Automated feature recognition in CFPD analyses of DMA or supply area flow data

Peter van Thienen and Ina Vertommen

ABSTRACT

The recently introduced comparison of flow pattern distributions (CFPD) method for the identification, quantification and interpretation of anomalies in district metered areas (DMAs) or supply area flow time series relies, for practical applications, on visual identification and interpretation of features in CFPD block diagrams. This paper presents an algorithm for automated feature recognition in CFPD analyses of DMA or supply area flow data, called CuBOid, which is useful for objective selection and analysis of features and automated (pre-)screening of data. As such, it can contribute to rapid identification of new leakages, unregistered changes in valve status or network configuration, etc., in DMAs and supply areas. The method is tested on synthetic and real flow data. The obtained results show that the method performs well in synthetic tests and allows an objective identification of most anomalies in flow patterns in a real life dataset.

Key words | demand patterns, flow pattern analysis, leakage, numerical methods

Peter van Thienen (corresponding author)
Ina Vertommen
KWR Watercycle Research Institute,
Post Box 1072,
3430 BB Nieuwegein,
The Netherlands
E-mail: *peter.vanthienen@kwrwater.nl*

INTRODUCTION

In recent years, utilities have been moving towards more data based decision making for network operation and management. Flow rate time series for district metered areas (DMAs) and distribution areas provide meaningful insights into the flow performance of the network, but due to their complexity these are not always fully explored and used. These data contain information about leakage (which continues to be an issue, with numbers worldwide ranging from 3% to more than 50% (Lambert 2002; Beuken *et al.* 2006)), unauthorized consumption, customer behavior, network configuration and isolation (valve statuses), among others. Many methods exist to obtain information out of these data, and most focus on leakage. Classically, the most important are top-down and bottom-up methods (Farley & Trow 2003; Wu 2011). The top-down method consists of a water balance in which the registered amount of water delivered to a supply area over the period of a year is compared to the billed amount of water. The bottom-up method essentially compares the minimum flow rate during the quiet night hours into a DMA or demand zone, or the integrated flow of a 24-hour period, to an estimate for the demand for this DMA or demand zone based on the number of connections (Puust *et al.* 2010).

Different methods to determine the amount of non-revenue water, leakage, bursts and the location of leakages have been the focus of research. These methods include inverse transient analysis (Liggett & Chen 1994; Savic *et al.* 2005; Vítkovský *et al.* 2007), alternative statistical (e.g. Palau *et al.* 2012; Romano *et al.* 2013) and machine learning methods (e.g. Aksela *et al.* 2009; Mounce *et al.* 2010; Mamo *et al.* 2014), or a combination of both (e.g. Romano *et al.* 2014), probabilistic leak detection (Poulakis *et al.* 2003; Puust *et al.* 2006), pressure dependent leak detection (Wu *et al.* 2010), and meta-methods including a comparison of results for neighboring DMAs (Montiel & Nguyen 2011, 2013).

The comparison of flow pattern distributions (CFPD) method was introduced (Van Thienen 2013; Van Thienen *et al.* 2013a) as a new tool to assess flow data of a DMA or supply area in order to pinpoint (in time, not space), identify,

doi: 10.2166/hydro.2015.056

and quantify changes in the amount of water supplied (see Figure 1). It has since been successfully applied in multiple projects with Dutch drinking water companies to identify, for example, leakages and incorrect valves statuses and network connections (Van Thienen *et al.* 2013b).

The interpretation of changes in water flow time series through CFPD block diagrams is intuitive in all but the most complex cases. However, it relies on the visual interpretation of these diagrams, which is still a limitation. This paper is aimed at overcoming this limitation by presenting a support algorithm for automated feature recognition in CFPD block diagrams. Such an algorithm offers several advantages: automated pre-screening of data to limit manual inspection and interpretation to the most interesting cases; objective rather than (to some degree) subjective selection and analysis of features.

This paper presents a method for automated feature recognition in CFPD block diagrams, called the CuBOid (CFPD Block Optimization) algorithm. Its principle is presented, and the method is applied to synthetic and real network data to evaluate its performance.

METHODS

For a complete description of the CFPD method, the reader is referred to Van Thienen (2013). A concise introduction is provided in Appendix 1 (available with the online version of this paper). An overview of the analysis and interpretation of the method is presented in Figure 1. In the matrices resulting from the analysis, each event (change in the flow pattern) is characterized by a typical structure (Figure 2). These matrices should be read as follows: going from the left to the right (i.e. time arrow), a change in color or color intensity represents a change of the CFPD parameter. The CFPD parameter changes as a consequence of a flow pattern alteration: the color intensity is proportional to magnitude of the alteration. The presence of this change in

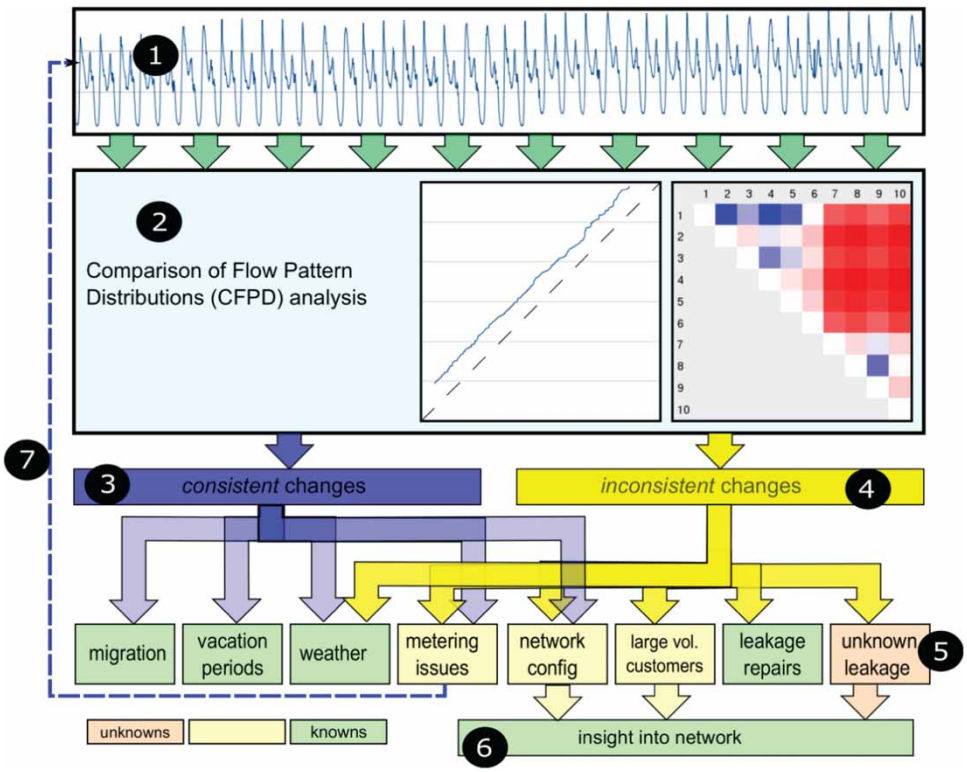

Figure 1 | CFPD analysis procedure and interpretation. (1) Flow time series; (2) CFPD analysis; (3), (4) identification of consistent and inconsistent changes; (5) interpretation of these changes in terms of known and unknown mechanisms; (6) discarding changes by known mechanisms such as vacation periods, weather, among others, results in a reduced list of unknown events that can be responsible for the change, making the interpretation easier; (7) any data quality issues which are found may initiate improvement of measurements. Copied from Van Thienen *et al.* (2013b).

