Mathematical Programming bounds for Large-Scale Unit Commitment Problems in Medium-Term Energy System Simulations

Alberto Ceselli[1], Alberto Gelmini[2], Giovanni Righini[1], and Andrea Taverna[2,3]

1 **Università degli Studi di Milano**
Dipartimento di Informatica, Via Bramante 65 – 26013 Crema (CR), Italy
{alberto.ceselli,giovanni.righini}@unimi.it

2 **Ricerca sul Sistema Energetico – RSE S.p.A., Via Rubattino 54 – 20134 Milano, Italy**
alberto.gelmini@rse-web.it

3 **Università degli Studi di Milano**
Dipartimento di Matematica, Via Saldini 50 – 20133 Milano, Italy
andrea.taverna@unimi.it

Abstract

We consider a large-scale unit commitment problem arising in medium-term simulation of energy networks, stemming from a joint project between the University of Milan and a major energy research centre in Italy. Optimal plans must be computed for a set of thermal and hydroelectric power plants, located in one or more countries, over a time horizon spanning from a few months to one year, with a hour-by-hour resolution. We propose a mixed-integer linear programming model for the problem. Since the complexity of this unit commitment problem and the size of real-world instances make it impractical to directly optimise this model using general purpose solvers, we devise ad-hoc heuristics and relaxations to obtain approximated solutions and quality estimations. We exploit an incremental approach: at first, a linear relaxation of an aggregated model is solved. Then, the model is disaggregated and the full linear relaxation is computed. Finally, a tighter linear relaxation of an extended formulation is obtained using column generation. At each stage, matheuristics are run to obtain good integer solutions. Experimental tests on real-world data reveal that accurate results can be obtained by our framework in affordable time, making it suitable for efficient scenario simulations.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases mathematical programming, unit commitment, power systems

Digital Object Identifier 10.4230/OASIcs.SCOR.2014.63

1 Introduction

The Unit Commitment Problem (UCP) consists in finding the optimal production levels for plants with discrete activation patterns, i.e. plants that can be turned off and on. The body of literature on UCPs is huge and spans both theory and applications, as recent reviews like [1] and [2] report.

The most common objective is to minimise the global cost of production. Indeed, for global system simulations like ours, used, for instance, by power exchange authorities, the minimisation of global costs allows to maximise also the general welfare of the system. The UCP has traditionally been used to model thermal power plants schedules in power systems

4th Student Conference on Operational Research (SCOR'14).
Editors: Pedro Crespo Del Granado, Martim Joyce-Moniz, and Stefan Ravizza; pp. 63–76
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

on daily or weekly horizons to support operational decisions. However, recent applications have an increasing need for simulating energy networks on substantially longer periods, and including other types of plants such as hydroelectric, nuclear and waste-to-energy plants, as well as plants from renewable sources like wind, solar and biomass.

UCP models admit several variations that require different solving techniques. A distinctive characteristic is the length of the simulation horizon. Short-term models, from a few hours to weeks, are used to guide operational decisions: they are detailed but relatively small. These include complicating elements such as non-linear costs or resource consumption functions, to accurately represent thermal units. Medium and long-term models, that aim to support strategic decisions, are larger in size and span longer time horizons: they require more robust and efficient solution methods, but are less detailed.

UCP models typically include inter-temporal constraints for thermal plants, that bound or penalise the change in activity level of each unit between consecutive periods. For example, ramping constraints limit the change in production levels, minimum up and down time constraints prevent units from switching state too frequently, and start-up penalties model the costs that producers incur when their plants are switched on [3]. A large body of literature covers short-term UCPs. Among the approximated methods greedy algorithms [4] and meta-heuristic approaches [5, 6] have been proposed. These offer flexibility, but require fine-tuning to be effective, and provide no optimality guarantee. Mathematical programming is employed when better control on solutions quality is needed: short-term UCPs can be conveniently modelled as mixed-integer non-linear programs (MINLPs). Unfortunately, these MINLPs are in general too hard to be solved on real-sized instances. Only the very special case of single-unit has been successfully handled with exact algorithms [10]. Instead, Lagrangian relaxation schemes [7, 8, 9] are often used to obtain good approximations.

Linear models and mixed-integer linear programming (MILP) techniques have also been tried. Linear UCP approximations with a weekly horizon and more than 100 plants have been effectively solved via commercial MILP solvers [11], or embedded in rounding algorithms and used to solve real instances [12]. In [13] a branch-and-cut scheme allows to iteratively improve a model including a piecewise linear approximation of thermal unit costs.

Instances involving up to 100 thermal units and 200 hydroelectric plants on a weekly scenario were solved in this way. More recently, MINLP and MILP models have been coupled in a hybrid scheme [14].

Fewer publications deal with medium-term UCPs, spanning over months or years. Recent contributions include [15], in which a MILP model is devised for the simulation of the annual power production in Denmark. The authors combine a set of constructive heuristics, that compute approximated and potentially infeasible solutions, with improving methods, that compute feasible and better solutions starting from the heuristic ones. Computational results are reported on instances with 20 thermal plants: annual solutions are produced with computing times ranging between 5 and 10 minutes.

