Beyond classical distributed optimization methods:

Optimal control of electrical hot water tanks and Unit Commitment problem.

Internship report

Prepared by: Omar SAADI

EDF Paris-Saclay LAB

From April 3rd, 2018 to August 17th, 2018

Supervisor: Nadia Oudjane

September 9th, 2018

Contents

1 Introduction **3**

2 Two specific examples **4**
2.1 Unit Commitment problem 4
2.2 Optimal Control of Electrical Hot Water Tanks 4
2.3 Unique formulation for UC and EHWT problems 5

3 Two classical approaches for distributed optimization **6**
3.1 Lagrangian decomposition 6
3.2 Block minimization 10

4 A hybrid algorithm **12**
4.1 First hybrid algorithm by using the same price 12
4.1.1 Formulation 12
4.1.2 Equivalence with Lagrangian decomposition in case of quadratic penalty . . . 13
4.2 Second hybrid algorithm using different prices 14
4.3 Convergence analysis 15
4.4 Regularization 16

5 Simulations **19**
5.1 Simulations for the algorithms BM, LD and HD 20
5.2 Simulations with the regularized algorithms gLD and gHD 25
5.3 Summary of the performances of the algorithms 28

6 Conclusion and perspectives **30**

7 Acknowledgment **30**

Appendices **32**

A Optimal Control of Electrical Hot Water Tanks **32**
A.1 General considerations 32
A.2 Discrete-Time Dynamics 34

B Duality result **36**

1 Introduction

My internship took place in the OSIRIS department within EDF R&D Paris-Saclay. This department is responsible of developing tools and methods for the optimal management of EDF's asset portfolio (production plants, customer contracts, ...). Among the problems of great impact treated by the department, there is the problem of management of electricity production in the short term (**Unit Commitment problem**), and the problem of management of electrical flexibilities, especially the **optimal control of electrical hot water tanks**. These kind of problems present two major challenges. The first one is the **large scale** of the problems due to a big number of flexibilities or production units and in presence of a cost that couples the variables. This cost presents the cost of deviation of the aggregated consumption from a given target in the case of controlling hot water tanks, and presents the cost of not matching the demand in the unit commitment problem. The second challenge that present these problems is the **non-convexity** of the admissible sets that define the operating constraints of each local user or producer.
Distributed optimization methods are very useful approaches for dealing with these large optimization problems. They allow numerically to reduce the complexity of the problem by decomposing it to smaller problems. And in terms of organization, they allow to preserve the privacy related to local constraints of the users. Moreover, they allow to deal with the coupling term in the cost function.
A first classical distributed approach is based on the *price decomposition* technique using Lagrangian dualisation. This method is very powerful when the admissible sets are convex, but it faces difficulties in case of non-convex admissible sets. Another classical approach called *block minimization* is to sequentially optimize each production unit or flexibility by assuming the schedule for the other units already fixed. This second approach has the advantage of always decreasing the cost of the problem, but it might converge only to a local optimum in case of non-convex admissible set which might not be a global optimum even if the cost function is convex due to the non-convexity of the admissible set. In this work we will propose a new algorithm that hybridize these two algorithms in order to benefit from their respective advantages hoping to go beyond their limits.
In this report we will first present, in part 2, the problem of Unit Commitment (**UC**), and the problem of managing Electric Hot Water Tanks (**EHWT**), and we will underline the similitude between these two problems since they can be formulated as a unique mathematical optimization problem. Then, in the part 3, we will present the two classical methods: Lagrangian decomposition and block minimization. In part 4, we will present a new idea of hybridizing the two previous methods. Finally, in the part 5, we will present the results of the simulations achieved by the three algorithms applied to the problem of optimal control of electrical hot water tanks, in order to evaluate their performances in practical cases.

2 Two specific examples

2.1 Unit Commitment problem

We consider an electric generation mix composed by a set of electrical generators $i = 1, \cdots, n$ (thermal units, nuclear power plants, Hydropower,...). The aim of the unit commitment problem is to minimize energy production costs while satisfying the demand. But, since the real demand can only be approached by forecasts. The constraint of satisfying exactly the demand is rather expressed as a cost penalizing the deviation of the production from the expected demand. To formulate the UC problem we consider:

- $d := (d_t)_{t=0,\cdots T} \in \mathbb{R}^{T+1}$ is the demand to satisfy (taken as deterministic) on the period of time $(0, 1, \cdots, T)$.
- $p := (p_i)_{i=1,\cdots,n}$ the vector determining the production of each power plant i over the horizon of time $(0, 1, \cdots, T)$: $p_i := (p_{i,t})_{t=0,\cdots,T} \in \mathbb{R}^{T+1}$.
- $X_i \subset \mathbb{R}^{T+1}$ the admissible domain for operating programs of the electrical generator i. It models the operating constraints imposed by the nature of the power plant.
- $c_i : \mathbb{R}^{T+1} \to \mathbb{R}$ the cost function of the producer i. In the same way, we denote c_g the cost function corresponding to the penalization of the production gap. It's supposed to be decomposable by time steps, i.e. $c_g(d - \sum_{i=1}^{n} p_i) = \sum_{t=0}^{T} c_{g,t}(d_t - \sum_{i=1}^{n} p_{i,t})$.

So the aim of the unit commitment problem is to find a program of production for the local production plants minimizing the global cost including the costs of production of local plants and the penalty related to the unsatisfied demand, i.e.

$$\min_{p_i \in X_i} \left\{ \sum_{i=1}^{n} c_i(p_i) + c_g(d - \sum_{i=1}^{n} p_i) \right\} .$$

2.2 Optimal Control of Electrical Hot Water Tanks

The physical model of an electrical hot water tank is given in the annexe A.

Objective Function

We consider a pool of electrical hot water tanks with n tanks. For the tank i, the consumption is denoted $u_i = (u_{i,t})_{t=0,\cdots,p-1}$, and given the initial values of $(a_{i,0}, \tau_{i,0}, \mu_{i,0}, d_i)$ (see annexe A), we denote the admissible set of constraints for the tank i by $U_i := U(a_{i,0}, \tau_{i,0}, \mu_{i,0}, d_i)$. We notice that the admissible sets for the tanks are separable. And due to constraints (A.5) and (A.7), the admissible sets U_i are non-convex.
In the optimal control of our set of tanks, we have two objectives: minimizing the cost of heating for a given price signal over time $(c_0, \cdots, c_{p-1})$, and minimizing the deviation cost from a collective target consumption profile, $P = (P_t)_{t=0,\cdots,p-1}$.
Given a price signal over time steps $(c_0, \cdots, c_{p-1})$, the cost of electricity consumption is given by

$$\sum_{j=1}^{n} \sum_{t=0}^{p-1} c_t u_{j,t}$$

The second objective of reaching a collective target profile $P = (P_t)_{t=0,\cdots,p-1}$, can be modeled by adding a penalty for deviating from this target, given by

$$\sum_{t=0}^{p-1} \frac{\gamma_t}{2} (P_t - \sum_{j=1}^{n} u_{j,t})^2$$

where $(\gamma_0, \cdots, \gamma_{p-1})$ are positive penalty coefficients. We notice that this aggregated objective is over the tanks.
Then the minimization problem for the control of electrical hot water tanks is formulated as follows:

$$\min_{\forall i,\ u_i \in U_i} \sum_{i=1}^{n} \sum_{t=0}^{p-1} c_t u_{i,t} + \sum_{t=0}^{p-1} \frac{\gamma_t}{2} (P_t - \sum_{i=1}^{n} u_{i,t})^2$$

2.3 Unique formulation for UC and EHWT problems

From the previous formulations of the problems of Unit Commitment and of controlling Electrical Hot Water Tanks, we deduce that they can be formulated in the same way as follows:

$$\min_{\forall i,\ u_i \in U_i} f_0(P - \sum_{i=1}^{n} u_i) + \sum_{i=1}^{n} f_i(u_i) \tag{2.1}$$

where for all i, $u_i = (u_{i,t})_{t=0,\cdots,p-1} \in U_i \subset \mathbb{R}^p$ with U_i being non-convex, and $P = (P_t)_{t=0,\cdots,p-1}$ is a target profile.
We notice that the admissible set for the problem $U_1 \times \cdots \times U_n$ is separable, and these admissible sets U_i are the only difference between the problems of UC and EHWT.
We remark also that the global cost contains a separable cost $\sum_{i=1}^{n} f_i(u_i)$ and a non-separable cost $f_0(P - \sum_{i=1}^{n} u_i)$ that depends on the sum of the variables u_i.
We denote the objective function by f such that:

$$f(u) = f_0(P - \sum_{i=1}^{n} u_i) + \sum_{i=1}^{n} f_i(u_i), \text{ for } u = (u_i)_i \in \prod_{i=1}^{n} U_i \tag{2.2}$$

Need for decomposition of the problem

It is possible to solve this global problem without decomposing it, by using solvers based on methods of branching, but the time needed to solve it increases in an exponential way with the number of tanks n by using the solver SCIP for example. This is shown in the paper of Nathanael Beeker [2]. In this paper we find the table 1 showing the computational time needed to solve this problem by using the SCIP solver for the number of tanks $n = 1$, 2, 3 and 4.

Number of tanks	Time
1	2.0s
2	9.4s
3	754.2s
4	4756.4s

Table 1: Computational times.

Therefore, looking for decomposition methods is needed to solve this problem with larger numbers of tanks and in reasonable computational times.
So, in the following parts of this report we will show two main methods of decomposition that are *Lagrangian Decomposition (LD)* and *Block Minimization (BM)*, and we will propose also a *Hybrid Decomposition (HD)* method that is inspired from the two previous ones.

3 Two classical approaches for distributed optimization

In this part, we will present the two classical distributed optimization methods: Lagrangian decomposition and block minimization. We will present the corresponding algorithms, analyze their convergence and underline the limits of these two methods.
We will base our presentation on the works of Tseng et al. [12], [11] and [8] on the block minimization methods, the works of Boyd et al. [6], and [5] and the book of Bertsekas [4] for the duality analysis and subgradient methods. We notice also, that some more general classical methods called PPA (*Principe du Problème Auxiliaire*) [7] exist and can also allow decomposed optimization approaches.

3.1 Lagrangian decomposition

In this section we will present the method of Lagrangian decomposition.
The original problem, called the primal problem (non-decomposed) is the following

$$\min_{\forall i,\ u_i \in U_i} \sum_{i=1}^{n} f_i(u_i) + f_0(P - \sum_{i=1}^{n} u_i) \tag{3.1}$$

This problem is not decomposable at this step because of the coupling term coming from the penalization function $f_0(\cdot)$. To decouple the variables we add a new variable $v = P - \sum_{i=1}^{n} u_i$, so that we have the following problem:

$$\min_{u_i \in U_i,\ v \in \mathbb{R}^p} \sum_{i=1}^{n} f_i(u_i) + f_0(v) \tag{3.2}$$

$$\text{s.t. } v_t = P_t - \sum_{i=1}^{n} u_{i,t}$$

We introduce Lagrangian coefficients $\lambda = (\lambda_0, \cdots, \lambda_{p-1}) \in \mathbb{R}^p$, and we define the *Lagrangian* of the problem as follows:

$$\mathcal{L}(u, v, \lambda) = \sum_{i=1}^{n} f_i(u_i) + f_0(v) + \sum_{t=0}^{p-1} \lambda_t (v_t + \sum_{i=1}^{n} u_{i,t} - P_t) \tag{3.3}$$

This Lagrangian can be written in a decomposed way

$$\mathcal{L}((u_i)_i, v, \lambda) = \sum_{i=1}^{n} \mathcal{L}_i(u_i, \lambda) + \mathcal{L}_0(v, \lambda) - \sum_{t=0}^{p-1} \lambda_t P_t$$

such that $\mathcal{L}_i(u_i, \lambda) = f_i(u_i) + \sum_{t=0}^{p-1} \lambda_t u_{i,t}$ and $\mathcal{L}_0(v, \lambda) = f_0(v) + \sum_{t=0}^{p-1} \lambda_t v_t$.
The dual function W is define by

$$W(\lambda) = \min_{u_i \in U_i,\ v \in \mathbb{R}^p} \mathcal{L}(u, v, \lambda)$$

The dual function is a minimum of affine functions in λ, therefore the dual function W is concave.

Duality

We have always the following **weak duality** result:

$$\max_{\lambda \in \mathbb{R}^p} W(\lambda) \leq \min_{u \in \prod_{i=1}^{n} U_i} f(u)$$

We say that the **strong duality** holds when the equality holds. But, in our case where the admissible set $U = \prod_{i=1}^{n} U_i$ is non-convex, the strong duality might not hold.

Differentiability of the dual function

Let us denote the objective function of the primal problem by F and the constraint function by h, we have for all $u = (u_i)_i \in \prod_{i=1}^{n} U_i$ and $v \in \mathbb{R}^p$;

$$F(u, v) := \sum_{i=1}^{n} f_i(u_i) + f_0(v)$$

$$h(u, v) = v + \sum_{i=1}^{n} u_i - P \in \mathbb{R}^p$$

For a given $\lambda \in \mathbb{R}^p$, we denote by $(u(\lambda), v(\lambda))$ a minimizer of the Lagrangian $\mathcal{L}(u, v, \lambda)$, over $u \in \prod_{i=1}^{n} U_i$ and $v \in \mathbb{R}^p$,

$$(u(\lambda), v(\lambda)) \in \underset{u \in \prod_{i=1}^{n} U_i, v \in \mathbb{R}^p}{\operatorname{argmin}} \mathcal{L}(u, v, \lambda) = \underset{u,v}{\operatorname{argmin}} F(u, v) + \langle \lambda, h(u, v) \rangle.$$

From the book by Bertsekas ([4], Chapter 6), we have that $h(u(\lambda), v(\lambda))$ is a *surgradient* of the dual function W at λ.
We notice that $v(\lambda) \in \mathbb{R}^p$ can be easily computed if we suppose that the function f_0 is strictly convex and separable $f_0(v) = \sum_{t=0}^{p-1} f_{0,t}(v_t)$, which is the case for our applications since f_0 is quadratic and separable. In this case the part $\min_{v \in \mathbb{R}^p} f_0(v) + \langle \lambda, v \rangle = f_0(v(\lambda)) + \langle \lambda, v(\lambda) \rangle$ is differentiable with respect to λ. So now we will deal only with the more difficult part of the Lagrangian which is $\min_{u \in U := \prod_{i=1}^{n} U_i} L(u, \lambda)$, where we define $L(u, \lambda) := \sum_{i=1}^{n} (f_i(u_i) + \langle \lambda, u_i \rangle)$. From the book by Bertsekas ([4], Chapter 6) we can deduce the following proposition giving the case were W is differentiable.

