JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 112, No. 2, pp. 295–314, February 2002 (©2002)

Unit Commitment by Augmented Lagrangian Relaxation: Testing Two Decomposition Approaches[1,2]

C. Beltran[3] and F. J. Heredia[4]

Communicated by M. Simaan

Abstract. One of the main drawbacks of the augmented Lagrangian relaxation method is that the quadratic term introduced by the augmented Lagrangian is not separable. We compare empirically and theoretically two methods designed to cope with the nonseparability of the Lagrangian function: the auxiliary problem principle method and the block coordinated descent method. Also, we use the so-called unit commitment problem to test both methods. The objective of the unit commitment problem is to optimize the electricity production and distribution, considering a short-term planning horizon.

Key Words. Augmented Lagrangian relaxation, auxiliary problem principle, block coordinate descent, classical Lagrangian relaxation, unit commitment, variable duplication.

1. Introduction

The problem dealt with here is called the unit commitment (UC) problem. The objective of the UC problem is to optimize electricity production and distribution, considering a short-term planning horizon (from one day to one week). Hydroelectric and thermal plants must be coordinated in order to satisfy the customer demand of electricity at minimum cost and with a reliable service. Some examples are Refs. 1–5.

[1] This research was supported by CICYT, Grant TAP96-1044.

[2] The authors would like to acknowledge the anonymous referees for their exhaustive and challenging revisions.

[3] PhD Student, Department of Statistics and Operations Research, Universitat Politècnica de Catalunya, Barcelona, Spain.

[4] Professor, Department of Statistics and Operations Research, Universitat Politècnica de Catalunya, Barcelona, Spain.

Nowadays, the Lagrangian relaxation (LR) method is the most widespread procedure for solving the UC problem. Within the LR method, there are two main approaches: the classical Lagrangian relaxation (CLR) method (Refs. 1, 6, 7) and the augmented Lagrangian relaxation (ALR) method (Refs. 5, 8, 9). The solution of the UC problem by the CLR method yields usually an infeasible primal solution due to the duality gap. One advantage of the ALR method over the CLR method is that the augmented Lagrangian may obtain a feasible primal solution in cases where the classical Lagrangian presents a duality gap; see page 279 in Ref. 10 or page 733 in Ref. 4.

Another advantage of the ALR method is that the dual function associated to the augmented Lagrangian is differentiable in cases where the CLR method presents a nondifferentiable dual function; see Ref. 11 or page 352 in Ref. 12. However, when solving nonconvex problems, as it is the case with the UC problem, the ALR method may obtain a local optimizer (see page 409 in Ref. 13 or page 228 in Ref. 14) without measure of its quality. In this case, the dual bound given by the CLR method can be used as a quality measure of a UC solution. Lai and Baldick (Ref. 15) compared both methods computationally, finding that in general ALR was slower, but providing feasible solutions, while CLR needed as usual a heuristic postprocess to achieve primal feasibility.

Although almost all the LR procedures used to solve the UC problem are based on either the CLR method or the ALR method, in Ref. 4 we find a two-phase approach. In the first phase, the authors use the CLR method to obtain the dual optimum. In the second phase, the authors use the ALR method to obtain a (local) optimal solution whose quality is assessed by the dual optimum obtained in the first phase. We also believe that this two-phase approach is advantageous; therefore, we see the ALR method as an approach to be used together with the CLR method within a two-phase frame.

One of the main drawbacks of the ALR method, apart from obtaining a local optimizer, is that the quadratic term introduced by the augmented Lagrangian is not separable. The most widely used method to overcome this problem is the auxiliary problem principle (APP) method (Ref. 16). In this work, we propose the block coordinate descent (BCD) method (Ref. 12, page 246) as an alternative decomposition approach for ALR. The aim of this paper is to compare, theoretically and practically, these two decomposition approaches when solving the UC problem.

Our starting point is the paper by Batut and Renaud (Ref. 8); therefore, we use variable duplication plus the augmented Lagrangian relaxation (ALR) method. However, the conservative APP method used in Ref. 8 is replaced by the BCD method, which proves to be faster, without any significant loss of accuracy. Several large-scale instances of the UC problem have been solved, showing the applicability of the proposed procedure.

This paper is divided into two parts. In the first part, a simple example is used to introduce the UC problem, the solution methodology, and the theoretical foundation. In the second part, a large-scale realistic UC problem is presented and solved in order to compare the APP method and BCD method.

2. Unit Commitment: Simple Example

As we mentioned above, the unit commitment (UC) problem arises in the electrical engineering field. However, the techniques developed in this paper can be used to solve many other problems. First, we start with the following simple UC problem:

$$\min \quad 2x^2 + \mathrm{Con}(x) + 2y^2 + \mathrm{Con}(y), \tag{1a}$$

$$\text{s.t.} \quad x + y = 2, \tag{1b}$$

$$x \in \{0\} \cup [1, 2], \tag{1c}$$

$$y \in \{0\} \cup [1, 2]. \tag{1d}$$

In this example, there are two thermal units: x stands for the production of the first thermal unit and y for the production of the second thermal unit. We must produce 2 MW of electrical power [see Eq. (1b)] at the lowest cost. The production cost increases quadratically, and we must also pay an extra starting cost whenever a unit is turned on

$$\mathrm{Con}(x) := \begin{cases} 1, & \text{if } x \in [1, 2], \\ 0, & \text{if } x = 0, \end{cases} \tag{2}$$

$$\mathrm{Con}(y) := \begin{cases} 2, & \text{if } y \in [1, 2], \\ 0, & \text{if } y = 0. \end{cases} \tag{3}$$

