



HAL
open science

Optimal deterministic and robust selection of electricity contracts

David Wu, Viet Hung Nguyen, Michel Minoux, Hai Tran

► **To cite this version:**

David Wu, Viet Hung Nguyen, Michel Minoux, Hai Tran. Optimal deterministic and robust selection of electricity contracts. *Journal of Global Optimization*, 2021, 10.1007/s10898-021-01032-z. hal-03257033

HAL Id: hal-03257033

<https://uca.hal.science/hal-03257033>

Submitted on 10 Jun 2021

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.



Distributed under a Creative Commons Attribution 4.0 International License

Optimal deterministic and robust selection of electricity contracts

David Wu · Viet Hung Nguyen · Michel Minoux · Hai Tran

Received: date / Accepted: date

Abstract We address the Electricity Contract Selection Problem (ECSP), of finding best parameters of an electricity contract for a client based on his/her past records of electricity consumption over a fixed time period. The objective is to optimize the electricity bill composed by some fixed cost, the cost of the subscription of the electricity contract and penalties due to overpowering when consumption exceeds subscribed power. The ECSP can be formulated as a convex separable optimization problem subject to total order constraints. Due to this special structure, ECSP is a special case of two well known classes of convex separable optimization problems, namely the minimum network flow under convex separable cost and minimizing convex separable functions under chain constraints. Both classes are well treated in the literature and can be solved in polynomial time [1, 2, 5, 11, 13, 14]. In particular, the algorithm in [2] achieves the best theoretical time complexity assuming that computing the objective function value at one specific point can be done in constant time. However, when we work on a big amount of historic data as in ECSP, the time required for evaluating the objective function cannot be assumed to be $O(1)$ anymore. In this paper, we propose a new algorithm for ECSP which is specially designed to reduce the computational effort over large scale historical data. We present numerical results showing that our algorithm outperforms the algorithm in [2] when applied to consumption data of various types of clients. A robust version of ECSP based on a Seasonal and Trend decomposition approach

David Wu
Sorbonne Université, LIP6 CNRS UMR 7606, Paris, France
E-mail: david.wu@lip6.fr

Viet Hung Nguyen
Université Clermont Auvergne, LIMOS, CNRS UMR 6158, Aubière, France
E-mail: vhunguyen@isima.fr
Corresponding author

Michel Minoux
Sorbonne Université, LIP6 CNRS UMR 7606, Paris, France
E-mail: michel.minoux@lip6.fr

Hai Tran
Energisme SAS, Boulogne-Billancourt, France
E-mail: hai.tran@energisme.com

for modelling consumption uncertainty is also investigated. The resulting worst-case cost minimization problem is shown to be efficiently solvable using the same algorithm as for deterministic ECSP.

Keywords electricity contract, convex separable optimization, robust optimization, active constraint set

1 Introduction

Optimally selecting an energy contract is an important issue for many industrial customers which have an electricity contract for each delivery point. Contracts may need to fix the maximum of total energy consumption in kilowatt-hours and the maximum of peak demand in kilowatt over a large period, typically one year.

The electricity contract problem discussed in [6] can be formulated as a linear program and solved as such in polynomial time. In the problem discussed in [6], there is a fixed cost when the peak demand does not exceed the contract capacity and some additional cost for excess demand.

Combining the optimal energy contract selection problem with a lot sizing problem by considering renewable energy sources has been investigated in [16].

Using a pricing based on the so-called "subscribed power" tends to enforce people to adapt their electricity consumption behaviour in order to match with the subscription [18]. This subscribed power is based on billing the cost of the electricity contract, the cost of some penalty of overpowering with respect to the maximum peak demand defined in the contract and some other fixed costs.

In this paper, we investigate an optimal electricity contract selection problem (ECSP) based on subscribed power which is formulated as minimizing a convex separable function subject to total order constraints. The problem is addressed both in the deterministic case and in the case when consumption data are subject to uncertainty. The paper is organized as follows. Section 2 presents the context and proposes a formulation of the optimization problem considered in its deterministic version. Section 3 presents an efficient algorithm based on an active constraint set approach to solve the optimal selection of electricity contracts. Section 4 investigates a robust version of the problem in which uncertainties on consumption are taken into account. Section 5 presents a series of computational results obtained, both for the deterministic case and for the robust version of the problem. Section 6 presents the main conclusions and some perspectives for future research.

2 Modelling the problem

2.1 Mathematical formulation of ECSP

In France, ENEDIS is responsible for the management of 95% of the electricity distribution network. The tariffs for using the electricity grid are referred to as TURPE (see enedis.fr).

Over the year, consumption periods are classified into several predefined categories, where each category may have its own consumption behaviour. Under the 5th version of TURPE, denoted TURPE5, $K = 5$ categories are considered, namely: winter peak hours (1), winter full hours (2), winter off-peak hours (3), summer

full hours (4) and summer off-peak hours (5). Electricity used during peak hours is billed at a higher rate than electricity used during off-peak hours, creating an incentive for homeowners to reduce energy usage during peak hours. Let T be the total number of 10 minute periods (e.g. $T = 52560$ for a year) and T_i the total number of 10 minutes period in category i . Let $M = \{1, \dots, 12\}$ be the set of all months of the study period, and $\forall i \in \{1, \dots, K\}$, M_i is the subset of M corresponding to category i . For example, for the winter peak hours category with $i = 1$, $M_1 = \{1, 2, 12\}$ which corresponds to the three months of winter. Each small 10 minute period t belongs to a category i and a month $m \in M_i$, noted as $t \in T_{i,m}$, and for each t there is a required consumption level c_t .

Selecting an electricity contract defined by TURPE5 is to choose for each category of consumption i a value of subscribed power $x_i \geq 0$ (in kW). The problem of selecting a best contract thus consists in determining a K -dimensional non-negative vector $x = (x_1, \dots, x_K)$ minimizing the bill due by the consumer. For each category i the corresponding subscribed power x_i represents thresholds of consumption, viewed as a commitment of maximum peak demand for each 10 minutes period.

There is an annual cost for the subscription equal to $\sum_i s_i x_i$ (for given $s_i > 0$ in €/kW/year), this cost is an increasing linear function of x . Excess demand of consumption, called overpowering, is allowed but in that case extra penalty costs are due. The overpowering quantity for t is $\delta_t(x_i) = \max(0, c_t - x_i)$. For each category i and month $m \in M_i$, the penalty cost is $p_i \sqrt{\sum_{t \in T_{i,m}} \delta_t(x_i)^2}$ (for given $p_i > 0$ in €/kWh), which is a decreasing function of x_i . There is also some fixed cost unrelated to subscribed power (e.g. consumption, transports and taxes) which is not necessary to our analysis. The objective function of the problem is the total annual cost $F(x)$ defined as the sum of the annual costs $F_i(x_i)$ for the various categories $i \in \{1, \dots, K\}$:

$$F_i(x_i) = s_i x_i + p_i \sum_{m \in M_i} \sqrt{\sum_{t \in T_{i,m}} \delta_t(x_i)^2}. \quad (1)$$

(see Figure 1 for a graphical representation of a function $F_i(x_i)$).

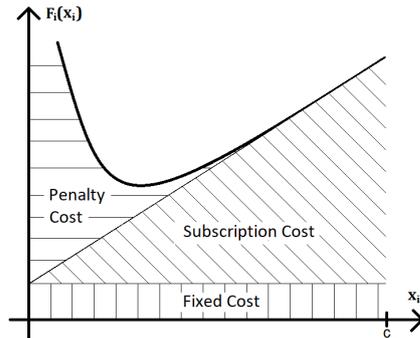


Fig. 1: Objective function for category i : $F_i(x_i)$

In order to ensure the stability of the network load, TURPE5 imposes that subscribed powers in high consumption categories (e.g. winter peak hours) must be lower than subscribed powers in low consumption categories (e.g. summer off-peak hours). This results into a series of $K-1$ constraints of the form $\forall i < K, x_i \leq x_{i+1}$, defining a total order on the x_i values.

