journal of scheduling manuscript No.
(will be inserted by the editor)

Solving electricity production planning by column generation

Antoine Rozenknop · Roberto Wolfler Calvo · Laurent Alfandari · Daniel Chemla · Lucas Létocart

Received: date / Accepted: date

Abstract This paper presents a heuristic method based on column generation for the EDF (Electricité De France) long term electricity production planning problem proposed as subject of the ROADEF/EURO 2010 Challenge. This is to our knowledge the first-ranked method among those methods based on mathematical programming, and was ranked 4^{th} overall. The problem consists in determining a production plan over the whole time horizon for each thermal power plant of the French electricity company, and for nuclear plants, a schedule of plant outages which are necessary for refueling and maintenance operations. The average cost of the overall outage and production planning, computed over a set of demand scenarios, is to be minimized. The method proceeds in two stages. In the first stage, dates for outages are fixed once for all for each nuclear plant. Data is aggregated with a single average scenario and reduced time steps, and a set-partitioning reformulation of this aggregated problem is solved for fixing outage dates with a heuristic based on column generation. The pricing problem associated with each nuclear plant is a shortest path problem in an appropriately constructed graph. In the second stage, the reload level is determined at each date of an outage, considering now all scenarios. Finally, the production quantities between two outages are optimized for each plant and each scenario by solving independent linear programming problems.

Keywords: Energy, electricity production, column generation, mathematical programming, heuristics.

1 Introduction

Optimization in the energy sector is a large subject. With the raise of raw materials prices and the growing awareness for environment concerns, energy consumption must be considered in a less greedy way in order to save resources. For years, several researchers have worked on optimizing the energy industry. In [7], the authors gather such type of problems, one of which is the electricity production planning. Electricity can be produced through different ways: nuclear and other thermal energies are the major part of production, but hydraulic energy or other renewable energies have sometimes a significant role (in Norway almost all the production is done using hydraulic energy). Producers can mix these different production resources in order to satisfy the energy demand while decreasing their costs and/or their impact on the environment. Electricity companies look for effective

Antoine Rozenknop · Roberto Wolfler Calvo · Daniel Chemla · Lucas Létocart
LIPN (UMR 7030) - Université Paris 13, 99 av. J-B Clement 93430 Villetaneuse
Tel.: +33-1-49403894
Fax: +33-1-48260712
E-mail: name.surname@lipn.fr

L. Alfandari
ESSEC Business School, av B. Hirsch 95021 Cergy-Pontoise
E-mail: alfandari@essec.edu

Daniel Chemla
CERMICS École des Ponts ParisTech, 6-8 av. Blaise Pascal, Cité Descartes - Champs sur Marne, 77455 Marne la Valle Cedex 2
E-mail: daniel.chemla@enpc.fr

optimization tools since even small percentages of gain through optimization translate into large savings. The problem of optimizing energy production planning has several difficulties, one of which is that the exact amount of the future demand and the exact price of raw materials are unknown. This difficulty naturally increases with the length of the time horizon, which can be in some cases very large.

A typical energy production planning problem with a short term horizon is the Unit Commitment Problem (UCP). In the UCP, several generators are available for producing energy. Costs and constraints are associated with generators depending on their type: nuclear and other thermal units, hydraulic energy, wind turbines. For example, in a short term horizon, the level of water in dams has to stay above some threshold. A forecast of the demand over a period of time is given. The objective is to find the best production assignment in order to satisfy the demand while minimizing production costs. Kazarlis *et al.* [8] propose a genetic algorithm to solve the UCP. Claudia Sagastizàbal *et al.* [5] present a Lagrangian relaxation method. Frangioni *et al.* in [6] extend Lagrangian relaxation methods to the UCP with ramping constraints, i.e., constraints that bound the difference of production of thermal units from one time step to the next. Claude Lemaréchal *et al.* [4] also studied UCP in the French context.

In long-term planning, typical constraints on energy production relate to maintenance operations that have to be executed periodically on nuclear plants. No production can be planned during these maintenance periods. The problem of scheduling the shut down of nuclear power plants for refueling and maintenance (see for example [9]) has been proposed as subject of the EURO/ROADEF 2010 challenge. EDF power generation facilities in France stand for a total of 98.8 GW of installed capacity and mix thermal energy (90 % in 2008, among which 86 % is produced by nuclear power and the rest by coal, fuel oil and gas), hydraulic and other renewable energies. The research project consists in modeling the electricity production assets, finding an optimal schedule of outages for each nuclear plant over a given time horizon, and determining an optimal production plan to satisfy demand. Moreover, in long-term electricity production planning, numerous uncertainties have to be taken into account, such as demand variation, generation units availability, spot market prices, quantities that can be bought or sold. This uncertainty leads to consider multiple scenarios and asks for a robust outage schedule that has to be feasible for every scenario. The scheduling of nuclear plant outages has to comply with various safety constraints and limitations on resources which are necessary to perform refueling and maintenance operations. The production planning fixes the quantity of energy to be produced by each plant at each time step for each scenario, and is also subject to technical constraints. The objective of the problem is to minimize the expected production cost over all scenarios. The proposed approach is based on mathematical programming and problem decomposition. An other approach used for solving the challenge ROADEF, based on a different decomposition method has been proposed by Lubsy *et al.* [11].

In this paper a column generation method (see [3] and [10] for surveys on this approach) is proposed to solve it. Column generation is often used to solve large-scale optimization problems when the problem can be formulated as a mixed integer linear program with a huge number of variables, or *columns* – called the *master problem* – as it is the case in the EDF problem. Creating and keeping all variables is not possible and useless as most of them would not enter the optimal solution. A *restricted master problem* is then defined, where instead of solving the problem on all its solution space, only a subset of variables (or columns) is considered. The column generation method iterates between two different problems: the restricted master problem and a *pricing subproblem*. The pricing subproblem aims at looking for new columns with negative reduced cost to add to the restricted master problem to improve its optimal value.

Even if the set of columns of the original master problem has a huge or exponential size, it is often possible to find the minimum reduced cost column in short or polynomial time, depending on the structure of the pricing problem. The method stops when no more negative reduced cost column is produced by the pricing subproblem, which means that the linear relaxation of the original master problem is solved to optimality. Nevertheless, in the case of integer variables the optimal solution of the linear relaxation could be far away from the optimal solution of the original master problem. In this case it is necessary either to start a branch and price algorithm (see for example [2] for a survey on this topic) or to have a complete pricing of the whole set of columns candidate to be in the optimal solution (see for example [1] for a reference on this topic).

Our approach is a heuristic method since only a subset of the whole set of possible columns are implicitly considered, due to the time limit imposed by the competition. This method is indeed heuristic for several reasons. The first reason is that production levels are discretized over time periods. This implies that the pricing subproblem is approximately solved using a shortest path algorithm in a graph with a finite number of nodes associated with production levels, which is much faster than solving it with continuous variables. The second

reason is that scenarios are aggregated into a single mean scenario. Finally, the column generation method leads to find the optimal solution of the relaxed problem on the subset of columns that could be produced with respect to the discretization considered. A branch and price algorithm was not envisaged due to the time limits. At the end of the column generation process, once the outages have been scheduled, two linear programming problems are solved to optimality for deciding the quantity of fuel and the optimal production planning over all scenarios for this set of outages.

The main contributions of this paper are the following:

- an original set partitioning-like reformulation of the problem;
- a pricing subproblem solved by shortest path computations on a particular state graph;
- a diversification method that changes the objective function, adding columns that may have a positive reduced cost;
- two linear programming problems solved to optimality for deciding the quantity of fuel and the production planning for the selected set of outages.

The discriminant strategy chosen for our approach is to spend time to generate a single high-quality schedule of outages, guided by the cost objective of column generation, rather than enumerating several schedules of outages in a shorter time.

This paper is organized as follow. The electricity production planning problem is presented in Section 2. Section 3 introduces the mathematical model that formulates the problem. Section 4 explains the decomposition applied and describes the overall column generation method used to solve it. Section 5 describes the formulation of the pricing subproblem and its resolution method. Section 6 gives the model of the final master problem that is solved for fixing dates of outages and critical stock-level periods, ending the first stage of the method. Section 7 shows how to improve the pricing subproblems in order to generate a greater number of columns with higher diversity. Section 8 presents the second-stage process used for fixing reload and production quantities for each plant. Section 9 gathers the numerical results obtained on the test instances while Section 10 concludes the paper.

2 Problem statement

The notations used in this paper mix the notations used in the ROADEF/EURO 2010 Challenge document [12] and notations issued from the paper by Lubsy *et al.* [11]. The various indexes used are scenarios $s \in S$, time steps $t \in T_= \{1,\dots,|T|\}$, weeks $w \in W = \{1,\dots,|W|\}$, non-nuclear plants $j \in J$, nuclear plants $i \in I$ and production cycles $k \in K_i = \{1,\dots,|K_i|\}$ for each nuclear power plant i, $|K_i|$ being the maximum number of cycles that can start during the considered period. The decision variables introduced in the Challenge document are:

- ha_{ik} : week of cycle k during which nuclear plant i goes offline (i.e. week of decoupling)
- p_{jts} : production of non-nuclear plant j during time step t over scenario s
- p_{its} : production of nuclear plant i during time step t over scenario s
- r_{ik} : reload performed during the outage of cycle k of nuclear plant i
- x_{its} : stock of fuel of nuclear power plant i at the beginning of time step t over scenario s
- x^f_{is} : final stock for plant i over scenario s

Variables ha are integer, all other variables are continuous variables.

The objective function is:

$$\min \sum_{i\in I}\sum_{k\in K} C_{ik} r_{ik} + \frac{1}{|S|}\sum_{s\in S}\left(\sum_{t\in T}\sum_{j\in J} C_{jts} p_{jts} D^t - \sum_{i\in I} C^f_i x^f_{is}\right)$$

where C_{ik} is the proportional cost of fuel during cycle k for nuclear plant i, C_{jts} is the proportional production cost for time t and scenario s for non-nuclear plant j, C^f_i is the selling price (or the gain) for the remaining fuel x^f_{is} and D^t is the length of time step t.

The set of constraints is the following, taking the same numbering as in the Challenge document. Some notations used throughout the paper are introduced with these constraints, amongst which :

- $T_{\mathscr{W}}$ is the set of timesteps contained in a set of weeks $\mathscr{W}$,

– L_{ik} is the length of an outage (defined in constraint `[CT13]` but used before).

`[CT1]` Demand satisfaction:

$$\sum_{j \in J} p_{jts} + \sum_{i \in I} p_{its} = DEM_{t,s} \quad \forall s,t$$

`[CT2]` For non-nuclear plants, we only have minimum and maximum production constraints:

$$\underline{P}_{jts} \leq p_{jts} \leq \overline{P}_{jts} \quad \forall t,s,j$$

where $\underline{P}_{jts}$ and $\overline{P}_{jts}$ are the minimum and maximum production for non-nuclear plants j on time step t for scenario s.

