J Sched
DOI 10.1007/s10951-012-0281-1

A constraint programming-based approach to a large-scale energy management problem with varied constraints

A solution approach to the ROADEF/EURO Challenge 2010

Felix Brandt · Reinhard Bauer · Markus Völker · Andreas Cardeneo

Abstract This paper addresses a large-scale power plant maintenance scheduling and production planning problem, which has been proposed by the ROADEF/EURO Challenge 2010. We develop two lower bounds for the problem: a greedy heuristic and a flow network for which a minimum cost flow problem has to be solved.

Furthermore, we present a solution approach that combines a constraint programming formulation of the problem with several heuristics. The problem is decomposed into an outage scheduling and a production planning phase. The first phase is solved by a constraint program, which additionally ensures the feasibility of the remaining problem. In the second phase we utilize a greedy heuristic—developed from our greedy lower bound—to assign production levels and refueling amounts for a given outage schedule. All proposed strategies are shown to be competitive in an experimental evaluation.

F. Brandt (✉) · A. Cardeneo
Logistics Systems Engineering, Research Center for Information Technology, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany
e-mail: brandt@fzi.de

A. Cardeneo
e-mail: cardeneo@fzi.de

R. Bauer · M. Völker
Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Box 6980, 76128 Karlsruhe, Germany

R. Bauer
e-mail: reinhard.bauer@kit.edu

M. Völker
e-mail: markus.voelker@kit.edu

Keywords Maintenance scheduling · Constraint programming · Lower bound · Flow network · Greedy heuristic

1 Introduction

In this work we take the perspective of a large utility company, tackling their problems in modeling and planning production assets, i.e., a multitude of power plants. Our goal is to fulfill the respective demand of energy over a time horizon of several years, while minimizing the total operating cost of all machinery.

Determining optimal maintenance schedules and production plans is not easy because of the number of alternatives to assess. As the exact electricity demand of each forthcoming day is unknown and depends on a large variety of factors (season, weather, holidays, etc.), this leads to the need of multiple uncertainty scenarios. Additionally, the increasing proportion of renewable energies in today's energy mix makes things more complicated for an utility company, because it has to feed the energy of third-party solar or wind power plants into its electricity networks and regulate its own power plants accordingly.

The problem discussed in this paper was proposed by the ROADEF/EURO Challenge 2010 (Pocheron et al. 2010), a competition announced by the French Operational Research and Decision Support Society (ROADEF) and the European Operational Research Society (EURO). The model has been developed by the French utility company Electricité de France (EDF).

The examined problem comprises three fields of optimization: maintenance scheduling, production planning and determining refueling amounts. It is a tactical model, neither

Published online: 07 August 2012

considering short-term operational restrictions (like intra-day load following) nor containing strategic decisions (like adding new power plants). However, the proposed model allows a generic formulation of other concerns like electricity network stability, safety considerations, availability of staff and tools, as well as legal restrictions. All of these limitations can be expressed as mathematical constraints.

1.1 Our contribution

We present two lower bounds for the overall problem. One is a fast greedy heuristic, the other is based on a flow network and yields closer bounds at the cost of high computational effort.

Our presented solution approach decomposes the problem into an outage scheduling phase and a power assignment phase. First, valid outage schedules are found by a CP approach with a randomized branching strategy (based on marginal cost) that prefers cheaper outage dates. All production level and refueling constraints are modeled into the CP in an approximate fashion to nearly guarantee the feasibility of the remaining problem. The latter phase utilizes a greedy production assignment and refueling heuristic, which is based on our greedy lower bound. Finally, rerunning the second phase with the realized cost from the first run yields our best results. As these procedures run very fast, the CP is solved in a randomized fashion and the whole process is repeated until a given time limit is reached.

Along with the presented solution methods we show our experimental results. All considered problem instances have been provided by the challenge and are extracted from real world data. Our solution approach performs competitive and provides good results for all proposed instances.

1.2 Related work

There exists a vast amount of literature concerning production planning of power generating facilities. Different models have been proposed that track either maintenance scheduling (Foong et al. 2008; Satoh and Nara 1991) or production planning (Ngundam et al. 2000; Dentcheva and Römisch 1997). A more general problem addressed frequently in the literature is the unit commitment problem (Padhy 2004; Feltenmark et al. 1996; Sen and Kothari 1998). When applied to power systems, the objective usually is to find low-cost short-term production plans (typically between 24 and 168 hours).

A former model proposed by EDF, comprising a subset of constraints and only considering maintenance scheduling, was solved by a combination of constraint programming and local search (Khemmoudj et al. 2006). While both models only accept solutions that fulfill all of the problem's constraints, we note that most other models from the literature employ penalties if constraints are violated, rather than looking for exact solutions.

Meanwhile, there also exists some literature considering the ROADEF/EURO Challenge 2010. Godskesen et al. (2010) prove the NP-hardness of the problem by describing a reduction from 1-in-3-SAT. Additionally, they describe a solution approach which consists of three phases. First, a simplified constraint programming model of the problem is solved. The solution is then improved by local search and in a final step overproduction is handled in greedy fashion. The experimental results indicate that the presented approach is very competitive. Lusby et al. (2010) describe a Benders decomposition approach for the problem. To cope with the non-linearity of the problem, their approach is divided into two stages. The first phase is used to produce feasible integer solutions for a subset of the constraints, in the second phase it is checked whether these solutions can be modified such that they also satisfy the remaining constraints. Given sufficient time, the algorithm returns optimal solutions. However, according to the authors it was unable to compete with the heuristical approaches for large instances due to the time limit. Gardi and Nouioua (2011) use a randomized local search technique to tackle the problem. They especially focus on an efficient evaluation of the effects of local search moves. This makes it possible to visit more feasible solutions within the given time limit. According to the provided experimental results, this approach also seems to be very competitive.

Finally, we note that this text is based on the thesis (Brandt 2010).

1.3 Overview

In the following we briefly outline the structure of this paper: we start with a sketch of the considered problem in Sect. 2. Two methods for computing lower bounds are developed in Sect. 3. The structure of the solution approach is reported in Sect. 4 while Sects. 5 and 6 give details on the first and second phase of the approach, respectively. Finally, the experimental results are stated in Sect. 7 followed by a conclusion in Sect. 8.

2 Problem statement

The given model extends over a period of time (e.g., 5 years). This period is split into uniform *time steps* of configurable length (e.g., 1 day). The main entities of the model are a set of various power plants and a set of uncertainty demand *scenarios*. For each scenario we are looking for a production assignment, such that the sum of energy produced by all available power plants equals the demand during each time step. The need for multiple scenarios arises

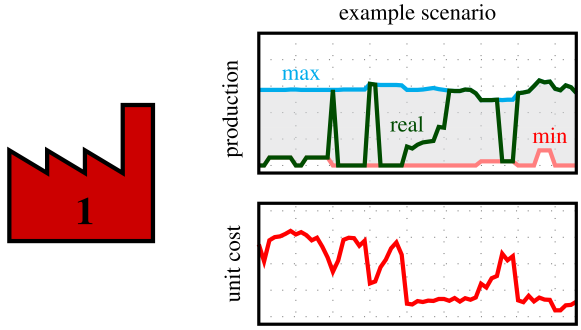

Fig. 1 Sample plots illustrating the development of production and unit cost over time of a type 1 power plant for an exemplary scenario. The *upper plot* shows the minimum/maximum allowed production levels together with a valid production plan. The *lower plot* shows the cost per unit of energy produced by this plant

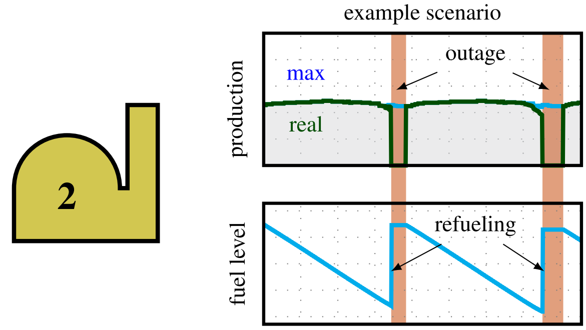

Fig. 2 Sample plots illustrating the production and fuel levels over time for a type 2 power plant. The *upper plot* shows maximum and actual production levels. The *gray area* represents the allowed production interval. The corresponding fuel level is depicted in the *lower plot*. Note that outage dates and refueling amounts are equal in all scenarios

from the numerous uncertainties that have to be taken into account such as unknown energy demand, generation units availability or spot market prices.

In the considered model there are two kinds of facility. The first, *type 1 power plants*, can operate continuously and their fuel supply is outside the scope of the problem. During each time step they can produce an amount of energy from a certain range that depends on the scenario and time step (see Fig. 1). Production at these power plants induces costs that are proportional to the power output and also depend on the scenario and time step. Power plants of this type might be coal- or gas-fired, or even virtual power plants for exporting and importing energy, whose available power levels and unit costs we cannot influence.

Facilities of the second kind, *type 2 power plants*, represent nuclear power plants. These plants have only limited capacities to store fuel and, as running out of it would stop the production of energy, they have to be refueled regularly (see Fig. 2). *Refueling* can only take place during an *outage*, i.e., when a plant is offline for several weeks, and is more complex than just adding fuel—a certain amount of fuel has to be removed to make the addition of new fuel possible (see Eq. (9)). Hence, the operation of a type 2 power plant is organized in so called *cycles*—successions of an offline period (an outage with refueling and maintenance) and the following *production campaign*. The event of taking a plant offline (online) is called *decoupling* (*coupling*). It is not always necessary to schedule all cycles of a type 2 power plant before the end of the considered time horizon, i.e., outages can be postponed. But the order of a power plant's cycles is fixed and so all successive cycles would have to be postponed, too.

There are two specialties of type 2 power plants included in the model. Firstly, running a plant at less than the specified maximum production amount, called *modulation*, is undesirable for technical reasons. Therefore, the aggregated modulation during a production campaign is limited. Secondly, if the fuel level of a plant is below a certain threshold, then the production level of the plant has to follow a given profile, called *imposed power profile*, and cannot any longer be chosen freely until the next outage.

Contrary to type 1 power plants, production at type 2 power plants is billed via the amount of fuel reloaded. In the considered model, the fuel unit cost depends on the cycle and power plant. For each type 2 power plant a number of (fuel-related) production constraints apply. Furthermore, outages of different type 2 power plants are dependent on each other to fulfill resource, staff, grid stability, production safety and legal restrictions.

To conclude, the proposed subject consists of modeling the production assets and finding an optimal outage schedule while satisfying all the given constraints. It includes three dependent subproblems: determining a schedule of type 2 power plant outages, choosing a refueling amount for each outage and setting up a production plan, i.e., a quantity of energy to produce by each plant at each time step for each scenario. The objective is to minimize the expected cost of production.

A complete description of the model can be found in (Brandt 2010) and the ROADEF/EURO Challenge 2010 subject document (Pocheron et al. 2010). A sample production plan is given as Fig. 3.

2.1 General definitions

2.1.1 Indices

A dataset comprises various sets of entities. Let $\mathcal{S}$ denote the set of demand scenarios, $\mathcal{J}$ the set of type 1 power plants and $\mathcal{I}$ the set of type 2 power plants. The timeline consists of T uniform time steps spanning over W weeks in total, where T is a multiple of W. The corresponding set for the time steps is defined as $\mathcal{T} = \{0, \dots, \mathrm{T}-1\}$ and for the weeks as $\mathcal{W} = \{0, \dots, \mathrm{W}-1\}$.

