



**HAL**  
open science

# Long term aging: an adaptative weights dynamic programming algorithm

Benjamin Heymann, Pierre Martinon, Frédéric Bonnans

► **To cite this version:**

Benjamin Heymann, Pierre Martinon, Frédéric Bonnans. Long term aging: an adaptative weights dynamic programming algorithm. 2016. hal-01349932v1

**HAL Id: hal-01349932**

**<https://hal.science/hal-01349932v1>**

Preprint submitted on 29 Jul 2016 (v1), last revised 11 Jun 2018 (v3)

**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.

# LONG TERM AGING : AN ADAPTATIVE WEIGHTS DYNAMIC PROGRAMMING ALGORITHM

BENJAMIN HEYMANN, PIERRE MARTINON, AND FRÉDÉRIC BONNANS

ABSTRACT. We introduce a class of optimal control problems with periodic data. A state variable that we call the age of the system represents the negative impact of the operations on the system qualities over time: other things being equal, older systems have higher operating costs. Many industrial problems relate to this class. If we envision to perform an optimization over a large number of periods, there is a tradeoff between minimizing repeatedly the one-period criterion in a short sighted way and taking into account the impact of the decision on the aging speed (which modifies the minimal one period criterion). In general, because the aging process is slow, short term optimization strategies- such as one period sliding horizon strategies- either neglect it or use rule-of-thumb penalization terms in the criterion, which leads to suboptimal solutions. On the other hand, for most applications it is unrealistic to envision a brute-force numerical resolution by dynamic programming of the long term problem because of the computation burden. We introduce a two-scale method to reduce this computation burden. The method relies on Lagrangian duality and some monotony properties. We expose the theoretical foundations of the method and discuss some practical aspects: approximation errors, asymptotic estimation, computation burden, possible extensions, etc. Since our initial motivation was the difficulty to take long term battery aging in Energy Management Systems into account, we implement the method on a toy long term microgrid energy management problem.

## 1. INTRODUCTION

The aging of physical systems is almost never taken into account in decision making. This can be a cause of sub-optimality. For instance, for many controlled industrial systems, a decision is taken every day. The decision should minimize a tangible criterion, such as the operating cost and is subject to some operating and physical constraints. As time goes on, the system gets older and its physical qualities decrease. On the one hand, this slow aging process depends on how the system is operated. On the other hand the aging of the system is responsible for a loss in efficiency that increases the daily operating cost. Therefore, in an ideal world, the operator should take into account the impact of the daily decision on the long term aging of the system. Nonetheless, this is often technically challenging: because of the time scales involved, the long term optimization problem requires a lot of time and memory to be solved (curse of dimensionality with respect to the state variables). In this work the system aging process is modeled with a one dimensional age variable  $a$  with values in a segment. The physical qualities of the system decrease as  $a$  increases. We point out that the choice of an increasing variable to measure the aging process is arbitrary, and we could have defined instead a decreasing health variable. Of course in this case the physical qualities of the system would have been increasing in the health process.

---

*Date:* July 29, 2016.

2010 *Mathematics Subject Classification.* 93A13, 93C15, 90C39, 49L20, 49M27, 49M29.

CMAP, Inria, Ecole polytechnique, CNRS, Université Paris-Saclay, 91128, Palaiseau, France.

We introduce a class of optimal control problems with periodic data. If we envision to perform an optimization over a large number of periods, there is a trade-off between minimizing repeatedly the one-period criterion in a short sighted way and taking into account the impact of the decision on the aging speed (which will later on in turn have an impact on the minimal one-period criterion). Microgrid energy management relates to the framework proposed in this paper when considering the aging of the battery. We discuss some existing approaches from this literature in §6.2. As far as we know, it appears that when taken into account, aging is constrained or penalized but not directly subject to a long term optimization. An important characteristic of our problem is the existence of two time scales. Some other works on multi time scales problem consider averaging techniques (see [5]).

A first essential observation is that if we knew an optimal aging profile over the whole time horizon, then the long term problem could be decomposed into a sum of micro problems with smaller time horizon, say one period. We could then use myopic approaches to solve the resulting one period (or *micro*) problems. The reverse is also true: the long term (or *macro*) problem can be reformulated by means of the micro problems. So we could precompute the solutions of the micro problems off-line to solve the long term aging problem afterward with a *macro* dynamic programming optimization. This is a first step to reduce the problem numerical complexity. It will be detailed in §3.

Another essential observation is that aging is a slow process, so we could neglect the age variations within a period to solve simplified micro problems without truly impacting the performances. Indeed, this would decrease by one the number of state variables. Yet it is still necessary to know the age at the end of the period to solve the long term problem. Moreover, since we want to take the age into account in the decision, we should be able to control the total aging over the period. So we look for a method to *control* the total aging over a period without requiring the age to be a state variable in the micro problem *numerical* resolution.

Our approach consists in penalizing the aging over the period in the criterion of the micro problems. Then we map the penalization coefficients with the resulting agings. By doing so, we do not need to keep track of the aging within the micro problems (see §4). We then optimize for each period the penalization coefficient in the reformulation of the long term problem. Observe that even if the penalization parameter could be interpreted as a Lagrangian multiplier, we will be performing a *minimization* over this coefficient, which can be confusing. An alternative understanding is to see this penalization parameters as a change of variable: instead of optimizing over the one-period age increments, we optimize over the corresponding penalization coefficients. Figure 1 illustrates this idea.

To sum-up, we propose a decomposition/parametrization method to solve a long term optimal control problem incorporating an age variable. We use the fact that the age (as a state variable) is slow to neglect its variations within a single period to limit the computation burden. We solve a collection of one period, penalized, optimal control problems and associate the resulting total agings with the corresponding coefficients, which allow us to perform a dynamic optimization over the coefficients to solve the long term problem. Observe that we do not fix only one coefficient for the whole problem as in penalization approaches: we control the penalization coefficients of every single period. Then the aging-conscious on-line one-period decision can be computed for an optimized time varying penalization parameter that incorporates the long term aging.

The long term and short term optimal control problems are presented in the next subsection, followed by the technical assumptions we use in this work. We present the bi-level dynamic programming in §3 and the adaptive weights approach in §4. We discuss the theoretical results and

some possible refinements and variations in §5. The last section is dedicated to an application of this approach to the microgrid energy management problem. Readers only interested in applications might want to start with this last section.

In the proofs we will use LHS and RHS as shorthand's for left-hand side and right-hand side.

## 2. SETTING

**2.1. Problem Formulation.** We consider a system with two state variables: the *age*  $a$  and the *fast state*  $c$ . The fast state should be interpreted as a form of wealth (available cash, energy, inventory). This is why we will refer to the fast state as the *charge*, in reference to this idea and to the toy microgrid model example proposed in the last section. The age  $a$  and the charge  $c$  follow a  $T$ -periodic dynamics controlled by a time dependent parameter  $u$ :

$$(2.1) \quad \begin{cases} \dot{a}(t) = F_a(a(t), c(t), u(t), t) \\ \dot{c}(t) = F_c(a(t), c(t), u(t), t), \end{cases}$$

or equivalently if we set  $x = (a, c)$  and  $F = (F_a, F_c)$ :

$$(2.2) \quad \dot{x}(t) = F(x(t), u(t), t).$$

Think of  $T > 0$  as the length of a day for instance. The whole horizon  $T_{tot}$  is a multiple of  $T$ , i.e.  $T_{tot} = NT$  with  $N$  being a large integer (think of  $T_{tot}$  as the length of five years for example). The control  $u$  is restricted to be in  $U = \{u \text{ s.t. } \forall t \in [0, T_{tot}], u(t) \in \bar{U}\}$ , and the charge  $c$  and the age  $a$  should belong to respectively  $\mathcal{C}$  and  $\mathcal{A}$ , where  $\bar{U}$  and  $\mathcal{C}$  are compact subsets of  $\mathbb{R}^n$  and  $\mathbb{R}$ , and there exists  $(a_-, a_+)$  such that  $\mathcal{A} = [a_-, a_+]$  and  $a_- \leq a(0) \leq a_+$ .

The age  $a$  of the system is non-decreasing in time, i.e.  $F_a \geq 0$ , and for all  $(u, t, c) \in \bar{U} \times [0, T] \times \mathcal{C}$ ,  $F_a(a_+, c, u, t) = 0$ , which ensures that for all time  $t$ ,  $a(t) \in [a_-, a_+]$ . One can interpret  $a = a_+$  as the aging component of the system being definitely dead. Then  $F_a(a_+, c, u, t) = 0$  means that once this component dead, it cannot get older anymore.

The behaviors of the system change as it get older. In fact, the older the system, the more effort should be needed to complete a given task. The aging process should decrease the *efficiency* of the system. In addition, as  $c$  is a form of wealth, the cost-to-go functions of our minimization problems should be decreasing in  $c$ . Those notions are expressed in the monotonicity assumption (see. §2.2).

While the relevant time scale for the charge  $c$  is the period  $T$ , the dynamics of the age  $a$  is so slow that one need to wait many periods to observe a non-microscopic change in  $a$ . Our objective is to minimize the sum of an integral criterion and a non-decreasing final cost  $\phi(a_{T_{tot}})$  while verifying some constraints. Like the dynamics, the (bounded) running cost  $\ell(u, t)$  is  $T$ -periodic. We end up with the following optimal control formulation:

$$(2.3) \quad P_1(a_0, c_0, t_0) \left\{ \begin{array}{l} V_1(a_0, c_0, t_0) = \inf_{u \in U} \int_{t_0}^{T_{tot}} \ell(u(t), t) dt + \phi(a_{T_{tot}}) \\ (a(t_0), c(t_0)) = (a_0, c_0) \\ a(T_{tot}) \leq a_{max} \\ \forall t \in [t_0, T_{tot}], \varphi(c(t), u(t), t) \in A \\ (\dot{a}(t), \dot{c}(t)) = (F_a(a(t), c(t), u(t), t), F_c(a(t), c(t), u(t), t)), \end{array} \right.$$

where  $A$  is a closed set,  $\varphi$  a continuous  $T$  periodic-function, and  $a_{max}$  a final constraint. Problem  $P_1$  is parametrized by the initial point  $(a_0, c_0, t_0)$ . The value function  $V_1$  associates any set of initial

time and state variables with the corresponding minimal cost-to-go function. Under Lipschitz conditions on  $F_a$  and  $F_c$ , the ordinary differential equation has a unique solution. The integrand  $\ell$  satisfies the following properties:

- $\ell$  is bounded
- for all  $t \in [0, T]$ ,  $u \rightarrow \ell(u, t)$  is continuous
- for all  $u \in \bar{U}$ ,  $t \rightarrow \ell(u, t)$  is measurable.

We use the standard flow notation i.e. for  $\tau > 0$ ,  $X_{x,t}^{u,t+\tau}$  is the value of the solution of the first order ordinary differential equation of the dynamics at time  $t+\tau$  when the initial point is  $x = (a, c)$  at time  $t$  and the control is  $u$ . We will use the notation  $a_{x,t}^{u,t+\tau}$  (resp.  $c_{x,t}^{u,t+\tau}$ ) to refer to the flow of the age (resp. the charge). When the context will be clear, we will sometimes simply write  $X(t)$ .

The dynamic programming principle applies and we can write the value function of problem  $P_1$  as the solution of:

$$(2.4) \quad V_1(a_0, c_0, t_0) = \inf_{u \in \bar{U}} \int_{t_0}^{t_0+T} \ell(u(t), t) dt + V_1(X_{x_0, t_0}^{u, t_0+T}, t_0 + T),$$

where the infimum is taken over controls such that  $\varphi(c_{x_0, 0}^{u, t}, u(t), t) \in A$  for any  $t \in [t_0, t_0 + T]$ . Observe that problem  $P_1$  is not always feasible, the optimization is by construction performed over the feasible  $V_1(X_{x_0, t_0}^{u, t_0+T}, t_0 + T)$ . Now set  $t = t_k = kT$  for  $k \in \mathbb{N}$ , by  $T$ -periodicity of the data, we get the formulation we will use throughout this article

$$(2.5) \quad V_1(a_0, c_0, t_k) = \inf_{u \in U_{a_0, c_0}} \int_0^T \ell(u(t), t) dt + V_1(X_{x_0, 0}^{u, T}, t_k),$$

where  $U_{a_0, c_0}$  is the set of controls  $u \in U_T = \{u \in L^\infty(0, T) \text{ s.t. } \forall t \in [0, T], u(t) \in \bar{U}\}$  such that  $\varphi(c_{x_0, 0}^{u, t}, u(t), t) \in A$  for any  $t \in [0, T]$ .

We propose a bi-level approach to solve problem  $P_1$ . First we introduce a collection of *micro* problems

$$(2.6) \quad P_1^\mu(a_0, \delta a, c_0, c_F) \begin{cases} V_1^\mu(a_0, \delta a, c_0, c_F) = \inf_{u \in U_T} \int_0^T \ell(u(t), t) dt \\ (a(0), c(0)) = (a_0, c_0) \\ a(T) \leq a_0 + \delta a \\ c(T) \geq c_F \\ \forall t \in [0, T], \varphi(c(t), u(t), t) \in A. \\ (\dot{a}(t), \dot{c}(t)) = (F_a(a(t), c(t), u(t), t), F_c(a(t), c(t), u(t), t))). \end{cases}$$

The superscript  $\mu$  stands for *micro*. The micro problem  $P_1^\mu$  is in many ways similar to  $P_1$ : the dynamics, the mixed constraints and the integrand of the criterion are the same. The difference is that the time horizon is only one period. Moreover, the final state condition on the age  $a$  and the final cost  $\phi(a_{T_{tot}})$  are replaced by two final state conditions on the age  $a$  and the state  $c$ . The problem is parametrized by  $\delta a$ , which represents the maximal amount of aging for the period. If  $\phi = 0$  then for both  $P_1$  and  $P_1^\mu$  the goal is to minimize the integral of the running cost  $\ell$  over a time horizon without increasing the age of the system  $a$  by more than a given quantity.

**2.2. Assumptions.** We make the following assumptions.

*Assumption 1* (Slow aging). There exists a constant  $L > 0$  such that

- $F_c$  is  $L$ -Lipschitz and uniformly bounded by  $L$ ,
- $F_a$  is  $L/T$ -Lipschitz and uniformly bounded by  $L/T$ .

