Paper accepted for presentation at PPT 2001
2001 IEEE Porto Power Tech Conference
10th –13th September, Porto, Portugal

Lagrangian Relaxation and Tabu Search Approaches for the Unit Commitment Problem

A. Borghetti, A. Frangioni, F. Lacalandra, A. Lodi, S. Martello, C. A. Nucci, A. Trebbi

***Abstract*-- The paper deals with the solution of the optimal short-term unit commitment (UC) problem in an electric power system. The optimization model takes into account the main operating constraints and physical characteristics of the power generation system. A comparison between Lagrangian heuristics and Tabu Search techniques on different classes of realistic instances is presented. Such a comparison is aimed at highlighting the strong features and the weaknesses of each technique, in view of their application in computer models for competitive electricity markets in progressive evolution. The comparison may provide insights for the construction of hybrid techniques that incorporate the best of both approaches.**

***Index Terms*— Lagrangian Relaxation, Optimization methods, Power generation dispatch, Tabu Search, Unit commitment**

I. Nomenclature

I	number of units.
T	number of time periods.
$\mathbf{D}$	T-dimensional vector of the load demands in each period t in the scheduling horizon.
$\mathbf{u}_i$	T-dimensional array of the 0-1 variables $u_{i,t}$ indicating the 'status' of unit i in time period t, i.e., whether unit i is committed in period t or not.
$\mathbf{p}_i$	T-dimensional array of the amount of power $p_{i,t}$ that the unit i is producing in period t;
$c_i(u_{i,t}, p_{i,t})$	per hour thermal generation cost of unit i as a function of MW provided, if committed, at current period t. The function is here assumed a quadratic function.
$s_i(u_{i,t-1}, u_{i,t})$	start-up cost which is charged whenever the unit is committed. In general, this cost depends on how long the unit has been uncommitted.
$p_i^{\min}$, $p_i^{\max}$	minimum and maximum output of unit i.
τ_i^u, τ_i^d	minimum up and down times of unit i.

This work is supported in part by the Italian National Research Council.

A. Borghetti, F. Lacalandra, C.A. Nucci are with the University of Bologna, Department of Electrical Engineering, 40136 Bologna, Italy.

A. Frangioni is with the University of Pisa, DI, 56125 Pisa Italy

A. Lodi, S. Martello are with the University of Bologna, DEIS, 40136 Bologna, Italy.

A. Trebbi is with Enel Produzione S.p.A., Processi e Sistemi, Rome, Italy.

II. Introduction

THIS paper deals with the solution of the optimal short-term unit commitment (UC) problem in an electric power system. Obtaining good schedules can considerably reduce the production costs, which is of increasing importance in the ongoing liberalization of the electricity market in many countries. Further, optimization models can assist market operators in determining which generators will be committed, and consequently paid, and this results in a great incentive to implement fair processes for choosing from alternative near-optimal solutions [1].

The optimization model considered in this paper takes into account the main operating constraints and physical characteristics of the power generation system [2]. The relevant mathematical formulation consists of a large-scale mixed-integer non-linear non-convex optimization problem.

Efficient Lagrangian-based heuristics have been proposed in the literature, where the problem is decomposed into independent single-unit problems by relaxing the constraint that the sum of the generated powers must equal the required load (e.g. [3-6]). Heuristics are then used to obtain a feasible schedule. Such an approach is referred to in the literature as Lagrangian relaxation (LR) method [7].

Different algorithms are available for the solution of the Lagrangian dual. The correct choice of the algorithm is critical both for the efficiency of the lower bound computation and for the quality of the primal solution obtained. In particular, since the dual exhibits a disaggregate structure, aggregate and disaggregate bundle approaches can be used (e.g. [3,4]), together with classical subgradient methods (e.g. [5,6]). With the proper choice, these techniques allow to efficiently obtain provable almost-optimal solutions, with relative gaps most often smaller than 1%.

More recently, Tabu Search (TS) approaches have been proposed for the solution of unit commitment problems (e.g. [8,9]). Tabu Search techniques [10] start in general by a feasible solution of the problem, and then iteratively improve it by using local search. In the case of the UC, the starting solution is an initial schedule indicating the status (on/off) of the units in each time interval, and can be generated using several types of greedy heuristics. The local search move upgrades this schedule, possibly improving the objective function, by switching on/off one of the units at a time in one or more periods. As soon as a local optimum is reached, i.e.,

no improving move can be performed, several escaping actions are settled up based on the acceptance of uphill moves with anticycle mechanisms. The anticycle mechanism is guaranteed by partially storing each move in a "tabu" list, thus identifying admissible and inadmissible moves. Fast convergence to good solutions is achieved by the method in very short computing times.

