DTU Library

A solution approach to the ROADEF/EURO 2010 challenge based on Benders' Decomposition

Lusby, Richard Martin; Muller, Laurent Flindt; Petersen, Bjørn

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

A solution approach to the ROAD-EF/EURO 2010 challenge based on Benders' Decomposition

Report 18.2010

DTU Management Engineering

Richard Lusby
Laurent Flindt Muller
Bjørn Petersen

November 2010

DTU Management Engineering
Department of Management Engineering

A solution approach to the ROADEF/EURO 2010 challenge based on Benders Decomposition

Richard Lusby Laurent Flindt Muller Bjørn Petersen

Department of Management Engineering, Technical University of Denmark
Produktionstorvet, Building 426, DK-2800 Kgs. Lyngby, Denmark
rlmu@man.dtu.dk, lafm@man.dtu.dk, bjop@man.dtu.dk

Abstract

Since 1999 the French Operations Research Society, Recherche Opérationnelle et d'Aide à la Décision (ROADEF), has organized the so-called ROADEF challenge, and international operations research contest in which participants must solve an industrial optimization problem. In 2010 it was jointly organized for the first time with the European Operational Research Society (EURO) and was run in collaboration with Electricité de France (EDF), one of the largest utility companies in the world, and required contestants to solve a large scale energy management problem with varied constraints. The challenge focused on the nuclear power plants, which need to be regularly shut down for refueling and maintenance, and asked contestants to schedule these outages such that the expected cost of meeting the power demand in a number of potential scenarios is minimized.

We present a Benders decomposition based framework for solving the problem. Because of the nature of the problem, not all constraints can be modelled satisfactorily as linear constraints and the approach is therefore divided into two stages: in the first stage Benders feasibility and optimality cuts are added based on the linear programming relaxation of the Benders Master problem, while in the second stage feasible integer solutions are enumerated and a procedure is applied to each solution in an attempt to make them satisfy the constraints that are not included in the mixed integer program. A number of experiments are performed on the available benchmark instances. These experiments show that the approach is competitive on the smaller instances, but not for the larger ones. We believe the exact approach gives insight into the problem, and additionally makes it possible to find lower bounds on the problem, which is typically not the case for the competing heuristics.

1 Introduction

Every two years[1] since 1999 the French Operations Research Society, Recherche Opérationnelle et d'Aide à la Décision (ROADEF), has organized the so-called ROADEF challenge, an international operations research contest in which participants must solve an industrial optimization problem. Given the success of previous contests, this year it was jointly organized for the first time with the European Operational Research Society (EURO) and known as the ROADEF/EURO 2010 challenge. The competition was run in collaboration with Electricité de France (EDF), one of the largest utility companies in the world, and required contestants to solve a large scale energy management problem with varied constraints.

EDF's power generation facilities in France stand for a total of 98.8 GW of installed capacity, most of which is produced using thermal, and in particular nuclear, power plants. In 2008 thermal power plants accounted for 90% of its total electricity production, 86% of which was delivered by nuclear power plants. This year's challenge focused on the nuclear power plants, since these need to be regularly shut down for refueling and maintenance, and asked contestants to schedule these outages in such a way that the various constraints regarding safety, maintenance, logistics, and

[1] Except for 2009-2010.

plant operation were satisfied, while minimizing the expected cost of meeting the power demand in a number of potential scenarios. The problem thus consisted of the following two dependent subproblems

1. Determine a schedule of nuclear power plant outages. This entails determining when the nuclear power plants should be taken *offline* and how much fuel should be reloaded at each. An outage lasts for some predefined (plant specific) period of time during which the nuclear power plant cannot be used for power generation. The coupling of an outage followed by a production period (until the next outage) for a nuclear power plant is termed a *cycle* and it is not uncommon to have to schedule up to six cycles for each nuclear power plant. In determining an outage schedule one must obey several safety requirements as well as observe restrictions arising from the limited resources available to perform the fuel reloading.

2. Given an outage schedule, determine a production plan for each of the *online* power plants, i.e. the quantity of electricity to produce in each time step, for each possible demand scenario. The power plants are divided into two categories termed type 1 and type 2, respectively. Type 2 power plants refer to the nuclear power plants and must be reloaded with fuel, while type 1 power plants represents thermal power plants, which can be supplied with fuel continuously, such as coal, gas, and oil powered plants. Several technical constraints govern the possible levels of power production at each power plant. Due to the stochastic nature of power markets, one is required to consider multiple demand scenarios.

The concepts of cycles, outages, and production plans for three power plants are illustrated in Figure 1. The gray area indicates the time steps during which the plants are offline.

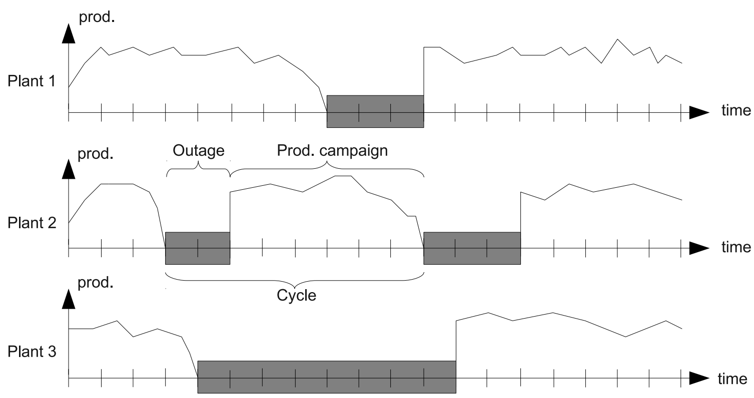

Figure 1: Outages, Cycles, Production Plans

One important aspect of this years competition is handling the size of the problem: There are approximately one hundred power plants and scenarios, and the planning horizon is in the order of years, with a granularity down to hours. Which means that a solution alone can contain in the order of 10^8 variables.

The remainder of the paper is organized as follows. Section 2 gives an overview of the problem constraints, while Section 3 presents a mixed integer programming (MIP) model for (parts of) the problem. In Section 4 we give a general outline of the proposed procedure and present the Benders decomposed model. Section 5 gives techniques for how the problem size can be reduced, and Section 6 describes a number of additional constraints that we add to the model in an attempt to reduce the number of infeasible subproblems. In Section 7 we present a procedure for taking a solution, which does not satisfy all the constraints, and making it do so. Extensive computational results are reported in Section 8 and conclusions from this research are drawn in Section 9.

2 Overview of problem constraints

Table 1 gives a brief overview of the different constraints of the problem. Due to space considerations, we do not include a full description of each constraint, but instead refer the reader to the official competition document by Porcheron et al. (2009). We first introduce a number of sets and constants, which will be used to state the mathematical model in Section 3. Additionally, we define some sets and variables that will only be used to describe the constraints in this section.

Sets

- I: Set of type 2 plants (nuclear). Indexed by i.
- J: Set of type 1 plants (other thermal). Indexed by j.
- T: Set of time steps. Indexed by t.
- W: Set of weeks. Indexed by w.
- S: Set of scenarios. Indexed by s.
- K_i: Set of cycles for each plant $i \in I$.

Constants

- L_{ik}: Length in weeks of the outage for cycle $k \in K_i$ at plant $i \in I$.
- $\underline{R}_{ik}$: Minimum reload amount for plant $i \in I$ in cycle $k \in K_i$.
- $\overline{R}_{ik}$: Maximum reload amount for plant $i \in I$ in cycle $k \in K_i$.
- P_{it}^{max}: Maximum production for plant $i \in I$ at time step $t \in T$
- F_t: Conversion factor between power and fuel in time step $t \in T$.
- D_{ts}: Required power in time step $t \in T$ of scenario $s \in S$.
- B_{ik}: Fuel stock level at which shutdown curve must begin in cycle k at plant $i \in I$.
- Q_{ik}: Proportion of fuel that can be kept during reload in cycle k at plant $i \in I$, $\tilde{Q}_{ik} := \frac{Q_{ik}-1}{Q_{ik}}$.
- S_{ik}^{max}: Maximum permitted fuel after reload in cycle k at plant $i \in I$, $M_i := \max_k S_{i,k}^{max}$.
- A_{ik}^{max}: Maximum permitted fuel prior to reload in cycle k at plant $i \in I$.
- $\overline{P}_{it}$: Maximum production capacity of plant $i \in I$ at time step $t \in T$.
- $\overline{P}_{jts}$: Maximum production capacity for plant $j \in J$ at time step $t \in T$ in scenario $s \in S$.
- $\underline{P}_{jts}$: Minimum production capacity for plant $j \in J$ at time step $t \in T$ in scenario $s \in S$.
- X_i: Starting stock of plant $i \in I$
- T_{ik}^{O}: First possible outage week for cycle $k \in K_i$ for $i \in I$.
- T_{ik}^{A}: Last possible outage week for cycle $k \in K_i$ for $i \in I$.

Constraint-specific variables and sets

- $ha(i,k)$: the first week of the outage of cycle $k \in K_i$ of plant $i \in I$.
- $p(j,t,s)$: production of plant $j \in J$ during the time step $t \in T$ of scenario $s \in S$.
- $p(i,t,s)$: production of plant $i \in I$ during the time step $t \in T$ of scenario $s \in S$.
- $r(i,k)$: reload performed during the outage of cycle $k \in K_i$ of plant $i \in I$.
- $x(i,t,s)$: stock of fuel of plant $i \in I$ at time step $t \in T$ for scenario $s \in S$.
- $ec(i,k)$: set of time steps composing the production campaign of cycle $k \in K_i$ of plant $i \in I$.
- $ea(i,k)$: set of weeks composing the outage of cycle $k \in K_i$ of plant $i \in I$.

