Efficient optimisation of wind power under acoustic constraints

Baldwin Dumortier, Emmanuel Vincent, Madalina Deaconu, Patrice Cornu

Efficient optimisation of wind power under acoustic constraints

Baldwin Dumortier, Emmanuel Vincent, *Senior Member, IEEE,* Madalina Deaconu and Patrice Cornu

***Abstract*—Great attention is currently paid to protecting residential areas from the noise pollution due to wind turbines. A family of acoustic constraints was developed to assess it. Maximizing electricity production under these constraints is difficult as they rapidly evolve with weather conditions and background noise. Today, this problem is addressed by computing a curtailment plan involving fixed operating modes, but the chosen modes are often suboptimal. In this article, we show that this problem can be expressed as a non-linear knapsack problem and we solve it using an efficient branch-and-bound (B&B) algorithm that converges asymptotically to the global optimum. The algorithm is initialised with a greedy heuristic that iteratively downgrades the turbines with the best acoustical to electricity loss ratio. The solution is then refined using a depth-first search strategy and a bounding stage based on a continuous relaxation problem solved with an adapted gradient algorithm. The results are evaluated using data from 28 real wind farms.**

***Keywords*—Wind turbines, knapsack problem, branch and bound, acoustic control.**

I. Introduction

Renewable energies have gained popularity all over the world and the use of solar and wind energy has experienced an exponential growth for the last ten years. The increased usage and the stochastic nature of wind energy have brought a large panel of research problems including storage-conversion, grid integration, dynamic control of the turbine parameters, power curve estimation [18] and wind prediction [14], [8].

However, up to now, acoustic constraints have never really been covered from an optimisation perspective. Yet, they are the constraints that limit most the deployment of new farms and the electricity production of existing farms today. Currently, acoustic studies are required to ensure tranquillity of the inhabitants around the farms. For this purpose, acoustic measurements are made over the duration of a couple of weeks. While measuring, the wind turbines are periodically stopped in order to evaluate the difference between the ambient noise level (with the turbines on) and the background noise level (with the turbines off). The data are then classified into homogeneous classes that correspond to different ranges of background noise and meteorological data. A curtailment plan is then computed for each homogeneous class and sent to the wind farm owner in order to set it up in the local control system of the turbines (SCADA).

In this paper, we study the problem of maximizing electricity production under acoustic constraints from a rigorous optimisation perspective. We show that it can be seen as a *convex integer non-linear programming problem* [11], [1], [5], or a *non-linear knapsack problem* [16], [19]. This designates an extension of a famous optimisation problem, the knapsack problem, in a non-linear context. It is an NP-complete class of problems [1] and different algorithms have been used over the last decades to address it such as *branch and bound* (B&B), *outer-approximation* [10], *generalized benders decomposition* [13], *extended cutting plane* [27], *LP/NLP based Branch-and-Bound* [22] and hybrid methods.

Here, we propose a B&B approach which is appropriate to solve discrete optimisation problems with a finite number of admissible solutions [26]. The algorithm builds a sequence of solutions that monotonously converges to an optimal solution. To achieve this, the B&B algorithm uses a smart browsing of the solution tree based on the ”divide and conquer” principle.

This algorithm is an exact approach in the sense that it is theoretically optimal when the used research tree is fully browsed. However, the computational efficiency strongly depends on the implementation choices made in the ”branching stage” and in the ”bounding stage” of the algorithm. For that reason, B&B algorithms require specific choices depending on the problem. Theoretical contributions and comparisons with other algorithms have been described in [23], [7] [21] [12], [19], [16], [3], [2]. B&B algorithms have been used for a wide range of resource allocation problems. Applications in the wind energy field have been proposed in [25], [6], [15], [20].

For our application, we chose to use two algorithms respectively to initialise and bound the nodes of the trees. This is inspired from an algorithm described in [26, p.201]. However, the originality of our approach comes from a greedy algorithm specifically designed for our problem and a gradient-based approach to solve the continuous relaxation with a piecewise-linear objective function. The algorithm was tested on real data obtained from measurement campaigns conducted by VENATHEC SAS, a company specialised in acoustic engineering.

This paper is organised as follows. The general context and assumptions are presented in Section II. Then the reformulation is given at Section III. To address this problem, an adapted algorithm is introduced in Section IV. Finally, the experiments

This work was done with the collaboration and the support of VENATHEC SAS. In particular, the authors thank Jeremy Schild from Venathec SAS for his implication.

This work was also supported by CPER MISN TALC and region Lorraine.

B. Dumortier and E. Vincent are with Inria, Villers-lès-Nancy, F-54600, France; CNRS, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France and Université de Lorraine, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France (e-mail: baldwin.dumortier@inria.fr, emmanuel.vincent@inria.fr).

M. Deaconu is with Inria, Villers-lès-Nancy, F-54600, France and Université de Lorraine, CNRS, Institut Elie Cartan de Lorraine - UMR 7502, Vandœuvre-lès-Nancy, F-54506, France (e-mail: madalina.deaconu@inria.fr).

P. Cornu is with VENATHEC SAS, 23 Boulevard de l'Europe, 54500, Vandœuvre-lès-Nancy, France (e-mail: p.cornu@venathec.com).

Country/institution	Day noise limit (dBA)		night noise limit (dBA)	
	$S_{e,o}$	$S_{s,o}$	$S_{e,o}$	$S_{s,o}$
WHO guidelines	$+\infty$	50	$+\infty$	45
Australia	5	35	5	35
France	5	35	3	35
Germany	$+\infty$	60 or 55 or 50	$+\infty$	45 or 40 or 35
UK	5	35 or 40	5	35 or 40
USA	$+\infty$	40	$+\infty$	40

TABLE I. WIND FARM NOISE REGULATION ACROSS COUNTRIES.

and the results are presented in Section V. We conclude in Section VI.

II. ASSUMPTIONS AND INITIAL FORMULATION

In this section, we give an overview of the main assumptions, introducing the logic behind the legal acoustic annoyance criteria commonly used in the world and the usual problem formulation.

