



**HAL**  
open science

## Optimal Control of Storage under Time Varying Electricity Prices

Md Umar Hashmi, Arpan Mukhopadhyay, Ana Bušić, Jocelyne Elias

► **To cite this version:**

Md Umar Hashmi, Arpan Mukhopadhyay, Ana Bušić, Jocelyne Elias. Optimal Control of Storage under Time Varying Electricity Prices. IEEE International Conference on Smart Grid Communications, Oct 2017, Dresden, Germany. hal-01672483

**HAL Id: hal-01672483**

**<https://hal.science/hal-01672483>**

Submitted on 25 Dec 2017

**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.

# Optimal Control of Storage under Time Varying Electricity Prices

Md Umar Hashmi<sup>1</sup>, Arpan Mukhopadhyay<sup>2</sup>, Ana Bušić<sup>1</sup>, and Jocelyne Elias<sup>3</sup>

**Abstract**—End users equipped with storage may exploit time variations in electricity prices to earn profit by doing energy arbitrage, i.e., buying energy when it is cheap and selling it when it is expensive. We propose an algorithm to find an optimal solution of the energy arbitrage problem under given time varying electricity prices. Our algorithm is based on discretization of optimal Lagrange multipliers of a convex problem and has a structure in which the optimal control decisions are independent of past or future prices beyond a certain time horizon. The proposed algorithm has a run time complexity of  $O(N^2)$  in the worst case, where  $N$  denotes the time horizon. To show the efficacy of the proposed algorithm, we compare its run-time performance with other algorithms used in MATLAB’s constrained optimization solvers. Our algorithm is found to be at least ten times faster, and hence has the potential to be used for in real-time. Using the proposed algorithm, we also evaluate the benefits of doing energy arbitrage over an extended period of time for which price signals are available from some ISO’s in USA and Europe.

## I. INTRODUCTION

Dynamic pricing of electricity in wholesale electricity markets has the potential to reduce peak demand [1]. An end user operating under such pricing has to alter its consumption pattern to reduce cost of operation by shifting its peak demand to hours of low price. However, it has been observed that consumption patterns of users do not change significantly with real time electricity price variations and hence consumers end-up paying more in their electricity bill [2]. Installing energy storage by an end user allows them to perform energy arbitrage, i.e., to buy energy when prices are low and sell it when it is expensive. The energy bought can be stored in the battery for later use when the demand arises. Hence, using the battery allows the user to keep the same consumption pattern without increasing their electricity bills under time-varying electricity prices [3]. Additionally, if the user is equipped with a renewable generation unit, then a battery also allows the user to reduce energy uncertainty by storing excess generations.

The real time optimal storage arbitrage requires an optimal control algorithm and accurate information about present and future states. Collectively these two requirements affect the end user arbitrage gains. In the present work we focus on the development of optimal energy arbitrage algorithm which allow users to perform energy arbitrage optimally and efficiently

under time varying electricity prices. We consider the optimal energy arbitrage problem for an end user as a convex problem and propose a solution based on finding the optimal Lagrange multipliers. The key contributions and insights in the paper are the following:

- *Tuning Lagrange multipliers*: Exploiting the piecewise linear cost structure of the arbitrage problem we find that the optimal Lagrange multipliers can only take a discrete set of values corresponding to buying and selling prices of electricity. This transforms the continuous optimization problem into a discrete optimization problem. We indicate how to tune the Lagrange multiplier variables to these prices to find their optimal values.
- *Complexity*: Using the discrete nature of the optimization, we explicitly characterize the worst case running time complexity of the proposed arbitrage algorithm. The worst case run-time complexity is found to be quadratic in the number of instants for which price values are available. Numerical simulations show that our algorithm computes the optimal solution at least ten times faster than standard MATLAB optimization solvers.
- *Sub-horizon*: From the structure of solution obtained using Lagrangian dual, we observe that to find optimal control decisions in a certain period within the total period it is sufficient to consider prices only within a sub-horizon much smaller than the whole duration. In the proposed algorithm, we show how to calculate these sub-horizons.
- *Arbitrage*: Using the proposed algorithm, we evaluate the benefits of performing energy arbitrage for an extended duration of operation (e.g. 5 years). We use real price data and also incorporate realistic losses in the battery.

The problem of optimal energy arbitrage using storage has been the subject of many recent works e.g., [4], [5], [6], [7], [3], [8], [9], [10], [11]. In [4], the demand and price of electricity are assumed to be correlated and stochastic. However, in this setting the user is not allowed to sell energy to the grid. In [5], a closed form solution based on stochastic dynamic programming has been found for the arbitrage problem without considering the ramp constraints of the battery. The objective in [6] is to reduce operational cost of the grid, where demands arrive randomly in time. The work closest to the current paper is [7], where the energy arbitrage problem has been considered for a single battery user incorporating ramp and capacity constraints of the battery. However, in [7] no special property of the cost function other than its convexity is assumed. As a result, characterization of the complexity for the optimal arbitrage algorithm was not possible. In the current paper, we consider a special case where the cost function is piecewise-linear which allows us to discretize the optimal

<sup>1</sup>M.U.H. and A.B are with INRIA, DI ENS, Ecole Normale Supérieure, CNRS, PSL Research University, Paris, France. Email: md-umar.hashmi@inria.fr

<sup>2</sup>A.M is with Computer Communications and Applications Laboratory-2 (LCA-2) of EPFL, Lausanne.

<sup>3</sup>J.E. is with INRIA and Paris Descartes University.

Lagrange multiplier values and characterize the complexity of the optimal arbitrage algorithm.

In [8], [9], [12], the authors consider application of energy storage for not only energy arbitrage but also for providing ancillary services to the grid. Authors in [13] propose a deterministic setting of revenue maximization using spot market prices of electricity available one day ahead. Authors in [11] investigate control of energy storage in context of data-centers. Their model assumes that the battery is fully efficient. In [11], [14], [15] authors use Lyapunov optimization to solve the optimal arbitrage problem.

The rest of the paper is organized as follows. In Section II we introduce the system model. Section III presents a mathematical framework and the proposed algorithm for solving the arbitrage problem. Section IV presents the numerical results. Section V analyses financial feasibility of energy storage performing arbitrage. Section VI concludes the work.

