

Solving Unit Commitment Problems with General Ramp Constraints

Antonio Frangioni * Claudio Gentile § Fabrizio Lacalandra ‡

September 24, 2007

Abstract

Lagrangian Relaxation (LR) algorithms are among the most successful approaches for solving large-scale hydro-thermal Unit Commitment (UC) problems; this is largely due to the fact that the Single-Unit Commitment (1UC) problems resulting from the decomposition, incorporating many kinds of technical constraints such as minimum up- and down-time requirements and time-dependent startup costs, can be efficiently solved by Dynamic Programming (DP) techniques. Ramp constraints have historically eluded efficient exact DP approaches; however, this has recently changed [18]. We show that the newly proposed DP algorithm for ramp-constrained (1UC) problems allows to extend existing LR approaches to ramp-constrained (UC); this is not obvious since the heuristic procedures typically used to recover a primal feasible solution are not easily extended to take ramp limits into account. However, dealing with ramp constraints in the subproblems turns out to be sufficient to provide the LR heuristic enough guidance to produce good feasible solutions even with no other modification of the approach; this is due to the fact that (sophisticated) LR algorithms to (UC) duly exploit the primal information computed by the Lagrangian Dual, which in the proposed approach is ramp feasible. We also show by computational experiments that the LR is competitive with those based on general-purpose Mixed-Integer Program (MIP) solvers for large-scale instances, especially hydro-thermal ones.

*Università di Pisa, Dipartimento di Informatica, Largo B. Pontecorvo 3, 56127 Pisa – Italy, e-mail: frangio@di.unipi.it

§Istituto di Analisi dei Sistemi ed Informatica “Antonio Ruberti” del C.N.R., Viale Manzoni 30, 00185 Rome – Italy, e-mail: gentile@iasi.cnr.it

‡*OptiME*, e-mail: fabrizio.lacalandra@fastwebnet.it

Keywords: *Hydro-Thermal Unit Commitment, Ramp Limits, Lagrangian Relaxation.*

1 Introduction

The short-term Unit Commitment (UC) problem in hydro-thermal power generation systems requires to optimally operate a set of hydro—possibly cascade connected—and thermal generating units, over a given time horizon (typically one day or one week), in order to satisfy a forecasted energy demand at minimum total cost. The generating units are subject to some technical restrictions, depending on their type and characteristics; for hydro units typical constraints concern the discharge rate, spillage limits, reservoir storage and effect on downstream units. As for the thermal units, they must usually satisfy minimum up- and down-time constraints and upper and lower bounds over the produced power when the unit is operational, besides having complex power production and start-up costs. Closely representing the actual operating behavior of generating units within mathematical optimization models is crucial for being able to effectively coordinate the production of the generating system taking into account each unit’s characteristics [27], which is of increasing importance in the ongoing liberalization of the electricity market in many countries [19]. Indeed, while the (UC) problem in the form treated in this paper originated from the era of monopolistic producers, it has numerous applications even in the liberalized regime; furthermore, algorithmic approaches developed for the “classical” (UC) problem can usually be easily extended to forms of the problem arising in a market environment [19, 9].

Despite having attracted the interest of researchers for over 30 years, (UC) still cannot be considered a well-solved problem for all practical sizes and operating environments; this should not be surprising, since it is a large-scale, Mixed-Integer NonLinear Program (MINLP). Despite the ever-increasing availability of cheap computing power and the advances in off-the-shelf software for MINLP, solving (UC) by general-purpose software, even using the most advanced approaches available, is not feasible when the number of units [17] and/or the length of the time horizon [18] grows large. Recently, an interesting approach has been devised where the nonlinearities in the problem are approximated by means of piecewise-linear functions, so that (UC) can be approximated by a Mixed-Integer Linear Program (MILP), for which more efficient general-purpose solvers are available [10, 11, 24]. This approach may provide good results especially for

low- to mid-size instances, while specialized approaches are still required for very-large-scale instances, or when very fast running times are required by the operational environment; this is also confirmed by the results reported in §6 of the present paper.

Among the most efficient specialized algorithmic approaches for (UC), Lagrangian Relaxation (LR) methods [2, 6, 7, 8, 14, 23, 28] surely play a major role. These approaches exploit the *spatial structure* of the problem, that is, the fact that removing the constraints that tie the different units together, one obtains a set of disjoint Single-Unit Commitment (1UC) problems, requiring to optimally operate one single (hydro or thermal) unit over the time horizon. These problems are typically easily solvable by network flow techniques—for hydro units—or Dynamic Programming (DP) techniques—for thermal units.

However, the success of LR methods critically depend on two factors:

- being able to optimally solving the (1UC) problems efficiently;
- being able to turn the infeasible solution(s) produced at each step of the approach into “good”, feasible ones.

In turn, both these depend on the specific details of the operational constraints of the generating units that are represented in the mathematical model; each time that new constraints have to be included in the model, a (usually, nontrivial) modification is required to the solution method to (1UC) (e.g., [21, 22, 12]), and possibly of the *heuristic* used to produce feasible solutions. Thus, here appears the typical trade-off between the accuracy of the mathematical models employed and the efficiency of the corresponding solution process. For hydro units, for instance, it is customary to assume a linear relationship between the discharged water and the generated power, disregarding the nonlinearities corresponding to the variation of the water head and the minimum discharge necessary to keep the turbines operational; these assumptions are reasonable when dealing with short term operation, and dramatically simplify the hydro (1UC) problems. For thermal units, it is usually assumed that the dynamic of the generating plant does not pose restrictions (other than on maximum and minimum power levels) on the amount of power generated at each time period of the time horizon; unfortunately, this is not realistic for large units or if the time periods are to be taken small (e.g., 15 minutes), since then *ramp constraints* need to be considered. These limit the maximum increase or decrease of generated power from one time period to the next, reflecting the thermal and mechanical inertia that has to be overtaken in order for the unit to

increase or decrease its output. Unfortunately, ramp constraints complicate the structure of thermal (1UC) problems, rendering the classical DP approaches unusable; this is due to the fact that the variables representing the power output at different time periods are no longer independent once that commitment (on/off) decision has been taken. Four approaches have been proposed in the literature to overcome this difficulty:

- Discretizing the power variables space one may keep using a standard DP procedure [5, 4]; however, the computational burden increases as the granularity of the discretization is decreased, and the obtained solution is in general an approximated one.
- Approximating the cost function with a piecewise-linear (convex) function one may use the DP approach of [13], in turn based on the efficient algorithm the *Single-Unit Economic Dispatch* (1ED) problems with ramp constraints of [3, 26]; however, the computational burden increases with the accuracy of the approximation (the number of linear pieces in the cost function), and the obtained solution is in general an approximated one.
- Solving (1UC) problems only approximately, e.g., by using two-stage Lagrangian techniques as in [20], where ramp constraints are dualized and the corresponding Lagrangian multipliers updated (keeping those of the system balance constraints fixed) until a solution that is ramp-feasible is attained; however, the computational burden increases as multiple solutions of each (1UC) are required for each LR iteration, and the obtained solution is in general an approximated one.
- Using general-purpose, off-the-shelf Mixed-Integer Quadratic Programming (MIQP) solvers for solving (1UC) in alternative to DP approaches. This has the advantage of allowing incorporation of several other constraints in the formulation, e.g. related to the electrical market; since MIQP effective solvers have only recently become available, this approach has usually been coupled to approximation of the cost function by a piecewise-linear (convex) function, thus making (1UC) solvable by a more common MILP solver [1]. However, the solution time for general-purpose solvers (as it should be expected) grows very rapidly with the size of the instance, making this approach unusable for longer time horizons [18], even if a state-of-the-art MIQP solver is used.