In this paper, we compare Lagrangian heuristics and Tabu Search techniques on several different classes of realistic instances. This comparison is aimed at highlighting the strong points and the weaknesses of each technique, in an attempt to evaluate their usefulness for the solution of the next generation of UC models. In addition, the comparison provides insights for the construction of hybrid techniques that incorporate the best of both approaches.

III. Problem formulation

The comparison is carried out on the following classical formulation of the UC problem, which takes into account the main operating constraints and the physical characteristics of the power generation system. In particular, we consider a power system of I thermal units.

The aim is to determine the optimal commitment that satisfies the forecasted demands **D** without violating physical and operating constraints of the generation equipment and demand specifications. The objective function to be minimized is the sum of thermal generation costs and start-up costs, i.e.,

$$\min_{\mathbf{u},\mathbf{p}} \left\{ \sum_{i=1}^{I} \sum_{t=1}^{T} \left[c_i\left(u_{i,t}, p_{i,t}\right) + s_i\left(u_{i,t-1}, u_{i,t}\right) \right] \right\} = \min_{\mathbf{u},\mathbf{p}} C(\mathbf{u},\mathbf{p}) \quad (1)$$

subject to initial conditions and subject to the system demand constraints and individual unit constraints. The system demand constraints require that the sum of all thermal generation should equal the (estimated) system demand at each hour, i.e.,

$$D_t - \sum_{i=1}^{I} p_{i,t} = 0 \qquad \forall t = 1,\ldots,T \quad (2)$$

The individual unit constraints that are considered are

$$\left.\begin{array}{l} u_{i,t} \cdot p_i^{\min} \le p_{i,t} \le u_{i,t} \cdot p_i^{\max} \\ \tau_i^d \text{ and } \tau_i^u \text{ constraints} \end{array}\right\} \quad \begin{array}{l} \forall i = 1,\ldots,I \\ \forall t = 1,\ldots,T \end{array} \quad (3)$$

Reserve requirement and power flow network constraints and losses have not been included in the model, since this is beyond the scope of the paper.

IV. Lagrangian-relaxation methods

A. Lagrangian relaxation approach

The Lagrangian relaxation approach allows decomposing the primal problem (1)-(3) into I independent sub-problems, each ones associated with each unit. By relaxing the system wide demand requirements (i.e., the "coupling" constraints) through Lagrangian multipliers, the following dual function is obtained

$$L(\boldsymbol{\lambda}) = \min_{\mathbf{u},\mathbf{p}} \left\{ c(\mathbf{u},\mathbf{p}) + \sum_{t=1}^{T} \lambda_t \cdot \left(D_t - \sum_{i=1}^{I} p_{i,t} \right) \right\} \quad (4)$$

subject to constraints (3). λ_t, with t=1,…,T, are Lagrangian multipliers. The dual function is rearranged as

$$L(\boldsymbol{\lambda}) = \sum_{i=1}^{I} L_i(\boldsymbol{\lambda}) + \sum_{t=1}^{T} \left(\lambda_t \cdot D_t \right) \quad (5)$$

where, for unit i

$$L_i(\boldsymbol{\lambda}) = \min_{\mathbf{u}_i,\mathbf{p}_i} \left\{ \sum_{t=1}^{T} \left[c_i\left(u_{i,t}, p_{i,t}\right) - \lambda_t \cdot p_{i,t} + s_i\left(u_{i,t-1}, u_{i,t}\right) \right] \right\} \quad (6)$$

subject to constraints (3).

Problems (6) are I sub-problems, one for each unit, that need to be solved. Since dual function (5) exhibits a disaggregate structure, this approach allows the separation of the problem into I uncoupled sub-problems associated with each unit. To solve each sub-problem (6), the values of λ are considered assigned, at first, as well as the values of the 0-1 decision-variables $\mathbf{u}_i$. It follows that this sub-problem, for each unit i, becomes

$$\tilde{L}_i(\boldsymbol{\lambda}) = \min_{\mathbf{p}_i} \left\{ \sum_{t=1}^{T} \left[c_i\left(u_{i,t}, p_{i,t}\right) - \lambda_t \cdot p_{i,t} \right] \right\} \quad (7)$$

subject to upper/lower limits specified in equation (3) (without the minimum up/down constraints).

The solution of Problem (7) gives the values of power productions $\mathbf{p}_i$. To find the optimal values for the decision variables $\mathbf{u}_i$, subproblem (6) is solved using the values of $\mathbf{p}_i$ in a forward dynamic programming algorithm, taking into account the start-up prices and the minimum up/down time constraints.

In order to find the optimal values of multipliers $\boldsymbol{\lambda}$, the following dual problem is solved

$$L^* = \max_{\boldsymbol{\lambda} \ge 0} L(\boldsymbol{\lambda}) \quad (8)$$

L^* provides a lower bound on the optimal objective value of the primal problem (1)-(3) [7]. As a by-product of the process of maximizing L we obtain the optimal Lagrange multipliers $\boldsymbol{\lambda}^*$ and a system schedule {**u,p**} resulting from the solution of the Lagrangian relaxation for $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$. Unfortunately, often this system schedule does not satisfy constraints (2) and, therefore, a heuristic approach for computing a near-optimal schedule has to be implemented [5,6].

