PWP-046

SOLVING UNIT COMMITMENT BY A UNIT DECOMMITMENT METHOD

Chung-Li Tseng, Chao-an Li, Shmuel S. Oren

February 1997

Solving Unit Commitment by a Unit Decommitment Method

Chung-Li Tseng*, Chao-an Li†, Shmuel S. Oren‡

February 10, 1997

Abstract

In this paper, we present an efficient and robust method for solving unit commitment problem using a unit decommitment method.

1 Introduction

A problem that must be frequently solved by a power utility is to economically determine a schedule of what units will be used to meet the forecast demand, and operating constraints such as spinning reserve requirements, over a short time horizon. This problem is commonly referred to as the unit commitment (UC) problem. The UC problem is a mixed integer programming problem, and is in the class of NP-hard problems [11]. Many optimization methods have been proposed to solve the UC problem (e.g. [4]). Among them, the Lagrangian relaxation (LR) methods [1, 4, 5] are the most widely used approaches for solving such a problem.

The LR approaches, though popular, are known to require many heuristics which strongly influence their performance [4, 13]. In this paper, we present an alternative method for solving the UC problem, which can be regarded as an approximate implementation of the LR approach. The basic idea is to employ the unit decommitment (UD) method proposed in [10, 7]. This method starts with a solution having all available units

*Edison Source, 13191 Crossroads Parkway North, Suite 405, City of Industry, CA 91746

†Consultant to Pacific Gas & Electric Company, San Francisco, CA 94177

‡Dept. of Industrial Engineering & Operations Research, University of California, Berkeley, CA 94720

on-line at all hours in the planning horizon and determines an optimal strategy for decommitting units, one at a time. We show that the number of iterations required by the method is bounded by the number of units. Empirical tests suggest that the proposed method is more efficient and robust than the LR approach.

This paper is organized as follows. In Section 2, the UC problem is formulated. Section 3 generalizes some important properties of economic dispatch. We generalize the UD method in Section 4. Section 5 presents the algorithm for solving UC using the UD method. The relation between the proposed method and the LR approach is discussed in Section 6. Finally we generate random instances of the UC problems and solve them by the proposed method. The numerical testing result and conclusion are given in Section 7.

2 Problem formulation

In this paper the following standard notation will be used. Additional symbols will be introduced when necessary.

i : index for the number of units $(i = 1, \cdots, I)$

t : index for time $(t = 0, \cdots, T)$

u_{it} : zero-one decision variable indicating whether unit i is up or down in time period t

x_{it} : state variable indicating the length of time that unit i has been up or down in time period t

t_i^{on} (t_i^{off}) : the minimum number of periods unit i must remain on (off) after it has been turned on (off)

p_{it} : state variable indicating the amount of power unit i is generating in time period t

$p_i^{\min}$ $(p_i^{\max})$: minimum (maximum) rated capacity of unit i

$r_i^{\max}$: maximum reserve for unit i

$r_i(p_{it})$: reserve available from unit i in time period t $(= \min(r_i^{\max}, p_i^{\max} - p_{it}))$

$C_i(p_{it})$: fuel cost for operating unit i at output level p_{it} in time period t

$S_i(x_{i,t-1}, u_{it}, u_{i,t-1})$: startup cost associated with turning on unit i at the beginning of time period t

D_t : forecast demand in time period t

R_t : spinning reserve requirement in time period t

The unit commitment problem is formulated as the following mixed-integer programming problem: (the underlined variables denote vectors, e.g. $\underline{u} = (u_{11}, \cdots, u_{IT})$.)

$$\min_{\underline{u},\underline{x},\underline{p}} \sum_{t=1}^{T} \sum_{i=1}^{I} [C_i(p_{it})u_{it} + S_i(x_{i,t-1}, u_{it}, u_{i,t-1})] \tag{2.1}$$

subject to the demand constraints,

$$\sum_{i=1}^{I} p_{it}u_{it} = D_t, \quad t = 1, \cdots, T, \tag{2.2}$$

and the spinning reserve constraints,

$$\sum_{i=1}^{I} r_i(p_{it})u_{it} \geq R_t, \quad t = 1, \cdots, T. \tag{2.3}$$

There are other unit constraints such as unit capacity constraints,

$$p_i^{\min} \leq p_{it} \leq p_i^{\max}, \; i = 1, \cdots, I; \; t = 1, \cdots, T, \tag{2.4}$$

the state transition equation for $i = 1, \cdots, I$,

$$x_{it} = \begin{cases} \max(x_{i,t-1}, 0) + 1, & \text{if } u_{it} = 1, \\ \min(x_{i,t-1}, 0) - 1, & \text{if } u_{it} = 0, \end{cases} \tag{2.5}$$

the minimum up/down time constraints for $i = 1, \cdots, I$,

$$u_{it} = \begin{cases} 1, & \text{if } 1 \leq x_{i,t-1} < t_i^{\text{on}}, \\ 0, & \text{if } -1 \geq x_{i,t-1} > -t_i^{\text{off}}, \\ 0 \text{ or } 1, & \text{otherwise}, \end{cases} \tag{2.6}$$

and the initial conditions on x_{it} at $t = 0$ for $\forall i$.