The feasible set of problem (1) is represented in Fig. 1. This disconnected feasible set is usually modeled by means of binary variables which represent the on/off state of the thermal units; see for example Ref. 2. An equivalent binary variable formulation of problem (1) would be

$$\min \quad 2x^2 + 1u_x + 2y^2 + 2u_y, \tag{4a}$$

$$\text{s.t.} \quad x + y = 2, \tag{4b}$$

$$u_x 1 \leq x \leq u_x 2, \tag{4c}$$

$$u_y 1 \leq y \leq u_y 2, \tag{4d}$$

$$u_x, u_y \in \{0, 1\}. \tag{4e}$$

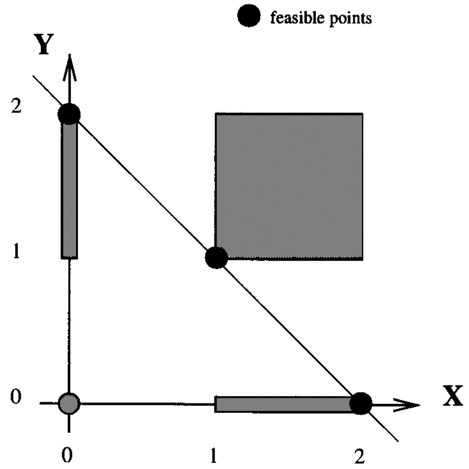

Fig. 1. Disconnected domain of the UC problem.

Following Ref. 8, we duplicate each variable in order to account, on the one hand, for the coupling constraint $x+y=2$ (a constraint that prevents the decomposition of the problem) and, on the other hand, for the disconnected set $(\{0\}\cup[1,2])^2 \subset R^2$. Obviously, problems (1) and (5) are equivalent regarding the optimal solution. We have

$$\min \quad x^2+y^2+\tilde{x}^2+\mathrm{Con}(\tilde{x})+\tilde{y}^2+\mathrm{Con}(\tilde{y}), \tag{5a}$$

$$\text{s.t.} \quad x+y=2, \tag{5b}$$

$$\tilde{x}\in\{0\}\cup[1,2], \tag{5c}$$

$$\tilde{y}\in\{0\}\cup[1,2], \tag{5d}$$

$$x=\tilde{x}, \qquad y=\tilde{y}. \tag{5e}$$

The previous duplicated variables problem can be recast in a more general form $[(x,y)\in R^2]$,

$$\min \quad f(x,y)+\tilde{f}(\tilde{x},\tilde{y}), \tag{6a}$$

$$\text{s.t.} \quad (x,y)\in\mathscr{D}, \tag{6b}$$

$$(\tilde{x},\tilde{y})\in\tilde{\mathscr{D}}, \tag{6c}$$

$$(x,y)=(\tilde{x},\tilde{y}), \tag{6d}$$

where $\mathscr{D}$ is the connected domain defined by the coupling constraints and $\tilde{\mathscr{D}}$ represents the disconnected domain. This is the primal problem to be solved.

2.1. Augmented Lagrangian Relaxation Method Our aim is to solve the following primal problem ($x \in R^n$, $\tilde{x} \in R^n$):

$$\min \quad f(x) + \tilde{f}(\tilde{x}), \tag{7a}$$

$$\text{s.t.} \quad x \in \mathcal{D}, \qquad \tilde{x} \in \tilde{\mathcal{D}}, \tag{7b}$$

$$x - \tilde{x} = 0. \tag{7c}$$

In order to decompose this problem into two subproblems, one in $\mathcal{D}$ and the other in $\tilde{\mathcal{D}}$, we relax the coupling constraint $x - \tilde{x} = 0$ in an augmented Lagrangian fashion.

This means adding the Lagrangian term $\lambda'(x - \tilde{x})$ and the penalty term $(c/2)\|x - \tilde{x}\|^2$ to the objective function $f(x) + \tilde{f}(\tilde{x})$. The resulting max–min problem is called the dual problem, that is,

$$\max_{\lambda \in R^n} \left\{ \min_{\substack{x \in \mathcal{D} \\ \tilde{x} \in \tilde{\mathcal{D}}}} f(x) + \tilde{f}(\tilde{x}) + \lambda'(x - \tilde{x}) + (c/2)\|x - \tilde{x}\|^2 \right\}. \tag{8}$$

As usual, the augmented Lagrangian is defined as

$$L_c(x, \tilde{x}, \lambda) := f(x) + \tilde{f}(\tilde{x}) + \lambda'(x - \tilde{x}) + (c/2)\|x - \tilde{x}\|^2, \tag{9}$$

and the associated dual function as

$$q_c(\lambda) := \left\{ \min_{\substack{x \in \mathcal{D} \\ \tilde{x} \in \tilde{\mathcal{D}}}} L_c(x, \tilde{x}, \lambda) \right\}. \tag{10}$$

Then, the dual problem can be recast as

$$\max_{\lambda \in R^n} q_c(\lambda). \tag{11}$$

Note that the augmented Lagrangian (9), unlike the plain Lagrangian, cannot be decomposed due to the quadratic penalty term $(c/2)\|x - \tilde{x}\|^2$.

2.2. Alternative Decomposition Methods. The focus of this paper is on the step at which the nonseparable augmented Lagrangian is minimized and how we decompose this problem into smaller subproblems,

$$\min_{\substack{x \in \mathcal{D} \\ \tilde{x} \in \tilde{\mathcal{D}}}} f(x) + \tilde{f}(\tilde{x}) + \lambda'(x - \tilde{x}) + (c/2)\|x - \tilde{x}\|^2. \tag{12}$$

The method used in Ref. 8 is the so-called auxiliary problem principle (APP) method. Roughly speaking, the APP method linearizes the quadratic term $(c/2)\|x - \tilde{x}\|^2$ of the augmented Lagrangian at the current iterate $(x_n, \tilde{x}_n)$ and adds a quadratic separable term

$$(b/2)(\|x - x_n\|^2 + \|\tilde{x} - \tilde{x}_n\|^2).$$

That is, we have

$$\min_{\substack{x\in\mathcal{D}\\ \tilde{x}\in\tilde{\mathcal{D}}}} f(x)+\tilde{f}(\tilde{x})+\lambda'(x-\tilde{x})$$

$$+\,c(x_n-\tilde{x}_n)'\,(x-\tilde{x})+(b/2)\,(\|x-x_n\|^2+\|\tilde{x}-\tilde{x}_n\|^2). \tag{13}$$

The chief utility of the added quadratic term,

$$(b/2)\,(\|x-x_n\|^2+\|\tilde{x}-\tilde{x}_n\|^2)=(b/2)\,\|(x,\tilde{x})'-(x_n,\tilde{x}_n)'\|^2,$$

is that it regularizes the problem (13) in the sense that it makes the (partially linearized) augmented Lagrangian strictly convex (Ref. 12, page 475). In the convex case, convergence of the APP method is guaranteed when $b\geq 2c$ (Ref. 16).