`[CT3]` Offline power: if the nuclear power plant is offline the production must be 0

$$t \in T_{[ha_{ik}, ha_{ik}+L_{ik}[} \Rightarrow p_{its} = 0 \quad \forall t,s,j$$

`[CT4]` Minimum power: if the nuclear power plant is online the production must be greater or equal than 0

$$p_{its} \geq 0 \quad \forall i,t,s$$

`[CT5]` Maximum power before activation of imposition of power profile constraint

$$\text{if} \quad (t \in T_{[ha_{ik}+L_{ik}, ha_{i(k+1)}[}) \text{ and } x_{its} \geq BO_{ik} \Rightarrow p_{its} \leq \overline{P}_{it} \quad \forall i,t,s,k$$

where BO_{ik} is a bound on stock of fuel that activates the imposition of power profile during cycle k, and $\overline{P}_{it}$ is the maximum power of plant i during time step t

`[CT6]` Maximum power after activation of imposition of power level constraint :

$$\left.\begin{array}{l} \text{if } t \in T_{[ha_{ik}+L_{ik}, ha_{i(k+1)}[} \text{ and } x_{its} < BO_{ik} \text{ then} \\ \left\{\begin{array}{ll} \text{if} & x_{its} \geq (PB_{ik}(x_{its}) \times \overline{P}_{it}) \times D^t \\ \text{then} & \frac{p_{its}}{PB_{ik}(x_{its}) \times \overline{P}_{it}} \in [1-\varepsilon, 1+\varepsilon] \\ \text{else} & p_{its} = 0 \end{array}\right. \end{array}\right\} \quad \forall s,t,i,k$$

where $PB_{ik}(x_{its})$ is an imposed decreasing production profile during the production campaign of cycle k (piecewise linear function dependent on the level of fuel), and ε is a tolerance parameter. It is important to point out that these constraints are non linear.

`[CT7]` Bounds on refueling for scheduled outages k:

$$\underline{R}_{ik} \leq r_{ik} \leq \overline{R}_{ik} \quad \forall i,k$$

where $\underline{R}_{ik}$ and $\overline{R}_{ik}$ are the minimum and maximum quantity that can be refueled, respectively.

`[CT8]` Initial fuel stock :

$$x_{i0s} = XI_i \quad \forall (i,s)$$

`[CT9]` Fuel stock variation during a production campaign

$$t \in T_{[ha_{ik}+L_{ik}, ha_{i(k+1)}[} \Rightarrow x_{i(t+1)s} = x_{its} - p_{its} \times D^t \quad \forall s,t,i,k$$

`[CT10]` Fuel stock variation during an outage. If t is the first time step of ha_{ik} then

$$x_{i(t+1)s} = ((Q_{ik}-1)/Q_{ik})(x_{its} - BO_{i(k-1)}) + r_{ik} + BO_{ik} \quad \forall i,t,s,k$$

where Q_{ik} is a refueling coefficient during the outage of cycle k

[CT11] Fuel stock variation during an outage

$$t \text{ is the first time step of } ha_{ik} \Rightarrow x_{its} \leq A_{ik},\ x_{i(t+1)s} \leq S_{ik} \quad \forall s,t,i,k$$

where A_{ik} (resp. S_{ik}) is the maximum bound on stock of fuel at the time of outage (resp. during production campaign) of cycle k

[CT12] Constraint on maximum modulation over a cycle

$$\sum_{t \in T_{[ha_{ik}+L_{ik}, ha_{i(k+1)}[} \text{ and } x_{its} \geq BO_{ik}} (\overline{P}_{it} - p_{its}).D^t \leq \bar{M}_{ik} \quad \forall s,i,k$$

where $\bar{M}_{ik}$ is the maximum modulation over the production campaign of cycle k. These constraints state that the gap between total production and production capacity, during a production campaign staying above the `BO` stock threshold, should not be too high.

[CT13] Earliest and latest date of outage

$$ha_{i(k+1)} \geq ha_{ik} + L_{ik} \quad \forall i,k$$

$$T^o_{ik} \leq ha_{ik} \leq T^a_{ik} \quad \forall i,k$$

where L_{ik} is the number of weeks (length) of outage k. These constraints bound the earliest and the latest date of an outage for each cycle $k \in K_i$ of each plant i.

[CT13bis] No outage $k \in K_i$ could be skipped as soon as it has a last possible week $T^a_{i,k}$ defined.

In the next series of constraints: N^{xx}_m are number of weeks associated to constraint $m \in M^{xx}$ and Q^{xx}_m represent quantities.

[CT14] Minimum spacing and maximum overlapping between outages of i and i'

$$ha_{ik} - ha_{i'k'} - L_{i'k'} \geq N^{14}_m \text{ or } ha_{i'k'} - ha_{ik} - L_{ik} \geq N^{14}_m \quad \forall (i,k),(i',k') \in G^{14}_m, \forall m \in M^{14}$$

[CT15] Minimum spacing and maximum overlapping between outages during a specific period. Let's define with $[b_m, e_m]$ a time interval, then
if

$$(b_m - L_{ik} + 1 \leq ha_{ik} \leq e_m) \text{ and } (b_m - L_{i'k'} + 1 \leq ha_{i'k'} \leq e_m)$$

then

$$ha_{ik} - ha_{i'k'} - L_{i'k'} \geq N^{15}_m \text{ or } ha_{i'k'} - ha_{ik} - L_{ik} \geq N^{15}_m \quad \forall (i,k),(i',k') \in G^{15}_m, \forall m \in M^{15}$$

[CT16] Minimum spacing between decoupling dates

$$|ha_{ik} - ha_{i'k'}| \geq N^{16}_m \quad \forall (i,k),(i',k') \in G^{16}_m, \forall m \in M^{16}$$

[CT17] Minimum spacing between coupling dates

$$|ha_{ik} + L_{ik} - ha_{i'k'} - L_{i'k'}| \geq N^{17}_m \quad \forall (i,k),(i',k') \in G^{17}_m, \forall m \in M^{17}$$

[CT18] Minimum spacing between coupling and decoupling dates

$$|ha_{ik} + L_{ik} - ha_{i'k'}| \geq N^{18}_m \quad \forall (i,k),(i',k') \in G^{18}_m, \forall m \in M^{18}$$

[CT19] Resource constraints on outages. Let define with

- w^m_{ik} represents the start of using the resources for the outage, expressed in terms of number of weeks after the beginning of the outage.
- i^m_{ik} the number of weeks during with the resources are not available for the next outage
- Q^{19}_m quantity of available resources

$$\sum_{(i,k)\in G_m^{19}} \delta(t,i,k) \leq Q_m^{19} \quad \forall t \in T \quad \forall m \in M^{19}$$

where $\delta(t,i,k) = 1$ if $ha_{ik} + w_{ik}^m \leq t \leq ha_{ik} + w_{ik}^m + i_{ik}^m$

[CT20] Maximum number of overlapping outages during a specific week w_m

$$\sum_{(i,k)\in G_m^{20}} \delta(w_m,i,k) \leq Q_m^{20} \quad \forall m \in M^{20}$$

where Q_m^{20} is the maximum number of outages at week w_m and $\delta(w,i,k) = 1$ if $ha_{ik} \leq w < ha_{ik} + L_{ik}$

[CT21] Maximum offline power capacity of a set of power plants during a time period. For a given time period $[b_m, e_m]$ the power capacity of a subset of plants C_m^{21} that are offline is bounded by a maximal quantity Q_m^{21}

$$\sum_{i\in C_m^{21}} \sum_k \delta(w,i,k)\overline{P}_{it} \leq Q_m^{21} \quad \forall m \in M^{21},\ \forall w \in [b_m, e_m],\ \forall t \in T_w$$

3 Compact formulation

The solution approach is based on mathematical programming. Our compact formulation is inspired from [11]. However, we provide a *complete* compact formulation as some constraints ([CT6] and [CT12]) were not directly modeled in [11]. Before defining the Mixed Integer Programming (MIP) model, we introduce some additional notations, sets, constants and variables.

Sets and data

- $w(t)$: week containing time step t
- T_w : set of timesteps in week w
- $q_{ik} = \frac{Q_{ik}-1}{Q_{ik}}$ the ratio of fuel kept from a cycle to the next one

Variables

Variables r_{ik}, p_{jts} are the same as defined before, but new binary variables are introduced and variables x_{its} are splitted as described below.

- y_{iwk} (binary): $= 1$ if cycle k for plant i begins in week $w \in W$, 0 otherwise,
- Y_{ik} (binary): $= 1$ if cycle k is used for plant i, 0 otherwise,
- x_{iks}^1 : fuel stock of nuclear power plant i at the beginning of cycle k (i.e. before refueling) over scenario s
- x_{iks}^2 : fuel stock of nuclear power plant i after refueling at cycle k over scenario s
- x_{is}^f : final stock for plant i over scenario s
- x_{ikts} : fuel stock of nuclear power plant i during time step $t \in T$ of cycle k over scenario s
- p_{ikts} : production of nuclear plant i during time step t of cycle k over scenario s
- z_{ikts} (binary) : $= 1$ iff $x_{ikts} \leq BO_{ik}$ (used for modeling [CT6] and [CT12])
- ξ_{ikts} (binary) : $= 1$ iff $PB_{ik}(x_{ikts})\underline{P}_{it}D^t \leq x_{ikts}$ (used for [CT6])

As in [11] we introduce intermediate variables $\eta(i,w,k) = \begin{cases} 1 - \sum_{w' \leq w} y_{iw'2} & k = 1 \\ \sum_{w' \leq w - L_{ik}} y_{iw'k} & k = |K_i| \\ \sum_{w' \leq w - L_{ik}} y_{iw'k} - \sum_{w' \leq w} y_{iw'(k+1)} & \text{otherwise} \end{cases}$
This binary variable indicates whether week w is a production week in cycle k for plant i.
Note that ha_{ik} can be rewritten as $ha_{ik} = \sum_{w\in W} w y_{iwk}$.

$$z = \min \sum_{i\in I}\sum_{k\in K} C_{ik}r_{ik} + \frac{1}{|S|}\sum_{s\in S}\left(\sum_{t\in T}\sum_{j\in J} C_{jts}p_{jts} - \sum_{i\in I} C_i^f x_{is}^f\right) \tag{1}$$

$$\text{s.t.} \quad \sum_{i\in I}\sum_{k\in K_i(w(t))} p_{ikts} + \sum_{j\in J} p_{jts} = DEM_{ts}, \qquad \forall t\in T, s\in S \qquad (2)$$

$$\underline{P}_{jts} \le p_{jts} \le \overline{P}_{jts} \qquad \forall j\in J, t\in T, s\in S \qquad (3)$$

$$\sum_{(iwk)\in G} y_{iwk} \le \alpha_G \qquad \forall G\in \mathscr{G} \qquad (4)$$

$$T^o_{ik} \le \sum_{w\in W} w y_{iwk} \le T^a_{ik} \qquad \forall i\in I, k\in K_i \qquad (5)$$

$$p_{ikts} \le \overline{P}_{it}\,\eta(i,w(t),k), \qquad \forall i\in I, k\in K_i, t\in T, s\in S \qquad (6)$$

$$Y_{ik} = \sum_{w\in W} y_{iwk} \qquad \forall i\in I, k\in K_i \qquad (7)$$

$$\underline{R}_{ik} Y_{ik} \le r_{ik} \le \overline{R}_{ik} Y_{ik} \qquad \forall i\in I, k\in K_i \qquad (8)$$

$$Y_{i(k-1)} \ge Y_{ik} \qquad \forall i\in I, \forall k\in K_i\setminus\{1\} \qquad (9)$$

$$x^1_{i(k+1)s} = x^2_{iks} - \sum_{t\in T} p_{ikts} D^t \qquad \forall i\in I, k\in K_i, s\in S \qquad (10)$$

