JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 125, No. 1, pp. 1–18, April 2005 (© 2005)
DOI: 10.1007/s10957-004-1708-4

An Effective Line Search for the Subgradient Method[1]

C. Beltran[2] and F. J. Heredia[3]

Communicated by M. Simaan

Abstract. One of the main drawbacks of the subgradient method is the tuning process to determine the sequence of steplengths. In this paper, the radar subgradient method, a heuristic method designed to compute a tuning-free subgradient steplength, is geometrically motivated and algebraically deduced. The unit commitment problem, which arises in the electrical engineering field, is used to compare the performance of the subgradient method with the new radar subgradient method.

Key Words. Lagrangian relaxation, multiplier updating, subgradient method, radar subgradient method, unit commitment.

1. Introduction

The objective of this paper is to improve the subgradient method which is used to solve nondifferentiable optimization problems such as for example the Lagrangian dual problem. One of the main drawbacks of the subgradient method is the tuning process to determine the sequence of steplengths to update successive iterates. To avoid this, we propose the radar subgradient method, a heuristic method designed to compute a tuning-free subgradient steplength.

It is well known that the dual function is a concave function over its domain (regardless of the cost structure and constraints of the primal problem), but not necessarily differentiable (Ref. 1, Chapter 5). If there is no duality gap, that is, if the optimal values of the primal problem and the

[1]This research was supported by the Spanish Government, CICYT Grant TAP99-1075-C02-01. We acknowledge the technical support from Logilab (HEC, University of Geneva) and especially the valuable remarks and suggestions of the referees.

[2]Researcher, Logilab, HEC, University of Geneva, Geneva, Switzerland.

[3]Professor, Department of Statistics and Operations Research, Polytechnical University of Catalonia, Barcelona, Spain.

dual problem are equal, then a solution of the dual problem provides a solution of the primal problem. We solve the dual problem whenever it is easier to solve than the primal problem and there is no duality gap. However, even if there is a duality gap, the solution of the dual problem provides a lower bound to the primal optimum that can be useful for example in combinatorial optimization. Further details can be found in Refs. 1–2 and a more in-depth theoretical point of view in Ref. 3.

There are two main methods to maximize the dual function (Refs. 1, 3): the subgradient method and the cutting-plane method (we could call them families of methods, since each method has its own variants).

First, in the subgradient method, the Lagrangian multipliers λ_n (dual variables) are updated as $\lambda_{n+1} = \lambda_n + \alpha_n \cdot s_n/\|s_n\|$, where s_n is the subgradient of the dual function at the current iterate λ_n. In this case, if $\lim_{n\to\infty} \alpha_n = 0$ and $\sum_{n=0}^{\infty} \alpha_n = +\infty$, convergence is guaranteed (Ref. 4). If the dual function is nondifferentiable, the subgradient method may progress slowly to the optimum in an oscillating fashion.

The steplength α_n is usually chosen as

$$\alpha_n = 1/(a + b_n) \text{ (basic rule)},$$

or more often as

$$\alpha_n = \beta_n(\hat{q}_n - q(\lambda_n))/\|s_n\| \text{ (Polyak II rule)},$$

where $q(\lambda_n)$ is the dual function at the current iterate, $\hat{q}_n$ is an upper bound to the dual optimum, and $\beta_n \in]0, 2[$ (Ref. 1). A less common choice is

$$\alpha_n = ab^n,$$

which may yield a geometric rate of convergence if the parameters a and b are selected carefully, but without guaranteed convergence (Ref. 5, page 9). Thus, another drawback of the subgradient method is the nontrivial tuning parameter associated with some of the available choices for α_n.

Second, in the Kelley cutting-plane method (Ref. 6), the new iterate is obtained by maximizing an outer approximation of the dual function, given by a collection of cutting planes (hyperplanes). Unlike the subgradient method, the cutting-plane method does not suffer from oscillations, proper stopping criteria can be used, but the computational burden can be high depending on the implementation. The dynamically constrained cutting-plane method avoids this high computational burden by limiting the number of approximating hyperplanes (Ref. 2).

The pure cutting-plane method gives usually a slow and unstable performance, especially around the optimum (Ref. 7). To overcome this

unstability, the cutting-plane method is used often in augmented form (bundle method, Refs. 3, 7–10), where a quadratic penalty is appended to the objective function.

The newest version of the cutting-plane method is the analytical center cutting plane method (ACCPM, Refs. 11–12). Roughly speaking, at each iteration, ACCPM searches for the analytical center of the polyhedron associated to the outer approximation of the dual function given by the cutting planes. The sophisticated ACCPM code has proved to be efficient and robust (Ref. 13).

In spite of the drawbacks of the subgradient method, it is still a popular method, since it is simple to implement and because its computational burden is small. Furthermore, although the theoretical rate of convergence for pure subgradient algorithms is at best linear (Ref. 14), experience shows that their practical efficiency is good, especially for large-scale, structured optimization problems.

For this reason, the subgradient method is still a subject of active research. Thus for example, in Ref. 15 faster initial convergence of the subgradient method is achieved using exponentially weighted subgradients. The volume algorithm (Ref. 16), derived from the subgradient method, has a much better stopping criterion compared to the basic subgradient method. Another variant of the subgradient method, the incremental subgradient method, results in a much better practical rate of convergence than the subgradient method (Ref. 17). The ball-step subgradient level method can be surprisingly effective when low solution accuracy is acceptable, as it happens in many applications (Ref. 18).