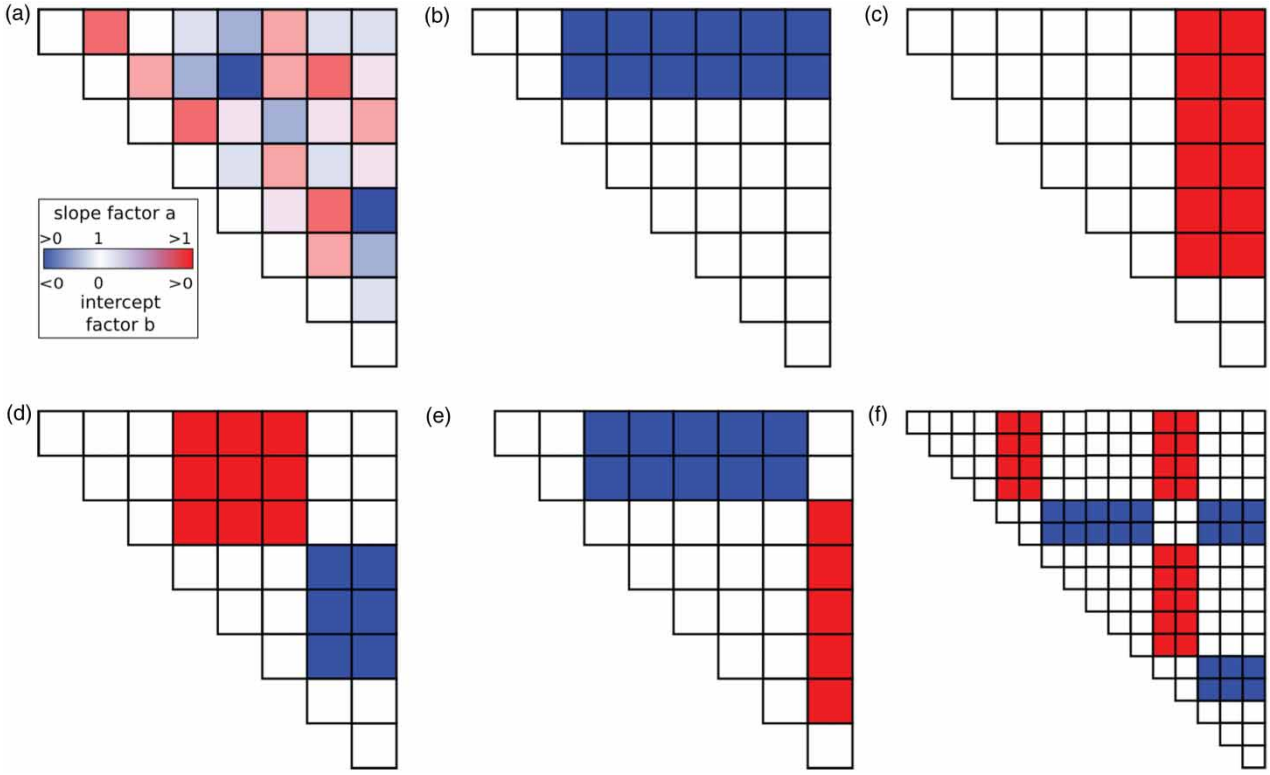

Figure 2 | CFPD block diagrams are upper triangle matrices with matrix values indicated by color (a). The diagonal is always 1 (slope factor *a*) or 0 (intercept factor *b*). Slope factor *a* values may range from 0 towards infinity; intercept factors *b* may range from minus infinity towards infinity. Logically permissible block patterns either meet both the upper and right hand side edge (b), (c) or appear in opposite sign combinations of two blocks, one of which touches the upper edge and the other the right hand side edge (d), (e). Weekdays and weekends are generally somewhat different, and show up as distinctive regular banded patterns (f). Please refer to the online version of this paper to see this figure in color: http://dx.doi.org/10.2166/hydro.2015.056.

multiple rows of the matrix means that the anomaly is not caused by an anomaly in the reference signal. Changes in flow patterns which are most interesting are systematic changes, which are indicative of changes in demand, network configuration, leakage, etc., rather than stochastic variations of short duration. These systematic changes show up in CFPD block diagrams as blocks with a similar color intensity (see Figure 2(b)–2(e)).

The CuBOid feature recognition algorithm presented in this paper seeks to describe these typical patterns observed in a CFPD block diagram as a summation of permissible block functions. In this way, it is somewhat similar to, for example, the Discrete Cosine Transform (Ahmed *et al.* 1974) or the Discrete Wavelet Transform (e.g. Akansu *et al.* 2009) which are used, for example, in image compression methods. The typical shape of the permissible blocks representing anomalies in the flow pattern stems from the nature of the CFPD analysis procedure (see Figure 2). This typical shape can be described by the following function:

$$f(i,j) = \begin{cases} 1 & \text{for } 1 \le i \le j_1, \quad j_1 \le j \le j_2 \\ -1 & \text{for } j_1 \le i \le j_2, \quad j > j_2 \\ 0 & \text{for all other } i, \quad j \end{cases} \tag{1}$$

In this expression, i and j are the row and column number, respectively, and j_1 and j_2 are the first and last column of a perturbation block. Note that consecutive columns in a CFPD block diagram correspond to consecutive days (or weeks, or some other duration), which are compared to each other in consecutive rows (for more details see Van Thienen (2013)).

Since people behave differently in the weekend compared to weekdays, CFPD block diagrams generally also show a distinct pattern setting apart weekdays from weekends and vice versa. An expression similar to the one above can be used to describe weekend day anomalies in CFPD block diagrams:

$$f(i,j) = \begin{cases} w_1 & \text{for } i \in [6,\ 7], \quad j \in [1,\ 5] \quad \text{Weekends compared to weekdays} \\ w_2 & \text{for } i \in [1,\ 5], \quad j \in [6,\ 7] \quad \text{Weekdays compared to weekends} \\ w_1 + w_2 & \text{for } i \in [6,\ 7], \quad j \in [6,\ 7] \quad \text{Weekends compared to weekends} \\ 0 & \text{for all other } i, \quad j \quad \text{Weekdays compared to weekdays} \end{cases} \tag{2}$$

In this expression, w_1 and w_2 are two weight factors.

The approach to automatically identify the block functions representing anomalies, whilst ignoring the regular weekday-weekend pattern, is by means of an optimization algorithm, in which only a limited number of block configurations, which can logically be present in a CFPD block diagram (shown in Figure 2), combined with the typical weekend pattern, are considered. The process has six steps, which are as follows (for a block (slopes or intercepts) of dimensions mxm).

1. Break detection: in order to quantify changes (breaks) between neighboring columns, firstly the L_x norm (with x having a value typically around 1.0) is computed for the difference vector of each pair of consecutive column vectors (skipping the first column of the matrix and the last item of each second vector). The values of these norms are divided by the size of the column vectors – m, obtaining a measure for the step sizes between consecutive columns, representing analysis periods. The exponent of the norm determines the focus of sensitivity of break detection within the matrix: smaller values result in more breaks being detected on the left side of the matrix, while larger values result in more breaks being detected on the right side of the matrix.

2. Generation of permissible blocks: using the n biggest changes (with n decided by the user), all permissible block functions are generated for $n \in \mathbb{N},\ n < m$. These permissible block functions correspond to all possible combinations of two steps (starting and ending) which are taken from the n biggest changes. The number of functions generated is $k = n(n-1)$.

3. Combination of block functions: the user choses the number of block functions p, with $p \in \mathbb{N},\ p < k$, which is used to resolve a single CFPD block diagram. All possible combinations of p block functions from the k functions generated in step 2 are generated. Thus, in total there are $k!/(k-p)!$ combinations.

4. For each combination, an optimization is performed in which the function amplitudes are the decision variables and the objective is to minimize the difference between the summation of this function combination and the block matrix which is being fitted. The weekday-weekend pattern is included in this computation, so the parameters w_1 and w_2 are free parameters of the optimization problem as well. This is described by the following objective function:

$$C_i = \left(\frac{\sum_{j=1}^{m}\sum_{k=1}^{m}\left(\left(\sum_{l=1}^{p} W_{il}B_{iljk} + \sum_{q=1}^{2} w_q b_{qjk}\right) - M_{jk}\right)^2}{m}\right)^{\frac{1}{2}} \tag{3}$$

for combination i, with j and k the indices for the matrix rows and columns, W_{il} the weights or amplitudes for block function l in combination i, B_{iljk} the amplitude of block function l of combination i at matrix row j and column k (expression (1)), w_q the weekend day factor for weekend day q, b_{ij} the amplitude of the weekend block function for day q at matrix row j and column k, and M_{jk} the actual CFPD matrix value at at matrix row j and column k. No constraints were applied to the optimization.

This optimization can be done in parallel. This step results in block amplitudes W_{il} for each combination generated in step 3. Note that the amplitude is dimensionless for the matrix of slope factors a and has the same unit of volumetric flow rate as the original input time series for the matrix of intercept factors b.

5. The performance of each combination (blocks and amplitudes) is quantified using the following expression:

$$F_i = C_i^*\left(1 + f_1{}^* n_p + f_2{}^* n_o/s\right) \tag{4}$$

in which F is the fitness of the solution, C_i is the Euclidian 2-norm of the difference between the original matrix and the reconstructed matrix (Equation (3)), f_1 is the penalty factor for the number of block functions, n_p is the number of block functions used ($\leq p$), f_2 is the overlap penalty factor, n_o is the number of overlapping blocks in the set of block functions ($\sum_{\text{matrix columns}}$ (number of block functions in column - 1)), and s is the sum of the lengths of the

block functions. This cost function reflects the fit of the candidate blocks with respect to the actual CFPD matrix. It is designed to penalize both a large number of block functions and a large degree of overlap. The fitness parameter F is minimized.