In this paper we face large-scale medium-term UCPs, to be solved on time periods ranging from a few months to the whole year. The aim is to support the simulation of the Italian energy market, where producers bid against one another to sell energy to buyers, i.e. private or public entities who provide energy to final consumers [16, 17]. Prices are then determined by the interaction between demand and supply. Cross-border flows are regulated: the pricing and the volumes exchanged are decided beforehand with a bidding mechanism according to a forecast of the future prices or demands in both systems. Indeed, mismatches in forecasting can cause discharges, blackouts, or simply the need to buy energy at very high prices, or sell it at negligible ones. Plant and system level features have to be accounted for. At plant-level, both dispatchable (e.g. thermal units) and non-dispatchable power sources (e.g. renewable energy units) appear in the system. The latter cannot vary their production levels according to market players and thus their contribution to the supply is assumed to be known in advance through forecasting. Dispatchable power sources can be of either hydroelectric or thermal type. Hydroelectric plants, as other renewables, have negligible marginal production costs. Thermal plants are the most critical entities to model, having significant costs for fuel, periodic operations and maintenance, constraints and costs associated with the changes in the production levels and plants' states between consecutive periods. Finally, there are cap-pricing schemes that impose penalties for pollution, namely the production of CO_2 and NO_X gases, in each country and at the European level.

In Section 2 we introduce the model. In Section 3 we describe how to relax it in order to effectively obtain lower bounds on optimal solutions. In Section 4 we describe matheuristics and rounding procedures providing upper bounds. Finally, in Section 5 we report and discuss computational results on real-world instances, and in Section 6 we draw some conclusions.

2 Model

In the UCP variant we consider, plants exchange production through a power network. The network connects different zones, each hosting a set of hydroelectric and thermoelectric plants, by means of power links of limited capacity. Each zone has its own hourly demand to be satisfied by either plants in the zone or by import from nearby ones. Dispatching and production decisions have to be taken hour-by-hour. Hydroelectric plants have basins filled by lateral water inflow or pumping systems. Their production in each period linearly depends on the outbound flow from the basin and is assumed to be costless. Thermal plants need to be ignited and heated to be active. As such they have a binary activation state, fixed production costs, non-zero technical minimum and reduced flexibility, i.e. they have to maintain their state for a given amount of time. Production costs linearly depend on production levels and include pollution penalties. Some thermal plants have "double-shaft" technology; that is, they can switch between two working states, with one of them employing more power units and allowing for higher production levels. In our formulation thermal plants in each zone are partitioned in groups. Each of them is characterised by plants with identical marginal cost. Each group is further partitioned in families, having also the same technical minima, maxima and fixed costs.

Let T be the set of time periods. Let Z be the set of zones, $Y \subseteq Z$ denote the subset of zones from which energy can be exchanged with external systems, and $A \subseteq Z \times Z$ be the set of links between zones. Let

- H_z be the set of hydroelectric plants with reservoir in zone $z \in Z$, and $H = \bigcup_{z \in Z} H_z$,
- G_z be the set of groups of thermal plants for zone $z \in Z$, and $G = \bigcup_{z \in Z} G_z$,
- M_{zg} be the set of families of plants for zone $z \in Z$ and group $g \in G_z$, $M = \bigcup_{z \in Z, g \in G_z} M_{zg}$,
- $M^D_{zg} \subseteq M_{zg}$ be the subset of families of plants in zone $z \in Z$ and group $g \in G_z$ implementing double-shaft technology.

For each zone $z \in Z$, group $g \in G_z$ and family $m \in M_{zg}$, thermal plants are characterised by the following data:

- k_{zgm} and k^D_{zgm} (number of plants in family $m \in M_{zg}$ and $m \in M^D_{zg}$, resp.),
- c_{tzg} and e_{tzgm} (marginal and fixed production cost at time $t \in T$, resp.),
- p_{zgm} and P_{zgm} (minimum and maximum power produced by plants in family $m \in M_{zg}$, resp.),

- p^D_{zgm} and P^D_{zgm} (minimum and maximum power produced by plants in family $m \in M^D_{zg}$, resp., when in double-shaft mode),
- on_{zgm} and off_{zgm} (minimum time for which plants in family $m \in M_{zg}$ have to stay active, resp. inactive, once turned on, resp. off).

Let

$$T^{on}_{tzgm} = \{t' \in T : t \le t' \le \min(|T|, t + on_{zgm} - 1)\} \cup \{t' \in T : 1 \le t' \le (t + on_{zgm} - |T|)\}$$

be the set of periods in which a plant $m \in M_{zg}$ has to remain active if turned on at time $t \in T$, and

$$T^{\mathit{off}}_{tzgm} = \{t' \in T : t \le t' \le \min(|T|, t + \mathit{off}_{zgm} - 1)\} \cup \{t' \in T : 1 \le t' \le (t + \mathit{off}_{zgm} - |T|)\}$$

be the set of periods in which it has to remain inactive if turned off.

For each zone $z \in Z$, each hydroelectric plant $h \in H_z$ is characterised by the following data:

- p_h and P_h (minimum and maximum power produced, resp.),
- P^β_h (maximum pumping power),
- q_{zh} and Q_{zh} (volume available in the reservoir at the beginning of simulation, and required to be in the reservoir at the end of simulation, resp.),
- V_h (basin capacity),
- α_h and β_h (energy conversion and pumping efficiency coefficients),
- f_h and n_h (hourly maximum outflow and lateral inflow).

Furthermore, at each time period $t \in T$, let

- b_{tij} be the maximum energy transfer capacity of link $(i, j) \in A$,
- d_{tz} be the demand of zone $z \in Z$,
- E_t be the price of imported energy.

We introduce, for each period $t \in T$, zone $z \in Z$ and group $g \in G_z$, continuous variables x_{tzg} that represent the overall production level, and, for each family $m \in M_{zg}$, integer variables y_{tzgm}, y^D_{tzgm}, up_{tzgm}, dn_{tzgm}, up^D_{tzgm} and dn^D_{tzgm}, that represent the number of plants that are resp. active, active in double-shaft mode, switched on, switched off, entered and exited from double-shaft mode. We also consider, for each period $t \in T$, zone $z \in Z$ and plant $h \in H_z$, continuous variables $l_{tzh}, m_{tzh}, s_{tzh}, o_{tzh}$, that represent production level, pumping power, excess outbound flow from the basin, and reservoir volume, resp., for hydroelectric plants.

