

# *Convex Optimization Theory*

*Athena Scientific, 2009*

by

Dimitri P. Bertsekas

Massachusetts Institute of Technology

## *Supplementary Chapter 6 on Convex Optimization Algorithms*

This chapter aims to supplement the book *Convex Optimization Theory*, Athena Scientific, 2009 with material on convex optimization algorithms. The chapter will be periodically updated. This version is dated

October 8, 2010

Your comments and suggestions to the author at [dimitrib@mit.edu](mailto:dimitrib@mit.edu) are welcome.

# 6

## *Convex Optimization*

### *Algorithms*

#### Contents

6.1. Special Problem Structures . . . . .	p. 250
6.1.1. Lagrange Dual Problems . . . . .	p. 250
6.1.2. Second Order Cone and Semidefinite Programming . . . . .	p. 255
6.2. Algorithmic Descent - Steepest Descent . . . . .	p. 266
6.3. Subgradient Methods . . . . .	p. 269
6.4. Polyhedral Approximation Methods . . . . .	p. 281
6.4.1. Outer Linearization - Cutting Plane Methods . . . . .	p. 281
6.4.2. Inner Linearization - Simplicial Decomposition . . . . .	p. 288
6.4.3. Duality of Outer and Inner Linearization . . . . .	p. 291
6.4.4. Generalized Simplicial Decomposition . . . . .	p. 293
6.4.5. Generalized Polyhedral Approximation . . . . .	p. 298
6.5. Proximal and Bundle Methods . . . . .	p. 311
6.5.1. Proximal Point Algorithm . . . . .	p. 313
6.5.2. Proximal Cutting Plane Method . . . . .	p. 322
6.5.3. Bundle Methods . . . . .	p. 324
6.6. Dual Proximal Point Algorithms . . . . .	p. 329
6.6.1. Augmented Lagrangian Methods . . . . .	p. 332
6.6.2. Proximal Inner Linearization Methods . . . . .	p. 335
6.7. Interior Point Methods . . . . .	p. 337
6.7.1. Primal-Dual Methods for Linear Programming . . . . .	p. 340
6.7.2. Interior Point Methods for Conic Programming . . . . .	p. 346
6.8. Approximate Subgradient Methods . . . . .	p. 347
6.8.1. $\epsilon$ -Subgradient Methods . . . . .	p. 347
6.8.2. Incremental Subgradient Methods . . . . .	p. 350
6.8.3. Subgradient Methods with Randomization . . . . .	p. 358
6.8.4. Incremental Proximal Methods . . . . .	p. 364
6.9. Optimal Algorithms and Complexity . . . . .	p. 369
6.10. Notes, Sources, and Exercises . . . . .	p. 380

In this supplementary chapter, we discuss several types of algorithms for minimizing convex functions. A major type of problem that we aim to solve is dual problems, which by their nature involve convex nondifferentiable minimization. The fundamental reason is that the negative of the dual function in the MC/MC framework is typically a conjugate function (cf. Section 4.2.1), which is generically closed and convex, but often nondifferentiable (it is differentiable only at points where the supremum in the definition of conjugacy is uniquely attained; cf. Prop. 5.4.3). Accordingly most of the minimization algorithms that we discuss do not require differentiability for their application. We refer to general nonlinear programming textbooks for methods that rely on differentiability, such as gradient and Newton-type methods.

## 6.1 SPECIAL PROBLEM STRUCTURES

Convex optimization algorithms have a broad range of application, but they are particularly useful for large/challenging problems with special structure, usually connected in some way to duality. We discussed in Chapter 5 two important duality structures. The first is Lagrange duality for constrained optimization, which arises by assigning dual variables to inequality constraints. The second is Fenchel duality together with its special case, conic duality. Both of these duality structures arise often in applications, and in this section we discuss some examples.

