AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 6(11): 11815–11836.
DOI:10.3934/math.2021686
Received: 01 February 2021
Accepted: 25 July 2021
Published: 13 August 2021

Research article

A sliding window algorithm for energy distribution system with storage

Jean-Paul Chehab, Vivien Desveaux and Marouan Handa*

LAMFA, UMR CNRS 7352, Université de Picardie Jules Verne, 33 rue Saint Leu, 80039 Amiens, France

* **Correspondence:** Email: marouan.handa@u-picardie.fr.

Abstract: This work is devoted to study optimization problems arising in energy distribution systems with storage. We consider a simplified network topology organized around four nodes: the load aggregator, the external grid, the consumption and the storage. The imported power from the external grid should balance the consumption and the storage variation. The merit function to minimize is the total price the load aggregator has to pay in a given time interval to enforce this balance.
Two optimization problems are considered. The first one is linear and standard. It can be solved through classical optimization methods. The second problem is obtained from the previous one by taking into account a power subscription, which makes it piecewise linear. We establish mathematical properties on both these models.
Finally, a new method based on a sliding window algorithm is derived. It allows to reduce drastically the computational time and makes feasible real time simulations. Numerical results are performed on real data to highlight both models and to illustrate the performance of the sliding window algorithm.

Keywords: optimization modeling; linear programming; nonlinear programming; energy distribution system modeling; sliding window algorithm
Mathematics Subject Classification: 90C05, 90C06, 90C30, 90C51, 90C90, 90-08

1. Introduction

Recently, large scale electric energy storage technologies have been developing fast and many of its applications are investigated (see [3, 4, 6, 14]). There is a growing interest in integrating storage in distribution networks to improve their economy. Indeed, the use of storage capabilities allows the load aggregator to have more flexibility in managing energy transaction and delivery. For example, we can buy extra energy and store it when the price is low and use the stored energy to deliver to customers when the price is high.

In order to develop decision tools aimed to reach economic objectives, the load aggregator must have predictions on both the consumption and the price. There are many methods developed for both price and consumption forecasting, such as those studied in [9, 11, 12], which are based on time series. In this work, we focus on the optimal storage strategy for given prices and consumptions. The next step that is not considered in this paper, would be to integrate uncertainties faced by the load aggregators both for prices and consumption.

The best energy management is presented as a solution of an optimization problem in which the merit function represents the effective energy cost. The decision variables are the operations of charging and discharging of the storage. We impose as constraints the physical limitations of the storage system (capacity and charging/discharging rates). Let us mention that our models have no spatial dependency; they can be obtained by geographical clustering. As a consequence, we do not consider transmission losses and transmission constraints in the formulation. Such models are known as standard discrete-time optimal control problems (see [5, 8]) and are widely used in the literature (for instance [16–18]). The definition of the merit function can lead to different type of models that we will consider in this paper.

Another important issue is the possibility to develop decision tools. To this end, it is necessary to perform real time simulations. The classical numerical optimization methods turn out to be very expensive when large scale problems are considered. For this reason, we introduce a sliding window algorithm inspired by the domain decomposition methods, which is based on same principles as existing methods such as limited foresight and myopic approach (see [7, 13]) or Rolling horizon approach horizon [10, 15]. These methods consist in computing successively optimal solutions on a staggered grid. However, in our case, we allow two successive intervals to overlap with variable length and we discuss the influence of the overlap length on the efficiency of our algorithm.

The paper is organized as follows. In Section 2, we describe the distribution system modeling including storage. We present a first (linear) optimization problem which can be presented as a linear programming. This problem is classical and was used in [16–18]. In order to take into account more realistic situations, we introduce a more advanced (piecewise linear) optimization problem which includes a power subscription in the cost function. Next we prove some qualitative properties satisfied by the solutions of both problems. Finally we recall some classical numerical methods to approximate the solutions.

Section 3 is dedicated to present in details the sliding window method. In Section 4 we perform numerous numerical simulations in order to illustrate both the models and the algorithms. Finally, in Section 5 we give a few concluding remarks and perspectives.

2. Distribution system modeling

2.1. System description

The model that we use in this paper is inspired by [18]. The model is obtained by clustering geographically the nodes of the same kind. This lets us with only four nodes:

- the load aggregator which is in charge of importing and distributing the energy;
- the external grid where traditional or renewable energy can be purchased;
- consumption which is the mutualised consumption of all the customers;
- the storage that can be used by the load aggregator to store or to deliver additional energy.

We consider a time interval $[0, T]$ which is discretized with regularly spaced times $(t_i)_{0\leq i\leq N}$ such that $t_{i+1} = t_i + \delta t$, where δt is the time step. For simplicity, δt is set to 1 (typically one hour). The following quantities will denote average values on the time interval $[t_{i-1}, t_i]$ (expressed in terms of Watt), (see Figure 1): L_i will be the consumption, C_i and D_i denote the charged and discharged power, respectively, and U_i is the imported power. There are losses during the charging and discharging operations. The efficiencies of these processes are denoted by η_C and η_D, respectively, and are assumed to belong to $(0, 1)$. The state of charge at time t_i will be denoted by S_i (expressed in terms of Watt Hour). The quantity L_i is assumed to be known and C_i, D_i and U_i are the decision variables that will be computed by the load aggregator.

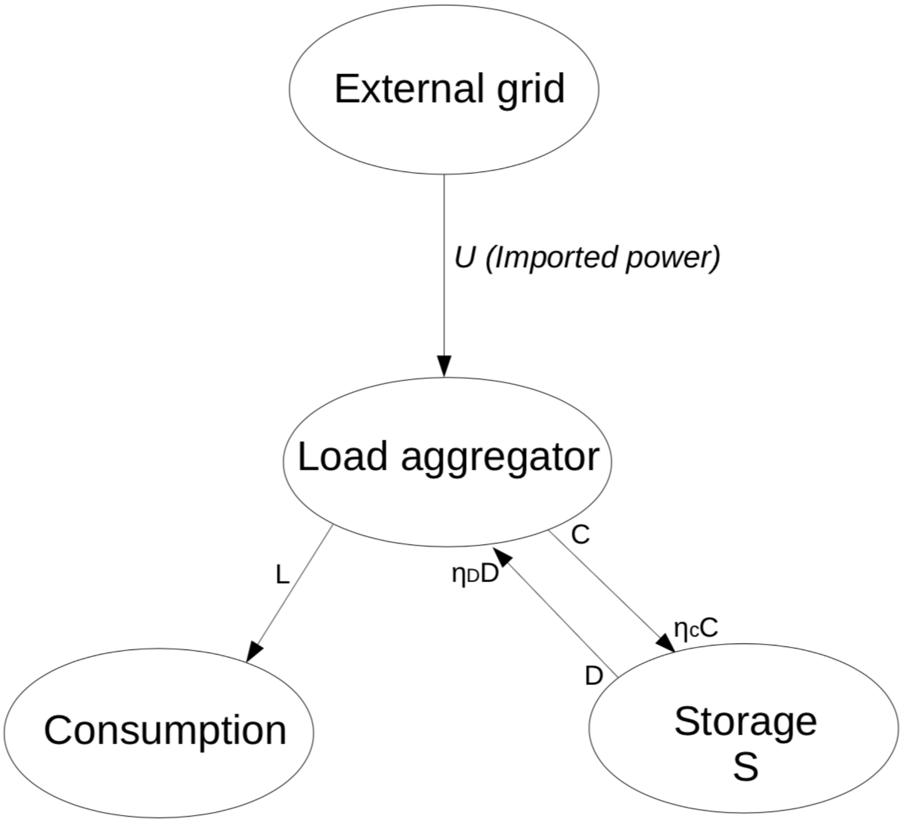

Figure 1. Distribution system.