The output of the heuristic algorithm is a feasible matrix **u**: then, a solution of the primal problem (1)-(3), for the given **u,** is computed by solving the so-called *Economic Dispatch Problem* (EDP). This solution constitutes an upper bound of the optimal objective value of the primal problem (1)-(3).

Since, at each iteration, the algorithm maintains both the available highest lower bound (LB) and the lowest upper bound (UB), then the solution can be considered reached as soon as the relative duality gap $(UB - LB)/LB$ becomes smaller than a specified value (able to guarantee that the

optimality has been reached).

The following three paragraphs briefly describe

- a bundle method, implemented to update multipliers $\boldsymbol{\lambda}$ at each iteration, in order to solve problem (8);
- a Lagrangian heuristic adopted to compute a feasible schedule from the solution of problem (8), exploiting the unique features of the bundle approach;
- the procedure used to calculate the initial value of the Lagrangian multipliers ($\boldsymbol{\lambda}_1$).

B. Bundle methods for updating the Lagrangian multipliers

The subgradient $\mathbf{g}(\boldsymbol{\lambda})$ of $L(\boldsymbol{\lambda})$ with respect to Lagrangian multipliers $\boldsymbol{\lambda}$ is a T-vector. The t-th element is

$$g_t(\boldsymbol{\lambda}) = D_t - \sum_{i=1}^{I} p_{i,t} \tag{9}$$

At iteration k, the bundle method accumulates multipliers $\boldsymbol{\lambda}_1$, ..., $\boldsymbol{\lambda}_k$, subgradients $\mathbf{g}(\boldsymbol{\lambda}_1)$, ..., $\mathbf{g}(\boldsymbol{\lambda}_k)$ and dual function values $L(\boldsymbol{\lambda}_1)$, ..., $L(\boldsymbol{\lambda}_k)$ in a *bundle* $\beta = < \boldsymbol{\lambda}_k, \mathbf{g}(\boldsymbol{\lambda}_k), L(\boldsymbol{\lambda}_k) >$. With this bundle, $L(\boldsymbol{\lambda})$ is upper approximated with the following *cutting plane (CP) model*

$$L_k^{\mathrm{CP}}(\boldsymbol{\lambda}) = \min_{1 \le j \le k} [L(\boldsymbol{\lambda}_j) + \mathbf{g}(\boldsymbol{\lambda}_j)' \cdot (\boldsymbol{\lambda} - \boldsymbol{\lambda}_j)] \tag{10}$$

where $\mathbf{g}(\boldsymbol{\lambda}_j)'$ is the transpose vector of $\mathbf{g}(\boldsymbol{\lambda}_j)$.

This approximation is tight at least in every point $\boldsymbol{\lambda}_j$. As described in [11], the method tries to maximize the known function L_k^{CP} instead of the unknown function L and to use the maximum as the next iteration. At iteration k, let $\bar{\boldsymbol{\lambda}}$ be the so called *current point*, i.e. the T-vector of $\boldsymbol{\lambda}$ values yielding the highest LB currently available, and let $\Delta L_j = L(\boldsymbol{\lambda}_j) + \mathbf{g}(\boldsymbol{\lambda}_j)' \cdot (\bar{\boldsymbol{\lambda}} - \boldsymbol{\lambda}_j) - L(\bar{\boldsymbol{\lambda}})$; then, the trial point at the next iteration $\boldsymbol{\lambda}_{k+1}$, that maximizes L_k^{CP}, is computed by the solution of the following linear problem

$$\begin{aligned} &\max_{\boldsymbol{\lambda}, v} \; v \\ &\text{subject to } v \le \Delta L_j + \mathbf{g}(\boldsymbol{\lambda}_j)' \cdot (\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}}) \qquad \forall 1 \le j \le k \end{aligned} \tag{11}$$

A major drawback of this approach is that (11) may be unbounded, especially in the first iterations. Moreover the cutting plane model will be a poor approximation of L if $\boldsymbol{\lambda}_{k+1}$ results to be too far from $\bar{\boldsymbol{\lambda}}$.

In order to overcome these drawbacks, the objective function of problem (11) is penalized by a quadratic term that discourages choosing $\boldsymbol{\lambda}_{k+1}$ far from $\bar{\boldsymbol{\lambda}}$; i.e., problem (11) is transformed into

$$\begin{aligned} &\max_{\boldsymbol{\lambda}, v} \; [v - \frac{1}{2 \cdot \alpha} \left\| \boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}} \right\|] \\ &\text{subject to } v \le \Delta L_j + \mathbf{g}(\boldsymbol{\lambda}_j)' \cdot (\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}}) \qquad \forall 1 \le j \le k \end{aligned} \tag{12}$$

where $\|\cdot\|$ is the Euclidean norm and α is a positive parameter. Problem (12) is a quadratic problem, which is always bounded.