Proposition 3.1. *We suppose that $U = \prod_{i=1}^{n} U_i$ is a compact set, that f_i is continuous on U_i for all i, and that for every $\lambda \in \mathbb{R}^p$, $L(u, \lambda)$ is minimized over $u \in U$ at a unique $u(\lambda)$. Then, W is every where continuously differentiable and*

$$\nabla W(\lambda) = h(u(\lambda), v(\lambda)) = v(\lambda) + \sum_{i=1}^{n} u_i(\lambda) - P, \ \forall \lambda \in \mathbb{R}^p.$$

Remark 3.1.

- *If The dual function is differentiable then strong duality holds and there is no duality gap, which means that the maximum of the dual function is equal to the minimum of the primal problem.*
- *Note that if the constraint function $h(u, v)$ is linear (which is the case for the problem EHWT), U is convex and compact (we have the compactness in EHWT but not the convexity), and f is strictly convex (we can get it by adding a small strict convex term to the functions f_i), then the assumptions of the proposition 3.1 are satisfied and the dual function W is differentiable.*
- *So in our example of EHWT, the main difficulty that, can lead to duality gap, is the non-convexity of the feasible set U.*

Subgradient ascent algorithm

The *dual problem* consists of the following maximization problem

$$\max_{\lambda \in \mathbb{R}^p} W(\lambda)$$

By weak duality that always holds, the value of this dual problem gives a lower bound to the value of the primal problem. And in case of strong duality these two values are equal. Unfortunately, in the case we are interested on where the admissible set is non-convex, we might lose strong duality, but the value of the dual problem is still a lower bound for the primal problem.
The dual function presents two interesting properties that make easy the resolution of the dual problem. First, since the dual function is defined as a maximum of affine functions in λ, then the dual function is a **concave function** which is a very useful property for maximization. In addition to that, the computation of a subgradient of the dual function in our case is simple and moreover it's a **separable gradient**, and this property allows the Lagrangian decomposition to be a distributed approach. Based on these properties we deduce that it's interesting to solve the dual problem using the gradient ascent method.
In order to solve the dual problem, which is a problem of maximization, we will use **Subgradient Ascent Algorithm** (Dual Ascent). This algorithm is given by starting from a given λ^0, and then updating it as follows (see Stephen Boyd [5]):

$$\lambda^{k+1} = \lambda^k + \rho_k \partial W(\lambda^k)$$

where $\partial W(\lambda^k)$ is a subgradient of W, in our case we can take $\partial W(\lambda^k) = h(u(\lambda^k), v(\lambda^k)) = v(\lambda^k) + \sum_{i=1}^{n} u_i(\lambda^k) - P$.
$\rho_k > 0$ are gradient stepsizes that must be chosen carefully. If they are too small the algorithm runs slow and if they are too big the algorithm is not converging. We will choose the *diminishing step size rule* (see Stephen Boyd [5]), i.e. $\rho_k > 0$ satisfying:

$$\rho_k \to 0, \ \sum_{k=0}^{\infty} \rho_k = \infty. \tag{3.4}$$

In that article [5], we have the convergence of the subgradient ascent algorithm:

Proposition 3.2. *If we suppose that the function W has a maximizer, and that the norm of the subgradients used during the iterations is bounded, then by choosing the diminishing step size rule ($\rho_k \to 0, \ \sum_{k=0}^{\infty} \rho_k = \infty$), the subgradient ascent algorithm converges to the maximum of W.*

Remark 3.2. *In the simulations, we will use the diminishing stepsize rule with $\rho_k = \frac{c}{\sqrt{k}}$, with c being a constant.*

An interesting particular case

We notice that a particular case that is interesting for us, is when there is no duality gap but the dual function is not differentiable while having non-convex admissible set. In this case, the Lagrangian method faces difficulties. Because even if we can maximize the dual function by subgradient method, we cannot construct a primal solution due to the non-convexity of the admissible set. In fact, when the admissible set is convex, we can construct a primal solution by an adequate convex combination of the solutions visited over the iterations, which is not possible in our case due to non-convexity of the admissible set. In the section 4 of this report, we prove that the hybrid algorithm we are proposing is particularly interesting in this case where there is no duality gap but the dual function is not differentiable.

Algorithm of Lagrangian decomposition

From the previous analysis we can now give the Lagrangian decomposition algorithm as follows: At iteration $k+1$, we have:

1. $u_i^{k+1} \in \underset{u_i \in U_i}{\operatorname{argmin}} \; f_i(u_i) + \sum_{t=0}^{p-1} \lambda_t^k u_{i,t}$

2. $v^{k+1} \in \underset{v}{\operatorname{argmin}} \; f_0(v) + \sum_{t=0}^{p-1} \lambda_t^k v_t$

3. $\lambda_t^{k+1} = \lambda_t^k + \rho_k(v_t^{k+1} + \sum_{i=1}^{n} u_{i,t}^{k+1} - P_t), \; \forall t \in [0, \cdots, p-1]$

During this iteration, the tanks all use the same dual price λ^k and calculate based on this price their optimal levels of consumption/production $(u_i^{k+1})_i$ (see Figure 1). The price λ^k is then updated to reflect the difference between the total consumption/production of the iteration and the target P.

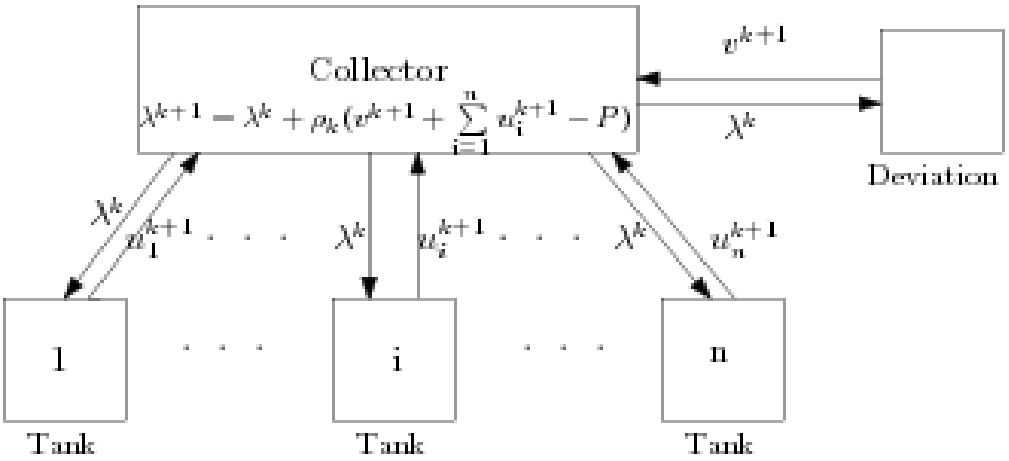

Figure 1: Schematic view of the Lagrangian decomposition

We notice that the local problems can be solved easily by dynamic programming or branching techniques because they have a small size even if they are non-convex. In particular, when the functions f_i, $i = 1, \cdots, n$ are linear and f_0 is quadratic, which is the case of our application for electric hot water tanks, all the local problems solved are linear with non-convex constraints.

3.2 Block minimization

We consider the original non-decomposed problem that we want to decompose into sub-problems to be able to solve it in reasonable time:

$$\min_{\forall i,\ u_i \in U_i} \sum_{i=1}^{n} f_i(u_i) + f_0(P - \sum_{i=1}^{n} u_i)$$

If we consider only the tank $i \in \{1, \cdots, n\}$, minimizing the global cost with respect only to u_i leads to the following local problem,

$$\min_{u_i \in U_i} f_i(u_i) + f_0(P - u_i - \sum_{j \neq i} u_j) \tag{3.5}$$

The idea of the block minimization method is to consider that all the other consumption u_j for $j \neq i$ are fixed, and based on their values, we solve the problem (3.5) and find the best response u_i of the tank i that we can make against the consumptions $(u_j)_{j \neq i}$. That is why this algorithm is called also *Best Response* algorithm and used in game theory problems to found the equilibrium.
To form the algorithm, we consider first an initial point $(u_1^0, \cdots, u_n^0)$. Then we will apply the local minimizations starting by $i = 1$, so that we will update u_1, and get u_1^1 given $(u_2^0, \cdots, u_n^0)$. Then we will move to $i = 2$, and we update u_2, and get u_2^1 given $(u_1^1, u_3^0, \cdots, u_n^0)$.
When we come to the i^{th} tank, we update u_i, and get u_i^1 given $(u_1^1, \cdots, u_{i-1}^1, u_{i+1}^0, \cdots, u_n^0)$, and so on until $i = n$. The passage on all the tanks from $i = 1$ to $i = n$ is considered as one iteration of the algorithm.
So the algorithm can be written as follows, going from $i = 1$ to $i = n$ (a cycling rule):
At iteration $k + 1$, we have
$u_i^{k+1} = \underset{u_i \in U_i}{\operatorname{argmin}}\ f_i(u_i) + f_0(P - u_i - S_i^{k+1})$
such that S_i^{k+1} is given by $S_i^{k+1} := \sum_{j<i} u_j^{k+1} + \sum_{j>i} u_j^k$.
We notice that the local problems can be solved easily by dynamic programming or branching techniques as for the Lagrangian decomposition algorithm, and in particular when the functions f_i, $i = 1, \cdots, n$ are linear and f_0 is quadratic, which is the case of our application for electric hot water tanks.

Remark 3.3. *At every local minimization we are getting a smaller value of the global cost. So the block minimization algorithm is always decreasing with the iterations even if the admissible set is non-convex.*

The following figure 2 gives a schematic view of the block minimization algorithm:

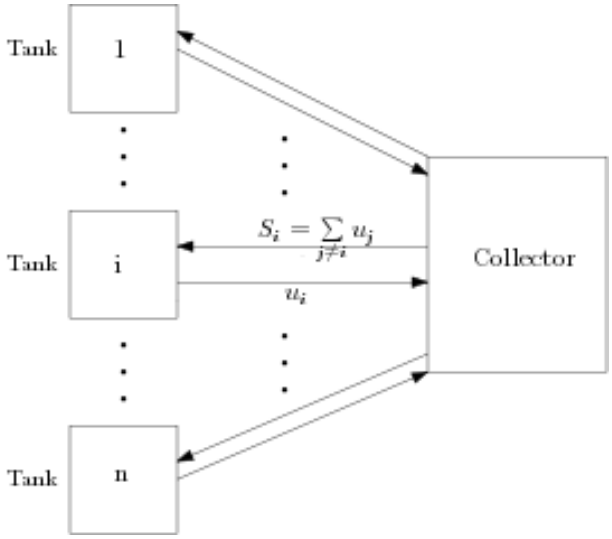

Figure 2: Schematic view of the block minimization algorithm.

Convergence analysis of the block minimization algorithm

A multitude of articles have dealt with the convergence analysis of the block minimization method. Among these articles there is the works of Tseng and Luo [11] that is the source of the following proposition (3.3), and also the articles [8] and [12]. There is also the works of Nutini et al. [9].
Let $\phi : \mathcal{D} \to \mathbb{R}$ be a function with $\mathcal{D}$ convex. Let's begin by some definitions.

- We denote the (lower) directional derivative of ϕ at x in the direction d by

$$\phi'(x;d) = \liminf_{\mu \to 0} \frac{\phi(x+\mu d) - \phi(x)}{\mu}.$$

- *Stationary points of a function:* A point x is a stationary point of $\phi(\cdot)$ if $\phi'(x;d) \geq 0$ for all d such that $x + d \in \mathcal{D}$.

- *Regularity of a function at a point:* The function $\phi : \mathbb{R}^m \to \mathbb{R}$ is regular at the point $x \in \mathcal{D} = \text{dom}(\phi)$ with respect to the coordinates $m_1, m_2, \ldots, m_n$, such that $m_1 + m_2 + \cdots + m_n = m$, if when $\phi'(x; d_k^0) \geq 0$ for all $d_k^0 = (0, \ldots, d_k, \ldots, 0)$ with $d_k \in \mathbb{R}^{m_k}$ and for all k, then it implies that $\phi'(x;d) \geq 0$ for all $d = (d_1, \ldots, d_n)$.
  If ϕ is differentiable then it is regular.

- *Pseudoconvexity of a function:* The function ϕ is pseudoconvex if

$$\phi(x+d) \geq \phi(x), \text{ whenever } x \in \text{dom } \phi \text{ and } \phi'(x;d) \geq 0;$$

  and if ϕ is convex then it's also pseudoconvex.

The objective function of our problem is denoted by $f : U = \prod_{i=1}^{n} U_i \to \mathbb{R}$, such that

$$f(u) = \sum_{i=1}^{n} f_i(u_i) + f_0(P - \sum_{i=1}^{n} u_i), \text{ for } u = (u_i)_{i=1\cdots,n}$$

We have the following proposition on the convergence of the block minimization algorithm.