Table 1: Overview of the constraints of the problem

Name	Description
CT1	Constraint coupling load and production: during every time step $t \in T$ of every scenario $s \in S$, the sum of production of type 1 and type 2 power plants has to be equal to the demand: $$\sum_{i \in I} p_{its} + \sum_{j \in J} p_{jts} = D_{ts}, \quad \forall (t,s)$$
CT2	Bound on production: During every time step $t \in T$ of every scenario $s \in S$, production of plant $j \in J$ has to be between minimum and maximum bounds: $$\underline{P}_{jts} \leq p_{jts} \leq \overline{P}_{jts}, \quad \forall (j,t,s)$$
CT3	Offline power: During every time step $t \in T$ of every scenario $s \in S$ where plant $i \in I$ is offline, its production is equal to zero: $$t \in ea(i,k) \Rightarrow p(i,t,s) = 0, \quad \forall (i,t,s)$$
CT4	Minimum power: During every time step $t \in T$ of every scenario $s \in S$ where plant $i \in I$ is online, its production is non-negative: $$0 \leq p(i,t,s), \quad \forall (i,t)$$
CT5	Maximum power before activation of imposition of power profile constraint (see CT6): During every scenario $s \in S$ and every time step $t \in T$ of the production campaign of cycle $k \in K_i$, if the current fuel stock of plant $i \in I$ is greater than or equal to B_{ik}, the production level has to be equal or less than its maximum bound: $$t \in ec(i,k) \land x(i,t,s) \geq B_{ik} \Rightarrow p(i,t,s) \leq \overline{P}_{it}, \quad \forall (i,t,s)$$

Table 1: Continued – Overview of the constraints of the problem

Name	Description
CT6	Maximum power after activation of imposition of power level constraint: During every scenario $s \in S$ and every time step $t \in T$ of the production campaign of cycle $k \in K_i$, if the current fuel stock of plant $i \in I$ is inferior to B_{ik}, production has to follow the power profile $\mathcal{P}_{ik} : \mathrm{R} \rightarrow [0;1]$ with a tolerance ϵ, where $\mathcal{P}_{ik}$ is a piecewise linear function of the stock level: $$t \in ec(i,k) \wedge x(i,t,s) \leq B_{ik} \Rightarrow p(i,t,s) \approx \mathcal{P}_{ik}(x(i,t,s)), \quad \forall (i,t,k,s)$$
CT7	Bounds on refueling: The reload performed during cycle $k \in K_i$ of plant $i \in I$ has to be inside its minimum and maximum bounds: $$\underline{R}_{ik} \leq r(i,k) \leq \overline{R}_{ik}, \quad \forall (i,k)$$
CT8	Initial fuel stock: $$x(i,0,s) = X_i, \quad \forall (i,s)$$
CT9	Fuel stock variation during a production campaign of a cycle: $$t \in ec(i,k) \Rightarrow x(i,t+1,s) = x(i,t,s) - p(i,t,s) \cdot F_t, \quad \forall (t,i,k,s)$$
CT10	Fuel stock variation during an outage: In the process of refueling a type 2 power plant at time $t \in T$, i.e., t is the first timestep of an outage, a certain amount of unspent fuel has to be removed to make the addition of new fuel possible: $$x(i,t+1,s) = \tilde{Q}_{ik} \cdot (x(i,t,s) - B_{i,k-1}) + r(i,k) + B_{ik}, \quad \forall (i,k,s)$$
CT11	Bounds on fuel stock at the instant, $t \in T$, of outage and after refueling $(t+1)$: $$x(i,t,s) \leq A_{ik}^{max}, \quad x(i,t+1,s) \leq S_{ik}^{max}, \quad \forall (i,k,s)$$
CT12	Constraint on maximum modulation over a cycle: Modulating the power output of a type 2 power plant leads to a certain amount of wear on the equipment involved. Therefore frequent power modulations at type 2 power plants are undesirable: $$\sum_{t \in \{t' \in ec(i,k) : x(i,t',s) \geq B_{ik}\}} (\overline{P}_{it} - p(i,t,s)) \cdot F_t \leq M_{ik}^{max}, \forall (i,k,s)$$

Table 1: Continued – Overview of the constraints of the problem

Name	Description
CT13	Constraint on the earliest and latest date of an outage: Outage of cycle $k \in K_i$ of plant $i \in K_i$ has to start during a given interval: $$T^O_{ik} \leq ha(i,k) \leq T^A_{ik}, \quad \forall (i,k),$$ $$ha(i,k+1) \geq ha(i,k) + L_{ik}, \quad \forall (i,k)$$ If no CT13 constraint is present, then scheduling the corresponding cycle is optional, but the cycle must still be scheduled in order for any subsequent cycle $k' > k$ to be scheduled.
CT14	Constraint on the minimum spacing/maximum overlapping between outages: a set of outages, A^{14}_m, have to be spaced by at least S^{14}_m weeks, with $m = 1, \ldots M_{14}$: $$ha(i,k) - ha(i',k') - L_{i'k'} \geq S^{14}_m \vee ha(i',k') - ha(ik) - L_{ik} \geq S_m, \quad \forall (i,k),(i',k') \in A^{14}_m$$
CT15	Minimum spacing/maximum overlapping between outages during a specific period: a set of outages, A^{15}_m, that intersect an interval $[U_m; V_m]$ have to be spaced by at least or can overlap by at most S^{15}_m weeks, with $m = 1, \ldots M_{15}$: $$U_m - L_{ik} + 1 \leq ha(i,k) \leq V_m \wedge U_m - L_{i'k'} + 1 \leq ha(i',k') \leq V_m$$ $$\Rightarrow ha(i,k) - ha(i',k') - L_{i'k'} \geq S^{15}_m \vee ha(i',k') - ha(ik) - L_{ik} \geq S_m,$$ $$\forall (i,k),(i',k') \in A^{15}_m$$
CT16	Minimum spacing constraint between decoupling dates: decoupling dates of of a set of outages, A^{16}_m, have to be spaced by at least S^{16}_m weeks, with $m = 1, \ldots M_{16}$: $$\lvert ha(i,k) - ha(i',k') \rvert \geq S^{16}_m, \quad \forall (i,k),(i',k') \in A^{16}_m$$
CT17	Minimum spacing constraint between dates of coupling: Coupling dates of a set of outages, A^{17}_m, have to be spaced by at least S^{17}_m weeks, with $m = 1, M_{17}$: $$\lvert ha(i,k) + L_{ik} - ha(i',k') - L_{i'k'} \rvert \geq S^{17}_m, \quad \forall (i,k),(i',k') \in A^{17}_m$$
CT18	Minimum spacing constraint between coupling and decoupling dates: coupling and decoupling dates for a set of outages, A^{18}_m, have to be spaced by at least S^{18}_m weeks, with $m = 1, \ldots M_{18}$: $$\lvert ha(i,k) + L_{ik} - ha(i',k') \rvert \geq S^{18}_m, \quad \forall (i,k),(i',k') \in A^{18}_m$$

Table 1: Continued – Overview of the constraints of the problem

Name	Description
CT19	Resource constraint: the use of resources on a given set of outages, A_m^{19}, is subject to constraints due to their limited availability, with U_{ikm}, and V_{ikm} indicating the start and the length of the resource usage period with $m = 1, \ldots M_{19}$: $$\sum_{(i,k)\in A_m^{19}} \delta(t,i,k) \leq Q_m, \quad \forall w,$$ where $\delta(t,i,k) = 1 \iff t \in [ha(i,k) + U_{ikm}; ha(i,k) + U_{ikm} + V_{ikm}]$
CT20	Constraint on the maximum number of overlapping outages during a given week: At most $N_m(w)$ outages of $A_m^{20}(w)$ can overlap during the week$w \in W$, with $m = 1, \ldots M_{20}$: $$\sum_{(i,k)\in A_m^{20}} \delta(t,i,k) \leq N_m(w), \quad \forall w,$$ where $\delta(t,i,k) = 1 \iff t \in [ha(i,k); ha(i,k) + L_{ik}]$.
CT21	Constraint on the offline power capacity of a set of power plants during a time period: For a given period, $[U_m; V_m]$ the power capacity of the set of plants C_m^{21} that are offline has to be inferior to a maximum bound, I_m^{max}, with $m = 1, \ldots, M_{21}$: $$\sum_{i\in C_m^{21}} \sum_{w\in[U_m;V_m]\cap ec(i,k)} \sum_{t\in w} \overline{P}_{it} \leq I_m^{max}$$

3 Model

As the approach to solving the problem will be based on applying mixed integer programming (MIP), we now give a MIP model of the problem. Before stating the model we introduce some additional sets, constants, and variables.

Sets

- W_{ik}^o: Set of allowed outage weeks for cycle $k \in K_i$ of plant $i \in I$.
- W_{ik}^p: Set of weeks where cycle $k \in K_i$ of plant $i \in I$ could be in a production campaign.
- T_{ik}^p: Set of time steps where cycle $k \in K_i$ of plant $i \in I$ could be in a production campaign.
- $K_i(w)$: Set of cycles for plant $i \in I$ which could be in a production campaign at in week w.
- $w(t)$: Week containing time step t
- T_w: Set of time steps in week w
- M_{21}: Set of CT21 constraints
- C_m: Set of type 2 power plants associated with $m \in M_{21}$

Constants

- c_{jt}: Cost of producing a unit of power at plant $j \in J$ in time step $t \in T$.
- c_i^f: Price of remaining fuel at plant $i \in I$.

Variables

- y_{iwk}: Binary variable indicating if cycle k for plant i begins in week $w \in W_{ik}^o$
- r_{ik}: The amount of fuel reloaded in cycle k for plant i
- x_{iks}^b: Stock at the beginning of cycle k for plant $i \in I$ in scenario $s \in S$.
- x_{iks}^e: Stock at end of cycle k for plant $i \in I$ in scenario $s \in S$.
- x_{is}^f: Final stock for plant $i \in I$ in scenario $s \in S$.
- p_{itks}: Amount of power produced at plant $i \in I$ in cycle k at time step $t \in T$ in scenario $s \in S$.
- p_{jts}: Amount of power produced at plant $j \in J$ in time step $t \in T$ in scenario $s \in S$.
- $$\rho(i, w, k) := \begin{cases} 1 - \sum_{w' \leq w} y_{i,w',k+1}, & k = 0 \\ \sum_{w' \leq w - L_{ik}} y_{i,w',k}, & k = |K_i| - 1 \\ \sum_{w' \leq w - L_{ik}} y_{i,w',k} - \sum_{w' \leq w} y_{i,w',k+1}, & \text{otherwise} \end{cases}$$

  Note that $\rho(i, w, k)$ is not a variable, but is included for ease of exposition. The following relation holds $\rho(i, w, k) = 1 \iff$ cycle (i, k) is in a production campaign in week w.