6. The best performing combination of block functions and weekend parameters is selected.

Thus, the process combines a combinatorial problem with a parameter fitting problem. The former is addressed in steps 2, 3, 5, 6, the latter in step 4.

Note that the method can be asked to fit a large number of functions simultaneously, but this will most certainly lead to overfitting the data, with noise being described by additional block functions. Therefore, parsimony is important to obtain meaningful results. This will be illustrated later. The penalty parameters become relevant for larger values of n and p. Choosing a larger value of n and/or p results in a significant increase in computation time, which is the reason why these parameters were introduced. With unlimited computation power, n should be the number of columns – 2, and p should be chosen to represent the largest number of anomalies expected in a single matrix.

The optimization criterion is formulated in terms of a Euclidean norm of the difference vector of the diagram data and sum of all block and weekday/weekend functions for which the optimization is being performed. For slope diagrams, the log of the actual values is taken, since multiple anomalies simply add up in log space and values are zero centered. The optimization method used for the parameter fitting part of the algorithm is the quasi-Newton method of Broyden, Fletcher, Goldfarb, and Shanno, as implemented in *scipy* (Oliphant 2007).

RESULTS AND DISCUSSION

In order to test the performance of the proposed approach and the influence of the different parameters on the results, a series of tests was performed on synthetic data. Bearing in mind that several combinations of parameter values are possible, note that only a limited set of tests that focus on the influence of each parameter one at a time (and hence facilitate interpretation) have been carried out and reported in this paper. Table 1 summarizes the considered tests and corresponding parameter values. In addition to this, the results of a series of tests on real flow data are also reported.

Table 1 | Summary of the performed tests. Distinguishing parameter values are indicated by a grey background color

Test description	Test code	Number of clusters	Number of steps (n)	L_x norm	f_1	f_2
Default	1	3	5	1	0.33	0.33
Influence of the number of clusters	2	2	5	1	0.33	0.33
	3	4	5	1	0.33	0.33
Influence of the number of steps	6	3	4	1	0.33	0.33
	7	3	6	1	0.33	0.33
Influence of the norm	8	3	5	0.7	0.33	0.33
	9	3	5	1.25	0.33	0.33
	10	3	5	2	0.33	0.33
Influence of the f_1 penalty	11	3	5	1	0.01	0.33
	12	3	5	1	0.2	0.33
	13	3	5	1	0.4	0.33
	14	3	5	1	0.7	0.33
	15	3	5	1	0.9	0.33
Influence of the f_2 penalty	16	3	5	1	0.33	0.01
	17	3	5	1	0.33	0.2
	18	3	5	1	0.33	0.4
	19	3	5	1	0.33	0.7
	20	3	5	1	0.33	0.9

Synthetic data

The synthetic data considered for the tests consist of repetitions of actual measured flow patterns, with in total three sequences of five identical weekdays and two identical weekend days, starting at day 2 and ending at day 22 of an arbitrary month in an arbitrary year. Different datasets were generated from these original data, by adding anomalies with different amplitude and duration, as well as different levels of normally distributed noise. The characteristics of the generated datasets are summarized in Table 2, and the flow patterns corresponding to the unperturbed signal and datasets 1a, 2a and 3a can be seen in Figure 3.

The difference between week and weekend days is clearly visible in the flow patterns. The added anomalies are also visible, corresponding to upward shifts in the patterns.

Table 2 | Summary of the generated datasets used to test the CuBOid algorithm

	Anomaly 1		Anomaly 2		
Dataset ID	Start-end day	Amplitude (m^3/h)	Start-end day	Amplitude (m^3/h)	(%) Gaussian noise
0	None	–	None	–	None
1a	05–08	10	15–18	5	None
1b	05–08	10	15–18	5	5
1c	05–08	10	15–18	5	10
1d	05–08	10	15–18	5	20
2a	04–10	10	14–20	5	None
2b	04–10	10	14–20	5	5
2c	04–10	10	14–20	5	10
2d	04–10	10	14–20	5	20
3a	05–15	10	12–20	5	None
3b	05–15	10	12–20	5	5
3c	05–15	10	12–20	5	10
3d	05–15	10	12–20	5	20

Block functions

The CuBOid algorithm identifies block functions representing anomalies in flow patterns of some days with respect to earlier days (or any other time scale – weeks, months, …) in a certain period of time. Since for the different datasets the anomalies were manually added to the data, it is known beforehand what the block functions should look like. The block functions are described by a start and an end column in the matrix diagram, and by an amplitude. The start and end columns should correspond to the start and end dates of the anomalies, and the amplitude should be equal to the amplitude of the actual anomaly (recall Table 2).

For datasets 1a to 1d, two block functions should be identified. The start and end days of the block function describing the first anomaly should be 5 and 8, and the amplitude should be 10 m^3/h. For the second anomaly the start and end columns of the block function should be 15 and 18, and the amplitude should be 5 m^3/h. Table 3 summarizes the start and end days as well as the amplitudes of the block functions obtained by the different tests. The last two columns in the table display the actual used steps to form the block functions and the C_i norm (Equation (3)), i.e. the difference (or distance) between the difference between the original matrix and the reconstructed matrix. This norm can also be used to compare results obtained by different tests in a more straightforward way.

First of all, the influence of added noise on the estimated amplitude of the block functions is clearly visible: when adding more noise to the data, the estimated amplitude decreases, and the estimated end day of the second block also tends to get worse, being shifted forward (this is, ending later than it should). Accordingly, the C_i norm increases with the increase of random Gaussian noise.

When no noise is added to the data (dataset 1a), all performed tests lead to the same resulting block functions, and these are a very close fit to the actual introduced anomalies. The slight deviation from the actual values is presumably due to numerical issues and/or the stop criterion for the optimization algorithm.

When adding 5% random Gaussian noise, not all tests lead to the same block functions. While most tests perform well in identifying the two anomalies, test 11 (lowest f_1 penalty coefficient), leads to the identification of three blocks instead. The third block is probably fitting the noise added to the data.

When adding 10% of noise, the tests perform generally worse, overestimating the duration of the second anomaly (by identifying the end column as being 20 instead of 18), and underestimating the amplitudes of the anomalies. The best results are obtained for test 7 and 8 (with higher number of steps and a lower L_x norm, respectively) Figure 4(a) and 4(b) illustrate the obtained results when performing test 7. In Figure 4(a), the matrix of *b*-factors is visible. In Figure 4(b), the estimated block functions are visible. The visual interpretation of Figure 4(b) is clearer, since the added noise is not visually represented.

For 20% of random noise, all tests underestimate the amplitudes of both anomalies, and overestimate the duration of the second anomaly. The best performance, in terms of the C_i norm, is obtained for test 11 (with the lowest f_1 penalty). However, for this case, the algorithm identifies three block functions, i.e. a false positive which is fitting the noise. For test 15 (highest f_1 penalty coefficient) the algorithm does not identify block functions, i.e. the results are false negatives. Figure 5(a) and 5(b) illustrate the obtained results when performing test 1. In Figure 5(a), the matrix of *b*-factors is visible.

