Water Planning and Management: An Extended Model for the Real-Time Pump Scheduling Problem

Louise Brac de la Perrière, Antoine Jouglet, Alexandre Nace and Dritan Nace

Abstract This work deals with the modeling of a water distribution system with a view to optimizing real-time pump schedules through the use of an integer linear programming (ILP) tool. We describe a number of different constraints that need to be modeled, and we provide mathematical models for each of them. Along with the more familiar capacity and demand satisfaction constraints, we consider specific constraints related to requirements such as water quality and maximum power consumption. Problems include taking the discrete behavior of pumps into account and modeling hydraulic phenomena. Solutions must be applicable to very large networks (about 100 pumps and 50 storage tanks in our example), and available computation time is limited in practice (30 min maximum in the application in hand) because of the real-time scheduling constraint. This problem is not new, but there are very few studies in the literature that address these various aspects of the problem at the same time, which implies a significant number of variables. We present as exhaustive a panorama as possible of the constraints and elements that have to be considered when modeling a water distribution system, and then describe our own model and computational work.

Keywords Real-time · Pump scheduling · Water distribution system · Integer linear programming

L. B. de la Perrière (✉) · A. Nace
Ondeo Systems, 38 rue du président Wilson 78230 Le Pecq, France
e-mail: louise.bracdelaperriere@suez-env.com

A. Nace
e-mail: alexandre.nace@suez-env.com

L. B. de la Perrière · A. Jouglet · D. Nace
Heudiasyc UMR CNRS 7253, BP 20529 60205 Compiegne cedex, France
e-mail: antoine.jouglet@hds.utc.fr

D. Nace
e-mail: dritan.nace@hds.utc.fr

P. Gourbesville et al. (eds.), *Advances in Hydroinformatics*,
Springer Hydrogeology, DOI: 10.1007/978-981-4451-42-0_13,

1 Introduction

Given a water distribution network and some forecasts about consumption at different nodes of the network over a specific time horizon, the problem addressed here is the scheduling of water pump jobs such that water demands are satisfied and conform to quality standards set by French and European legislation while minimizing operating costs (water treatment and electricity).

Among the approaches in the literature for solving this problem, we find linear programming (LP) [1], nonlinear programming [2], ant colony optimization [3], and genetic algorithms [4].

Each method has its own advantages and drawbacks, and a strategy may be more or less efficient, depending on the characteristics of any given network. The characteristics of the networks that concern us in particular are as follows:

- The considered networks are very large (about 100 pumps and 50 storage tanks).
- The computation time available for devising optimized solutions is relatively short (30 min maximum) because of the real-time scheduling constraint.
- The discrete behavior of pumps needs to be expressed through the model. The pumping stations work at some defined level of pumping depending on the type and the power of the station and changes in the pumping regime can take place only periodically.

We chose to use LP for several reasons. First, our intention is to enhance an existing tool that is based on LP. Secondly, LP provides an appropriate modeling tool for most of the constraints encountered in drinking water distribution networks, including some related pressure constraints (known to be difficult to handle within an LP framework but that we have managed to express in a linear form). Finally, efficient solvers are already available.

The starting point is an existing software tool, Saphir [5], which already models the pump-scheduling problem with LP. Our aim is to enhance this tool so as to achieve the following:

- Modeling, through the use of LP, new constraints that have not been expressed before.
- Expressing in a linear form the hydraulic phenomena in the system.
- Placing a limit on computation time while dealing with integer LP models of large dimensions.

The first difficulty is to model new elements of the network (such as pumps with variable flow) and to take into account the discrete behavior of pumps. This discrete behavior makes the optimization problem more difficult to handle: while the continuous problem can be solved with a polynomial algorithm, the mixed integer linear programming (MILP) problem is NP-hard and requires a branch and price procedure [6], but it represents the real-life operation of the pumps much better than a continuous approach. We therefore included in this formalization new constraints that have not been expressed before. The second difficulty is to respect or even reduce the limits placed on computation time, and to this end, we have looked at a number of different strategies.

In this chapter, we describe the previous existing formalization of a water network with MILP, we describe the new constraints, we provide some results, and we show how these results allow a better choice of parameters in the optimization.

2 Water Distribution System Modeling

In this section, we present the main elements of a water distribution network, and we list the associated structural constraints.

2.1 Elements

A water distribution system can be described by nodes and edges; each element in the network implies the creation of a set of constraints. The time horizon is divided into different time slots. The aim of the optimization is to determine whether each element, and in particular each pump, should operate or not operate within each time slot.

Let us consider the nodes and their associated constraints. We consider four different kinds of node:

- *Production points* must respect the maximum flow production and the discrete behavior of the pumps. Some of these production points have a semi-discrete behavior, as their flow can vary between two fixed values. Each production point has only one outgoing pipe in our model.
- *Storage tanks* must respect the maximum and minimum volumes of water they can contain. The level of each storage at the end of the day also has a required value, and the non-respect of this constraint penalizes the objective function (the penalty is commensurate with the difference observed).
- *Consumption points* have only one input pipe in our model.
- *Connection nodes* represent points where several pipes are connected.

Each node in the model needs to respect an additional constraint, namely continuity of flow. For every connection node and storage tank, the volume of water entering the node during a given time slot is equal to the outgoing volume (with an acceptable gap corresponding to the variation of volume inside the storage tank).

The volume produced during a time slot by a production element is equal to the volume of water carried by the outgoing pipe during this time slot. Finally, the volume required by a consumption point during a given period is equal to the volume carried by its input pipe.