Assumption 1 expresses that the aging process is slow. The dependence of  $L_a^N$  in  $N$  (which is the number of periods in the macro problem  $P_1$ ) is needed to perform an asymptotic estimation. (see §5.1). Assumption 1 is used in the proof of Lemma 5.1.

*Assumption 2* (Monotonicity). (a) The value functions  $V_1^\mu$  and  $V_1$  are non decreasing in  $a_0$  and non increasing in  $c_0$ .

(b) The value function  $V_1^\mu$  is non-increasing in  $\delta a$  and non-decreasing in  $c_F$ .

The first item of Assumption 2 corresponds to the fact that youth and wealth are always preferable. The second item is just a remark that comes from the definition of  $V_1^\mu$ .

*Assumption 3* (Regularity of the aging process). For any  $\epsilon > 0$ ,  $\Delta > 0$ , there exists  $\epsilon_1 > 0$  such that

if  $x_0 = (a_0, c_0) \in \mathcal{A} \times \mathcal{C}$ ,  $u \in U_{x_0}$  and  $\Delta \leq \int_0^T F_a(X(t), u(t), t)dt \leq \Delta + \epsilon_1$ ,  
then there exists  $u' \in U_{x_0}$  such that

$$(2.7) \quad \int_0^T F_a(X(t), u'(t), t)dt = \Delta \quad \text{and} \quad \left| \int_0^T [\ell(u(t), t) - \ell(u'(t), t)]dt \right| \leq \epsilon.$$

We can interpret  $\Delta$  as a maximum aging level. Assumption 3 ensures that if we set a maximum aging level and a precision level, then we can modify any ‘‘almost admissible’’ control for the maximum aging level into an admissible one, and the change in integral cost will not be more than the precision level. This Assumption is used in the proof of Lemma 3.1.

### 3. BILEVEL DYNAMIC PROGRAMMING

**3.1. Mathematical Justification.** We start with the following (intuitive) result.

**Lemma 3.1.** *For all  $(k, a_0, c_0) \in \mathbb{N} \times \mathcal{A} \times \mathcal{C}$ ,*

$$(3.1) \quad V_1(a_0, c_0, t_k) = \inf_{\delta a \in \mathbb{R}^+, c_F \in \mathcal{C}} V_1^\mu(a_0, \delta a, c_0, c_F) + V_1(a_0 + \delta a, c_F, t_{k+1}).$$

*Proof.*  $LHS \geq RHS$ : Set  $x_0 = (c_0, a_0)$ . First we establish that  $LHS \geq RHS$ . Take a control  $u \in U_{x_0}$  and set  $\delta a = a_{x_0,0}^{u,T} - a_0$  and  $c_F = c_{x_0,0}^{u,T}$ . By construction of  $\delta a$  and  $c_F$ ,  $u$  is admissible for  $P_1^\mu(a_0, \delta a, c_0, c_F)$  so by definition of  $V_1^\mu$ ,  $\int_0^T \ell(u(t), t)dt \geq V_1^\mu(c_0, c_F, a_0, \delta a)$ . Moreover, trivially  $V_1(X_{x_0,0}^{u,T}, t_{k+1}) = V_1(a_0 + \delta a, c_F, t_{k+1})$  therefore

$$(3.2) \quad \int_0^T \ell(u(t), t)dt + V_1(X_{x_0,0}^{u,T}, t_{k+1}) \geq V_1^\mu(c_0, c_F, a_0, \delta a) + V_1(a_0 + \delta a, c_F, t_{k+1}) \geq RHS.$$

Since this is true for any  $u \in U_{x_0}$ , we can apply the dynamic programming principle (2.5) to get  $LHS \geq RHS$ . We point out that here we did not use the existence of minimizers.

$LHS \leq RHS$  Take  $\delta a$  and  $c_F$  an  $\epsilon$ -optimal decision for the RHS and  $u \in U_{x_0}$  an  $\epsilon$ -optimal control for  $P_1^\mu(a_0, \delta a, c_0, c_F)$ . By definition of  $P_1^\mu(a_0, \delta a, c_0, c_F)$  and admissibility of  $u$ ,  $c_{x_0,0}^{u,T} \geq c_F$  and  $a_{x_0,0}^{u,T} \leq a_0 + \delta a$ . By  $\epsilon$ -optimality,

$$(3.3) \quad RHS + 2\epsilon \geq \int_0^T \ell(u(t), t)dt + V_1(a_0 + \delta a, c_F, t_{k+1})$$

By monotonicity of  $V_1$ ,

$$(3.4) \quad V_1(a + \delta a, c_F, t_{k+1}) \geq V_1(a + \delta a, c_{x_0,0}^{u,T}, t_{k+1}) \geq V_1(a_{x_0,0}^{u,T}, c_{x_0,0}^{u,T}, t_{k+1}).$$

Therefore,

$$(3.5) \quad RHS + 2\epsilon \geq \int_0^T \ell(u(t), t) dt + V_1(X_{x_0,0}^{u,T}, t_{k+1}) \geq LHS,$$

where we used the dynamic programming principle (2.5) and the fact that  $u \in U_{x_0}$  for the last inequality. We can conclude that  $RHS = LHS$ .  $\square$

**3.2. Complexity Analysis.** We proceed with a complexity analysis of the previous results. Assume we characterize the discretization of the space and time grid with the integer parameters  $N_a$ ,  $N_c$ ,  $N_u$  and  $N_t$ , which are the discretization levels of  $a$ ,  $c$ ,  $u$  and one unit of time. On the one hand, if we solve problem  $P_1$  directly by dynamic programming (for a fixed initial state), the computation burden is proportional to  $N_a N_c N_u N_t T N$ . On the other hand if we use Lemma 3.1, we first solve  $P_1^\mu$  offline for each possible parameters. Problem  $P_1^\mu$  computation burden for one numerical resolution is proportional to  $N_a N_c N_u N_t T$ , but we need to solve it for each final parameters ( $c_F, a_F = a_0 + \delta a$ ), ie  $N_a N_c$  times (indeed, observe that one resolution solves the problem for all possible initial states). Then we have to do the macro resolution ( $O(N_a N_c N)$ ). So the total cost with the micro/macro formulation is  $O(N_a^2 N_c^2 N_u N_t T) + O(N_a N_c N)$ , which can be competitive against a brute-force dynamic programming if  $N$  is large compared to  $N_a N_c$ .

Consider a specific case where the charge  $c$  has to be the same at the beginning of each period. In this case we get:

- Direct dynamic programming resolution:  $O(N_a N_c N_u N_t T N)$
- One micro problem resolution:  $O(N_a N_c N_u N_t T)$
- Number of parameters for the micro problem:  $O(N_a)$
- Resolution of the macro problem:  $O(N_a^2 N)$
- Total computation burden for the bilevel approach:  $O(N_a^2 N) + O(N_a^2 N_c N_u N_t T)$

Then this approach becomes competitive compared to the direct DP approach if  $N$  is larger than  $N_a$ . Note that when solving the micro problem, the complexity is proportional to  $N_a$ , but the grid does not need to contain the whole  $\mathcal{A}$  segment for a given set of final ages. The pseudo code of such an algorithm is straightforward.

- Compute the value function of  $P_1^\mu$  for all  $\delta a$ .
- Compute the value function of the macro problem using the previous results.

Now assume we do an online implementation (with periodic  $c_F = \hat{c}_F$ ): at the beginning of each period, we compute an optimal control for the period, with a final constraint  $c_F = \hat{c}_F$ . With direct dynamics programming, either we recompute the value function every time (a), or we keep it in memory (which requires a lot of ressources)(b). If we keep the whole value function in memory, it requires a memory space proportional to  $N_a N_c N_t N T$ , which is not realistic. If we only keep the value function at the end of each period, then we only need a memory space proportional to  $N_a N$ , but we need to recompute the intermediate value function for a computation burden of  $N_a N_c N_u N_t T$ .

With a bilevel approach, we can keep in memory either the whole value function of the macro problem (c), the value functions of the micro problems at  $t = 0$  for different maximal aging (d), or the mapping of the optimal  $\delta a$  for each  $(a, t)$  (e). The first solution is similar to (a) and is not realistic. The second solution requires a memory space of  $O(N_a^2)$ , which is not proportional to  $N$ .

Then we need to get the optimal aging by dynamic programming on the macro problem  $O(N_a^2 N)$  and compute an optimal control  $O(N_a N_c N_u N_t T)$ . The last possibility requires a  $N_a \times N$  table. The online computation of the control will then require two states, which represents a computation burden proportional to  $N_a N_c N_u N_t T$ . Note that this last possibility is equivalent to (b) for the online phase.

The complexity analysis is summarized in Tables 1 and 2.

We point out that we have not used much of the problem specificity, as only Assumptions 2 is needed in the proof of Lemma 3.1. Moreover, such online computation burden may be too big for some applications. This is what motivates the next section.

#### 4. ADAPTATIVE WEIGHTS

**4.1. Preliminary results.** Note that we could replace the final constraint and the final cost in  $P_1$  by a penalization on the age variation. We would get the criterion

$$(4.1) \quad \int_{t_0}^{T_{tot}} \ell(u(t), t) dt + \alpha [a(T_{tot}) - a_{max}] + \phi(a_{T_{tot}}).$$

where  $\alpha \in \mathbb{R}_+$  is a penalization coefficient. Since  $a_{max}$  and  $\hat{a}_0$  are constants over which we are not optimizing, we get an equivalent optimization problem by using instead the criterion

$$(4.2) \quad \int_{t_0}^{T_{tot}} \ell(u(t), t) dt + \alpha \int_{t_0}^{T_{tot}} F_a(X(t), u(t), t) dt + \phi(a_{T_{tot}}).$$

We then get the penalized problem (with free final age)

$$(4.3) \quad P_2(a_0, c_0, \alpha, t_0) \left\{ \begin{array}{l} V_2(a_0, c_0, \alpha, t_0) = \inf_{u \in U} \int_{t_0}^{T_{tot}} [\ell(u(t), t) + \alpha F_a(X(t), u(t), t)] dt + \phi(a_{T_{tot}}) \\ \dot{X}(t) = F(X(t), u(t), t) \\ (a(t_0), c(t_0)) = (a_0, c_0) \\ \varphi(c(t), u(t), t) \in A \end{array} \right.$$

Let us introduce the corresponding micro-problem:

$$(4.4) \quad P_2^\mu(a_0, c_0, c_F, \alpha) \left\{ \begin{array}{l} V_2^\mu(a_0, c_0, c_F, \alpha) = \inf_{u \in U} \int_0^T [\ell(u(t), t) + \alpha F_a(X(t), u(t), t)] dt \\ \dot{X}(t) = F(X(t), u(t), t) \\ (a(0), c(0)) = (a_0, c_0) \\ \varphi(c(t), u(t), t) \in A \\ c(T) \geq c_F. \end{array} \right.$$

As the notations implies,  $P_2^\mu$  is to  $P_2$  what  $P_1^\mu$  is to  $P_1$ : a one day version. Just note that the final constraint in  $P_1$  is considered as fixed and is a parameter in  $P_1^\mu$  whereas the penalization coefficient is a parameter for both  $P_2^\mu$  and  $P_2$ . In addition, note that we cannot write a dynamic programming principle directly with  $P_2^\mu$  and  $P_2$  as we did for  $P_1^\mu$  and  $P_1$  in Lemma 3.1.

If problem  $P_1$  is strictly feasible, then for  $\alpha$  big enough the final constraint is satisfied by the solutions of  $P_2(a_0, c_0, \alpha, t_0)$  (see the proof of Lemma 6). A standard way to deal with aging is to



replace  $P_1(a_0, c_0, t_0)$  by an approximation of  $P_2(a_0, c_0, c_F, \alpha)$  where the age  $a$  is fixed:

$$(4.5) \quad \tilde{P}_2(a_0, c_0, \alpha, t_0) \begin{cases} V_2(a_0, c_0, \alpha, t_0) = \inf_{u \in U} \int_{t_0}^{T_{tot}} [\ell(u(t), t) + \alpha F_a(a_0, c(t), u(t), t)] dt + \phi(a_{T_{tot}}) \\ \dot{c}(t) = F_c(a_0, c(t), u(t), t) \\ (a(t_0), c(t_0)) = (a_0, c_0) \\ \varphi(c(t), u(t), t) \in A \end{cases}$$

To set  $\alpha$ , practitioners would often compute a collection of solutions of  $\tilde{P}_2$  for different values of  $\alpha$ , and then take the best admissible solution in the sense of  $P_1$ . Nonetheless such an approach neglects the impact of the aging on the efficiency and more generally the fact that the dynamics are coupled.

In addition, there may be a duality gap. In this case any (theoretical) solution of  $P_2$  admissible for  $P_1$  is suboptimal for  $P_1$ . Ekeland and Aubin propose an estimate of this sub-optimality in a finite dimensional setting [1].

The following lemma is a relation between the value functions of the two micro problems  $P_2^\mu$  and  $P_1^\mu$ .

**Lemma 4.1.** *For any  $(a_0, c_0, c_F, \alpha) \in \mathcal{A} \times \mathcal{C}^2 \times \mathbb{R}_+$ ,*

$$(4.6) \quad V_2^\mu(a_0, c_0, c_F, \alpha) = \inf_{\delta a \in \mathbb{R}_+} V_1^\mu(a_0, \delta a, c_0, c_F) + \alpha \delta a$$

*Proof. LHS  $\geq$  RHS:*

Take  $\epsilon > 0$ ,  $u \in U_{x_0}$  an  $\epsilon$ -optimal solution of the LHS. Set  $\delta a = a_{x_0, 0}^{u, T} - a_0$ . By  $\epsilon$ -optimality of  $u$

$$(4.7) \quad LHS + \epsilon \geq \int_0^T \ell(u(t), t) dt + \alpha \delta a.$$

In addition note that  $u$  is admissible for  $P_1^\mu(a_0, \delta a, c_0, c_F)$  therefore by definition of  $V_1^\mu$ ,  $\int_0^T \ell(u(t), t) dt \geq V_1^\mu(a_0, \delta a, c_0, c_F)$ . Then

$$(4.8) \quad LHS + \epsilon \geq V_1^\mu(a_0, \delta a, c_0, c_F) + \alpha \delta a \geq \inf_{\delta a \in \mathbb{R}_+} V_1^\mu(a_0, \delta a, c_0, c_F) + \alpha \delta a = RHS.$$

Therefore  $LHS \geq RHS$ .