Section 1.1 contains notations for the subgradient and cutting-plane methods. In Section 2, an effective steplength for the subgradient method is motivated geometrically and described algebraically. In Section 3, a set of instance of the well-known unit commitment problem are used to compare the efficiency of the basic subgradient method with that of the new method. Conclusions are given in Section 4.

1.1. Notation. Let us suppose that we wish to solve the following primal problem (P):

$$\min \quad f(x), \tag{1a}$$

$$\text{s.t.} \quad h(x)=0, \tag{1b}$$

$$x \in \mathcal{D}, \tag{1c}$$

where $f(x): \mathbb{R}^n \longmapsto \mathbb{R}$, $h(x): \mathbb{R}^n \longmapsto \mathbb{R}^m$, and $\mathcal{D}$ is a nonempty compact set in $\mathbb{R}^n$.

As usual, the Lagrangial dual Problem (D) of (P) is

$$\begin{aligned} &\max && \min f(x)+\lambda' h(x) && \text{(2a)}\\ &\lambda\in\mathbb{R}^m && x\in\mathcal{D}. && \text{(2b)} \end{aligned}$$

Equivalently, with the Lagrangian function

$$L(x,\lambda):=f(x)+\lambda' h(x),$$

the dual problem can be written as

$$\begin{aligned} &\max && \min L(x,\lambda) && \text{(3a)}\\ &\lambda\in\mathbb{R}^m && x\in\mathcal{D}. && \text{(3b)} \end{aligned}$$

In short, defining the dual function

$$q(\lambda):=\min\{L(x,\lambda): x\in\mathcal{D}\}, \tag{4}$$

the dual problem has the expression

$$\begin{aligned} &\max && q(\lambda). && \text{(5a)}\\ &\lambda\in\mathbb{R}^m && && \text{(5b)} \end{aligned}$$

Other symbols used through the paper are:

n,	iteration index;
$\{\alpha_n\}$,	sequence of positive scalars (for example, a subgradient steplength sequence);
λ_n,	Lagrange multiplier vector at iteration n;
q_n	$:=q(\lambda_n)$;
$\partial q(\lambda_n)$	subdifferential of q at λ_n, $\partial q(\lambda_n)=\{s\in\mathbb{R}^m: q(\lambda)\le q(\lambda_n)+s'(\lambda-\lambda_n)\}$;
s_n	subgradient of q at λ_n, that is, $s_n\in\partial q(\lambda_n)$;
$z_n(\lambda)$	$:\mathbb{R}^m\longmapsto\mathbb{R}$, affine function defined as $z_n(\lambda)=q_n+s_n'(\lambda-\lambda_n)$;
SP_n	$:=\left\{\begin{pmatrix}\lambda\\ z\end{pmatrix}\in\mathbb{R}^m\times\mathbb{R}: z=z_n(\lambda)\right\}$, supporting plane of $q(\lambda)$ at λ_n (associated to s_n), i.e., $z_n(\lambda_n)=q_n$ and $z_n(\lambda)\ge q(\lambda)$, for all $\lambda\in R^m$;

2. Radar Subgradient Method

2.1. Motivation. On the one hand, unlike the subgradient method, the cutting-plane method takes advantage of the first-order information generated each time $L(x, \lambda)$ is minimized. Given that $h(x_n)$ is a subgradient of $q(\lambda_n)$, eventually the cutting-plane method builds up an accurate first-order approximation of the dual function $q(\lambda)$. Then, instead of tuning a sequence of steplengths, as in the subgradient method, the cutting-plane method works upon an increasingly accurate knowledge of the dual function.

On the other hand, unlike the cutting-plane method, the subgradient method takes advantage of the subgradient direction generated each time $L(x, \lambda)$ is minimized; that is, the subgradient method converges to an optimal point using $h(x_n) \in \partial q(\lambda_n)$.

Although $h(x_n)$ may not be an ascent direction, it determines a new iterate closer to an optimizer if a suitable steplength is taken (Ref. 1, Chapter 6). Furthermore, considering that the dual function is the lower envelope of a set of affine functions,

$$\varphi_x(\lambda) := f(x) + \lambda' h(x),$$

very often the dual function will not be smooth at the optimal points, but in many other points it will be smooth; that is, $h(x_n)$ will be the steepest ascent direction $\nabla q(\lambda_n)$. All in all, in many iterations, the subgradient $h(x_n)$ should be effective direction if complemented with a careful choice of the steplength. Consequently, it seems promising to try to avoid the steplength tuning of the subgradient method by incorporating the first-order information about the dual function $q(\lambda)$ exploited by the cutting-plane method, while keeping the low computational burden of the subgradient method. This is the philosophy of the radar subgradient method presented herein.

The objective of the radar subgradient method is to maximize any concave function $q(\lambda)$ without constraints, as it is the case of the unconstrained dual problem (5). This method uses the same information as the cutting-plane method but in a different way. The supporting planes obtained in the course of the optimization give a first-order approximation of $q(\lambda)$. The cutting-plane method maximizes directly the function induced by the successive approximations of $q(\lambda)$, whereas the radar subgradient method uses the approximation to $q(\lambda)$ in order to compute the steplength along the subgradient direction. We used this idea to develop the radar gradient method (Ref. 19) within the framework of augmented Lagrangian relaxation, whose dual function is differentiable. The radar gradient

method was compared to the multiplier method from a practical point of view and the performances of the two method were similar.