Different countries in the world have set a maximum ambient noise level around wind farms. However, the ambient level itself is not always a good measure of discomfort. Indeed, it appears that the human ear is rather sensitive to the ratio of the energy of the annoying noise signals (noise of the wind turbines called *particular noise* in this article) over the energy of the other acoustic signals (called *residual noise* level). One can easily imagine two different situations with about the same ambient level but with very different ratios of particular noise and residual noise levels. For that reason, the residual noise level is often directly or indirectly taken into account in the regulation. The ratio of ambient noise energy over the residual noise energy is called *acoustic emergence* and is defined as follows, for each a given measurement point j around the farm:

$$e_j = b_j - r_j \tag{1}$$

where b_j designates the total ambient noise power when the turbines are turned on and r_j designates the residual noise power when the turbines are turned off.

The acoustic emergence is directly considered in some countries like Australia or France by measuring the residual noise level when the the turbines are turned off. In some other countries, like Germany, the acoustic emergence is indirectly used by considering different maximum admissible values of the ambient noise level depending on the type of residential area. In any case, the legal criterion can be always formulated as follows :

$$\forall j \in \{1, .., J\},\ e_j < S_{e,o} \text{ or } b_j < S_{s,o} \tag{2}$$

where $S_{e,o}$ is the emergence threshold, and $S_{s,o}$ is the ambient noise threshold. Here J denotes the number of measurement points considered and j one of these measurement points. Table I presents a summary of noise thresholds for several countries and institutions based on a review written by Schild and Chavand [24] working in the field of acoustics in France. The data are in terms of $S_{e,o}$ and $S_{s,o}$.

The criterion 2 cannot be applied in real time since the particular noise and the residual noise can't be measured directly and constantly evolve. For this reason, homogeneous classes are defined by environment conditions that allow to gather measurements that are supposed to happen in similar conditions. The particular noise level and the residual noise level are considered to be constant in each class and for each measurement point j. For instance, in France the homogeneous classes are standardised and defined by the subclasses $\mathcal{P} = \{day, night\}$, $\mathcal{D} = \{d_1, d_2, d_3, d_4, d_5, d_6\}$ and $\mathcal{M} = \{m_3, m_4, ..., m_{10}\}$, that respectively designates partitions of the period of the day, of the wind directions (60 angular sectors) and wind speed (interval of wind speed centred on an integer value between 3 and 10 m/s and of width 1 m/s). The set of homogeneous classes can be defined as the Cartesian product of all subclasses and written as $\mathcal{O} = \mathcal{P} \times \mathcal{D} \times \mathcal{M}$.

More generally, we consider $\mathcal{O}$ to be the set of homogeneous classes defined by the Cartesian product of the subclasses used in a given country.

The legal criterion is then computed off-line for each homogeneous class in order to find out which optimal operating modes fulfil the constraints for each homogeneous class and wind turbine. Optimal operating modes are then searched under the assumptions that the uncertainties on the acoustic and electric parameters of the system are considered negligible in each homogeneous class. The problem of finding the optimal operating modes can be expressed as follows for each homogeneous class $o \in \mathcal{O}$:

$$(P) \begin{cases} \max_{\mathbf{c} \in \mathcal{C}^I} \sum_{i=1}^{I} \mathfrak{p}_v(c_i) \\ \quad \forall j \in \{1, ..., J\} \\ \quad b_j - r_j \leq S_{e,o} \\ \quad \text{or } b_j \leq S_{s,o} \end{cases}$$

where c_i is an integer command variable for the i-th wind turbine and I the number of wind turbines. The possible commands are given by the set $\mathcal{C} = 1, ..C$ defining all the manufacturers operating modes sorted by decreasing order of production and acoustic power. $\mathfrak{p}_v$ is the function returning the production power for the current value of wind speed v.

This problem is often solved in a heuristic way by acoustic engineers today. This results in suboptimal production power. In the following we propose to solve it using a principled optimisation approach.

III. REFORMULATION AS A NON LINEAR CONVEX KNAPSACK PROBLEM

From now on, we consider a simple propagation model which is currently used in the field of acoustics. We denote by x_i the noise emitted by wind turbine i and by $\mathbf{A}_t = [a_{i,j}]$ the propagation matrix that defines the acoustical energy loss from each turbine i to each measurement point j. The acoustic parameters are basically in decibels but we use a wavy line ($\sim$) to denote the parameters expressed in linear scale (*i.e* $\tilde{y} = 10^{\frac{y}{10}}$ for any acoustic parameter y) Now we can write the following relationship:

$$\forall j, \tilde{b}_j = \sum_{i=1}^{I} \tilde{a}_{i,j}\tilde{x}_i + \tilde{r}_j. \quad (3)$$

In order to obtain a more usual formulation for the optimisation problem, we use a variable change and the linear scale. The constraints in (2) become:

$$\begin{cases} \forall j \in \{1, ..., J\} \\ \sum_{i=1}^{I} \tilde{a}_{i,j}\tilde{x}_i \leq 10^{\frac{S_{e,o}+r_j}{10}} - \tilde{r}_j \\ \text{or } \sum_{i=1}^{I} \tilde{a}_{i,j}\tilde{x}_i \leq 10^{\frac{S_{s,o}}{10}} - \tilde{r}_j \end{cases} \quad (4)$$

Now if we set $\mathbf{q} = [q_j]$ as follows:

$$\forall j \in \{1, ..., J\}, q_j = \max(10^{\frac{S_{e,o}+r_j}{10}} - \tilde{r}_j, 10^{\frac{S_{s,o}}{10}} - \tilde{r}_j) \quad (5)$$

then, we have $\forall j \in \{1, ..., J\} \quad \sum_{i=1}^{I} \tilde{a}_{i,j}\tilde{x}_i \leq q_j$ that we write in the following:

$$\tilde{\mathbf{A}}^T \tilde{\mathbf{x}} \leq \mathbf{q} \quad (6)$$

We thus obtain a linear formulation for the acoustical constraints. Still, the dependencies between the constraints and the objective function are not explicit. We then propose to consider $\tilde{\mathbf{x}}$ as the new choice variable instead of $\mathbf{c}$. This allows us to reformulate the problem without loss of generality as a bijection between the operating modes $\mathbf{c}$ and the acoustic emission $\tilde{\mathbf{x}}$. To do this, we define the function $\mathfrak{f}$ as follows:

$$\begin{aligned} \mathfrak{f} : \tilde{\mathfrak{x}}_v(\mathcal{C}) &\to \mathbb{R} \\ \tilde{x}_i &\mapsto \mathfrak{p}_v(\tilde{\mathfrak{x}}_v^{-1}(\tilde{x}_i)). \end{aligned} \quad (7)$$

where $\tilde{\mathfrak{x}}_v$ is the function returning the acoustic power for the current value of wind speed v. In other words, $\mathfrak{f}$ is the function that relates the acoustic power in $\mathrm{W/m^2}$ to the electricity production power of each wind turbine. We also define the total production power $\mathfrak{F}$ as $\forall \tilde{\mathbf{x}} \in \tilde{\mathfrak{x}}_v(\mathcal{C})^I, \mathfrak{F}(\tilde{\mathbf{x}}) = \sum_{i=1}^{I} \mathfrak{f}(\tilde{x}_i)$. Then we get:

$$(P) \Leftrightarrow \begin{cases} \max_{\tilde{\mathbf{x}} \in \tilde{\mathfrak{x}}_v(\mathcal{C})^I} \mathfrak{F}(\tilde{\mathbf{x}}) \\ \tilde{\mathbf{A}}^T \tilde{\mathbf{x}} \leq \mathbf{q} \end{cases} \quad (8)$$