Let \underline{c}_i (resp.: \bar{c}_i) be the smallest (resp.: greatest) value of consumed power during the time slots in category i . Then we denote by \underline{c} (resp.: \bar{c}) the smallest (resp.: greatest) consumed power among the instance. Then the x_i variables should meet the bound constraints $0 \leq \underline{c} \leq x_i \leq \bar{c}$. Let $C = \bar{c} - \underline{c} + 1$ be the number of integer values among which the optimal solution is searched. This parameter C together with the number of categories K will be useful for expressing the worst-case time complexity of our algorithm.

The ECSP can then be formulated as the following mathematical program \mathcal{P} :

$$\mathcal{P} : \min F(x) = \sum_{i=1}^K F_i(x_i) \quad (2)$$

$$\text{s.t.} \quad x_i \leq x_{i+1}, \quad 1 \leq i \leq K-1, \quad (3)$$

$$\underline{c} \leq x_i \leq \bar{c}, \quad 1 \leq i \leq K, \quad (4)$$

The constraints (3) are called *the order constraints* and the constraints (4) are *the bound constraints*. A solution for \mathcal{P} is a K -dimensional vector $\mathbf{x} = (x_1, \dots, x_K)^T$ satisfying the constraints (3) and (4). The objective function (2) is separable and convex as stated in the following Proposition 1.

Proposition 1 *Each F_i is convex with respect to x_i .*

Proof For all $i \in \{1, \dots, T\}$, let $g_t(x_i)$ be an univariate convex function such that $g_t(x_i) \geq 0$ for all $x_i \geq 0$.

For all $x \in \mathbb{R}^T$, let us denote $G(x) = \sqrt{\sum_{i=1}^T g_t(x_i)^2}$ the Euclidean Norm (L_2)

$$\text{of } \begin{pmatrix} g_1(x_1) \\ \vdots \\ g_T(x_T) \end{pmatrix} \text{ i.e. } G(x) = \left\| \begin{pmatrix} g_1(x_1) \\ \vdots \\ g_T(x_T) \end{pmatrix} \right\|.$$

$$\text{Let } \bar{x} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_T \end{pmatrix} \geq 0, \hat{x} = \begin{pmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_T \end{pmatrix} \geq 0 \text{ and } \lambda \in [0, 1].$$

From the convexity of the g_t functions and the fact that the norm is a non decreasing function on the set of non negative vectors, it holds:

$$\begin{cases} g_1(\lambda \bar{x}_1 + (1-\lambda)\hat{x}_1) \leq \lambda g_1(\bar{x}_1) + (1-\lambda)g_1(\hat{x}_1) \\ \vdots \\ g_T(\lambda \bar{x}_T + (1-\lambda)\hat{x}_T) \leq \lambda g_T(\bar{x}_T) + (1-\lambda)g_T(\hat{x}_T) \end{cases}$$

Therefore:

$$\left\| \begin{pmatrix} g_1(\lambda \bar{x}_1 + (1-\lambda)\hat{x}_1) \\ \vdots \\ g_T(\lambda \bar{x}_1 + (1-\lambda)\hat{x}_1) \end{pmatrix} \right\| \leq \left\| \begin{pmatrix} \lambda g_1(\bar{x}_T) + (1-\lambda)g_1(\hat{x}_T) \\ \vdots \\ \lambda g_T(\bar{x}_T) + (1-\lambda)g_T(\hat{x}_T) \end{pmatrix} \right\|$$

Now, from the convexity of the Euclidean norm, it follows:

$$\left\| \begin{array}{c} \lambda g_1(\bar{x}_1) + (1-\lambda)g_1(\hat{x}_1) \\ \vdots \\ \lambda g_T(\bar{x}_T) + (1-\lambda)g_T(\hat{x}_T) \end{array} \right\| \leq \lambda \left\| \begin{array}{c} g_1(\bar{x}_1) \\ \vdots \\ g_T(\bar{x}_T) \end{array} \right\| + (1-\lambda) \left\| \begin{array}{c} g_1(\hat{x}_1) \\ \vdots \\ g_T(\hat{x}_T) \end{array} \right\|$$

In view of this, the following inequality is shown to hold:

$$\left\| \begin{array}{c} g_1(\lambda\bar{x}_1 + (1-\lambda)\hat{x}_1) \\ \vdots \\ g_T(\lambda\bar{x}_T + (1-\lambda)\hat{x}_T) \end{array} \right\| \leq \lambda \left\| \begin{array}{c} g_1(\bar{x}_1) \\ \vdots \\ g_T(\bar{x}_T) \end{array} \right\| + (1-\lambda) \left\| \begin{array}{c} g_1(\hat{x}_1) \\ \vdots \\ g_T(\hat{x}_T) \end{array} \right\|$$

and from this, we conclude that:

$$\sqrt{\sum_{i=1}^T g_t(\lambda\bar{x}_i + (1-\lambda)\hat{x}_i)^2} \leq \lambda \sqrt{\sum_{i=1}^T g_t(\bar{x}_i)^2} + (1-\lambda) \sqrt{\sum_{i=1}^T g_t(\hat{x}_i)^2},$$

which proves the desired convexity property of G .

For each category i the subscription cost is linear in x_i and the penalty cost is of the form: $p_i G_i(x_i)$ with $G_i(x_i) = \sqrt{\sum_{t=1}^{T_i} g_t(x_i)^2}$, where $g_t(x_i) = \delta_t(x_i)$ is a convex function of x_i . Thus, for each category i , F_i is convex w.r.t. x_i . \square

2.2 State of the art and motivation

The ECSP can be reformulated as a minimum separable convex cost network optimization problem in a graph featuring K nodes and $2K-2$ arcs (see Appendix). Thus, in order to solve \mathcal{P} , one might well consider applying some of the existing solution procedures, in particular:

- the algorithm proposed in [13, 14] which, in our case, would lead to a worst-case complexity $O(K^3 \log C)$;
- the algorithms proposed in [1] and in [11] which would lead to a worst-case complexity $O(K^2 \log(K) \log(KC))$.

For a survey of solution algorithms for convex cost network optimization problems and related nonlinear optimization problems, we refer the reader to [9]. Note that all the previous algorithms use various versions of minimum cost flow network algorithms running on various specific types of graphs. Some of these graphs could be of very big size (e.g. the number of the arcs could be KC in the graph considered in [1]) and the construction of these graphs is not taken into account in the time complexity of algorithm. Moreover, the algorithms in [1] and [11] need advanced data structures such as dynamic trees to achieve the time complexity of $O(K^2 \log(K) \log(KC))$. Without them, the worst-case complexity would remain $O(K^3 \log C)$.

The ECSP also belongs to the class of convex separable function minimization problems under total order constraints. This class of problem appears in the context of isotonic regression which is a well-known problem in statistics. Given K

data c_1, \dots, c_K , the Isotonic Regression Problem (IRP) consists in finding K values x_1, \dots, x_K minimizing the norm p distance between the solution and the data subject to order constraints $x_1 \leq \dots \leq x_K$. Formally, it is formulated as follows:

$$\begin{aligned} \min \quad & \sqrt[p]{\sum_i (x_i - c_i)^p} \\ \text{s.t.} \quad & x_1 \leq x_2 \leq \dots \leq x_K. \end{aligned}$$

The *generalized IRP* is obtained when the objective function is replaced by any convex separation function.

One of the most efficient algorithms for solving generalized IRP has been proposed by Ahuja and Orlin [2] under the name of 'Scaling_PAV'. Assuming that computing (1) at any x in \mathbb{R}^K can be done in constant time $O(1)$, its complexity is $O(K \log(C))$. In Scaling_PAV, instead of calculating the exact optimal solution of each $F_i(x_i)$ at each iteration, one only updates the intervals to which the optimal solution should belong. The K intervals are all initially set equal to (\underline{c}, \bar{c}) and reduced by one half after each iteration. As new order constraints are successively detected and saturated, the number of intervals is progressively decreased as variables are fused. After $\log_2(\frac{C}{\epsilon})$ iterations, all the intervals have reduced length less than ϵ , and an ϵ -optimal solution is found. Since, in each iteration, one works with a at most K intervals, the worst-case time complexity of Scaling_PAV is $O(K \log(C))$.

