Editorial Manager(tm) for Journal of Scheduling
Manuscript Draft

Manuscript Number:

Title: Solving eletricity production planning by column generation

Article Type: Maintenance Scheduling

Keywords: Energy; electricity production; column generation

Corresponding Author: Roberto Wolfler Calvo

Corresponding Author's Institution: University of Paris Nord

First Author: Antoine Rozenknop

Order of Authors: Antoine Rozenknop;Roberto Wolfler Calvo;Laurent Alfandari;Daniel Chemla;Lucas Létocart

Manuscript
Click here to download Manuscript: WolflerS23ChallengeRoadef.tex Click here to view linked References

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 the column generation method used by the authors to solve the EDF (Electricité De France) electricity production planning problem of the ROADEF/EURO 2010 Challenge, which is to our knowledge the first-ranked method among column generation methods proposed in the Challenge, and was ranked 4^{th} overall. The problem consists in determining, for the set of all thermal power plants of the French electricity company, a schedule of plant outages, which are necessary for refueling and maintenance operations, and for each plant, a production plan over the whole time horizon. The average cost of the global outage and production planning, computed over a set of demand scenarios, is to be minimized. The method proceeds in two main stages. In the first stage, which is the most time-consuming, dates for outages are fixed for each nuclear plant by transforming the initial problem into an aggregate set partitioning-like master problem with a single average scenario and reduced time steps. This reformulated problem is solved by column generation, where the pricing problem associated with each nuclear plant is a shortest path problem in an appropriate graph. Then, in the second stage the reload level is determined at each date of an outage, considering now all scenarii. Finally, the production quantities between two outages are optimized for each plant and each scenario by solving independent Linear Programs.

Keywords: Energy, electricity production, outage, column generation, shortest path, linear programming.

1 Introduction

Optimization in the energy sector is a large subject. With the raise of raw materials prices and the growing awareness of environment concerns, energy consumption must be thought in a less greedy way in order to save resources. For years several researchers have worked on that topic as gathered in book [6]. The increasing part of renewable energies in the global energy production enhances this trend. We report here a sample of works within that huge area of research. Bruglieri and Liberti in [3] propose a model to improve the efficiency of biomass energy production plants. Mustakerov and Borissova in [10] give tools to help to optimally deploy a grid of wind turbines on a park knowing the wind direction. Finally several works have also been done on optimally planning energy production when several means can be used to produced it. This has been done with both a heuristic method in [5] by Gupta *et al.* and an exact method by Morais *et al.* in [9].

Energy production planning is one of the most studied topics because the high volume effect often leads to achieve large cost improvements through optimization. Indeed in most countries energy can be produced through

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

different ways: nuclear and other thermal energies provide 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 waterpower). Producers can make a trade-off between these different production resources in order to satisfy the energy demand while decreasing their costs and/or their impact on the environment. This subject has several difficulties, one of which is that the exact amount of the future demand and the exact price of raw materials are unknown. Moreover, there are many constraints on the energy production plants: for example, the level of water in dams has to stay above a certain threshold or maintenance operations have to be executed periodically on nuclear plants, periods during which they are not producing. This problem that has been studied for years now ([7] for instance) was the starting idea 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. This subject is focused on thermal power plants and especially on those that have to be periodically shut down for refueling and maintenance, e.g. nuclear ones. The research project consists of 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. In medium-term electricity management 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), which leads to consider multiple scenarios and ask 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 produce 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 scenarii. In this paper a column generation method (see for example in [4] for a survey 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*) (see [8] for a set of problems where column generation was applied). The column generation resolution method iteratively creates two different problems: a *Restricted Master Problem* and a *Pricing Subproblem*. A *Restricted Master Problem* is the Master Problem where instead of solving the problem on all its solution space, only a subset of variables (or columns) is considered. If the decision variables are integer, these constraints are relaxed to have continuous decision variables. Each restricted master problem is solved to optimality and iteratively enriched with new columns output from the resolution of a *Pricing Subproblem* which uses as input data the optimal dual variables of the restricted master problem. The *Pricing Subproblem* aims at looking for new columns with negative reduced cost (associated with the former dual variables), that if added to the master problem would improve its optimal value (except in case of degeneracy). 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. The method stops when no more negative reduced cost column is output from the pricing subproblem, the linear relaxation of the original master problem is solved then 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 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).

The approach developed in this paper is a column generation method, where only a subset of the whole set of possible columns are implicitely considered due to the time limit imposed by the competition. Several approximations have been done in the Pricing Subproblem and therefore the optimality of the final solution is not guaranteed. The main contributions of this paper are the following: original set partitioning-like reformulation of the problem, decomposition of the whole problem in two subproblems (master and slave), pricing subproblem solving based on shortest path computations in a particular state graph, since the slave problem is close to the Uncapacitated Lot Sizing problem, diversification method obtained by changing the objective function and adding different columns which may have a positive reduced cost. Moreover, the method is not limited to the idea of using columns generation but also uses rows generation. The discriminant option chosen for our approach is to spend time to generate a high-quality schedule of outages, as guided by the cost objective of column generation.

This paper is organized as follow. The electricity production planning problem is presented in Section 2. Section 3 describes the overall method used to solve it. Section 4 gives the model of the master problem that is solved for fixing dates of outages and critical stock-level periods as first stage. Section 5 describes the pricing

subproblem and its resolution method. Section 6 shows to improve the pricing subproblems in order to generate a greater number of columns assuring diversity. Section 7 exposes the second stage of the method for fixing reload and production quantities for each plant. Section 8 gathers the results obtained on the test instances.

