



HAL
open science

Accurate and Optimal Sensor Placement for Source Identification of Water Distribution Networks

Hervé Ung, Olivier Piller, Denis Gilbert, Iraj Mortazavi

► **To cite this version:**

Hervé Ung, Olivier Piller, Denis Gilbert, Iraj Mortazavi. Accurate and Optimal Sensor Placement for Source Identification of Water Distribution Networks. *Journal of Water Resources Planning and Management*, 2017, 143 (8), 04017032, 12 p. 10.1061/(ASCE)WR.1943-5452.0000777 . hal-01547294

HAL Id: hal-01547294

<https://hal.science/hal-01547294>

Submitted on 26 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An accurate and optimal sensor placement for source identification of Water Distribution Networks

Ung Hervé¹, Piller Olivier², Gilbert Denis³, Mortazavi Iraj⁴

(1): Researcher, Irstea, 50, avenue de Verdun, F-33612 Cestas, France ; e-mail : herve.ung@gmail.com

(2): Senior Research Scientist, Irstea, 50, avenue de Verdun, F-33612 Cestas, France; e-mail : olivier.piller@irstea.fr

(3): Research Engineer, Irstea, 50, avenue de Verdun, F-33612 Cestas, France; e-mail : denis.gilbert@irstea.fr

(4): Professor; M2N, IMATH, CNAM, 3 rue Conté F-75141 Paris Cedex 03, France; e-mail : iraj.mortazavi@cnam.fr

Abstract

The problem of sensor placement for early-warning detection system is a topical issue for industry and utilities who want to equip their networks with such technology. It consists of finding the best sensor locations that optimize a criterion such as detection rate or time to detection. Few methods exist concerning the sensor placement that optimizes the result of a source identification method. This paper fills the gap by coupling an adjoint source identification method and a Monte-Carlo sensor placement algorithm. The first one is treated through the use of a backtracking algorithm. It uses binary responses at sensors to calculate the ranked list of potential contamination location nodes and contamination times. A criterion is then defined based on the source identification accuracy and specificity. Finally, two optimizing methods that maximize this criterion are proposed: a greedy algorithm and a local search algorithm which are both coupled with a Monte Carlo method to give the locations of sensors that are the best suited for allocating the source of a contamination. These methods are tested on a 2,500 node network to evaluate their efficiency.

Keywords

Water distribution network – Contamination – Inverse problem – Source identification – Sensor placement
– Monte Carlo – Greedy algorithm– Online contaminant detection

1. Introduction

Real-time sensors are quite a new topic. Multiple studies have focused on where to place the sensors and how to treat the data they produce. One possible use is the identification of the source of a contamination. Taking into account that we have an alarm generator based on non-specific quality sensors, the aim is to find the location of the source of the contamination by using the time and location of detection as well as the history of the network dynamics. We consider binary sensor responses: positive or negative. Both can be used to deduce the location, starting time and duration of a contamination.

The problem of source identification has been widely studied, and can be split into three categories. The first one is the enumeration type that consists of determining a subgroup of nodes that may be a source of contamination. For instance, De Sanctis et al. (2010) use a particle backtracking algorithm (PBA) and compute the system in reverse time to find the list of potential sources of contamination. Propato et al. (2007) and Propato et al. (2010) simulate all possible scenarios of contamination to create the matrix that links any source to any sensors. Any positive coefficient indicates that a contamination at a node for a corresponding time implies a sensor response at another time. That matrix can also be used to find the list of potential nodes of contamination.

The second type is the exploration type, which does the ranking. It uses the previous list and gives the probability for each node to be the true source of contamination. Guan et al. (2006) formulate a Least-Squares optimization problem. Laird et al. (2006) use a mixed-integer formulation to solve the source identification problem with concentration measurements at sensors. Liu et al. (2011) solve the same problem with the use of a logistic regression model; the work has also been done with binary sensors and as expected results are less good. Propato et al. (2007) and Propato et al. (2010) use the matrix of contamination they created and apply a minimum relative entropy method (MRE), to compute

probabilities and confidence bounds. Also, Preis et al. (2006a) couple Epanet with a genetic algorithm method.

Finally the last type regroups the methods that take into account the sensor's error rate and uses stochastic methods to calculate contamination source probabilities through the network. For instance, Dawsey et al. (2006) propose a Bayesian Belief Network (BBN). Also Perelman et al. (2010) develop a stochastic method on directed acyclic graphs (DAG) which creates clusters of nodes upon which the probabilities are then calculated.

The method developed here does both enumeration and exploration. It uses a backtracking algorithm to find the potential sources of contamination while creating the input/output matrix of contamination. The created matrix is the same as the one created by Propato et al. (2010) but restricted to only positive responses; it is also extended to large networks. Indeed, the previous method is forward in time and, therefore, needs to calculate all possible scenarios of contamination that may launch an alarm, which is time consuming. The reverse method proposed here uses sensor responses, either positive or negative, to tackle the adjoint equations. It takes considerably less time for large networks, which is important for a real time solving. That matrix will then be used to give rankings through counting positive values on rows and columns. The more a node can explain positive sensor responses, the more likely it has the chance to be source of contamination. By crossing information on two or more nodes, it is also possible to define a probability for multi-contamination for each group of nodes it may come from.

The paper written by Seth et al. (2016) introduces an evaluation method to test the capabilities of a source identification method. They define two criteria, the accuracy (Eq. 1) and the specificity (Eq. 2) as follows:

$$accuracy (\%) = \frac{Likeliness\ measure\ of\ the\ true\ injection\ node}{Highest\ likeliness\ measure\ over\ all\ candidate\ nodes} \times 100, \quad (1)$$

$$specificity (\%) = \frac{Number\ of\ nodes\ with\ lower\ likeliness\ than\ true\ injection\ node}{Total\ number\ of\ candidate\ nodes} \times 100. \quad (2)$$

Two close definitions of those criteria are used in this paper. The accuracy taken in this paper is a binary value based on whether or not the true source of contamination is in the list of potential contamination nodes. It is the characteristic function of the value given by Seth et al. (2016) formula. The specificity is then defined in this paper as to only concern scenarios where the source identification is accurate. This criterion also includes the number of nodes with lower likeliness than the true injection node. However it is here divided by the number of all the nodes of the network instead of the number of potential contamination nodes. Those two new definitions are adapted to an evaluation of a large number of contamination scenarios. The accuracy percentage indicates the number of scenarios when the source of contamination is included in the potential source node list. Therefore a large accuracy limits the number of scenarios when the true source of contamination is not found. And the specificity focuses on the capability of the source identification method to give a good rank to the true source of contamination among all the nodes of the network.

Another important issue for water network managers is the sensor placement design. Installations cost as well as material and data managing are factors that define how operators should buy and set their sensors. Optimization design algorithms are necessary to make the best suited choices for each network and objectives. Lee et al. (1992) were the first to tackle the problem on water distribution systems for the location of quality monitoring station. Then multiple authors proposed their formulation and methods which are reviewed in Rathi et al. (2014). At first it was with one single objective, such as the time to detection or the coverage, and then multi-objective methods appeared. For the Battle of the Water Sensor Networks (Ostfeld et al., 2008), several algorithms have been proposed to solve the sensor placement problem. One notable algorithm is based on the resolution of the p-median facility location problem by Berry et al. (2006) or Krause et al. (2006). It uses the fact that greedy algorithms based on non-decreasing submodular functions give near optimal solution. Propato et al. (2006) treat a close method with a mixed integer linear programming (MILP) formulation.

A few tackled the problem of sensor placement by optimizing the source identification criteria. Preis et al. (2006b) perform a clustering of the network and use a genetic algorithm. Propato et al. (2006) propose a method that minimizes the set of solutions given by the matrix of contamination, improving the source identification. Tryby et al. (2010) carry on that work and propose a method that improves the regularization of the matrix. They have reached the conclusion that detection likelihood and source identification are correlated. Liu et al. (2014) use a multi-genetic optimization algorithm and draw Pareto fronts for those two parameters.

This paper proposes new sensor placement methods that favor locations helping the source identification process. The backtracking source identification method will be used to do the ranking. A Monte Carlo method is associated with a greedy algorithm to calculate the best sensor locations. Even though the method is quite costly in time, it can be parallelized. Also, another solution is used based on local search methods and graph theory. From an initial guess of the best placement, the solution seeks placements that better comply with source identification criteria. The search is done with three distance adjacent nodes through multiple iterations.

This paper first presents the source identification method used as well as the backtracking algorithm. After that, it describes the multi-contamination method and then the criteria used for sensor placement. The algorithms based both on the Monte Carlo method, either greedy or local search, are given. Finally, results are interpreted concerning source identification and sensor placement on a test network.

2. Source identification

The source identification problem consists in estimating the location of the sources of contamination, their time of injection and duration in case of contamination of the water distribution network. The entries being the responses of the sensors located at strategic point inside the network.

Algorithm such as PBA can be used on sensor responses to define binary response of contamination of the network at the sensor location.

2.1. Equations and method

The equations (Eq. 3) and (Eq. 4) that are mostly used in water distribution network quality models are the one-dimension advection-reaction equation coupled with perfect mixing at all junctions. However, the scenario with no reaction term is the one that has the most impact in case of contamination of the water distribution network, therefore this will be the one that will be considered in the following.

$$\frac{\partial C}{\partial t} + u \frac{\partial C}{\partial x} = 0 \quad (3)$$

with C the concentration, t the time, x the space and u the average velocity of the pipe.

The following equation for perfect mixing is used:

$$C_{out} = \frac{\sum_{i \in \text{entries}} Q_i C_i}{\sum_{i \in \text{entries}} Q_i} \quad (4)$$

With Q_i is the flow rate of inflow pipe i ; and C_{out} is the concentration at the beginning of the outflow pipes calculated from the flow rate balance of the concentration at the end of the inflow pipes.