Obviously, the Scaling_PAV algorithm could be applied to the ECSP addressed here. However, in this context of application, the evaluation of the resulting complexity has to be completely reconsidered, because careful examination reveals that the computational effort required to compute each value $F_i(x_i)$ involved in the objective function (2) cannot be assumed to be $O(1)$. Indeed, from the expression given in (1), it is readily seen that computing $F_i(x_i)$ for any given value of x_i , requires $O(N^+(x_i))$ arithmetic operations, where $N^+(x_i)$ is defined as the number of time steps t in T_i such that $c_t > x_i$. Thus, for large values of x_i (i.e. values close to $\max(c_t, t \in T_i)$), both $N^+(x_i)$ and the computation time are reduced. By contrast, for smaller values of x_i , $N^+(x_i)$ can be of the same order of magnitude as $|T_i|$, the total number of time steps in the definition of F_i , thus leading to possibly much bigger computation time. For further discussion and illustration of this dependence of $N^+(x_i)$ on x_i , please refer to the forthcoming Section 3.2. Clearly, since $|T_i|$ (the number of consumption data in the category i) can be much larger than K , the number of decision variables, this parameter turns out to be a key factor to be taken into account in the evaluation of an algorithm for solving \mathcal{P} . Thus, in the search for computational efficiency in the solution of ECSP, the dependence of $N^+(x_i)$ (a measure of the computational effort) with respect to x_i has to be taken into account.

In the present paper, we propose a special-purpose algorithm for solving ECSP which will turn out to be practically more efficient than Scaling_PAV, thanks to a proper exploitation of the specific features of the problem pointed out above. The proposed algorithm, which will be called OPTIM_SP, is based on the same scheme of iterative active constraint set detection as Scaling_PAV. However, its implementation makes essential use of an extended asymmetric binary search procedure, thanks to which the computational effort required in each iteration to minimize the relevant part of the objective function (2) can be significantly reduced, as compared with usual binary search. It is also worth mentioning here

that this possibility of resorting to this extended binary search procedure could not be applied within the Scaling_PAV algorithm for the following reason: when running Scaling_PAV, at any given step all the intervals *must have the same length*, and this cannot be the case in case of using the proposed extended binary search.

3 OPTIM_SP: an efficient algorithm based on an active constraint set approach

For $k \in \{1, \dots, K-1\}$, let the k^{th} order constraint denote the constraint $x_k \leq x_{k+1}$. For $k \in \{0, \dots, K-1\}$ let \mathcal{P}^k denote the relaxation of the problem \mathcal{P} where the last $K-1-k$ order constraints are relaxed, i.e. the order constraints in \mathcal{P}^k are only the first k order constraints:

$$\begin{aligned} \mathcal{P}^k : \min F(x) \\ \text{s.t.} \quad & x_i \leq x_{i+1}, & 1 \leq i \leq k, \\ & \underline{c} \leq x_i \leq \bar{c}, & 1 \leq i \leq K. \end{aligned}$$

Note that \mathcal{P}^0 is the relaxation of \mathcal{P} without the order constraints and \mathcal{P}^{K-1} is \mathcal{P} itself.

More generally, for $B \subseteq \{1, \dots, K-1\}$, let $\mathcal{P}(B)$ denote the relaxation of \mathcal{P} where all the order constraints are relaxed except those indexed in B :

$$\begin{aligned} \mathcal{P}(B) : \min F(x) \\ \text{s.t.} \quad & x_i \leq x_{i+1}, & i \in B, \\ & \underline{c} \leq x_i \leq \bar{c}, & 1 \leq i \leq K. \end{aligned}$$

In addition, we denote $\mathcal{P}^=(B)$ the restriction of $\mathcal{P}(B)$ where the order constraints in B are set to equality:

$$\begin{aligned} \mathcal{P}^=(B) : \min F(x) \\ \text{s.t.} \quad & x_i = x_{i+1}, & i \in B, \\ & \underline{c} \leq x_i \leq \bar{c}, & 1 \leq i \leq K. \end{aligned}$$

Note that, if $B = \emptyset$, then $\mathcal{P}^=(B) = \mathcal{P}^0$.

3.1 Algorithm's overview

OPTIM_SP initially solves \mathcal{P}^0 and then performs $K-1$ iterations thus solving successively $\mathcal{P}^1, \mathcal{P}^2, \dots, \mathcal{P}^{K-1}$. For $k = 0, \dots, K-1$, let $\mathbf{x}^k = (x_1^k, \dots, x_K^k)^T$ denote the optimal solution of \mathcal{P}^k found by OPTIM_SP and let B^k denote the set of the indices of the active order constraints in \mathcal{P}^k associated with \mathbf{x}^k , i.e. the order constraints in \mathcal{P}^k that \mathbf{x}^k satisfies at equality. At each iteration $k = 1, \dots, K-1$, OPTIM_SP sets $B \leftarrow B^{k-1}$ and $\mathbf{x} \leftarrow \mathbf{x}^{k-1}$. Then OPTIM_SP iteratively finds the order constraint of greatest index i in \mathcal{P}^k violated by \mathbf{x} , updates $B \leftarrow B \cup \{i\}$ and finds a new x by solving $\mathcal{P}^=(B)$. If no such index can be found then OPTIM_SP sets $\mathbf{x}^k \leftarrow \mathbf{x}$ and $B^k \leftarrow B$ and terminates the iteration k . OPTIM_SP can be formally stated as Algorithm 1.

Algorithm 1 OPTIM_SP

```

1: Set  $B \leftarrow \emptyset$ 
2: Solve  $\mathcal{P}^=(B)$  and set  $\mathbf{x}$  equal to the obtained optimal solution.
3: for  $k$  from 1 to  $K-1$  do
4:   Set  $b \leftarrow True$ .
5:   repeat
6:     Find the greatest index  $1 \leq i \leq k$  such that  $x_i > x_{i+1}$ .
7:     Set  $b \leftarrow False$  if such an index does not exist.
8:     if  $b=True$  then
9:       Set  $B \leftarrow B \cup \{i\}$ 
10:      Solve  $\mathcal{P}^=(B)$  and update  $\mathbf{x}$ .
11:     end if
12:   until  $b=False$ 
13: end for
14: return  $\mathbf{x}$ 

```

Algorithm 2 Ajustable binary search

```

1: function  $y_{opt} \leftarrow A\_SEARCH(F_I(y), \underline{y}, \bar{y}, a)$ 
2:    $\epsilon \leftarrow 0.1$ 
3:   repeat
4:      $ymid \leftarrow \lfloor (y + a * \bar{y}) / (1 + a) \rfloor$ 
5:      $left\_cost \leftarrow F_I(ymid - \epsilon)$ 
6:      $right\_cost \leftarrow F_I(ymid + \epsilon)$ 
7:     if  $left\_cost \leq right\_cost$  then
8:        $\bar{y} \leftarrow ymid$ 
9:        $a \leftarrow 1$ 
10:    else
11:       $y \leftarrow ymid$ 
12:    end if
13:  until  $y + 1 = \bar{y}$ 
14:   $y_{opt} \leftarrow ymid$ 
15:  return  $y_{opt}$ 
16: end function

```

3.2 Adjustable binary search for optimizing univariate convex functions $F(x)$

Each step of Algorithm 1 requires the solution of a problem of univariate convex function minimization over a given interval. To achieve this, we propose an extended binary search procedure called A_SEARCH, formally stated as Algorithm 2. A_SEARCH is an adjustable version of binary search which depends on the input parameter $a > 0$. Thus, for given univariate convex function F_I and an interval $[\underline{y}, \bar{y}]$ where $I \subset \{1, \dots, K\}$ and $F_I(y) = \sum_{j \in I} F_j(y)$, the function A_SEARCH returns:

$$y_{opt} = \underset{\substack{\underline{y} \leq y \leq \bar{y} \\ y \text{ integer}}}{\operatorname{argmin}} F_I(y).$$

The adjustable binary search starts with a lower bound \underline{y} and an upper bound \bar{y} . Until y_{opt} is found, the algorithm computes the integer middle point $ymid = \lfloor (y + a * \bar{y}) / (1 + a) \rfloor$ according to a parameter a , and computes objective values $left_cost$ and $right_cost$ for the two neighbours at $ymid - \epsilon$ and $ymid + \epsilon$, then it updates either y or \bar{y} .