2 Problem statement

The model proposed by the EDF company is the following. For sake of consistency, the notations used in the ROADEF/EURO 2010 Challenge document [11] are almost totally kept in this paper. The various indexes used are the index of scenarios $s = 0,\ldots,S-1$, the index of time steps $t = 0,\ldots,T-1$, the index of weeks $h = 0,\ldots,H-1$, the index of non-nuclear plants $j = 0,\ldots,J-1$, the index of nuclear plants $i = 0,\ldots,I-1$ and the index $k = 1,\ldots,K-1$ of production cycles for nuclear plants, $K-1$ being the maximum number of cycles that can start during the considered period. The decision variables are:

- $ha(i,k)$: week of cycle k during which nuclear plant i goes offline (i.e. week of decoupling)
- $p(j,t,s)$: production of non-nuclear plant j during time step t over scenario s
- $p(i,t,s)$: production of nuclear plant i during time step t over scenario s
- $r(i,k)$: reload performed during the outage of cycle k of nuclear plant i
- $x(i,t,s)$: stock of fuel of nuclear power plant i at the beginning of time step t over scenario s
- $ec(i,k)$: set of time steps composing the production campaign of cycle k of nuclear plant i
- $ea(i,k)$: set of weeks composing the outage of cycle k of plant i

The objective function is:

$$\min \sum_i \sum_k C_{ik} r(i,k) + \frac{1}{S} \sum_s \Big(\sum_t \sum_j C_j^{ts} p(j,t,s) D^t - \sum_i C_{i,T+1} x(i,T,S) \Big) \tag{1}$$

where C_{ik} is the proportional cost of fuel during cycle k for nuclear plant i, C_j^{ts} is the proportional production cost for time t and scenario s for non-nuclear plant j, $C_{i,T+1}$ is the selling price (or the gain) for the remaining fuel and D^t is the length of time step t.

As the framework of this paper is column generation, we decompose the set of constraints into the *global* constraints that link different plants i and i', $i \neq i'$, and the *local* constraints that are specific to a single plant i.

2.1 Global constraints linking different plants

The global constraints that have been used for writing the set partitioning-like formulation of the problem are the following:

(i) Demand satisfaction:

$$\text{CT1} \quad \sum_j p(j,t,s) + \sum_i p(i,t,s) = DEM^{t,s} \quad \forall s,t$$

(ii) Constraints on outages of different plants i and i', concerning variables $ha(i,k)$:

- CT14 Minimum spacing and maximum overlapping between outages of i and i'
- CT15 Minimum spacing and maximum overlapping between outages between a specific period
- CT16 Minimum spacing between decoupling dates
- CT17 Minimum spacing between coupling dates
- CT18 Minimum spacing between coupling and decoupling dates
- CT19 Resource constraints on outages
- CT20 Maximum number of overlapping outages during a week
- CT21 Maximum offline power capacity of a set of power plants during a time period

All constraints [CT14-CT21] involving variables of dates of outages $ha(i,k)$ are not detailed in this paper. Nevertheless, they have been used for improving the column generation process and they have been used for the row generation process. We rather report the local constraints on production and stock as we shall refer to them in the definition of the pricing subproblem and its associated graph.

2.2 Local constraints holding for each plant

The local constraints that will belong to the pricing subproblem associated with each *nuclear* plant in the column generation scheme are the following:

CT3 Offline power

CT4 Minimum power

CT5 Maximum power before activation of imposition of power profile constraint

$$(t \in ec(i,k)) \text{ and } x(i,t,s) \geq BO_{ik} \Rightarrow p(i,t,s) \leq PMAX_i^t \quad \forall s,t,i,k$$

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

CT6 Maximum power after activation of imposition of power level constraint :

$$\begin{aligned} \forall s,t,i,k, \quad & \text{if} \quad t \in ec(i,k)\ x(i,t,s) < BO_{ik} \text{ then} \\ \{ \quad & \text{if} \quad x(i,t,s) \geq (PB_{ik}(x(i,t,s)) \times PMAX_i^t) \times D^t \\ & \text{then} \ \frac{p(i,t,s)}{PB_{ik}(x(i,t,s)) \times PMAX_i^t} \in [1-\varepsilon, 1+\varepsilon] \\ & \text{else} \ \ p(i,t,s) = 0\} \end{aligned}$$

where $PB_{ik}(x(i,t,s))$ 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:

$$RMIN_{ik} \leq r(i,k) \leq RMAX_{ik}$$

CT8 Initial fuel stock : $x(i,0,s) = XI_i$

CT9 Fuel stock variation during a production campaign

$$\forall s,t,i,k, \quad t \in ec(i,k) \Rightarrow x(i,t+1,s) = x(i,t,s) - p(i,t,s) \times D^t$$

CT10 Fuel stock variation during an outage : $\forall s,t,i,k$ if t is the first time step of $ea(i,k)$ then

$$x(i,t+1,s) = ((Q_{ik}-1)/Q_{ik})(x(i,t,s) - BO_{i,k-1}) + r(i,k) + BO_{ik}$$

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

CT11 Fuel stock variation during an outage

$$\forall s,t,i,k, \quad t \text{ is the first time step of } ea(i,k) \Rightarrow x(i,t,s) \leq AMAX_{ik},\ x(i,t+1,s) \leq SMAX_{ik}$$

where $AMAX_{ik}$ (resp. $SMAX_{ik}$) 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

$$\forall s,i,k \quad \sum_{t \in ec(i,k) x(i,t,s) \geq BO_{ik}} (PMAX_i^t - p(i,t,s)).D^t \leq MMAX_{ik}$$

where $MMAX_{ik}$ 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

CT13bis No outage $k = 0, \ldots, K-1$ could be skipped as soon as it has a last possible week $TA_{i,k}$ defined.

For non-nuclear plants, we only have minimum and maximum production constraints:

$$\text{CT2} \quad PMIN_j^{ts} \leq p(j,t,s) \leq PMAX_j^{ts} \quad \forall t,s,j$$

where $PMIN_j^{ts}$ and $PMAX_j^{ts}$ are the minimum and maximum production for non-nuclear plants j on time step t for scenario s.

3 The solving approach

As the combinatorial complexity of the problem is mainly due to the binary variables $ha(i,k)$ representing dates for outages, we fix these decision variables as a first stage. We also fix the periods when stock reaches a critical level equal to BO_{ik} (if such periods exist for a given nuclear plant) and the corresponding imposed decreasing profile for the successives periods in order to avoid to deal explicitly with non linear constraints [CT6]. Therefore, the only decisions that remain to be made in further steps are the reload and production variables, which are easier to settle once the binary outage variables have been fixed. This fixing, once for all, of outage variables $ha(i,k)$ is guided by the cost objective, solving by Column Generation (CG) a set partitioning-like reformulation of the problem. The reload and production variables are fixed for each plant in a second stage.

The two-stage method is stated below.