$LHS \leq RHS$ :

Take  $\epsilon > 0$ ,  $\delta a \in \mathbb{R}_+$  an  $\epsilon$ -optimal solution of the RHS and  $u \in U_{x_0}$  an  $\epsilon$ -optimal solution of problem  $P_1^\mu(a_0, \delta a, c_0, c_F)$ . By  $\epsilon$ -optimality

$$(4.9) \quad RHS + \epsilon \geq V_1^\mu(a_0, \delta a, c_0, c_F) + \alpha \delta a \quad \text{and} \quad V_1^\mu(a_0, \delta a, c_0, c_F) + \epsilon \geq \int_0^T \ell(u(t), t) dt,$$

so

$$(4.10) \quad RHS \geq \int_0^T \ell(u(t), t) dt + \alpha \delta a - 2\epsilon.$$

Moreover, since  $u$  is  $P_1^\mu(a_0, \delta a, c_0, c_F)$ -admissible,  $\delta a \geq \int_0^T F_a(X(t), u(t), t) dt$ , so

$$(4.11) \quad RHS \geq \int_0^T [\ell(u(t), t) + \alpha F_a(X(t), u(t), t)] dt - 2\epsilon \geq LHS - 2\epsilon.$$

Therefore  $LHS \leq RHS$  and the proof is done.  $\square$

We point out that this result does not depend on  $\phi$  and  $a_{max}$  which are macro problem specific parameters.

*Corollary 4.2.* For any  $(a_0, c_0, c_F, \delta a, \alpha) \in \mathcal{A} \times \mathcal{C}^2 \times \mathbb{R}_+^2$ ,

$$(4.12) \quad V_1^\mu(a_0, \delta a, c_0, c_F) \geq V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha \delta a.$$

**4.2. Nice Case: No Duality Jumps.** To proceed we need some additional notations. First for any  $x_0 \in \mathcal{A} \times \mathcal{C}$ , and any control  $u \in U_{x_0}$ , we define

$$(4.13) \quad \Delta a(u) = \int_0^T F_a(X(t), u(t), t) dt \quad \text{and} \quad \mathcal{L}(u) = \int_0^T \ell(u(t), t) dt.$$

Note that for readability the initial conditions are kept implicit. For any  $(a_0, c_0, c_F, \alpha) \in \mathcal{A} \times \mathcal{C}^2 \times \mathbb{R}_+$ , let

$$(4.14) \quad \Gamma(a_0, c_0, c_F, \alpha) = \left\{ \begin{array}{l} \delta a = \lim_n \Delta a(u_n); \\ u_n \text{ minimizing sequence of } P_2^\mu(a_0, c_0, c_F, \alpha) \end{array} \right\}$$

Roughly speaking, given some  $(a_0, c_0, c_F) \in \mathcal{A} \times \mathcal{C}^2$ ,  $\alpha \rightarrow \Gamma(a_0, c_0, c_F, \alpha)$  associates the penalization coefficients with the set of optimal aging levels for  $P_2^\mu$ . Lemma 4.3 is a key result for what follows.

**Lemma 4.3.** Let  $(a_0, c_0, c_F, \alpha, \Delta a) \in \mathcal{A} \times \mathcal{C}^2 \times \mathbb{R}_+^2$  such that  $\Delta a \in \Gamma(a_0, c_0, c_F, \alpha)$ . Then

$$(4.15) \quad V_1^\mu(a_0, \Delta a, c_0, c_F) = V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha \Delta a.$$

*Proof.* We deduce  $LHS \geq RHS$  from Corollary 4.2, so we only need to show that  $LHS \leq RHS$ . Let  $\epsilon > 0$ . By Assumption 3, there exists  $\epsilon_1 \geq 0$  such that if  $u_2 \in U_{x_0}$  and  $\Delta a \leq \Delta a(u_2) \leq \Delta a + \epsilon_1$ , then there exists  $u' \in U_{x_0}$  such that

$$(4.16) \quad \Delta a(u') = \Delta a \quad \text{and} \quad |\mathcal{L}(u_2) - \mathcal{L}(u')| \leq \epsilon.$$

By hypothesis,  $\Delta a \in \Gamma(a_0, c_0, c_F, \alpha)$  so by definition of  $\Gamma$  there exists  $u_2$   $\epsilon$ -optimal for  $P_2^\mu(a_0, c_0, c_F, \alpha)$  that satisfies  $|\Delta a(u_2) - \Delta a| \leq \min(\epsilon_1, \frac{\epsilon}{\alpha})$ .

If  $\Delta a(u_2) \leq \Delta a$ , then

$$(4.17) \quad V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha \Delta a \geq V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha \Delta a(u_2) - \epsilon \geq$$

$$(4.18) \quad -2\epsilon + \mathcal{L}(u_2) \geq -2\epsilon + V_1^\mu(c_0, c_F, a_0, \Delta a).$$

We used  $\alpha|\Delta a(u_2) - \Delta a| \leq \epsilon$  for the first inequality, the  $\epsilon$ -optimality of  $u_2$  for the second inequality, and the admissibility of  $u_2$  for  $P_1^\mu(c_0, c_F, a_0, \Delta a)$  in the third.

Else we have the existence of a control  $u'$  satisfying (4.16). Then by  $\epsilon$ -optimality of  $u_2$  for  $P_2^\mu$ , (4.16) and  $\Delta a(u_2) > \Delta a$ , and the fact that  $u'$  is admissible for  $P_1^\mu$  and  $\Delta a(u') = \Delta a$ :

$$(4.19) \quad V_2^\mu(a_0, c_0, c_F, \alpha) \geq -\epsilon + \mathcal{L}(u_2) + \alpha \Delta a(u_2) \geq$$

$$(4.20) \quad -2\epsilon + \mathcal{L}(u') + \alpha \Delta a(u') \geq$$

$$(4.21) \quad -2\epsilon + V_1^\mu(c_0, c_F, a_0, \Delta a) + \alpha \Delta a.$$

Then we can conclude. □

From Lemma 4.3 it is trivial that

*Corollary 4.4.* Let  $(a_0, c_0, c_F, \alpha, \delta a, u) \in \mathcal{A} \times \mathcal{C}^2 \times \mathbb{R}_+^2 \times U_T$  and  $\epsilon > 0$  such that  $u$  is an  $\epsilon$ -optimal solution of  $P_1^\mu(a_0, \delta a, c_0, c_F)$  and  $\Delta a(u) \in \Gamma(a_0, c_0, c_F, \alpha)$  then

$$(4.22) \quad V_1^\mu(a_0, \Delta a(u), c_0, c_F) = V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha \Delta a(u)$$

Now we have the tools to prove one of our main results.

**Theorem 4.5.** *Let  $(a_0, c_0, \alpha_n, u) \in \mathcal{A} \times \mathcal{C} \times \mathbb{R}_+^n \times U$  and  $\epsilon > 0$  such that  $u$  is an  $\epsilon$ -optimal solution for  $P_1(a_0, c_0, 0)$  and  $(a_{k+1} - a_k) \in \Gamma(a_k, c_k, c_{k+1}, \alpha_n)$ , where  $a_k = a_{x_0, 0}^{u, t_k}$  and  $c_k = c_{x_0, 0}^{u, t_k}$ . Then for all  $k = 0 \dots N - 1$*

$$(4.23) \quad |V_1(a_k, c_k, t_k) - \inf_{(\alpha, c_F, \delta a)} \{V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha \delta a + V_1(a_k + \delta a, c_F, t_{k+1})\}| \leq \epsilon,$$

where the optimization is performed over the  $(\alpha, \delta a, c_F)$  such that  $\alpha \in \mathbb{R}_+$ ,  $c_F \in \mathcal{C}$  and  $\delta a \in \Gamma(a_k, c_k, c_F, \alpha)$ .

*Proof.* Without loss of generality, we deal with the case  $k = 0$ . First note that the restriction  $u_T$  of  $u$  to  $[0, T]$  is an  $\epsilon$ -optimal solution for  $P_1^\mu(a_0, a_1 - a_0, c_0, c_1)$ .

By  $\epsilon$ -optimality, admissibility of  $u_T$  for  $P_1^\mu(a_0, a_1 - a_0, c_0, c_1)$ , Corollary 4.4 and the fact that  $a_1 - a_0 \in \Gamma(a_0, c_0, c_1, \alpha_0)$ ,

$$(4.24) \quad V_1(a_0, c_0, 0) + \epsilon \geq \mathcal{L}(u_T) + V_1(a_1, c_1, T) \geq V_1^\mu(a_0, a_1 - a_0, c_0, c_1) + V_1(a_1, c_1, T)$$

$$(4.25) \quad \geq V_2^\mu(a_0, c_0, c_1, \alpha_0) - \alpha_0(a_1 - a_0) + V_1(a_0 + a_1 - a_0, c_1, T)$$

$$(4.26) \quad \geq \inf_{\alpha, \delta a, c_F} V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha \delta a + V_1(a_0 + \delta a, c_F, T)$$

Therefore  $RHS \leq LHS + \epsilon$ .

For any  $(\alpha, \delta a, c_F) \in \mathbb{R}_+ \times \mathbb{R}_+ \times \mathcal{C}$  such that  $\delta a \in \Gamma(a_0, c_0, c_F, \alpha)$ , by Lemma 4.3 and Lemma 3.1:

$$(4.27) \quad V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha \delta a + V_1(a_0 + \delta a, c_F, T) =$$

$$(4.28) \quad V_1^\mu(a_0, \delta a, c_0, c_F) + V_1(a_0 + \delta a, c_F, T) \geq V_1(a_0, c_0, 0).$$

Therefore  $RHS \geq LHS$  and the conclusion follows.  $\square$

We point out that the result is still true if we want to fix the  $c_k$  as operational constraints. Under the hypothesis of the previous theorem, assume that  $\Gamma$  is a singleton, then  $V_1$  can be computed by dynamic programming over  $\alpha$ . We will get a similar result in the next section as a consequence of Theorem 4.12. Note that since in  $P_2^\mu$  there is no final constraint on the age  $a$  we could approximate this problem by fixing the age in the dynamics. With Assumption 1 we should be able to get an error estimate. Last but not least, beware that the optimal  $\alpha$  in Theorem 4.5 is not the  $\alpha$  that would relate by duality two macro problems  $P_1$  and  $P_2$ .

**4.3. Some comments on  $\Gamma$ .** We say that  $P_1^\mu(a_0, \delta a, c_0, c_F)$  is strictly feasible if there exists  $u \in U_{x_0}$  such that  $\Delta a(u) < \delta a$ . We start with the following classical result:

**Lemma 4.6.** *If problem  $P_1^\mu(a_0, \delta a, c_0, c_F)$  is strictly feasible, then there is an  $\alpha_0$  such that for  $\alpha$  bigger than  $\alpha_0$ ,*

$$(4.29) \quad \forall \Delta a \in \Gamma(a_0, c_0, c_F, \alpha), \Delta a \leq \delta a.$$

*Proof.* Since  $P_1^\mu(a_0, \delta a, c_0, c_F)$  is strictly feasible, there exists  $u \in U_{x_0}$  such that  $\Delta a(u) < \delta a$ . Assume that there exists  $\alpha_n$  an increasing sequence such that

$$(4.30) \quad \alpha_n \rightarrow +\infty \text{ and } \forall n \in \mathbb{N}, \exists \Delta a_n \in \Gamma(a_0, c_0, c_F, \alpha_n), \Delta a_n > \delta a.$$

Then

$$(4.31) \quad n(\Delta a_n - \Delta a(u)) \geq n(\delta a - \Delta a(u))$$

Then for  $n$  big enough, since  $\ell$  is bounded, we would have  $\mathcal{L}(u) + n\Delta a(u) < V_2^\mu(a_0, c_0, c_F, n)$ , which is absurd.  $\square$

The result of the next lemma is clear to the intuition:  $\Gamma$  should be decreasing in  $\alpha$ .

**Lemma 4.7.** *Let  $(a_0, c_0, c_F) \in \mathcal{A} \times \mathcal{C}^2$ , then for any  $0 \leq \alpha_1 < \alpha_2$ ,*

$$(4.32) \quad (\delta_1, \delta_2) \in \Gamma(a_0, c_0, c_F, \alpha_1) \times \Gamma(a_0, c_0, c_F, \alpha_2) \Rightarrow \delta_1 \geq \delta_2$$

*Proof.* To simplify, we omit  $(a_0, c_0, c_F)$  because they do not intervene in the proof. Assume that for some non-negative  $\alpha_1 < \alpha_2$ , there exist some  $(\delta_1, \delta_2)$  such that  $\delta_1 < \delta_2$ . Then by corollary 4.2

$$(4.33) \quad V_2^\mu(\alpha_2) \leq V_1^\mu(\delta_1) + \alpha_2\delta_1.$$

This implies with Lemma 4.3 that

$$(4.34) \quad \alpha_2(\delta_2 - \delta_1) \leq V_1^\mu(\delta_1) - V_1^\mu(\delta_2)$$

Then since  $\alpha_1 < \alpha_2$  and  $\delta_2 - \delta_1 > 0$

$$(4.35) \quad \alpha_1(\delta_2 - \delta_1) < V_1^\mu(\delta_1) - V_1^\mu(\delta_2)$$

which in turn implies that

$$(4.36) \quad V_1^\mu(\delta_2) + \alpha_1\delta_2 < V_1^\mu(\delta_1) + \alpha_1\delta_1 = V_2^\mu(\alpha_1),$$

which is not coherent with the optimality of  $V_2^\mu(\alpha_1)$ . We conclude that  $\delta_1 \geq \delta_2$ .  $\square$

Since the data are bounded,  $\Gamma$  is included in a compact set. Since  $P_2^\mu$  has a value, and  $u \rightarrow \Delta(u)$  is valued in a compact set,  $\Gamma$  is not empty. We display in Figure 1 a sketch of  $\Gamma$  as a function of  $\alpha$  (the other variables being fixed). By Lemma 4.7  $\Gamma$  is non-increasing. There is no reason a priori why it could not be locally constant. Indeed, even the solution of  $P_2^\mu$  could be locally constant with respect to  $\alpha$ . Observe that  $\Gamma$  not necessarily a singleton: we can have some jumps. In addition it is not necessarily convex valued.