Recently, a DP algorithm for thermal (1UC) with ramping constraints has been proposed [18] that can solve to optimality problems on a time horizon

of n time periods in $O(n^3)$ for convex cost functions whose unconstrained minimum can be found in $O(1)$, such as the quadratic ones typically used in operational settings. The algorithm solves thermal (1UC) problems with a very general definition of ramping constraints and complex, time-dependent startup costs. In this paper, we report on the use of that algorithm within LR approaches to (UC). We show that the increased complexity of the (1UC) subproblems—w.r.t. $O(n^2)$ of the case without ramp constraints—does not represent a serious computational bottleneck for this class of approaches. Also, we show that existing LR approaches that duly use the primal information provided by the Lagrangian Dual (e.g. [14, 8]) are fairly “resistant” to the introduction of ramp constraints, once these are properly taken into account in the Lagrangian subproblem. This is not obvious, since the heuristic procedures typically used to recover a primal feasible solution are not easily extended to take into account ramp constraints. Yet, solving ramp-constrained (1UC) subproblems turns out to be sufficient to provide the LR heuristic enough guidance to produce good feasible solutions even with no other substantial modification of the approach, due to the fact that the primal information computed by the Lagrangian Dual always remains ramp-feasible. To the best of our knowledge, this is the first LR approach that has been shown capable of solving—very large-scale—(UC) problems with ramp constraints with exact solution of the (1UC) subproblems and without any form of approximation of either the power production levels or the objective function. We show on a set of realistic thermal and hydro-thermal instances that the proposed approach consistently obtains much better solutions than a comparable approach solving non-ramp-constrained (1UC) problems. Also, we show that good quality solutions are obtained in a small fraction of the time required for the same task by state-of-the-art, general-purpose MIQP technology. Using a MILP (approximate) formulation within a general-purpose MIQP solver turns out to be, for a somewhat surprising reason, much more efficient than using the MIQP formulation on exactly the same solver; however, the LR approach still looks competitive for large-scale instances, especially hydro-thermal ones.

2 The UC model

Consider a set P of thermal units and a set H of hydro cascades, each comprising one or more basin units, and let $\mathcal{T} = \{1, \dots, n\}$ be the set of time periods defining the time horizon (the time period “0” will be used for indicating the initial conditions of the power system). Introducing status

and power production variables of the thermal units, u_t^i and p_t^i , respectively, with $i \in P$, $t \in \mathcal{T}$, the objective function of (UC), representing the total power production cost to be minimized, has the form

$$\sum_{i \in P} c^i(\mathbf{p}^i, \mathbf{u}^i) = \sum_{i \in P} \left(c^i(\mathbf{u}^i) + \sum_{t \in \mathcal{T}} c_t^i(p_t^i) \right) ; \quad (1)$$

that is, while the power production cost at each hour is typically approximated via a (convex) quadratic separable form (neglecting for instance the so called valve points, see e.g. [27]) in the p_t^i variables. In this paper, we allow the cost function to be nonseparable per hour due to time-dependent *start-up costs*, whose exact form has no impact on the proposed approach and is not reported here for the sake of notational simplicity; the interested reader can refer to [18] for a detailed discussion of the form of start-up cost which can be accommodated by the proposed algorithmic framework. The constraints of (UC) can be partitioned into three sets: local constraints for thermal units, local constraints for hydro units, and global (system wide) constraints.

- *Local constraints for thermal units*: for each thermal unit $i \in P$, let τ_+^i and τ_-^i be respectively the minimum up- and down-time requirements, Δ_+^i and Δ_-^i be respectively the maximum ramp-up and ramp-down rates, \bar{p}_{min}^i and \bar{p}_{max}^i be respectively the maximum and minimum power output of unit when operating in steady state, and \bar{l}^i and \bar{u}^i be the maximum power that can be produced by the unit in the time period where it is committed or decommitted, respectively. Then, the local constraints corresponding to each unit $i \in P$ are

$$\bar{p}_{min}^i u_t^i \leq p_t^i \leq \bar{p}_{max}^i u_t^i \quad t \in \mathcal{T} \quad (2)$$

$$p_t^i \leq p_{t-1}^i + u_{t-1}^i \Delta_+^i + (1 - u_{t-1}^i) \bar{l}^i \quad t \in \mathcal{T} \quad (3)$$

$$p_{t-1}^i \leq p_t^i + u_t^i \Delta_-^i + (1 - u_t^i) \bar{u}^i \quad t \in \mathcal{T} \quad (4)$$

$$u_t^i \geq u_r^i - u_{r-1}^i \quad t \in \mathcal{T}, r \in [t - \tau_+^i, t - 1] \quad (5)$$

$$u_t^i \geq 1 - u_{r-1}^i - u_r^i \quad t \in \mathcal{T}, r \in [t - \tau_-^i, t - 1] \quad (6)$$

$$u_t^i \in \{0, 1\} \quad t \in \mathcal{T} \quad (7)$$

Note that we assume knowledge of the *complete state* of each unit prior to the beginning of the current operation, that is, its commitment u_0^i , the generated power p_0^i ; for the sake of minimum up- and down-time constraints (5), (6), it is also necessary to know how long it has been on or off at time period 0.

- *Local constraints for hydro cascade units:* each cascade $h \in H$ is composed by a set $H(h)$ (possibly containing only one element) of individual hydro units; for each $j \in H(h)$, variables q_t^j , v_t^j and w_t^j represent respectively discharged water, the volume of the reservoir and the spilled water at time period $t \in \mathcal{T}$. Constants \bar{v}_{min}^j and \bar{v}_{max}^j represents respectively the minimum and maximum volume for the reservoir, \bar{q}_{max}^j represents the technical maximum of discharged water (the technical minimum is assumed to be zero in order to avoid nonlinearities in the model), while \bar{w}_t^j represents the natural inflows at time period $t \in \mathcal{T}$. Finally, let $\mathcal{S}(j)$ be the (possibly empty) set of the immediate predecessors of unit j —those whose discharge and spillage reaches j without passing through other reservoirs—and t_{kj} be the water time delay from plant $k \in \mathcal{S}(j)$ to the basin feeding plant j . Then, the local constraints corresponding to each unit $j \in H(h)$ are:

$$0 \leq q_t^j \leq \bar{q}_{max}^j \quad t \in \mathcal{T} \quad (8)$$

$$\bar{v}_{min}^j \leq v_t^j \leq \bar{v}_{max}^j \quad t \in \mathcal{T} \quad (9)$$

$$v_t^j - v_{t-1}^j = \bar{w}_t^j - w_t^j - q_t^j + \sum_{k \in \mathcal{S}(j)} \left(q_{t-t_{kj}}^k + w_{t-t_{kj}}^k \right) \quad t \in \mathcal{T} \quad (10)$$

Note that, in order for the balance equations (10) to be well-defined, we assume knowledge of the volume of each reservoir at time period $t = 0$, as well as water discharged and spilled at all time periods prior to $t = 1$ for which the water is still arriving to one of the downstream basins (i.e., those $k \in \mathcal{S}(j)$ such that $t < t_{kj}$).