Several important issues must be addressed in order to implement an efficient bundle algorithm: among these, the dynamic choice of parameter α at each iteration, the rules for updating of current point $\bar{\boldsymbol{\lambda}}$ and the stopping criteria for the iterative algorithm [11].

As previously observed, the dual function $L(\boldsymbol{\lambda})$ exhibits a disaggregate structure; therefore, a *disaggregate* bundle approach has been also implemented by replacing the aggregated CP model $L_k^{\mathrm{CP}}(\boldsymbol{\lambda})$ with the sum of I CP models, one for each unit i, and by replacing (12) with

$$\begin{aligned} &\max_{\boldsymbol{\lambda}, v} \; [\sum_{i=1}^{I} v_i - \frac{1}{2 \cdot \alpha} \left\| \boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}} \right\| + (\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}})' \cdot \mathbf{D}] \\ &\text{subject to: } v \le \Delta L_{i,j} + \mathbf{g}_i(\boldsymbol{\lambda}_j)' \cdot (\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}}) \qquad \forall 1 \le j \le k \end{aligned} \tag{13}$$

where the t-th element of vector $\mathbf{g}_i$ is $g_{i,t}(\boldsymbol{\lambda}) = -p_{i,t}$ and $\Delta L_{i,j} = L_i(\boldsymbol{\lambda}_j) + \mathbf{g}_i(\boldsymbol{\lambda}_j)' \cdot (\bar{\boldsymbol{\lambda}} - \boldsymbol{\lambda}_j) - L_i(\bar{\boldsymbol{\lambda}})$. The basic structure of the disaggregate algorithm remains similar to the structure of the aggregate one.

The sum of I disaggregated CP models is a much better description of $L(\boldsymbol{\lambda})$ than (10). Therefore, the longer computational time needed to solve problem (13) than (12) is largely compensated by the strong reduction in the number of iterations required to converge to optimality.

C. Lagrangian Heuristic

As already pointed out, in the case of UC, the solution of the dual problem (8) may not produce a feasible solution; hence, a heuristic procedure has to be implemented in order to obtain a feasible schedule for the units.

In the literature on the subject (e.g. [5,6]), some techniques have been proposed that can be applied independently of the method used to compute the solution of the dual problem. However, the feasible schedules obtained by adopting these techniques may be quite distant from the optimal solution.

By solving the dual problem by a bundle method, without an extra computational effort, a "convexified" solution of the original problem is also available [12,13]. This solution is a matrix with the same dimensions of matrix $\mathbf{u}$, whose elements $u_{i,t} \in [0,1]$ can be interpreted as the "probability" for unit i to be committed at period t. Making use of this matrix, a new heuristic procedure has been implemented, which includes also a following "shutdown" phase, trying to uncommit units that are not really needed. Such a heuristic procedure results in significantly improving the overall performance of the algorithm.

D. Preconditioning procedure

A preconditioning procedure is also implemented, in order to compute the initial value of the Lagrangian multipliers $\boldsymbol{\lambda}_1$ and to improve the convergence of the bundle method. This procedure is described in the following paragraphs.

First, a continuous relaxed version of problem (1)-(3) is solved, which consists on relaxing the binary variables, neglecting start-up costs and formulating the min up/down time constraints (3b) in a linear form as follows

$$\begin{aligned} u_{i,\min(T,t+r)} &\geq u_{i,t} - u_{i,t-1} \quad \forall i = 1...I,\ t = 1...T,\ r = 1...\tau_j^{up} \\ u_{i,\min(T,t+r)} &\geq 1 - u_{i,t-1} - u_{i,t} \quad \forall i = 1...I,\ t = 1...T,\ r = 1...\tau_j^{d} \end{aligned} \tag{14}$$

The computed lagrangian multipliers of such a problem are used as $\boldsymbol{\lambda}_1$ and the resulting continuous matrix $\hat{\mathbf{u}}$ ($u_{i,t} \in [0,1]$) is used, instead of $\tilde{\mathbf{u}}$, in the heuristic procedure described in section C, in order to compute an integer feasible schedule for each unit. By using this feasible solution, the subgradients values of each disaggregated CP model are computed. These subgradients are added to bundle β, improving the convergence of the method.

Figure 1a) shows the values of dual function L computed at each iteration, during the solution of the 10-unit UC problem of [14]. Although typically non-differentiable methods are not ascent approaches, i.e., improvement in the dual value at each iteration cannot be guaranteed, the extra information obtained with the implemented preconditioning procedure makes the bundle method to behave essentially as an ascent algorithm.