### 6.1.1 Lagrange Dual Problems

We first focus on Lagrange duality (cf. Sections 5.3.1-5.3.4). It involves the problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && x \in X, \quad g(x) \leq 0, \end{aligned} \tag{6.1}$$

where  $X$  is a set,  $g(x) = (g_1(x), \dots, g_r(x))'$ , and  $f : X \mapsto \Re$  and  $g_j : X \mapsto \Re$ ,  $j = 1, \dots, r$ , are given functions. The dual problem is

$$\begin{aligned} & \text{maximize} && q(\mu) \\ & \text{subject to} && \mu \in \Re^r, \quad \mu \geq 0, \end{aligned} \tag{6.2}$$

where the dual function  $q$  is given by

$$q(\mu) = \inf_{x \in X} L(x, \mu), \quad \mu \geq 0, \tag{6.3}$$

and  $L$  is the Lagrangian function defined by

$$L(x, \mu) = f(x) + \mu'g(x), \quad x \in X, \quad \mu \geq 0.$$

While in Section 5.3 we made various convexity assumptions on  $X$ ,  $f$ , and  $g_j$  in order to derive strong duality theorems, the dual problem can be defined even in the absence of such assumptions. Then the optimal dual value yields a lower bound on the optimal value of the primal problem (6.1) (cf. the Weak Duality Theorem of Prop. 4.1.2). This is significant in the context of discrete optimization as described in the following example.

**Example 6.1.1: (Discrete Optimization and Lower Bounds)**

Many practical optimization problems of the form (6.1) have a finite constraint set  $X$ . An example is *integer programming*, where the components of  $x$  must be integers from a bounded range (usually 0 or 1). An important special case is the linear 0-1 integer programming problem

$$\begin{aligned} &\text{minimize} && c'x \\ &\text{subject to} && Ax \leq b, \quad x_i = 0 \text{ or } 1, \quad i = 1, \dots, n. \end{aligned}$$

A principal approach for solving such problems is the *branch-and-bound method*, which is described in many sources. This method relies on obtaining lower bounds to the optimal cost of restricted problems of the form

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in \tilde{X}, \quad g(x) \leq 0, \end{aligned}$$

where  $\tilde{X}$  is a subset of  $X$ ; for example in the 0-1 integer case where  $X$  specifies that all  $x_i$  should be 0 or 1,  $\tilde{X}$  may be the set of all 0-1 vectors  $x$  such that one or more components  $x_i$  are restricted to satisfy  $x_i = 0$  for all  $x \in \tilde{X}$  or  $x_i = 1$  for all  $x \in \tilde{X}$ . These lower bounds can often be obtained by finding a dual-feasible (possibly dual-optimal) solution  $\mu$  of this problem and the corresponding dual value

$$q(\mu) = \inf_{x \in \tilde{X}} \{f(x) + \mu'g(x)\}, \tag{6.4}$$

which by weak duality, is a lower bound to the optimal value of the restricted problem  $\min_{x \in \tilde{X}, g(x) \leq 0} f(x)$ . Because  $\tilde{X}$  is finite,  $q$  is a concave piecewise linear function, and solving the dual problem amounts to minimizing the polyhedral function  $-q$  over the nonnegative orthant.

Let us now outline some problem structures that involve convexity, and arise frequently in applications.

**Separable Problems - Decomposition**

Consider the problem

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^n f_i(x_i) \\ &\text{subject to} && a'_j x \leq b_j, \quad j = 1, \dots, r, \end{aligned} \tag{6.5}$$

where  $x = (x_1, \dots, x_n)$ , each  $f_i : \mathfrak{R} \mapsto \mathfrak{R}$  is a convex function of the single scalar component  $x_i$ , and  $a_j$  and  $b_j$  are some vectors and scalars, respectively. Then by assigning a dual variable  $\mu_j$  to the constraint  $a_j'x \leq b_j$ , we obtain the dual problem [cf. Eq. (6.2)]

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^n q_i(\mu) - \sum_{j=1}^r \mu_j b_j \\ & \text{subject to} && \mu \geq 0, \end{aligned} \tag{6.6}$$

where

$$q_i(\mu) = \inf_{x_i \in \mathfrak{R}} \left\{ f_i(x_i) + x_i \sum_{j=1}^r \mu_j a_{ji} \right\},$$

and  $\mu = (\mu_1, \dots, \mu_r)$ . Note that the minimization involved in the calculation of the dual function has been decomposed into  $n$  simpler minimizations. These minimizations are often conveniently done either analytically or computationally, in which case the dual function can be easily evaluated. This is the key advantageous structure of separable problems: it facilitates computation of dual function values (as well as subgradients as we will see in Section 6.3), and it is amenable to decomposition/distributed computation.