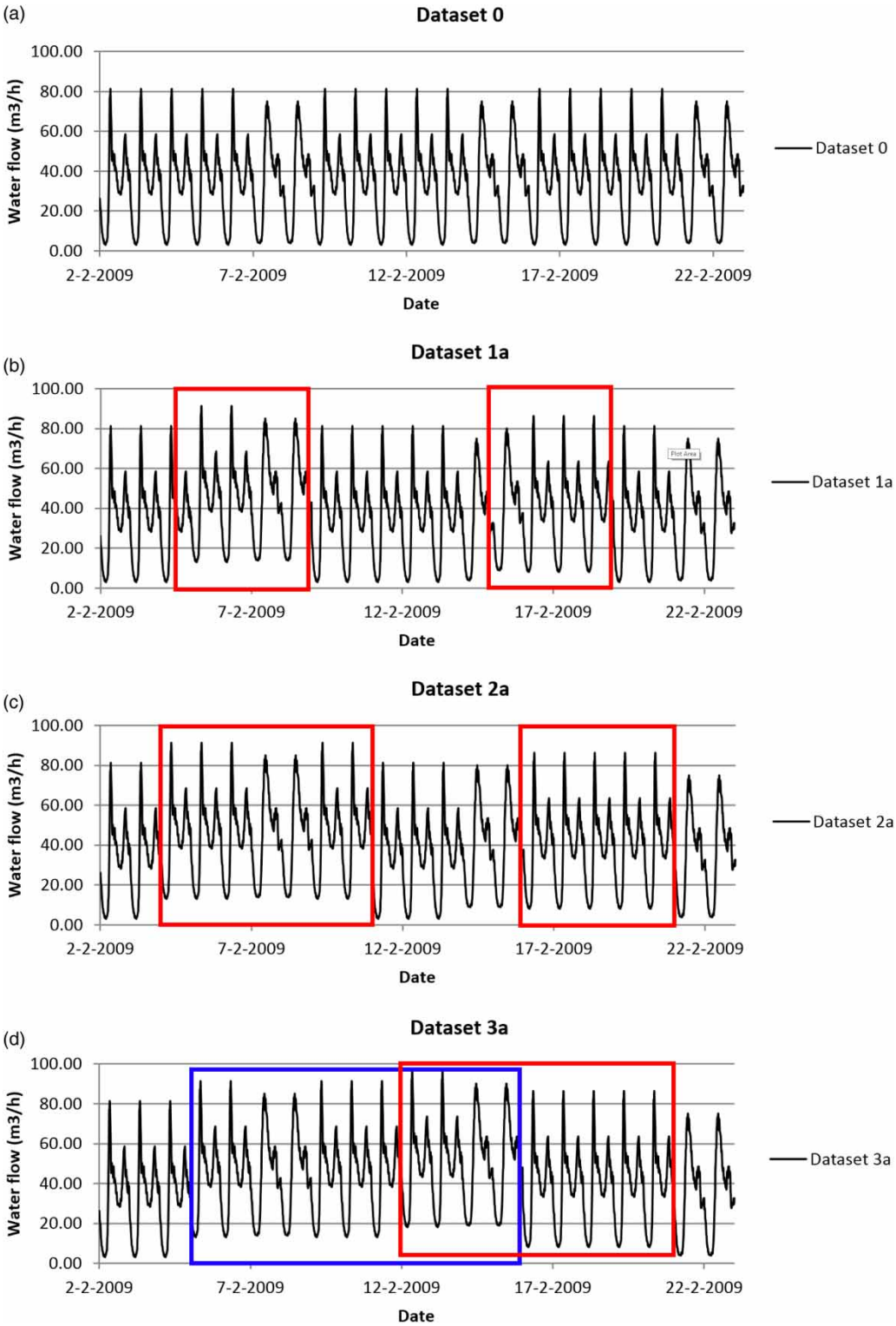

Figure 3 | Considered flow patterns: (a) original flow pattern, with distinct week and weekend days; (b) flow pattern corresponding to dataset 1a; (c) flow pattern corresponding to dataset 2a; (d) flow pattern corresponding to dataset 3a.

Table 3 | Characteristics of the block functions obtained by the performed tests on datasets 1a–1d, corresponding used steps and C_i norm

		Block function 1			Block function 2			Block function 3				
Dataset	**Test**	**Start**	**End**	**Amplitude (m^3/h)**	**Start**	**End**	**Amplitude (m^3/h)**	**Start**	**End**	**Amplitude (m^3/h)**	**C_i**	**Number of steps used**
1a	All	5	8	10.05	15	18	5.04	.	.	.	3.5	4
1b	All except	5	8	9.84	15	18	4.48	.	.	.	4.3	4
	11	5	8	9.75	15	18	4.39	9	14	–0.21	4.6	4
1c	All except	5	8	8.96	15	20	3.75	.	.	.	22.0	4
	7, 8	5	8	9.03	15	18	5.04	.	.	.	7.6	4
1d	All except	5	8	8.18	15	20	3.68	.	.	.	19.2	4
	6, 10	5	8	8.51	15	22	3.32	.	.	.	24.1	4
	11	5	8	8.82	15	20	4.50	6	13	1.68	17.1	5
	15	.	.	.	.	.	.	.	.	.	50.8	0

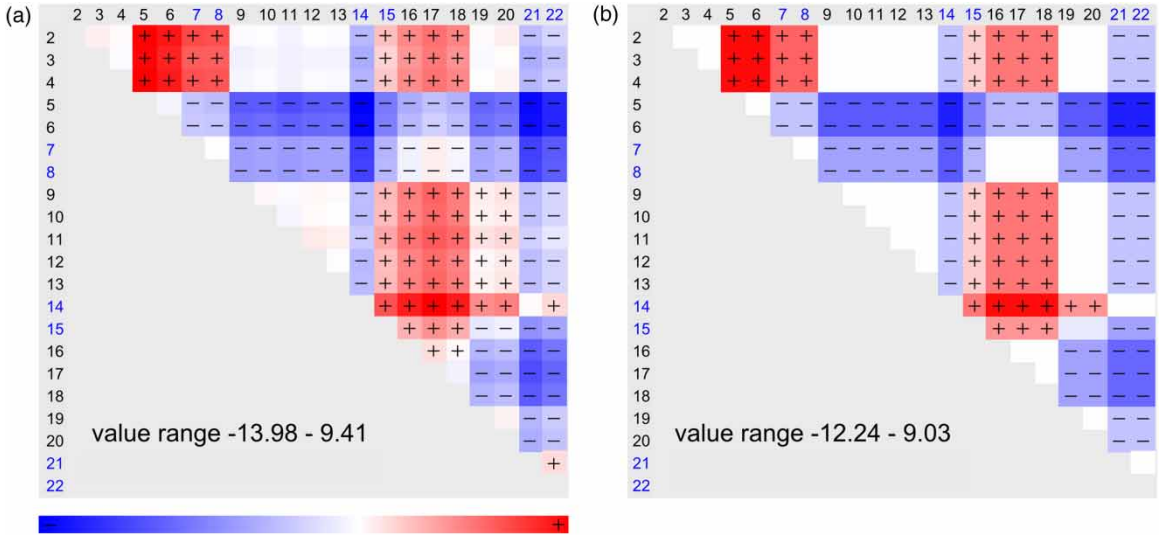

Figure 4 | Graphical results for test 7 performed on dataset 1c: (a) diagram with matrix of *b*-factors; (b) diagram with estimated block functions.

In Figure 5(b), the estimated block functions are visible. With the increased noise, it becomes more difficult to interpret and visually identify anomalies in the matrix of *b*-factors. The visual interpretation of Figure 5(b) is much easier, since the added noise is not visually represented. The longer duration of the second block function and the lower estimated amplitude are also clear in Figure 5(b).

Regarding datasets 2a–2d, two block functions should be identified. The start and end columns of the block function describing the first anomaly should be 4 and 10, and the amplitude should be 10 m^3/h. For the second anomaly the start and end columns of the block function should be 14 and 20, and the amplitude should be 5 m^3/h. Table 4 summarizes the obtained results from the different tests performed to datasets 2a–2d.

For datasets 2a, the majority of the performed tests identify three block functions. The start day of the second anomaly is identified 2 days later than the actual start date of the anomaly.

For dataset 2b, with 5% of Gaussian noise, several tests identify three block functions, overestimate the amplitude of both anomalies, and test three even identifies four blocks. Tests 2, 10, 13, 14, 15 lead to the identification of two block functions, solving the false positive issue. When adding 10% of noise to the data, the average C_i norm

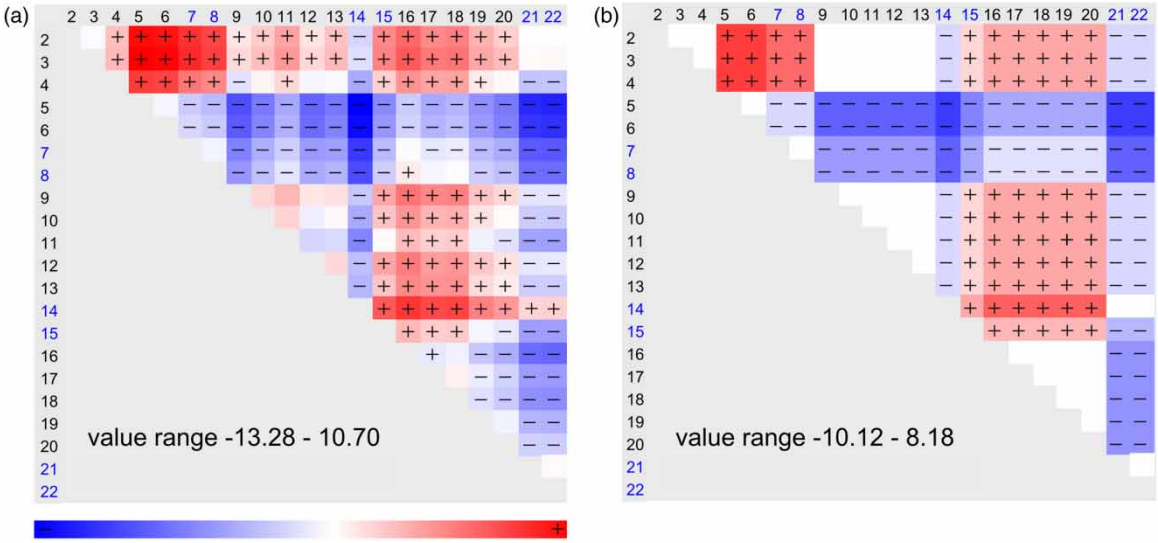

Figure 5 | Graphical results for test 1 performed on dataset 1d: (a) diagram with matrix of *b*-factors; (b) diagram with estimated block functions.

increases. The algorithm continues to overestimate the amplitude of the anomalies. Several tests identify the two anomalies, although the best results in terms of the C_i norm are obtained for test 7. Figure 6(a)–6(c) represents the results for dataset 2c. Figure 6(a) represents the matrix of *b*-factors. Figure 6(b) represents the block functions estimated by test 1, and Figure 6(c) represents the block function estimated by test 14. In Figure 6(a), the effect of the added noise is visible. When performing test 1, this noise is approximated by a third block function, visible in Figure 6 (b). Test 14 is able to ignore this noise and identifies only two block functions (Figure 6(c)).