The contaminant goes from a source, travels through the network and may reach an installed sensor. The usual transport models used are either Eulerian or Lagrangian. The adjoint equations (Eq. 5) associated allow to go back in time from the sensor response to find the possible sources in the network.

$$\frac{\partial \psi}{\partial \tau} - u \frac{\partial \psi}{\partial \chi} = 0 \quad (5)$$

and $\tau = T_0 - t$ and $\chi = X_0 - x$.

2.2. Backtracking algorithm

The backtracking algorithm is a method based on the adjoint formulation of the transport equation on a graph. Similar solution was proposed by De Sanctis et al. (2010) to enumerate the potential sources of contamination. PBA is used here to build an input/output matrix as defined in Propato et al. (2010) but

beginning at positive sensor responses to accelerate the calculations. The negative responses are also used as presented afterwards but not as part of the matrix of contamination. That matrix is a binary matrix, where true values link the potential source of contamination to positive sensor responses. Negative responses help identifying times where nodes can't be source of contamination and therefore limit true values in the matrix.

The backtracking algorithm used is one of the Lagrangian adjoint forms. It takes advantage of the method of characteristics to find the time needed to cover the whole pipe length. The Lagrangian solution (Eq. 6) takes the form:

$$C(x + dx, t + \frac{dx}{u}) = C(x, t), \quad (6)$$

and its adjoint formulation (Eq. 7) is:

$$\psi(X - dx, \tau - \frac{dx}{u}) = \psi(X, \tau). \quad (7)$$

In case of a particle leaving a pipe, if u is constant, and $dx = L$ (length of the pipe), then $\frac{dx}{u}$ is the time needed to reach the pipe exit from the other end of the pipe. Particularly, hydraulic models often use constant velocity in each hydraulic time step. In this case the time of injection is obtained with the formula (Eq. 8):

$$\int_{T_B}^{T_{end}} u(t) dt = \begin{cases} 0 & \text{if reached first} \\ L & \text{else} \end{cases} \quad (8)$$

with T_{end} given (the time step calculated) and T_B looked for.

This may give multiple solutions, the one chosen is the first (smallest) that is not the trivial one ($T_B = T_{end}$).

Actually, hydraulic models often use constant velocity in each hydraulic time step. In this case, an algorithm has been developed considering such a time step Δt_h . For each pipe, it computes the time that

has been needed, for a particle to go through the whole pipe considering the velocity values in previous time steps used to compute this displacement. To consider the fact that velocity direction may change, a condition is added to stop the algorithm if the particle reaches its initial position:

$$i = 0; t_B = t_{end}$$

While $l < L$ AND $l > 0$: //with l the position of the particle

$l = l + u(t)\Delta t_h$ //with $u(t)$ constant for $t \in [(i - 1)\Delta t_h, i\Delta t_h]$ and i an integer representing current time step.

$$t_B = t_B - \Delta t_h$$

$$i = i - 1$$

End

$$\text{If } l < 0: t_B = t_B + |l|/u(t)$$

$$\text{If } l > L: t_B = t_B + |l - L|/u(t)$$

An example network is given in **Fig. 1**, with an injection at node 3. Constant velocities (no direction change) have been set for the water flows inside the pipes. The network is the one used in Neupauer et al. (2010) and its matrix of contamination is given in **Fig. 2**. The contaminant is injected at node 3 (black square). There are 3 sensors, at nodes 7, 35 and 43 (triangles).

To get the whole solution around the network, a recursive algorithm is used for the example in **Fig. 1**. The sensors 7 and 35 have detected a contamination as contrary to the sensor 43. The dark grey circle nodes (and node 3) are potential sources of contamination that can explain positive responses at sensors 7 and 35. The light grey circle nodes are either safe (nodes 1, 15 and 29 that are monitored by sensor 43) or not watched by the sensor set.

The computation is performed for each sensor positive response defining a location (sensor node) and a time. We calculate for every inflow pipe the node and the time from when the contamination could

have originated and repeat the algorithm for each new node reached. This method can go back far in time and induce a high computational cost. Indeed, the algorithm traces back the possible hydraulic way the contamination could have gone through and therefore could go back to the sources and the backtracked time can be several days. Therefore, the backtracking time (BT) is defined from when it is assumed that the contamination cannot have lasted that much time. This time needs to be calibrated from the maximum residence time inside the network or the average time to detection. Also, as will be shown in the results part, the efficiency of the source identification method depends on both BT and the average time to detection.

This algorithm allows from a sensor node and a response time to calculate every node location and time that can explain where this detection is coming from. Now, it can also be used in another way; if no contamination has been detected by the sensor, the algorithm can tell which node at which time cannot be a source of contamination. The negative responses are also backtracked and fill a preprocessing matrix that is used during the creation of the matrix of contamination. Indeed, for any backtracking, either for a positive or a negative answer, a test is done for every intersecting node of the network. If that node is marked as negative, from the preprocessing matrix, it will not generate another backtracking from that node.

Finally, both ways can be used simultaneously and for a large range of sensor times by launching all simulations and crossing the information. This allows the creation of the input/output matrix of contamination. It is the matrix that can illustrate which node at which time can explain which positive sensor at which time. Henceforth, it gives the list of potential sources of contamination.

2.3. The matrix of contamination

In **Fig. 2**, each row represents a positive sensor response at a time and the column a potential node at a time. The black points mean that the node at a time corresponding to a column can explain the positive sensor response at a time on a row. Each sub-diagonal matrix, such as the area delimited by

sensor 7 and potential node 9, represents the contamination duration. Indeed, in case of a velocity that does not vary a lot, subsequent contamination time steps can explain subsequent sensor responses.

Now, by using the input/output matrix of contamination, it is possible to find which node may be more likely to be the real source of contamination. Indeed, the node that best explains the sensor response is the one with the most non zero values in every row, all times taken together.

Table 1 shows the results for simplification (aggregation) in time of the matrix in **Fig. 2**, on the rows and the columns. Firstly, columns of the same node are combined in one column that takes the value 1 on each row having at least one non-zero value. Then all columns are averaged on the number of rows to give a score. **Table 1** gives the average score for each sensor. In fact, all sensors are weighted with their number of rows, or positive responses, but in this example both sensors have the same weight. The averaging on the rows gives an efficient weighting for ranking nodes to be the likely source of contamination. In this example, node 3 is the one that is most likely be the true source of contamination. It can explain 98% of the positive responses at sensors. However the mean taken is calculated on all responses, no matter the sensor it belongs to, and therefore the method favors the information given by the sensors with more responses.

The averaging method used is algebraic; instead, a product may have been used to define probabilities. Indeed, in case of one contamination only, node 7 in **Table 1** has a zero probability to be the source of contamination if the product of the scores in the rows of sensor 7 and 35 score is done. The method used here is more robust, all positive sensor responses have the same weight and, therefore, the results are less affected by sensor failures. Also it can be used in case of multi-contamination as contrary of the product probabilities, as explained in the next part.

Two variables of the backtracking algorithm are defined: the backtracking time (BT) and the observation time (OT) from first time to detection. These two variables define two real aspects of source identification. The first one is the time the algorithm goes back in time to find the source of contamination. The second one is the period of observation used after the first observation. The ability of a

sensor placement to do the source identification will depend on these two parameters. A large BT will allow finding contamination that happened far in the past, and a large OT means more information to solve the inverse problem. A good sensor placement tries to minimize these two parameters while keeping the same results. A small BT induces a good early warning as the contamination is detected fast. A small OT proves a good performance; the amount of observation time needed to accurately identify the contamination is small.

Note that this article will only focus on one source contamination but multi source contamination can also be treated with operations on the contamination matrix increasing considerably the computational time.

3. Sensor placement

Piller et al. (2015) perform several mono-objective optimizations solving on conflicting criteria such as minimization of time to detection, maximization of the detection likelihood and minimization of the fraction of population exposed. The method uses a greedy algorithm which is easy to compute. However, in some cases, it may not give the optimal solution. The aim of the optimization is to place the sensor so as to respect as much as possible the criterion chosen. A criterion defined as a weighted sum of the simple criteria can also be used instead.

In this paper, it is proposed to complete this approach with a new parameter being the best ranking for source identification, or *Contribution*. The aim is to look for the best placement of sensors in a water distribution network to monitor and detect contaminations. The method uses a coupling between the backtracking method presented before and a Monte Carlo method associated with a greedy algorithm. Then a local search on graph method to accelerate the algorithm is also given.

3.1. Sensor placement criterion

The usual criteria (Rathi et al. (2014)) for sensor placement are: average time to detection, detection likelihood, volume of water consumed, population exposed and extent of contamination. Some

other criteria may be used such as detection redundancy but few in the literature take into account the performance of a method of source identification linked with a specific set of sensor locations.

The criterion for ranking the potential sources of being the true source of contamination explained in part 2.3 (see Table 1) is modified and used in the following method to perform the sensor placement. The objective, which is named *Contribution* (Eq. 9), is defined as a real number between 0 and 1. It is defined as a non-dimensional ranking of the potential source over the total number of nodes. Considering a certain sensor placement and a certain contamination scenario, it estimates the capacity of that sensor placement to rank best the true source of contamination, as a potential node, among the node list.

$$Contribution = 1 - \frac{rank - 1}{N_{nodes} - 1} \quad (9)$$

with N_{nodes} the number of nodes and the last rank; and 1 the best rank.

If the rank is 1 then there is no node higher in the ranking, if is 0 then the node is the least likely to be the source of contamination. Also, it has been chosen that when a node is not a potential source of contamination, it is given the *Contribution* equals to 0.

Table 2 gives the example (injection at node 3) associated with the results of **Table 1** and the matrix of contamination of **Fig. 2**. The node 3 is the first in the list, therefore it has a *Contribution* equals to 1 associated with it. The others have an intermediate value between 0 and 1. In case of same criteria values, the smallest ranking is chosen, therefore nodes « 7 », « 9 », « 31 », « 33 » and « 35 » have equal rankings of 8. Then they are associated the value 0.63 equal to $1 - \frac{8-1}{20-1}$, 20 being the number of nodes in the network. The true source of contamination being the node 3, the criterion for the sensor placement is 1. If this would have been node 9 and the same values would have been found, the criterion would have been 0.63.