Given an initial value of the storage S_0, the state of charge at time t_i, $1 \leq i \leq N$, is defined recursively by

$$S_i = S_{i-1} + \eta_C C_i - D_i. \tag{2.1}$$

Let us notice that S_i can be written equivalently as

$$S_i = S_0 + \sum_{k=1}^{i} (\eta_C C_k - D_k). \tag{2.2}$$

The variables C_i, D_i and S_i are bounded by their operation limits, expressed as

$$S_{min} \leq S_i \leq S_{max}, \tag{2.3}$$

$$0 \leq C_i \leq C_{max}, \tag{2.4}$$

$$0 \leq D_i \leq D_{max}. \tag{2.5}$$

Here S_{min} and S_{max} are respectively the minimum and the maximum energy storage capacities, C_{max} and D_{max} are respectively the charging and discharging power limits.

For simplicity, the external stock of energy is assumed to be infinite. The load aggregator must enforce the power balance, given by

$$U_i = L_i + C_i - \eta_D D_i. \tag{2.6}$$

We derived above the modeling of the physical constraint, that will be unchanged in the various economic situations we will consider from now on; each economical model (power delivery with or without subscription) will be governed by the choice of the merit function to minimize.

2.2. *The model without subscription*

For the sake of simplicity, we first consider a linear merit function which corresponds to the simplest economical model. In short, the load aggregator imports power from the external grid at a given price.

2.2.1. Formulation

The average price for the imported electricity from the external grid on $[t_{i-1}, t_i]$ is denoted by P_i and is assumed to be known and fixed. Therefore, the total cost of imported power on $[0, T]$ is given by

$$\sum_{i=1}^{N} P_i U_i.$$

According to (2.6), and since L_i is fixed, minimizing this cost is equivalent to minimize the following merit function

$$J(C, D) = \sum_{i=1}^{N} P_i (C_i - \eta_D D_i). \tag{2.7}$$

For a given $S_0 \in [S_{min}, S_{max}]$, the optimization problem can be written as

$$\begin{aligned} &\min_{C,D \in \mathbb{R}^N} J(C, D), \\ &\text{s.t. } \forall 1 \leq i \leq N, \quad \begin{cases} S_i = S_{i-1} + \eta_C C_i - D_i, \\ S_{min} \leq S_i \leq S_{max}, \\ 0 \leq C_i \leq C_{max}, \\ 0 \leq D_i \leq D_{max} \end{cases} \end{aligned} \tag{P1}$$

Remark 2.1. *Let us notice that the investment cost of the storage system does not appear in the merit function J. Indeed, it would appear as a constant with respect to decision variables C and D of Problem* (P1).

2.2.2. Properties

The Problem (P1) does not necessarily have a unique solution since J is not strictly convex. Here we are going to prove that all the minimizers of J satisfy some qualitative properties.

Physically, the charging and discharging operations cannot be realized simultaneously. Theoretically, we should add the constraint $C_iD_i = 0$ to Problem (P1). The first result we present shows that this constraint is automatically taken into account in the optimization problem, thus it is not necessarily to include it as a constraint in the resolution.

Theorem 2.2. *Let (C^*, D^*) be a solution of Problem (P1). Then $C_i^* D_i^* = 0$ for all $1 \leq i \leq N$.*

Proof. Assume that there exists j such that $C_j^* D_j^* > 0$. Let us define

$$\overline{C}_i = \begin{cases} C_i^* & \text{if } i \neq j, \\ (C_j^* - \frac{1}{\eta_C} D_j^*)^+ & \text{if } i = j, \end{cases}$$

$$\overline{D}_i = \begin{cases} D_i^* & \text{if } \neq j, \\ (\eta_C C_j^* - D_j^*)^- & \text{if } i = j, \end{cases}$$

where x^+ and x^- will denote respectively the positive part and the negative part of a given number x, i.e. $x^+ = \max(x, 0)$ and $x^- = -\min(x, 0)$. This definition ensures for all i

$$0 \leq \overline{C}_i \leq C_i^*,$$

$$0 \leq \overline{D}_i \leq D_i^*,$$

so $\overline{C}$ and $\overline{D}$ satisfy constraints (2.4) and (2.5). Let S^* and $\overline{S}$ be the state of charge related to (C^*, D^*) and $(\overline{C}, \overline{D})$, respectively. Then according to (2.2), we have

$$\begin{aligned} \overline{S}_i &= S_0 + \sum_{\substack{k=1 \\ k \neq j}}^{i} (\eta_C \overline{C}_k - \overline{D}_k) + \eta_C \overline{C}_j - \overline{D}_j, \\ &= S_0 + \sum_{\substack{k=1 \\ k \neq j}}^{i} (\eta_C C_k^* - D_k^*) + (\eta_C C_j^* - D_j^*)^+ - (\eta_C C_j^* - D_j^*)^-. \end{aligned}$$

Since $x^+ - x^- = x$, we deduce

$$\overline{S}_i = S_0 + \sum_{\substack{k=1 \\ k \neq j}}^{i} (\eta_C C_k^* - D_k^*) + (\eta_C C_j^* - D_j^*) = S_i^*.$$

Therefore, we have $\overline{S} = S^*$, so $\overline{S}$ satisfies constraint (2.3).

However, since $\eta_C \eta_D < 1$, we have

$$\begin{aligned} \overline{C}_j - \eta_D \overline{D}_j &= (C_j^* - \frac{1}{\eta_C} D_j^*)^+ - \eta_D(\eta_C C_j^* - D_j^*)^-, \\ &< (C_j^* - \eta_D D_j^*)^+ - (C_j^* - \eta_D D_j^*)^- = C_j^* - \eta_D D_j^*. \end{aligned}$$

Then, the merit function is given by

$$J(\overline{C},\overline{D}) = \sum_{i\neq j} P_i(\overline{C}_i - \eta_D \overline{D}_i) + P_j(\overline{C}_j - \eta_D \overline{D}_j),$$
$$< \sum_{i\neq j} P_i(C_i^* - \eta_D D_i^*) + P_j(C_j^* - \eta_D D_j^*) = J(C^*, D^*).$$

This contradicts the optimality of (C^*, D^*). □

Next theorem describes the energy storage level at final time N before and after the time $(S_0 - S_{min})/D_{max}$, which is the minimal time to empty the storage S at maximum discharge rate D_{max}.

Theorem 2.3. *Let (C^*, D^*) be a solution of Problem (P1) and S^* be the associated state of charge. We have the following alternatives:*

(i) if $N \leq (S_0 - S_{min})/D_{max}$, then $S^*_N = S_0 - ND_{max}$;

(ii) if $N > (S_0 - S_{min})/D_{max}$, then the energy storage level reverts to its minimum value at time N, i.e. $S^*_N = S_{min}$.

Proof. We start by proving (i). Let us define $C_i^* = 0$ and $D_i^* = D_{max}$. Obviously C^* and D^* satisfy constraints (2.4) and (2.5). Moreover the hypothesis on N ensures that the associated state of charge S^* satisfies constraint (2.3). We are going to show that this define an optimal solution. Let C and D be two vectors satisfying constraints (2.4) and (2.5) such that the associated state of charge S satisfies constraint (2.3). We have

$$J(C^*, D^*) - J(C, D) = -\sum_{i=1}^{N} P_i(C_i + \eta_D(D_{max} - D_i) \leq 0.$$

Therefore, (C^*, D^*) is an optimal solution. Let us notice that the solution is unique in this case. It follows immediately that $S_i^* = S_0 - ND_{max}$.