This last formulation is the most interesting as it defines a theoretical context for the given problem which allows us to construct an optimal algorithm. We also experimentally verified that $\mathfrak{f}$ is non linear and concave, and therefore that the optimisation problem is convex (see Figure 2). It should also be noticed that with this formulation, the problem can be solved with the same algorithm for every country: only the value of $\mathbf{q}$ will change.

IV. Proposed optimisation algorithm

B&B algorithms are a class of algorithms based on an intuitive principle. The admissible solutions[1] are stored in a search tree where each node defines a subset of all the admissible solutions. Each node is labelled with an upper bound that indicates the maximal possible value of the objective function for the subset of solutions that it represents. Then, the tree is iteratively browsed by using the following pruning rule.

> Pruning rule:
> For a given admissible solution $\mathbf{c}$ of electricity production p, it is impossible to get a better solution from the nodes whose upper bound is smaller than p.

This property is the core of the algorithm. All the branches of the tree are either browsed or pruned and the solution is optimal when the computation terminates [26, p.135].

The main difficulties are to bound and to browse the tree efficiently. Coarse bounding of the nodes induces more useless browsing of the tree and fine bounding induces long computation time of the bounding stage. Moreover, a good initialisation is also essential to reduce computation time.

A. Proposed Branch and bound algorithm

In our algorithm, the whole set of admissible solutions is stored in the research tree in the following way: each new node defines a new operating mode for a given wind turbine compared to its father node. Fig. 1 gives an example of a typical tree.

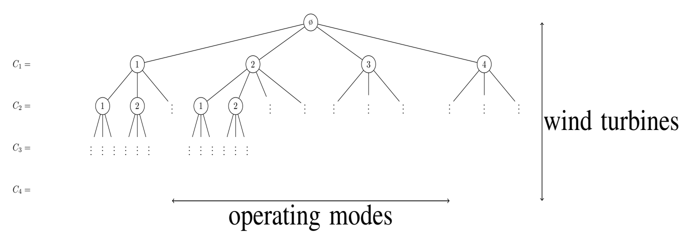

Fig. 1. Example tree for 4 wind turbines and 4 operating modes.

The tree is browsed using a depth-first strategy in the following order:

- In depth, the wind turbines are sorted by order of decreasing acoustical impact considering, for a given wind turbine i, the value $\max_j a_{i,j}$.
- In width, the nodes are sorted by decreasing value of upper bounds. The computation is explained in Section IV-C.

The general pseudo-code is given in Algorithm 1.

B. The greedy heuristic

We propose a greedy heuristic to initialise the first solution of the sequence built by the B&B algorithm. It allows us to get

[1]those who satisfy the acoustic constraints

Algorithm 1 *Branch and Bound* algorithm

```
function BRANCHANDBOUND
    Initialisation of the root of tree;
    current_node points to tree;
    Initialisation of current_solution with an
    admissible solution;
    while root of tree is not bounded OR tree is not empty do
        if current_node is a leaf then
            Write the solution in current_solution;
            Delete current_node and go back to its
            father;
        else if the sons of current_node have not been built yet then
            Build the sons of current_node;
            Compute the bounds of the sons of
            current_node;
            Sort the sons by decreasing value of bounds;
        end if
        if power production of current_solution < bound of the first son of current_node then
            Assign for current_node its son;
        else
            Delete current_node and go back to
            its father;
        end if
    end while
end function
```

close to the optimal solution with a computation time smaller than 1 s. The principle is to start from a "full-power" strategy and to downgrade one wind turbine at a time by choosing the wind turbine i that has the best "loss of acoustical excess" (of emergence or ambient noise) per "loss of production power" ratio. The ratio for the i-th wind turbine can be written as:

$$\text{criterion}(i) = \frac{\|\tilde{\mathbf{d}}^{+}_{i_{\text{down}}} - \tilde{\mathbf{d}}^{+}\|_1}{\|\mathfrak{F}(\tilde{\mathbf{x}}_{i_{\text{down}}}) - \mathfrak{F}(\tilde{\mathbf{x}})\|_1} \tag{9}$$

where $\tilde{\mathbf{d}}^{+}$ is the $J \times 1$ vector of the positive parts of the entries of $\tilde{\mathbf{d}}$, the vector of acoustical excesses:

$$\tilde{\mathbf{d}} = \tilde{\mathbf{A}}^T \tilde{\mathbf{x}} - \mathbf{q} \tag{10}$$

$\tilde{\mathbf{d}}^{+}_{i_{\text{down}}}$ and $\tilde{\mathbf{x}}_{i_{\text{down}}}$ are respectively the vector of positive parts of the acoustic excesses and the vector of acoustic powers after downgrading turbine i from c_i to $c_i + 1$. $\|.\|_1$ is the usual 1-norm defined by $\|\boldsymbol{x}\|_1 = |x_1| + \ldots + |x_n|$ with $\boldsymbol{x}$ any $n \times 1$ vector. When no excess is observed anymore (all the entries of $\tilde{\mathbf{d}}$ are negative), the algorithm terminates. The pseudo code of the algorithm is given by Algorithm 2.