## II. SYSTEM DESCRIPTION

We consider the operation of a single household user of electricity over a fixed period of time. The user is assumed to be equipped with a renewable generation unit and a battery to store excess generation. It is also connected to the electricity grid from where it can buy or to which it can sell energy. Figure 1 shows the block diagram of energy flow for an end user. The objective is to minimize the cost or maximize the profit of the user's operation over a fixed period of time. Qualitatively, the optimal decisions consist of buying energy from the grid when electricity prices are low and selling energy to the grid when the electricity prices are high. Our objective is to find an efficient algorithm for user to make optimal decisions over a period of varying electricity prices. We assume that the energy consumed or sold by the end user over period of constant price is too small to affect the price of the electricity in the same period. So in our case the end user acts as a price-taker.

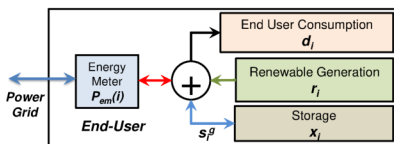


Fig. 1: Power Flow Block Diagram

The total duration,  $T$ , of operation is divided into  $N$  steps, where in each step  $i \in \{1, \dots, N\}$  the price of electricity is assumed to be constant at  $p_i$ . The duration of step  $i$  is denoted as  $h_i$ . Hence,  $T = \sum_{i=1}^N h_i$ . The time duration  $T$  of the user's operation is typically chosen as one day [13] since the pattern in electricity prices repeats with a period of one day; being high during the day and low during the night. We assume each user has a non-elastic demand of  $d_i$  units of energy in time instant  $i$  and it generates  $r_i$  units of energy through renewable sources in the same time instant. We define  $z_i = d_i - r_i \in \mathbb{R}$ . This difference between the demand and the generation can be satisfied either by buying energy from the grid (at price

$p_i$ ) or by the energy obtained from a battery installed at the user's home. In case of excess energy generation, it can be either stored in the battery or sold to the grid. The efficiency of charging and discharging of the battery are denoted by  $\eta_{ch} \in (0, 1]$  and  $\eta_{dis} \in (0, 1]$ , respectively. We denote the change in the energy level of the battery at  $i^{th}$  instant by  $x_i = h_i \delta_i$ , where  $\delta_i$  denotes the storage ramp rate at  $i^{th}$  instant;  $\delta_i > 0$  implies charging and  $\delta_i < 0$  implies discharging. Amount of energy that comes from the grid or renewables or goes out to the grid or end user from the battery at  $i^{th}$  instant is given by

$$s_i = \frac{1}{\eta_{ch}} [x_i]^+ - \eta_{dis} [x_i]^-, \quad (1)$$

where function  $[a]^+ = \max(0, a)$  and function  $[a]^- = -\min(0, a)$ . Hence, the net energy consumption at  $i^{th}$  instant is given as  $P_{em}(i) = d_i - r_i + s_i$ . The cost at  $i^{th}$  instant without the battery is given by  $C_1(i) = z_i p_i$ . Similarly, the cost at  $i^{th}$  instant with the battery is given by  $C_2(i) = (z_i + s_i) p_i$ . The objective of the energy arbitrage problem is to minimize  $\sum_{i=1}^N C_2(i)$  with respect to  $x_i$ . Note that the energy level  $b_i$  in the battery at the end of the  $i^{th}$  instant evolves as  $b_i = b_{i-1} + x_i$ .

*Storage Constraints:* The battery has limited capacity. We incorporate the battery capacity constraints by imposing  $b_i \in [b_{min}, b_{max}]$ ,  $\forall i$ , where  $b_{min}, b_{max}$  is the minimum and the maximum battery capacity (kWh). The *ramp rate constraint* is given as  $\delta_i \in [\delta_{min}, \delta_{max}]$ ,  $\forall i$ , where  $\delta_{min} \leq 0, \delta_{max} \geq 0$  are the minimum and the maximum ramp rates (kW).

## III. OPTIMAL ENERGY ARBITRAGE: FRAMEWORK

The optimal arbitrage problem with battery is defined as the minimization of  $\sum_{i=1}^N C_2(i)$  subjected to the battery constraints. We have

$$\sum_{i=1}^N C_2(i) = \sum_{i=1}^N (z_i + s_i) p_i = \sum_{i=1}^N C_1(i) + C_{storage}^{(i)}(x_i), \quad (2)$$

where  $C_{storage}^{(i)}(x_i)$  is defined as

$$C_{storage}^{(i)}(x_i) = s_i p_i = p_i \left( \frac{1}{\eta_{ch}} [x_i]^+ - \eta_{dis} [x_i]^- \right). \quad (3)$$

Hence, minimizing  $\sum_{i=1}^N C_2(i)$  is equivalent to minimizing  $\sum_{i=1}^N C_{storage}^{(i)}(x_i)$  as  $C_1(i)$  is a constant for every  $i$ . Therefore, the optimal arbitrage problem is given as

$$\begin{aligned} & \text{Minimize}_{x_i \in [X_i^{min}, X_i^{max}], \forall i} \sum_{i=1}^N C_{storage}^{(i)}(x_i) \\ & \text{subject to} \quad b_{min} \leq b_i \leq b_{max}, \quad \forall i, \end{aligned} \quad (P)$$

where  $X_i^{min} = h_i \delta_{min}$  and  $X_i^{max} = h_i \delta_{max}$  for all  $i$ . From (3) it is clear that  $C_{storage}^{(i)}(x_i)$  is a piecewise linear, continuous increasing function in  $x_i$  for all  $i$ . Hence, the objective function for the above problem is convex. The Lagrangian of problem (P) is given as  $\mathcal{L}(x, \alpha, \beta)$

$$= \sum_{i=1}^N \left( C_{storage}^{(i)}(x_i) + \alpha_i (b_{min} - b_i) + \beta_i (b_i - b_{max}) \right). \quad (4)$$

Hence, the Lagrangian dual of (P) is given by

$$\begin{aligned}
\text{(LD)} \quad & \max \quad \phi(\alpha, \beta) \\
& \text{subject to,} \quad \alpha_i, \beta_i \geq 0 \quad \forall i \\
\text{where} \quad & \phi(\alpha, \beta) = \inf_{x_i \in [X_i^{\min}, X_i^{\max}]} \mathcal{L}(x, \alpha, \beta)
\end{aligned}$$

**Theorem III.1.** *There exists a pair  $(x^*, \mu^*)$  with  $\mu^* = (\mu_1^*, \dots, \mu_N^*)$  such that:*

(1.)  $x^* = (x_1^*, \dots, x_N^*)$  is a feasible solution of optimal arbitrage problem (P) implying  $x_i^* \in [X_i^{\min}, X_i^{\max}]$  and  $b_i^* = b_0 + \sum_{j=1}^i x_j^* \in [b_{\min}, b_{\max}]$  for all  $i$ .

