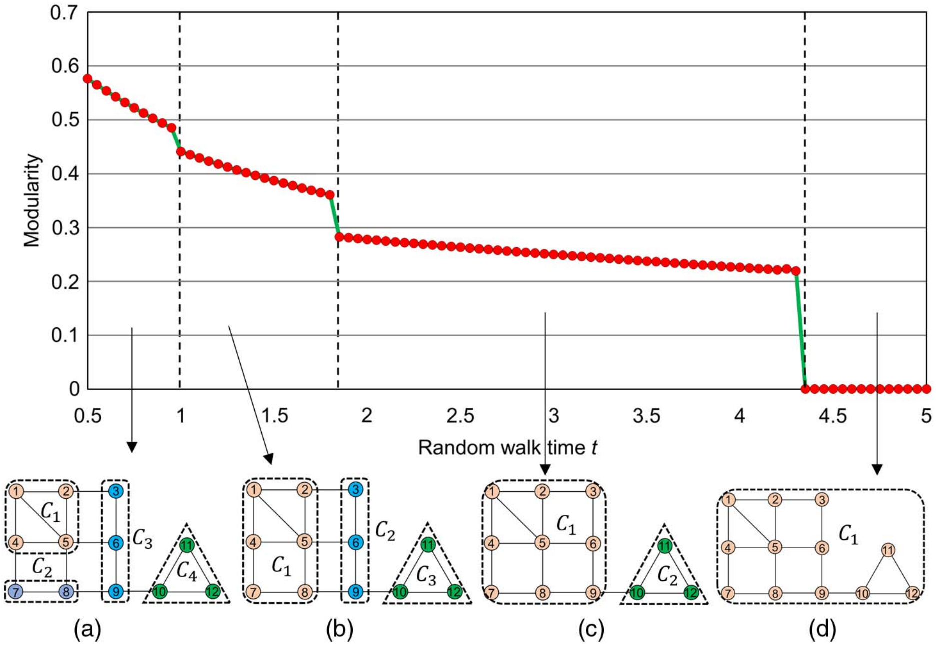

Fig. 4. Network partition results at different time periods

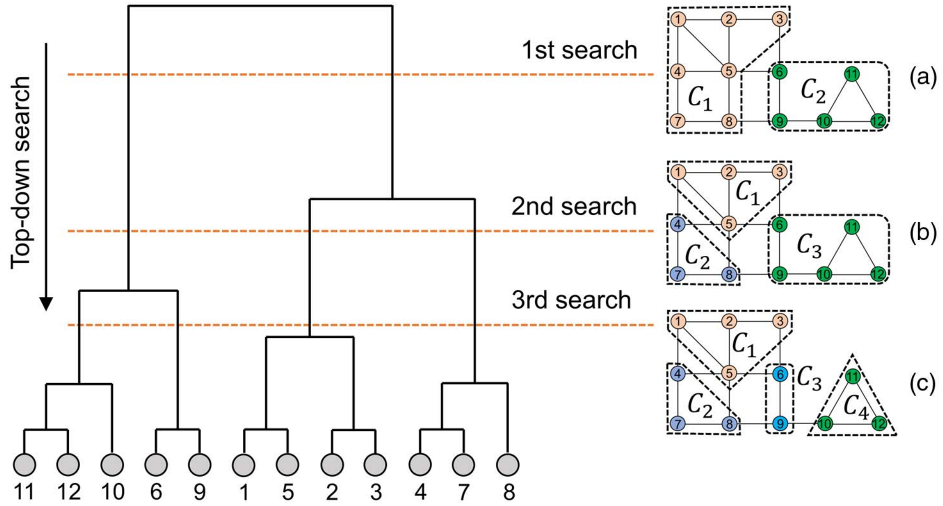

Fig. 5. Partition results using the community detection method proposed by Diao et al. (2013)

of the spread between numbers in a data set, the variance of node pressure can be used to compare the quality of partition of two methods. It measures how far each node pressure in the same community is from the mean. The variance of node pressure of community C_i is defined as

$$\mathrm{var}(C_i) = \frac{1}{N}\sum_{j\in C_i}(H_j^{\mathrm{avg}} - \bar{H})^2 \tag{10}$$

where N = number of nodes in C_i, H_j^{avg} = average pressure of node j, and $\bar{H} = \sum_{j\in C_i} H_j^{\mathrm{avg}}/N$. For instance, when the network is decomposed into two communities using the GN modularity-based partitioning method [Fig. 5(a)], the variance of node average pressure in community C_1 is calculated as

$$\begin{aligned}\bar{H} &= \frac{1}{N}\sum_{j\in C_i} H_j^{\mathrm{avg}}\\ &= \frac{1}{7}(H_1^{\mathrm{avg}} + H_2^{\mathrm{avg}} + H_3^{\mathrm{avg}} + H_4^{\mathrm{avg}} + H_5^{\mathrm{avg}} + H_7^{\mathrm{avg}} + H_8^{\mathrm{avg}})\\ &= \frac{1}{7}(14.4 + 14.7 + 16.4 + 15 + 15 + 15.3 + 15.6) = 15.2\end{aligned}$$

$$\begin{aligned}\mathrm{var}(C_1) &= \frac{1}{n}\sum_{i=1}^{n}(H_i - \bar{H})\\ &= \frac{1}{7}[(14.4 - 15.2)^2 + (14.7 - 15.2)^2 + (16.4 - 15.2)^2\\ &\quad + (15 - 15.2)^2 + (15 - 15.2)^2 + (15.3 - 15.2)^2\\ &\quad + (15.6 - 15.2)^2] = 0.369\end{aligned}$$