- *Global constraints:* for each time period $t \in \mathcal{T}$, let \bar{d}_t be the forecasted load to be satisfied, and for each hydro unit j let α^j be the power-to-discharged-water efficiency (assumed constant, again to avoid nonlinearities); then, the system-wide constraints—linking the different units among themselves—are:

$$\sum_{i \in P} p_t^i + \sum_{h \in H} \sum_{j \in H(h)} \alpha^j q_t^j = \bar{d}_t \quad t \in \mathcal{T} \quad (11)$$

We refer to (UC) as the problem of minimizing (1) subject to constraints (2)—(11). This is a large-scale, mixed-integer nonlinear program; however, it should be noted that when the u_t^i variables are fixed, the remaining *Economic Dispatch* (ED) problem is a linearly constrained convex problem, hence easily solvable. For future reference, we will denote the set defined

by constraints (2)—(7) for a given thermal unit $i \in P$ as \mathcal{U}^i , and the set defined by constraints (8)—(10) for a given hydro cascade $h \in H$ by \mathcal{H}^h .

We remark that several of the (widely accepted) simplifying assumptions in the above model can be relaxed without hindering the applicability of the proposed approach; in particular, more sophisticated models of hydro cascades, e.g., taking into account nonlinear effects of the water head on the power-to-discharged-water efficiency and/or nonzero technical minima for discharged water, could be used at the cost of more difficult hydro subproblems and (ED)s. Conversely, thermal units are modeled in a fairly sophisticated way, and the approach can be easily extended (cf. [18]) to handle issues such as:

- Data dependent on the time period t , i.e., different ramp limits Δ_{t+}^i and Δ_{t-}^i or start-up and shut-down power output \bar{l}_t^i and \bar{u}_t^i for each $t \in \mathcal{T}$, modeling, e.g., the reaction to varying operating parameters such as the external temperature.
- Data dependent on the history of the unit, i.e., different ramp limits, start-up and shut-down power output, maximum and minimum power levels, and even (coefficients of the) cost function according to *how long the unit has remained consecutively committed* before the current time period.
- Different discretization intervals for commitment and power variables, e.g., power production levels to be specified for each quarter of hour of the time horizon whereas the commitment decisions can only be changed hourly.

Thus, the UC model here considered is well-suited for being employed in a free market regime, both at the stage where GenCos need to optimize their production schedule once that their own load profile has been established by the market procedures, and within approaches for computing optimal bidding strategies [19, 9].

3 The Lagrangian relaxation

The LR approach is based on dualizing each of the coupling constraint (11) via a Lagrangian multiplier λ_t , thereby forming the *Lagrangian Relaxation* of (UC)

$$\mathcal{L}(\boldsymbol{\lambda}) = \sum_{i \in P} \phi_i^1(\boldsymbol{\lambda}) + \sum_{h \in H} \phi_h^2(\boldsymbol{\lambda}) + \sum_{t \in \mathcal{T}} \lambda_t \bar{d}_t \quad (12)$$

where

$$\begin{aligned}\phi_i^1(\boldsymbol{\lambda}) &= \min \{ c^i(\mathbf{p}^i, \mathbf{u}^i) - \boldsymbol{\lambda} \mathbf{p}^i : (\mathbf{p}^i, \mathbf{u}^i) \in \mathcal{U}^i \} \\ \phi_h^2(\boldsymbol{\lambda}) &= \min \left\{ -\boldsymbol{\lambda} \sum_{j \in H(h)} \alpha^j \mathbf{q}^j : [\mathbf{q}^j]_{j \in H(h)} \in \mathcal{H}^h \right\}\end{aligned}$$

It is well known (e.g., [16]) that for each $\boldsymbol{\lambda} \in \mathbb{R}^n$, $\mathcal{L}(\boldsymbol{\lambda})$ gives a lower bound on the optimal value of (UC). Therefore, one is interested in the $\boldsymbol{\lambda}^*$ such that this lower bound is the best (maximum), i.e., in the optimal solution of the *Lagrangian Dual* of (UC):

$$\max \{ \mathcal{L}(\boldsymbol{\lambda}) : \boldsymbol{\lambda} \in \mathbb{R}^n \} \quad . \quad (13)$$

Since $\mathcal{L}(\cdot)$ is a non differentiable function, proper algorithms must be chosen for solving (13); *bundle* methods [15], particularly in their *disaggregated* variant [2], have been repeatedly reported to be quite efficient in solving (13), much more so [8] than alternative algorithms such as subgradient methods [4, 28].

However, solving (13) is not, in general, enough to solve (UC); even for $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$, the optimal solution to (12) is not guaranteed to—and will not in general—satisfy the relaxed constraints (11). Therefore, a number of *Lagrangian heuristics* have been devised that attempt to convert the (sequence of) infeasible solution(s) provided by (13) in a (sequence of) feasible, hopefully “good”, one(s). Roughly speaking, there are two possible ways for doing this:

- either combinatorial heuristics are run at every step of the iterative solution of (13), thus for several different values of $\boldsymbol{\lambda}$, that use the infeasible optimal solution of the corresponding (12) to produce a feasible solution for (UC);
- or the Lagrangian problem is modified with further terms that try to enforce feasibility of the obtained solutions.

Most approaches from the literature belong to the first group; it is outside the scope of the present paper to describe them in detail, the interested reader being referred, e.g., to [2, 4, 8, 7, 14, 28]. These approaches suffer from the drawback that the combinatorial heuristic ran at each iteration is dependent on the exact details of the model to be solved, and may be difficult to adapt to different situations; this motivated the development of the second group of approaches, where a further quadratic term (either an Augmented Lagrangian one [6] or a proximal one [12]) is added to the problem to try to

enforce feasibility of the obtained solutions. Since the quadratic term may make the Lagrangian relaxation harder to solve, approximate solution via sequential linearization may have to be employed.

