Chapter 11
Cutting Plane Methods

Subgradient methods described in the previous chapter use only one arbitrary subgradient at a time, without memory of past iterations. If the information from previous iterations is kept, it is possible to define a model—the so-called cutting plane model—of the objective function. In this way, more information about the local behavior of the function is obtained than what an individual arbitrary subgradient can yield. The cutting plane idea was first developed independently in [60, 129]. In this chapter, we first introduce the basic ideas of the *standard cutting plane method* (CP) and then the more advanced *cutting plane method with proximity control* (CPPC) [86]. In addition, the history of the so-called bundle methods (see Chap. 12) originates from the cutting plane idea.

11.1 Standard Cutting Plane Method

In this section we describe the ideas of the standard cutting plane method (CP) by Kelley for convex nonsmooth minimization [129] (see also [60]). Due to the Theorem 2.30 in Part I, a convex function f has the representation

$$f(\boldsymbol{x}) = \max\{f(\boldsymbol{y}) + \boldsymbol{\xi}^T(\boldsymbol{x} - \boldsymbol{y}) \mid \boldsymbol{\xi} \in \partial f(\boldsymbol{y}),\ \boldsymbol{y} \in \mathbb{R}^n\} \text{ for all } \boldsymbol{x} \in \mathbb{R}^n. \quad (11.1)$$

However, for this representation we need the whole subdifferential $\partial f(\boldsymbol{y})$, which, in practice, is too big a requirement. For this reason we have to approximate it somehow. We now suppose that in addition to the current iteration point $\boldsymbol{x}_k$ we have some auxiliary points $\boldsymbol{x}_j \in \mathbb{R}^n$ and subgradients $\boldsymbol{\xi}_j \in \partial f(\boldsymbol{x}_j)$ for $j \in \mathcal{J}_k$, where the index set $\mathcal{J}_k$ is such that $\emptyset \neq \mathcal{J}_k \subset \{1, \ldots, k\}$. Now instead of Eq. (11.1) we can define a finite piecewise affine approximation of f at the current iteration k by

$$\hat{f}^k(\boldsymbol{x}) := \max\{f(\boldsymbol{x}_j) + \boldsymbol{\xi}_j^T(\boldsymbol{x} - \boldsymbol{x}_j) \mid j \in \mathcal{J}_k\} \quad \text{for all} \quad \boldsymbol{x} \in \mathbb{R}^n. \quad (11.2)$$

The minimization of the approximation $\hat{f}^k$ on a convex compact set S containing the minimum point of f gives a new iterate $\boldsymbol{x}_{k+1}$. By the definition of the approximation we have for all k

A. Bagirov et al., *Introduction to Nonsmooth Optimization*,
DOI: 10.1007/978-3-319-08114-4_11

$$\hat{f}^k(\boldsymbol{x}) \leq f(\boldsymbol{x}), \qquad \hat{f}^k(\boldsymbol{x}_j) = f(\boldsymbol{x}_j), \qquad \text{and} \quad \hat{f}^k(\boldsymbol{x}) \leq \hat{f}^{k+1}(\boldsymbol{x}).$$

The minimization of (11.2) can be transformed to a problem of finding a solution $(\boldsymbol{d}, v) \in \mathbb{R}^{n+1}$ to a linearly constrained smooth cutting plane problem

$$\begin{cases} \text{minimize} & v \\ \text{subject to} & -\alpha_j + \boldsymbol{\xi}^T \boldsymbol{d}_k \leq v \text{ for all } j \in \mathcal{J}_k \\ & \boldsymbol{x}_k + \boldsymbol{d} \in S \text{ and } v \in \mathbb{R}, \end{cases}$$

where α_j is the so-called *linearization error* between the actual value of the objective function at $\boldsymbol{x}_k$ and the linearization generated at $\boldsymbol{x}_j$ and evaluated at $\boldsymbol{x}_k$, that is,

$$\alpha_j := f(\boldsymbol{x}_k) - f(\boldsymbol{x}_j) + \boldsymbol{\xi}_j^T (\boldsymbol{x}_k - \boldsymbol{x}_j) \quad \text{for all } j \in \mathcal{J}_k.$$

and $s_i = \|\boldsymbol{x}_k - \boldsymbol{x}_j\|$. The cutting planes for some iterations of the CP are illustrated in Fig. 11.1.

