



Piecewise bounding and Integer Linear Programming for the optimal management of a water pumping and desalination process

Sandra Ulrich Ngueveu, Bruno Sareni, Xavier Roboam

► To cite this version:

Sandra Ulrich Ngueveu, Bruno Sareni, Xavier Roboam. Piecewise bounding and Integer Linear Programming for the optimal management of a water pumping and desalination process. 2014. hal-01163064

HAL Id: hal-01163064

<https://hal.science/hal-01163064>

Preprint submitted on 12 Jun 2015

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.

Piecewise bounding and Integer Linear Programming for the optimal management of a water pumping and desalination process

Sandra Ulrich Ngueveu^{2,3}, Bruno Sareni^{1,2} and Xavier Roboam^{1,2}

¹LAPLACE, UMR CNRS 5213, INPT, UPS, 2 rue Camichel, 31071 Toulouse, France

² Univ de Toulouse, INP, LAAS, LAPLACE, F-31400 Toulouse, France

³ CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse Cedex 4, France

ngueveu@laas.fr

02/05/2014

Abstract

The problem considered is the optimization of water production for autonomous water pumping and desalination units supplied by renewable energy sources, designed to be a viable solution to fresh water scarcity for remote areas. Non-linear gyrators as well as the non-linear efficiency of energy and flow transfers model the mechanical-hydraulic power conversion systems involved. This paper presents a generic formulation and resolution algorithms based on piece-wise bounding and integer linear programming to solve to optimality the global optimization problem of finding an optimal energy management strategy.

1 Problem statement

The system studied corresponds to an architecture previously designed by Roboam *et al* [9] and illustrated on figure 1. It is composed of motor pumps, desalination process and hydraulic network (pipes and valves). The first pump draws from the groundwater and pours salt water into the first tank. The second pump extracts salt water from the first tank and is coupled with a desalination subsystem (Reverse Osmosis module) so that only filtered drinking water is poured in tank 2, whereas the rejected water, which has a very high salt concentration, is discarded. The third pump draws from the tank 2 to fill the third and final tank at the top of the water tower with clean water available for consumption. For the remainder of the paper, the intake tank and discharge tank of a given pump will refer to the tank from which the pump draws water and the one into which the water is poured, respectively. All tanks are identical, with a height l_{\max} and a base area of $hq = 1m^2$.

The power p required by a pump to pull a quantity of water q is function of the pump characteristics (fixed parameters available from the manufacturer), but also depends on the level of water l in its intake tank. Indeed, following Pascal's law (physics), the higher the

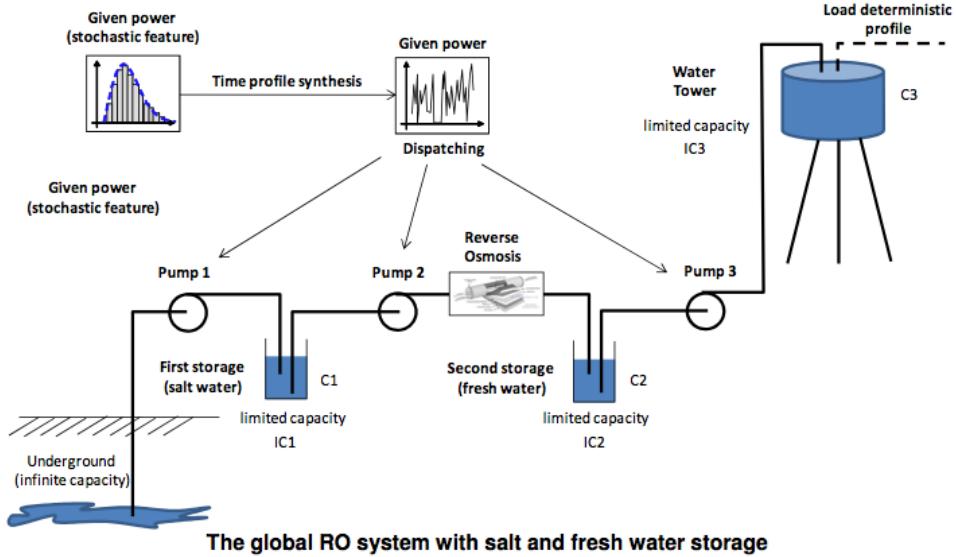


Figure 1: The global RO system with salt and fresh water storage (Source: B. Sareni)

height difference between the intake point and the discharge point, the higher the pressure difference, and thus the more suction power is needed. The discharge point of each pump is fixed, but the intake point depends on the water level, which can vary from the minimum water level l_{\min} from which the pump can work to the maximal tank height l_{\max} . Only the first pump does not take into account water height variation, as it pumps from the groundwater whose level is not affected by the amount of water pumped. The second pump is connected to a Reverse Osmosis (RO) desalination module. It is modeled as a full hydraulic system in which a membrane with a given conductivity splits the input water flow into two flows, the fresh desalinated water flow $q_{\text{permeate}} = q_{\text{totank}(2)}$ and the rejected water $q_{\text{concentrate}}$ which has a very high concentration of salt. Details on the system characteristics as well as the resulting expressions of the electrical power from the three pumps and the RO module are available in [9] and recalled in appendix. It should be noted that presence of RO module and the absence of groundwater level variation mean that even if identical pumps were used, three different flow transfer functions would be obtained.