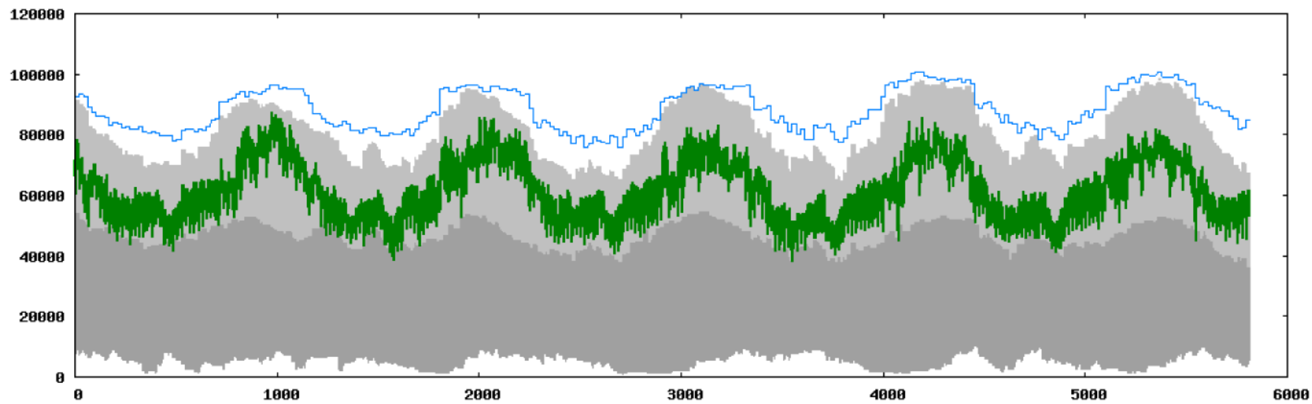

Fig. 3 A sample production plan showing the units of demand (*thick line in the middle*) per time step and the available capacity of all type 1 (*light gray*) and type 2 (*dark gray*) power plants. The *top line* indicates the gross production capacity of all plants, i.e. without any outages. Note that in the presented plan there is a trend to schedule outages during times of low demand (data are taken from instance 08, scenario 0)

Let K denote the number of cycles of each type 2 power plant and $\mathcal{K} = \{0, \ldots, \mathrm{K}-1\}$ the corresponding set. K is equal for all type 2 power plants. Furthermore, let the cycle and production campaign each plant starts with (i.e., during time step 0) be denoted as the *initial cycle* ($k = -1$). Throughout this paper K, T and W are also used to denote the end of the time horizon.

The following lower case indices will be used throughout this document to access single entities of the given type: $s \in \mathcal{S}$, $j \in \mathcal{J}$, $i \in \mathcal{I}$, $t \in \mathcal{T}$, $w \in \mathcal{W}$, $k \in \mathcal{K}$.

2.1.2 *Decision variables*

Decoupling dates: Each type 2 power plant regularly goes offline for refueling and maintenance. Let $ha_{i,k} \in \{-1\} \cup \mathcal{W}$ denote the week of decoupling of plant i in cycle k. Note that there are two special cases: for the initial cycle of each plant we define $ha_{i,-1} := 0$ and if a cycle is not scheduled, i.e., its decoupling date is postponed behind the time horizon, then $ha_{i,k}$ will be set to -1.

Refueling amounts: During each outage every type 2 power plant can be supplied with a certain amount of fuel. Let $r_{i,k} \in \mathbb{R}^{\geq 0}$ denote this amount of plant i in cycle k. We define the refueling amount of postponed cycles as 0.

Production levels: In a solution all power plants (i.e., both types) need to have an absolute real-valued production level assigned for each time step and scenario. This level of power plant j (resp. i) during time step t in scenario s is denoted by $p_{j,s,t}$ (resp. $p_{i,s,t}$) $\in \mathbb{R}^{\geq 0}$.

Note that a power plant's production levels may vary between demand scenarios of a dataset, while outage dates and refueling amounts are first-stage decisions, i.e., equal in all scenarios.

2.1.3 *Parameters*

Global parameters

$\mathrm{DEM}^{s,t}$	demand in time step t of scenario s
D	duration of each time step

Parameters of each type 1 power plant j

For each time step t of scenario s exist:

$\mathrm{PMIN}_j^{s,t}$	minimum production
$\mathrm{PMAX}_j^{s,t}$	maximum production
$\mathrm{C}_j^{s,t}$	production unit cost

Parameters of each type 2 power plant i

PMAX_i^t	maximum production in time step t
XI_i	initial fuel level, i.e., in time step 0
$\mathrm{C}_{i,\mathrm{T}}$	final refund per unit of residual fuel

The following parameters are provided for each cycle k:

$\mathrm{DA}_{i,k}$	outage duration in weeks (note: $\mathrm{DA}_{i,-1} := 0$)
$\mathrm{C}_{i,k}$	refueling unit cost
$\mathrm{RMIN}_{i,k}$	minimum refueling amount
$\mathrm{RMAX}_{i,k}$	maximum refueling amount
$\mathrm{MMAX}_{i,k}$	maximum modulation during cycle
$\mathrm{Q}_{i,k}$	refueling coefficient
$\mathrm{BO}_{i,k}$	imposed production threshold
$\mathrm{PB}_{i,k}$	imposed power profile
$\mathrm{AMAX}_{i,k}$	upper fuel bound before refueling
$\mathrm{SMAX}_{i,k}$	upper fuel bound after refueling

Additionally, the parameters $\mathrm{MMAX}_{i,-1}$, $\mathrm{BO}_{i,-1}$ and $\mathrm{PB}_{i,-1}$ are also provided.

2.1.4 Auxiliary Constructs

Definition 1 Given an arbitrary type 2 power plant i and a production cycle k, let $t^-_{i,k}$ denote the first time step of this cycle (which is also the first time step of the outage), $t^*_{i,k}$ denote the first time step of this cycle's production campaign and $t^+_{i,k}$ denote the first time step after the end of this cycle. For a cycle that is not scheduled all three variables equal T. Formally we define:

$$ha_{i,k} \neq -1 \Longrightarrow t^-_{i,k} = ha_{i,k} \cdot \mathrm{T} / \mathrm{W}$$

$$ha_{i,k} \neq -1 \Longrightarrow t^*_{i,k} = (ha_{i,k} + \mathrm{DA}_{i,k}) \cdot \mathrm{T} / \mathrm{W}$$

$$ha_{i,k+1} \neq -1 \Longrightarrow t^+_{i,k} = ha_{i,k+1} \cdot \mathrm{T} / \mathrm{W}$$

Note that $t^+_{i,k}$ generally corresponds to $t^-_{i,k+1}$ with exceptions for the initial cycle, where $t^-_{i,-1} = t^*_{i,-1} = 0$, and the last cycle, where $t^+_{i,\mathrm{K}-1} = \mathrm{T}$.

Definition 2 Let $x(i,s,t)$ denote the fuel level of plant i at the beginning of time step t in scenario s. For each plant and scenario, $x(i,s,t)$ depends on the fuel level of the previous time step and the production level and refueling performed during the previous time step. Note that $x(i,s,\mathrm{T})$ denotes the fuel level at the end of the time horizon.

2.2 Constraints

The problem contains a variety of constraints that deal for example with allowed production levels, fuel level restrictions, and maintenance requirements. A brief tabular overview of the constraints is given in Table 1. For further details on the constraints and exact definitions we refer to Brandt (2010).

2.3 Objective function

While satisfying all given constraints (CT 1–21) the sum of the expected production costs of all type 1 power plants over all scenarios and the total refueling cost of all type 2 power plants is to be minimized. Besides, there is a refund for residual fuel. The objective function is formalized as follows:

$$\underbrace{\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}} \mathrm{C}_{i,k} \cdot r_{i,k}}_{\text{type 2 refueling cost}} + \underbrace{\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}}\sum_{j\in\mathcal{J}} \mathrm{C}^{s,t}_j \cdot p_{j,s,t} \cdot \mathrm{D}}_{\text{type 1 production cost}}$$
$$- \underbrace{\frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{I}} \mathrm{C}_{i,\mathrm{T}} \cdot x(i,s,\mathrm{T})}_{\text{type 2 residual fuel refund}}$$

Note that the refueling amounts and refueling unit costs are constant in all scenarios. Contrary, the unit production costs of type 1 power plants and residual fuel amounts of type 2 power plants can vary between scenarios and have to be averaged.

Table 1 Overview of the model's constraint types

Constraint	Description
CT 1	demand equals production
CT 2	bound production of type 1 power plants
CT 3	no production during maintenance
CT 4/5	bound production of type 2 power plants
CT 6	power profile imposition
CT 7	bounds on refueling
CT 8	initial fuel level
CT 9	fuel consumption during production
CT 10	fuel variation during refueling
CT 11	bounds on fuel before and after refueling
CT 12	maximum modulation
CT 13A	earliest and latest date of outages
CT 13B	order of outages
CT 14–18	minimum spacing between outages
CT 19	limited resources
CT 20	maximum number of parallel outages
CT 21	maximum offline power capacity

3 Lower bounds

In this section we present two strategies for computing lower bounds for the overall solution cost of a given dataset. As a preparatory step, we introduce a lower bound for a type 2 power plant's unit production cost in Sect. 3.1. The first presented lower bound is the source of a simple greedy production planning heuristic presented in Sect. 6.

A more sophisticated lower bound, which also considers most of the fuel level and refueling constraints, is presented in Sect. 3.3. It employs a flow network to model production levels and refueling amounts.

3.1 Type 2 power plant unit production cost

The production of type 2 power plants is charged with the fuel reloaded during each outage and thus the exact cost of production per time step is unknown during production planning. In this section, we set up a simple lower bound for the *unit production cost* (UPC) of each type 2 power plant in a given dataset. This lower bound enables us to compare the UPC of all plants and—in a next step—assign production to the cheapest plants.

There are several hurdles to overcome while transforming reloading costs into production costs. First of all the initial fuel level of each plant is provided for free and there is a refund for residual fuel at the end of the time horizon. Furthermore, refueling unit costs vary with each plant and cycle (CT 7) and refueling is more complex than just adding fuel (CT 10).

For this bound we assume that type 2 power plants can produce in each time step, i.e., there are no scheduled outage periods. Furthermore, we relax the constraints for power profile imposition (CT 6), fuel level tracking (CT 7–11), maximum modulation over a cycle (CT 12), and all outage dependencies (CT 13–21).

In the remainder of this section, we set up a lower bound on the UPC c_i of a type 2 power plant i. Therefore, we first define the total cost C_i of a plant i, which is extracted from the objective function (cf. Sect. 2.3) as the sum of costs from all refuelings reduced by the refund for residual fuel at the end of the time horizon:

$$\mathrm{C}_i := \sum_{k\in\mathcal{K}} \mathrm{C}_{i,k} \cdot r_{i,k} - \mathrm{C}_{i,\mathrm{T}} \cdot x(i,s,\mathrm{T}) \tag{1}$$

We relate the total cost C_i to the amount of energy produced by plant i and define c_i as follows:

$$\mathrm{c}_i := \frac{\mathrm{C}_i}{\sum_{t\in\mathcal{T}} p_{i,s,t}\cdot \mathrm{D}} \tag{2}$$

Since all type 2 power plant parameters (cf. Sect. 2.1.3) are independent of the concrete scenario, we ignore s in the following without loss of generality. To transform refueling costs into costs of production, we make an aggregate analysis starting with some helpful definitions.