Let us now suppose that we have a convex compact set S containing the minimum point of f available. The pseudo-code of the CP is the following:

```
PROGRAM CP
  INITIALIZE x_1 ∈ S, J_1 = {1}, and ε > 0;
  Set f̂^0(x) = −∞, α_1 = ∞ and k = 1;
  WHILE the termination condition α_k ≤ ε is not met
    Generate the search direction
      d_k = argmin_{x_k + d ∈ S} {f̂_k(x_k + d)};
    Find step size t^k;
    Set x_{k+1} = x_k + t^k d_k and compute α_{k+1} = f(x_{k+1}) − f̂_k(x_{k+1});
    Update J_k according some updating rules;
    Set k = k + 1;
  END WHILE
  RETURN final solution x_k;
END PROGRAM CP
```

Like in subgradient methods (see Chap. 10) the sequence $(\boldsymbol{x}_k)$ generated by the CP does not necessarily have decreasing objective values $f(\boldsymbol{x}_k)$. The step size t^k can be selected by using some kind of line-search procedure or just use a constant step size (e.g. $t^k = 1$). At the initial iterations the minimization of the cutting plane model $\hat{f}_k$ may be unbounded from below unless a suitable set S is introduced. Thus, the choice of set S is a key element to overcome the instability of cutting planes.

By the convexity of f, the graph of the cutting plane model $\hat{f}_k$ approaches the graph of f from below with increasing accuracy as k grows. This guarantees the global convergence of the method (for more details, see e.g. [42] Chap. 9). Furthermore, this property provides an implementable stopping criterion (note the decreasing values of the distances α_k in Fig. 11.1)

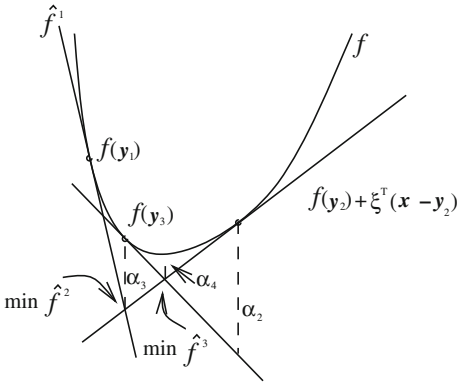

Fig. 11.1 Cutting planes

There are two main disadvantages in the CP: first, the choice of set S such that the minimization problem has a solution in the set and, second, the method generally attains rather poor convergence results in practice. However, it is also obvious, that if the original objective function f is piecewise linear or almost piecewise linear, then the cutting plane method may convergence in a reliable way and rapidly—in the piecewise linear case the convergence is finite—to the exact global minimum.

11.2 Cutting Plane Method with Proximity Control

The extension of the cutting plane method for nonconvex functions is not straightforward. A basic observation is that, in nonconvex case, the first order information does not necessarily provide the lower approximation of the objective function any longer. In this section, we briefly introduce the cutting plane method with proximity control (CPPC) for nonconvex NSO developed by Fuduli, Gaudioso, and Giallombardo. For more details, see [86].

Let us denote the set of available information—*the bundle*—as

$$(\boldsymbol{x}_j, f(\boldsymbol{x}_j), \boldsymbol{\xi}_j, \alpha_j, s_j), \quad j \in \mathcal{J}_k,$$

where again $\boldsymbol{x}_j \in \mathbb{R}^n$ are auxiliary points, $\boldsymbol{\xi}_j \in \partial f(\boldsymbol{x}_j)$ and α_j is a linearization error. Note that the linearization error α_j can be negative in nonconvex case.

The CPPC is based on the construction of both a lower and an upper polyhedral approximation of the objective function. That is, instead of just one index set $\mathcal{J}_k$ (cf. standard cutting plane method in Sect. 11.1), we have two sets $\mathcal{J}_+$ and $\mathcal{J}_-$ defined as follows:

$$\mathcal{J}_+ = \{j \mid \alpha_j \geq 0\} \quad \text{and} \quad \mathcal{J}_- = \{j \mid \alpha_j < 0\}.$$

The bundles defined by index sets $\mathcal{J}_+$ and $\mathcal{J}_-$ are characterized by points that somehow exhibit, respectively, the "convex behavior" and the "nonconvex behavior" of the objective function relative to point $\boldsymbol{x}_k$. Notice that the set $\mathcal{J}_+$ is never empty since at least the element $(\boldsymbol{x}_j, f(\boldsymbol{x}_j), \boldsymbol{\xi}_j, 0, 0)$ belongs to the bundle. The basic idea of the CPPC is to treat differently the two bundles in the construction of a piecewise affine model.