Finally, we assume the energy distribution network to have tree topology, since this is the case in Italy. Nevertheless our model can be extended to arbitrary structures. Continuous variables w_{tij} represent the amount of energy flowing through link $(i, j) \in A$ at time $t \in T$, and imp_{tz} and exc_{tz} the energy imported from external systems and the exceeding production in zone $z \in Y$ at time $t \in T$, respectively. As explained in the introduction, cross-border flows are regulated beforehand and unforeseen imports or exports are not expected. Therefore imp_{tz} and exc_{tz} variables are introduced only to detect issues in data forecasting or actual problems in the simulated system. Then E_t is meant to be set to a high value to minimise the use of imported energy and the exceeding production is assumed to be lost. Our UCP can be formulated as the following MILP.

$$\min \quad \phi = \sum_{\substack{t\in T, z\in Z,\\ g\in G_z}} c_{tzg} x_{tzg} + \sum_{\substack{t\in T, z\in Z,\\ g\in G_z, m\in M_{zg}}} e_{tzgm} y_{tzgm} + \sum_{t\in T, z\in Z} imp_{tz} E_t \tag{1a}$$

$$\text{s. t. } x_{tzg} \geq \sum_{m\in M_{zg}} p_{zgm} y_{tzgm} + \sum_{m\in M^D_{zg}} (p^D_{zgm} - p_{zgm}) y^D_{tzgm} \ \forall t\in T, z\in Z, g\in G_z \tag{1b}$$

$$x_{tzg} \leq \sum_{m\in M_{zg}} P_{zgm} y_{tzgm} + \sum_{m\in M^D_{zg}} (P^D_{zgm} - P_{zgm}) y^D_{tzgm} \ \forall t\in T, z\in Z, g\in G_z \tag{1c}$$

$$y^D_{tzgm} \leq y_{tzgm} \ \forall t\in T, z\in Z, g\in G_z, m\in M^D_{zg} \tag{1d}$$

$$up_{tzgm} \geq y_{tzgm} - y_{(t-1)zgm} \ \forall t\in T, z\in Z, g\in G_z, m\in M_{zg} \tag{1e}$$

$$dn_{tzgm} \geq y_{(t-1)zgm} - y_{tzgm} \ \forall t\in T, z\in Z, g\in G_z, m\in M_{zg} \tag{1f}$$

$$up^D_{tzgm} \geq y^D_{tzgm} - y^D_{(t-1)zgm} \ \forall t\in T, z\in Z, g\in G_z, m\in M^D_{zg} \tag{1g}$$

$$dn^D_{tzgm} \geq y^D_{(t-1)zgm} - y^D_{tzgm} \ \forall t\in T, z\in Z, g\in G_z, m\in M^D_{zg} \tag{1h}$$

$$y_{tzgm} \geq \sum_{\tau\in T: t\in T^{on}_\tau} up_{\tau zgm} \ \forall t\in T, z\in Z, g\in G_z, m\in M_{zg} \tag{1i}$$

$$y_{tzgm} \leq k_{zgm} - \sum_{\tau\in T: t\in T^{off}_\tau} dn_{\tau zgm} \ \forall t\in T, z\in Z, g\in G_z, m\in M_{zg} \tag{1j}$$

$$y^D_{tzgm} \geq \sum_{\tau\in T: t\in T^{off}_\tau} up^D_{\tau zgm} \ \forall t\in T, z\in Z, g\in G_z, m\in M^D_{zg} \tag{1k}$$

$$y^D_{tzgm} \leq k^D_{zgm} - \sum_{\tau\in T: t\in T^{off}_\tau} dn^D_{\tau zgm} \ \forall t\in T, z\in Z, g\in G_z, m\in M^D_{zg} \tag{1l}$$

$$o_{1zh} = q_{zh} \ \forall z\in Z, h\in H_z \tag{1m}$$

$$o_{(|T|+1)zh} = Q_{zh} \ \forall z\in Z, h\in H_z \tag{1n}$$

$$o_{tzh} + n_{th} + \beta_h \cdot m_{tzh} = o_{(t+1)zh} + s_{tzh} + l_{tzh} \ \forall t\in T, z\in Z, h\in H_z \tag{1o}$$

$$\begin{aligned} &\sum_{h\in H_z} \alpha_h \cdot l_{tzh} + \sum_{g\in G_z} x_{tzg} + \sum_{(i,z)\in A} w_{tiz} + \sum_{z\in Y} imp_{tz} \geq \\ &\quad d_{tz} + \sum_{h\in H_z} m_{tzh} + \sum_{(z,j)\in A} w_{tzj} + \sum_{z\in Y} exc_{tz} \end{aligned} \quad \forall t\in T, z\in Z \tag{1p}$$

$$y_{tzgm}, up_{tzgm}, dn_{tzgm} \in [0, k_{zgm}] \cap \mathbb{Z}^+_0 \ \forall t\in T, z\in Z, g\in G_z, m\in M_{zg} \tag{1q}$$

$$y^D_{tzgm}, up^D_{tzgm}, dn^D_{tzgm} \in [0, k^D_{zgm}] \cap \mathbb{Z}^+_0 \ \forall t\in T, z\in Z, g\in G_z, m\in M^D_{zg} \tag{1r}$$

