A new MILP-based approach for unit commitment in power production planning

Ana Viana, João Pedro Pedroso

A B S T R A C T

This paper presents a complete, quadratic programming formulation of the standard thermal unit commitment problem in power generation planning, together with a novel iterative optimisation algorithm for its solution. The algorithm, based on a mixed-integer formulation of the problem, considers piecewise linear approximations of the quadratic fuel cost function that are dynamically updated in an iterative way, converging to the optimum; this avoids the requirement of resorting to quadratic programming, making the solution process much quicker.

From extensive computational tests on a broad set of benchmark instances of this problem, the algorithm was found to be flexible and capable of easily incorporating different problem constraints. Indeed, it is able to tackle ramp constraints, which although very important in practice were rarely considered in previous publications.

Most importantly, optimal solutions were obtained for several well-known benchmark instances, including instances of practical relevance, that are not yet known to have been solved to optimality. Computational experiments and their results showed that the method proposed is both simple and extremely effective.

Keywords:
Combinatorial optimisation
Mixed-integer programming
Unit commitment

1. Introduction

The Unit Commitment Problem (UCP) is the problem of deciding which power generating units must be committed/decommitted over a planning horizon (that lasts from 1 day to 2 weeks, generally split into periods of 1 h each). The production levels at which units must operate (pre-dispatch) must also be determined to optimise a given objective function. The committed units must usually satisfy the forecasted system load and reserve requirements, as well as a large set of technological constraints.

This problem has great practical significance because the effectiveness of the schedules obtained has a strong economical impact on power generation companies. Due to this reason and to the problem's high complexity (a prove that it is NP-hard has been given in [1]), it has received considerable research attention. Even after several decades of intensive study, it is still a rich and challenging topic of research.

The proposed optimisation techniques for unit commitment encompass very different paradigms. These range from exact approaches and Lagrangian relaxation to rules of thumb or very elaborate heuristics and metaheuristics. In the past, the combinatorial nature of the problem and its multi-period characteristics have prevented exact approaches from being successful in practice: they resulted in highly inefficient algorithms that were only capable of solving small problem instances with virtually no practical interest. Heuristic techniques, such as those based on priority lists, did not totally succeed either, as they often lead to low quality solutions. Metaheuristics had very promising outcomes when they were first explored. The quality of their results was better than those achieved using well established techniques and good solutions were obtained very quickly. However, some drawbacks can be highlighted when metaheuristics are used. If one considers that the ultimate goal is to design a technique that can be accepted and used by a company, one major drawback of metaheuristics is their dependence on parameter tuning. Parameter tuning is time consuming and the complex tuning procedure requires profound knowledge of the algorithm implemented. Furthermore, accurate tuning is vital for algorithm performance. A second drawback is related to the lack of information that metaheuristics provide in terms of the quality of the solution (i.e., how far it is from the optimal solution). Some proposals have been presented to address these drawbacks; but this still remains an open line of research.

An open issue is related to solution optimality and how it affects individual pay-offs in restructured markets where an independent system operator performs a centralised unit commitment. As stated in [2], only if problems are solved to optimality can one guarantee that units will receive their optimal dispatch and pay-off. Therefore, the design and development of optimisation techniques that provide optimal results to unit commitment problems are of crucial importance.

Nomenclature

Constants

T	length of the planning horizon
$\mathcal{T} = \{1, \ldots, T\}$	set of planning periods
$\mathcal{U}$	set of units
$P_u^{\min}, P_u^{\max}$	minimum and maximum production levels for unit u
$T_u^{\text{on}}, T_u^{\text{off}}$	minimum number of periods that unit u must be kept switched on/off
$r_u^{\text{up}}, r_u^{\text{down}}$	maximum up/down rates of unit u
D_t	system load requirements in period t
R_t	spinning reserve requirements in period t
a_u, b_u, c_u	fuel cost parameters for unit u
$a_u^{\text{hot}}, a_u^{\text{cold}}$	hot and cold start up costs for unit u
t_u^{cold}	number of periods after which the start up of unit u is evaluated as cold
y_u^{prev}	previous state of unit u (1 if on, 0 if off)
t_u^{prev}	number of periods unit u has been on or off prior to the first period of the planning horizon

Decision variables

y_{ut}	1 if unit u is on in period t, 0 otherwise
p_{ut}	production level of unit u, in period t

Auxiliary variables

Auxiliary variables	Due to model structure, some of the auxiliary variables can be relaxed to the set [0,1], as discussed later
$x_{ut}^{\text{on}}, x_{ut}^{\text{off}}$	1 if unit u is started/switched off in period t, 0 otherwise
s_{ut}^{hot}	1 if unit u has a hot start in period t, 0 otherwise
s_{ut}^{cold}	1 if unit u has a cold start in period t,0 otherwise
$p_{ut}^{\max}$	maximum production levels for unit u in period t (due to ramp constraints)

Production costs

$F(p_{ut})$	fuel cost for unit u in period t
$S(x_{ut}^{\text{off}}, y_{ut})$	start up cost for unit u in period t
H_{ut}	shut down cost for unit u in period t

The dramatic increase in the efficiency of mixed-integer programming (MIP) solvers has encouraged the thorough exploitation of their capabilities. Some research has already been directed towards defining of alternative, more efficient, mixed-integer linear programming (MILP) formulations of this problem (see e.g., [3]). Extensive surveys of different optimisation techniques and modelling issues are provided in [4–6].

This paper proposes a MIP formulation for quadratic optimisation of the UCP, and also presents a method based on a linear formulation. The method has proven to be effective at solving instances of a practically relevant size. Instead of considering a quadratic representation of the fuel cost, the linear model considers a piecewise linear approximation of the function and updates it in an iterative process, by including additional pieces. Function updating is based on the solutions obtained in the previous iteration.

The solution approach developed in this research was tested on several well-known test instances that were not known to have been solved to optimality. For each of them, the new approach iteratively converged to the optimal solution, even for the largest benchmark instances.

2. Problem variants

Different modelling alternatives that reflect different problem issues, such as fuel, multiarea and emission constraints have been published (e.g., [7–9]). Security constraints [10] and market related aspects [11] have been addressed more recently.

The decentralised management of production has also introduced new issues to the area [12] and in some markets the problem has now been reduced to single-unit optimisation. However, for several decentralised markets the traditional problem is still very much similar to that of the centralised markets [3]; the main difference is the objective function that, rather than minimising production costs, maximises total welfare. Therefore, the techniques that apply for centralised production management will also be effective at solving many decentralised market production problems.

This paper considers the centralised UCP model. The objective of the problem is to minimise total production costs over a given planning horizon. The total production cost is expressed as the sum of fuel costs (quadratic functions that depend on the production level of each unit) and start-up costs. Start-up costs are represented by constants that depend on the last period when the unit was operating. In addition to the uninterrupted operation of the unit (i.e., no start-up cost[1]), two constants are defined: one constant for hot start-up costs when the unit has been off for a number of periods smaller than or equal to a given value, and the other for cold start-up costs. The following constraints will be included in the formulation: system power balance, system reserve requirements, unit initial conditions, unit minimum up and down times, generation limits and ramp constraints. For a standard quadratic mathematical formulation refer to [13].