It is relevant for the present context that all the most recent and efficient approaches, both of the first [8, 14] and of the second [6, 12] type, make use of the *continuous* primal information generated by the solution of (13). In fact, it is well-known that the Lagrangian Dual of a problem with convex objective function and nonconvex feasible set is equivalent to the *convexified relaxation* of the problem where the original objective function is optimized over the intersection of the convex hull of the (integer) solutions of the LR and the relaxed constraints. Most importantly, this equivalence is *computational* in the sense that basically all algorithms for solving the Lagrangian Dual either naturally compute, or can be modified to do so, a solution to the convexified relaxation [16]. In particular, bundle methods applied to (13) compute at each iteration a continuous solution that satisfies all constraints (2)—(9), and that pretty rapidly becomes almost feasible w.r.t. the relaxed constraints (11). This continuous, almost feasible solution is very useful to drive the heuristic search for integer, entirely feasible solutions to (UC), as described in more details in §5.

The LR approach has proven to be capable of solving non ramp-constrained large-scale (UC) problems with a provable high degree of accuracy in relatively short time. However, extending the approach to ramp constraints requires, in principle, three steps:

- i) solving (possibly approximately) ramp-constrained (1UC) subproblems;
- ii) modifying the heuristics in order to take into account the ramping constraints while fixing the commitment u_t^i variables;
- iii) solving ramp-constrained Economic Dispatch problems at each iteration to recover the p_t^i and q_t^j variables once that the u_t^i variables have been (heuristically) fixed.

Of those, (i) has only been made possible—without approximations or an excessive growth of the computational burden—by the development in [18], while (iii) is trivial in that it only requires to add some linear constraints to the classical (ED) formulation. As for the point (ii), the rationale behind the “augmented” approaches [6, 12] was exactly to provide a general substitute for specialized combinatorial heuristics in order to avoid the problem.

The aim of this paper is to show that point (ii) is—at least for ramp constraints, but possibly also in other cases—much less of a problem than

often perceived, that is, the combinatorial heuristics originally developed for (UC) without ramp constraints are efficient in the ramp-constrained case also, only provided that ramp-constrained (1UC) (and, obviously, ramp-constrained (ED)s) are solved. This should not be too surprising considering that, when a ramp-constrained (12) is solved, the latest combinatorial heuristics construct the feasible solutions using, as starting points, two solutions—an integer infeasible one and the continuous almost feasible one—that are both ramp-feasible. Hence, ramp-constrained (1UC) provide—through the ramp-feasibility of the primal solutions, as well as through better dual multipliers—the standard LR heuristic with enough guidance to produce good feasible solutions even with no other modification of the approach.

4 Solving the (1UC) problem

In this section we briefly sketch the algorithm of [18] for solving the ramp-constrained (1UC) problem. We assume that each term c_t^i in (1) is a “simple” convex function, in the sense that unconstrained minimization of c_t^i can be carried upon in $O(1)$. Then, the overall cost function c^i of (1) is the sum of c_t^i plus a start-up cost function that only depends (but possibly in sophisticated ways) on how long the unit has been turned off. Because we are only interested in one unit at a time, for notational simplicity in the remainder of the section we will drop the superscript i .

The approach of [18] is a DP procedure, that is, it requires the computation of a shortest path in a suitable acyclic directed *state space* graph. Following an idea originally proposed in [13], the state space graph comprises a node for all pairs (h, k) for $h, k \in \mathcal{T}$ and $k \geq h$, plus a source s and a sink d . Each state (h, k) represents the unit being turned on at time period h (i.e., being uncommitted at time period $h - 1$), and being turned off again at time period k ; clearly, all states that correspond to operations violating the minimum up-time requirements are not constructed. The graph has arcs between any node (h, k) and (r, q) such that it is feasible to turn on the unit at time instant r given that it has been turned off at time instant k ; each of these arcs are labeled with the start-up cost of the unit at time instant r . There are also arcs from the source s to all nodes (h, k) compatible with the initial state of the unit; finally, there is a zero-cost arc from each node to the sink d . Clearly, every $s - d$ path on this graph represents a feasible solution to (1UC), arc costs representing the contribution of start-up costs to the objective function (fixed generating costs, if any, can also be easily

included). It is now necessary to compute, for each node (h, k) , the optimal contribution of the variable generating costs, that depend on the p_t variables. This requires the solution of the *Economic Dispatch with ramp constraints* problem for the interval $[h, k]$, that is, the minimization of (the continuous part of) (1) subject to (2)—(7) if all u_t variables for $t \notin [h, k]$ are fixed to zero and all those for $t \in [h, k]$ are fixed to one; we will denote this problem as (ED_{hk}) . Since all the binary variables are fixed, this is an optimization problem with convex objective function and linear constraints, hence easily solvable. By assigning its optimal objective function value $z_{hk}^* = z(ED_{hk})$ to the corresponding node, the (node plus arc) cost of each $s - d$ path on the graph is that of the feasible solution it represents. Thus, (1UC) can be solved with a visit of a graph with $O(n^2)$ nodes and $O(n^4)$ arcs, once that all the data has been computed; exploiting some structural properties of the state-space graph, the complexity of the visit can be reduced to $O(n^3)$ [18].

However, computing the node costs requires the solution of $O(n^2)$ convex problems, each with up to n variables. Hence, solving (ED_{hk}) efficiently—or, more to the point, solving *all* the $O(n^2)$ of them efficiently—is crucial. Indeed, the main contribution of [18] is the proposal of an efficient dynamic programming algorithm that can solve all $O(n^2)$ Economic Dispatch problems in $O(n^3)$ for cost functions whose unconstrained minimization is $O(1)$, i.e., given by a closed formula, such as the quadratic ones. The algorithm is based on the fact that, defining $z_{hk}(\bar{p})$ to be the optimal solution value of (ED_{hk}) if the further constraint $p_k = \bar{p}$ is imposed, one has a constructive way to compute $z_{h(k+1)}$ given z_{hk} . In particular, one can prove [18] the following fact:

Proposition 1 *The function z_{hk} is finite-valued only in $v + 1$ intervals $[m_0, m_1], [m_1, m_2] \dots [m_v, m_{v+1}]$, with $v \leq 2(k - h)$, in which*

$$z_{hk}(\bar{p}) = z^j(\bar{p}) \quad \text{if } \bar{p} \in [m_i, m_{i+1}] \text{ for } i = 0, \dots, v$$

where each function z^j is the sum of at most $k - h + 1$ functions c_t for $t \in \{h, h + 1, \dots, k\}$.

The proof of the above proposition is constructive, and provides an algorithm for explicitly computing $z_{h(k+1)}$ given the piecewise representation of z_{hk} (the case z_{hh} is trivial). The overall complexity of the procedure depends on the actual form of the functions c_t ; for quadratic functions, all the problems $(ED_{hh}), (ED_{h(h+1)}), \dots, (ED_{hk})$ can be solved in $O((k - h)^2)$, hence solving all the $O(n^2)$ (ED) problems is $O(n^3)$. Note that this complexity is only for computing the optimal objective function values; however,

the optimal solutions of all the (ED_{hk}) problems corresponding to all nodes in the optimal path are also needed. Those solutions are easily found with a “backward pass”, using the available information constructed in the “forward pass”: the (both primal and dual) optimal solution of any (ED_{hk}) can be found in $O(k-h)$, so computing the optimal solution to (1UC), in terms of the power variables, once the “upper” dynamic programming procedure is computed, is $O(n)$. Thus, with the approach of [18] one can solve (1UC), for the quadratic case, in $O(n^3)$ overall.