In summary, the main subsystems (pump 1, pump 3 and the combination pump 2 + RO module) are modeled by non linear transfer functions as follows:

$$p^{\text{pump1}} = f^1(q_{\text{totank}(1)}) = f^{1\&3}(q_{\text{totank}(1)}, 0) \quad (1)$$

$$p^{\text{pump2+ROmodule}} = f^2(q_{\text{concentrate}}, l_{\text{tank}(1)}) = f^2(q_{\text{fromtank}(1)} - q_{\text{totank}(2)}, l_{\text{tank}(1)}) \quad (2)$$

$$p^{\text{pump3}} = f^3(q_{\text{totank}(3)}, l_{\text{tank}(2)}) = f^{1\&3}(q_{\text{totank}(3)}, l_{\text{tank}(2)}) \quad (3)$$

$$q_{\text{fromtank}(1)} = q_{\text{permeate}} + q_{\text{concentrate}} = f^4(q_{\text{concentrate}}) \quad (4)$$

where $f^i, \forall i \in \{1, 2, 3, 4\}$ are four different non linear functions whose analytical expressions are explicated in appendix 8. An illustration of f^3 is provided on figure 2.

The objective was to minimize the total time necessary to fill all the three tanks given an input wind power profile.

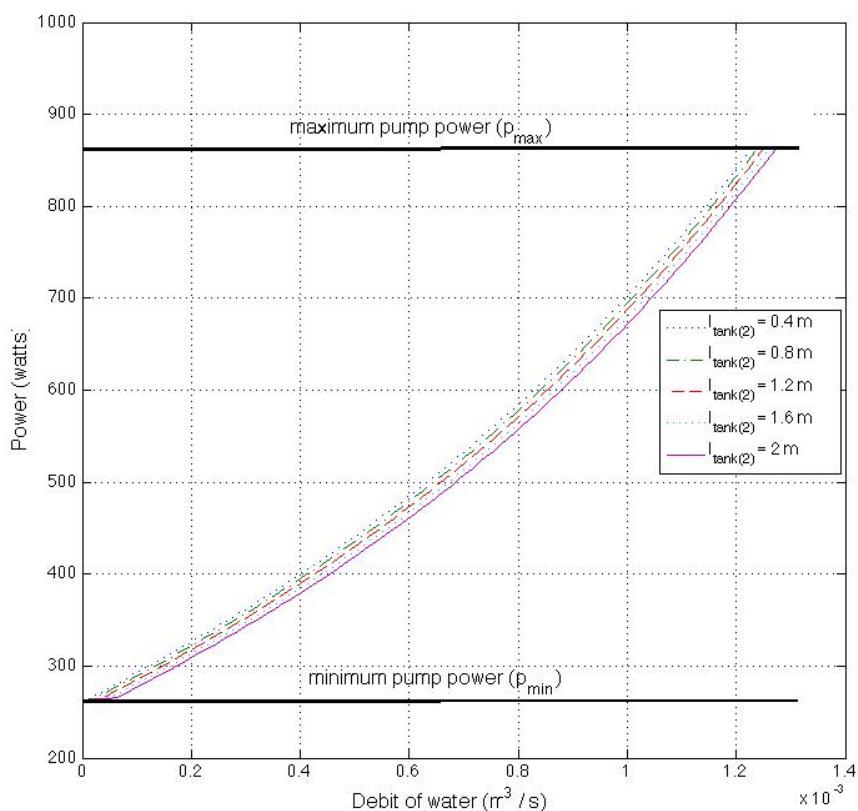


Figure 2: Non linear transfer function related to pump 3

2 Literature review

The problem considered was introduced in [9]. The authors first presented the dynamic model of the system, based on an equivalent hydraulic circuit and its corresponding Bond Graph. Then they introduced a static model of the system behavior with non-linear equations of the conversion of the electrical power consumed and state of the system into water flow. They showed that it is sufficient to focus on the static model when trying to find an efficient water management strategy. Then they designed and compared different online heuristics, based on the local optimization, at each instant, of a local non-linear objective function. Different local objective functions were compared, corresponding to a maximization of respectively the system efficiency, the total flow of water transferred, the total flow weighted by the tank water level, quadratic flows and the output hydraulic powers. Tested on three different waveforms of input powers, they showed that the best results were obtained from locally optimizing the output hydraulic powers. Finally, the authors compared the results obtained using different pump characteristics from a pump manufacturer database, to highlight the importance of the pump sizing during the conception of such autonomous process: less powerful pumps have the obvious disadvantage of being limited in the water flow they can generate at each instant, but more powerful pumps present the disadvantage of requiring a higher water level from their intake tank to be able to pump (higher l_{\min} values). Hence the importance of pump sizing when implementing the system in a specific region subject to a particular wind profile. No global optimization method was proposed, and the efficiency of the online heuristic in comparison to the global optimum was not evaluated.

In the literature, a few promising mathematical programming-based approaches on similar problems can be found [3, 4], either based on MINLP or transformations into approximated MILP. A detailed review of MINLP techniques has been proposed by Grossman [7]. Let us briefly recall the basic elements of the most used algorithms to help positioning the scientific contribution of our work. Classical resolution methods are branch-and-bound (BB), Outer-Approximation (OA), LP/NLP based branch-and-bound (LPNLP), Generalized Benders Decomposition (GBD) and Extended Cutting Plane Method (ECP). These methods differ on the way they generate and use subproblems. There are four basic subproblems that have been considered for solving MINLPs: three non-linear subproblems (NLP with no more integer/binary variables) and a MILP cutting plane (MILPCP). The MILPCP is obtained by replacing the nonlinear convex functions with supporting hyperplanes or Outer Approximation cuts at some chosen boundary points (usually derived at the solution of an NLP subproblem). New points are obtained by identifying violated inequalities. The solution of a MILPCP subproblem yields a valid lower bound for the MINLP problem. This bound is nondecreasing with the number of linearization points. Based on these components, BB for MINLP is a branch-and-bound solves at each node either an LP relaxation or an NLP. OA is an iterative procedure that alternates between the resolution of a MILPCP to obtain a lower bound and the resolution of a NLP to obtain upper an upper bound. The NLP is obtained by fixing the value of integer variables of the original problem to their values in the previous MILPCP solution. The NLP solution and the corresponding supporting hyperplane is then added to the MILPCP before the next iteration. The algorithm ends when the two bounds assume the same value (within a fixed tolerance). GBD can be regarded as a particular case of the OA algorithm where in the MILPCP the set of hyperplanes considered is restricted to