**Lemma 4.8.** *Let  $(a_0, c_0, c_F, \alpha, \Delta a_1, \Delta a_2) \in \mathcal{A} \times \mathcal{C}^2 \times \mathbb{R}_+^2$  such that  $\Delta a_i \in \Gamma(a_0, c_0, c_F, \alpha)$  for  $i = 1, 2$  then*

$$(4.37) \quad V_1^\mu(a_0, \Delta a_1, c_0, c_F) - V_1^\mu(a_0, \Delta a_2, c_0, c_F) = \alpha(\Delta a_2 - \Delta a_1)$$

*Proof.* This is a direct consequence of Lemma 4.3.  $\square$

**4.4. Generic Case.** In general we do not have any guarantee of the existence of the  $\alpha$ 's as in Theorem 4.3. In relation with problem  $P_2^\mu$ , we introduce for any  $(x_0, c_F, \alpha) \in (\mathcal{A} \times \mathcal{C}) \times \mathbb{R}_+$

$$(4.38) \quad \Delta^-(x_0, c_F, \alpha) = \inf_{u \in S(x_0, c_F, \alpha)} \liminf_{n \rightarrow +\infty} \Delta a(u_n),$$

and

$$(4.39) \quad \Delta^+(x_0, c_F, \alpha) = \sup_{u \in S(x_0, c_F, \alpha)} \limsup_{n \rightarrow +\infty} \Delta a(u_n),$$

where

$$(4.40) \quad S(x_0, c_F, \alpha) = \{u \in (U_T)^\mathbb{N}; (u_n)_{n \in \mathbb{N}} \text{ minimizing sequence of } P_2^\mu(x_0, c_F, \alpha)\}.$$

In addition, we denote by  $\hat{\Gamma}$  the set

$$(4.41) \quad \hat{\Gamma}(x_0, c_F, \alpha) = \{\Delta^-(x_0, c_F, \alpha), \Delta^+(x_0, c_F, \alpha)\}$$

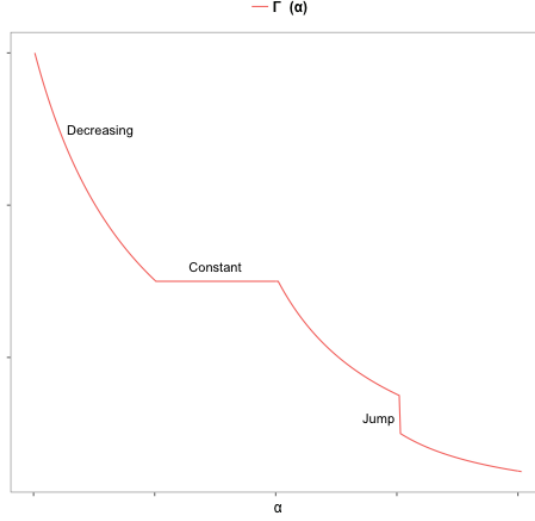


FIGURE 1. AN EXAMPLE OF  $\Gamma(\alpha)$  PROFILE: This drawing summarize the possible behaviors of  $\Delta a(\alpha)$ , i.e the influence of the penalization parameter over the aging in the micro problem. It can be either continuous and strictly decreasing or constant, or it may “jump” for a given value of  $\alpha$ . If there is a jump at  $\alpha = \alpha_0$  then we cannot a priori say that  $\Gamma(\alpha_0)$  is a singleton. In addition, it may happen that  $\Gamma$  ‘misses’ some values at  $\alpha_0$ .

which is either a 2-uplet if there is a jump, and a singleton otherwise. The set  $\hat{\Gamma}$  corresponds to the minimal and maximal optimal age increments for a given  $\alpha$ . Last, we denote by  $\gamma$  the quantity:

$$(4.42) \quad \gamma = \sup_{(x_0, c_F, \alpha) \in (\mathcal{A} \times \mathcal{C}) \times \mathcal{C} \times \mathbb{R}_+} \left\{ \lim_{\alpha^- < \alpha} \Delta^+(x_0, c_F, \alpha^-) - \lim_{\alpha^+ > \alpha} \Delta^-(x_0, c_F, \alpha^+) \right\},$$

which corresponds to the size of the biggest possible duality jump. By monotonicity of  $\Gamma$  with respect to  $\alpha$ ,  $\gamma$  is non negative, and by Assumption 1,  $\gamma$  is finite. We denote by  $P_3(x_k, t_k)$  for  $k = 0 \dots N - 1$  the problem

$$(4.43) \quad V_3(x_k, t_k) = \inf_{(x_i, \alpha_i)} \sum_{i=k}^{N-1} \{V_2^\mu(x_i, c_{i+1}, \alpha_i) - \alpha_i(a_{i+1} - a_i)\} + \tilde{\phi}(a_N),$$

where the optimization is performed over the  $(x_i, \alpha_i)_{i=k+1, \dots, N} \in (\mathcal{A} \times \mathcal{C} \times \mathbb{R}_+)^{N-k}$  such that for any  $i = k \dots N - 1$ ,

$$(4.44) \quad (a_{i+1} - a_i) \in \hat{\Gamma}(x_i, c_{i+1}, \alpha_i)$$

and  $\tilde{\phi}(a) = \phi(a)$  if  $a < a_{max}$ ,  $+\infty$  else.

**Lemma 4.9.** *Let  $(a_0, c_0, c_F, \alpha, \delta a) \in \mathcal{A} \times \mathcal{C}^2 \times \mathbb{R}_+^2$ , such that  $\delta a \in \hat{\Gamma}(a_0, c_0, c_F, \alpha)$ , then*

$$(4.45) \quad V_1^\mu(a_0, c_0, c_F, \delta a) = V_2^\mu(a_0, c_0, c_F, \alpha) - \alpha \delta a$$

*Proof.* We just observe that  $\hat{\Gamma}(x_0, c_F, \alpha) \in \Gamma(x_0, c_F, \alpha)$  and apply Lemma 4.3.  $\square$

**Lemma 4.10.** *Let  $(x, k) \in \mathcal{A} \times \mathcal{C} \times [0 \dots N - 1]$ , then  $V_3(x, t_k) \geq V_1(x, t_k)$ .*

*Proof.* By Lemma 3.1, we have

$$(4.46) \quad V_1(x, t_k) = \inf_{x_i} \sum_{i=k}^{N-1} V_1^\mu(x_i, c_{i+1}, a_{i+1} - a_i) + \tilde{\phi}(a_N).$$

and by lemma 4.9 and the definition of  $V_3$ ,

$$(4.47) \quad V_3(x, t_k) = \inf_{x_i} \sum_{i=k}^{N-1} V_1^\mu(x_i, c_{i+1}, a_{i+1} - a_i) + \tilde{\phi}(a_N).$$

where the optimization is performed over the  $x_i$  such that there exist  $(\alpha_i)_{i=k+1, \dots, N}$  to satisfy 4.44. QED.  $\square$

Remember the definition of  $\gamma$  in (4.42).

**Lemma 4.11.** *For any  $(x_0, c_F, u) \in (\mathcal{A} \times \mathcal{C}) \times \mathcal{C} \times U_T$  such that  $u$  is admissible for problem  $P_1^\mu(a_0, \Delta a(u), c_0, c_F)$ , there exists  $\alpha$  such that*

$$(4.48) \quad \text{dist}(\Delta a(u), \hat{\Gamma}(x_0, c_F, \alpha)) \leq \gamma.$$

*Proof.* We know that for  $\alpha \in \mathbb{R}_+$  big enough,  $\delta \in \hat{\Gamma}(x_0, c_F, \alpha)$  implies  $\delta \leq \Delta a(u)$ . Take  $\alpha_0$  the infimum of those  $\alpha$ . Then if there exists  $\delta \in \hat{\Gamma}(x_0, c_F, \alpha_0)$  such that  $\delta = \Delta a(u)$  we just take  $\alpha = \alpha_0$  and we are done. Else by definition of  $\alpha_0$ , for all  $\alpha^- < \alpha$ ,  $\Delta^+(x_0, c_F, \alpha^-) > \Delta a(u)$  and for all  $\alpha^+ > \alpha$ ,  $\Delta^-(x_0, c_F, \alpha^+) \leq \Delta a(u)$ . This rewrites, for all  $\delta \in \hat{\Gamma}(x_0, c_F, \alpha)$

$$(4.49) \quad \Delta^-(x_0, c_F, \alpha^+) - \delta \leq \Delta a(u) - \delta \leq \Delta^+(x_0, c_F, \alpha^-) - \delta$$

Taking the limit in  $\alpha^-$  and  $\alpha^+$ , we get

$$(4.50) \quad |\Delta a(u) - \delta| \leq \gamma$$

i.e.

$$(4.51) \quad \text{dist}(\Delta a(u), \hat{\Gamma}(x_0, c_F, \alpha)) \leq \gamma. \quad \square$$

The next result contains one of the main ideas of the paper: the set of the  $x_i$  satisfying 4.44 is rich enough to approximate a trajectory with a precision of  $\gamma$ .

**Theorem 4.12.** *Let  $x_0 \in \mathcal{A} \times \mathcal{C}$  and  $u \in U$  admissible for  $P_1(x_0, 0)$ . Then there exist  $(\alpha, \delta a) \in \mathbb{R}_+^{2N}$  such that for all  $k = 0, \dots, N$ :*

$$(4.52) \quad \delta a_k \in \hat{\Gamma}(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$$

$$(4.53) \quad \sum_{i=0}^{k-1} \delta a_i \leq a(t_k) - a_0$$

$$(4.54) \quad \sum_{i=0}^{k-1} \delta a_i \geq a(t_k) - a_0 - \gamma,$$

whereand  $c_k = c(t_k)$ .

*Proof.* First note that (4.53) and (4.54) are satisfied for  $k = 0$ . Now consider  $k = 0, \dots, N$  such that the three properties are satisfied until  $(k - 1)$ . If there exist  $\alpha$  such that

$$(4.55) \quad a(t_{k+1}) - a(t_k) \in \hat{\Gamma}(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$$

then set  $\delta a_k = a(t_{k+1}) - a(t_k)$  and  $\alpha_k = \alpha$ . and then the three properties are still trivially satisfied for  $k$ . Else, we apply Lemma 4.11 to justify the existence of an  $\alpha_k$  such that

$$(4.56) \quad \text{dist}(a(t_{k+1}) - a(t_k), \hat{\Gamma}(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)) \leq \gamma.$$

then if

$$(4.57) \quad \sum_{j=0}^{k-1} \delta a_j + \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \leq a(t_{k+1}) - a_0$$

then we set  $\delta a_k = \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$ . Then constraints (4.52) and (4.53) are satisfied. We need to check that (4.54) is also satisfied. To see this, observe that

$$(4.58) \quad \sum_{j=0}^{k-1} \delta a_j + \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$$

$$(4.59) \quad \geq a(t_k) - a_0 - \gamma + \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$$

$$(4.60) \quad \geq a(t_{k+1}) - a_0 + \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) - (a(t_{k+1}) - a(t_k)) - \gamma$$

$$(4.61) \quad \geq a(t_{k+1}) - \gamma,$$

where we applied, the induction and the fact that  $\Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) \geq a(t_{k+1}) - a(t_k)$ .

Else we set  $\delta a_k = \Delta^-(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$ . We have

$$(4.62) \quad \sum_{j=0}^{k-1} \delta a_j + \Delta^-(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$$

$$(4.63) \quad > a(t_{k+1}) - a_0 - \Delta^+(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) - \Delta^-(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$$

$$(4.64) \quad > a(t_{k+1}) - a_0 - \gamma$$

and

$$(4.65) \quad \sum_{j=0}^{k-1} \delta a_j + \Delta^-(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$$

$$(4.66) \quad \leq a(t_k) - a_0 + \Delta^-(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$$

$$(4.67) \quad \leq a(t_{k+1}) - a_0$$

which concludes the induction.  $\square$

We point out that the proof of this result is constructive. We propose in the next result an a posteriori error estimate based on this construction. Since  $a(t_k)$  is the age of the system piloted by the  $\epsilon$ -optimal solution  $u$  and  $a_0 + \sum_{i=1}^k \delta a_i$  is the age of the system piloted by the solution we are building with adaptative weights, relation (4.52) means that the system is always in better shape (i.e. younger) in the solution we are building, and at the same time, it cannot be more than  $\gamma$  better (reminder:  $\gamma$  is the maximal diameter of  $\Gamma$ ).

**Theorem 4.13** (Error Estimate). *For any  $\epsilon$ -optimal solution of  $P_1(a_0, c_0, t_0)$ , the construction of the previous theorem gives the estimate:*

$$(4.68) \quad V_3(a_0, c_0, t_0) - V_1(a_0, c_0, t_0) \leq \sum_{k=0}^{N-1} \alpha_k (a(t_{k+1}) - a(t_k) - \delta a_k) + \epsilon$$

Note that  $(a(t_{k+1}) - a(t_k) - \delta a_k) = 0$  when there is no jump at  $\alpha_k$ , which can be numerically checked. Thus the theorem give an a posteriori estimate of the error when optimizing with an adaptative weights approach. For instance, if the structure of the problem allows us to claim that there are no jumps, then we do not make any approximation error.

*Proof.* Take  $\epsilon > 0$  and  $u \in U$  an  $\epsilon$ -optimal control for problem  $P_1(c_0, a_0, t_0)$ . We apply Theorem 4.12 to  $u$ .