There are also other separable problems that are more general than the one of Eq. (6.5). An example is when  $x$  has  $m$  components  $x_1, \dots, x_m$  of dimensions  $n_1, \dots, n_m$ , respectively, and the problem has the form

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m f_i(x_i) \\ & \text{subject to} && \sum_{i=1}^m g_i(x_i) \leq 0, \quad x_i \in X_i, \quad i = 1, \dots, m, \end{aligned} \tag{6.7}$$

where  $f_i : \mathfrak{R}^{n_i} \mapsto \mathfrak{R}$  and  $g_i : \mathfrak{R}^{n_i} \mapsto \mathfrak{R}^r$  are given functions, and  $X_i$  are given subsets of  $\mathfrak{R}^{n_i}$ . The advantage of convenient computation of the dual function value using decomposition extends to such problems as well.

### Partitioning

An important point regarding large-scale optimization problems is that there are several different ways to introduce duality in their solution. For example an alternative strategy to take advantage of separability, often called *partitioning*, is to divide the variables in two subsets, and minimize first with respect to one subset while taking advantage of whatever simplification may arise by fixing the variables in the other subset. In particular, the problem

$$\begin{aligned} & \text{minimize} && F(x) + G(y) \\ & \text{subject to} && Ax + By = c, \quad x \in X, \quad y \in Y, \end{aligned}$$

can be written as

$$\begin{aligned} & \text{minimize } F(x) + \inf_{By=c-Ax, y \in Y} G(y) \\ & \text{subject to } x \in X, \end{aligned}$$

or

$$\begin{aligned} & \text{minimize } F(x) + p(c - Ax) \\ & \text{subject to } x \in X, \end{aligned}$$

where  $p$  is the primal function of the minimization problem involving  $y$  above:

$$p(u) = \inf_{By=u, y \in Y} G(y);$$

(cf. Section 4.2.3). This primal function and its subgradients can often be conveniently calculated using duality.

### Additive Cost Problems

We now focus on a structural characteristic of dual problems that arises also in other contexts: a cost function that is the sum of a large number of components,

$$f(x) = \sum_{i=1}^m f_i(x), \quad (6.8)$$

where the functions  $f_i : \mathfrak{R}^n \mapsto \mathfrak{R}$  are convex. Such functions can be minimized with specialized methods, called *incremental*, which exploit their additive structure (see Section 6.8.2).

An important special case is the cost function of the dual/separable problem (6.6); after a sign change to convert to minimization it takes the form (6.8). Additive functions also arise in other important contexts. For example, in data analysis/machine learning problems, where each term  $f_i(x)$  corresponds to error between data and the output of a parametric model, with  $x$  being a vector of parameters. A classical example is least squares problems, where  $f_i$  has a quadratic structure, but other types of cost functions, including nondifferentiable ones, have become increasingly important. An example is the so called  *$\ell_1$ -regularization problem*

$$\begin{aligned} & \text{minimize } \sum_{j=1}^m (a'_j x - b_j)^2 + \gamma \sum_{i=1}^n |x_i| \\ & \text{subject to } (x_1, \dots, x_n) \in \mathfrak{R}^n, \end{aligned}$$

(sometimes called the *lasso method*), which arises in statistical inference. Here  $a_j$  and  $b_j$  are given vectors and scalars, respectively, and  $\gamma$  is a positive scalar. The nondifferentiable penalty term affects the solution in a different

way than a quadratic penalty (it tends to set a large number of components of  $x$  to 0).

For a related context where additive cost functions arise, consider the minimization of the expected value

$$E\{F(x, w)\},$$

where  $w$  is a random variable taking a finite but very large number of values  $w_i, i = 1, \dots, m$ , with corresponding probabilities  $\pi_i$ . Then the cost function consists of the sum of the  $m$  functions  $\pi_i F(x, w_i)$ .