3. MILP formulations for the UCP

For many years, approaches to solving the UCP were mainly based on Lagrangian relaxation and (meta) heuristics. This was due to the non-existence of exact approaches capable of coping with the computational complexity of the problem using reasonable resources. However, the dramatic improvement of MIP solvers in recent years suggested that an effort should be applied to studying "good" mathematical formulations of the problem, so that it can be handled by relevant solvers.

The first requirement is the linearisation of the various nonlinearities in the problem; namely, minimum up and down time constraints, minimum and maximum power production constraints (for problems that consider ramps), and the objective function.

Several efforts have been made to improve and strengthen the formulation of the UCP; pioneering work can be found in [14]. This work considers three sets of binary variables that model the state of each unit, start-ups, and shut-downs. The quadratic fuel cost function is represented by a piecewise-linear cost function. Initial attempts to solve the problem with standard branch-and-bound (B&B) proved to be inefficient. As a result, an extended version of the algorithm that considers problem-specific characteristics in the branching process was proposed. Results are provided for problems with up to 16 units and 14 time periods.

A thorough discussion of model linearisation, considering a perfect electricity spot market and a single unit (self-scheduling), is provided in [15]. The quadratic cost function is approximated using a piecewise linear cost function with L segments. Three extra sets of variables are required when compared to the model in [14]: 0/1

[1] Notice that even in this situation there is a fixed component in the quadratic cost function.

variables that are set to 1 if the unit is started-up at the beginning of hour t and it has been off for k hours; integer variables to set the number of hours the unit has been on or off at the end of hour t; and 0/1 variables that are set to 1 if the power output of the unit at hour t exceeds segment l of the piecewise linear approximation of the quadratic cost function.

The authors of [16] focus their study on the accurate modelling of start-up and shut-down power trajectories (that depend on ramps). The authors of [17] present a model similar to [15], that can adapt to centralised or competitive markets. The model is validated with a 27 unit × 24 h instance but no information on the efficiency of the algorithm is reported. Later, a model introduced in [3] reduced both the number of binary variables and the number of constraints from previous formulations. In [18] Wu presents a methodology to determine segment partition points that will provide a tighter piecewise linear approximation of the quadratic cost function. In this case the quadratic cost function is still approximated using a piecewise linear cost function with L segments but each segment may have a different length.

Still within the scope of MILP formulations for the UCP, a study on the quality of previously proposed valid inequalities was conducted in [19]. The authors focus on the valid inequalities used for minimum up and down time constraints and they show how some inequalities can be improved. In [20] the authors proposed a MILP formulation in which the "perspective cuts" concept (a family of valid inequalities) is used to approximate the non-linear objective function; according to the authors, this improves the solver's effectiveness and efficiency.

4. A MILP formulation

Although the quality of MILP solvers has improved dramatically in recent years, mathematical models such as the one given in [13] are not suitable for these solvers due to the various non-linearities. In order to fully utilise MILP solvers, these non-linearities must be removed from the model, if possible. In the following subsections a linearised mathematical model for the UCP is presented and discussed. The model was implemented in AMPL (*A Modelling Language for Mathematical Programming*) [21], and the CPLEX MIP solver was used to solve it.

4.1. System constraints

Two types of system constraints are considered: system power balance (1) and spinning reserve requirements (2). The impact of ramp constraints can be considered when setting reserve constraints. This can be achieved using the variable $p_{ut}^{\max}$, rather than the constant $P_u^{\max}$, in (2).

$$\sum_{u\in\mathcal{U}} p_{ut} = D_t, \quad \forall t \in \mathcal{T}, \tag{1}$$

$$\sum_{u\in\mathcal{U}} p_{ut}^{\max} \geqslant D_t + R_t, \quad \forall t \in \mathcal{T}, \tag{2}$$

with

$$p_{ut}^{\max} \leqslant y_{ut} P_u^{\max}, \quad \forall u \in \mathcal{U}, \text{ for } t = 2\ldots T,$$
$$p_{ut}^{\max} \leqslant p_{u,t-1} + y_{u,t-1} r_u^{\text{up}} + P_u^{\max}(1 - y_{u,t-1}),$$
$$\forall u \in \mathcal{U}, \text{ for } t = 2\ldots T.$$

4.2. Technical constraints

Technical constraints represent limitations of the generating units and constrain the system's overall performance (e.g., units' minimum up and down times, production limits, and ramps).

4.2.1. Minimum up and down times

When a unit u is switched on (off), it must remain on (off) for at least T_u^{on} (T_u^{off}) consecutive periods. Constraints (3) and (4) model this aspect for the initial state, while constraints (5) and (6) do the same for the remaining planning horizon. In (3) θ_u^{on} represents $\max\left(0, T_u^{\text{on}} - t_u^{\text{prev}}\right)$, and θ_u^{off} in (4) stands for $\max\left(0, T_u^{\text{off}} - t_u^{\text{prev}}\right)$.

$$y_{ut} = 1, \quad \forall u \in \mathcal{U} : y_u^{\text{prev}} = 1, \text{ for } t = 0, \ldots, \theta_u^{\text{on}}, \tag{3}$$

$$y_{ut} = 0, \quad \forall u \in \mathcal{U} : y_u^{\text{prev}} = 0, \text{ for } t = 0, \ldots, \theta_u^{\text{off}}. \tag{4}$$

In (5) and (6), τ_{ut}^{on} and τ_{ut}^{off} stand for $\max\left(t - T_u^{\text{on}} + 1, 1\right)$ and $\max\left(t - T_u^{\text{off}} + 1, 1\right)$, respectively.

$$\sum_{i=\tau_{ut}^{\text{on}}}^{t} x_{ui}^{\text{on}} \leqslant y_{ut}, \quad \forall u \in \mathcal{U}, \ \forall t \in \mathcal{T}, \tag{5}$$

$$\sum_{i=\tau_{ut}^{\text{off}}}^{t} x_{ui}^{\text{off}} \leqslant 1 - y_{ut}, \quad \forall u \in \mathcal{U}, \ \forall t \in \mathcal{T}. \tag{6}$$

In [22] the authors show that these inequalities are facets to the convex hull of the set $C_T(T^{\text{on}}, T^{\text{off}})$ which is the projection of the problem in the space of variables y and x^{on}.