Then, the minimization of this approximation to L_c decomposes into two subproblems [$(x_{n+1},\tilde{x}_{n+1})$ is the new iterate],

$$x_{n+1}:=\arg\min_{x\in\mathcal{D}} f(x)+\lambda_n'x+c(x_n-\tilde{x}_n)'x+(b/2)\,\|x-x_n\|^2, \tag{14}$$

$$\tilde{x}_{n+1}:=\arg\min_{\tilde{x}\in\tilde{\mathcal{D}}} \tilde{f}(\tilde{x})-\lambda_n'\tilde{x}-c(x_n-\tilde{x}_n)'\,\tilde{x}+(b/2)\,\|\tilde{x}-\tilde{x}_n\|^2. \tag{15}$$

An alternative is to use the block coordinate descent (BCD) method, also called the nonlinear Gauss–Seidel method (Ref. 12, page 247). Unlike the APP method, which uses an approximation to L_c, the BCD method directly minimizes L_c. When minimizing in the domain $\mathcal{D}$, the variables in the domain $\tilde{\mathcal{D}}$ are frozen at their best known value, say $\tilde{x}_n$; analogously, when minimizing in $\tilde{\mathcal{D}}$, the variables in $\mathcal{D}$ are frozen at their best known value, say x_{n+1}; and so on. In the convex case, convergence is guaranteed; see Ref 12, page 246. Then, the minimization of L_c decomposes into two subproblems [$(x_{n+1},\tilde{x}_{n+1})$ is the new iterate],

$$x_{n+1}:=\arg\min_{x\in\mathcal{D}} f(x)+\lambda_n'x+(c/2)\,\|x-\tilde{x}_n\|^2, \tag{16}$$

$$\tilde{x}_{n+1}:=\arg\min_{\tilde{x}\in\tilde{\mathcal{D}}} \tilde{f}(\tilde{x})-\lambda'^{n}\tilde{x}+(c/2)\,\|x_{n+1}-\tilde{x}\|^2. \tag{17}$$

As we will see in Section 3.2, our modeling of subproblems (14) and (16) belongs to the class of nonlinear network flow problems with side constraints, which can be solved through specialized codes (Refs. 17–18). Subproblems (15) and (17), having a disconnected domain, are of combinatorial nature. We solve them by means of the dynamic programming. Full details on the solution of subproblems (14)–(17) can be found in Ref. 19.

2.3. Alternative Decomposition Algorithms. This section uses the well-known fact that the gradient of the dual function at the current dual iterate is equal to the relaxed constraint evaluated at the current primal iterate, i.e.,

$$\nabla q_{c_n}(\lambda_n) = (x_n - \tilde{x}_n);$$

see Ref. 12, page 352.

The augmented Lagrangian relaxation (ALR) method combined with the auxiliary problem principle (APP) is summarized in the following ALR + APP algorithm [the ALR algorithm is also called the multiplier method (Ref. 12, page 340)].

ALR + APP Algorithm.

Step 1. Check the Stopping Criterion. If the norm of the gradient of the dual function $\|x_n - \tilde{x}_n\| < \epsilon$, then stop. $(x_n, \tilde{x}_n, \lambda_n)$ is a primal–dual solution.

Step 2. Compute

$$x_{n+1} := \arg\min_{x \in \mathcal{D}} f(x) + \lambda_n' x + c_n(x_n - \tilde{x}_n)' x + (b_n/2)\|x - x_n\|^2.$$

Step 3. Compute

$$\tilde{x}_{n+1} := \arg\min_{\tilde{x} \in \tilde{\mathcal{D}}} \tilde{f}(\tilde{x}) - \lambda_n' \tilde{x} - c_n(x_n - \tilde{x}_n)' \tilde{x} + (b_n/2)\|\tilde{x} - \tilde{x}_n\|^2.$$

Step 4. Dual Variable Updating. Set

$$\lambda_{n+1} = \lambda_n + c_n(x_{n+1} - \tilde{x}_{n+1}).$$

Step 5. Penalty Parameter Updating. If

$$\|x_{n+1} - \tilde{x}_{n+1}\| > \alpha \cdot \|x_n - \tilde{x}_n\|,$$

then set

$$c_{n+1} := \beta c_n, \qquad b_{n+1} := \gamma c_{n+1}.$$

A suitable choice is $\alpha := 1.10$, $\beta := 2$, $\gamma := 2$.

Analogously, the ALR method combined with the block coordinate descent (BCD) method is summarized in the following ALR + BCD algorithm.

ALR + BCD Algorithm.

Step 1. Check the Stopping Criterion. If the norm of the gradient of the dual function $\|x_n - \tilde{x}_n\| < \epsilon$, then stop. $(x_n, \tilde{x}_n, \lambda_n)$ is a primal–dual solution.

Step 2. Compute

$$x_{n+1} := \arg\min_{x \in \mathcal{D}} f(x) + \lambda_n' x + (c_n/2)\|x - \tilde{x}_n\|^2.$$

Step 3. Compute

$$\tilde{x}_{n+1} := \arg\min_{\tilde{x}\in\tilde{\mathcal{D}}} \tilde{f}(\tilde{x}) - \lambda_n'\tilde{x} + (c_n/2)\|x_{n+1} - \tilde{x}\|^2.$$

Step 4. Dual Variable Updating. Set

$$\lambda_{n+1} = \lambda_n + c_n(x_{n+1} - \tilde{x}_{n+1}).$$

Step 5. Penalty Parameter Updating. If

$$\|x_{n+1} - \tilde{x}_{n+1}\| > \alpha\|x_n - \tilde{x}_n\|,$$

then set

$$c_{n+1} := \beta c_n.$$

A suitable choice is $\alpha := 1.10, \beta := 2$.