the active inequalities (the others are removed). ECP is based on an iterative resolution of a MILPCP subproblem and given the optimal solution of the MILPCP which can be infeasible for MINLP, the determination of the most violated inequalities whose linearization are added at the next MILPCP. LPNLP can be seen as an extention of branch-and-Cut to convex MINLPs. The idea is to solve a BB on MILPCP, but everytime an integer feasible solution is found, the corresponding NLP subproblem is solved (with the same fixed variables values) to generate new cuts and local enumerations are performed at some nodes of the search tree. It is equivalent to applying OA at some nodes of the search trees. Recently hybrid algorithms and frameworks have been proposed, usually based on the same key components, but able to switch between the different algorithms during the resolution of a problem, to take advantage of the strengths of each approach [2], [6], [8].

Classical global optimization methods (OA, BB,...) are able to find globally optimal solutions of MINLP problems, but may require high computing times and present scalability challenges. The MILP-based approaches with classical piecewise linear approximations have been presented as a practical and theoretically sound alternative for solving some problems but the presence of approximation errors may result into the loss of either the guarantee of global optimality of the solution obtained, or the guarantee of feasibility of such solution. Furthermore, it requires the introduction of additional integer variables, and yet there is no guarantee that a lower discretization level (leading to more variables) would not have led to a better solution. In the light of the foregoing, as we will describe in next Section, we propose an alternative approach based on piecewise linear lower and upper bounding, rather than approximating, the non linear efficiency functions. Related work in scheduling problems involving non-linear efficiency functions can also be found in the field of scheduling with continuous resources [1], mainly associated with parallel scheduling applications and considering theoretical complexity studies of remarkable special cases. In contrast with these studies, we aim at rather proposing a resolution scheme to solve (relatively) more general problems.

3 Resolution scheme proposed

Using the non linear efficiency functions f^i from equations (1)-(3), it is possible to model the resulting problem as a MINLP with discrete variables on the state of the pumps (on or off) or tanks (full or not) and continuous variables measuring the water flows as well as the levels of water in the tanks.

The resolution scheme proposed is based on two main ideas: (i) the piece-wise bounding of the nonlinear energy efficiency function, then (ii) the reformulation of the problem, which originally is a mixed integer non linear problem (MINLP), into two mixed integer linear problems (MILP) using the pair of bounding functions previously defined. The piece-wise bounding of a function f of m variables within a tolerance value ϵ consists in identifying two piece-wise linear functions denoted \bar{f}^ϵ and \underline{f}^ϵ that verify equations (5) to (7). The two MILP, denoted MILP and MILP respectively, are obtained by substituting f with \bar{f}^ϵ and \underline{f}^ϵ ,

respectively.

$$\underline{f}^\epsilon(x) \leq f(x) \leq \bar{f}^\epsilon(x), \quad \forall x \in \mathbb{R}^m \quad (5)$$

$$f(x) - \underline{f}^\epsilon(x) \leq \epsilon f(x), \quad \forall x \in \mathbb{R}^m \quad (6)$$

$$\bar{f}^\epsilon(x) - f(x) \leq \epsilon f(x), \quad \forall x \in \mathbb{R}^m \quad (7)$$

Performing the linearizations before the optimization allows not only to generate \bar{f}^ϵ and \underline{f}^ϵ with respect to a predefined tolerance value ϵ , but also to minimize the number of sectors of the resulting piecewise functions and therefore to minimize the number of additional integer variables in $\overline{\text{MILP}}$ and $\underline{\text{MILP}}$. Solving a $\overline{\text{MILP}}$ generates solutions that are feasible for the original MINLP, and that have a total cost less than $\epsilon\%$ higher than the optimal solution cost. Solving a $\underline{\text{MILP}}$ generates solutions that may not be feasible for the original MINLP, but whose total cost is less than $\epsilon\%$ lower than the optimal solution cost. Note that if feasible, the solution of $\underline{\text{MILP}}$ is also optimal. Theorem 3.1 states the existence of ϵ values leading to optimal solutions of the original problem. Even if not feasible, the solution of $\underline{\text{MILP}}$ can still help proving the global optimality of the solution of $\overline{\text{MILP}}$ if both have the same cost, or evaluate the gap to optimality if not. Also note that both problems share the same structure and only differ in terms of the numerical data of their respective piecewise functions. Therefore, either a MILP black-box solver is used, or a single dedicated resolution method needs to be developed and applied to solve both problems.

Theorem 3.1 $\exists \epsilon^*$ such that $\forall f^\epsilon$, the optimal solution of the corresponding $\underline{\text{MILP}}$ is the global optimal solution of the original MINLP.

Proof The proof is based on two properties. (i) The value of the solution of $\underline{\text{MILP}}$ does not decrease with the decrease of ϵ . (ii) The theorem of Duran and Grossman [5] which is the basis of the OA algorithm states that if all feasible discrete variables are used as linearization points then the resulting MILPCP problem (refered to in [7] as M-OA) has the same optimal solution than the original MINLP. Let ϵ^+ be the maximum relative error between M-OA and the original MINLP. From (i) and (ii), it can be deduced that any solution of $\underline{\text{MILP}}$ obtained with $\epsilon \leq \epsilon^+$ has a cost higher or equal to the optimal solution cost of M-OA, and therefore equal to the optimal cost of the original MINLP.