Another multiplier updating method that shares the same philosophy (subgradient plus cutting planes) can be found in Ref. 20, although it is based on the bundle method (rather than on the basic cutting-plane method) and the resulting updating procedure is different from the one that we propose.

2.2. Geometry and Algebra. To derive the radar subgradient method, we distinguish three cases.

Case 1. Radar Steplength. The basic geometric intuition of the radar steplength is displayed in Figure 1.

Note that we use a one-dimensional $q(\lambda)$ to introduce the radar subgradient method. The central idea of the radar subgradient method is to perform a line search along the subgradient direction by using the first-order approximation to $q(\lambda)$. In the optimization process of $q(\lambda)$, we obtain the function values q_k and subgradients s_k of $q(\lambda)$, $k = 1, \ldots, n$, which allow us to build the supporting planes $\mathrm{SP}_1, \ldots, \mathrm{SP}_n$. The lower envelope of these supporting planes gives a first-order approximation of $q(\lambda)$. To simplify, in this one-dimensional example, we assume that the

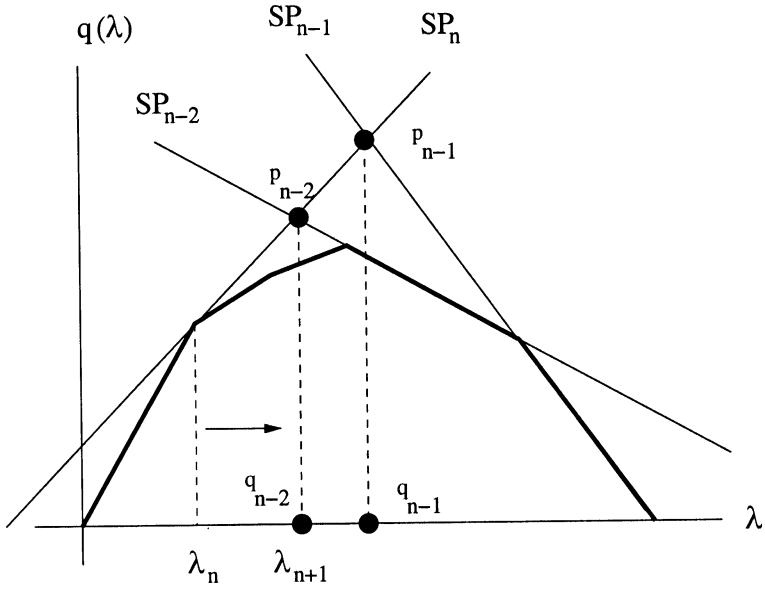

Fig. 1. Radar steplength.

first-order approximation to $q(\lambda)$ is based on three supporting planes SP_n, SP_{n-1} and SP_{n-2}. In this case, to compute the steplength from λ_n to λ_{n+1}, first we compute the points

$$p_{n-1} := SP_n \bigcap SP_{n-1} \quad \text{and} \quad p_{n-2} := SP_n \bigcap SP_{n-2}.$$

Second, we project p_{n-1} and p_{n-2} on the subgradient direction obtaining

$$q_{n-1} := \text{proj}(p_{n-1}) \quad \text{and} \quad q_{n-2} := \text{proj}(p_{n-2}).$$

Third, λ_{n+1} is defined as the closest projection to λ_n, that is, q_{n-2}.

Arbitrarily enough, we call the previous procedure the radar subgradient method considering that, as we can see in Figure 1, the method detects the first supporting plane that would be detected by a radar station pointing in the direction given by SP_n. Naturally, the radar steplength will refer to the steplength associated to the radar subgradient method.

In the following proposition, we show how to compute the radar steplength β_n. Note that, unlike in the bundle method or in the cutting-plane method, we update the vector of multipliers λ_n without solving any optimization problem.

We employ the following notation:

$\Lambda_n(\beta): \mathbb{R} \longmapsto \mathbb{R}^m$, vector-valued mapping defined as $\Lambda_n(\beta) = \lambda_n + \beta s_n$;
$R_n := \{\Lambda_n(\beta) \in \mathbb{R}^m : \beta \in \mathbb{R}\}$, straight line of candidates to λ_{n+1};
$y_{nk}(\beta): \mathbb{R} \longmapsto \mathbb{R}$, affine function defined as $y_{nk}(\beta) = q_k + s_k'(\Lambda_n(\beta) - \lambda_k)$, $k = 0, \ldots, n$;
$r_{nk} := \left\{ \binom{\beta}{y} \in \mathbb{R} \times \mathbb{R} : y = y_{nk}(\beta) \right\}$, straight line defined on SP_k when λ moves along R_n, $k = 0, \ldots, n$;
$(\beta_{nk}, \hat{y}_{nk}) \in \mathbb{R} \times \mathbb{R}$, intersection point of the straight lines r_{nk} and r_{nn}, $k = 0, \ldots, n-1$;
β_n steplength from λ_n to λ_{n+1} given by the radar subgradient method, i.e., $\lambda_{n+1} = \lambda_n + \beta_n s_n$;
Ω_n set of positive steps from λ_n; i.e., $\Omega_n := \{\beta_{nk} : \beta_{nk} > 0, k = 0, \ldots, n-1\}$;