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

substage 1 Solve an aggregate set partitioning-like formulation by column generation

- Aggregate data passing from $(T,S) = (5000 \times 500)$ to $(H,\bar{s}) = (260 \times 1)$
- 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 master problem and solve it

substage 2 Give the final Master problem to a MIP solver adding all the clique inequalities [CT14,...,CT21] exploiting the fact that only a small number of nuclear power plants can be refueled at the same time. Outputs are dates for outages and *BO* stock-level periods.

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

Stage 2 will be described in Section 7. Sections 4 and 5 detail the Stage 1 of the method. In Stage 1, for every nuclear plant $i = 0,\ldots,I-1$, two decisions are fixed: dates for outages respecting the scheduling constraints on outages [CT13-CT21] and periods when stock reaches the BO_{ik} level, with imposed a non linear decreasing power profile on the following periods. This step is the most-consuming step of the method.

The efficiency of this column generation approach largely depends on the tractability of the subproblems associated with the I nuclear plants, which are of very large scale given the number S of scenarios and the number T of time steps. The option we chose to get round this complexity consists in simplifying the original set of data by aggregating the S scenarios into a single scenario $\bar{s}$ and the time steps into weeks. We first simplify the problem by considering weeks $h = 0,\ldots,H-1$ instead of time steps $t = 0,\ldots,T-1$, and we note $\bar{s}$ the *average* scenario, we thus note for every week h:

$$DEM^{h,\bar{s}} = \frac{1}{S}\sum_{s=0}^{S-1}\sum_{t\in week\ h} D^t.\ DEM^{t,s}$$

$$C_j^{h,\bar{s}} = \frac{1}{S}\sum_{s=0}^{S-1}\frac{H}{T}\sum_{t\in week\ h} C_j^{t,s}$$

$$EMIN_j^{h,\bar{s}} = \frac{1}{S}\sum_{s=0}^{S-1}\sum_{t\in week\ h} D^t.\ PMIN_j^{t,s}$$

$$EMAX_j^{h,\bar{s}} = \frac{1}{S}\sum_{s=0}^{S-1}\sum_{t\in week\ h} D^t.\ PMAX_j^{t,s}$$

Therefore, let $\mathscr{P}$ be the index set of all feasible production plans for the whole set of nuclear power plants (i.e. sequences of productions, outages and reloads that satisfy the *local* constraints [CT3-CT13]), and let $\mathscr{P}_i$ be the subset of production plans (respecting constraints ([CT3-CT13])) associated to nuclear power plant i. Let's note that the in the following we will use the quantity of energy produced for each week h which means to use either the quantity $e_{h\ell}$ in the case of nuclear power plants or the variables $e_{j\ell}$ in the case of thermal power plant. Recalling that the energy is related to the power by the formula $E_h = \int_{t\in h} p(t)dt$. Therefore, each production plan $\ell \in \mathscr{P}_i$ is associated a production quantity $e_{h\ell}$ of energy for each week h. Moreover, to each production plan

$\ell \in \mathscr{P}_i$ is associated c_ℓ which is the total cost of refueling plant i (reduced by the expected value of remaining fuel).

Then, we can write a set partitioning-like reformulation of the original problem where binary variables y_ℓ indicates whether production plan $\ell \in \mathscr{P}_i$ is selected for plant i, and continuous variables e_{hj} indicate the total production of non-nuclear plant j at week h.

$$(F) \quad z(F) = \min \sum_{i=0}^{I-1} \sum_{\ell \in \mathscr{P}_i} c_\ell y_\ell + \sum_{j=0}^{J-1} \sum_{h=0}^{H-1} C_j^{h,\bar{s}} e_{hj} \tag{2}$$

$$\text{s.t.} \quad \sum_{i=0}^{I-1} \sum_{\ell \in \mathscr{P}_i} e_{h\ell} y_\ell + \sum_{j=0}^{J-1} e_{hj} = DEM^{h,\bar{s}} \qquad \forall h = 0,\dots,H-1 \tag{3}$$

$$\sum_{\ell \in \mathscr{P}_i} y_\ell = 1 \qquad \forall i = 0,\dots,I-1 \tag{4}$$

$$EMIN_j^{h,\bar{s}} \le e_{hj} \le EMAX_j^{h,\bar{s}} \qquad \forall j = 0,\dots,J-1,\ h = 0,\dots,H-1 \tag{5}$$

$$\text{Constraints [CT14,...,CT21]} \tag{6}$$

$$y_\ell \in \{0,1\} \qquad \forall \ell \in \mathscr{P}, \tag{7}$$

$$e_{hj} \ge 0 \qquad \forall j = 0,\dots,J-1,\ h = 0,\dots,H-1 \tag{8}$$

Constraints (3) specify that the energy demand of each week $h = 0,\dots,H-1$ must be satisfied. Constraints (4) impose that for each nuclear power plant one and only one column must be chosen. Constraints (5) bound the quantity of energy that a thermal power plant can produce each week.

Let LF be the continuous relaxation of formulation F and let $z(LF)$ be its optimal solution cost. Both formulations F and LF are impractical to solve even for moderate size, since the number of variables is typically exponential, which justifies a column generation approach.

4 Solving the Restricted Master Problem

Let $\widehat{\mathscr{P}}$ be a subset of production plans. We call Restricted Master Problem (*RMP*) associated with $\widehat{\mathscr{P}}$ the above formulation F restricted to columns of $\widehat{\mathscr{P}}$ and without constraints [CT14,...,CT21]. When relaxing the integrality constraints we obtain the linear relaxation (*LRMP*):

$$(LRMP) \quad z(LRMP) = \min \sum_{i=0}^{I-1} \sum_{\ell \in \widehat{\mathscr{P}}_i} c_\ell y_\ell + \sum_{j=0}^{J-1} \sum_{h=0}^{H-1} C_j^{h,\bar{s}} e_{hj} \tag{9}$$

$$\text{s.t.} \quad \sum_{i=0}^{I-1} \sum_{\ell \in \widehat{\mathscr{P}}_i} e_{h\ell} y_\ell + \sum_{j=0}^{J-1} e_{hj} = DEM^{h,\bar{s}} \qquad \forall h = 0,\dots,H-1 \tag{10}$$

$$\sum_{\ell \in \widehat{\mathscr{P}}_i} y_\ell = 1 \qquad \forall i = 0,\dots,I-1 \tag{11}$$