2.1 Model of cost functions

The generating cost of a thermal unit includes fuel costs and the startup costs. In this paper, the fuel cost C_i of a unit, say i, is assumed to be a smooth and strictly convex function of the power output(MWh) of the unit. The startup costs $S_i(x_{i,t-1}, u_{it}, u_{i,t-1})$ vary with the temperature of the boiler and therefore depend on the length of time that the unit has been off. The longer a unit remains off, the greater will be the cost to re-start that unit. To further simplify the notation, we let $S_i(\underline{u}, t) = S_i(x_{i,t-1}, u_{it}, u_{i,t-1})$.

3 Reserve-constrained economic dispatch

Given a known commitment $\underline{\tilde{u}} = \{\tilde{u}_{it}\}$ satisfying (2.5) and (2.6), economic dispatch (ED) is a problem of allocating system demand among all on-line generating units while satisfying (2.2), (2.3) and (2.4) at any time over the planning horizon, i.e. to determine the corresponding $\underline{\tilde{p}} = \{\tilde{p}_{it}\}$. (In this paper, variables denoted with a *tilde* hat denote a fixed realization of the corresponding variables.) If the spinning reserve constraints (2.3) are not considered, the ED problem is a conventional resource allocation (RA) problem (e.g. [6]), which has the form:

$$\min\{\sum_i C_i(p_i) | \sum_i p_i = D;\ p_i^{\min} \le p_i \le p_i^{\max}\}. \tag{3.7}$$

Optimality of such a RA-type ED problem requires that all generators operate at a marginal cost that either equals same fixed value Lambda (Lagrange multiplier) or equals the marginal cost corresponding to the upper or lower bound of a generator's output level, whichever is closer to Lambda. This property is commonly referred to as the 'equal-Lambda' rule (e.g. [12]). We use the term 'an equal-Lambda method' to refer to a method which solves the RA-type ED problem and presents the solution as well as the Lambda. An equal-Lambda method can be implemented very efficiently such that it obtains the optimal solution within strongly polynomial time if C_i are quadratic convex functions [2].

With the presence of the reserve constraints (2.3), the problem becomes a reserve-constrained economic dispatch (RCED). Methods for obtaining approximate solutions for RCED has been proposed, e.g. [8, 9]. In this section, we shall present an efficient algorithm for obtaining the optimal solution for RCED. Note that the RCED problem is separable in time, it can be solved sequentially by hour t. Define the index set of on-line units at time t with respect to this feasible commitment $J(t; \underline{\tilde{u}}) \equiv \{i | \tilde{u}_{it} = 1\}$. For simplicity, $\tilde{J}_t = J(t; \underline{\tilde{u}})$. The RCED problem in time t is denoted by

$$rced(\tilde{J}_t, t) \equiv \min_{p_i^{\min} \le p_{it} \le p_i^{\max}} \{\sum_{i \in \tilde{J}_t} C_i(p_{it}) | \sum_{i \in \tilde{J}_t} p_{it} = D_t; \sum_{i \in \tilde{J}_t} r_{it}(p_{it}) \ge R_t\}, \forall t. \tag{3.8}$$

Also assume $\underline{\tilde{p}} = \{\tilde{p}_{it}\}$ solves $rced(\tilde{J}_t, t)$, if the solution exists.

Proposition 1 The solution of $rced(\tilde{J}_t, t)$ exists, *if and only if*, the following conditions hold.

$$\sum_{i \in \tilde{J}_t} p_i^{\min} \le D_t \le \sum_{i \in \tilde{J}_t} p_i^{\max}, \tag{3.9a}$$

$$\sum_{i\in\tilde{J}_t} r_i^{\max} \geq R_t, \tag{3.9b}$$

and

$$\sum_{i\in\tilde{J}_t} p_i^{\max} \geq D_t + R_t. \tag{3.9c}$$

Proof. The *if* part is obvious. To show the *only if* part, note that (3.9b) implies that there exists $\{\tilde{r}_i\}$ such that $\sum_{i\in\tilde{J}_t}\tilde{r}_i = R_t$, $0 \leq \tilde{r}_i \leq r_i^{\max}$, $\forall i \in \tilde{J}_t$. Since $\sum_{i\in\tilde{J}_t} p_i^{\min} \leq D_t \leq \sum_{i\in\tilde{J}_t} p_i^{\max} - R_t = \sum_{i\in\tilde{J}_t}(p_i^{\max} - \tilde{r}_i)$, there exists $\{\tilde{p}_{it}\}$ such that $\sum_{i\in\tilde{J}_t}\tilde{p}_{it} = D_t$, and $p_i^{\min} \leq \tilde{p}_{it} \leq p_i^{\max} - \tilde{r}_i$, $\forall i \in \tilde{J}_t$. (Note: $\sum_{i\in\tilde{J}_t} r_i(\tilde{p}_{it}) \geq R_t$.) ■