Now let us turn to the proof of (ii). In order to obtain a contradiction, suppose that $S^*_N > S_{min}$. If $D_i^* = D_{max}$, for all $1 \leq i \leq N$, then according to the hypothesis on N, we have

$$S^*_N = S_0 - ND_{max} < S_{min}.$$

This contradicts the hypothesis $S^*_N > S_{min}$, so there exists i such that $D_i^* < D_{max}$. Let us define $j = \max\{i : D_i^* < D_{max}\}$. This ensures that $S^*_k > S^*_N$, for all k such that $j \leq k < N$.

Let $\overline{C}$ and $\overline{D}$ be two vectors defined by

$$(\overline{C}_i, \overline{D}_i) = \begin{cases} (C_i^*, D_i^*) & \text{if } i \neq j, \\ (C_j^* - \epsilon_1, 0) & \text{if } i = j \text{ and } C_j^* > 0, \\ (0, D_j^* + \epsilon_2) & \text{if } i = j \text{ and } C_j^* = 0, \end{cases}$$

with

$$\epsilon_1 = \min\left(\frac{S^*_N - S_{min}}{\eta_C}, C_j^*\right)$$

and

$$\epsilon_2 = \min\left(S_N^* - S_{min}, D_{max} - D_j^*\right).$$

This definition ensures that $\overline{C}$ and $\overline{D}$ satisfy constraints (2.4) and (2.5). Let $\overline{S}$ be the state of charge related to $\overline{C}$ and $\overline{D}$. We are going to check that $\overline{S}$ satisfy constraint (2.3). First, for $i \leq j$, we have $\overline{S}_i = S_i^*$. Next, for $i > j$, we have in one hand $\overline{S}_i < S_i^* \leq S_{max}$. It remains to prove that $S_{min} \leq \overline{S}_i$. In order to do so, let us notice that

$$\eta_C \overline{C}_j - \overline{D}_j \geq (\eta_C C_j^* - D_j^*) + S_{min} - S_N^*. \tag{2.8}$$

Indeed, two cases can be distinguished:

- if $C_j^* = 0$, then $\eta_C \overline{C}_j - \overline{D}_j = -D_j^* - \epsilon_2 \geq -D_j^* + S_{min} - S_N^*$;
- if $C_j^* > 0$, then $\eta_C \overline{C}_j - \overline{D}_j = \eta_C (C_j^* - \epsilon_1) \geq \eta_C C_j^* + S_{min} - S_N^*$.

Therefore we have according to (2.8)

$$\begin{aligned}\overline{S}_i &= S_0 + \sum_{\substack{k=1\\k\neq j}}^{i} (\eta_C C_k^* - D_k^*) + (\eta_C \overline{C}_j - \overline{D}_j),\\ &\geq S_0 + \sum_{\substack{k=1\\k\neq j}}^{i} (\eta_C C_k^* - D_k^*) + (\eta_C C_j^* - D_j^*) + S_{min} - S_N^*.\end{aligned}$$

Then thanks to (2.2), we conclude

$$\overline{S}_i \geq S_i^* + S_{min} - S_N^* \geq S_{min}.$$

Finally, we prove that $(\overline{C}, \overline{D})$ realizes a lower value for J than (C^*, D^*). This is immediate since

$$\overline{C}_j - \eta_D \overline{D}_j < (C_j^* - \eta_D D_j^*).$$

This contradicts the optimality of (C^*, D^*).

□

2.2.3. Resolution methods

We rewrite Problem (P1) in the following standard matrix form which is suitable for the application of the classical resolution methods:

$$\begin{aligned}&\min_{C,D\in\mathbb{R}^N} && \begin{pmatrix} P^T & -\eta_D P^T \end{pmatrix} \begin{pmatrix} C \\ D \end{pmatrix},\\ &\text{s.t} && a \leq A\,(\eta_C C - D) \leq b,\\ &&& 0 \leq C \leq c,\\ &&& 0 \leq D \leq d,\end{aligned}$$

where C and D are the decision variables corresponding to charge and discharge, respectively, $P \in \mathbb{R}^N$ is the price vector and $a, b, c, d \in \mathbb{R}^N$ are the given vectors defined by

$$a = (S_{min} - S_0)\mathbf{1},$$

$$b = (S_{max} - S_0)\mathbf{1},$$

$$c = D_{max}\mathbf{1},$$

$$d = C_{max}\mathbf{1},$$

with $\mathbf{1} = (1, \cdots, 1)^T$. Finally, $A \in M_N(\mathbb{R})$ is the matrix defined by

$$A = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ 1 & \cdots & \cdots & 1 \end{pmatrix}.$$

With this formalism, we can see that (P1) is expressed in the framework of linear programming. It can be solved either by the simplex method or the interior point method for very large scale problems. We refer the reader to [1] for a complete description of both methods.

Remark 2.4. *Let us notice that it is possible to reformulate (P1) using only one decision vector Δ defined by $\Delta = \eta_C C - D$. Indeed, we can write the merit function as*

$$J(\Delta) = \sum_{i=1}^{N} P_i \Big(\frac{1}{\eta_C} \Delta_i^+ + \eta_D \Delta_i^- \Big).$$

Hence, we consider the problem

$$\begin{aligned} &\min_{\Delta \in \mathbb{R}^N} J(\Delta), \\ &s.t.\ \forall 1 \le i \le N, \quad \begin{cases} S_{min} - S_0 \le \sum_{k=1}^{i} \Delta_k \le S_{max} - S_0, \\ -D_{max} \le \Delta_i \le \eta_C C_{max}. \end{cases} \end{aligned} \tag{P1'}$$

Both Problems (P1) and (P1') are equivalent in the following sense:

1. *If (C^*, D^*) is a solution of Problem (P1), then $\Delta^* = \eta_C C^* - D^*$ is a solution of Problem (P1').*
2. *If Δ^* is a solution of Problem(P1'), then $(C^*, D^*) = (\frac{1}{\eta_C}\Delta^{*+}, \Delta^{*-})$ is a solution of Problem (P1).*

Dealing with only one variable allows to use less memory, however the problem is now nonlinear and its resolution requires a larger computational effort.

2.3. The model with subscription

In this section, we take into account a more realistic economical model. Instead of buying power freely from the external grid, as in the previous model, the load aggregator is now limited by a power subscription.

2.3.1. Framework

We start from the same nodes as in the previous model (see Figure 1). Let us assume that the load aggregator has a subscription for a given power U_s. The price of this subscription is a constant which has no influence on the optimization problem that will follow. Therefore we will not take it into account. This subscription is based on the two alternatives:

- if the imported power U_i does not exceed U_s, its price is P_i;
- else, the imported power U_i is still imported with price P_i, but the exceeding power $U_i - U_s$ is penalized with a higher price Q_i.

Therefore the cost of the imported power is

$$\sum_{i=1}^{N} P_i U_i + \sum_{i=1}^{N} Q_i (U_i - U_s)^+,$$

where

$$(U_i - U_s)^+ = \max(U_i - U_s, 0).$$

As previously, using equation (2.6), this cost can be reformulated using only the decision variables C and D. Thus we obtain the following merit function to minimize

$$J_s(C, D) = \sum_{i=1}^{N} P_i(C_i - \eta_D D_i) + \sum_{i=1}^{N} Q_i(L_i - U_s + C_i - \eta_D D_i)^+. \tag{2.9}$$

The optimization problem is then expressed as

$$\begin{aligned} &\min_{C,D\in\mathbb{R}^N} J_s(C, D),\\ &\text{s.t. } \forall 1 \le i \le N, \quad \begin{cases} S_i = S_{i-1} + \eta_C C_i - D_i,\\ S_{min} \le S_i \le S_{max},\\ 0 \le C_i \le C_{max},\\ 0 \le D_i \le D_{max}. \end{cases} \end{aligned} \tag{P2}$$

Remark 2.5. *The model without subscription can be seen as a particular case of the current model with U_s large enough.*

2.3.2. Properties

The results presented here state that the solutions of (P2) satisfy the same properties as the solutions of (P1).