Given an arbitrary type 2 power plant i and a cycle k, we define the *refueling difference* $d(i,k)$ between the fuel level after refueling and the previous fuel level plus the refueling amount as

$$d(i,k) := x\left(i,s,t^-_{i,k}+1\right) - \left(x(i,s,t^-_{i,k}) + r_{i,k}\right).$$

This is a simplification of the reloading constraint CT 10, where we neglect fuel level thresholds $\mathrm{BO}_{i,*}$ and the refueling ratio $\mathrm{Q}_{i,k}$. By using $d(i,k)$, we can formulate a first lemma which holds for all scheduled cycles:

Lemma 1 *Given an arbitrary type* 2 *power plant i and a cycle k, the amount of fuel gained from all sources* (*initial fuel level, amount of refueling*) *equals the amount of fuel delivered to all consumers* (*production, residual fuel level*):

$$x\left(i,s,t^-_{i,k}\right) + r_{i,k} + d(i,k) = \sum_{t=t^-_{i,k}}^{t^+_{i,k}-1} p_{i,s,t}\cdot \mathrm{D} + x\left(i,s,t^+_{i,k}\right) \tag{3}$$

Proof We derive this lemma from the given constraints that influence the fuel level during a cycle. Starting with the fuel level variation during a single time step of a production campaign (CT 9), we aggregate over all production time steps of this cycle:

$$x\left(i,s,t^+_{i,k}\right) = x\left(i,s,t^*_{i,k}\right) - \sum_{t=t^*_{i,k}}^{t^+_{i,k}-1} p_{i,s,t}\cdot \mathrm{D}$$

According to CT 3, we can substitute the time step $t^*_{i,k}$ in the production sum by $t^-_{i,k}$ as there is no production in between these time steps. Similarly, corresponding to CT 10, the fuel level at $t^*_{i,k}$ equals the level at $t^-_{i,k}+1$. We get

$$x\left(i,s,t^+_{i,k}\right) = x\left(i,s,t^-_{i,k}+1\right) - \sum_{t=t^-_{i,k}}^{t^+_{i,k}-1} p_{i,s,t}\cdot \mathrm{D}$$

By replacing the fuel level after refueling $x(i,s,t^-_{i,k}+1)$ with the simplified reloading formula from the definition of the refueling difference $d(i,k)$ we gain Eq. (3). □

Equation (3) can be summarized over all cycles of a plant i to relate the fuel levels/variation to i's aggregated production. Since generally $t^+_{i,k} = t^-_{i,k+1}$, the initial and residual fuel levels of successive cycles can be canceled out. Furthermore, the initial fuel level can be set according to CT 8:

$$\mathrm{XI}_i + \sum_{k\in\mathcal{K}} r_{i,k} + \sum_{k\in\mathcal{K}} d(i,k) = \sum_{t\in\mathcal{T}} p_{i,s,t}\cdot \mathrm{D} + x(i,s,\mathrm{T}) \tag{4}$$

For each type 2 power plant i, let $\underline{\mathrm{C}}_i := \min_k(\mathrm{C}_{i,k})$ denote the minimum refueling cost. To get rid of varying refueling unit costs, we substitute the unit cost $\mathrm{C}_{i,k}$ of each cycle by $\underline{\mathrm{C}}_i$ in the following. This way, we get a lower bound for the total cost of plant i from Eq. (1):

$$\mathrm{C}_i \geq \underline{\mathrm{C}}_i \sum_{k\in\mathcal{K}} r_{i,k} - \mathrm{C}_{i,\mathrm{T}} \cdot x(i,s,\mathrm{T}) \tag{5}$$

By replacing the refueling amounts of Eq. (5) with Eq. (4) (solved for the sum of refuelings) and inserting this to Eq. (2), we estimate c_i:

$$\mathrm{c}_i \geq \underline{\mathrm{C}}_i + \frac{\underline{\mathrm{C}}_i\left(-\mathrm{XI}_i - \sum_{k\in\mathcal{K}} d(i,k)\right) - \left(\mathrm{C}_{i,\mathrm{T}} - \underline{\mathrm{C}}_i\right)\cdot x(i,s,\mathrm{T})}{\sum_{t\in\mathcal{T}} p_{i,s,t}\cdot \mathrm{D}} \tag{6}$$

We assume a non-negative numerator in Eq. (6), thereby ignoring all degenerate cases where a plant has negative total cost C_i. Given upper bounds for the sum of refueling differences $d(i,k)$, the sum of production levels $p_{i,s,t}$ and the amount of residual fuel $x(i,s,\mathrm{T})$, we have found a computable lower bound for c_i.

During any campaign, the fuel level never exceeds the maximum fuel level allowed after refueling. Hence, the residual fuel level is limited by

$$x(i,s,\mathrm{T}) \le \max\bigl(\mathrm{XI}_i, \max_k(\mathrm{SMAX}_{i,k})\bigr) \tag{7}$$

Assuming maximum production and exploiting CT 4, the sum of production is limited by

$$\sum_{t\in\mathcal{T}} p_{i,s,t} \le \sum_{t\in\mathcal{T}} \mathrm{PMAX}_i^t \tag{8}$$

Lemma 2 *The sum of refueling differences* $\sum_{k\in\mathcal{K}} d(i,k)$ *can be bounded from above by*

$$\max_{\kappa\in\mathcal{K}}\left(\mathrm{BO}_{i,\kappa} + \left(\sum_{k=0}^{\kappa-1}\frac{\mathrm{BO}_{i,k}}{\mathrm{Q}_{i,k+1}}\right) - \frac{\mathrm{Q}_{i,0}-1}{\mathrm{Q}_{i,0}}\mathrm{BO}_{i,-1}\right).$$

Proof Remember the refueling equation from CT 10:

$$x\bigl(i,s,t_{i,k}^{-}+1\bigr) = \frac{\mathrm{Q}_{i,k}-1}{\mathrm{Q}_{i,k}}\bigl(x\bigl(i,s,t_{i,k}^{-}\bigr) - \mathrm{BO}_{i,k-1}\bigr) + r_{i,k} + \mathrm{BO}_{i,k} \tag{9}$$

We insert this into the definition of the refueling difference $d(i,k)$ to approximate the amount of fuel gained or lost during refueling in the given outage:

$$d(i,k) = \mathrm{BO}_{i,k} - \frac{\mathrm{Q}_{i,k}-1}{\mathrm{Q}_{i,k}}\mathrm{BO}_{i,k-1} - \frac{1}{\mathrm{Q}_{i,k}}x\bigl(i,s,t_{i,k}^{-}\bigr)$$

Assuming that there is no fuel left before refueling, we get an upper bound for $d(i,k)$:

$$d(i,k) \le \mathrm{BO}_{i,k} - \frac{\mathrm{Q}_{i,k}-1}{\mathrm{Q}_{i,k}}\mathrm{BO}_{i,k-1}$$

Note that $d(i,k)$ might be negative. We now aggregate $d(i,k)$ for the first κ cycles utilizing a telescoping series:

$$\sum_{0\le k<\kappa} d(i,k) = \mathrm{BO}_{i,\kappa} + \left(\sum_{k=0}^{\kappa-1}\frac{\mathrm{BO}_{i,k}}{\mathrm{Q}_{i,k+1}}\right) - \frac{\mathrm{Q}_{i,0}-1}{\mathrm{Q}_{i,0}}\mathrm{BO}_{i,-1}$$

Since we do not consider any outages here, we cannot determine how many cycles of each plant will be exactly scheduled and as $d(i,k)$ might be negative, the sum over all cycles is probably not the maximum. In order to limit the sum of $d(i,k)$, we choose the maximum reached when only scheduling the first κ cycles. □

By inserting Eqs. (7) and (8) as well as Lemma 2 into Eq. (6), we get the sought lower bound of the unit production cost c_i of a type 2 power plant i.

Theorem 1 *For an arbitrary type* 2 *power plant i we can calculate a lower bound of*

$$\underline{\mathrm{C}}_i - \frac{\underline{\mathrm{C}}_i(\mathrm{XI}_i + \Delta_i) + (\mathrm{C}_{i,\mathrm{T}} - \underline{\mathrm{C}}_i)\cdot\max(\mathrm{XI}_i, \max_k(\mathrm{SMAX}_{i,k}))}{\sum_{t\in\mathcal{T}}\mathrm{PMAX}_i^t\cdot\mathrm{D}}$$

for the unit production cost c_i *where* $\underline{\mathrm{C}}_i$ *is the minimum refueling cost of i and* Δ_i *is defined as*

$$\Delta_i := \max_{\kappa\in\mathcal{K}}\left(\mathrm{BO}_{i,\kappa} + \left(\sum_{k=0}^{\kappa-1}\frac{\mathrm{BO}_{i,k}}{\mathrm{Q}_{i,k+1}}\right) - \frac{\mathrm{Q}_{i,0}-1}{\mathrm{Q}_{i,0}}\mathrm{BO}_{i,-1}\right).$$

3.2 An auction-based lower bound

The basic idea of this approach is that each plant offers its production capacity at a certain price that depends on the scenario and time step. The production levels are then assigned in a greedy way to the cheapest plants. We relax the imposed power profile constraint (CT 6), as well as all fuel level tracking (CT 7–12) and outage scheduling constraints (CT 13–21). As the unit production costs of the type 2 power plants we use the lower bound presented in Sect. 3.1.

Definition 3 The gross production amounts (as an interval [pmin, pmax]) and unit costs offered by all plants are defined in the following table:

	type 1 power plant o	type 2 power plant o
pmin_o	$\mathrm{PMIN}_o^{s,t}$	0
pmax_o	$\mathrm{PMAX}_o^{s,t}$	PMAX_o^t
cost_o	$\mathrm{C}_o^{s,t}$	c_i

Subtracting already assigned production levels gives the net amount of production that can be offered. Since we relaxed all fuel tracking constraints, the auction can be run for each time step of each scenario independently. Algorithm 1 presents such a single auction consisting of two steps. First, the minimum required production level of each plant is assigned. Afterwards, as much production as possible is assigned to the cheapest plants until the demand is covered.

Lemma 3 *Given an arbitrary scenario s and a time step t, Algorithm* 1 *finds a cheapest possible production assignment with respect to CT* 1, 2 *and* 4.

Proof The conformance with CT 1, 2 and 4 can easily be shown by using invariants on Algorithm 1. We will proof the remaining statement by contradiction, assuming that Algorithm 1 produced an assignment p, but there exists a cheaper assignment p'. In this case, there have to be two plants φ and

φ' whose production in p and p' differs and which have different unit costs as follows:

$$p_{\varphi,s,t} > p'_{\varphi,s,t} \tag{10}$$

$$p_{\varphi',s,t} < p'_{\varphi',s,t} \tag{11}$$

$$\cos t_{\varphi'} < \cos t_{\varphi} \tag{12}$$

Since both solutions have to meet the minimum offers, Algorithm 1 must have assigned different values in the maximum offer step. According to Eq. (12), the cheaper plant φ' is processed before plant φ. From Eq. (11), it follows that $p_{\varphi',s,t}$ is strictly less than the maximum amount offered by φ', which implies that the assigned production level was bounded by the demand and all more expensive plants (especially φ) do not get any production assigned. But this is a contradiction to Eq. (10) because $p_{\varphi,s,t}$ has to be strictly greater than 0. □

Hence, executing Algorithm 1 for all scenarios and time steps yields a cheapest production plan for the model without outages and fuel tracking. Algorithm 2 does this and returns the average total cost of production per scenario.