We have

$$(4.69) \quad V_3(c_0, a_0) \leq \sum_{k=1}^N \{V_2^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) - \alpha_k \delta a_k\} + \tilde{\phi}(a_0 + \sum_{k=0}^{N-1} \delta a_k)$$

We deal with the first term of the RHS. According to Lemma 4.9, for all  $k = 0 \dots N - 1$

$$(4.70) \quad V_2^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k) - \alpha_k \delta a_k = V_1^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, \delta a_k, c_k, c_{k+1})$$

Therefore

$$(4.71) \quad V_2^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_{k-1}, c_k, \alpha_k) - \alpha_k \delta a_k - V_1^\mu(a(t_k), a(t_{k+1}) - a(t_k), c_k, c_{k+1})$$

$$(4.72) \quad = V_1^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, \delta a_k, c_k, c_{k+1}) - V_1^\mu(a(t_k), a(t_{k+1}) - a(t_k), c_k, c_{k+1})$$

$$(4.73) \quad = V_1^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, \delta a_k, c_k, c_{k+1}) - V_1^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, a(t_{k+1}) - a(t_k), c_k, c_{k+1}) +$$

$$(4.74) \quad V_1^\mu(a_0 + \sum_{j=0}^{k-1} \delta a_j, a(t_{k+1}) - a(t_k), c_k, c_{k+1}) - V_1^\mu(a(t_k), a(t_{k+1}) - a(t_k), c_k, c_{k+1})$$

The second difference is negative because  $V_1^\mu$  is decreasing in its first variable and  $a_0 + \sum_{j=0}^{k-1} \delta a_j \leq a(t_k)$  by construction. We thus concentrate on the first difference. If there is no jump, this quantity



is zero. Otherwise, we need to compare (simplifying the notations)  $V_1^\mu(\delta a_k)$  and  $V_1^\mu(a(t_{k+1}) - a(t_k))$ . By definition of  $P_2^\mu$ , we have

$$(4.75) \quad V_1^\mu(a(t_{k+1}) - a(t_k)) + \alpha_k(a(t_{k+1}) - a(t_k)) \geq V_2^\mu(\alpha_k).$$

Remember that by definition the pair  $(\alpha_k, \delta a_k)$  satisfies  $\delta a_k \in \hat{\Gamma}(a_0 + \sum_{j=0}^{k-1} \delta a_j, c_k, c_{k+1}, \alpha_k)$ . We can apply Lemma 4.9 to get

$$(4.76) \quad V_2^\mu(\alpha_k) = V_1^\mu(\delta a_k) + \alpha_k \delta a_k.$$

Therefore, combining relations (4.75) and (4.76) we get

$$(4.77) \quad V_1^\mu(\delta a_k) - V_1^\mu(a(t_{k+1}) - a(t_k)) \leq \alpha_k(a(t_{k+1}) - a(t_k) - \delta a_k)$$

Therefore if we denote by  $K_{jumps}$  the  $k \in [0..N-1]$  such there is a jump in the construction:

$$(4.78) \quad V_3(a_0, c_0, t_0) - V_1(a_0, c_0, t_0) \leq$$

$$(4.79) \quad \sum_{k \in K_{jumps}} \alpha_k(a(t_{k+1}) - a(t_k) - \delta a_k) + \tilde{\phi}(a_0 + \sum_{k=0}^{N-1} \delta a_k) - \phi(a(t_N))$$

$$(4.80) \quad \leq \sum_{k \in K_{jumps}} \alpha_k(a(t_{k+1}) - a(t_k) - \delta a_k).$$

Since  $a(t_N) \geq a_0 + \sum_{k=0}^{N-1} \delta a_k$  by construction,  $\phi$  is monotone decreasing and  $u$  is admissible for  $P_1(x_0, t_0)$ .  $\square$

**4.5. Complexity Analysis.** We denote by  $N_\alpha$  the number of elements in the discretization of  $\alpha$ . We get the following offline computation burden for the adaptive weights algorithm if we neglect the age variations in the micro problem.

- Micro problem:  $O(N_c N_u N_t T)$
- Parameters for micro:  $O(N_\alpha N_a N_c)$
- Macro:  $O(N_a N_c N_\alpha N)$
- Total bilevel:  $O(N_a N_\alpha N) + O(N_\alpha N_a N_c^2 N_u N_t T)$

We proceed with a complexity analysis for the case where we have the constraint  $c_{t_k} = c_0$  and we neglect the age variations in the micro problem:

- Micro problem:  $O(N_c N_u N_t T)$
- Parameters for micro:  $O(N_\alpha N_a)$
- Macro:  $O(N_a N_\alpha N)$
- Total bilevel:  $O(N_a N_\alpha N) + O(N_\alpha N_a N_c N_u N_t T)$

If  $N_a$  and  $N_\alpha$  are of the same order, then the offline computation burdens for the adaptive weights and bilevel dynamic programming algorithms are of the same order. We then store an  $N_a \times N_\alpha$  matrix. The online optimization complexity is proportional to  $N_a N_\alpha N + N_c N_u N_t T$ .

The complexity analysis is summarized in Tables 1 and 2.

## 5. DISCUSSION

**5.1. Asymptotic Analysis.** We propose in this subsection an asymptotic error estimate. The derivation of the estimate relies mostly on Assumption 1, which by the way we did not use in the

	Offline Computation burden
Approach	offline
BF	$N_a N_c N_u N_t T N$
Bilevel	$N_a^2 N_c^2 N + N_a^2 N_c^2 N_u N_t T$
AWA	$N_\alpha N_a N_c^2 N + N_\alpha N_a N_c^2 N_u N_t T$

TABLE 1. The offline computation burdens for the general case

		Computation burden		Memory requirement
Approach		offline	online	
BF	(b)	$N_a N_c N_u N_t T N$	$N_a N_c N_u N_t T$	$N_a N$
Bilevel	(d)	$N_a^2 N + N_a^2 N_c N_u N_t T$	$N_a^2 N + N_a N_c N_u N_t T$	$N_a^2$
	(e)	"	$N_a N_c N_u N_t T$	$N_a N$
AWA		$N_a N_\alpha N + N_\alpha N_a N_c N_u N_t T$	$N_a N_\alpha N + N_c N_u N_t T$	$N_a N_\alpha$

TABLE 2. The computation burdens for the case where  $c$  has to be the same at the beginning of each period

previous sections. We now express formally the approximation of  $V_2^\mu$  envisioned in the complexity analysis:

$$(5.1) \quad \tilde{P}_2^\mu(a_0, c_0, c_F, \alpha) \left\{ \begin{array}{l} \tilde{V}_2^\mu(a_0, c_0, c_F, \alpha) = \inf_{u \in U} \int_0^T [\ell(u(t), t) + \alpha F_a(a_0, c(t), u(t), t)] dt \\ \dot{c}(t) = F_c(a_0, c(t), u(t), t) \\ (a(0), c(0)) = (a_0, c_0) \\ \varphi(c(t), u(t), t) \in A \\ c(T) \geq c_F. \end{array} \right.$$

This approximation consists in neglecting the evolution of  $a$  in the micro optimal control problem. We then set  $\hat{\Delta}a(u) = \int_0^T F_a(a_0, c(t), u(t), t) dt$  and define  $\tilde{\Gamma}$  for  $\tilde{P}_2^\mu$  the same way we defined  $\hat{\Gamma}$  for  $P_2^\mu$ . Last we define  $\tilde{V}_3$  by replacing  $\hat{\Gamma}$  and  $V_2^\mu$  by  $\tilde{\Gamma}$  and  $\tilde{V}_2^\mu$  in the definition of  $V_3$  and  $\tilde{U}_x$  as the set of controls  $u \in U_T = \{u \in L^\infty(0, T) \text{ s.t. } \forall t \in [0, T], u(t) \in \bar{U}\}$  such that  $\varphi(\tilde{c}_{x_0,0}^{u,t}, u(t), t) \in A$  for any  $t \in [0, T]$ , where  $\tilde{c}$  is the flow corresponding to the dynamics (5.1).

First we estimate with Gronwall's lemma the error made on the trajectories.

**Lemma 5.1.** *There exists a constant  $K$  such that for any  $(a_0, c_0, u, t) \in \mathcal{A} \times \mathcal{C} \times U_T \times [0, T]$ ,*

$$(5.2) \quad |a_0 - a_{x_0,0}^{u,t}| \leq K/N \text{ and } |c_{x_0,0}^{u,t} - \tilde{c}_{x_0,0}^{u,t}| \leq K/N,$$

and

$$(5.3) \quad |\Delta a(u) - \tilde{\Delta}a(u)| \leq K/N^2.$$

*Proof.* We get the first inequality using  $F_a \leq L/N$ , the second inequality by combining the  $L$ -Lipschitzianity of  $F_c$ , the first inequality and Gronwall lemma. We get (5.3) combining the  $L/N$ -Lipschitzianity of  $F_a$  with (5.2).  $\square$

We continue with the error estimate.

**Theorem 5.2.** *Let  $x_0 = (a_0, c_0) \in \mathcal{A} \times \mathcal{C}$ . Let  $(\alpha_i, a_i, c_i)$  be some minimizers of  $V_3(x_0, 0)$ . Let  $u_i \in U_{a_i, c_i}$  be a minimizer of  $F_2^\mu(a_i, c_i, c_{i+1}, \alpha_i)$  such that  $u_i \in \tilde{U}_{a_i, c_i}$ . Let  $\psi \in \mathbb{R} \rightarrow \mathbb{R}$  and  $C \geq 0$  be such that*

- $a_{i+1} - a_i \in \tilde{\Gamma}(x_i, c_{i+1}, \psi(\alpha_i))$
- $|\psi(\alpha_i) - \alpha_i| \leq C$
- $\alpha_i \leq NC$

Then

$$(5.4) \quad \frac{\tilde{V}_3(x_0, 0) - V_3(x_0, 0)}{N} \leq \frac{3CK}{N}$$

*Proof.* Set  $\delta a_i = a_{i+1} - a_i$ . By definition of  $\tilde{V}_3$ , we have

$$(5.5) \quad \tilde{V}_3(x_0, 0) - V_3(x_0, 0) \leq$$

$$(5.6) \quad \sum_{i=0}^{N-1} \tilde{V}_2^\mu(x_i, c_{i+1}, \psi(\alpha_i)) - V_2^\mu(x_i, c_{i+1}, \alpha_i) + (\alpha_i - \psi(\alpha_i))\delta a_i + \phi(a_n) - \phi(a_0)$$

$$(5.7) \quad \leq \sum_{i=0}^{N-1} \inf_{u \in \tilde{U}_{x_i}} [\mathcal{L}(u) + \psi(\alpha_i)\tilde{\Delta}a(u)] - \inf_{u \in U_{x_i}} [\mathcal{L}(u) + \alpha_i\Delta a(u)] + (\alpha_i - \psi(\alpha_i))\delta a_i$$

$$(5.8) \quad \leq \sum_{i=0}^{N-1} \inf_{u \in \tilde{U}_{x_i}} [\mathcal{L}(u) + \psi(\alpha_i)\tilde{\Delta}a(u)] - \inf_{u \in \tilde{U}_{x_i}} [\mathcal{L}(u) + \alpha_i\Delta a(u)] + CK/N$$

$$(5.9) \quad \leq \sum_{i=0}^{N-1} \sup_{u \in \tilde{U}_{x_i}} [\psi(\alpha_i)\tilde{\Delta}a(u) - \alpha_i\Delta a(u)] + CK/N$$

$$(5.10) \quad \leq \sum_{i=0}^{N-1} \frac{CK}{N} + \frac{\alpha_i K}{N^2} + \frac{CK}{N} \leq 3CK$$

□

The reverse result can be proved similarly.

To give an intuition of the estimate of  $\alpha$ , observe that  $\Gamma(x_0, c_F, N\alpha)$  is equal to a constant divided by  $N$  for any  $\alpha$ .

**5.2. Extensions.** In this subsection we propose some possible extensions for the adaptative weights algorithm.

**5.2.1. Obstacle.** In many applications, it is possible to buy off the shelf spare parts to replace worn components. Hence we may want to introduce the possibility to buy a replacement in the optimization problem. This could be done with an impulse control: for a fixed price  $p$ , we should be able to reset the age  $a$  to zero. We would get the following dynamic programming principle:

$$(5.11) \quad V_3(a, c, t_k) = \min \left\{ \inf_{c_f, \alpha, \delta a} V_2^\mu(a, c, c_f, \alpha) - \alpha \delta a + V_3(a + \delta a, c_f, t_{k+1}), V_3(0 + \delta a, c_f, t_{k+1}) + p \right\}$$

**5.2.2. Periodicity.** We can include seasonality by having different kinds of periods. For instance, we could model winter and summer days. In this case, one need to perform an offline pre-processing for each kind of day.

5.2.3. *Short-term and long term Randomness.* With Markovian dynamics and final constraints in expectancy, the same arguments should apply. The algorithm should work with stochastic dynamics. Moreover, we could add an integer state with Markovian dynamics to model the type of 'days'.

5.2.4. *Infinite Horizon.* If we add an actualization rate, the arguments should apply for infinite horizon. Then one needs to replace the macro dynamic programming algorithm by either a policy iteration algorithm or a value iteration algorithm.