$$EMIN_j^{h,\bar{s}} \le e_{hj} \le EMAX_j^{h,\bar{s}} \qquad \forall j = 0,\dots,J-1,\ h = 0,\dots,H-1 \tag{12}$$

$$y_\ell \ge 0, \qquad \forall \ell \in \widehat{\mathscr{P}}, \tag{13}$$

$$e_{hj} \ge 0 \qquad \forall j = 0,\dots,J-1,\ h = 0,\dots,H-1 \tag{14}$$

The initial subset $\widehat{\mathscr{P}}$ of production plans of the Restricted Master Problem is empty at the beginning, therefore there are no y_ℓ variables, but only e_{hj} variables in the first program. The Linear relaxation of the Restricted Master Problem *LRMP* is solved by CPLEX at each iteration of the process. This LP-solving requires to find for every nuclear plant i a production plan verifying local constraints [CT3-CT13] with minimum reduced cost, i.e. to find for every nuclear plant $i = 0,\dots,I-1$:

$$\ell^* = \underset{\ell \in \mathscr{P}}{\operatorname{argmin}} \bar{c}_\ell \tag{15}$$

where $\bar{c}_\ell = c_\ell - u_i - \sum_h \mu_h e_{h\ell}$, μ_h are the dual variables associated with each week h (i.e. constraints 10) and u_i is the dual variable associated with the assignment constraint for plant i (i.e. constraints 11). The dual variables for the first run are chosen as explained in subsection 5.3. Note that c_ℓ in this equation depends on both $e_{h\ell}$ and $r(i,k)$ as we shall see later.

Although there are some differences, the pricing problem for each nuclear power plant is quite close to an Uncapacitated Lot Sizing problem. Therefore as explained in [12], [13] and [14] it can be modeled as a shortest path problem on a specific graph. The main difficulty of the method is how to build the graph.

The pricing subproblem and the associated graphs for the shortest path computations are described in section 5. Due to time limit in the competition, an approximated pricing subproblem is solved at each iteration: the columns that are added have to be supported by the specific graph the construction of which is explained in subsection 5.1. Hence *LRMP* is not solved to optimality but its projection on the space where new columns have to be supported by these graphs, called Approximated Linear Relaxed Master Problem (*ALRMP*), is solved to optimality. When no more negative reduced cost columns are found, it means that the ALRMP is solved to optimality.

4.1 MIP Solving for fixing outages and BO-periods

Once the *ALRMP* has been solved to optimality by the Column Generation Process, the last *LRMP* (i.e., containing all columns added since the beginning of the process) is solved as a Mixed Integer Program (*MIP*), i.e. with $y_\ell \in \{0,1\}$.

This *MIP* includes incompatibility constraints of the form $\sum_{\ell \in \widehat{\mathscr{P}^l}} y_\ell \leq b_{\widehat{\mathscr{P}^l}}$ for some subsets $\widehat{\mathscr{P}^l}$ of production plans, corresponding to constraints [CT14-CT21] 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-CT21]).

If ℓ_i denotes the column (i.e. production plan or path) selected for nuclear plant i in the optimal *MIP* solution, then the dates of outages in this column are fixed for the rest of the whole method (i.e., variables $ha(i,k)$), as well as the dates when the stock reaches the *BO*-level and the corresponding imposed decreasing profile for the following weeks.

5 Solving the pricing subproblems of column generation

The generation of columns with optimal reduced cost has been implemented as a three-steps process:

1. create an acyclic static *stock graph* $G_i = (N_i, A_i)_i$, for each nuclear power plant i,;
2. for each arc $a \in A_i$ compute weight w_a (optimal reduced cost on the arc) ;
3. generate a new column by finding in G_i a path of minimum weight;

Step 1 is quite time consuming (5 to 30 minutes for all power plants), but occurs only once at the beginning of the whole process. Steps 2 and 3 are light processes (1 to 10 seconds) and can be serialized with the computation of new dual variables μ_h. Figure 1 illustrates the overall column generation process.

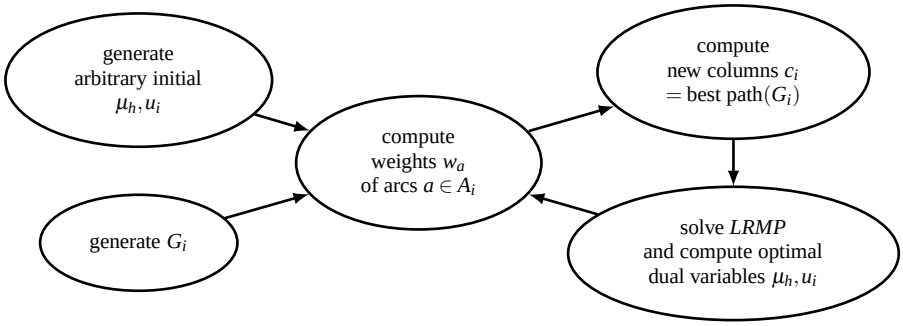


Fig. 1 Column generation process

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 caracterized by a 4-uplet $(type_n, x_n, h_n, k_n)$, where:

- h_n is the week associated with n ;
- x_n is the fuel stock of power plant i *at the beginning* of week h_n ;
- k_n is the id of the campaign in which the power plant will be during week h_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 its h_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_{i,k_n} and the power level will be imposed by constraint `[CT6]` ; the stock x_n of a `BO` node is indeed a little higher than BO_{i,k_n}, but ensures that BO_{i,k_n} can be reached *during* the week h_n by imposing the maximal power production of constraint `[CT5]` at the beginning of the week.