Algorithm 1: AUCTION (s, t)

Output: Total cost of assigned production for the given scenario and time step

```
begin
    demand ⟵ DEM^{s,t}
    foreach plant o do
        p_{o,s,t} ⟵ pmin_o
        demand ⟵ demand − pmin_o
    foreach plant o ascending by cost_o do
        produce ⟵ min(demand, pmax_o − p_{o,s,t})
        p_{o,s,t} ⟵ p_{o,s,t} + produce
        demand ⟵ demand − produce
    return
    ∑_{j∈J} cost_j · p_{j,s,t} · D + ∑_{i∈I} cost_i · p_{i,s,t} · D
end
```

Algorithm 2: AUCTIONS

Output: Average cost of assigned production across all scenarios

```
begin
    cost ⟵ 0
    foreach s ∈ S do
        foreach t ∈ T do
            cost ⟵ cost + AUCTION(s, t)
    return cost/|S|
end
```

From Lemma 3 we know that Algorithm 2 aggregates lower bound production assignments for each time step and thus finds a global cheapest production assignment for the offers of Definition 3. By formalizing the result of Algorithm 2 and transforming it into the original objective function one can show the following lemma.

Lemma 4 *Given production intervals from Definition 3, then Algorithm 2 returns a lower bound of the objective function's value for a given dataset.*

Since we neither care for outages nor for fuel levels, this bound is not as close to real solutions as one might wish and we will present a more sophisticated bound in the next section.

3.3 A flow-based lower bound

In this section, we present a flow network that models a power assignment and considers outage restrictions (CT 13) as well as fuel consumption in an approximate fashion (CT 7–12) to deduce a tighter lower bound of the objective function for a given dataset. Still, all outage dependency constraints (CT 14–21) are relaxed. The network operates on a single scenario. Hence, the overall lower bound of an instance corresponds to the average cost realized by the flow network in all scenarios. This cost can be computed by any standard black-box network-flow solver. In this text, we only give an intuition and present the main ideas. A formal definition of this approach as well as a sketch of the proof of its lower bound property can be found in Brandt (2010).

The previously shown auction-based bound has two major drawbacks, which do not appear in the flow-based bound. Firstly, the auction strongly underestimates the unit production cost of type 2 power plants. Although this is not a big problem when distributing demand (we assume all type 2 power plants are nearly equally underestimated), it clearly distorts any lower bound. Secondly, the auction overestimates type 2 power plant production capacity because neither outages are considered nor the amount of fuel consumed over time is restricted.

3.3.1 Intuition

The network consists of two logical parts (see Fig. 4). The left part handles the distribution of the demand level of each time step to the power plants. The lower right segments model the internals of type 2 power plants, more precisely: the sources of fuel, which is consumed for production. The network's commodity is fuel, i.e., production levels and demands are multiplied by their duration (see CT 9). Arcs are annotated with minimum and maximum capacities, as well as the cost per unit of flow.

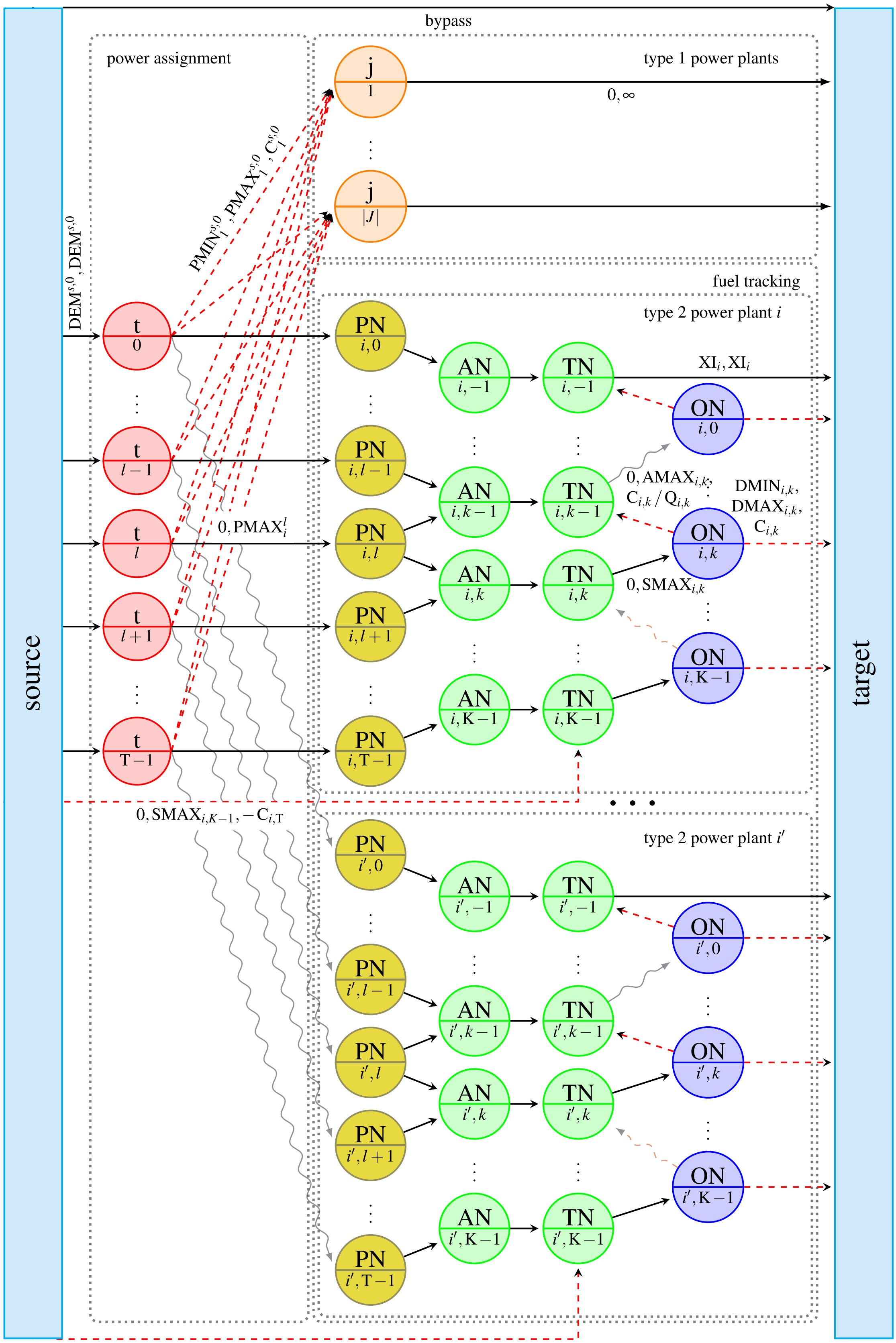

Fig. 4 Sketch of the flow network: the *left part* performs the power assignment, while the *lower right parts* assign the fuel consumed by type 2 power plants to a potential cycle, where it is charged. Only the *dashed arcs* incur cost

For each *time step node* (t), there is a certain amount of fuel required to cover the demand level. This amount is distributed among all power plants' *production nodes* (type 1: j, type 2: PN). Since the costs of type 1 power plants are charged with the produced amounts of energy, the respective $t \rightarrow j$ arc captures this costs.

To bill the production of a type 2 power plant, the amount of fuel which is consumed during each time step has to be mapped to its source (either the initial fuel level or a specific outage). Therefore, an arc is added from the PN of each time step to the *aggregation nodes* (AN) of all cycles that might be active during this time step. To restrict the overall fuel consumption during a cycle, an arc with limited capacity from AN to its corresponding *transfer node* (TN) is added. From a cycle's point of view, there are two sources of fuel incident to the cycle's *outage node* (ON): the residual fuel of the previous cycle (ON→TN) and the refueling done during the cycle's outage (ON→target). To model the residual fuel levels in the network, an arc with negative cost (the refund) from the source directly into the TN of the last cycle of each type 2 power plant is introduced. Furthermore, a bypass arc of zero cost is added for technical reasons. Its capacity equals the sum of capacities from all residual arcs. This way, a minimum cost flow problem has to be solved for the flow amount: $\sum_t \mathrm{DEM}^{s,t} + \max(bypass)$. We relax CT 6 and all dependencies between outages of different plants (CT 14–21) in the presented lower bound.

In our flow network the refueling amount cannot be handled directly (see CT 10). Therefore, we define the simpler *reloading difference* $\delta(i,s,k)$, which can be embedded in the network.

$$\delta(i,s,k) := x\left(i,s,t_{i,k}^{-}+1\right) - x\left(i,s,t_{i,k}^{-}\right)$$

From constraints CT 7, 10 and 11 one can derive a lower bound $\mathrm{DMIN}_{i,k}$ and upper bound $\mathrm{DMAX}_{i,k}$ for $\delta(i,s,k)$:

$$\begin{aligned}
&[\mathrm{DMIN}_{i,k}, \mathrm{DMAX}_{i,k}]\\
&\quad := \left[\mathrm{RMIN}_{i,k} - \frac{\mathrm{AMAX}_{i,k}}{\mathrm{Q}_{i,k}}, \mathrm{RMAX}_{i,k}\right]\\
&\qquad + \mathrm{BO}_{i,k} - \frac{\mathrm{Q}_{i,k}-1}{\mathrm{Q}_{i,k}}\mathrm{BO}_{i,k-1}
\end{aligned}$$

Nevertheless, the real cost of refueling is required for the lower bound and has to be reconstructed. This is done by splitting the cost into three components: one depending on the fuel level before refueling, another depending on the refueling difference, and a third fixed offset:

$$\begin{aligned}
\mathrm{C}_{i,k} \cdot r_{i,k} = {} & \underbrace{\frac{\mathrm{C}_{i,k}}{\mathrm{Q}_{i,k}} \cdot x\left(i,s,t_{i,k}^{-}\right)}_{\text{initial fuel level}} + \underbrace{\mathrm{C}_{i,k} \cdot \delta(i,s,k)}_{\text{reloading difference}}\\
& + \underbrace{\mathrm{C}_{i,k} \cdot \left(\frac{\mathrm{Q}_{i,k}-1}{\mathrm{Q}_{i,k}}\mathrm{BO}_{i,k-1} - \mathrm{BO}_{i,k}\right)}_{\text{global cost offset}}
\end{aligned} \tag{13}$$

Besides the refueling issues, the compliance with CT 3 and CT 12 is not fully enforced in our flow network. As the outage dates are not determined yet, it is unknown during which time steps a power plant is shut down (CT 3). But since the network allows that the consumed fuel is gathered from any cycles that might be active during this step, this constraint is modeled indirectly in a weaker form. Maximum modulation (CT 12) effectively creates a minimum production amount during a production campaign. Unfortunately, since the production campaign's duration is not fixed, this amount can only be bound by the minimum duration and is therefore not a strong restriction.

3.3.2 *Handling non-scheduled cycles*

At building time of a concrete flow network instance, the number of scheduled cycles of each power plant in a final (and maybe optimal) solution is unknown. But, since parts of the network will be set up per cycle, we have to take care that the cost induced by modeled but postponed cycles does not destroy the lower bound property. We guarantee this by relaxing the problematic network properties, i.e., arc capacities and cost.

First, we identify three categories of cycles: mandatory cycles which have to be scheduled in any solution, impossible cycles which cannot be scheduled in any solution[1] and optional cycles.

In the following let $\mathrm{K}_i^{\min}$ denote the number of the last mandatory cycle and $\mathrm{K}_i^{\max}$ denote the number of the last optional cycle of a type 2 power plant i. When building a concrete network instance, only the first $\mathrm{K}_i^{\max}$ cycles of each power plant are added. For each optional cycle the minimum capacity of the AN → TN arc is set to 0. Furthermore, the cost components identified in Eq. (13) have to be adapted to behave gracefully if they belong to an unscheduled cycle.