$$x^2_{iks} = \left[q_{ik}(x^1_{iks} - \mathtt{BO}_{i(k-1)}) + r_{ik} + \mathtt{BO}_{ik}\right] Y_{ik} + x^1_{iks}(1-Y_{ik}) \qquad \forall i\in I, k\in K_i, s\in S \qquad (11)$$

$$x^1_{iks} \le A_{ik} + (1-Y_{ik})N \qquad \forall i\in I, k\in K_i, s\in S \qquad (12)$$

$$x^2_{iks} \le S_{ik} + (1-Y_{ik})N \qquad \forall i\in I, k\in K_i, s\in S \qquad (13)$$

$$x^f_{is} = \sum_{k\in K_i} x_{i(k+1)s}(Y_{ik} - Y_{i(k+1)}) \qquad \forall i\in I, s\in S \qquad (14)$$

$$x_{ikts} = x^2_{iks} - \sum_{t'\le t} p_{ikt's} D^t \qquad \forall i\in I, k\in K_i, t\in T, s\in S \qquad (15)$$

$$\sum_{t\in T} (\underline{P}_{it} - p_{ikts}) D^t (1 - z_{ikts}) \le \overline{M}_{ik} \qquad \forall i\in I, k\in K_i \qquad (16)$$

$$(1-\varepsilon)\xi_{ikts} z_{itks} \le \frac{p_{ikts}}{PB_{ik}(x_{ikts})\overline{P}_{it}} \xi_{ikts} z_{itks} \le (1+\varepsilon)\xi_{ikts} z_{itks} \qquad \forall i\in I, k\in K_i, t\in T, s\in S \qquad (17)$$

$$\eta(i,w,k)(BO_{ik} - x_{itks}) z_{ikts} \ge 0, \qquad \forall i\in I, \forall k\in K_i, \forall t\in T, \forall s\in S \qquad (18)$$

$$\eta(i,w,k)(x_{itks} - BO_{ik})(1 - z_{ikts}) \ge 0, \qquad \forall i\in I, \forall k\in K_i, \forall t\in T, \forall s\in S \qquad (19)$$

$$(PB_{ik}(x_{itks})\overline{P}_{it} D_t - x_{itks})\xi_{itks} \le 0 \qquad \forall i\in I, \forall k\in K_i, \forall t\in T, \forall s\in S \qquad (20)$$

$$(x_{itks} - PB_{ik}(x_{itks})\overline{P}_{it} D_t)(1 - \xi_{itks}) \le 0 \qquad \forall i\in I, \forall k\in K_i, \forall t\in T, \forall s\in S \qquad (21)$$

$$y_{iwk}, Y_{ik} \in \{0,1\} \qquad \forall i\in I, \forall k\in K_i, \forall w\in W \qquad (22)$$

$$z_{ikts}, \xi_{ikts} \in \{0,1\} \qquad \forall i\in I, \forall k\in K_i, \forall t\in T, \forall s\in S \qquad (23)$$

$$p_{ikts}, p_{jts}, x^1_{iks}, x^2_{iks}, x^f_{is}, x_{ikts} \ge 0 \qquad \forall i\in I, \forall k\in K_i, \forall t\in T, \forall s\in S \qquad (24)$$

The objective function (1) minimizes the sum of the total reload cost and the average production cost (reduced by the profit value of remaining fuel) over all scenarios. Constraints (2) specify that the total quantity produced by nuclear power plants and thermal power plants must be equal to the energy demand for each period and scenario. Constraints (4) are a general way of writing constraints `[CT14]` to `[CT21]`. The right hand side α_G is an integer number, generally equal to one (corresponding then to incompatibility constraints between dates of outages). Constraints (5) ensure that outages start in the imposed intervals of `[CT13]`. Constraints (3) and (6) limit the production of thermal and nuclear power plants. Equality constraints (7) define the binary variables Y_{ik} that indicate whether cycle k is selected or not in the schedule. Constraints (8) bound the reloaded amount of fuel in a cycle. Constraints (9) say that when a cycle k is selected, then all cycles before k are also selected in the schedule. Constraints (10) say that the level of fuel stock at the end of cycle k (i.e., at the beginning of cycle $k+1$) is the level of stock after refueling minus the total production of cycle k. Constraints (11) express that the level of fuel stock after refueling is equal to the level before refueling (i.e., at the beginning of cycle k), plus a function of the reload quantity according to constraints `[CT10]`. The two levels of stock after and before refueling are bounded in constraints (12) and (13), corresponding to constraints `[CT11]`. $N > 0$ is a very big parameter so that no bound holds if $Y_{ik} = 0$, similarly to [11]. The computation of the final level of fuel at the end of the time horizon is made in constraints (14). In these constraints, only one term in the right-hand-side

sum will be non-zero by (9) and will correspond to the final stock. Constraints (15) define the level of stock during a production time step as the stock level after reload minus the total production of the campaign up to this step. The modulation constraints [CT12] are expressed in (16). Constraints (17) [CT6] impose the piecewise linear decreasing production profile as soon as $PB_{ik}(x_{ikts})\underline{P}_{it}D^t \leq x_{ikts} < BO_{ik}$, which implies $\xi_{ikts}z_{ikts} = 1$ via constraints (18-21), or no production otherwise. Both later constraints use variables z_{ikts} that are forced to 0 as soon as $x_{ikts} \geq BO_{ik}$ by constraints (18), and to 1 otherwise by constraints (19). Constraints (20) and (21) ensure that the level of fuel is enough to start the BO profile. Constraints (22), (23) and (24) define the range of the variables.

The complexity of the model is mainly due to the combinatorial choice for binary variables y_{iwk}, z_{itks} and ξ_{itks}, and the nonlinearity of constraints (11), (14) and (16) - (21).

4 Decomposition and solution method

This Section presents how the proposed approach can be derived and explained on the basis of the compact formulation and the outline of the complete solution method.

4.1 Decomposition

The set of constraints can be decomposed in two sets. The *global* constraints are demand constraints (2) and incompatibility constraints (4) which link all power plants, and constraints (3) that bound the production in thermal plants. The *local* constraints (5)-(21) can be decomposed by each nuclear power plant i.

Assume that we are able to find for each nuclear power plant i the set of all possible production plans (i.e., feasible values of $p_{ikts}, \forall t \in T$, outages and reload quantities). Define $\mathscr{P}$ as the index set of all feasible production plans for the set of nuclear power plants, and let $\mathscr{P}_i$ be the subset of production plans associated to nuclear power plant i. Therefore, we can write a set partitioning-like reformulation (i.e., a set partitioning model with some more constraints) of the original problem where binary variables y_ℓ indicates whether production plan $\ell \in \mathscr{P}_i$ is selected for plant i, and variables p_{jts} represent the thermal energy used for each scenario $s \in S$ and each period of time $t \in T$.

$$(F) \quad z(F) = \min \sum_{i\in I}\sum_{\ell\in\mathscr{P}_i} c_\ell y_\ell + \frac{1}{|S|}\sum_{s\in S}\sum_{t\in T}\sum_{j\in J} C_{jts}p_{jts} \tag{25}$$

$$\text{s.t.} \quad \sum_{i\in I}\sum_{\ell\in\mathscr{P}_i} p_{\ell ts}y_\ell + \sum_{j\in J} p_{jts} = DEM_{ts}, \qquad \forall t \in T, \forall s \in S \tag{26}$$

$$\sum_{\ell\in\mathscr{P}_i} y_\ell = 1 \qquad \forall i \in I \tag{27}$$

$$\underline{P}_{jts} \leq p_{jts} \leq \overline{P}_{jts}, \qquad \forall j \in J, \forall t \in T, \forall s \in S \tag{28}$$

$$\sum_{\ell\in\mathscr{P}} a_{\ell G}y_\ell \leq \alpha_G, \qquad \forall G \in \mathscr{G} \tag{29}$$

$$y_\ell \in \{0,1\} \qquad \forall \ell \in \mathscr{P}, \tag{30}$$

$$p_{jts} \geq 0 \qquad \forall j \in J, \forall t \in T, \forall s \in S \tag{31}$$

where $p_{\ell ts}$ is the quantity produced in plan ℓ in time step t over scenario s, and $a_{\ell G}$ are 0-1 coefficients associated with plan ℓ in constraint set G.

Equation (25) is the resulting objective function to be minimized. Constraints (26) specify that the energy demand of each period $t \in T$ and for each scenario $s \in S$ must be satisfied. Constraints (27) impose that for each nuclear power plant one and only one column (refuel and production plan) must be chosen. Constraints (28) bound the quantity of energy that a thermal power plant can produce each time period. Constraints (29) limit the number of nuclear power plants that can stop together. Constraints (30) and (31) define the range of the variables.

Note that constraints (26), (28) and (29) correspond to constraints (2), (3) and (4).

4.2 Solution method

The efficiency of the approach largely depends on the tractability of problem (F) and the $|I|$ subproblems associated with each nuclear power plant. These problems have very large scale given the number $|S|$ of scenarios and the number T of time steps. To deal with this complexity, we chose to simplify the original set of data by aggregating the S scenarios into a single *average* scenario $\bar{s}$, and the time steps $t = 1,\dots,T$ into weeks $w = 1,\dots,W$. We note for every week $w \in W$:

$$
\begin{aligned}
DEM_w &= \frac{1}{|S|}\sum_{s\in S}\sum_{t\in T_w} D^t.\, DEM_{ts} \\
C_{jw} &= \frac{1}{|S|}\sum_{s\in S}\frac{|W|}{|T|}\sum_{t\in T_w} C_{jts} \\
\underline{E}_{jw} &= \frac{1}{|S|}\sum_{s\in S}\sum_{t\in T_w} D^t.\, \underline{P}_{jts} \\
\overline{E}_{jw} &= \frac{1}{|S|}\sum_{s\in S}\sum_{t\in T_w} D^t.\, \overline{P}_{jts} \\
\overline{E}_{iw} &= \sum_{t\in T_w} \overline{P}_{it}\ .\ D^t
\end{aligned}
$$

From now on, production quantities stand for each week $w \in W$, and we use notations $e_{\ell w}$ for the production of column $\ell \in \mathscr{P}_i$ in week w, and production variables e_{jw} (resp. e_{ikw}) for thermal (resp. nuclear) power plants. Recall that the energy is related to the power by the formula : $E_w = \sum_{t\in T_w} p_t D^t$. Therefore the problem (F) to solve becomes

$$
\begin{aligned}
(\bar{F}) \quad z(\bar{F}) = \min & \sum_{i\in I}\sum_{\ell\in\mathscr{P}_i} c_\ell y_\ell + \sum_{w\in W} C_{jw} e_{jw} && && (32)\\
\text{s.t.} \quad & \sum_{i\in I}\sum_{\ell\in\mathscr{P}_i} e_{w\ell} y_\ell + \sum_{j\in J} e_{wj} = DEM_w && \forall w\in W && (33)\\
& \sum_{\ell\in\mathscr{P}_i} y_\ell = 1 && \forall i\in I && (34)\\
& \underline{E}_{jw} \le e_{jw} \le \overline{E}_{jw} && \forall j\in J, \forall w\in W && (35)\\
& \sum_{i\in I}\sum_{\ell\in\mathscr{P}_i} a_{\ell G} y_\ell \le \alpha_G && \forall G\in\mathscr{G} && (36)\\
& y_\ell \in \{0,1\} && \forall \ell\in\mathscr{P} && (37)\\
& e_{jw} \ge 0 && \forall j\in J, w\in W && (38)
\end{aligned}
$$