5 The combinatorial heuristic

Here we briefly recall the combinatorial heuristic, proposed in [8], which is ran at each step of the iterative method used to solve (13). The heuristic uses as starting points the current value of the Lagrangian multipliers λ , the corresponding optimal solution $\bar{s} = [\bar{\mathbf{p}}, \bar{\mathbf{u}}, \bar{\mathbf{q}}]$ of (12)—in which $\bar{\mathbf{u}}$ is integral and constraints (2)—(10) are satisfied, but (11) are *not*—and the “convexified” solution $\tilde{s} = [\tilde{\mathbf{p}}, \tilde{\mathbf{u}}, \tilde{\mathbf{q}}]$ produced by the bundle approach, where $\tilde{\mathbf{u}}$ is not integral, constraints (2)—(10) are satisfied and constraints (11) are typically almost satisfied. Then, the following three steps are performed:

- i) the “convexified” hydro power production $\tilde{\mathbf{q}}$ is considered as fixed, and the total power demand is decreased accordingly;
- ii) a *greedy-type* heuristic is used to set a commitment status $\hat{\mathbf{u}}$ of the thermal units in order to try to guarantee that the remaining power demand can be satisfied;
- iii) finally, the actual power production $[\hat{\mathbf{p}}, \hat{\mathbf{q}}]$ of thermal and hydro units is determined by solving the *Economic Dispatch* problem (1)—(11) for the fixed value $\hat{\mathbf{u}}$ of the commitment variables; note that (ED) is large-scale convex quadratic program, hence theoretically “easy” but still relatively costly to solve in practice.

This heuristic is motivated by the fact that adjusting the commitment status of thermal units is relatively simple because the commitment decision at time t directly impacts only commitment decisions in a small set of time instants centered on t , depending on the minimum up- and down-time constraints, while changing the power output of some hydro units at a certain time instant potentially impacts the hydro power output of the units in all the time horizon. However, once the combinatorial decisions have been taken, the remaining continuous problem (which, however, is not guaranteed

to have a feasible solution) is “easy”. In particular, the greedy heuristic at step (ii) checks for each time period t whether the residual demand

$$\tilde{d}_t = \bar{d}_t - \sum_{h \in H} \sum_{j \in H(h)} \alpha^j \hat{q}_t^j$$

can be satisfied by the active thermal units in the integral solution $\bar{\mathbf{u}}$ by simply checking that it belongs to the range $[\bar{u}_t^-, \bar{u}_t^+]$, where

$$\bar{u}_t^- = \sum_{i \in P} \bar{p}_{min}^i \bar{u}_t^i \quad \bar{u}_t^+ = \sum_{i \in P} \bar{p}_{max}^i \bar{u}_t^i .$$

If $\tilde{d}_t > \bar{u}_t^+$, then the time period t is said *undercommitted*, while if $\tilde{d}_t < \bar{u}_t^-$ it is said *overcommitted*; in either case, the solution $\bar{\mathbf{u}}$ has to be modified by turning some units on or off at t . For this purpose, a priority list of units is formed to decide which ones are more “promising” at any given time instant; the list is based on a combination of the Lagrangian cost of turning on the unit and on the “convexified” commitment status \tilde{u}_t^i of the unit, interpreted as a “probability” that the unit i should be on at time period t in the optimal solution.

We remark that this is only one of the several possible implementations of the algorithmic scheme; several others are presented and compared in [8], showing that the one just described is usually the best. Furthermore, the heuristic has been developed for the non-ramp-constrained case: the definition of \bar{u}_t^- and \bar{u}_t^+ does *not* take into account the ramping constraints, and therefore may trick the heuristic into concluding that a time period is “feasible” while in actuality it is not because, due to ramping, the maximum (minimum) amount of power that can actually be produced in t , given the chosen commitment, is smaller (larger) than \bar{u}_t^+ (\bar{u}_t^-).

It would be possible (although nontrivial) to try to adapt the heuristic to take into account somewhat ramp rates. Furthermore, one may go the route of [6, 12] and avoid the heuristic altogether; finally, the two approaches may be combined. However, the purpose of this paper is to show that the information about ramp rates “embedded” into \bar{s} and \tilde{s} is enough to provide the heuristic, with no modification, enough guidance to produce (good quality) feasible solutions.

6 Computational Experiences

In this section we present some numerical results aimed at showing the efficiency and the effectiveness of the proposed approach. Our algorithm

has been coded within a C++ commercial code, **PowerSched**® . All tasks are implemented by custom-written routines except the solution of the ED—a large-scale, linearly constrained convex quadratic program—in step (iii) of the heuristic, for which the commercial solver **Mosek** was used. All the results have been obtained on an Opteron 246 (2 GHz) computer with 2 GigaBytes of RAM, running Linux Fedora Core 3.

For our tests, we have randomly generated several sets of realistic pure thermal and hydro-thermal instances, with number of thermal units ranging from 20 to 200 and number of hydro units ranging from 10 to 100, on a daily problem ($n = 24$). The generator produces a generating set with “small”, “medium” and “large” thermal units in realistic proportions; the characteristics of each unit are then randomly generated within a set of realistic parameters, depending on the type of the unit. Ramping restrictions are also randomly generated within realistic measures, resulting in large units to require between two and three hours to ramp from the technical minimum to the technical maximum. All the instances have *time-invariant start-up costs*; this is a “worst case” situation for the new approach, in that the new DP procedure requires $O(n^3)$ regardless to the fact that start-up costs are time-dependent or time-invariant, while the “classical” DP is $O(n^2)$ in the former case, but only $O(n)$ in the latter. Furthermore, introducing time-dependent startup costs in the MIQP formulation increases the number of continuous variables and constraints (although not the number of binary variables [25]) in the model, thereby making it more difficult to solve, while—apart from the possible effect on the DP procedure—time-dependent startup costs are basically handled for free in the LR approach. The (UC) instances are freely available at

<http://www.di.unipi.it/optimize/Data>

6.1 Impact of the new DP procedure on the LR approach

We first analyze the impact of the newly proposed DP procedure by comparing two versions of the LR approach, identical except for the fact that one uses the ramp-constrained DP of [18] and the other uses the “classical” DP disregarding ramps.