The cost component of the initial fuel level factor models the opportunity cost of lost (or not gained free) fuel during an outage. As there is no such loss for unscheduled cycles, the cost of the ON→TN arc for an optional cycle is set to 0. Furthermore, the final fuel level of a type 2 power plant might have to pass several optional cycles in our flow network. Hence, the maximum fuel level before and after refueling (see CT 11) of each optional cycle has to be modified

[1] Such cycles can be identified by a constraint reasoning, see Sect. 5.

for the flow network:

$$\mathrm{AMAX}'_{i,k} := \begin{cases} \mathrm{XI}_i & \text{if } k=0 \\ \max(\mathrm{AMAX}_{i,k}, \mathrm{SMAX}_{i,k-1}) & \text{if } k=\mathrm{K}_i^{\min}+1 \\ \max(\mathrm{AMAX}_{i,k}, \mathrm{SMAX}'_{i,k-1}) & \text{else} \end{cases}$$

$$\mathrm{SMAX}'_{i,k} := \max\bigl(\mathrm{AMAX}'_{i,k}, \mathrm{SMAX}_{i,k}\bigr)$$

Furthermore, the refueling amount CT 7 has to be relaxed ($0 \le r_{i,k} \le \mathrm{RMAX}_{i,k}$) and the range of the reloading difference $\delta(i,s,k)$ has to be adapted for optional cycles:

$$\bigl[\mathrm{DMIN}'_{i,k}, \mathrm{DMAX}'_{i,k}\bigr] := \bigl[\min(0, \mathrm{DMIN}_{i,k}), \mathrm{DMAX}_{i,k}\bigr]$$

The global cost offset also has to be adapted to always return a lower bound. Therefore, the minimum subsum when scheduling any possible number of optional cycles is chosen as the effective global cost offset of power plant i:

$$\min_{\mathrm{K}_i^{\min} \le \kappa \le \mathrm{K}_i^{\max}} \left(\sum_{k=0}^{\kappa} \mathrm{C}_{i,k} \cdot \left(\frac{\mathrm{Q}_{i,k}-1}{\mathrm{Q}_{i,k}} \mathrm{BO}_{i,k-1} - \mathrm{BO}_{i,k} \right) \right)$$

In Sect. 7.3 we show how the presented lower bounds compare to the best known solutions that have been found for the test instances.

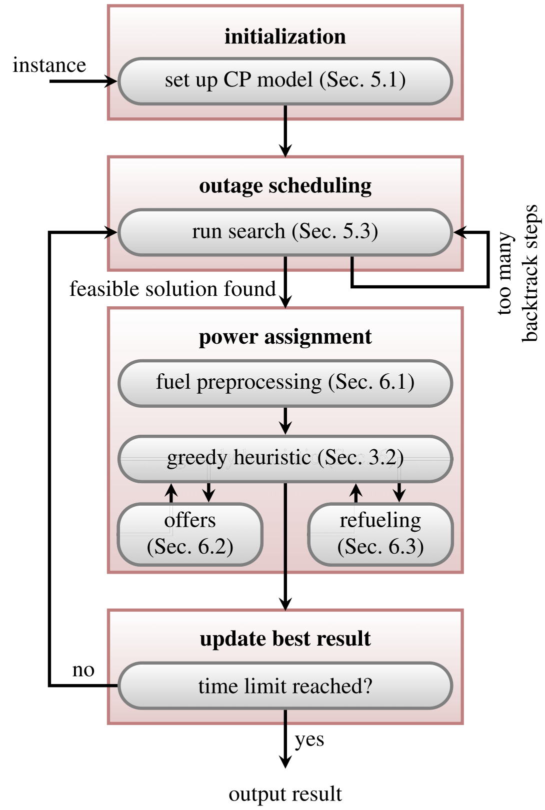

Fig. 5 Illustration of the overall solution process

4 Overall solution process

Since there is probably no way to solve even medium-sized instances exactly in a reasonable time, we develop several heuristics. Therefore, we decompose the problem into an outage scheduling and a power assignment phase. Outage dates $ha_{i,k}$ are fixed in the first phase, while power levels $p_{j,s,t}/p_{i,s,t}$ and refuelings $r_{i,k}$ are calculated in the second phase for a given outage schedule.

Section 5 deals with the outage scheduling phase. Due to the large number of constraint types (see Sect. 2.2) and their dependencies we use a CP formulation to find a feasible schedule. In Sect. 6, we present a valid production assignment heuristic for a given set of outages. Therefore, we utilize a demand auction derived from the lower bound presented in Sect. 3.2.

The overall solution process is of randomized, iterative nature. In a preparatory step, the CP-model of the outage scheduling part is initialized. An iterative step starts with searching for a solution of the outage scheduling part. If no feasible solution is found after a given number of backtracking steps, the whole iterative step is terminated and the next iteration starts. As soon as a feasible solution has been found, the power assignment phase is executed. Afterwards, the found solution is compared with the best solution found so far and the next step starts. The whole process ends when a given timelimit is reached.

Note that the branchers used in the outage scheduling part are partly randomized and hence new solutions are found in every step. An overview of the process is given in Fig. 5.

In Table 2 we give an overview at which point of the solution process we take care of each constraint type of the problem. We mark parts of our heuristics with an orange square if the consideration of the given constraint is necessary, i.e., not checking this property in the particular step might result in invalid solutions. A green dot stands for a sufficient mean to achieve this constraint, i.e., after performing this step the remaining solution space cannot fail this constraint.

5 Outage scheduling

The first step of the presented solution is to search for feasible and promising outage schedules, i.e., a fixed week of decoupling for each outage such that the scheduling related constraints CT 13–21 are fulfilled and the expected production cost is low. As this subset of constraints does not guarantee the feasibility of the remaining production assignment and refueling problem, the constraints CT 1–12 are considered in an approximate fashion.

Table 2 Illustration at which step each constraint type is considered. In the outage scheduling part, we restrict the variables' *domains* (DO), add the given *constraints* (CT) to our CP model or add our own *additional* (AC) constraints. The production assignment part consists of a *preprocessing* (PP) step, the calculation of *minimum* (MN) and *maximum* (MX) offers for the *auctions* (AU) of each time step and the fixation of *refueling* (RF) amounts. Orange squares symbolize a step where ignoring the constraint might result in invalid solutions. Green dots mark a step whose entire resulting search space fulfills the constraint, i.e., we do not need to care anymore about it

CT	Outage Scheduling			Production Assignment				
	DO	CT	AC	PP	MN	MX	AU	RF
1	○	○	■	○	○	○	●	○
2	○	○	○	○	●	●	○	○
3	○	○	■	■	●	●	○	○
4/5	○	○	■	■	●	●	○	○
6	○	○	■	■	●	●	○	○
7	■	○	○	■	○	○	○	●
8	■	○	○	○	○	○	●	○
9	○	○	■	■	○	○	●	○
10	○	■	○	○	○	○	○	●
11	■	○	○	■	●	○	○	○
12	○	○	■	○	●	○	○	○
13A	●	○	○	○	○	○	○	○
13B	○	●	○	○	○	○	○	○
14-18	○	●	○	○	○	○	○	○
19	○	●	○	○	○	○	○	○
20	○	●	○	○	○	○	○	○
21	○	●	○	○	○	○	○	○

Table 3 Finite domain integer variables instantiated for each cycle of a type 2 power plant i. The *Initial* column comprises the predefined values of this variable for the initial cycle $k = -1$, while the *Min* and *Max* columns represent the domain limits for all successive cycles

Var.	$k = -1$	$k > -1$		Description
	Initial	Min	Max	
$sch_{i,k}$	1	0	1	Is campaign scheduled?
$dec_{i,k}$	0	$\mathrm{TO}_{i,k}$	$\mathrm{TA}_{i,k}$	Date of decoupling
$ref_{i,k}$	0	$\mathrm{RMIN}_{i,k}$	$\mathrm{RMAX}_{i,k}$	Amount of fuel provided
$pre_{i,k}$	0	0	$\mathrm{AMAX}_{i,k}$	Fuel before refueling
$post_{i,k}$	XI_i	0	$\mathrm{SMAX}_{i,k}$	Fuel after refueling

We use a CP formulation to find a feasible outage schedule as well as reasonable lower and upper bounds for the refueling amounts. To keep the number of variables at a minimum, only the fuel levels before and after each outage are part of the model. Furthermore, the initial cycle $k = -1$ of each type 2 power plant is included in the model and all its decision variables are assigned to suitable values. The model contains five variables per cycle: a boolean variable to decide if the outage is scheduled or not ($sch_{i,k}$) as well as integer variables for the decoupling date ($dec_{i,k}$), refueling amount ($ref_{i,k}$), and the fuel levels before and after each outage ($pre_{i,k}$ and $post_{i,k}$). The domains of the variables and the values for the initial cycle are presented in Table 3.

The following sections sketch our CP formulation in a bottom-up fashion. First the relations within a cycle and between successive cycles of a type 2 power plant are introduced, then three additional constraints are presented that approximate the fuel consumption to create effective spacing constraints between successive outages. Finally, the chosen branching strategy is explained.

5.1 Basic modeling

The key variable of each cycle is the decoupling date $dec_{i,k}$. If there is no CT 13 for a cycle, the CP model does not bound $dec_{i,k}$ to easily handle unscheduled (or "postponed") cycles. The decision to schedule a cycle is reduced to the decision to schedule the outage's decoupling date before the time horizon. Accordingly, if for any reason an outage cannot be scheduled before the time horizon, then the CP does not schedule it:

$$sch_{i,k} \iff dec_{i,k} < \mathrm{W} \tag{14}$$

Although no production levels are planned in the CP, it is crucial to track fuel levels in order to keep the remaining production assignment problem feasible. The change of the fuel level in an outage can easily be taken from CT 10:

$$post_{i,k} = \frac{\mathrm{Q}_{i,k} - 1}{\mathrm{Q}_{i,k}} (pre_{i,k} - \mathrm{BO}_{i,k-1}) + ref_{i,k} + \mathrm{BO}_{i,k} \tag{15}$$

So far, the formulated constraints model a single cycle. According to CT 13, the order of cycles is fixed:

$$dec_{i,k+1} \geq dec_{i,k} + \mathrm{DA}_{i,k} \tag{16}$$

To complete the model, all instances of CT 14 to CT 21 have to be added. This can be done with little effort by using scheduling or linear constraints.

5.2 Additional constraints

In the following, three additional constraints are formulated that relate the fuel levels of successive cycles to each other and set up lower and upper bounds for the duration of production campaigns. Although these constraints do not for-

Table 4 Forward propagation parameters to calculate the *upper limit* and *lower limit* of $pre_{i,k+1}$

	Upper limit	Lower limit
start	$\min(dec_{i,k}) + \mathrm{DA}_{i,k}$	$\max(dec_{i,k}) + \mathrm{DA}_{i,k}$
end	$\max(dec_{i,k+1})$	$\min(dec_{i,k+1})$
fuel	$\min(post_{i,k})$	$\max(post_{i,k})$
modulation	0	$\mathrm{MMAX}_{i,k}$

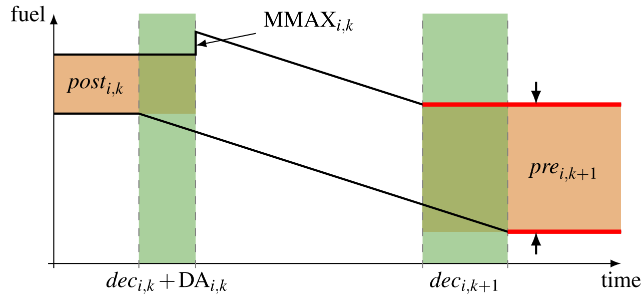

Fig. 6 Fuel level propagation: by assuming maximum production (*lower line*) and minimum production (*upper line*) starting with $post_{i,k}$ fuel the domain of the final fuel level $pre_{i,k+1}$ can be bound

mally guarantee the feasibility of the remaining problem,[2] our experience shows that this is the case in almost all situations.