C. Continuous relaxation

For each node, it is possible to define a continuous problem that allows us to determine an upper bound for the subset of solutions that the node represents. We consider continuous operating modes for the remaining wind turbines that are not

Algorithm 2 *Greedy* algorithm

```
function GREEDY
    initialise c with ∀i, c(i) = 1
    while d̃⁺ is not the null vector do
        for i between 1 and n do
            criterion(i) == ‖d̃⁺_{i_down} − d̃⁺‖₁ / ‖𝔉(x̃_{i_down}) − 𝔉(x̃)‖₁
        end for
        i_chosen = arg max(criterion);
        c(i_chosen) = c(i_chosen) + 1;
    end while
end function
```

determined yet. The solution to the continuous problem is always greater or equal to the discrete solution and therefore constitutes a bound for the node. This property is intuitive and easy to prove. Indeed, the continuity of the function and the compactness of the two sets of solutions (continuous and discrete) imply that an upper bound exists for each set. The order of those bounds can then be deduced from the preservation of inclusion by a function.

From now on, we suppose that the remaining wind turbines below a given node can have an infinity of operating modes whose values of acoustic emission and production are included between those of the closest discrete operating modes. If we denote by I_0 the number of determined wind turbines of a given node, new acoustical constraints can be deduced and we obtain a continuous problem for the $I - I_0$ remaining turbines. We consider $\tilde{\mathbf{x}}_{\text{det}}$, the $I \times 1$ emission power vector of the determined wind turbines and $\tilde{\mathbf{x}}_{\text{free}}$, the $I \times 1$ emission power vector of the undetermined wind turbines:

$$\tilde{\mathbf{x}}_{\text{det}} = \begin{pmatrix} \tilde{\mathfrak{x}}_v(c_1) \\ \tilde{\mathfrak{x}}_v(c_2) \\ \ldots \\ \ldots \\ \tilde{\mathfrak{x}}_v(c_{I_0}) \\ 0 \\ 0 \\ \ldots \\ 0 \end{pmatrix} \quad \tilde{\mathbf{x}}_{\text{free}} = \begin{pmatrix} 0 \\ 0 \\ \ldots \\ 0 \\ \tilde{\mathfrak{x}}_v(c_{I_0+1}) \\ \tilde{\mathfrak{x}}_v(c_{I_0+2}) \\ \ldots \\ \ldots \\ \tilde{\mathfrak{x}}_v(c_I) \end{pmatrix}. \tag{11}$$

We can set the following continuous problem:

$$(P_{\text{cont}_{\text{I}_0}}) \begin{cases} \max_{(\tilde{x}_{I_0+1},\ldots,\tilde{x}_I) \in [0, \tilde{\mathfrak{x}}_v(1)]^{(I-I_0)}} \sum_{i=I_0+1}^{I} \mathfrak{f}(\tilde{x}_i) \\ \tilde{\mathbf{A}}^T \tilde{\mathbf{x}}_{\text{free}} \leq \mathbf{q} - \tilde{\mathbf{A}} \tilde{\mathbf{x}}_{\text{det}}. \end{cases} \tag{12}$$

To extend the function $\mathfrak{f}$, any continuous and concave function can be chosen without any consequence on the optimality of the discrete algorithm. We chose to extend $\mathfrak{f}$ with linear interpolations between the discrete operating modes. The reason is that the upper bound should be as small as possible to reduce computation time and a linear interpolation gives the smallest concave continuous function. An example of extended function $\mathfrak{f}$ is given in Figure 2.

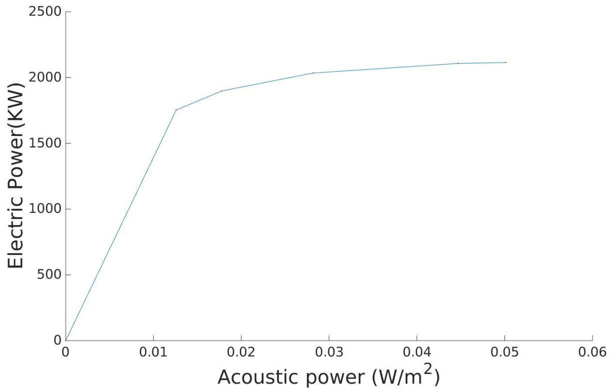

Fig. 2. Example of extended $\mathfrak{f}$ function

Notice that the continuous relaxation is a convex problem. It can be solved using any algorithm for constrained convex optimisation (projected gradient, Lagrange multipliers, ...) [4]. We propose below a projected gradient algorithm adapted to the piecewise nature of the objective function.

D. Projected gradient algorithm

Projected gradient is a classical algorithm to solve continuous optimisation problems with a differentiable objective function. It converges to a global optimum for a convex problem when the objective function is smooth enough. A variation to the classic algorithm is proposed in this paper in order to handle the specificities of the problem. The algorithm is adapted to the linear constraints of the problem $\tilde{\mathbf{A}}^T\tilde{\mathbf{x}} \leq \mathbf{q}$ and to the objective function which is a concave piecewise linear function $\mathfrak{F}(\tilde{\mathbf{x}}) = \sum_i \mathfrak{f}(\tilde{x}_i)$ where $\mathfrak{f}$ is piecewise linear.

Let us start by introducing the algorithm on an example. We consider a problem with 2 wind turbines ($I = 2$), 2 measurement points ($J = 2$) and five possible operating modes ($C = 5$) with the following values:

- We consider the following values of acoustical power for the operating modes: $\tilde{\mathfrak{x}}_v(\mathcal{C}) = \{0, 1, 2, 3, 4\}$ and $\mathfrak{p}_v(\mathcal{C}) = \{0, 8, 12, 14, 15\}$. We set $\tilde{\mathfrak{x}}_v(1) = 4$, $\tilde{\mathfrak{x}}_v(2) = 3$, $\tilde{\mathfrak{x}}_v(3) = 2$, $\tilde{\mathfrak{x}}_v(4) = 1$, $\tilde{\mathfrak{x}}_v(5) = 0$. We also consider for the power production: $\mathfrak{p}_v(1) = 15$, $\mathfrak{p}_v(2) = 14$, $\mathfrak{p}_v(3) = 12$, $\mathfrak{p}_v(4) = 8$, $\mathfrak{p}_v(5) = 0$.
- We extend the operating modes to continuous values of $\tilde{x}_1$ and $\tilde{x}_2$ in the interval $[\tilde{\mathfrak{x}}_v(C), \tilde{\mathfrak{x}}(1)] = [0, 4]$ for the acoustic power. Then, we extend the function $\mathfrak{f}$ with piecewise linear interpolation:

$$\mathfrak{f} : [\tilde{\mathfrak{x}}_v(C), \tilde{\mathfrak{x}}(1)] \rightarrow [\tilde{\mathfrak{p}}_v(C), \tilde{\mathfrak{p}}(1)]$$
$$\tilde{x} \mapsto \frac{\mathfrak{p}_v(k) - \mathfrak{p}_v(k+1)}{\tilde{\mathfrak{x}}_v(k) - \tilde{\mathfrak{x}}_v(k+1)} \times (\tilde{x} - \tilde{\mathfrak{x}}_v(k+1)) + \mathfrak{p}_v(k+1)$$

- We set $\tilde{\mathbf{A}}^T = \begin{bmatrix} 3 & 1 \\ \frac{1}{8} & 1 \end{bmatrix}$ and $\mathbf{q} = \begin{bmatrix} 11 \\ 5 \end{bmatrix}$.