The results are displayed in Table 1. In the Table, column “ p ” reports the total number of thermal generating units, while column “ h ” reports the total number of hydro units. The first half of the table, with $h = 0$, is therefore composed by “pure thermal” instances; each row reports averaged results of 5 instances of the same size. Columns “RCDP” report results for the

LR approach using the ramp-constrained DP, while columns “UDP” report results for the LR approach using the standard unconstrained DP. In both cases, column “time” reports the required running time (in seconds), column “iter” reports the total number of iterations of the bundle method used to solve (13)—that is, the number of times that the heuristic is attempted—, column “sol” reports the total number of feasible solutions found and column “gap” reports the obtained gap (in percentage)

$$\frac{\text{best feasible solution} - \text{best lower bound}}{\text{best lower bound}} \times 100;$$

the number in parenthesis next to the gap, if any, is the number of instances in that group for which no feasible solution at all has been found, so that the reported gap is the average among those for which at least a solution was found. Note that we have used the best overall lower bound for computing the gap, as opposed to the best lower bound produced by each single algorithm, so that any difference in gaps is only due to the quality of the obtained feasible solutions. For UDP, column “ Δlb ” reports the gap (in percentage) between the obtained lower bound and that of RCDP; the latter is always higher because the Lagrangian Relaxation is “more constrained”, and the Lagrangian Dual is solved to optimality.

p	h	RCDP				UDP				
		time	iter	sol	gap	time	iter	sol	gap	Δlb
20	0	8	189	34	0.44	6	202	1	11.30(3)	2.49
50	0	17	195	33	0.26	16	247	1	5.25 (3)	1.48
75	0	30	206	33	0.38	22	278	1	9.25	2.38
100	0	46	213	21	0.48	29	285	1	8.69	2.21
150	0	72	277	23	0.20	54	341	1	7.66	2.31
200	0	134	317	67	0.06	78	369	1	8.53	2.46
20	10	16	162	159	0.22	7	206	3	3.80	1.50
50	20	41	165	146	0.07	16	231	6	0.63	1.19
75	35	89	209	166	0.02	28	274	5	1.73	1.19
100	50	135	218	143	0.04	38	301	1	1.86	1.27
150	75	222	223	164	0.01	71	318	1	4.10 (1)	1.20
200	100	353	244	192	0.05	90	305	2	4.38	1.25

Table 1: Comparison of the two different DP approaches

It is clear from the table that RCDP is much more effective than UDP. On pure thermal instances it obtains much smaller gaps—always at least one order of magnitude smaller, and up to three orders of magnitude smaller in the largest instances—and it is always capable of finding at least a feasible

solution, while UDP fails to solve several of the smallest instances. This is clearly explained by looking at column “sol”: while RCDP finds feasible solutions at least once in roughly ten dual iterations, UDP only finds on average *one* solution during all the course of the algorithm. This is not surprising, since the heuristic in UDP has absolutely no information about ramp constraints, that only come up in the solution of the ED problem, when the critical (binary) decisions have already been taken; on the contrary, RCDP benefits from much more accurate information coming from the ramp-constrained Lagrangian subproblems. This also explains why UDP is faster (though never more than a factor of two): it is not due to the extra cost of the DP procedure, that is negligible in the context of the overall approach, but rather to the fact that UDP solves a much smaller number of ED problems, which are rather costly. Clearly, the extra time required by RCDP is very well-spent in this case, especially considering the much higher robustness of the approach; the fact that RCDP also improves the lower bound, most often to the tune of more than 2%, is also a nice extra effect of using the more sophisticated DP procedure.

The results of hydro-thermal instances confirm those of the pure thermal case. Hydro-thermal instances are easier to solve for LR approaches, since the flexibility of hydro units allow the system to react better to changes of the demand, possibly without requiring (too many) thermal units to switch state. This is testified by the much smaller gaps, by the much larger fraction of dual iterations that actually produce a feasible solution, and by the fact that—barring an occasional (150, 75) instance—even UDP consistently finds feasible solutions. However, the gaps obtained by RCDP are in this case *most often* better than those of UDP by three orders of magnitude, and (with a comparable number of dual iterations) the former finds feasible solutions in the hundredths, while the latter only finds a few of them. Due to the much smaller number of ED solved, UDP is also roughly three times faster than RDCP; however, the latter never requires more than 6 minutes even for the largest instances, and the obtained solutions are of excellent quality.

6.2 Comparison with a general-purpose MIQP solver

We now present results comparing the LR using the ramp-constrained DP of [18] with an approach using state-of-the-art, general-purpose MIQP technology. For this, we simply passed the MIQP formulation (1)—(11) to the commercial solver `Cplex 9.1`.

The results are displayed in Table 2. For RCDP, the results of Table 1 are reported, albeit limited to the pure thermal cases for reasons to become

clear shortly. Columns “Cplex MIQP” reports results obtained by the MIQP solver Cplex 9.1. The code was ran with a time limit of one hour, which is already unrealistic in most production environments; however, none of the instances were solved to optimality within that time. Thus, we report in column “first” the (average) time required by the MIQP solver for finding any integer solution, in column “best” the time required for finding its best solution, and in column “eqv” the time required for finding a solution with equivalent or better quality than the best one found by RCDP (counting one hour if this never happens). Finally, columns “gap” and “ Δ lb” have the same meaning as for UDP in Table 1.

p	RCDP				Cplex MIQP				
	time	iter	sol	gap	first	best	eqv	gap	Δ lb
20	8	189	34	0.44	24	2229	858	0.29	-0.14
50	17	195	33	0.26	249	1491	2563	0.22	0.34
75	30	206	33	0.38	447	1514	2195	0.10	0.44
100	46	213	21	0.48	940	2327	1236	0.13	0.47
150	72	277	23	0.20	2348	2483	3280	0.24(1)	0.42
200	134	317	67	0.06	3600	3600	3600	* (5)	0.44

Table 2: Comparing the LR and a MIQP solver

The Table shows that, given enough time, the MIQP approach may deliver more accurate solutions than the LR approach on smaller-size instances. For very small instances ($p = 20$) this is true both for the lower bound and for the upper bound, while for instances with p up to 100 the lower bound is always dominated by the Lagrangian one, while the upper bound may be better. However, this comes at a hefty cost in terms of time, as Cplex takes anywhere between 25 and 150 times the running time of RCDP to deliver solutions of the same quality (even when it succeeds in doing this); we should also remark that RCDP may—and typically does—find its best solution way before the actual termination of the approach. Furthermore, the MIQP solver requires a long time just to find *any* feasible solution (possibly of bad quality), which is unacceptable in many operating environments. Finally, for $p > 100$ it may fail to find a feasible solution at all within an hour of running time; this *always* happens for the largest instances ($p = 200$), which are, incidentally, those that are better solved by the LR approach. We have therefore avoided to report the results of Cplex for hydro-thermal instances, since the much larger size of the formulation (for the same number of thermal units) make it even less efficient at solving these instances.

6.3 Comparison with a general-purpose MILP solver

During the first revision of this paper, an anonymous referee suggested us to test the approximated MILP formulation of [10]; this is obtained from (1)—(11) by replacing the quadratic terms in the objective function by a (convex) piecewise-linear approximation. While the number of pieces in the approximation is in general arbitrary, we used a four-piece approximation as suggested in [10] (the interested reader is referred to the original paper for details); this results in a MIP with roughly 4 times the number of continuous variables and more constraints than the original quadratic formulation, but without quadratic terms in the objective function.