2.4. First Computational Test. In this first test, we use an n-dimensional version of the simple unit commitment (UC) problem presented above. The notation employed is as follows: n is the number of thermal units; x_i is the output of thermal unit i; Con_i is the starting cost, i.e., the cost of turning a unit on; d is the demand of electrical power; and l_i and u_i are lower and upper bounds for x_i. The UC problem is given below,

$$\min \quad \sum_{i=1}^{n} [2x_i^2 + \mathrm{Con}_i(x_i)], \tag{18a}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_i = d, \tag{18b}$$

$$x_i \in \{0\}\cup[l_i, u_i], \qquad i = 1, \ldots, n, \tag{18c}$$

where arbitrarily we choose

$$\mathrm{Con}_i(x_i) := \begin{cases} 10 + [10/(n-1)](i-1), & \text{if } x_i \in [l_i, u_i], \\ 0, & \text{if } x_i = 0. \end{cases} \tag{19}$$

Given that the starting cost function Con_i is monotone on i and that the generating cost is the same for all the units ($2x_i^2, i = 1, \ldots, n$), this problem can be solved exactly (algebraic method). For example, let us solve problem (18) for the particular case of 3 thermal units ($n = 3$), demand $d = 6$, lower bound $l_i = 1$, and upper bound $u_i = 6$, with $i = 1, 2, 3$. That is,

$$\min \quad \sum_{i=1}^{3} [2x_i^2 + \mathrm{Con}_i(x_i)], \tag{20a}$$

$$\text{s.t.} \quad x_1 + x_2 + x_3 = 6, \tag{20b}$$

$$x_i \in \{0\}\cup[1, 6], \qquad i = 1, 2, 3, \tag{20c}$$

with

$$\mathrm{Con}_i(x_i) := \begin{cases} k_i := 10 + [10/(3-1)]\,(i-1), & \text{if } x_i \in [1, 6], \\ 0, & \text{if } x_i = 0. \end{cases} \tag{21}$$

Note that

$$k_1 = 10, \qquad k_2 = 15, \qquad k_3 = 20;$$

i.e., the starting cost is monotone on i as we have already pointed out. If n^* is defined as the optimal number of thermal units, then a priori there are three possibilities: $n^* = 1, 2$, or 3.

If $n^* = 1$, given that unit 1 has the cheapest starting cost ($k_1 = 10$) and that the generating cost ($2x_i^2$, $i = 1, 2, 3$) is the same for the three units, then the optimal solution would be $x^* = (6, 0, 0)$, with an optimal cost of $2 \times 6^2 + 10 = 82$. Analogously, if $n^* = 2$, then it is easy to see that $x^* = (3, 3, 0)$ (units 1 and 2 are the cheaper ones), and in that case, the optimal cost would be $2 \times 3^2 + 10 + 2 \times 3^2 + 15 = 61$. Finally, if $n^* = 3$, then it is also easy to see that $x^* = (2, 2, 2)$ and the optimal cost would be $2 \times 2^2 + 10 + 2 \times 2^2 + 15 + 2 \times 2^2 + 20 = 69$. Thus, the optimal solutions is $n^* = 2, x^* = (3, 3, 0)$ with optimal cost 61. Note that we have computed a global optimizer.

The parameters used in this first computational test are

$$n = 10, 20, 30, \ldots, 100,$$
$$d = n,$$
$$l_i = 1,$$
$$u_i = n.$$

Regarding the tuning parameters for both the ALR + APP algorithm and the ALR + BCD algorithm, the initial penalty c_0 has been set equal to 1, 0.1, 0.05 for each method in three different trials. The parameter b_0 has been set equal to $2c_0$, and the scalars α, β, γ have been set equal to 1.1, 2, 2 respectively. The reported results in this paper correspond to the best trial (lowest local optimum). Table 1 displays the best c_0 for each method and each case.

This first test was performed in three steps.

Step 1. In the first step, we solve exactly the UC problem for 10 cases ($n = 10, 20, 30, \ldots, 100$) by the above algebraic method in order to know its global optimizer. The optimal number of thermal units and the optimal cost, both obtained with the algebraic method, are displayed in Table 2 and Table 3 respectively (label ALG).

Table 1. Initial penalty parameter c_0 for Test 1.

	c_0	
Thermal units	APP	BCD
10	0.10	0.10
20	0.05	0.05
30	0.10	0.10
40	1.00	0.10
50	1.00	0.10
60	0.10	0.05
70	0.10	0.10
80	0.10	0.05
90	0.10	0.10
100	0.10	0.10

Step 2. In the second step we solve the same 10 problems using the augmented Lagrangian relaxation (ALR) method + block coordinate descent (BCD) method, and the ALR method + the auxiliary problem principle (APP) method to check the quality of the computed optimizers: are they local or global optimizers?. The relative error $100 * (f_{\text{computed}} - f^*)/f^*$ of the computed solution is displayed for the BCD and the APP methods in Table 3. For example, in case $n = 70$, neither method reaches the global optimizer. Table 3 shows that these methods obtain either the global

Table 2. Optimal number of thermal units, Test 1.

	Optimal number of thermal units		
Thermal units	ALG	APP	BCD
10	4	4	4
20	8	7	8
30	11	11	11
40	15	16	15
50	19	20	19
60	23	23	22
70	27	26	29
80	30	30	31
90	34	33	33
100	38	38	37
Average	20.9	20.8	20.9

Table 3. Quality of the computed optimizers, Test 1.