Formulation $(\bar{F})$ is impractical to solve even for instances of moderate size, since the number of variables is typically exponential, which justifies a column generation approach. We define a linear relaxation $(L\bar{F})$ of $\bar{F}$:

$$
\begin{aligned}
(L\bar{F}) \quad z(L\bar{F}) = \min & \sum_{i\in I}\sum_{\ell\in\mathscr{P}_i} c_\ell y_\ell + \sum_{w\in W} C_{jw} e_{jw} && && (39)\\
\text{s.t.} \quad & \sum_{i\in I}\sum_{\ell\in\mathscr{P}_i} e_{\ell w} y_\ell + \sum_{j\in J} e_{jw} = DEM_w && \forall w\in W && (40)\\
& \sum_{\ell\in\mathscr{P}_i} y_\ell = 1 && \forall i\in I && (41)\\
& \underline{E}_{jw} \le e_{jw} \le \overline{E}_{jw} && \forall j\in J, w\in W && (42)\\
& 0 \le y_\ell \le 1 && \forall \ell\in\mathscr{P}, && (43)\\
& e_{jw} \ge 0 && \forall j\in J, \forall w\in W && (44)
\end{aligned}
$$

When passing from $\bar{F}$ to $(L\bar{F})$, constraints (36) have been removed since their dual variables are hard to compute in the pricing problem within the limited computational time imposed by the competition. For compensation we allow to add to problem $(L\bar{F})$ some columns with positive reduced costs as described in Section 7. The solution of problem $(L\bar{F})$ provides the dual variables μ_w and u_i associated with constraints (33) and (34), respectively. The pricing problem exploits them for generating the column(s) with minimum reduced cost:

$$\ell^* = \underset{\ell \in \mathscr{P}}{\operatorname{argmin}} \bar{c}_\ell \quad \text{with} \quad \bar{c}_\ell = c_\ell - \sum_{w \in W} e_{\ell w} \mu_w - u_i \tag{45}$$

The pricing subproblem (P_i) associated with each nuclear power plant $i \in I$ can be written as follows, given the decomposition previously described and aggregation of time steps into weeks. Indexes of various scenarios have disappeared and the t indexes have been replaced by w indexes in data and variables.

$$(P_i) \quad \min \sum_{k \in K_i} C_{ik} r_{ik} - C_i^f x_i^f - \sum_{w \in W, k \in K_i(w)} e_{ikw} \mu_w - u_i \tag{46}$$

$$\text{s.t.} \quad e_{ikw} \leq \overline{E}_{iw} \eta(i,w,k), \qquad k \in K_i, w \in W \tag{47}$$

$$Y_{ik} = \sum_{w \in W} y_{ikw} \qquad \forall k \in K_i \tag{48}$$

$$\underline{R}_{ik} Y_{ik} \leq r_{ik} \leq \overline{R}_{ik} Y_{ik} \qquad \forall k \in K_i \tag{49}$$

$$Y_{i(k-1)} \geq Y_{ik} \qquad \forall k \in K_i \setminus \{1\} \tag{50}$$

$$x^1_{i(k+1)} = x^2_{ik} - \sum_{w \in W} e_{ikw} \qquad \forall k \in K_i \tag{51}$$

$$x^2_{ik} = \left[q_{ik}(x^1_{ik} - \mathtt{BO}_{i(k-1)}) + r_{ik} + \mathtt{BO}_{ik} \right] Y_{ik} + x^1_{ik}(1 - Y_{ik}) \qquad \forall k \in K_i \tag{52}$$

$$x^1_{ik} \leq A_{ik} + (1 - Y_{ik}) N \qquad \forall k \in K_i \tag{53}$$

$$x^2_{ik} \leq S_{ik} + (1 - Y_{ik}) N \qquad \forall k \in K_i \tag{54}$$

$$x_i^f = \sum_{k \in K_i} x^1_{i(k+1)} (Y_{ik} - Y_{i(k+1)}) \tag{55}$$

$$\sum_{w \in W'_{ik}} (\overline{E}_{iw} - e_{ikw}) z_{ikw} \leq \overline{M}_{ik} \qquad \forall k \in K_i \tag{56}$$

$$(1 - \varepsilon) \xi_{ikt} z_{itk} \leq \frac{p_{ikt}}{PB_{ik}(x_{ikt}) \overline{P}_{it}} \xi_{ikt} z_{itk} \leq (1 + \varepsilon) \xi_{ikt} z_{itk} \qquad \forall i \in I, k \in K_i, t \in T \tag{57}$$

$$\eta(i,w,k)(BO_{ik} - x_{ikt}) z_{ikt} \geq 0, \qquad \forall i \in I, \forall k \in K_i, \forall t \in T \tag{58}$$

$$\eta(i,w,k)(x_{ikt} - BO_{ik})(1 - z_{ikt}) \geq 0, \qquad \forall i \in I, \forall k \in K_i, \forall t \in T \tag{59}$$

$$(PB_{ik}(x_{ikt}) \overline{P}_{it} D_t - x_{ikt}) \xi_{itk} \leq 0 \qquad \forall i \in I, \forall k \in K_i, \forall t \in T \tag{60}$$

$$(x_{ikt} - PB_{ik}(x_{ikt}) \overline{P}_{it} D_t)(1 - \xi_{itk}) \leq 0 \qquad \forall i \in I, \forall k \in K_i, \forall t \in T \tag{61}$$

$$x_{ikt} = x^2_{ik} - \sum_{\tau < t} p_{ik\tau} D^\tau \qquad \forall k \in K_i, \forall t \in T \tag{62}$$

$$\sum_{t \in T_w} p_{ikt} D_t = e_{ikw} \sum_{w' \leq w} (y_{ikw'} - y_{i(k+1)w'}) \qquad \forall k \in K_i, \forall w \in W \tag{63}$$

$$y_{iwk}, Y_{ik}, z_{ikt}, \xi_{ikt} \in \{0,1\} \qquad \forall k \in K_i, \forall w \in W, \forall t \in T \tag{64}$$

$$e_{ikw}, p_{ikt}, x^1_{ik}, x^2_{ik}, x^f_i, x_{ikt} \geq 0 \qquad \forall k \in K_i, \forall w \in W, \forall t \in T \tag{65}$$

The Pricing problem (P) consists in optimizing every problem (P_i) for $i \in I$ and output the column with best optimal value (minimum reduced cost). The columns with negative reduced cost output by (P) are added to problem $(L\bar{F})$ and the process iterates between $(L\bar{F})$ and (P) until (P) finds no more negative reduced cost column. As we mentioned in Section 1, the proposed column generation method is a heuristic scheme. Indeed, due to the time constraints imposed by the competition, only an approximation of the pricing problem P is solved at each iteration. This holds because the pricing subproblem is a shortest path problem solved on a graph whose paths do not explicitly represent all feasible plans, as explained in Section 5. At the end of the column generation,

the set of columns forms a relaxed restricted master program which is called the Restricted $L\bar{F}$ ($RL\bar{F}$), and solved to optimality as explained in Section 6.

The obtained optimal solution is a set of columns and variables p_{wj} satisfying constraints (40)-(42) at minimum cost. Each column is a production plan containing the dates for the outages and the periods when stock reaches a critical level equal to BO_{ik} (if such periods exist for a given nuclear plant). Therefore the solution obtained at the end of the first phase is used for fixing the decision variables on the outages and the date when the BO-threshold is attained. The advantage is to avoid to deal explicitly with non linear constraints, since once one has fixed BO, the corresponding decreasing profile is completely defined for the successive periods until the next outage.

The only decisions that remain to be made in the second phase are the reload variables r_{ik} and the production variables p_{ikts} , which are easier to settle once the binary outage variables y_{iwk} have been fixed. For setting the values of p_{ikts}, p_{jts} and r_{ik}, it is necessary to use the whole model with all time steps $t = 1,\ldots,T$ and all scenarios $s \in S$. Since it has no more binary variables, it can be solved efficiently by Linear Programming.

The two-stage method is stated below.

Phase 1 Fixing dates for outages and BO stock-level periods

subphase 1 Solve the aggregate set partitioning-like formulation $(L\bar{F})$ by column generation

- Aggregate data, passing from $|T \times S| = (5000 \times 500)$ to $|W \times \{\bar{s}\}| = (260 \times 1)$
- Create an acyclic graph G_i, for each nuclear power plant i
  - Repeat until a time limit is elapsed or no new column of negative reduced cost exists
    - Solve a shortest path subproblem in G_i, for each nuclear power plant i
    - Add the new columns (corresponding to shortest paths) to the $(L\bar{F})$ master problem and solve it by the Simplex method

subphase 2 Give the final restricted master problem to a *MIP* solver adding all the clique-like inequalities (36), exploiting the fact that only a small number of nuclear power plants can be refueled at the same time. Outputs are the dates for outages and BO stock-level periods.

Phase 2 Solve LP problems for defining the reload level for each date of outages and optimizing the production planning for each power plant and each scenario.

Phase 2 is described in Section 8. Sections 5 and 6 detail the first phase of the method. In phase 1, for every nuclear plant $i \in I$, two decisions are fixed: dates for outages respecting the scheduling constraints on outages and periods when stock reaches the BO_{ik} level, with imposed non linear decreasing power profile on the following periods. This step is the most time-consuming step of the method.

5 Solving the pricing problem

The pricing problem (P_i) consists in finding a reload and production plan for each nuclear power plant i. The problem contains a lot sizing problem and therefore can be solved by a shortest path algorithm in a graph (see for example [13], [14] and [15]). The main point is how to build the graph. The difficulty of this step is that the decision variables are a combination of continuous and binary variables. The binary variables define the possible outage dates while the continuous variables define the quantity of fuel that must be reloaded and the production for each time step. The further difficulty is due to the BO piecewise non linear function. Last but not least comes from the limited computational time imposed by the competition. Therefore, in the proposed approach, we chose to design a graph in which a production plan would be made of a relatively small number of arcs, each production campaign being represented by at most two arcs, and in such a way that cost computations of each arc could be managed efficiently. For efficiency reasons, we made the simplification of discretizing the possible fuel stock levels at the beginning of outages, so that the stock level could be fixed once for all at both ends of each arc of the graph. In the construction of the graph, constraints (47) to (61) are taken into account either when a node is defined or either when an arc is defined, such that any path on this graph represents a feasible solution.

The pricing problem has been implemented as a three-step process:

Algorithm 1 The pricing algorithm

for all nuclear power plants i **do**
1) create an acyclic static *stock graph* $G_i = (N_i, A_i)$;
2) for each arc $a \in A_i$ compute weight $\bar{c}_a$ (optimal reduced cost on the arc);
3) generate a new column by finding in G_i a path of minimum weight.
end for

Step 1 is quite time consuming but occurs only once at the beginning of the whole process. Steps 2 and 3 are light processes and can be serialized with the computation of new dual variables μ_w.