Therefore, from a sensor placement, and a given contamination, the method finds the *Contribution* of any potential source of contamination. The one corresponding to the true source of contamination can then be used to test the source identification method performance.

This criterion can also be used to perform the sensor placement method that is explained in the next section. The larger the *Contribution*, the higher is the source contamination potential node probability of being the true source of contamination. Depending on the contamination scenario, the sensor placement and the parameters BT and OT, three cases exist. The first one concerns the scenarios where the contamination is not detected, their *Contribution* is fixed at 0. The same applies for the scenarios where contamination source nodes are not inside the potential contamination node list. It can happen due to imprecision on either sensor response, water velocities or algorithm uncertainties. The accuracy (Eq. 10) of the source identification algorithm is then defined: the algorithm is accurate only if the source node is among the list of potential nodes of contamination,

$$accuracy (\%) = 100 \times \begin{cases} 0 & \text{if true source of contamination is not in the potential contamination list} \\ 1 & \text{else.} \end{cases} \quad (10)$$

Lastly, the remaining contamination scenarios are left. For each scenario we have computed the estimated rank of the real source of contamination as a potential contamination node. The specificity (Eq. 11) of the source identification algorithm is then defined as follows:

$$specificity (\%) = \left(1 - \frac{average\ ranking - 1}{N_{nodes} - 1} \right) \times 100 \quad (11)$$

with *average ranking* computed by considering the computed rank of the real source of contamination in all the remaining contamination scenarios. The specificity represents the percentage of nodes that are ranked worse than the source of contamination. A 100% specificity indicates that the source node is ranked first.

The scenarios of contamination belong either to the not detected scenarios, the not accurate or the ranked scenarios (Eq. 12):

$$N_{scenarios} = N_{not\ detected} + N_{not\ accurate} + N_{ranked} \quad (12)$$

The average Contribution (Eq. 13) can be calculated with the following formula:

$$\overline{Contribution} = \frac{\sum_{i=1}^{N_{scenarios}} (1 - \frac{rank(i) - 1}{N_{nodes} - 1})}{N_{scenarios}}. \quad (13)$$

with $rank(i)$ the computed rank of the true source of contamination for the scenario i .

It can also be calculated by combining Eq. 10, Eq. 11, Eq. 12 and Eq. 13 as a function of the specificity percentage, the accurate percentage and the detection likelihood with the following formula (Eq. 14):

$$\overline{Contribution} = \frac{specificity (\%)}{100} \times \frac{accurate (\%)}{100} \times detection\ likelihood. \quad (14)$$

3.2. Greedy algorithm

In this section, a new optimization criterion which allows to select the best sensor location for source identification, is proposed. It is defined as the placement that gives the best ranking to the real source of contamination-associated candidates. It uses the criterion defined Eq. 13, based on the ranking among all nodes of the network, that is the *Contribution* of a sensor placement for the localization of the source of contamination. Firstly, several contamination scenarios are defined as well as an empty list of sensors. All node locations are added to this list separately as temporary sensors and all contamination scenarios are tested giving the average *Contribution* for that temporary sensor. The node with the highest *Contribution* is then added permanently to the list of sensors. That list of sensors grows with one sensor at each iteration and is used as initialization for the next iteration. Each new sensor chosen will cover a new area of the network, indeed, a sensor placed next to a sensor already on the list won't improve the *Contribution* by much. The greedy algorithm can be used again, until the number of sensors corresponds to the value wanted.

An example of the result is given in **Table 3** which correspond to the network of Fig. 1 (but not the same contamination). Only the first 10 nodes and 11 (among 20) contamination scenarios are shown.

Each cell gives the value of the *Contribution* of the potential sensor associated to the contamination scenarios. Here, node 47 is, on average, the one which gives the best results. Then after several uses of the algorithm, node 35, and then 49 are chosen. These are in the same order in **Table 3** but it may not be the same for other examples.

It is an effective method; however it is quite time-consuming. The method has only been used for single contamination cases. For multi-contaminations, more computation time is needed and the algorithm has to be modified.

3.3. Local search on graph algorithm

The previous algorithm looks at each iteration for the node that best suits the criteria of source identification. As the search is done on the whole node list, this method is quite costly as shown in the next part. To accelerate the algorithm, an optimization based on local search on graphs method is used (example of local search on graph for coloring in Galinier et al. (2005)). It begins with an initial set of guessed best locations, which can be a list given by another method such as the average time to detection optimal placement given by Piller *et al.* (2015). Then the optimal solution is found by testing the *Contribution* source identification criteria on adjacent nodes. If the criterion has increased on adjacent nodes compared to the actual node then it is taken as the new sensor location in the next iteration. It is proven later that it may not give the optimal solution but local minima. To improve the method, adjacent to already-used adjacent nodes can be explored or a genetic algorithm be used. In this paper we have taken a three adjacent nodes distance.

3.4. Parallelization

Parallelization of the code has been necessary because of the large number of contamination scenarios that have been taking too much computational time in the next part (large network). Both precedent algorithms have been parallelized and launched on a cluster for computation. For the first algorithm, for each potential sensor node 2,000 contamination scenarios are tested on a test network case. Firstly, these 2,000 cases are divided into 20 groups of 100 because of the memory limit of the processors

needed for large networks. The contamination scenario groups are then divided by groups of processors depending on their number. Each processor is given a certain amount of scenarios to process. These scenarios are taken from the 100 scenarios assigned to that processor. To accelerate the algorithm an optimization process has been performed on the work list of each processor. To each scenario, a certain amount of time is associated that depends mostly on the scenario and on the sensor placement; therefore, a shuffle on the scenario list has been applied, and the same scenario might not be done by the same processor for another potential sensor. This allows us to average the calculation time charge for all processors, decreasing the total time of calculation. Indeed, for each iteration, all processors need to synchronize to decide on the sensor to add to the list, therefore the total time of calculation is based on the processor that takes the most time to calculate, and that mostly depends on the contamination scenario it needs to perform.

In the next parts the results are presented for both source identification method and sensor placement optimization.

4. Source identification results

A sufficiently large number of simulations of random one-contamination have been performed for the test network which has around 2,500 nodes. Each simulation lasts for 72 hours, and the contaminations are simulated between 24h and 48h for durations of 1h to 6h. During the first 24h no contamination is simulated not to affect the backtracking resolution accuracy; indeed 0h is, by construction, an indication that no contamination exists before that time. Simulations with a fixed concentration of 1mg/L of conservative contaminant have been taken to simulate sensor responses. The sensors are considered respecting the following hypothesis: any contamination value that is higher than 0.001 mg/L in the simulation is taken as a positive response, otherwise it is negative. The time of backtracking BT is taken as 24h and the *Contribution* criterion will be plotted as a function of the time of observation OT for both average time to detection and detection likelihood optimal placement given by Piller et al. (2015).

4.1. Ranking for 10 sensors

In **Fig. 3** the *Contribution* for both average time to detection (ATDOP) and detection likelihood (DLOP) optimal placement with 10 sensors has been plotted as a function of OT. For both, the *Contribution* increases with OT, proving the efficiency of exploiting new information by the source identification algorithm. The DLOP sensor placement is doing better than ATDOP, indeed it has a higher probability of detection and therefore is less impacted by the constraints (the *Contribution* equals to 0 if not detected).

When looking at the accuracy and the specificity, ATDOP is doing slightly better. In **Fig. 4**, the accuracy is increasing from 95 or 96% to 99% with OT increasing. Most contamination scenarios where the real source of contamination is not found by backtracking, and therefore not accurate, are probably because that the contamination happened before the backtracked time. This explains why ATDOP, which is the average time to detection optimal placement, performs better than the DLOP when observation time is low. **Table 4** shows, as expected, that ATOP has an average time to detection lower than DLOP but also has a lower detection likelihood. In **Fig. 4**, the specificity is going from near 93% to at most 96%, which is correct; indeed, this implies that the source node is on average ranked among the first 7% nodes which is equal to at most 175 nodes.

4.2. Ranking for 10, 20 and 50 sensors

It is also interesting to see the influence of the number of sensors on the ranking. In **Fig. 5** the *Contribution* for both ATDOP and DLOP have been plotted as a function of the number of sensors. The *Contribution* increases with the number of sensors for both sensor placements. This increase is not linear, the gain from 10 sensors to 20 sensors is not the same as between 20 and 50 and the asymptotic value seems lower for ATDOP than for DLOP.

In **Fig. 6** the *Contribution* is also given but this time as a function of the observation time for 10/20/50 sensors. For average time to detection optimal placement, the more sensors, the less it is impacted by the observation time. A similar behavior can be seen with the detection likelihood optimal

placement but with less sensitivity. Finally, average time to detection optimal placements are doing better than respective detection likelihood optimal placements for both accuracy and specificity as seen in **Fig. 7**. The sensor number impacts greatly on the source identification results but with less impact for each new sensor added, and the decision to choose the number of sensors to place depends on the four desired criteria: the type of optimization, the Contribution, the average time to detection and the detection likelihood, as given in **Table 5**.

5. Sensor placement results

The Contribution criterion studied in the previous section is now used as an objective for the optimal sensor placement problem. The greedy and local search on graph methods have been used on the test network with the same scenarios of contamination. Two cases have been taken, BT = 4h, OT = 2h and BT = 24h, OT = 2h with 10 sensors. The results are presented hereafter.

5.1. Greedy algorithm

The test network has been tested for the greedy algorithm. Two cases have been performed BT = 4h, OT = 2h (GOP4) and BT = 24h, OT = 2h (GOP24) with 10 sensors. It consists of testing every potential sensor location and determining, for all generated contamination scenarios, which location is best for the placement of the next sensor. Concerning the Contribution (see **Fig. 8**), both DLOP and GOP24 are performing best compared to GOP4 in third and ATDOP in last place. **Table 6** gives the average time to detection and the detection likelihood of the two sensor placements. The detection likelihood is in the following ascending order ATDOP, GOP4, GOP24 and DLOP. Concerning the accuracy and specificity (see **Fig. 9**), GOP4 performs better and equal respectively with GOP24. Depending on the chosen performance criteria, either of the following algorithms is optimal: GOP4 for average time to detection and accuracy; DLOP for detection likelihood; and GOP24 for specificity.