	Optimum cost			Relative error (%)	
Thermal units	ALG	APP	BCD	APP	BCD
10	96.67	96.70	96.67	0.03	0.00
20	194.74	195.38	194.74	0.33	0.00
30	292.60	292.60	292.60	0.00	0.00
40	390.36	390.91	390.60	0.08	0.00
50	488.06	488.84	488.10	0.16	0.01
60	585.92	585.92	586.41	0.00	0.08
70	683.83	684.02	686.82	0.03	0.44
80	781.73	781.73	781.76	0.00	0.00
90	879.50	880.20	880.20	0.08	0.08
100	977.33	977.33	977.85	0.00	0.05
Average	537.10	537.36	537.58	0.07	0.07

optimizer or a high-quality local optimizer (relative error always under 0.5%). On average, the relative error of the computed optima is 0.07% for both the APP method and the BCD method.

Step 3. Once we know that the quality of the solutions is similar for the two methods, we investigate their performance. The CPU time for the two methods can be found in Table 4 as well as the time ratio between the APP method versus the BCD method. In most cases, the BCD method is the fastest one. Thus, for example, in the first case ($n = 10$), the APP time is 17% greater than the BCD time (see time ratio column). On

Table 4. APP method versus BCD method, Test 1.

	Time (sec)		Time ratio
Thermal units	APP	BCD	APP/BCD
10	2.7	2.3	1.17
20	6.8	7.9	0.86
30	9.6	5.4	1.78
40	18.0	11.1	1.62
50	29.3	19.5	1.50
60	38.0	36.4	1.04
70	49.0	68.0	0.72
80	66.6	89.3	0.75
90	91.3	49.9	1.83
100	149.3	74.5	2.00
Average	46.1	36.4	1.33

average, the APP method needs 33% more time than the BCD method.

The result obtained in this section encouraged us to compare these methods theoretically.

2.5. Theoretical Insight In the last section, the BCD method has shown to be the fastest one from an empirical point of view. In this section, we compare the BCD method to the APP method from a theoretical point of view.

Proposition 2.1. A single iteration of the auxiliary problem principle (APP) method is equivalent to

$$x_{n+1} := \arg\min_{x \in \mathscr{D}} f(x) + \lambda_n' x + (c/2)\|x - \tilde{x}_n\|^2$$

$$+ [(b-c)/2]\|x - x_n\|^2, \tag{22}$$

$$\tilde{x}_{n+1} := \arg\min_{\tilde{x} \in \tilde{\mathscr{D}}} \tilde{f}(\tilde{x}) - \lambda_n' \tilde{x} + (c/2)\|x_n - \tilde{x}\|^2$$

$$+ [(b-c)/2]\|\tilde{x}_n - \tilde{x}\|^2, \tag{23}$$

where $(x_n, \tilde{x}_n)$ is the input current iterate and $(x_{n+1}, \tilde{x}_{n+1})$ is the output next iterate.

Proof. Given that

$$c(x_n - \tilde{x}_n)'x + (b/2)\|x - x_n\|^2 \tag{24}$$

$$= cx_n'x - c\tilde{x}_n'x + (b/2)\|x\|^2 + (b/2)\|x_n\|^2$$

$$- bx'x_n \pm (c/2)\|x\|^2 \pm (c/2)\|x_n\|^2 \pm (c/2)\|\tilde{x}_n\|^2 \tag{25}$$

$$= (c/2)\|x\|^2 + (c/2)\|\tilde{x}_n\|^2 - c\tilde{x}_n'x + [(b-c)/2]\|x\|^2$$

$$+ [(b-c)/2]\|x_n\|^2 - (b-c)x_n'x + (c/2)\|x_n\|^2 - (c/2)\|\tilde{x}_n\|^2 \tag{26}$$

$$= (c/2)\|x - \tilde{x}_n\|^2 + [(b-c)/2]\|x - x_n\|^2 + (c/2)[\|x_n\|^2 - \|\tilde{x}_n\|^2], \tag{27}$$

then the minimization over $\mathscr{D}$,

$$\min_{x \in \mathscr{D}} f(x) + \lambda_n' x + c(x_n - \tilde{x}_n)'x + (b/2)\|x - x_n\|^2, \tag{28}$$

can be rewritten as

$$\min_{x \in \mathscr{D}} f(x) + \lambda_n' x + (c/2)\|x - \tilde{x}_n\|^2$$

$$+ [(b-c)/2]\|x - x_n\|^2 + (c/2)[\|x_n\|^2 - \|\tilde{x}_n\|^2]. \tag{29}$$

Analogously the minimization over $\tilde{\mathscr{D}}$,

$$\min_{x \in \tilde{\mathscr{D}}} \tilde{f}(\tilde{x}) - \lambda'_n \tilde{x} - c(x_n - \tilde{x}_n)' \tilde{x} + (b/2) \|\tilde{x} - \tilde{x}_n\|^2, \tag{30}$$

can be rewritten as

$$\begin{aligned} &\min_{x \in \tilde{\mathscr{D}}} \tilde{f}(\tilde{x}) - \lambda'_n \tilde{x} \\ &+ (c/2) \|x_n - \tilde{x}\|^2 + [(b-c)/2] \|\tilde{x} - \tilde{x}_n\|^2 + (c/2)[\|\tilde{x}_n\|^2 - \|x_n\|^2]. \end{aligned} \tag{31}$$

Now, the constant terms $(c/2)[\|x_n\| - \|\tilde{x}_n\|]$ and $(c/2)[\|\tilde{x}_n\| - \|x_n\|]$ of the objective functions in (29) and (31) can be removed without changing the optimal iterate. Therefore, a single iteration of the APP method is equivalent to

$$\begin{aligned} x_{n+1} &:= \arg\min_{x \in \mathscr{D}} f(x) + \lambda'^n x + (c/2) \|x - \tilde{x}_n\|^2 \\ &\quad + [(b-c)/2] \|x - x_n\|^2, \end{aligned} \tag{32}$$

$$\begin{aligned} \tilde{x}_{n+1} &:= \arg\min_{\tilde{x} \in \tilde{\mathscr{D}}} \tilde{f}(\tilde{x}) - \lambda'_n \tilde{x} + (c/2) \|x_n - \tilde{x}\|^2 \\ &\quad + [(b-c)/2] \|\tilde{x}_n - \tilde{x}\|^2, \end{aligned} \tag{33}$$

as we wanted to prove. □

Corollary 2.1. For $b = c$, the APP method is nothing but the Jacobi version of the nonlinear Gauss–Seidel or BCD method.