5.3. **Algorithm.** We propose a bi-level approach that consists in an offline and an online part. For readability we assume that  $c$  should have the same value at the end of each period  $\tilde{c}$ . We already defined  $\Delta a(u)$  and  $\mathcal{L}(u)$ . The inputs of the algorithm are the discretization grids of the age and the parameters, namely  $I_a$  and  $I_\alpha$ . We denote by  $k_0$  the current period number and by  $a_0$  the current age. We use the notation  $\tilde{\phi}(a) = \phi(a)$  if  $a < a_{max}$ ,  $+\infty$  else. For a table  $T$  indexed by  $I_a$   $\mathcal{F}T$  is an interpolation of  $T$  over the grid. The output of the offline algorithm is the pair of tables  $(\Delta_{\alpha,a_0}, \mathcal{L}_{\alpha,a_0})$ . The output of the online algorithm is a control  $u^* \in U_T$ .

```

Data:  $I_a, I_\alpha$ 
Result:  $\Delta_{\alpha,a_0} \mathcal{L}_{\alpha,a_0}$ 
for  $\alpha \in I_\alpha$  do
  for  $a_0 \in I_a$  do
    Solve  $P_2^\mu(a_0, \tilde{c}, \tilde{c}, \alpha)$  ;
    Compute an optimal control  $u$  for  $P_2^\mu(a_0, \tilde{c}, \tilde{c}, \alpha)$  ;
     $\Delta_{\alpha,a_0} \leftarrow \Delta a(u)$  and  $\mathcal{L}_{\alpha,a_0} \leftarrow \mathcal{L}(u)$  ;
  end
end

```

**Algorithm 1:** OFFLINE Algorithm

```

Data:  $k_0, a_0, I_a, I_\alpha, \Delta_{\alpha,a_0}, \mathcal{L}_{\alpha,a_0}$ 
Result:  $u^*$ 
Initialize  $\tilde{V}_{:,k} \in I_a \times [k_0 + 1 \dots N]$ ;
 $\tilde{V}_{:,k} \leftarrow +\infty$ ;
 $\tilde{V}_{:,N} \leftarrow \tilde{\phi}(\cdot)$ ;
for  $k \leftarrow N - 1$  to  $k_0 + 1$  do
  for  $a \in I_a$  do
    for  $\alpha \in I_\alpha$  do
       $\tilde{V}_{a,k} \leftarrow \min\{\mathcal{L}_{\alpha,a} + \mathcal{F}\tilde{V}(a + \Delta_{\alpha,a}, k + 1); \tilde{V}_{a,k}\}$  ;
    end
  end
end
 $\alpha^* \leftarrow \operatorname{argmin}_{I_\alpha} \{\mathcal{L}_{\alpha,a} - \alpha \Delta_{\alpha,a_0} + \tilde{V}(a_0 + \Delta_{\alpha,a_0}, k_0 + 1)\}$ ;
Compute an optimal control  $u^*$  for  $P_2^\mu(a_0, \tilde{c}, \tilde{c}, \alpha^*)$  ;
Return  $u^*$ ;

```

**Algorithm 2:** ONLINE Algorithm

## 6. SIMULATION AND IMPLEMENTATION ON A MICROGRID MODEL

**6.1. Problem Presentation.** A microgrid is an electric system that includes electricity generation units (dispatchable and non dispatchable) and a battery to store energy for later use. Although the battery price is a non negligible part of the total infrastructure cost, the battery aging is rarely taken into account in the control of the grid: this is the source of sub-optimality we propose to deal with. The profusion and complexity of battery aging models is a reason why, to the extent of our knowledge, no generic optimization tools have been proposed yet. It is nonetheless natural to model the aging through a quantity representing the age of the battery that would increase as the battery is used. The scope of our framework is the models for which the battery age dynamics is a controlled first order ordinary differential equation. We use a severity factor model for the aging and solve the simplified optimal control formulation (introduced in [11] and [10] by dynamic programming).

Here is a brief description of the microgrid we consider. The electricity is produced by some non dispatchable units (solar panels) and a dispatchable unit (a diesel generator or the network for instance). At each instant, there is an instantaneous demand (non controllable) for electricity. If there is a production surplus, one can store this surplus in the battery. If there is not enough electricity produced at this instant, and there is some energy left in the battery, one can use the battery to fill the gap or increase the production from the dispatchable unit. Note that the battery is not a perfect storage: if one unit of energy is stored in an empty battery, the total amount of energy we can get from the battery is strictly lower than 1. Of course, there is a cost associated with the production of electricity from the dispatchable unit.

A more detailed and technical description of the system, with the underlying equations, is proposed in §6.3. As already discussed, the battery state can be described by two variables: the state of charge  $c$  and the age  $a$ . The state of charge  $c$  is a normalized quantity that is the ratio of energy stored in the battery over the maximum quantity the battery can store. It is zero when the battery is empty, and 1 when the battery is full.

The age  $a$  is also a normalized quantity in  $[0, 1]$ . We set  $a = 0$  for a brand new battery and  $a = 1$  for a dead one. Obviously, we need to precise the dynamics of  $a$  and  $c$  so that the model makes sense from a physics perspective. Such dynamics can be found in the literature (see §6.2). Very often in the literature instead of the notion of age we find the concept of state of health  $h = 1 - a$ . We prefer the notion of age in this work, to stay coherent with the previous sections.

In order to make the document self contained and the results reproducible we use analytical inputs for the solar power production and for the power consumption (load)

**6.2. Battery Aging Model.** As noted by Koller et al. in [12], battery based solutions for energy storage present the advantage of being deployable without any consideration of the geographic factors and within short schedule thanks to their modularity. In [15] the authors propose three ways to model battery aging: a physico-chemical model, a weighted Amp-hour (Ah) throughput model (or charge counting model) and an event oriented model. In [4] Borhan et al. propose a model predictive control approach where the aging is penalized in the criterion. They implement a weighted Ah throughput model. In [12] Koller et al. propose a discrete time, model predictive control where the aging factor is the Depth of Discharge (DoD), which is modeled with piece-wise affine dynamics.

In [9] Haessig et al. propose a simulation that includes an aging model in order to perform a cost analysis, yet the optimization of the operations is not in the scope of this work. In his PhD thesis [7] Haessig describes a battery aging model (among others) based on the total amount of energy

exchanged during the lifetime of the battery. In [14] Riffonneau et al. use a discrete time dynamic programming approach to solve an optimal power flow problem. The battery aging is proportional to the discharge of the battery. It linearly decreases the capacity of the battery.

In [13], Palma et al. integrate the battery aging in a rolling horizon strategy model. The aging is taken into account in the model using a working zones approach as proposed in [6] and the penalty parameter is the investment cost of the battery.

As argued in the final comments of [7] instead of using a penalization approach, one could impose a maximal aging constraint. The supporting argument is that the appropriateness of an aging profile depends on the time horizon over which the battery is supposed to be in operation. Then in [8] Haessig et al. propose to implement an aging constraint by introducing the notion of exchangeable energy for exchanged energy counting aging model.

To our knowledge there are basically three approaches in the literature to take battery aging into account in an optimization model:

- Some constraints on the control and state variables (for instance to avoid extreme State of Charge values). This requires deciding which constraints to implement. The aging is not directly taken into account in the optimization process. The constraints can be too or not enough conservative.
- A penalization of the aging. This requires choosing an aging model and a penalization parameter.
- An aging constraint. This requires choosing an aging model and the aging level. In addition, one may need a heuristic to implement this constraint if a direct numerical optimization is too burdensome.

Observe that ideally, we should perform an optimization over the whole remaining existence of the microgrid and take into account the impact of the aging on the battery performance. If the battery scheduled lifetime is shorter than the microgrid one, then the optimal aging profile should take into account the possibilities and the conditions (price, etc. ) to buy a new battery. Then it appears that in this very idealistic viewpoint, the optimal control of the microgrid would take into account the aging without requiring the implementation of any penalization or constraint. The aging penalization and the aging constraint are in fact rule of thumbs to incorporate those long term considerations.

Nonetheless, it is hard to conciliate long term optimization and the modeling of the aging related performances variations. Moreover, the numerical resolution of the optimal control formulation problem needs to be fast enough if one envisions a real *online* implementation.

Note that the adaptative weights approach presented in this paper allows for a long term *offline* optimization. The output of the offline optimization is a closed loop optimal penalization parameter which can be then used as input for the *online* (and short term) optimal control problem. Since the age variation within a single day has a negligible impact on the performance, we can neglect those variations for the numerical resolution of the *online* optimal control problem. By doing so, the *online* problem approximation is one dimensional and can be solved efficiently with dynamic programming.

Note that this approach should work as long as we have a continuous time model for the aging (i.e. an ordinary differential equation). In the following we will apply our framework to a severity factor model based on the state of charge of the battery. The whole quantitative formulation is presented in §6.3.

**6.3. The Optimal Control Formulation.** We implement the adaptative weights framework on a simplified version of the continuous time optimal control formulation we introduced in [11] and extended to a stochastic setting in [10]. The unit of time is the hour and  $T = 24$  corresponds to a day. The long term optimal control problem writes

$$(6.1) \quad P_1(a_0, c_0, t_0) \begin{cases} V_1(a_0, c_0, t_0) = \inf_{u \in U} \int_{t_0}^{T_{tot}} \ell(u(t)) dt \\ (a(t_0), c(t_0)) = (a_0, c_0) \\ a(T_{tot}) \leq a_{max} \\ c(t) \in [0.1, 1] \\ (\dot{a}(t), \dot{c}(t)) = (F_a(a(t), c(t), u(t), t), F_c(a(t), c(t), u(t), t)), \end{cases}$$

The control  $u$  corresponds to the power produced by the dispatchable unit (a diesel generator). If the battery is full and renewable production is greater than demand, we can disconnect the battery. To simplify the model, we implement this by allowing  $u$  to be negative. So we have the integral cost

$$(6.2) \quad \ell(u) = \beta(u^+)^2,$$

is a quadratic (and so convex) function associated with the generator consumption of fuel. The value of  $\beta$  as well as the other model parameters are detailed in Table 3. Observe that only the product of  $U_{bat}$  and  $Ah_{bat}$  matters, so to decrease the number of parameters, we only indicate their product. If we denote by  $P_s(t)$  the power produced by the solar panels and by  $P_L(t)$  the load then the state of charge dynamics is

$$(6.3) \quad F_c(c, u, t) = \frac{\rho_i(a)P_i(a, c, u, t) - P_o(a, c, u, t)/\rho_o}{C}$$

with  $P_i(a, c, u, t) = (-u - P_s(t) + P_l(t))^+$  being the power that gets into the battery and  $P_o(a, c, u, t) = (-u - P_s(t) + P_l(t))^-$  being the power that gets out of the battery. We make the choice for simplicity purpose to model the aging impact on the performances by decreasing the efficiency ratio  $\rho_i$ :

$$(6.4) \quad \rho_i(a) = (1 - a)\rho$$

Where  $\rho$  is the initial coefficient for  $a = 0$ . We denote by  $C$  the capacity of the battery. In order to make our numerical experiment reproducible, we take  $T$  periodic functions with analytic expression for the data input. The functions were chosen to be realistic enough. For  $t \in [0, 24]$

$$(6.5) \quad P_S(t) = \max(0, 13 - 0.3(4t - 48)^2)$$

and

$$(6.6) \quad P_L(t) = 3 + 3e^{-0.1(4t-32)^2} + 12e^{-0.03(4t-74)^2}.$$

The aging dynamics corresponds to a severity factor model

$$(6.7) \quad F_a(a(t), c(t), u(t), t) = \eta(c) \frac{P_o(a, c, u, t)}{U_{bat} Ah_{bat}}$$

where

$$(6.8) \quad \eta(c) = \frac{(-4c^2 + 5)}{5}$$

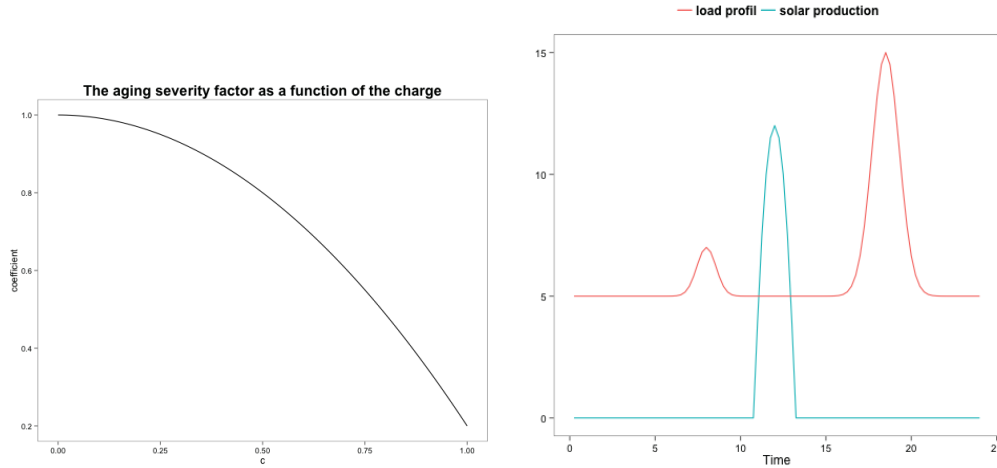


FIGURE 2. ON THE LEFT: the severity factor, assuming the aging is stronger when the battery is depleted, as a function of  $c$ . This function was originally obtained by interpolation of a piece-wise constant severity factors model. Note that in the model the battery aging happens only when energy is taken *from* the battery. ON THE RIGHT: solar production and load during the day. Observe that the solar production only happens during mid-day, whereas the load profile has a constant component and two peaks, one in the morning and one in the evening. For both plots, we made the choice to use analytic expressions to produce synthetic data.

Constant	Interpretation	Value
$\beta$	running cost coef.	0.5
$\rho_i$	efficiency factor (in)	0.95
$\rho_o$	efficiency factor (out))	0.95
$U_{bat}Ah_{bat}$	*	12.5

TABLE 3. Model Contants

is the severity factor (see Figure 2 ) and  $U_{bat}$ ,  $Ah_{bat}$  are parameters that depend on the battery (see3). We set  $a_- = 0$  and  $a_+ = 1000$ . For the numerical experiment, we will set  $a_{max} = 500$  and  $N = 600$ .

## 6.4. Implementation.

6.4.1. *Periodicity.* We impose that at the end of each day, the charge  $c$  should be equal to  $\bar{c}$ , which is a parameter decided upfront. The origine of this additional constraint is operational. Indeed, in a real setting, the optimization is performed regularly on a 24-hours sliding horizon window. Without any final constraint, the optimization program will tend to deplete the battery at the end of its horizon (*end of the world* effect). What is often done to deal with the undesirable effect is to impose that the battery at the end of the time horizon should be as charged as at the beginning. This is



Parameter	$N_a$	$N_c$	$N_u$	$N_t$
Value	100	100	100	4

TABLE 4. Discretization parameters

algorithm	micro (total)	macro	total AWA	bruteforce
Comp. time	12.8 min	0.2 sec	12.8 min	> 10 hours

TABLE 5. The computing times

more or less what we implement here. In dynamic programming, we cannot in general impose hard equality constraints such as  $c_T = \bar{c}$  when doing numerics. So we implement the periodicity of the daily final state of charge using a penalty function (in the running cost for the long term bruteforce problem, in the final constraints for the micro problem):

$$(6.9) \quad \Psi(c) = \begin{cases} (c - \bar{c})M_1 & \text{if } c \geq \bar{c} \\ M_2 & \text{else} \end{cases}.$$

Where  $M_1$  and  $M_2$  are two penalty parameters.