To summarize, the use of piecewise linear bounding as we propose allows the application of combinatorial optimization tools and techniques only, and yet the obtention and certification of the global optimal solution of the non linear problem. The resulting approach differs from the usual generic MINLP resolution methods (BB, OA, ...) because the linearizations are performed before the start of the optimization, the definition of two MILP problems is never modified and no non-linear subproblem resolution is performed at any time. This approach also differs from the MILP-based reformulations proposed in the literature for similar problems ([4],[3]) because the piecewise linear bounding of a function differs from its piecewise linear approximation. Indeed, piece-wise bounding as expressed with equations (5) to (7) imposes that each of the two functions generated remains in its half-space delimited by the nonlinear function. Such constraint is not necessarily imposed when performing linear approximation, as illustrated on figure 3. In addition, we explicitly consider the two bounding functions whereas in the literature the resulting approximation function may correspond to none or one of them coincidentally. Also note that neither \bar{f}^ϵ or \underline{f}^ϵ is required to be continuous.

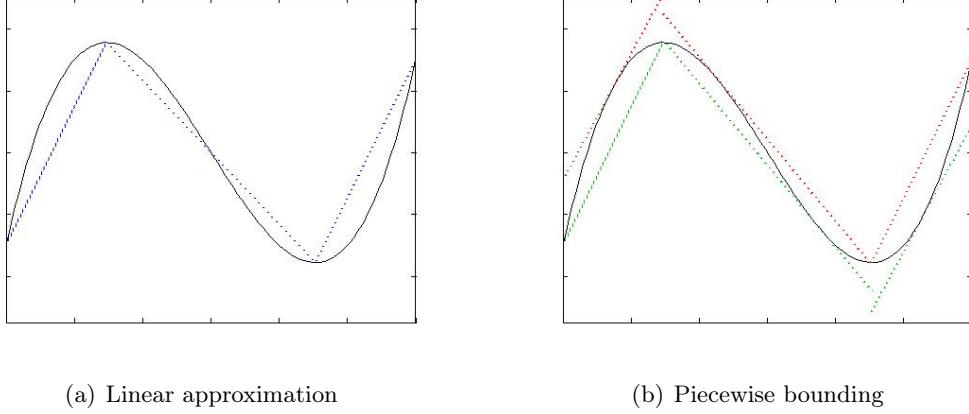


Figure 3: Difference between piecewise linear approximation and piecewise linear bounding

4 Piecewise linear bounding heuristics

Bounding functions \bar{f}^ϵ and f^ϵ must be generated with respect to a predefined tolerance value ϵ , but their number of sectors must be minimized to ensure the minimization of number of additional integer variables in MILP and MILP. We proposed a fast yet efficient heuristic to that end for the nonlinear efficiency functions from the water production optimization problem considered. The heuristic takes advantage of the convexity of the transfer functions. Note that if the functions were concave instead, the heuristic would remain applicable with a simple permutation of the upper and lower bounding procedures.

4.1 Piecewise linear bounding of the transfer function of Pump 1

4.1.1 Generation of lower bounding linear sectors

Each sector i of the linear bounding function f_ϵ verifies a linear equation in the form $p = a_i q + b_i$ and is therefore defined by four parameters: the slope a_i , the y-intercept b_i and the limits q_{\min_i} and q_{\max_i} . The goal of the heuristic is to identify the minimum number of sectors n , and the parameters of each sector. Each pair of consecutive sectors $i - 1$ and i satisfies $q_{\min_i} = q_{\max_{i-1}}$ with q_{\min_1} verifying $p_{\min} = f^1(q_{\min_1})$ and q_{\max_n} verifying $p_{\max} = f^1(q_{\max_n})$. The heuristic we designed proceeds iteratively, from the first sector to the last. One challenge of the sectorization is to ensure respect of constraints (6). The solution proposed was to define each sector using a supporting linear function tangent to f^1 at a predefined point, ensuring in this way that the maximum error would be located either at the extremities of the sector, or at the predefined tangent point of the sector, facilitating its computation.

Let us consider sector i with its origin q_{\min_i} . Let \tilde{q} be a potential tangent point. The resulting slope and y-intercept would be $a_i = \frac{df^1}{dq}(\tilde{q})$, $b_i = f^1(\tilde{q}) - \tilde{q} \frac{df^1}{dq}(\tilde{q})$. Because of the convexity of the function, the error between the sector and the real curve grows as the distance $|x - \tilde{q}|$ increases. Therefore, the maximum errors would be located at the extremities of sector i .

If $f(q_{\min_i}) - a_i q_{\min_i} - b_i \leq \epsilon f(q_{\min_i})$, then constraint (6) is verified for each $q \in [q_{\min_i}, \tilde{q}]$.

The corresponding a_i and b_i values can be kept. Only the upper limit q_{\max_i} of the sector i remains to be identified. The goal is to maximize q_{\max_i} subject to constraint (6). The heuristic procedure starts by setting $q_{\max_i} = q_{\max}$ and then reduces the value by a parameter θ until the constraint (6) is verified by q_{\max_i} and therefore by all $q \in [\tilde{q}, q_{\max_i}]$. If $f(q_{\min_i}) - a_i q_{\min_i} - b_i > \epsilon f(q_{\min_i})$, it means that the value \tilde{q} had been chosen too far from the origin of the sector q_{\min_i} . Therefore its value should be reduced and the slope and y-intercept updated accordingly. The complete heuristic H_{f^1} is summarized by Algorithm 1.

4.1.2 Generation of upper bounding linear sectors

The upper bounding procedure for pump1 $H_{\overline{f^1}}$ is similar to H_{f^1} , except this time the resulting y-intercept is $b_i = f^1(q_{\min_i}) - q_{\min_i} \frac{df^1}{dq}(\tilde{q})$ ensuring that the resulting line crosses the origin of the sector q_{\min_i} . In this case, the maximum error for every $q \in [q_{\min_i}, \tilde{q}]$, is obtained at the point \tilde{q} . Therefore, the value of \tilde{q} is to be decreased until constraint (6) is verified by \tilde{q} . The resulting a_i and $b_i = f^1(q_{\min_i}) - q_{\min_i} \frac{df^1}{dq}(\tilde{q})$ are kept and only q_{\max_i} remains to be computed. The procedure starts by setting $q_{\max_i} = q_{\max}$ and then reduces the value by a parameter θ until $f(q_{\max_i}) - a_i q_{\max_i} - b_i \leq 0$ is verified. The complete heuristic $H_{\overline{f^1}}$ is summarized by Algorithm 2.