4.2.2. Generation limits and ramps

Power production levels of thermal power units are within the range defined by the technical minimum and maximum production levels in (7).

$$P_u^{\min} y_{ut} \leqslant p_{ut} \leqslant P_u^{\max} y_{ut}, \quad \forall u \in \mathcal{U}, \ \forall t \in \mathcal{T}. \tag{7}$$

If ramps are considered (i.e., if the difference of values in production levels is limited to a maximum value in consecutive periods) additional constraints are needed. Constraints (8) and (9) model, respectively, maximum up and down rates for each unit in consecutive periods of time.

$$p_{ut} - p_{u,t-1} \leqslant r_u^{\text{up}}, \quad \forall u \in \mathcal{U}, \ \forall t \in \mathcal{T}, \tag{8}$$

$$p_{u,t-1} - p_{ut} \leqslant r_u^{\text{down}}, \quad \forall u \in \mathcal{U}, \ \forall t \in \mathcal{T}. \tag{9}$$

4.3. Additional constrains

A set of additional constraints for the computation of auxiliary variables allows relaxation of integrality for variables x_{ut}^{on} and x_{ut}^{off}, as discussed below.

4.3.1. Setting and computation of variables s_{ut}^{hot} and s_{ut}^{cold}

Constraints (10) state that every time a unit is switched on, a start-up cost will be incurred. The same type of constraint is used in [23] for a more general case, where three start-up types are considered: hot, warm and cold start-ups.

$$s_{ut}^{\text{hot}} + s_{ut}^{\text{cold}} = x_{ut}^{\text{on}}, \quad \forall u \in \mathcal{U}, \ \forall t \in \mathcal{T}. \tag{10}$$

Constraints (11) determine the start-up type of each unit, i.e., decide whether it is a cold or a hot start type. It will be a cold start if the unit remained off for more than t_u^{cold} periods of time, and a hot start otherwise.

$$y_{ut} - \sum_{i=t-t_u^{\text{cold}}-1}^{t-1} y_{ui} \leqslant s_{ut}^{\text{cold}}, \quad \forall u \in \mathcal{U}, \ \forall t \in \mathcal{T}. \tag{11}$$

The same constraint is modelled in a similar way in [23]. However, instead of using variables y_{ut} the authors consider variables x_{ut}^{on}. In both cases, the constraint only models the problem properly if hot start-up costs are smaller than cold start-up costs.

4.3.2. Setting and computation of variables x_{ut}^{on} and x_{ut}^{off}

Constraints (12) determine each unit's switch-on variables, and (13) determine the switch-off variables.

$$y_{ut} - y_{u,t-1} \leqslant x_{ut}^{on}, \quad \forall u \in \mathcal{U}, \ \forall t \in \mathcal{T}, \tag{12}$$

$$x_{ut}^{off} = x_{ut}^{on} + y_{u,t-1} - y_{ut}, \quad \forall u \in \mathcal{U}, \ \forall t \in \mathcal{T}. \tag{13}$$

4.3.3. Relaxation of integrality constraints on variables x_{ut}^{on} and x_{ut}^{off}

Constraints (10) and (13) make it possible to relax variables x_{ut}^{on} and x_{ut}^{off}. In fact, if s_{ut}^{hot} and s_{ut}^{cold} are defined as binary variables, using (10) x_{ut}^{on} will always be 0 or 1. Furthermore, since y_{ut} is binary, using (13) x_{ut}^{off} will always be set to 0 or 1, for feasible y_{ut}.

4.4. Objective function

The objective of this problem is to minimise the total production cost over the planning horizon, expressed as the sum of fuel, start-up costs and shut-down costs (14).

$$\text{minimise} \sum_{t \in \mathcal{T}} \sum_{u \in \mathcal{U}} \left(F(p_{ut}) + S(x_{ut}^{off}, y_{ut}) + H_{ut} \right). \tag{14}$$

We consider the traditional quadratic function for $F(p_{ut})$, as follows:

$$F(p_{ut}) = \begin{cases} c_u p_{ut}^2 + b_u p_{ut} + a_u & \text{if } y_{ut} = 1, \\ 0 & \text{otherwise}. \end{cases} \tag{15}$$

Shut-down costs H_{ut} are assumed to be zero and start-up costs are modelled as:

$$S(x_{ut}^{off}, y_{ut}) = a_u^{hot} s_{ut}^{hot} + a_u^{cold} s_{ut}^{cold}. \tag{16}$$

This is a linearised version of the (non-linear) function proposed in [24] to represent start-up-costs:

$$S(x_{ut}^{off}, y_{ut}) = y_{ut}(1 - y_{u,(t-1)}) S_x(x_{ut}^{off}), \tag{17}$$

where S_x depends on the last period the unit was operating as follows:

$$S_x = \begin{cases} a_u^{hot} & \text{if } \gamma_{ut}^{off} \leqslant t_u^{cold}, \\ a_u^{cold} & \text{otherwise}, \end{cases} \tag{18}$$

with γ_{ut}^{off} as the number of consecutive periods unit u was off before period t.

5. Iterative linear algorithm

The new solution approach considers a piecewise linear approximation of the quadratic fuel cost function (15), where a linear MILP model is iteratively solved. The MILP will provide increased precision at each iteration, until a user-defined proximity to the quadratic function is reached.

The algorithm is an application of Kelley's theorem on the cutting plane method for convex programs (see [25]), which states the conditions and method of obtaining a sequence of points that converge to the minimum of a convex function in a compact, convex set, using successive linear approximations. The theorem is the following: let $G(\boldsymbol{x})$ be a continuous convex function defined in the n-dimensional compact convex set S. Feasible points for this problem are $\boldsymbol{x} \in S$: $G(\boldsymbol{x}) \leqslant 0$. An extreme support to the graph of G is an $(n+1)$-dimensional hyperplane that intersects the boundary of the convex set $P = \{(\boldsymbol{x}, y): \boldsymbol{x} \in S, y \geqslant G(\boldsymbol{x})\}$ and does not cut the interior of P. Consider the extreme support $y = p(\boldsymbol{x}; \boldsymbol{t})$ to the graph of G in a point $\boldsymbol{t} \in S$; this can be written as $p(\boldsymbol{x}; \boldsymbol{t}) = G(\boldsymbol{t}) + \nabla p(\boldsymbol{x}; \boldsymbol{t})(\boldsymbol{x} - \boldsymbol{t})$. Assume that, for some finite constant K, $\|\nabla p(\boldsymbol{x}; \boldsymbol{t})\| \leqslant K$ for all $\boldsymbol{x} \in S$. Let $\boldsymbol{cx}$ be a linear form such that $\|\boldsymbol{c}\| < \infty$ and let $R = \{\boldsymbol{x} \in S: G(\boldsymbol{x}) \leqslant 0\} \subset S$ be non-empty. The optimisation problem is that of finding a vector $\boldsymbol{\tau}$ such that $\boldsymbol{c}\,\boldsymbol{\tau} = \min\{\boldsymbol{cx}: \boldsymbol{x} \in R\}$. The theorem states that if $\boldsymbol{t}_k \in S_k$ is determined such that $\boldsymbol{ct}_k = \min\{\boldsymbol{cx}: \boldsymbol{x} \in S_k\}$, for $k = 0, 1, \ldots,$ where $S_0 = S$ and $S_k = S_{k-1} \cap \{\boldsymbol{x}: p(\boldsymbol{x}; \boldsymbol{t}_{k-1}) \leqslant 0\}$, then the sequence $\boldsymbol{t}_k$ contains a subsequence that converges to a point $\boldsymbol{\tau} \in R$ with $\boldsymbol{c\tau} \leqslant \boldsymbol{cx}$ for all $\boldsymbol{x} \in R$. Therefore, it converges to the minimum of the optimisation problem.