5.2.1 *Fuel level propagation*

A core requirement to track a type 2 power plant's fuel level is to approximate the minimum and maximum fuel consumption during a production campaign. The consumption depends on the coupling and decoupling dates, the respective maximum production levels, the maximum allowed modulation, and the present fuel level.

Relating all the fuel levels of a type 2 power plant to each other requires the ability to reason from $post_{i,k}$ (fuel level after refueling) to $pre_{i,k+1}$ (fuel level before refueling of the next cycle) and vice versa. Therefore, we consider a minimum and a maximum production scenario. Maximum production is done with the given maximum production level PMAX_i^t of each time step. Similarly, minimum production is modeled by adding the allowed modulation $\mathrm{MMAX}_{i,k}$ to the fuel level and also assuming maximum production.

First, we limit the final fuel level of the production campaign $pre_{i,k+1}$ by performing a maximum and minimum production starting with $post_{i,k}$. The maximum possible final fuel level is reached by taking the upper domain limit of the opening fuel level, choosing the shortest duration of the production campaign and fully exhausting modulation. Analogously, the minimum possible final fuel level is reached by taking the lower domain limit of the opening fuel level, choosing the longest duration and no modulation. Table 4 gives an overview of the used parameters, while Fig. 6 illustrates the process.

To receive a real benefit from the fuel level propagation, its opposite direction also has to be implemented, i.e., reasoning from $pre_{i,k+1}$ to $post_{i,k}$. Calculating the maximum fuel level of a previous time step can be done by using CT 9 as long as production is not imposed and by searching for the right segment of the imposed power profile function $\mathrm{PB}_{i,k}$ otherwise. Just as in the forward calculation, the maximum allowed fuel level at the beginning of the production campaign is reached by taking the upper domain limit of the final fuel level, choosing the longest possible duration of the production campaign and using no modulation. Backward propagation of the minimum fuel level is of little use and therefore skipped.

5.2.2 *Outage spacing*

Given the minimum fuel level before a production campaign $\min(post_{i,k})$ as well as the maximum after it $\max(pre_{i,k+1})$ and the plant's maximum production levels for all time steps in between, it is possible to deduce the minimum period of time the plant needs to run to consume this fuel difference. Scheduling the neighboring outages in a shorter distance still would result in a feasible outage schedule but give an infeasible production assignment problem.

Consequently, an outage spacing constraint has to be added to the model. In forward direction, the date of decoupling $dec_{i,k+1}$ can be shifted until the fuel level falls below $\max(pre_{i,k+1})$ starting maximum production at the earliest coupling date with a minimum fuel level (see Fig. 7). Analogously, the latest possible decoupling date of a cycle $dec_{i,k}$ can be determined by a reasoning going backward in time and starting at the latest possible decoupling date of the next cycle $\max(dec_{i,k+1})$ with $\max(pre_{i,k+1})$ fuel and maximum production up to the week where the fuel level exceeds the lower bound of $post_{i,k}$.

5.2.3 *Online propagator*

Besides a minimum spacing between outages it is useful to set up a maximum spacing. If the scheduled production campaigns are too long, then the plant will run out of fuel and demand has to be covered by more expensive plants. Hence, we restrict the latest date of decoupling to the week when the plant will run out of fuel. The date which achieved the best results in our preliminary experiments is determined by performing maximum production, starting with $\max(post_{i,k})$

[2] In some degenerate cases the principle of maximizing the available type 2 production capacity might turn out wrong. If there is little demand over a longer period of time, then type 2 power plants might not be able produce enough to comply with CT 11 and CT 12. But, this was not the case in the provided datasets.

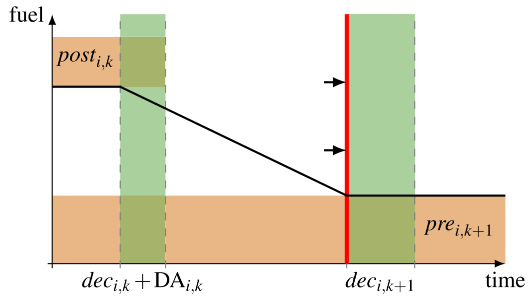

Fig. 7 Outage spacing: starting maximum production at the earliest possible date with the minimum fuel level $\min(post_{i,k})$, the earliest date of the next outage can be delayed to the time step, where the current fuel level falls below the maximum allowed fuel level $\max(pre_{i,k+1})$ before the next outage

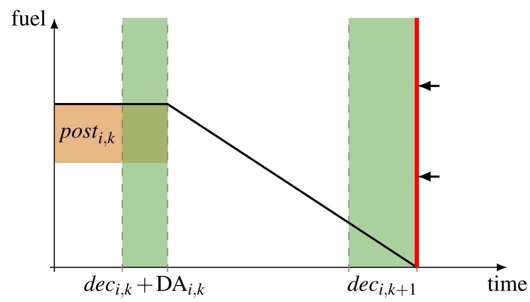

Fig. 8 Online propagator: the latest possible decoupling date of the upcoming outage is limited to the week where the plant runs definitely out of fuel

fuel at the latest possible coupling date (see Fig. 8). In case a plant definitely runs out of fuel before the next decoupling date, $dec_{i,k+1}$ is set to its domain minimum.

Note that by adding this constraint the CP model might become infeasible. Although this problem did not occur in any experiment, the model should be rerun without this constraint if no solution is found in a reasonable amount of time.

5.3 Branching strategy

As some of the constraints are not modeled precisely in the CP, we do not solve a classical branch-and-bound constraint optimization problem but rather use the constraint solver to produce a small number of feasible and promising outage schedules for further inspection. While *constraint propagation* ensures the feasibility of the solution, the *branching strategy* has to be chosen carefully to select cheap schedules first. We identify cheap outage dates by their additional cost, i.e., the cost of shifting production capacity to a more expensive plant. This additional cost can be approximated by the marginal cost, i.e., the unit cost of the most expensive power plant still scheduled for production, realized by the lower bound presented in Sect. 3.2.

In a preprocessing step, we run the lower bound calculation once before starting the search. This provides for each scenario and time step the marginal cost assuming that all plants are available. By using these marginal costs, an approximation of the additional cost when scheduling an outage during a certain week can be quantified (see Fig. 9). Aggregating these additional costs over all affected time steps of a specific decoupling date gives the additional cost of the whole outage.

After preprocessing a depth first search is started in the search tree. As it is common in CP setups, the chosen branching strategy consists of two decisions: variable selection and value selection. At each branching step our strategy selects an unfixed decoupling variable and assigns it to a certain week in the first branch, while excluding the chosen week from the second branch's search space. In constraint satisfaction setups the quality of a branching is primarily measured by how early the CP solver detects that no feasible solution exists in the current subtree. Generally, it is a good idea to assign closely related decision variables within a few branching steps to detect infeasible situations as early as possible. Therefore, our variable selection strategy will consider only the first (i.e., minimum k) unassigned decoupling date $dec_{i,k}$ of each type 2 power plant at each branching step. Among those the variable with the minimal domain minimum is chosen.

In the preprocessing step the additional costs for each possible decoupling date of an outage were determined. To prefer cheaper decoupling dates our value selection strategy chooses decoupling dates that incur low additional costs with a high probability. Formally, for a given outage the mth best out of n still available decoupling dates is chosen with probability $\mathbf{P}[m] := 0.5^m$. The most expensive date is chosen with the residual probability $\mathbf{P}[n] := 1 - \sum_{m=1}^{n-1} \mathbf{P}[m] = 0.5^{n-1}$.

Our proposed search strategy stops at the first found feasible solution. If no solution is found after $2 \cdot |\mathcal{I}| \cdot \mathrm{K}$ backtrack steps the search is restarted from the root node. This is necessary as our randomized branching strategy might get stucked in infeasible regions of the search space.

6 Production assignment

Now, we present a power assignment heuristic derived from the auction-based lower bound presented in Sect. 3.2. Power assignments are performed for a given outage schedule (i.e., all $ha_{i,k}$ are fixed) and assign the resulting decision variables: production levels $p_{j,s,t}/p_{i,s,t}$ and refueling amounts $r_{i,k}$. We use the same auction method and type 1 power plant offers as presented in Sect. 3 but augment the offers of type 2 power plants by their current fuel level and consumed modulation, i.e., the difference between the maximum and realized production throughout the current campaign.

The auction is run in ascending order of the time steps and for all scenarios in parallel. This way, it is easy to adjust

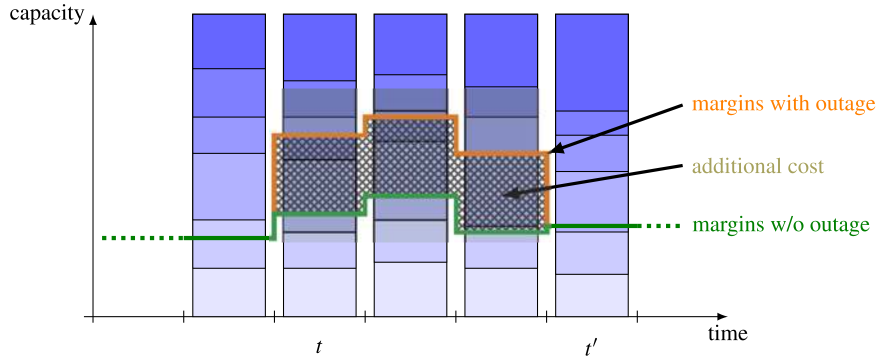

Fig. 9 Illustration of the additional costs that arise when scheduling an outage. The plants production capacities are represented by the *boxes* ordered by ascending cost (*light* = cheap, *dark* = expensive). Capacities below the lower marginal costs line are utilized for production. When putting a plant on outage between the time steps t and t', its production capacity has to be transferred to the next more expensive plants (*checkered region*), resulting in new marginal cost (*upper line*)

the fuel levels from the previous time step and fix an outage's refueling amount consistently at the second time step of the outage.

Although cheap production is the ultimate goal, sometimes a type 2 power plant has to be utilized at any cost to comply with the given constraints. Fortunately, the provided datasets suggest that type 2 power plant production is generally cheaper than type 1 power plant production. But this is nowhere stated explicitly and we do not take it for granted. Nevertheless, we assume in this section that it is desirable for each type 2 power plant to produce as much as possible and adjust our refueling strategy accordingly.

6.1 Type 2 power plant preprocessing

During each production campaign a certain amount of fuel has to be consumed to reach the fuel level limits that apply before or after the upcoming refueling (see CT 11). In Sect. 5.2.1 we introduced a constraint that assured this property between successive outages. In this preprocessing step we define this limit for each time step, relying on the fact that the assignment problem is feasible.

We start at the last time step and move backward in time. Since all outage dates are known at this state of the solution process, a maximum allowed fuel level of each time step can be assigned by assuming maximum production in each time step and minimum refueling during each outage. See Algorithm 3 for a pseudo-code illustration of this preprocessing step.

Note that these maximum fuel levels impose a minimum production if the current fuel level comes close to its maximum.

6.2 Type 2 power plant offers

Besides the maximum allowed fuel levels, we have to take care of several other constraints when assembling the offered production amounts of a type 2 power plant.