Table 3. Comparison of Two Different Community Detection Methods

Number of communities	Proposed approach	Average variance	GN modularity-based partitioning method (Diao et al. 2013)	Average variance
2	$\mathrm{var}(C_1) = 0.579$, $\mathrm{var}(C_2) = 0.020$	0.300	$\mathrm{var}(C_1) = 0.369$, $\mathrm{var}(C_2) = 1.174$	0.772
3	$\mathrm{var}(C_1) = 0.150$, $\mathrm{var}(C_2) = 0.002$, $\mathrm{var}(C_3) = 0.020$	0.057	$\mathrm{var}(C_1) = 0.587$, $\mathrm{var}(C_2) = 0.060$, $\mathrm{var}(C_3) = 1.174$	0.607
4	$\mathrm{var}(C_1) = 0.062$, $\mathrm{var}(C_2) = 0.023$, $\mathrm{var}(C_3) = 0.002$, $\mathrm{var}(C_4) = 0.020$	0.027	$\mathrm{var}(C_1) = 0.587$, $\mathrm{var}(C_2) = 0.060$, $\mathrm{var}(C_3) = 0.000$, $\mathrm{var}(C_4) = 0.020$	0.167

Similarly, $\mathrm{var}(C_2) = 1.174$. The average variance is $[\mathrm{var}(C_1) + \mathrm{var}(C_2)]/2 = 0.772$. The detailed calculation result is shown in Table 3. As shown in Table 3, in the same number of partitions, the average variance using the proposed approach is obviously less than the average variance using GN modularity-based partitioning method (Diao et al. 2013). That is to say, the proposed approach can partition the network into communities with more uniform pressure.

In addition, the GN modularity-based partitioning may result in many more boundary links. For example, as shown in Figs. 4(c) and 5(a), the network was decomposed into two communities using the two respective methods. GN modularity-based partitioning, as shown in Fig. 5(a), results in many more boundary links than the partitioning as shown in Fig. 4(c) by the proposed approach. Hence, for actual complex water networks, the approach proposed here can identify smaller communities and achieve satisfactory results compared with the traditional modularity-based community detection method.

Boundary Pipe Optimization

After the community structure of WDNs is identified, the next step is to determine which pipe can be closed among the boundary pipes. That is generally a multiobjective, discrete nonlinear combinatory optimization problem. In this paper, three objective functions are formulated by taking into account the number of boundary pipes, network pressure uniformity, and water age uniformity. The boundary pipes optimization model is generalized as follows:

$$\mathbf{X} = (x_1, x_2, \ldots, x_{np}) \tag{11}$$

Minimize the number of boundary pipes

$$f_1(\mathbf{X}) = \sum_{l=1}^{np} S_l \tag{12}$$

the pressure uniformity (Alhimiary and Alsuhaily 2007)

$$f_2(\mathbf{X}) = \sum_t \left[\frac{1}{n} \sum_{i=1}^{n} \left(\frac{H_i^t - H_i^{\min}}{H_i^{\min}} \right) + \frac{\sqrt{\frac{\sum_{i=1}^{n} (H_i^t - H_{\mathrm{avg}}^t)^2}{n}}}{H_{\mathrm{avg}}^t} \right] \tag{13}$$

and the water age uniformity (Marchi et al. 2014)

$$f_3(\mathbf{X}) = \frac{\sum_{i=1}^{n} \sum_t \sigma_i^t q_i^t (WA_i^t - WA_{\lim})}{\sum_{i=1}^{n} \sum_t q_i^t} \tag{14}$$

Subject to junction pressure constraints

$$H_i^{\min} \le H_i^t \le H_i^{\max}, \quad \forall\ t, i = 1, \ldots, n \tag{15}$$

tank storage constraints

$$TL_j^{\min} \le TL_j^t \le TL_j^{\max}, \quad \forall\ j, \quad \forall\ t \tag{16}$$