Theorem 2.6. *Let (C^*, D^*) be a solution of Problem (P2). Then $C_i^* D_i^* = 0$ for all $1 \le i \le N$.*

Theorem 2.7. *Let (C^*, D^*) be a solution of Problem (P2) and S^* be the associated state of charge. We have the following alternatives:*

(i) if $N \le (S_0 - S_{min})/D_{max}$, then $S_N^* = S_0 - N D_{max}$;

(ii) if $N > (S_0 - S_{min})/D_{max}$, then the energy storage level reverts to its minimum value at time N, i.e. $S_N^* = S_{min}$.

The proofs of the latest theorems are similar as in the model without subscription. The additional term in the merit function J_s does not raise any technical difficulty to adapt the proof.

2.3.3. Resolution method

The Problem (P2) is piecewise linear and classically can be written as a linear program as follows (see [2]). We introduce an auxiliary variable $q \in \mathbb{R}^N$ and the merit function

$$\widehat{J_s}(C, D, q) = \sum_{i=1}^{N} P_i(C_i - \eta_D D_i) + \sum_{i=1}^{N} Q_i q_i.$$

Next we consider the following optimization problem

$$\min_{C,D,q \in \mathbb{R}^N} \widehat{J_s}(C, D, q),$$

$$\text{s.t. } \forall 1 \le i \le N, \quad \begin{cases} S_i = S_{i-1} + \eta_C C_i - D_i, \\ S_{min} \le S_i \le S_{max}, \\ 0 \le C_i \le C_{max}, \\ 0 \le D_i \le D_{max}, \\ q_i \ge 0, \\ q_i \ge L_i + C_i - \eta_D D_i - U_s. \end{cases} \tag{$\widehat{\text{P2}}$}$$

We now establish the equivalence between Problems (P2) and ($\widehat{\text{P2}}$).

Proposition 2.8.

1. *If $(C^\star, D^\star)$ is a solution of Problem (P2), then by setting*

$$q_i^\star = (L_i - U_s + C_i^\star - \eta_D D_i^\star)^+, \tag{2.10}$$

the triplet $(C^\star, D^\star, q^\star)$ is a solution of Problem ($\widehat{\text{P2}}$).

2. *If $(C^\star, D^\star, q^\star)$ is a solution of Problem ($\widehat{\text{P2}}$), then $(C^\star, D^\star)$ is a solution of the Problem (P2).*

Proof. First, we notice that if (C, D, q) satisfies the constraints of Problem ($\widehat{\text{P2}}$), then we have

$$\widehat{J_s}(C, D, q) \ge J_s(C, D).$$

1. By definition (2.10) of $q^\star$, the triplet $(C^\star, D^\star, q^\star)$ satisfies the constraints of Problem ($\widehat{\text{P2}}$). Let (C, D, q) a triplet satisfying the constraints of Problem ($\widehat{\text{P2}}$). We have

$$\widehat{J_s}(C, D, q) \ge J_s(C, D) \ge J_s(C^\star, D^\star) = \widehat{J_s}(C^\star, D^\star, q^\star).$$

The last equality follows from the definition (2.10) of $q^\star$. Therefore, $(C^\star, D^\star, q^\star)$ is a solution of Problem ($\widehat{\text{P2}}$).

2. The couple $(C^\star, D^\star)$ trivially satisfies the constraints of Problem (P2). Let (C, D) a couple satisfying the constraints of Problem (P2) and let

$$q_i = (L_i - U_s + C_i - \eta_D D_i)^+.$$

This definition ensures that

$$J_s(C, D) = \sum_{i=1}^{N} P_i(C_i - \eta_D D_i) + \sum_{i=1}^{N} Q_i q_i,$$

therefore

$$J_s(C, D) = \widehat{J_s}(C, D, q) \geq \widehat{J_s}(C^\star, D^\star, q^\star) \geq J_s(C^\star, D^\star).$$

Thus $(C^\star, D^\star)$ is a solution of Problem (P2).

□

The equivalent reformulation $(\widehat{\text{P2}})$ leads to a linear programming problem, we can thus use the same numerical methods as for Problem (P1).

3. Sliding window algorithm

In many concrete applications, the computation of the solutions of (P1) or (P2) can be included in a decisional system which aims to apply an optimal storage strategy. Therefore it is crucial to compute the solution in real time. When one considers a long period of time, the computing time can quickly become very large. In order to overcome this difficulty, we propose to apply iteratively the optimization method on small overlapping time intervals recovering the whole time period. This approach leads to a resolution method that can be seen as a sliding window algorithm.

Let us notice that this method has similarities with existing algorithms like limited foresight and myopic approach [7, 13]. However we can point out some differences: The length of the overlap between consecutive intervals is variable in our method and the influence of this length is discussed; also in these existing methods there is a boundary condition imposed at the final time of each subinterval, while this is not the case in our method.

The main idea is to decompose the global problem on the discrete time interval $[|0, N|]$ into a succession of simulations on small time subintervals. The solutions are then concatenated in order to obtain an approximated solution of the global problem.

In the following, we assume that the subintervals are large enough to enter the second alternative of Theorems 2.3 and 2.7. Therefore this theorems ensure that the storage level will be equal to S_{min} at the end of each simulation. This creates artificial boundary conditions that will lead to a non optimal global solution. In order to avoid these artifacts, we impose that two successive sub-intervals must overlap.

Now we present in details the suggested method. Let $L \in \mathbb{N}^*$, $L \leq N$, the length of the time window. The other parameter of the method is the length of the overlap, that we will denote by $r < L$. We define the subintervals $[|a_i, b_i|] \subset [|0, N|]$, with $b_i - a_i = L$, using the following procedure:

- $a_0 = 0$;

- $a_i = b_{i-1} - r, \quad i > 0.$

In order to compute the solution on the subinterval $[|a_k, b_k|]$, we need an initial condition S_0^k for the state of charge at time a_k. Since a_k belongs to the sub-interval $[|a_{k-1}, b_{k-1}|]$, we can set this initial condition as the value of the state of charge computed at time a_k obtained during the simulation on the subinterval $[|a_{k-1}, b_{k-1}|]$. This method is summarized in Algorithm 1.

Algorithm 1

Require: L, r and S_0
  $S = (S_0)$
  for $k = 0, 1, 2...$ **do**
    solve Problem (P1) or (P2) on $[|\ a_k, b_k\ |]$ with the last component of S as initial state of charge
    compute $S^k = (S_1^k, ..., S_L^k)$ with equation (2.3)
    $S \leftarrow$ concatenation of S and $(S_1^k, \cdots, S_{L-r}^k)$
  end for

Figure 2 illustrates how the proposed algorithm works in the first iterations. In black is displayed what we call the reference solution, which was computed directly on the whole time interval $[|0, N|]$ (without the Sliding Window Algorithm). The first iteration generates a solution on the time interval $[|a_0, b_0|] = [|0, L|]$. The state of charge at time $L - r$ is used as the initial condition for the second iteration, which is then computed on the interval $[|a_2, b_2|] = [|L - r, 2L - r|]$.