6.4.2. *Discretization and numerical resolution.* We display in Figure 4 the discretization parameters.  $N, T$ . We take  $\alpha$  between 0 and 500 with a discretization of 10 points, to which we add a  $\alpha = +\infty$  point. We use a discretization of 20 points for the adaptive weights algorithm.

We use the optimal control toolbox BOCOPHJB (see [2] and [3]) to solve the optimal control problems. The macro algorithm is coded in the R scripting language. We performed the computation on laptop running OSX 10 with 1.3 GHz and 4 logical cores. The computing times are displayed in Table 5.

6.5. **Results for the micro problem.** Before commenting the results of the numerical experiment for the adaptive weights dynamic programming algorithm (that we will refer to as AWA), it is worth having a closer look at the micro problem (one day time horizon). We display in Figure 3 three simulations for  $(\alpha, a)$  equal to  $(0, 0)$  (solid line),  $(0, 400)$  (dot and dash) and  $(250, 0)$  (dot). We see that the increase in age or  $\alpha$  is associated with a decrease in the total aging within the day. This is done by diminishing the use of the battery: the maximal value of  $c$  is greater for the red curve. The explanations are different for the case  $a = 400$  and  $\alpha = 250$ . For the first one, because the battery efficiency is poor, the quantity that gets effectively stored in the battery is low, so that there is not much to take from the battery during the peaks. The diesel needs to compensate the battery age. For the second one, the battery is efficient, but its use is penalized, so the diesel generator is used during the load peak to decrease the quantity of energy taken from the battery. Observe that, unlike what is seen in the two other cases, the control is flat for a new battery with no penalty. The aging occurs during the two load peaks for all profiles, when the battery is discharging.

We display  $\Delta a(\alpha)$  for  $\alpha = 0$  in Figure 4 (solid curve). Observe that this picture is qualitatively similar to the sketch in Figure 1. Yet it is likely that some *jumps* are the result of the discretization.

We display in Figure 4 and 5  $\Delta a(\alpha)$  and  $\mathcal{L}(\alpha)$  with respect to  $\alpha$  for two values of  $a$  and make the following observations:

- monotonicity of  $\Delta a(\alpha)$ : we observe that  $\Delta a(\alpha)$  is monotone in  $a$ . Then it seems that some discretization artifacts occur.

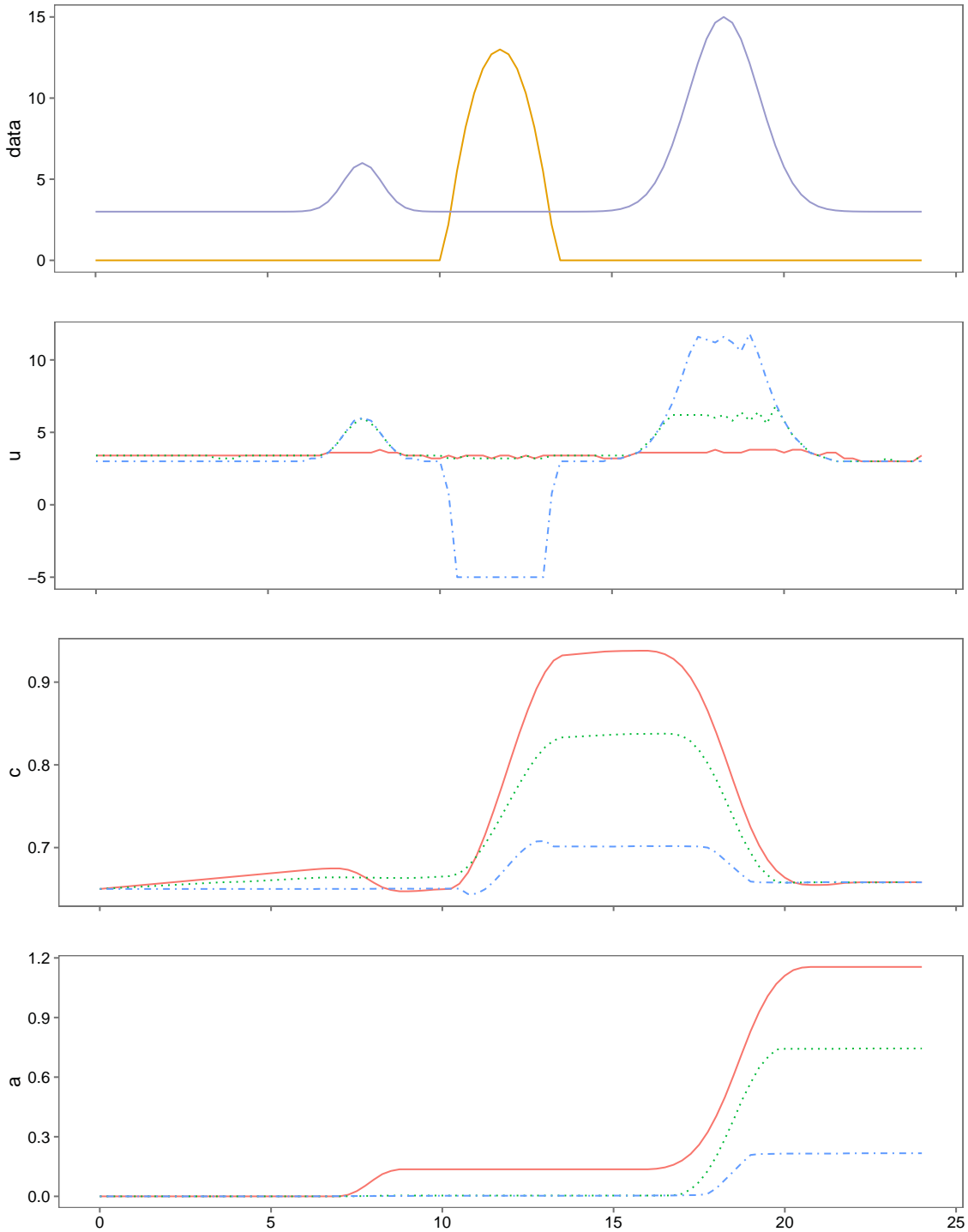
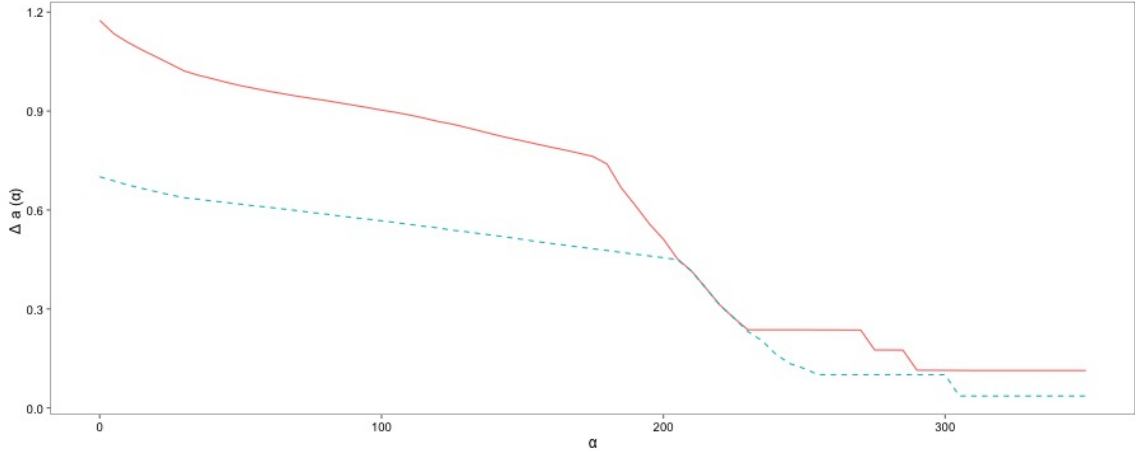
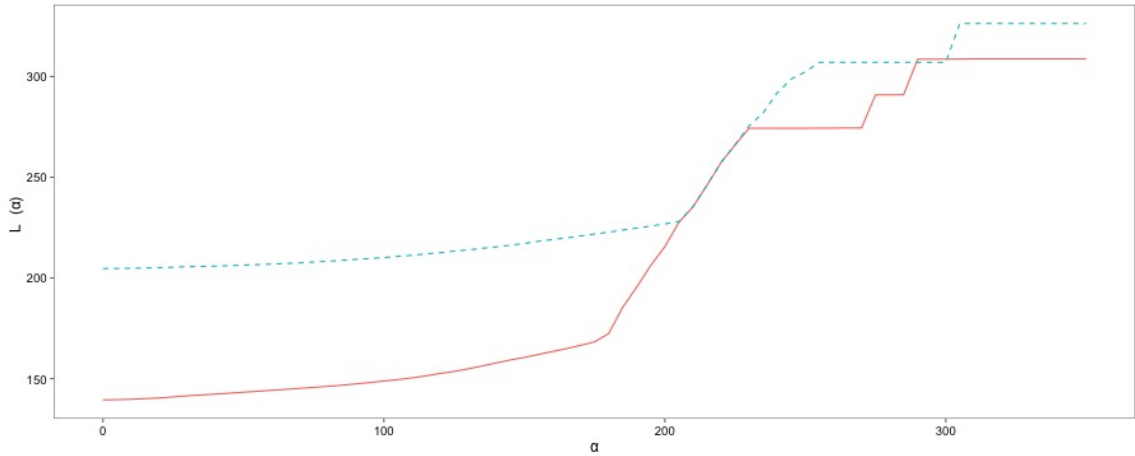


FIGURE 3. Three simulations for  $(\alpha, a)$  equal to  $(0, 0)$  (solid),  $(0, 400)$  (dot and dash) and  $(250, 0)$  (dot). The bell shaped curve correspond to the solar production  $P_s$  and the other one to the load  $P_L$ .

FIGURE 4.  $\Delta a(\alpha)$  for  $a = 0$  (solid line) and  $a = 450$  (dashed line)FIGURE 5.  $\mathcal{L}(\alpha)$  for  $a = 0$  (solid line) and  $a = 450$  (dashed line)

- regularity of  $\Delta a(\alpha)$ : There seem to be smooth and non smooth ranges for  $\alpha$ . It is probable that some of the jumps are due to the discretization choice.
- as expected  $\mathcal{L}(\alpha)$  and  $\Delta a(\alpha)$  are respectively non-decreasing and non-increasing with respect to  $\alpha$  for  $a$  fixed.

Remark on the periodicity of  $c$ . We chose to impose a periodicity condition on the charge  $c$  for simplicity (in particular, the results are easier to represent) and because it makes sense from an operational perspective (see for instance [13]). We compare in Figure 6 the optimal trajectories (computed by dynamic programming) with and without this periodicity condition.

We now proceed with the analysis of the macro part of the adaptive weights algorithm.

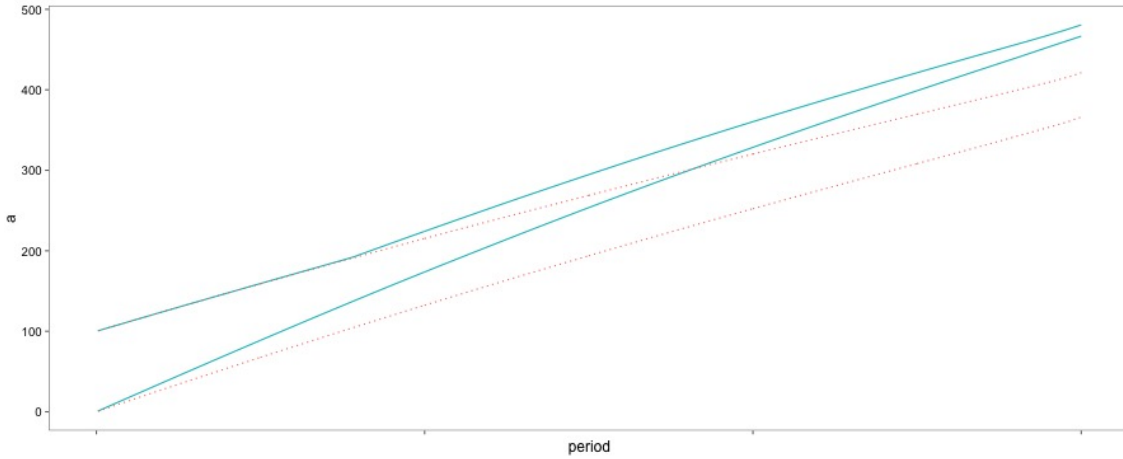


FIGURE 6. The age  $a$  trajectories for  $a(0) = 0$  and  $a(0) = 100$ . The solid lines correspond to the *periodic case* and the dotted line to the unconstrained case. The periodicity constraint increases the overall aging.

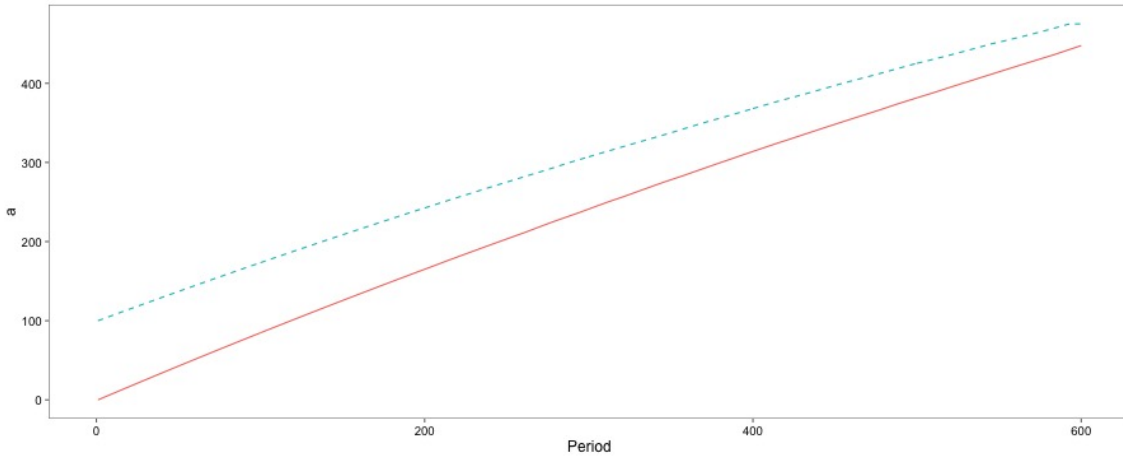


FIGURE 7. Two trajectories computed with AWA for  $a(0) \in \{0, 100\}$

**6.6. Results for the macro dynamic programming phase of AWA (adaptative weights algorithm).** We display in Figure 7 two trajectories corresponding to two different initial ages. We observe that as long as the age is far from  $a_{max}$ , the lines *look* smooth. We display in Figures 8 and 9 the daily weights and the age increments along those same trajectories. The oscillations of the weights have two possible explanations: first the discretization of the set to which  $\alpha$  belongs, second, a jump in  $\Delta a$ , which the oscillations smooth out on average. As explained in the discretization section, on Figure 8 the maximal value of the dotted curve ( $a(0) = 100$ ) corresponds to  $\alpha = +\infty$  as we have added such point in the discretization of  $\alpha$  to freeze the aging.