Fig. 10. shows the time it takes, at each iteration, to find a new sensor location with 120 processors, which is not the cumulated time. The first iteration takes less than half an hour, then the

execution time increases, until it reaches an asymptotic slope, adding less than 10 minutes for each new sensor from the last execution time. The time of execution increases with the number of sensors set because the complexity increases. As sensor placement is not done in real time conditions, the execution times are yet not a problem on a sufficient cluster. However, it is believed that for bigger networks it might be problematic because the problem is NP-hard. Another algorithm, based on local search on graph is developed to decrease the execution time.

5.2. Local search algorithm

A local search algorithm with 3 adjacent node distance has been used for 10 sensors. At each iteration, a local search around the 10 potential sensors location is performed concerning the source identification criteria. Each potential sensor is replaced with the local neighbor that performed the best. Firstly the evolution of the criteria is given, defined in the first section, as a function of the iteration. Then the last sensor placement performance is tested for source identification for both initialization with average time to detection and detection likelihood optimal placement.

Four cases have been performed $BT = 4h$, $OT = 2h$ (LSATDOP4, resp. LSDLOP4) and $BT = 24h$, $OT = 2h$ (LSATDOP24, resp. LSDLOP24) with ATDOP (resp. DLOP) initialization and 10 sensors.

In **Fig. 11** and **Fig. 12** the evolution of the *Contribution* is shown, defined in the Sensor Placement criterion section, as a function of the iterations. It does not converge to one solution, but oscillate around one, because all potential sensors are changed at the same time for each iteration to accelerate the algorithm. A convergence criterion can be defined to stop the algorithm and returns the best placement found. In such case, it can be seen that the algorithm has converged in all cases in fewer than 10 iterations. The *Contribution* values are slightly better when the initialization is taken as detection likelihood optimal sensor placement. The time for each iteration is constant around 700 seconds for 60 processors and therefore the algorithm converges in around one hour, which is much faster than the previous algorithm.

The *Contribution* scores (**Fig. 13**) in ascending order are ATDOP, LSATDOP4, LSATDOP24, LSDLOP4, LSDLOP24 and DLOP. The *Contribution* values are between ATDOP and DLOP

Contribution. The same applies for average time to detection and detection likelihood as shown in Table 7. Finally, as seen in **Fig. 14**, LSATDOP24 and LSATDOP4 perform best concerning accuracy and specificity. As discussed before, the *Contribution* criteria is not enough to decide which sensor placement to choose, detection likelihood, average time to detection, accuracy and specificity are also decision factors. If average time to detection is favored, LSATDOP4 may be chosen because it has better results than ATDOP and almost the same average time to detection. In case of favouring detection likelihood, LSATDOP24 and LSDLOP24 may be chosen because they have better accuracy and specificity than DLOP. LSDLOP4 is good on average for average time to detection and detection likelihood. In conclusion this method is giving better solutions overall than initial ones and which parameters and initialization to use depend on the results we want to favor.

5.3. Evaluation

We have seen that different objectives can influence the choice for a source identification optimal sensor placement: the average time to detection, the detection likelihood, the accuracy, the specificity and finally the *Contribution* which is a combination of the last three. The rankings for all these parameters have been written in **Table 8** and the mean over all parameters is given with $BT = 24h$ and $OT = 2h$. Overall, the greedy algorithm solutions are the best followed by the local search algorithm solution LSATDOP4, LSATDOP24 and LSDLOP24. This ranking should however be used cautiously and different weights can be used depending on which is considered more important for the decision maker.

To conclude this section, we have described two methods: a greedy algorithm and a local search on graph to maximize the *Contribution* criterion. Both were better for source identification (accuracy and specificity) than optimization based on average time to detection and detection likelihood. However, these methods are time-consuming. The greedy algorithm execution time is between 30 minutes to 3 hours on 120 processors with each new sensor added. The local search on graph showed best results with both average time to detection initialization, $BT = 4h$, $BT = 24h$, $OT = 2h$ criteria and detection likelihood initialization, $BT = 24h$, $OT = 2h$ criteria. The algorithm converges in approximately one hour in total on

60 processors. The local search solutions are not as good as the greedy ones but can be chosen as their results are not too far and the time of execution is much faster.

Conclusions and perspectives

The objective of this paper is to determine optimal sensor placement methods that give the highest *Contribution* for the source identification. One key step is to have a method that allows the measurement of that *Contribution* for a given sensor placement. For that, a new algorithm of source identification has been developed that uses binary sensor responses and works for large networks.

To take into consideration detection likelihood, accuracy and specificity criteria in the sensor placement for source identification, one criterion is defined that is called *Contribution*. This is a dimensionless variable between 0 and 1 that depends on the ranking. The lowest ranked nodes are given the value 0 and the most likely to be the source of contamination is given the value 1. This criterion is averaged over several contamination scenarios. The not-detected scenarios as well as the not-backtracked scenarios are given the value 0. The *Contribution* is then computed as the average of the results given by all the contamination scenarios simulated.

The novel method of source identification uses a backtracking algorithm to construct the input/output contamination matrix in real time. The backtracking is based on the adjoint of the transport equations. It computes those in reverse time, beginning at sensors, to enumerate potential nodes of contamination. The use of the backtracking methods allows handling large-size networks that would require huge calculation time if it was performed with a forward scheme. Indeed, the time needed to run every possible contamination is exponential with the size of the network. The matrix obtained is then analyzed (with simple manipulation such as aggregation) to evaluate the number of positive sensor responses that each potential source of contamination can explain. This score allows the creation of a

ranking list among the potential source of contamination. The method uses positive answers to find potential nodes but also processes negative answers to eliminate candidates. It can also be extended to multi-contamination but is quite costly in time; extended covering methods may be developed to overcome this drawback.

The methods have been tested on a real French network with around 2,500 pipes. The influence of the backtracking time (BT) as well as the initial sensor placement for the local search method has been carried out. Eight parameter/method combinations have been defined: Two come from the application of the Piller et al. (2015) method each based on average time to detection (ATDOP) and detection likelihood (DLOP) maximization through a greedy algorithm. Six method/parameter scenarios have been added to ATDOP and DLOP. The following two come from a greedy algorithm but with objective being the maximization of the *Contribution* criteria in the two parameter cases BT being long, 24h, or short, 4h, and observation time (OT) being short set at 2h. The fourth last sensor placements assessed are results of a local search algorithm in the same configuration than for the greedy algorithm but with two different initializations. The first two begin with ATDOP, the other two launched from DLOP.

It was found that the criteria are conflicting. When comparing ATDOP and DLOP, the first one is performing better with average time to detection, accuracy and specificity, however it performs worse concerning detection likelihood than the second one. It can be concluded that ATDOP performs better the source identification but on less coverage of the network. Also, adding new sensors to ATDOP and DLOP improves the results for the criteria, however each new added sensor brings fewer result improvements each time. Particularly ATDOP gets less detection likelihood improvement than the other placement, impacting the overall score for the source identification performance.

Concerning the backtracking time BT parameter, a small value (e.g. 4h) induces a small average time to detection score. This parameter is important, because the results show that a small average time to detection is performing better but on a smaller scale than with a big detection likelihood. For the

observation time OT, 2h period seems to be enough to have an ability to identify the contamination, still a longer observation time improves the results.

The placements based on the first method presented in this paper, greedy algorithm with *Contribution* objective, perform the best overall, however they are the longest with an increasing iteration execution time going from half an hour to three hours on 120 processors. The solutions given by the local search are second and the method is much faster than the previous one taking three hours overall on 60 processors. Finally ATDOP and DLOP are less suited for source identification than the two previous methods but are fast to compute.

The choice of the sensor placement method for source identification will depend on the objectives and the network properties as well as the number of sensors available. More research work should focus on evaluating the impact of hydraulic and transport model error in the final optimal designs. In the first case the velocity may not be accurate or not the same as in the simulations used to place the sensors. In the second case, transport model, reaction coefficients and sensor thresholds need careful attention. Results from the SMaRT-Online^{WDN} project have shown that imperfect mixing at cross and double T-junction may have an impact on the contamination spreading, the same applies when adding the dispersion effect to the transport model. The two modifications can influence the simulation of the contamination scenarios, therefore also change the sensor responses, and finally influence the sensor placement optimization.

Acknowledgments

We wish to acknowledge the following institutes for letting us use their computation cluster to carry out the simulations performed in this project: Mésocentre de Calcul Intensif Aquitain (MCIA), Avakas supercomputer

The research part of the SMaRT-Online^{WDN} project (SMaRT-Online^{WDN}, 2017) is supported by the German Federal Ministry of Education and Research (BMBF; project: 13N12180) and by the French Agence Nationale de la Recherche (ANR; project: ANR-11-SECU-006).