5.1 Creation of a static graph

Description of the graph. Each node $n \in N_i$ in graph G_i represents a possible state of power plant i and is characterized by a 4-tuple $(type_n, x_n, w_n, k_n)$, where:

- w_n is the week associated with n ;
- x_n is the fuel stock of power plant i *at the beginning* of week w_n ;
- k_n is the id of the campaign in which the power plant will be during week w_n ;
- $type_n$ is the node type. There are two main types of nodes:
  - an "`OUTAGE`" node n marks the start of a new production campaign and w_n is the week during which the outage occurs ;
  - a "`BO`" node n marks the start of a time period during which the fuel stock will be lower than BO_{ik_n} and the power level will be imposed by constraints `[CT6]` ; the stock x_n of a `BO` node is indeed a little higher than BO_{ik_n}, but ensures that BO_{ik_n} can be reached *during* week w_n by imposing the maximal power production of constraints `[CT5]` at the beginning of the week. In the model, it corresponds to setting the constraints (57)-(63).

  Besides, two nodes with special types are added, corresponding to the `initial` and `final` states of a power plant.

  Each arc $a = (n_1, n_2) \in A_i$ in graph G_i represents a possible direct transition from state n_1 to n_2.

- If n_1 is an `OUTAGE` node, this means that it is possible to reach state n_2 by proceeding to an outage on week w_{n_1} followed by a production campaign, without need for any other outage and without reaching the `BO` stock level during the time interval $[w_{n_1}, w_{n_2}]$. Note that n_2 can be either a `BO` node or an `OUTAGE` node. If it is a `BO` node, then $k_{n_2} = k_{n_1}$; else $k_{n_2} = k_{n_1} + 1$. Arcs leaving from `OUTAGE` nodes will be referred to as `OUTAGE` arcs.
- If n_1 is a `BO` node, this means there exists a production plan leading from n_1 to n_2 that respects the imposed profile of constraints `[CT6]` with the acceptable ε tolerance. Note that here n_2 has to be an `OUTAGE` node and thus $k_{n_2} = k_{n_1} + 1$. Arcs leaving from `BO` nodes will be referred to as `BO` arcs.

Furthermore, arcs are constructed so as to guarantee that technical constraints `[CT3]` to `[CT13bis]` (eq. (47) to (56)) can be respected for any path from the `initial` to the `final` node of G_i.

Building the graph. We build graph G_i in the following way:

Algorithm 2 Creation of graph G_i

1: create the `initial` node at week 1
2: **for** week $w = 1$ **to** $|W|$ **do**
3: **for all** previously created nodes n with $w_n = w$ **do**
4: create new nodes n' with $w_{n'} > w$ so that there exists a production plan leading the power plant from state n to state n' without violating constraints `[CT3]` to `[CT13bis]`
5: **end for**
6: **end for**

In order to find new nodes (step 4 of Algorithm 2), we follow the beam of possible stock interval(s) over time from a node n, as stated in Algorithm 3:

Algorithm 3 Creation of new nodes in G_i from $n = (type_n, x_n, w_n, k_n)$

1: **for** week $w = w_n + 1$ **to** $|W|$ **do**
2: compute the possible stock interval $[x_{min_{n,w}}, x_{max_{n,w}}]$ of power plant i at the beginning of week w
3: **if** the outage of campaign $k_n + 1$ is allowed at week w according to constraints `[CT13]` **then**
4: compute the intersection of $[x_{min_{n,w}}, x_{max_{n,w}}]$ with $[0, A_{ik_n+1}]$
5: create new nodes $n' = (\texttt{OUTAGE}, x_{n'}, w, k_n + 1)$ with $x_{n'}$ in this intersection (constraints `[CT11]`)
6: **end if**
7: **if** $\texttt{BO}_{ik_n} \in [x_{min_{n,w}}, x_{max_{n,w}}]$ **then**
8: create new nodes $n' = (\texttt{BO}, x_{n'}, w-1, k_n)$ with $x_{n'} \in [x_{min_{n,w-1}}, x_{max_{n,w-1}}]$ computed so that if the power plant follows the imposed profile of `[CT6]` during week $w-1$, its stock will fall under $\texttt{BO}_{ik_n}$ at the beginning of week w
9: **end if**
10: **end for**

For steps 5 and 8 of Algorithm 3, we discretize the interval $[x_{\min_{n,w}}, x_{\max_{n,w}}]$ of possible stocks of fuel at week w by a fixed amount of fuel Δ_i. New nodes are only created when needed, so as to ensure that there will be a node in every non-empty intervals $[x_{\min_{n,w}}, x_{\max_{n,w}}] \cap [x, x + \Delta_i]$, for all $x \in \mathbb{R}_+$. We compute Δ_i relatively to the mean capacity of powerplant i after choosing *a priori* a global discretisation parameter ρ :

$$\Delta_i = \rho \ . \ \frac{\sum_{w \in W} \overline{E}_{iw}}{|W|}$$

Step 2 of Algorithm 3 depends on the node type.

- If $type_n = \texttt{BO}$, we follow the imposed profile of constraints `[CT6`, i.e. (57)`]` with the authorized ε-deviation, counted positively for $x_{min_{n,w}}$ and negatively for $x_{max_{n,w}}$.
- If $type_n = \texttt{OUTAGE}$, we use `[CT10`, i.e. (52)`]` to compute stock variation during the outage, `[CT7`,(49)`]` (bounds on refueling) and `[CT11`,(53)-(54)`]` (maximum stock after refueling) to compute the interval after refueling ; when the production campaign starts, $x_{max_{n,w}}$ is computed using the maximum allowed modulation of `[CT12`,(56)`]` – and $x_{min_{n,w}}$ using no modulation at all.

The node creation process is illustrated by figure 1 for the second case (starting from an `OUTAGE` node).

When all nodes have been created by the process of Algorithm 2, a second pass takes place in order to create the arcs of the graph between all nodes that can be connected.

5.2 Weighting the arcs of a graph

The goal of this graph is to compute a production plan $\ell \in \mathscr{P}_i$ for power plant i with minimal reduced cost $\bar{c}_\ell$, which amounts to find according to equation (45) :

$$\min_{r,e,x_i^f} \sum_{k \in K} C_{ik} r_{ik} - C_i^f x_i^f - \sum_{w \in W} \mu_w e_w - u_i$$

where

- x_i^f is the quantity of remaining fuel at the end of the timeline in the aggregated scenario ;
- e_w is the energy produced at week w in production plan ℓ ;
- μ_w is the dual variable associated to constraint (33) ;
- u_i is the dual variable associated to constraint (34)

Let us recall that a production plan is supported by some unique path $\mathscr{A}_\ell \subset A_i$ in graph G_i, and that each campaign k in a production plan is carried either by a single `OUTAGE` arc, or by an `OUTAGE` arc followed by a `BO` arc.

It comes that the previous expression can be rewritten as:

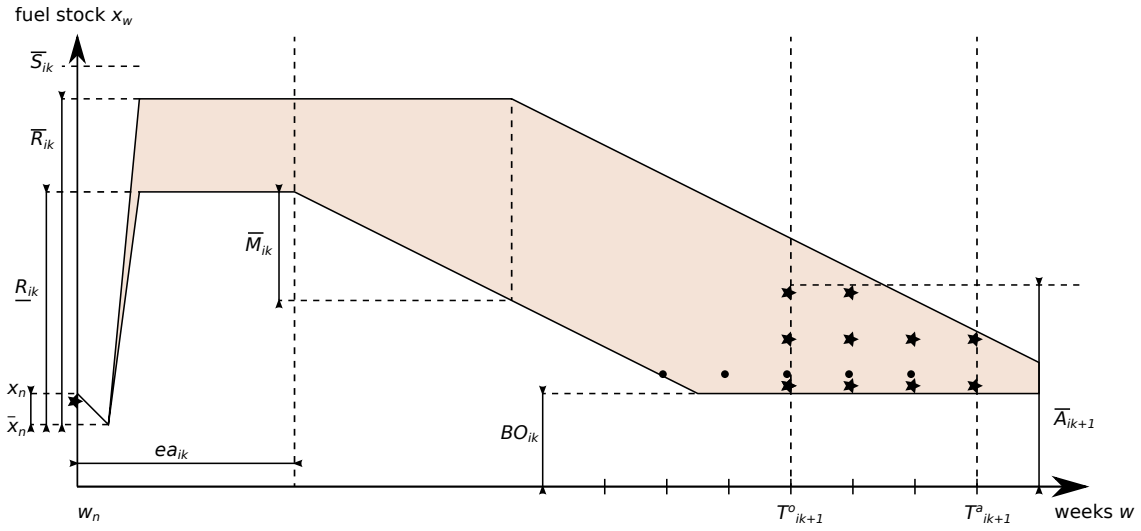


Fig. 1 Creating new nodes from an `OUTAGE` node n of campaign k in graph G_i inside the beam of possible stock intervals. Stars inside the beam correspond to the creation of new `OUTAGE` nodes associated with campaign $k+1$, whereas circles are new `BO` nodes associated with campaign k. Note that no `BO` node is created at week $T^a_{i(k+1)}$, because the paths that would go through such a node would violate constraints `[CT13bis]`.

$$\min \bar{c}_\ell = \sum_{a=(n,n')\in\mathscr{A}_\ell} \left(C_{ik_n} r_a - C_i^f x_a^f - \sum_{w\in W(a)} \mu_w e_w \right) - u_i$$

where:

- r_a is the refuel quantity at the beginning of arc a, i.e. :

$$r_a = \begin{cases} r_{ik_n} & \text{if } a \text{ is an } \texttt{OUTAGE} \text{ arc} \\ 0 & \text{if } a \text{ is a } \texttt{BO} \text{ arc} \end{cases}$$

- x_a^f is the remaining fuel at the end of the timeline if a is the final arc in path $\mathscr{A}_\ell$, i.e. :

$$x_a^f = \begin{cases} x_i^f & \text{if } type_{n'} = \texttt{FINAL} \\ 0 & \text{otherwise} \end{cases}$$

- $W(a) = [w_n, w_{n'}[$ is the set of weeks spanned by arc a.

With this rewriting, we see that each variable that plays a part in the reduced cost of path $\mathscr{A}_\ell$ is *uniquely linked* with a variable associated with some arc of ℓ.

Let us call the *reduced cost* of arc $a \in A_i$ the value c_a defined by:

$$\bar{c}_a = \min_{r,e,x_a^f} C_{ik_n} r_a - C_i^f x_a^f - \sum_{w\in W(a)} \mu_w e_w \tag{66}$$

While ensuring that constraints of (P) hold true for every path in ℓ, we can easily find the production plan with minimum reduced cost by solving a shortest path in the graph. The reduced cost of this production plan is obtained by substracting u_i from the minimum weight path value.

Determining the weight of a `BO` arc is straightforward because there is no refuel at the beginning of such an arc ($r_a = 0$) and the productions $e_{(W(a))}$ are fixed for each timestep by constraints `[CT6`, i.e. (57)`]`, if we disregard the allowed deviation ε. If the arc is `FINAL`, we can compute x_a^f by substracting the productions from its initial fuel stock. The weight of a `BO` arc a is given by the following expressions, where everything is known:

$$\bar{c}_a = \begin{cases} -C_i^f \cdot \left(x_n - \sum_{w\in W(a)} e_w\right) - \sum_{w\in W(a)} \mu_w \cdot e_w & \text{if } a \text{ is } \texttt{FINAL} \\ -\sum_{w\in W(a)} \mu_w \cdot e_w & \text{otherwise} \end{cases} \tag{67}$$

Determining the weight of a non final `OUTAGE` *arc* is more intricate because the allowed modulation gives some latitude to productions $e_{(W(a))}$. However, the campaign k_n, the production weeks $W(a) = [w_n, w_{n'}[$ and the levels of fuel $(x_n, x_{n'})$ at the beginning and at the end of the arc are known.

