Supporting Information

S1 The pseudo-codes for the improved Bayesian-MCMC, RWM, and GA

The flowchart of the improved Bayesian-MCMC, RWM and GA is presented in Fig. S1.

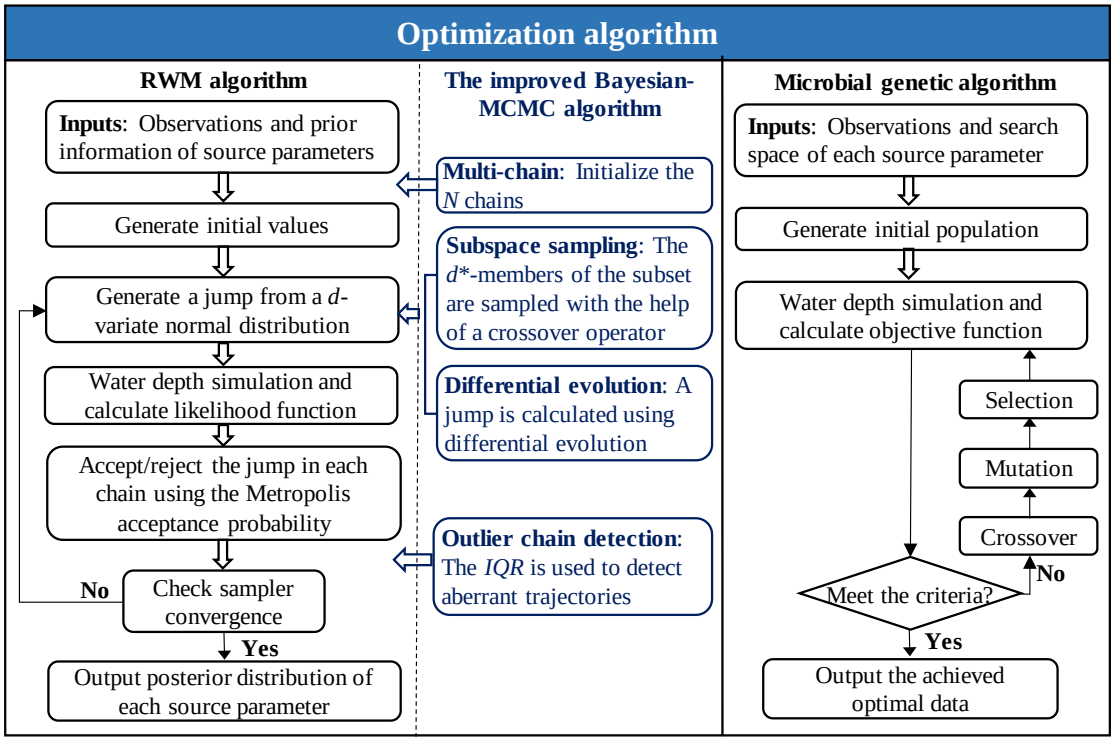

Fig. S1 Flowchart of the improved Bayesian-MCMC, RWM and microbial GA.

The pseudo-codes to perform the improved Bayesian-MCMC, RWM and GA are provided below. To obtain the complete source codes of the two cases in this manuscript, please contact us via E-mail address: yinhailong@tongji.edu.cn, m15801793070@163.com

1) Pseudo-code for random walk metropolis (RWM) algorithm

```
Function [theta, p_ theta] = RWM (prior, pdf, T, d)
theta (1, 1: d) = prior (1, d); % Initialize chain by sampling from prior
p_ theta (1) = pdf (theta (1, 1: d)); % Compute density initial state chain
sigma = (2.38/sqrt (d))^2 * eye(d); % covariance matrix proposal distribution
```

```
for t = 2: T, % Dynamic part
   d_ theta = mvnrnd (zeros (1, d), sigma); % d-variate normal proposal distribution.
theta p = theta (t-1, 1: d) + d_ theta; % Create candidate point
p_ theta p = pdf (theta p); % Calculate density proposal
p_acc= min (1, p_ theta p/ p_ theta (t-1)); % Compute p_acc
if p_acc > rand, % p_acc larger than U [0,1] ?
      theta (t, 1: d) = theta p; % accept proposal
      p_ theta (t) = p_ theta p;
else
theta (t, 1: d) = theta (t-1, 1: d); % reject proposal
      p_ theta (t) = p_ theta (t-1);
end
end
```