Algorithm 1 H_{f^1}

```

1:  $i := 0$ ;  $q_{\min_0} = q_{\min}$ ;  $\theta = 0.9$ 
2: repeat
3:    $i := i + 1$ 
4:    $q_{\min_i} = q_{\min_{i-1}}$ 
5:    $\tilde{q} = q_{\max}$ 
6:    $a_i := \frac{df^1}{dq}(\tilde{q})$ ;  $b_i := f^1(\tilde{q}) - \tilde{q} \frac{df^1}{dq}(\tilde{q})$ 
7:   while  $f(q_{\min_i}) - a_i q_{\min_i} - b_i > \epsilon f(q_{\max_i})$  do
8:     if  $\tilde{q} \geq \frac{q_{\min_i}}{\theta}$  then
9:        $\tilde{q} := q_{\min_i} + \theta(\tilde{q} - q_{\min_i})$ 
10:    else
11:       $\tilde{q} := \theta\tilde{q}$ 
12:    end if
13:     $a_i := \frac{df^1}{dq}(\tilde{q})$ ;  $b_i := f^1(\tilde{q}) - \tilde{q} \frac{df^1}{dq}(\tilde{q})$ 
14:   end while
15:    $q_{\max_i} = q_{\max}$ 
16:   while  $f(q_{\max_i}) - a_i q_{\max_i} - b_i > \epsilon f(q_{\max_i})$  do
17:     if  $q_{\max_i} \geq \frac{\tilde{q}}{\theta}$  then
18:        $q_{\max_i} := \tilde{q} + \theta(q_{\max_i} - \tilde{q})$ 
19:     else
20:        $q_{\max_i} := \theta q_{\max_i}$ 
21:     end if
22:   end while
23: until  $q_{\max_i} = q_{\max}$ 
24:  $n_{\text{sectors}} := i$ 

```

Algorithm 2 $H_{\overline{f^1}}$

```

1:  $i := 0$ ;  $q_{\min_0} = q_{\min}$ ;  $\theta = 0.9$ 
2: repeat
3:    $i := i + 1$ 
4:    $q_{\min_i} = q_{\min_{i-1}}$ 
5:    $\tilde{q} = q_{\max}$ 
6:    $a_i := \frac{df^1}{dq}(\tilde{q})$ ;  $b_i := f^1(q_{\min_i}) - q_{\min_i} \frac{df^1}{dq}(\tilde{q})$ 
7:   while  $a_i q_{\min_i} + b_i - f(\tilde{q}) > \epsilon f(\tilde{q})$  do
8:     if  $\tilde{q} \geq \frac{q_{\min_i}}{\theta}$  then
9:        $\tilde{q} := q_{\min_i} + \theta(\tilde{q} - q_{\min_i})$ 
10:    else
11:       $\tilde{q} := \theta\tilde{q}$ 
12:    end if
13:     $a_i := \frac{df^1}{dq}(\tilde{q})$ ;  $b_i := f^1(q_{\min_i}) - q_{\min_i} \frac{df^1}{dq}(\tilde{q})$ 
14:   end while
15:    $q_{\max_i} = q_{\max}$ 
16:   while  $f(q_{\max_i}) - a_i q_{\max_i} - b_i > 0$  do
17:     if  $q_{\max_i} \geq \frac{\tilde{q}}{\theta}$  then
18:        $q_{\max_i} := \tilde{q} + \theta(q_{\max_i} - \tilde{q})$ 
19:     else
20:        $q_{\max_i} := \theta q_{\max_i}$ 
21:     end if
22:   end while
23: until  $q_{\max_i} = q_{\max}$ 
24:  $n_{\text{sectors}} := i$ 

```

4.2 Extensions to the transfer functions related to Pump 3 and 'pump 2 + RO'

The transfer function f^3 related to pump 3 varies from the one of pump 1 because it includes the water level variation from its the intake tank. Therefore it is a function of two variables

instead of one. To maintain the linearity of the bounding function, this variation was taken into account with a linear correction term in function of water level in the tank, which results into piecewise bounding functions in the form $p = aq + b - sv$ where v is the variable of the water level and s is the correction parameter. The idea of using a linear correction term is not new, as it has already been applied by [3] to enhance the approximation of the power production in function of a Hydro Scheduling and Unit Commitment problem. The novelty of our work is to ensure that constraints (5)-(7) remain verified. To that end, for \bar{f}^ϵ the reference function is $f^3(:, l_{\min})$ and the correction term s must never lead to overestimation of the power saving resulting from water level v ; whereas for \underline{f}^ϵ the reference function is $f^3(:, l_{\max})$ and the correction term s must never lead to underestimation of the power saving resulting from water level v .

The bounding functions for f^2 also require the usage of a correction term to account for the level of water in intake tank 1. Note that it is possible to consider and bound separately the transfer function related to pump 2 and the one related to the RO module, especially if pump 2 and 3 are identical. However, considering a unique global transfer function contributes to the minimization of the total number of sectors, and also ensures that the complete subsystem 'pump 2 + RO' respects the tolerance level ϵ , instead of $2 * \epsilon$ if the functions are separated. Finally the bounding functions for f^4 are obtained by simply replacing f^1 by f^4 in heuristics $H_{\bar{f}^1}$ and $H_{\underline{f}^1}$.

4.3 Piecewise linear bounding output

The parameters of the piecewise linear functions computed by the heuristics, can be summarized as follows.