Finally, let us note that cost functions that are separable or additive arise in the class of problems collectively known as *stochastic programming*. For a classical example, consider the case where a vector  $x \in X$  is selected, a random event occurs that has  $m$  possible outcomes  $w_1, \dots, w_m$ , and then another vector  $y \in Y$  is selected with knowledge of the outcome that occurred. Then for optimization purposes, we need to specify a different vector  $y_i \in Y$  for each outcome  $w_i$ . The problem is to minimize the expected cost

$$F(x) + \sum_{i=1}^m \pi_i G_i(y_i),$$

where  $G_i(y_i)$  is the cost associated with the occurrence of  $w_i$  and  $\pi_i$  is the corresponding probability. This is a problem with additive cost function. Furthermore, if there are linear constraints coupling the vectors  $x$  and  $y_i$ , the problem has a separable form.

### Large Number of Constraints

Problems of the form

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && a'_j x \leq b_j, \quad j = 1, \dots, r, \end{aligned} \tag{6.9}$$

where the number  $r$  of constraints is very large often arise in practice, either directly or via reformulation from other problems. They can be handled in a variety of ways. One possibility is to adopt a penalty function approach, and replace problem (6.9) with

$$\begin{aligned} & \text{minimize} && f(x) + c \sum_{j=1}^r P(a'_j x - b_j) \\ & \text{subject to} && x \in \mathfrak{R}^n, \end{aligned} \tag{6.10}$$

where  $P(\cdot)$  is a scalar penalty function satisfying  $P(t) = 0$  if  $t \leq 0$ , and  $P(t) > 0$  if  $t > 0$ , and  $c$  is a positive penalty parameter. For example, one may use the quadratic penalty

$$P(t) = (\max\{0, t\})^2.$$

An interesting alternative is to use

$$P(t) = \max\{0, t\},$$

in which case it can be shown that the optimal solutions of problems (6.5) and (6.10) coincide when  $c$  is sufficiently large (see e.g., [Ber99], Section 5.4.5, [BNO03], Section 7.3). The cost function of the penalized problem (6.10) is of the additive form (6.8).

Another possibility, which points the way to some major classes of algorithms, is to initially discard some of the constraints, solve the corresponding less constrained problem, and later reintroduce constraints that seem to be violated at the optimum. This is known as an *outer approximation* of the constraint set; see the cutting plane algorithms of Section 6.4.1. Another possibility is to use an *inner approximation* of the constraint set consisting for example of the convex hull of some of its extreme points; see the simplicial decomposition methods of Sections 6.4.2 and 6.4.4.

### 6.1.2 Second Order Cone and Semidefinite Programming

Another major problem structure is the conic programming problem discussed in Section 5.3.6:

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && x \in C, \end{aligned} \tag{6.11}$$

where  $f : \mathbb{R}^n \mapsto (-\infty, \infty]$  is a proper convex function and  $C$  is a convex cone in  $\mathbb{R}^n$ . An important special case, called *linear conic problem*, arises when  $\text{dom}(f)$  is an affine set and  $f$  is linear over  $\text{dom}(f)$ , i.e.,

$$f(x) = \begin{cases} c'x & \text{if } x \in b + S, \\ \infty & \text{if } x \notin b + S, \end{cases}$$

where  $b$  and  $c$  are given vectors, and  $S$  is a subspace. Then the primal problem can be written as

$$\begin{aligned} & \text{minimize} && c'x \\ & \text{subject to} && x - b \in S, \quad x \in C. \end{aligned} \tag{6.12}$$

To derive the dual problem, we note that

$$\begin{aligned} f^*(\lambda) &= \sup_{x-b \in S} (\lambda - c)'x \\ &= \sup_{y \in S} (\lambda - c)'(y + b) \\ &= \begin{cases} (\lambda - c)'b & \text{if } \lambda - c \in S^\perp, \\ \infty & \text{if } \lambda - c \notin S^\perp. \end{cases} \end{aligned}$$