2) Pesudo-code for the improved Bayesian-MCMC algorithm

```
Function [theta, p_ theta] =Improved MCMC (prior, pdf, N, T, d)
theta = prior (N, d); % Create initial population
for i = 1:N, % Compute density initial population
p_theta (i, 1) = pdf (theta (i, 1:d));
end
[delta, c, c_star, n_CR] = deal (3, 0.1, 1e-6, 3); % Default of algorithmic parameters
CR = [1: n_CR]/ n_CR; pCR = ones (1, n_CR) / n_CR;
for t = 2: T % Dynamic part: Evolution of N chains
[-, draw] = sort(rand(N-1, N)); %Permute [1, …, N-1] N times
d theta = zeros (N,d); % Set N jump vectors to zero
      lambda = unifrnd (-c, c, N, 1); % Draw N lambda values
      for i = 1: N,
      D = randsample ([1: delta], 1, 'true'); % Set select. prob.
      a = R (i, draw(1: D, i)); b = R(i, draw(1: D, i)); % Extract vectors a and b not
equal i
      id = randsample (1:nCR, 1, 'true', pCR); % Select index of crossover value
z = rand (1, d); % Draw d values from U[0,1]
A = find (z < CR (id)); % A must contain at least one value
d_star = numel (A); % How many dimensions sampled?
gamma_d = 2.38/sqrt (2*D*d_star); % Calculate jump rate
d theta (i, A) = c_star*randn(1, d_star) + (1+lambda(i))* gamma_d*sum(theta (a,A)-
        theta(b,A),1); % Compute ith proposal
p_ theta p(i,1) = pdf (theta p (i, 1:d)); % Calculate density ith proposal
p_acc = min (1, p_theta p(i,1)./ p_theta (i,1)); % Compute acceptance probability
if p_acc > rand,
theta (i, 1:d) = theta p(i, 1:d); % p_acc larger than U[0.1]?
           p_theta (i, 1:d) = p_theta p(i, 1:d); % Accept proposal
else
d theta (i, 1:d) =0; %Set jump back to zero for pCR
```

```
end
      end
   [theta, p_ theta] = check (theta, mean (log (p_ theta(ceil(t/2):t, 1:N)))); % Outlier
detection and correction
end
```

3) Pesudo-code for the microbial genetic algorithm (MGA)

```
ga = MGA(dna_size, dna_bound, cross_rate, mutation_rate, pop_size)
[cross_rate, mutation_rate, pop_size] = deal (0.5, 0.5,100); % Default of algorithmic
parameters
Generate a population P randomly;
generation = 1
for generation <= max_gen
   Use a fitness function f(.) to evaluate each individual in P;
   % Adapting crossover rate
   if CP_avg > MP_avg
theta = 0.01 * (max(f(.))-avg(f(.)))/max(f(.))-avg(f(.));
      cross_rate = cross_rate + theta;
   else
      cross_rate = cross_rate – theta;
   end
   % Crossover
   for i: dna_size
if rand()< cross_rate
cross_idx(i) = True
else
cross_idx(i) = False
end
      loser_winner (0, cross_idx) = loser_winner (1, cross_idx) % assign winner gens
to loser
   end
   % Adapting mutation rate
   if CP_avg > MP_avg
theta = 0.01 * (max(f(.))-avg(f(.)))/max(f(.))-avg(f(.))
      mutation_rate = mutation_rate + theta;
   else
      mutation_rate = mutation_rate – theta;
   end
   % Mutation
   for i: dna_size
if rand()< mutation_rate
mutation _idx(i) = True
else
mutation _idx(i) = False
```

```
end
   loser_winner (0, mutation _idx) = ~ loser_winner (0, cross_idx)
   end
   % evolve
   % tournSelect performs a tournament selection
tournSize=options(2); % Get the number of tournaments
loser_winner_idx = sort (fitness)
loser_winner = crossover(loser_winner)
loser_winner = mutate(loser_winner)
   % compute CP_avg
CP = sum(f_child)-sum(f_parents); % sum(f_child) is the fitness sum of the two
offspring
   CP_avg = 1/nc * sum(CP);
   % compute MP_avg
   MP = f_new – f_old %f_new is the fitness of the new offspring
   MP_avg = 1/nm * sum(MP);
   Generation = generation +1;
End
```

S2 Running of the improved Bayesian-MCMC for the design case

For each of the four source parameters (i.e., source location x, source discharge q, starting discharge time T_1, and ending discharge time T_2), $\hat{R}$-statistic used to measure the convergence of the sampled chains during the iterations is shown in Fig. S2. The evolutions of the parallel Markov chains for each source parameter during the iterations are presented in Fig. S3.