Model

$$\min \sum_{i\in I}\sum_{k\in K} c_{ik}r_{ik} + \frac{1}{|S|}\sum_{s\in S}(\sum_{t\in T}\sum_{j\in J} c_{jts}F_t p_{jts} - \sum_{i\in I} c_i^f x_{is}^f) \quad (1)$$

$$\text{s.t. } r_{ik} \geq \underline{R}_{ik}\cdot \sum_{w\in W_{ik}} y_{iwk} \qquad \forall i\in I, \forall k\in K_i \quad (2)$$

$$r_{ik} \leq \overline{R}_{ik}\cdot \sum_{w\in W_{ik}} y_{iwk} \qquad \forall i\in I, \forall k\in K_i \quad (3)$$

$$\sum_{w\in W_{ik}} y_{iwk} \geq \sum_{w\in W_{i,k+1}} y_{i,w,k+1} \qquad \forall i\in I, \forall k\in K_i \quad (4)$$

$$\sum_{i\in C_m}\sum_{k\in K_i}\sum_{w\in IT_m}\sum_{w'=w-L_{ik}+1}^{w} y_{iw'k}\cdot \sum_{t=t(w)}^{t(w+1)-1} P_{it}^{max} \leq I_m^{max} \qquad \forall m\in M_{21}, \forall w\in W \quad (5)$$

$$\sum_{(iwk)\in H} y_{iwk} \leq K_H \qquad \forall H\in\mathcal{H} \quad (6)$$

$$x_{iks}^e = x_{iks}^b - \sum_{t\in T} p_{itks}\cdot F_t \qquad \forall i\in I, \forall k\in K_i, \forall s\in S \quad (7)$$

$$x_{iks}^b = r_{ik} + B_{ik}\sum_{w\in W_{ik}^o} y_{iwk} + \tilde{Q}_{ik}\left(x_{i,k-1,s}^e - B_{i,k-1}\sum_{w\in W_{ik}^o} y_{iwk}\right) \qquad \forall i\in I, \forall k\in K_i, \forall s\in S \quad (8)$$

$$x_{iks}^e \leq A_{i,k+1}^{max} + \left(1 - \sum_{w\in W_{ik}^o} y_{i,w,k+1}\right)\left(M_i^1 - A_{i,k+1}^{max}\right) \qquad \forall i\in I, \forall k\in K_i, \forall s\in S \quad (9)$$

$$x_{iks}^b \leq S_{ik}^{max} + \left(1 - \sum_{w\in W_{ik}^o} y_{iwk}\right)(M_i - S_{ik}^{max}) \qquad \forall i\in I, \forall k\in K_i, \forall s\in S \quad (10)$$

$$x_{is}^f \leq \sum_{k'>k}\sum_{w\in W_{ik}^o} y_{iwk'}M_i + x_{iks}^e \qquad \forall i\in I, \forall k\in K_i, \forall s\in S \quad (11)$$

$$p_{itks} \leq \overline{P}_{it}\cdot\rho(i,w(t),k) \qquad \forall i\in I, \forall k\in K_i, \forall t\in T_{ik}^p, \forall s \quad (12)$$

$$\underline{P}_{jts} \leq p_{jts} \leq \overline{P}_{jts} \qquad \forall j\in J, \forall t\in T, \forall s\in S \quad (13)$$

$$\sum_{i\in I}\sum_{k\in K_i(w(t))} p_{itks} + \sum_{j\in J} p_{jts} = D_{ts} \qquad \forall t\in T, \forall s\in S \quad (14)$$

$$y_{iwk}\in\{0,1\} \qquad \forall i\in I, \forall k\in K_i, \forall w\in W_{ik}^o \quad (15)$$

All variables other than y_{iwk} are continuous and non-negative. The objective (1) minimizes the sum of the costs of the reloading pattern and the sum of the production costs for each scenario, reduced by the profit for any remaining fuel for a scenario. Constraints (2) and (3) ensure the reloaded amount in any cycle is always within the possible reloading bounds. Constraints (4) ensure that if a cycle for a plant is set, then all preceding cycles must be set. Constraints (5) model CT21, and all other CT13-C20 constraints are modelled in the form of (6). These typically have a right hand side of one (i.e. give pairwise conflicts), but in some cases can exceed this (in the case of CT19 and CT20). Constraints (7) ensure stock level consistency between the starting stock level of a cycle and its end stock level (taking into account any production), while constraints (8) reflect the requirement that some fuel is lost as a plant goes through a reload. The CT11 constraints are enforced by constraints (9) and constraints (10) respectively, and (11) ensures the stock at the end of the last cycle is the plant's final stock level. Maximum and minimum production required by the respective plants are enforced by (12) and (13). Constraints (14) ensure all demand in each time step is met.

Two constraints, CT6 and CT12, are too complicated to include in the MIP model. The first states that once the fuel stock level at a given nuclear power plant falls below a certain threshold production must follow a piecewise linear decreasing function, while the second tries to ensure a high utilization of the nuclear power plants by stipulating that the average deviation of the production cannot be more than a certain tolerance from the maximum possible production level (however, only prior to the aforementioned threshold). As a result, these constraints are not enforced in the model, but rather in a post-processing step that attempts to repair a solution. This is described in Section 7.1.

4 Methodology

In this section we present a Benders Decomposition based framework to solve the compact formulation (1)-(15). We begin by providing a short introduction to Benders Decomposition in general before describing the Benders reformulation of (1)-(15). Once the necessary models have been introduced, we discuss, in detail, the components of the algorithm developed to solve this reformulation.

4.1 Benders Decomposition

Benders Decomposition is a well-known technique for solving large scale mixed integer programming (MIP) problems that have a special block structure (see Benders, 1962). It is commonly found in stochastic applications where one is required to make a so-called first stage decision and then, upon the realization of some random event, solve a second problem that ameliorates the first stage decision. This is often the case in the power industry, where the demand is highly stochastic. Recent applications of Benders in the power industry include (see Canto, 2008; Santos and Diniz, 2009; Cabero et al., 2010; Wu and Shahidehpour, 2010). However, it has also been applied in a variety of other areas including telecommunication network design (see Naoum-Sawaya and Elhedhli, 2010), staff scheduling (see Guyon et al., 2010), aircraft routing and crew planning (see Mercier et al., 2005), and uncapacitated hub location (see Contreras et al., 2010).

The Benders approach decomposes the original problem into a mixed integer *master* problem and one or more independent, linear *subproblems*. Consider the following formulation as an example.

$$
\begin{aligned}
\mu = \min \ & c^T x + f^T y \\
\text{s.t. } & \mathcal{A}x = b && (16)\\
& \mathcal{B}x + \mathcal{D}y = d && (17)\\
& x \in \mathcal{X} \subseteq \mathbb{R}^p, y \in \mathcal{Y} \subseteq \mathbb{R}^q,
\end{aligned}
$$

where x and y are vectors of decision variables with dimension p and q, $\mathcal{X}$ and $\mathcal{Y}$ are polyhedrons, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{D}$ are matrices, and c, f, b, and d are vectors (all with appropriate dimensions). The first set of constraints, (16) restrict the values of x, while the second set, (17) restrict the values of both x and y. With Benders Decomposition this problem is decomposed into the following two smaller problems, P1 and P2.

$$
\begin{aligned}
P1: \min \ & c^T x + z(x) \\
\text{s.t. } & \mathcal{A}x = b \\
& x \in \mathcal{X}
\end{aligned}
\qquad\qquad
\begin{aligned}
P2: z(x) = \min \ & f^T y \\
\text{s.t. } & \mathcal{D}y = d - \mathcal{B}x \\
& y \in \mathcal{Y}
\end{aligned}
\qquad (18)
$$

Observe that P1 is an optimization problem in terms of the x variables only, where $z(x)$ is the objective function value of P2 given the solution to P1. If one assumes that P2 is not unbounded, then one can also calculate $z(x)$ by solving it's dual formulation. If u denotes the vector of dual variables associated with constraints (18), then the dual formulation of P2 can be stated as:

$$
\begin{aligned}
\text{D2: } \max \ & u^T(d - \mathcal{B}x) \\
\text{s.t. } & D^T u \le f
\end{aligned}
$$

The feasible region of this optimization problem is completely independent of the values of x, which only affect the objective function. Assuming that the feasible region of D2 is not empty, then exactly one of two cases will occur when solving D2 for a given solution $\hat{x} \in \mathcal{X}$. Either D2 is unbounded from above, or D2 has a finite optimal solution. In the first case there must exist an extreme ray r_j such that $r_j^T(d - \mathcal{B}\hat{x}) > 0$, while in the second case there must exist an extreme point u_j of the feasible region such that $z(\hat{x}) = u_j^T(d - \mathcal{B}\hat{x})$. If we denote the set of all extreme rays of D2 as R and the set of all extreme points of D2 as U, then D2 can be restated as follows.

$$
\begin{aligned}
\text{D2*: } \min\ & z \\
\text{s.t. } & (r_i)^T(d - \mathcal{B}x) \leq 0 \ \forall r_i \in R && (19) \\
& (u_i)^T(d - \mathcal{B}x) \leq z \ \forall u_i \in U && (20)
\end{aligned}
$$

P2 now consists of the single variable z. The first set of constraints, (19), restricts the set of solutions to P1 to those which are also feasible for P2 (termed feasibility cuts), while the second set, (20), restrict the set of solutions to P1 to those that minimize the objective function value of P2 (termed optimality cuts). Hence, the original problem can be restated as:

$$
\begin{aligned}
\text{RMP: } \min\ & c^T x + z \\
\text{s.t. } & \mathcal{A}x = b \\
& (r_i)^T(d - \mathcal{B}x) \leq 0 \ \forall r_i \in R \\
& (u_i)^T(d - \mathcal{B}x) \leq z \ \forall u_i \in U \\
& x \in \mathcal{X}
\end{aligned}
$$

Since there can be an exponential number of constraints of the form (19) and (20), it is impractical to generate them all and include them initially. The so-called Restricted Master Problem (RMP) starts with a subset of these and dynamically identifies violated ones as needed. Thus, one usually adopts an iterative process where at any iteration a candidate solution (x^*, z^*) is found. The subproblem is then solved to calculate $z(x^*)$. If $z(x^*) = z^*$, the algorithm terminates, otherwise a violated feasibility or optimality cut exists. One adds the respective cut to the RMP and iterates again. In what follows we provide the Benders reformulation of (1)-(15).

4.2 Benders Reformulation

For the problem under consideration one can observe that once the reload dates and reload amounts have been fixed, one can independently solve each scenario and find the cheapest way of supplying the respective power demand power of each. That is, the problem naturally decomposes into n independent subproblems, where n is the number of different possible scenarios. Thus, the role of the master problem in this context is to identify good outage/reloading schedule. We model this as a MIP since it contains binary decision variables, which govern reload dates, and continuous variables that reflect the corresponding reload amounts. The Benders RMP (without the addition of any feasibility and optimality cuts) can be stated as follows.