Proposition 2 Assume $\{\tilde{p}_{it}; \tilde{u}_{it}\}$ is an optimal solution of RCED. Then there exist two mutually exclusive subsets of $\tilde{J}_t$, $\tilde{\Omega}_t$ and $\tilde{\Lambda}_t$, i.e. $\tilde{\Omega}_t \cup \tilde{\Lambda}_t = \tilde{J}_t$, and $\tilde{\Omega}_t \cap \tilde{\Lambda}_t = \emptyset$, and (Lagrange multipliers) $\tilde{\lambda}_t$, $\tilde{\alpha}_t$ and $\tilde{\mu}_t$ $t = 1, \cdots, T$, such that

$$\left.\begin{array}{ll} C_i'(\tilde{p}_{it}) = \tilde{\alpha}_t, & \text{for } p_i^{\max} - r_i^{\max} < \tilde{p}_{it} < p_i^{\max} \\ C_i'(\tilde{p}_{it}) \leq \tilde{\alpha}_t, & \text{for } \tilde{p}_{it} = p_i^{\max}, \end{array}\right\}, \forall i \in \tilde{\Omega}_t \tag{3.10}$$

$$\left.\begin{array}{ll} C_i'(\tilde{p}_{it}) = \tilde{\lambda}_t, & \text{for } p_i^{\min} < \tilde{p}_{it} < p_i^{\max} - r_i^{\max} \\ C_i'(\tilde{p}_{it}) \leq \tilde{\lambda}_t, & \text{for } \tilde{p}_{it} = p_i^{\max} - r_i^{\max} \\ C_i'(\tilde{p}_{it}) \geq \tilde{\lambda}_t, & \text{for } \tilde{p}_{it} = p_i^{\min} \end{array}\right\}, \forall i \in \tilde{\Lambda}_t \tag{3.11}$$

$$\tilde{\mu}_t\Big(\sum_{i\in\tilde{J}_t} \min(p_i^{\max} - \tilde{p}_{it}, r_i^{\max}) - R_t\Big) = 0 \tag{3.12}$$

$$\tilde{\mu}_t = \tilde{\lambda}_t - \tilde{\alpha}_t \tag{3.13}$$

$$\tilde{\lambda}_t \geq 0;\ \tilde{\alpha}_t \geq 0;\ \tilde{\mu}_t \geq 0. \tag{3.14}$$

for $\forall t$. ■

The proof of Proposition 2 is straightforward and is omitted here. An intuitive way to interpret the optimality condition is to divide the units into two categories: $\tilde{\Omega}_t$ is the set of units with 'cheap' reserve but 'expensive generation', and $\tilde{\Lambda}_t$ is the counterpart. Inspired by the proof of Proposition 1 and the optimality conditions above, we state the following algorithm for obtaining the optimal solution for RCED in time t, and its associated multipliers $\tilde{\lambda}_t$ and $\tilde{\mu}_t$ as defined in Proposition 2.

Algorithm for solving RCED

Step 1: Apply an equal-Lambda method to solve the reserve-unconstrained case:

$$\begin{aligned} \min_{p_{it}} \quad & \textstyle\sum_{i\in\tilde{J}_t} C_i(p_{it}) \\ \text{s.t.} \quad & \textstyle\sum_{i\in\tilde{J}_t} p_{it} = D_t \\ & p_i^{\min} \le p_{it} \le p_i^{\max}, \ \forall i \in \tilde{J}_t. \end{aligned}$$

Assume $\{\tilde{p}_{it}\}$ is the solution and the Lambda is denoted by $\tilde{\lambda}_t$. If $\sum_{i\in\tilde{J}_t} r_i(\tilde{p}_{it}) \ge R_t$, then $\tilde{\alpha}_t = \tilde{\lambda}_t$, stop and $\{\tilde{p}_{it}\}$ is also an optimal solution of RCED. Otherwise discard $\{\tilde{p}_{it}\}$ and go to Step 2.

Step 2: Apply an equal-Lambda method to solve the following RA problem:

$$\begin{aligned} \min_{p_{it}} \quad & \textstyle\sum_{i\in\tilde{J}_t} C_i(p_{it}) \\ \text{s.t.} \quad & \textstyle\sum_{i\in\tilde{J}_t} (p_i^{\max} - p_{it}) = R_t \\ & p_i^{\max} - r_i^{\max} \le p_{it} \le p_i^{\max}, \ \forall i \in \tilde{J}_t. \end{aligned}$$

Assume $\{\tilde{p}_{it}\}$ is the solution, and the Lambda is denoted by $\tilde{\alpha}_t$. Let $\tilde{\Omega}_t = \{i | p_i^{\max} - r_i^{\max} < \tilde{p}_{it}\}$ and $\tilde{\Lambda}_t = \tilde{J}_t \backslash \tilde{\Omega}_t$.