Proposition 2.1. If $q(\lambda): \mathbb{R}^m \longmapsto \mathbb{R}$ is a concave function, then:

(a) If $(s_n - s_k)' s_n = 0$, then r_{nn} is parallel to r_{nk}; therefore, one should not compute β_{nk}.
(b) If $(s_n - s_k)' s_n \neq 0$, then r_{nn} is not parallel to r_{nk} and

$$\beta_{nk} = [q_k - q_n + (\lambda_n - \lambda_k)' s_k] / [(s_n - s_k)' s_n], \quad k = 0, \ldots, n-1. \qquad (6)$$

(c) If $\Omega_n \neq \emptyset$, the radar steplength can be computed as

$$\beta_n = \min \Omega_n. \tag{7}$$

Proof.

(a) If $(s_n - s_k)' s_n = 0$, then $s_n' s_n = s_k' s_n$, which means (as we will see in Proposition 2.2) that the straight lines r_{nn} and r_{nn} have the same slope, that is, r_{nn} and r_{nn} are parallel and it does not make sense to compute β_{nk}.

(b) Analogously to case (a), we have that r_{nn} and r_{nn} have different slopes; therefore, it makes sense to compute β_{nk} as follows. By definition,

$$\begin{aligned}
&\Lambda_n(\beta) = \lambda_n + \beta s_n,\\
&y_{nn}(\beta) = q_n + s_n'(\Lambda_n(\beta) - \lambda_n),\\
&y_{nk}(\beta) = q_k + s_k'(\Lambda_n(\beta) - \lambda_k).
\end{aligned}$$

Then,

$$\begin{aligned}
y_{nn}(\beta) &= q_n + s_n'(\lambda_n + \beta s_n - \lambda_n)\\
&= q_n + s_n' \beta s_n,\\
y_{nk}(\beta) &= q_k + s_k'(\lambda_n + \beta s_n - \lambda_k).
\end{aligned}$$

At the intersection point of r_{nn} and r_{nk}, it happens that $y_{nn}(\beta) = y_{nk}(\beta)$; hence,

$$\begin{aligned}
q_n + s_n' \beta s_n &= q_k + s_k'(\lambda_n + \beta s_n - \lambda_k),\\
q_n + \beta s_n' s_n &= q_k + s_k'(\lambda_n - \lambda_k) + \beta s_n' s_k,\\
\beta (s_n - s_k)' s_n &= q_k - q_n + (\lambda_n - \lambda_k)' s_k.
\end{aligned}$$

Therefore, the intersection point $(\beta_{nk}, \hat{y}_{nk})$ is defined by the step

$$\beta_{nk} = [q_k - q_n + (\lambda_n - \lambda_k)' s_k] / [(s_n - s_k)' s_n], \quad k = 0, \ldots, n-1.$$

(c) Given that the aim of the radar subgradient method is to maximize $q(\lambda)$ following the subgradient direction, only positive steps lengths β_{nk}, $k = 0, \ldots, n-1$, will be taken. We wish to stop with the first intersection point among the points $\{(\beta_{nk}, \hat{y}_{nk}) : k = 0, \ldots, n-1\}$; thus, the minimum positive β_{nk} must be chosen,

$$\beta_n = \min\{\beta_{nk} : \beta_{nk} > 0, k = 0, \ldots, n-1\} = \min \Omega_n. \qquad \square$$

Case 2. Positive and Negative Planes. In Figure 2, we repeat the computation of a radar step for the one-dimensional example; i.e., we compute the intersection of all the supporting planes with SP_n, we project these intersections on the subgradient direction, and we take λ_{n+1} as the closest projection to λ_n. Now, the difference is that we do not take into account SP_{n-1} because its slope has the same sign as the slope of SP_n. Our intuition is that taking into account supporting planes with the slope sign of SP_n would stop the radar subgradient progress prematurely. This intuition has been confirmed by preliminary computational tests.

Let us define m_{nk} as the slope of the straight line $r_{nk}, k=0,\ldots,n$. The next proposition shows how to compute m_{nk}.

Proposition 2.2. Let $q(\lambda):\mathbb{R}^m \longmapsto \mathbb{R}$ be a concave function and let us follow the notation of the above proposition. The slope of the straight line $r_{nk}, k=0,\ldots,n$, can be computed as

$$m_{nk} = s_n' s_k. \tag{8}$$

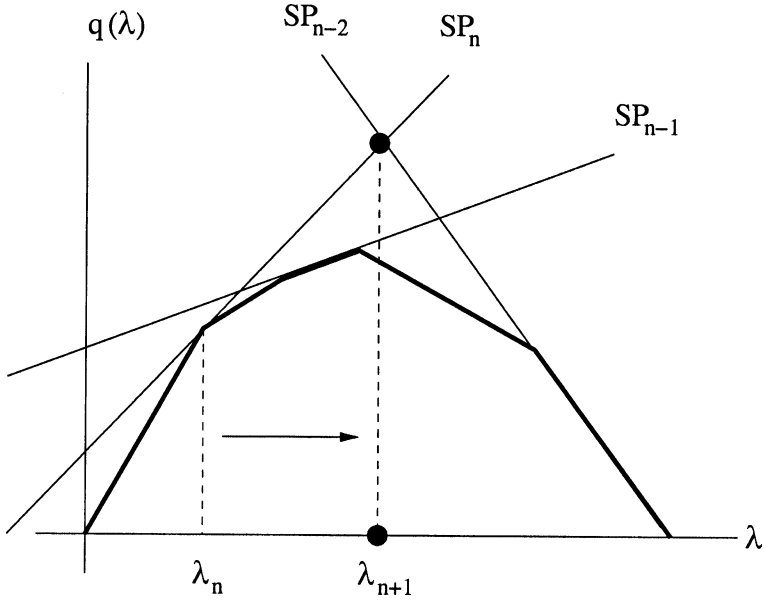

Fig. 2. Positive and negative planes.