Fig.3 gives a graphical representation of the gradient algorithm on the example with 2 wind turbines. The solution space is a polyhedron, whose border represents the legal constraints (ambient and emergence constraints) and the physical constraints (the emission power is always positive and reaches a maximal emission corresponding to full power operation). They are all indifferently referred to *constraints* in the rest of this article. The polyhedron is partitioned into square subspaces corresponding to the different modes. In each subspace, the gradient is constant because of the piecewise-linearity of the objective function. The hyperplanes between such subspaces are called *gradient transitions* and define the points where a brutal gradient change is observed. Moreover, we distinguish the **active constraints** (respectively active transitions) which are the constraints applying to the vector obtained and the previous iteration, from the *hypothesized constraints* (respectively hypothesized transitions) that are the constraints on the tested vectors and from the *reached constraints* (new reached transitions) that are the additional constraints in the current iteration (see Fig 4).

Now, here is a short description of the execution based on Fig 3:

- We start from $\tilde{\mathbf{x}}^{(0)} = (0, 0)$ and we move in the polyhedron following the gradient until a gradient transition or a constraint is reached.
- Here, we have stopped in $\tilde{\mathbf{x}}^{(1)}$ where we reached a gradient transition. For now, the execution is simple as there are no active constraints, the new gradient is computed and followed.
- In $\tilde{\mathbf{x}}^{(1)}$ and $\tilde{\mathbf{x}}^{(2)}$, we have the same situation. We compute the new gradient and then we follow the gradient until $\tilde{\mathbf{x}}^{(3)}$
- This time, in ($\tilde{\mathbf{x}}^{(3)}$) a first constraint is reached. However, we have not find the optimum yet. We compute the projection of the gradient on the constraints that is followed until a new constraint or transition.
- In $\tilde{\mathbf{x}}^{(4)}$ a new transition is reached. The new gradient is computed and projected once again on the active constraint.
- Then we reach a new constraint in $\tilde{\mathbf{x}}^{(5)}$. The algorithm stops as it is not possible to follow the projection of the gradient on any of the constraints or on the intersection of the constraints without violating any of the constraints.

More generally, we developed a gradient algorithm that moves inside an $I - I_0$ dimensional polyhedron. In this respect, the constraints and the gradient transitions are delimited by hyperplanes of dimension $I - I_0 - 1$ and each time a new constraint or a new transition is reached all the intersections of combinations of hyperplanes must be tested in theory.

To avoid an exploding computational cost, we use the following heuristic : if k constraints/gradient transitions are reached at a given step, only the intersection of all the constraints/transitions and of the combinations of $k - 1$ constraints/transitions are tested. We found this heuristic to be efficient and optimal in a huge majority of the test cases (see Section V). To describe the general algorithm, we need to introduce new notations:

- $\mathcal{X}_{\text{poss}} = [0, \tilde{\mathfrak{x}}(1)]^I$ set of possible continuous strategies.

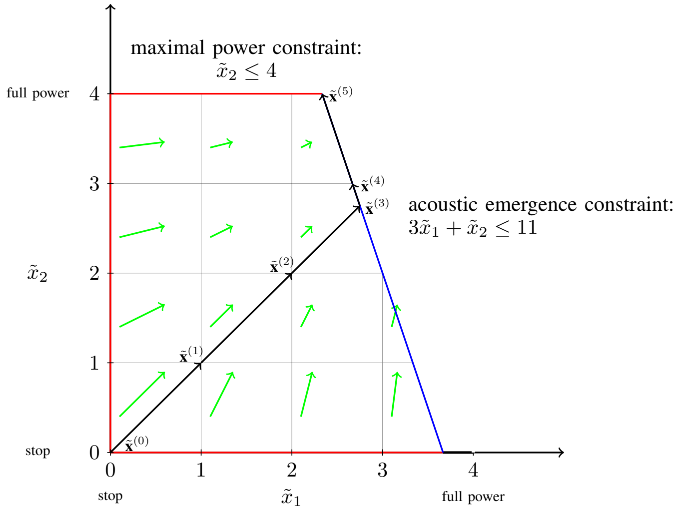

Fig. 3. Graphical representation of the gradient algorithm on an example with 2 wind turbines. The gradient are shown as grey arrows.

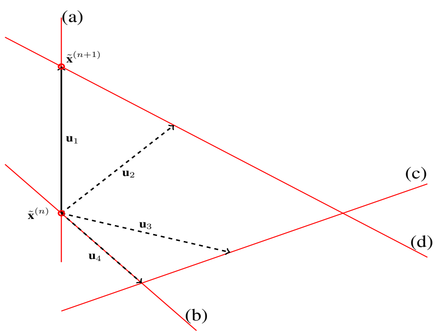

Fig. 4. Graphical representation to introduce the active, the hypothesized and the new reached constraints (or transitions) for one iteration of the algorithm. $\tilde{\mathbf{x}}^{(n)}$ is the current solution at iteration n. It is located on constraints/transitions (a) and (b) that are called *active constraints/transitions* for that reason. The vectors $\mathbf{u}_1$, $\mathbf{u}_2$, $\mathbf{u}_3$, $\mathbf{u}_4$ are the tested vectors for the next iteration in the algorithm. They may release one or several constraints and hence the remaining constraints are called *hypothesized constraints*. On the figure, the hypothesized constraint of $\mathbf{u}_1$ is (a), of $\mathbf{u}_4$ is (b). $\mathbf{u}_2$ and $\mathbf{u}_3$ don't have hypothesized constraints. Finally, we call *reached constraints*, the last constraint that was added compared to the last iteration. On the figure, for $\tilde{\mathbf{x}}^{(n+1)}$, the *reached constraint* is (d). Indeed, the active constraints of each iteration consist of the sum of the new reached constraints plus the hypothesized constraints of the last chosen vector. On the figure, the active constraints of $\tilde{\mathbf{x}}^{(n+1)}$ are (a) and (d), which are obtained by adding the reached constraint (d) to the hypothesized constraint (a) of the chosen vector u_1.