Please note that the standard binary search procedure corresponds to the choice of $a = 1$ in A_SEARCH. For this value of a , $ymid$ is the middle of the interval

(\underline{y}, \bar{y}) . For $a > 1$, $ymid$ is no longer the middle of the interval, and we refer to this case as ‘asymmetric binary search’. The motivation behind the use of $a > 1$ stems from the observation that the computational effort $O(N_i^+(x_i))$ required to compute the value of one (individual aggregated) component of the objective function as defined in (1) is strongly dependent on the value of the parameter x_i . This dependence is illustrated in Figure 3 on three typical examples of ECSP corresponding to data sets D_1 and D_5 , two of the data sets used in the computational experiments discussed in Section 3 below. More precisely, Figure 3(A) corresponds to dataset D_1 and $i = 1$; Figure 3(B) corresponds to data set D_1 and $i = 5$; Figure 3(C) corresponds to data set D_5 and $i = 4$.

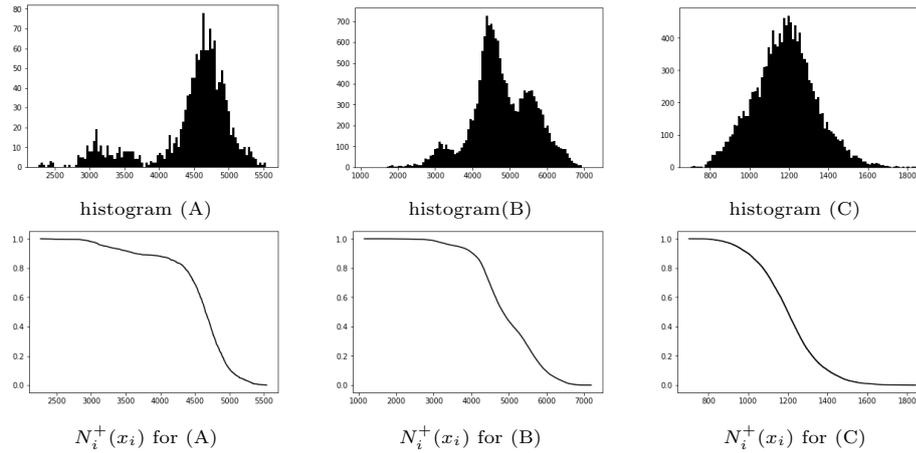


Fig. 3: Histograms and curves showing the dependence of $N_i^+(x_i)$ on x_i for three examples which are part of the data sets D_1 and D_5 described in Section 5.

Note that when $a > 1$ does not allow anymore the binary search to continue on the smaller subinterval on the right of $ymid$, we set the value of a to 1 (see line 9 of Algorithm 2) and resume the last iterations with the standard symmetric binary search. With this switch on the value of a , as our binary search scheme performs at least as well as the symmetric binary search scheme on all iterations except perhaps one, it can be checked that the number of iterations in A_SEARCH is at most $\log_2(\bar{y} - \underline{y}) + 1$.

3.3 Solving $\mathcal{P}^=(B)$

To perform line 2 in OPTIM_SP which consists in solving $\mathcal{P}^=(\emptyset)$, i.e. solving \mathcal{P}^0 , we call the A_SEARCH function for minimizing each $F_i(x)$ for $i = 1, \dots, K$, i.e. $x_i = \text{A_SEARCH}(F_i(x), \underline{c}_i, \bar{c}_i, a)$. It is easy to see that after execution of line 2, \mathbf{x} is an optimal solution of $\mathcal{P}^=(B)$ with $B = \emptyset$.

We now explain the procedure corresponding to line 10 of OPTIM_SP which consists in solving $\mathcal{P}^=(B)$ with $B \neq \emptyset$. Let us suppose that we are in iteration k ($1 \leq k \leq K - 1$) and we are ready to execute line 10 after having added an

index i to B in line 9. Let $B' = B \setminus \{i\}$, i.e. the value of B before executing line 9. Let \mathbf{x}' be the optimal solution of $\mathcal{P}^=(B')$, i.e. the value of \mathbf{x} before the line 10, we have then $x'_i > x'_{i+1}$. Let \underline{i} and \bar{i} respectively denote the smallest and the greatest indices of the block of consecutive indices in B which contains i . Let $I = \{\underline{i}, \underline{i} + 1, \dots, \bar{i}, \bar{i} + 1\}$. Line 10 is executed by just calling $\text{A_SEARCH}(F_I(x), x'_i, x'_i, a)$ where $F_I(x) = \sum_{j \in I} F_j(x)$. This function call returns an integer value y_{opt} and the update of \mathbf{x} (in line 10 of Algorithm 1) is done as follows: $x_j \leftarrow x'_j$ for $j \notin I$ and $x_j \leftarrow y_{opt}$ for $j \in I$.

3.4 Correctness of OPTIM_SP

A partition of the index set $\{1, 2, \dots, K\}$ (of the variables) $J = \{J_1, \dots, J_k\}$ is called a *block partition* if its elements are subsets (or blocks) of consecutive indices. Given an $\mathbf{x} \in \mathbb{R}^K$, \mathbf{x} is *conform to J* if $x_i = x_j$ for all i and j belonging to the same block in J . The correctness of OPTIM_SP is based on well-known results in the literature. In particular, we quote here the following lemma (Lemma 2 in [2] which uses results in [5]) appropriately restated to fit our context.

Lemma 1 [2] *If $J = \{J_1, \dots, J_k\}$ is a block partition of $\{1, 2, \dots, K\}$ such that there exists a $\mathbf{x} \in \mathbb{R}^K$ conform to J which is also feasible for \mathcal{P} then \mathbf{x} is an optimal solution for \mathcal{P} .*

Then we can state:

Theorem 1 *The solution \mathbf{x} output by OPTIM_SP is optimal for \mathcal{P} .*

Proof As we cannot find a greatest index $1 \leq i \leq K - 1$ such that $x_i > x_{i+1}$, \mathbf{x} is feasible for \mathcal{P} . Let \bar{B} denote the last value of B after termination of OPTIM_SP then \bar{B} is the set of active constraints that \mathbf{x} satisfies at equality. Let us build a block partition J of $\{1, 2, \dots, K\}$ from \bar{B} as follows:

- For each of block of consecutive indices I in \bar{B} with \underline{i} and \bar{i} as respectively the smallest and the greatest indices in I , let us refer to $\{\underline{i}, \underline{i} + 1, \dots, \bar{i}, \bar{i} + 1\}$ as a block in J .
- For an index $1 \leq j \leq K$, if neither j nor $j - 1$ belongs to \bar{B} , then let us refer to $\{j\}$ as a single block in J .

It is then easy to see that \mathbf{x} is a feasible solution of \mathcal{P} conform to J . Hence, by Lemma 1, \mathbf{x} is optimal for \mathcal{P} . \square

3.5 Complexity of OPTIM_SP

Proposition 2 *The worst-case time complexity of OPTIM_SP is $O(K^2 \log(C))$ provided that the evaluation of each $F_i(x)$ at a specific point x can be done in $O(1)$.*

Proof We can see that the worst case occurs when the set B in line 10 of OPTIM_SP contains successively $1, 2, \dots, K - 1$ elements and OPTIM_SP calls $K - 1$ times $\text{A_SEARCH}(F_I(x), x'_i, x'_i, a)$ in line 10 with the sets I (as defined in Section 3.3) containing successively $2, 3, \dots, K$ elements. Hence, the total number of times that the functions $F_i(x)$ ($i = 1, \dots, k$) are minimized in the executions of A_SEARCH

is $\frac{K(K+1)}{2} - 1$. We should also include the minimization of each $F_i(x)$ in the K calls of A_SEARCH for solving \mathcal{P}^0 at line 2 of OPTIM_SP. Hence, overall there are $\frac{K(K+1)}{2} - 1 + K = \frac{K(K+3)}{2} - 1$ minimizations of $F_i(x)$ in the calls of A_SEARCH. In each minimization of $F_i(x)$ by A_SEARCH, $F_i(x)$ is evaluated at most $\log_2(C) + 1$ times as pointed out in Section 3.2. Hence, the number of evaluations of the functions $F_i(x)$ is $O(K^2 \log(C))$ in the worst case. \square

4 OPTIM_SP for a robust version of the problem

Now, if we consider input data as forecasted consumption, actual realisation of electricity consumption can be anticipated to be different from the forecast. Indeed the deviation from predicted consumption can create greater overpowering, the consequence being to increase the electricity bill. In the sequel, we discuss a robust version of the problem of optimal selection of electricity contract and show how it can still be solved using OPTIM_SP algorithm.