For the last dataset, with 20% of added noise, most tests are able to identify the two block functions describing the added anomalies. The best results in terms of the C_i norm are again obtained by test 7. For test 10, affecting the step size, only one block function is identified and the C_i norm is the highest of all tests. For test 15 (highest f_1 penalty coefficient), no blocks are identified.

Datasets 3a–3d consider two anomalies that overlap during a few days. Two block functions should be identified. The start and end columns of the block function describing the first anomaly should be 5 and 15, and the amplitude should be 10 m^3/h. For the second anomaly the start and end columns of the block function should be 12 and 20, and the amplitude should be 5 m^3/h. Table 5 summarizes the results obtained by performing the different tests.

For dataset 3a, all tests except the default test identify two block functions. The estimated amplitudes are close to real amplitudes of the anomalies. For test 1, where the start and end dates are less accurate, the algorithm also identifies a third block function with positive amplitude.

When adding 5% Gaussian noise, the results are similar. However, for test 1 the first identified block function actually describes the overlap of both anomalies, by estimating an amplitude equal to 15.46 m^3/h, and estimating accurately the start and end days of the overlap, this is 12–15.

For dataset 3c, most of the tests identify two block functions.

For dataset 3d, with 20% added noise, the C_i becomes significantly higher. Again, test 1 leads to the best fit between the block functions and the anomalies. Figure 7(a) and 7(b) illustrate some of the obtained results when considering dataset 3c, namely the block functions obtained by test 1. Between days 12 and 15 the block functions overlap and the color of the block is darker, illustrating the higher amplitude.

Influence of noise and parameters

Table 6 gives an overview of the influence of the noise, gap between anomalies, and the parameters considered to run the algorithm on the obtained results.

Table 4 | Characteristics of the block functions obtained by the performed tests on datasets 2a–2d, corresponding used steps and C_i norm

		Block function 1			Block function 2			Block function 3			Block function 4				
Dataset	**Test**	**Start**	**End**	**Amplitude (m^3/h)**	**Start**	**End**	**Amplitude (m^3/h)**	**Start**	**End**	**Amplitude (m^3/h)**	**Start**	**End**	**Amplitude (m^3/h)**	**C_i**	**Number of steps used**
2a	All except	4	10	8.80	16	20	4.19	21	22	–3.51	.	.	.	16.4	4
	2, 14, 15	4	10	9.18	16	20	4.62	.	.	.	.	.	.	19.6	4
	6, 9, 10	4	10	10.50	11	20	3.92	.	.	.	.	.	.	27.2	3
2b	All except	4	10	10.56	16	20	5.86	11	15	2.2	.	.	.	15.5	4
	2, 13, 14, 15	4	10	9.26	16	20	4.61				.	.	.	17.9	4
	3	4	6	10.23	7	10	7.96	16	20	4.42	21	22	−3.56	14.1	5
	10	4	10	10.55	11	20	3.78	.	.	.	.	.	.	26.8	3
	16	4	10	12.39	11	20	5.68	7	16	–3.01	.	.	.	14.9	5
2c	1, 3, 11, 12, 16, 17	4	10	10.61	16	20	7.189	9	16	2.08	.	.	.	17.6	5
	2, 13, 14, 15, 18, 19, 20	4	10	10.002	16	20	5.815	.	.	.	.	.	.	21.5	4
	6, 9, 10	4	10	11.37	11	20	4.41	.	.	.	.	.	.	34.7	3
	7	4	10	12	9	20	7.5	7	16	–5.61	.	.	.	16.7	6
2d	All except	4	10	9.51	16	20	5.08	.	.	.	.	.	.	23.9	4
	7	4	10	11.74	9	20	6.22	6	15	–4.54	.	.	.	17.7	6
	10	4	10	7.9	.	.	.	.	.	.	.	.	.	40.9	2
	11, 12, 16	4	10	10.17	16	20	6.56	9	15	2.24	.	.	.	23.7	5
	15	.	.	.	.	.	.	.	.	.	.	.	.	69.6	0

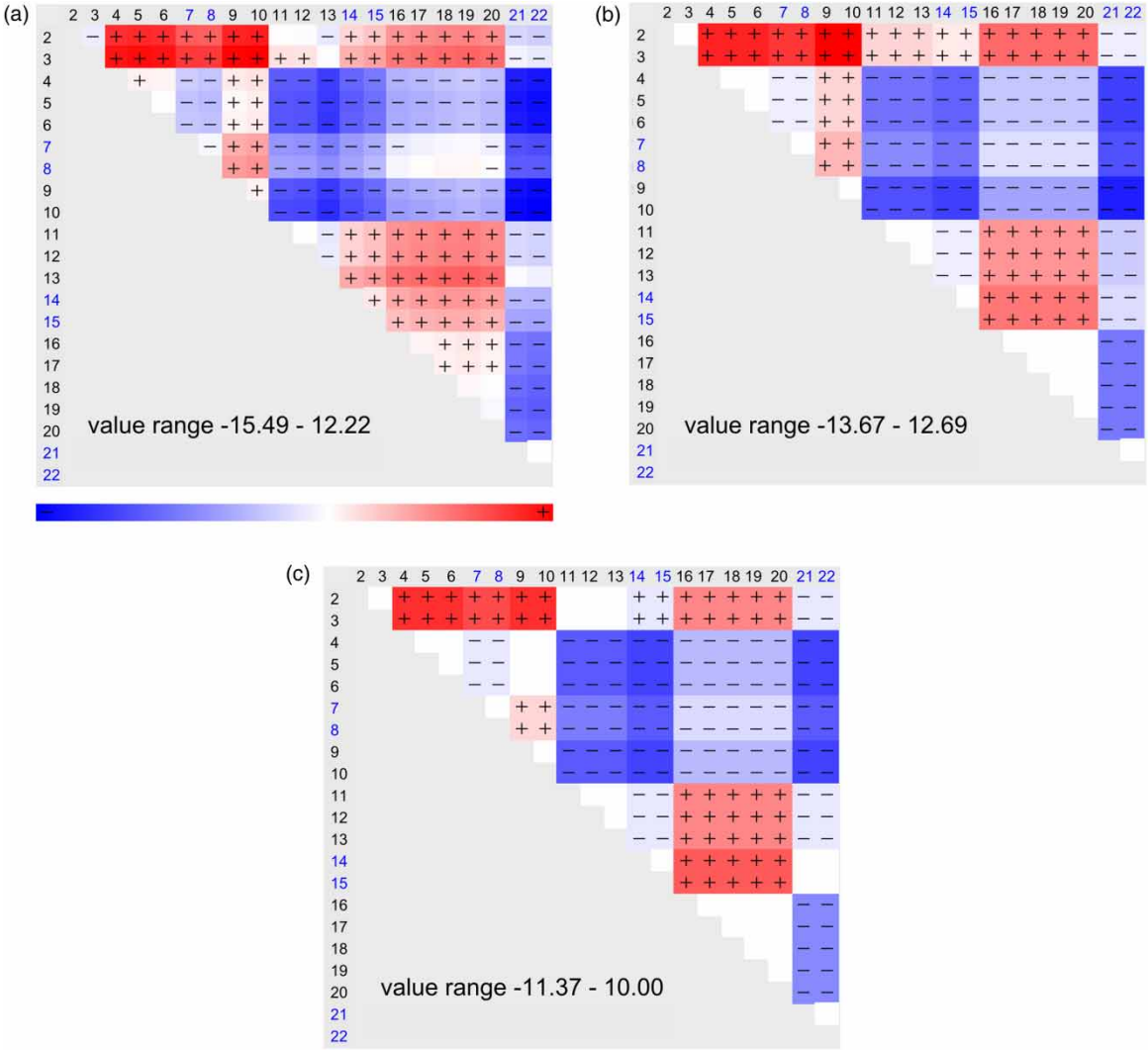

Figure 6 | Graphical results for dataset 2c: test 1 and 14 performed on dataset 2c: (a) diagram with matrix of *b*-factors; (b) diagram with estimated block functions by test 1; (c) diagram with estimated block functions by test 14.