- $n_{p_1}, n_{p_2}, n_{p_3}$: number of sectors of the piece-wise power functions of pump1, "pump2+filter" and pump3 respectively
- $a_i^j, b_i^j, \forall i \in 1..3, \forall j \in 1..n_{p_i}$: coefficients (slope and y-intercept) of the piece-wise power functions of pump i found
- $s_i^j, \forall i \in 2..3, \forall j \in 1..n_{p_i}$: coefficient for the surplus of pump i ($s_1^j = 0, \forall j \in 1..n_{p_1}$)
- $\alpha^j, \beta^j, \forall j \in 1..n_{p_2}$: coefficients (slope and y-intercept) of the piece-wise functions of the total flow of water ($Q_{\text{permeate}} + Q_{\text{concentrate}}$) versus the flow of rejected water ($Q_{\text{concentrate}}$) exiting from the "pump 2 + RO filter module" system
- $Q_{\min}^{i,j}, \forall i \in 1..3, \forall j \in 1..n_{p_i}$: starting breakpoint of the j^{th} sector of the piece-wise power function of pump i
- $Q_{\max}^{i,j}, \forall i \in 1..3, \forall j \in 1..n_{p_i}$: ending breakpoint of the j^{th} sector of the piece-wise power function of pump i

For the remainder of the paper, an upper bar will be added to refer specifically to the upper limit of the parameter (e.g $\overline{n_{p_1}}$) whereas an underline will be added to refer specifically to its lower limit (e.g $\underline{n_{p_1}}$). The absence of bar will refer to both (e.g n_{p_1}).

5 Integer linear programming reformulation of the energy management problem

Data

- \mathbb{N}_I : set of time intervals
- \mathbb{N}_P : set of pumps
- \mathbb{N}_T : set of tanks
- ts: scale of time, duration of the time intervals
- hq: section of the tanks, used to convert the debit into water level
- $P_{\min}^i, P_{\max}^i, \forall i \in \mathbb{N}_P$: pumping power limits of pump i
- $L_{\min}^i, L_{\max}^i, \forall i \in \mathbb{N}_P$: capacity limits of tank i
- $l_{\text{init}}^j, j \in \mathbb{N}_T, \geq 0$: initial water level of tank j
- $P_{\text{in}}^i, \forall i \in \mathbb{N}$: input power available at time interval i
- piece-wise functions data computed in Section 4: $\{n_{p_i}, a_i^j, b_i^j, s_i^j, \alpha^j, \beta^j, Q_{\min}^{i,j}, Q_{\max}^{i,j}\}, \forall i \in \mathbb{N}_P, \forall j \in 1..n_{p_i}$

Binary variables

- $r_i, \forall i \in \mathbb{N}_I$: binary variable equal to 0 if all tanks are full at time interval i , and 1 otherwise
- $sect_i^{j,k}, \forall i \in \mathbb{N}_I, \forall j \in 1..n_{p_1}, \forall k \in 1..3$: binary variable equal to 1 if pump k is used at the j^{th} section of the piece-wise power function during time interval i

Continuous variables

- $q_i^{j,k}, \forall i \in \mathbb{N}_I, \forall j \in 1..n_{p_1}, \geq 0$: continuous variable equal to the flow of water pumped by pump k at time interval i if it is used at the j^{th} section of the piece-wise power function and 0 otherwise. Note that for pump it corresponds to $Q_{\text{concentrate}}$.
- $l_i^j, \forall i \in \mathbb{N}_I, \forall j \in \mathbb{N}_T, \geq 0$: continuous variable that specifies the level of water going in tank j at time interval i
- $v_i^{j,k}, \forall i \in \mathbb{N}_I, \forall j \in 1..n_{p_2}, \geq 0$: continuous variable equal to the level of water in tank k if pump $k+1$ is used at the j^{th} section of the piece-wise power function at time interval i and 0 otherwise.

Mathematical formulation proposed:

$$\min \sum_{i \in \mathbb{N}} r_i * \text{ts} \quad (8)$$

subject to

$$r_0 = 1 \quad (9)$$

$$l_0^j = l_{\text{init}}^j, \quad \forall j \in \mathbb{N}_T \quad (10)$$

$$r_i - r_{i-1} \leq 0, \quad \forall i \in \mathbb{N}_I \quad (11)$$

$$\sum_{j \in \mathbb{N}_T} l_i^j + (\sum_{j \in \mathbb{N}_T} L_{\max}^j) r_i \geq \sum_{j \in \mathbb{N}_T} L_{\max}^j, \quad \forall i \in \mathbb{N}_I \quad (12)$$

$$\sum_{k \in \mathbb{N}_P} \sum_{j \in 1..n_{p_k}} (a_k^j q_i^{j,k} + b_k^j \text{sect}_i^{j,k} - s_k^j v_i^{j,k}) \leq \text{Pin}_i, \quad \forall i \in \mathbb{N}_I \quad (13)$$

$$l_i^1 - l_{i-1}^1 - \sum_{j \in 1..n_{p_1}} \text{hq} * \text{ts} * q_{j,1}^i + \sum_{j \in 1..n_{p_2}} \text{hq} * \text{ts} * (\alpha^j q_{j,2}^i + \beta^j \text{sect}_i^{j,2}) \leq 0, \quad \forall i \in \mathbb{N}_I \quad (14)$$

$$l_i^2 - l_{i-1}^2 - \sum_{j \in 1..n_{p_2}} \text{hq} * \text{ts} * ((\alpha^j - 1) q_{j,2}^i + \beta^j \text{sect}_i^{j,2}) + \sum_{j \in 1..n_{p_3}} \text{hq} * \text{ts} * q_{j,3}^i \leq 0, \quad \forall i \in \mathbb{N}_I \quad (15)$$

$$l_i^3 - l_{i-1}^3 - \sum_{j \in 1..n_{p_3}} \text{hq} * \text{ts} * q_{j,3}^i \leq 0, \quad \forall i \in \mathbb{N}_I \quad (16)$$