Algorithm 3: PREPROCESS (i)

Output: Vector of upper bounds of fuel levels after production for each time step

begin

`// init last element of the result vector`

$fuelmax_{\mathrm{T}} \longleftarrow \infty$

foreach scheduled $k \in \mathcal{K}$ in descending order **do**

`// back propagation of the max fuel`

for $t = t^{+}_{i,k} - 1$ to $t^{*}_{i,k}$ **do**

$fuelmax_t \longleftarrow$ PREVIOUS$(i, t, fuelmax_{t+1})$

`// fuel level limit after refueling`

$fuelmax_{t^{-}_{i,k}+1} \longleftarrow \min(\max(post_{i,k}), fuelmax_{t^{*}_{i,k}})$

`// reconstruct fuel before min refueling`

$fuelmax_{t^{-}_{i,k}} \longleftarrow \frac{\mathrm{Q}_{i,k}}{\mathrm{Q}_{i,k}-1}(fuelmax_{t^{-}_{i,k}+1} - \min(ref_{i,k}) - \mathrm{BO}_{i,k}) + \mathrm{BO}_{i,k-1}$

`// fuel level limit before refueling`

$fuelmax_{t^{-}_{i,k}} \longleftarrow \min(\max(pre_{i,k}), fuelmax_{t^{-}_{i,k}})$

return *fuelmax*

end

Firstly, offers have to comply with imposed power profiles if the fuel level threshold $\mathrm{BO}_{i,k}$ is under-run. This is achieved by setting an imposed power level as the minimum offer. If the fuel levels of each time step are tracked, this production level can be easily calculated. Algorithm 4 gives a

Algorithm 4: MINOFFER (i, k, s, t)

```
Output: Minimum required amount of production
begin
    amount ⟵ 0
    // Is this a production time step? [CT3]
    if t*_{i,k} ≤ t then
        // Imposed power profile active? [CT6]
        if fuellevel_{i,t} < BO_{i,k} then
            amount ⟵ PB_{i,k}(fuellevel_{i,t}) · PMAX^t_i
        else
            // maximum fuel level [CT11]
            if fuellevel_{i,t} > fuelmax_{i,t} then
                amount ⟵ (fuellevel_{i,t} − fuelmax_{i,t}) / D
            consumed_modulation = Σ_{τ=t*_{i,k}}^{t} (PMAX^τ_i − p(i, s, τ)) · D
            // maximum modulation [CT12]
            if consumed_modulation > MMAX_{i,k} then
                amount ⟵ max(amount, (consumed_modulation − MMAX_{i,k}) / D)
    return amount
end
```

Algorithm 5: MAXOFFER (i, k, s, t)

```
Output: Maximum available amount of production
begin
    amount ⟵ 0
    // Is this a production time step? [CT3]
    if t*_{i,k} ≤ t then
        // production not imposed? [CT6]
        if fuellevel_{i,t} ≥ BO_{i,k} then
            // choose minimal upper bound from [CT4/5] and [CT9]
            amount ⟵ min(fuellevel_{i,t} / D, PMAX^t_i − p(i, s, t))
    return amount
end
```

pseudo-code notation of the process of calculating minimum offers and considers the constraints CT 6, 11 and 12.

CT 6: If the production is imposed, the exact level is calculated and set as the minimum offer.

CT 11: If the current fuel level exceeds the maximum allowed fuel level of this time step, it is not possible to reach the maximum fuel level before or after refueling of the next cycle. Therefore, the minimum offer is set such that the fuel level is reduced to its maximum allowed value.

CT 12: If the consumed modulation of the current cycle (including the time step of the auction) would exceed the allowed maximum modulation, the minimum offer is set such that the modulation equals $\mathrm{MMAX}_{i,k}$.

For time steps where a type 2 power plant is offline or its production is imposed, the maximum production was already set by the minimum offer. During all other time steps, the only constraints limiting maximum production are CT 4, stating the maximum production of the plant, and CT 9, limiting the fuel level to non-negative values. Algorithm 5 assembles the maximum offered production amount.

6.3 Refueling amounts

Finally, we have to decide how much fuel to reload during each outage. If the auction is run for a time step in which a refueling for plant i has to be performed, the current fuel levels of i in all scenarios are known because the previous auctions have already been executed. Since the maximum allowed fuel levels of all scenarios were calculated in the preprocessing step, a possible refueling interval for each scenario can be determined. The intersection of these intervals is the globally possible refueling interval of the considered outage.

Since the realized fuel level in each scenario is less or equal to its upper bound found in the preprocessing step we can conclude the following: Given an arbitrary type 2 power plant i and cycle k, the intersection of the scenarios' refueling intervals $r_{i,k}$ is not empty.

We still have to decide which value to take from the refueling interval of the outage. A higher amount of refueling probably results in higher utilization of the plant. This seems desirable, as the provided instances suggest that type 2 power plant production is generally cheaper than production by type 1 power plants. On the other hand, a larger refueling results in more residual fuel and thus a worse refueling difference, i.e., more lost fuel during refueling (see CT 10). Algorithm 6 demonstrates the calculation of the concrete value.

6.4 Tuning refueling amounts

The basic refueling strategy presented in Sect. 6.3 is fast to compute, but not desirable for several reasons. Firstly, if the plant is operating at its maximum allowed fuel level, its production level is imposed in order to stay below the maximum

Algorithm 6: REFUELING (i, k, t)

Output: The reloading amount to perform
begin
  $a \longleftarrow \mathrm{RMAX}_{i,k}$
  foreach scenario s **do**
    $a \longleftarrow \min\left(a, fuelmax_{i,t^-_{i,k}+1} - \mathrm{BO}_{i,k} - \frac{\mathrm{Q}-1}{\mathrm{Q}}(fuellevel_{s,t^-_{i,k}} - \mathrm{BO}_{i,k-1})\right)$
  return a
end

allowed fuel level at the end of the production campaign (cf. Sect. 6.2). So, the plant might have to produce although it is not among the cheapest plants. Secondly, reloading more fuel is expensive and as more fuel is reloaded, there is probably a higher loss of fuel at the end of the production campaign. On the other hand, reloading too little amounts of fuel results in less production, and hence also higher cost.

To determine a good refueling amount, we set up a model to forecast the profit induced by a certain refueling amount. We estimate this profit from the cost of refueling, the production amounts, and an approximation of the lost fuel at the end of the production campaign. The cost of refueling and the cost of lost fuel can be easily determined. The key idea to transfer production amounts into revenue is to run the production assignment phase twice. In the first run, we store the marginal cost realized for each time step. These costs are used in a second run as the price of each produced unit of energy.

At the instant when a concrete refueling amount is chosen, the decoupling dates $dec^*_{i,k}$ have already been fixed and the fuel level before refueling of the plant is known for each scenario. Algorithm 7 gives a pseudo-code notation, how to determine the profit for a given refueling amount.

As an analytical approach is hard, our heuristic just calculates the profit for different refueling amounts (see Fig. 10). We assume that the revenue function is unimodal and perform a ternary search to choose the refueling amount with the highest expected profit.

Algorithm 7: PROFIT (s, i, k, fuel level f, refueling r, marginal cost c)

Output: Estimated profit of the cycle when refueling r
begin
  $profit \leftarrow -r \cdot \mathrm{C}_{i,k}$
  $f \leftarrow$ fuel level, when refueling r on top of f
  for *time step* $t \leftarrow t^*_{i,k}$ to $t^+_{i,k}$ **do**
    $prod \leftarrow$ production with respect to the fuel level, modulation, unit cost, ...
    $profit \leftarrow profit + prod \cdot c_t$
  if *next cycle is scheduled* **then**
    $profit \leftarrow profit - \frac{\mathrm{C}_{i,k+1}}{\mathrm{Q}_{i,k+1}} \cdot f$
  else
    $profit \leftarrow profit + f \cdot \mathrm{C}_{i,\mathrm{T}}$
  return *profit*
end

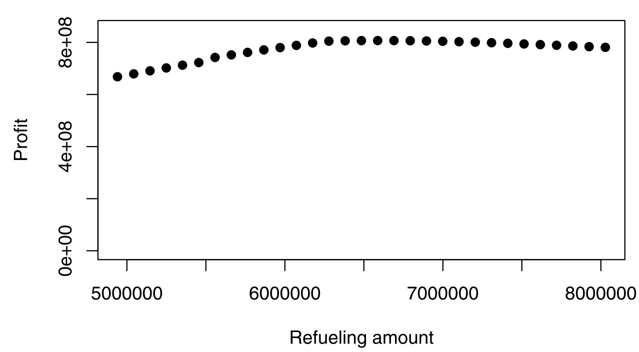

Fig. 10 Fuel level tuning: a sample plot of the profit gained for different refueling amounts. The data are taken from A05, power plant 0, cycle 1

7 Evaluation

7.1 Implementation and testing environment

We implemented our solution in C++ and utilize two external libraries. For the solution of the outage scheduling phase (cf. Sect. 5) we extend the constraint programming toolkit Gecode 3.3.2 (GECODE 2010). To solve the minimum cost flow problem of the network-based lower bound (cf. Sect. 3.3) we use the implementation of the cost-scaling algorithm from the Lemon Graph Library 1.2 (LEMON Graph 2010).

All code has been compiled with GCC 4.3, using optimization level 3. Experiments have been performed on one core of a 2x dual-core AMD Opteron Processor 2218 clocked at 2.6 GHz, equipped with 1 MB L2 Cache per core and 32 GB RAM. This setup is slightly less powerful than the reference setup used in the competition (2x quad-core Intel Xeon 5420 at 2.5 GHz, 12 MB L2 Cache, 8 GB RAM). Our solution runs as a single thread and does not consume more than 8 GB of memory per process.

7.2 Provided datasets

Three datasets (A, B and X) have been provided by the ROADEF/EURO Challenge 2010. Set A contains six small instances, while B and X comprise five big instances each. Their key figures are shown in Table 5. Note that the ratio between type 1 and type 2 power plants is nearly 1 in set A, while B and X comprise 2–3 times more type 2 power plants than type 1 power plants. As most of our constraint types are

Table 5 Overview of the instances provided by the ROADEF/EURO Challenge 2010. Column 14* is the sum of CT 14–18. The column for CT 19 was omitted since each instance (except A00 which had none) contained exactly one constraint of this type

Set	# Constraints								
	J	I	S	W	T	13	14*	20	21
A00	1	2	2	89	623	4	1	0	0
A01	11	10	10	250	1750	46	11	1	1
A02	21	18	20	250	1750	84	17	1	1
A03	21	18	20	250	1750	80	18	1	1
A04	31	30	30	250	1750	122	23	1	1
A05	31	28	30	250	1750	120	22	1	3
B06	25	50	50	277	5817	222	77	50	5
B07	27	48	50	265	5565	192	70	50	5
B08	19	56	121	277	5817	114	86	50	5
B09	19	56	121	277	5817	114	86	50	5
B10	19	56	121	265	5565	235	86	50	5
X11	25	50	50	277	5817	239	77	50	5
X12	27	48	50	263	5523	207	70	50	5
X13	19	56	121	277	5817	260	86	50	5
X14	19	56	121	277	5817	256	86	50	5
X15	19	56	121	263	5523	245	86	50	5

Table 6 Lower bounds in millions: we compare both lower bounds with respect to the obtained objective function *value* and its *score* (deviation from the best known solution, cf. Sect. 7.2)

Set	Auction		Network	
	Value	Score	Value	Score
A00	8 676 507 M	−0.66 %	8 701 730 M	−0.34 %
A01	160 847 M	−5.13 %	165 560 M	−2.35 %
A02	130 148 M	−10.09 %	139 991 M	−4.15 %
A03	137 744 M	−10.80 %	148 454 M	−3.87 %
A04	82 428 M	−23.45 %	102 326 M	−8.30 %
A05	94 379 M	−24.99 %	112 467 M	−10.61 %
B06	36 338 M	−56.44 %	69 592 M	−16.58 %
B07	38 263 M	−52.86 %	68 528 M	−15.58 %
B08	28 410 M	−65.32 %	62 594 M	−23.60 %
B09	30 000 M	−63.30 %	63 991 M	−21.72 %
B10	30 389 M	−60.92 %	63 747 M	−18.03 %
X11	33 377 M	−57.81 %	66 931 M	−15.40 %
X12	36 740 M	−52.65 %	66 558 M	−14.22 %
X13	25 789 M	−66.27 %	62 155 M	−18.70 %
X14	26 903 M	−64.68 %	63 045 M	−17.23 %
X15	28 444 M	−62.13 %	61 866 M	−17.62 %

related to type 2 power plants the B and X instances require much more effort.