Thus, this preconditioning procedure avoids the typical large fluctuations in the dual function values and results to be beneficial also to the quality of the feasible primal solutions generated by the heuristic as shown in Fig. 1b).

It must be noted that this warm-start and preconditioning procedure results to be effective only if the bundle method makes use of the disaggregated model.

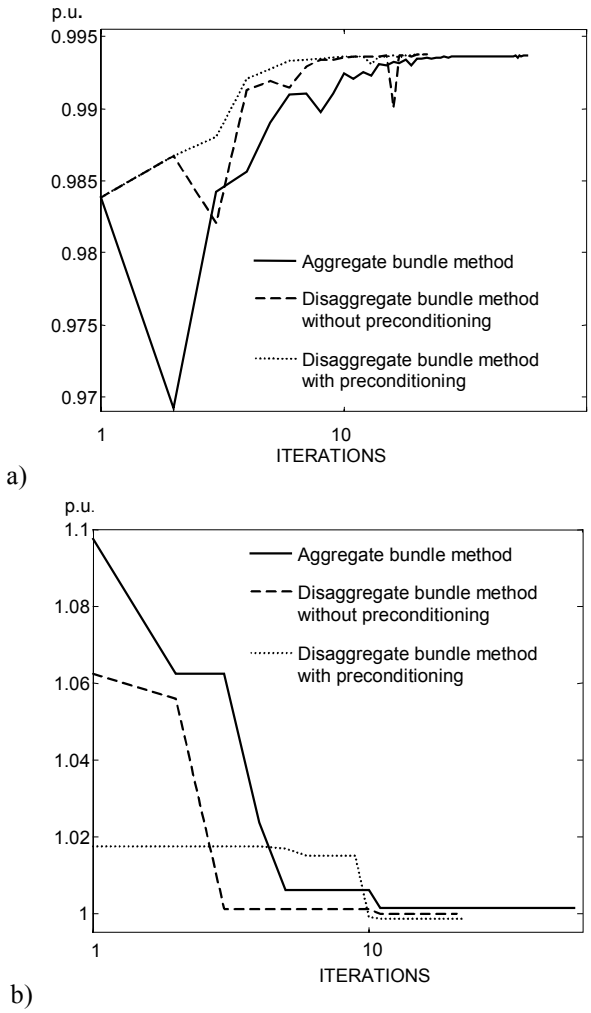

Fig. 1. a) Normalized values of dual function L and b) normalized values of the total production costs of the associated feasible schedules, computed at each iteration, for the 10-unit UC of [14].

V. TABU-SEARCH METHODS

Classical approximation algorithms start from a feasible solution, frequently obtained by executing a simple greedy algorithm, and then iteratively generate new solutions through *moves*, i.e., through modifications of the current solution. The set of solutions that a move can produce from a given one constitutes its *neighborhood*. Whenever a solution improving the current one is found, this is stored as the new current solution, and the search proceeds by exploring its neighborhood. As soon as no solution in the neighborhood of the current one can improve it, the search terminates. This approach is known as *local search*. The main drawback of this strategy is that it can be trapped in a local optimum, without possibility of exploring other (more promising) regions of the solution space.

Modern heuristic techniques overcome the above disadvantage by occasionally accepting moves that produce solutions worse than the current one, in the hope that the exploration of their neighborhood may produce better solutions or lead to a better region. This strategy is the base of the so-called *metaheuristic algorithms*.

The most popular metaheuristic technique is nowadays *Tabu Search* (see, e.g., [10]). Let x^* denote the best solution found so far, and x the current solution. The general Tabu Search strategy is to always execute the best move, i.e., to accept as new current solution the best neighbor of x, say x', even if it is worse than x (*uphill move*). In practice, the algorithm alternates a local search to obtain a local optimum and, once this has been reached, a worsening move. Performing a new local search on the neighborhood of x' could however produce a return to x. In order to avoid such a behavior, the algorithm stores information on the more recent moves in a *Tabu List*, which is used to prohibit execution of moves leading to recently explored solutions. Hence, in practice, the local search phase is usually followed by a series of uphill moves, until a non-tabu downhill move allows starting a new local search. Obviously, whenever a move produces a solution better than the incumbent one, x^* is updated. See Figure 2 for an illustrative representation of the process.