The application of Kelley's theorem to the model presented in Section 4 is straightforward, and visually easy to explain. In this case, besides being convex, the cost function is separable into the sum of one-dimensional functions (one for each generator). For each of these one-dimensional functions, a set of extreme supports is determined and, for each of them, the cost is constrained to be greater than or equal to the value of the corresponding linear function. This leads to a lower approximation of the cost. The process proposed is to dynamically find linear functions that are tangent to the true cost at points where it is being underestimated (i.e., the extreme support at these points) and add them to a set. The cost at any production level p must then be greater than the maximum of these linear functions, evaluated at p.

For clarity, let us remove the indices u and t identifying the generator and time period, respectively. For any generator and any period, we start by approximating its cost by means of two linear functions: one going through $(P^{min}, F(P^{min}))$ and another going through $(P^{max}, F(P^{max}))$, as shown in Fig. 1.

After solving the problem using this approximation, a production level of, say, p is obtained for a unit. The operating cost at this point will be underestimated as the value of the highest straight line at p. In Fig. 1 this is given as the value $\widetilde{F}$. In order to exclude this point from the feasible region, we add a constraint based on the extreme support of the cost at point p; this is the line that is tangent to the quadratic function evaluated at p, represented as a solid grey line in Fig. 2. When the problem is solved with this additional constraint added, the solution may change; the optimal production level for this same unit may now be another possible value p', as shown in Fig. 2. As more and more extreme supports are added and the highest is selected, this converges to an exact representation of the true cost function.

5.1. Algorithm description

For each unit, we start with the corresponding quadratic fuel cost function $F(p)$ approximated by two linear functions, the first being tangent to $F(p)$ at the minimum power $(P^{min}, F(P^{min}))$, and the second being tangent at $(P^{max}, F(P^{max}))$ (see Fig. 1).

Thereafter, more extreme support lines are iteratively added into a set, until reaching one iteration with all production levels correctly evaluated (up to an acceptable error).

Let $\mathcal{P}$ be a set of numbers identifying the power at which new extreme supports to the true cost are added for a given unit; initially $\mathcal{P} = \{P^{min}, P^{max}\}$. At a given iteration, let the production level

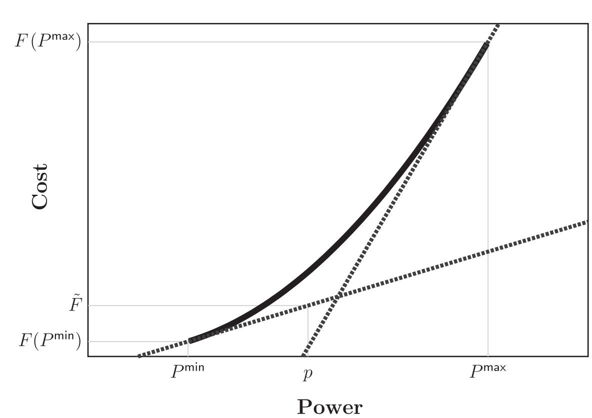

Fig. 1. Initial approximation of the cost function by two straight lines going through the minimum and maximum operating power of the unit. If the current production level for this unit is p, its cost (in this iteration) will be approximated by $\widetilde{F}$.

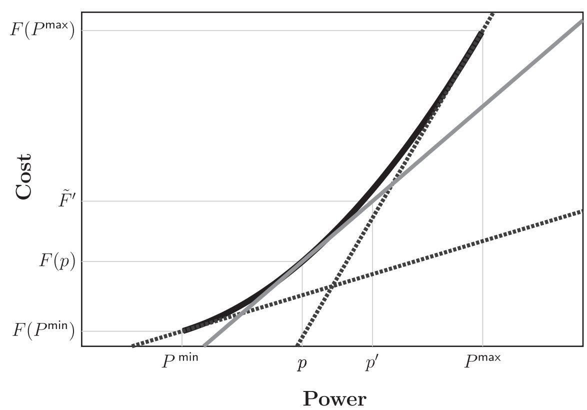

Fig. 2. Approximation of the cost function by the maximum of three straight lines after obtaining production at level p in the previous iteration.

obtained in the MILP solution be p and the corresponding cost approximation (i.e., the maximum of the linear cost functions evaluated at p) be $\widetilde{F}$. For a given unit and a given period, point p is added to the set $\mathcal{P}$ whenever $|F(p) - \widetilde{F}|/F(p) > \epsilon$, where ϵ is a user-defined tolerance. Otherwise, the current approximation is accepted as accurate enough.

In the MILP solved at each iteration, the following constraints are added (which, in this form, are only imposed if the corresponding unit is on during the period considered):

$$F_{ut} \geqslant (\alpha_{un} - \beta_{un}\bar{p}_n)y_{ut} + \beta_{un}p_{ut}, \quad \text{for } n = 1, \ldots, |\mathcal{P}|,$$

now using the actual variables p_{ut} for production level and F_{ut} for production cost, for a given unit at a given period. For a given unit, and for each production level $\bar{p}_n$ where the approximation does not satisfy the tolerance, the constants for the above straight lines are obtained by:

$$\alpha_{un} = c_u\bar{p}_n^2 + b_u\bar{p}_n + a_u,$$
$$\beta_{un} = 2c_u\bar{p}_n + b_u.$$

The algorithm stops when in a given iteration the set $\mathcal{P}$ is unchanged, thus no extreme support constraints are added, meaning that all the production costs are already being correctly evaluated up to the specified tolerance ϵ, for all units, in all periods. In this experiment, ϵ is set to 10^{-6}; this allows an excellent approximation of the quadratic function in all of the instances tested (actually, no difference was observed between quadratic costs and the linear approximation, concerning the solutions obtained).

Even though cutting plane algorithms are usually employed in sophisticated branch-and-cut methods, in the context of this paper it must be noted that they were only applied in iterative calls to a solver black-box. This was very simply implemented in a mathematical modelling language [26]. Therefore, the burden of implementing a solver with a problem-specific branch-and-cut method, as has been done in [20], is not required. Furthermore, different solvers can be used with no additional programming effort.

6. Computational results

The algorithm was tested in two sets of instances: one without ramp constraints but that is a reference for comparing UC algorithms [24] (instances P1 through P6); and the other with ramp constraints (instances R1 through R6). CPU times were obtained with CPLEX 12.1 on a computer with a Quad-Core Intel Xeon processor at 2.66 GHz and running Mac OS X 10.6.6. Only one thread was assigned to this experiment. Models were written in the AMPL language [21] and the default CPLEX parameters were used for the linear models. For the quadratic models to converge to the correct

Table 1
Results for instances P1–P6 (standard Kazarlis set).