$$w_{tij} \in [0, b_{ij}] \ \forall t\in T, (i,j)\in A \tag{1s}$$

$$s_{tzh} \in [0, f_h], o_{tzh} \in [0, V_h], l_{tzh} \in [p_h, P_h], m_{tzh} \in [0, P^\beta_h] \ \forall t\in T, z\in Z, h\in H_z \tag{1t}$$

$$imp_{tz} \geq 0, exc_{tz} \geq 0 \ \forall t\in T, z\in Y \tag{1u}$$

Constraints (1b) and (1c) impose that production level is 0 for inactive plants, and within production bounds for active ones. Constraints (1d) impose that only active plants can enter double-shaft mode. Constraints (1e)–(1h) enforce consistency between variables describing activation patterns. Constraints (1i)–(1l) impose that activation patterns respect minimum on and off times after switching. Finally (1m)–(1p) are flow conservation constraints ensuring energy balance between zones and consistency with thermal and hydroelectric productions inside each zone. The objective (1a) is to minimise the sum of production and additional energy import costs.

Due to the peculiarity of the UCP we consider, the resulting formulation (1) is significantly different than those previously proposed in the literature. A more detailed discussion on modelling issues is presented in [16] and [17].

3 Lower bounds

Preliminary experiments revealed that in large scale instances (a) even solving the continuous relaxation of formulation (1a)–(1u) (CR in the remainder) is computationally demanding, and (b) the bound obtained in this way has a non-negligible optimality gap. Therefore, we first propose to aggregate parts of the model, to obtain a relaxation that, although potentially weaker, can be solved more efficiently, coping with issue (a). Then we propose a decomposed model that has an exponential number of variables, but a reduced number of constraints. By optimising it through column generation we are able to obtain tighter bounds, coping with issue (b). These techniques are then meant to be used sequentially.

3.1 Aggregate Continuous Relaxation

For each $t \in T, z \in Z, g \in G_z$, let

$$\begin{aligned}
\tilde{e}_{tzg} &= \min_{m \in M_{zg}} \{e_{tzgm}\}, \\
P^x_{zg} &= \sum_{m \in M_{zg}} P_{zgm} k_{zgm} + \sum_{m \in M^D_{zg}} (P^D_{zgm} - P_{zgm}) k^D_{zgm} \\
\tilde{c}_{tzg} &= \frac{\tilde{e}_{tzg}}{P^x_{zg}}.
\end{aligned}$$

We consider an aggregate continuous relaxation (ACR) given by the following model:

$$\min \quad \tilde{\phi} = \sum_{\substack{t \in T, z \in Z, \\ g \in G_z}} \tilde{c}_{tzg} x_{tzg} + \sum_{t \in T, z \in Z} imp_{tz} E_t \tag{2a}$$

$$\text{s.t.} \quad 0 \le x_{tzg} \le P^x_{zg} \ \forall t \in T, z \in Z, g \in G_z \tag{2b}$$

$$(1m) - (1p)$$

that intuitively is a linear continuous model obtained by removing all integer variables, and by approximating the piecewise-linear cost function of each thermal group in each period, which may be non-continuous or non-differentiable, with a continuous linear lower-bound.

Model (2) can be shown to provide weaker bounds than CR, unless each group is composed by a single unit, in which case they coincide. On the other hand it can be solved more efficiently than CR as it is smaller and can be formulated as a network flow problem, for which well-known exact polynomial time algorithms can be used.

3.2 Decomposed model

For all $z \in Z, g \in G_z, m \in M_{zg}$ let

$$S_{zgm} = \{ \left(\hat{y}^u_{tzgm}, \hat{y}^{uD}_{tzgm}, \hat{up}^u_{tzgm}, \hat{up}^{uD}_{tzgm}, \hat{dn}^u_{tzgm}, \hat{dn}^{uD}_{tzgm} \right)^{\mathsf{T}}_{t \in T} | (1d) - (1l), (1q) - (1r) \}$$

be the set of feasible activation patterns of thermal plants of a given family on the whole horizon. The following linear program represents the Dantzig-Wolfe reformulation [18] of the continuous relaxation of model (1), where the feasible region defined by constraints (1d)–(1l), (1q)–(1r) has been replaced by the convex hull of its extreme integer points:

$$\min \quad \hat{\phi} = \sum_{\substack{t\in T, z\in Z,\\ g\in G_z}} c_{tzg} x_{tzg} + \sum_{\substack{z\in Z, g\in G_z,\\ m\in M_{zg}, u\in S_{zgm}}} \gamma^u_{zgm} (\sum_{t\in T} \hat{y}^u_{tzgm} e_{tzgm}) + \sum_{t\in T, z\in Z} imp_{tz} E_t \tag{3a}$$

$$\text{s.t.} \quad x_{tzg} \geq \sum_{\substack{m\in M_{zg},\\ u\in S_{zgm}}} (\hat{y}^u_{tzgm} p_{zgm} + \hat{y}^{uD}_{tzgm} p^D_{zgm}) \gamma^u_{zgm} \;\; \forall t \in T, z \in Z, g \in G_z \tag{3b}$$

$$x_{tzg} \leq \sum_{\substack{m\in M_{zg},\\ u\in S_{zgm}}} (\hat{y}^u_{tzgm} P_{zgm} + \hat{y}^{uD}_{tzgm} P^D_{zgm}) \gamma^u_{zgm} \;\; \forall t \in T, z \in Z, g \in G_z \tag{3c}$$

$$\sum_{u\in S_{zgm}} \gamma^u_{zgm} = 1 \;\; \forall z \in Z, g \in G_z, m \in M_{zg} \tag{3d}$$

$$\gamma^u_{zgm} \in [0,1] \;\; \forall z \in Z, g \in G_z, m \in M_{zg}, u \in S_{zgm} \tag{3e}$$