(2.) For each  $i$ ,  $x_i^*$  minimizes  $C_{\text{storage}}^{(i)}(x) - \mu_i^* x$ . Here,  $\mu_i^*$  is called the optimal accumulated Lagrange multiplier for time instant  $i$  and is related to the dual optimal solution  $(\alpha^*, \beta^*)$  as follows  $\mu_i^* = \sum_{j=i}^N (\alpha_j^* - \beta_j^*)$ .

(3.) Optimal accumulated Lagrange multiplier,  $\mu_i^*$ , at any time instant  $i$  satisfies the following recursive conditions:

- $\mu_{i+1}^* = \mu_i^*$ , if  $b_{\min} < b_i^* < b_{\max}$
- $\mu_{i+1}^* \leq \mu_i^*$ , if  $b_i^* = b_{\min}$
- $\mu_{i+1}^* \geq \mu_i^*$ , if  $b_i^* = b_{\max}$

where  $b_i^* = b_0 + \sum_{j=1}^i x_j^*$

(4.) Additionally, the optimal accumulated Lagrange multiplier  $\mu_N^*$  at the last instant  $N$  satisfies

- $\mu_N^* = 0$ , if  $b_{\min} < b_N^* < b_{\max}$
- $\mu_N^* \geq 0$ , if  $b_N^* = b_{\min}$

For any pair  $(x^*, \mu^*)$  satisfying the above conditions,  $x^*$  solves the optimal arbitrage problem (P).

The proof of Theorem III.1 is provided in Appendix A. We note that the optimality conditions stated in Theorem III.1 do not depend on the particular structure of the cost function and are valid as long as  $C_{\text{storage}}^{(i)}(x_i)$  is a convex function with respect to  $x_i$  for each  $i$ . Next, we characterize, for each instant  $i$ , the relationship between the optimal decision  $x_i^*$  and the optimal accumulated Lagrange multiplier  $\mu_i^*$  using the particular nature of the cost function  $C_{\text{storage}}^{(i)}(x_i)$ . This will be useful in formulating the optimal arbitrage algorithm.

**Remark 1.** From condition (2) of Theorem III.1 we have that the optimal control decision  $x_i^*$  in the  $i$ th instant minimizes the function  $C_{\text{storage}}^{(i)}(x) - \mu_i^* x$  for  $x \in [X_i^{\min}, X_i^{\max}]$ . Now from (3) we obtain that for a given  $\mu_i^* = \mu$  the optimal decision  $x_i^*(\mu)$  is given by

$$x_i^*(\mu) = \begin{cases} [X_i^{\min}, X_i^{\min}], & \text{if } \mu < p_{\text{dis}}(i), \\ [X_i^{\min}, 0], & \text{if } \mu = p_{\text{dis}}(i), \\ [0, 0], & \text{if } p_{\text{ch}}(i) > \mu > p_{\text{dis}}(i), \\ [0, X_i^{\max}], & \text{if } \mu = p_{\text{ch}}(i), \\ [X_i^{\max}, X_i^{\max}], & \text{if } \mu > p_{\text{ch}}(i), \end{cases} \quad (5)$$

where  $p_{\text{ch}}(i) = p_i/\eta_{\text{ch}}$  and  $p_{\text{dis}}(i) = p_i\eta_{\text{dis}}$ . Note that  $x_i^*(\mu)$  is a set-valued function in  $\mu$ : for  $\mu = p_{\text{ch}}(i)$  or  $\mu = p_{\text{dis}}(i)$ ,  $x_i^*(\mu)$  takes an *envelope* of values and for any other value of  $\mu$  it is a singleton set. It is also important to note from (5) that if  $\mu_1 \leq \mu_2$  then  $x_i^*(\mu_1) \preceq x_i^*(\mu_2)$ , where for two sets  $A$

and  $B$  we say  $A \preceq B$  (resp,  $A \prec B$ ) if  $a \leq b$  (resp  $a < b$ ) for all  $a \in A$  and for all  $b \in B$ . The above monotonicity property also holds for the sets  $b_i^*(\mu)$ , defined recursively as  $b_i^*(\mu) = b_{i-1}^*(\mu) + x_i^*(\mu)$  for  $i \geq 1$  and  $b_0^*(\mu) = b_0$ . Here, addition of two intervals  $[a, b]$  and  $[c, d]$  denotes the interval  $[a + c, b + d]$ .

#### A. Proposed Algorithm

We now propose an algorithm which finds a pair  $(x^*, \mu^*)$  that satisfies all the conditions in Theorem III.1 and therefore solves (P). The pseudo code of the proposed algorithm is shown as Algorithm 1 below. Note from Theorem III.1 condition (3) that  $\mu_{i+1}^*$  may differ from  $\mu_i^*$  only when  $b_i^* = b_{\max}$  or  $b_i^* = b_{\min}$ . Hence, if the battery level at the end of a time instant lies strictly within the battery capacity limits, then there is no change in the value of the optimal accumulated Lagrange multiplier. Using this key idea, in the proposed algorithm, we divide the total duration  $T$  into groups, indexed as  $\{1, 2, \dots, M\}$ , of consecutive time instants such that for all instants  $i$  belonging to the same group  $K \in \{1, 2, \dots, M\}$  the value of the accumulated Lagrange multiplier  $\mu_i^*$  is the same, denoted as  $\mu_K$ . We call each such group as a sub-horizon. At the end of each sub-horizon, the battery energy level touches either  $b_{\max}$  or  $b_{\min}$ . The number  $M$  of sub-horizons, the start and end instants of each sub-horizon  $K$ , and the value of  $\mu_K$  and optimal actions in each sub-horizon  $K$  depend on the problem instance and are determined recursively as described below.