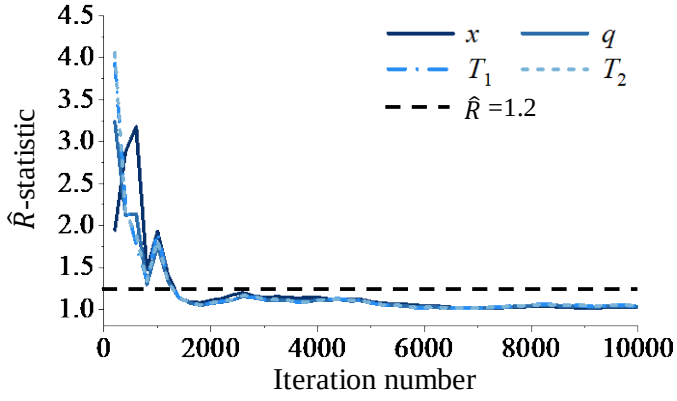

Fig. S2 The convergence of $\hat{R}$ values of the four parameters in Case I.

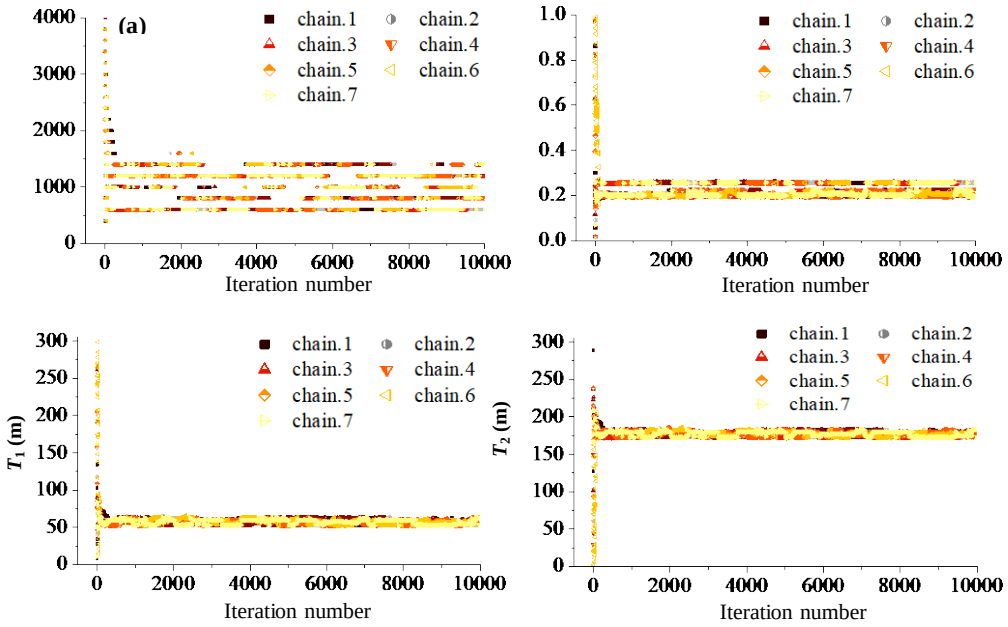

Fig. S3 Evolutions of the seven parallel Markov chains of the four source parameters with improved Bayesian-MCMC algorithm: (a) source discharge; (b) source location; (c) starting discharge time; (d) ending discharge time.

S3 Running of the improved Bayesian-MCMC for the real case

For the identified four source parameters (i.e., sewage inputs of q_3, q_{11}, q_{18}, q_{22}), $\hat{R}$-statistics to measure the convergence of the sampled chains during the iterations are presented in Fig. S4. The evolutions of the seven parallel Markov chains and the resulting posterior distributions are presented in Fig. S5. Figs. S4 and S5 correspond to the simulations under computational grid of 500 m.

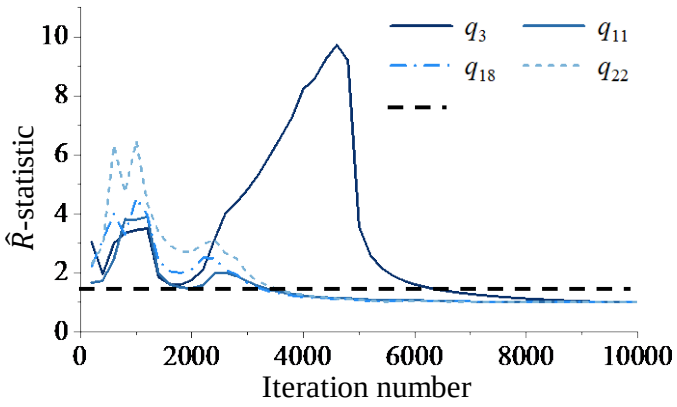

Fig. S4 Convergence of $\hat{R}$ values of the four identified source flows.

