Designing an Optimal Water Quality Monitoring Network

Xiaohui Zhu[1,3], Yong Yue[1,*], Yixin Zhang[2], Prudence W.H. Wong[3], Jianhong Tan[4]

[1] Department of Computer Science and Software Engineering, Xi'an-Jiaotong Liverpool University, Suzhou, Jiangsu Province,
215123 P.R. China
[2] Research Institute of New-type Urbanization, Huai'an, Jiangsu Province,
223005 P.R. China
[3] Department of Computer Science, University of Liverpool, Liverpool,
L69 3BX United Kingdom
[4]Jiangsu Province Hydrology and Water Resources Investigation Bureau Suzhou Branch
Suzhou, Jiangsu Province 215011 P.R. China
{xiaohui.zhu, yong.yue, yixin.zhang}@xjtlu.edu.cn, pwong@liverpool.ac.uk,
jianhong_tan@qq.com

Abstract. The optimal design of water quality monitoring network can improve the monitoring performance. In addition, it can reduce the redundant monitoring locations and save the investment and costs for building and operating the monitoring system. This paper modifies the original Multi-Objective Particle Swarm Optimization (MOPSO) to optimize the design of water quality monitoring network based on three optimization objectives: minimum pollution detection time, maximum pollution detection probability and maximum centrality of monitoring locations. We develop a new initialization procedure as well as a discrete velocity and position updating function to optimize the design of water quality monitoring network. The Storm Water Management Model (SWMM) is used to model a hypothetical river network which was studied in the literature for comparative analysis of our work. We simulate pollution events in SWMM to obtain all the pollution detection time for all the potential monitoring locations. Experimental results show that the modified MOPSO can obtain steady Pareto frontiers and better optimal deployment solutions than genetic algorithm (GA).

Keywords: Optimal water quality monitoring network, multi-objective optimization, SWMM, closeness centrality

1 Introduction

River systems play a crucial role in the sustainable development of a community. However, industry and living activities are creating more and more pollutants to freshwater sources. It is estimated that 40 billion dollars are lost each year in China due to freshwater pollution events [1]. Water quality monitoring has become one of the routine efforts for environmental protection all over the world. However, monitoring water quality remains a very complex process due to the large number of factors to consider such as monitoring locations, selection of water quality parameters, monitoring frequency and identification of monitoring objectives [3]. The costs of building and operating an automatic monitoring station are also very high (about

500,000-600,000 dollars per station for construction and 14,000 dollars per year for operating and maintaining). Planning and optimizing water quality monitoring networks have been addressed since 1940s. Dozens of papers have been published on this subject [3,4,5,6].

In this paper, we proposed a modified MOPSO algorithm to optimize the design of water quality monitoring network based on three optimization objectives: maximum pollution detection probability, minimum pollution detection time and maximum centrality of a monitoring network. Two discrete functions were developed to calculate particle's velocity and update particle's position respectively. Experimental results show that our algorithm can get steady Pareto frontiers and obtain better optimal deployment solutions than genetic algorithm (GA).

2 Methodology

2.1 Hypothetical River Network

To compare our study results with the achievements given by the literature (Quyang et al. 2008; Telci et al. (2008, 2009)), we use the same hypothetical river network as Figure 1 shows. There are 6 inlet locations (1, 3, 5, 8, 10, 11), 5 intermediate locations (2, 4 ,6 ,7, 9) and 1 outlet location (12) in the river network. We assume that a pollution event can occur at any location randomly with the same amount of pollutant spilling and there is only one pollution event at each time. We simulate the water flows for 24 hours from 00:00 to 23:59 with a steady water flow of 10*ft*^{3}*/s* for each inlet location. The pollution event occurs at 10:00 and lasts for 1 hour during the simulation. We also assume that the pollutant concentration is 10*mg/L*. The remaining characteristics of the river network is shown in Table 1, which is the same as Telci used [7].