Suppose we have identified the first  $K - 1$  ( $K \geq 1$ ) sub-horizons and the optimal actions in all instants belonging to these sub-horizons. Call the last instant identified to be in the  $(K - 1)$ th sub-horizon as  $i_{K-1}$ . If  $i_{K-1} = N$ , then we have already covered whole period  $T$ . If  $i_{K-1} < N$ , then we proceed to identify the next sub-horizon  $K$ , i.e., the values of  $i_K$ ,  $\mu_K$ , and the optimal decisions for the time instants  $i \in [i_{K-1} + 1, i_K]$ . To identify the sub-horizon  $K$ , we start with instant  $i_{K-1} + 1$  and a guess value of  $\mu_K \geq 0$  for that sub-horizon.<sup>1</sup> Now, for the chosen value of  $\mu_K$ , the values of  $x_i^*(\mu_K)$  and  $b_i^*(\mu_K)$  are computed as described in Remark 1 for all consecutive time instants  $i > i_{K-1}$  until we reach a time instant  $i = i_{\text{break}}$  for which one of the following conditions is satisfied (we call these as the *violation conditions*):

- C1:  $b_{i_{\text{break}}}^*(\mu_K) \prec \{b_{\min}\}$ .
- C2:  $\{b_{\max}\} \prec b_{i_{\text{break}}}^*(\mu_K)$ .
- C3:  $i_{\text{break}} = N, b_{\min} \notin b_N^*(\mu_K), \mu_K > 0$ .

If no such  $i$  is found even after reaching  $i = N$ , then  $K$  is identified as the last sub-horizon and we set  $i_K = N$  (and lines 24–30 of the pseudo code are executed). If  $\mu_K > 0$ , then  $b_N^* = b_{\min}$ ; else  $b_N^*$  is taken to be some value in the set  $[b_{\min}, b_{\max}] \cap b_N^*(\mu_K)$  to satisfy condition (4) of Theorem III.1. The optimal decisions  $x_i^*$  and  $b_i^*$  for  $i \in [i_{K-1} +$

<sup>1</sup>For the first sub-horizon  $K = 1$  (that includes the first time instant) the starting guess value of  $\mu_1$  is taken to be 0 and for every other sub-horizon  $K > 1$ , the starting guess value of  $\mu_K$  is taken to be equal to  $\mu_{K-1}$ . Note that these choices are arbitrary and the algorithm does not depend on these choices.

1, N] are then found by using the algorithm `BackwardStep`, shown as in Algorithm 2 below. The proposed algorithm then terminates. The algorithm `BackwardStep` will be discussed in more detail later.

If condition C1 above is satisfied, then for the chosen value of  $\mu_K$ , the battery capacity limit is violated at the instant  $i_{\text{break}}$  since the set  $b_{i_{\text{break}}}(\mu)$  lies strictly below  $b_{\min}$ . The value of  $\mu_K$  is then increased to  $\min\{p > \mu : p \in (p_{\text{ch}}(i), p_{\text{dis}}(i); i_{K-1} < i \leq i_{\text{break}})\}$ . Otherwise, if C2 or C3 above is satisfied, then  $\mu_K$  is decreased to  $\max\{p < \mu : p \in (p_{\text{ch}}(i), p_{\text{dis}}(i); i_{K-1} < i \leq i_{\text{break}})\}$ . With the updated value of  $\mu_K$  we again repeat the same process as discussed above to obtain a new value of  $i_{\text{break}}$ . Since  $x_i^*(\mu_K)$  and  $b_i^*(\mu_K)$  are monotonically non-decreasing functions in  $\mu_K$ , the potential effect of the update of  $\mu_K$  is that  $i_{\text{break}}$  is pushed to a later instant. The update of  $\mu_K$  is repeated so long as  $i_{\text{break}}$  increases (or remains the same) as compared to its old value (stored in  $i_{\text{memory}}$ ). If the value of  $i_{\text{break}}$  decreases with an updated value of  $\mu_K$ , then for the previous value  $\mu_{\text{memory}}$  of  $\mu_K$  there must have been an instant  $i \in [i_{K-1} + 1, i_{\text{memory}}]$ , where  $b_{\max} \in b_i^*(\mu_K)$  (if the violation occurred due to C1) or  $b_{\min} \in b_i^*(\mu_K)$  (if violation occurred due to C2 or C3). This is because both  $\mu_{\text{memory}}$  and  $\mu_K$  always lie in the range  $[p_{\text{dis}}(i), p_{\text{ch}}(i)]$  for all  $i > i_{K-1}$ . Since  $b_i^*(p_{\text{dis}}(i)) \cap b_i^*(p_{\text{ch}}(i)) \neq \emptyset$  an update of  $\mu_K$  cannot cause  $b_i^*(\mu_K)$  to completely go above  $b_{\max}$  or below  $b_{\min}$  if  $\{b_{\min}\} < b_i^*(\mu_{\text{memory}}) < \{b_{\max}\}$ .

At this point in the algorithm,  $\mu_K$  and  $i_{\text{break}}$  are switched back to their previous values stored in  $\mu_{\text{memory}}$  and  $i_{\text{memory}}$ , respectively. This value of  $\mu_K$  is identified to be the final value of the optimal accumulated Lagrange multiplier in the sub-horizon  $K$ . We set  $i_K$  to be the latest time instant  $i \in [i_{K-1} + 1, i_{\text{break}}]$  for which  $b_{\min} \in b_i^*(\mu_K)$  or  $b_{\max} \in b_i^*(\mu_K)$ . The value of  $b_{i_K}^*$  is chosen to be  $b_{\min}$  in the former case and  $b_{\max}$  in the later case.

Finally, for each  $i$  in the range  $i_{K-1} + 1 \leq i < i_K$ , the optimal battery level  $b_i^*$  is found from  $b_{i+1}^*$  through the function `BackwardStep` which uses the backward recursion  $b_i^* = (b_{i+1}^* - x_{i+1}^*(\mu_K)) \cap b_i^*(\mu_K) \cap [b_{\min}, b_{\max}]$ . If the above backward recursion returns a set, then any arbitrary value in the set is chosen to be the optimal battery level. We note here that the optimal solution to (P) need not be unique since its objective function is not strictly convex.

*Complexity Analysis:* In order to derive the worst case time-complexity of the proposed algorithm, we consider the computation of  $x_i^*(\mu)$  for a given value of  $\mu$ , in a given time step  $i \in \{1, 2, \dots, N\}$  as the basic operation. Let the length of the  $K^{\text{th}}$  sub-horizon be denoted by  $W_K$ ,  $K \in \{1, 2, \dots, M\}$ . Clearly, in sub-horizon  $K$  we may have to update the value of  $\mu_K$  at most  $2W_K$  times (for each instant  $i$  in the sub-horizon two possible values  $\{p_{\text{ch}}(i), p_{\text{dis}}(i)\}$  may be checked) to obtain the optimal Lagrange multiplier value  $\mu_K^*$ . For each update of  $\mu_K$ , the optimal decisions  $x_i^*(\mu_K)$  need to be calculated at most for all instants  $i$  in the sub-horizon  $K$ . Hence, the total number of basic operations in the forward step is  $W_K^2$ . Finally, for the `BackwardStep` the value of  $x_i^*(\mu_K^*)$  has to be computed for all  $i$  in the sub-horizon  $K$ .