  Besides, there are two nodes with special types, 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 it is possible to reach state n_2 by proceeding to an outage on week h_{n_1} followed by a production campaign, without need for any other outage and without reaching the BO stock level during the time interval $[h_{n_1}, h_{n_2}]$. Note that n_2 can be either a `BO` node or an `OUTAGE` node. 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 constraint `[CT6]` with the acceptable ε tolerance. Note that here n_2 has to be an `OUTAGE` node. 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]` 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 1 Creation of graph G_i

1: create the `initial` node at week 0
2: **for** week $h = 0$ **to** $H - 1$ **do**
3: **for all** previously created node n with $h_n = h$ **do**
4: create new nodes n' with $h_{n'} > h$ 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 1), we follow the beam of possible stock interval(s) over time from a node n, as stated in Algorithm 2:

Algorithm 2 Creation of new nodes in G_i from $n = (type_n, x_n, h_n, k_n)$

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

For steps 5 and 8 of algorithm 2, we discretize the interval $[x_{\min_{n,h}}, x_{\max_{n,h}}]$ of possible stocks of fuel at week h 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 $[x_{\min_{n,h}}, x_{\max_{n,h}}] \cap [x, x + \Delta_i]$, for all $x \in \mathbb{R}^+$. Let us define the maximum production for power plant $i \in I$ over the week $h \in H$ as

$$EMAX_{h,i} = \sum_{t \in timesteps(h)} PMAX_i^t \,.\, D^t$$

and

$$\overline{weekProd_i} = \frac{\sum_{c \in H} EMAX_{h,i}}{H}$$

the mean maximum production per week for power plant $i \in I$, Δ_i is then computed as follows:

$$\Delta_i = \rho \,.\, \overline{weekProd_i}$$

where ρ is a value called the discretisation parameter that is chosen.

Step 2 of algorithm 2 depends on the node type.

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

The node creation process is illustrated by figure 2 in the second case (starting from an `OUTAGE` node).

When all nodes have been created by the process of Algorithm 1, 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

Let us recall the goal of this graph: we want to produce a production plan $\ell \in \mathscr{P}_i$ for power plant i with minimal reduced cost $\bar{c}_\ell$ as explained by equation 15. Nevertheless, in order to point out that they are variables in this pricing subproblem, $e_{h\ell}$ becomes $e(h,l)$ and $\bar{c}_\ell$ becomes $\bar{c}_\ell(r,e,x)$:

$$\bar{c}_\ell(r,e,x) = \sum_k C_{i,k} r(i,k) - C_{i,T+1} x(i,T) - \sum_h \mu_h e(h,\ell) - u_i$$

where:

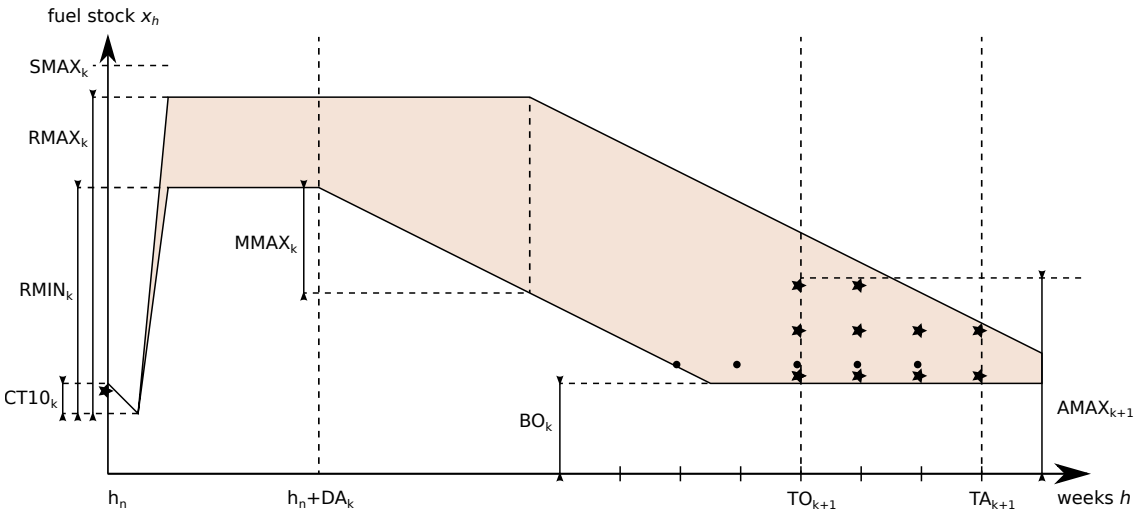


Fig. 2 Creating new nodes from an `OUTAGE` node n in graph G_i inside the beam of possible stock intervals. Stars figure out `OUTAGE` nodes, circles are `BO` nodes. Note that no `BO` node is created at week $\texttt{TA}_{k+1}$, because the paths that would go through such a node would break constraint `CT13bis`.

- $C_{i,k}$ is the cost of a unit of fuel for campaign k and power plant i and $C_{i,T+1}$ is the price for power plant i of a unit of fuel that can be sold at the end of the timeline ;
- μ_h is the dual variable associated to constraint 10 ;
- $x(i,T)$ is the quantity of remaining fuel at the end of the timeline ;
- $r(i,k)$ is the refuel quantity of campaign k ;
- $e(h,\ell)$ is the energy produced at week h ;
- u_i is the dual variable associated to constraint 11

As a production plan is represented by some path $\ell \subset A_i$ in graph G_i, we can rewrite this expression as:

$$\bar{c}_\ell(r,e,x) = \sum_{a=(n,n')\in\ell} \left(C_{i,k_n} r(a) - \delta_{type_{n'}=\texttt{FINAL}} C_{i,T+1} x(i,T) - \sum_{h\in[h_n,h_{n'}[} \mu_h e(h,\ell) \right) - u_i$$

where:

- $r(a)$ is the refuel quantity at the beginning of arc a ;
- $\delta_{type_{n'}=\texttt{FINAL}}$ is 1 if n' is the final node of G_i, 0 otherwise.

Let us call the expression in parenthesis the *reduced cost* $\bar{c}_a(r,e,x)$ of arc a. This decomposition on the arcs of the graph means that if we weight each arc a with its minimal reduced cost

$$w_a = \min_{r,e,x}(\bar{c}_a(r,e,x))$$

we can easily find the production plan with minimal reduced cost by solving a shortest path in the graph. The reduced cost of this production plan is obtained by substracting u_i to the minimum weight path value.

Weighting a `BO` arc is straightforward because there is no refuel at the beginning of such an arc and the productions $e(h,\ell)$ are fixed for each timestep by constraint `CT6`, if we neglect the allowed deviation ε. The weight of a `BO` arc a is given by the following expression, where everything is known:

$$w_a = \left(-\delta_{type_{n'}=\texttt{FINAL}} \cdot C_{i,T+1} \cdot \left(x_n - \sum_{h\in[h_n,h_{n'}[} e(h,\ell) \right) - \sum_{h\in[h_n,h_{n'}[} \mu_h \cdot e(h,\ell) \right) \tag{16}$$

Weighting a non final `OUTAGE` *arc* is more intricate because the allowed modulation gives some latitude to productions $e(h,\ell)$. Only the weeks of production $(h_n, h_{n'})$, and the levels of fuel $(x_n, x_{n'})$ at the beginning and at the end of the arc, are known.