References

- Berry, J., Hart, W. E. and Phillips, C. A., Uber, J. G., Watson, J.-P. (2006).** *Sensor Placement in Municipal Water Networks with Temporal Integer Programming Models*. s.l. : Journal of Water Resources Planning and Management, 2006. Volume 132, SPECIAL ISSUE:Drinking Water Distribution Systems Security.
- Dawsey, W., Minsker, B. and VanBlaricum, V. (2006).** *Bayesian Belief Networks to Integrate Monitoring Evidence of Water Distribution System Contamination*. s.l. : Journal of Water Resources Planning and Management, 2006. Volume 132, SPECIAL ISSUE:Drinking Water Distribution Systems Security.
- De Sanctis, A., Shang, F. and Uber, J. (2010).** *Real-Time Identification of Possible Contamination Sources Using Network Backtracking Methods*. s.l. : Water Resour. Plann. Manag., 2010. ASCE 136 (4). 444-453..
- Galinier, P. and Hertz, A. (2006).** “A survey of local search methods for graph coloring.” *Comput. Oper. Res.*, 33(9), 2547–2562.
- Guan, J., Aral, M. and Maslia, M., Grayman, W. (2006).** *Identification of Contaminant Sources in Water Distribution Systems Using Simulation Optimization Method: Case Study*. s.l. : Journal of Water Resources Planning and Management, 2006. 07/2006; 132:252-262. DOI: 10.1061/(ASCE)0733-9496(2006)132:4(252).
- Krause, Andreas, Leskovec, Jure and Isovitsch, Shannon, Xu, Jianhua, Guestrin, Carlos, VanBriesen, Jeanne, Small, Mitchell, Fischbeck, Paul. (2006).** *optimizing sensor placements in water distribution systems using submodular function maximization*. s.l. : PROCEEDINGS - Water Distribution Systems Analysis Symposium 2006, 2006. 10.1061/40941(247)109.
- Krause, A., Leskovec, J., Isovitsch, S., Xu, J., Guestrin, C., VanBriesen, J., Small, M., and Fischbeck, P. (2006).** “*Optimizing sensor placements in water distribution systems using submodular function maximization*”. *Proc., 8th Annual Water Distribution System Analysis Symp., Dept. of Civil and Environmental Engineering, University of Cincinnati, Cincinnati, Ohio USA, August 27-30 2006*, printed by ASCE, 17 pages. doi:10.1061/40941(247)109.
- Laird, C. D., Biegler, L. T. and Van Bloemen Waanders, B. G. (2006).** *Mixed-integer approach for obtaining unique solutions in source inversion of water networks*. s.l. : Journal of Water Resources Planning and Management, 2006. Special Issue on Drinking Water Distribution Systems Security 132, 242 (2006).
- Lee, B. H. and Deininger, R. A. (1992).** *Optimal Locations of Monitoring Stations in Water Distribution System*. s.l. : Procedia Engineering - Computing and Control for the Water Industry (CCWI2015) Sharing the best practice in water management, 1992. Volume 119, 2015, Pages 1308–1317.
- Liu, L., Sankarasubramanian, A. and Ranjithan, S. R. (2011).** *Logistic regression analysis to estimate contaminant sources in water distribution systems*. s.l. : Journal of Hydroinformatics - & IWA Publishing 2011, 2011. 13.39 545.

- Liu, S. and Auckenthaler, P. (2014)** “*Optimal sensor placement for event detection and source identification in water distribution networks.*” *Journal of Water Supply: Research and Technology-AQUA*, 63(1), 51-57; DOI:10.2166/aqua.2013.106.
- Neupauer, M. R., Records, K. M., and Ashwood, H. M. (2010).** “*Backward probabilistic modeling to identify contaminant sources in water distribution systems.*” *J. Water Resour. Plann. Manage.*, 10.1061/(ASCE)WR.1943-5452.0000057, 587–591.
- Ostfeld, Avi, Uber, James G. and Salomons, Elad, Berry, Jonathan W., Hart, William E., Phillips, Cindy A., Watson, Jean-Paul, Dorini, Gianluca, Jonkergouw, Philip, Kapelan, Zoran, Pierro, Francesco di, Khu, Soon-Thiam, Savic, Dragan, Eliades, Demetrios, et al. (2008).** *The Battle of the Water Sensor Networks (BWSN): A Design Challenge for Engineers and Algorithms.* s.l.: *Journal of Water Resources Planning and Management-ASCE*, 2008. 10.1061/(ASCE)0733-9496(2008)134:6(556).
- Perelman, L., and Ostfeld, A. (2010).** “*Bayesian networks for estimating contaminant source and propagation in a water distribution system using cluster structure.*” *Proc. 12th Annual Water Distribution Systems Analysis Symp, ASCE.* 426-435.
- Piller, O., Deuerlein, J., Gilbert, D., and Weber, J. M. (2015).** “*Installing fixed sensors for double calibration and early-warning detection purposes.*” *Procedia Engineering*, 119, 564–572.
- Preis, A. and Ostfeld, A. (2006a).** *Multiobjective sensor design for water distribution systems security.* *Water Distribution Systems Analysis Symposium 2006, ASCE*, 17 pages, DOI: 10.1061/40941(247)107.
- Preis, A. and Ostfeld, Avi. (2006b).** *Optimal Sensors Layout for Contamination Source Identification in Water Distribution Systems.* *Water Distribution Systems Analysis Symposium 2006, ASCE*, 12 pages, DOI: 10.1061/40941(247)127.
- Propato, M. and Piller, O. (2006).** *Battle of the Water Sensor Networks.* s.l.: 8th annual Water Distribution System Analysis Symposium, 2006. August 27-30 2006, ASCE, 8 pages, DOI: 10.1061/40941(247)112.
- Propato, M., Sarrazy, F. and Tryby, M. (2010).** *Linear Algebra and Minimum Relative Entropy to Investigate Contamination Events in Drinking Water Systems.* s.l.: *Journal of Water Resources Planning and Management*, 2010. Volume 136, Issue 4 (July 2010) 10.1061/(ASCE)WR.1943-5452.0000059.
- Propato, M., Tryby, M. E., and Piller, O. (2007).** “*Linear algebra analysis for contaminant source identification in water distribution systems.*” *World Environmental and Water Resources Congress 2007, Tampa, Florida, USA, 15-19 May 2007*, 10 pages, ASCE.
- Rathi, S. and Gupta, R. (2014).** *Sensor placement methods for contamination detection in water distribution networks: a review.* s.l.: 16th Water Distribution System Analysis Conference, WDSA2014 — Urban Water Hydroinformatics and Strategic Planning, 2014. Volume 89, 2014, Pages 181–188.
- Seth, A., Klise, K., Siirola, J., Haxton, T., and Laird, C. (2016).** *Testing Contamination Source Identification Methods for Water Distribution Networks.* s.l.: *Journal of Water Resources Planning and Management*, 2016. 04016001.Klise, K., Siirola, J., Haxton, T., and Laird, C.
- SMaRT-Online^{W_{DN}} (2017). <http://www.smart-onlinewdn.eu/>, accessed: 2017-03-28.

Tryby, M. E., Propato, M. and Ranjithan, S. R. (2010). *Monitoring Design for Source Identification in Water Distribution Systems.* s.l. : Journal of Water Resources Planning and Management, 2010. Volume 136, Issue 6 (November 2010) 10.1061/(ASCE)WR.1943-5452.0000080.

Tables:

Table 1. Simplified input/output contamination matrix.

Parameters	NODE ID										
	3	5	7	9	17	19	21	23	31	33	35
sensor 7	0.98	0.69	1.00	1.00	0.56	0.64	0.64	0.64	0.00	0.00	0.00
sensor 35	0.98	0.30	0.00	0.00	0.98	0.89	0.33	0.00	1.00	1.00	1.00
mean	0.98	0.49	0.50	0.50	0.77	0.76	0.48	0.32	0.50	0.50	0.50
rank	1	9	8	8	2	3	10	11	8	8	8

Table 2. Contribution for a source identification with source of contamination being at node 3.

node Id	3	5	7	9	17	19	21	23	31	33	35
<i>Contribution</i>	1.00	0.58	0.63	0.63	0.95	0.89	0.53	0.47	0.63	0.63	0.63

Table 3. Average Contribution calculations for 20 scenarios.

node Id		47	35	49	33	45	31	23	19	7	21
scenario	contamination node	<i>Contribution</i>									
1	19	0.53	0.53	0.42	0.58	0	0	0.63	0.74	0.63	0.68
2	23	0	0	0	0	0	0	0.74	0	0	0
3	51	0	0	0	0	0	0	0	0	0	0
4	47	0.95	0	0.83	0	0	0	0	0	0	0
5	23	0	0	0	0	0	0	0.63	0	0	0
6	3	0	0	0	0	0.63	0.74	0	0.74	0.58	0.68
7	19	0.84	0.79	0.67	0.84	0	0	0.89	0.95	0.74	0.89
8	9	0	0	0	0	0	0	0	0	0.58	0
...
18	51	0	0	0	0	0	0	0	0	0	0
19	31	0.53	0.53	0.42	0.58	0.63	0.73	0	0	0	0
20	31	0.53	0.53	0.42	0.58	0.63	0.73	0	0	0	0
	mean	0.28	0.27	0.27	0.25	0.23	0.22	0.22	0.20	0.19	0.19

Table 4. Average time to detection and detection likelihood for ATDOP and DLOP.

criteria	ATDOP	DLOP
average time to detection	3.8h	6h
detection likelihood	75%	82%

Table 5. Average time to detection and detection likelihood for ATDOP 10/20/50, DLOP 10/20/50

criteria	ATDOP 10	ATDOP 20	ATDOP 50	DLOP 10	DLOP 20	DLOP 50
average time to detection	3.8h	2.7h	1.8h	6h	4.4h	3.4h
detection likelihood	74%	80%	84%	82%	91%	95%

Table 6. Average time to detection and detection likelihood for ATDOP, DLOP, DOP4 and GOP24.

criteria	ATDOP	DLOP	GOP4	GOP24
average time to detection	3.8h	6h	3.5h	5.1h
detection likelihood	74%	82%	76%	80%

Table 7. Average time to detection and detection likelihood for ATDOP, DLOP, LSATDOP4, LSATDOP24, LSDLOP4 and LSDLOP24.

criteria	ATDOP	DLOP	LSATDOP4	LSATDOP24	LSDLOP4	LSDLOP24
average time to detection	3.8h	6h	3.9h	5.4h	4.3h	5.2h
detection likelihood	74%	82%	77%	79%	80%	81%

Table 8. Evaluation for all parameters.

algorithm cases	average time to detection rank	detection likelihood rank	accuracy rank	specificity rank	contribution rank	mean rank	range
GOP24	5	3	3	2	2	3	3-5
GOP4	1	7	1	4	6	3.8	1-7
LSATDOP24	7	5	2	3	5	4.4	2-7
LSATDOP4	3	6	5	1	7	4.4	1-7
LSDLOP24	6	2	7	6	1	4.4	1-7
LSDLOP4	4	4	4	8	4	4.8	4-8
DLOP	8	1	8	7	3	5.4	1-8
ATDOP	2	8	6	5	8	5.8	2-8