Let us now consider the different pipes to be found in the network model:

- Non-gravitational pipes have exactly the same constraints as for production points, because both are composed of pumps.
- Gravitational pipes have a continuous behavior: They only need to respect the maximum volume they are able to carry during a given time slot.

Each element of the network has to respect this basic set of structural constraints, to ensure continuity of flow and the coherence of the model. A large network therefore implies a large number of variables and constraints to be taken into account, which is one of the difficulties of our problem. Because of the discrete behavior of the pumps, we have to employ integer variables, which means using integer linear programming (ILP) models. The larger the network, the larger the number of these integer variables, which seriously impacts computation time.

2.2 Mathematical Formulation

We now introduce the mathematical formulation for the main elements of the network.

- For a production element p:

$$q_{p,t} = f_{p,t} \times Q_{p,t} \tag{1}$$

$$q_{p,t} = q_{k,t} \tag{2}$$

where parameter t represents the index of the considered time slot; element p is the index of the production node. Parameter $Q_{p,t}$ is the water volume that can be produced if element p operates during time slot t.

Element k is the index of the single output pipe from production node p.

Variable $f_{p,t}$ is a binary variable that represents the operation ($f_{p,t} = 1$) or non-operation state ($f_{p,t} = 0$) of element p. Variable $q_{p,t}$ is the volume of water actually produced by production node p during time slot t.

Variable $q_{k,t}$ is the volume of water carried by pipe k during time slot t.

Constraint 1 represents the flow of element p at each time slot.

Constraint 2 is the flow continuity constraint, which gives a flow value to element k.

Most of the time production elements have a discrete behavior as described above. But some production elements have a partial continuous behavior, which means that operation is discrete between 0 and a first flow value, and continuous between this value and the maximum flow. To take this behavior into account in the model, two variables $f_{p,t}$ are created and the equations rearranged.

$$q_{p,t} = f_{p,t} \times Q_{p,t} + f'_{p,t}\left(Q'_{p,t} - Q_{p,t}\right) \tag{3}$$

$$f'_{p,t} \leq f_{p,t} \tag{4}$$

$$0 \leq f'_p,\ t \leq 1 \tag{5}$$

where $f_{p,t}$ still represents the operation or non-operation of pump, $Q_{p,t}$ is the first acceptable flow value, $Q'_{p,t}$ is the maximum flow, and $f'_{p,t}$ is a continuous variable that allows the flow to fluctuate between the first flow value and the maximum flow.

- For a consumption element c:

$$q_{k,t} = q_{c,t} \tag{6}$$

where k is the index of the single input pipe into consumption point c, variable $q_{k,t}$ represents the volume transported by pipe k during time slot t, and variable $q_{c,t}$ represents the consumption demand of node c during time slot t.

- For each connection node o, the flow continuity constraint can be expressed as follows:

$$\sum_{k \in U_0^+} q_{k,t} = \sum_{k \in U_0^-} q_{k,t} \tag{7}$$

where set U_o^+ is the set of successors of element o, and set U_o^- is the set of predecessors of element o.

- Finally, a very similar expression can be obtained for a storage point l:

$$\sum_{k \in U_l^+} q_{k,t} = \sum_{k \in U_l^-} q_{k,t} + \Delta(n_{l,t}) \tag{8}$$

where variable $\Delta(n_{l,t})$ represents the variation in the volume held in storage tank l during time slot t.

3 Operational Constraints

In addition to these structural constraints that describe the functioning network, a number of operational constraints have to be considered. We first provide an exhaustive list of the operational constraints to be taken into account and then focus on two of them in particular.

3.1 Operational Constraints

Constraints are more or less hard to express. The following are qualified as "simple" because they can be expressed through LP using conventional tools.

The constraints to be found already in the previous LP model are as follows:

- The maximum or minimum volume to be pumped in a day for a given production element.
- The maximum electricity power levels defined by the electricity supplier's contractual obligations.
- The list of elements that must work only in cases of emergency.
- The maximum incremental rate for the flow in some elements of the network.

The constraints added in our new ILP model are as follows:

- Simultaneity and exclusion constraints (pumps that cannot work at the same time or that are required to work at the same time).
- A maximum inactivity period length for some pumps.
- Ensuring a required water quality level through the mixing of water from different sources in some storage tanks (for instance by guaranteeing a percentage from different water sources).
- Avoiding stagnation in storage tanks to ensure water quality.
- Taking account of transfer delays when using very long pipes.

We now focus on the stagnation avoidance constraint and on a specific constraint that does not appear in the list.

To maintain a high quality of distributed water, stagnation has to be avoided. Most of the time water renewal occurs naturally, but in some areas with low consumption, a constraint needs to be added in relation to storage.

We denote as δ the maximum time water may remain in the tank. For each time slot *t,* we can write:

$$n_{l,t} \leq \sum_{t}^{t+\delta} \sum_{k \in U_l^+} q_{k,t} \tag{9}$$

where $n_{l,t}$ represents the volume held in tank l at time slot t, and set U_l^+ is the set of pipes leading out of tank l. In other words, the volume held in the tank at a given time slot may not exceed the outgoing volume over the following δ hours, in order to force a water turnover.

Because of the diversity of existing networks, highly specific constraints may sometimes be encountered. The constraint described below was encountered in only one network and required particular modeling. Two tanks are connected by both a gravitational and a non-gravitational pipe. The use of a gravitational pipe is possible only when the difference in levels is sufficiently large (Fig. 1).