Proof. This is shown directly comparing Proposition 2.5 with the BCD method below (nonlinear Gauss–Seidel method),

$$x_{n+1} := \arg\min_{x \in \mathscr{D}} f(x) + \lambda'_n x + (c/2) \|x - \tilde{x}_n\|^2, \tag{34}$$

$$\tilde{x}_{n+1} := \arg\min_{\tilde{x} \in \tilde{\mathscr{D}}} \tilde{f}(\tilde{x}) - \lambda'_n \tilde{x} + (c/2) \|x_{n+1} - \tilde{x}\|^2. \tag{35}$$

Note that, for $b = c$, the last term in (22)–(23) vanishes. Therefore, the difference between both methods is only that, in (23), the APP method uses the old iterate x_n (Jacobi method) whereas in (35) the BCD method uses the new iterate x_{n+1} computed in (34) (nonlinear Gauss–Seidel method). □

It is well known in numerical analysis that the nonlinear Gauss–Seidel method is likely to outperform the Jacobi method (Refs. 20–22). The main reason is that the nonlinear Gauss–Seidel method incorporates immediately the new generated information x_{n+1} within the current iteration [like in (17)], whereas the Jacobi method incorporates this new information at the beginning of the next iteration and uses x_n instead [like in (23)]. Thus, a first consequence of the precedent corollary is that a faster convergence to the optimum can be expected using the BCD method than using the APP method for the particular case $b = c$.

Let us study now the APP method for the more general case $b > c$. In Proposition 2.1, if we consider the minimization of the objective function in $\mathscr{D} \times \tilde{\mathscr{D}}$, the terms $[(b-c)/2]\|x - x_n\|^2$ and $[(b-c)/2]\|\tilde{x}_n - \tilde{x}\|^2$ of (22)–(23), respectively, can be joined as

$$[(b-c)/2]\|(x, \tilde{x})' - (x_n, \tilde{x}_n)'\|^2. \tag{36}$$

For $b = c$, the term (36) vanishes, but for any $b > c$ it penalizes the iterates $(x_{n+1}, \tilde{x}_{n+1})$ that lie far from $(x_n, \tilde{x}_n)$. Therefore, the greater is b, the closer will be $(x_{n+1}, \tilde{x}_{n+1})$ to $(x_n, \tilde{x}_n)$. Then, in general, it is likely that, for $b > c$, the APP method will take more iterations to converge to a particular optimizer than for $b = c$, since $b > c$ penalizes long steps from the current iterate.

For the reasons given above and considering that, in the APP method, b must fulfill the condition $b \geq 2c$, then we can expect faster performance of the BCD method when compared to the APP method.

3. Unit Commitment: Realistic Example

In Section 2.4, we presented the unit commitment (UC) problem for one interval. In practical situations, one solves the UC problem for several intervals (from 24 to 168 hours). This problem has been studied amply in the past by many authors (Refs. 1–2). Researchers are now attempting to solve enhanced versions of the UC problem in which, in addition to the system of thermal units, one takes simultaneously into account other related systems, such as hydroelectric plants and transmission networks (Refs. 4, 7, 8).

As we said already, the objective of the UC problem is to optimize electricity production and distribution, considering a short-term planning horizon (from one day to one week). Hydroelectric and thermal plants must be coordinated in order to satisfy the customer demand of electricity at minimum cost and with a reliable service. The model for the UC problem presented here considers the thermal system, hydroelectric system, and distribution network.

3.1. Formulation of the Unit Commitment Problem. The optimization problem considered here is

$$\min \quad f(x) = C_{htd}(x) + C_m(x), \tag{37a}$$

$$\text{s.t.} \quad x \in \mathscr{D}_{htd}, \tag{37b}$$

$$x \in \mathscr{D}_m. \tag{37c}$$

Here, $\mathscr{D}_{htd}$ represents the domain defined by the constraints that couple the hydroelectric, thermal, and distribution systems (load constraints, spinning reserve constraints, etc.); $\mathscr{D}_m$ represents the domain of the management for the thermal units (minimum up and down times, minimum and maximum output levels, etc.); $C_{htd}(x)$ represents the costs associated with $\mathscr{D}_{htd}$, and $C_m(x)$ represents the costs associated with $\mathscr{D}_m$.

3.2. Modeling the Unit Commitment Problem. The general expression of the UC problem (37a) could be developed in several different ways. The approach adopted in this paper follows the so-called coupled model presented in Ref. 17. This model takes into account the hydroelectric energy generation system together with the thermal system and transmission network. The variable vector x of problem (37a) splits into three different vectors; x_H for the variables related with the hydroelectric system (volume, discharges, and spillages of each reservoir), x_T for the thermal variables (power output and spinning reserve of each thermal unit), and x_E for the variables which account for the power flow through the electric transmission network.

In the coupled model, the constraints relating all these variables [domain $\mathscr{D}_{htd}$ of problem (37a)] are expressed through a network flow model with side constraints,

$$A_{HTT}\begin{bmatrix} x_H \\ x_T \\ x_E \end{bmatrix} = b_{HTT}, \tag{38}$$

$$h(x_H, x_E) = 0, \tag{39}$$

$$T_{ISR} x_T \geq b_{ISR}, \tag{40}$$

$$T_{DSR} x_T \geq b_{DSR}, \tag{41}$$

$$T_{KVL} x_E = 0, \tag{42}$$

$$\underline{x}_H \leq x_H \leq \bar{x}_H, \tag{43}$$

$$\underline{x}_T \leq x_T \leq \bar{x}_T, \tag{44}$$

$$\underline{x}_E \leq x_E \leq \bar{x}_E. \tag{45}$$

The following comments are pertinent.