Step 3: Apply an equal-Lambda method to solve the following RA problem:

$$\begin{aligned} \min_{p_{it}} \quad & \textstyle\sum_{i\in\tilde{\Lambda}_t} C_i(p_{it}) \\ \text{s.t.} \quad & \textstyle\sum_{i\in\tilde{\Lambda}_t} p_{it} = D_t - \sum_{i\in\tilde{\Omega}_t} p_{it} \\ & p_i^{\min} \le p_{it} \le p_i^{\max} - r_i^{\max}, \ \forall i \in \tilde{\Lambda}_t \end{aligned}$$

Assume the solution is $\hat{p}_{it}$, $i \in \tilde{\Lambda}_t$, and the Lambda is denoted by $\tilde{\lambda}_t$.

Step 4: Let $\tilde{p}_{it} \leftarrow \hat{p}_{it}$, $i \in \tilde{\Lambda}$. $\{\tilde{p}_{it}\}$ is the solution of RCED. ■

It can be verified that the $\{\tilde{p}_{it}\}$ as well as $\tilde{\lambda}_t$ and $\tilde{\mu}_t$ satisfy the optimality condition stated in Proposition 2.

4 A unit decommitment method

A UD method was proposed in [10, 11] as a post-processing method to improve solution quality of existing UC algorithms. Given a feasible schedule $(\underline{\tilde{u}}, \underline{\tilde{p}})$, the UD method determines an optimal strategy, based on dynamic programming, for decommitting over-committed units. To start the UD method, a feasible solution has to be given. While in theory obtaining a feasible solution of the UC problem is an NP-hard problem [11], it is a relatively easy task in real world instances of that problem. Methods based on priority lists to sequentially commit units (e.g. [3]) can be used to construct an initial feasible solution. Since the purpose of this paper is to develop a UC algorithm, and the UD method plays a central part of the algorithm to be proposed, we shall defer the discussion of how to obtain an initial feasible solution to a later section. In this section, we have only included results which are either new, provide insight, or are simpler than previously published in [10].

Given a feasible schedule $(\tilde{\underline{u}}, \tilde{\underline{p}})$ (assumed economically dispatched), consider the problem (P_j^*) of optimally decommitting a unit, say unit j, with other units' commitments fixed. The formulation is as follows.

$$(P_j^*) \quad \min_{p_{it,i\neq j},u_{jt}} \quad \textstyle\sum_{t=1}^{T}[C_j(\tilde{p}_{jt})u_{jt} + S_j(\underline{u},t)] + \sum_{i\neq j}\sum_{t=1}^{T}[C_i(p_{it})\tilde{u}_{it} + S_i(\tilde{\underline{u}},t)]$$

$$\text{s.t.} \quad \textstyle\sum_{i\neq j} p_{it}\tilde{u}_{it} + \tilde{p}_{jt}u_{jt} = D_t,\ \forall t \tag{4.15a}$$

$$\textstyle\sum_{i\neq j} r_i(p_{it})\tilde{u}_{it} + r_j(\tilde{p}_{jt})u_{jt} \geq R_t,\ \forall t \tag{4.15b}$$

$$u_{jt} = 0 \text{ if } \tilde{u}_{jt} = 0, \tag{4.15c}$$

and minimum up/down time constraints (2.6) and initial conditions for unit j.

Note that the variables in (P_j^*) are u_{jt} for $\forall t$ and p_{it} for $\forall i \neq j$, $\forall t$ as well, since decommitting unit j in hour t would result in a redistribution of its generated power $\tilde{p}_{jt}$ to other on-line units in hour t in order to satisfy (2.2) and (2.3). In this paper, the solution of (P_j^*) will be called a *tentative* commitment of unit j. To solve (P_j^*), we observe that if $u_{jt} = 1$, the optimal p_{it} for $\forall i \neq j$ can be determined by the economic dispatch $rced(\tilde{J}_t, t)$; if $u_{jt} = 0$ is feasible, the resultant objective value will be the economic dispatch over all units but j, i.e., $rced(\tilde{J}_t\backslash\{j\}, t)$. The startup cost of unit j is imposed whenever applicable. Based on this observation, (P_j^*) can be transformed to the following 0-1 integer programming problem:

$$\min_{u_{jt}\in\{0,1\}} \sum_{t=1}^{T}[(rced(\tilde{J}_t,t) + S_j(\underline{u},t))u_{jt} + rced(\tilde{J}_t\backslash\{j\},t)(1-u_{jt})] + \sum_{i\neq j}\sum_{t=1}^{T} S_i(\tilde{\underline{u}},t) \tag{4.16}$$

subject to the same constraints except that (4.15a) now can be removed.