Hence, the total number of operations in sub-horizon  $K$  is  $O(W_K^2)$ . Therefore, the total time-complexity of the algorithm is  $O(\sum_{K=1}^M W_K^2) = O(N^2)$  since  $\sum_{K=1}^M W_K = N$ .

---

### Algorithm 1 Proposed optimal arbitrage algorithm

---

**Inputs:**  $N, T, h = (h_1, h_2, \dots, h_N), p = (p_1, p_2, \dots, p_N), b_0$   
**Parameters:**  $b_{\max}, b_{\min}, \delta_{\max}, \delta_{\min}$   
**Outputs:**  $x^* = (x_1^*, x_2^*, \dots, x_N^*), b^* = (b_1^*, b_2^*, \dots, b_N^*), \mu^* = (\mu_1^*, \mu_2^*, \dots, \mu_N^*)$   
**Initialize:**  $K = 1; \mu_K = \mu_{\text{memory}} = 0; i_{K-1} = i_K = i_{\text{memory}} = 0; \text{BreakFlag} = 0$

- 1: **while**  $i_K < N$  **do**
- 2:     **for**  $i = i_{K-1} + 1$  to  $N$  **do**
- 3:         Compute  $x_i^*(\mu_K)$  and  $b_i^*(\mu_K)$
- 4:         **if** C1 or C2 or C3 holds **then**
- 5:              $\text{BreakFlag} \leftarrow 1; i_{\text{break}} \leftarrow i$
- 6:             **Break**
- 7:         **end if**
- 8:     **end for**
- 9:     **if**  $\text{BreakFlag} = 1$  and  $i_{\text{break}} \geq i_{\text{memory}}$  **then**
- 10:          $\text{BreakFlag} \leftarrow 0; i_{\text{memory}} \leftarrow i_{\text{break}}; \mu_{\text{memory}} \leftarrow \mu_K$
- 11:         **if**  $b_i^*(\mu_K) < \{b_{\min}\}$  **then**
- 12:              $\mu_K \leftarrow \min\{p > \mu : p \in (p_{\text{ch}}(i), p_{\text{dis}}(i); i_{K-1} < i \leq i_{\text{break}})\}$
- 13:             **else**
- 14:                  $\mu_K \leftarrow \max\{p < \mu : p \in (p_{\text{ch}}(i), p_{\text{dis}}(i); i_{K-1} < i \leq i_{\text{break}})\}$
- 15:             **end if**
- 16:         **else if**  $\text{BreakFlag} = 1$  and  $i_{\text{break}} < i_{\text{memory}}$  **then**
- 17:             **if** C1 is True **then**
- 18:                  $i_K \leftarrow \max\{i \in [i_{K-1} + 1, i_{\text{memory}}] : b_{\max} \in b_i^*(\mu_{\text{memory}})\}$
- 19:                  $b_{i_K}^* = b_{\max}$
- 20:             **else if** C2 or C3 is True **then**
- 21:                  $i_K \leftarrow \max\{i \in [i_{K-1} + 1, i_{\text{memory}}] : b_{\min} \in b_i^*(\mu_{\text{memory}})\}$
- 22:                  $b_{i_K}^* = b_{\min}$
- 23:             **end if**
- 24:              $\mu_K \leftarrow \mu_{\text{memory}}; \text{BreakFlag} \leftarrow 0; i_{\text{break}} \leftarrow i_{\text{memory}}$
- 25:              $i_{\text{memory}} \leftarrow i_K$
- 26:              $\text{BackwardStep}(\mu_K, i_{K-1}, i_K, b^*, x^*, \mu^*)$
- 27:              $\mu_{K+1} \leftarrow \mu_K; K \leftarrow K + 1$
- 28:         **else**
- 29:              $i_K \leftarrow N;$
- 30:             **if**  $\mu_K > 0$  **then**
- 31:                  $b_N^* \leftarrow b_{\min}$
- 32:             **else**
- 33:                  $b_N^* \leftarrow [b_{\min}, b_{\max}] \cap b_N^*(\mu_K)$
- 34:             **end if**
- 35:              $\text{BackwardStep}(\mu_K, i_{K-1}, i_K, b^*, x^*, \mu^*)$
- 36:         **end if**
- 37:     **end while**

---



---

### Algorithm 2 BackwardStep( $\mu_K, i_{K-1}, i_K, b^*, x^*, \mu^*$ )

---

**Inputs:**  $\mu_K, i_{K-1}, i_K, b^*, x^*, \mu^*$   
**Function:** Computes components of the optimal vectors  $b^*, x^*$  in the range  $[i_{K-1} + 1, i_K - 1]$  using backward recursion  
**Initialize:**  $i \leftarrow i_K - 1$

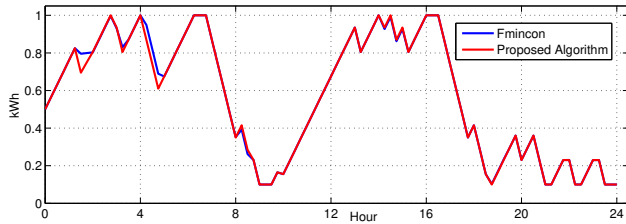
- 1: **while**  $i \geq i_{K-1} + 1$  **do**
- 2:      $b_i^* \leftarrow (b_{i+1}^* - x_{i+1}^*(\mu_K)) \cap b_i^*(\mu_K) \cap [b_{\min}, b_{\max}]$
- 3:      $x_{i+1}^* \leftarrow b_{i+1}^* - b_i^*$
- 4:      $\mu_i^* \leftarrow \mu_K$
- 5: **end while**