Master problem

$$\min \sum_{i\in I}\sum_{k\in K} c_{ik} r_{ik} + \frac{1}{|S|}\sum_{s\in S}\theta_s \tag{21}$$

$$\text{s.t. } r_{ik} \geq \underline{R}_{ik} \cdot \sum_{w\in W_{ik}} y_{iwk} \qquad \forall i \in I, \forall k \in K_i \tag{22}$$

$$r_{ik} \leq \overline{R}_{ik} \cdot \sum_{w\in W_{ik}} y_{iwk} \qquad \forall i \in I, \forall k \in K_i \tag{23}$$

$$\sum_{w\in W_{ik}} y_{iwk} \geq \sum_{w\in W_{i,k+1}} y_{i,w,k+1} \qquad \forall i \in I, \forall k \in K_i \tag{24}$$

$$\sum_{i\in C_m}\sum_{k\in K_i}\sum_{w\in IT_m}\sum_{w'=w-L_{ik}+1}^{w} y_{iw'k} \cdot \sum_{t=t(w)}^{t(w+1)-1} P_{it}^{max} \leq I_m^{max} \qquad \forall m \in M_{21}, \forall w \in W \tag{25}$$

$$\sum_{(iwk)\in H} y_{iwk} \leq K_H \qquad \forall H \in \mathcal{H} \tag{26}$$

$$r_{ik} \geq 0 \qquad \forall i \in I, \forall k \in K_i \tag{27}$$

$$y_{iwk} \in \{0,1\} \qquad \forall i \in I, \forall k \in K_i, \forall w \in W_{ik}^o, \tag{28}$$

Associated with each scenario $s \in S$ is a decision variable θ_s that reflects the cost of supplying the power demanded in scenario. The constraints are as described in Section 3. In addition to the constraints described here, a number of additional constraints, which will be described in Section 6, are added to the master problem. Given a candidate solution (r, y, θ) to this problem, one can solve $|S|$ power production subproblems to separate any violated feasibility and optimality cuts. The structure of the subproblems that must be solved is given below

Subproblem (for given $s \in S$)

$$\min \sum_{t\in T}\sum_{j\in J} c_{j,t} F_t p_{j,t} - \sum_{i\in I} c_{i,|T|+1} x_i^f \tag{29}$$

$$\text{s.t. } x_{iks}^e = x_{iks}^b - \sum_{t\in T} p_{itks} \cdot F_t \qquad \forall i \in I, \forall k \in K_i, \forall s \in S \tag{30}$$

$$x_{iks}^b = r_{ik} + B_{ik} \sum_{w\in W_{ik}^o} y_{iwk} + \tilde{Q}_{ik}\left(x_{i,k-1,s}^e - B_{i,k-1}\sum_{w\in W_{ik}^o} y_{iwk}\right) \qquad \forall i \in I, \forall k \in K_i, \forall s \in S \tag{31}$$

$$x_{iks}^e \leq A_{i,k+1}^{max} + \left(1 - \sum_{w\in W_{ik}^o} y_{i,w,k+1}\right)\left(M_i^1 - A_{i,k+1}^{max}\right) \qquad \forall i \in I, \forall k \in K_i, \forall s \in S \tag{32}$$

$$x_{iks}^b \leq S_{ik}^{max} + \left(1 - \sum_{w\in W_{ik}^o} y_{iwk}\right)(M_i - S_{ik}^{max}) \qquad \forall i \in I, \forall k \in K_i, \forall s \in S \tag{33}$$

$$x_{is}^f \leq \sum_{k'>k}\sum_{w\in W_{ik}^o} y_{iwk'} M_i + x_{iks}^e \qquad \forall i \in I, \forall k \in K_i, \forall s \in S \tag{34}$$

$$p_{itks} \leq \overline{P}_{it} \cdot \rho(i, w(t), k) \qquad \forall i \in I, k \in K_i, t \in T_{ik}^p, \forall s \tag{35}$$

$$\underline{P}_{jts} \leq p_{jts} \leq \overline{P}_{jts} \qquad \forall j \in J, \forall t \in T, \forall s \in S \tag{36}$$

$$\sum_{i\in I}\sum_{k\in K_i(w(t))} p_{itks} + \sum_{j\in J} p_{jts} = D_{ts} \qquad \forall t \in T, \forall s \in S \tag{37}$$

Again each constraint is as described in Section 3. Each subproblem is modelled as a linear program (LP) and determines how much each power plant should produce in each time step so that the demand in each time step for the given scenario is satisfied and the various constraints regarding fuel stock levels are respected. In addition to this one must respect several production level bounds at each power plant. We remind the reader that two constraints, CT6 and CT12, were too complicated to include in the LP formulation and are instead enforced in a post-processing step that attempts to repair the subproblem solution. This is as mentioned earlier described in Section 7.1.

In typical Benders Decomposition fashion, optimality cuts are separated using solutions to each of the subproblems and are added to the master problem to direct it towards more promising outage/reloading schedules. In order to minimize the need for feasibility cuts to the master problem, constraints are preemptively added to the master problem and try to enforce CT11. These constraints also partly enforce CT6 and are discussed in Section 6.

4.3 Solution Approach

In this section we provide an overview of the algorithm we propose for solving the Benders reformulation. Here we simply provide a sketch of the approach, more detailed discussions on certain components of the algorithm are provided in the subsequent sections. The algorithm can be separated into three distinct phases, and we discus each in turn.

Stage 1 In this stage, the root node of the relaxed master problem is solved. The relaxed master problem is obtained by removing the integrality restriction on the y_{iwk} variables. Solving the root node is an iterative procedure between the master problem and the subproblems, where the subproblems are used to separate any violated feasibility and optimality cuts given a solution to the master problem. Note that we do not solve all subproblems per Benders iteration since this would simply take too long. A round robin approach is adopted in which only one subproblem is solved per Benders iteration. Since even solving a single instance of the subproblem can be quite time consuming, an aggregated version is used (see Section 5.2). In the aggregated subproblem, the time step is considered to be weeks as opposed to days or even hours. When no optimality cut has a magnitude of violation greater than some prespecified epsilon, or some predetermined time limit is reached, this stage terminates. Cplex 12.1 is used to solve both the master and the subproblems.

Stage 2 In the final stage of the algorithm the master problem is solved to integrality without the addition of anymore optimality cuts using a standard branch-and-bound technique. Cplex's populate routine is used to collect integeral solutions found in the branch-and-bound tree. Once a certain number of integer solutions have been found, all subproblems are solved to obtain a complete solution. However, the complete solution may violate CT6 and CT12. To remedy this, the solution to each subproblem is repaired so that CT6 and CT12 are satisfied. The routine to do this is described in Section 7.1. Once a complete solution satisfying all constraints has been found, a heuristic is used to improve its quality. This is detailed in Section 7.2. The best found solution is retained. Stage 2 continues until either all integer solutions from the branch-and-bound tree are enumerated, or a prespecified time limit is reached. The pseudo code for the complete methodology is given in Algorithm 1.

5 Reducing the problem size

As the problems may contain a huge number of variables, it is an advantage both with respect to computational time and memory consumption to reduce the problem size. In the following we describe two such reduction procedures.

5.1 Preprocessing

For the master problem employed, there is a y_{iwk} variable for each possible week w the outage of cycle k for plant i can occur. Because many of the constraints (CT13-CT21) concern these outage dates, many of them are infeasible, and removing them in a preprocessing step will reduce the size of the master problem. In the following we present a simple, yet effective preprocessing procedure.

Algorithm 1 Core Methodology

Preprocess problem instance
{Stage 1}
repeat
  Solve relaxed master problem
  Solve next aggregated subproblem
  Separate violated optimality/feasibility cut and append
until No violated optimality/feasibility cuts exist or time limit exceeded
{Stage 2}
Convert to MIP and run branch-and-bound
repeat
  Populate integral solution pool with a certain number of solutions
  for $s \in S$ **do**
    Solve subproblem associated with scenario s
    Repair subproblem solution
    Run 2-opt heuristic to improve solution quality
  end for
  if A feasible solution is found for each subproblem **then**
    Update best known solution if total cost is better than that of the current best solution
  end if
until All integer solutions have been enumerated or time limit exceeded

Let $G = (V, E)$ be a graph, where each node $v \in V$ corresponds to the outage date, w_v of some cycle, k_v, of plant i_v. There is an edge $(u, v) \in E$, if there is a conflict between the two corresponding outage dates, i.e., it is infeasible for cycle k_u to start its outage in week w_u while cycle k_v starts its outage in week w_v. How the conflicts are derived is explained later. For a set $S \subseteq V$ let $N(S) = \{v \in V \backslash S : \forall u \in S : \exists (u, v) \in E\}$, i.e., the set of nodes incident to all nodes in S. Now if $S \subseteq V$ is a set of nodes, for which it is known that at least one of the corresponding outage dates must be chosen in any solution, then the set $N(S)$ may be removed from the graph, as the corresponding outage dates can never be used. As it is known from the input data that some of the cycles must be scheduled, the set of nodes corresponding to the outage dates of these cycles can be used to perform the above described elimination.

Conflicts between outage dates are derived as follows:

1. All outage dates of the same cycle are in conflict.

2. Assume that the outage of cycle k of plant i occurs in week w, then the outage of any following cycle of the same plant must occur after week $w + L_{ik} - 1$ (see constraint CT13). This can be represented as conflicts between the individual outage dates.

3. Similarly assume that the outage of cycle k of plant i occurs in week w. Because of constraint CT11, the stock must be below $A^{max}_{i,k+1}$ before the outage of the next cycle can occur, and be below $S^{max}_{i,k+1}$ after the reload. Let LB be a lower bound on the stock at the beginning of cycle k, and let $UB(w_0, w_1)$ be an upper bound on the production capacity from week w_0 to week w_1 for plant i. A lower bound on the stock at any time after w, may then be calculated as $LBS(w_1) = LB - UB(w, w_1)$. The outage of cycle $k + 1$ must occur after

$$w_{min} = \arg\min\{w_1 : LBS(w_1) \leq A^{max}_{i,k+1} \wedge LBS(w_1) \leq f(S^{max}_{i,k+1}, \underline{R}_{i,k+1})\},$$

   where $f(x, r)$ returns the stock after a reload of r given end stock x, as specified by CT10. We set $LB = \underline{R}_{ik}$, and $UB(w_0, w_1)$ is calculated by assuming a production of $\overline{P}_{it}$ as long as the stock is above B_{ik}, and then the shutdown curve is followed. Again this can be represented as conflicts between the individual outage dates.