Table 5 | Characteristics of the block functions obtained by the performed tests on datasets 3a–3d, corresponding used steps and C_i norm

		Block function 1			Block function 2			Block function 3				
Dataset	**Test**	**Start**	**End**	**Amplitude (m^3/h)**	**Start**	**End**	**Amplitude (m^3/h)**	**Start**	**End**	**Amplitude (m^3/h)**	**C_i**	**Number of steps used**
3a	All except	5	15	10.16	12	20	5.01	.	.	.	4.2	4
	1	5	13	10.25	14	16	9.7	12	20	5.16	4	5
3b	All except	5	15	10.08	12	20	5.39	.	.	.	4.4	4
	1	12	15	15.46	7	11	10.11	16	20	5.43	4.4	4
3c	All except	5	15	9.98	12	20	5.12	.	.	.	7.1	4
	1	5	15	9.69	12	20	4.86	21	22	–1.28	5.9	4
3d	All except	5	15	10.71	14	20	4.49	.	.	.	20.4	4
	1	9	15	12.76	5	9	9.42	16	20	5.59	16.9	4
	10	5	15	8.12	9	20	4.45	.	.	.	18.8	4

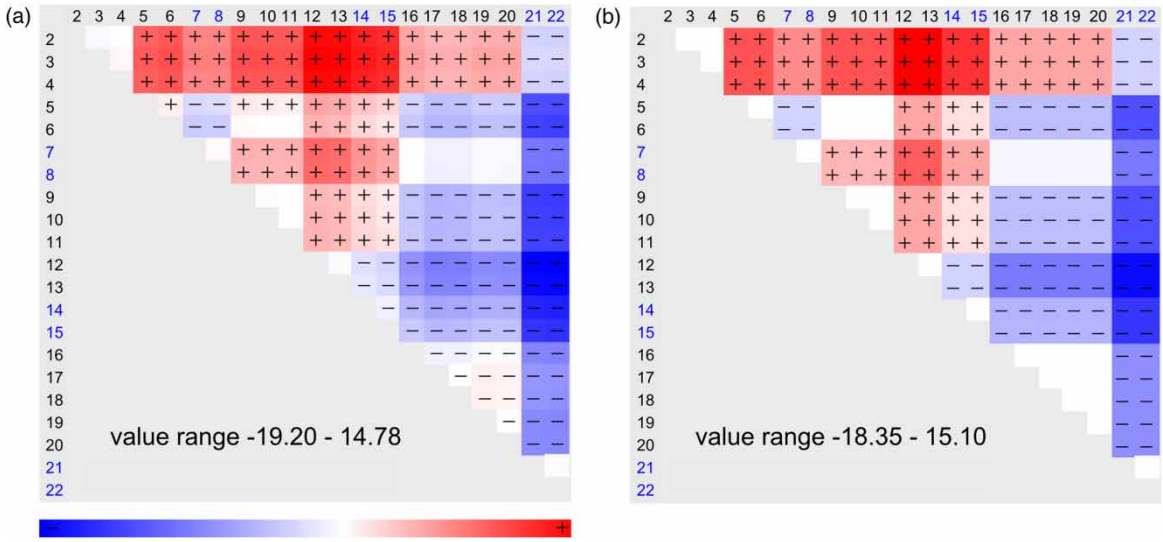

Figure 7 | Graphical results for dataset 3c: (a) diagram with matrix of *b*-factors; (b) diagram with estimated block functions by test 1.

Real data

The aforementioned synthetic data share a common characteristic: the introduced anomalies are relatively abrupt. In real data anomalies can occur either in a progressive or in an abrupt manner and the signal may be noisier than that in the considered synthetic tests. Therefore, anomalies can be harder to detect. Thus, to assess the performance and capability of the CuBOid algorithm on the detection of natural anomalies in real data with varying noise conditions, flow measurements series from the municipal drinking water company of the city of Paris, Eau de Paris, were considered. The water company serves 4 million consumers during the day, and 2 million during night time, and has an average water consumption of 550,000 m³/day. For a detailed description of the Parisian drinking water distribution system the reader is referred to Montiel & Nguyen (2011, 2013).

The flow data considered in this paper are an extract from Paris real-time SCADA system historical records. The quality of the registered data is varying, with data gaps and periods of anomalous signals and many periods of continuous, good quality registration occurring in all of the DMAs. Van Thienen & Montiel (2014) have presented a non-censored list of registered leaks, and the results obtained by the application of the CFPD block analyses (non-automatized, so no application of CuBOid) to these data. In most cases, the leaks could be recovered. Presently, we applied the CuBOid algorithm to the same data in order to retrieve the anomalies from the same leakage list. Different sets of parameter values controlling the CuBOid algorithm were tested and results were compared. For all of the tested combinations the algorithm was able to identify almost all of the registered leaks (success of identification and its practical meaning are discussed below). The differences consisted of the estimated amplitudes and the number of identified blocks describing each anomaly. The best results, in terms of amplitudes and number of blocks, were obtained for the following set of parameters: number of steps = 3, number of clusters = 5, $L_x = 0.7$, $f_1 = 0.01$, $f_2 = 0.7$. The results obtained for this set are shown in Table 7.

As can be seen in Table 7, for most cases, the CuBOid algorithm has succeeded in autonomously detecting the anomalies. The algorithm failed to detect four of the 22 registered leaks, namely the leaks at the DMAs of Belleville Réservoir, Cité Universitaire, Plaine Vaugirard (1) and Sorbonne. The registered leak at Belleville Réservoir is a single day event, harder to detect by the algorithm. In the case of Cité Universitaire, data gaps prevented the CuBOid algorithm from finding good solutions. Incomplete event registrations of anomalies at Plaine Vaugirard (1) and Sorbonne hinder the interpretation of results, although in the latter case, the anomaly which is detected seems unrelated.

For the identified anomalies the results were assessed in two ways: accuracy of identified start and end-dates and

Table 6 | Overview of the influence of some characteristics of the datasets and the parameters considered to run the algorithm on the obtained results

Characteristics of dataset and parameter	Effect
Noise	Higher noise values lead to a decrease of the estimated amplitude of the block functions – especially visible in dataset 1 Higher noise values make the algorithm more sensitive to the f_1 penalty coefficient: for datasets 1 and 2 the algorithm fails to identify block functions when higher values for the f_1 penalty coefficient are considered
Gap between anomalies	Overall results for datasets 1 are better than the results for datasets 2. The difference between sets 1 and 2 is the duration of the added anomalies: for sets 2 anomalies last longer, and the gap between them is shorter. This makes it harder for the algorithm to clearly identify two separate block functions For datasets 2, the algorithm has more difficulties in identifying the four necessary steps to describe the block functions. For several tests, the algorithm uses, or less or more steps, than the ones required for the block identification. For datasets 1 and 3, and for the majority of the tests, the four necessary steps are well identified
Number of clusters	The number of clusters significantly influences the computational time. When three and four clusters are considered the average computational times are respectively 6 to 17 times longer than when two clusters are considered. Since the generated datasets have only two anomalies, setting the number of clusters equal to two is ideal. However, when performing the test to real data, from which anomalies are not known beforehand, but instead are desired to be identified, setting the number of clusters to two can entail some risks such as not identifying more anomalies than two, if they exist. On the other hand increasing the number of clusters can lead to the identification of more blocks than the actual anomalies, mainly if anomalies occur soon after each other and there is some noise in the data. A suitable value for the f_1 penalty factor should be chosen to prevent this issue
Number of steps	The number of considered steps also influences the computational time. When using five or six steps instead of four, the computational times are five and eight times longer, respectively Increasing the number of steps can lead to better results, especially when more noise is added to the data. However, it also leads to the identification of extra block functions in some cases. A suitable value for the f_1 penalty factor should be chosen to prevent this issue
L_x norm	Using the L_2 norm to determine the steps size leads to worse results in terms of the distance between the identified block functions and the matrix of *b*-factors. This effect becomes even more evident when the added noise increases. On the other hand, the use of the L_2 norm seems to decrease the risk of identifying a third block Two intermediate values for the L_x norm were also considered (0.7 and 1.25). In some tests the lower value lead to better results, while the higher value leads to worse results
Penalty f_1	For several tests, when using a very small f_1 penalty, (0.01), the algorithm identifies a third block function, located between the anomalies. With this very small penalty, the algorithm is not penalizing the use of more block functions and adds a block which is fitting the added noise. Increasing the f_1 penalty solves this problem. For datasets 1a–1c, it is sufficient to consider a f_1 penalty of 0.33 However, for datasets 2a–d, the algorithm benefits from higher f_1 penalty values, and in some cases to avoid the identification of a third block it is necessary to increase the f_1 value to 0.7
Penalty f_2	For most of the performed tests the value of the f_2 penalty has no influence on the results. The exceptions are for datasets 2c where increasing the f_2 penalty avoids identifying a third block

estimated intensity. Regarding the start and end-dates three situations were identified: good agreement of a single block, good agreement of combination of blocks and identification of leakage repair. The first situation refers to anomalies that are identified by a single block and for which the start and/or end-dates are the same, or within 1 day difference, of the corresponding reported dates. Since analysis were carried out on a monthly basis, in some cases the end-date matches the last day of a month. This happens, for instance, for Belleville. The start-date is 1 day from the registered date, but the end-date corresponds to the last day of the period of analysis (30-4-2011). To overcome this issue the analysis could be repeated considering a 2 month period. The second situation refers to anomalies that were