2.2 Hydraulic Simulations

The Storm Water Management Model (SWMM) is a dynamic rainfall-runoff simulation model used for single event or long-term (continuous) simulation of runoff quantity and quality from primarily urban areas [8]. It is widely used for dynamically simulating storm water runoff and drainage systems in urban areas. Here we use

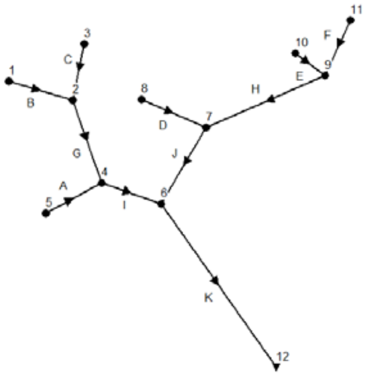


Fig. 1. Hypothetical river network

Table 1. Hydraulic characteristics of the river network

Catchment	Width (ft)	Channel's slope	Manning's coefficient	Flow rate (ft^3/s)
A	10	0.0001	0.02	10
B	10	0.0001	0.02	10
C	10	0.0001	0.02	10
D	10	0.0001	0.02	10
E	10	0.0001	0.02	10
F	10	0.0001	0.02	10
G	10	0.0001	0.02	20
H	10	0.0001	0.02	20
I	10	0.0001	0.02	30
J	10	0.0001	0.02	30
K	10	0.0001	0.02	60

Table 2. Pollution detection time at each potential locations

Locations	Pollution detection time and probability at each location											
	1	2	3	4	5	6	7	8	9	10	11	12
1	0	27	_	81	_	118	_	_	_	_	_	198
2	_	0	_	40	_	75	_	_	_	_	_	152
3	_	27	0	81	_	118	_	_	_	_	_	198
4	_	_	_	0	_	23	_	_	_	_	_	96
5	_	_	_	28	0	62	_	_	_	_	_	139
6	_	_	_	_	_	0	_	_	_	_	_	62
7	_	_	_	_	_	38	0	_	_	_	_	113
8	_	_	_	_	_	79	27	0	_	_	_	157
9	_	_	_	_	_	111	57	_	0	_	_	190
10	_	_	_	_	_	133	78	_	10	0	_	213
11	_	_	_	_	_	156	99	_	27	_	0	236
12	_	_	_	_	_	_	_	_	_	_	_	0

SWMM to simulate the hydraulic model, pollution events and pollutants transport along the river system.

We set the pollution detection threshold to 0.01*mg/L* and run hydraulic simulations in SWMM. Table 2 shows the simulation results of pollution detection time for each potential monitoring location. The "_" in Table 2 represents an infinite value, which means the pollution event cannot be successfully detected at a monitoring location.

2.3 Optimization objectives

a) Minimum Pollution Detection Time and Maximum Pollution Detection Probability

Telci et al. (2008) proposed a real-time optimal monitoring network design in river networks. They designed two optimization objectives of minimum pollution detection time and maximum pollution detection. A genetic algorithm (GA) was used to optimize the water quality monitoring network according to these two optimization objectives.

Here we also use the same two objectives in our algorithm. The detailed definition of these two optimization objectives can be found in [7].

b) Maximum Centrality of Monitoring Network

We argue that in practical environment, different potential monitoring locations may have different monitoring priorities, which should be considered when we design a monitoring network. In river network simulations, graph theory and network analysis are usually used to model river systems. The centrality is one of the most important indicators in graph theory. It can identify the importance of vertexes in a graph or network. Here we use the closeness centrality as an evaluation criterion of location priority. The closeness centrality for each potential monitoring location is described as Formula 1.

$$\mathrm{C(i)} = \left[\frac{\sum_{j=1}^{m} d(i,j)}{m-1}\right]^{-1} \tag{1}$$

where *m* is the total number of potential monitoring locations, *d(i,j)* is the length from location *i* to location *j*. The total closeness centrality for a potential deployment solution S_k is shown in Formula 2.

$$\mathrm{C}(S_K) = \sum_{i=1}^{n} C(S_{ki}) \tag{2}$$

where *n* is the number of monitoring devices deployed in a river network, S_{ki} is a monitoring location in deployment solution S_k.

The third optimization objective is to maximize the total closeness centrality for all deployment solutions, which is shown in Formula 3.

$$C(S) = \max\{C(S_1), C(S_2), \ldots, C(S_\mathrm{T})\} \tag{3}$$

where *T* is the total number of potential deployment solutions.