We were initially doubtful about the possible effectiveness of this formulation, since we were going to solve it with the very same general-purpose solver, `Cplex 9.1`, thus most of the sophisticated techniques (preprocessing, branching rules, valid inequalities, ...) available to the MILP solver were also available in the MIQP case, too. Hence, the only fundamental difference between the two cases seemed to be whether a Quadratic Program (QP) or a Linear program (LP) was solved at each relaxation instead; since most often a QP can be solved in no more than three times the time it takes to solve an LP of the same size, we expected nothing more than an improvement of at most a factor of three in the running times, even less since the MILP formulation is significantly larger than the MIQP one. However we followed the advice of the referee, and the actual results, reported in Table 3 proved us blatantly wrong. Columns “RCDP” have the same meaning as in the previous paragraphs; the results are however different since, having been obtained over an year after the previous ones, a more advanced and efficient version of the code was used, with a significant part of the performance improvements being due to using a more recent version of `Mosek` (version 5 vs. version 3.2). Columns “`Cplex MILP`” reports results obtained by the MILP solver `Cplex 9.1`; as in [10], the code was instructed to stop as soon as a relative gap of 0.5% or less is attained, and it always managed to achieve this result. We could mention at this point that, being the piecewise-linear approximation of the objective function an *upper* estimate, the lower bounds computed by the MILP solver are not, in general, valid for the “true” problem, so the estimate of the relative gap is approximated. Columns “time” and “gap” have the same meaning as in the other cases, columns “ftime” and “gap” report respectively the time it takes to generate the first integer feasible solution (the column being empty if the time is very small, so that `Cplex` does not correctly report it) and its relative gap, column “nodes” reports the number of visited nodes in the enumeration tree and,

finally, column “LPs” reports the total number of LP solved; this is much larger than the number of nodes because `Cplex 9.1` employs a sophisticated “Branch & Cut” approach where valid inequalities are automatically derived and added to the formulation to improve the lower bound.

p	h	RCDP			Cplex MILP					
		time	gap	iter	time	gap	ftime	fgap	nodes	LPs
10	0	0.75	0.99	187	0.95	0.33		1.18	0	23
20	0	1.83	0.46	189	3.72	0.36		1.00	0	23
50	0	4.84	0.28	195	21.93	0.21	15.98	0.36	0	25
75	0	9.41	0.34	206	56.31	0.20	47.08	1.62	10	59
100	0	14.74	0.33	213	94.09	0.17	69.75	2.18	16	76
150	0	21.20	0.17	277	218.69	0.12	177.35	6.58	16	115
200	0	34.80	0.09	317	267.78	0.09	247.12	1.85	6	87
20	10	1.76	0.39	170	93.53	0.21		0.59	140	258
50	20	6.36	0.06	160	17.98	0.06	17.98	0.06	0	60
75	35	15.01	0.04	198	96.86	0.11	96.86	0.11	170	300
100	50	24.74	0.04	209	130.86	0.06	130.86	0.06	180	266
150	75	37.41	0.02	189	467.62	0.06	467.62	0.06	300	554
200	100	50.91	0.01	175	427.71	0.05	427.71	0.05	205	321

Table 3: Comparing the LR and a MILP solver

The table shows that the MILP approach is remarkably efficient in obtaining good quality solutions for UC instances, dramatically more so than the MIQP one. A close inspection of the solution logs revealed us that the difference mainly lies in the effectiveness of *the primal heuristic*, i.e., the (unknown) approach used within the `Cplex` solver to construct a feasible integer solution out of the continuous one produced by the relaxation. In fact, while in the MIQP case the solution, if at all produced, had at least initially abysmal gaps (to the tune of 7%), the very first solution produced by the MILP solver quite often, although not always, has a very small gap (much less than 1%); even when the first solution is not satisfactory, the MILP solver would typically find a very good one shortly thereafter.

The reasons of this dramatic difference in the effectiveness of the primal heuristics in the MIQP and MILP case is not very clear to us; perhaps it can be due to the fact that the continuous solution generated by the MIQP code is not, usually, a vertex of the feasible polyhedron, and therefore may be “more fractional” than the one generated by the MILP code. This may conceivably impact on the performances of the heuristic, which typically uses rounding techniques and therefore greatly benefits from having very many variables already set to integer values. Whatever the reason, the

MILP approach has proved capable to obtaining very good quality solutions to even the largest instances in times not exceeding 8 minutes.

On the pure thermal instances, the MILP code consistently outperforms the LR as far as the quality of the solutions is concerned; however, the gap between the two narrows as the size of the instances increase, and for the largest instances the two approaches provide solutions of comparable quality. Furthermore, the LR approach is always faster, with the speedup increasing as the size of the instances does and reaching one order of magnitude. Also, column “ftime” reveals that the first feasible solution is obtained very late, sometimes right before termination; thus, there does not seem to be a way to improve on these running times, not even by accepting the very first solution produced (which, as column “fgap” indicates, can be of very bad quality).

This trend is even clearer for hydro-thermal instances. There, for all instances with $p > 20$ the MILP code stops as soon as the first solution is obtained; unlike the pure thermal case, that solution invariably turns out to be of very good quality. However, unlike the pure thermal case the LR approach almost always provides solutions of even better quality than the MILP, and requires much less time to do so, with a speedup of up to a factor of 12 on the largest instances. So, while the MILP approach is highly competitive w.r.t. the LR one on small- to mid-size instances, especially pure thermal ones, when running time is not much of a concern, the LR approach still is the method of choice for very-large-scale instances, especially hydro-thermal ones, when a fast response time is required.

7 Conclusions and directions for future work

In this paper, we have proposed a Lagrangian Relaxation (LR) approach for solving large-scale hydro-thermal Unit Commitment (UC) problems with ramp constraints on the thermal generating units. The keys of the effectiveness of the approach are the efficient algorithm for Single-Unit Commitment (1UC) problems with ramp constraints recently proposed in [18], that exactly solves the Lagrangian subproblems without resorting to any form of approximation, and the sophisticated heuristics for producing an integer ramp-feasible and demand-feasible solution out of the two infeasible ones (the integer demand-infeasible and the continuous demand-almost-feasible) computed by the LR approach. Despite the fact that the heuristic has been developed for problems without ramping constraints, and does not take them into account, solving ramp-constrained subproblems turns out to be sufficient for solving with high provable accuracy very-large-scale realistic

instances in reasonable computational time on low-end hardware.

Our approach is competitive with using general-purpose MIP technology, especially Mixed-Integer Quadratic Program solvers. Using the approximated formulation of [10] turns out to provide consistent improvements, up to the point that the resulting MILP approach may be preferable to the LR one on small- to mid-size instances, especially pure thermal ones; however, the LR approach still is the method of choice for very-large-scale instances, especially hydro-thermal ones, when a fast response time is required. Furthermore, our LR approach could be improved in at least three ways:

- by improving the heuristic in order to take into account ramp constraints directly in the logic of the greedy approach, as opposed to only indirectly by the use of ramp-constrained starting solutions;
- by using an “augmented” approach a-la [6, 12] in alternative to, or in combination with, the combinatorial heuristic;
- by embedding the whole process in a Branch&Bound approach, like that of `Cplex`, in order to refine both the upper and the lower bound at the cost of a longer running time.