Instance	Size	Iterative linear alg.		Quadratic model	
		Objective	CPU	Objective	CPU
P1	10	565,827.7	0.32	565,827.7	1.29
P2	20	1,125,997.4	4.99	1,125,997.4	115.8
P3	40	2,248,284.7	63.3	2,248,284.8	6080
P4	60	3,368,949.7	534		
P5	80	4,492,173.1	54,330		
P6	100	5,612,686.1	15,625		

Table 2
Results for insctances R1–R6 (including ramp constraints).

Instance	Size	Iterative linear alg.		Quadratic model	
		Objective	CPU	Objective	CPU
R1	10	570,396.4	3.27	570,396.4	5.17
R2	20	1,135,452.3	215	1,135,452.4	1752
R3	40	2,267,535.7	4059		
R4	60	3,398,614.4	4457		
R5	80	4,531,720.8	38,690		
R6	100	5,662,901.4	112,791		

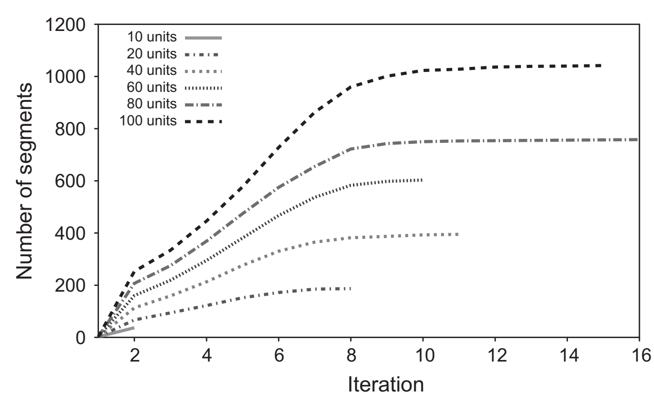

Fig. 3. Number of extreme support constraints (i.e., the number of piecewise linear segments) added up for all units, in terms of the iteration number.

solution, the following parameters had to be changed: `mipgap`, `absmipgap` and `qcpconvergetol` were set to 10^{-12}, and `integrality` was set to 10^{-9}.

Tables 1 and 2 present the results obtained using the algorithm proposed for different sets of UCP instances. Instances P1–P6, in Table 1, are the standard Kazarlis benchmarks [24], which do not include ramp constraints. Ramp constraints are considered in instances R1–R6 (Table 2), resulting from instances P1–P6, by setting ramp up-and-down maximum values identical to the minimum production level for each unit. All instances are for a 24-h planning horizon, with one period per hour, and the number of units ranged from 10 to 100. Empty entries in the tables mean that the solver could not find the solution within 24 h of CPU time.

In Tables 1 and 2, columns *Quadratic model* provide the optimal result for the base problem using quadratic programming, and columns *Iterative linear algorithm* provide the results obtained using the method we propose, confirming that the algorithm converges to the optimal solution. Columns CPU refer to the time (in seconds) taken by each of the methods to solve the problem. Attempts to solve the problem with the quadratic formulation without ramps were not successful for instances with more than 40 units. For the problem with ramps, the quadratic formulation was unsuccessful for instances with more than 20 units.

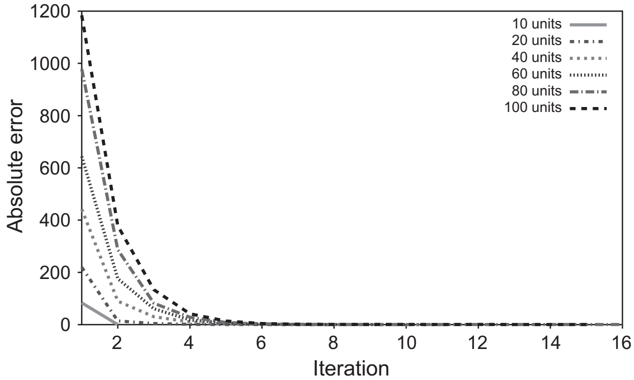

Fig. 4. Sum, for all units and all periods, of the absolute error of evaluation of the quadratic costs by piecewise linear approximations, in terms of the iteration number. True cost is underestimated by this amount, at most.

The number of segments added during the solution process as a function of the iteration number is shown in Fig. 3 for the standard instances P1–P6. The evolution of the error with respect to the iteration number for the same instances, is shown in Fig. 4. It can be seen that the total absolute error of evaluation for the quadratic costs rapidly decreases with the iteration number.

In order to compare the effectiveness of the iterative linear algorithm, with respect to standard piecewise linear formulations, Table 3 provides the value of the absolute error in the representation of the cost function (i.e., the quadratic cost minus the linear approximation, added up for all units and all periods) for the case of a fixed number of five segments for each unit, as well as the measure for the solution of the iterative linear algorithm with $\epsilon = 10^{-6}$. In the former the error is considerable and in the latter is under half a unit for all instances. The cost for this increased precision is an increase in the CPU time required, which is higher for the linear iterative algorithm (although it is much lower than that required by the quadratic programming solver). One explanation for having relatively few segments and greatly increased accuracy in the iterative linear algorithm concerns the distribution of the support points. Indeed, they are added as needed in irregular patterns that vary widely from unit to unit, as shown in Fig. 5.

It should be mentioned that the CPU time required to solve these instances shows a rather random pattern; it seems that the solver used (CPLEX) is sometimes trapped, taking much more time to finish than usual. This was observed in the solution of successive iterations of the same instance: the CPU time required varied widely, as shown in Fig. 6 (note that ordinate is in logarithmic scale).

Table 4 presents the results obtained using the algorithm proposed in this paper when start-up costs are evaluated by (19); these instances will be referred to as M1–M6. This equation was first proposed in [27] and was later used by several authors.

$$S_x = \begin{cases} a_u^{\text{hot}} & \text{if } T_u^{\text{off}} < \gamma_{ut}^{\text{off}} \leqslant T_u^{\text{off}} + t_u^{\text{cold}}, \\ a_u^{\text{cold}} & \text{otherwise}. \end{cases} \tag{19}$$

Again, γ_{ut}^{off} is the number of consecutive periods unit u was off before period t.

Table 3
Comparison of results obtained for a model with a fixed number of five segments, and the results of the iterative method.