Proposition 3.3. *Assume that the level set $U^0 = \{u \in U : f(u) \leq f(u^0)\}$ is compact, and that f is continuous on U^0. If we assume also that f is pseudoconvex in (u_i, u_k) for every $i, k \in \{1, \cdots, n\}$ and that f is regular at every $u \in U^0$, then every cluster point of the iterations $\{u^k\}$ is a stationary point.*

Remark 3.4. *The assumptions of this proposition holds in the case of the EHWT problem because the set U is compact, the cost function f is continuous, differentiable and convex. But since the set U is not convex, a stationary point is only a local minimum of f and not a global minimum.*

So the block minimization algorithm has the advantage of leading to a decreasing sequence of costs that converges to a local minimum, but it might not be the global minimum in the case of non-convex admissible set U.

4 A hybrid algorithm

In the previous chapter of this report, we presented the two classical methods allowing distributed optimization. The first one was the Lagrangian decomposition method. We showed that this method is well adapted to the case of strong duality, when there is no duality gap. But it faces the difficulty related to the fact that we can not use a convex combination of the previous primal solutions since the admissible set is not convex. It has the benefits of being very efficient when it works and uses a decomposition with prices λ. The second method was the block minimization method that has the benefit of leading to decreasing costs over the iterations. However, this block minimization method is in practice slower than the Lagrangian decomposition when the last one converges.
In this chapter we want to exhibit a new algorithm that can hybridize the two previous ones in order to benefit from their advantages and surpass their limits. The idea is to introduce prices as in the Lagrangian decomposition but to use specific price λ_i for each tank i, and we want to derive the formulations of these prices by inspiration from the block minimization approach.

4.1 First hybrid algorithm by using the same price

4.1.1 Formulation

We consider the original non-decomposed problem:

$$\min_{\forall i,\ u_i \in U_i} \sum_{i=1}^{n} f_i(u_i) + f_0(P - \sum_{i=1}^{n} u_i) \tag{4.1}$$

Our idea is to have sub-problems like in Lagrangian decomposition of the form $\min_{u_i \in U_i} f_i(u_i) + \langle \lambda, u_i \rangle$, where $\langle \lambda, u_i \rangle = \sum_{t=0}^{p-1} \lambda_t u_{i,t}$ We can obtain this by linearizing the non-separable term $f_0(P - \sum_{i=1}^{n} u_i)$.
As in the block minimization we will update the values of u_i starting by $i = 1$ and ending by $i = n$ at each iteration.
So at iteration $k+1$ and while updating the i^{th} tank, we have in the block minimization algorithm the following local problem:

$$u_i^{k+1} = \underset{u_i \in U_i}{\operatorname{argmin}}\ f_i(u_i) + f_0(P - u_i - S_i^{k+1}) \tag{4.2}$$

with $S_i^{k+1} := \sum_{j<i} u_j^{k+1} + \sum_{j>i} u_j^k$.
We consider a linearization of the term $f_0(P - u_i - S_i^{k+1})$ by a Taylor development at the first order. We have:

$$f_0(P - u_i - S_i^{k+1}) = f_0(P - (u_i - u_i^k) - \sum_{j<i}(u_j^{k+1} - u_j^k) - \sum_{j=1}^{n} u_j^k) \tag{4.3}$$

$$\approx f_0(P - \sum_{j=1}^{n} u_j^k) - \nabla f_0(P - \sum_{j=1}^{n} u_j^k) \cdot (u_i - u_i^k + \sum_{j<i}(u_j^{k+1} - u_j^k)) \tag{4.4}$$

We suppose that the terms $(u_j^{k+1} - u_j^k)$ for $j < i$ and $(u_i - u_i^k)$ to have a sum $(\sum_{j<i}(u_j^{k+1} - u_j^k) + u_i - u_i^k)$ small enough so that the linearization can be accurate.
Then the approximated local problem can be written as $u_i^k(\lambda^k) \in \underset{u_i \in U_i}{\operatorname{argmin}}\ f_i(u_i) + \langle \lambda^k, u_i \rangle$, with the following update of λ:

$$\lambda^{k+1} = -\nabla f_0(P - \sum_{j=1}^{n} u_j^k(\lambda^k)) \tag{4.5}$$

We notice that this price λ^{k+1} is the same for all tanks. We remark also that the equation (4.5) is an iteration looking for the fixed point $\lambda = -\nabla f_0(P - \sum_{j=1}^{n} u_j^k(\lambda))$. We remark also that updating

λ with the value $-\nabla f_0(P - \sum_{j=1}^{n} u_j^k)$ will lead to iterations giving to λ very different values at each iteration, which leads to large oscillations of the global cost over the iterations. We remarked this fact with simulations and we will explain it the following subpart. To deal with this problem, a good method is to update λ by taking into account its previous value and also the term $-\nabla f_0(P - \sum_{j=1}^{n} u_j^k)$ by doing a convex combination as follows:

$$\lambda^{k+1} = (1 - r_k)\lambda^k - r_k \ \nabla f_0(P - \sum_{j=1}^{n} u_j^k)$$

where $0 < r_k < 1$ has to be chosen carefully.
Therefore the algorithm that we propose is the following:
At iteration $k+1$, we have

1. $u_i^{k+1} = \underset{u_i \in U_i}{\text{argmin}} \ f_i(u_i) + \sum_{t=0}^{p-1} \lambda_t^k u_{i,t}$, for all $i = 1, \cdots, n$
2. $\lambda^{k+1} = (1 - r_k)\lambda^k - r_k \ \nabla f_0(P - \sum_{j=1}^{n} u_j^{k+1})$

4.1.2 Equivalence with Lagrangian decomposition in case of quadratic penalty

In our two applications of EHWT and UC the penalty term related to the target profile of consumption or production is quadratic of the form:

$$f_0(v) = \sum_{t=0}^{p-1} \frac{\gamma_t}{2}(v_t)^2$$

for all $v = (v_t)_{t=0,\cdots,p-1} \in \mathbb{R}^p$. In our application we consider that $\gamma_t = \gamma$ for all $t = 0, \cdots, p-1$. Then we have:

$$\nabla f_0(v) = (\gamma v_t)_{t=0,\cdots,p-1} \in \mathbb{R}^p$$

Therefore the hybrid algorithm can be written as follows: At iteration $k+1$, we have

1. $u_i^{k+1} = \underset{u_i \in U_i}{\text{argmin}} \ f_i(u_i) + \sum_{t=0}^{p-1} \lambda_t^k u_{i,t}$, for $i = 1, \cdots, n$
2. $\lambda_t^{k+1} = (1 - r_k)\lambda_t^k + r_k\gamma(\sum_{j=1}^{n} u_{j,t}^{k+1} - P_t)$, for $t = 0, \cdots, p-1$

Lagrangian Decomposition: Now we go back to the Lagrangian decomposition algorithm to simplify it. The algorithm was written as follows (see section 3.1): At iteration $k+1$, we have

1. $u_i^{k+1} \in \underset{u_i}{\text{argmin}} \ f_i(u_i) + \sum_{t=0}^{p-1} \lambda_t^k u_{i,t}$
2. $v^{k+1} \in \underset{v}{\text{argmin}} \ f_0(v) + \sum_{t=0}^{p-1} \lambda_t^k v_t$
3. $\lambda_t^{k+1} = \lambda_t^k + \rho_k(v_t^{k+1} + \sum_{i=1}^{n} u_{i,t}^{k+1} - P_t)$

We will develop the computation of v^{k+1}, we have

$$\min_{(v_t)_t \in \mathbb{R}^p} f_0(v) + \sum_{t=0}^{p-1} \lambda_t^k v_t = \sum_{t=0}^{p-1} \min_{v_t \in \mathbb{R}} \frac{\gamma}{2}(v_t)^2 + \lambda_t^k v_t$$

So by derivation we can easily compute $v_t^{k+1} = \underset{v_t \in \mathbb{R}}{\operatorname{argmin}} \frac{\gamma}{2}(v_t)^2 + \lambda_t^k v_t = -\frac{\lambda_t^k}{\gamma}$.
Therefore the update above update of λ becomes of the form:

$$\lambda_t^{k+1} = \lambda_t^k + \rho_k(-\frac{\lambda_t^k}{\gamma} + \sum_{i=1}^{n} u_{i,t}^{k+1} - P_t) = (1 - \frac{\rho_k}{\gamma})\lambda_t^k + \frac{\rho_k}{\gamma}\gamma(\sum_{i=1}^{n} u_{i,t}^{k+1} - P_t)$$

In conclusion, the Lagrangian decomposition algorithm makes the same update of λ as the first hybrid algorithm shown above. To see it, we have just to see that the choice of the weighting coefficient r_k and the choice of the step size ρ_k are equivalent by the relation $r_k = \frac{\rho_k}{\gamma}$.
The fact that r_k plays the role of ρ_k explains why choosing carefully r_k such that it's not allowed to be to big so that the algorithm will maybe not converge, and also to don't be allowed to be too small so that the algorithm becomes very slow. In fact not using the weighting that we introduced with r_k by taking $r_k = 1$ is not necessarily the best choice. So this justifies the introduction of weighting that we made.

4.2 Second hybrid algorithm using different prices

We go back to the update of u_i^{k+1} (see (4.2)), we have in the block minimization algorithm the following local problem:

$$u_i^{k+1} = \underset{u_i \in U_i}{\operatorname{argmin}} f_i(u_i) + f_0(P - u_i - S_i^{k+1})$$

with $S_i^{k+1} := \sum_{j<i} u_j^{k+1} + \sum_{j>i} u_j^k$.
We consider know a more realistic approximation by linearization of the term $f_0(P - u_i - S_i^{k+1})$ by a Taylor development at the first order. By denoting $\tilde{S}_i^{k+1} = \sum_{j<i} u_j^{k+1} + \sum_{j=i}^{n} u_j^k = S_i^{k+1} + u_i^k$, we have :

$$\begin{aligned} f_0(P - u_i - S_i^{k+1}) = & \quad f_0(P - (u_i - u_i^k) - \sum_{j<i} u_j^{k+1} - \sum_{j=i}^{n} u_j^k) && (4.6)\\ = & \quad f_0(P - (u_i - u_i^k) - \tilde{S}_i^{k+1}) && (4.7)\\ \approx & \quad f_0(P - \tilde{S}_i^{k+1}) - \nabla f_0(P - \tilde{S}_i^{k+1}) \cdot (u_i - u_i^k) && (4.8) \end{aligned}$$

This approximation is more accurate than the one in the first hybrid algorithm because here we have to suppose that only the term $(u_i - u_i^k)$ is small enough, without imposing this assumption to the hole sum $(\sum_{j<i}(u_j^{k+1} - u_j^k) + u_i - u_i^k)$ as before.
Therefore the second hybrid algorithm that we propose is the following:
At iteration $k+1$, begin with $i = 1$ and do

1. $\lambda_i^{k+1} = (1 - r_k)\lambda_i^k - r_k \nabla f_0(P - \sum_{j=1}^{i-1} u_j^{k+1} - \sum_{j=i}^{n} u_j^k)$
2. $u_i^{k+1} = \underset{u_i \in U_i}{\operatorname{argmin}} f_i(u_i) + \sum_{t=0}^{p-1} \lambda_{i,t}^{k+1} u_{i,t}$
3. increase i

We notice that in this algorithm we update for each $i = 1, \cdots, n$, first λ_i which is specific for the tank i and then we use it to update u_i.

Application in the case of quadratic penalization

We consider the case of quadratic penalization $f_0(v) = \sum_{t=0}^{p-1} \frac{\gamma_t}{2}(v_t)^2$ that interests us for the applications for EHWT and UC. The second hybrid algorithm becomes:
At iteration $k+1$, begin with $i = 1$ and do

1. $\lambda_{i,t}^{k+1} = (1-r_k)\lambda_{i,t}^k + r_k\gamma(\sum_{j=1}^{i-1} u_{j,t}^{k+1} + \sum_{j=i}^{n} u_{j,t}^k - P_t)$, for $t = 0, \cdots, p-1$

2. $u_i^{k+1} = \underset{u_i \in U_i}{\operatorname{argmin}}\ f_i(u_i) + \sum_{t=0}^{p-1} \lambda_{i,t}^{k+1} u_{i,t}$

3. increase i

With this algorithm we update sequentially the variables u_i from $i = 1$ to $i = n$, with a specific price λ_i to each u_i. Contrary to the Lagrangian decomposition, the use of specific prices allows us to differentiate the users and to avoid the synchronization of the responses u_i.
Since the first hybrid algorithm coincides with the Lagrangian decomposition algorithm in the case of our applications, we will denote by *Hybrid Decomposition* (HD), in the rest of this report, only the second hybrid algorithm.