4.1 Modelling customer demand

Customer demands are often represented as stochastic process based on influencing factors. Those influencing factors are often external condition (temperature, climate...), physical characteristic of dwelling (type, age...), devices and occupants (occupation and behaviour...), prices and subscriptions [3, 12, 21]. Works on lighting energy in large office building show that consumption can be accurately simulated by considering occupant behaviour and seasonal variations, and can be described using e.g. Poisson and Normal distribution [21]. The consumption model is useful to predict consumption in the short-term (1 day), medium-term (3 days), long-term (7 days) [19] and very long-term (1 year) [3]. Many works are based on precise modelling of each device in household [10, 15, 17]. The authors of [8] present an approach based on hidden Markov chains for the modelling and statistical analysis of electric consumption curves.

The "Seasonal and Trend decomposition using Loess" (STL) method [7] decomposes the consumption as:

$$raw_data(t) = trend(t) + seasonal(t) + remainder(t).$$

The 'trend' component corresponds to non-stationary long-term evolution of consumption, the 'seasonal' component corresponds to periodical evolution (e.g. period = 1 week), and the remainder (or the residual) variation is what is left over after fitting the model and will be viewed as a realisation of some random process.

In the following, consumption is viewed as a time series in which each sample is collected every 10 minutes. For each category i , we apply the STL method to decompose the time series and the important parameter to characterize uncertainty is taken to be the standard deviation σ_i of the remainder. Then for 10 minute period $t \in \{1, \dots, T_i\}$ we will typically consider that consumption can deviate from the trend + seasonal forecast by a value represented as an independent realisation of the appropriate probability distribution. In Section 5 below, it will be assumed

that the residuals are independent realizations of some known probabilistic distribution (typically a truncated normal distribution) with zero mean and known standard deviation.

4.2 Robust problem formulation

We propose here to define a robust version of the problem using a concept of uncertainty set similar to the one introduced in [4]. Following the latter, the uncertainty set, corresponding to ν uncertain parameters μ_1, \dots, μ_ν , each taking bounded values between 0 and β , and having a very small probability of all simultaneously taking their maximum value β , would typically be the polyhedron defined by the inequalities $0 \leq \mu_i \leq \beta$ and $\sum_{i=1}^{\nu} \mu_i \leq \Gamma\beta$, where $\Gamma\beta$ is the so-called budget of uncertainty. Increasing $\Gamma\beta$ increases the range of possible scenarios against which robustness is to be achieved, thus improving the robustness of the solutions obtained but at the expense of increasing the solution cost.

For each category i , let $v = (v_1, \dots, v_{|T_i|})$ be the variable vector of possible deviations, where deviations typically are bounded as $-b_t \leq v_t \leq b_t$. However, in view of the max operator in the expression of the penalty cost, negative deviations will not increase the objective function value. We consider that for a given category i , all b_t ($t \in T_i$) are equal and we denote b_i this common value. Finally, we consider each deviation is bounded as $0 \leq v_t \leq b_i$, where the value b_i is chosen to be of the same order of magnitude as the standard deviation σ_i (typically $b_i = r * \sigma_i$ with r chosen between 2 and 3).

In line with [4], denoting B_i the budget of uncertainty for category i , we allow a total deviation at most equal to B_i , thus the deviation v has to satisfy the constraint: $\sum_{t=1}^{T_i} v_t \leq B_i$. As an indication about how to choose B_i , consider n realisations of a random variable normally distributed with zero mean and standard deviation σ_i . The sum of those n random realisations has a variance $\sigma^2 = n\sigma_i^2$, i.e. its standard deviation is $\sigma = \sqrt{n}\sigma_i$. In line with this remark, taking V_i to be an integer value of the same order as $\sqrt{T_i}$, we define the uncertainty set U_i as:

$$U_i = \{v | 0 \leq v_t \leq b_i, \forall t, \sum_{t=1}^{T_i} v_t \leq B_i = V_i b_i\}.$$

For defining the robust version of the problem, the overpowering quantity for period t is now $\delta_t^r(x_i, v_t) = \max(0, (c_t + v_t) - x_i)$, note that $\delta_t^r(x_i, v_t) = \delta_t(x_i - v_t)$. For given x_i , the worst-case penalty cost for category i in the robust version of the problem is obtained as the maximum value over the uncertainty set U_i of the penalty cost function, which is defined as:

$$\Pi_i^r(x_i) = \max_{v \in U_i} \sum_m \sqrt{\sum_{t \in T_{i,m}} (\delta_t^r(x_i, v_t))^2}. \quad (5)$$

The objective function for the robust formulation is $F^r(x) = \sum_i F_i^r(x_i) = \sum_i s_i x_i + p_i \Pi_i^r(x_i)$ and the robust version of ECSP can be stated as:

$$\begin{aligned} \mathcal{P}^r : \quad & \min F^r(x) \\ \text{s.t.} \quad & x_i \leq x_{i+1}, \quad 1 \leq i \leq K-1, \\ & \underline{c} \leq x_i \leq \max_{1 \leq i' \leq K} \{\bar{c}_{i'} + b_{i'}\}, \quad 1 \leq i \leq K. \end{aligned}$$

We note that in the above formulation, the upper bound values on the variables x_i have been changed to take into account the possible increase of the maximum value of consumption by b_i for category i .

Proposition 3 *Each $\Pi_i^r(x_i)$ is convex in x_i .*

Proof For any fixed $v \in U_i$, we know from Proposition 1 that the function:

$$\sum_m \sqrt{\sum_{t \in T_{i,m}} (\delta_t^r(x_i, v_t))^2}$$

is convex in x_i . Now, since the maximum value in (5) is to be determined with respect to the (finite) set of extreme points of U_i , Π_i^r is recognized as the pointwise maximum of a finite collection of functions, each of which is convex in x_i . This proves the claimed result. \square

Corollary 1 *Each F_i^r is convex in x_i , and hence \mathcal{P}^r is a convex program.*

4.3 Using OPTIM_SP to solve the robust version of the problem

Since it appears that the robust version of the optimum contract selection problem has the same structure as the deterministic version, OPTIM_SP algorithm turns out to be readily applicable to the robust version of the optimum contract selection problem. However in each call to A_SEARCH function, for a given category i and a given value x_i , whenever we have to evaluate the value $F_i^r(x_i)$, we need now to solve the sub-problem (5).

Proposition 4 *For a given x_i , any optimal solution to the sub-problem (5) features V_i components equal to b_i , all the other components being 0.*

Proof For a category i , the objective of the sub-problem is to maximize a convex function w.r.t. v , and it is a well-known fact that the maximum is reached at an extreme point of the polyhedron representing the uncertainty set U_i . Now, we just have to observe that, since V_i has been assumed to be an integer value, each extreme point of the polyhedron U_i features exactly V_i non zero components equal to their upper bound values b_i . \square

For the category i and given x_i , according to proposition 4 the maximization sub-problem (5) is solved when the V_i uncertain parameters that maximize the objective function are found i.e. we need to find how to best dispatch those V_i time slots over the months M_i . This optimum dispatch problem, (5) can be cast into a *Multiple-Choice Knapsack Problem* (MCKP) with integer variables, which according to [20] can be solved in worst-case time complexity $O(N)$ ($N = V_i * |M_i|$ is the number of items). Let us explain how this MCKP is formulated.