$$TL_j^{\mathrm{final}} \ge TL_j^0, \quad \forall\ j \tag{17}$$

and hydraulic constraints

$$G(\mathbf{h}, \mathbf{q}) = 0 \tag{18}$$

where $\mathbf{X}$ = vector of decision variable, in which $x_l = \{IP_l, S_l\}$ $(l = 1, \ldots, np)$; IP_l is the pipe index for boundary pipe l; S_l is the status of boundary pipe l ($S_l = 1$, if pipe l is open, otherwise, $S_l = 0$); np is the number of boundary pipes; H_i^t = pressure at node i at time t; $H_i^{\min}$ and $H_i^{\max}$ are, respectively, the minimum required and maximum allowable pressure for node i; n is the number of nodes in the network; $H_{\mathrm{avg}}^t = \sum_{i=1}^{n} H_i^t/n$ is the average pressure of the network at time t; WA_i^t is the water age at junction i (excluding tanks and reservoirs) at time t; $WA_{\lim}$ is the upper limit of water age (in hours); q_i^t is the actual demand at node i at time t; σ_i^t is the index variable ($\sigma_i^t = 1$ if $WA_i^t \ge WA_{\lim}$, otherwise, $\sigma_i^t = 0$); $TL_j^{\min}$ and $TL_j^{\max}$ are, respectively, the minimum required and maximum allowable level for tank j; TL_j^t is the water level of tank j at time t; TL_j^{final} and TL_j^0 are, respectively, the water level at the beginning and end of simulation of tank j; $\mathbf{h}$ is the vector of nodal head; $\mathbf{q}$ is the vector of nodal demand; and G is a nonlinear equation of hydraulic model, namely, flow continuity and energy conservation equations. Additionally, considering the connectivity to the source, it has to be noted that the connectivity can be guaranteed when all the nodes in the network meet the three aforementioned constraints.

In this paper, the BORG algorithm (Hadka and Reed 2013) was applied to solve the multiobjective optimization problem. The BORG algorithm contains some new features compared with traditional multiobjective optimization algorithms such as NSGAII (Reed et al. 2013). These new features include: (1) an ε-box dominance archive established to maintain convergence and diversity in the whole search process; (2) a restart strategy presented to avoid premature convergence; and (3) a feedback mechanism established to adjust the adaptability for different problems. A new algorithm was tested and compared with other algorithms. A comparison between the BORG algorithm and other algorithms can be found in the literature (Reed et al. 2013; Wang et al. 2015; Zheng et al. 2016). The flowchart of boundary pipe optimization using the BORG algorithm is shown in Fig. 6.

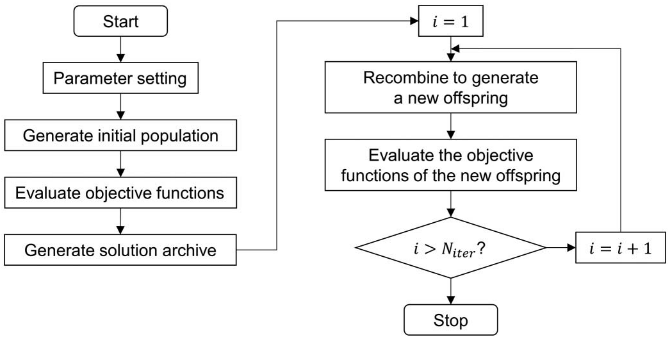
Start
Parameter setting
Generate initial population
Evaluate objective functions
Generate solution archive
$i = 1$
Recombine to generate a new offspring
Evaluate the objective functions of the new offspring
$i > N_{iter}$?
$i = i + 1$
Stop

Fig. 6. Flowchart of boundary pipes optimization using BORG algorithm

Using the BORG algorithm for boundary pipe optimization can be performed in the following seven steps: (1) set the required parameters; the minimum required and maximum allowable pressure of each node, the minimum required and maximum allowable level of each tank, the upper limit of water age and the parameters of BORG algorithm; (2) generate initial parent population; for each individual, random initializes the status of boundary pipes; (3) evaluate objective functions; for each individual, set hydraulic model parameters of the specified individual, then run hydraulic simulation using *EPANET* to calculate the objective functions and the aggregate constraint violation; and (4) generate a solution archive. First, sort the parent population based on their objective values and constraints. For any two individuals, (1) if both solutions are feasible, then choose the one with Pareto optimality; (2) if solution 1 is feasible while solution 2 violates constraints, then choose solution 1; (3) if both solutions are in violation of constraints, then choose the one that has a lower aggregate constraint violation. Finally, add the Pareto optimal solution to the archive; if no feasible solution have been found, then add the solution with the least aggregate constraint violation to the archive; and (5) recombination. Generate a new offspring using auto-adaptive multioperator procedure; assuming that a recombination operator requires d parents, select one parent randomly from the archive and select the remaining $d-1$ parents from the population using tournament selection; (6) evaluate the objective functions of the new offspring; give priority to the archive: if the offspring dominates at least one archive members, then add the offspring to the archive and remove a random dominated member; and else if the offspring is nondominated, then add the offspring to the archive. Otherwise, consider the population: if at least one population member is dominated by the offspring, then add the offspring to the population and remove a random dominated member; and else if no population members are dominated, then discard the offspring; and (7) repeat steps (5) and (6) until reach a fixed number of iterations. After a certain number of iterations, check if a restart process is required; if yes, interrupt the iterative process, clear and refill the population with all of the archive members, and the remaining population members are randomly generated by step (2). Once the restart process has completed, the iterative process continues until termination. Eventually, the individuals in the archive are the optimized solutions of boundary pipes.