Since the last term in (4.16) is a constant, it can be removed from the objective function without altering the solution. Furthermore, adding the same term, $C_j(\tilde{p}_{jt})\tilde{u}_{jt} - rced(\tilde{J}_t,t)$, to both (mutually exclusive) choices, u_{jt} and $(1-u_{jt})$ would not alter the optimal decision. Thus, hereafter we will consider the following equivalent problem (P_j) to improve the commitment of unit j.

$$(P_j) \quad \min_{u_{jt}\in\{0,1\}} \sum_{t=1}^{T}[(C_j(\tilde{p}_{jt}) + S_j(\underline{u},t))u_{jt} + \Delta C_j(\underline{\tilde{u}},\underline{\tilde{p}},t)(1-u_{jt})] \tag{4.17}$$

subject to

$$\Delta C_j(\underline{\tilde{u}},\underline{\tilde{p}},t) = rced(\tilde{J}_t\backslash\{j\},t) + C_j(\tilde{p}_{jt})\tilde{u}_{jt} - rced(\tilde{J}_t,t). \tag{4.18}$$

$$u_{jt} = \begin{cases} 0 & \text{if } \tilde{u}_{jt} = 0, \\ 1 & \text{if } \tilde{u}_{jt} = 1, \text{ and the removal of } j \text{ from } \tilde{J}_t \text{ would} \\ & \text{result in violation of (3.9a) to (3.9c).} \end{cases} \tag{4.19}$$

and the minimum uptime, downtime constraints and the initial conditions for unit j.

(P_j) is an integer programming problem and can be solved using dynamic programming. $\Delta C_j(\underline{\tilde{u}},\underline{\tilde{p}},t)$ in (4.18) denotes the increased fuel cost of all on-line units other than unit j due to its decommitment in hour t. Obtaining $\Delta C_j(\underline{\tilde{u}},\underline{\tilde{p}},t)$ involves solving two single-hour RCED's, $rced(\tilde{J}_t,t)$ and $rced(\tilde{J}_t\backslash\{j\},t)$.

Assume that solving $rced(\tilde{J}_t\backslash\{j\},t)$ yields generation level $\tilde{p}_{it} + \Delta\tilde{p}_{it}$ for unit $i \in \tilde{J}_t\backslash\{j\}$ such that $\sum_{i\in\tilde{J}_t\backslash\{j\}} \Delta\tilde{p}_{it} = \tilde{p}_{jt}$. Based the fact that all the fuel cost functions C_i are smooth and convex, and also from Proposition 2, we have

$$\begin{aligned} \Delta C_j(\underline{\tilde{u}},\underline{\tilde{p}},t) &= \sum_{i\in\tilde{J}_t\backslash\{j\}} C_i(\tilde{p}_{it}+\Delta\tilde{p}_{it}) - C_i(\tilde{p}_{it}) \\ &\approx \sum_{i\in\tilde{J}_t\backslash\{j\}} C_i'(\tilde{p}_{it})\Delta\tilde{p}_{it} \\ &\approx \tilde{\lambda}_t \sum_{i\in\tilde{\Lambda}_t\backslash\{j\}} \Delta\tilde{p}_{it} + (\tilde{\lambda}_t - \tilde{\mu}_t) \sum_{i\in\tilde{\Omega}_t\backslash\{j\}} \Delta\tilde{p}_{it} \\ &= \tilde{\lambda}_t\tilde{p}_{jt} - \tilde{\mu}_t \sum_{i\in\tilde{\Omega}_t\backslash\{j\}} \Delta\tilde{p}_{it} \\ &\approx \tilde{\lambda}_t\tilde{p}_{jt} - \tilde{\mu}_t\tilde{r}_{jt}. \end{aligned} \tag{4.20}$$

(Note that in (4.20) we use the fact that the loss of reserve originally provided by unit j, can only be made up by the units in $\tilde{\Omega}_t\backslash\{j\}$.)

In the sequel, we will approximate $\Delta C_j(\underline{\tilde{u}},\underline{\tilde{p}},t)$ by $\Delta C_{jt} \equiv \tilde{\lambda}_t\tilde{p}_{jt} - \tilde{\mu}_t\tilde{r}_{jt}$, and will only focus on the case where $\Delta C_j(\underline{\tilde{u}},\underline{\tilde{p}},t)$ is replaced by its 'first order approximation', $\tilde{\lambda}_t\tilde{p}_{jt} - \tilde{\mu}_t\tilde{r}_{jt}$.

In the following algorithm, superscript k denotes the k-th iteration of the algorithm. Let $\tilde{\Theta}_i^k$, $i = 1,\cdots,I$ be the total generating cost (fuel cost and startup cost) of unit i of the feasible schedule $(\underline{\tilde{u}}^k,\underline{\tilde{p}}^k)$; and Θ_i^k, $i = 1,\cdots,I$, the optimal objective value of (P_i^k) solved with respect to feasible solution $(\underline{\tilde{u}}^k,\underline{\tilde{p}}^k)$. We now state the decommitment algorithm.

The UD algorithm

Data: Feasible solution $(\underline{\tilde{u}}^0, \underline{\tilde{p}}^0)$ and the corresponding $\tilde{\Theta}_i^0$, $i = 1, \cdots, I$ are given.

Step 0: $k \leftarrow 0$.

Step 1: Solve (P_i^k) with respect to $(\underline{\tilde{u}}^k, \underline{\tilde{p}}^k)$ and obtain Θ_i^k for all $i = 1, \cdots, I$.