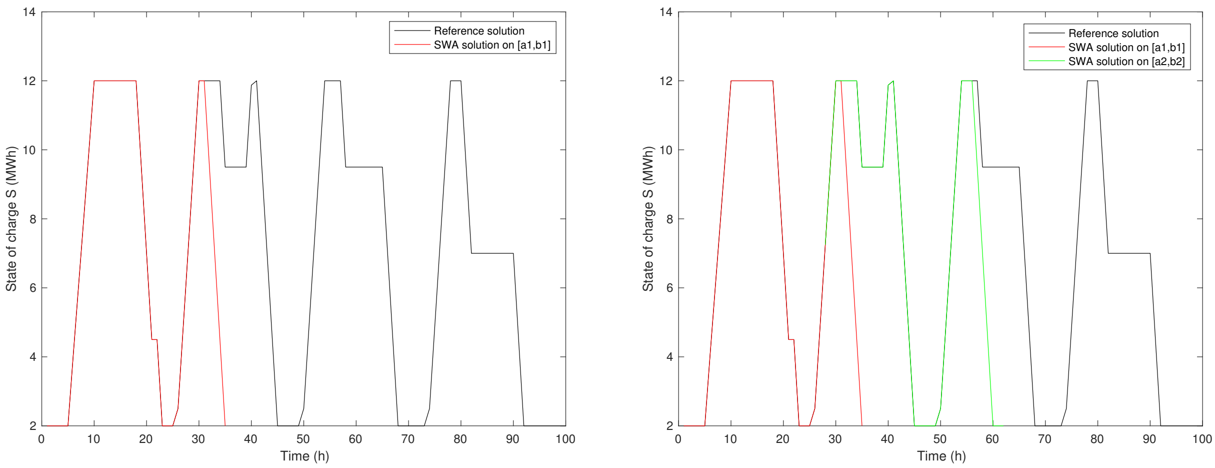

Figure 2. The first iteration (left) and the second iteration (right) of the sliding window algorithm.

The choice of the parameters L and r and their effect on the quality of the approximation, as well as their effect on the CPU time, will be discussed in section 4.

4. Numerical results and discussion

Simulations are performed with MATLAB® on a personal computer with an Intel Core i7 processor running at 2,8 GHz. The storage parameters are set as in Table 1 to reproduce all the figures except for Figure 4 where the effect of maximum storage S_{max} is studied and then is variable. The values for consumption and prices are coming from real life data for a small town of about $10,000$ inhabitants. All the simulations are performed with a simplex algorithm using the function `linprog`. The dual feasibility tolerance is set by default to $1e-07$ and the primal feasibility tolerance is set by default to $1e-04$.

Table 1. Storage parameters values.

S_{min}	2 MWh
S_{max}	12 MWh
C_{max}, D_{max}	2.5 MW
η_C, η_D	0.95

4.1. *Model without subscription*

To validate the model (P1), we consider a time frame of 24 hours. Figure 3 represents the evolution of the state of charge together with the price. We notice that the results agree with the basic intuition: the state of charge S tends to increase when the price is low and to decrease when the price is high. To point out the advantage of using storage we compare the cost with and without storage. In this example the cost without storage is $1.0353e+04$ (euros), against $1.0175e+04$ (euros) when storage is considered.

As stated in Theorem 2.3, we can see from Figure 3 that the state of charge level S reverts to its minimum ($S_{24} = S_{min} = 2$).

Next, we study the influence of the maximum storage capacity S_{max} on the cost. We can see in Figure 4 that the merit function decreases when storage capacity increases. However, as it can be excepted, there is a threshold for the storage capacity (about 20 MWh for these data) beyond which the total cost stays constant.

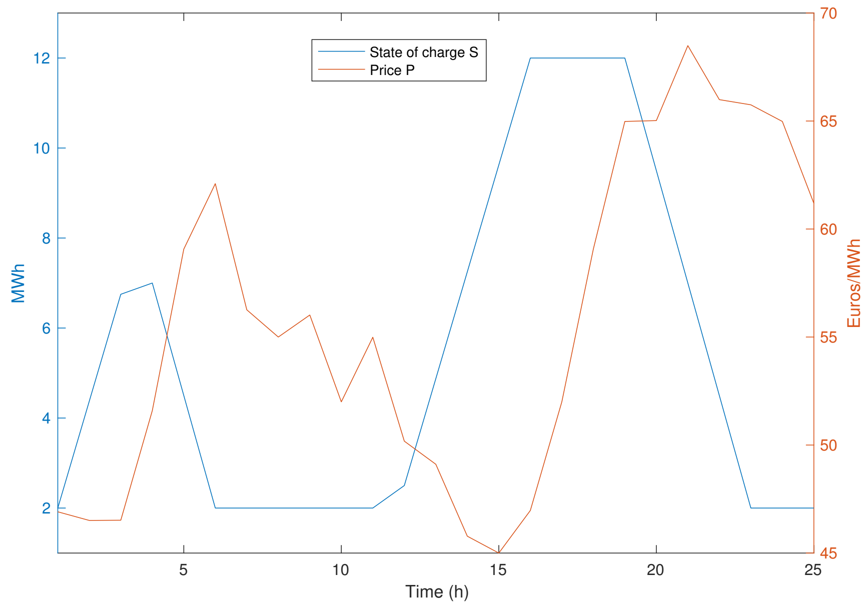

Figure 3. State of charge variation versus the price (Model (P1)).

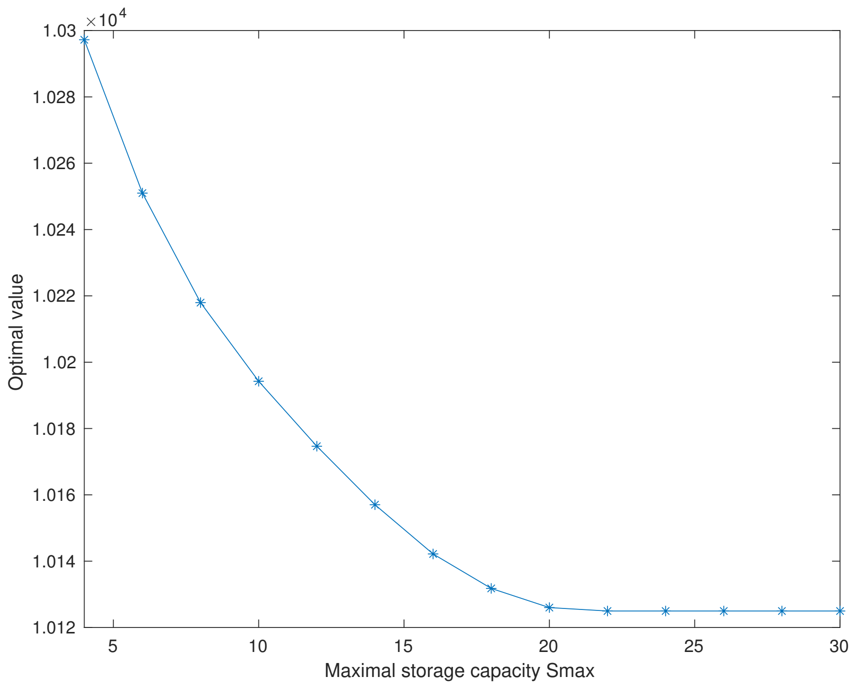

Figure 4. Variation of the optimal value with respect to the maximum storage capacity (Model (P1)).

4.2. Model with subscription

Now we present the results obtained with model (P2). The subscription U_{ab} is set to 7 MW, and the price curve Q is taken equal to P.

The evolution of the state of charge and of the price are shown in Figure 5. We compare the cost with and without storage: In this example the cost without storage is $1.1016e + 04$ (euros), against $1.0749e + 04$ (euros) when the storage is considered. The evolution of the state of charge has a similar shape as the one obtained with model (P1) and displayed in Figure 3.