$$(1m) - (1u)$$

The reformulation details are omitted. Indeed, without loss of optimisation potential, each set S_{zgm} can include only those patterns corresponding to extreme integer points. For each such a point $u \in S_{zgm}$, integer coefficients $\hat{y}^u_{tzgm}$ and $\hat{y}^{uD}_{tzgm}$ represent the number of plants that are active in normal and double-shaft mode, resp., in the corresponding pattern. Each variable γ^u_{zgm} is 1 if pattern $u \in S_{zgm}$ is fully selected, 0 if it is not selected at all. Fractional values are feasible: constraints (3d) and (3e) enforce that a linear convex combination of points in S_{zgm} is selected for each $z \in Z, g \in G_z, m \in M_{zg}$. Constraints (3c) and (3b) are the reformulated counterparts of constraints (1c) and (1b), resp..

▶ Proposition 1. The lower bound provided by (3) is at least as tight as that given by CR.

The proof follows immediately by the Dantzig-Wolfe decomposition principle [18].

Model (3) contains a combinatorial number of variables. In fact, neglecting double shaft and inter-temporal constraints, $|S_{zgm}| = k^{|T|}_{zgm}$. Therefore we optimise it by means of column generation techniques: we start with a restricted model where each set S_{zgm} is replaced by a subset $\overline{S}_{zgm}$ containing only patterns generated by heuristics. Then we iteratively solve the restricted model, obtain a vector of dual variables, and search for columns of negative reduced cost by solving the following *pricing problem*, for each $z \in Z$, $g \in G_z$ and $m \in M_{zg}$:

$$\begin{aligned} \min \; \pi^u_{zgm} = & (\sum_{t\in T} \hat{y}^u_{tzgm} e_{tzgm}) - \eta_{zgm} + \\ & - \sum_{t\in T} (\hat{y}^u_{tzgm} p_{zgm} + \hat{y}^{uD}_{tzgm} p^D_{zgm}) \lambda_{zgm} - \sum_{t\in T} (\hat{y}^u_{tzgm} P_{zgm} + \hat{y}^{uD}_{tzgm} P^D_{zgm}) \mu_{zgm} \end{aligned} \tag{4a}$$

$$\text{s.t.} \;\; (\hat{y}^u_{tzgm}, \hat{y}^{uD}_{tzgm}, \hat{up}^u_{tzgm}, \hat{up}^{uD}_{tzgm}, \hat{dn}^u_{tzgm}, \hat{dn}^{uD}_{tzgm})_{t\in T} \in S_{zgm} \tag{4b}$$

where λ_{zgm}, μ_{zgm} and η_{zgm} are the dual variables associated with constraints (3b), (3c) and (3d), resp..

If any column of negative reduced cost is found, then the corresponding pattern is inserted in $\overline{S}_{zgm}$, and the process is iterated; otherwise, the optimal solution of the restricted model is optimal also for the full model, and therefore the process is halted.

We point out that each model (4), although being an integer linear program, asks to optimise a *single* family of plants, in a single zone and a single group. This makes it still manageable by general purpose solvers even for long time horizons.

4 Upper Bounds

In order to obtain good feasible integer solutions quickly, we designed several upper bounding heuristics [16, 17]. Two of them showed appealing results: an alternating matheuristic to be run after ACR and CR, described in subsection 4.1, and a rounding heuristic to be run at each column generation iteration, described in subsection 4.2.

4.1 Plan&Combine

As soon as an initial relaxation is computed, we run the following matheuristic, that we indicate as "Plan&Combine" (P&C). Intuitively, we first fix production levels and search for activation patterns of minimum cost for which the production levels are feasible (Plan). Then, we fix activation patterns and optimise production levels (Combine). We iterate Plan and Combine phases until no more changes are made in either phase.

Plan: for each $t \in T, z \in Z, g \in G_z$, let

$$\tilde{x}_{tzg} \in [0, \sum_{m \in M_{zg}} P_{zgm} k_{zgm} + \sum_{m \in M^D_{zg}} (P^D_{zgm} - P_{zgm}) k^D_{zgm}]$$

be a given set of feasible production levels.

We compute the minimum cost activation patterns allowing such production levels by solving the following integer linear program:

$$\min \phi_{Plan} = \sum_{\substack{t \in T, z \in Z, \\ g \in G_z, m \in M_{zg}}} e_{tzgm} y_{tzgm} \tag{5a}$$

$$\text{s.t. } \tilde{x}_{tzg} \leq \sum_{m \in M_{zg}} P_{zgm} |M_{zg}| + \sum_{m \in M^D_{zg}} (P^D_{zgm} - P_{zgm}) |M^D_{zg}| \quad \forall t \in T, z \in Z, g \in G_z \tag{5b}$$

$$(1d) - (1l)$$

It is easy to note that (5) decomposes in one independent subproblem for each $t \in T, z \in Z, g \in G_z$, making it well solvable with general purpose solvers.