Case Study

The case study system, as shown in Fig. 7, is the battle of water sensor networks 2 (BWSN2) (Ostfeld et al. 2008). The system supplies over 8.9×10^4 m^3 water per day and serves for about 150,000 people. The hydraulic model is made of 12,523 junctions, 14,822 pipes, 2 reservoirs, 2 tanks, and 4 pumps. Auto-partitioning of water distribution network is undertaken for the following three processes:

1. Mapping the network into a weighted undirected graph. Firstly, for the network without partitioning, run hydraulic model using *EPANET* to calculate the average pressure for each node, then calculate the weighted adjacency matrix according to the average pressure of each node. The corresponding computing method is described earlier in this paper.
2. Community detection. The range of random walk time is set to $[t_{\min}, t_{\max}] = [0.1, 10]$ and the random walk time increment is set to $\Delta t = 0.5$ according to the scale and complexity of the network. Considering the number of communities, since Grayman et al. (2009) decomposed the system into 43 DMAs, for the purpose of comparison, the system was decomposed into 43 communities (DMAs) using the proposed approach.

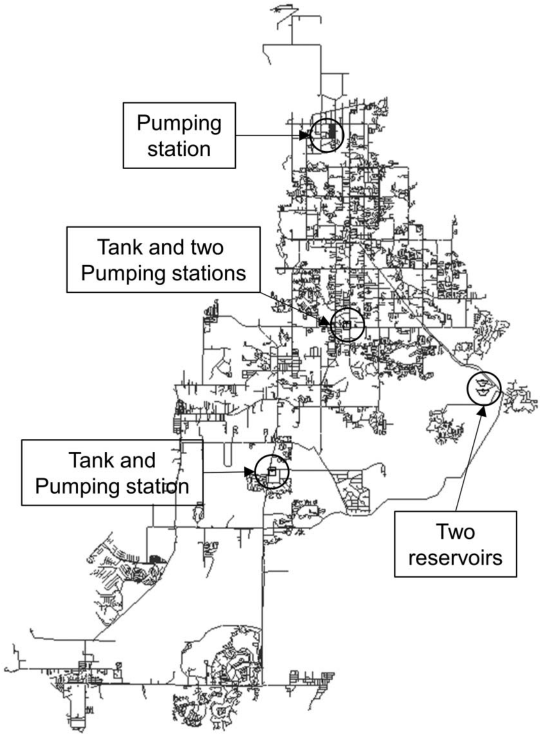

Fig. 7. Layout of example network

3. Boundary pipe optimization. The BORG algorithm (Hadka and Reed 2013) is applied to determine which pipe should be closed among the boundary pipes. The selection of the corresponding parameters mainly depends upon the water utility operators and local conditions. In this case, the minimum required

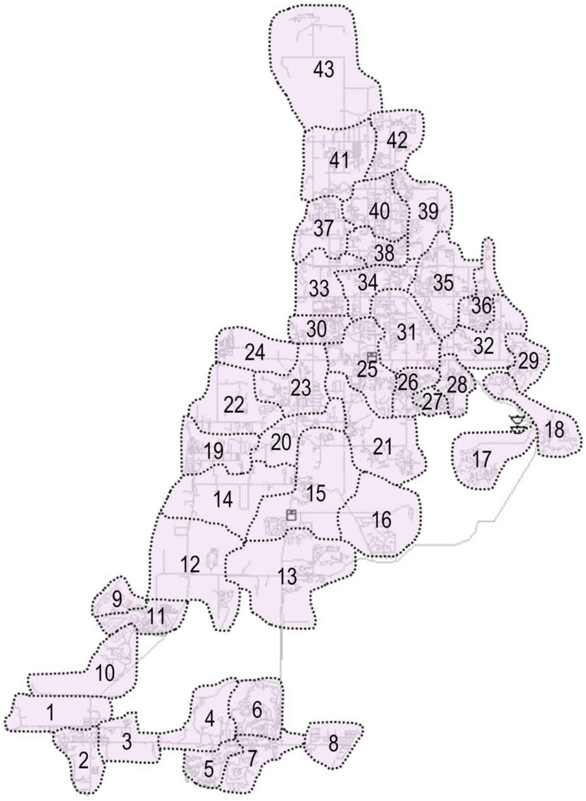

Fig. 8. Results of network partitioning

and maximum allowable pressure for each node is set to 15 and 70 m, respectively. The water age WA_{lim} limit is set to 60 h according to the operation data of the network. Since the network does not have tanks, there is no need to set up tank parameters. In addition, the parameters of the BORG algorithm are set to default values. Fig. 9 shows the calculating results.

Fig. 8 shows that the network is decomposed into 43 communities (DMAs) using the Louvain algorithm when random walk time $t = 3.6$. The corresponding modularity is 0.963 and reveals a strong community structure. A Pareto set of 11 solutions generated by the BORG algorithm is shown in Fig. 9. The set represents the approximate Pareto optimal from a total of 50,000 hydraulic model simulations, which took about 278 h using a desktop with a 3.00-GHz Intel Core i7-4710HQ with 16 GB of RAM in Windows 10. The computational efficiency of BORG is mainly affected by computer's performance and network's size. For large-scale and highly looped water supply systems, parallel distributed computing can significantly improve the computational efficiency.