4. Constraints CT14-CT18 can be represented as conflicts between individual outage dates.

5. (Optional) The previous methods are exact in the sense that only outage dates which are infeasible are removed. These methods derive a large number of conflicts, and as a consequence a large number of outage dates may be removed. Even so, for some less tightly constrained instances (see Section 8 on computational results) this may not reduce the size of the problems enough and we thus include a heuristic for deriving conflicts. The working assumption for this heuristic is that it is not optimal to have a type 2 plant without stock for too long before the next reload occurs. Assume that the outage of cycle k of plant i occurs in week w. Let UB be an upper bound on the stock at the beginning of cycle k, and let $LB(w_0, w_1)$ be a lower bound on the stock which must be consumed from week w_0 to week w_1. Note that this lower bound is not zero because of constraint CT12. Let

$$w_{max} = (1 + \alpha) \arg\min\{w^1 : UB - LB(w, w^1) \leq 0\},$$

where $\alpha \geq 0$. We add conflicts between w and all outage dates $w' > w$ of the following cycle $k + 1$ for the same plant. The value α controls how long we allow a plant to lay idle in the worst case. As UB we use S_{ik}^{max}, and $LB(w_0, w_1)$ is calculated by assuming a production of zero until CT12 is violated, then production at $\overline{P}_{it}$ as long as the stock is above B_{ik}, and then the shutdown curve is followed.

In addition to the above conflicts we remove certain outage date as follows: Let k be a cycle *that does not necessarily* have to be scheduled. Let $w = T^A + L_{ik}$ be the latest point in time the production campaign of cycle k can start, let w_{max} be defined as above. We remove all outage dates $w' > w$ for the following cycle $k+1$. This may remove additional outage dates.

Additional conflicts can be deduced by the calculation described in point 3 above if the A_{ik}^{max}, S_{ik}^{max}, or R_{ik}^{max} values can be tightened. For any $i \in I$ and $k \in K_i$ the values may be tightened as follows:

$$\begin{aligned} A_{ik}^{max} &:= \min\{A_{ik}^{max}, \tilde{f}(S_{ik}^{max}, \underline{R}_{ik}\}) \\ S_{ik}^{max} &:= \min\{S_{ik}^{max}, f(A_{ik}^{max}, \overline{R}_{ik})\} \\ \overline{R}_{ik} &:= \min\{\overline{R}_{ik}, \hat{f}(0, S^{max})\} \end{aligned}$$

where $f(x, r)$ is as earlier, $\tilde{f}(y, r)$ gives the end stock which results in stock y after a reload of r as specified by CT10, and $\hat{f}(x, y)$ gives the reload which results in stock y given end stock x, as specified by CT10.

All conflicts of the conflict graph are added to the mater problem as clique constraints, which thus include the constraints CT13–CT18, and only CT19 and CT20 are included using the form (26). The complete preprocessing algorithm is sketched in Algorithm 2.

Algorithm 2 Preprocessing of outage dates

```
Tighten A^max and S^max values.
Construct conflict graph G.
repeat
  for all i ∈ I do
    for all cycles, k, of i which must be scheduled do
      Eliminate vertices of G.
    end for
  end for
until No vertices could be eliminated
```

5.2 Aggregation

Unlike the preprocessing technique described in Section 5.1, which attempts to remove as many redundant variables as possible from the master problem, the aggregation technique focuses on the subproblem and reduces the size of this problem by aggregating the individual time step production variables into variables that determine the weekly production level for each power plant (both type 1 and type 2). Since the time discretization of the master problem is weekly, one does not need the production levels for each individual time step (which can be as short as 4 hours) when solving the subproblem in the cutting phase of our methodology. This simple aggregation approach can dramatically reduce the size of the subproblem; the number of production variables can be reduced by a factor 42 at best. This primarily allows faster Benders iterations to be performed; however, it can also be used to determine the likelihood of finding a feasible solution to the subproblem. If the aggregated version is infeasible, then the disaggregated version will also be infeasible. The reverse, however, is not true. In Section 8 we assess the impact of using the aggregated version in the repair phase of the algorithm. Next, we formalize how both the aggregation and the necessary disaggregation are performed.

Minimal changes are required to model (29)-(37) in order to obtain the aggregated version. In introducing the weekly production variables p_{iwks} and p_{jws}, one is required to update the minimum and maximum production levels for each power plant, i.e. (35) and (36), the demand constraints, i.e. (37), and the production cost for each plant of type 2 to reflect the weekly structure. That is, (35), (36), and (37) become

$$p_{iwks} \leq \rho(i,w,k) \cdot \sum_{t \in w_t} \overline{P}_{it} \quad \forall (i,k,w \in W^p_{ik}, s) \tag{38}$$

$$\sum_{t \in w_t} \underline{P}_{jt} \leq p_{jws} \leq \sum_{t \in w_t} \overline{P}_{jts} \quad \forall (j,w,s) \tag{39}$$

$$\sum_{i \in I} \sum_{k \in K_i(w)} p_{iwks} + \sum_{j \in J} p_{jws} \geq D_{ws}, \quad \forall (w,s) \tag{40}$$

where $D_{ws} = \sum_{t \in w_t} D_{ts}$. The cost of each p_{iwks} variable is assumed to be the average cost of production for the aggregated time intervals, and we assume that the number of time steps per week is constant.

In order to provide a feasible solution to the subproblem in stage 2, any aggregated solution must be disaggregated (if this is possible). This routine works in a similar way to the repair heuristic described in Section 7.1. In an aggregated solution one has the weekly production levels of each plant which must be disaggregated into time step production levels in such a way that the demand of each time step is satisfied. Since each type 1 plant has a certain minimum production level in each time step, the procedure begins by first identifying the type 1 plant contribution to the demand in each of the time steps. The respective time step demands are then reduced accordingly. Next, the type 2 power plants are considered in order and an attempt is made to disaggregate their weekly production levels in each of their scheduled cycles. In this disaggregation step one proceeds by assigning the plant's maximum production level in each of the time steps, or the remaining demand for that time step, whichever is the smaller. If disaggregation fails (i.e. the assigned weekly production level for the plant cannot be met), an attempt is made to identify a time step (or as many as required) within the week for which there is unmet demand and for which the plant is currently not producing. If this cannot consume the surplus fuel, one repeats this process but looks across the weeks in the cycle. Finally, an attempt is made to push the remaining fuel to the subsequent cycle as long as CT11 is satisfied. If CT11 is violated, disaggregation is deemed not possible, although there is no guarantee that it is actually not possible. Once disaggregation has been successfully performed for each type 2 power plant, any unmet demand in any time step is satisfied by the cheapest type 1 power plant.

6 Feasibility

The following so-called stock-bounding constraints are added to the master problem in order to decrease the number of infeasible subproblems due to the CT11 constraints. One can think of

Algorithm 3 Disaggregation Algorithm

Require: A feasible solution to an aggregated subproblem
  for all $j \in J$ **do**
    for all $t \in T$ **do**
      Reduce the demand in time step t by the minimum required production level for plant j.
    end for
  end for
  for all $i \in I$ **do**
    for all cycles k of i that *must* be scheduled **do**
      for all weeks w of k **do**
        Disaggregate weekly production level
        if Surplus power remains **then**
          Try to consume the surplus fuel in the given week. If this is not possible, try to consume the fuel in the given cycle. If the remaining fuel still cannot be used, try to move it to the subsequent cycle. If a CT11 violation occurs, disaggregation is deemed impossible.
        end if
      end for
    end for
  end for

these constraints as adding an artificial stock variable to the master problem, which must satisfy these constraints given an upper bound on production. In addition, a number of constraints, not described here due to space consideration, are added which makes the complete solution less likely not to be repairable because of CT6, by bounding the amount of fuel which can be consumed between time periods when satisfying the shutdown curve.

Stock Bounding

- x^b_{ik}: lower bound on stock at the beginning of cycle (i, k).
- x^e_{ik}: lower bound on stock at the end of cycle (i, k).

$$x^e_{ik} \geq x^b_{ik} - \sum_{t \in T^p_{ik}} \overline{P}_{it} \cdot F_t \cdot \rho(i, w(t), k) \qquad \forall (i,k) \qquad (41)$$

$$x^b_{ik} = r_{ik} + BO_{ik} \sum_{w \in W_{ik}} y_{iwk} + \frac{Q_{ik} - 1}{Q_{ik}} \left(x^e_{i,k-1} - BO_{i,k-1} \sum_{w \in W_{ik}} y_{iwk} \right) \qquad \forall (i,k) \qquad (42)$$

$$x^e_{ik} \leq A^{max}_{i,k+1} + \left(1 - \sum_{w \in W_{ik}} y_{i,w,k+1} \right) \left(M^1_i - A^{max}_{i,k+1} \right) \qquad \forall (i,k) \qquad (43)$$

$$x^b_{ik} \leq S^{max}_{ik} + \left(1 - \sum_{w \in W_{ik}} y_{iwk} \right) \left(M^1_i - S^{max}_{ik} \right) \qquad \forall (i,k) \qquad (44)$$

Similar to in the subproblem, constraints (41) ensure stock level consistency between the starting stock level of a cycle and its end stock level assuming maximal production in all time steps, while constraints (42) reflect the requirement that some fuel is lost as a plant goes through a reload. The A^{max} bounds and S^{max} bounds are enforced by constraints (43) and constraints (44) respectively.

6.1 Stock Cuts

The Stock Cuts are introduced to enforce some of the structure of the shutdown curve on the stock bound variables $x^b_{ik}, x^e_{ik} : i \in I, k \in K$ defined above. The cuts are divided into three sets described in the following:

Cut-SI

$$x^b_{ik} - \sum_{w'>w} y_{i,w',k+1}\left(UB^1_{ik}(w,w') + A^{max}_{i,k+1}\right) \leq \left(1 - \sum_{w'>w} y_{i,w',k+1}\right) S^{max}_{ik} + (1 - y_{iwk})\, S^{max}_{ik} \quad \forall (i,w,k) \tag{45}$$

where $UB^1_{ik}(w,w') :=$ max production from w to w' assuming S^{max}_{ik} at time w with no intermediate refueling. $UB^1_{ik}(w,w')$ is bounded from above by S^{max}_{ik}.