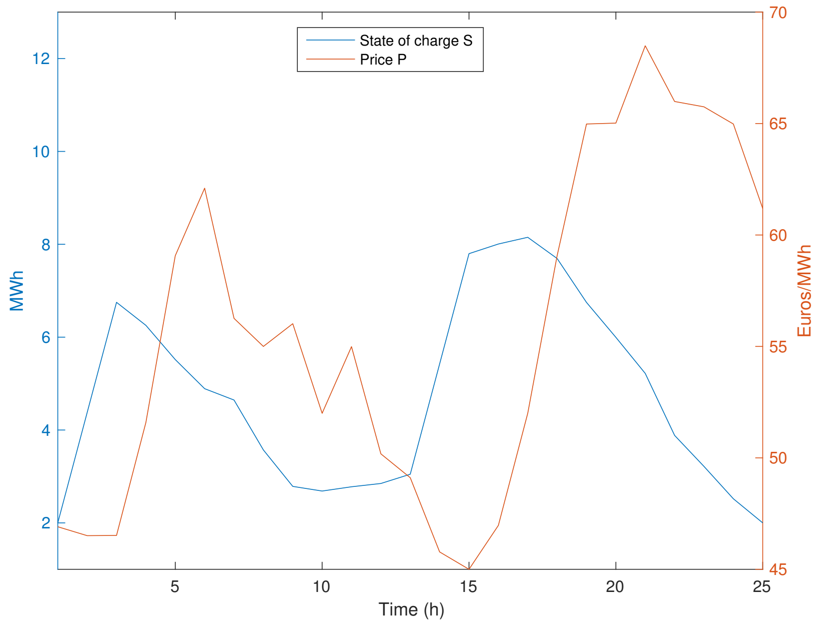

Figure 5. State of charge variation versus the price (Model (P2)).

Similarly as in the model without subscription case, we study the influence of the maximum storage capacity on the optimal energy cost. The results are displayed on Figure 6. The threshold value beyond which it is no longer interesting to increase the storage capacity (here about 10 MWh) is not the same as in model (P1) because it depends on the subscribed power. Indeed, we also show in Figure 6 the influence of the subscribed power on the optimal energy cost. The behaviour is similar: the energy cost does not decrease when the subscribed power goes beyond a threshold value (here about 8 MW). Since it can be excepted that the cost of the subscription increases with the power subscribed, the load aggregator should not subscribe to a power greater than this threshold value.

In order to emphasize the differences between the two models, we compare the imported powers for each model, displayed on Figures 7 and 8. The consumption curve and the price curve are also plotted in order to see their impact for both models. Let us recall that the consumption does not intervene in model (P1). We notice that the shape of the imported power curve are slightly different. In model (P2), the imported power tends to be equal to U_s most of the time in order to avoid the penalties. There are only two short time intervals in which this is not the case and this corresponds to time intervals when the price is very low. In contrast, the curve obtained with model (P1) is mostly driven by the price all the time.

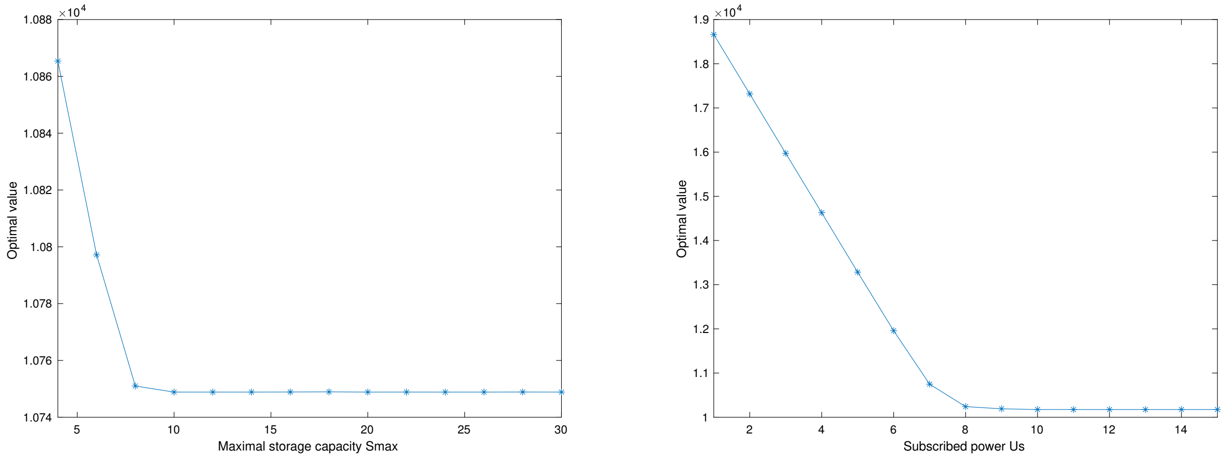

Figure 6. Variation of the optimal value (Model (P2)) with respect to the maximum storage capacity (left) and the subscribed power U_s (right).

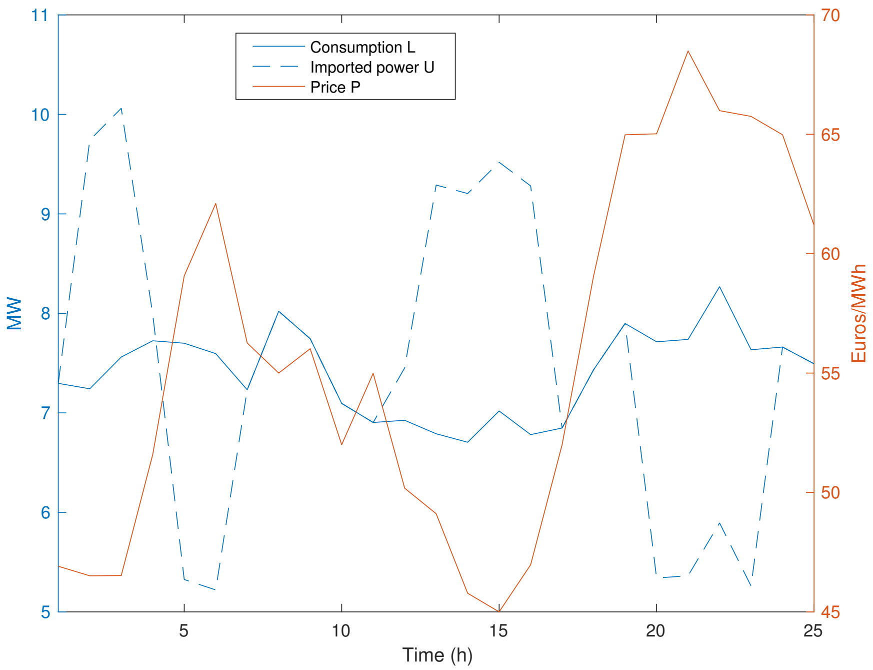

Figure 7. Imported power variation versus the consumption and the price (Model (P1)).