Let the variable $z_{m,n} = 1$ if the algorithm assigns $n \in \{0, \dots, V_i\}$ deviations to the month $m \in M_i$, else $z_{m,n} = 0$. Let $T_{i,m}^n$ denote the subset of $t \in T_{i,m}$ achieving the n highest values of c_t .

Proposition 5 *For a given category i and month m , if exactly n deviation parameters v_t , $t \in T_{i,m}$, may take on their upper bound value b_i , then the worst-case cost value for month m is obtained when $v_t = b_i$, for all $t \in T_{i,m}^n$.*

Proof For a given category i and month m , the penalty cost function is maximized by setting maximum deviations for period t with the highest nonzero δ_t^r values, i.e. exactly n deviation parameters may take on their upper bound value b_i , then the maximum cost value is obtained when $v_t = b_i$ for all $t \in T_{i,m}^n$. \square

According to proposition 5, when n parameters are set equal to their upper bound b_i the partial objective cost for month m is:

$$\gamma_{m,n} = \sqrt{\sum_{t \in T_{i,m}^n} (\delta_t^r(x_i, b_i))^2 + \sum_{t \in T_{i,m} \setminus T_{i,m}^n} (\delta_t^r(x_i, 0))^2}.$$

The MCKP to be solved in order to determine the worst-case cost is then:

$$\begin{aligned} \text{MKCP:} \quad & \max \sum_{m=1}^{M_i} \sum_{n=0}^{V_i} \gamma_{m,n} z_{m,n} \\ & \text{s.t.} \quad \sum_{m=1}^{M_i} \sum_{n=0}^{V_i} n z_{m,n} \leq V_i, \\ & \quad \sum_{n=0}^{V_i} z_{m,n} \leq 1, \forall m, \\ & \quad z_{m,n} \in \{0, 1\}, \forall m \forall n. \end{aligned}$$

Recall that $M_i \subseteq M$ and V_i is an integer value of the same order of magnitude as $\sqrt{T_i}$ ($\leq \sqrt{T}$), thus for a given x_i determining the optimal value of (5) takes $O(|M| * \sqrt{T})$ worst-case time complexity. Therefore we can state:

Proposition 6 *The worst-case time complexity of OPTIM_SP applied to the robust version of the optimum contract selection problem is $O(K^2 M \sqrt{T} \log C)$.*

5 Computational experiments

5.1 Experiments on the deterministic version of ECSP

This section is devoted to a detailed computational study of the OPTIM_SP algorithm on a series of typical real instances of the ECSP problem. For each instance, the evolution of CPU time as a function of the value chosen for the a parameter is analyzed. The results obtained are also compared against those which would be obtained by using the Scaling_PAV algorithm on the same data.

In our computational experiments, we consider 5 data sets referred to as D_1 to D_5 . Each data set is defined by specifying the 52560 values of electricity consumption of the customer under consideration, recorded in every 10-minute period along a full year (2017). This set of values is decomposed into 5 subsets (time categories) T_1 to T_5 , and each T_i is in turn decomposed into several subsets (corresponding to the relevant months during which consumption has been observed (these subsets are denoted $T_{i,m}$ for some m in M). Data set D_1 corresponds to a big factory in the food industry with subscribed power typically in the range 5000-7500 KW; data set D_2 corresponds to a company providing maintenance of transportation equipment with subscribed power typically in the range 1500-2500 KW; data set

	D_1	D_2
$[c_1, \dots, c_K]$	[2269, 2056, 72, 1769, 1152]	[1430, 1242, 1189, 910, 358]
$[\bar{c}_1, \dots, \bar{c}_K]$	[5536, 6732, 6667, 7539, 7189]	[2419, 2446, 2307, 1822, 1768]
\mathbf{x}^0	[5536, 6573, 6395, 6949, 6738]	[2419, 2425, 2254, 1730, 1691]
\mathbf{x}^*	[5536, 6479, 6479, 6914, 6914]	[2317, 2317, 2317, 2317, 2317]

	D_3	D_4
$[c_1, \dots, c_K]$	[533, 284, 208, 183, 2]	[480, 193, 321, 375, 315]
$[\bar{c}_1, \dots, \bar{c}_K]$	[1514, 1596, 1603, 1610, 1588]	[723, 737, 667, 722, 680]
\mathbf{x}^0	[1514, 1568, 1465, 1541, 1481]	[723, 736, 643, 700, 627]
\mathbf{x}^*	[1514, 1526, 1526, 1533, 1533]	[711, 711, 711, 711, 711]

	D_5
$[c_1, \dots, c_K]$	[735, 545, 664, 572, 665]
$[\bar{c}_1, \dots, \bar{c}_K]$	[1302, 1464, 1818, 2292, 1594]
\mathbf{x}^0	[1302, 1449, 1238, 1716, 1449]
\mathbf{x}^*	[1302, 1382, 1382, 1662, 1662]

Table 1: Main characteristics of the data sets D_1, \dots, D_5

D_3 corresponds to an industrial bakery with subscribed power typically in the range 1000-1500 KW; D_4 and D_5 correspond to two big hotels with subscribed power typically in the range 500-1500 KW.

The main characteristics of each data set are shown in Table 1, namely:

- for each time category, the minimum and maximum values of the c_t values for $t \in T_i$;
- for each time category i , the minimum value x_i^0 of $F_i(x_i)$, the component of the objective function corresponding to time category i ; it is observed that in all cases, the resulting 5 values do not meet the order constraints imposed in ECSP;
- the five components x_1^*, \dots, x_5^* of the optimum solution \mathbf{x}^* to the ECSP problem, which can be observed to satisfy the order constraints $x_i \leq x_{i+1}$ ($i = 1, \dots, 4$).

Table 2 displays the computational results obtained with scaling_PAV (first line of the table labelled 'sPAV') and OPTIM_SP (in the following lines for values of a ranging from 1 to 8) for the various instances corresponding to the data sets D_1 to D_5 . For each instance, the column labelled '#evalF' provides the number of calls to the A_SEARCH procedure; the column labelled 'Total_N+' provides the the evaluation of the resulting total number of arithmetic operations as measured using the $N_i^+(x_i)$ values; the column labelled 'cpu' indicates the resulting total cpu time (in milliseconds).

All tests reported have been carried out using Python 3.6.8 on an environment of 8-thread quad-core processor with 16 GB RAM and 2.8 GHz CPU running Windows 10 (64 bits).

The results shown in Table 2 suggest the following comments:

- in all cases but D_5 , the OPTIM_SP algorithm significantly outperforms Scaling_PAV in terms of CPU time for a chosen between 4 and 8, and the comparison remains favorable in the wider range $2 \leq a \leq 8$. For the data set D_5 , conclusions are less clear, but one can observe that there are still two values of a for which OPTIM_SP outperforms Scaling_PAV.
- in most cases, it can be observed that there is a fairly wide range of values of a for which OPTIM_SP achieves improved efficiency as compared with one of the

		D_1			D_2		
a	#evalF	Total_N ⁺	cpu	#evalF	Total_N ⁺	cpu	
sPAV	130	113117	44	110	62766	28	
1	192	103498	50	253	72739	34	
2	174	39913	25	256	29724	15	
3	172	18923	10	234	14463	8	
4	167	11286	6	246	12082	8	
5	166	7754	5	252	15993	9	
6	183	5929	4	248	13183	8	
7	168	4801	5	257	11984	7	
8	169	4037	5	255	10989	7	

		D_3			D_4		
a	#evalF	Total_N ⁺	cpu	#evalF	Total_N ⁺	cpu	
sPAV	110	101162	38	90	93442	41	
1	162	90373	44	199	77807	37	
2	149	49375	24	171	34229	17	
3	149	24729	11	169	18330	9	
4	144	11882	6	164	9760	5	
5	134	6426	4	162	6597	4	
6	139	4248	3	159	22971	11	
7	146	3434	3	160	20586	11	
8	147	2977	2	174	19567	11	

		D_5		
a	#evalF	Total_N ⁺	cpu	
sPAV	110	15022	7	
1	167	23838	12	
2	162	24938	12	
3	155	14982	8	
4	161	11746	6	
5	157	9481	5	
6	160	24505	13	
7	167	22706	12	
8	167	21914	11	

Table 2: Computational results on the deterministic version of ECSP for the data sets D_1, \dots, D_5

best previously known algorithms for solving problems of minimizing a convex separable function under order constraints. This shows that OPTIM_SP features a good deal of robustness with respect to the a parameter, and that the choice of a particular value for the parameter a is not a critical issue: a practical consequence of this is that a good value for a will be easily obtained on an experimental basis, by observing the behavior of the OPTIM_SP algorithm on a sample of a few typical instances of the problem for a few values of a in the range $(2, 8)$.