2.4 MOPSO Algorithm

Assume that we will deploy *n* monitoring devices in a river system out of *m* potential monitoring locations (*n* <= *m*). It is easy to know that the total number of potential deployment combinations T is:

$$T = \mathrm{C_m}^{\mathrm{n}} \tag{4}$$

We can find from Formula 4 that when we increase the value of *m* and/or *n*, the number of deployment solutions will be also increased exponentially, which is too large to calculate using enumeration search methods within a reasonable time. In addition, these optimization objectives normally conflict with each other, which means we aim to find some good trade-off solutions among multi objectives [9,10].

Multi-objective Particle Swarm Optimization (MOPSO) is one of the popular evolution algorithms used in recent year. Coello et al. (2004) compared MOPSO against three state-of-the art multi-objective evolutionary algorithms of Nondominated Sorting Genetic Algorithm II (NSGA-II), Pareto Archived Evolution Strategy (PAES) and

```
procedure INITIALIZATION (Integer k)
for i=1 to k do
    particle(i).position = [ ];
    particle(i).velocity = [ ];
    for j=1 to n do
        particle(i).position(j) = randomi(1, m);
        particle(i).velocity(j) = 0;
    end for
end for
end procedure
```

Fig. 2. Pseudocode of MOPSO initialization

```
procedure VEL_POS_UPDATING (int k)
MaxVel = round ((m − 1)/10);
for i=1 to k do
    for j=1 to n do
        particle(i).velocity(j) = round(w ∗ particle(i).velocity(j)
        +$c_1$ ∗ $r_1$ ∗ (particle(i).best.position(j) − particle(i).position(j))
        +$c_2$ ∗ $r_2$ ∗ (reph.position(j) − particle(i).position(j)));
        particle(i).velocity(j) = min(max(particle(i).velocity(j), −MaxVel), +MaxVel);
        particle(i).position(j) = particle(i).position(j) + particle(i).velocity(j);
        if particle(i).position(j) < 1 or particle(i).position(j) > m then
            particle(i).velocity(j).flag = −particle(i).velocity(j).flag;
            particle(i).position(j) = min(max(particle(i).position(j), 1), m);
        end if
    end for
end for
end procedure
```

Fig. 3. Pseudocode of velocity and position updating

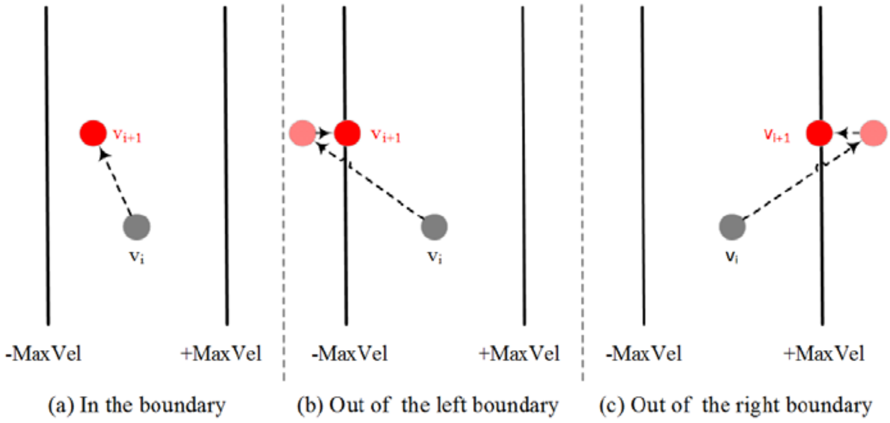
Microgenetic Algorithm for Multi-Objective Optimization (MicroGA) using 5 different test functions. Experimental results show that MOPSO has a highly competitive performance and can be considered a viable alternative to solve multi-objective optimization problems with low computational time. Here we use MOPSO to design an optimal water quality monitoring network. The velocity and position of particles during the computing iteration are updated by the following equations:

$$V_\mathrm{i}(t+1) = \mathrm{w}\, V_\mathrm{i}(t) + \mathrm{c}_1\mathrm{r}_1(pbest(i,t) - P_\mathrm{i}(t)) + \mathrm{c}_2\mathrm{r}_2(gbest(t) - P\,(t)) \tag{5}$$