Step 2: Select a unit m such that $(\tilde{\Theta}_m^k - \Theta_m^k) > 0$. If there is no such a unit, stop; otherwise update the commitment of unit m in $\underline{\tilde{u}}^k$ by the tentative commitment obtained in (P_m^k). The resultant unit commitment is assigned to be $\underline{\tilde{u}}^{k+1}$.

Step 3: Perform the RCED on $\underline{\tilde{u}}^{k+1}$ to obtain $\underline{\tilde{p}}^{k+1}$ and evaluate $\tilde{\Theta}_i^{k+1}$, the total generating cost of unit i, $i = 1, \cdots, I$.

Step 4: $k \leftarrow k + 1$, go to Step 1. ■

The algorithm chooses the tentative commitment which can yield savings to replace the original commitment. The rule for selecting a unit to be improved in Step 2 (corresponding to choosing a descent direction in continuous optimization) is not unique. In our implementation, we considered two selection rules.

1. $m = \arg\max_i\{\tilde{\Theta}_i^k - \Theta_i^k\}$ – select the unit which can yield the most savings.
2. $m = \arg\max_i\{(\tilde{\Theta}_i^k - \Theta_i^k)/\sum_{t=1}^{T}|\tilde{u}_{it}^k - \tilde{u}_{it}^{k+1}|\}$ – select the unit which can yield the most savings per decommitted power.

Generally, we found that the second selection rule outperforms the first one.

Suppose unit j is selected at iteration k', and its tentative commitment $\{\hat{u}_{jt}\}$ will be the same as $\{\tilde{u}_{jt}^{k'+1}\}$, so

$$\sum_{t=1}^{T}[(C_j(\tilde{p}_{jt}^{k'}) + S_j(\underline{\hat{u}}, t))\hat{u}_{jt} + \Delta C_{jt}^{k'} \cdot (1 - \hat{u}_{jt})] \leq$$

$$\sum_{t=1}^{T}[(C_j(\tilde{p}_{jt}^{k'}) + S_j(\underline{\tilde{u}}, t))\tilde{u}_{jt} + \Delta C_{jt}^{k'} \cdot (1 - \tilde{u}_{jt})], \tag{4.21}$$

for any $\{\tilde{u}_{jt}\}$ satisfying (4.19). Let $\Gamma = \{t|\tilde{u}_{jt} \neq \hat{u}_{jt}, t = 1, \cdots, T\}$, (4.21) is reduced to

$$\sum_{t\in\Gamma} C_j(\tilde{p}_{jt}^{k'}) + (\sum_{t=1}^{T}(S_j(\underline{\hat{u}}, t)\hat{u}_{jt} - S_j(\underline{\tilde{u}}, t)\tilde{u}_{jt}) \leq \sum_{t\in\Gamma}\Delta C_{jt}^{k'}. \tag{4.22}$$

It can be shown that for some on-line unit j in time t that before it is decommitted, both $\{C(\tilde{p}_{jt}^k)\}$ and $\{\Delta C_{jt}^k\} = \{\tilde{\lambda}_t^k \tilde{p}_{jt}^k - \tilde{\mu}_t^k \tilde{r}_{jt}^k\}$ are nondecreasing in k. Furthermore, it can be shown that $\{C(\tilde{p}_{jt}^k)\}$ increases faster than it first approximation $\{\Delta C_{jt}^k\}$ because of convexity. This implies that (4.22) should continue to hold for all $k \geq k'$. That is, unit j should not be selected again in Step 2 at any subsequent iteration of iteration k'. We conclude that with $\Delta C_j(\underline{\tilde{u}}, \underline{\tilde{p}}, t)$ approximated by $\tilde{\lambda}_t \tilde{p}_{jt} - \tilde{\mu}_t \tilde{r}_{jt}$, the UD algorithm terminates within I iterations, where I is the number of the units.

5 Solving UC Using UD

A natural question is whether it is possible to use a UD approach as a direct primal solution technique for solving the unit commitment problem. An intuitive approach is to initially turn on as many units as possible in all hours without violating the minimum up/down time constraints. A schematic algorithm for implementing the outlined procedure is given below.

The UC algorithm

Step 0: $\tilde{u}_{i0}$ and $\tilde{x}_{i0}$ are given for $\forall i$; $i \leftarrow 1$; $t \leftarrow 1$.

Step 1: If $i > I$, stop and $\underline{\tilde{u}}$ is the initial commitment. Go to Step 4.

Step 2: If $t > T$, then $i \leftarrow i+1$ and go to Step 1. Otherwise,

$$\tilde{u}_{it} = \begin{cases} 1 & \text{if } \tilde{x}_{i,t-1} \geq -t_i^{\text{off}} \\ 0 & \text{otherwise} \end{cases} \tag{5.23}$$

$$\tilde{x}_{it} = \begin{cases} \max(\tilde{x}_{i,t-1}, 0) + 1, & \text{if } \tilde{u}_{it} = 1, \\ \min(\tilde{x}_{i,t-1}, 0) - 1, & \text{if } \tilde{u}_{it} = 0. \end{cases} \tag{5.24}$$

Step 3: $t \leftarrow t+1$, go to Step 2.