4.3 Convergence analysis

We show in the following lemma that when the hybrid algorithm converges then it converges to the global minimum of the objective function, in the case $r_k = 1$, i.e. without averaging.
In this case the update of λ is given by $\lambda_i^{k+1} = -\nabla f_0(P - \sum_{j=1}^{i-1} u_j^{k+1} - \sum_{j=i}^{n} u_j^k)$, and the $(k+1)^{\text{th}}$ iteration of the hybrid algorithm becomes:

$$u_i^{k+1} = \underset{u_i \in U_i}{\operatorname{argmin}}\ f_i(u_i) - u_i \cdot \nabla f_0(P - \sum_{j=1}^{i-1} u_j^{k+1} - \sum_{j=i}^{n} u_j^k),\ \ i = 1, \cdots, n$$

Lemma 4.1. *We assume that f_0 is convex. If the sequence of iterations (u^k) of the hybrid decomposition converges to a limit u^*, in the case of $r_k = 1$, $\forall k$, then u^* is a global minimizer of the cost function f.*

Proof. By the iteration $u_i^{k+1} = \underset{u_i \in U_i}{\operatorname{argmin}}\ f_i(u_i) - u_i \cdot \nabla f_0(P - \sum_{j=1}^{i-1} u_j^{k+1} - \sum_{j=i}^{n} u_j^k)$, we deduce that:

$$f_i(u_i^{k+1}) - u_i^{k+1} \cdot \nabla f_0(P - \sum_{j=1}^{i-1} u_j^{k+1} - \sum_{j=i}^{n} u_j^k) \leq f_i(u_i) - u_i \cdot \nabla f_0(P - \sum_{j=1}^{i-1} u_j^{k+1} - \sum_{j=i}^{n} u_j^k), \forall u_i \in U_i \quad (4.9)$$

Therefore by tending $k \to \infty$ and by summing over i we obtain that

$$\nabla f_0(P - \sum_{i=1}^{n} u_i^*) \cdot \sum_{i=1}^{n} (u_i^* - u_i) + \sum_{i=1}^{n} (f_i(u_i) - f_i(u_i^*)) \geq 0, \forall\ u_i \in U_i,\ \forall\ i, \quad (4.10)$$

And by the convexity of f_0 we have that,

$$\begin{aligned} f_0(P - \sum_{i=1}^{n} u_i) - f_0(P - \sum_{i=1}^{n} u_i^*) &\geq \nabla f_0(P - \sum_{i=1}^{n} u_i^*) \cdot [(P - \sum_{i=1}^{n} u_i) - (P - \sum_{i=1}^{n} u_i^*)] \quad (4.11) \\ &= \nabla f_0(P - \sum_{i=1}^{n} u_i^*) \cdot \sum_{i=1}^{n} (u_i^* - u_i) \quad (4.12) \end{aligned}$$

By the two previous inequalities, we deduce that:

$$f_0(P - \sum_{i=1}^{n} u_i) + \sum_{i=1}^{n} f_i(u_i) - f_0(P - \sum_{i=1}^{n} u_i^*) - \sum_{i=1}^{n} f_i(u_i^*)) \geq 0, \forall\ u_i \in U_i,\ \forall\ i,$$

i.e. $f(u) - f(u^*) \geq 0$, $\forall u \in U$, which proves that u^* is a global minimizer of f. □

We recall that for $\lambda \in \mathbb{R}^p$, $u_i(\lambda^*)$ denotes a minimizer of $\mathcal{L}_i(u_i, \lambda) = f_i(u_i) + \langle \lambda, u_i \rangle$, and we suppose that f_0 is differentiable and strictly convex.

Lemma 4.2. *The problem (4.1) presents no duality gap if and only if* $\exists \lambda^* \in \mathbb{R}^p$ *and* $(u_i(\lambda^*))_i$ *such that* $\lambda^* = -\nabla f_0(P - \sum_{i=1}^{n} u_i(\lambda^*))$.

Proof.

We suppose that there no duality gap, let λ^* be a maximizer of the dual function W, by the duality result (B.1) presented in the Annexe B, we know that there exists $(u(\lambda^*), v(\lambda^*))$ which is solution to the primal problem (3.2). So we have $v(\lambda^*) = P - \sum_{i=1}^{n} u_i(\lambda^*)$. And since $v(\lambda^*) \in \underset{v \in \mathbb{R}^p}{\operatorname{argmin}} f_0(v) + \langle \lambda^*, v \rangle$, we obtain by differentiability of f_0 that $lambda^* = -\nabla f_0(v(\lambda^*))$. Therefore $\lambda^* = -\nabla f_0(P - \sum_{i=1}^{n} u_i(\lambda^*))$.

We suppose now that $\exists \lambda^* \in \mathbb{R}^p$ and $(u_i(\lambda^*))_i$ such that $\lambda^* = -\nabla f_0(P - \sum_{i=1}^{n} u_i(\lambda^*))$. By differentiability and strict convexity of f_0, the unique minimizer of $f_0(v) + \langle \lambda^*, v \rangle$ is the unique element $v(\lambda^*)$ satisfying $\lambda^* = -\nabla f_0(v(\lambda^*))$. And since $\lambda^* = -\nabla f_0(P - \sum_{i=1}^{n} u_i(\lambda^*))$ then $v(\lambda^*) = P - \sum_{i=1}^{n} u_i(\lambda^*)$. Therefore,

$$W(\lambda^*) = \sum_{i=1}^{n} f_i(u_i(\lambda^*)) + f_0(v(\lambda^*)) + \langle \lambda, v(\lambda^*) + \sum_{i=1}^{n} u_i(\lambda^*) - P \rangle \tag{4.13}$$

$$= f_i(u_i(\lambda^*)) + f_0(P - \sum_{i=1}^{n} u_i(\lambda^*)) \tag{4.14}$$

On the other side, since $\lambda^* = -\nabla f_0(P - \sum_{i=1}^{n} u_i(\lambda^*))$, then using the same reasoning as in the proof of lemma (4.1), we deduce that $(u_i(\lambda^*))_i$ is a global minimizer of f. And with (4.14), we conclude that $W(\lambda^*) = \min_{u \in U} f(u)$, which means that there is no duality gap. □

With this lemma, we deduce that when the sequence of iterations obtained by our hybrid algorithm converges then there is no duality gap. So the hybrid algorithm has the hope to work well only for problems that present no duality gap, which can also be the case for some problems with non-convex admissible sets.

Corollary 4.3. *If the sequence of iterations* $(u^k)_k$ *of the hybrid algorithm converges (to some* u^**) then the problem has no duality gap.*

Proof. We tend k to ∞ in the inequality (4.9), and we obtain that $u_i^* \in \underset{u_i \in U_i}{\operatorname{argmin}} f_i(u_i) + \langle \lambda^*, u_i \rangle$ where $\lambda^* = -\nabla f_0(P - \sum_{i=1}^{n} u_i^*)$, an by taking $u(\lambda^*) = u^*$ we deduce that $\lambda^* = -\nabla f_0(P - \sum_{i=1}^{n} u_i(\lambda^*))$. And by the lemma (4.2), we conclude that there is no duality gap. □

4.4 Regularization

In the simulations for our problem of controlling electric hot water tanks, we remark that adding a term of regularization of the form $\frac{g}{2}||u||^2$, $g > 0$, to the cost function $f(u)$, allows the Lagrangian decomposition to make better performances.
So adding this regularization term to Lagrangian Decomposition (LD) leads to the following regularized Lagrangian Decomposition (gLD):

1. $u_i^{k+1} \in \underset{u_i}{\operatorname{argmin}} f_i(u_i) + \frac{g}{2}||u_i||^2 + \langle \lambda^k, u_i \rangle$
2. $v^{k+1} \in \underset{v}{\operatorname{argmin}} f_0(v) + \langle \lambda^k, v \rangle$

3. $\lambda^{k+1} = \lambda^k + \rho_k(v^{k+1} + \sum_{i=1}^{n} u_i^{k+1} - P)$

And adding this regularization term to Hybrid Decomposition (HD) leads to the following regularized Hybrid Decomposition (gHD): At iteration $k+1$, go from $i=1$ to $i=n$

1. $\lambda_i^{k+1} = (1-r_k)\lambda_i^k - r_k \ \nabla f_0(P - \sum_{j=1}^{i-1} u_j^{k+1} - \sum_{j=i}^{n} u_j^k)$

2. $u_i^{k+1} = \underset{u_i \in U_i}{\operatorname{argmin}} \ f_i(u_i) + \frac{g}{2}||u_i||^2 + \langle \lambda_i^{k+1}, u_i \rangle$

In the last month of the internship, we founded in the literature a new algorithm presented by **Xu and Yin** ([13],2017). This algorithm include our idea of using a linearization of f_0 with ∇f_0, and adds a proximal term of the form $\frac{\nu L_k}{2}||u_i - u_i^k||$ to the subproblems. We precise that we didn't perform simulations with this algorithm since it was discovered at the end of the internship. But we will present simulations using a quadratic term $\frac{g}{2}||u||^2$.
We will present in the following this algorithm of Xu and Yin. The more general objective function they consider is of the form:

$$\min_{u_i \in U_i, i=1,\cdots,n} f(u_1, \cdots, u_n) = f_0(u_1, \cdots, u_n) + \sum_{i=1}^{n} f_i(u_i)$$

With U_i that can be convex or not. And if we consider a cyclic rule in the choice of the block i to update, their algorithm is of the form:

$$u_i^{k+1} \in \underset{u_i}{\operatorname{argmin}} \ \langle \nabla_{u_i} f_0(u^k), u_i \rangle + \frac{\nu L_k}{2}||u_i - u_i^k||^2 + f_i(u_i)$$

where $\nabla_{u_i} f_0(u^k)$ is the gradient of the function $u_i \to f_0(u_{\neq i}^k, u_i)$ at $u_i = u_i^k$, $\nu > 1$ and L_k being the Lipschitz constant of the function $u_i \to \nabla_{u_i} f_0(u_{\neq i}^k, u_i)$ (supposed to be Lipschitz).
In this article of Xu and Yin [13], it's shown that under some assumptions, in particular assuming the Kurdyka–Łojasiewicz property on f (that contains also the indicator function of the admissible set U), the hole sequence u^k generated by this algorithm converges to a critical point u^* of f, i.e. satisfying $0 \in \partial f(u^*)$.

We will present now an advanced block minimization algorithm given by Razaviyayn et al. ([10],2013), this algorithm is called the Block Successive Upper-bound Minimization Algorithm (BSUM).

We will present the BSUM algorithm. By using a cyclic rule in the choice of the block (go from $i=1$ to $i=n$), we have at iteration $k+1$:

$$u_i^{k+1} \in \underset{u_i \in U_i}{\operatorname{argmin}} \ \phi_i(u_i, u^{\text{previous}}) \quad \text{, with } u^{\text{previous}} = (u_1^{k+1}, \cdots, u_{i-1}^{k+1}, u_i^k, \cdots, u_n^k) \tag{4.15}$$

The approximation functions $\phi_i(\cdot,\cdot)$, $i = 1, \cdots, n$ has to satisfy the following assumptions:

$$\phi_i(y_i, y) = f(y), \qquad \forall y \in U \tag{4.16}$$

$$\phi_i(u_i, y) \geq f(y_1, \ldots, y_{i-1}, u_i, y_{i+1}, \ldots, y_n), \qquad \forall u_i \in U_i, \forall y \in U \tag{4.17}$$

$$\phi_i'(u_i, y; d_i)\Big|_{u_i = y_i} = f'(y; d), \qquad \forall d = (0, \ldots, d_i, \ldots, 0) \text{ with } y_i + d_i \in U_i \tag{4.18}$$

$$\phi_i(u_i, y) \text{ is continuous in } (u_i, y) \tag{4.19}$$

In the article of Razaviyayn et al. [10], it's shown that under some assumptions, in particular assuming $\phi_i(u_i, y)$ to be quasi-convex in u_i, and that the subproblems (4.15) have unique solutions for every $u^{previous} \in U$, every limit point u^* of the iterates generated by the BSUM algorithm is a stationary point, i.e. $f'(u^*; d) \geq 0$ for all d such that $u^* + d \in U$.

An important point is that we remarked that the algorithm of Xu and Yin can be seen as a particular case of the BSUM, especially when $L_k = L$ didn't depend on the iteration k in the algorithm of Xu and Yin.
To show that we have to take:

$$\phi_i(u_i, y) = \langle \nabla_{u_i} f_0(y), u_i - y_i \rangle + \frac{\nu L}{2} ||u_i - y_i||^2 + f_i(u_i) + \sum_{j \neq i} f_j(y_j) + f_0(\sum_{j=1}^{n} y_j)$$

So verify easily that the condition 4.16, 4.18, and 4.19 holds. And to verify the condition 4.17, we use the fact that $y_i \to \nabla_{u_i} f_0(y_{\neq i}, y_i)$ is L-Lipschitz, to have

$$f_0(u_i + \sum_{j \neq i} y_j) \leq f_0(\sum_{j=1}^{n} y_j) + \langle \nabla_{u_i} f_0(y), u_i - y_i \rangle + \frac{L}{2} ||u_i - y_i||^2 \tag{4.20}$$

And since $\nu > 1$ we deduce that the condition 4.17 also holds. So we conclude that in this case the algorithm of Xu and Yin [13] (2017) can be seen as a particular case of the BSUM algorithm of Razaviyayn et al. [10] (2013).

Now, we will show that the property of our hybrid decomposition algorithm given by lemma (4.1) is specific to this algorithm by presenting a particular problem for which the BM algorithm and the algorithm of Xu and Yin can lead to a sequence of iterates that converges to a point that is not a global minimum of the cost function.

A particular problem (counter example)

In this particular problem, we consider $n = 2$, $U_1 = \{-1, 0\}$ and $U_2 = \{0, 2\}$, we take $f_1 = f_2 = 0$, $P = 1$ and $f_0(v) = v^2$, so the problem can be written as:

$$\min_{u_1 \in \{-1,0\}, u_2 \in \{0,2\}} (u_1 + u_2 - 1)^2$$

We can see easily that the only global minimizer of this problem is $(u_1^*, u_2^*) = (-1, 2)$.
We can also easily show that the Block Minimization (BM) algorithm and the algorithm of Xu and Yin can lead both of them to the following sequence of iterates $(u_1^k, u_2^k) = (0, 0)$, for all $k \geq 0$, which converges to $(0, 0)$ but this is not a global minimizer of f.
This proves that the BM algorithm and the algorithm of Xu and Yin can lead to convergent sequence but it's limit is not a global minimizer.

Advantages of the hybrid decomposition

From the previous results, we deduce that the first main advantage of the hybrid decomposition algorithm is that when it converges then it converges towards the a global minimum of the cost function, which not the case of the other algorithms found in the literature. The other advantages is that in our specific application for EHWT, the functions f_i, $i = 1, \cdots, n$, are linear which means that all the local optimization problems solved $\underset{u_i \in U_i}{\operatorname{argmin}} f_i(u_i) + \langle \lambda_i^{k+1}, u_i \rangle$ are linear problems which make them easier to solve. We will show that in the next part of this report that presents the results of our simulations.