Fig. 9 shows the Pareto approximate set from three-objective minimization. Engineers can choose the optimal solution according to the motivation of network sectorization. For example, if only for pressure control, with the results of Fig. 9(c), it is possible to see that the greater the number of boundary pipes, the worse the uniformity of pressure. Hence, engineers will choose the solution with the lower number of boundary pipes and install pressure control valves and flow meters on those pipes to improve pressure management.

A parallel line plot in Fig. 10 is used to further clarifies the relationships between f_1, f_2, and f_3, in which each solution is represented by a line across three objectives along the x-axis. The trade-off among objectives can be distinctly observed. The parallel line plot shows that the best solution should balance well with overall good performance. This case study considers the number of instruments that should be installed on the boundary pipes and water quality as well as the priority of the number of boundary pipes and water age uniformity. There is a total of 136 boundary pipes in the network; for the best solution (heavy line in Fig. 10), 33 pipes are closed, with only 103 boundary pipes remaining open.

In this case study, a 192-h extended period analysis was executed and the results of the last 24-h period are selected to calculate the average pressure and average water age for each DMA. Fig. 11 shows the number of boundary connections, Fig. 12 shows the average pressure, and Fig. 13 shows the average water age of each DMA before and after optimization. It can be observed that the number of boundary connections is significantly reduced after the optimization. The average pressure and the average water age of

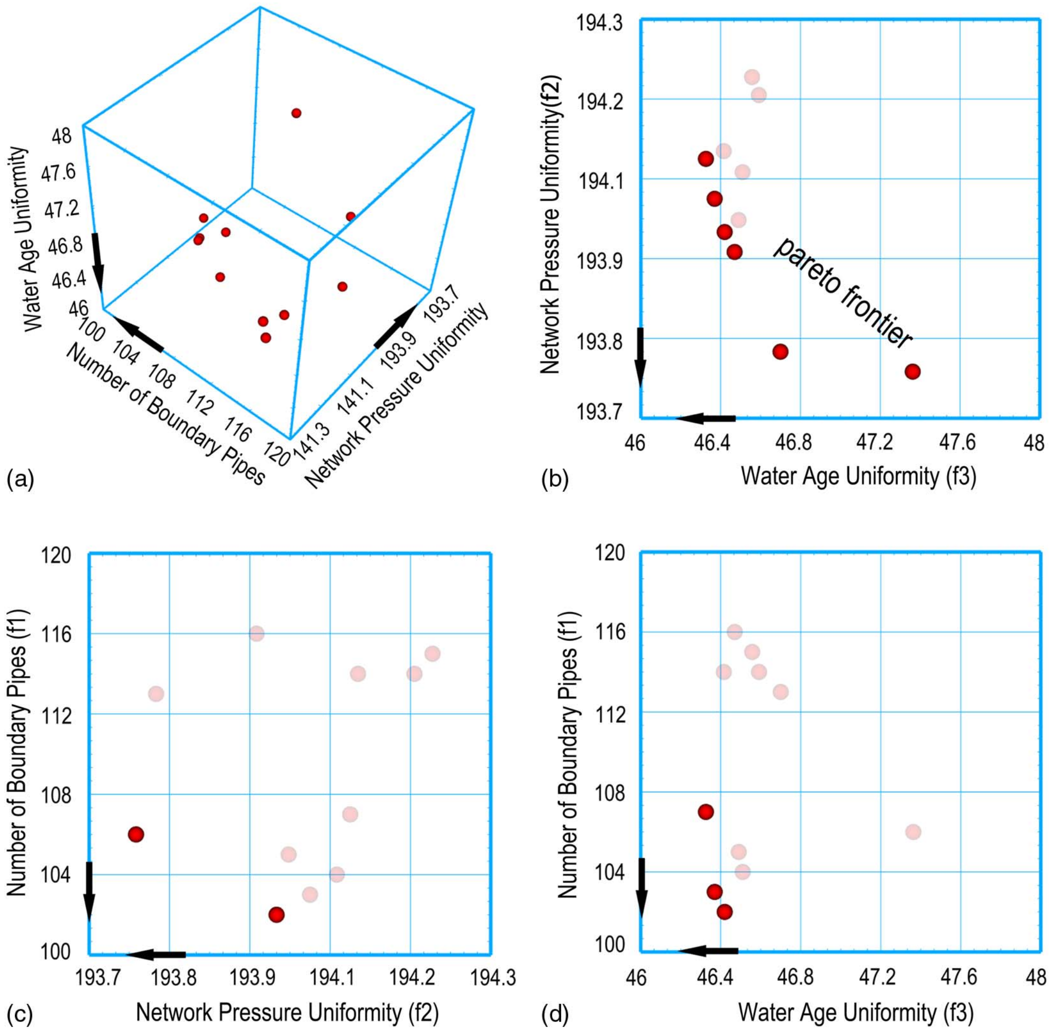

Fig. 9. Pareto approximate set from three-objective minimization with arrows showing directions of increasing preference: (a) the global view of Pareto approximate set; (b) network pressure uniformity versus water age uniformity; (c) number of boundary pipes versus network pressure uniformity; (d) number of boundary pipes versus water age uniformity