Table 7 | Overview of registered leaks and anomalies identified using CFPD analysis with the CuBOid algorithm

	Reported			CFPD Block Analysis			
DMA	Start	End	Amplitude	Start	End	Amplitude	Succes
Belleville	27-4-2011 0:00	5-5-2011 0:00	80	28-4-2011	30-4-2011	65	✓
Belleville Réservoir[1]	9-12-2011 04:00	9-12-2011 12:00	peak 3500				×
Chapelle	12-3-2012 0:00	23-3-2012 10:30	300	6-3-2012	23-3-2012	4662	✓
				24-3-2012	31-3-2012	4386	
Cité Universitaire[4]	5-1-2012 10:00	24-10-2012 11:00	35				×
Convention	11-6-2011 3:00	15-6-2011 14:00	40	11-6-2011	23-6-2011	28	✓
Courcelles (1)	11-2-2012 21:30	16-2-2012 15:00	700	12-2-2012	16-2-2012	451	✓
Courcelles (2)	27-4-2012 0:00	2-5-2012 0:00	300	27-4-2012	30-4-2012	686	✓
Courcelles* (3)	26-9-2012 5:15	27-9-2012 12:00	1100	27-9-2012	27-9-2012	280	✓
				28-9-2012	29-9-2012	-512	
Daumesnil	11-9-2012 3:00	8-10-2012 14:00	100	11-9-2012	22-9-2012	406	✓
				23-9-2012	28-9-2012	336	
				10-9-2012	29-9-2012	-312	
Fabien (1)	15-1-2011	20-1-2011	700	18-1-2011	19-1-2011	1506	✓
				17-1-2011	19-1-2011	548	
				20-1-2011	30-1-2011	222	
Fabien (2)	8-7-2011 0:00	3-8-2011 0:00	640	27-7-2011	31-7-2011	-986	✓
Maine	24-5-2012 5:10	26-5-2012 20:50	250	25-5-2012	28-5-2012	215	✓
Menilmontant	13-5-2011 13:00	15-5-2011 9:00	2200	14-5-2011	15-5-2011	3346	✓
Nation	25-12-2011 4:30	27-12-2011 0:00	1100	25-12-2011	26-12-2011	248	✓
				26-12-2011	29-12-2011	918	
				27-12-2011	29-12-2011	-707	
Olympiades (1)[3]	28-2-2012 14:00	18-10-2012 9:00	50	1-10-2012	15-10-2012	-266	✓
				15-10-2012	30-12-2012	-73	
Olympiades (2)[1]	24-9-2012 10:50	24-9-2012 17:30	700	24-9-2012	30-9-2012	-230	✓
				25-9-2012	30-9-2012	214	
Plaine Vaugirard (1)[2]	29-11-2010		260				×
Plaine Vaugirard (2)	30-3-2011	27-4-2011	60	26-4-2011	30-4-2011	-57	✓
Rivoli (1)	20-12-2010 0:00	25-1-2011 12:00	300	13-12-2010	22-12-2010	165	✓
				23-12-2010	24-12-2010	247	
				25-12-2010	31-12-2010	304	
Rivoli (2)[1]	27-2-2012 1:30	27-2-2012 18:45	670	25-2-2012	27-2-2012	210	✓
Sorbonne[2]		17-6-2011 12:00	50	30-5-2011	5-12-2011	-26	×
Vaugirard	12-6-2011	17-6-2011	110	12-6-2011	16-6-2011	147	✓

Legend:

good agreement of single block
good agreement of combination of blocks
end period of analysis
Identification of leakage repair
[1]leakage with one day duration
[2]incomplete registration
[3]requires one year period analysis
[4]data continuity issues

amplitude (deviation from registered value):

>100%
75-100%
50-75%
25-50%
0-25%

*block width=1

found not as one single block, but as a succession of blocks. This is probably related to the noisy character of the dataset (in the sense that many things are going on). Even though it would have been more elegant for the algorithm to find these as a single block, for operational purposes it does not really make a difference. A special case of this type is presented by Chapelle. The corresponding signal shows a huge anomaly, apparently unrelated to the leak, of more than 4,000 m^3/h which drops by approximately 300 m^3/h at the reported date of the fixing of the leak. The third situation refers to blocks that identify not the leakage, but the leakage repair. In these cases, the start-date of the block is closer to the end-date of the registered anomaly. For instance, for Plaine Vaugirard (2), the algorithm identifies a block starting at 26-04-2011, 1 day earlier than the end-date of the registered anomaly. In this case the estimated intensity of the anomaly is also negative, due to the reduction in measured flow. This leads us to the estimated intensities: results were classified using different shades of green (or grey), representing the relative deviation from the registered value. In many cases, the amplitude matches the amplitude estimated by Eau de Paris quite well. Note, however, that a mismatch in the start date or amplitude may also be due to an inaccuracy in the original estimate.

Figure 8 illustrates the obtained graphical results for Courcelles (month of February 2012), Maine (month of May 2012) and Vaugirard (month of June 2011). It is visible that the inherent noise of the data make human interpretation of these block diagrams more difficult, while the algorithm performs well on clearly identifying the anomalies, emphasizing the capability and usefulness of the algorithm.

As mentioned above, some natural anomalies have a smooth rather than an abrupt initiation (a leak with growing flow rate over time). In an extension of this work, these could also be included in the analyses with a separate type of block function, with two non-zero segments, the first linearly rising from 0 to 1 and the second a constant 1.

Operational application of the CFPD method and the CuBOid algorithm will clearly not focus on the rapid detection of large bursts. More suitable methods exist, e.g. monitoring for combined flow increases and pressure drops above threshold levels. As CFPD depends on duration of anomalies for their detection, it is more appropriate for detecting smaller, less urgent leakages which nevertheless may represent a significant amount of water lost over longer periods of time. As such, an accurate determination of the amplitude of anomalies is more important than an accurate determination of the start and end dates. Also, representation by the method of a single anomaly as a succession of multiple blocks rather than a single block, as sometimes seen in our results, does not present a problem. The method can be implemented as part of a monitoring system for relatively small leakages, identifying anomalies, e.g. one per week or month and sending suspect anomalies (for which a grading or classification may need to be developed) to human operators for further analysis.

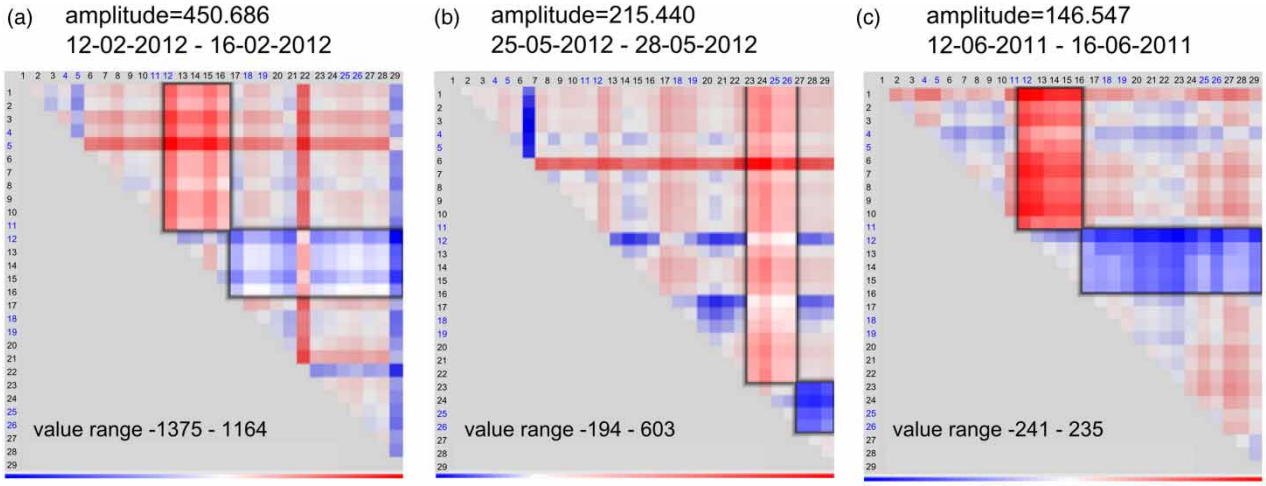

Figure 8 | Graphical results for: (a) Courcelles, February 2012; (b) Maine, May 2012; (c) Vaugirard, June 2011.