5 Simulations

According to the previous chapters, we have now three algorithms that we will compare with simulations. The first algorithm is the Lagrangian Decomposition (LD) algorithm. The second algorithm is the Block Minimization (BM) algorithm. And the third algorithm is the Hybrid Decomposition (HD) algorithm that was presented in the previous chapter as the second hybrid algorithm.
In this chapter, we will present and compare the simulations obtained by these three algorithms for different numbers of tanks. The tanks parameters correspond to an Atlantis ATLANTIC VMRSEL 200L water tank. The computations were made using the CPLEX 12.6 commercial solver and the coding was made by using MATLAB.
We notice that for example with $n = 100$ tanks, the size of the problem is such that, there is 33600 variables to find and 76800 constraints.

Rescaling the problem

In the simulations, we want to vary the number of tanks. In order to balance the terms of the objective function when n becomes large, we don't want that a some term of the objective function dominates. So we consider the problem with the average consumption $\bar{u} = \frac{1}{n}\sum_{i=1}^{n} u_i$, $\bar{u^2} = \frac{1}{n}\sum_{i=1}^{n} u_i^2$, and the average target $\bar{P} = \frac{P}{n}$. So we want that the problem is written as following:

$$\sum_{i=1}^{n}(\langle c, \bar{u}\rangle + \frac{\gamma_0}{2}(\bar{P} - \bar{u})^2$$

Which leads to

$$\min_{u_i \in U_i} \sum_{i=1}^{n}(\langle c, u_i\rangle) + \frac{\gamma_0}{2n}(P - \sum_{i=1}^{n} u_i)^2$$

Therefore, we choose γ to be inversely proportional to the number of tanks n : $\gamma = \frac{\gamma_0}{n}$.

The stopping criteria

We define a stopping criteria based on the costs $f(u^k)$ obtained for the primal problem after each iteration. We stop the algorithm when the decrease in the cost is less than a given value η. So the stopping criteria is:

$$f(u^{k+1}) - f(u^k) < \eta$$

The idea is that we will stop when we estimate that the gain we realize with each additional iteration $f(u^{k+1}) - f(u^k)$ is less than the cost of computation η needed to perform this iteration.
Since we will vary the number n of tanks, we choose the constant value to be proportional to n : $\eta = \eta_0 \times n$.
We fix also a maximum number of iterations K that cannot be exceeded to stop the algorithms in case they don't converge.

Parameters

In all the simulations we choose the penalization parameter $\gamma_0 = 50000$, the regularization parameter $g = 100$, the stopping criteria constant $\eta_0 = 0.1$, the weighting parameter $r_k = \frac{0.002}{\sqrt{k}}$ for the LD algorithm (which equivalent to saying that the diminishing stepsize in the Lagrangian decomposition algorithm is $\rho_k = r_k \times \gamma = \frac{100/n}{\sqrt{k}}$) and $r_k = \frac{0.003}{\sqrt{k}}$ for the HD algorithm. The time step is equal to half an hour so that we have $p = 48$ time steps in a 24h period of simulation.

5.1 Simulations for the algorithms BM, LD and HD

We will make the simulations for different numbers of tanks $n = 5$, 10, 20, 50, and 100. And for every value of n, we generate 50 different admissible sets U. So we have 50 problems that we solve with the three algorithms BM, LD and HD. Let the value of n be fixed, to show the performances of some algorithm over the 50 problems we present the evolution of the cost obtained by this algorithm at each iteration. At each iteration, we present the 50 values given by the algorithm with a box of the following form

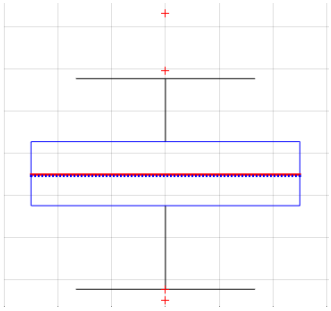

Figure 3: Form of the box presenting 50 simulations at each iteration.

This presentation is such that the central box (in blue) contains 50% of the points, the two bars (in black) delimit 90% of the points, and above the upper bar, we present the highest 5% of the points (in red), and bellow the lower bar, the smallest 5% of the points are presented (in red). The red bar presents the mean of the 50 values, and the blue dotted bar presents their median.

Simulations with $n = 5$ tanks

We begin our simulations with $n = 5$ tanks. The following figure (4) shows the cost values obtained over the iterations by the three algorithms, we retrieve the dual lower bound for the three algorithms. We notice that the dual lower bound is obtained by the maximization of the dual function done by the LD algorithm. We notice that in the following figures BM is always presented in blue, LD is always presented in red and HD is always presented in black.
In order to have a clear figure we aggregate every 4 neighbor iterations, and we place the boxes presenting the three algorithms shifted relative to each other.

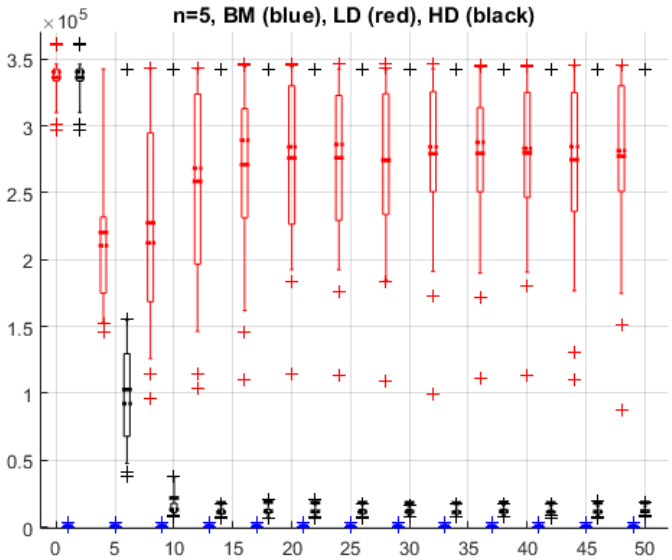

Figure 4: Algorithms performances for $n = 5$

First, we remark that the LD algorithm didn't converge and is far from the optimum. We notice also that the block minimization (BM) algorithm decreases very rapidly in his first iteration, so that in the previous figure BM is close to zero (which means that the cost given by BM is close to the maximum of the dual function and therefore to the minimum of the problem). However after some iterations the HD algorithm also becomes close to zero. In order to show more accurately these two algorithms we make a zoom (figure 5) on the figure with a focus on the two interesting algorithms BM and HD:

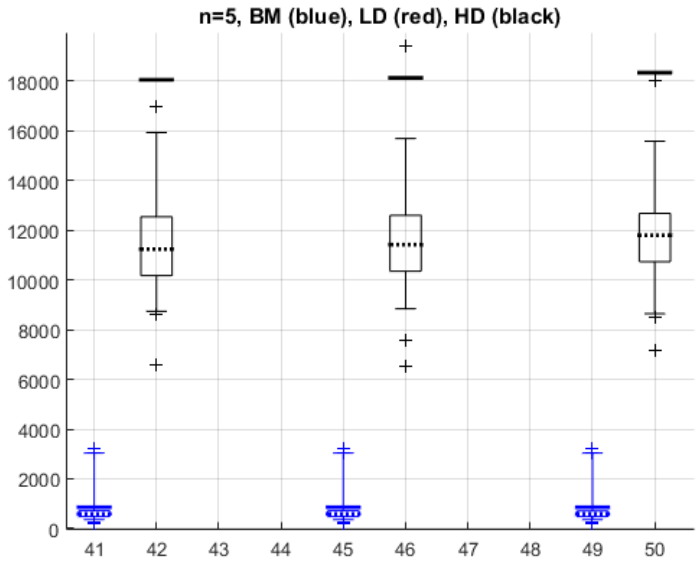

Figure 5: Zoom on the algorithms performances for $n = 5$

We remark from this figure that the algorithm BM gives better performance than the algorithm HD for $n = 5$.

Simulations with $n = 10$ tanks

Now we show the performances of the three algorithms for $n = 10$ tanks in the following figure (6):

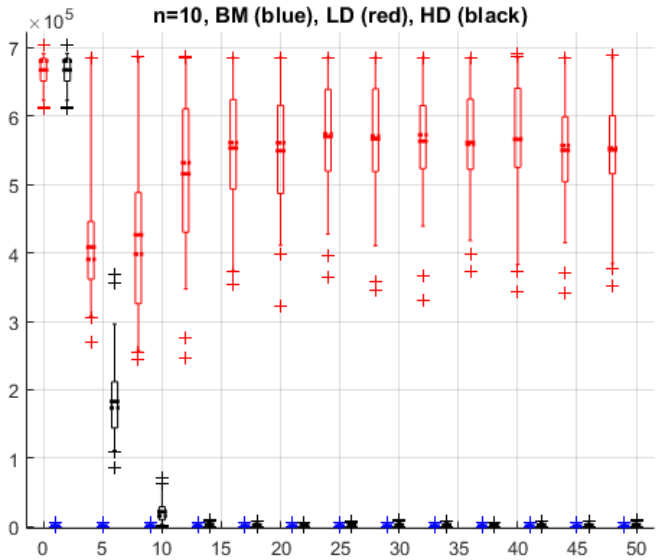

Figure 6: Algorithms performances for $n = 10$

We remark that the LD algorithm didn't converge and is far from the optimum as for $n = 5$. We notice also that the BM algorithm decreases a lot at its first iteration but the HD algorithm also decreases significantly after few iterations. We make a zoom at the performances of these two interesting performances of the algorithms BM and HD (figure 7):

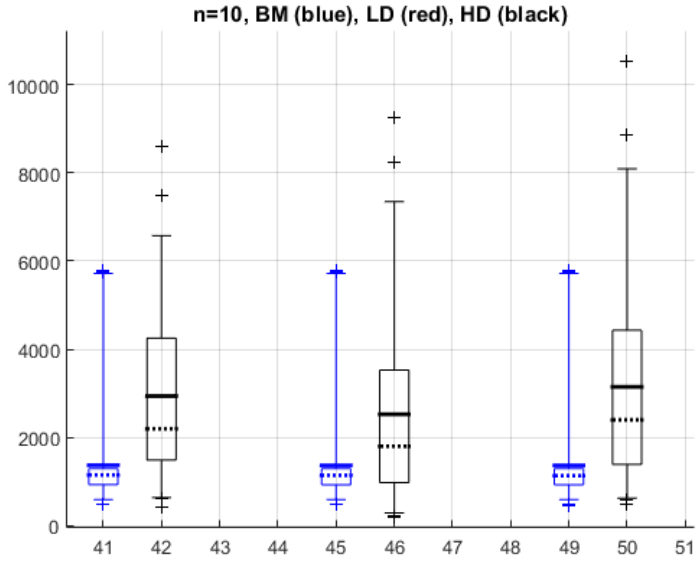

Figure 7: Zoom on the algorithms performances for $n = 10$

We notice also that the block minimization (BM) algorithm is still better than the hybrid algorithm (HD) for $n = 10$.

Simulations with $n = 20$ tanks

Now we show the performances of the three algorithms for $n = 20$ tanks in the following figure (8):

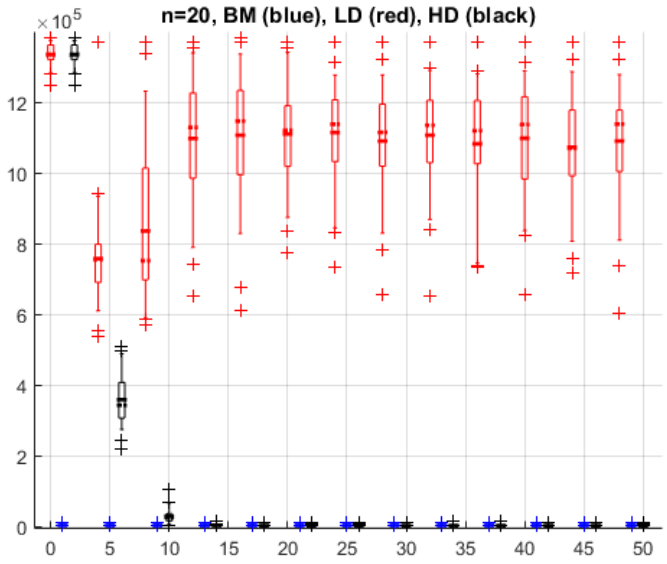

Figure 8: Algorithms performances for $n = 20$

We remark that the LD algorithm didn't converge and is far from the optimum as for the precedent cases. We make a zoom at the performances of the two interesting performances of the algorithms BM and HD (figure 9):

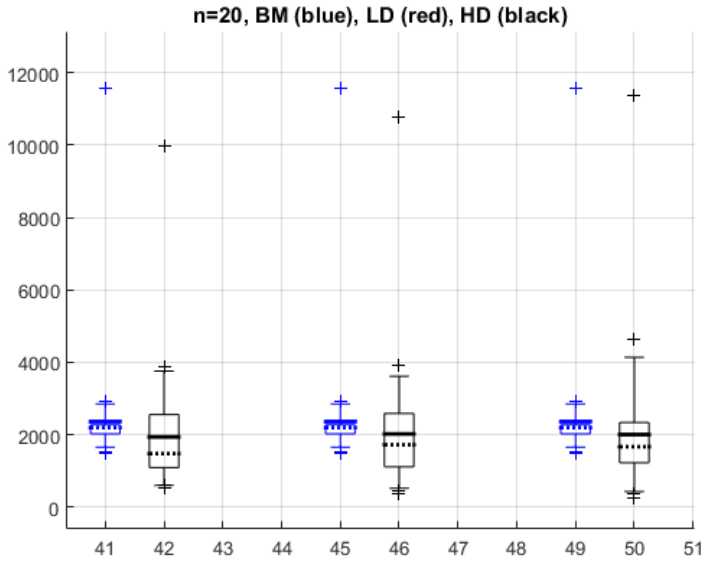

Figure 9: Zoom on the algorithms performances for $n = 20$

We notice that for $n = 20$, the two algorithms BM and HD have comparable performances.