There are three parts to these cuts:

- $y_{iwk} = 0$: This means that for plant i, week w is not the date of outage for cycle k. In this case the cut evaluates to $x^b_{ik} \leq S^{max}_{ik} + \rho$ with ρ being some positive number. This does not bound x^b_{ik} further than the already existing bound of S^{max}_{ik}.
- $y_{iwk} = 1$ and $\sum_{w'>w} y_{i,w',k+1} = 0$: This means that for plant i, week w is the date of outage for cycle k and for cycle $k+1$ there is no outage. In this case the cut evaluates to $x^b_{ik} \leq S^{max}_{ik}$, which does not bound x^b_{ik} further.
- $y_{iwk} = 1$ and $y_{i,w',k+1} = 1$ for some $w' > w$: This means that for plant i, week w is the date of outage for cycle k and for the following cycle $k+1$ week w' is the date of outage. In this case the cut evaluates to $x^b_{ik} - UB^1_{ik}(w,w') \leq A^{max}_{i,k+1}$, which ensures that the begin stock x^b_{ik} of cycle k is small enough for the maximum permitted fuel prior to reload in cycle $k+1$ is not violated, assuming maximum production in cycle k and no interactions from other plants $j \in I : i \neq j$.

Special case. For $k = 0$:

$$XI - \sum_{w'>w} y_{i,w',k+1}\left(UB^1_{ik}(w,w') + A^{max}_{i,k+1}\right) \leq \left(1 - \sum_{w'>w} y_{i,w',k+1}\right) XI \quad \forall (i,0,0) \tag{46}$$

Similar to above, several cases exist:

- $\sum_{w'>w} y_{i,w',k+1} = 0$: Evaluates to $XI \leq XI$, which requires no further comments.
- $y_{i,w',k+1} = 1$ for some $w' > w$: Evaluates to $XI - UB^1_{ik}(w,w') \leq A^{max}_{i,k+1}$. By following the argumentation above, this can be shown to be valid.

Cut-SII

$$\underline{R}_{ik} - \sum_{w'>w} y_{i,w',k+1} UB^2_{ik}(w,w') \leq x^e_{ik} + \left(1 - \sum_{w'>w} y_{i,w',k+1}\right) \underline{R}_{ik} + (1 - y_{iwk})\, \underline{R}_{ik} \quad \forall (i,w,k) \tag{47}$$

where $UB^2_{ik}(w,w') :=$ max production from w to w' assuming $\underline{R}_{i,k}$ at time w.

As for *Cut-SI* there are three parts to these cuts:

- $y_{iwk} = 0$: In this case the cut evaluates to $0 \leq x^e_{ik} + \rho$ with ρ being some positive number. This does not bound x^b_{ik} further than the already existing bound of 0.
- $y_{iwk} = 1$ and $\sum_{w'>w} y_{i,w',k+1} = 0$: In this case the cut evaluates to $0 \leq x^e_{ik}$, which does not bound x^b_{ik} further.

- $y_{iwk} = 1$ and $y_{i,w',k+1} = 1$ for some $w' > w$: In this case the cut evaluates to $\underline{R}_{ik} - UB^2_{ik}(w,w') \leq x^e_{ik}$, which ensures that the begin stock x^b_{ik} is large enough compared to the minimum fuel reload, assuming maximum production in cycle k and no interactions from other plants $j \in I : i \neq j$.

Special case. For $k = 0$:

$$XI - \sum_{w'>w} y_{i,w',k+1} UB^2_{ik}(w,w') \leq x^e_{ik} + \left(1 - \sum_{w'>w} y_{i,w',k+1}\right) XI \quad \forall (i,0,0) \tag{48}$$

Similar to above, several cases exist:

- $\sum_{w'>w} y_{i,w',k+1} = 0$: Evaluates to $0 \leq x^e_{ik}$, which requires no further comments.
- $y_{i,w',k+1} = 1$ for some $w' > w$: Evaluates to $XI - UB^2_{ik}(w,w') \leq x^e_{ik}$. By following the argumentation above, this can be shown to be valid.

Cut-SIII

$$x^b_{ik} - UB^1_{ik}(w,w') \leq x^e_{ik} + (2 - y_{iwk} - y_{i,w',k+1})\, S^{max}_{ik} \quad \forall i \in I, \forall k \in K, \forall w \in W_{ik}, \forall w' \in W_{i,k+1}, w < w' \tag{49}$$

where $UB^1_{ik}(w,w')$ is defined as before.

There are two parts to these cuts:

- $y_{iwk} + y_{i,w',k+1} \leq 1$: In this case the cut evaluates to $x^b_{ik} \leq x^e_{ik} + \rho$ with ρ being some positive number, since $UB^1_{ik}(w,w') \leq S^{max}_{ik}$. This is clearly dominated by the constraint $x^b_{ik} \leq x^e_{ik}$.
- $y_{iwk} = 1$ and $y_{i,w',k+1} = 1$: In this case the cut evaluates to $x^b_{ik} - UB^1_{ik}(w,w') \leq x^e_{ik}$, which ensures that the end lower bound on stock x^e_{ik} compared to the start lower bound on stock is not smaller than what can be explained by a maximum production in the cycle.

7 Postprocessing

The role of the postprocessing stage is to try to convert a solution, in the following also referred to as the reference solution, which does not satisfy the CT6 and CT12 constraints into one that does. This process is divided in two stages: in the first stage, called the repair stage, the production levels and reload amounts are altered in an attempt to satisfy CT6, and CT12, without violating any other constraints. If the solution can not be repaired, it is discarded. In the second stage, called the postoptimization stage, the production levels are shuffled between plants in an attempt to reduce the cost of the solution. The two stages are now described in further detail.

7.1 Repair

The input to this stage is a solution that satisfies all constraints except perhaps CT6, and CT12, i.e., the production curve may not follow the shutdown curve it should, or the maximum modulation is exceeded. The assumption is that the structure of the reference solution is good, and by making small adjustments, it is possible to make it satisfy these two additional constraints without changing the cost too much. Thus we want alterations to be as local as possible and since changes in start and end stock of a cycle propagates to the remaining cycles, the changes in these should be as small as possible. Satisfying CT6 means reducing production levels in some places, while satisfying CT12 means increasing production in some places. Changing the production levels from the reference solution, means that the stock levels passed from one cycle to the next will change from the reference solution. One observes that these changes in stocks can be kept small if a decrease of production in one time step of a cycle can be absorbed by an increase in production in another place of the cycle (see Figure 2 for an example). Changes in production within a cycle could also be absorbed by a change of the amount of fuel reloaded, but this is at odds with the principle of locality, as all scenarios are affected and previously repaired scenarios would thus have to be repaired again.

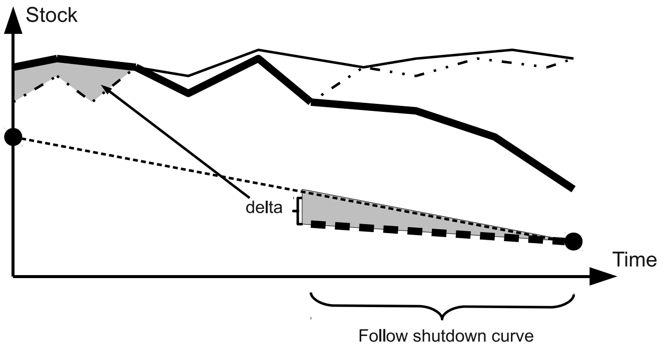

Figure 2: Example of repairing a cycle, such that the shutdown curve is respected by shuffling production to an earlier part of the cycle, such that the end stock remains the same. The upper think line is the production capacity, the upper dashed line is the production levels before repair, the thick line is the repaired shutdown curve. The lower think dashed line is the stock levels in the reference solution, while the lower thick dashed line is the stock levels assuming the shutdown curve is followed backwards from the end stock. δ is the extra stock that must be consumed earlier in the cycle for the end stock to remain the same. The gray area represents the increase in production in order to consume the extra stock.

We now describe the consequences on the remaining constraints, when the production levels are changed: Lowering the production will raise the end stock of the cycle, which may lead to CT11 becoming violated, either for that or a later cycle. Raising production will lower the end stock, which may lead to shortage of fuel in later cycles, where demand can no longer be met.

The repair procedure is divided in two stages: In stage 1 only type 2 plants are considered, and the production curves of these are adjusted so that they satisfy all constraints except perhaps the demand constraints. Then in stage 2, the production of type 1 plants is adjusted such that demand is covered. If any of the stages fail, the entire reference solution is discarded. We now describe these two stages in detail.

Stage 1 For each plant $i \in I$, each cycle $k \in K_i$ is treated one at a time, starting with the earliest. Each time a cycle is repaired one of two cases may happen:

1. No change in the stock levels at the start or end occurred. This means all changes of production levels within the cycle, were absorbed by increasing of decreasing production somewhere else within that cycle.

2. Given the repaired production levels, the end stock would be increased by δ. In this case the algorithm has two possibilities: either backtrack and try to have δ less stock at the beginning, i.e., consume δ more earlier, or push the stock excess to the next cycle. The algorithm first backtracks, and if this is not possible pushes the stock to the next cycle.

The repair algorithm is sketched in Algorithm 4, where x^b_{ik} and x^e_{ik} is the stock at the beginning and end of the cycle respectively, t^b_{ik} and t^e_{ik} is the beginning and end respectively of the production campaign of that cycle, and $\delta_{ik} \in \mathrm{R}$ is the amount of stock to clear, either from a backtrack from a later cycle, or from an earlier cycle.

Algorithm 4 Repair algorithm for a single cycle

Require: A plant $i \in I$ and cycle $k \in K_i$
  if is_backtrack **then**
    $x^e_{ik} := \max\{0, x^e_{ik} - \delta_{ik}\}$.
  else
    $x^s_{ik} := x^e_{ik} + \delta_{ik}$.
  end if
  Calculate shutdown curve backwards from x^e_{ik}. Let t_B and x_B be the resulting time step and stock right before entering this shutdown curve, and let x_{t_B} be the current stock at time t_B.
  Set $\delta := x_B - x_{t_B}$.
  Raise production by δ (if possible) in the time interval $[t_{ik}; t_B]$. Let δ be what is left.
  if $\delta = 0$ **then**
    Check if CT12 is violated, if so augment production. This may change the point of the shutdown curve and the end stock. Let x be the (new) end stock.
    $\delta_{i,k+1} := x - x^e_{ik}$
    Set is_backtrack := **false**.
    Set $x^e_{ik} := x$.
    Proceed with next cycle.
  else
    Since all production is at the upper bound, CT12 is satisfied.
    if $k = 0$ **then**
      Set can_backtrack := **false**
    end if
    if can_backtrack **then**
      Set $\delta_{i,k-1} := \delta$
      Backtrack to previous cycle.
    else
      Set $\delta_{i,k+1} := \delta$.
      Set $x^e_{ik} := x^e_{ik} + \delta$.
      Proceed with next cycle.
    end if
  end if

Stage 2 This stage is quite simple. For each $t \in T$ it is checked whether demand is either oversupplied or under-supplied. If demand is oversupplied the production of the most expensive type 1 plants are reduced, if demand is under-supplied the production of the least expensive type 1 plants are raised. It may happen that demand can not be met because of the bounds on the production of plant type 1. If so, an attempt to shuffle production within each cycle is made in a manner similar to the procedure described in the next section. If demand can not be met, the reference solution is discarded.