Equations (38). These are the network constraints associated with the so-called hydro-thermal-transmission extended network (HTTEN). The HTTEN integrates the replicated hydro network (which accounts for the time and space coupling between the reservoirs of the river basin), the thermal equivalent network (which defines the relation between the power output and the spinning reserve level of each thermal unit), and the transmission network (which formulates the conservation of the power flow at the busses of the transmission system).

Equations (39). These nonlinear equality constraints define the injection of the hydroelectric generation (a nonlinear function of the variables x_H) into the appropriate busses of the transmission network. The solution procedure will be based on a successive linearization of these constraints.

Inequalities (40), (41). These two sets of linear side constraints impose the satisfaction of the incremental and decremental spinning reserve requirements of the whole system.

Equation (42). This last set of linear equality constraints is the formulation of the Kirchhoff voltage law. These constraints, together with the power flow conservation equations formulated in (38), represent a dc approach to the transmission network.

Inequalities (43)–(45). These are upper and lower bounds to the variables.

The formulation of the domain $\mathscr{D}_{htd}$ as a network flow problem with equality and side constraints allows one the use of specialized network optimization codes (Refs. 17–18). Also, the flexibility of this model is such that any other relevant system constraints can be added easily, for instance, security constraints and emission constraints (see Ref. 23).

The thermal management domain $\mathscr{D}_m$ of problem (37a) deals with the physical and economic constraints of the thermal units. First, the minimum up and down times of each thermal unit must be respected, as too many on/off switches will stress the system. Second, for economic reasons, each unit has a minimum and maximum output level which bounds its production level.

To be more precise, the management domain D_m takes into account the disconnected domain of the variables p^i_t (power output of the thermal unit t at interval i), that is,

$$p^i_t \in \{0\} \cup [\underline{p}^i{}_t, \bar{p}^i{}_t], \tag{46}$$

whereas in D_{htd} the variables $p^i{}_t$ are allowed to oscillate between 0 and the upper bound $\bar{p}^i_t$ in order to have a connected domain D_{htd}.

The domain D_m describes also the restrictions to the startup and shutdown processes of the thermal generators. When a thermal generator t has been shut down, then for mechanical reasons, it must be off at least a given time min_off_t (minimum down time). Equivalently, when a thermal generator t is started-up, it must be on at least a given time min_on_t (minimum up time). Mathematically,

$$\text{if } p_t^{i-1} = 0 \text{ and } p_t^i > 0, \text{ then } p_t^j > 0 \ (j = i, \ldots, i + \text{min\_on}_t - 1), \tag{47}$$

$$\text{if } p_t^{i-1} > 0 \text{ and } p_t^i = 0, \text{ then } p_t^j = 0 \ (j = i, \ldots, i + \text{min\_off}_t - 1). \tag{48}$$

Therefore, D_m is defined by Eqs. (46)–(48). This kind of constraints are difficult to handle with ready-to-use optimization packages. So far, dynamic programming is the most common method used to deal with them within the Lagrangian relaxation approach (Refs. 1–3).

The first term $C_{htd}(x)$ of the objective function of (37a) represents 50% of the cost of the fuel consumption of the thermal units, and it is modeled as a quadratic function of the power output of each thermal unit. This term includes also an estimation of the cost of the power losses through a quadratic function of some of the variables x_E. The second term $C_m(x)$ includes the remaining 50% of the fuel cost, the startup and shutdown costs of the thermal units, and depends on only the thermal variables x_T.

3.3. Second Computational Test. In this second test, seven instances of the unit commitment (UC) problem are solved. In Table 5, we describe their main features; they range from small size (2 intervals, 0 reservoirs, 2 thermal units, and 4 binary variables) up to medium size (168 intervals, 4 reservoirs, 7 thermal units, and 1176 binary variables). The UC problems solved here consider the hydroelectric and thermal systems without the distribution network.

Unlike in the first test, obviously we cannot know a priori the real optimizer for these large-scale nonlinear combinatorial problems. Consequently, this second test consists of only two steps.

Table 5. Description of UC instances, Test 2.

Case	Number of intervals	Number of reservoirs	Thermal units	Continuous variables	Binary variables
1	2	0	2	16	4
2	6	2	4	138	24
3	48	2	4	1104	192
4	48	4	7	1920	336
5	48	2	7	1680	336
6	168	4	2	3360	336
7	168	4	7	6720	1176

Table 6. Initial penalty parameter c_0, Test 2.

Case	1	2	3	4	5	6	7
c_0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-4}	10^{-4}	10^{-4}

Step 1. We solve the same seven problems using the augmented Lagrangian relaxation (ALR) method + block coordinate descent (BCD) method and the ALR method + the auxiliary problem principle (APP) method to compare the quality of the computed optimizers.

Step 2. We compare the efficiency of the ALR + BCD method and the ALR + APP method in terms of CPU time. The CPU times correspond to a Sun/Ultra2 2200 workstation with 200 MHz clock, 256 Mbytes of main memory.

The parameters used in both the ALR + APP and the ALR + BCD algorithms are: the stopping criterion with $\epsilon = 10^{-4}$, the initial penalty parameter c_0 is listed in Table 6, and the parameters used to update c_0 are $\alpha = 1.10$ and $\beta = 2$. The ALR + APP algorithm requires the additional parameter b_n, which has been set equal to $2c_n$ in all cases. The results of this second test are given below.

Results of Step 1. Comparing the cost columns in Table 7 we observe that the quality of the computed optimizers is very similar for the two algorithms, except for Case 2.