Let $\bar{x_n} = \left(\frac{Q_{i,k_n}-1}{Q_{i,k_n}} x_n - \texttt{BO}_{i,k_n-1} + \texttt{BO}_{i,k_n}\right)$. The problem can then be stated as follows:

$$w_a = \min_{r,e,x} C_{i,k_n} r(a) - \sum_{h\in[h_n,h_{n'}[} \mu_h e(h,\ell) \tag{17}$$

$$\text{s.t.} \quad \bar{x_n} + r(a) = x_{n'} + \sum_{h\in[h_n,h_{n'}[} e(h,\ell) \qquad [\texttt{CT10},\texttt{CT9}] \tag{18}$$

$$0 \le e(h,\ell) \le EMAX_h \quad \forall h \in [h_n,h_{n'}[\qquad [\texttt{CT4},\texttt{CT5}] \tag{19}$$

$$RMIN_{k_n} \le r(a) \le RMAX_{k_n} \qquad [\texttt{CT7}] \tag{20}$$

$$\bar{x_n} + r(a) \le SMAX_{i,k_n} \qquad [\texttt{CT11}] \tag{21}$$

$$\sum_{h\in[h_n,h_{n'}[} e(h,\ell) \ge -MMAX_{k_n} + \sum_{h\in[h_n,h_{n'}[} EMAX_h \qquad [\texttt{CT12}] \tag{22}$$

We can use (18) to rewrite $r(a)$ and remove it from the problem. It comes that we are looking for the minimum under constraints (19) to (22) of:

$$w_a = C_{i,k_n}\left(x_{n'} - \bar{x_n}\right) + \sum_{h\in[h_n,h_{n'}[} \left(C_{i,k_n} - \mu_h\right) e(h,\ell)$$

As the first part of the sum is *known* for each arc, we only have to find the productions $e(h,\ell)$ that minimize $\sum_{h\in[h_n,h_{n'}[} \left(C_{i,k_n} - \mu_h\right) e(h,\ell)$. We tried two methods to achieve this goal:

The exact solution is obviously given by the simplex method, since (19) to (22) is a linear model or by Algorithm 3 below, where *nullify* has to be understood as "set as close to zero as possible without violating constraints (19) to (22):

Algorithm 3 Exact algorithm for minimizing arc weight

set all $e(h,\ell)$ to $EMAX_h$
order the set of weeks by decreasing values of $\left(C_{i,k_n} - \mu_{h_m}\right) : h_1, h_2 \ldots$
start with $m = 1$
while constraints 19,20,21,22 are not satisfied **do**
  nullify $e_{h_m\ell}$
  increment m
end while
while $\left(C_{i,k_n} - \mu_{h_m}\right) > 0$ **and** constraints 19,20,21,22 can be satisfied **do**
  nullify $e_{h_m\ell}$
  increment m
end while

However, we encountered two problems with the exact solution:

- the overall column generation process was converging too fast, and no new column was generated after about two or three iterations ; this is not redhibitory though, because this can be overcome by the *diversification* strategy exposed in section 6 ;
- considering there are hundreds of thousands of arcs in the graph, the method is quite time consuming because it has to walk along the ordered set of weeks.

We used a heuristic to give a good solution, even if not optimal: we chose to *spread* the modulation on all weeks proportionally to $EMAX_h$, but with a different factor whether the *reduced unit cost* $\bar{c}_{i,k,h} = C_{i,k} - \mu_h$ is positive or negative. For that purpose, we split the set of weeks into two sets: $\mathscr{H}_k^+ = \{h \in [0,H] \text{ s.t. } C_{i,k} - \mu_h > 0\}$ and $\mathscr{H}_k^- = \{h \in [0,H] \text{ s.t. } C_{i,k} - \mu_h \leq 0\}$. 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 $e(h,\ell)$ from these variables:

$$\forall h \in [h_n, h_{n'}[\cap \mathscr{H}_{k_n}^+, \;\; e(h,\ell) = \alpha_a \,.\, EMAX_h$$
$$\forall h \in [h_n, h_{n'}[\cap \mathscr{H}_{k_n}^-, \;\; e(h,\ell) = \beta_a \,.\, EMAX_h$$

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

$$\alpha_a \left(\sum_{h \in [h_n, h_{n'}[\cap \mathscr{H}_{k_n}^+} (C_{i,k_n} - \mu_h) EMAX_h \right) + \beta_a \left(\sum_{h \in [h_n, h_{n'}[\cap \mathscr{H}_{k_n}^-} (C_{i,k_n} - \mu_h) EMAX_h \right)$$

The parenthesized expressions do not have to be computed for each arc. Instead, we can compute them *in advance* for each $(k,h,h')_{h<h'}$ triplet. Similarly, the constraints (2-5) of the problem can be rewritten as linear constraints on α_a and β_a and we will save a lot of computing overhead by precomputing $\sum_{h \in [h_n, h_{n'}[\cap \mathscr{H}_{k_n}^-} EMAX_h$ for each couple $(h,h')_{h<h'}$ and their counterparts on $\mathscr{H}_{k_n}^+$.

As it reduces the number of variables to a couple, this heuristic has proved to be much more efficient than looking for the exact solution and as effective when placed in the context of this column generation problem.

5.3 Obtaining dual variables μ_h and u_i

At each iteration we solved *LRMP* to optimality and the dual variables (μ_h) and (u_i) are respectively associated with constraints (10) and (11). For the first run the variables are arbitrarily chosen as follows: $(u_i) = 0$ for all $i \in I$ and $(\mu_h) = \max_{i \in I, k \in [0,K-1]} \{C_{i,k} + 1\}$ so that $\mathscr{H}_k^+ = [0,H]$ for all power plants. This will create a first pool of production plans where nuclear power plants are used at their maximal capacity.