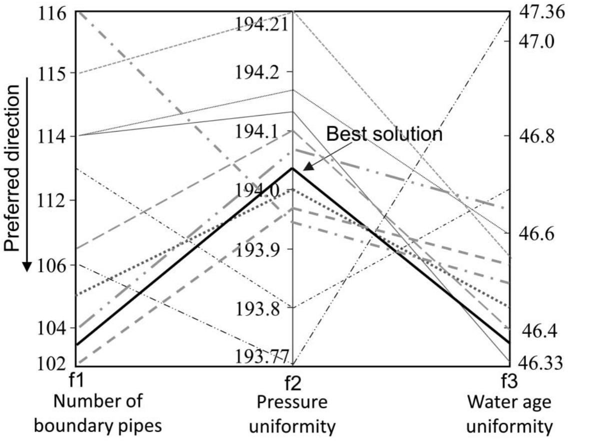

Fig. 10. Parallel line plot for the Pareto approximate set (each solution is represented by a line across three objectives in the *x*-axis)

each DMA have no apparent changes before and after optimization in this case study. This may be related to the characteristics of each WDN, such as the topological structure and nodal elevation. Each network has its own characteristics, e.g., for a mountainous urban network, the changes can vary significantly; however, for a plain urban network, the changes may not be obvious. Despite the small improvements in the average pressure and water age with the partition method for this specific case, the network is successfully divided into more appropriate DMAs. Thus, the proposed method in this paper will contribute to more effective DMA management. In addition, engineers can install pressure control valves in the inlet of some DMAs to further improve pressure management.

Conclusions

This paper presents a new method for network partitioning design based on the multiscale community detection method and multiobjective optimization. The method includes four main steps: (1) mapping the water distribution network into a weighted undirected graph; (2) a multiresolution modularity based on random walk theory, as presented in this paper, to measure the strength of division of WDNs; (3) network partitioning using a Louvain algorithm; and (4) determining which boundary pipes can be closed. A new hyperheuristic algorithm, BORG, is used here to optimize boundary pipes by minimizing three objectives: number of boundary pipes, network pressure uniformity, and water age uniformity. The proposed approach was applied to a real water distribution system. The results indicate that the proposed method can decompose the network into reasonable subnetworks and similar pressure and water age can be achieved for each DMA. Finally, it is worth noting that with the proposed method, like many other modeling techniques,

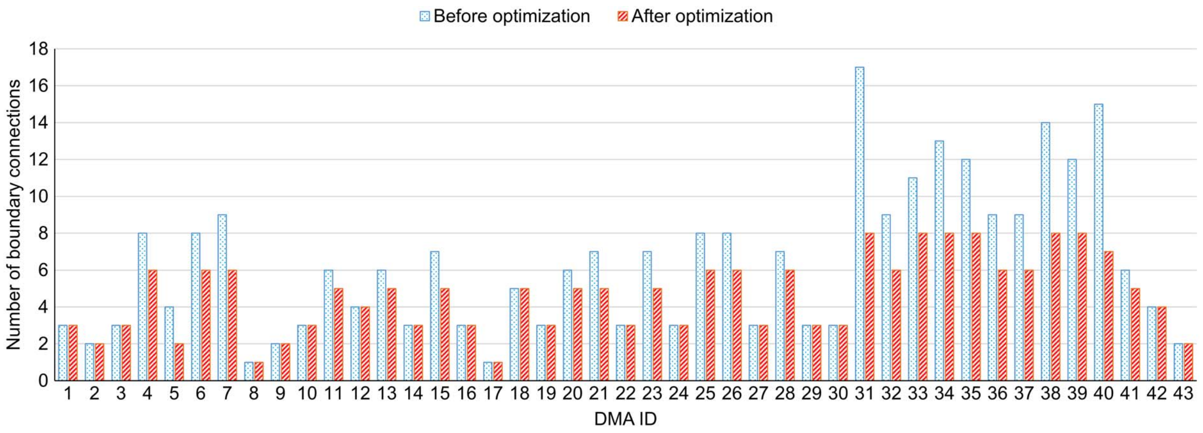

Fig. 11. Number of boundary connections of each DMA before and after optimization