7.2 Postoptimization

The input to this stage is a solution that satisfies all constraints, and the role of the postoptimization is to try and reduce the cost of the solution by performing alterations, which do not lead to any new constraint violations but reduces the overall cost. As performing alterations which result in changes to the end stock of a cycle propagates, calculating the consequence of such alterations can be cumbersome, and we thus restrict our attention to alterations, where this is not the case.

One such alteration is the following: Let t_1 and t_2 be two points in time lying before the start of the shutdown curve within the same cycle for some $i \in I$, let $j_1, j_2 \in J$ be two plants such that $c_{j_1t_1} < c_{j_2t_2}$, and let $\delta = \min\{\overline{P}_{it_2} - p_{it_2}, p_{it_1}, \overline{P}_{j_1t_1} - p_{j_1,t_1}, p_{j_2,t_2} - \underline{P}_{j_2t_2}\}$, where p is the current production level. Then updating $p_{it_1} := p_{it_1} - \delta, p_{it_2} := p_{it_2} + \delta, p_{j_1t_1} := p_{j_1t_1} + \delta, p_{it_2} := p_{it_2} - \delta$, results in an improved cost while satisfying all constraints and not altering the end stock, nor the shutdowncurve of the cycle in question. The postoptimization heuristic is sketched in Algorithm 5.

Algorithm 5 Postoptimization heuristic

Require: A solution satisfying all constraints.
  for all $i \in I$ **do**
    for all cycles, k, of i **do**
      for some number of iterations **do**
        Select at random some t_1, t_2 lying within k and before the start of a shutdown curve.
        Select at random some $j_1, j_2 \in J$ such that $c_{j_1 t_1} < c_{j_2 t_2}$.
        $\delta = \min\{\overline{P}_{it_2} - p_{it_2}, p_{it_1}, \overline{P}_{j_1 t_1} - p_{j_1,t_1}, p_{j_2,t_2} - \underline{P}_{j_2 t_2}\}$,
        Update $p_{it_1} := p_{it_1} - \delta, p_{it_2} := p_{it_2} + \delta, p_{j_1 t_1} := p_{j_1 t_1} + \delta, p_{it_2} := p_{it_2} - \delta$
      end for
    end for
  end for

8 Computational results

In this section we present the computational experiments performed. The challenge instances are divided in three groups: data0–data5 are the initial instances used for the qualification phase, data6–data10 are the instances made public after the qualification phase, finally data11–data15 are the instances used for the final ranking of the competitors. These instances were not made available until after the end of the challenge. As only data instances data0–data10 were available, we restrict the experiments to these 11 instances, and consider only all the instances for the final computational results. For some experiments we further restrict our attention to a representative sample: data1, data5, data7, data8, and data10.

Table 2 lists the characteristics of the instance, where $|T|$ is the number of time steps, $|W|$ is the number of weeks, $|K|$ is the number of cycles, $|S|$ is the number of scenarios, $|J|$ is the number of plants of type 1, $|I|$ is the number of plants of type 2, and **13–21** are the number of corresponding CTXX constraints.

Setup The computational experiments were performed on a machine with 2 Intel(R) Xeon(R) CPU X5550 @ 2.67GHz (16 cores in total), with 24 GB of RAM, and running Ubuntu 10.4. The version of CPLEX used is 12.1.

Preprocessing We here examine the effect of the preprocessing described in Section 5.1. Table 3 shows the results: It gives the total number of possible outage dates before preprocessing (**Total**), and the percentage of these removed by the preprocessing (**Rem.**). As can be seen the preprocessing is very effective removing $80\% - 90\%$ of the possible outage dates for all instances except the very small instance data0, and data8 and data9. For the two latter the number of variables is around twice as large as the largest of the other instances, and fewer variables are removed (around 63% and 68% respectively). The reason can be gleamed from Table 2: data8 and data9 have much fewer CT13 constraints, i.e., few cycles *must* be scheduled, which means the problem is less constrained and eliminating outage dates is harder.

Heuristic preprocessing As mentioned earlier, the preprocessing is very effective, but it is still struggles on certain large instances (data8 and data9), where few cycles *have* to be scheduled. For this reason the heuristic conflict detector described as point 5 in Section 5.1 is included. As described there the value of α controls the aggressiveness of the heuristic (lower values means more conflicts). We here examine the effect of including the heuristic preprocessing. The value of α is fixed to 0.05 (for smaller values some instances were infeasible). Table 4 shows the percentage of variables removed (**Rem.**) and the final solution (**Sol.**), without the heuristic (**No**

Table 2: Characteristics of the problem instances

Name	$\|T\|$	$\|W\|$	$\|K\|$	$\|S\|$	$\|J\|$	$\|I\|$	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
data0	623	89	2	2	1	2	4	1	0	0	0	0	0	0	0
data1	1750	250	6	10	11	10	46	7	0	1	3	0	1	1	1
data2	1750	250	6	20	21	18	84	13	0	1	3	0	1	1	1
data3	1750	250	6	20	21	18	80	10	2	1	3	2	1	1	1
data4	1750	250	6	30	31	30	122	19	0	1	3	0	1	1	1
data5	1750	250	6	30	31	28	120	18	0	1	3	0	1	1	3
data6	5817	277	6	50	25	50	222	33	40	1	3	0	1	50	5
data7	5565	265	6	50	27	48	192	31	35	1	3	0	1	50	5
data8	5817	277	6	121	19	56	114	37	45	1	3	0	1	50	5
data9	5817	277	6	121	19	56	114	37	45	1	3	0	1	50	5
data10	5565	265	6	121	19	56	235	37	45	1	3	0	1	50	5
data11	5817	277	6	50	25	50	239	33	40	1	3	0	1	50	5
data12	5523	263	6	50	27	48	207	31	35	1	3	0	1	50	5
data13	5817	277	6	121	19	56	260	37	45	1	3	0	1	50	5
data14	5817	277	6	121	19	56	256	37	45	1	3	0	1	50	5
data15	5523	263	6	121	19	56	245	37	45	1	3	0	1	50	5

Table 3: Number of variables removed by preprocessing

Name	**Total**	**Rem.**	**Name**	**Total**	**Rem.**
data0	36	28.78%	data6	24683	85.65%
data1	3920	87.53%	data7	35817	80.61%
data2	7941	88.47%	data8	69481	68.03%
data3	8207	89.83%	data9	69136	62.70%
data4	17514	89.41%	data10	30061	85.43%
data5	15415	82.13%			

Heur.), and with the heuristic (**Heur.**) given 3600 seconds respectively. For this experiment the stock constraints were added as described for Run 6 in Table 6, aggregation was enabled and postoptimization was disabled. As can be seen, the effect on the final solution quality is minor for data1–data3, for data4–data5 the solution is improved, while it is worse for data6, data7, and data10, finally we are now able to provide a solution for data8, which was not possible earlier.

Aggregation Next we examine the effect of aggregation on the solution quality. To that effect the algorithm is run for 3600 seconds with aggregation enabled and disabled for stage 2 (aggregation is always performed for stage 1). Table 5 shows the results, where **Vars.** is the number of variables in the subproblem, **Cons.** is the number of constraints in the subproblem, and **Sol.** is the final solution. For this experiment the stock constraints were added as described for Run 7

Table 4: Effect of using heuristic preprocessing for different values of α

	No Heur.		Heur.	
Name	**Rem.**	**Sol.**	**Rem.**	**Sol.**
data0	28.78%	**8.7371e12**	28.78%	**8.7371e12**
data1	87.53%	1.6990e11	87.63%	**1.6971e11**
data2	88.47%	**1.4654e11**	88.64%	1.4672e11
data3	89.83%	**1.5537e11**	89.92%	1.5578e11
data4	89.41%	1.1342e11	89.64%	**1.1309e11**
data5	82.13%	1.3272e11	83.28%	**1.3153e11**
data6	85.65%	**9.0945e10**	85.72%	9.2508e10
data7	80.61%	**1.2307e11**	80.68%	1.3663e11
data8	68.03%	–	77.15%	**3.2392e12**
data9	62.70%	–	74.13%	–
data10	85.43%	**1.5303e11**	85.44%	1.7455e11

in Table 6, heuristic conflicts were included with $\alpha = 0.05$ and postoptimization was disabled. As can be seen the aggregation results in a big reduction in the number of variables and constraints of the subproblem. For data8, and data10 no solution is found without the use of aggregation.

Table 5: Effect on final solution of aggregating versus not aggregating in stage 2

	Enabled			Disabled		
Name	**Vars.**	**Cons.**	**Sol.**	**Vars.**	**Cons.**	**Sol.**
data1	5514	8693	1.**6971e11**	37692	58871	1.6968e11
data5	16696	25199	1.3147e11	114346	170849	**1.3100e11**
data7	25898	34181	1.3663e11	529438	686121	**1.3412e11**
data8	41762	48309	**3.2392e12**	860182	977529	–
data10	23150	29457	**1.7455e11**	469330	581637	–

Stock constraints We here examine the effect of including the constraints described in Section 6, in an attempt to ensure the feasibility of the subproblem. One can choose to include either all the constraints, or only a subset, and to included them only in stage 2 or in both stages. Nine runs are performed, with the settings described in Table 6. The results can be seen in Table 7. For this experiment heuristic conflicts were included with $\alpha = 0.05$, aggregation was enabled and postoptimization was disabled. As can be seen only Run 7 and Run 10 completes for all the tested instances. Run 10 achieves the best average results.