Following constraints (52), let $\bar{x}_n = \left(\frac{Q_{ik_n}-1}{Q_{ik_n}}(x_n - \mathtt{BO}_{ik_n}) + \mathtt{BO}_{ik_n-1}\right)$. The problem can then be stated as follows:

$$(LP_a) \quad \bar{c}_a = \min_{r_a, e_{(W(a))}} C_{ik_n} r_a - \sum_{w \in W(a)} \mu_w e_w \tag{68}$$

$$\text{s.t.} \quad \bar{x}_n + r_a = x_{n'} + \sum_{w \in W(a)} e_w \qquad [\mathtt{from(51)}] \tag{69}$$

$$0 \leq e_w \leq \overline{E}_{iw} \qquad \forall w \in W(a) \quad [\mathtt{from(47)}] \tag{70}$$

$$\underline{R}_{ik_n} \leq r_a \leq \overline{R}_{ik_n} \qquad [\mathtt{from(49)}] \tag{71}$$

$$\bar{x}_n + r_a \leq \overline{S}_{ik_n} \qquad [\mathtt{from(54)}] \tag{72}$$

$$\sum_{w \in W(a)} (\overline{E}_{iw} - e_w) \leq \overline{M}_{ik_n} \qquad [\mathtt{from(56)}] \tag{73}$$

We can use (69) to rewrite r_a and remove it from the problem. It comes that we are looking for the minimum under constraints (70) to (73) of:

$$\bar{c}_a = C_{ik_n}(x_{n'} - \bar{x}_n) + \sum_{w \in W(a)} (C_{ik_n} - \mu_w) e_w$$

As the first part of the sum is *known* for each arc, we only have to find the productions e_w that minimize $\sum_{w \in W(a)} (C_{ik_n} - \mu_w) e_w$. We tried two methods to achieve this goal:

The exact solution is obviously given by the Simplex method, since (LP_a) is a linear program, or by Algorithm 4 below, where *nullify* has to be understood as "set as close to zero as possible without violating constraints (70) to (73)":

Algorithm 4 Exact algorithm for minimizing arc weight

```
set all e_w to Ē_iw
order the set of weeks by decreasing values of (C_{ik_n} − μ_{w_m}) : w_1, w_2 ...
start with m = 1
while constraints (70) to (73) are not satisfied do
    nullify e_{w_m}
    increment m
end while
while (C_{ik_n} − μ_{w_m}) > 0 and constraints (70) to (73) can be satisfied do
    nullify e_{w_m}
    increment m
end while
```

However, we encountered two problems with the exact method:

- the overall column generation process was converging too fast, and no new column was generated after about two or three iterations ; this can be overcome however by the *diversification* strategy exposed in Section 7 ;
- this method was 3 to 10 times slower than the heuristic exposed below (depending on the instances), because it has to walk through the ordered set of weeks for each arc. Thus, on some instances, we were not able to produce enough columns in the imposed time and ended with no valid solution when using this exact algorithm.

We used a heuristic to obtain a good solution, even if not optimal: we chose to *spread* the modulation on all weeks proportionally to $\overline{E}_{iw}$, but with a different factor whether the *reduced unit cost* $\bar{c}_{ikw} = C_{ik} - \mu_w$ is positive or negative. For that purpose, we split the set of weeks into two sets: $W_k^+ = \{w \in W \text{ s.t. } C_{ik} - \mu_w > 0\}$ and $W_k^- = \{w \in W \text{ s.t. } C_{ik} - \mu_w \leq 0\}$. Then, when looking for the minimal reduced cost of our arc a, we consider two variables α_a and β_a in the $[0,1]$ interval and we *set* the values of e_w from these variables:

$$\forall w \in W(a) \cap W_{k_n}^+, \;\; e_w = \alpha_a \,.\, \overline{E}_{iw}$$
$$\forall w \in W(a) \cap W_{k_n}^-, \;\; e_w = \beta_a \,.\, \overline{E}_{iw}$$

The problem becomes: find the factors α_a and β_a that minimize the expression:

$$\alpha_a \left(\sum_{w \in W(a) \cap W_{k_n}^+} (C_{ik_n} - \mu_w) \overline{E}_{iw} \right) + \beta_a \left(\sum_{w \in W(a) \cap W_{k_n}^-} (C_{ik_n} - \mu_w) \overline{E}_{iw} \right)$$

The parenthesized expressions do not have to be computed for each arc. Instead, we can compute them *in advance* for each $(k, [w, w'[)_{w<w'}$ couple. Similarly, the constraints (70) to (73) can be rewritten as linear constraints on α_a and β_a and we save a lot of computing overhead by precomputing $\sum_{w \in [w,w'[} \cap W_k^- \overline{E}_{iw}$ for each couple $(k, [w, w'[)_{w<w'}$ and their counterparts on W_k^+.

As it reduces the number of variables to a couple, this heuristic has proven to be more efficient and effective than looking for the exact solution, when placed in the context of this column generation problem : even if the minimum reduced cost is not reached by this computation, the induced time savings allow the production of a greater set of columns amongst which better combinations can be found.

5.3 Obtaining dual variables μ_w and u_i

At each iteration we solved $RL\bar{F}$ to optimality and the dual variables μ_w and u_i are respectively associated with constraints (33) and (34). For the first run the variables are arbitrarily chosen as follows: $u_i = 0$ for all $i \in I$ and $\mu_w = \max_{i \in I, k \in K_i} \{C_{ik} + 1\}$ so that $W_k^+ = W$ for all power plants. This will create a first pool of production plans where nuclear power plants are used at their maximal capacity.

6 Solving the final master problem

Once $L\bar{F}$ has been solved to optimality by the Column Generation process, the following last restricted $R\bar{F}$ is solved to optimality

$$(R\bar{F}) \quad z(\bar{F}) = \min \sum_{i \in I} \sum_{\ell \in \widehat{\mathscr{P}_i}} c_\ell y_\ell + \sum_{w \in W} C_{jw} e_{jw} \tag{74}$$

$$\text{s.t.} \quad \sum_{i \in I} \sum_{\ell \in \widehat{\mathscr{P}_i}} y_\ell e_{\ell w} + \sum_{j \in J} e_{jw} = DEM_w \qquad \forall w \in W \tag{75}$$

$$\sum_{\ell \in \widehat{\mathscr{P}_i}} y_\ell = 1 \qquad \forall i = 1, \ldots, I \tag{76}$$

$$\underline{E}_{jw} \leq e_{jw} \leq \overline{E}_{jw} \qquad \forall j \in J, \forall w \in W \tag{77}$$

$$\sum_{i=1}^{I} \sum_{\ell \in \widehat{\mathscr{P}_i}} a_{\ell G} y_\ell \leq \alpha_G \qquad \forall G \in \mathscr{G} \tag{78}$$

$$y_\ell \in \{0,1\} \qquad \forall \ell \in \widehat{\mathscr{P}} \tag{79}$$

$$e_{jw} \geq 0 \qquad \forall j \in J, \forall w \in W \tag{80}$$

where $\widehat{\mathscr{P}}$ contains the subset of production plans generated from the beginning. This *MIP* includes constraints (78) which are the incompatibility constraints for some subsets $\widehat{\mathscr{P}}$ of production plans, corresponding to constraints `[CT14-21]` on the scheduling of outages. These constraints are generated dynamically during the column generation process. All of them are based on the same principle: given a set of columns, it is possible to build an incompatibility graph and we look for the maximal clique on this graph. The graph has a node for each column and an edge between any pair of incompatible columns (i.e. columns that cannot be at the same time in the optimal solution due to constraints `[CT14-21]`).

If $\ell \in \mathscr{P}_i$ denotes the column selected for each nuclear plant i in the optimal *MIP* solution, then the variables y_{iwk}, z_{itks} and ξ_{itks}, representing the dates of outages, the dates when the stock reaches the `BO`-level and the corresponding imposed decreasing profile for the following weeks, are fixed for the rest of the whole method.

7 Columns diversification

The main difficulty encountered with the method described so far comes when solving the *MIP* for column selection since constraints `[CT14-21]` are not part of the column generation process. We are not able to find a proper subset of columns that satisfies these constraints, most of the time. To overcome this problem, we used an alternate scheme for weighting the columns: instead of using weights $\bar{c}_a$ given by equations (67) and (68), we compute and use weights $\bar{c}'_a$ in the following way:

$$\bar{c}'_a = \begin{cases} 0 & \text{for } \texttt{BO} \text{ arcs} \\ nbConflicts(a, col) & \text{for } \texttt{OUTAGE} \text{ arcs} \end{cases} \tag{81}$$

where $nbConflicts(a, cols)$ heuristically figures out the current number of (column, constraint) couples that would conflict with a new column going through arc a.

More precisely, we compute $\bar{c}'_a$ with the following algorithm:

Algorithm 5 Alternate weight computation during a *diversification* pass for an `OUTAGE` arc $a = (n, n')$

$\bar{c}'_a \leftarrow 0$
for all existing column $\ell \in \cup_i \mathscr{P}_i$ **do**
  for all constraint m of type `[CT14]` to `[CT18]` **do**
    if column ℓ conflicts with an outage of power plant i_a starting at week w_n by constraint m **then**
      $\bar{c}'_a \leftarrow \bar{c}'_a + 1$
    end if
  end for
  for all constraint m of type `[CT19]` **do**
    if for some outage k of ℓ, (i_a, k_a) and (i_ℓ, k) are part of set G_m^{19} **and** w_n is interval $[ha_{i_\ell k} + w^m_{i_\ell k} - w^m_{i_a k_a}, ha_{i_\ell k} + w^m_{i_\ell k} - w^m_{i_a k_a} + i^m_{i_\ell k}]$ **then**
      $\bar{c}'_a \leftarrow \bar{c}'_a + \frac{1}{Q_m^{19}}$
    end if
  end for
  for all constraint m of type `[CT20]` concerning week w_m **do**
    if $w_m \in [w_n, w_n + L_{i_a k_a}] \cap [ha_{i_\ell k}, ha_{i_\ell k} + L_{i_\ell k}]$ for some outage k of column ℓ **then**
      $\bar{c}'_a \leftarrow \bar{c}'_a + \frac{1}{Q_m^{20}}$
    end if
  end for
end for
for all constraint m of type `[CT21]` concerning time period $[b_m, e_m]$ such that $i_a \in C_m^{21}$ **do**

$$\bar{c}'_a \leftarrow \bar{c}'_a + \max_{w \in [w_n, w_n + L_{i_a k_a}] \cap [b_m, e_m]} \left(\sum_{(\ell,k) \in \cup_i (\mathscr{P}_i \times K_i) \ / \ i_a \neq i_\ell \ \wedge \ i_\ell \in C_m^{21} \ \wedge \ w \in [ha_{i_\ell k}, ha_{i_\ell k} + L_{i_\ell k}] \cap [b_m, e_m]} \frac{\overline{P}_{i_\ell w}}{Q_m^{21}} \right)$$

end for

Note that $\bar{c}'_a$ only depends on the power plant i_a, the week of outage w_a and the campaign k_a ; furthermore, when new columns are generated, they only modify weights $\bar{c}'_a$ by an additive value, independently of other columns. These facts allow major factorizations in the algorithm.