Our Tabu Search algorithm for the solution of UC problems can be outlined as follows (see [15] for more details):

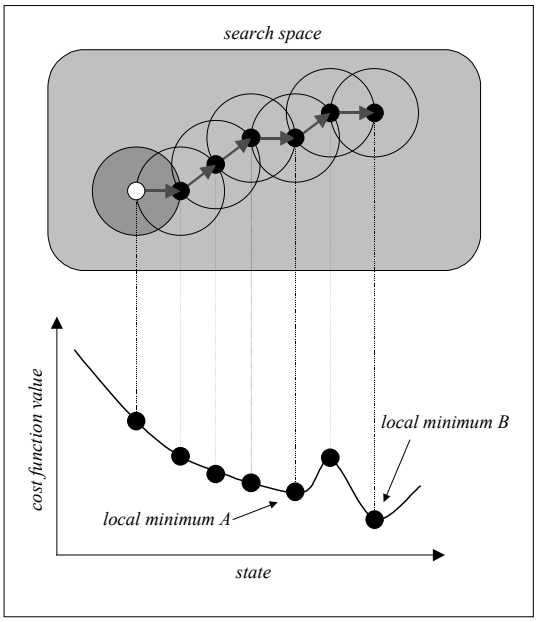

Fig. 2. Neighborhood search in Tabu Search algorithms.

1. find an initial feasible solution x;
   $x^* := x$;
   counter := 0; *did_not_improve* := 0;
   do
      counter ++; *did_not_improve* ++;
2. x' := best non-tabu solution in the neighbourhood of x;
   if cost(x') < cost(x^*) **then**
      $x^* := x'$;
      did_not_improve := 0
   endif ;
   $x := x'$
   while (*counter* ≤ *max_counter*) **and**
         (*did_not_improve* ≤ *max_not_improve*)

The initialization of Step 1 is obtained by executing two heuristic algorithms. The first one starts by committing all units in all time periods. It then determines, among the time intervals for which (2) is violated, the one for which $\sum_{i=1}^{I} p_{i,t} - D_t$ is a maximum, and uncommits units until (2) is satisfied. The process is iterated until the solution becomes feasible. The second algorithm, dual with respect to the previous one, starts by uncommitting all units in all time periods. It then determines, among the time intervals for which (2) is violated, the one for which $\sum_{i=1}^{I} p_{i,t} - D_t$ is a minimum, and commits units until (2) is satisfied. The process is iterated until the solution becomes feasible. The best of the two solutions is finally selected.

The neighborhood exploration of Step 2 is performed in three different sub-steps. In order to describe them, let the current solution x be represented by the 0-1 matrix **u** characterized, as in the previous sections, by one row for each unit i and one column for each period t. An entry (i,t) of value 0 (1) corresponds to a time period t in which unit i is uncommitted (committed). Step 2.1 in the neighborhood exploration implies the generation of a complete set of *trial* solutions, i.e., solutions obtained by manipulating in all the "acceptable" ways (see below for a precise definition of acceptable) the sequences of 0's and 1's in the rows of the matrix representing x. These trial solutions are feasible just with respect to the minimum up and down times of the unit, while their feasibility with respect to constraints (2) is tested in Step 2.2. Finally, each fully feasible solution certified by Step 2.2 is evaluated in Step 2.3 (possibly, by solving the associated EDP), and the best non-tabu solution is chosen as x'.

In order to complete the description, it remains to discuss the Tabu List. A unique Tabu List is maintained, which stores the time period indices t and k corresponding to the start and end, respectively, of the considered sequence of 0's or 1's, and the unit i of the sequence. A move is prohibited if it involves a vector j, k, i contained in the Tabu List.

Concerning the tuning of the parameters, which is usually a heavy task in the design of metaheuristic algorithms, in this case we have just three parameters to play with, namely the length of the Tabu List, say l, the maximum number of iterations without any improvement of x^*, *max_not_improve*, and the maximum number of overall iterations, *max_counter*. The last two parameters determine the stopping condition of the algorithm: as soon as either *max_not_improve* iterations have been performed without improving x^*, or *max_counter* iterations in total have been executed, the computation is terminated.

Preliminary computational experiments suggested for the three parameters the following values: $l = 28$, *max_not_improve* = 1000, and *max_counter* = 5000. These values turn out to be on average quite robust.

VI. Numerical results

A. Implementation of the algorithms

The Lagrangian relaxation algorithm was implemented in the algebraic modeling language for mathematical programming called AMPL [16]. A major advantage of the modular structure resulting with the AMPL implementation is the ability to use a library of solvers. In this paper the CPLEX [17] solver has been used to solve problems (12) (or (13)) and the EDP, i.e. the primal problem (1)-(3) with matrix **u** known as computed by the heuristic procedure.

The Tabu Search algorithm was implemented in ANSI C.

The aim of the computational section is to show the capability of these different approaches to cope with UC instances. More sophisticated implementations would obviously improve the quality of the results. As discussed in the following, this is one of the topics of a further research program aimed at integrating the presented methodologies within a single framework.

B. Instances description.

These software packages have been used to perform a number of numerical studies to assess the characteristics of the two approaches.

In the following section, the results obtained for 11 different UC problems will be presented.