Proof. The straight line r_{nk} is defined by the affine function

$$y_{nk}(\beta) = q_k + s_k'(\Lambda_n(\beta) - \lambda_k).$$

Let us express $y_{nk}(\beta)$ as $b_k + m_{nk}(\beta)$,

$$\begin{aligned} y_{nk}(\beta) &= q_k + s_k'(\Lambda_n(\beta) - \lambda_k) \\ &= q_k + s_k'(\lambda_n + \beta s_n - \lambda_k) \\ &= q_k + s_k'(\lambda_n - \lambda_k) + s_n' s_k \beta, \end{aligned}$$

whence

$$b_k = q_k + s_k'(\lambda_n - \lambda_k), \quad m_{nk} = s_n' s_k,$$

as we wanted to prove. □

Inspired by Figure 2 and the above proposition, we say that the supporting plane SP_k defined by the point (λ_k, q_k) and the subgradient $s_k, k < n$, is a positive plane relative to λ_n if r_{nk} has a positive slope, that is, if $m_{nk} > 0$. If m_{nk} is not greater than zero, we say that SP_k is a negative plane relative to λ_n.

Case 3. Lack of Negative Planes. If the first-order approximation of $q(\lambda)$ along the subgradient direction is based on only a set of positive planes (i.e., $\Omega_n = \emptyset$), then the radar steplength from λ_n to λ_{n+1} cannot be computed and arbitrary we take the classical subgradient steplength instead.

2.3. Algorithm. Joining the three cases discussed in the above section, the radar subgradient algorithm can be summarized as follows.

Radar Subgradient Algorithm.

Step 0. Initialize. Set $n = 0, \lambda_0$ and N_{iter}.

Step 1. Compute $q(\lambda_n)$. Compute q_n and $s_n \in \partial q(\lambda_n)$. Store s_n, q_n, λ_n.

Step 2. Check the stopping criterion. If λ_n does not improve for the last N_{iter} iterations or n reaches a prefixed value, then stop.

Step 3. Compute the steplength.
Compute $m_{nn} := s_n' s_n$.
For $k = 0, \ldots, n-1$:
Compute $m_{nk} := s_n' s_k$.

If $m_{nk} > 0$, reject the positive plane SP_k by setting $\beta_{nk} := -1$. Otherwise, if $m_{nn} - m_{nk} \neq 0$, compute

$$\beta_{nk} := [q_k - q_n + (\lambda_n - \lambda_k)' s_k]/(m_{nn} - m_{nk}).$$

There are two cases depending on

$$\Omega_n := \{\beta_{nk} : \beta_{nk} > 0,\ k = 0, \ldots, n-1\}.$$

(a) If $\Omega_n \neq \emptyset$, then set $\beta_n := \min \Omega_n$.
(b) If $\Omega_n = \emptyset$, then set $\beta_n := \alpha_n$, for a prefixed sequence $\{\alpha_n\}$.

Step 4. Compute $\lambda_{n+1} = \lambda_n + \beta_n s_n$. Set $n = n + 1$ and go back to Step 1.

In our implementation of the radar subgradient algorithm, we use sequences $\{\alpha_n\}$ such that $\lim_{n\to\infty} \alpha_n = 0$ and $\sum_{n=0}^{\infty} \alpha_n = +\infty$ (classical subgradient step length sequences). Furthermore, good computational results are obtained using pure classical subgradient steps, say, for the first 10 iterations, that is,

$$\beta_n = \alpha_n, \quad \text{for } n = 1, 2, \ldots, 10.$$

Note that, in Step 2, the radar subgradient method inherits the typical stopping criterion used by the classical subgradient method. In spite of its generalized use, with this criterion we cannot ensure the dual optimality of the best computed point. Nevertheless, in many applications, the accuracy obtained using this stopping criterion will suffice. For example, this is the case of the dual bound computing in combinatorial optimization.

3. Numerical Results

The objective of this section is to compare the performance of the subgradient method with the radar subgradient method. The two methods are compared by using a set of large-scale unit commitment instances. The aim of the unit commitment problem is to optimize electricity production, 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 for electricity at minimum cost and with a reliable service. Some examples of unit commitment literature are Refs. 19, 21, 22. The unit commitment problem decides, for each subperiod of the planning horizon, when a thermal unit is on or off, that is, when it is producing electricity and when it is shut down. These on/off decisions are of a binary

nature and turn the unit commitment problem into a difficult mixed integer programming problem. Solving the associated dual problem is a key point for most of the methodologies used to solve the unit commitment problem.