Combine: for each $z \in Z, g \in G_z, m \in M_{zg}$, let

$$(\tilde{y}_{tzgm} \ldots \tilde{y}^D_{tzgm})^{\intercal}_{t \in T} \in S_{zgm}$$

be a feasible activation pattern. We compute the minimum cost production respecting minimum and maximum power levels by solving the following linear program:

$$\min \phi_C = \sum_{\substack{t \in T, z \in Z, \\ g \in G_z}} c_{tzg} x_{tzg} + \sum_{\substack{z \in Z, g \in G_z, \\ m \in M_{zg}}} e_{tzgm} \tilde{y}_{tzgm} + \sum_{t \in T, z \in Z} imp_{tz} E_t \tag{6a}$$

$$\text{s.t. } x_{tzg} \leq \sum_{m \in M_{zg}} P_{zgm} \tilde{y}_{tzgm} + \sum_{m \in M^D_{zg}} (P^D_{zgm} - P_{zgm}) \tilde{y}^D_{tzgm} \; \forall t \in T, z \in Z, g \in G_z \tag{6b}$$

$$x_{tzg} \geq \sum_{m \in M_{zg}} p_{zgm} \tilde{y}_{tzgm} + \sum_{m \in M^D_{zg}} (p^D_{zgm} - p_{zgm}) \tilde{y}^D_{tzgm} \; \forall t \in T, z \in Z, g \in G_z \tag{6c}$$

$$(1d) - (1l)$$

that is comparable in complexity to ACR (2).

The P&C heuristic works as follows: for each $i \in \mathbb{N}_0^+$

1. let $\tilde{\mathbf{x}}^i$ be a feasible vector of production levels.
2. (plan) solve mod.(5) yielding activation patterns $\tilde{\boldsymbol{y}}^i$
3. (combine) solve mod.(6) yielding a feasible solution U^{i+1} for the original model (1). Let $\tilde{\boldsymbol{x}}^{i+1}$ be the corresponding production levels for thermal plants
4. if $\tilde{\boldsymbol{x}}^i \neq \tilde{\boldsymbol{x}}^{i+1}$ then $i := i+1$, go to step 2. Otherwise stop.

It is worth noting that a solution given by a 'plan' step is always feasible for the subsequent 'combine' step, and a solution given by a 'combine' step is always feasible for the 'plan' step of the subsequent iteration. Hence the solutions produced never worsen during the iterations of P&C. It follows also that loops may occur only between solutions with the same value. Therefore, the convergence of P&C is guaranteed by considering a lexicographic order between solutions as a secondary objective function.

4.2 Column Generation rounding

At each column generation iteration we search for good integer solutions with the following Rounding Heuristic (RH). After pricing, we consider model (3): for each $z \in Z$, $g \in G_z$ and $m \in M_{zg}$, we fix to one the γ^u_{zgm} variable of highest fractional value, and we fix to zero all the remaining variables. Ties are broken according to the lexicographic order. In this way, no more integer variables are left free, and model (3) becomes a linear program: by optimising it using a suitable algorithm, like dual simplex, we obtain a full UCP solution.

5 Implementation and Results

We combined our algorithms to produce Upper Bounds (UB) and Lower Bounds (LB) on the optimal solution value with the following approach:

1. Solve ACR (obtain LB); run P&C starting from ACR optimal x_{tzg} values (obtain UB).
2. If (UB = LB) then stop (optimality is proved).
3. Solve CR (update LB); run P&C starting from CR optimal x_{tzg} values (update UB).
4. If (UB = LB) then stop (optimality is proved).
5. Populate sets $\bar{S}_{zgm}$ with solutions from steps 1 and 3, and from a pricing round using as dual values those corresponding to constraints (1c) and (1b) in the CR solution.
6. Run column generation until convergence, using RH at each iteration, updating UB.
7. Output the best UB found as final UB, and the final solution of model (3) as LB.

That is, we incrementally compute tighter bounds at the expense of higher computing efforts, stopping as soon as upper and lower bounds match.

We implemented our algorithms in AMPL [19], using IBM ILOG CPLEX 12.4 for solving both MILPs and LPs. CPLEX network simplex algorithm was used to solve ACR and Plan instances, while the barrier algorithm was selected for column generation LPs, including those in RH. P&C was stopped as soon as no improvements in the solution values were obtained, as in preliminary experiments no further improvement occurred afterwards. In each test, column generation was stopped after a time limit of $3600s$.

We performed a set of experiments on a notebook equipped with an Intel Core 2 Duo 1.2GHz processor, 4GB of RAM and running a Linux Operating System. As a benchmark, we used real data collected by RSE S.p.A.. They refer to the Italian energy market, and consist of $|Z| = 7$ zones in a tree network, three of which connected to external markets, 148 thermal plants partitioned in $|G| = 98$ groups and $|M| = 103$ families, and $|H| = 34$ groups of hydroelectric plants. Thermal plants are split in three parts, the first including 68 plants having minimum on and off times of 12 and 6 hours, resp., the second including 48 plants having minimum on and off times of 60 and 20 hours, resp., and the third including

Table 1 CPLEX performance on Italian Energy Market data.

Size	Id	Size after presolving				CPU time (s)	gap %
		Constraints	Continuous var.	Integral var.	Binary var.		
1 month	1	152,541	110,184	13,870	67,158	-	33.34
	2	152,360	110,050	13,853	67,104	-	19.52
	3	151,649	109,485	13,750	66,852	3396	0.0
	4	151,578	109,451	13,740	66,836	1452	0.0
	5	151,624	109,477	13,753	66,831	842	0.0
	6	152,406	110,075	13,864	67,112	-	41.55
	7	152,266	109,956	13,855	67,062	-	< 0.01
	8	152,492	110,149	13,861	67,150	-	45.22
	9	152,147	109,879	13,819	67,013	-	0.11
	10	152,218	109,932	13,829	67,043	-	18.29
	11	152,311	109,992	13,838	67,093	2454	0.0
	12	151,774	109,591	13,813	66,908	-	19.65
2 months	1	304,887	220,249	27,721	134,246	-	31.42
	2	302,960	218,771	27,478	133,544	-	48.63
	3	304,033	219,579	27,616	133,931	-	< 0.01
	4	304,776	220,149	27,725	134,199	-	36.53
	5	304,423	219,867	27,646	134,080	-	46.56
	6	303,740	219,332	27,595	133,864	-	40.79
3 months	1	456,617	329,831	41,493	201,113	-	47.41
	2	455,838	329,210	41,386	200,855	-	59.97
	3	457,109	330,187	41,563	201,281	-	57.96
	4	456,073	329,383	41,460	200,918	-	57.11
6 months	1	912,782	659,305	82,904	402,076	-	60.74
	2	913,597	659,919	83,019	402,375	-	60.15
12 months	1	1,826,563	1,319,378	165,951	804,489	-	-

32 plants with no constraints on minimum on and off times. However, since minimum on and off times have a strong impact on the computing performances of CPLEX, after creating groups we considered all thermal plants to have $on_{zgm} = off_{zgm} = 1$.