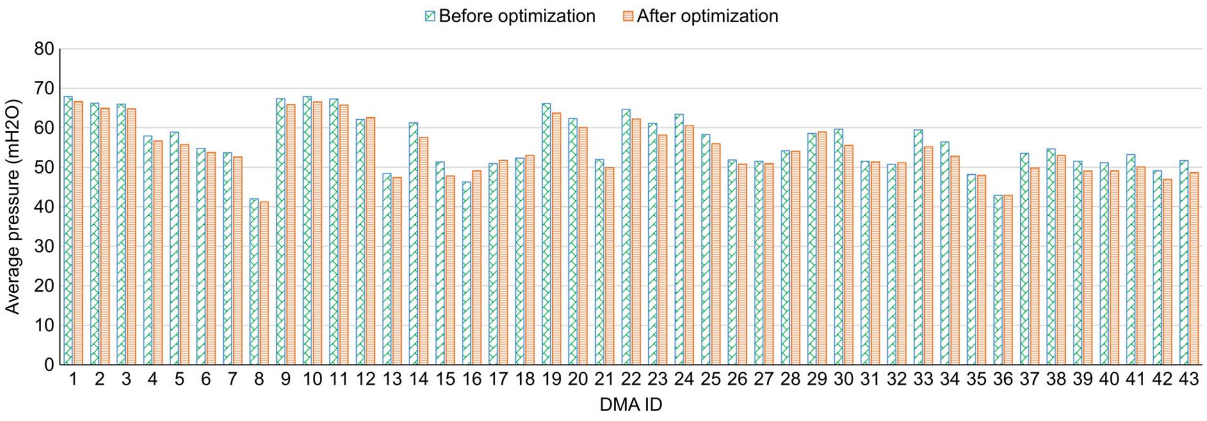

Fig. 12. Average pressure of each DMA before and after optimization

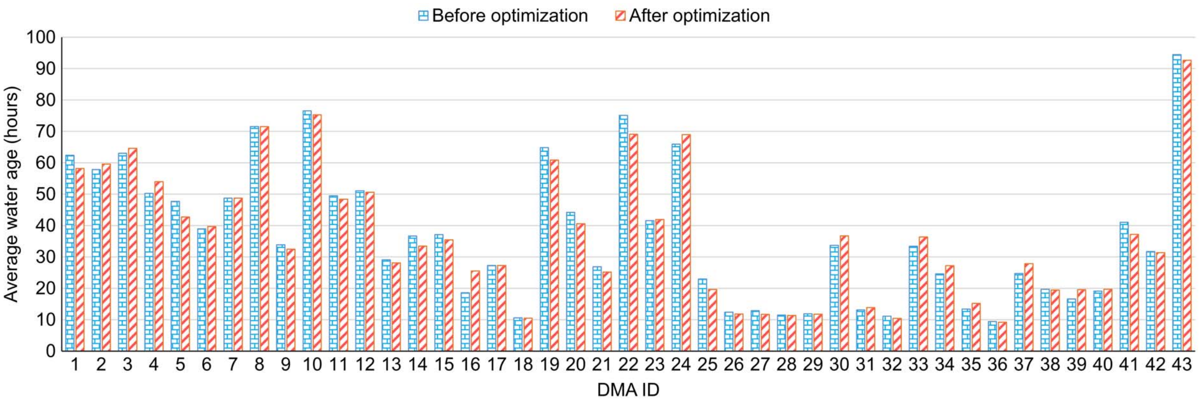

Fig. 13. Average water age of each DMA before and after optimization

good results for network partitioning can only be achieved in practical engineering by carefully applying the method with sound engineering judgment.

Acknowledgments

The part of work is financially supported by the National Science and Technology Major Project (2014ZX07406003).

References

Alhimiary, H., and Alsuhaily, R. (2007). "Minimizing leakage rates in water distribution networks through optimal valves settings." *World Environmental and Water Resources Congress 2007*, ASCE, Reston, VA, 1–13.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). "Fast unfolding of communities in large networks." *J. Stat. Mech: Theory Exp.*, 2008(10), P10008.

Campbell, E., et al. (2016). "A flexible methodology to sectorize water supply networks based on social network theory concepts and multi-objective optimization." *J. Hydroinf.*, 18(1), 62–76.

Campbell, E., Ayala-Cabrera, D., Izquierdo, J., Pérez-García, R., and Tavera, M. (2014). "Water supply network sectorization based on social networks community detection algorithms." *Procedia Eng.*, 89, 1208–1215.

Candelieri, A., Conti, D., and Archetti, F. (2014). "A graph based analysis of leak localization in urban water networks." *Procedia Eng.*, 70, 228–237.

Ciaponi, C., Murari, E., and Todeschini, S. (2016). "Modularity-based procedure for partitioning water distribution systems into independent districts." *Water Resour. Manage.*, 30(6), 2021–2036.

Deuerlein, J. (2008). "Decomposition model of a general water supply network graph." *J. Hydraul. Eng.*, 10.1061/(ASCE)0733-9429(2008)134:6(822), 822–832.

Diao, K. G., Zhou, Y. W., and Rauch, W. (2013). "Automated creation of district metered areas boundaries in water distribution systems." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)WR.1943-5452.0000247, 184–190.

Di Nardo, A., Di Natale, M., Musmarra, D., Santonastaso, G. F., Tzatchkov, V. G., and Alcocer-Yamanaka, V. H. (2015). "Dual-use value of network partitioning for water system management and protection from malicious contamination." *J. Hydroinf.*, 17(3), 361–376.