Simulations with $n = 50$ tanks

Now we show the performances of the three algorithms for $n = 50$ tanks in the following figure (10):

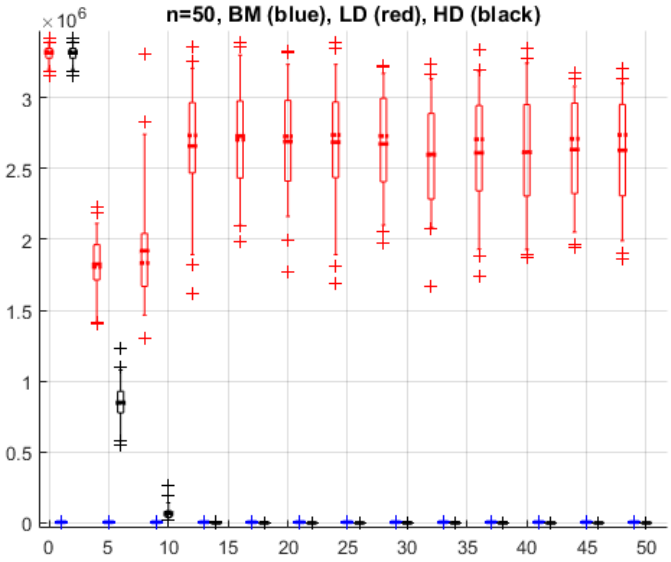

Figure 10: Algorithms performances for $n = 50$

We remark that the LD algorithm didn't converge and is far from the optimum as in the precedent cases. We make a zoom at the performances of the two interesting performances of the algorithms BM and HD (figure 11):

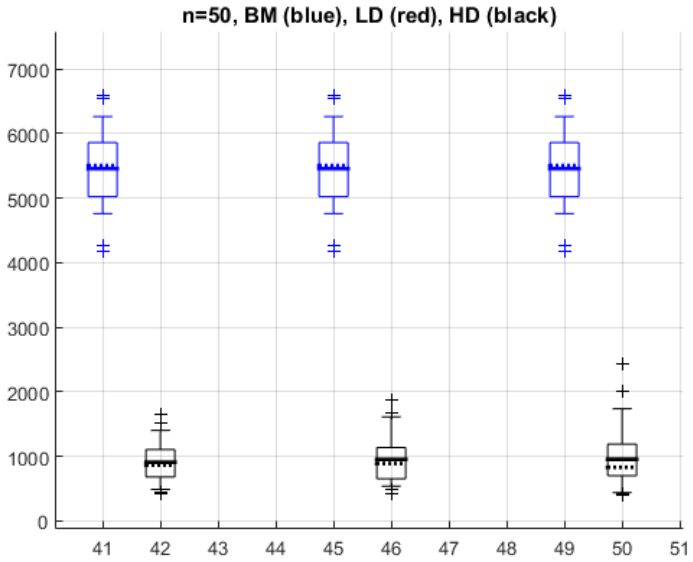

Figure 11: Zoom on the algorithms performances for $n = 50$

We notice that for $n = 50$, the HD algorithm becomes clearly better than the BM algorithm.

Simulations with $n = 100$ tanks

Now we show the performances of the three algorithms for $n = 100$ tanks in the following figure (12):

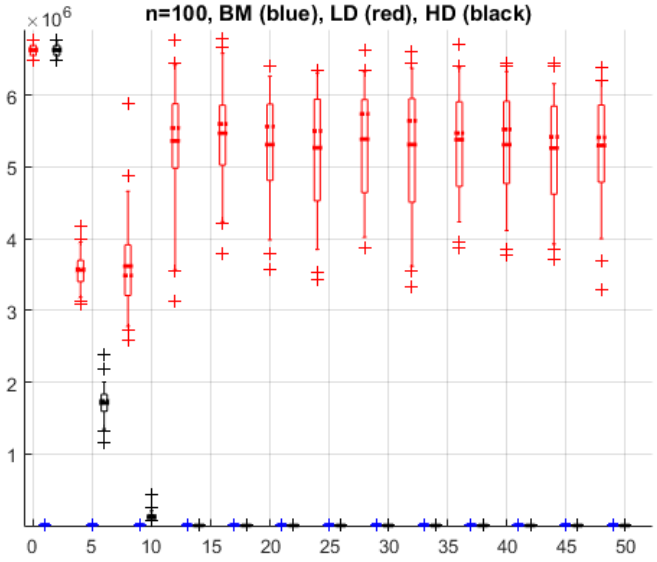

Figure 12: Algorithms performances for $n = 100$

We remark that the LD algorithm didn't converge and is far from the optimum as in the precedent cases. We make a zoom at the performances of the two interesting algorithms BM and HD (figure 13):

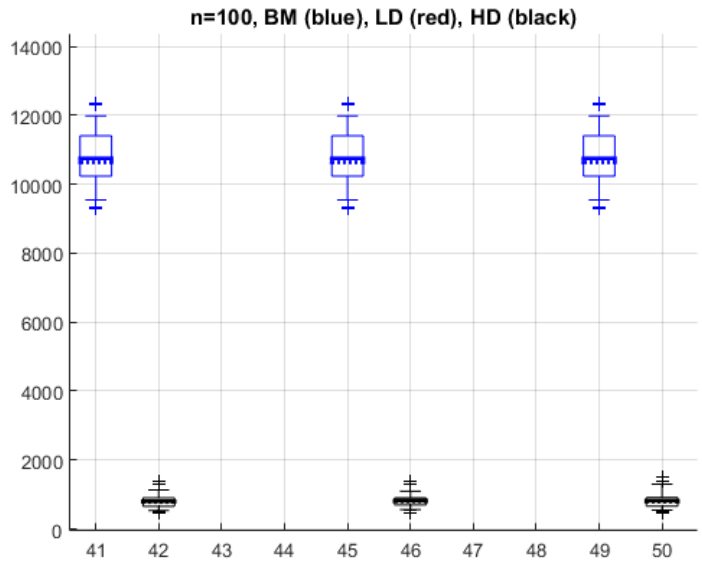

Figure 13: Zoom on the algorithms performances for $n = 100$

We notice also that for $n = 100$, the HD algorithm is clearly better than the BM algorithm.

To summarize the precedent results, we notice that the Lagrangian Decomposition algorithm (LD) didn't converge and is far from the optimum for different values of n.
We remarked also that the algorithms BM and HD are giving for each value of n interesting performances with a small gap to the dual lower bound of the problem. In practice BM decreases a lot in his first iteration and after that it decreases more slowly. And we notice that for small numbers of tanks $n < 20$, the algorithm BM has better performances than our hybrid algorithm HD, and that for high numbers of tanks $n > 20$, it's our algorithm HD that gives better performances.
We remark also that the HD algorithm has a globally decreasing shape contrary to the LD algorithm, which means that HD has inherited a part of the decreasing property from the BM algorithm.

5.2 Simulations with the regularized algorithms gLD and gHD

Now we will show the performances of the regularized Lagrangian Decomposition (gLD) and the regularized Hybrid Decomposition (gHD) that we obtain from the LD and HD algorithms by adding the quadratic term $\frac{g}{2}||u||^2$ to the cost function of the problem as described in section 4.4. We will show the simulations for $n = 5$, 10 and 20.

Simulations with $n = 5$ tanks

We begin our simulations with $n = 5$ tanks. The following (figure 14) shows the cost values obtained over the iterations by the two algorithms gLD (in red) and gHD (in black).

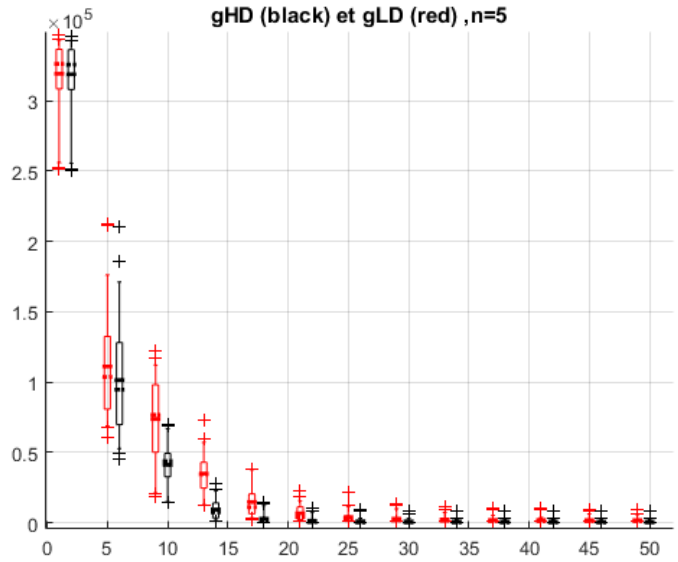

Figure 14: Regularized algorithms performances for $n = 5$

We notice that the two algorithms gLD and gHD are giving interesting performances. In order to show more accurately these performances, we make a zoom on the last iterations (figure 15):

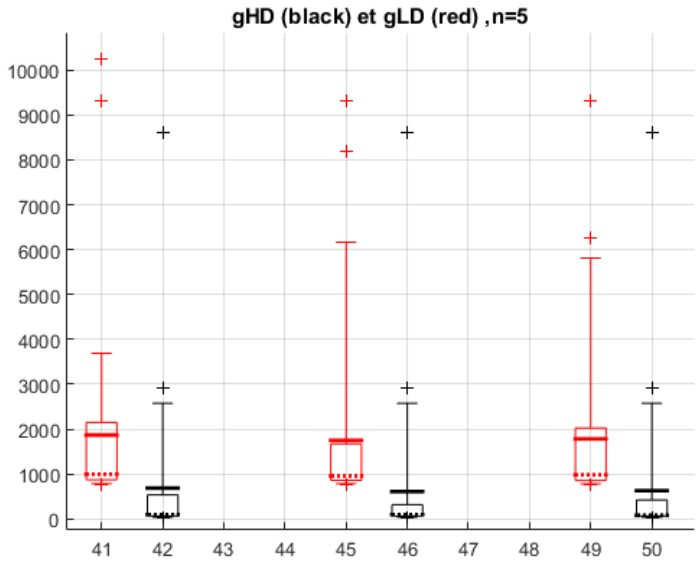

Figure 15: Zoom on the regularized algorithms performances for $n = 5$

We remark that the algorithm gHD is giving better performances than the algorithm gLD for $n = 5$ tanks.

Simulations with $n = 10$ tanks

Now we show the performances of the two algorithms gLD and gHD for $n = 10$ tanks in the following (figure 16):

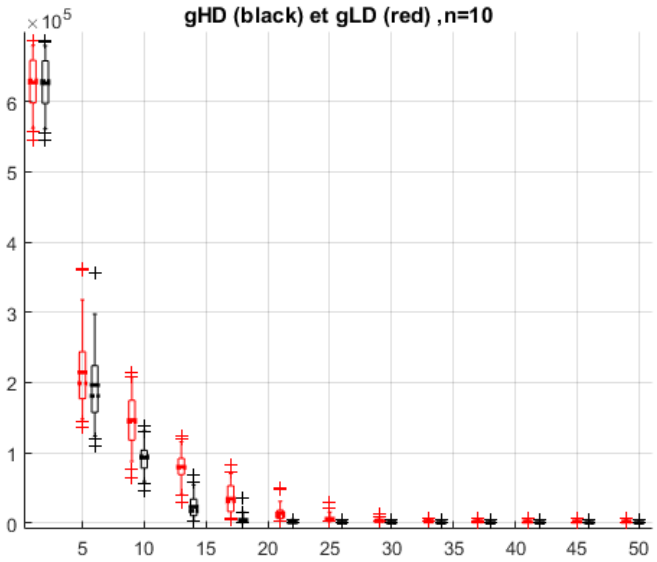

Figure 16: Regularized algorithms performances for $n = 10$

We make a zoom at the performances of these two algorithms gLD and gHD (figure 17):

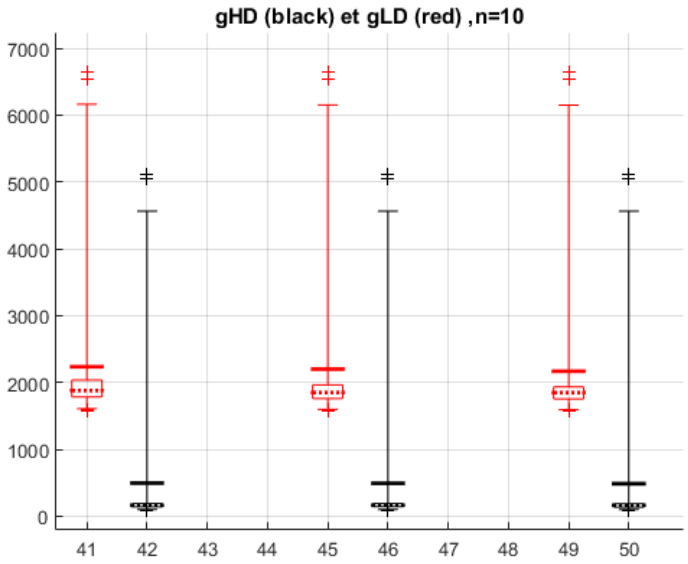

Figure 17: Zoom on the regularized algorithms performances for $n = 10$

We remark that the algorithm gHD is also giving better performances than the algorithm gLD for $n = 10$ tanks.