- $\mathcal{X}_{\text{adm}} = \mathcal{X}_{\text{poss}} \cap \{\tilde{\mathbf{x}} \in \mathbb{R}^I | \tilde{\mathbf{A}}^T \tilde{\mathbf{x}} \leq \mathbf{q}\}$ set of continuous admissible strategies.
- $\mathcal{W} = \{(w_1, w_2, .., w_I) \in \mathcal{C}^I\}$ set of possible discrete strategies.
- $\forall \mathbf{w} \in \mathcal{W}, \mathcal{X}_{\text{adm}}(\mathbf{w}) = \mathcal{X}_{\text{adm}} \cap \{\tilde{\mathbf{x}} \in \mathbb{R}^I | \forall i \in \{1, .., I\}, \tilde{\mathfrak{r}}_v(w_i) \leq \tilde{x}_i \leq \tilde{\mathfrak{r}}_v(w_i - 1)\}$ subspace of admissible strategies where the gradient is constant and associated to the discrete strategy $\mathbf{w}$.

We have the following properties :

- $\mathcal{X}_{\text{adm}} = \cup_{\mathbf{w} \in \mathcal{W}} \mathcal{X}_{\text{adm}}(\mathbf{w})$.
- $\forall \tilde{\mathbf{x}} \in \mathcal{X}_{\text{adm}}, \exists \mathbf{w} \in \mathcal{W}, \tilde{\mathbf{x}} \in \mathcal{X}_{\text{adm}}(\mathbf{w})$ (generally $\mathbf{w}$ is not unique).
- $\mathfrak{F}$ is differentiable for all interior values of the subsets $\mathcal{X}_{\text{adm}}(\mathbf{w})$. Also, the gradient is constant in each subset and its value is

$$\left[\frac{\mathfrak{p}_v(w_1) - \mathfrak{p}_v(w_1 - 1)}{\mathfrak{r}_v(w_1) - \mathfrak{r}_v(w_1 - 1)} \; \frac{\mathfrak{p}_v(w_2) - \mathfrak{p}_v(w_2 - 1)}{\mathfrak{r}_v(w_2) - \mathfrak{r}_v(w_2 - 1)} \; \cdots \; \frac{\mathfrak{p}_v(w_I) - \mathfrak{p}_v(w_I - 1)}{\mathfrak{r}_v(w_I) - \mathfrak{r}_v(w_I - 1)} \right]^T$$

As in the example, at each iteration n, we need to determine a new direction that increase the objective function and the biggest step size we can achieve without leaving the current subspace $\mathcal{X}_{\text{adm}}(\mathbf{w})$ we are in. At each iteration n, $\mathfrak{F}(\tilde{\mathbf{x}}^{(n+1)}) > \mathfrak{F}(\tilde{\mathbf{x}}^{(n)})$ is guaranteed by the fact that $(\tilde{\mathbf{x}}^{(n+1)} - \tilde{\mathbf{x}}^{(n)}) \cdot \boldsymbol{\nabla}\mathfrak{F}(\tilde{x}^{(n)}) > 0$ and $\| \tilde{\mathbf{x}}^{(n+1)} - \tilde{\mathbf{x}}^{(n)} \|_2$ is chosen small enough to avoid overpassing a constraint/gradient transition. Notice that $\|.\|_2$ stands for the usual Euclidian norm. This is always true since we follow $\boldsymbol{\nabla}\mathfrak{F}(\tilde{x}^n)$ or its orthogonal projection on an intersection of constraints. In other words, $\tilde{\mathbf{x}}^{(n+1)} = \tilde{\mathbf{x}}^{(n)} + \lambda^{(n)}\mathbf{u}^{(n)}$ where $\mathbf{u}^{(n)}$ is the chosen direction and $\lambda^{(n)}$ the step size. By convention, we consider in the following normalised directions ($\| \mathbf{u}^{(n)} \|_2 = 1$).

We use the *Moore-Penrose* inverse to compute the projection of the gradient on the intersections of constraints and gradient transitions [9, p. 855]:

$$\mathbf{u}^{(n)} = \boldsymbol{\nabla}\mathfrak{F}(\tilde{\mathbf{x}}^{(n)}) - \mathbf{M}((\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\boldsymbol{\nabla}\mathfrak{F}(\tilde{\mathbf{x}}^{(n)})) \tag{13}$$

where $\boldsymbol{\nabla}\mathfrak{F}(\tilde{\mathbf{x}}^{(n)})$ denotes the vector to be projected, $\mathbf{M}$ denotes the matrix of vectors that defines the hyperplanes, and $\mathbf{u}^{(n)}$ denotes the resulting projected vector.

To compute the step size $\lambda^{(n)}$ for a given direction $\mathbf{u}^{(n)}$, we use the piecewise form of $\mathfrak{f}$. For $w \in \mathcal{W}$ and $\lambda^{(n)} \in \mathbb{R}$, such that $\tilde{\mathbf{x}}^{(n)} \in \mathcal{S}_{\text{adm}}(w)$ and $(\tilde{\mathbf{x}}^{(n)} + \lambda^{(n)}\mathbf{u}^{(n)}) \in \mathcal{S}_{\text{adm}}(w)$, we have $\mathfrak{F}(\tilde{\mathbf{x}}^{(n)} + \lambda^{(n)}\mathbf{u}^{(n)}) = \mathfrak{F}(\tilde{\mathbf{x}}^{(n)}) + \lambda^{(n)}\boldsymbol{\nabla}\mathfrak{F}(\tilde{\mathbf{x}}^{(n)}) \cdot \mathbf{u}^{(n)}$. Then, the best step size $\lambda^{(n)}$ is the maximum value such that $\tilde{\mathbf{x}}^{(n+1)}$ is located in the same subset $\mathcal{S}_{\text{adm}}(w)$ than $\tilde{\mathbf{x}}^{(n)}$. If we write $\mathbf{y} = \boldsymbol{\nabla}\mathfrak{F}(\tilde{\mathbf{x}}^{(n)}) \cdot \mathbf{u}^{(n)}$ and $\mathbf{z} = \mathfrak{F}(\tilde{\mathbf{x}}^{(n)} + \lambda^{(n)}\mathbf{u}^{(n)}) - \mathfrak{F}(\tilde{\mathbf{x}}^{(n)})$, the best step size is then $\lambda^{(n)} = \min_{\{i \in \{1,...,I\} | y_i > 0\}} \frac{z_i}{y_i}$.