EPANET 2.0 [Computer software]. National Risk Management Research Laboratory, Cincinnati.

Farley, M. (2001). "Leakage monitoring and control." *Leakage management and control: A best practice training manual*, World Health Organization, Geneva, 58–98.

Ferrari, G., Savic, D., and Becciu, G. (2014). "Graph-theoretic approach and sound engineering principles for design of district metered areas." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)WR.1943-5452.0000424, 04014036.

Fortunato, S. (2010). "Community detection in graphs." *Phys. Rep.*, 486(3–5), 75–174.

Fortunato, S., and Barthelemy, M. (2007). "Resolution limit in community detection." *Proc. Natl. Acad. Sci. U.S.A.*, 104(1), 36–41.

Galdiero, E., De Paola, F., Fontana, N., Giugni, M., and Savic, D. (2016). "Decision support system for the optimal design of district metered areas." *J. Hydroinf.*, 18(1), 49–61.

Girvan, M., and Newman, M. E. J. (2002). "Community structure in social and biological networks." *Proc. Natl. Acad. Sci. U.S.A.*, 99(12), 7821–7826.

Giustolisi, O., and Ridolfi, L. (2014). "New modularity-based approach to segmentation of water distribution networks." *J. Hydraul. Eng.*, 10.1061/(ASCE)HY.1943-7900.0000916, 04014049.

Giustolisi, O., Ridolfi, L., and Berardi, L. (2015). "General metrics for segmenting infrastructure networks." *J. Hydroinf.*, 17(4), 505–517.

Grayman, W. M., Murray, R., and Savic, D. A. (2009). "Effects of redesign of water systems for security and water quality factors." *Proc., World Environmental and Water Resources Congress 2009*, ASCE, Reston, VA, 504–514.

Hadka, D., and Reed, P. (2013). "BORG: An auto-adaptive many-objective evolutionary computing framework." *Evol. Comput.*, 21(2), 231–259.

Hajebi, S., Roshani, E., Cardozo, N., et al. (2016). "Water distribution network sectorisation using graph theory and many-objective optimisation." *J. Hydroinf.*, 18(1), 77–95.

Herrera, M., Abraham, E., and Stoianov, I. (2016). "A graph-theoretic framework for assessing the resilience of sectorised water distribution networks." *Water Resour. Manage.*, 30(5), 1685–1699.

Lambiotte, R., Delvenne, J.-C., and Barahona, M. (2014). "Random walks, Markov processes and the multiscale modular organization of complex networks." *IEEE Trans. Network Sci. Eng.*, 1(2), 76–90.

Maier, H., et al. (2014). "Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions." *Environ. Modell. Software*, 62(9), 271–299.

Marchi, A., et al. (2014). "Battle of the water networks II." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)WR.1943-5452.0000378, 04014009.

Newman, M. E. J., and Girvan, M. (2004). "Finding and evaluating community structure in networks." *Phys. Rev. E*, 69(2), 026113.

Ostfeld, A., et al. (2008). "The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)0733-9496(2008)134:6(556), 556–568.

Perelman, L., and Ostfeld, A. (2011). "Topological clustering for water distribution systems analysis." *Environ. Modell. Software*, 26(7), 969–972.

Perelman, L. S., Allen, M., Preis, A., Iqbal, M., and Whittle, A. J. (2015). "Automated sub-zoning of water distribution systems." *Environ. Model1. Software*, 65, 1–14.

Perelman, L. S., Allen, M., Preisc, A., Iqbal, M., and Whittle, A. J. (2014). "Multi-level automated sub-zoning of water distribution systems."

Proc., 7th Int. Congress on Environmental Modelling and Software, San Diego.

Reed, P. M., Hadka, D., Herman, J., Kasprzyk, J., and Kollat, J. (2013). "Evolutionary multiobjective optimization in water resources: The past, present, and future." *Adv. Water Resour.*, 51(1), 438–456.

Scarpa, F., Lobba, A., and Becciu, G. (2016). "Elementary DMA design of looped water distribution networks with multiple sources." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)WR.1943-5452.0000639, 04016011.

Scibetta, M., Boano, F., Revelli, R., and Ridolfi, L. (2014). "Community detection as a tool for district metered areas identification." *Procedia Eng.*, 70, 1518–1523.

Wang, Q., Guidolin, M., Savic, D., and Kapelan, Z. (2015). "Two-objective design of benchmark problems of a water distribution system via MOEAs: Towards the best-known approximation of the true Pareto front." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)WR.1943-5452.0000460, 04014060.

Zheng, F., Simpson, A. R., Zecchin, A. C., and Deuerlein, J. W. (2013). "A graph decomposition-based approach for water distribution network optimization." *Water Resour. Res.*, 49(4), 2093–2109.

Zheng, F., Zecchin, A., Maier, H., and Simpson, A. (2016). "Comparison of the searching behavior of NSGA-II, SAMODE, and Borg MOEAs applied to water distribution system design problems." *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)WR.1943-5452.0000650, 04016017.