Figures:

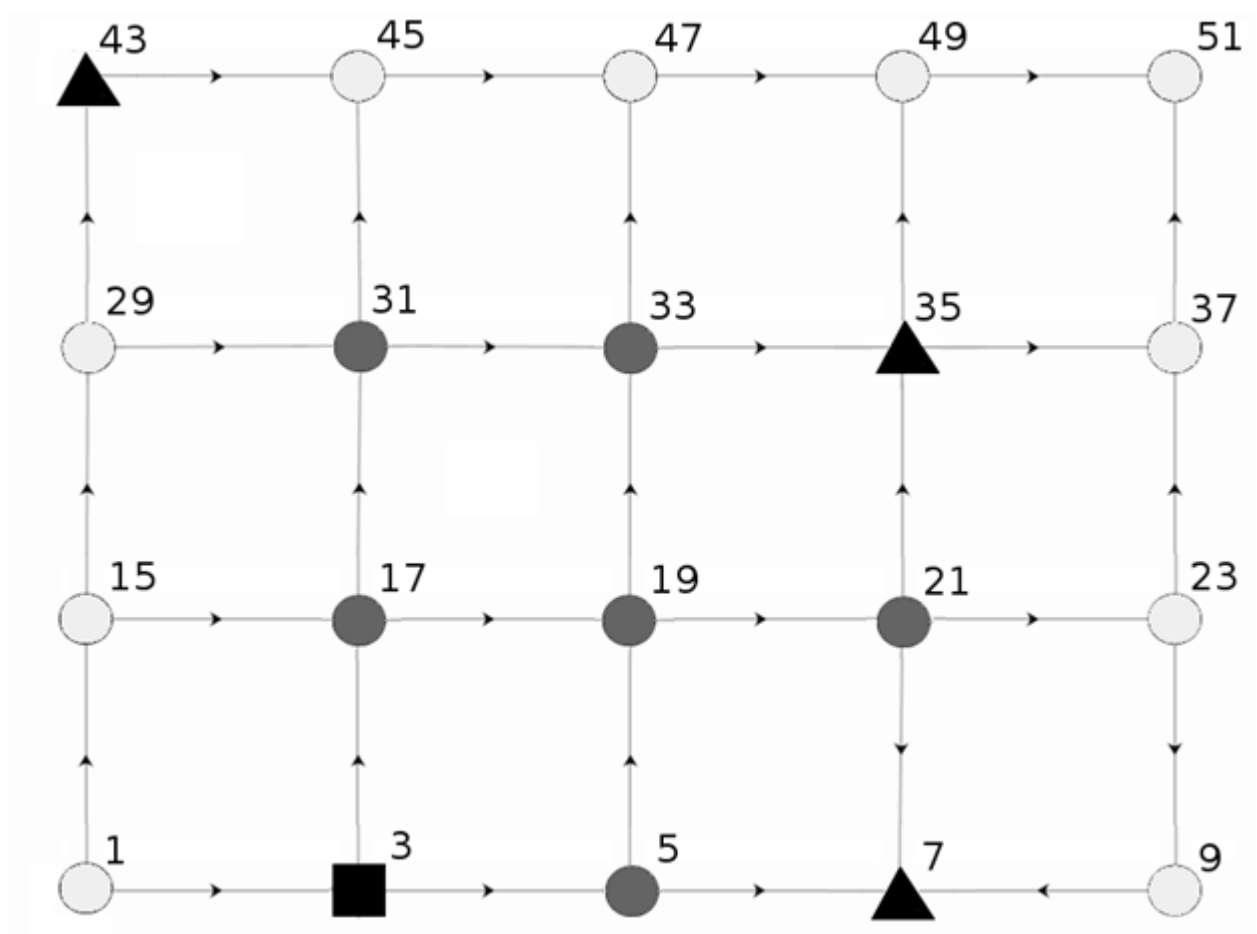


Fig. 1. Small network; solid square: contamination node; triangle: sensor; dark-gray circle: potential contamination node; light gray: ordinary node (adapted from Neupauer et al. 2010).

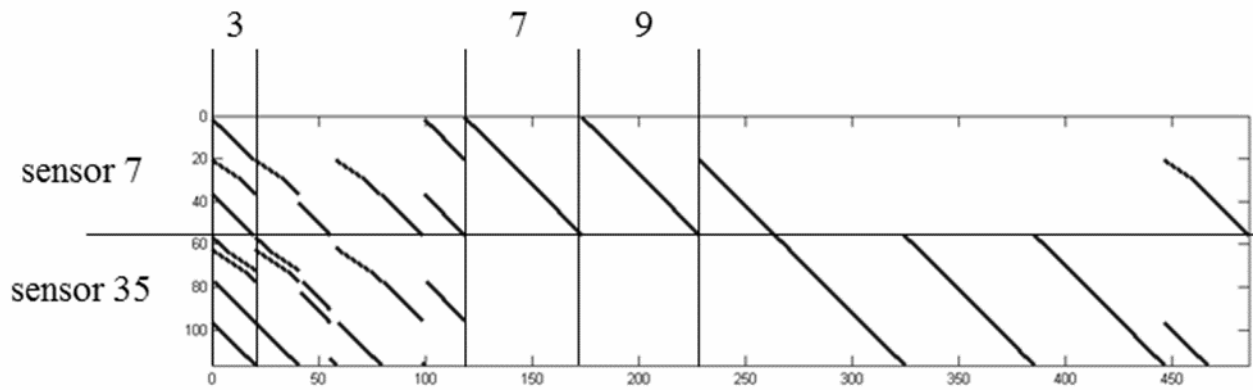


Fig. 2. Input/output matrix of contamination; solid points are non-zero values.

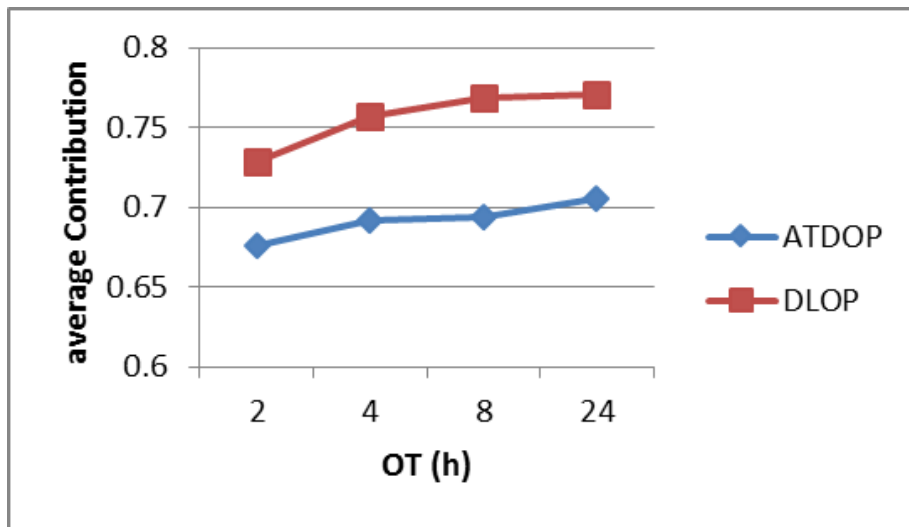


Fig. 3. Contribution for average time to detection and detection likelihood optimal sensor placement with 10 sensors.

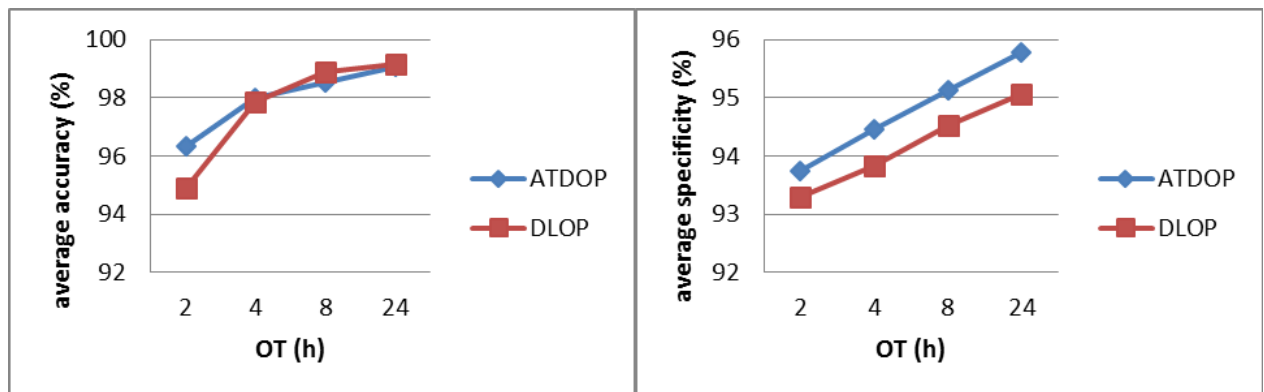


Fig. 4. Accuracy and specificity for average time to detection and detection likelihood optimal sensor placement with 10 sensors.