The proximity control [132] is introduced by defining the *proximal trajectory* $\boldsymbol{d}^\gamma$ of the piecewise affine function $\max_{j\in\mathcal{J}_+}\{\boldsymbol{\xi}_j^T\boldsymbol{d}-\alpha_j\}$. The optimal proximal trajectory $\boldsymbol{d}^\gamma$ is computed by solving a quadratic direction finding problem ($v \in \mathbb{R}$ and $\boldsymbol{d} \in \mathbb{R}^n$ are variables) parametrized by scalar $\gamma > 0$ (see the pseudo-code given below):

$$\begin{cases} \text{minimize} & \gamma v + \frac{1}{2}\|\boldsymbol{d}\|^2 \\ \text{subject to} & v \geq \boldsymbol{\xi}_j^T\boldsymbol{d} - \alpha_j, \quad j \in \mathcal{J}_+, \\ & v \leq \boldsymbol{\xi}_j^T\boldsymbol{d} - \alpha_j, \quad j \in \mathcal{J}_-. \end{cases} \tag{11.3}$$

In what follows we denote by $\boldsymbol{\xi}_t$ the subgradient computed at $\boldsymbol{x}_k + t\boldsymbol{d}_{\hat{\gamma}}$ and by α_t the corresponding linearization error, that is, $\alpha_t = f(\boldsymbol{x}_k) - f(\boldsymbol{x}_k + t\boldsymbol{d}_{\hat{\gamma}}) + t\boldsymbol{\xi}_t^T\boldsymbol{d}_{\hat{\gamma}}$. The pseudo-code of the CPPC is the following:

```
PROGRAM CPPC
  INITIALIZE x_1 ∈ R^n, ε > 0, δ > 0, m ∈ (0,1), ρ ∈ (m,1), and r ∈ (0,1);
  COMPUTE f(x_1) and ξ_1 ∈ ∂f(x_1) and set k = 1;
  SET the bundle (x_1, f(x_1), ξ_1, 0, 0), so that J_- = ∅ and J_+ = {1};
  MAIN ITERATION
    Initialize θ > 0, γ_min > 0 and γ_max > γ_min;
    WHILE the termination condition ||ξ_k|| ≤ ε is not met
      Solve (11.3) for increasing values of γ to obtain (v_k^γ, d_k^γ);
      Choose γ̂ = min{γ | γ ∈ (γ_min, γ_max) and f(x_k + d_k^γ) > f(x_k) + m v_k^γ}
        if it exists, otherwise, set γ̂ := γ_max;
      IF ||d_k^γ̂|| > θ THEN
        Set x_γ̂ = x_k + d_γ̂;
        Compute ξ_γ̂ ∈ ∂f(x_γ̂) and α_γ̂ = f(x_k) − f(x_γ̂) + ξ_γ̂^T d_γ̂;
        BUNDLE INSERTION
          IF α_γ̂ < 0 and ||d_γ̂|| > δ THEN
            Insert (x_γ̂, f(γ̂), ξ_γ̂, α_γ̂, ||d_γ̂||) in the bundle with j ∈ J_-;
            Set γ̂ = γ̂ − r(γ̂ − γ_min);
          ELSE IF ξ_γ̂^T d_γ̂ ≥ ρ v_γ̂ THEN
            Insert (x_γ̂, f(x_γ̂), ξ_γ̂, max{0, α_γ̂}, ||d_γ̂||) in the bundle
              with j ∈ J_+;
          ELSE
            Find step size t ∈ (0,1) such that ξ_t^T d_γ̂ ≥ ρ v_γ̂;
```

```
            Insert (x_k + t d_ŷ, f(x_k + t d_ŷ), ξ_t, max{0, α_t}, t‖d_ŷ‖) in the
              bundle with j ∈ J_+;
          END IF
        END BUNDLE INSERTION
        IF ‖d_ŷ‖ ≤ θ go to BUNDLE DELETION;
        IF f(x_ŷ) ≤ f(y) + m v_ŷ THEN
          Set the new stability center x_{k+1} = x_ŷ;
        ELSE
          Solve (11.3) with γ = ŷ to obtain (v_k^ŷ, d_k^ŷ);
          Go to BUNDLE INSERTION;
        END IF
      ELSE
        BUNDLE DELETION
        Set J_+ := J_+\{j ∈ J_+ | s_j > δ} and J_- := J_-\{j ∈ J_- | s_j > δ};
        Compute ξ* = min_{ξ∈{ξ_j | j∈J_+}} ‖ξ‖;
        IF ‖ξ*‖ ≤ ε THEN
          STOP with the solution x_k + d_k^ŷ;
        ELSE
          Set γ_max := γ_max − r(γ_max − γ_min);
        END IF
      END IF
    END WHILE
    Update J_+ and J_- with respect to x_{k+1};
    Set k = k + 1 and go to next MAIN ITERATION;
  END MAIN ITERATION
  RETURN final solution x_k;
END PROGRAM CPPC
```

The global convergence of the CPPC to a stationary point is proved for weakly semi-smooth objective functions [86].