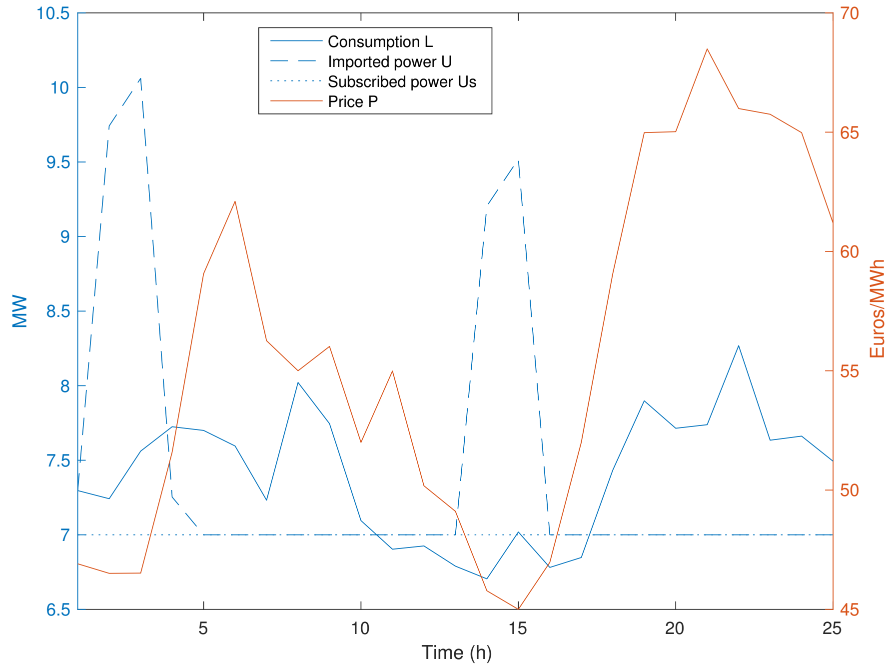

Figure 8. Imported power variation versus the consumption and the price (Model (P2)).

4.3. The sliding window algorithm

In this section, we are going to compare the solution obtained by the sliding window algorithm with the reference solution computed in one step on the whole time frame $[|0, N|]$. Before giving the criteria used for this comparison, we need to introduce a few notations:

- (C, D) represents the reference solution, S the associated state of charge and m the optimal value (equal to $J(C, D)$ or $J_s(C, D)$, depending on the model);
- (C^{swa}, D^{swa}) represents a solution computed with the sliding window algorithm with a given L and r that will be specified for each simulation, S^{swa} the associated state of charge and m^{swa} the optimal value (equal to $J(C^{swa}, D^{swa})$ or $J_s(C^{swa}, D^{swa})$, depending on the model).

To analyze the efficiency of the method and the sensitivity of the parameters L and r, we use the three following criteria:

- the CPU time (in seconds);
- the relative L^1 error on the state of charge $E_1 = \dfrac{\sum_{i=1}^{N} |S_i - S_i^{swa}|}{\sum_{i=1}^{N} S_i}$;
- the relative error on the optimal value $E_2 = \dfrac{|m - m^{swa}|}{m}$.

The first numerical simulations we present here are dedicated to illustrate how the sliding window algorithm works. Therefore we consider a short period of time with $N = 100$, $L = 30$ and $r = 5$.

The evolution of the state of charge for both the reference solution and the sliding window solution are displayed in Figure 9 (model (P1)) and in Figure 10 (model (P2)). We remark that in Figure 9 both curves coincide, which is not the case in Figure 10. Since the solutions of problems (P1) and (P2) are not unique, it is also useful to compare their optimal value. For model (P1), we obtain $E_1 = 4.33e-05$ and $E_2 = 6.79e-11$, while for model (P2), we find $E_1 = 1.78e-01$ and $E_2 = 6.38e-04$. As we can expect from Figure 9, these errors are very small for model (P1). For model (P2), although the L^1 error is important, the optimal values are close. It means that the two methods have probably converged toward different minimizers.

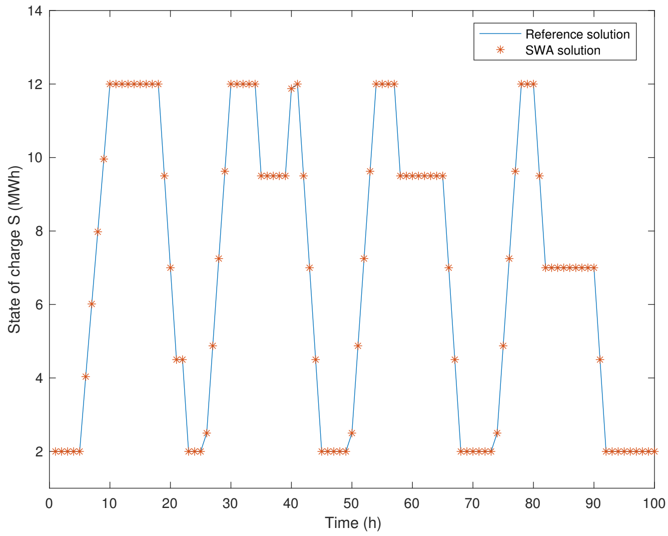

Figure 9. State of charge for the reference solution and the sliding window solution with $L = 30$ and $r = 5$ (model (P1)).

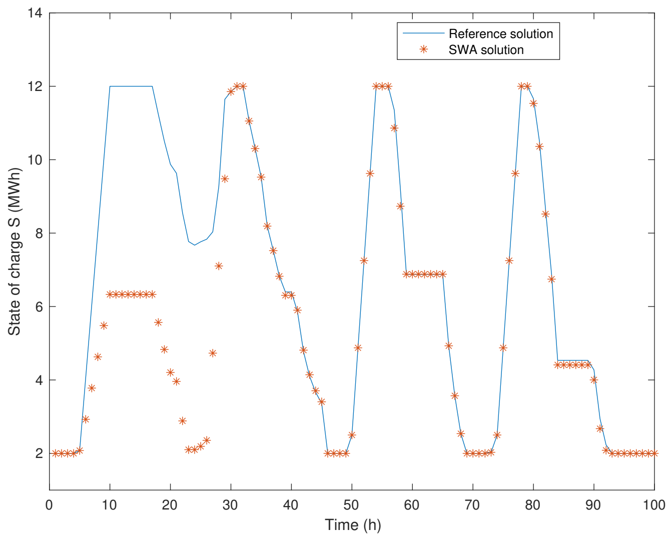

Figure 10. State of charge for the reference solution and the sliding window solution with $L = 30$ and $r = 5$ (model (P2)).

Since the main objective of the sliding window algorithm is to decrease the CPU time, we are going to compute simulations on a much longer time frame. We set $N = 2160$ hours, which correspond to a three months simulation. Let us recall that the sliding window algorithm is characterized by two parameters: The length of the time window L and the length of the overlap r.

The next series of results has a double goal. First, we want to emphasize that the gain in CPU time obtained with the sliding window algorithm does not deteriorate too much the accuracy of the solution. Secondly, we want to study the impact of the choice of the parameters L and r on the accuracy and the CPU time. Therefore, we are going to perform a number of simulations with the sliding window algorithm and with different values of L and r. Each of these simulations will be compared to the reference solution.

The results presented in Tables 2 and 3 display the values of errors E_1, E_2 and the CPU time for models (P1) and (P2), respectively, while varying the time window L and keeping the length of the overlap r constant to 5. Similar results are presented in Tables 4 and 5, but this time we set the time window L to 40 while varying the length of the overlap r.

In model (P1), the reference solution is computed with the same algorithm parameters as in Subsection 4.1. The CPU time to compute the reference solution is approximately 390 seconds. We can observe from Table 2 that increasing the time window L, both errors E_1 and E_2 decrease while the CPU time increase. We can observe from Table 4 that the length of the overlap r has less influence on the errors than to the length of the time window L has. Indeed, the CPU time increases with r; however for $r \geq 15$, we do not observe a significant decrease in both errors.

The total CPU time to compute the reference solution with this method is approximately 1077 seconds. We can observe from Table 3 that the length of the time window L does not have much influence on both errors E_1 and E_2. As for model (P1), we notice that the parameter r should not be too small. However, for $r \geq 15$ there is no significant improvement.

Table 2. Errors and CPU time for the sliding window algorithm with varying L and fixed $r = 5$ (model (P1)).