In this section, we perform a test where we solve eight instances of the dual problem associated to the unit commitment problem with duplicated variables (9), introduced in Ref. 23,

$$\begin{aligned} \min \quad & f(x,\tilde{x}) = C_{ht}(x) + C_m(\tilde{x}), && \text{(9a)} \\ \text{s.t.} \quad & x \in \mathcal{D}_{ht}, && \text{(9b)} \\ & \tilde{x} \in \mathcal{D}_m, && \text{(9c)} \\ & x - \tilde{x} = 0. && \text{(9d)} \end{aligned}$$

Here, $\mathcal{D}_{ht}$ represents the domain defined by the constraints that couple the hydroelectric and thermal systems: load constraints, spinning reserve constraints, etc.; $\mathcal{D}_m$ represents the domain of the management for the thermal units: minimum up and down times, minimum and maximum output levels, etc.; $C_{ht}(x)$ represents the costs associated with $\mathcal{D}_{ht}$; $C_m(x)$ represents the costs associated with $\mathcal{D}_m$. The complete description of this model can be found in Ref. 24, Chapter 2.

The dual problem that we solve corresponds to the relaxation of the equality constraint (9d), that is,

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^M} \quad \min \quad & C_{ht}(x) + C_m(\tilde{x}) + \lambda'(x - \tilde{x}), && \text{(10a)} \\ \text{s.t} \quad & x \in \mathcal{D}_{ht}, && \text{(10b)} \\ & \tilde{x} \in \mathcal{D}_m, && \text{(10c)} \end{aligned}$$

where M, the dimension of λ, is the product of the number of thermal units by the number of intervals.

In Table 1, we describe eight unit commitment instances, the main features of which range from very small size (2 intervals, 0 reservoirs, 2 thermal units, and 4 binary variables) up to medium size (168 intervals, 4 reservoirs, 11 thermal units, and 1848 binary variables).

This test consists of two steps. In Step 1, we solve the dual (10) for each unit commitment instance, using both the subgradient method and the radar subgradient method, to compare the quality of the computed optima. In Step 2, we compare the efficiency of the subgradient and radar subgradient methods in terms of CPU time.

Within the subgradient method, we use the Polyak step defined by $\beta_n = \alpha_n(\hat{q}_n - q_n)/\|s_n\|^2$, where $\alpha_n = \alpha_0/n, n > 0, q_n = q(\lambda_n)$ and $s_n \in \partial q(\lambda_n)$.

Table 1. Description of the unit commitment instances.

Case	Number of intervals	Number of intervals	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
8	168	4	11	9408	1848

Based on Ref. 1, $\hat{q}_n$, an approximation to the optimum q^*, is chosen as

$$\hat{q}_n = (1+\delta_n) \cdot \max_{i=1,\ldots,n} \{q_i\},$$

where δ_n is updated at each iteration as

$$\delta_{n+1} = \begin{cases} \min\{\max\{\underline{\delta}, \delta_n \cdot \Delta\delta\}, \bar{\delta}\}, & \text{if } q_n > q_{n-1}, \\ \min\{\max\{\underline{\delta}, \delta_n / \Delta\delta\}, \bar{\delta}\}, & \text{if } q_n \le q_{n-1}, \end{cases} \tag{11}$$

$\Delta\delta$ being a constant factor. $\underline{\delta}$ and $\bar{\delta}$ are, respectively, a lower and upper bound to δ_n; i.e., $\delta_n \in [\underline{\delta}, \bar{\delta}]$ for all n.

The stopping criterion used by both the subgradient algorithm and the radar subgradient algorithm is

$$\sum_{i=0}^{4} \left\| \lambda_{n-i} - \lambda_{n-i-1} \right\| \infty / 5 < \epsilon_\lambda; \tag{12}$$

i.e., both the subgradient and the radar subgradient methods stop whenever the average variation of λ_n for the last 5 iterations is small enough. We use $\epsilon_\lambda = 10^{-5}$ and a maximum number of multiplier updates equal to 1000 for both the subgradient method and the radar step method. The value of the parameter α_0 used in the radar subgradient method to set the first 10 steps is 10^{-3} for all cases except for Cases 1 and 2, where we have used 10^0. The best parameters that we have found for the Polyak step (11), after extensive computational tests, are

$$\alpha_0 = 5, \quad \delta_0 = 0.5, \quad \Delta\delta = 1.5, \quad \underline{\delta} = 0.1, \quad \bar{\delta} = 0.5$$

for all the eight cases.

An important difference between these two methods is the tuning process. The classical subgradient method with Polyak step, as we have implemented it, depends on the parameters $\alpha_0, \Delta\delta, \underline{\delta}, \bar{\delta}$, which may need a demanding tuning as it has been in this test. On the contrary, the radar subgradient method depends only on the parameter α_0, which is very easy to tune. This is because normally the radar subgradient method uses only α_0 in the first iterations, when the approximation to the dual function is still too poor to compute the radar step.

Results of Step 1. As can be appreciated in Table 2, the quality of the computed dual optima is very similar for the two methods. In general the subgradient methods obtain slightly better results; that is, on average, the subgradient optima are 1.1% better than the radar subgradient optima.