The algorithm stops when we can't find any positive step size for all possible directions.

The pseudo code of the algorithm is given in Algorithm 3. Notice that, it can be shown using the Karush-Kuhn-Tucker [17] [4] criteria that this algorithm converges to a global optimum in a finite number of iterations.

V. Experiments and Results

A. Experimental settings

We use real data coming from acoustical studies done by the company VENATHEC SAS. We extract the data of 28 wind farms in France. For each wind farms, measured data is classified along with explicative parameters (wind speed, wind direction, temperature,...) as commonly done in the field of acoustics. Each classification defines about 40 optimisation problems, totalling 1210 test cases. We also used the French threshold values for the acoustic constraints.

To evaluate the errors between the obtained solution and the theoretical maximal value, we used an exact B&B algorithm with a simplified bounding stage but a very long computation

Algorithm 3 Gradient algorithm

function GRADIENT
  Initialisation of $\tilde{x}^{(0)} = (0, 0, ..., 0)$ of size $I - I_0$;
  Initialisation of the gradient `grad` of size $I - I_0$;
  Initialisation of the first chosen direction:
  `p_chosen_grad` $= \mathbf{u}^{(0)} =$ `grad`;
  Computation of `lambda` from the gradient until
  the first gradient transition/constraint is reached;
  while make a step is possible **do**
    make a step of `lambda` $*$ `p_chosen_grad`;
    if a new gradient transition is reached **then**
      Compute the gradient `grad`;
      Project `grad` on the active constraints;
      Compute `number` = number of
      active gradient transitions;
      Compute the list of vectors
      `possible_projections` by projecting
      the previous vector on
      all the intersections of the
      (`number`-1)-combinations
      of the active gradient transitions;
    end if
    if a new constraint is reached **then**
      Compute `number` = number of active
      constraints and gradient transitions;
      Compute the list of vectors
      `possible_projections` by projecting
      the previous vector on
      all the intersections of the
      (`number`-1)-combinations
      of the active constraints and gradient transitions;
    end if
    Update the set of hypothesized
    constraints and transitions
    of each vector;
    while there are still vectors in `possible_projections` and `p_chosen_grad` is not assigned **do**
      compute `current_lambda`
      associated with the current vector `current_p`
      if `current_lambda` $\neq 0$ **then**
        Select the direction
        `p_chosen_grad` = `current_p`;
        Keep the size of the step :
        `lambda` = `current_lambda`;
      end if
    end while
    Update the reached gradient transitions
    or the reached constraints;
    Update the set of active constraints and gradient
    transitions with the set of hypothesized transitions
    and constraints;
  end while
end function

time that we evaluate on a supercomputer. Then, we considered the obtained solutions as the ground truth.

In the following we compare the proposed greedy solution and the proposed B&B solution (including efficient bounding) with the considered ground truth.

B. Results

The results obtained for the *greedy* algorithm, and the B&B algorithm are summarised in Table 5.

algorithm	**greedy**	**B&B**
maximal computation time (s)	0.25 s	240 s
global optimal not found	32 %	1.5 %
maximal relative error	30 %	2.3 %
mean relative error	0.15 %	9.10^{-3} %
maximal excess	-1.14×10^{-4} dB	$-6.46 \times 10^{-5} dB$
mean excess	-0.86 dB	-0.84 dB

Fig. 5. Results of the greedy heuristic and the proposed B&B algorithm. Relative error is defined by the difference between the obtained electricity production and the ground truth production divided by the ground truth production.

With real data, the *greedy* algorithm appears to be very fast (less than 0.25 s) and to be optimal in 70 % of the cases. However, when the optimum is not reached, the relative error can be up to 30 %. Nevertheless, the error is well distributed and we notice in Fig 6 that the relative error is smaller than 4 % in 90% of the test cases.

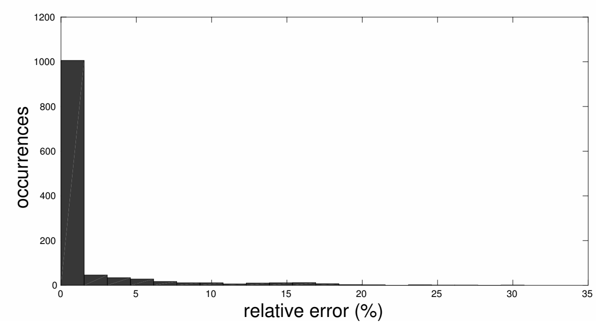

Fig. 6. Distribution of the relative error of the *greedy* initialisation

Starting with the solution of the greedy algorithm, the B&B algorithm refines the solution enhancing the production power at each iteration. Fig 7 shows the distribution of the relative error. As expected, the algorithm reaches the optimal solution in a huge majority (98.5%) of the test cases. For the remaining cases, the error is smaller than 2.3 %. This error is due to the heuristic used to limit the computation time. However it is still acceptable if we all the consider that the error is equal to only 9.10^{-3} on average over all test cases.

Regarding computation time, the important issue is to know how the relative error evolves with time, as the B&B algorithm can be manually stopped at any iteration and output the best current solution at that iteration. Fig 8 represents the temporal evolution of the long test cases (those with a computation time longer than 5 s). It shows the worst case among them and different quantiles. It can be seen that the algorithm converges within 2 min in all cases.