Demand data are given, and planning decisions required, for the full year with a hour-by-hour resolution, that is considering $T = 8760$ time slots. Besides testing our algorithms on the full 12 months horizon, we extracted three sets of instances corresponding to single months (12 instances, $T = 730$), pairs of consecutive months (6 instances, $T = 1460$), quarters (3 instances, $T = 2190$) and semesters (2 instances, $T = 4380$). The price of imported energy E_t was set to a very high value: our algorithms were always able to find solutions requiring neither energy import nor excess.

First, as a term of comparison, we performed a set of test by running the CPLEX MILP solver with default settings using model (1). The corresponding results are reported in Table 1 whose columns contain, in turn, instance size and reference, number of constraints, continuous, integer and binary variables after presolving, CPU time spent in optimising (or dash when a time limit of 4800s was hit), optimality gap at the end of computation. It can be noticed that such an approach leaves in general very large gaps even for small instances; CPLEX is able to close such a gap on four cases only, but the required computing time is very high. This preliminary check stresses the need for more computationally effective methods.

Then, we ran a set of tests with our incremental approach. In order to highlight the behaviour of each step of our method, and the relative impact of ACR and CR, we measured also intermediate bounds. Our results are reported in Table 2, which is composed by four

Table 2 Computational results of the incremental approach on Italian energy market data.

Size	Id	Continuous Lower Bounds		Plan&Combine				P&C + column generation			
		ACR	CR	after ACR		after CR		ACR init		CR init	
		Time (s)	Time (s)	Time (s)	Gap %	Time (s)	Gap %	Iter.	Gap %	Iter.	Gap %
1 month	1	7	30	46	3.55	70	0.38	30	1.44	42	0.22
	2	6	25	38	3.02	64	0.33	34	0.59	42	0.17
	3	7	25	39	2.89	65	0.30	42	0.41	50	0.17
	4	7	19	43	3.26	70	0.36	40	0.64	30	0.35
	5	6	22	39	3.07	61	0.32	38	1.44	44	0.13
	6	6	25	38	3.17	64	0.30	36	0.31	48	0.14
	7	7	26	45	3.10	68	0.29	34	0.54	41	0.16
	8	7	39	42	4.55	87	0.34	32	1.86	40	0.34
	9	8	33	42	3.20	71	0.29	36	0.52	37	0.15
	10	7	31	40	3.13	71	0.35	38	0.72	39	0.17
	11	7	27	44	3.10	67	0.32	39	0.45	44	0.15
	12	7	25	51	3.55	75	0.30	48	0.36	49	0.13
2 months	1	13	86	99	3.23	173	0.35	18	0.79	20	0.19
	2	14	84	81	2.94	168	0.33	17	0.42	20	0.12
	3	15	98	107	3.01	188	0.32	14	0.63	16	0.16
	4	18	69	97	3.57	171	0.32	18	0.75	19	0.18
	5	13	105	79	3.07	192	0.29	18	0.44	18	0.19
	6	13	105	80	3.00	194	0.31	19	0.67	20	0.15
3 months	1	23	178	153	3.09	321	0.34	11	0.76	12	0.23
	2	25	155	153	3.10	319	0.35	12	1.02	13	0.16
	3	25	112	176	3.39	271	0.30	12	0.64	11	0.28
	4	23	132	228	2.91	266	0.31	11	0.65	13	0.16
6 months	1	53	396	395	3.37	910	0.34	5	0.63	5	0.27
	2	59	303	379	3.44	891	0.33	6	1.29	4	0.22
12 months	1	125	1195	1301	3.50	-	-	-	-	-	-

blocks. The first block contains the instance size and reference. The second one contains the CPU time needed to compute ACR and CR. The third one refers to the P&C heuristics, and consists of two sub-blocks, reporting the cumulative CPU time needed to run both ACR and P&C (resp. CR and P&C) and the optimality gap $(UB - LB)/LB$ obtained, where UB is the value produced by P&C and LB is that given by ACR (resp. CR). The final block refers to the column generation process, and also consists of two sub-blocks, reporting the number of column generation iterations performed within the time limit and the optimality gap reached, when the sets $\bar{S}_{zgm}$ are initially populated with heuristic solutions obtained by running P&C after either ACR or CR.

For the 12 months full instance, the last six columns are marked with a dash, as we encountered out-of-memory problems while running P&C after CR. Therefore also the column generation process could not be started. As can be noticed by looking at columns in the second block, ACR can be computed five to ten times faster than CR. Results in the third block show that the subsequent effort for computing P&C is instead similar, but P&C on CR solutions produces much better approximations, always reaching an optimality gap lower than 0.4%. Finally, column generation is able to reduce the optimality gap, consistently reaching values below 1%, even if very few iterations are made. Computing times tend to increase slowly as the size of the instance increases, while the optimality gaps remain stable. A closer look at the computational details of our simulations reveal that Pricing and Plan models are easy to solve: CPLEX is able to solve most of the instances via presolving, and in any case is able to prove optimality at the root node of the branch-and-bound tree by performing a few LP iterations. As an overall assessment, best quality results are obtained with column generation, when the RMP is initialised with P&C using CR solutions. However, such an approach is not viable on the full-scale instance. Instead, P&C using ACR solutions shows to offer a good trade off between solutions quality and computational scalability.