Instance	Size	Five linear segments			Iterative linear algorithm			
		Error	#Segm	CPU	Error	#Iter	#Segm	CPU
P1	10	3.54	50	0.20	0	2	37	0.32
P2	20	12.16	100	0.62	0.072	8	187	4.99
P3	40	27.36	200	3.62	0.153	11	395	63.3
P4	60	44.23	300	43.0	0.259	10	603	534
P5	80	69.83	400	1213	0.327	16	758	54,329
P6	100	83.69	500	261	0.497	15	1042	15,624

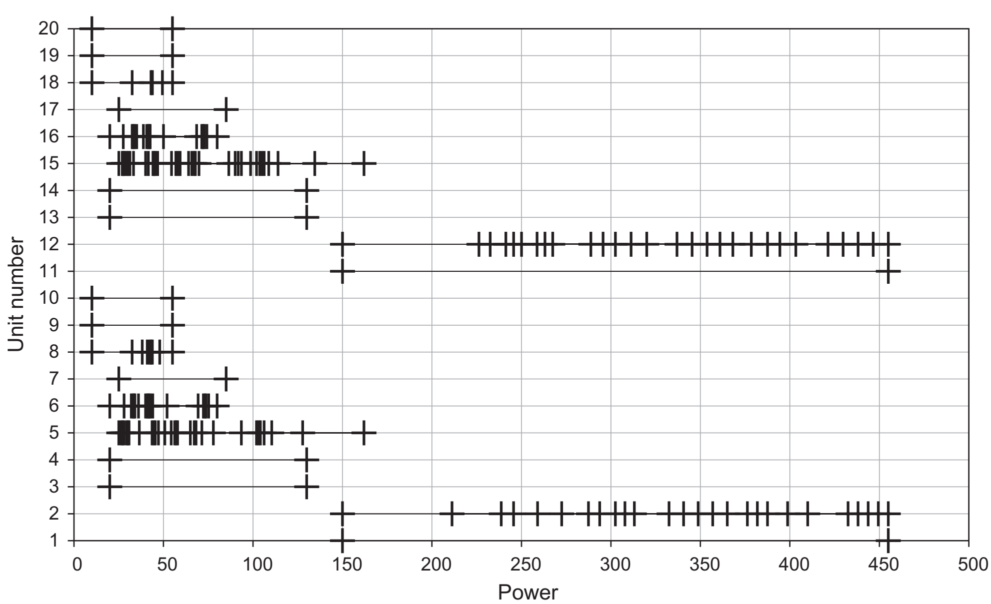

Fig. 5. Power at which support points have been added with instance P3: results for the first 20 units.

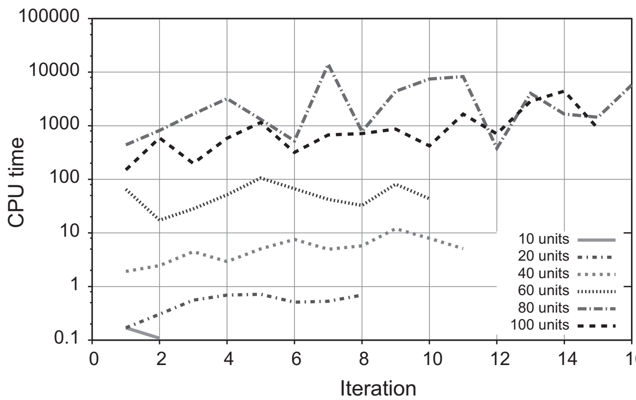

Fig. 6. CPU time required to solve the linear problem in each iteration.

Table 4
Results for instances M1–M6 (modified cost function).

Instance	Size	Iterative linear alg.		Quadratic model	
		Objective	CPU	Objective	CPU
M1	10	563,937.7	0.58	563,937.7	1.73
M2	20	1,123,297.4	16.4	1,123,297.4	150
M3	40	2,242,575.4	960		
M4	60	3,359,954.8	271		
M5	80				
M6	100	5,597,770.1	6341		

Table 5
Previous results for instances P1–P6: best solution found.

Instance	Size	LR [24]	GA [24]	LR–MA [28]
P1	10	565,825	565,825	565,827
P2	20	1,130,660	1,126,243	1,127,254
P3	40	2,258,503	2,251,911	2,249,589
P4	60	3,394,066	3,376,625	3,370,595
P5	80	4,526,022	4,504,933	4,494,214
P6	100	5,657,277	5,627,437	5,616,314
		ICGA [29]	GRASP [13]	CON [30]
P1	10	566,404	565,825	565,825
P2	20	1,127,244	1,126,805	1,126,070
P3	40	2,254,123	2,255,416	2,248,490
P4	60	3,378,108	3,383,184	3,370,530
P5	80	4,498,943	4,524,207	4,494,140
P6	100	5,630,838	5,668,870	5,615,410

Table 5 presents the results reported in the literature for instances P1–P6, using different heuristic methods. Although the objective function value reported in this paper (565827.7) for the 10 unit instance using the iterative linear algorithm is higher than the one reported in some papers (565,825), the actual solution is the same. Small differences in values can be attributed to the possible rounding of values by other authors. Similarly, Table 6 presents the results reported in the literature for instances M1–M6, when start-up costs are modelled using (19). Naturally, the results reported in Tables 5 and 6 cannot be compared. At most, the results in Table 6 could be seen as lower bounds for instances P1–P6, if they were optimal. Note that some of the results presented are below the optimum values in Table 4; for the cases that were possible to verify, not all problem contraints were being observed. Limited floating point precision may also be a reason to explain those results.

To the best of our knowledge, no optimal solutions have ever been reported for instances P1–P6, with a quadratic cost function, not even for the smallest one. We have shown that for instances up to 40 units, optimal solutions can be obtained using current, state-of-the-art MIP solvers. Moreover, with the iterative approach based on the linear approximative model, it was possible to reach the optimal solution with dramatic reductions in CPU times, when compared to the direct solution approach with the quadratic solver of CPLEX, having determined the optimal solutions (with a tolerance $\epsilon = 10^{-6}$ on production costs) for all of the instances. These can be compared to the best published values for the quadratic models (see Tables 5 and 6).

Table 6
Previous results for instances M1–M6: best solution found (using the modified cost function).

Instance	Size	MILP-UC [3]	OGS [31]	ELR [32]
M1	10		561,436	563,937
M2	20		1,121,751	1,122,637
M3	40			2,243,245
M4	60			3,363,376
M5	80			4,484,915
M6	100	5,605,189		5,604,470
		IQEA-UC* [33]	SFLA [34]	MM [35]
M1	10	563,938	564,769	
M2	20	1,123,297	1,123,261	
M3	40	2,242,980	2,246,005	2,247,162
M4	60	3,362,010	3,368,257	3,366,874
M5	80	4,482,826	4,503,928	4,490,208
M6	100	5,602,387	5,624,526	5,609,782

Similar conclusions may be drawn for the ramp problem. The quadratic solver of CPLEX was capable of reaching optimal solutions for problem instances of up to 20 units. Optimal values for the whole set of instances were, again, reached using the iterative linear algorithm, albeit using larger CPU times. All the solutions obtained are available for download in [26].