Results of Step 2. Now that we know the quality of the solutions is very similar for both methods, we investigate their performance. The relative CPU time $\text{Time}_{\text{APP}}/\text{Time}_{\text{BCD}}$ is displayed in Table 7 (last column). For example, in the first case, the APP method takes 56% more time than the BCD method. On average, the APP method takes 84% more time than the

Table 7. Results using the APP and BCD algorithms, Test 2.

	Iterations		Cost (10^6 Pesetas)		CPU Time (sec)		
Case	APP	BCD	APP	BCD	APP	BCD	Ratio
1	30	17	0.004	0.004	14	9	1.56
2	99	67	13.652	13.883	37	25	1.48
3	85	41	0.985	0.985	45	21	2.14
4	67	41	6.413	6.410	56	40	1.40
5	111	28	1.105	1.105	64	20	3.20
6	63	45	4.445	4.454	107	76	1.41
7	135	54	3.152	3.152	455	269	1.69
Average	84	42	4.251	4.285	111	66	1.84

BCD method in this particular test, as expected from a theoretical point of view (see Section 2.5).

4. Conclusions

The unit commitment problem has been solved successfully using variable duplication plus the augmented Lagrangian relaxation (ALR) method. Theoretically and practically, the block coordinate descent method is shown to be faster than the auxiliary problem principle method to deal with the nonseparable augmented Lagrangian.

With the above methodology (the ALR method), one obtains either a local or a global optimizer. The quality of the computed solution is unknown and therefore more research is needed. At present, the authors are working toward an improvement of the solutions and the measure of their quality.

References

1. Merlin, A., and Sandrin, P., *A New Method for Unit Commitment at Electricité de France*, IEEE Transactions on Power Apparatus and Systems, Vol. 102, pp. 1218–1225, 1983.
2. Virmani, S., Adrian, C., et al., *Implementation of a Lagrangian Relaxation Based Unit Commitment Problem*, IEEE Transactions on Power Systems, Vol. 4, pp. 1373–1380, 1989.
3. Wang, S. J., Shahidehpour, S. M., et al, *Short-Term Generation Scheduling with Transmission and Environmental Constraints Using an Augmented Lagrangian Relaxation*, IEEE Transactions on Power Systems, Vol. 10, pp. 1294–1301, 1995.
4. Pellegrino, F., Renaud, A., et al, *Bundle and Augmented Lagrangian Methods for Short-Term Unit Commitment*, 12th Power System Computation Conference, Dresden, Germany, Vol. 2, pp. 730–739, 1996.
5. Beltran, C., and Heredia, F. J., *Short-Term Hydrothermal Coordination by Augmented Lagrangian Relaxation: A New Multiplier Updating*, Investigación Operativa, Vol. 8, pp. 63–76, 1999.
6. Tong, S. K., and Shahidehpour, S. M., *An Innovative Approach to Generation Scheduling in Large-Scale Hydro-Thermal Power Systems with Fuel Constrained Units*, IEEE Transactions on Power Systems, Vol. 5, pp. 665–673, 1990.
7. Jiménez, N., and Conejo, A., *Short-Term Hydrothermal Coordination by Lagrangian Relaxation: Solution of the Dual Problem*, IEEE Transactions on Power Systems, Vol. 14, pp. 89–95, 1999.

8. Batut, J., and Renaud, A., *Daily Generation Scheduling Optimization with Transmission Constraints: A New Class of Algorithms*, IEEE Transactions on Power Systems, Vol. 7, pp. 982–989, 1992.
9. Baldick, R., *The Generalized Unit Commitment Problem*, IEEE Transactions on Power Systems, Vol. 10, pp. 465–475, 1995.
10. Minoux, M., *Programation Mathematique*, Dunod, Paris, France, Vol. 1, 1983.
11. Rockafellar, R. T., *A Dual Approach to Solving Nonlinear Programming Problems by Unconstrained Optimization*, Mathematical Programming, Vol. 5, pp. 354–373, 1973.
12. Bertsekas, D. P., *Nonlinear Programming*, Athena Scientific, Belmont, Massachusetts, 1995.
13. Luenberger, D. G., *Linear and Nonlinear Programming*, 2nd Edition, Addison-Wesley, Reading, Massachusetts, 1984.
14. Gill, P. E., Murray, W., et al., *Practical Optimization*, Academic Press, London, England, 10th Edition, 1995.
15. Lai, S., and Baldick, R., *Unit Commitment with Ramp Multipliers*, IEEE Transactions on Power Systems, Vol. 14, pp. 58–64, 1999.
16. Cohen, G., *Auxiliary Problem Principle and Decomposition of Optimization Problems*, Journal of Optimization Theory and Applications, Vol. 32, pp. 277–305, 1980.
17. Heredia, F. J., and Nabona, N., *Optimum Short-Term Hydrothermal Scheduling with Spinning Reserve through Network Flows*, IEEE Transactions on Power Systems, Vol. 10, pp. 1642–1651, 1995.
18. Heredia, F. J., *Constrained Nonlinear Network Flow Problems through Projected Lagrangian Methods*, Problems in Modern Applied Mathematics, Edited by N. E. Mastorakis, Mathematics and Computers in Science and Engineering, World Scientific and Engineering Society Press, Athens, Greece, pp. 406–411, 2000.
19. Beltran, C., and Heredia, F. J., *A New Dual Method for the Short-Term Hydrothermal Coordination Problem*, Technical Report DR 98/09, Departament d'Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, Barcelona, Spain, 1998.
20. Acton, F. S., *Numerical Methods That Work*, 2nd Edition, Mathematical Association of America, Washington, DC, 1990.
21. Burden, R. L., and Faires, J. D., *Numerical Analysis*, 5th Edition, PWS Publishing Company, Boston, Massachusetts, 1993.
22. Dahlquist, G., and Bjorck, A., *Numerical Methods*, Prentice Hall, Englewood Cliffs, New Jersey, 1974.
23. Chiva, A., Nabona, N., et al, *Network Model of Short-Term Optimal Hydrothermal Power Flow with Security Constraints*, IEEE Stockholm Power Technology Proceedings, Volume PS, pp. 67–73, 1995.