- if, for each result shown in Table 2, one computes the ratio $\text{cpu}/\text{TotalN}^+$, it is observed that all values obtained typically range between $5e - 4$ and $7e - 4$, which shows that cpu is, up to small fluctuations, proportional to TotalN^+ . This confirms the relevance of our analysis based on the use of TotalN^+ as a measure of computational effort.

Category	Months	c_i	\bar{c}_i	T_i	$\lfloor \sqrt{T_i} \rfloor$	σ_i
Winter peak hours	Dec-Feb	2269	5536	1488	38	115
Winter full hours	Nov-Mar	2056	6732	8688	93	124
Winter off-peak hours	Nov-Mar	72	6667	11562	107	104
Summer full hours	Apr-Oct	1769	7539	13920	117	144
Summer off-peak hours	Apr-Oct	1152	7189	16902	130	134

Table 3: Information on remainder of STL decomposition applied the dataset D_1

5.2 Results with OPTIM.SP on the robust version of ECSP

This section presents various computational experiments carried out on the robust version of ECSP.

5.2.1 Analysis of uncertainties through STL decomposition

Information on categories and values of the standard deviation (σ_i) of remainder obtained by applying STL decomposition to the data set D_1 are shown in table 3 (similar results would be observed for the other data sets). For example, the winter peak hours cover three months from December to February and $T_i = 1488$ 10-minute periods, and the standard deviation of remainder is $\sigma_i = 116$.

Statistical tests on remainders reveal that they are uncorrelated, have zero means and are normally distributed, then by considering a truncated normal distribution between $-3\sigma_i$ and $3\sigma_i$ will give correct representation of deviation of the consumption. In our experiments, the robust parameter b_i is chosen to be $3 * \sigma_i$ ($r = 3$), and we will use parameter corresponds to $B_i = V_i * b_i$ (with $V_i = \lfloor q * \sqrt{T_i} \rfloor$) for different values of $q \in [0, 0.2, 0.5, 1]$. Notice that for $q = 0$ the deterministic version of the problem is obtained. For the summer off-peak hours, the different values of q considered correspond to the values 0, 26, 65 and 130 for V_i .

5.2.2 Numerical results concerning the robust version for the data set D_1

Table 4 provides comparative information about the costs of various optimal or sub-optimal solutions to both the deterministic version ($q = 0$) and the robust version ($q > 0$) of ECSP for data set D_1 . Table 4 decomposes into three parts:

- the first part of the table concerns the contract currently used by the company (a big factory in the food industry), the corresponding solution (6200, 7000, 7000, 7300, 7300), is called the reference solution; the deterministic cost of this reference solution, as well as its costs in terms of the robust objective function for $q = 0.2, 0.5, 1$ are provided.
- the second part concerns the optimal solutions to the deterministic ECSP (for $q = 0$) and the robust versions of ECSP (for $q = 0.2, 0.5, 1$);
- the third part provides:
 - (a) the deterministic cost 111296 of the optimal robust solution obtained for $q = 0.5$ and $q = 1$, namely: (5881, 6837, 6837, 7341, 7341);
 - (b) for $q = 0.2, 0.5, 1$, the robust objective function values of the optimal deterministic solution (5536, 6479, 6479, 6914, 6914).

The results in Table 4 suggest the following comments:

q	Subscribed Power (SP)	Cost	cpu
0	6200, 7000, 7000, 7300, 7300	112275	
0.2	6200, 7000, 7000, 7300, 7300	114274	
0.5	6200, 7000, 7000, 7300, 7300	114315	
1	6200, 7000, 7000, 7300, 7300	114315	
0	5536, 6479, 6479, 7316, 7316	107576	25
0.2	5881, 6837, 6837, 7287, 7287	113750	2235
0.5	5881, 6837, 6837, 7341, 7341	113771	2404
1	5881, 6837, 6837, 7341, 7341	113771	2755
0	5881, 6837, 6837, 7341, 7341	111296	
0.2	5536, 6479, 6479, 6914, 6914	118212	
0.5	5536, 6479, 6479, 6914, 6914	121448	
1	5536, 6479, 6479, 6914, 6914	123161	

Table 4: Comparing the costs of various solutions to the deterministic and robust versions of ECSP for data set D_1

- (i) the comparison of the costs of the optimal solutions (part 2 of the table) with the costs of the reference solution shows that the latter is suboptimal in all cases: the difference is 4.2% for $q = 0$, 0.4% for $q=0.2$, and 0.5% for $q = 0.5$ and $q = 1$. It is thus observed that the reference solution turns out to be a fairly good approximation (to within 0.5% or so) of the optimal robust solutions for all the (strictly positive) values of q considered.
- (ii) the comparison between the last three lines of part 2 and the last three lines of part 3 shows the benefit provided by the optimal robust solution over the optimal deterministic solution in the presence of uncertainty: when $q = 0.2$, the optimal robust function value is 113750, whereas the robust objective function value for the optimal deterministic solution is 118212, showing that the former leads to an improvement in cost 3.8% over the latter. The same comparison for $q = 0.5$ and $q = 1$ would lead to 6.3% and 7.6% improvement respectively.
- (iii) The comparison between the first line of part 2 of the table and the first line of part 3 shows that the difference between the deterministic objective function value of the optimal robust solution for $q = 0.5$ and $q = 1$, and the optimal deterministic solution value ($111296-107576= 3720$) represents an increase by 3.3%. This value can be interpreted as the price of robustness.
- (iv) the cpu times required for solving the robust versions of ECSP are about 100 times more than for solving the deterministic version, however they typically do not exceed two or three seconds, which is quite acceptable from the point of view of practical applicability.

To conclude, let us point out that, in spite of the fact that the results in Table 4 only concern the instance D_1 , quite similar conclusions would be obtained by analyzing the other data sets D_2, \dots, D_5 .

5.2.3 Computational efficiency of OPTIM_SP on the Robust ECSP

In addition to the results discussed in Section 5.2.2, we provide in Table 5 results showing the computational efficiency of OPTIM_SP applied to the robust version of ECSP, in a form similar to Table 2. In these experiments, the value $q = 0.5$ has been taken. The main comments suggested by the results shown in Table 5 are the following:

- since the computational effort required to calculate the value of each component of the objective function of the robust problem is significantly bigger than for the deterministic case, it is all but surprising to observe significantly increased cpu values. On average, they can be seen to be multiplied by a factor in the range 50-100.
- concerning the influence of varying the value of the a parameter in the range $a = 1$ to $a = 8$, a behavior similar to the one observed for the deterministic case can be noticed: for all instances (except D_5) OPTIM.SP outperforms Scaling_PAV for a wide range of values for a .
- the relationship between cpu and the Total_N⁺ measure of computational effort, which was almost linear in the deterministic case, is more intricate in the case of the robust ECSP; this is mainly due to the significant overhead induced by the need of repeatedly solving the MCKP discussed in Section 4.3.