Postoptimization We here examine the effect of the postoptimization procedure described in Section 7.2. For each of the 11 instances three runs are performed, with the number of iterations respectively set to $50,000$, $150,000$, and $300,000$. The results can be seen in Table 8. For this experiment the stock constraints were added as described for Run 10 in Table 6, heuristic conflicts

Table 6: Description of the different settings used for the stock constraint runs

Run	Description
1	No stock constraints included
2	Stock constraints (41)–(44) included in stage 1.
3	Stock constraints (41)–(44), SI and SII included in stage 1.
4	All stock constraints included in stage 1.
5	Stock constraints (41)–(44) included in stage 2.
6	Stock constraints (41)–(44), SI and SII included in stage 2.
7	All stock constraints included in stage 2.
8	Stock constraints (41)–(44) included in stage 1, SI and SII included in stage 2.
9	Stock constraints (41)–(44) included in stage 1, remaining included in stage 2.
10	Stock constraints (41)–(44), SI and SII included in stage 1, remaining included in stage 2.

Table 7: Effect of including stock constraints. See Table 6 for a description of each run.

Name	Run 1	Run 2	Run 3	Run 4	Run 5
data 1	1.6986e11	1.6987e11	1.6981e11	1.6986e11	1.6973e11
data 5	–	1.2656e11	**1.2593e11**	1.2707e11	1.3163e11
data 7	–	1.3535e11	**1.0514e11**	1.2517e11	1.3363e11
data 8	–	–	–	2.8047e12	–
data10	–	1.3055e11	1.3785e11	1.3667e11	**1.1532e11**
Avg.	–	–	–		–

Name	Run 6	Run 7	Run 8	Run 9	Run 10
data 1	**1.6972e11**	**1.6972e11**	1.6987e11	1.6979e11	1.6977e11
data 5	1.3039e11	1.3148e11	1.2656e11	1.2654e11	**1.2593e11**
data 7	1.3392e11	1.3663e11	1.4035e11	1.3373e11	1.0563e11
data 8	–	3.2393e12	–	–	**2.1963e12**
data10	1.5162e11	1.7455e11	1.4273e11	1.6093e11	1.3864e11
Avg.	–	7.7034e11	–	–	5.4725e11

were included with $\alpha = 0.05$. The reason for the factor 10 increase in solution quality for data8 over previous results, is a newer version of the code, where the conflict graph more effectively is added as cliques. Due to time constraints the previous tests could not be rerun. As can be seen there is a clear correlation between the number of postoptimization iterations and the final solution quality.

Table 8: Effect of postoptimization procedure. The number of iterations for the runs are respectively 50,000, 150,000, and 300,000

Name	Run 1	Run 2	Run 3
data1	1.69711e11	1.69710e11	**1.69709e11**
data5	1.26452e11	1.26453e11	**1.26451e11**
data7	1.10602e11	1.09518e11	**1.09013e11**
data8	2.88400e11	2.79264e11	**2.75348e11**
data10	1.30261e11	1.28788e11	**1.28443e11**

Time We finally examine the solution quality as a function of total time given to the algorithm. Each of the 16 instance is run for respectively 3600 seconds, and 10800 seconds. The results can be seen in Table 9, and Table 10 respectively, where the number in parenthesis is the deviation from the best known solutions reported on the ROADEF/EURO 2010 challenge website (`http://challenge.roadef.org/2010`). For the challenge a maximum time of 1800 seconds was allowed for the first 6 instances, and 3600 seconds for the remaining 10. Each table gives the following information: the number of optimality cuts added (**#Cuts**), the number of solutions found in stage 2(**#Sols**), the number of solutions found in stage 2 that were repairable (**#Rep**), the solution value (**Sol.**), the percentage deviation from the best known solution (**#Dev**), and the average deviation for the two test sets of five instances (**#Avg**).

As can be seen from the tables, the solution approach performs satisfactorily on instances zero to five. These were the test instances used in the qualification phase of the contest and are less complicated than the second and third set of instances (data6 to data10, and data11 to data15). For the latter sets, the algorithm runs into difficulty due to the large number of binary variables, particularly for data8 and data9 which are far from the best known solutions, and for data13 which is not solved at all. Furthermore, formulating and solving the subproblem as an LP and repairing its solution so that it satisfies CT6 and CT12 appears to be an expensive process, despite the aggregation.

For the the smaller instances, many of the solutions are repairable, while for the larger instances, there is a lot more variation. It is sunrising that for two of the instances where the algorithm performs poorly (data8 and data13), there is a large number of solutions found but only two are repairable in one case, while none in the other. Generally it appears that for the larger instances, either the solutions to the master problem can not be adjusted such that they satisfy CT6 and CT12, or the repair algorithm does a poor job.

Doubling the amount of time (Table 10) does not significantly change the results and only data1, data5, and data6 are improved. The trend of finding many solutions which are non-repairable remain the same.

9 Conclusion

In conclusion, we have developed a Benders Decomposition approach to solve the large scale energy management problem posed for the ROADEF/EURO 2010 challenge. The approach includes a MIP model of the problem along with additional constraints for ensuring feasibility of the subproblems, a very effective preprocessing and aggregation scheme, which reduces the size of the problem significantly, and an algorithm for repairing a solution which only satisfies a subset of the constraints.

On the first set of instances the approach is competitive, while on the the second two set of instances it is not. This is mainly due to the size of the problems, and the time allotted. On the second set of instances and 5 blind instances we placed 14th out of 19 teams in the final of

Table 9: Results for different problem instances given 3600 seconds

Name	**#Cuts**	**#Sols.**	**#Rep.**	**Sol.**	**Dev.**	**Avg.**
data0	5	2517	2517	8.7372e12	0.0709%	
data1	6	352	312	1.6971e11	0.1008%	
data2	17	82	82	1.4629e11	0.1639%	
data3	14	86	86	1.5475e11	0.2050%	
data4	23	36	35	1.1206e11	0.4157%	
data5	21	38	37	1.2645e11	0.4997%	0.2427 %
data6	14	12	12	9.0113e10	8.0173%	
data7	5	121	1	1.0901e11	34.2953%	
data8	3	1486	2	2.7535e11	236.0938%	
data9	9	7	3	3.5103e12	4193.8891%	
data10	37	7	5	1.2844e11	62.3461%	906.9283 %
data11	61	447	11	8.8464e10	14.0143%	
data12	20	12	12	8.8135e10	15.2850%	
data13	16	1017	0	–	–%	
data14	10	8	4	1.0092e11	32.4820%	
data15	46	35	2	1.5758e11	109.8220%	42.9008%

the competition. One of the few optimal methods proposed, it was unable to compete with the heuristics given only 3600 seconds of computing time. The sophisticated approach can, however, provide information as to the quality of solutions through the lower bound information which can be obtained at each iteration of the Benders algorithm as well as insights into the structure on the problem.

Table 10: Results for different problem instances given 10800 seconds

Name	#Cuts	#Sols.	#Rep.	Sol.	Dev.	Avg.
data0	5	2517	2517	8.7372e12	0.0709%	
data1	6	872	583	1.6971e11	0.1007%	
data2	17	153	153	1.4629e11	0.1639%	
data3	14	165	165	1.5475e11	0.2050%	
data4	23	73	72	1.1206e11	0.4156%	
data5	21	76	74	1.2643e11	0.4842%	0.2401 %
data6	14	23	23	8.9659e10	7.4733%	
data7	5	481	22	1.0901e11	34.2953%	
data8	3	4292	2	2.7535e11	236.0938%	
data9	9	12	10	3.5103e12	4193.8891%	
data10	37	10	9	1.2844e11	62.3461%	906.8179 %
data11	61	458	22	8.8464e10	14.0143%	
data12	20	25	25	8.8135e10	15.2850%	
data13	16	2187	0	–	–%	
data14	10	118	8	1.0092e11	32.4820%	
data15	46	126	2	1.5758e11	109.8220%	42.9008%

References

Jacques F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4:238 – 252, 1962.

Jordi Cabero, Mariano J. Ventosa, Santiago Cerisola, and Álvaro Baíllo. Modeling risk management in oligopolistic electricity markets: A benders decomposition approach. *IEEE Transactions on power systems*, 25(1):263 – 271, 2010.

Salvador Perez Canto. Application of benders decomposition to power plant preventive maintenance scheduling. *European Journal of Operational Research*, 184:759 – 777, 2008.

Ivan Contreras, Jean-François Cordeau, and Gilbert Laporte. Benders decomposition for large scale uncapacitated hub location. Technical Report CIRRELT-2010-26, Interuniversity research centre on enterprise networks, logistics, and transportation, 2010.

Olivier Guyon, P Lemaire, Eric Pinson, and David Rivreau. Cut generation for an integrated employee timetabling and production scheduling problem. *European Journal of Operational Research*, 201:557 – 567, 2010.

Anne Mercier, Jean-François Cordeau, and François Soumis. A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers and Operations Research*, 32:1451 – 1476, 2005.

Joe Naoum-Sawaya and Samir Elhedhli. A nested benders decomposition approach for telecommunication network planning. *Naval Research Logistics*, 57:519 – 539, 2010.

Marc Porcheron, Agns Gorge, Olivier Juan, Tomas Simovic, and Guillaume Dereu. Challenge roadef/euro 2010 : A large-scale energy management problem with varied constraints. http://challenge.roadef.org/2010/sujetEDFv22.pdf, 2009.

T. N. Santos and A. L. Diniz. Feasibility and optimality cuts for the multi-stage benders decomposition approach: Application to the network constrained hydrothermal scheduling. In *Power & Energy Society General Meeting, 2009. PES '09. IEEE*, 2009.

Lei Wu and Mohammad Shahidehpour. Accelerating the benders decomposition for network-constrained unit commitment problems. *Energy Systems*, 1:339 – 376, 2010.

We present a Benders' decomposition based framework for solving a large scale energy management problem with varied constraints posed as the ROADEF/EURO 2010 challenge. Because of the nature of the problem, not all constraints can be modeled satisfactorily as linear constraints and the approach is therefore divided into two stages: in the first stage Benders feasibility and optimality cuts are added based on the linear programming relaxation of the Benders Master problem, and in the second stage feasible integer solutions are enumerated and procedure is applied to each solution in an attempt to make them satisfy the constraints not part of the mixed integer program. A number of experiments are performed on the available benchmark instances. These experiments show that the approach is competitive on the smaller instances, but not for the larger ones. We believe the exact approach gives insight into the problem and additionally makes it possible to find lower bounds on the problem, which is typically not the case for the competing heuristics

DTU Management Engineering
Department of Management Engineering
Technical University of Denmark

Produktionstorvet
Building 424
DK-2800 Kongens Lyngby
Denmark
Tel. +45 45 25 48 00
Fax +45 45 93 34 35

www.man.dtu.dk