Results of Step 2. To compare the efficiency of the two methods, we compute the CPU time ratio for each case, which is on average 13.97. Thus, we can say that, in this test, on average, the subgradient method and the radar subgradient method have obtained similar dual bounds. Also on average, the radar subgradient time is less than nearly 1/14 of the time required by the subgradient method. Once again, it should be stressed that the number of iterations is not a good measure to compare the two methods. In this case, on average, the number of subgradient iterations is 26 times of radar subgradient iterations. However, the actual CPU time ratio is 13.97.

In Table 2, we can observe that the radar subgradient method obtains dual costs slightly worse than the subgradient method. Even if we impose a more strict stopping parameter ϵ_λ, the final radar subgradient dual cost does not improve. This illustrates the fact that the radar subgradient method may converge to a suboptimal point; therefore, no convergence

Table 2. Results using the radar subgradient (RS) method versus the subgradient (SG) method.

	Iterations		CPU time (sec)			Best dual cost q^* ($\times 10^6$ PTA)	
Case	SG	RS	SG	RS	Ratio	SG	RS
1	43	22	17.6	9.51	1.85	0.00374	0.00373
2	1000	30	297.2	9.40	31.61	5.99310	5.82620
3	774	25	418.3	17.6	23.76	0.97040	0.96149
4	1000	37	632.0	81.8	7.72	6.35305	6.32430
5	665	18	390.2	16.5	23.64	1.00943	1.00387
6	580	25	555.9	156.5	3.55	4.42936	4.28276
7	941	28	3975.4	497.3	7.99	2.53527	2.53023
8	820	40	203872.9	17454.4	11.68	85.67975	84.86457
Average	727.9	28.1	26269.9	2280.4	13.97	13.37	13.22

proof can be envisaged. Nevertheless, this computational analysis shows that the radar subgradient method can give a good approximation to the optimal dual function in much less time than the subgradient method.

Another interesting experiment would be to compare the radar subgradient solutions with the solutions obtained by the subgradient method after the same number of iterations. Table 3 presents the same information shown in Table 2, but truncating the subgradient method at the iteration where the radar subgradient reaches its best solution. Looking at the results in this table, it appears that the radar subgradient method has a faster approach to a good estimation of the optimal dual function than the truncated subgradient method. While the times spent by both methods are similar (average CPU time ratio of 1.055), the average of the relative dual error $(q^*_{SG} - q^*)/q^*_{SG}$, which measures the difference between the current solution q^* and q^*_{SG} (the solution found by the subgradient method in Table 2) is reduced by the radar subgradient method by a factor greater than 2; see the last two columns of Table 3. Figure 3 illustrate graphically this improvement of the dual function value for instance 7 of this computational test.

4. Conclusions

In this paper, we have introduced the radar subgradient method. The philosophy of the radar subgradient method is to incorporate into the subgradient method the first-order information about the dual function already exploited by the cutting-plane method. The cutting-plane method maximizes directly the successive approximations to the dual function, whereas the

Table 3. Comparison of the radar subgradient (RS) method with the truncated subgradient (TSG) method.

	Iterations		CPU time (sec)		Best dual cost q^* ($\times 10^6$ PTA)		Relative dual error $(q^*_{SG} - q^*)/q^*_{SG}$	
Case	TSG, RS	TSG	RS	Ratio TSG/RS	TSG	RS	TSG	RS
1	22	10.0	9.51	1.051	0.00373	0.00373	0.002%	0.235%
2	30	9.9	9.40	1.053	5.91992	5.8262	1.221%	2.784%
3	25	18.3	17.6	1.039	0.94358	0.96149	2.763%	0.918%
4	37	33.5	81.8	0.409	6.05337	6.32430	4.717%	0.452%
5	18	22.1	16.5	1.339	0.96279	1.00387	4.620%	0.550%
6	25	107.3	156.5	0.685	4.38704	4.28276	0.955%	3.309%
7	28	806.4	497.3	1.621	2.44635	2.53023	3.507%	0.198%
8	40	21635.4	17454.4	1.239	85.42933	84.86457	0.292%	0.951%
	Average	2830.4	2280.4	1.241	13.16826	13.22590	2.342%	1.017%

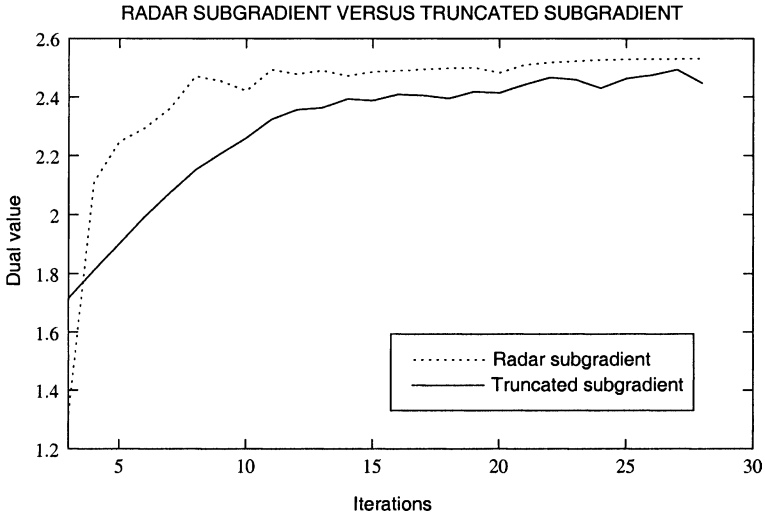

Fig. 3. Dual function evolution, Case 7 of the test.

radar subgradient method uses these approximations in order to compute an effective steplength following the subgradient direction. Thus, the tuning of the steplength sequence of the subgradient method is avoided, while the subgradient low computational burden is maintained.