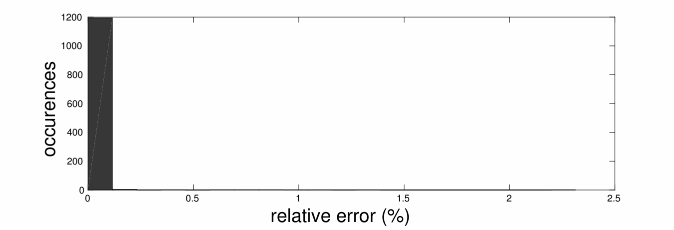

Fig. 7. Distribution of the relative error of the *B&B* algorithm

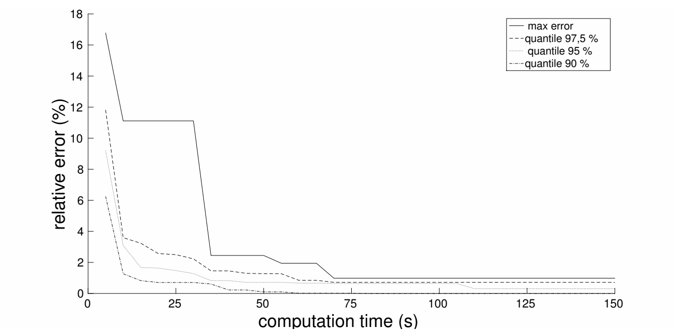

Fig. 8. Evolution of the relative error for the test data with long computation time (over 5 s).

VI. Conclusion

In this article, we have proposed a mathematical model of the problem of allocating acoustic operating modes as a nonlinear convex discrete programming problem. This allowed us to develop an efficient algorithm based on a B&B approach by proposing a gradient algorithm to the piece-wise nature of the relaxation by testing the algorithm on numerous real cases. This will allow companies in the field of acoustics to ensure the optimality of their results and might be used as a basis for further research.

Concerning perspectives, acoustic control of wind farms can be improved by taking the uncertainties into account and the algorithm might be adapted for that purpose. Note also that in certain cases, a wind turbine show some response delay (when it is fully stopped for instance). For these specific cases, a predictive model might also be coupled.

References

[1] Pierre Bonami, Mustafa Kilinç, and Jeff Linderoth. Algorithms and software for convex mixed integer nonlinear programs. In *Mixed integer nonlinear programming*, pages 1–39. Springer, 2012.

[2] Pierre Bonami, Jon Lee, Sven Leyffer, and Andreas Wächter. More branch-and-bound experiments in convex nonlinear integer programming. *Preprint ANL/MCS-P1949-0911, Argonne National Laboratory, Mathematics and Computer Science Division*, 2011.

[3] Brian Borchers and John E. Mitchell. An improved branch and bound algorithm for mixed integer nonlinear programs. *Computers & Operations Research*, 21(4):359–367, 1994.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2009.

[5] Samuel Burer and Adam N. Letchford. Non-convex mixed-integer nonlinear programming: a survey. *Surveys in Operations Research and Management Science*, 17(2):97–106, 2012.

[6] Chun-Lung Chen. Optimal wind–thermal generating unit commitment. *Energy Conversion, IEEE Transactions on*, 23(1):273–280, 2008.

[7] Arthur I. Cohen and Miki Yoshimura. A branch-and-bound algorithm for unit commitment. *Power Apparatus and Systems, IEEE Transactions on, (United States)*, 102(2):444–451, 1983.

[8] Alexandre Costa, Antonio Crespo, Jorge Navarro, Gil Lizcano, Henrik Madsen, and Everaldo Feitosa. A review on the young history of the wind power short-term prediction. *Renewable and Sustainable Energy Reviews*, 12(6):1725–1744, 2008.

[9] V.S. Donchenko and N.F. Kirichenko. Generalized inverse in control with constraints. *Cybernetics and Systems Analysis*, 39(6):854–861, 2003.

[10] Marco A Duran and Ignacio E Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, 36(3):307–339, 1986.

[11] Claudia DAmbrosio and Andrea Lodi. Mixed integer nonlinear programming tools: a practical overview. *4OR*, 9(4):329–349, 2011.

[12] Ya Ping Fang, Kaiwen Meng, and Xiao Qi Yang. Piecewise linear multicriteria programs: the continuous case and its discontinuous generalization. *Operations research*, 60(2):398–409, 2012.

[13] Arthur M Geoffrion. Generalized benders decomposition. *Journal of optimization theory and applications*, 10(4):237–260, 1972.

[14] Gregor Giebel, Richard Brownsword, George Kariniotakis, Michael Denhard, and Caroline Draxl. The state-of-the-art in short-term prediction of wind power: A literature overview. Technical report, ANEMOS. plus, 2011.

[15] Dawei He, Zhenyu Tan, and Ronald G. Harley. Chance constrained unit commitment with wind generation and superconducting magnetic energy storages. In *Power and Energy Society General Meeting, 2012 IEEE*, pages 1–6. IEEE, 2012.

[16] Dorit S. Hochbaum. A nonlinear knapsack problem. *Operations Research Letters*, 17(3):103–110, 1995.

[17] Kenneth Lange. Elementary optimization. In *Optimization*, pages 1–17. Springer, 2004.

[18] M. Lydia, Sahoo Subhendu Kumar, G. Edwin Prem Kumar, and A. Immanuel Selvakumar. Advanced algorithms for wind turbine power curve modeling. *Sustainable Energy, IEEE Transactions on*, 4(3):827–835, 2013.

[19] Kamlesh Mathur, Harvey M. Salkin, and Susumu Morito. A branch and search algorithm for a class of nonlinear knapsack problems. *Operations Research Letters*, 2(4):155–160, 1983.

[20] Y. Nian, Siyuan Liu, Dalei Wu, and Jiangchuan Liu. A method for optimal sizing of stand-alone hybrid pv/wind/battery system. 2013.

[21] Vladimir I. Norkin, Georg Ch. Pflug, and Andrzej Ruszczyński. A branch and bound method for stochastic global optimization. *Mathematical programming*, 83(1-3):425–450, 1998.

[22] Ignacio Quesada and Ignacio E Grossmann. An lp/nlp based branch and bound algorithm for convex minlp optimization problems. *Computers & chemical engineering*, 16(10):937–947, 1992.

[23] G. Terry Ross and Richard M. Soland. A branch and bound algorithm for the generalized assignment problem. *Mathematical programming*, 8(1):91–103, 1975.

[24] Jeremy Schild and Vincent Chavand. State of the art and new perspectives for the development of noise regulation of wind farms. *6th International Meeting on Wind Turbine Noise*, 2015.

[25] H. Siahkali and M. Vakilian. Electricity generation scheduling with large-scale wind farms using particle swarm optimization. *Electric Power Systems Research*, 79(5):826–836, 2009.

[26] Jacques Teghem. La recherche opérationnelle tome 1: Les méthodes d'optimisation. 2012.

[27] Tapio Westerlund and Frank Pettersson. An extended cutting plane method for solving convex minlp problems. *Computers & Chemical Engineering*, 19:131–136, 1995.