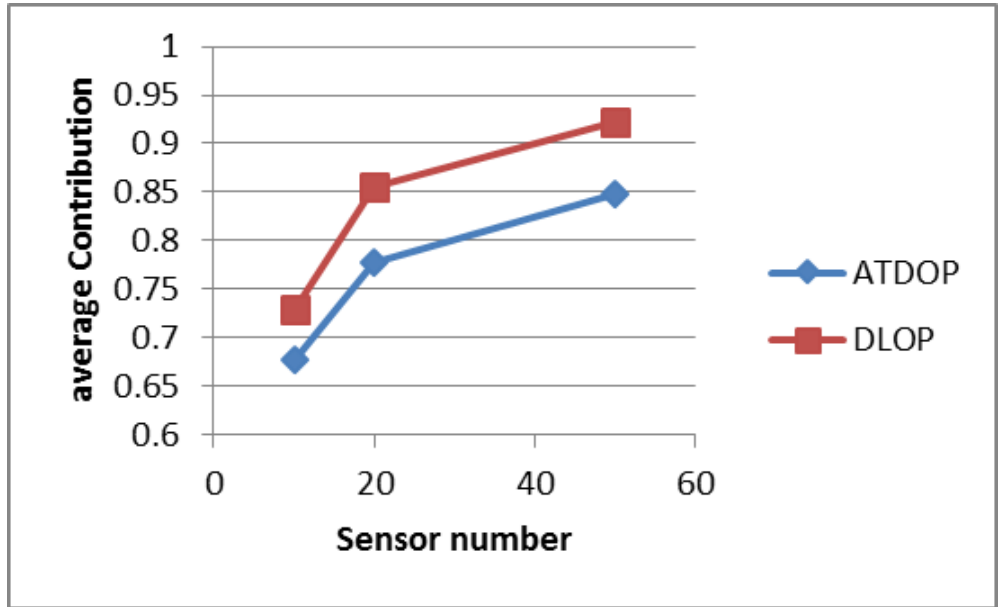


Fig. 5. Contribution for average time to detection and detection likelihood optimal sensor placement with 10, 20, and 50 sensors.

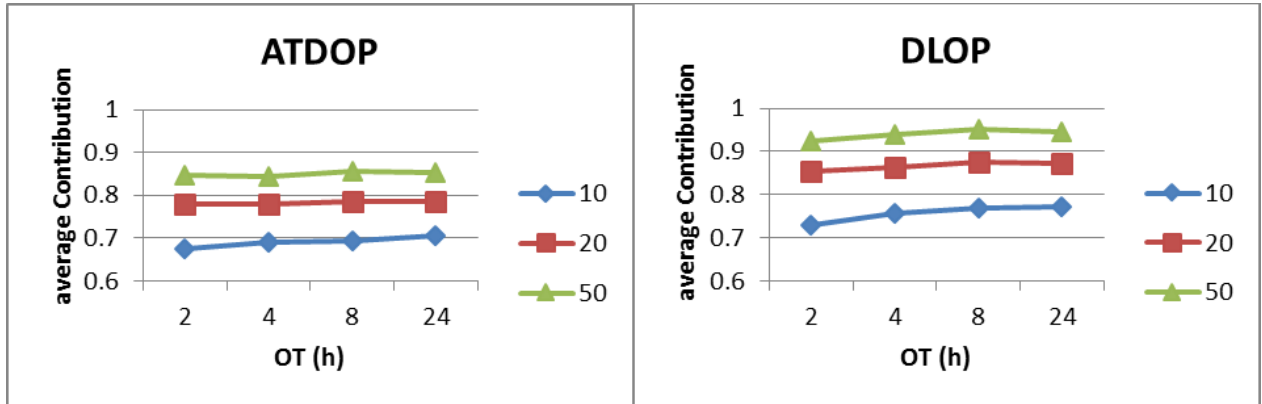


Fig. 6. Contribution for average time to detection and detection likelihood optimal sensor placement with 10, 20, and 50 sensors as a function of observation time.

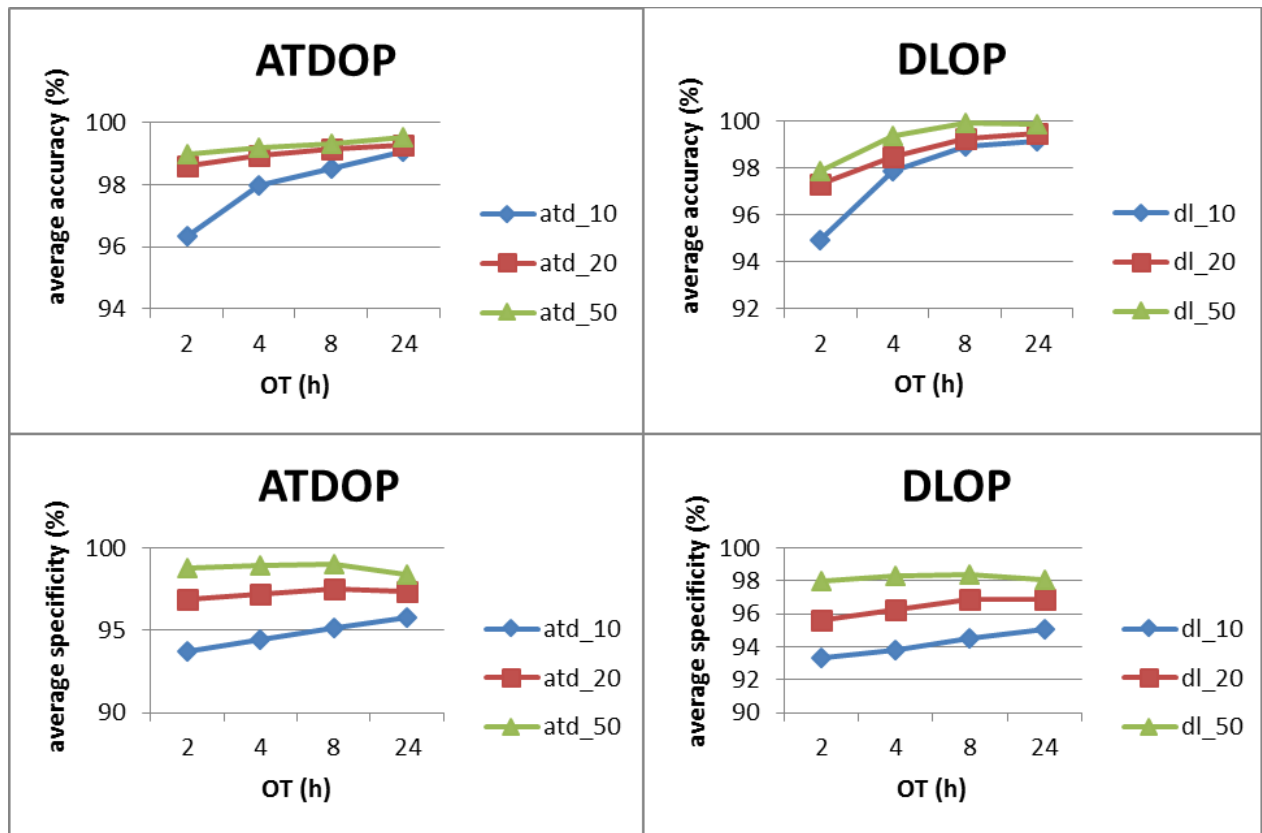


Fig. 7. Accuracy and specificity for average time to detection and detection likelihood optimal sensor placement with 10, 20, and 50 sensors.

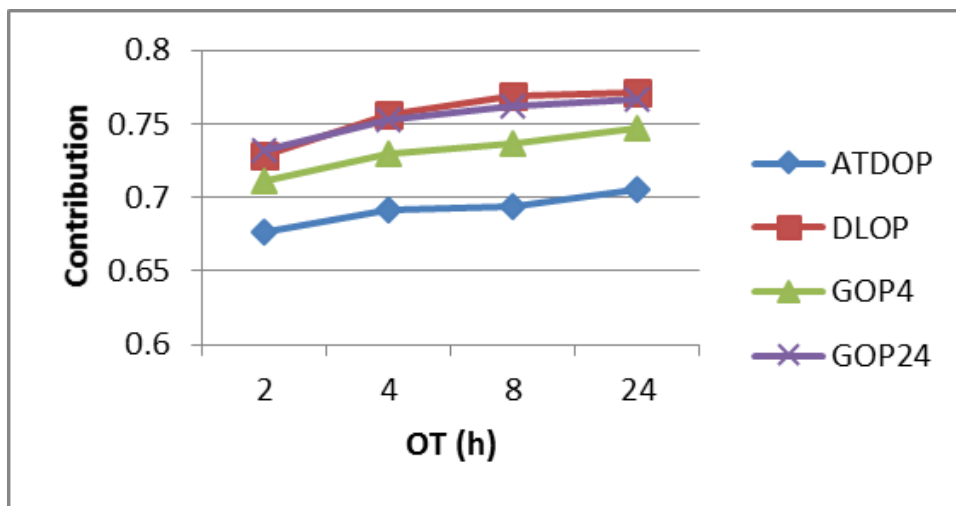


Fig. 8. Contribution for average time to detection, detection likelihood, and greedy algorithm optimal sensor placement with 10 sensors.

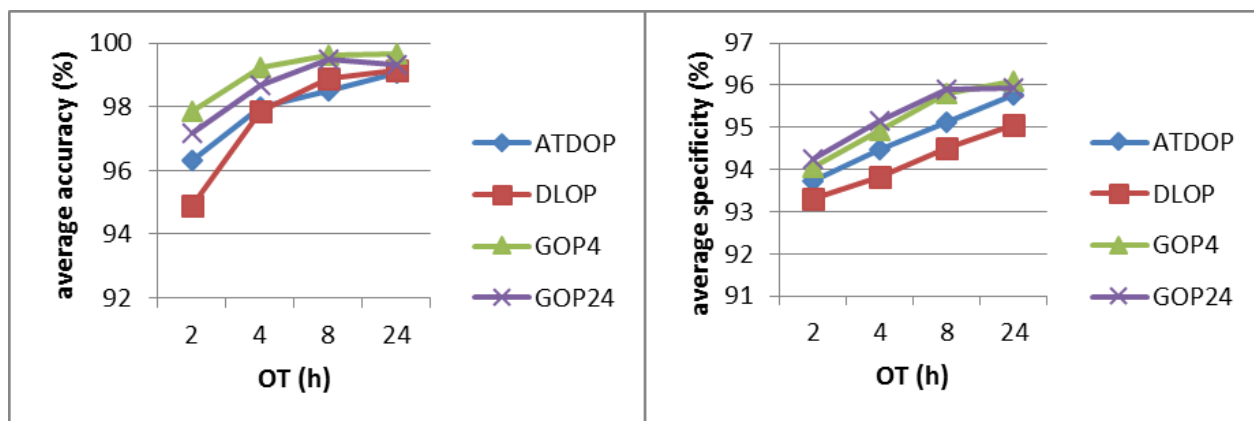


Fig. 9. Accuracy and specificity for average time to detection, detection likelihood, and greedy algorithm optimal sensor placement with 10 sensors.