Step 4: Apply the RCED with respect to $\underline{\tilde{u}}$ to obtain $\underline{\tilde{p}}$.

Step 5: Apply the UD algorithm with respect to $(\underline{\tilde{u}}, \underline{\tilde{p}})$. ■

Solving UC by means of UD requires finding first an initial feasible solution $(\underline{\tilde{u}}, \underline{\tilde{p}})$. Initially when all units are committed, such a commitment tend to violate (3.9a), i.e. the so called minimum load conditions. In other words, the loop between Step 1 and Step 3 does not necessarily result in an initial feasible schedule as required by the UD algorithm, i.e., the RCED phase in Step 4 may not be feasible. Since the RCED is a subroutine required at each iteration of the UD algorithm stated in Step 5 of the above algorithm, we need to extend the RCED subroutine to also handle cases where the minimum load conditions are not satisfied. A possible modification is to dispatch the on-line generators so as to equalize the marginal costs to the extent possible, even if the minimum load conditions are not satisfied. That is, when

$$\sum_{i \in \tilde{J}_t} p_i^{\min} > D_t, \tag{5.25}$$

all on-line units are dispatched to their minimum capacities respectively,

$$\tilde{p}_{it} \leftarrow p_i^{\min}, \ \forall i, \tag{5.26}$$

and the corresponding Lambda is the minimum of the marginal costs of the corresponding dispatches in (5.26),

$$\tilde{\lambda}_t = \tilde{\mu}_t \leftarrow \min_{i \in \tilde{J}_t} C'(p_i^{\min}); \ \tilde{\alpha}_t \leftarrow 0. \tag{5.27}$$

Such a modification of the RCED phase above is based on the expectation that as the decommitment procedure proceeds, the commitment obtained will eventually satisfy the minimum load conditions, thus producing a feasible schedule. From a theoretical perspective, determining a feasible solution of the UC problems is NP-hard as mentioned. However, in extensive numerical tests, we have found that the above approach worked satisfactorily. In all observed cases, the UD method performed well as a UC algorithm and obtained feasible solutions. Furthermore, we have the following theorem.

Theorem 1 With the modification in (5.27) of the RCED phase, the UC algorithm terminates within I iterations, where I is the number of units.

Proof. One only needs to verify that the modification in (5.27) does not change the fact that both $\{C(\tilde{p}_{jt}^k)\}$ and $\{\Delta C_{jt}^k\} = \{\tilde{\lambda}_t^k \tilde{p}_{jt}^k - \tilde{\mu}_t^k \tilde{r}_{jt}^k\}$ are nondecreasing in k, and $\{C(\tilde{p}_{jt}^k)\}$ increases faster than $\{\Delta C_{jt}^k\}$. ∎

6 UD is a LR-like method

In this section, we present an intuitive discussion hoping to shed some light on the relationship between the UD method and the LR method for solving the UC problem. Let λ_t and μ_t $(t = 1, \cdots, T)$ be the corresponding nonnegative Lagrange multipliers to (2.2) and (2.3). Conventional LR approaches solve the following dual problem (D):

$$(D) \ \max_{\underline{\lambda}, \underline{\mu} \geq 0} d(\underline{\lambda}, \underline{\mu}), \tag{6.28}$$

where

$$d(\lambda_t, \mu_t) \ = \ \min_{\lambda_t, \mu_t \geq 0} \sum_{i=1}^{I} \sum_{t=1}^{T} [C_i(p_{it}) u_{it} + S_i(\underline{u}, t)$$

$$+ \lambda_t(D_t - \sum_{i=1}^{I} p_{it}u_{it}) + \mu_t(R_t - \sum_{i=1}^{I} r_i(p_{it})u_{it})]$$

$$= \sum_{i=1}^{I} d_i(\lambda_t, \mu_t) + \sum_{t=1}^{T}(\lambda_t D_t + \mu_t R_t), \tag{6.29}$$

and

$$d_i(\lambda_t, \mu_t) = \min_{u_{it}, p_{it}} \sum_{t=1}^{T} [C_i(p_{it})u_{it} + S_i(\underline{u}, t) - \lambda_t p_{it} u_{it} - \mu_t r_i(p_{it})u_{it}]. \tag{6.30}$$

The minimization problem (6.30) is subject to (2.4) to (2.6) and the initial conditions.

It is straightforward to show that each decommitment problem (P_j) and the dual subproblem with $i = j$, $d_j(\tilde{\lambda}_t, \tilde{\mu}_t)$, are equivalent in the sense that given $\{\tilde{\lambda}_t\}$, $\{\tilde{\mu}_t\}$ and $\{\tilde{p}_{jt}; \tilde{u}_{jt}\}$, if $\{\hat{u}_{jt}\}$ solves (P_j), then $(\tilde{p}_{jt}, \hat{u}_{jt})$ also solves $d_j(\tilde{\lambda}_t, \tilde{\mu}_t)$. Therefore, the UD method is a LR-like method. The differences are that the multipliers λ_t and μ_t are taken from RCED phase rather than updated by the subgradient iteration, and that UD is a primal method which maintains primal feasibility once achieved.