6 Columns diversification

The main difficulty encountered with the method described so far comes when solving the *MIP*for column selection (see section 4.1) : as 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 w_a given by equations (16) and (17-22), we compute and use weights w'_a in the following way:

$$w'_a = \begin{cases} 0 & \text{for } \texttt{BO} \text{ arcs} \\ nbConflicts(a,col) & \text{for } \texttt{OUTAGE} \text{ arcs} \end{cases} \tag{23}$$

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 w'_a with the following algorithm:

Algorithm 4 Alternate weight computation during a *diversification* pass

$w'_a \leftarrow 0$
for all existing column $c \in cols$ **do**
  for all constraint m of type `[CT14]` to `[CT18]` **do**
    if column c conflicts with an outage of power plant i_a at week h_a by constraint m **then**
      $w'_a \leftarrow w'_a + 1$
    end if
  end for
  for all constraint m of type `[CT19]` **do**
    if for some outage k of c, (i_a, k_a) and (i_c, k) are part of set A_m **and** h_a is interval $[h_k + L_{i_c,k,m} - L_{i_a,k_a,m}, h_k + L_{i_c,k,m} - L_{i_a,k_a,m} + TU_{i_c,k_c,m}]$ **then**
      $w'_a \leftarrow w'_a + \frac{1}{Q_m}$
    end if
  end for
  for all constraint m of type `[CT20]` concerning week h_m **do**
    if $h_m \in [h_a, h_a + DA_{i_a,k_a}] \cap [h_k, h_k + DA_{i_c,k}]$ for some outage k of column c **then**
      $w'_a \leftarrow w'_a + \frac{1}{N_m(h_m)}$
    end if
  end for
end for
for all constraint m of type `[CT21]` concerning interval IT_m such that $i_a \in C_m$ **do**
  $w'_a \leftarrow w'a + \max_{h \in [h_a, h_a + DA_{i_a,k_a}] \cap IT_m} \left[\sum_{(c,k) \in cols \times outages / k \in outages(c) \ \wedge \ i_a \neq i_c \ \wedge \ i_c \in C_m \ \wedge \ h \in [h_k, h_k + DA_{i_c,k}] \cap IT_m} \frac{PMAX^h_{i_c}}{TMAX_m} \right]$
end for

Note that w'_a only depends on the power plant i_a, the week of outage h_a and the campaign k_a ; furthermore, when new columns are generated, they only modify weights w'_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.

7 Reload and production

Let us remark that once the dates for outages and BO-profiles have been fixed as described in section 4.1, 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 $r(i,k)$ during each outage k for every nuclear plant i and the optimal production values $p(j,t,s)$ and $p(i,t,s)$ for non-nuclear and nuclear plants, respectively, and the optimal stock values $x(i,t,s)$ for time steps t and scenario s such that the stock level is over the BO threshold.

7.1 LP-solving for fixing reload quantities

The reload quantities $r(i,k)$ are determined by solving a partially-disaggregated LP where every scenario s is considered instead of the average scenario $\overline{s}$, but time steps t are still aggregated into weeks h so as to accelerate solving. The optimal reload quantities $r(i,k)$ are found by solving a Linear Program (LP_R) with CPLEX, where non-negative decision variables are reload variables $r(i,k)$, production variables $e(j,h,s)$ and $e(i,h,s)$, and stock variables $x(i,h,s)$. The constraints in this LP are all constraints [CT1-CT12], but for weeks h instead of time steps t. The objective function is thus the following:

$$z(LP_R) = \min \sum_i \sum_k C_{ik} r(i,k) + \frac{1}{S} \sum_s (\sum_h \sum_j C_j^{hs} e(j,h,s) - \sum_i C_{i,H+1} x(i,H,s))$$

This linear problem LP_R is solved for the different combinations of columns that produce the optimal value of the Master Problem 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.

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

At this final step, for every scenario $s = 0, \ldots, S-1$, we go back to time steps t. Therefore, S Linear Programs (LP_s) are solved consequently where decision variables are:

(i) the production variables $p(j,t,s)$ and $p(i,t,s)$ for non-nuclear and nuclear plants, respectively,
(ii) the stock variables $x(i,t,s)$ 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-CT12], but for time steps t now, i.e. original constraints [CT1-CT12]. The objective functions are the total production cost for each scenario s, reduced by the objective value of remaining fuel:

$$z(LP_s) = \min \sum_t \sum_j C_j^{ts} p(j,t,s) D^t - \sum_i C_{i,T+1} x(i,T,s))$$

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:

$$z = \min \sum_i \sum_k C_{ik} r(i,k) + \frac{1}{S} \sum_s (\sum_t \sum_j C_j^{ts} p(j,t,s) D^t - \sum_i C_{i,T+1} x(i,T,s)) \quad (24)$$

8 Computational Results

The 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.

In our computational experiments, we considered 16 instances given by EDF during the Challenge (see [11]). 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 scenarii 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 scenarii considered can rise till 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 scenarii 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.

In table 1, the best results that were obtained and published on the website of the Challenge are given in the first column and the one that was obtained by the current method. All teams were ranked by the Challenge

Data number	ROADEF best found results	our ROADEF results	best found results since	ρ factor
data0	8.73099e+12	8.73623e+12	8.73696e+12	0.9
data1	1.69538e+11	1.70030e+11	1.69902e+11	0.449
data2	1.46048e+11	1.46409e+11	1.46435e+11	0.9
data3	1.54430e+11	1.54711e+11	1.54747e+11	0.9
data4	1.11591e+11	1.12808e+11	1.12528e+11	0.45
data5	1.25822e+11	1.28366e+11	1.26932e+11	0.449
data6	8.34247e+10	8.47093e+10	8.46439e+10	0.5
data7	8.11742e+10	8.18391e+10	8.18501e+10	0.45
data8	8.19262e+10	8.29770e+10	8.29548e+10	0.449
data9	8.17509e+10	8.36228e+10	8.29290e+10	0.5
data10	7.77670e+10	7.87302e+10	7.88065e+10	0.449
data11	7.91168e+10	TIMEOUT	8.00418e+10	3.0
data12	7.75899e+10	7.81931e+10	7.81252e+10	0.45
data13	7.64492e+10	INFEASIBLE	7.67848e+10	0.45
data14	7.61730e+10	TIMEOUT	7.69035e+10	0.449
data15	**7.51014e+10**	**7.51014e+10**	7.49233e+10	0.449

Table 1 Results of the ROADEF 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 the one we have been developed is the only one, as far as we know, having some mathematical programming foundations.

The third column gives the best results 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 (ALRMP).