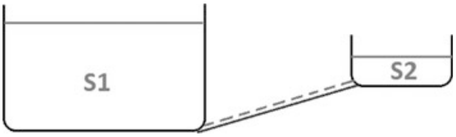


Fig. 1 Two tanks connected by a gravitational and a non-gravitational pipe

We expressed this constraint with ILP as follows:

$$h_{t+1} \geq m\left(1 - f_{g,t}\right) \tag{10}$$

$$h_t \geq m\left(1 - f_{g,t}\right) \tag{11}$$

$$0 \leq q_{g,t} \leq f_{g,t} \times Q^{+}_{g,t} \tag{12}$$

where variable h_t represents the difference in levels between tanks S1 and S2 at the beginning of time slot t (h_{t+1} represents this difference at the beginning of the next time slot following t, that is to say $t+1$).

Parameter m is the minimum value possible for variable h_t.

Parameter $Q^{+}_{g,t}$ is the greatest possible volume that can be transported by gravitational pipe during time slot t.

As before, the binary variable $f_{g,t}$ stands for the operation or non-operation of gravitational pipe g during time slot t.

These equations represent the possibility or the impossibility of using gravitational pipe g; the same equations can be used to represent the availability of pipe g', by changing variable h_t.

If the difference in levels is negative at the beginning of the time step (represented by h_t) or at the end of the time step (represented by h_{t+1}), then the variable $f_{g,t}$ is equal to zero. It is therefore impossible to use the gravitational pipe because the flow $q_{g,t}$ will be null.

3.2 Production-Related Constraints and Simplification

As explained above, expressing the problem through ILP increases computation time, which is a problem when seeking real-time solutions. To improve the solving process, we propose the aggregation of some pumps in a “super-pump” structure, in order to keep track of only the feasible flows. For example, having three identical pumping stations linked to the same connection node means that to obtain the same flow on this node, three pump combinations are possible, and so we have more variables than are needed to represent the same information, that is feasible flow. Therefore, under certain conditions, it will be more efficient to aggregate these three pumping stations into a single one.

The delicate aspects of this are as follows:

- Deciding when pumps cannot be aggregated (for example, if a particular constraint affects only one of the pumping stations).
- Transferring constraints from the previously separate pumps to the new aggregate one.

Conversely, in order to express certain constraints, it may be helpful to separate a pumping station with several allowed flows into different “simple pumps”.

For example, consider four pumps in the same pumping station. One of them can send water in two different directions, and only three pumps out of the four can work at the same time because of the maximum power consumption allowed by the electricity supply contract. To simplify, we separate the first pump into two (with an exclusion constraint meaning that they cannot operate during the same time slot), and express the constraints on the five pumps (Figs. 2, 3).

Depending on the constraint difficulty, two strategies may be helpful:

Dividing a complex production element into “pump units” may be helpful if there is no other simple way to express particular constraints, but this means increasing the number of variables.

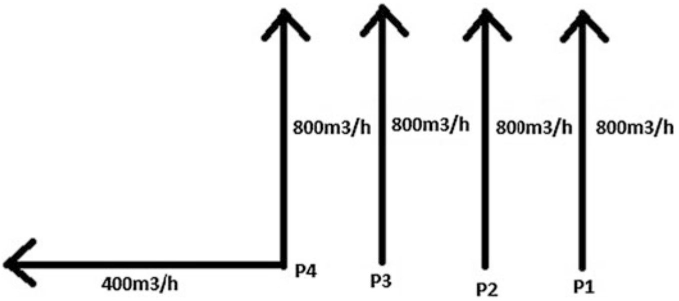


Fig. 2 Initial real configuration of pumps

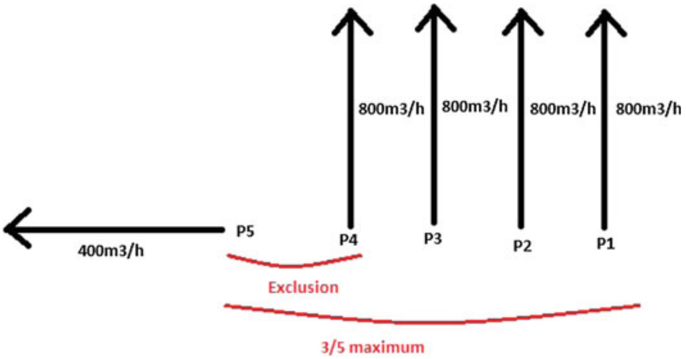


Fig. 3 New configuration and constraints associated

Where possible we aggregate pump units, in order to limit the number of variables and equations. For instance, situation involving only the pumps P1, P2, and P3 can led to a simple case of aggregation of pumps.

3.3 Hydraulic Phenomena

The hardness of modeling hydraulic phenomena is the first reason cited in the literature to disqualify the use of LP for the pump-scheduling problem. It was therefore especially important for us to show that hydraulic phenomena can be taken into account with a simplification precise enough for our needs. If the linearization of the hydraulic behavior of components is not possible, or unsatisfactory—as a result, for example, of an unacceptable increase in the number of constraints and the length of computation time—an alternative strategy may be to work with a hydraulic module during the optimization, in order to obtain the real flow values. This is already done in some applications such as the Derceto software [7].

A situation involving hydraulic phenomena arises when there are different gravitational pipes that bring water to or from a particular location, while the water distribution depends on pressure. The equations representing this situation are clearly nonlinear.