$$P_\mathrm{i}(t+1) = P_\mathrm{i}(t) + V_\mathrm{i}(t+1) \tag{6}$$

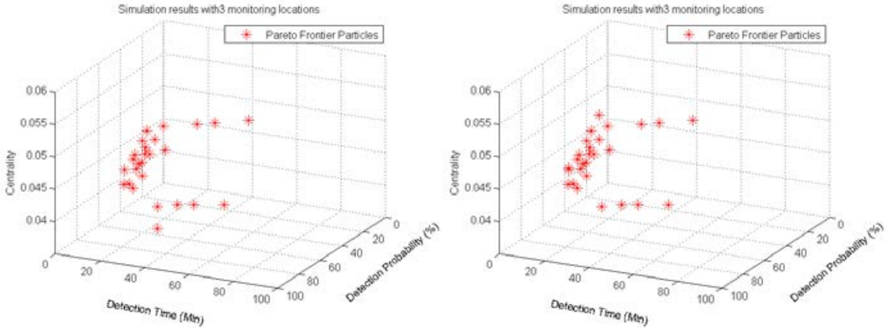
where V denotes the particle's velocity, w is an inertia weight constant, r_1 and r_2 are uniformly distributed random variables within range [0, 1], *pbest(i,t)* is the best position that the particle *i* has had, *gbest(t)* is the best position in all current particles, and c_1 and c_2 are positive constant coefficients for acceleration.

The classical MOPSO is a powerful algorithm to get global optimal results for continuous definition domains. However, it cannot be applied to discrete problems



(a) In the boundary (b) Out of the left boundary (c) Out of the right boundary

Fig. 4. Velocity updating process



(a) Pareto frontier 1 (b) Pareto frontier 2

Fig. 5. Hypothetical river network

directly. In this paper, we redesign the initialization procedure and the velocity and position calculation function to optimize this discrete issue.

a) Particle design and swarm initialization

Assume we select *n* locations to deploy water quality monitoring devices out of *m* potential monitoring locations in a river network (*n* <= *m*). Each potential monitoring location is named from *1* to *m* respectively resulting in a location set *S*={*1, 2, 3, …, m*}. Each particle in a swarm denotes a deployment solution with *n* monitoring locations. Therefore, each particle has *n* positions and each position represents a monitoring location in set *S*. As a result, particle *P* can be defined as a vector with *n* elements shown in Formula 7.

$$P = [p_1\, p_2 \ldots p_\mathrm{i} \ldots p_\mathrm{n}] \qquad \text{(7)}$$
$$subject\ to\ n \leq m\ \&\ 1 \leq p_\mathrm{i} \leq m$$

Assume we create *k* initial particles, we use a random integer function to initialize *n* positions and velocities for each particle in MOPSO. The swarm initialization

Table 3. Optimal deployment solutions

Monitoring locations	Detection time (min)	Detection probability	Centrality
6,9,12	45.8	100%	0.0414
6,7,12	54.8	100%	0.0455
4,6,9	34.9	91.7%	0.0500
2,6,7	36.4	91.7%	0.0514
4,6,7	44.6	91.7%	0.0561
4,7,9	29.4	83.3%	0.0487
2,4,7	34.3	83.3%	0.0505
2,7,9	14.8	66.7%	0.0451
2,4,9	14.9	66.7%	0.0455
2,5,9	13	58.3%	0.0420
5,7,9	10.7	50%	0.0447
7,8,9	7.4	41.7%	0.0444
2,4,5	10.8	41.7%	0.0466
5,9,11	2.5	33.3%	0.0379
2,3,5	6.75	33.3%	0.0401
5,8,10	0	25%	0.0399

procedure is as Figure 2 shows. The position value for all the particles is constrained between 1 and m. All the velocities are initialized to 0.

b) Velocity and Position Updating

Equations 5 and 6 show that the original velocity and position in MOPSO are both real values. However, we use integers to denote monitoring locations in a particle. Figure 3 shows a procedure to update particle's velocity and position. We let *MaxVel* be a maximum velocity during calculation. A *round* function is used to calculate a new integer value of velocity for each particle based on current *gbest* and *pbest* particles. Because the new velocity may be out of the boundary of [-*MaxVel*,+*MaxVel*], we use *max* and *min* functions to restrict the velocity scope. The detailed velocity updating process is shown in Figure 4.