Simulations with $n = 20$ tanks

Now we show the performances of the two algorithms gLD and gHD for $n = 20$ tanks in the following (figure 18):

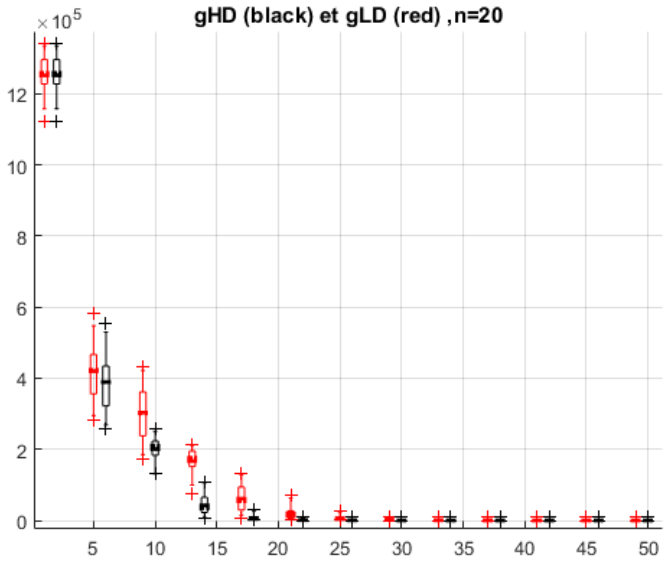

Figure 18: Regularized algorithms performances for $n = 20$

We make a zoom at the performances of these two algorithms gLD and gHD for $n = 20$ (figure 19):

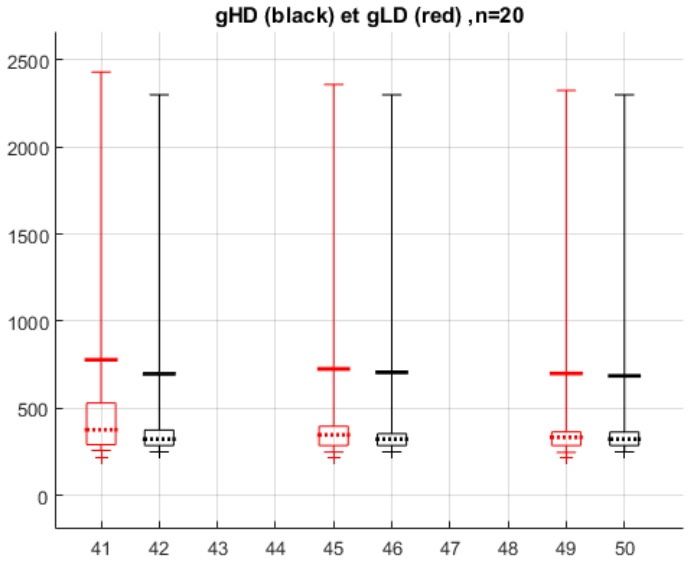

Figure 19: Zoom on the regularized algorithms performances for $n = 20$

We notice that for $n = 20$, the two algorithms gLD and gHD have comparable performances.

We remark also that the performances of the algorithm gLD become better when the number of tanks n is big. This can be explained by the intuition that when the number of tanks is big the admissible set become 'close' to be a convex set.

5.3 Summary of the performances of the algorithms

In the following table 2 we summarize the best performances achieved by the different algorithms for each number of tanks $n = 5$, 10, 20, 50 and 100. We indicate by "cv" that the algorithm is converging in simulations to the optimum. We indicate with a red color the algorithms with the best performances. We indicate by "ncv" that the algorithm is not converging in the simulations. And if the simulations was not performed we indicate it by a hyphen.

Algorithm	$n = 5$	$n = 10$	$n = 20$	$n = 50$	$n = 100$
HD	*cv*	*cv*	*cv*	*cv*	*cv*
gHD	*cv*	*cv*	*cv*	*cv*	–
LD	*ncv*	*ncv*	*ncv*	*ncv*	*ncv*
gLD	*cv*	*cv*	*cv*	–	–
BM	*cv*	*cv*	*cv*	*cv*	*cv*

Table 2: Summary of the best algorithms.

So the hybrid algorithm HD (and his regularized form gHD) present the best performances compared to the classical algorithms block minimization (BM) and Lagrangian decomposition (LD) and his regularized form (gLD).

6 Conclusion and perspectives

In this work, we are trying to solve large-scale optimization problems that can be applied either to the problem of optimal control of electrical hot water tanks or to the problem of management of electricity production in the short term (Unit Commitment problem) since they can be formulated in the same way. The problems we are dealing with present two main challenges. The first one is the presence of a cost that couples the variables. This cost presents the cost of deviation of the aggregated consumption from a given target in the case of controlling hot water tanks, and it presents the cost of not matching the demand in the unit commitment problem. The second challenge that present these problems is the non-convexity of the constraint sets that define the operating programs constraints of each local user or producer.
In order to get rid of high computational costs due to high dimensions of the problem and of the coupling term, we are using distributed optimization methods. In these methods, we decompose the global problem into sub-problems in a decentralized system. By this way, each decentralized local user solve his own problem locally. This allows to have less computational cost, and the user is the only one that has access to his constraints which lets more privacy.
In this work, we consider two classical distributed optimization methods that are the Lagrangian decomposition algorithm and the block minimization algorithm. The first one uses a (unique) price decomposition and appears to be very efficient when there is no duality gap in the problem and when the dual function is differentiable. So that in the case of non-convex constraints sets, where the dual function is potentially non-differentiable, this Lagrangian method faces difficulties in converging to the minimal cost. The second method (block minimization) has the good property of giving decreasing costs over the iterations even if the constraints sets are non-convex, but it seems to decrease slowly after his first iteration.In order to go beyond these two methods and to achieve interesting optimization of our problem even in the case of non-convex sets of constraints, we propose in this work a new algorithm that aims to hybridize the two previous methods with the aim to go beyond their limits. This algorithm takes the idea of using a price decomposition from the Lagrangian decomposition but by using different prices, and takes the update of these prices by inspiration from the block minimization method.
As seen in the simulations, this hybrid algorithm gives in all the examples interesting performances by being always close to the optimum contrary to the Lagrangian decomposition that can be very far from the optimum due to the non-convexity of the admissible sets. Moreover, in most of the cases we remark that the hybrid algorithm achieves better performances than the block minimization algorithm. In addition to that, we remark that the hybrid algorithm gives a globally decreasing costs contrary to the Lagrangian method, which means that it inherited this property from the block minimization.
In the future works, it's possible to pursue with the following points:

- Apply these three algorithms to the problem of Unit Commitment that has the same formulation as the problem of controlling electrical hot water tanks, but with different sets of constraints. This will allow us to evaluate the performance of the hybrid algorithm and compare it to the two classical methods, in order to know if the good properties of the hybrid algorithm are intrinsic or are related to the specific problem of electric hot water tanks.

- Pursue the work on the theoretical aspects of the hybrid algorithm. In particular, we want to find some results about the decreasing property that we noticed during the simulations. It's also possible to think about a hybrid algorithm where the approximation of the function f_0 can be done differently, for example by adding some quadratic term.

7 Acknowledgment

The internship opportunity I am having with my supervisor Nadia Oudjane at EDF Paris-Saclay LAB is a great chance for learning and for my professional development. I am grateful to her, many thanks for all our constructive discussions. I thank also the researchers of the OSIRIS department

at EDF Paris-Saclay LAB: Olivier Beaude, Arnaud Lenoir and Olivier Juan for their interest on the subject and all the discussions we had. I thank Nathanaël Beeker-Adda for his help and also Dilek Savas for their previous works on the subject.

Appendices

A Optimal Control of Electrical Hot Water Tanks

In this annexe, we will present the physical model that allows to describe the discrete-time optimal control of Electrical Hot Water Tank (EHWT). This model was proposed by Nathanael Beeker [2] and [3]. We will present some general considerations on EHWT and the modeling assumptions.

A.1 General considerations

A typical electric hot water tank has a vertical cylindrical shape and is filled with water. A heating element, which converts electricity into heat through an electrical resistance is plunged at the bottom of the tank. Hot water is drained from the top while cold water is injected at the bottom at exactly the same flow-rate. Hence the tank is always full (see Figure 20).

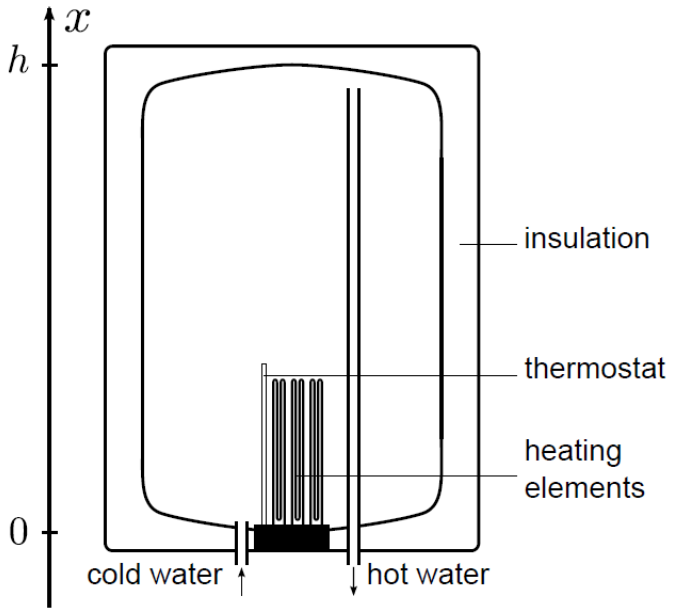

Figure 20: Schematic view of an electric hot water tank

The heating element is at the bottom of the tank. Layers of water with various temperatures coexist inside the tank. These layers are mixed only by heat diffusion. There is a non uniform quasi-equilibrium temperature profile which is increasing with height in the tank. This quasi-equilibrium is called *stratification*. Thanks to the stratification the heating element at the bottom is able to heat the cold water, while the hot water near the outlet of the tank is available for consumption. We can assume that the temperature is homogeneous at each height and continuously increasing function of height because of the stratification effect and the cylindrical symmetry of the tank (see Figure 21).

Temperature Profile: We assume that the cold water injected at the bottom has a constant temperature T_{in}, which represents the lower bound of the temperature profile. The heating process is driven by turbulence generated by buoyancy property during the heating process, which is the cause of a local mixing at the bottom of the tank. We consider that this mixing does not affect the temperature profile in the upper part of the tank which has a higher temperature. The user can specify a temperature T_{max} at which the heating has to be stopped to prevent overheating. Also the user can set a comfort temperature T_{com} for his comfort. Water having temperature higher than T_{com} is also useful because it can be blended with cold water to reach T_{com}, on the other hand

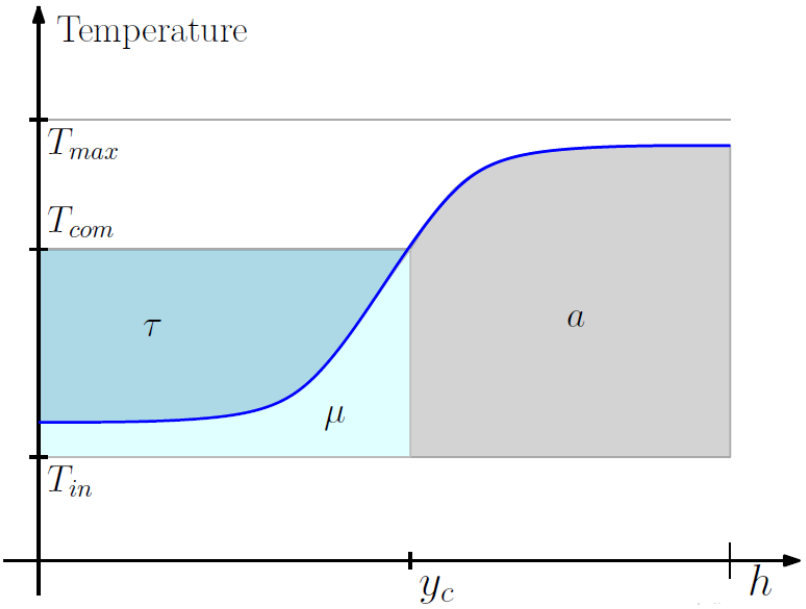

Figure 21: Temperature Profile, Available, Delay and Reserve Energy after a Drain

water having temperature lower than T_{com} is useless.
In order to simplify the system, we define a few variables of interests which are called state variables, to avoid the control of the entire temperature profile (see Figure 21).
The available energy a : it is defined as the energy contained in the zone having temperature greater than the comfort temperature T_{com}. If a reaches the value 0 and a water drain is applied, it means that the user is trying to consume hot water when none is available, and therefore that the comfort constraints are broken.
The delay energy τ : itis defined as the energy required by the system to reach the temperature T_{com}.
The reserve energy μ : it is defined as the energy contained in the tank under T_{com} that is currently unavailable for consumption. When, during the heating process, τ reaches the value 0, the energy μ becomes available to consumption. This generates an immediate increase of a and decrease of μ up to 0. This is a source of a discontinuity aspect on the model of the electric hot water tank which leads to non-convex admissible sets on which we perform the optimization.
A drain is mainly characterized by a decrease of a and an increase of τ with a insignificant raise of μ due to an energy transfer from a.
Note that h is the height of the tank, S its cross-section and $T(\cdot)$ is the increasing temperature profile of the water defined on $[0; h]$. Then, the definitions yield the following expressions of a, τ , μ as:

$$a = S\rho c_p \int_{y_c}^{h} T(y)dy$$
$$\tau = S\rho c_p \int_{0}^{y_c} (T_{\text{com}} - T(y))dy$$
$$\mu = S\rho c_p \int_{0}^{y_c} T(y)dy$$

where ρ and c_p are the density and the heat capacity of water, respectively, and y_c is defined as $y_c := \min\{y \mid T(y) = T_{\text{com}}\}$.
Hot water consumption $d = (d_t)_{t=0,\cdots,p-1}$ is an (uncontrolled) input of our problem and represented by drain in the system. For his comfort, the user consumes certain quantities of energy each hour. On the other hand, the heat injected via the heating element in the tank is a control variable $u = (u_t)_{t=0,\cdots,p-1}$. The control have a number of objectives. The most obvious one is the cost reduction for a single unit in response to a price signal and the second one is reaching a load profile for the aggregated consumption in which the tank participates.