Another problem with pressure is water exchanges between storage tanks that supply the same consumption area. Given the difference in pressure between them their water levels tend to become balanced. To understand this phenomenon, we studied experimentally the behavior of two tanks supplying the same consumption area, with and without consumption.

The two storage tanks provide a certain amount of water to satisfy demand, and when consumption is low or null the water in the two of them will move toward the same level (Fig. 4). The equations above (obtained with Kirchhoff's nodal rule and Kirchhoff's mesh rule) represent the mathematical formalization of the flows

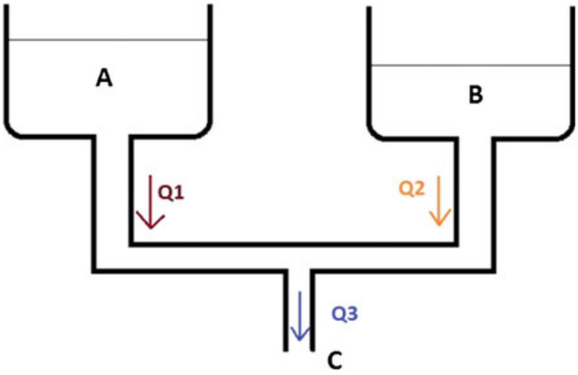


Fig. 4 Parallel tanks supplying the same consumption area *C*

$$\begin{cases} Q_1 + Q_2 - Q_3 = 0 \\ \dfrac{P_A}{\rho} + z_A + \dfrac{8Q_A^2}{\pi^2 g D_A^4} - \dfrac{P_C}{\rho} - z_C - \dfrac{8Q_C^2}{\pi^2 g D_C^4} - \dfrac{8 f_1 L_1 Q_1^2}{\pi^2 g D_1^5} - \dfrac{8 f_3 L_3 Q_3^2}{\pi^2 g D_3^5} = 0 \\ \dfrac{P_B}{\rho} + z_B + \dfrac{8Q_B^2}{\pi^2 g D_B^4} - \dfrac{P_C}{\rho} - z_C - \dfrac{8Q_C^2}{\pi^2 g D_C^4} - \dfrac{8 f_2 L_2 Q_2^2}{\pi^2 g D_2^5} - \dfrac{8 f_3 L_3 Q_3^2}{\pi^2 g D_3^5} = 0 \end{cases}$$

Fig. 5 Hydraulic behavior equations

Q_1, Q_2, and Q_3 between points A, B, and C (Fig. 5). We shall not explain the notation here, as our aim is simply to show the nonlinearity of the equations.

We attempted to model experimentally the movements using the hydraulic software tool EPANET. We found that the curve representing the exchanges between A and B when there is no demand can be linearized (Fig. 6).

We performed other experiments with nonzero consumption and for different pipe parameters, and we concluded that in the case of two parallel storage tanks, the different flows can be calculated from the experimental curves, based only on knowledge of consumption and water levels.

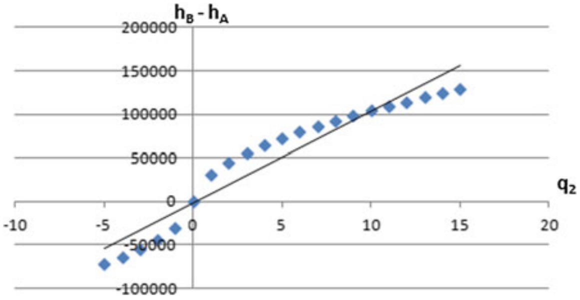


Fig. 6 Flow from storage B to A with respect to the difference in levels between B and A

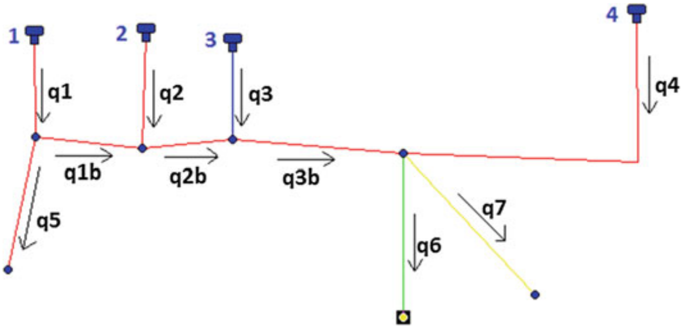


Fig. 7 New test network modeled with EPANET

We generalized this approach to a slightly more complicated case with 4 tanks, 10 pipes, and 3 consumption points (Fig. 7).