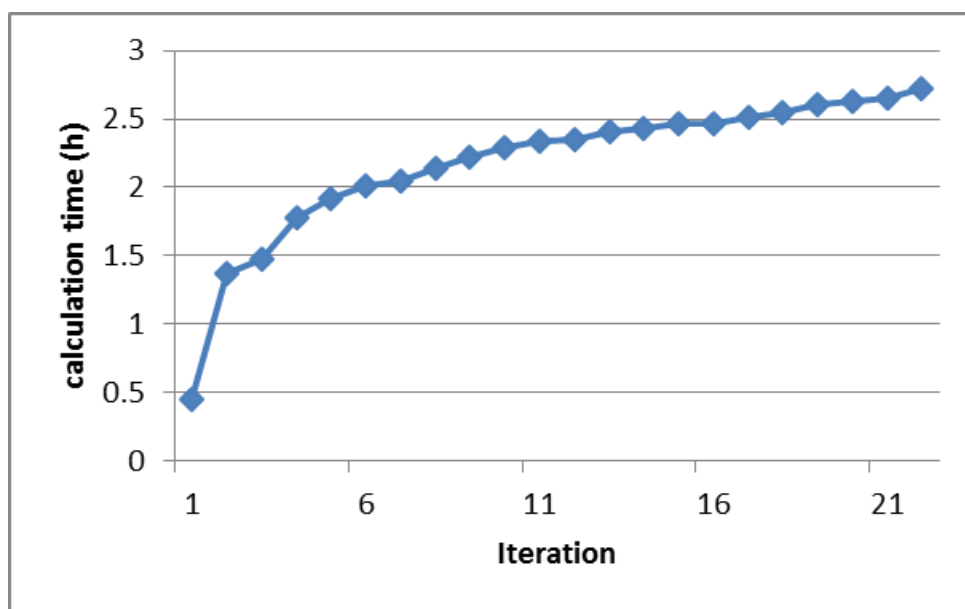


Fig. 10. Execution time per iteration for the greedy algorithm with BT = 24 h and OT = 2 h with 120 processors for 23 sensors.

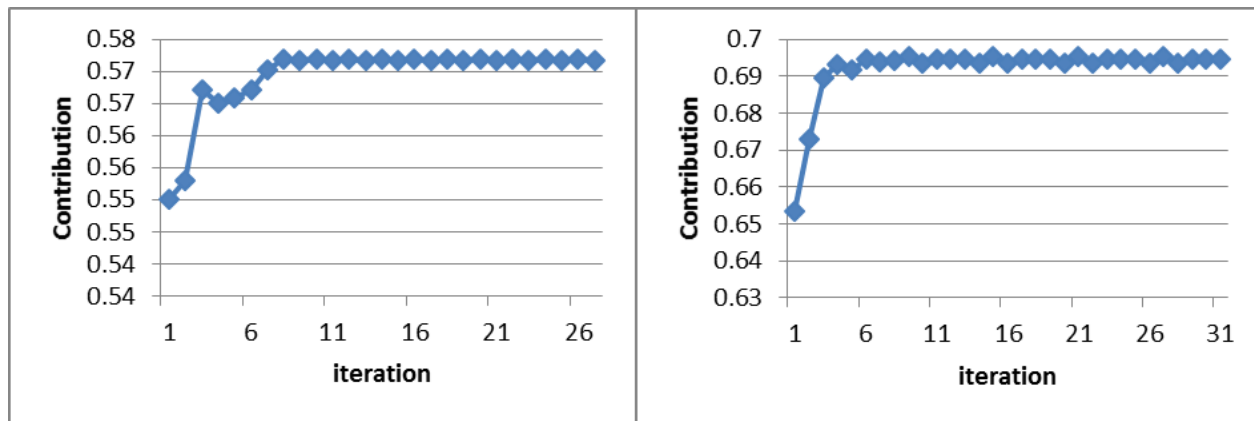


Fig. 11. Contribution evolution as a function of iteration for local search with ATDOP initialization and BT = 4 or 24 h and OT = 2 h for 10 sensors.

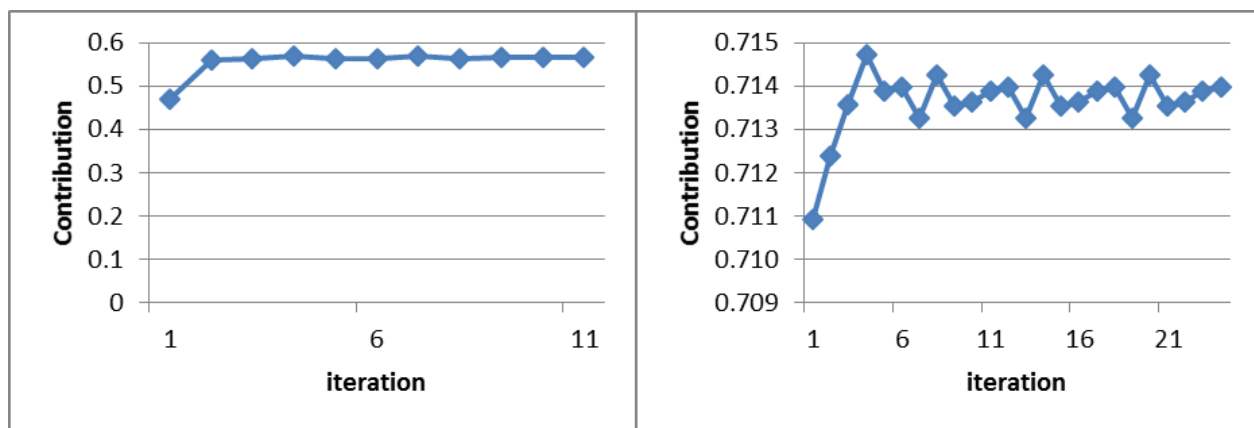


Fig. 12. Contribution evolution as a function of iteration for local search with DLOP initialization and BT 1= 4 or 24 h and OT 1= 2 h for 10 sensors.

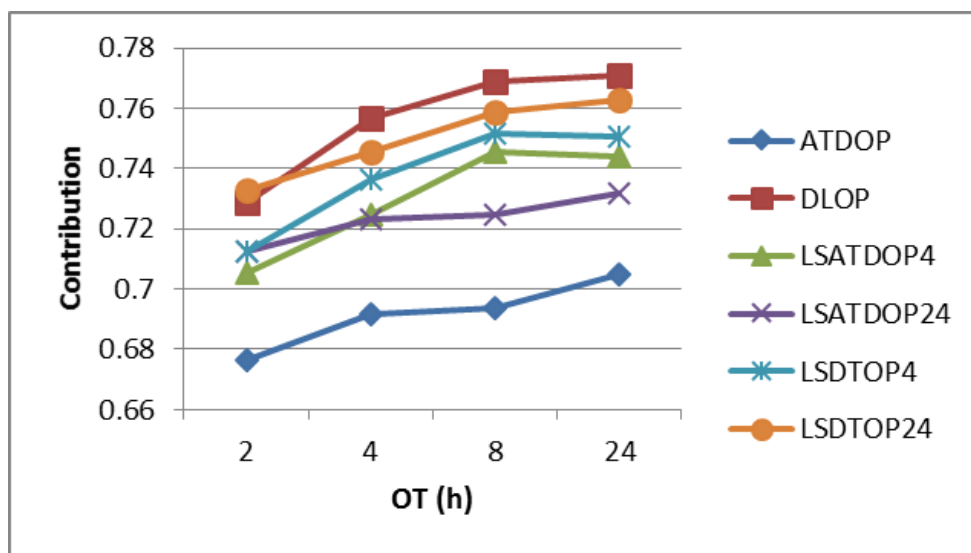


Fig. 13. Contribution for average time to detection, detection likelihood, and local search algorithm optimal sensor placement with 10 sensors.

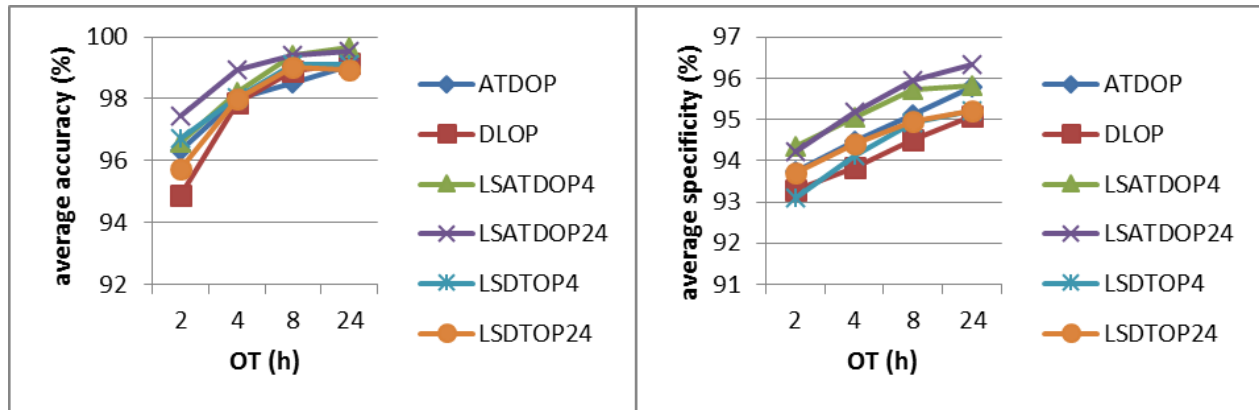


Fig. 14. Accuracy and specificity for average time to detection, detection likelihood, and local search algorithm optimal sensor placement with 10 sensors.