		D1			D2		
a	#evalF	Total_N ⁺	cpu	#evalF	Total_N ⁺	cpu	
sPAV	130	5733875	3815	110	2781354	2363	
1	198	4786268	4058	213	3537792	3271	
2	173	1398527	2046	183	1067315	1803	
3	181	490332	1534	180	461635	1578	
4	173	223949	1401	176	302938	1406	
5	170	130601	1254	176	178908	1362	
6	182	82507	1327	196	324828	1559	
7	175	55507	1267	185	274221	1457	
8	172	50018	1255	183	243549	1472	

		D3			D4		
a	#evalF	Total_N ⁺	cpu	#evalF	Total_N ⁺	cpu	
sPAV	110	4399366	3024	100	3949268	2834	
1	163	4121953	3512	201	3074664	2971	
2	148	1900657	2237	184	949165	1832	
3	152	570448	1413	174	295506	1387	
4	151	202383	1282	174	129488	1325	
5	143	106005	1099	172	76850	1279	
6	144	64673	1076	175	56555	1290	
7	139	45094	1036	177	916144	1747	
8	145	39596	1065	185	864616	1757	

		D5		
a	#evalF	Total_N ⁺	cpu	
sPAV	110	539235	1115	
1	171	825202	1695	
2	160	171727	1273	
3	156	643125	1472	
4	156	432818	1388	
5	161	355929	1398	
6	166	1192940	1930	
7	163	1078631	1810	
8	167	997721	1773	

Table 5: Computational results on the robust version of ECSP ($q = 0.5$) for the data sets D_1, \dots, D_5 .

6 Conclusion and future works

This paper proposes an efficient polynomial-time algorithm based on an active constraint set approach for the electricity contract selection problem (ECSP). This problem is formulated as minimizing a convex separable function subject to total order constraints. The computational results obtained show that the algorithm features improved CPU time as compared with Scaling_PAV [2], one of the best existing methods for this problem.

The robust version of the problem has also been investigated. The construction of an uncertainty set representing realistic scenarios of deviation between realized consumption and forecasted values has been proposed, based on the so-called Seasonal and Trend decomposition method (STL). It has been shown that the robust version of ECSP can be efficiently solved using the same algorithm as for the deterministic version. However, taking uncertainty into account requires computing worst-case values of the objective function. The latter problem is reduced to solving a multiple-choice knapsack problem, which is polynomially solvable.

An interesting direction for future research will be to generalize the approach of robustness. In the present paper, it has been assumed that each consumption of a given category i can only deviate by a value b_i which is the same for all the time periods in the category. The next step is to have more detailed evaluation of the σ values within each category, the standard deviation may be different from one month to the next (e.g. the consumption of summer off-peak hours in June may be quite different than the one in September). Another interesting subject for future investigation would be to use more accurate stochastic models, possibly capturing statistical dependence from one time instant to the next to better represent uncertainties.

Acknowledgements This work was supported by Energisme SAS in supplying data and validating the experiments.

References

1. Ahuja, R.K., Hochbaum, D.S., Orlin, J.B.: Solving the Convex Cost Integer Dual Network Flow Problem. *Management Science* **49**(7), 950–964 (2003)
2. Ahuja, R.K., Orlin, J.B.: A Fast Scaling Algorithm for Minimizing Separable Convex Functions Subject to Chain Constraints. *Operations Research* **49**(5), 784–789 (2001)
3. Barbier, T.: Modélisation de la consommation électrique à partir de grandes masses de données pour la simulation des alternatives énergétiques du futur. Ph.D. thesis, Université de recherche Paris Sciences et Lettres PSL Research University (2017)
4. Bertsimas, D., Sim, M.: The Price of Robustness. *Operations Research* **52**(1), 35–53 (2004)
5. Best, M., Chakravarti, N., Ubhaya, V.: Minimizing separable convex functions subject to simple chain constraints. *SIAM Journal on Optimization* **10**, 658–672 (2000)

6. Chen, C.Y., Liao, C.J.: A linear programming approach to the electricity contract capacity problem. *Applied Mathematical Modelling* **35**(8), 4077–4082 (2011)
7. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: STL: a seasonal-trend decomposition. *Journal of official statistics* **6**(1), 3–73 (1990)
8. Durand, J.B., Bozzi, L., Celeux, G., Derquenne, C.: Analyse de courbes de consommation électrique par chaînes de Markov cachées. *Revue de statistique appliquée* **52**(4), 71–91 (2003)
9. Hochbaum, D.S.: Complexity and algorithms for convex network optimization and other nonlinear problems. *4OR* **3**(3), 171–216 (2005)
10. Issi, F., Kaplan, O.: The Determination of Load Profiles and Power Consumptions of Home Appliances. *Energies* **11**(3), 607 (2018)
11. Karzanov, A.V., McCormick, S.T.: Polynomial Methods for Separable Convex Optimization in Unimodular Linear Spaces with Applications. *SIAM Journal on Computing* **26**(4), 1245–1275 (1997)
12. Kavousian, A., Rajagopal, R., Fischer, M.: Determinants of residential electricity consumption: Using smart meter data to examine the effect of climate, building characteristics, appliance stock, and occupants’ behavior. *Energy* **55**, 184–194 (2013)
13. Minoux, M.: A polynomial algorithm for minimum quadratic cost flow problems. *European Journal of Operational Research* **18**(3), 377–387 (1984)
14. Minoux, M.: Solving integer minimum cost flows with separable convex cost objective polynomially. In: G. Gallo, C. Sandi (eds.) *Netflow at Pisa, Mathematical Programming Studies*, pp. 237–239. Springer, Berlin, Heidelberg (1986)
15. Pipattanasomporn, M., Kuzlu, M., Rahman, S., Teklu, Y.: Load Profiles of Selected Major Household Appliances and Their Demand Response Opportunities. *IEEE Transactions on Smart Grid* **5**(2), 742–750 (2014)
16. Rodoplu, M., Arbaoui, T., Yalaoui, A.: Energy Contract Optimization for the Single Item Lot Sizing Problem in a Flow-Shop Configuration and Multiple Energy Sources. *IFAC-PapersOnLine* **51**(11), 1089–1094 (2018)
17. Salerno, V., Rabbeni, G.: An Extreme Learning Machine Approach to Effective Energy Disaggregation. *Electronics* **7**(10), 235 (2018)
18. Sæle, H., Bremdal, B.A., Engan, T.T., Kristoffersen, V., Foosnaes, J.A., Nordal, T.E., Sletner, J.M.: Subscribed power-testing new power based network tariffs stimulating for demand response. *CIREN 2015, Paper 1085* (2015)
19. Tatsa, S.: Modélisation et prévision de la consommation horaire d’électricité au Québec: Comparaison de méthodes de séries temporelles. Master’s thesis, Université Laval (2014)
20. Zemel, E.: An $O(n)$ Algorithm for the Linear Multiple Choice Knapsack Problem and Related Problems. *Inf. Process. Lett.* **18**(3), 123–128 (1984)
21. Zhou, X., Yan, D., Hong, T., Ren, X.: Data analysis and stochastic modeling of lighting energy use in large office buildings in China. *Energy and Buildings* **86**, 275–287 (2015)

Appendix

We state in this appendix the reformulation of \mathcal{P} as a separable convex cost network flow problem.

The graph to be considered contains K nodes numbered $1, 2, \dots, K$ and a set of $2K - 1$ arcs, decomposed into K 'primary arcs' and $K - 1$ 'secondary arcs'.

- the set of primary arcs contains the $K - 1$ arcs of the form $(i, i + 1)$ for $i = 1, \dots, K - 1$, together with a 'return arc' $(K, 1)$. The flow value x_i on each of these arcs (including x_K for the return arc) has to meet the bound constraints $0 \leq x_i \leq C$ (and $0 \leq x_K \leq C$) and the corresponding cost function is F_i (F_K for the return arc);
- the set of secondary arcs is composed of $K - 1$ arcs of the form $(1, j)$ for $j = 2, \dots, K$. The flow value s_j on each of these arcs has to meet the bound constraints $0 \leq s_j \leq C$, and the associated cost function is identically 0.

Problem \mathcal{P} then reduces to determining a minimum cost circulation on the above graph. We observe that the flow values s_j on the secondary arcs play the role of slack variables for the constraints $x_{j-1} \leq x_j$ for $j = 2, \dots, K$.