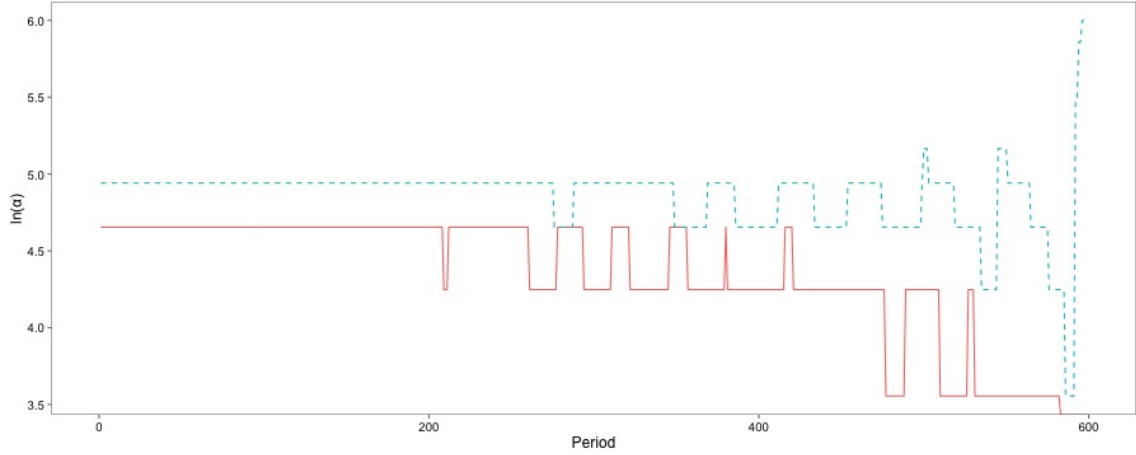


FIGURE 8. The daily weights along the trajectories for  $a(0) \in \{0, 100\}$

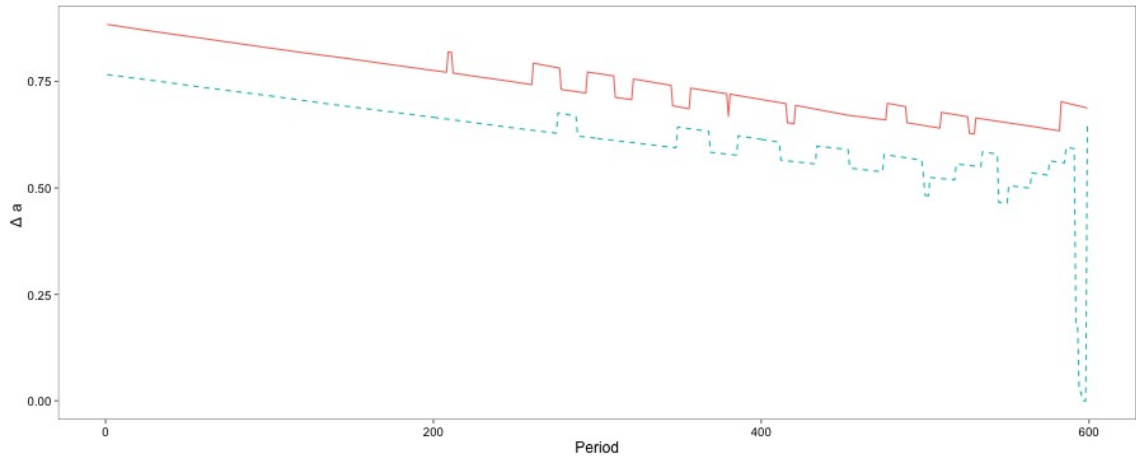


FIGURE 9. The daily age increments along the trajectories for  $a(0) \in \{0, 100\}$

**6.7. Comparison with the brute force results.** We display in Figures 10 and 11 the envelopes of  $c$  and  $u$  for  $a(0) = 0$  obtained with the brute force dynamic programming algorithm. We see that the battery utilization rate decreases as time goes on, while the diesel generator's increases. We observe that the state of charge  $c$  is always slightly above its expected terminal value during the first month, which is possible since the constraint is implemented through a piecewise linear penalt, for numerical reasons.

Last but not least we display on the same plot in Figure 12, two pairs of trajectories computed with AWA and a brute force dynamic programming approach for two initial ages. The corresponding values are displayed in Table 6.

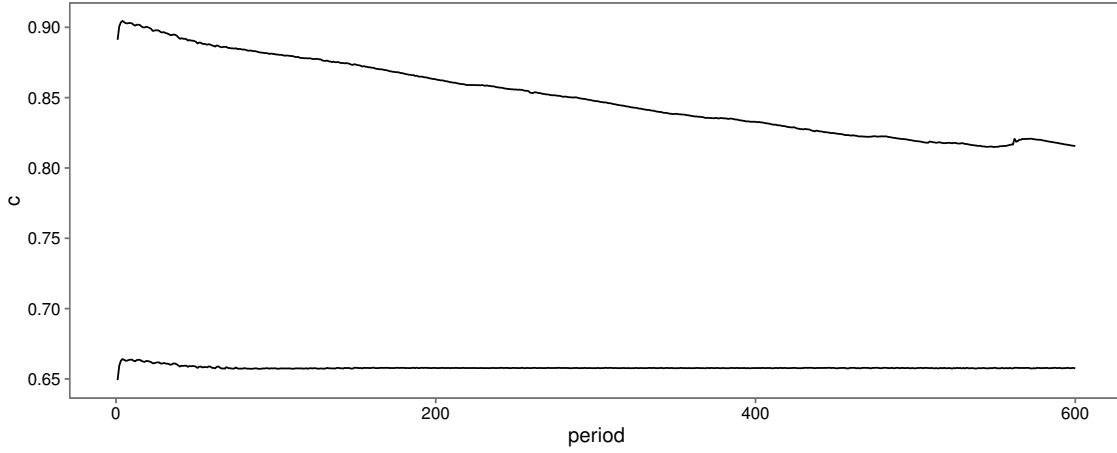


FIGURE 10. Envelope of  $c$  obtained by dynamic programming for the long term problem

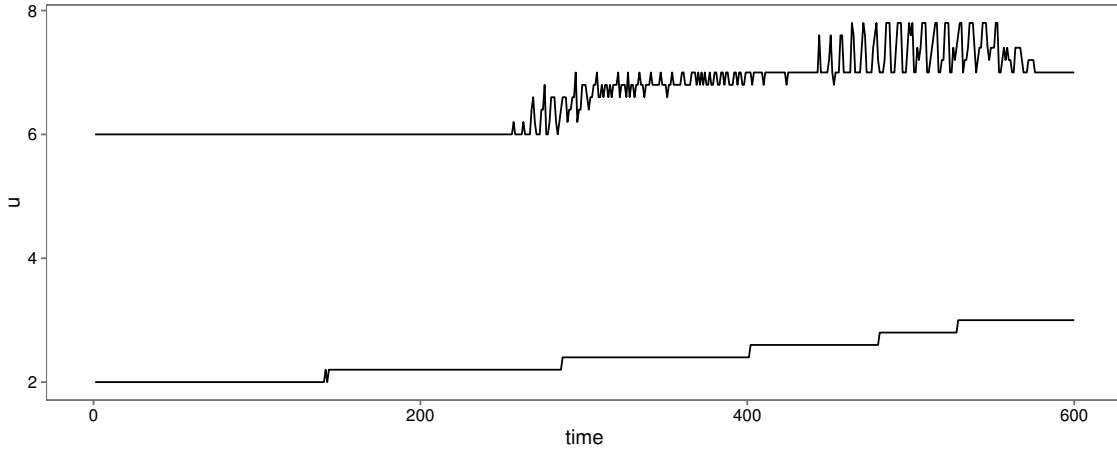


FIGURE 11. Envelope of  $u$  obtained by dynamic programming for the long term problem

start. age.	AWA	BF	(AWA-BF)/BF
$a(0) = 0$	106544	105213	0.012
$a(0) = 100$	116901	111279	0.050

TABLE 6. Estimates of the value function

### 7. CONCLUSION

We have introduced the adaptive weight dynamic programming algorithm (AWA), which is a decomposition technique for problems with periodic data. We tested this algorithm on a toy micro grid problem to integrate the battery aging within the decision process. The trajectories and the

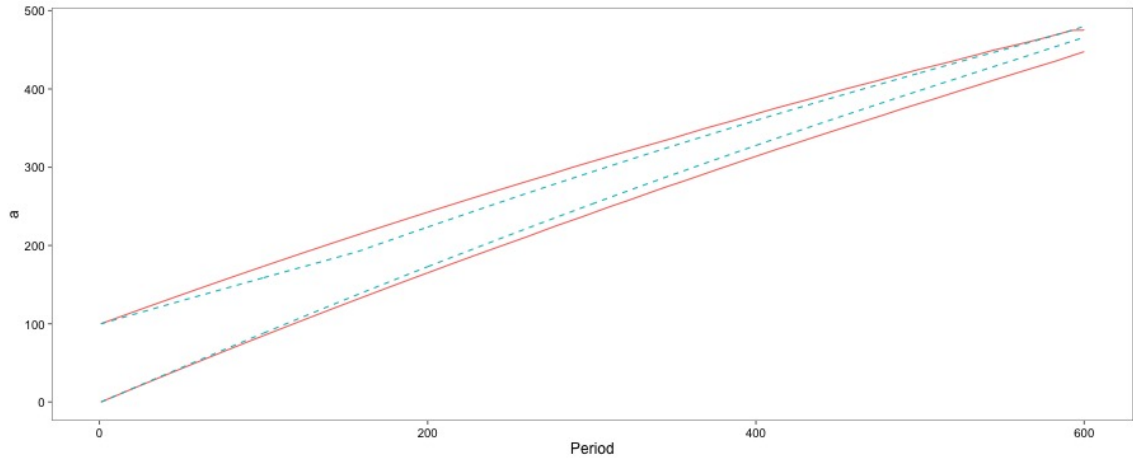


FIGURE 12. The age profile computed with AWA (solid line) and bruteforce dynamic programming (dotted line) for two initial age values.

value functions obtained with AWA are close to those obtained with a bruteforce approach, and the computing times are way smaller.

#### REFERENCES

- [1] J.-P. Aubin and I. Ekeland. Estimates of the duality gap in nonconvex optimization. *Mathematics of Operations Research*, 1(3):225–245, 1976.
- [2] F. Bonnans, D. Giorgi, B. Heymann, P. Martinon, and O. Tissot. Bocophjb 1.0. 1–user guide. Technical report, 2015.
- [3] F. Bonnans, P. Martinon, D. Giorgi, V. Grélard, B. Heymann, L. Jinyan, S. Maindrault, and O. Tissot. Bocop - a collection of examples. Technical report, 2016.
- [4] H. Borhan, M. A. Rotea, and D. Viassolo. Optimization-based power management of a wind farm with battery storage. *Wind Energy*, 16(8):1197–1211, 2013.
- [5] F. Chaplais. Averaging and deterministic optimal control. *SIAM journal on control and optimization*, 25(3):767–780, 1987.
- [6] D. Guasch and S. Silvestre. Dynamic battery model for photovoltaic applications. *Progress in Photovoltaics: Research and applications*, 11(3):193–206, 2003.
- [7] P. Haessig. *Dimensionnement et gestion d’un stockage d’énergie pour l’atténuation des incertitudes de production éolienne*. PhD thesis, Cachan, Ecole normale supérieure, 2014.
- [8] P. Haessig, H. Ben Ahmed, and B. Multon. Energy storage control with aging limitation. In *PowerTech, 2015 IEEE Eindhoven*, pages 1–6. IEEE, 2015.
- [9] P. Haessig, B. Multon, H. Ben Ahmed, S. Lascaud, and L. Jamy. Aging-aware nas battery model in a stochastic wind-storage simulation framework. In *PowerTech (POWERTECH), 2013 IEEE Grenoble*, pages 1–6. IEEE, 2013.
- [10] B. Heymann, F. Bonnans, J Frédéric Bonnans Silva, and G. Jimenez. A stochastic continuous time model for microgrid energy management. 2016.
- [11] B. Heymann, J. F. Bonnans, P. Martinon, F. Silva, F. Lanás, and G. Jimenez. Continuous optimal control approaches to microgrid energy management. 2015.
- [12] M. Koller, T. Borsche, A. Ulbig, and G. Andersson. Defining a degradation cost function for optimal control of a battery energy storage system. In *PowerTech (POWERTECH), 2013 IEEE Grenoble*, pages 1–6. IEEE, 2013.
- [13] R. Palma-Behnke, C. Benavides, F. Lanás, B. Severino, L. Reyes, J. Llanos, and D. Saez. A microgrid energy management system based on the rolling horizon strategy. *IEEE Transactions on Smart Grid*, 4:996–1006, 2013.

- [14] Y. Rifonneau, S. Bacha, F. Barruel, and S. Ploix. Optimal power flow management for grid connected PV systems with batteries. *Sustainable Energy, IEEE Transactions on*, 2(3):309–320, 2011.
- [15] D. U. Sauer and H. Wenzl. Comparison of different approaches for lifetime prediction of electrochemical systems—using lead-acid batteries as example. *Journal of Power Sources*, 176(2):534–546, 2008.