This method is called each time negative reduced cost columns are found through finding the shortest path on the former graph. Then, when no more negative reduced cost columns are found with the shortest path algorithm, only the diversification method is processed and the minimum path algorithm is not ran unless zero reduced cost columns are added by diversification. Indeed, their addition could modify the value of the dual variables though not decreasing the value of the dual problem. Note that this diversification method creates positive reduced cost columns that are added if not already present.

8 Calculate the reload and the production quantities

Once the dates for outages and BO-profiles have been fixed as described in Section 6, it is still possible to improve the value of the optimal solution by looking for the optimal value of the remaining variables which are continuous. In other words, we have to determine the optimal reload quantities r_{ik} during each outage k for every nuclear plant i, and the optimal production values p_{jts} and p_{its} for non-nuclear and nuclear plants, and finally the optimal stock values p_{its} for time steps t and scenario s such that the stock level is over the `BO` threshold.

8.1 LP-solving for fixing reload quantities

The reload quantities r_{ik} are determined by solving a partially-disaggregated LP where every scenario s is considered instead of the average scenario $\bar{s}$, but time steps t are still aggregated into weeks w so as to accelerate solving. The optimal reload quantities r_{ik} are found by solving a linear program (LP_R) with CPLEX, where non-negative decision variables are reload variables r_{ik}, production variables e_{jws} and e_{iws}, and stock variables x_{iWs}. The constraints in this LP are all constraints `[CT1-12]`, but for weeks w instead of time steps t. Let us define the binary values $\eta^{\#}(i,w,k)$ and $Y^{\#}_{ik}$ as the values of $\eta(i,w,k)$ and Y_{ik} after they have been fixed at the end of Phase 1.

Since the two former values are known, now the outage periods have been fixed.

Let us also define the energy demand for week $w \in W$ for scenario $s \in S$ as

$$DEM_{ws} = \sum_{t \in T_w} D^t . DEM_{ts}$$

and the maximum production for power plant $i \in I$ over week $w \in W$ as

$$\overline{E}_{iw} = \sum_{t \in T_w} \overline{P}^t_i . D^t$$

and the maximum and minimum productions for power plant $j \in J$ and scenario $s \in S$ over the week $w \in W$ as

$$\overline{E}_{jws} = \sum_{t \in T_w} \overline{P}_{jts} . D^t$$

and

$$\underline{E}_{jws} = \sum_{t \in T_w} \underline{P}_{jts} . D^t$$

and

$$C_{jws} = \frac{|W|}{|T|} \sum_{t \in T_w} C_{jts}$$

the mean cost of thermal energy production over a week $w \in W$ for $j \in J$ and $s \in S$.

The objective function is thus the following:

$$z(LP_R) = \min \sum_{i \in I} \sum_{k \in K} C_{ik} r_{ik} + \frac{1}{|S|} \sum_{s \in S} \left(\sum_{w \in W} \sum_{j \in J} C_{jws} e_{jws} - \sum_{i \in I} C^f_i x^f_{is} \right)$$

$$\text{s.t.} \quad \sum_{i \in I} \sum_{k \in K_i} e_{ikws} + \sum_{j \in J} e_{jws} = DEM_{ws}, \qquad \forall w \in W, \forall s \in S \qquad (82)$$

$$e_{iws} \leq \overline{e}_{iw} \sum_{k \in K_i} \eta^{\#}(i,w,k), \qquad \forall i \in I, w \in W, \forall s \in S \qquad (83)$$

$$\underline{E}_{jws} \leq e_{jws} \leq \overline{E}_{jws}, \qquad \forall j \in J, w \in W, s \in S \qquad (84)$$

$$\underline{R}_{ik} Y^{\#}_{ik} \leq r_{ik} \leq \overline{R}_{ik} Y^{\#}_{ik} \qquad \forall i \in I, k \in K_i \qquad (85)$$

$$x^1_{i(k+1)s} = x^2_{iks} - \sum_{w \in W'_{ik}} e_{iws} \qquad \forall i \in I, k \in K_i s \in S \qquad (86)$$

$$x^2_{iks} = \left[\left(\frac{Q_{ik} - 1}{Q_{ik}} \right) (x^1_{iks} - \mathtt{BO}_{i(k-1)}) + r_{ik} + \mathtt{BO}_{ik} \right] Y^{\#}_{ik} + x^1_{iks}(1 - Y^{\#}_{ik}) \qquad \forall i \in I, k \in K_i, s \in S \qquad (87)$$

$$x^1_{iks} \leq A_{ik} + (1 - Y^{\#}_{ik})N \qquad \forall i \in I, k \in K_i, s \in S \qquad (88)$$

$$x^2_{iks} \leq S_{ik} + (1 - Y^{\#}_{ik})N \qquad \forall i \in I, k \in K_i, s \in S \qquad (89)$$

$$x^f_{is} = \sum_{k \in K_i} x_{i(k+1)s}(Y^{\#}_{ik} - Y^{\#}_{i(k+1)}) \qquad \forall i \in I, s \in S \qquad (90)$$

$$\sum_{w \in W'_{ik}} (\overline{E}_{iw} - e_{iws}) \leq \overline{M}_{ik} \qquad \forall i \in I, k \in K_i, s \in S \qquad (91)$$

$$e_{jws}, e_{ikws}, r_{ik}, x^1_{iks}, x^2_{iks}, x^f_{is} \geq 0 \qquad (92)$$

This linear problem LP_R is solved for the different combinations of columns that produce the optimal value of the restricted master problem $R\bar{F}$ and the combination that gives the lowest value is kept to go to the next step where the original problem will be solved with outage periods and reload quantities fixed.

8.2 Disaggregate LP-solving for every scenario and initial time steps

At this final step, for every scenario $s \in S$, we go back to time steps t. Therefore, S Linear Programs (LP_s) are solved where decision variables are:

(i) the production variables p_{jts} and p_{its} for thermal and nuclear plants, respectively,
(ii) the stock variables x_{its} for time steps t such that the stock level is over the `BO` threshold (for the other time steps, the stock level is already fixed as imposed by the decreasing profile)

Constraints are the same as in the previous step, i.e. constraints `[CT1-12]`, but for time steps t now, i.e. original constraints `[CT1-12]`. The objective functions are the total production cost for each scenario s, reduced by the objective value of remaining fuel:

$$z(LP_s) = \min \sum_{t \in T} \sum_{j \in J} C_{jts} p_{jts} D^t - \sum_{i \in I} C^f_i x^f_{is}$$

The final objective value z is the refueling cost and the average total production cost reduced by the objective value of remaining fuel over all S scenarios:

$$z = \min \sum_{i \in I} \sum_{k \in K_i} C_{ik} r_{ik} + \frac{1}{|S|} \sum_{s \in S} \left(\sum_{t \in T} \sum_{j \in J} C_{jts} p_{jts} D^t - \sum_{i \in I} C^f_i x^f_{is} \right) \qquad (93)$$

9 Computational Results

All algorithms described in this paper were coded in C++ and compiled with the gcc (Debian 4.4.4-8) 4.4.5 compiler. CPLEX 12.1 was used as the LP solver and as the integer programming solver. The experiments were performed on a Squeeze-x64-64 (image based on Debian version sid for AMD64/EM64T) Bi AMD Opteron Dual core 3,2 Ghz with 16 GB of physical memory and 16 GB of swap memory.

9.1 Instances

In our computational experiments, we considered 16 instances given by EDF during the Challenge (see [12]). Almost all of them ask for a planification of 6 campaigns for a time period of 5 years. The instances have been partitioned into the following three classes.

Class 1. A set of 6 instances (A0-A5) used for the qualification phase. In these instances the number of different scenarios is at most 30 for about 20 nuclear plants and 30 other thermic power plants. The time step is the day and there are less than 27 constraints linking the nuclear power plants.

Class 2. A set of 5 instances (B6-B10) used for the second phase. In these instances the number of different scenarios considered can rise up to 121. There are about 50 nuclear plants and 20 other thermic power plants. The time step is 8 hours and there are about 140 constraints linking the nuclear power plants. There are between 114 and 235 constraints `[CT13]` on the dates of the outages.

Class 3. A set of 5 instances (X11-X15) used for the final ranking with the 5 former instances. They were not released until the end of the challenge. In these instances the number of different scenarios are 50 or 121. There are about 50 nuclear plants and 20 other thermic power plants. The time step is 8 hours and there are about 140 constraints linking the nuclear power plants. There are between 207 and 260 constraints `[CT13]` on the dates of the outages, which is the only noticeable difference with the *class 2* instances.

9.2 Results and discussion

In table 1, the best results that were obtained and published on the website of the Challenge are given in the first column whereas the second column displays the ones obtained by the current method. All teams were ranked by the Challenge organizers and with the former results our method ranked 4^{th} although we were penalized by the fact that the method was not able to give answers for three of the instances. It is interesting to point out that among the first five methods proposed, our method is the only one, as far as we know, having some mathematical programming foundations.

Data number	ROADEF best found results	our ROADEF results	best found results since	ρ factor
A0	8.73099e+12	8.73623e+12	8.73696e+12	0.9
A1	1.69538e+11	1.70030e+11	1.69902e+11	0.449
A2	1.46048e+11	1.46409e+11	1.46435e+11	0.9
A3	1.54430e+11	1.54711e+11	1.54747e+11	0.9
A4	1.11591e+11	1.12808e+11	1.12528e+11	0.45
A5	1.25822e+11	1.28366e+11	1.26932e+11	0.449
B6	8.34247e+10	8.47093e+10	8.46439e+10	0.5
B7	8.11742e+10	8.18391e+10	8.18501e+10	0.45
B8	8.19262e+10	8.29770e+10	8.29548e+10	0.449
B9	8.17509e+10	8.36228e+10	8.29290e+10	0.5
B10	7.77670e+10	7.87302e+10	7.88065e+10	0.449
X11	7.91168e+10	TIMEOUT	8.00418e+10	3.0
X12	7.75899e+10	7.81931e+10	7.81252e+10	0.45
X13	7.64492e+10	INFEASIBLE	7.67848e+10	0.45
X14	7.61730e+10	TIMEOUT	7.69035e+10	0.449
X15	**7.51014e+10**	**7.51014e+10**	7.49233e+10	0.449

Table 1 Results of the ROADEF challenge

The fourth column gives the best results we found since then, after the few bugs were corrected. The last column gives the parameter ρ that was chosen to obtain the former result, i.e., the parameter used to build the graphs to solve the $L\bar{F}$.