3 Simulations and Analysis

We run the MOPSO algorithm several times based on data in Table 2 and get several Pareto frontiers as Figure 5 shows. The simulation results show that these Pareto frontiers are almost the same except for few (one or two) Pareto frontier particles. It means that the simulation results of MOPSO algorithm are quite steady and we can use it to design an optimal water quality monitoring network. Table 3 shows the optimal deployment solutions based on the Pareto frontier particles.

We can find from Table 3 that if we deploy 3 monitoring devices at locations 6, 9 and 12, we can detect all the potential pollution events and the pollution detection time is only 45.8 minutes. The centrality of this monitoring network is 0.0414. If we deploy monitoring devices at locations 6, 7 and 12, potential pollution events can still be all detected. However, the pollution detection time is increased to 54.8 minutes. If the monitoring devices are deployed at locations 4, 6 and 9. The pollution detection time is significantly decreased to 34.9 minutes while the detection probability is slightly decreased to 91.7%, which is also the second highest pollution detection probability solution in our deployment solutions and we can get a better centrality of 0.05. However, the second highest pollution detection probability in Telci's paper is only 83%.

4 Conclusions and Future Work

We presented a novel method based on a modified MOPSO algorithm to design an optimal water quality monitoring network with three optimization objectives of minimum pollution detection time, maximum pollution detection probability, maximum closeness centrality. Results show that the modified MOPSO can get a better optimal deployment solutions than GA.

In the future, this novel approach will be applied to a real case of water quality monitoring network. Further research is planned to explore the feasibility of redesigning the velocity and position calculation procedure to avoid same positions in a particle, which can further improve the computing performance.

Acknowledgements

This work was partly supported by the Natural Science Foundation of Jiangsu Province (BK20151245), the Natural Science Foundation of Huai'an City (HAG2015007), the Natural Science Foundation of Nantong City (MS12016048).

References

1. Yuqing Yang. How to Look at the GDP Receiving Much Attention[J]. Revolution, 2015(12):26-32.
2. Behmel S, Damour M, Ludwig R, et al. Water quality monitoring strategies - A review and future perspectives[J]. Science of The Total Environment, 2016: 1312-1329.
3. Park S Y, Choi J H, Wang S, et al. Design of a water quality monitoring network in a large river system using the genetic algorithm[J]. Ecological modelling, 2006, 199(3): 289-297.
4. Chilundo M, Kelderman P. Design of a water quality monitoring network for the Limpopo River Basin in Mozambique[J]. Physics and Chemistry of the Earth, Parts A/B/C, 2008, 33(8): 655-665.
5. Chang, Chia-Ling, and You-Tze Lin. A water quality monitoring network design using fuzzy theory and multiple criteria analysis[J]. Environmental monitoring and assessment 186.10 (2014): 6459-6469.
6. Ouyang, H. T., Yu, H., Lu, C. H., & Luo, Y. H. (2008). Design optimization of river sampling network using genetic algorithms. Journal of Water Resources Planning and Management, 134(1), 83-87.
7. Telci, I. T., Nam, K., Guan, J., & Aral, M. M. (2008). Real time optimal monitoring network design in river networks. In World Environmental and Water Resources Congress 2008: Ahupua'A (pp. 1-10).
8. Rossman, Lewis A. Storm water management model user's manual, version 5.0. Cincinnati, OH: National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency, 2010.

9. Reyes-Sierra, M., & Coello, C. C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International journal of computational intelligence research, 2(3), 287-308.

10. Coello C A, Lechuga M. MOPSO: a proposal for multiple objective particle swarm optimization[C]. congress on evolutionary computation, 2002: 1051-1056.

11. Telci, I. T., Nam, K., Guan, J., & Aral, M. M. (2009). Optimal water quality monitoring network design for river systems. Journal of environmental management, 90(10), 2987-2998.