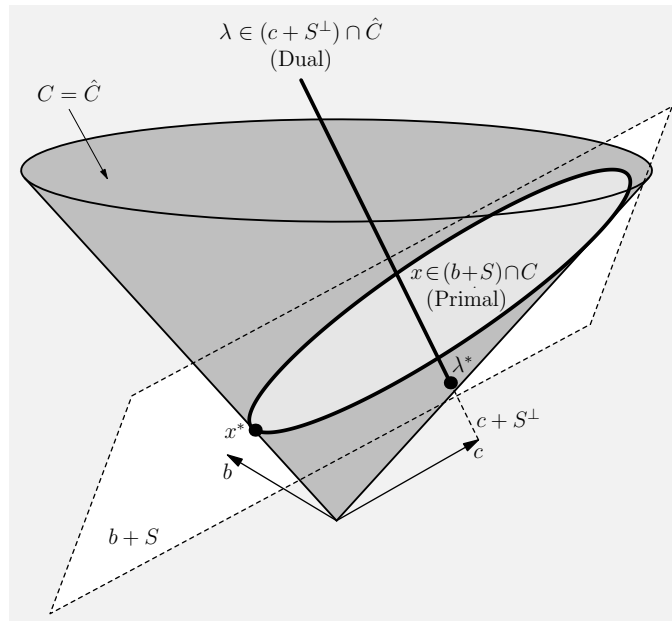
It can be seen that the dual problem  $\min_{\lambda \in \hat{C}} f^*(\lambda)$  (cf. Section 5.3.6), after discarding the superfluous term  $c'b$  from the cost, can be written as

$$\begin{aligned} &\text{minimize} && b'\lambda \\ &\text{subject to} && \lambda - c \in S^\perp, \quad \lambda \in \hat{C}, \end{aligned} \tag{6.13}$$

where  $\hat{C}$  is the dual cone:

$$\hat{C} = \{\lambda \mid x'\lambda \geq 0, \forall x \in C\};$$

cf. Fig. 6.1.1.



**Figure 6.1.1.** Illustration of primal and dual linear conic problems for the case of a 3-dimensional problem, 2-dimensional subspace  $S$ , and a self-dual cone ( $C = \hat{C}$ ); cf. Eqs. (6.12) and (6.13).

The following proposition translates the conditions of Prop. 5.3.9 to the polyhedral conic duality context.

**Proposition 6.1.1: (Linear-Conic Duality Theorem)** Assume that the primal problem (6.12) has finite optimal value. Assume further that either  $(b + S) \cap \text{ri}(C) = \emptyset$  or  $C$  is polyhedral. Then, there is no duality gap and the dual problem has an optimal solution.

**Proof:** Under the condition  $(b + S) \cap \text{ri}(C) = \emptyset$ , the result follows from Prop. 5.3.9. For the case where  $C$  is polyhedral, the result follows from the more refined version of the Fenchel Duality Theorem (Prop. 5.3.8), discussed at the end of Section 5.3.5. **Q.E.D.**

### Special Forms of Linear Conic Problems

The primal and dual linear conic problems (6.12) and (6.13) have been placed in an elegant symmetric form. There are also other useful formats that parallel and generalize similar formats in linear programming (cf. Example 4.2.1 and Section 5.2). For example, we have the following dual problem pairs:

$$\min_{Ax=b, x \in C} c'x \quad \iff \quad \max_{c-A'\lambda \in \hat{C}} b'\lambda, \quad (6.14)$$

$$\min_{Ax=b \in C} c'x \quad \iff \quad \max_{A'\lambda=c, \lambda \in \hat{C}} b'\lambda, \quad (6.15)$$

where  $x \in \mathfrak{R}^n$ ,  $\lambda \in \mathfrak{R}^m$ ,  $c \in \mathfrak{R}^n$ ,  $b \in \mathfrak{R}^m$ , and  $A$  is an  $m \times n$  matrix.