Due to the peculiar features of our problem, no direct comparison with methods from the literature is possible. However, with respect to similar applications like [15] we were able to tackle instances (a) involving 7 times more thermal plants and including (b) minimum up/-down constraints, (c) double shaft operating modes, (d) hydroelectric plants and (e) requiring energy transfers among zones, with a comparable computing effort and solutions quality. Moreover, the algorithms proposed in [15] require the fine-tuning of several parameters, while ours are almost parameter-free.

6 Conclusions

We faced a large-scale medium-term UCP arising in practice, introducing both compact and extended MILP models. We designed an incremental approach, computing lower bounds of increasing complexity and accuracy, and upper bounds exploiting the corresponding relaxations. We performed experiments on instances spanning a time horizon of up to one year. In all our tests, solutions within a few percentage points from optimality can be found very early in the incremental optimisation process. On instances with a time horizon up to six months and hour-by-hour resolution, our incremental approach reaches last stage, and provides solutions that, for practical purposes, are provably within negligible distance from optimality.

References

1 B. Saravanan, S. Das, S. Sikri, and D.P. Kothari. *A solution to the unit commitment problem – a review*. Frontiers in Energy, 7(2): 223–236, 2013.

2 R. Mallipeddi, and P.N. Suganthan. *Unit commitment - A survey and comparison of conventional and nature inspired algorithms*. International Journal of Bio-Inspired Computation, 6 (2):71–90, 2014.

3 M. Peik-herfeh, H. Seifi, and M.K. Sheikh-El-Eslami. *Two-stage approach for optimal dispatch of distributed energy resources in distribution networks considering virtual power plant concept*. European Transactions on Electrical Power, in press, 2014.

4 Y. Tingfang and T. O. Ting. *Methodological priority list for unit commitment problem*. Proceedings of the 2008 International Conference on Computer Science and Software Engineering, 1:176–179, Washington DC, USA, 2008.

5 N. P. Padhy. *Unit commitment – a bibliographical survey*. IEEE Trans. on Power Systems, 19(2):1196–1205, 2004.

6 S. Salam. *Unit commitment solution methods*. World Academy of Science, Engineering and Technology, 11, 2007.

7 A. Belloni, A. L. D. S. Lima, M. E. P. Maceira, and C. A. Sagastizabal. *Bundle relaxation and primal recovery in unit problems. The Brazilian case.*. Annals of Operations Research, 120(1):21–44, 2003.

8 A. Borghetti, A. Frangioni, F. Lacalandra, and C. A. Nucci. *Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment*. IEEE Trans. on Power Systems, 18(1):313–323, 2003.

9 E. C. Finardi and E. L. Da Silva. *Solving the unit commitment problem of hydropower plants via lagrangian relaxation and sequential quadratic programming*. IEEE Trans. on Power Systems, 21(2):83–844, 2006.

10 A. Frangioni. *Solving nonlinear single-unit commitment problems with ramping constraints*. Operations Research, 54:775, 2006.

11 G. W. Chang, Y. D. Tsai, C. Y. Lai, and J. S. Chung. *A practical mixed integer linear programming based approach for unit commitment*. IEEE Power Engineering Society General Meeting, 1:221–225, 2004.

12 G. Migliavacca, A. Formaro, and A. Zani *Guida all'uso dell'interfaccia utente del simulatore MTSIM ed alla metodologia di creazione della base dati*, RSE technical report n. 08003692, 2008.

13 A. Frangioni, C. Gentile, and F. Lacalandra, *Tighter Approximated MILP Formulations for Unit Commitment Problems*. IEEE Trans. on Power Systems, 24(1): 105–113, 2009.

14 A. Frangioni, M. De Santis, and C. Gentile. *A new hybrid lagrangian-milp approaches for unit commitment problems*. Contribution at 26th European Conference on Operations Research, Rome, Italy, 2013.

15 N. H. Kjeldsen, and M. Chiarandini. *Heuristic solutions to the long-term unit commitment problem with cogeneration plants*. Computers and Operations Research, 39(2):269–282, 2012

16 A. Ceselli, and G. Righini. *Modelli ed algoritmi per problemi di ottimizzazione con variabili miste intere (Unit Commitment)*, Technical Report (in italian), RSE, Segrate, Italy, 2013.

17 A. Taverna. *Analisi e sperimentazione di algoritmi di programmazione intera per la simulazione annuale del sistema elettrico italiano (Unit Commitment Problem)*, Master's thesis, Dipartimento di Informatica, Università degli Studi di Milano, 2011.

18 G. Dantzig, and P. Wolfe. *Decomposition Principle for Linear Programs*, Operations Research 8(1), 1960.

19 R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*, Pacific Grove, CA: Brooks/Cole – Thomson Learning, 2003.

Revision Notice

This is a revised version of the eponymous paper appeared in the proceedings of SCOR 2014 (OASIcs, volume 37, `http://www.dagstuhl.de/dagpub/978-3-939897-67-5`, published in August, 2015), in which the following sentence was added in Section 5, page 72: *"However, since minimum on and off times have a strong impact on the computing performances of CPLEX, after creating groups we considered all thermal plants to have* $on_{zgm} = off_{zgm} = 1.$ ".

Dagstuhl Publishing – May 21, 2015.