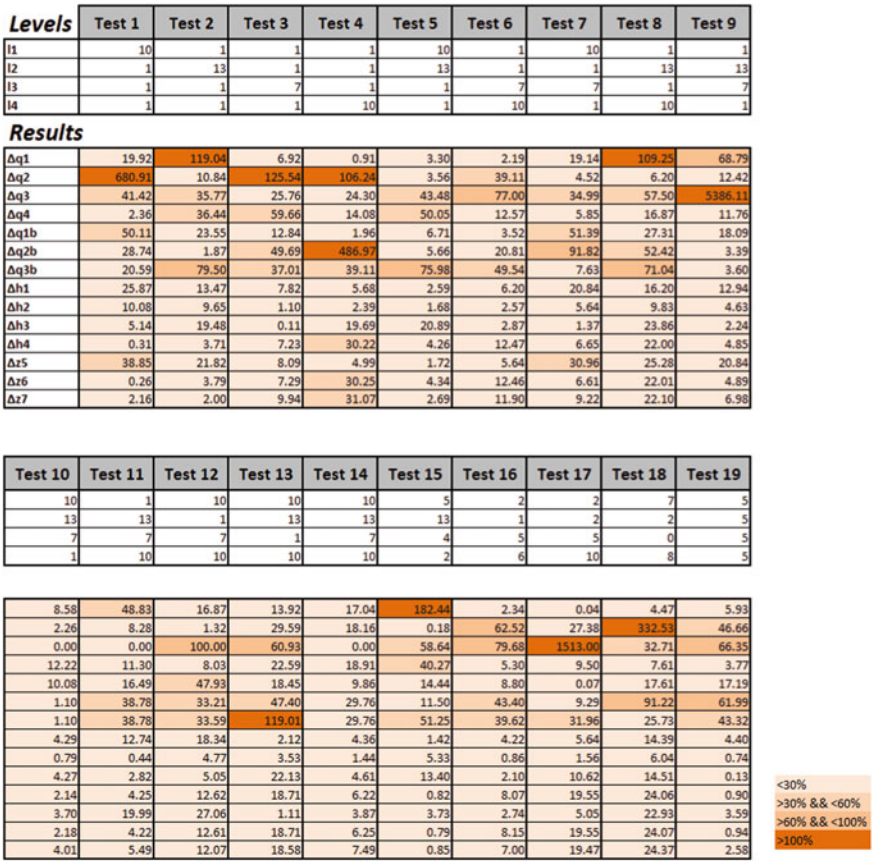
In addition to the flow values in the pipes, we add a pressure value on the nodes. On each pipe, we add a linear equation of the following form, where f is a linear function:

$$q_{k,t} = f\left(h_{i,t}, h_{f,t}, H_k\right) \tag{13}$$

The flow through a pipe q_k depends on the pressure $h_{i,t}$ in its initial node, in the pressure $h_{f,t}$ in its final node, and on certain known parameters of the pipes H_k (such as diameter and length).

For 19 different combinations of storage level values, we calculated the flow and pressure values on this small network, both with our linear model and with the hydraulic software tool EPANET (Table 1).

Table 1 Comparison in percentages between the results from EPANET and from our linear model



Levels	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9
l1	10	1	1	1	10	1	10	1	1
l2	1	13	1	1	13	1	1	13	13
l3	1	1	7	1	1	7	7	1	7
l4	1	1	1	10	1	10	1	10	1
Results									
Δq1	19.92	119.04	6.92	0.91	3.30	2.19	19.14	109.25	68.79
Δq2	680.91	10.84	125.54	106.24	3.56	39.11	4.52	6.20	12.42
Δq3	41.42	35.77	25.76	24.30	43.48	77.00	34.99	57.50	5386.11
Δq4	2.36	36.44	59.66	14.08	50.05	12.57	5.85	16.87	11.76
Δq1b	50.11	23.55	12.84	1.96	6.71	3.52	51.39	27.31	18.09
Δq2b	28.74	1.87	49.69	486.97	5.66	20.81	91.82	52.42	3.39
Δq3b	20.59	79.50	37.01	39.11	75.98	49.54	7.63	71.04	3.60
Δh1	25.87	13.47	7.82	5.68	2.59	6.20	20.84	16.20	12.94
Δh2	10.08	9.65	1.10	2.39	1.68	2.57	5.64	9.83	4.63
Δh3	5.14	19.48	0.11	19.69	20.89	2.87	1.37	23.86	2.24
Δh4	0.31	3.71	7.23	30.22	4.26	12.47	6.65	22.00	4.85
Δz5	38.85	21.82	8.09	4.99	1.72	5.64	30.96	25.28	20.84
Δz6	0.26	3.79	7.29	30.25	4.34	12.46	6.61	22.01	4.89
Δz7	2.16	2.00	9.94	31.07	2.69	11.90	9.22	22.10	6.98

Levels	Test 10	Test 11	Test 12	Test 13	Test 14	Test 15	Test 16	Test 17	Test 18	Test 19
l1	10	1	10	10	10	5	2	2	7	5
l2	13	13	1	13	13	13	1	2	2	5
l3	7	7	7	1	7	4	5	5	0	5
l4	1	10	10	10	10	2	6	10	8	5
Results										
Δq1	8.58	48.83	16.87	13.92	17.04	182.44	2.34	0.04	4.47	5.93
Δq2	2.26	8.28	1.32	29.59	18.16	0.18	62.52	27.38	332.53	46.66
Δq3	0.00	0.00	100.00	60.93	0.00	58.64	79.68	1513.00	32.71	66.35
Δq4	12.22	11.30	8.03	22.59	18.91	40.27	5.30	9.50	7.61	3.77
Δq1b	10.08	16.49	47.93	18.45	9.86	14.44	8.80	0.07	17.61	17.19
Δq2b	1.10	38.78	33.21	47.40	29.76	11.50	43.40	9.29	91.22	61.99
Δq3b	1.10	38.78	33.59	119.01	29.76	51.25	39.62	31.96	25.73	43.32
Δh1	4.29	12.74	18.34	2.12	4.36	1.42	4.22	5.64	14.39	4.40
Δh2	0.79	0.44	4.77	3.53	1.44	5.33	0.86	1.56	6.04	0.74
Δh3	4.27	2.82	5.05	22.13	4.61	13.40	2.10	10.62	14.51	0.13
Δh4	2.14	4.25	12.62	18.71	6.22	0.82	8.07	19.55	24.06	0.90
Δz5	3.70	19.99	27.06	1.11	3.87	3.73	2.74	5.05	22.93	3.59
Δz6	2.18	4.22	12.61	18.71	6.25	0.79	8.15	19.55	24.07	0.94
Δz7	4.01	5.49	12.07	18.58	7.49	0.85	7.00	19.47	24.37	2.58

<30%
>30% && <60%
>60% && <100%
>100%

We know that our results cannot be exactly the same as those calculated from a real hydraulic model, owing to the simplifications we introduced. Regarding the margins of error calculated, we conclude that they are acceptable for our needs. Moreover, the largest differences in percentages correspond to only small differences in absolute values.