In tables 2 and 3 we give several results about the method:

- The first column gives the considered instance ;
- The second column shows the optimal result of the *ALRMP* ;
- 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,...,CT21] 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 eighth column gives the value of LP_R ;
- The ninth column gives the result of z defined in equation (24) ;
- The last column gives the elapsed time for the whole algorithm

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 Approximated Relaxed Pricing Subproblem is reached then.

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.

Let us 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 process is necessary to find quickly the integer solution of the *ALRMP*.

Finally, note that between column *MIP* and LP_R the quality of solutions found decreases. It means that if we had a more accurate model with only one scenario instead of several scenarii, the quality of the method would be improved.

Data	*ALRMP*	It to LB	Tot It	Pl Nb	Cs Nb	*MIP*	LP_R	*z*	Time (s)
data0	8.73548e+12	4	24	32	40	8.73696e+12	8.73696e+12	8.73696e+12	0
data1	1.52339e+11	8	149	540	1106	1.52489e+11	1.68891e+11	1.69978e+11	7
data2	1.45276e+11	18	449	2111	1947	1.45745e+11	1.45747e+11	1.46435e+11	300
data3	1.52835e+11	8	764	1 929	1753	1.53190e+11	1.54055e+11	1.54747e+11	46
data4	1.02524e+11	18	461	2617	2626	1.02974e+11	1.11857e+11	1.12558e+11	253
data5	1.19655e+11	18	521	3226	3020	1.20145e+11	1.26698e+11	1.27744e+11	187
data6	7.75175e+10	22	761	6642	5630	7.90051e+10	8.16399e+10	8.48905e+10	513
data7	7.48221e+10	348	1094	8437	6105	7.57638e+10	7.87608e+10	8.21897e+10	896
data8	7.35414e+10	22	406	3256	7497	7.60027e+10	7.95534e+10	8.32061e+10	422
data9	7.31604e+10	28	372	3630	7477	7.56930e+10	7.96609e+10	8.34030e+10	577
data10	7.03327e+10	36	3075	11157	5944	7.13487e+10	7.52558e+10	7.90224e+10	851
data11	7.34759e+10	40	1239	9061	5719	7.45279e+10	7.71169e+10	8.00708e+10	1908
data12	7.17523e+10	44	1958	10869	5640	7.22878e+10	7.49525e+10	7.82294e+10	939
data13	6.98451e+10	34	409	6299	6687	7.07385e+10	7.39050e+10	7.69910e+10	524
data14	6.92277e+10	40	611	7961	6911	6.99999e+10	7.36125e+10	7.69212e+10	1133
data15	6.73328e+10	44	935	10026	5955	6.78149e+10	7.13536e+10	7.50613e+10	804

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

Data	*ALRMP*	It to LB	Tot It	Pl Nb	Cs Nb	*MIP*	LP_R	*z*	Time (s)
data0	8.73548e+12	4	24	32	40	8.73698e+12	8.73698e+12	8.73698e+12	0
data1	1.52342e+11	8	58	269	1072	1.52536e+11	1.68836e+11	1.69962e+11	7
data2	1.45284e+11	20	391	1614	1931	1.45769e+11	1.46151e+11	1.46840e+11	286
data3	1.52830e+11	8	776	1735	1694	1.53188e+11	1.54089e+11	1.54777e+11	33
data4	1.02543e+11	22	625	2687	2632	1.03002e+11	1.12506e+11	1.13198e+11	217
data5	1.19685e+11	18	367	2903	3059	1.20247e+11	1.27393e+11	1.28352e+11	173
data6	7.75271e+10	22	740	6283	5637	7.88730e+10	8.15095e+10	8.47410e+10	465
data7	7.48317e+10	38	661	9574	6130	7.55816e+10	7.85357e+10	8.19583e+10	936
data8	7.36053e+10	20	350	3311	7401	7.76852e+10	8.13098e+10	8.51822e+10	355
data9	7.31967e+10	30	1419	6187	7544	7.53907e+10	7.90747e+10	8.27811e+10	607
data10	7.03706e+10	34	3320	11417	5928	7.13445e+10	7.51940e+10	7.89976e+10	762
data11	7.34802e+10	38	1232	9508	5690	7.45335e+10	7.70939e+10	8.00418e+10	1563
data12	7.17785e+10	50	975	10821	5687	7.22479e+10	7.48730e+10	7.81379e+10	905
data13	6.98541e+10	24	535	6529	6667	7.08168e+10	7.38722e+10	7.68983e+10	575
data14	6.92260e+10	36	713	8243	6870	6.99515e+10	7.36255e+10	7.69829e+10	1122
data15	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 $\rho = 3$

Acknowledgment

The authors of this paper 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. *Mathematical Programming*, 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. Maurizio Bruglieri and Leo Liberti. Optimal running and planning of a biomass-based energy production process. *Energy Policy,*, 36:2430 – 2438, 2008.
4. Guy Desaulniers, Jacques Desrosiers, and Marius M.Solomon; GERAD. *Column Generation*. Springer, 2005.
5. Ajay Gupta, Rajsafah P. Saini, and Mahendra P. Sharma. Modelling of hybrid energy system. *Renewable Energy,*, 36:456 – 481, 2011.
6. Josef Kallrath, Panos M.Pardalos, Steffen Rebennack, and Michel Scheidt. *Optimization in the Energy Industry*. 2009.
7. Mohand I. Khemmoudj, Marc Porcheron, and Hachémi Bennaceur. When constraints programming and local search solve the scheduling problem of edf nuclear power plant aoutages. *12th Internationnal Conference on Principles and Practice of Constraint Programming*, pages 271–283, 2006.

8. Marco E. Lubbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, pages 1007–1023, November-December 2005.
9. Hugo Morais, Péter Kádárb, Pedro Fariaa, Zita A. Valea, and H.M. Khodr. Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming. *Renewable Energy,*, 35:151 – 156, 2010.
10. Ivan Mustakerov and Daniela Borissova. Wind turbines type and number choice using combinatorial optimization. *Renewable Energy,*, 35:1887 – 1894, 2010.
11. 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. Master document and data instances A and B available at `http://challenge.roadef.org/2010/en/`.
12. 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.
13. Harvey M. Wagner and Thomson M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5:89 – 96, October 1958.
14. Laurence A. Wolsey. *Integer Programming*. 1998.