In the tests which we carried out, based on the unit commitment problem, the new radar subgradient method outperformed the classical subgradient method from a practical point of view: the subgradient method obtained slightly better dual optima than the radar subgradient method, but on average the radar subgradient method reduced the execution CPU time by a factor of 14. That is, the radar subgradient method approaches the optimal set in a considerably faster way than the subgradient method.

Therefore, even if the radar subgradient method, in general, will give only a good approximation to the optimum, we can use it to effectively approach the optimal set. For some applications, this will be enough. For other applications, we will need to continue the optimization process by a more sophisticated method (bundle method for example) to achieve convergence to the optimum.

References

1. Bertsekas, D. P., *Nonlinear Programming*, 2nd Edition, Athena Scientific, Belmont, Massachusetts, 1999.

2. Luenberger, D. G., *Linear and Nonlinear Programming*, 2nd Edition, Addison-Wesley, Reading, Massachusetts, 1984.
3. Hiriart-Urruty, J. B., and Lemaréchal, C., *Convex Analysis and Minimization Algorithms*, Vols. I and II, Springer Verlag, Berlin, Germany, 1996.
4. Shor, N. Z., *Nondifferentiable Optimization and Polynomial Problems*, Nonconvex Optimization and Its Applications, Kluwer Academic Publishers, Dordrecht, Holland, 1998.
5. Feltenmark, S., *On Optimization of Power Production*, PhD Thesis, Department of Mathematics, Optimization, and Systems Theory, Royal Institute of Technology, Stockholm, Sweden, 1997.
6. Kelley, J. E., *The Cutting-Plane Method for Solving Convex Programs*, SIAM Journal on Applied Mathematics, Vol. 8, pp. 703–712, 1960.
7. Feltenmark, S., and Kiwiel, K. C., *Dual Applications of Proximal Bundle Methods, Including Lagrangian Relaxation of Nonconvex Problems*, SIAM Journal on Optimization, Vol. 10, pp. 697–721, 2000.
8. Kiwiel, K. C., *A Survey of Bundle Methods for Nondifferentiable Optimization*, Proceedings of the 13th International Symposium on Mathematical Programming, Tokyo, Japan, 1988.
9. Schramm, H., and Zowe, J., *A Combination of the Bundle Approach and the Trust-Region Concept*, Advances in Mathematical Optimization, Edited by J. Guddat, B. Bank, et al., Akademie Verlag, Berlin, Germany, 1988.
10. Schramm, H., and Zowe, J., *A Version of the Bundle Idea for Minimizing a Nonsmooth Function: Conceptual Idea, Convergence Analysis, Numerical Results*, SIAM Journal on Optimization, Vol. 2, pp. 121–152, 1992.
11. Du Merle, O., Goffin, J. L., et al., *On Improvements to the Analytic Center Cutting Plane Method*, Computational Optimization and Applications, Vol. 11, pp. 37–52, 1998.
12. Goffin, J. L., and Vial, J. P., *Convex Nondifferentiable Optimization: A Survey Focused on the Analytic Center Cutting Plane Method*, Technical Report 99.02, HEC, Logilab, Geneva University, 1999.
13. Gondzio, J., Sarkissian, R., et al., *Parallel Implementation of a Central Decomposition Method for Solving Large-Scale Planning Problems*, Computational Optimization and Applications, Vol. 19, pp. 5–29, 2001.
14. Nemirovsky, A. S., and Yudin, D. B., *Problem Complexity and Method Efficiency in Optimization*, John Wiley and Sons, New York, NY, 1983.
15. Baker, B. M., and Sheasby, J., *Accelerating the Convergence of Subgradient Optimization*, European Journal of Operational Research, Vol. 117, pp. 136–144, 1999.
16. Barahona, F., and Anbil, R., *The Volume Algorithm: Producing Primal Solutions with a Subgradient Method*, Mathematical Programming, Vol. 87, pp. 385–399, 2000.
17. Nedić, A., and Bertsekas, D. P., *Incremental Subgradient Methods for Nondifferentiable Oprimization*, SIAM Journal on Optimization, Vol. 12, pp. 109–138, 2001.

18. Kiwiel, K. C., Larsson, T., et al., *The Efficiency of Ballstep Subgradient Level Methods for Convex Optimization*, Mathematics of Operations Research, Vol. 24, pp. 237–254, 1999.
19. 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.
20. Lai, S., and Baldick, R., *Unit Commitment with Ramp Multipliers*, IEEE Transactions on Power Systems, Vol. 14, pp. 58–64, 1999.
21. 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.
22. 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.
23. Beltran, C., and Heredia, F. J., *Unit Commitment by Augmented Lagrangian Relaxation: Testing Two Decomposition Approaches*, Journal of Optimization Theory and Applications, Vol. 112, pp. 295–314, 2002.
24. Beltran, C., *Generalized Unit Commitment by the Radar Multiplier Method*, PhD Thesis, Statistics and Operations Research, Polytechnic University of Catalonia, Barcelona, Catalonia, Spain, 2001; see http://www.tdcat.cesca.es/TDCat-0720101-132715/, 2001.