$$L_{\min}^k \leq l_i^k \leq L_{\max}^k, \quad \forall i \in \mathbb{N}_I, k \in \mathbb{N}_P \quad (17)$$

$$P_{\min}^k \leq \sum_{k \in N_p \setminus \{1\}} \sum_{j \in 1..n_{p_k}} (a_k^j q_i^{j,k} + b_k^j \text{sect}_i^{j,k} - s_k^j v_i^{j,k}) \leq P_{\min}^k, \quad \forall i \in \mathbb{N}_I, k \in \mathbb{N}_P \quad (18)$$

$$Q_{\min}^{i,j} \text{sect}_i^{j,k} \leq q_i^{j,k} \leq Q_{\max}^{i,j} \text{sect}_i^{j,k}, \quad \forall i \in \mathbb{N}_I, k \in \mathbb{N}_P, j \in 1..n_{p_k} \quad (19)$$

$$\sum_{j \in 1..n_{p_k}} \text{sect}_i^{j,k} \leq 1, \quad \forall i \in \mathbb{N}_I, k \in \mathbb{N}_P \quad (20)$$

$$v_i^{j,k} - l_i^j \leq 0, \quad \forall k \in \mathbb{N}_P, i \in \mathbb{N}_I, j \in 1..n_{p_k} \quad (21)$$

$$v_i^{j,k} - L_{\max}^j \text{sect}_i^{j,k} \leq 0, \quad \forall k \in \mathbb{N}_P, i \in \mathbb{N}_I, j \in 1..n_{p_k} \quad (22)$$

The objective function (8) minimizes the total finishing time. Constraints (9) states that all tanks are not full at time interval 0. Constraints (10) initialize the level of water in each tank at time interval 0. Constraints (11) ensure that once all tanks have been full, they remain full. Constraints (12) enforces r_i equal to 1 if all tanks are not yet full at time interval i . Constraints (13) ensure that the total power consumed by all pumps do not exceed the total power available at time interval i . Constraints (14) to (16) compute the level of water in tanks 1 to 3 respectively, in function of the level of water in the tanks at the previous instant and the debit of pump upstream and downstream of each tank. Constraints (17) and (18) enforce the bounds on the levels of water and the power of the pumps. Constraints (19) ensure that the debit of water from each pump is within the bounds of the sector in which each pump has been activated. Constraints (20) limit to at most one the number of sectors that can be activated for each pump. Constraints (21) connect variables $v_i^{j,k}$ with l_i^j and $p_i^{j,k}$. Note that the increase of the values of variables $v_i^{j,k}$ goes in the direction of the optimization (because the higher the values, the lower the power required), therefore only upper bounds of $v_i^{j,k}$ were considered in Constraints (21), not lower bounds.

6 Computational results

The heuristics and resulting MILP were implemented and tested on a wind profile that exceeds the total maximum power of the pumps. The results obtained are summarized by the

Table 1 summarizes the piecewise bounding outputs. It can be seen the total number of sectors and resulting bounding functions differ significantly for the three main transfer functions. The comparison between \underline{n}_{p_1} , \overline{n}_{p_1} and \underline{n}_{p_3} , \overline{n}_{p_3} highlights the influence of the water height variation, which has more impact on the upper bounding than on the lower bounding function, certainly because of the convexity of the transfer function. The comparison between \underline{n}_{p_2} , \overline{n}_{p_2} and \underline{n}_{p_3} , \overline{n}_{p_3} highlights the impact of the RO module (filter), and show a significant increase of the non linearity. It is interesting to note that in this case only is the lower bounding function more segmented than the upper bounding function.

	Pump 1		Pump 2 (+ filter)		Pump 3	
tolerance ϵ	\overline{n}_{p_1}	\underline{n}_{p_1}	\overline{n}_{p_2}	\underline{n}_{p_2}	\overline{n}_{p_3}	\underline{n}_{p_3}
5%	2	2	11	21	3	2
1%	5	5	21	29	8	5
0.5%	8	7	35	62	13	7
0.3%	10	9	43	74	17	9

Table 1: Number of sectors per tolerance value

Table 2 shows the results obtained on the water production optimization problem in function of the tolerance value ϵ . For reference, note that the solution obtained by the step-by-step online heuristic proposed by [9] obtained a solution cost of 22100 seconds. UB is the optimal solution value (in seconds) of $\overline{\text{MILP}}$ whereas LB is the optimal solution value of $\underline{\text{MILP}}$. The gap is computed as the ratio $\frac{\text{UB} - \text{LB}}{\text{LB}}$. Note that its value is lower than the tolerance value. The feasibility of the solutions obtained is always verified with regard to the original MINLP. The solution of $\overline{\text{MILP}}$ is of course always feasible. UB* is the optimal solution value of $\underline{\text{MILP}}$ if the corresponding solution happens to be feasible for the original MINLP.

tolerance ϵ	UB	s	LB	s	Gap UB vs LB	UB*	optimality proven?
5%	20580	4	19740	15	4.25%	-	no
1%	20100	15	19920	140	0.9%	-	no
0.5%	20040	178	19980	117	0.3%	-	no
0.3%	20040	64	19980	321	0.3%	19980	yes

Table 2: Upper and Lower bounds values obtained for the instance solved

The results show the efficiency of the resolution scheme proposed. More instances deriving from real-world wind profiles provided by the LAPLACE electrical engineering team are currently being tested.

7 Conclusion

We proposed a new resolution scheme based on the piecewise linear bounding and integer programming, applied on a water production optimization problem with non linear transfer functions. Efficient bounding heuristics for convex and concave transfer functions are also introduced. The resulting algorithm solves to optimality the global optimization problem of finding an optimal energy management strategy and validates the adequacy of the resolution methodology designed.

Acknowledgements

This work was funded in part by The Gaspard Monge Program for Optimization and operations research launched by EDF and the Jacques Hadamard Mathematical Foundation.