CONCLUSIONS

In this paper, we presented the CuBOid algorithm for the automated detection of anomalies in CFPD block diagrams.

The automated recognition of features in CFPD block diagrams has several advantages. The tests which have been performed demonstrate clearly that the method works well to objectively identify anomalies in synthetic data, with automated estimation of start and end dates as well as amplitudes. Successful application of the method to real flow data from Paris, showing autonomous detection of 82% of known anomalies, shows that the CuBOid algorithm can also perform well in operational conditions. However, a broader application to different datasets and distribution systems is required to generalize this conclusion. This algorithm can remove the need for human interpretation of matrices of *a* and *b*-factors in the CFPD block analysis method. This means that analysis time is reduced and greater objectivity and reproducibility of the analyses are achieved. Moreover, it opens the possibility of application to automatized alarms. Therefore, the logical next step would be application in a real distribution network as part of the operational framework.

Even though the CuBOid algorithm has been shown to provide a useful addition to the CFPD algorithm, it will fail to recognize anomalies with amplitudes significantly below system noise levels (e.g. stochastic variability). This is a limitation of the CFPD method rather than the CuBOid algorithm, which is investigated in more detail in Van Thienen (2013), and is a limitation of other leak detection methods as well. Also, the main power of the CFPD method is in recognizing events which last multiple days. The CuBOid algorithm does not change this, as this issue is intrinsic in the CFPD method. For the rapid detection of anomalies within minutes or hours, more suitable methods exist.

There is, however, room for improvement in the CuBOid algorithm in the sense that events with a less block-like shape, such as slowly increasing leakage rates, can be included in the future by defining specific shape functions for these.

Fine tuning the algorithms' parameters is important to obtain better results. At this point, the need for setting the adequate values for these several parameters might be a drawback of the presented method. This paper provides some insights on the influence of these parameters on the outcoming results. For practical applications it would be easier to provide some rules of thumb for the choice of these parameters. Deriving these rules requires more extensive tests, considering series of water flow data from several distribution systems with different characteristics. That is why future developments should also include: (1) a more extensive investigation on the influence of the algorithms' parameter values on subsequent results, including combinations not considered in the present paper (Table 1); (2) tests on real flow data coming from water distribution systems with different characteristics and containing different types of anomalies.

ACKNOWLEDGEMENTS

Considerate comments resulting from thorough reviews by three anonymous reviewers have helped to significantly improve the quality and clarity of the paper. These are gratefully acknowledged. The authors would also like to acknowledge W-SMART and the utilities participating in and sponsoring its INCOM project (Eau de Paris and Eaux du Nord), Frank Montiel of Eau de Paris for providing the analyzed data, Cédric Auliac of CEA for fruitful discussions and thoughtful comments on an early version of the paper, and also the European Water Innovation fund of the Dutch water companies for additional funding. Part of the work was performed in the SmartWater4Europe project. This project has received funding from the European Union's Seventh Programme for research, technological development and demonstration under grant agreement number 619024.

REFERENCES

Ahmed, N., Natarajan, T. & Rao, K. R. 1974 Discrete cosine transforms. *Comp. IEE Trans.* **100** (1), 90–93.

Akansu, A. N., Serdijn, W. A. & Selesnick, I. W. 2009 Emerging applications of wavelets: a review. *Phys. Commun.* **3**, 1–18.

Aksela, K., Aksela, M. & Vahala, R. 2009 Leakage detection in a real distribution network using a SOM. *Urban Water J.* **6** (4), 279–289.

Beuken, R. H. S., Lavooij, C. S. W., Bosch, A. & Schaap, P. G. 2006 Low leakage in the Netherlands confirmed. In: *Water Distribution Systems Analysis Symposium*. Cincinnati.

Farley, M. & Trow, S. 2003 *Losses in Water Distribution Networks*. IWA Publishing, London.

Lambert, A. O. 2002 Water losses management and techniques. *Water Supply* **2** (4), 1–20.

Liggett, J. A. & Chen, L.-C. 1994 Inverse transient analysis in pipe networks. *J. Hydraul. Eng.* **120** (8), 934–955.

Mamo, T. G., Juran, I. & Shahrour, I. 2014 Virtual DMA municipal water supply pipeline leak detection and classification using advance pattern recognizer multi-class SVM. *J. Pattern Recog. Res.* **1**, 25–42.

Montiel, F. & Nguyen, B. 2011 Efficient real and differed time tools and method for leakage detection in the city of Paris. In: *6th IWA Specialist Conference on Efficient Use and Management of Water*. Dead Sea, Jordan.

Montiel, F. & Nguyen, B. 2013 Real-time water leak detection and analysis tools. In: *7th International Conference on Efficient Use and Management of Water*. Paris, France.

Mounce, S. R., Boxall, J. B. & Machell, J. 2010 Development and verification of an online artificial intelligence system for detection of bursts and other abnormal flows. *J. Water Resour. Plann. Manage.* **136** (3), 309–318.

Oliphant, T. E. 2007 Python for scientific computing. *Comp. Sci. Eng.* **9**, 90.

Palau, C. V., Arregui, F. J. & Carlos, M. 2012 Burst detection in water networks using principal component analysis. *J. Water Resour. Plann. Manage.* **138** (1), 47–54.

Poulakis, Z., Valougeorgis, D. & Papadimitriou, C. 2003 Leakage detection in water pipe networks using a Bayesian probabilistic framework. *Prob. Eng. Mech.* **18**, 315–327.

Puust, R., Kapelan, Z., Savic, D. & Koppel, T. 2006 Probabilistic leak detection in pipe networks using the SCEM-UE algorithm. In *Water Distribution Systems Analysis Symposium*. Cincinnati, USA.

Puust, R., Kapelan, Z., Savic, D. A. & Koppel, T. 2010 A review of methods for leakage management in pipe networks. *Urban Water J.* **7** (1), 25–45.

Romano, M., Kapelan, Z. & Savić, D. 2013 Geostatistical techniques for approximate location of pipe burst events in water distribution systems. *J. Hydroinform.* **15** (3), 634–651.

Romano, M., Kapelan, Z. & Savić, D. 2014 Automated detection of pipe bursts and other events in water distribution systems. *J. Water Resour. Plann. Manage.* **140** (4), 457–467.

Savic, D., Lambert, A. & Kapelan, Z. 2005 Water losses management and leakage detection techniques for water distribution systems. *Water Sewer. J.* **2**, 25–27.

Van Thienen, P. 2013 A method for quantitative discrimination in flow pattern evolution of water distribution supply areas with interpretation in terms of demand and leakage. *J. Hydroinform.* **15** (1), 86–102.

Van Thienen, P. & Montiel, F. 2014 Flow analysis and leak detection with the CFPD method in the Paris drinking water distribution system. In: *11th International Conference on Hydroinformatics*. New York City, USA.

Van Thienen, P., Pieterse-Quirijns, I., Vreeburg, J. H. G., Vangeel, K. & Kapelan, Z. 2013a Applications of discriminative flow pattern analysis using the CFPD method. *Water Sci. Technol. Water Supply* **13** (4), 906–913.

Van Thienen, P., Vreeburg, J. & De Kater, H. 2013b Water flow data key to pinpointing change. *Water21*, June 2013, 36.

Vítkovský, J. P., Lambert, M. F., Simpson, A. R. & Liggett, J. A. 2007 Experimental observation and analysis of inverse transients for pipeline leak detection. *J. Water Resour. Plan. Manage.* **133** (6), 519–530.

Wu, Z. Y. (ed.) 2011 *Water Loss Reduction*. Bentley Institute Press, Exton, Pennsylvania.

Wu, Z. Y., Sage, P. & Turtle, D. 2010 Pressure-dependent leak detection model and its application to a district water system. *J. Water Resour. Plan. Manage.* **136** (1), 116–128.

First received 13 March 2015; accepted in revised form 14 October 2015. Available online 27 November 2015