We now have to test this linear model on a real network. The most significant concern is the likely increase in computation time as a result of the new equations added. It may be possible to address this problem by reducing the number of integer time steps, as shown in Sect. 3.2.

4 Solution Strategies

The model as it is formalized can provide the network manager with a pumping schedule for the day. But the efficiency and the speed of the search for a solution can still be improved, as we see in this section. First, we discuss the formulation of the objective function. Then, we describe the optimization parameters that can help to improve the efficiency of the search for a solution.

4.1 Two-Stage Optimization Strategy

The objective function is formulated as follows:

$$f = \sum_{t=1}^{H} \sum_{x} C_{x,t} f_{x,t} + \sum_{t=1}^{H} \sum_{y} K_y k_{y,t} \tag{14}$$

where parameter H is the time horizon (the number of time slots in the optimization). Parameter $C_{x,t}$ represents economic cost penalties, and variable $f_{x,t}$ represents the operation or non-operation of the corresponding element x during time slot t. Parameter $K_{y,t}$ is the penalty for not respecting constraint y; variable $k_{y,t}$ is different from 0 only if constraint y is not respected at time slot t.

One problem of this objective function is its heterogeneity. Penalties $C_{x,t}$ represent real economic costs, while penalties K_y are set arbitrarily to enforce the respect of some constraints if possible.

Introducing these two goals into the objective function (to ensure that operational constraints are met as far as possible while minimizing the economic cost) is not the most efficient way to proceed. It might even be counterproductive, in that it can lead to aberrations if the arbitrary penalty is not well chosen (for example, failure to respect a particular constraint, because this is a less expensive option, whereas optimization of economic cost should only begin once all operational constraints have been met).

A two-stage optimization might be an answer to this problem. First, we optimize taking into account only the constraint penalties. At the end of this step, we are able to set the maxim penalties allowed for the non-respect of each constraint. For example, we know the smallest deficit value for the storage tanks that will still allow us to reach a feasible solution.

Once these new constraints have been established, we can start to optimize in the second step, taking into account only the economic penalties.

We tested this approach with a time limit of 1,800 s (30 min). We compared a direct optimization with a two-stage resolution. After a few trials, we decided to devote the largest amount of available computation time to the first stage (1,790 s) and only 10 s to the second stage (Fig. 8).

The cost of the objective function is always less for direct optimization, but most of the time (except for five instances out of twelve) two-stage optimization better respects the constraints of the problem.

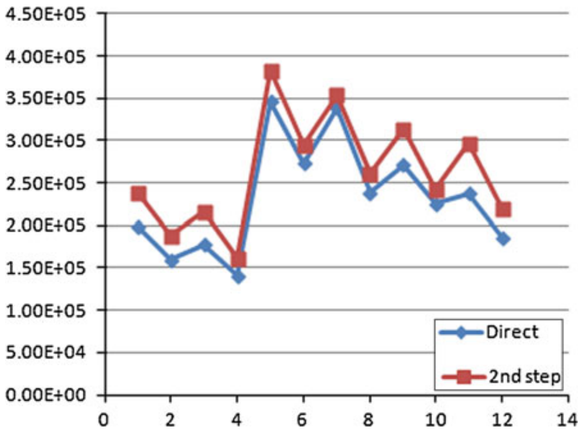
Our test showed us that the most difficult part of the optimization problem was emphatically not the economic aspect, but the respect of all constraints (as stated above, we had to set the first step of the optimization to 1,790 s, and the second step to only 10 s). Two-stage optimization is therefore relevant to our problem, but might be refined to obtain better results. The values of the penalties, in particular, have to be chosen very carefully.

4.2 Partially Continuous Relaxation Strategy

Another idea for reducing computation time is to reduce the number of integer time steps. As we are periodically recalculating the pump scheduling because consumption predictions are constantly being updated, it is possible to use integer time steps during an initial phase only, before switching to continuous time steps for the remainder of the solution process.

The correct schedule obtained in this fashion for the first time steps can be implemented directly. For subsequent time steps, we will have continuous values

Fig. 8 Comparison between economic costs of a direct or two-stage optimization



(somewhere between 0 and 1) for the pump operation. It will not be possible to use these values directly, but they will give us an idea of the volumes to be exchanged.

We tested the computation time for 20, 12, and 4 initial integer time steps, followed by continuous time steps (Table 2, Fig. 9).

Reducing the number of integer time steps considerably increases the efficiency of the solution process. For the test with only four integer time steps, we obtained an optimal solution in less than 10 s for each instance.

But we have to determine the smallest number of time steps necessary to guarantee the quality of the solution.