A.2 Discrete-Time Dynamics

We establish a discrete-time model for the dynamics of the triplet $x = (a, \tau, \mu)$ over a finite horizon which is discretized into uniform time-steps $[0, \cdots, p-1]$. Note that $(a_0, \cdots, a_{p-1})$, $(\tau_0, \cdots, \tau_{p-1})$ and $(\mu_0, \cdots, \mu_{p-1})$ respectively state the values of a, μ, τ at each of these time-steps.
At each time-step $t \in [0, \cdots, p-1]$, energy is consumed by the user via draining an amount d_t that is already known. Via the heating element, an amount of energy $u_t \in [0, u_{\max}]$ is introduced. We divide this energy into two parts v_t and w_t representing respectively the part introduced in a_t and μ_t . A flow of energy is defined with the variable ϕ_t from μ_t to a_{t+1}.
For any t, the dynamics of x_t is given by the energy balance,

$$a_{t+1} = (1-p)a_t \quad -\alpha d_t \quad +v_t + \phi_t \tag{A.1}$$
$$\tau_{t+1} = \tau_t + p\mu_t \quad +\beta d_t \quad -w_t \tag{A.2}$$
$$\mu_{t+1} = (1-p)\mu_t \quad -(1-\alpha)d_t \quad +w_t - \phi_t \tag{A.3}$$

The heat losses, modeled with an exponential decay, are characterized by the erosion of a and μ at a rate p, the energy from the later contributing to a raise of τ (see Figure 21). The energy consumed by the user during one time step is split between a and μ with a coefficient α, $1-\alpha$ and affects τ with a coefficient β. Using the model presented in Beeker et al. [1] for identification purposes, we set the values $\alpha \approx 1.2$ and $\beta \approx 0.4$.
The energy is injected at the bottom of the tank via the heating element. When $\tau > 0$ the heating has no impact on the available energy a, but instead tends to reduce τ and increase μ until $\tau = 0$. When the value of τ is 0 , the injected energy is added to a and then becomes immediately available. This is modeled by dividing u_t into two shares w_t and v_t representing, respectively, the part of the injected energy going into μ and a , and subject to the following conditions,

$$u_t = v_t + w_t \tag{A.4}$$
$$0 = v_t \tau_{t+1} \tag{A.5}$$
$$0 \leq v_t, w_t, u_t \leq u_{\max} \tag{A.6}$$

Given these conditions, if $\tau_{t+1} > 0$ then no energy can be introduced in a_{t+1}. Moreover $v_t = 0$ and $w_t = u_t$, and if $\tau_{t+1} = 0$ the value of w_t and energy drain d_t have to compensate for heat losses $p\mu_t$ in the balance equations and while the remainder is introduced in a_{t+1}.
The energy flow ϕ_t from μ_t to a_{t+1} is generally equal to 0, except when $\tau_{t+1} = 0$. Then, the value of ϕ_t is defined by the fact that all the energy μ_t suddenly becomes available. This can be described as follows,

$$0 = \phi_t \tau_{t+1} \tag{A.7}$$
$$0 \leq \phi_t \tag{A.8}$$
$$\tau_{t+1} = 0 \Rightarrow \mu_{t+1} = 0 \tag{A.9}$$

we determine some bounds to the state x_t for all t: This defines the admissible controls that allow each x_t to respect these constraints. Moreover, to ensure that the dynamics are properly defined, we assume that $a_t \geq 0$, $\tau_t \geq 0$ and $\mu_t \geq 0$ for all t.
Let $m = Sh\rho c_p(T_{\max} - T_{\text{in}})$ be the maximal energy that can be contained in the tank, and $\lambda = \frac{T_{\text{com}} - T_{\text{in}}}{T_{\max} - T_{\text{in}}}$. Then from the definitions of a, τ and μ, we have $a_t \leq m$, $\tau_t \leq \lambda m$ and $\mu_t \leq \lambda m$ for all t.
And from physical constraints on the total energy we get,

$$\lambda a_t + \tau_t + \mu_t \leq \lambda m$$
$$\lambda m \leq a_t + \tau_t + \mu_t$$

We define $\Omega = \left\{ (a, \tau, \mu) \in \mathcal{R}_+^3 | \lambda a_t + \tau_t + \mu_t \leq \lambda m \leq a_t + \tau_t + \mu_t \right\}$.
Then for all t, x_t is subject to the constraint

$$x_t \in \Omega \tag{A.10}$$

This ensures that no energy is drained more than the tank can provide and the tank is not overheated.
Admissible Controls: For given $(a_0;\tau_0;\mu_0) \in \Omega$ and $d = (d_0, \cdots, d_{p-1}) \in \mathbb{R}_+^p$ and for chosen control sequence $d = (u_0, \cdots, u_{p-1})$ the relations of constraints for $t \in [0, \cdots, p-1]$ uniquely define $(a, \tau, \mu, v, w, \phi)$. This allows us to define the admissible set U:

$$U(a_0, \tau_0, \mu_0, d) = \Big\{ u \in \mathbb{R}^p |\ (a, \tau, \mu, v, w, \phi) \text{ defined by (A.1)-(A.9), } \forall t \in [0, \cdots, p-1] \text{ and satisfy (A.10), } \forall t \in [0, \cdots, p] \Big\}$$

Strengthening of (A.9):
If we except the two product conditions (A.5) and (A.7), and the condition (A.9), the relations (A.1)-(A.10) are linear equalities and inequalities in the variables $(u_t, a_t, \tau_t, \mu_t, v_t, w_t, \phi_t)$ and therefore define a polytope of $\mathbb{R}^{7p}$ for each tank. In order to solve linear/quadratic program via commercial software we need to modify the condition (A.9) since its structure is not suitable for linear programming yet. Given the set

$$A = \{(\tau, \mu) \in [0, \lambda m]^2 | \tau = 0 \Rightarrow \mu = 0\}$$

then we have

$$A = [0, \lambda m]^2 \setminus \{(\tau, \mu) \in [0, \lambda m]^2 | \tau = 0, \mu > 0\}$$

and can be adapted to

$$\{(\tau, \mu) \in [0, \lambda m]^2 | M\tau \geq \mu\} \subset A$$

where $M > 0$ is a given large coefficient as seen in the Figure 22. Then instead of considering (A.9) we consider the following linear constraint:

$$M\tau_{t+1} \geq \mu_{t+1} \tag{A.11}$$

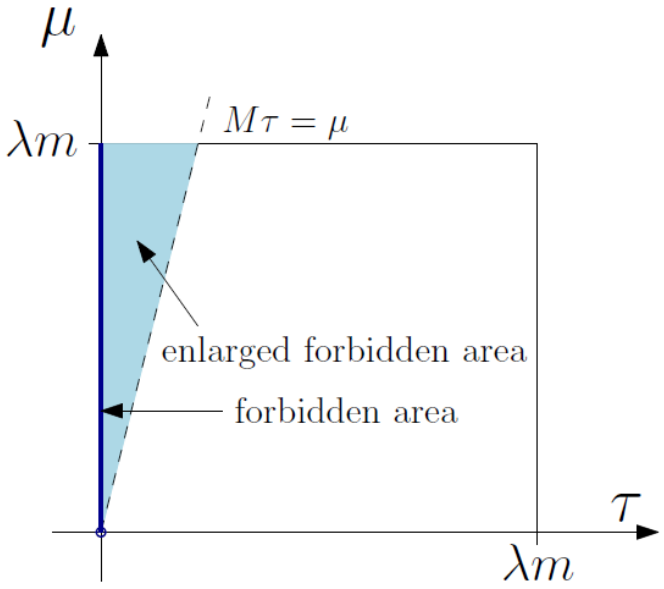

Figure 22: Strengthening of (A.9)

Given this new relation, finally we redefine the admissible set as

$$U(a_0, \tau_0, \mu_0, d) = \Big\{ u \in \mathbb{R}^p |\ (a, \tau, \mu, v, w, \phi) \text{ defined by (A.1)-(A.8),(A.10),(A.11)} \Big\} \tag{A.12}$$

The product conditions (A.5) and (A.7) contains the variables that can not be positive at the same time. We will take them into account by using SOS (Special-Ordered Set) features. This SOS constraints make our admissible sets non-convex.

Compactness of the feasible set for EHWT

We will show here that the set feasible set $U := U(a_0, \tau_0, \mu_0, d) \subset \mathbb{R}^p$ for EHWT defined above is a compact set, for any fixed choice of the initial parameters (a_0, τ_0, μ_0, d).

Lemma A.1. *The feasible set U is a compact set.*

Proof. Since we are in finite dimension $\mathbb{R}^p$, it suffices to prove that U is bounded and closed. The boundedness of U is immediate since by the constraint (A.6) we have $U \subset [0, u_{\max}]^p$. It remains to prove that U is closed.
Let $(u^k)_{k\geq 0}$ be a sequence of elements in U, such that (u^k) converges to an element $u \in \mathbb{R}^p$. We have to show that u must be an element of U.
For every k since $u^k \in U$ then by definition there exist $(a^k, \tau^k, \mu^k, v^k, w^k, \phi^k) \in \mathbb{R}^{6p}$ such that $(u^k, a^k, \tau^k, \mu^k, v^k, w^k, \phi^k)$ verifies the constraints [(A.1)-(A.8),(A.10),(A.11)]. Knowing that all the variables $a^k, \tau^k, \mu^k, v^k, w^k, \phi^k$ are nonnegative and bounded by the maximal energy m that can be contained in the tank, we have that these sequences are bounded an therefore we can extract from them converging subsequences $a^{n_k}, \tau^{n_k}, \mu^{n_k}, v^{n_k}, w^{n_k}, \phi^{n_k}$ for some strictly increasing indexes $(n_k)_{k\geq 0}$, $(a^{n_k}, \tau^{n_k}, \mu^{n_k}, v^{n_k}, w^{n_k}, \phi^{n_k}) \underset{k\to\infty}{\to} (a, \tau, \mu, v, w, \phi) \in \mathbb{R}^{6p}$.
Now in the constraints ((A.1)-(A.8),(A.10),(A.11)) applied to $(u^{n_k}, a^{n_k}, \tau^{n_k}, \mu^{n_k}, v^{n_k}, w^{n_k}, \phi^{n_k})$, we tend $k \to \infty$. We obtain directly that these constraints are also valid for the limits $(u, a, \tau, \mu, v, w, \phi)$. Therefore u is an element of U. This shows that U is a closed set. And then we deduce that U is a compact set. □

B Duality result

We consider a minimization problem of the form

$$\min_{x\in X,\ c(x)=0} f(x)$$

With $X \subset \mathbb{R}^n$ and $c : \mathbb{R}^n \to \mathbb{R}^p$. We denote $X^* = \underset{x\in X,\ c(x)=0}{\operatorname{argmin}} f(x)$ and for every $\lambda \in \mathbb{R}^p$, we denote $X(\lambda) = \underset{x\in X}{\operatorname{argmin}}\ f(x) + \langle \lambda, c(x)\rangle$.
We define also W the dual function by $W(\lambda) = \min_{x\in X} f(x) + \langle \lambda, c(x)\rangle$. We suppose that there is no duality gap, i.e. $W^* := \max_{\lambda\in\mathbb{R}^p} W(\lambda) = \min_{x\in X,\ c(x)=0} f(x) =: f^*$.
Let $x^* \in X^*$ be a solution of the primal problem, and let $\lambda^* \in \underset{\lambda\in\mathbb{R}^p}{\operatorname{argmax}} W(\lambda)$ be any maximizer of the dual function. We have:

$$\min_{x\in X} f(x) + \langle \lambda^*, c(x)\rangle = W(\lambda^*) = W^* = f^* = f(x^*) + \langle \lambda^*, \underbrace{c(x^*)}_{=0}\rangle$$

Which means that

$$x^* \in X(\lambda^*) \tag{B.1}$$

In conclusion

$$X^* \subset X(\lambda^*),\ \text{for all } \lambda^* \in \underset{\lambda\in\mathbb{R}^p}{\operatorname{argmax}} W(\lambda)$$

References

[1] Nathanael Beeker, Paul Malisani, and Nicolas Petit. Dynamical modeling for electric hot water tanks. *IFAC-PapersOnLine*, 48(11):78–85, 2015.

[2] Nathanael Beeker, Paul Malisani, and Nicolas Petit. Discrete-time optimal control of electric hot water tank. *IFAC-PapersOnLine*, 49(7):882–888, 2016.

[3] Nathanaël BEEKER-ADDA. *Modélisation et contrôle des ballons d'eau chaude sanitaire à effet Joule : du ballon individuel au parc.* 2016.

[4] Dimitri P Bertsekas. *Nonlinear programming.* Athena scientific Belmont, 1999.

[5] Stephen Boyd and Almir Mutapcic. Subgradient methods. *Lecture notes of EE364b, Stanford University, Winter Quarter*, 2007, 2006.

[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[7] Pierre Carpentier and Guy Cohen. *Décomposition-coordination en optimisation déterministe et stochastique*, volume 81. Springer, 2017.

[8] Zhi-Quan Luo and Paul Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

[9] Julie Nutini, Issam Laradji, and Mark Schmidt. Let's make block coordinate descent go fast: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence. *arXiv preprint arXiv:1712.08859*, 2017.

[10] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[11] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[12] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009.

[13] Yangyang Xu and Wotao Yin. A globally convergent algorithm for nonconvex optimization based on block coordinate update. *Journal of Scientific Computing*, 72(2):700–734, 2017.