All instances—except the dummy-instance A00—contain six cycles for each type 2 power plant. From the original problem description we derive that a time step's duration in real time is between 8 and 24 hours. The demand scenarios show similar characteristics and seem realistic (e.g., clear seasonal oscillation). Instances B08 and B09 are a bit special as they contain less outage date range constraints CT 13, which dramatically increases the search space, making it harder to find good solutions. Furthermore, in B08 and X13 all refueling amounts are fixed (i.e., $\mathrm{RMIN}_{i,k} = \mathrm{RMAX}_{i,k}$).

All instances contain one type 1 power plant that is able to cover the complete demand—at extremely high cost though. This plant can be seen as a backup plant to keep the problems feasible. As it is never desirable to utilize this plant, we omit it in all plots and figures.

To make solution qualities comparable, the ROADEF /EURO Challenge 2010 defines a so called *score* on each result, which corresponds to the result's deviation from the best known solution in percent. Assume that R is our solution and R' the best known solution so far, then the score is calculated as:

$$score(R) := \frac{cost(R)}{cost(R')} - 1$$

where *cost* returns the result's value of the objective function. In the competition teams are ranked by their average score for all B and X instances.

7.3 Lower bounds

In this section we evaluate the proposed lower bounds, the results can be seen in Table 6. Both methods yield stable deviations from the best known results. Obviously, the network-based lower bound gives far better results than the auction-based bound. The reason is that the network is designed to overcome the disadvantages of the auction (underestimated unit cost of type 2 power plants and overestimated production capacity). In the flow network, the cost of production by type 2 power plants is really mapped onto a concrete outage and billed with the refueling. Furthermore, the network models outages by limiting the production of the power plants over a certain period of time, while outages are not considered in the auction approach.

Therefore, the remaining gap between the network-based lower bound and an optimal solution originates mainly from the unconsidered spacing and resource constraints (CT 14–21) and the caveats against optional outages.

The lower bounds for instances from sets B and X differ much more from their best known solution than instances from set A do. This probably results from the larger number of outages and spacing constraints in the respective instances. Consequently, the influence of the spacing constraints, which are not modeled in the lower bound, grows

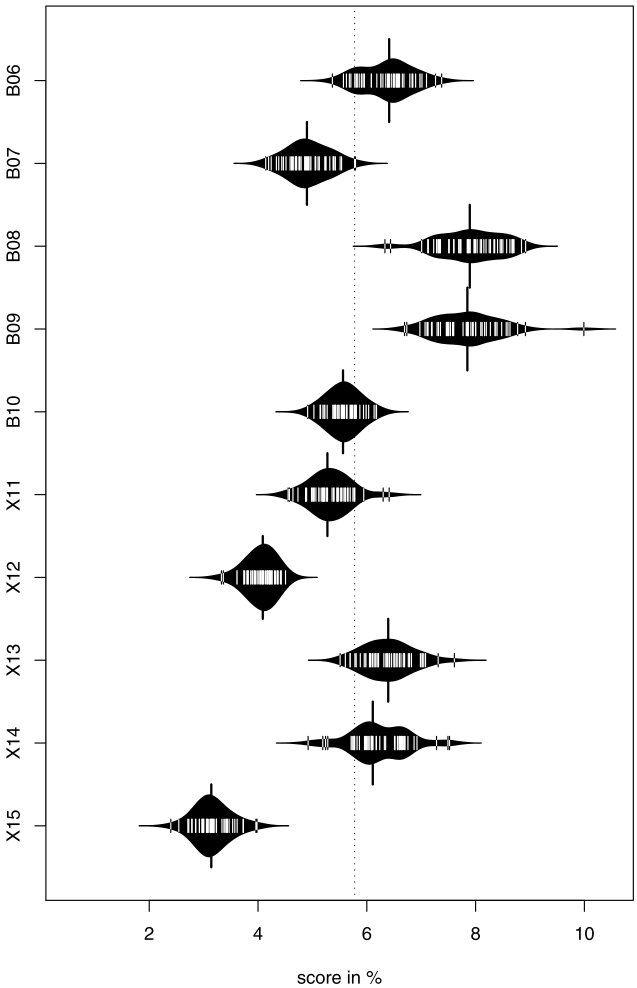

Fig. 11 Robustness of our solution: we show the *scores* (cf. Sect. 7.2) of 50 iterations on the B and X instances. *Ticks* mark single solutions. The *thick vertical lines* mark the median score of each instance. Note that the average expected score (*dotted line*) after one iteration of each instance is less than 6 %

and makes real solutions close to the lower bound hard to obtain.

As the obtained scores of our lower bounds are similar among the instances from sets B and X, we assume that these instances are somewhat equally hard to solve. Thus, a good solution should also yield similar scores for all instances.

The computational effort to compute the bounds differs dramatically: while the auction-based bound requires less than 10 seconds to compute, solving the flow-network for all scenarios takes up to an hour for the biggest datasets.

7.4 Best solutions

At last, we present the best solutions found by our final approach (see Table 7). For insights on the impact of different components of the method we refer to Brandt (2010).

For all experiments we used a computation time as allowed in the ROADEF/EURO Challenge 2010, i.e., 30 minutes per instance for dataset A and 60 minutes per instance for datasets B and X. First of all, we note that our obtained solution quality for A instances is much better than for B and X instances. This probably originates from some nice properties of the A instances: they comprise less spacing constraints and outages to schedule, thus the search spaces are smaller.

Table 7 Best solutions in millions. We compare the costs of the best known results as published after the competition (see Pocheron et al. 2010) to our best achieved results

Set	Best known solution	Own best solution	Score
A00	8 730 985 M	8 735 652 M	0.05 %
A01	169 538 M	169 661 M	0.07 %
A02	146 048 M	146 226 M	0.12 %
A03	154 429 M	154 775 M	0.22 %
A04	111 591 M	112 106 M	0.46 %
A05	125 822 M	126 509 M	0.55 %
B06	83 424 M	87 901 M	5.37 %
B07	81 174 M	84 535 M	4.14 %
B08	81 926 M	86 308 M	5.35 %
B09	81 750 M	87 092 M	6.53 %
B10	77 767 M	81 587 M	4.91 %
X11	79 116 M	82 718 M	4.55 %
X12	77 589 M	80 171 M	3.33 %
X13	76 449 M	80 345 M	5.10 %
X14	76 172 M	79 921 M	4.92 %
X15	75 101 M	76 901 M	2.40 %

Among the B and X instances we achieve good results with a robust deviation from the best known solution (see Fig. 11), but we have not been able to improve the best known results for the provided instances. Across these instances we gained an average score of 4.66 % which would result in the second best rating in the competition. However, we have to acknowledge that probably most of the participating teams are able to improve their results after all instances have been revealed or provide solutions to instances they failed to solve in the competition.

8 Conclusion

In this paper we developed a constraint programming and greedy heuristics based approach for the problem posed by the ROADEF/ EURO Challenge 2010, which contained an industry scale power plant production planning problem in an uncertain and highly constrained environment. The overall objective function was to reduce the total operating cost.

We developed two approaches that compute lower bounds for the problem. One is a fast greedy heuristic, the other is based on a flow network and yields closer bounds at the cost of high computational effort. We decomposed the main problem into two stages: outage scheduling and production assignment. For the first stage we formulated the given model as a constraint program and added further constraints which ensure the feasibility of the remaining production assignment problem. In the second stage we utilized a greedy heuristic—developed from our greedy lower bound—to assign production levels and refueling amounts for a given outage schedule.

We implemented all the proposed methods and evaluated their performance on 16 challenge instances. Further work might be done to improve the lower bounds by a better preprocessing (e.g., to reduce the outage date domains). Besides, local search techniques are not considered yet and might be able to improve the presented solution approach.

Finally, we related this work to the results obtained by the participating teams of the ROADEF/EURO Challenge 2010. Our approach is able to solve all proposed instances, which only 4 of 21 finalist teams did, and achieves competitive results with an average deviation from the best known results of less than 5 %. This corresponds to the second best score achieved in the competition.

References

Brandt, F. (2010). *Solving a large-scale energy management problem with varied constraints*. Karlsruhe Institute of Technology (KIT), Faculty of Informatics. http://digbib.ubka.uni-karlsruhe.de/volltexte/1000022490.

Dentcheva, D., & Römisch, W. (1997). Optimal power generation under uncertainty via stochastic programming. In K. Marti & P. Kall (Eds.), *Lecture notes in economics and mathematical systems: Vol. 458. Stochastic programming methods and technical applications* (pp. 22–56). Berlin: Springer.

Feltenmark, S., Kiwiel, K. C., & Lindberg, P. (1996). Solving unit commitment problems in power production planning. In *Operations research proceedings* (pp. 236–241). Berlin: Springer.

Foong, W. K., Maier, H. R., & Simpson, A. R. (2008). Power plant maintenance scheduling using ant colony optimization: an improved formulation. *Engineering Optimization*, *40*, 309–329.

Gardi, F., & Nouioua, K. (2011). Local search for mixed-integer nonlinear optimization: a methodology and an application. In *Lecture notes in computer science: Vol. 6622. Proceedings of EvoCOP 2011, the 11th European conference on evolutionary computation in combinatorial optimisation* (pp. 167–178).

GECODE—a generic constraint development environment. (2010). http://www.gecode.org.

Godskesen, S., Jensen, T. S., Kjeldsen, N., & Larsen, R. (2010). Solving a real-life large-scale energy management problem. ArXiv:1012.4691.

Khemmoudj, M. O. I., Porcheron, M., & Bennaceur, H. (2006). When constraint programming and local search solve the scheduling problem of electricité de France nuclear power plant outages. In *Lecture notes in computer science: Vol. 4204. Principles and practice of constraint programming* (pp. 271–283).

LEMON Graph (2010). *Library*. http://lemon.cs.elte.hu.

Lusby, R. M., Muller, L. F., & Petersen, B. (2010). *A solution approach to the ROADEF/EURO 2010 challenge based on Benders' decomposition* (Tech. Rep. DTU Management 2010; 18). Technical University of Denmark, Department of Management Engineering.

Ngundam, J. M., Kenfack, F., & Tatietse, T. T. (2000). Optimal scheduling of large-scale hydrothermal power systems using the Lagrangian relaxation technique. *International Journal of Electrical Power & Energy Systems*, *22*(4), 237–245. doi:10.1016/S0142-0615(99)00054-X.

Padhy, N. (2004). Unit commitment—a bibliographical survey. *IEEE Transactions on Power Systems*, *19*(2), 1196–1205.

Pocheron, M., Gorge, A., Juan, O., Simovic, T., & Dereu, G. (2010). *Challenge ROADEF/EURO 2010: a large-scale energy management problem with varied constraints*. http://challenge.roadef.org/2010/.

Satoh, T., & Nara, K. (1991). Maintenance scheduling by using simulated annealing method [for power plants]. *IEEE Transactions on Power Systems*, *6*, 850–857.

Sen, S., & Kothari, D. P. (1998). Optimal thermal generating unit commitment: a review. *International Journal of Electrical Power & Energy Systems*, *20*(7), 443–451.