L	E_1	E_2	CPU time (s)
20	$2.30e-01$	$4.26e-04$	2.07
40	$1.82e-01$	$1.71e-04$	1.50
60	$1.38e-01$	$9.06e-05$	1.78
100	$1.03e-01$	$1.95e-05$	2.10
140	$8.57e-02$	$4.00e-05$	3.35
180	$3.72e-02$	$9.62e-05$	5.07
220	$4.48e-02$	$2.41e-06$	7.55
580	$1.99e-02$	$1.72e-11$	10.01
700	$1.41e-02$	$1.24e-11$	15.62
820	$1.11e-02$	$4.42e-10$	23.51
2160	0	0	390

Table 3. Errors and CPU time for the sliding window algorithm with varying L and fixed $r = 5$ (model ($\widehat{\text{P2}}$)).

L	E_1	E_2	CPU time (s)
20	$3.34e-01$	$4.47e-03$	15.80
40	$2.36e-01$	$1.71e-03$	7.55
60	$1.80e-01$	$1.40e-03$	6.25
100	$1.15e-01$	$1.90e-04$	5.21
140	$8.48e-02$	$5.18e-04$	5.86
180	$7.70e-02$	$3.80e-04$	7.82
220	$4.54e-02$	$5.54e-05$	10.88
580	$1.60e-02$	$2.27e-05$	54.98
700	$2.28e-02$	$3.00e-04$	82.79
820	$6.70e-03$	$2.10e-05$	105.78
2160	0	0	1077

Table 4. Errors and CPU time for the sliding window algorithm with fixed $L = 40$ and varying r (model (P1)).

r	E_1	E_2	CPU time (s)
5	$1.82e-01$	$1.71e-04$	1.50
10	$1.45e-01$	$2.07e-06$	1.94
15	$1.34e-01$	$3.8e-08$	2.18
20	$1.30e-01$	$1.39e-08$	2.69
25	$1.17e-01$	$2.00e-08$	3.31
30	$1.10e-01$	$1.91e-08$	4.35
35	$1.05e-01$	$2.01e-08$	9.35

Table 5. Errors and CPU time for the sliding window algorithm with fixed $L = 40$ and varying r (modèle ($\widehat{\text{P2}}$)).

r	E_1	E_2	CPU time (s)
5	$2.36e-01$	$1.71e-03$	8.54
10	$1.92e-01$	$5.73e-04$	9.88
15	$1.56e-01$	$2.51e-04$	12.37
20	$1.40e-01$	$1.38e-04$	15.65
25	$1.35e-01$	$1.68e-04$	21.42
30	$1.16e-01$	$9.58e-05$	32.68
35	$1.05e-01$	$8.07e-05$	67.80

5. Conclusion and perspectives

We have presented and studied two optimization problems arising from the modeling of an energy distribution system with storage. We have proven some qualitative mathematical properties of the models. In addition we have performed several numerical experiments that allowed to study the influence of the storage and to confirm the theoretical properties: indeed the storage is an interesting ingredient to decrease the energy cost.

Next, we have proposed and investigated a sliding window algorithm as a way to compute an approximation of a solution in case of large scale problems in a very short time. This new algorithm is characterized by two parameters: The length of the time window and the length of the overlap. We have shown in numerical examples that by choosing suitably these parameters, we can obtain a very good compromise between the CPU time and the accuracy of the solution.

Several interesting extensions of this work could be considered:

- Consider the consumption and price uncertainties. It could be done by introducing stochastic variables in the problem, that allow to better take into account the variability of these quantities.
- Take into account distributed generation such as renewable energy. This can be made either by having prediction models of renewable production, or by including stochastic variables in the model. In the modeling point of view, a node can be added for the distributed generation. The load aggregator can then buy energy in limited quantities via this node at a certain price that can be different from the energy price purchased from the external grid. It is then possible to add constraints to the problem, such as the obligation to use in priority the energy produced locally.
- Reformulate the problem in order to evaluate the economic benefits of storage with respect to the investment and the maintenance costs of the storage units. The benefits will thus depend on the storage parameters and the time period considered. Such a model would be multi-objective.

Acknowledgments

This work was achieved within the project VERTPOM® supported by the French Environment and Energy Management Agency (ADEME), as part of the ”Future Investment Program of the French government.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, C. A. Sagastizábal, *Numerical optimization: Theoretical and practical aspects*, Springer Science & Business Media, 2006.
2. S. Boyd, S. P. Boyd, L. Vandenberghe, *Convex optimization*, Cambridge university press, 2004.
3. T. Chen, Y. Jin, H. Lv, A. Yang, M. Liu, B. Chen, et al. Applications of lithium-ion batteries in grid-scale energy storage systems, *Trans. Tianjin Univ.*, **26** (2020), 208–217.

4. U. D. E. A. Committee, *Bottling electricity: Storage as a strategic tool for managing variability and capacity concerns in the modern grid*, tech. rep., 2008.
5. M. Drouin, H. Abou-Kandil, M. Mariton, *Basic concepts of discrete-time optimal control theory*, In: Control of Complex Systems, Springer, 1991, 11–38.
6. J. Eyer, G. Corey, Energy storage for the electricity grid: Benefits and market potential assessment guide, *Sandia Nat. Laboratories*, **20** (2010), 5.
7. I. Keppo, M. Strubegger, Short term decisions for long term problems–the effect of foresight on model based energy systems analysis, *Energy*, **35** (2010), 2033–2042.
8. B. Kouvaritakis, M. Cannon, *Model predictive control*, Switzerland: Springer International Publishing, (2016), 38.
9. Y. Lee, K. Tay, Y. Choy, Forecasting electricity consumption using time series model, *Int. J. Eng. Technol.*, **7** (2018), 218–223.
10. J. F. Marquant, R. Evins, J. Carmeliet, Reducing computation time with a rolling horizon approach applied to a milp formulation of multiple urban energy hub system, *Procedia Comput. Sci.*, **51** (2015), 2137–2146.
11. F. J. Nogales, J. Contreras, A. J. Conejo, R. Espínola, Forecasting next-day electricity prices by time series models, *IEEE Trans. Power Syst.*, **17** (2002), 342–348.
12. A. Peshkov, O. Alsova, *Short-term forecasting of the time series of electricity prices with ensemble algorithms*, In: Journal of Physics: Conference Series, vol. 1661, IOP Publishing, 2020, p. 012070.
13. K. Poncelet, E. Delarue, D. Six, W. D'haeseleer, *Myopic optimization models for simulation of investment decisions in the electric power sector*, In: 2016 13th International Conference on the European Energy Market (EEM), IEEE, 2016, 1–9.
14. M. Rawa, A. Abusorrah, Y. Al-Turki, S. Mekhilef, M. H. Mostafa, Z. M. Ali, et al. Optimal allocation and economic analysis of battery energy storage systems: Self-consumption rate and hosting capacity enhancement for microgrids with high renewable penetration, *Sustainability*, **12** (2020), 10144.
15. J. Su, T. Lie, R. Zamora, A rolling horizon scheduling of aggregated electric vehicles charging under the electricity exchange market, *Appl. Energy*, **275** (2020), 115406.
16. Y. Xu, C. Singh, Adequacy and economy analysis of distribution systems integrated with electric energy storage and renewable energy resources, *IEEE Trans. Power Syst.*, **27** (2012), 2332–2341.
17. Y. Xu, C. Singh, Power system reliability impact of energy storage integration with intelligent operation strategy, *IEEE Trans. Smart Grid*, **5** (2013), 1129–1137.
18. Y. Xu, L. Xie, C. Singh, *Optimal scheduling and operation of load aggregators with electric energy storage facing price and demand uncertainties*, In: 2011 North American Power Symposium, IEEE, 2011, 1–7.

AIMS Press

© 2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)