To verify the duality relation (6.14), let  $\bar{x}$  be any vector such that  $A\bar{x} = b$ , and let us write the primal problem on the left in the primal conic form (6.12) as

$$\begin{aligned} &\text{minimize} && c'x \\ &\text{subject to} && x - \bar{x} \in N(A), \quad x \in C, \end{aligned} \quad (6.16)$$

where  $N(A)$  is the nullspace of  $A$ . The corresponding dual conic problem (6.13) is to solve for  $\mu$  the problem

$$\begin{aligned} &\text{minimize} && \bar{x}'\mu \\ &\text{subject to} && \mu - c \in N(A)^\perp, \quad \mu \in \hat{C}. \end{aligned} \quad (6.17)$$

Since  $N(A)^\perp$  is equal to  $\text{Ra}(A')$ , the range of  $A'$ , the constraints of problem (6.17) can be equivalently written as  $c - \mu \in -\text{Ra}(A') = \text{Ra}(A')$ ,  $\mu \in \hat{C}$ , or

$$c - \mu = A'\lambda, \quad \mu \in \hat{C},$$

for some  $\lambda \in \mathfrak{R}^m$ . Making the change of variables  $\mu = c - A'\lambda$ , the dual problem (6.17) can be written as

$$\begin{aligned} &\text{minimize} && \bar{x}'(c - A'\lambda) \\ &\text{subject to} && c - A'\lambda \in \hat{C}. \end{aligned}$$

By discarding the constant  $\bar{x}'c$  from the cost function, using the fact  $A\bar{x} = b$ , and changing from minimization to maximization, we see that this dual

problem is equivalent to the one in the right-hand side of the duality pair (6.14). The duality relation (6.15) is proved similarly.

We next discuss two important special cases of conic programming: *second order cone programming* and *semidefinite programming*. These problems involve some special cones, and an explicit definition of the affine set constraint. They arise in a variety of practical settings, and their computational difficulty tends to lie between that of linear and quadratic programming on one hand, and general convex programming on the other hand.

### Second Order Cone Programming

Consider the cone

$$C = \left\{ (x_1, \dots, x_n) \mid x_n \geq \sqrt{x_1^2 + \dots + x_{n-1}^2} \right\},$$

known as the *second order cone* (see Fig. 6.1.2). The dual cone is

$$\hat{C} = \{y \mid 0 \leq y'x, \forall x \in C\} = \left\{ y \mid 0 \leq \inf_{\|(x_1, \dots, x_{n-1})\| \leq x_n} y'x \right\},$$

and it can be shown that  $\hat{C} = C$ . This property is referred to as *self-duality* of the second order cone, and is fairly evident from Fig. 6.1.2. For a proof, we write

$$\begin{aligned} \inf_{\|(x_1, \dots, x_{n-1})\| \leq x_n} y'x &= \inf_{x_n \geq 0} \left\{ y_n x_n + \inf_{\|(x_1, \dots, x_{n-1})\| \leq x_n} \sum_{i=1}^{n-1} y_i x_i \right\} \\ &= \inf_{x_n \geq 0} \{ y_n x_n - \|(y_1, \dots, y_{n-1})\| x_n \} \\ &= \begin{cases} 0 & \text{if } \|(y_1, \dots, y_{n-1})\| \leq y_n, \\ -\infty & \text{otherwise.} \end{cases} \end{aligned}$$

Combining the last two relations, we have

$$y \in \hat{C} \quad \text{if and only if} \quad 0 \leq y_n - \|(y_1, \dots, y_{n-1})\|,$$

so  $\hat{C} = C$ .

Note that linear inequality constraints of the form  $a'_i x - b_i \geq 0$  can be written as

$$\begin{pmatrix} 0 \\ a'_i \end{pmatrix} x - \begin{pmatrix} 0 \\ b_i \end{pmatrix} \in C_i,$$

where  $C_i$  is the second order cone of  $\Re^2$ . As a result, linear conic problems involving second order cones contain as special cases linear programming problems.