A different line of research involves increasing the accuracy of the employed mathematical models of hydro and thermal units to improve the real-life quality of the obtained solution, for instance by taking into account *reserve constraints*, i.e., reserving spare capacity on the active thermal units in order to be able to cope with the inherent uncertainty of the load and system failures. These constraints are necessary in a monopolistic context, and may be useful in a liberalized regime since extra available capacity is a valuable commodity that is normally traded in a separated market. Incorporating reserve constraints in the “classical” (UC) problem is usually not difficult; indeed, they are often considered in the literature [8]. However, extending our approach to reserve-constrained versions of (UC) is not straightforward, and it will be the subject of a future work.

Acknowledgment

The authors gratefully acknowledge the contributions of an anonymous referee for its numerous and useful suggestions, in particular the one about using of the approximated MILP formulation of [10], which have allowed us to greatly improve upon a previous version of the paper.

References

- [1] J.M. Arroyo and A.J. Conejo. Optimal Response of a Thermal Unit to an Electricity Spot Market. *IEEE Transactions on Power Systems*, 15(3):1098–1104, 2000.
- [2] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal. Bundle Methods in Stochastic Optimal Power Management: A Disaggregated Approach Using Preconditioners. *Computational Optimization and Applications*, 20:227–244, 2001.
- [3] C.H. Bannister and R.J. Kaye. A Rapid Method for Optimization of Linear Systems With Storage. *Operations Research*, 39(2):220–232, 1991.
- [4] J.F. Bard. Short-term Scheduling of Thermal-electric Generators using Lagrangian Relaxation. *Operations Research*, 36(5):765–766, 1988.
- [5] T.E. Bechert and H.G. Kwatny. On the Optimal Dynamic Dispatch of Real Power. *IEEE Transactions on Power Apparatus and Systems*, PAS-91(1):889–898, 1972.
- [6] A. Belloni, A. Diniz, M.E. Maceira, and C. Sagastizábal. Bundle Relaxation and Primal Recovery in Unit Commitment Problems. The Brazilian Case. *Annals of Operations Research*, 120:21–44, 2003.
- [7] A. Borghetti, A. Frangioni, F. Lacalandra, A. Lodi, S. Martello, C.A. Nucci, and A. Trebbi. Lagrangian Relaxation and Tabu Search Approaches for the Unit Commitment Problem. In Saraiva, J.T. and Matos, M.A., editors, *Proceedings IEEE 2001 Powertech Porto Conference*, volume Paper n. PSO5-397, 2001.
- [8] A. Borghetti, A. Frangioni, F. Lacalandra, and C.A. Nucci. Lagrangian Heuristics Based on Disaggregated Bundle Methods for Hydrothermal Unit Commitment. *IEEE Transactions on Power Systems*, 18:313–323, 2003.
- [9] A. Borghetti, A. Frangioni, F. Lacalandra, C.A. Nucci, and P. Pelacchi. Using of a Cost-based Unit Commitment Algorithm to Assist Bidding Strategy Decisions. In Borghetti, A., Nucci, C.A., and Paolone, M., editors, *Proceedings IEEE 2003 Powertech Bologna Conference*, volume Paper n. 547, 2003.

- [10] M. Carrión and J.M. Arroyo. A Computationally Efficient Mixed-integer Linear Formulation for the Thermal Unit Commitment Problem. *IEEE Transactions on Power Systems*, 21(3):1371–1378, 2006.
- [11] G.W. Chang, M. Aganagic, J.G. Waight, J. Medina, T. Burton, S. Reeves, and M. Christoforidis. Experiences with Mixed Integer Linear Programming Based Approaches on Short-term Hydro Scheduling. *IEEE Transactions on Power Systems*, 16(4):743–749, 2001.
- [12] L. Dubost, R. Gonzalez, and C. Lemaréchal. A Primal-proximal Heuristic Applied to the French Unit Commitment Problem. *Mathematical Programming*, 104(1):129–151, 2005.
- [13] W. Fan, X. Guan, and Q. Zhai. A New Method for Unit Commitment with Ramping Constraints. *Electric Power Systems Research*, 62:215–224, 2002.
- [14] S. Feltenmark and K.C. Kiwiel. Dual Applications of Proximal Bundle Methods, Including Lagrangian Relaxation of NonConvex Problems. *SIAM Journal on Optimization*, 10(3):697–721, 2000.
- [15] A. Frangioni. Generalized Bundle Methods. *SIAM Journal on Optimization*, 13(1):117–156, 2002.
- [16] A. Frangioni. About Lagrangian Methods in Integer Optimization. *Annals of Operations Research*, 139:163–193, 2005.
- [17] A. Frangioni and C. Gentile. Perspective Cuts for a Class of Convex 0–1 Mixed Integer Programs. *Mathematical Programming*, 106(2):225 – 236, 2006.
- [18] A. Frangioni and C. Gentile. Solving Nonlinear Single-Unit Commitment Problems with Ramping Constraints. *Operations Research*, 54(4):767 – 775, 2006.
- [19] B.F. Hobbs, M. Rothkopf, R.P. O’Neill, and H.P. Chao. *The Next Generation of Unit Commitment Models*. Kluwer Academic Press, Boston, 2001.
- [20] S.-Y. Lai and R. Baldick. Unit Commitment with Ramp Multipliers. *IEEE Transactions on Power Systems*, 14(1):58–64, 1999.
- [21] Z. Li and M. Shahidehpour. Generation Scheduling With Thermal Stress Constraints. *IEEE Transactions on Power Systems*, 18(4):1402–1409, 2003.

- [22] B. Lu and M. Shahidehpour. Short-term scheduling of combined cycle units. *IEEE Transactions on Power Systems*, 19(3):1616–1625, 2004.
- [23] M. Madrigal and V.H. Quintana. An Interior-Point/Cutting-Plane Method to Solve Unit Commitment Problems. *IEEE Transactions on Power Systems*, 15(3):1022–1027, 2000.
- [24] O. Nilsson and D. Sjelvgren. Mixed-integer Programming Applied to Short-term Planning of a Hydro-thermal System. *IEEE Transactions on Power Systems*, 11(1):281–286, 1996.
- [25] M.P. Nowak and W. Römisich. Stochastic Lagrangian Relaxation Applied to Power Scheduling in a Hydro-thermal System Under Uncertainty. *Annals of Operations Research*, 100:251–272, 2000.
- [26] D.L. Travers and R.J. Kaye. Dynamic Dispatch by Constructive Dynamic Programming. *IEEE Transactions on Power Systems*, 13:72–78, 1998.
- [27] A.J. Wood and B.F. Wollemborg. *Power Generation Operation and Control*. John Wiley & Sons, 1996.
- [28] F. Zhuang and F.D. Galiana. Towards a More Rigorous and Practical Unit Commitment by Lagrangian Relaxation. *IEEE Transactions on Power Systems*, 3:763–773, 1988.