Table 2 Gap between best bound and best feasible solutions for different numbers of integer time steps

Instance	All integer time steps (24)				20 integer time steps			
	CPU	Solution	Bound	Gap (%)	CPU	Solution	Bound	Gap (%)
1	60	–	2.27E+06	–	60	2.17E+06	2.17E+06	0.04
2	60	–	2.23E+06	–	60	2.13E+06	2.13E+06	0.06
3	60	1.75E+06	8.15E+05	53.54	60	7.35E+05	7.34E+05	0.11
4	60	1.22E+07	7.75E+05	93.64	60	6.95E+05	6.93E+05	0.19
5	60	2.98E+06	7.76E+06	40.99	60	1.32E+06	1.32E+06	0.10
6	60	2.79E+06	1.93E+06	30.83	60	1.24E+06	1.24E+06	0.04
7	60	–	1.73E+06	–	60	1.29E+06	1.29E+06	0.13
8	60	–	1.89E+06	–	60	1.21E+06	1.21E+06	0.02
9	60	–	1.45E+06	–	60	9.79E+05	9.78E+05	0.03
10	60	–	1.39E+06	–	60	9.20E+05	9.19E+05	0.05
11	60	1.82E+07	1.41E+06	92.24	60	9.49E+05	9.48E+05	0.03
12	60	–	1.34E+06	–	60	8.88E+05	8.88E+05	0.02
12 integer time steps					4 integer time steps			
1	60	2.09E+06	2.09E+06	0.02	1.15	2.05E+06	2.05E+06	0.01
2	60	2.05E+06	2.05E+06	0.02	1.95	2.01E+06	2.01E+06	0.01
3	60	6.64E+05	6.63E+05	0.06	0.7	6.26E+05	6.26E+05	0.01
4	60	6.18E+05	6.18E+05	0.04	0.89	5.80E+05	5.79E+05	0.01
5	60	1.21E+06	1.21E+06	0.05	4.52	1.13E+06	1.13E+06	0.01
6	60	1.13E+06	1.13E+06	0.04	4.56	1.05E+06	1.05E+06	0.00
7	60	1.18E+06	1.18E+06	0.07	6.36	1.11E+06	1.11E+06	0.00
8	60	1.09E+06	1.09E+06	0.06	1.78	1.02E+06	1.02E+06	0.01
9	60	8.82E+05	8.82E+05	0.01	0.81	8.29E+05	8.29E+05	0.01
10	60	8.23E+05	8.23E+05	0.04	1.9	7.67E+05	7.67E+05	0.00
11	60	8.53E+05	8.53E+05	0.02	1.09	8.02E+05	8.02E+05	0.00
12	60	7.92E+05	7.92E+05	0.02	1.01	7.39E+05	7.39E+05	0.00

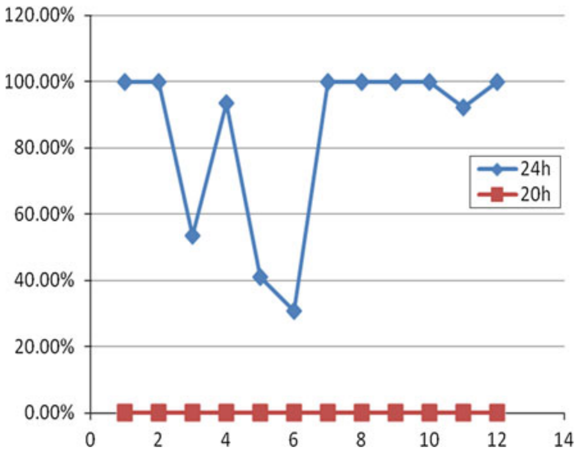


Fig. 9 Comparison of gap for 20 and 24 integer time steps

4.3 Enlarging the Horizon Time

Another problem is planning for the final time slots of the day. Some constraints are hard to take into account unless we have some knowledge concerning the following periods. For example, in the case of the constraint designed to avoid stagnation in the storage tanks, we need to know if consumption will be low or high just after the end of the optimization.

To overcome this difficulty, we planned to extend the optimization horizon: If the requested planning was over 24 h, the optimization would be over a period of 48 h so as to avoid the problem of non-visibility affecting the final periods. But extending the optimization horizon increases the computation time, which is a serious drawback in a real-time context. We therefore retained the principle while slightly modifying the model for the following 24 h in order to have no more than an overall vision of the volume and constraints present the following day, thus avoiding a large increase in computation time.

We tested the differences between a 24-hour and a 48-hour optimization. A computation time limit of 30 min was specified in advance, and we compared the final discrepancy between the feasible solution and the best bound found. The storage tanks were required to be full at the end of the optimization, and the first time slot was fixed at 6 a.m. The different instances tested had different initial water levels in the tanks and the tests concerned different periods of the year or week, which impacts the level of consumption and certain energy-related constraints (price and maximum authorized power) (Tables 3, 4).

The first column indicates the instance tested: the initial level in the tanks, the season, the consumption prediction used for the first 24 h, and the consumption prediction for the latest 24 h (because the consumption is different during the week and during the week end). There are more instances for the 48-hour optimization because there are a larger number of possibilities when we are dealing with two