---

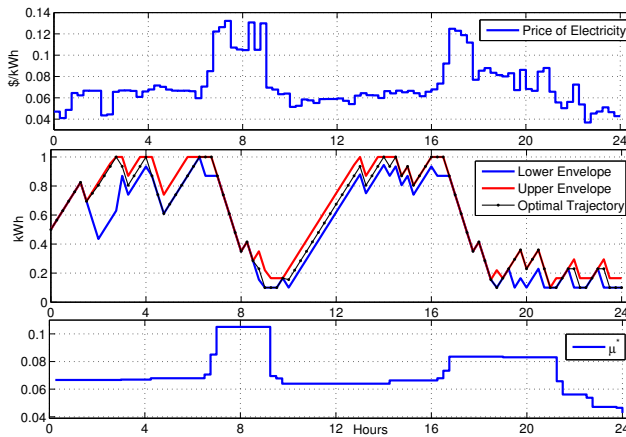
## IV. NUMERICAL EVALUATION

We solve the optimal arbitrage problem using the proposed algorithm described in Section III. The performance of the

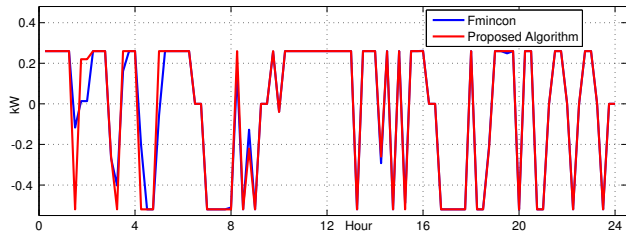
proposed algorithm is compared with Linear Programming (LP) and Matlab's Fmincon based constrained minimization (function evaluations set to 9600), in terms of run time and energy arbitrage gains. However, LP can only be evaluated for lossless battery as the objective function of (P) is linear in the lossless case.



(a) Comparison of optimal battery level ( $b^*$ )



(b) Proposed algorithm results for lossless battery



(c) Comparison of optimal ramp rates

Fig. 2: Comparison for Lossless battery

The battery parameters used for numerical evaluation are:  $b_{\max} = 1$  kWh,  $b_{\min} = 0.1$  kWh,  $\delta_{\max} = 0.26$  kW,  $\delta_{\min} = -0.52$  kW. Real-Time locational marginal pricing data for 21st December 2016 from NYISO [16] is used to calculate optimal ramping trajectory. The sampling time of price signal is  $h = 0.25$  hours. Simulations are conducted using a laptop PC with Intel Core i5-4200M, 2.50GHz processor and 8 GB RAM.

Fig 2 shows the results for lossless battery, i.e.  $\eta_{\text{ch}} = 1, \eta_{\text{dis}} = 1$ . The first plot of Fig 2b shows the electricity price for a day [16], and the other plots show the optimal trajectory of battery level and the value of  $\mu$  with time. It can be observed from Fig 2a and 2c that the proposed algorithm

has not violated the constraints for the battery and the results are using proposed algorithm and Matlab's Fmincon are very similar. However, the run time of the proposed algorithm is significantly lower than other methods as shown in Table I.

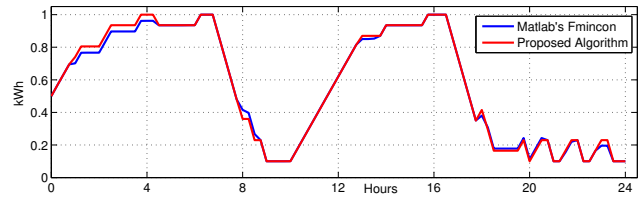
TABLE I: Comparison of performance for lossless battery

Algorithm Type	Run Time (sec)	Profit (\$)
Proposed Algo	0.1967	0.1403245
Linear Program	1.4873	0.1403245
Matlab's Fmincon	23.0526	0.14027568

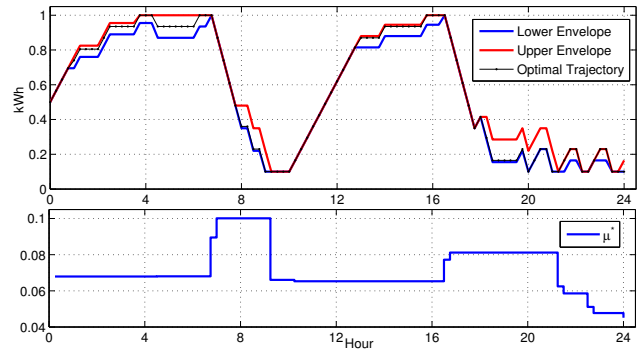
Fig 3 shows the results of the proposed algorithm for a lossy battery, i.e.  $\eta_{\text{ch}} = 0.95, \eta_{\text{dis}} = 0.95$ . From numerical

TABLE II: Comparison of performance for lossy battery

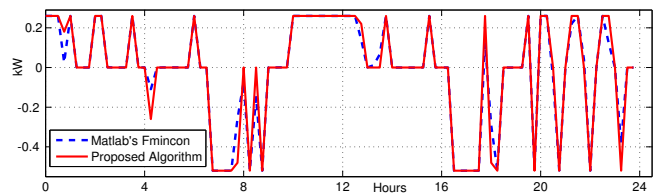
Algorithm Type	Run Time (sec)	Profit (\$)
Proposed Algo	0.164189	0.1193289
Matlab's Fmincon	23.41217	0.11923956



(a) Comparison of optimal battery level ( $b^*$ )



(b) Proposed algorithm results for lossy battery



(c) Comparison of optimal ramp rates

Fig. 3: Comparison for lossy battery

evaluation it is observed that the proposed algorithm significantly reduces the run time compared to Matlab's Fmincon with reasonable number of evaluations, 9600 in this case. Table II provides a comparison of results for a lossy battery.

## V. FEASIBILITY OF ENERGY STORAGE ARBITRAGE

Energy storage will act as energy buffers for future power-grid. However, energy storage being expensive its financially feasibility analysis is essential. We present an approximate methodology to consider the net average capacity of the battery over its entire life. Using this average available capacity, we calculate energy arbitrage gains in a deterministic setting for a 1 kWh rated capacity battery with 1 day as optimization horizon.

### A. Net Average Available Capacity of Battery

The average battery capacity available over its entire life to the user will be close to 50 to 70 % of the original storage capacity when the battery is new. Oversizing is a crucial factor jeopardizing the financial viability of energy storage. We consider following sources of discounting:

(1.) *Efficiency of Cycling* : The losses incurred during cycles of charging and discharging ( $\approx 70$  to  $95\%$ )

(2.) *Efficiency of Converter* : AC to DC and DC to AC conversion incurs losses ( $\approx 90$  to  $98\%$ )

(3.) *Performance Degradation*: Battery reaches End-of-Life (EOL) if the maximum battery capacity reduces to 80 % of the rated battery capacity. Life of a battery in years ranges from 5-20 years and in cycles from 1000 to 20000 cycles depending on type of battery, *whatever is achieved first marks EOL*.