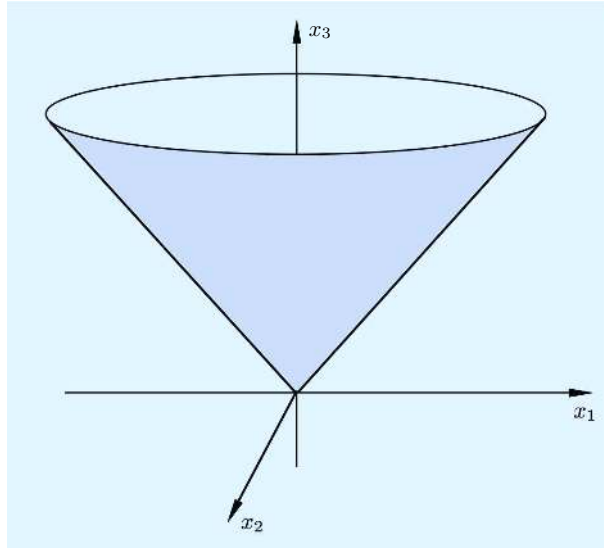


Figure 6.1.2. The second order cone

$$C = \left\{ (x_1, \dots, x_n) \mid x_n \geq \sqrt{x_1^2 + \dots + x_{n-1}^2} \right\},$$

in  $\mathbb{R}^3$ .

The second order cone programming problem (SOCP for short) is

$$\begin{aligned} & \text{minimize} && c'x \\ & \text{subject to} && A_i x - b_i \in C_i, \quad i = 1, \dots, m, \end{aligned} \tag{6.18}$$

where  $x \in \mathbb{R}^n$ ,  $c$  is a vector in  $\mathbb{R}^n$ , and for  $i = 1, \dots, m$ ,  $A_i$  is an  $n_i \times n$  matrix,  $b_i$  is a vector in  $\mathbb{R}^{n_i}$ , and  $C_i$  is the second order cone of  $\mathbb{R}^{n_i}$ . It is seen to be a special case of the primal problem in the left-hand side of the duality relation (6.15), where

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_m \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}, \quad C = C_1 \times \dots \times C_m.$$

Thus from the right-hand side of the duality relation (6.15), we see that the corresponding dual linear conic problem has the form

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^m b'_i \lambda_i \\ & \text{subject to} && \sum_{i=1}^m A'_i \lambda_i = c, \quad \lambda_i \in C_i, \quad i = 1, \dots, m, \end{aligned} \tag{6.19}$$

where  $\lambda = (\lambda_1, \dots, \lambda_m)$ . By applying the duality result of Prop. 6.1.1, we have the following proposition.

**Proposition 6.1.2: (Second Order Cone Duality Theorem)**

Consider the primal SOCP (6.18), and its dual problem (6.19).

- (a) If the optimal value of the primal problem is finite and there exists a feasible solution  $\bar{x}$  such that

$$A_i \bar{x} - b_i \in \text{int}(C_i), \quad i = 1, \dots, m,$$

then there is no duality gap, and the dual problem has an optimal solution.

- (b) If the optimal value of the dual problem is finite and there exists a feasible solution  $\bar{\lambda} = (\bar{\lambda}_1, \dots, \bar{\lambda}_m)$  such that

$$\bar{\lambda}_i \in \text{int}(C_i), \quad i = 1, \dots, m,$$

then there is no duality gap, and the primal problem has an optimal solution.

Note that while Prop. 6.1.1 requires a relative interior point condition, the preceding proposition requires an interior point condition. The reason is that the second order cone has nonempty interior, so its relative interior coincides with its interior.

The SOCP arises in many application contexts, and significantly, it can be solved numerically with powerful specialized algorithms that belong to the class of interior point methods, discussed in Section 6.7. We refer to the literature for a more detailed description and analysis (see e.g., Ben-Tal and Nemirovski [BeT01], and Boyd and Vandenbergue [BoV04]).

Generally, SOCPs can be recognized from the presence of convex quadratic functions in the cost or the constraint functions. The following are illustrative examples.

**Example 6.1.2: (Robust Linear Programming)**

Frequently, there is uncertainty about the data of an optimization problem, so one would like to have a solution that is adequate for a whole range of the uncertainty. A popular formulation of this type, is to assume that the constraints contain parameters that take values in a given set, and require that the constraints are satisfied for all values in that set.

As an example, consider the problem

$$\begin{aligned} & \text{minimize} && c'x \\ & \text{subject to} && a'_j x \leq b_j, \quad \forall (a_j, b_j) \in T_j, \quad j = 1, \dots, r, \end{aligned} \tag{6.20}$$

where  $c \in \mathfrak{R}^n$  is a given vector, and  $T_j$  is a given subset of  $\mathfrak{R}^{n+1}$  to which the constraint parameter vectors  $(a_j, b_j)$  must belong. The vector  $x$  must be chosen so that the constraint  $a'_j x \leq b_j$  is satisfied for all  $(a_j, b_j) \in T_j$ ,  $j = 1, \dots, r$ .