Table 3 24-hour optimization

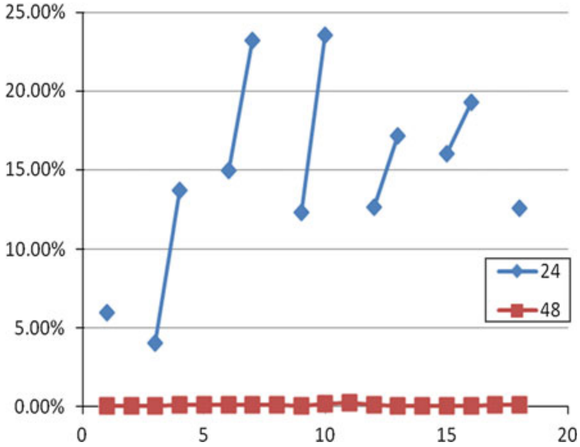
Characteristics	Solution	Best bound	Gap (%)
Full storages–summer–week/week	2.81E+06	2.81E+06	0.05
Full storages–summer–week/Sunday	2.62E+06	2.62E+06	0.05
Full storages–summer–Sunday/week	2.77E+06	2.77E+06	0.05
Full storages–winter–week/week	1.36E+06	1.36E+06	0.11
Full storages–winter–week/Sunday	1.30E+06	1.30E+06	0.12
Full storages–winter–Sunday/week	1.32E+06	1.32E+06	0.13
Low storages–summer–week/week	1.86E+06	1.86E+06	0.15
Low storages–summer–week/Sunday	1.63E+06	1.63E+06	0.13
Low storages–summer–Sunday/week	1.79E+06	1.79E+06	0.07
Low storages–winter–week/week	1.82E+06	1.82E+06	0.18
Low storages–winter–week/Sunday	1.76E+06	1.75E+06	0.24
Low storages–winter–Sunday/week	1.75E+06	1.75E+06	0.12
Half-full storages–summer–week/week	1.58E+06	1.58E+06	0.04
Half-full storages–summer–week/Sunday	1.36E+06	1.36E+06	0.07
Half-full storages–summer–Sunday/week	1.52E+06	1.52E+06	0.08
Half-full storages–winter–week/week	1.53E+06	1.53E+06	0.08
Half-full storages–winter–week/Sunday	1.47E+06	1.47E+06	0.09
Half-full storages–winter–Sunday/week	1.48E+06	1.47E+06	0.08

Table 4 48-hour optimization

Characteristics	Solution	Best bound	Gap (%)
Full storages–summer–week/week	2.41E+06	2.27E+06	5.98
Full storages–summer–week/Sunday			
Full storages–summer–Sunday/week	2.32E+06	2.23E+06	4.03
Full storages–winter–week/week	9.45E+06	8.16E+06	13.67
Full storages–winter–week/Sunday			
Full storages–winter–Sunday/week	9.11E+06	7.75E+06	14.98
Low storages–summer–week/week	2.31E+06	1.77E+06	23.22
Low storages–summer–week/Sunday			
Low storages–summer–Sunday/week	2.20E+06	1.93E+06	12.30
Low storages–winter–week/week	2.27E+06	1.73E+06	23.57
Low storages–winter–week/Sunday			
Low storages–winter–Sunday/week	2.16E+06	1.89E+06	12.64
Half-full storages–summer–week/week	1.76E+06	1.45E+06	17.18
Half-full storages–summer–week/Sunday			
Half-full storages–summer–Sunday/week	1.69E+06	1.42E+06	16.03
Half-full storages–winter–week/week	1.75E+06	1.41E+06	19.30
Half-full storages–winter–week/Sunday			
Half-full storages–winter–Sunday/week	1.54E+06	1.35E+06	12.55

separate days. The three other columns respectively show the final feasible solution obtained, the best bound, and the difference between these two values. These results are presented in the following graphic (Fig. 10):

Fig. 10 Comparison of gap for a 24-hour and a 48-hour optimizations concerning 18 different instances



We can see that the 48-hour optimization converges faster on the best bound than the 24-hour solution. This is encouraging and will enable us to reduce the 30-min limit on computation time.

5 Conclusions

The aim of this study is to build a realistic mathematical model giving rise to pertinent, feasible solutions, where computation time has been reduced as much as possible and the solutions are of a high quality.

The use of integer variables in our model takes into account the discrete behavior of pumps, and the solutions obtained are therefore easy to implement. We also take new constraints into account. The next important step for us will be to increase the relevance of our model by taking fully into account a number of hydraulic phenomena that can be observed in the network.

We succeeded in reducing computation time and the discrepancy between the best bound and feasible solutions by changing some parameters in our optimization. Our intention is now to refine our optimization techniques and to introduce new ways of reasoning with the aim of reducing the solution space and the computation time. We also plan to look at other optimization tools (such as constraint programming) that may be able to solve the same model more efficiently.

References

1. Guhl, F. (1999). *Gestion optimale des réseaux d'eau potable*. Ph.D. thesis, Université Louis Pasteur.
2. Ulanicki, B., Kahler, J., & See, H. (2007). Dynamic optimization approach for solving an optimal scheduling problem in water distribution systems. *Journal of Water Resources Planning and Management, 133*, 23–32.

3. Lopez-Ibanez, M., Prasad, T. D., & Paechter, B. (2008). Ant colony optimization for optimal control of pumps in water distribution networks. *Journal of Water Resources Planning and Management, 134*, 337–346.
4. Nicklow, J. (2010). State of the art for genetic algorithms and beyond in water resources planning and management. *Journal of Water Resources Planning and Management, 136*, 412–432.
5. Nace, D., Demotier, S., Carlier, J., Daguinos, T., & Kora, R. (2001). Using linear programming methods for optimizing the real-time pump scheduling. World Water and Environmental Resource Congress.
6. Wosley, L. A. (1998). *Integer linear programming*. New York: Wiley.
7. Pegg, S. (2001). An online optimised pump scheduling system. *Proceedings of the ORSNZ Conference Twenty Naught One*.