References

- [1] J. Blazewicz, K.H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Handbook on Scheduling: From Theory to Applications*. Springer, 2007.
- [2] Pierre Bonami, Lorenz T. Biegler, Andrew R. Conn, Gérard Cornuéjols, Ignacio E. Grossmann, Carl D. Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas Sawaya, and Andreas Wöchter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186 – 204, 2008. In Memory of George B. Dantzig.
- [3] A. Borghetti, C. D'Ambrosio, A. Lodi, and S. Martello. An milp approach for short-term hydro scheduling and unit commitment with head-dependent reservoir. *Power Systems, IEEE Transactions on*, 23(3):1115–1124, Aug 2008.
- [4] E. Camponogara, M.P. de Castro, and A. Plucenio. Compressor scheduling in oil fields: A piecewise-linear formulation. In *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*, pages 436–441, Sept 2007.
- [5] M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3):307 – 339, 1986.
- [6] C.A. Floudas and C.E. Gounaris. A review of recent advances in global optimization. *Journal of Global Optimization*, 45:3 – 38, 2009.
- [7] Ignacio E. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, 3:227 – 252, 2002.
- [8] P.K. Polisetty and E.P. Gatzke. A decomposition-based minlp solution method using piecewise linear relaxations. Technical report, University of South Carolina, August 2006.
- [9] X. Roboam, B. Sareni, D. T. Nguyen, and J. Belhadj. Optimal system management of a water pumping and desalination process supplied with intermittent renewable sources. In M. Fadel and S. Caux, editors, *8th Power Plant and Power System Control Symposium*, volume 8, pages 369–374, Toulouse, 2012. International Federation of Automatic Control.

8 Appendix: Non-linear expressions of the electrical power and water flow transfers modelling the mechanical-hydraulic power conversion systems

For pump 1, $h = 0$ at any instant (groundwater) whereas for pump 3, $h = \text{water level in tank 2}$, therefore:

$$p^{\text{pump}1} = f^{1\&3}(q_{\text{totank}(1)}, 0) \quad (23)$$

$$p^{\text{pump}3} = f^{1\&3}(q_{\text{totank}(3)}, l_{\text{tank}(2)}) \quad (24)$$

where

$$\begin{cases} f^{1\&3}(\mathbf{q}, \mathbf{h}) = ((f_m + f_p) * G(\mathbf{h}, \mathbf{q}) + \mathbf{q} * (a * G(\mathbf{h}, \mathbf{q}) + (b * \mathbf{q}))) * G(\mathbf{h}, \mathbf{q}) + r * K(\mathbf{h}, \mathbf{q}) \\ G(\mathbf{h}, \mathbf{q}) = (((-(b * \mathbf{q}) + \sqrt{((b * \mathbf{q})^2) - 4 * a * (-p_0 + \rho_g * (\mathbf{h} - l_{\text{discharge}}) + (k + c) * (\mathbf{q}^2))})))/(2 * a)) \\ K(\mathbf{h}, \mathbf{q}) = (((((f_m + f_p) * G(\mathbf{h}, \mathbf{q}) + \mathbf{q} * (a * G(\mathbf{h}, \mathbf{q}) + (b * \mathbf{q}))) / k_{\phi}).^2) \end{cases}$$

and $a, b, c, k, f_m, f_p, p_0, \rho_g, l_{\text{discharge}}$ are parameters deduced from the pump characteristics.

For the subsystem resulting from the combination of pump 2 and the Reverse Osmosis module:

$$p^{\text{pump}2+\text{ROmodule}} = f^2(q_{\text{totank}(2)}, l_{\text{tank}(1)}) \quad (25)$$

where

$$\begin{cases} f^2(\mathbf{q}_c, \mathbf{h}_2) = ((f_{m2} + f_{p2}) * G_2(\mathbf{q}_c, \mathbf{h}_2) + (\mathbf{q}_c + F(\mathbf{q}_c)/R_{Me}) * M_2(\mathbf{q}_c, \mathbf{h}_2)) * G_2(\mathbf{q}_c, \mathbf{h}_2) + r_2 * K_2(\mathbf{q}_c, \mathbf{h}_2) \\ M_2(\mathbf{q}_c, \mathbf{h}_2) = (a_2 * G_2(\mathbf{q}_c, \mathbf{h}_2) + H_2(\mathbf{q}_c)) \\ G_2(\mathbf{q}_c, \mathbf{h}_2) = \frac{(-H_2(\mathbf{q}_c) + \sqrt{((H_2(\mathbf{q}_c))^2) - 4 * a_2 * (-p_{02} + \rho_g * (\mathbf{h}_2 - l_{\text{discharge}}) + (k_2 + c_2) * ((\mathbf{q}_c + F(\mathbf{q}_c)/R_{Me})^2) + (F(\mathbf{q}_c))))})}{(2 * a_2)} \\ K_2(\mathbf{q}_c, \mathbf{h}_2) = (((((f_{m2} + f_{p2}) * G_2(\mathbf{q}_c, \mathbf{h}_2) + (\mathbf{q}_c + F(\mathbf{q}_c)/R_{Me}) * (a_2 * G_2(\mathbf{q}_c, \mathbf{h}_2) + H_2(\mathbf{q}_c))) / k_{\phi_2}).^2) \\ H_2(\mathbf{q}_c) = (b_2 * (\mathbf{q}_c + (R_{Mod} + R_{Valve}) * \mathbf{q}_c.^2 / R_{Me})) \\ F(\mathbf{q}_c) = (R_{Mod} + R_{Valve}) * \mathbf{q}_c.^2 \end{cases}$$

and $a_2, b_2, c_2, k_2, f_{m2}, f_{p2}, r_2, p_{02}, \rho_g, k_{\phi_2}, R_{Me}, R_{Mod}, R_{Valve}$ are parameters deduced from the pump characteristics.