(4.) *Optimal SoC Band*: overcharging or overdischarging a battery significantly affect its life.

The discount factors assumed are: (1.)  $\eta_{ch} = 0.95, \eta_{dis} = 0.95$ , thus the roundtrip battery efficiency equals 0.9025, (2.) Converter Efficiency = 0.95, (3.) Average Capacity due to Degradation = 0.9, (4.)  $[B_{max} - B_{min}]/\text{Capacity} = [0.98 - 0.1]/1 = 0.88$ . Therefore Net Available Capacity = 0.679. This implies that the earnings of 0.679 kWh available capacity in 5 years should match price paid by end user in buying 1 kWh, for achieving 5 yrs. simple payback period. Discount factors due to maintenance cost and self-discharge will further reduce the net average capacity of the battery.

### B. Evaluation

The present battery cost is around \$ 350 - 500 per kWh. One year (2016) simulations for real time electricity prices of Nord Pool, PJM, ISONE, MISO, NYISO, ERCOT, CAISO and PG&E's TOU prices are done to calculate the value of energy storage. The discounted returns calculated for Battery

TABLE III: The price signals evaluated are listed below

Region/ISO	Pricing	Sample
NordPool [17]	Real Time	1 hour
PG&E [18]	ToU	-
CAISO (Average Price) [19]	Real Time	5 min
PJM (Zone AEP) [19]	Real Time	1 hour
ERCOT (Zone LZ-Huston) [19]	Real Time	1 hour
ISONE (Zone .Z.SEMASS) [19]	Real Time	1 hour
MISO(Zone Michigan Hub) [19]	Real Time	1 hour
NYISO (Zone N.Y.C.) [19]	Real Time	1 hour

model 1, 2, 3 listed in Table IV, V and VI. The results take

TABLE IV: **Battery 1:**  $\delta_{max}=0.26$  kW,  $\delta_{min}=-0.52$  kW

Region or ISO	Cumulative Gains in 2016	Operational Cycles in 2016	Discounted Gains in 5 yrs
NordPool	€0.991	1748	€4.3
PG&E	\$ 4.38	184	\$ 18.7
CAISO	\$ 37.9	914	\$ 162.0
PJM	\$ 11.2	573	\$ 47.9
ERCOT	\$ 18.6	430	\$ 79.5
ISONE	\$ 15.3	687	\$ 65.4
MISO	\$ 10.5	595	\$ 44.9
NYISO	\$ 23.3	700	\$ 99.6

TABLE V: **Battery 2:**  $\delta_{max} = 1kW$ ,  $\delta_{min} = -1kW$

Region or ISO	Cumulative Gains in 2016	Operational Cycles in 2016	Discounted Gains in 5 yrs
NordPool	€1.09	2836	€4.7
PG&E	\$ 4.38	184	\$ 18.7
CAISO	\$ 73.2	2008	\$ 312.9
PJM	\$ 16.1	825	\$ 68.8
ERCOT	\$ 25.02	534	\$ 107.0
ISONE	\$ 23.51	1082	\$ 100.5
MISO	\$ 15.52	860	\$ 66.34
NYISO	\$ 36.32	1225	\$ 155.3

TABLE VI: **Battery 3:**  $\delta_{max} = 2kW$ ,  $\delta_{min} = -2kW$

Region or ISO	Cumulative Gains in 2016	Operational Cycles in 2016	Discounted Gains in 5 yrs
CAISO	\$ 125.03	3240	\$ 534.4

into account the net average battery capacity calculated in section V-A. The discounted returns are significantly lower than the present cost of battery. CAISO, NYISO and ERCOT is relatively more profitable but still lower than the initial investment made by end user. For small price variations arbitrage could still be profitable if (Selling Price) > (Buying Price)/(Roundtrip Efficiency), but the revenue generated with per cycle of operation of the battery will be small. It is evident from Table IV, V that the arbitrage gains are lower than the cost of battery, therefore subsidies would be required for end user participation. For battery model 3, energy arbitrage using CAISO prices tend to over use the battery, which is evident from the cycles of operation shown in Table VI. Note that for other ISOs the gains remain as in Table V, due to the one hour sampling time. Energy storage arbitrage could become more profitable if the price of energy storage decreases drastically *or/and* the price of electricity becomes more volatile *or/and* cycle life of batteries increase significantly.

## VI. CONCLUSION

We formulate the the optimal arbitrage algorithm for storage operation and propose an efficient algorithm to find an optimal solution. The method transforms a continuous, convex optimization problem into a discrete one by exploiting the piecewise linear structure of the cost function. We show that optimal storage control decisions do not depend on prices beyond a certain sub-horizon. In the proposed algorithm, we

indicate a method to calculate these sub-horizons and finding the optimal solution in the sub-horizon using backward step algorithm. The worst case run-time complexity of the proposed algorithm is found to be quadratic in terms of number of time instants for which price values are available.

Using the proposed algorithm, we conducted extended simulations for real price data from several ISOs in USA and Europe for the year 2016. We extrapolate the arbitrage gains for an end user for a five year period, considering detailed losses in the battery. The numerical evaluation suggests that only arbitrage cannot create positive net present value for storage, thus subsidies are required to incentivize investment.