The reference case is a 10-unit 24-hour UC problem, whose parameters of the cost functions $c_i(p_i) = a + b \cdot p_i + c \cdot p_i^2$ are published in [14]. The other ten UC test cases are generated from the reference with the aim to assess the influence of the various parameters of the problem on the behavior of the two different approaches. In particular, Cases 2 and 3 are used to show the influence of the number of units, Cases 4 to 7 the influence of the size of the units, Cases 8 to 11 the influence of different demand profiles.

Cases 2 and 3 are 50-unit problems, generated from the data of the reference case by repeating 5 times the generating units. The cost function parameters of the 5 sets of units of Case 2 are randomly generated according to a normal distribution with mean value equal to the relevant value of the unit parameters of the reference case and standard deviation parameter *a*, *b* and *c* equal to 100, 5 and 0.0001 respectively. All the 5 sets of unit parameters of Case 3 are taken equal to the relevant value of the unit parameters of the reference case.

Cases 4 and 5 are 10-unit problems, generated from the data of the reference case by considering only the units with $p^{\max}$ lower and higher than the relevant median value, respectively. The cost function parameters are randomly obtained as already described in Case 2.

Cases 6 and 7 are 50-unit problems, generated from the data of Case 4 and 5 respectively, with the random generation of the parameters of the cost function as in Case 2.

In each of the previous cases, the system demand data are generated by resizing those of the reference case proportionally to the value of the maximum load that can be met in each case. As already mentioned, *Cases 8 to 11* are chosen in order to assess the influence of the load profile on the behavior of the two algorithms, with the same unit parameters as in the reference case. In particular, the demand profile of Case 8 is obtained by increasing all the demand data of the reference case. This increment is equal to half of the margin between the peak load and the value of the maximum load that can be met by the units. The demand profile of Case 9 is obtained so that each demand higher than the load mean value is increased and each demand lower than the average load is decreased, with variations proportional to the increment used in Case 8. The demand profile of Case 10, instead, is obtained so that each demand higher than the load mean value is decreased and each demand lower than the average load is increased, with the same variations of Case 9. Case 11 is obtained in a similar way of Case 10, but, in this case, the variations are 5 times more pronounced, resulting in a more flat demand profile.

The influence of additional constraints, such as reserve, time-variant start-up costs and tighter constraints on the initial status of the units, can be taken into account by both approaches, and will be explicitly considered in further research activities.

C. Simulation results

Reference [14] gives the value of the actual optimum of the production cost of the reference case (Case 1), which is equal to 610,646.5. In order to assess the success of the optimization also for the other cases, whose actual optimum is unknown, we refer to the highest dual value computed by the Lagrangian relaxation algorithm.

Details of the simulation results are reported in Table I.

TABLE I
COMPUTATIONAL RESULTS

Case	Best dual value	Solution by L R	L R Gap (%)	Solution by TS	TS Gap (%)
1	$607,420.31	$611,214	0.625	$610,751	0.548
2	$3,162,766.76	$3,165,555	0.088	$3,169,274	0.206
3	$3,037,101.09	$3,042,094	0.164	$3,048,813	0.386
4	$403,593.70	$408,099	1.116	$408,087	1.113
5	$937,297.53	$944,275	0.744	$943,019	0.610
6	$2,004,813.58	$2,009,639	0.241	$2,010,108	0.264
7	$4,570,335.65	$4,576,011	0.124	$4,580,889	0.231
8	$626,913.75	$631,683	0.761	$631,921	0.799
9	$610,335.54	$616,216	0.963	$616,214	0.963
10	$604,773.96	$609,074	0.711	$609,623	0.802
11	$596,503.07	$600,399	0.653	$600,779	0.717

These computational results show:

- a good behavior of both approaches in finding approximate solutions for the UC;
- comparable difficulties of the algorithms with respect to the various instances;
- the percentage gap for the instances with 50 units is typically smaller than the one for 10 units;
- as expected, Case 8 corresponding to an augmented demand profile turns out to be more costly and difficult of the reference case (instance 1). Case 9 is even more difficult, but less costly.
- the quality of the solution obtained by the implemented Lagrangian relaxation seems not to be influenced by the number of units.

This preliminary computation shows a satisfactory behavior of both approaches, and suggests that there is definite room to integrate them in a single framework in order to exploit their different peculiarities.

In particular, the Lagrangian relaxation algorithm, widely applied in recent years for the solution of UC problems for large-scale systems due to its ability to include more detailed system representation than would be possible with other techniques, is able to compute at the same time both lower and upper bound values for the problem. In this way, the overall algorithm can, for example, be stopped as soon as a

performance guarantee is reached, i.e., a percentage gap less of equal than a given threshold (e.g., 1 %).

On the other hand, the Tabu Search approach, even in a quite straightforward implementation, proves to give good results (also within 1% of average gap) in very short computing times: at most 5 and 60 CPU seconds on a Pentium II 300 MHz for instances with 10 and 50 units, respectively. In particular, the Tabu Search algorithm is very competitive for instances with 10 units, while more sophisticated diversification techniques have to be applied to better explore the huge solution space in the case of 50 units.