7. Conclusions and further developments

The main contributions of this paper are a complete formulation of the standard thermal unit commitment problem in quadratic programming and a novel and efficient methodology for approximating the quadratic cost of electricity generating units, with an iterative method that uses a linear model, converging to the exact solution. Using this method we were able to solve all instances of a widely used benchmark testbed, for which to the best of our knowledge no optimum results were previously reported. The paper also establishes optimal solution for small instances of the quadratic model (with and without ramp constraints) showing that a simple method exploring the potential of current state-of-the-art solvers can tackle problems that were not previously solvable. This achievement is particularly relevant in markets were the independent system operator performs a centralised commitment and where there is a guarantee that units will receive their optimal dispatch and pay-off only when problems are solved to optimality. Previously proposed techniques for unit commitment could not, in general, guarantee that this goal was achieved as they were mostly based on (meta) heuristics. Prior approaches based on mixed-integer programming provided approximations, and did not seek for convergence to optimality.

A computational analysis has shown that the iterative linear method is capable of reaching the optimum of the quadratic model, when it is known, using much less computational time than required for its quadratic programming solution. For large problem instances, where the quadratic model could not be solved directly in a reasonable time, the iterative linear algorithm has found the optimal solution.

Similar conclusions could be drawn when ramp constraints are considered. The iterative linear algorithm was also capable of reaching the optimum for all of the instances. This is particularly important for instances with more than 20 units, for which the straightforward use of the quadratic programming solver was not successful.

As future work, we plan to extend the algorithm in order to solve a broader set of models, for example, to include other electricity generating technologies in addition to thermal units. The approach used to approximate the quadratic cost can be applied to any convex function. In terms of practical applications, this is a feature that should be explored, as it may lead to a better model for the true cost function (instead of quadratic).

Acknowledgements

Financial support for this work was provided by the Portuguese Foundation for Science and Technology (under Project PTDC/EGE-GES/099120/2008) through the "Programa Operacional Temático Factores de Competitividade (COMPETE)" of the "Quadro Comunitário de Apoio III", partially funded by FEDER.

Appendix A. Satisfaction of problem's constraints

In order to illustrate how the problem's constraints for minimum up and down times are verified, let us consider solutions to the 10 units instance P1 with and without considering them, as shown in the left-hand side of Table A.1. In this instance, the minimum number of periods that unit 3 must be off is five, which in the relaxed instance is not verified in period 18. In the left-hand solution, we observe that, for generator 3, Eq. (13) implies that $x^{\text{off}}_{3,16} = x^{\text{on}}_{3,16} + y_{3,15} - y_{3,16} = 1$. However, Eq. (6) implies $x^{\text{off}}_{3,14} + x^{\text{off}}_{3,15} + x^{\text{off}}_{3,16} + x^{\text{off}}_{3,17} + x^{\text{off}}_{3,18} \leqslant 1 - y_{3,18}$; as the left-hand side here is 1, this equation forces $y_{3,18}$ to be 0 (which is not observed in the relaxed solution).

Similarly, for generator 7—which has a minimum up time of three: in period 20 Eq. (12) becomes $x^{\text{on}}_{7,20} \geqslant y_{7,20} - y_{7,19} = 1$. Eq. (5) implies that $x^{\text{on}}_{7,20} + x^{\text{on}}_{7,21} + x^{\text{on}}_{7,22} \leqslant y_{7,22}$ forces $y_{7,22}$ to be 1 (which, again, is not observed in the relaxed solution).

It can be trivially shown that the correct solution (on the right-hand side of Table A.1) satisfies these equations, in both cases.

Appendix B. Solution to 100 units instance

Table B.1 displays the solution to the 100 units instance using the original cost function (instance P6). The way the instance is generated, by replicating 10 times the units of instance P1, leads to profuse symmetry; many solutions with the same objective

Table A.1
Solutions to the 10 units instance P1. Left: without considering minimum off and on periods. Right: actual instance (with those constraints).

Generator	Period (0 2 4 6 8 10 12 14 16 18 20 22 24)	Generator	Period (0 2 4 6 8 10 12 14 16 18 20 22 24)
1	1111111111111111111111111	1	1111111111111111111111111
2	1111111111111111111111111	2	1111111111111111111111111
3	000000111111111**1001**111000	3	000000111111111**1111**111000
4	0000011111111111111111100	4	0000011111111111111111000
5	0001111111111111111111100	5	0001111111111111111111100
6	0000000001111110000011110	6	0000000001111110000011110
7	00000000011111100000**110**00	7	00000000011111100000**111**00
8	0000000000111100000010000	8	0000000000111100000010000
9	0000000000011000000000000	9	0000000000011000000000000
10	0000000000001000000000000	10	0000000000001000000000000

Table B.1
Solution to the 100 units instance P6.

Period	Generator 1 3 5 7 9	11 13 15 17 19	21 23 25 27 29	31 33 35 37 39	41 43 45 47 49	51 53 55 57 59	61 63 65 67 69	71 73 75 77 79	81 83 85 87 89	91 93 95 97 99
0	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000
1	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000
2	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000
3	1100100000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100000000	1100100000
4	1100100000	1100000000	1100100000	1101100000	1100100000	1100100000	1101000000	1100000000	1100100000	1100100000
5	1101100000	1101000000	1100100000	1101100000	1100100000	1101100000	1101000000	1101000000	1100100000	1100100000
6	1101100000	1101100000	1111100000	1111100000	1101100000	1101100000	1101100000	1101000000	1101100000	1101100000
7	1101100000	1111100000	1111100000	1111100000	1111100000	1101100000	1101100000	1101100000	1101100000	1111100000
8	1111100000	1111100000	1111100000	1111100000	1111100000	1111100000	1111100000	1111100000	1111100000	1111100000
9	1111110000	1111111000	1111100000	1111110000	1111111000	1111100000	1111110000	1111110000	1111111000	1111111000
10	1111111100	1111111100	1111111100	1111111100	1111111100	1111111000	1111111100	1111111100	1111111100	1111111000
11	1111111100	1111111110	1111111110	1111111100	1111111110	1111111110	1111111110	1111111110	1111111110	1111111110
12	1111111111	1111111111	1111111110	1111111111	1111111111	1111111111	1111111110	1111111111	1111111111	1111111111
13	1111111100	1111111100	1111111100	1111111100	1111111100	1111111100	1111111000	1111111000	1111111100	1111111100
14	1111111000	1111111000	1111110000	1111101000	1111100000	1111110000	1111110000	1111111000	1111110000	1111110000
15	1111100000	1111100000	1111110000	1111100000	1101100000	1111100000	1111110000	1101100000	1111100000	1111100000
16	1111100000	1111100000	1111100000	1111100000	1101100000	1111100000	1111100000	1101100000	1111100000	1111100000
17	1111100000	1111100000	1111100000	1111100000	1101100000	1111100000	1111100000	1101100000	1111100000	1111100000
18	1111100000	1111100000	1111100000	1111100000	1101100000	1111100000	1111100000	1101100000	1111100000	1111100000
19	1111100000	1111100000	1111100000	1111100000	1101100000	1111100000	1111100000	1101101000	1111100000	1111110000
20	1111110110	1111110111	1111110110	1111110110	1111110110	1111110110	1111110110	1111111110	1111110110	1111110111
21	1111110100	1111110000	1111110000	1111110000	1111110000	1111110000	1111110000	1111111000	1111110100	1111110000
22	1101110000	1101010000	1101010000	1101010000	1111110000	1101010000	1101110000	1111110000	1101010000	1101010000
23	1100100000	1100000000	1100000000	1100000000	1110010000	1100000000	1100100000	1110100000	1100000000	1100000000
24	1100000000	1100000000	1100000000	1100000000	1010000000	1100000000	1100000000	1110000000	1100000000	1100000000

can be obtained by exchanging the states of equivalent generators, when they are different in a given period (e.g., generators 5 and 15 in period 23). The amount of symmetry in the solution space is typically one of the causes of difficulty in solving exactly a combinatorial optimisation problem.