## REFERENCES

- [1] S. Borenstein, "The long-run efficiency of real-time electricity pricing," *The Energy Journal*, vol. 26, no. 3, pp. 93–116, 2005.
- [2] H. Allcott, "Rethinking real-time electricity pricing," *Resource and energy economics*, vol. 33, no. 4, pp. 820–842, 2011.
- [3] P. Mokrian and M. Stephen, "A stochastic programming framework for the valuation of electricity storage," in *26th USAEE/IAEE North American Conference*. Citeseer, 2006, pp. 24–27.
- [4] P. M. van de Ven, N. Hegde, L. Massoulié, and T. Salonidis, "Optimal control of end-user energy storage," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 789–797, 2013.
- [5] J. Qin, R. Sevlian, D. Varodayan, and R. Rajagopal, "Optimal electric energy storage operation," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–6.
- [6] I. Koutsopoulos, V. Hatzi, and L. Tassiulas, "Optimal energy storage control policies for the smart power grid," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*. IEEE, 2011, pp. 475–480.
- [7] J. Cruise, L. Flatley, R. Gibbens, and S. Zachary, "Optimal control of storage incorporating market impact and with energy applications," *arXiv preprint arXiv:1406.3653*, 2014.
- [8] K. Anderson and A. El Gamal, "Co-optimizing the value of storage in energy and regulation service markets," *Energy Systems*, pp. 1–19, 2016.
- [9] X. Xi, R. Sioshansi, and V. Marano, "A stochastic dynamic programming model for co-optimization of distributed energy storage," *Energy Systems*, vol. 5, no. 3, pp. 475–505, 2014.
- [10] J. H. Kim and W. B. Powell, "Optimal energy commitments with storage and intermittent supply," *Operations research*, vol. 59, no. 6, pp. 1347–1360, 2011.
- [11] R. Uргаonkar, B. Uргаonkar, M. J. Neely, and A. Sivasubramaniam, "Optimal power cost management using stored energy in data centers," in *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. ACM, 2011, pp. 221–232.
- [12] R. Walawalkar, J. Apt, and R. Mancini, "Economics of electric energy storage for energy arbitrage and regulation in new york," *Energy Policy*, vol. 35, no. 4, pp. 2558–2568, 2007.
- [13] W. Hu, Z. Chen, and B. Bak-Jensen, "Optimal operation strategy of battery energy storage system to real-time electricity price in denmark," in *Power and Energy Society General Meeting, 2010 IEEE*. IEEE, 2010, pp. 1–7.
- [14] Y. Guo, M. Pan, and Y. Fang, "Optimal power management of residential customers in the smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1593–1606, 2012.
- [15] T. Li and M. Dong, "Real-time energy storage management with renewable integration: Finite-time horizon approach," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2524–2539, 2015.
- [16] "Real time lmp," Online, <https://tinyurl.com/2flowo6>, 2016.
- [17] "Real time electricity price data," Online, [Energinet.dk](http://www.energinet.dk), 2016.
- [18] "Time-of-use rates," Online, <https://tinyurl.com/mwztd3>, 2016.
- [19] "Energy prices," Online, <http://www.energyonline.com/Data/>, 2016.

## APPENDIX A

### PROOF OF THEOREM III.1

We first prove existence of  $(x^*, \alpha^*, \beta^*)$  such that:

- 1)  $x^*$  is the primal optimal solution,
- 2)  $(\alpha^*, \beta^*)$  is the dual optimal solution, and
- 3) the optimality gap is zero (strong duality).

Since the constraints of the primal problem are all linear, weak Slater's constraint qualification conditions (which imply strong duality) follow simply from the feasibility of the primal problem. Clearly, under the assumptions  $b_{\min} \leq b_{\max}$ ,  $\delta_{\min} \leq \delta_{\max}$ ,  $b_0 \in [b_{\min}, b_{\max}]$ ,  $0 \in [\delta_{\min}, \delta_{\max}]$  a feasible solution exists ( $x_i = 0$  for all  $i = 1, 2, \dots, N$  is feasible). Furthermore, since the primal objective function is continuous and the constraints define a convex compact set, its minimum must be finite and achieved at the some  $x^*$  in the feasibility region. According to the strong duality theorem, the above facts imply that the dual problem must be maximized at some  $(\alpha^*, \beta^*)$  and the duality gap must be zero.

From the above reasoning it also follows that  $(x^*, \alpha^*, \beta^*)$  must be the saddle point satisfying the KKT conditions. Hence, using RHS inequality of the Saddle Point conditions,

$$\begin{aligned} \mathcal{L}(x^*, \alpha^*, \beta^*) &\leq \mathcal{L}(x, \alpha^*, \beta^*) \\ \implies \sum_{i=1}^N \left\{ C_{\text{storage}}^{(i)}(x_i^*) + \alpha_i^*(b_{\min} - b_i^*) + \beta_i^*(b_i^* - b_{\max}) \right\} \\ &\leq \sum_{i=1}^N \left\{ C_{\text{storage}}^{(i)}(x_i) + \alpha_i^*(b_{\min} - b_i) + \beta_i^*(b_i - b_{\max}) \right\} \end{aligned}$$

Substituting  $b_i = b_0 + \sum_{j=1}^i x_j$  we get,

$$\sum_{i=1}^N \left\{ C_{\text{storage}}^{(i)}(x_i^*) - \mu_i^* x_i^* \right\} \leq \sum_{i=1}^N \left\{ C_{\text{storage}}(x_i) - \mu_i^* x_i \right\} \quad (6)$$

where  $\mu_i^* = \sum_{j=i}^N (\alpha_j^* - \beta_j^*)$ .  $\mu_i^*$  is the accumulated Lagrange multiplier for time instant  $i$  to  $N$ . Hence,

$$\mu_k^* - \mu_{k+1}^* = (\alpha_k^* - \beta_k^*) \quad (7)$$

The complementary slackness conditions for the Lagrangian are defined as

$$\alpha_i(b_{\min} - b_i) = 0, \quad \beta_i(b_i - b_{\max}) = 0, \quad \forall i \text{ s.t. } \alpha_i, \beta_i \geq 0$$

Equation (8) derived above and complementary slackness conditions implies following relation of  $\mu_k^*$  and  $\mu_{k+1}^*$ ,

$$\mu_{k+1}^* \begin{cases} = \mu_k^*, & \text{if } b_{\min} < b_k^* < b_{\max} \quad \text{as } \alpha_k^* = \beta_k^* = 0 \\ \leq \mu_k^*, & \text{if } b_k^* = b_{\min} \quad \text{as } \alpha_k^* \geq 0 \text{ and } \beta_k^* = 0 \\ \geq \mu_k^*, & \text{if } b_k^* = b_{\max} \quad \text{as } \alpha_k^* = 0 \text{ and } \beta_k^* \geq 0 \end{cases}$$

The accumulated Lagrangian i.e.  $\mu$  for the  $N^{\text{th}}$  (last) instant is  $\mu_N^* = \alpha_N^* - \beta_N^*$ , therefore

$$\mu_N^* = \begin{cases} = 0^*, & \text{if } b_{\min} < b_N^* < b_{\max} \\ \geq 0, & \text{if } b_N^* = b_{\min} \\ \leq 0, & \text{if } b_N^* = b_{\max} \end{cases}$$

Such a  $x^*$  solves the optimal arbitrage problem (P) and  $\alpha^*, \beta^*$  solves the dual problem.  $\square$