Tables 2 and 3 show several results about the method:

- The first column specifies the considered instance ;
- The second column shows the optimal result of the $RL\bar{F}$;

- The third column gives the iteration at which the former value is reached: the following iterations do not decrease the former value ;
- The fourth column shows the total number of iterations of the algorithm ;
- The fifth column gives the final number of plans that have been computed at the end ;
- The sixth column gives the final number of constraints `[CT14-21]` that have been added to the *MIP* ;
- The seventh column gives the value of *MIP* when it is called at the end of the column generation process ;
- The eigth column gives the value of LP_R ;
- The ninth column gives the result of z defined in equation (93) ;
- The last column gives the elapsed time for the whole algorithm

Data	$RL\bar{F}$	It to LB	Tot It	Pl Nb	Cs Nb	*MIP*	LP_R	z	Time (s)
A0	8.73548e+12	4	24	32	40	8.73696e+12	8.73696e+12	8.73696e+12	0
A1	1.52339e+11	8	149	540	1106	1.52489e+11	1.68891e+11	1.69978e+11	11
A2	1.45276e+11	18	449	2111	1947	1.45745e+11	1.45747e+11	1.46435e+11	224
A3	1.52835e+11	8	764	1 929	1753	1.53190e+11	1.54055e+11	1.54747e+11	34
A4	1.02524e+11	18	461	2617	2626	1.02974e+11	1.11857e+11	1.12558e+11	157
A5	1.19655e+11	18	521	3226	3020	1.20145e+11	1.26698e+11	1.27744e+11	165
B6	7.75175e+10	22	761	6642	5630	7.90051e+10	8.16399e+10	8.48905e+10	287
B7	7.48221e+10	348	1094	8437	6105	7.57638e+10	7.87608e+10	8.21897e+10	896
B8	7.35414e+10	22	406	3256	7497	7.60027e+10	7.95534e+10	8.32061e+10	366
B9	7.31604e+10	28	372	3630	7477	7.56930e+10	7.96609e+10	8.34030e+10	512
B10	7.03327e+10	36	3075	11157	5944	7.13487e+10	7.52558e+10	7.90224e+10	1053
X11	7.34759e+10	40	1239	9061	5719	7.45279e+10	7.71169e+10	8.00708e+10	1533
X12	7.17523e+10	44	1958	10869	5640	7.22878e+10	7.49525e+10	7.82294e+10	737
X13	6.98451e+10	34	409	6299	6687	7.07385e+10	7.39050e+10	7.69910e+10	501
X14	6.92277e+10	40	611	7961	6911	6.99999e+10	7.36125e+10	7.69212e+10	1112
X15	6.73328e+10	44	935	10026	5955	6.78149e+10	7.13536e+10	7.50613e+10	676

Table 2 Performance of the overall method with $\rho = 0.9$

Data	$RL\bar{F}$	It to LB	Tot It	Pl Nb	Cs Nb	*MIP*	LP_R	z	Time (s)
A0	8.73548e+12	4	24	32	40	8.73698e+12	8.73698e+12	8.73698e+12	0
A1	1.52342e+11	8	58	269	1072	1.52536e+11	1.68836e+11	1.69962e+11	7
A2	1.45284e+11	20	391	1614	1931	1.45769e+11	1.46151e+11	1.46840e+11	286
A3	1.52830e+11	8	776	1735	1694	1.53188e+11	1.54089e+11	1.54777e+11	33
A4	1.02543e+11	22	625	2687	2632	1.03002e+11	1.12506e+11	1.13198e+11	217
A5	1.19685e+11	18	367	2903	3059	1.20247e+11	1.27393e+11	1.28352e+11	173
B6	7.75271e+10	22	740	6283	5637	7.88730e+10	8.15095e+10	8.47410e+10	465
B7	7.48317e+10	38	661	9574	6130	7.55816e+10	7.85357e+10	8.19583e+10	936
B8	7.36053e+10	20	350	3311	7401	7.76852e+10	8.13098e+10	8.51822e+10	355
B9	7.31967e+10	30	1419	6187	7544	7.53907e+10	7.90747e+10	8.27811e+10	607
B10	7.03706e+10	34	3320	11417	5928	7.13445e+10	7.51940e+10	7.89976e+10	762
X11	7.34802e+10	38	1232	9508	5690	7.45335e+10	7.70939e+10	8.00418e+10	1563
X12	7.17785e+10	50	975	10821	5687	7.22479e+10	7.48730e+10	7.81379e+10	905
X13	6.98541e+10	24	535	6529	6667	7.08168e+10	7.38722e+10	7.68983e+10	575
X14	6.92260e+10	36	713	8243	6870	6.99515e+10	7.36255e+10	7.69829e+10	1122
X15	6.73713e+10	40	1032	11123	5977	6.78117e+10	7.12779e+10	7.49589e+10	690

Table 3 Performance of the overall method with $\rho = 3$

The value of the relaxation of the master problem decreases as negative reduced cost columns are added. As the lowest value is reached in a few dozens of iterations and the columns added afterwards do not change this value, the optimal value of the restricted linear $\bar{F}$ ($RL\bar{F}$) is reached then. The number of columns generated is smaller than $|I| \times$ Tot It and as expected it decreases with the number of iterations.

Note that the number of lines (Cs Nb) and variables (Pl Nb) has the same order of magnitude. It means that both the rows and columns generation processes are necessary to find quickly the integer solution of the $RL\bar{F}$.

The value of $RL\bar{F}$ is always lower than the value of *MIP* (i.e., $R\bar{F}$) as expected since $z(RL\bar{F}) \leq z(R\bar{F})$, but it is not true that $z(RL\bar{F}) \leq z(F) \leq z$ since it is not a lower bound for the original problem.

Nevertheless, it interesting to note that the value of the objective function reported in column *MIP* is always smaller than the one reported in column LP_R which is always smaller than the value reported in the column z. It is clear that this is due to the approximations that have been done for solving $\bar{F}$. The increase in the objective function between *MIP* and (LP_R) can be explained by the fact that *MIP* uses an average scenario instead of the whole set of scenarios. Therefore, when problems (LP_R) are solved with the dates of outage and `BO` fixed, can happen that, for satisfying scenarios with a demand greater than the average, it is necessary to use more expensive source of energy.

Less intuitive is the explanation of why the values reported in LP_R is smaller than z. This difference is due to the modulation. In fact, when the model considers the demand aggregated over a week it applies a certain quantity of modulation. Nevertheless, can happen that when the time interval considered is the time step it is impossible to maintain the same modulation and therefore more expensive source of energy must be used for satisfying the excess of demand.

Table 4 gives CPU time in seconds of the method:

- The first column specifies the considered instance ;
- The second column shows the time to read the input and to write the output ;
- The third column gives the time to build the graph for the pricing problem ;
- The fourth column shows the total time of column generation ;
- The fifth column gives the time to solve the integer master problem obtained at the end of the column generation process ;
- The sixth column gives the time of the reload phase ;
- The seventh column gives the time of the production phase ;
- The last column gives the elapsed time for the whole algorithm

Data	Read/Write	Graph	CG	MIP	Reload	Production	Total
A0	0	0	0	0	0	0	0
A1	1	0	1	5	3	1	11
A2	4	173	15	12	19	1	224
A3	3	1	6	8	15	1	34
A4	8	95	12	12	28	2	157
A5	7	26	76	25	28	3	165
B6	56	5	27	93	94	12	287
B7	52	18	122	543	152	9	896
B8	60	48	94	101	45	18	366
B9	143	66	164	52	65	22	512
B10	136	8	104	576	209	20	1053
X11	48	537	152	577	199	10	1533
X12	53	11	76	434	155	8	737
X13	143	126	57	102	42	31	501
X14	139	76	129	643	104	21	1112
X15	132	8	54	193	267	22	676

Table 4 CPU Time (s)

Actually the whole method never reaches the set time limit (3600 seconds). It can be used, for example, as a very good initial solution which could be a starting point for a local search procedure. The most time-consuming phase is the *MIP* phase at the end of the column generation process. The reload phase is more time consuming than the production phase due to the multiple resolutions for the different combinations of columns that produce the *MIP*. We can notice that the time taken by the input and output is not negligible due to the large size of the files to read and write.

10 Conclusion

We presented a mathematical formulation for the EDF (Électricité De France) electricity production planning problem submitted as subject of the ROADEF/EURO 2010 Challenge, and a heuristic method based on column generation. To the best of our knowledge this is the first ranked method using mathematical formulation among the final results of the Challenge. The method proceeds in two stages : first we fix dates for outages for each nuclear plant on aggregated data, and in a second step we determine the reload and production quantities for each plant and each scenario. Computational results show the effectiveness of the proposed approach, given that the method was far from using all the time allowed in the Challenge. Indeed, part of the time is kept for solving the (LP_R) and the (LP_s) problems which may be time consuming. Thus, the first stage of the method is stopped after a time limit which has been experimentally set to 30 minutes. Nevertheless, in some case it stops before, since no more columns are produced.

Amongst possible future improvements of the proposed approach and therefore future directions of research, it could be interesting, once a real lower bound is obtained, to add more positive reduced cost columns which are in the gap. The current shortest path solving algorithm is not suited to that, since it produces columns with minimal reduced costs. A further improvement could consist in generating more sets of schedules for the outages using the local search on the set of outage dates found in the first phase of the method. Last but not least, it would be interesting to improve the method by entering the clique-like constraints (36) into the pricing problem. Therefore the associated dual variables could be included in the pricing subproblem. Introducing them could improve the number of columns found by the current shortest path method and alleviate the need for our diversification algorithm.

Acknowledgment

We wish to thank anonymous reviewers for fruitful suggestions which help improve a previous version of this paper. Moreover, we would like to thank Aristide Mingozzi and Guillaume Turri for their useful help and/or comments they gave at the beginning of this project.

References

1. Roberto Baldacci, Nicos Christophides, and Aristide Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Math. Program.*, pages 351–385, 2008.
2. Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1996.
3. Guy Desaulniers, Jacques Desrosiers, and Marius M.Solomon; GERAD. *Column Generation*. Springer, 2005.
4. Louis Dubost, Robert Gonzalez, and Claude Lemaréchal. A primal-proximal heuristic applied to the French unit-commitment problem. *Math. Program.*, 104(1):129–151, September 2005.
5. Erlon C. Finardi, Edson L. da Silva, and Claudia Sagastizàbal. Solving the unit commitment problem of hydropower plants via lagrangian relaxation and sequential quadratic programming. *Computational & Applied Mathematics*, 24:317 – 342, 12 2005.
6. Antonio Frangioni, Claudio Gentile, and Fabrizio Lacalandra. Solving unit commitment problems with general ramp constraints. *International Journal of Electrical Power; Energy Systems*, 30(5):316 – 326, 2008.
7. Josef Kallrath, Panos M.Pardalos, Steffen Rebennack, and Michel Scheidt. *Optimization in the Energy Industry*. 2009.
8. S.A. Kazarlis, A.G. Bakirtzis, and V Petridis. A genetic algorithm approach to solve the unit commitment problem. *IEEE Transactions on Power Systems*, 11:83 – 92, February 1996.
9. Mohand I. Khemmoudj, Marc Porcheron, and Hachémi Bennaceur. When constraints programming and local search solve the scheduling problem of edf nuclear power plant outages. *12th Internationnal Conference on Principles and Practice of Constraint Programming*, pages 271–283, 2006.
10. Marco E. Lubbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, pages 1007–1023, November-December 2005.
11. Richard Lusby, Laurent Flindt Muller, and Bjorn Petersen. A solution approach to the roadef/euro 2010 challenge based on benders decomposition. Technical report, November 2010.
12. Marc Porcheron, Agnès Gorge, Olivier Juen, Tomas Simovic, and Guillaume Dereu ; EDF R&D. Challenge roadef/euro 2010 :a large-scale energy management problem with varied constraints. February 2010.
13. Albert Wagelmans, Stan van Hoesel, and Antoon Kolen. Economic lot sizing: An o(n log n) algorithm that runs in linear time in the wagner-whitin case. *Operations Research*, 40:145–156, January-February 1992.
14. Harvey M. Wagner and Thomson M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5:89 – 96, October 1958.
15. Laurence A. Wolsey. *Integer Programming*. 1998.