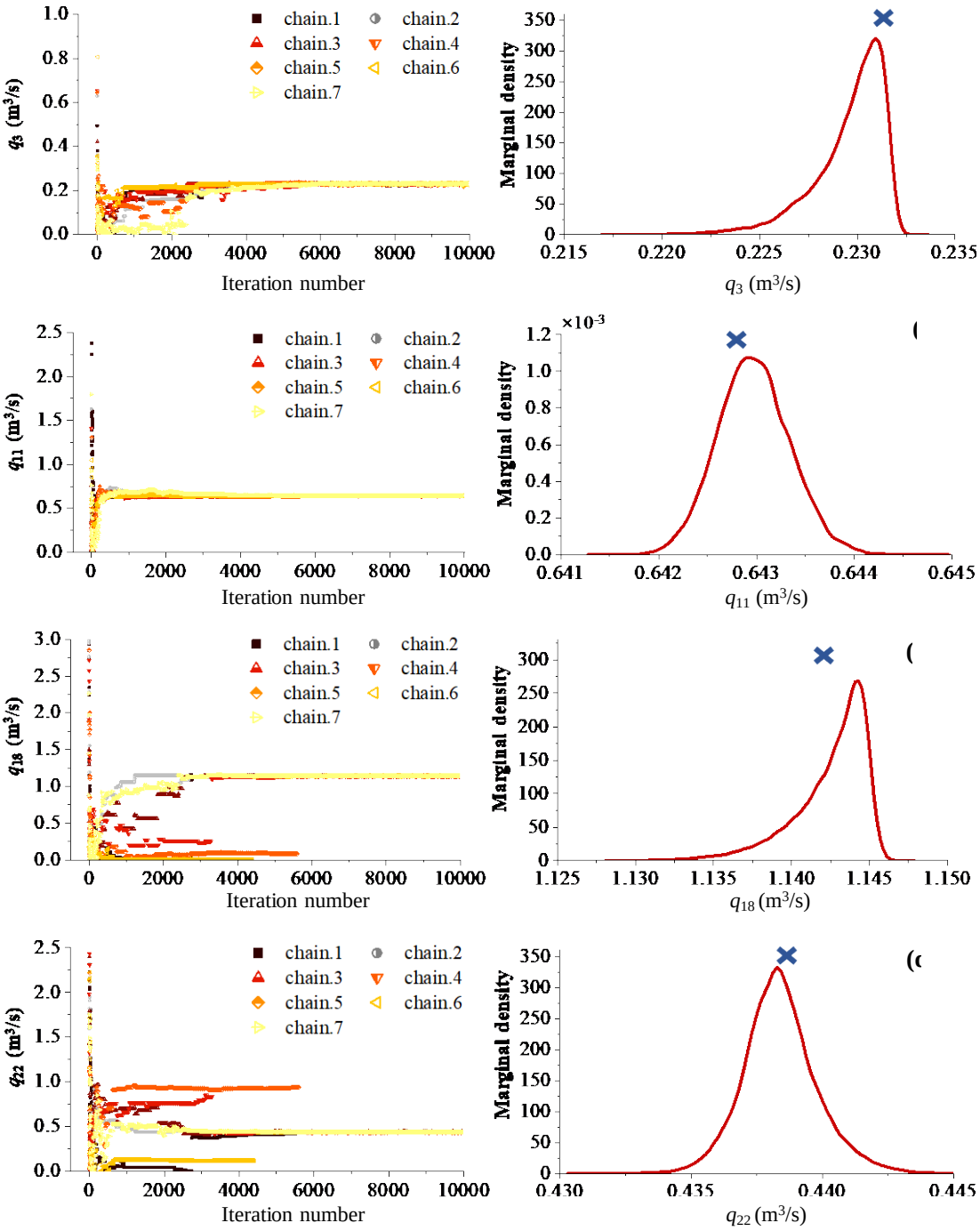

Fig. S5 Evolutions of the seven parallel Markov chains and posterior probability distributions of the four sewage inputs at (a) urban village, (b) Sili river, (c) Banqiao River and (d) municipal outlet.