A possible integration into a single algorithm of the two approaches could be effectively performed by using the feasible solutions obtained by the Lagrangian approach (through the Lagrangian heuristic) as advanced starting points for the Tabu Search approach which proved to be very effective in optimizing them. In this context, the Tabu Search could be performed for a limited number of iterations and then the Lagrangian optimization is resumed, possibly obtaining a faster convergence.

This unified approach, which is currently under investigation, would exhibit the big advantage of providing not only good feasible solutions but also a strong indication on their distance from the optimal one.

VII. Conclusions

In this paper, a Lagrangian relaxation algorithm for the solution of UC problems has been illustrated, wherein the dual problem solution is achieved through the implementation of an improved bundle method and the feasible solution for the primal problem is computed by a heuristic procedure that exploits available hints given by the bundle algorithm. The results obtained by the implemented Lagrangian relaxation algorithm are compared with those obtained by a completely different program that applies a Tabu Search algorithm, explicitly conceived for the solution of UC problems.

This comparison, carried out on a set of classical UC problems that take into account the main operating constraints and physical characteristics of the power generation system, has shown a good behavior of both approaches in finding approximate solutions. Moreover, the analysis of the different and complementary characteristics of the two approaches suggests further research activity to obtain an integrated algorithm of them, able to provide adequate solutions of the new UC problems peculiar of competitive electricity markets.

VIII. References

[1] B.F. Hobbs, M. Rothkopf, R.P. O'Neill and H.P. Chao (eds.), The Next Generation of Unit Commitment Models, Boston: Kluwer Academic Press, 2001.

[2] J. Gruhl, F. Schweppe and M. Ruane, "Unit commitment scheduling of electric power systems", In System Engineering for Power: Status and Prospects, L.H. Fink and K. Carlsen (eds.), Henniker, NH, 1975.

[3] C. Lemaréchal, C. Sagastizabal, F. Pellegrino and A. Renaud, "Bundle methods applied to the unit commitment problem", in System modelling and optimization, pp. 396- 402, Chapman & Hall, London, 1996.

[4] D. Zhang, P. B. Luh, and Y. Zhang, "A Bundle Method for Hydrothermal Scheduling", IEEE Transactions on Power Systems, vol. 14, pp. 1355-1361, 1999.

[5] J.F. Bard, "Short-term scheduling of thermal-electric generators using Lagrangian relaxation", Operation Research, vol. 36, pp. 756-766, 1988.

[6] F. Zhuang and F.D. Galiana, "Towards a more rigorous and practical unit commitment by Lagrangian relaxation", IEEE Trans. on Power Systems, vol. 3, pp. 763-773, 1988.

[7] A. Geoffrion, "Lagrangian relaxation for integer programming", Mathematical programming study, vol. 2, pp. 82-114, 1974.

[8] H. Mori and T. Usami, "Unit commitment using tabu search with restricted neighborhood", in Proc. ISAP'96, IEEE International Conference on Intelligent System Application to Power Systems, pp. 422-428, Orlando FL, 1996.

[9] A.H. Mantawy, Y.L. Abdel-Magid and S.Z. Selim, "Unit commitment by tabu search", IEE Proc. Gener. Transm. Distrib, vol. 145, pp. 56-64, January 1998.

[10] F. Glover and M. Laguna, Tabu Search, Kluwer Academic Publishers, 1997.

[11] A. Frangioni, "Dual Ascendent Methods and Multicommodity Flows", Ph.D Thesis, University of Pisa, Dipartimento di Informatica, 1997.

[12] C. Lemarechàl and A. Renaud, "Dual equivalent convex and nonconvex problems", Research report, INRIA, Rocquencourt, 1996.

[13] S. Feltenmark, "On Optimization of Power Production", Ph.D Thesis, Dept. of Mathematics/Optimization, Royal Institute of Technology, Stockholm, Sweden.

[14] V. Petridis, S. Kazarlis and A. Bakirtzis, "Varying Fitness Functions in Genetic Algorithm Constrained Optimization: The Cutting Stock and Unit Commitment Problems", IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics, Vol. 28, No. 5, October 1998.

[15] A. Lodi, S. Martello, A. Trebbi, "A Tabu Search Algorithm for the Unit Commitment Problem", Technical Report, D.E.I.S., University of Bologna, 2001.

[16] R. Fourer, D.M. Gay and B.W. Kernighan, "A modeling language for mathematical programming", Management Science, 36, pp. 519-554, 1990.

[17] R. E. Bixby, S. Ceria, C. M. McZeal and M. W. P. Savelsbergh, "An Updated Mixed Integer Programming Library: MIPLIB 3.0", Optima, 54, pp. 12{15), 1998.