7 Numerical results and conclusions

We conduct numerical tests to compare the performance of UD and LR. All algorithms are implemented in FORTRAN on a HP 700 workstation. Four cases of systems with combinations of 10, 30 units, and 24, 168 hours of planning horizon are tested. For each case, we randomly generate 100 instances of the UC problem. (Detailed configuration of the random instances are available upon request to the authors.) Each instance is solved by LR and UD of selection rules 1 and 2, under the columns of LR, UD1 and UD2, respectively. The column under D.G. records the duality gap of the LR approach in terms of the percentage of the dual value. Since the comparison are made with regard to the value of the LR approach, the columns under LR are all "ones". Also the two numbers in a parenthesis define a range of the sample points. The mean of the sample points is recorded on the top of the corresponding parentheses. The test results including solution qualities and CPU times required for both methods are summarized in Table 1.

From Table 1, the error between LR and UD is within 0.2%, and the UD methods takes much less (save at least 50%) CPU time than the LR approach. Besides, the only heuristic in UD is the unit selection rule, which is analogous to choosing a descent direction in continuous optimization. We test for two selection rules, and both are equally good. To sum up, the numerical testing results show that UD serves as a reliable, efficient and robust approximate method of the LR approaches for solving UC.

Table 1: Comparison of UD and LR

Case	Solution Quality				CPU Time		
	LR	UD1	UD2	D.G. (%)	LR	UD1	UD2
10×24	1	1.0010 (0.9933-1.0105)	1.0007 (0.9933-1.0086)	0.9 (0.09-2.85)	1	0.2185 (0.1028-0.4773)	0.1784 (0.1056-0.3636)
10×168	1	1.0008 (0.9976-1.0054)	1.0004 (0.9973-1.0053)	0.9 (0.38-2.04)	1	0.1500 (0.0978-0.3446)	0.0978 (0.0661-0.2953)
30×24	1	1.0013 (0.9981-1.0090)	1.0007 (0.9986-1.0048)	0.28 (0.06-0.81)	1	0.5214 (0.3344-0.8608)	0.4183 (0.2762-0.7371)
30×168	1	1.0017 (0.9997-1.0058)	1.0007 (0.9994-1.0025)	0.35 (0.15-1.78)	1	0.2745 (0.1513-0.4152)	0.3215 (0.1597-0.4886)

8 Acknowledgment

This work was partly supported by the National Science Foundation under Grant IRI-9120074. This support is greatly appreciated.

References

[1] D. P. Bertsekas, G. S. Lauer, N. R. Sandell, Jr., and T. A. Posbergh, "Optimal Short-Term Scheduling of Large-Scale Power Systems," *IEEE Trans. Automatic Control*, vol. AC-28, no. 1, pp. 1-11, 1983.

[2] P. Brucker, "An $O(n)$ algorithm for quadratic knapsack problems", *Operations Research Letter*, vol. 3, no. 3, pp. 163-166, August 1984.

[3] W. G. Chandler, P. L. Dandeno, A. F. Gilmn and L. K. Kirchmayer, "Short-range operation of a combined thermal and hydroelectric power system", *AIEE, Transactions*, paper no. 53-326, 1953.

[4] A. Cohen and V. Sherkat, "Optimization-Based Methods for Operations Scheduling," *Proceedings of IEEE*, vol. 75, no. 12, pp. 1574-1591, 1987.

[5] X. Guan, P. B. Luh, H. Yan and J. A. Amalfi, "An Optimization-Based Method for Unit Commitment," *Electrical Power & Energy Systems*, vol. 14, no. 1, pp. 9-17, February 1992.

[6] T. Ibaraki and N. Katoh, *Resource Allocation Problems : Algorithmic Approaches*, MIT Press, 1988.

[7] C. A. Li, R. B. Johnson, and A. J. Svoboda, "A New Unit Commitment Method", presented at the IEEE/PES 1995 Winter meeting.

[8] G. B. Sheble, "Real-time Economic Dispatch and Reserve Allocation Using Merit Order Loading and Linear Programming Rules", *IEEE Trans. Power System*, vol. 4, no. 4, pp. 1414-1420, 1989.

[9] W. O. Stadlin, "Economic allocation of regulating margin", *IEEE Trans. Power Apparatus and Systems*, vol. PAS-89, no. 4, pp. 1776-1781, July/Aug. 1971.

[10] C. L. Tseng, S. S. Oren, A. J. Svoboda, and R. B. Johnson, "A Unit Decommitment Method in Power System Scheduling", to appear in *Electrical Power & Energy Systems.*

[11] C. L. Tseng, "On Power System Generation Unit Commitment Problems", Ph.D. Dissertation, Department of Industrial Engineering and Operations Research, University of California at Berkeley, Berkeley, December 1996.

[12] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation and Control*, John Wiley & Sons, New York, 1984.

[13] F. Zhuang, "Optimal generation unit commitment in thermal electric power systems", Ph.D. Thesis, McGill University, December 1988.