References

[1] Tseng C. On power system generation unit commitment problems. Ph.D. thesis, Department of Industry Engineering and Operations Research, University of California at Berkeley; 1996.
[2] Sioshansi R, O'Neill R, Oren SS. Economic consequences of alternative solution methods for centralized unit commitment in day-ahead electricity markets. IEEE Trans Power Syst 2008;23(2):344–52.
[3] Carrion M, Arroyo JM. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Trans Power Syst 2006;21(3):1371–8.
[4] Sen S, Kothari D. Optimal thermal generating unit commitment: a review. Int J Electr Power Energy Syst 1998;20(7):443–51.
[5] Padhy NP. Unit commitment – a bibliographical survey. IEEE Trans Power Syst 2004;19(2):1196–205.
[6] Yamin HY. Review on methods of generation scheduling in electric power systems. Electr Power Syst Res 2004;69:227–48.
[7] López JA, Ceciliano-Meza JL, Moya IG, Gómez RN. A MIQCP formulation to solve the unit commitment problem for large-scale power systems. Int J Electr Power Energy Syst 2012;36(1):68–75.
[8] Ouyang Z, Shahidehpour SM. Heuristic multi-area unit commitment with economic dispatch. IEE Proc – C 1991;138:242–52.
[9] Srinivasan D, Tettamanzi AGB. An evolutionary algorithm for evaluation of emission compliance options in view of the clean air act amendments. IEEE Trans Power Syst 1997;12(1):336–41.
[10] Kumar VS, Mohan M. Solution to security constrained unit commitment problem using genetic algorithm. Int J Electr Power Energy Syst 2010;32(2):117–25.
[11] Xu J, Christie RD. Decentralised unit commitment in competitive energy markets. In: Hobbs BF, Rothkopf MH, O'Neill RP, Chao H, editors. The next generation of electric power unit commitment models. Kluwer Academic Publishers; 2001. p. 293–315.
[12] Hobbs BF, Rothkopf MH, O'Neill RP, Chao H, editors. The next generation of electric power unit commitment models. Kluwer Academic Publishers; 2001.
[13] Viana A, Sousa JP, Matos MA. Using GRASP to solve the unit commitment problem. Ann Operat Res 2003;120(1):117–32.
[14] Dillon TS, Edwin KW, Kochs HD, Taud RJ. Integer programming approach to the problem of optimal unit commitment with probabilistic reserve determination. IEEE Trans Power Appar Syst 1978;PAS-97(6):2154–66.
[15] Arroyo JM, Conejo AJ. Optimal response of a thermal unit to an electricity spot market. IEEE Trans Power Syst 2000;15(3):1098–104.
[16] Arroyo JM, Conejo AJ. Modelling of start-up and shut-down power trajectories of thermal units. IEEE Trans Power Syst 2004;19:1562–8.
[17] Chang GW, Tsai YD, Lai CY, Chung JS. A practical mixed integer linear programming based approach for unit commitment. In: Proceedings of the 2004 IEEE Power Engineering Society general meeting, vol. 1. Toronto, Canada; 2004. p. 221–5.
[18] Wu L. A tighter piecewise linear approximation of quadratic cost curves for unit commitment problems. IEEE Trans Power Syst 2011;26(4):2581–3.
[19] Hedman KW, O'Neill RP, Oren SS. Analyzing valid inequalities of the generation unit commitment problem. In: Proceedings of 2009 IEEE power system conference and exposition (PSCE). Seattle, USA; 2009. p. 1–6.
[20] Frangioni A, Gentile C, Lacalandra F. Tighter approximated MILP formulations for unit commitment problems. IEEE Trans Power Syst 2009;24(1):105–13.
[21] Fourer R, Gay DM, Kernighan BW. AMPL: a modeling language for mathematical programming. 2nd ed. Brooks/Cole Publishing Company; 2003.
[22] Rajan D, Takriti S. Minimum up/down polytopes of the unit commitment problem with start-up costs. Research Report RC 23628 (W0506–050). IBM; 2005.
[23] Simoglou CK, Biskas PN, Bakirtzis AG. Optimal self-scheduling of a thermal producer in short-term electricity markets by MILP. IEEE Trans Power Syst 2010;25(4):1965–77.
[24] Kazarlis SA, Bakirtzis AG, Petridis V. A genetic algorithm solution to the unit commitment problem. IEEE Trans Power Syst 1996;11:83–92.
[25] Kelley J. The cutting-plane method for solving convex programs. J Soc Indust Appl Math 1960;8:703–12.
[26] Viana A, Pedroso JP. Unit commitment problem: models, methods, and results. Internet repository, version 1.0; 2012. <http://www.dcc.fc.up.pt/ jpp/code/ucp>.
[27] Juste KA, Kita H, Tanaka E, Hasegawa J. An evolutionary programming solution to the unit commitment problem. IEEE Trans Power Syst 1999;14:1452–9.
[28] Valenzuela J, Smith AE. A seeded memetic algorithm for large unit commitment problems. J Heurist 2002;8(2):173–95.
[29] Damousis IG, Bakirtzis AG, Dokopoulos PS. A solution to the unit commitment problem using integer-coded genetic algorithm. IEEE Trans Power Syst 2004;19:1165–72.
[30] Viana A, Sousa JP, Matos MA. Fast solutions for UC problems by a new metaheuristic approach. Electr Power Syst Res 2008;78:1385–95.
[31] Abookazemi K, Ahmad H, Tavakolpour A, Hassan MY. Unit commitment solution using an optimized genetic system. Int J Electr Power Energy Syst 2011;33(4):969–75.
[32] Benhamida F, Abdelbar B. Enhanced lagrangian relaxation solution to the generation scheduling problem. Int J Electr Power Energy Syst 2010;32(10):1099–105.
[33] Chung CY, Yu H, Wong KP. An advanced quantum-inspired evolutionary algorithm for unit commitment. IEEE Trans Power Syst 2011;26(2):847–54.
[34] Ebrahimi J, Hosseinian SH, Gharehpetian GB. Unit commitment problem solution using shuffled frog leaping algorithm. IEEE Trans Power Syst 2011;26(2):573–81.
[35] Chandram K, Subrahmanyam N, Sydulu M. Unit commitment by improved pre-prepared power demand table and Muller method. Int J Electr Power Energy Syst 2011;33(1):106–14.