Generally, when  $T_j$  contains an infinite number of elements, this problem involves a correspondingly infinite number of constraints. To convert the problem to one involving a finite number of constraints, we note that

$$a'_j x \leq b_j, \quad \forall (a_j, b_j) \in T_j \quad \text{if and only if} \quad g_j(x) \leq 0,$$

where

$$g_j(x) = \sup_{(a_j, b_j) \in T_j} \{a'_j x - b_j\}. \quad (6.21)$$

Thus, the robust linear programming problem (6.20) is equivalent to

$$\begin{aligned} & \text{minimize} \quad c'x \\ & \text{subject to} \quad g_j(x) \leq 0, \quad j = 1, \dots, r. \end{aligned}$$

For special choices of the set  $T_j$ , the function  $g_j$  can be expressed in closed form, and in the case where  $T_j$  is an ellipsoid, it turns out that the constraint  $g_j(x) \leq 0$  can be expressed in terms of a second order cone. To see this, let

$$T_j = \{(\bar{a}_j + P_j u_j, \bar{b}_j + q'_j u_j) \mid \|u_j\| \leq 1\}, \quad (6.22)$$

where  $P_j$  is a given matrix,  $\bar{a}_j$  and  $q_j$  are given vectors, and  $\bar{b}_j$  is a given scalar. Then, from Eqs. (6.21) and (6.22),

$$\begin{aligned} g_j(x) &= \sup_{\|u_j\| \leq 1} \{(\bar{a}_j + P_j u_j)'x - (\bar{b}_j + q'_j u_j)\} \\ &= \sup_{\|u_j\| \leq 1} (P'_j x - q_j)'u_j + \bar{a}'_j x - \bar{b}_j, \end{aligned}$$

and finally

$$g_j(x) = \|P'_j x - q_j\| + \bar{a}'_j x - \bar{b}_j.$$

Thus,

$$g_j(x) \leq 0 \quad \text{if and only if} \quad (P'_j x - q_j, \bar{b}_j - \bar{a}'_j x) \in C_j,$$

where  $C_j$  is the second order cone; i.e., the “robust” constraint  $g_j(x) \leq 0$  is equivalent to a second order cone constraint. It follows that in the case of ellipsoidal uncertainty, the robust linear programming problem (6.20) is a SOCP of the form (6.18).

**Example 6.1.3: (Quadratically Constrained Quadratic Problems)**

Consider the quadratically constrained quadratic problem

$$\begin{aligned} & \text{minimize} && x'Q_0x + 2q_0'x + p_0 \\ & \text{subject to} && x'Q_jx + 2q_j'x + p_j \leq 0, \quad j = 1, \dots, r, \end{aligned}$$

where  $Q_0, \dots, Q_r$  are symmetric  $n \times n$  positive definite matrices,  $q_0, \dots, q_r$  are vectors in  $\mathbb{R}^n$ , and  $p_0, \dots, p_r$  are scalars. We show that the problem can be converted to the second order cone format. A similar conversion is also possible for the quadratic programming problem where  $Q_0$  is positive definite and  $Q_j = 0$ ,  $j = 1, \dots, r$ .

Indeed, since each  $Q_j$  is symmetric and positive definite, we have

$$\begin{aligned} x'Q_jx + 2q_j'x + p_j &= \left(Q_j^{1/2}x\right)' Q_j^{1/2}x + 2\left(Q_j^{-1/2}q_j\right)' Q_j^{1/2}x + p_j \\ &= \|Q_j^{1/2}x + Q_j^{-1/2}q_j\|^2 + p_j - q_j'Q_j^{-1}q_j, \end{aligned}$$

for  $j = 0, 1, \dots, r$ . Thus, the problem can be written as

$$\begin{aligned} & \text{minimize} && \|Q_0^{1/2}x + Q_0^{-1/2}q_0\|^2 + p_0 - q_0'Q_0^{-1}q_0 \\ & \text{subject to} && \|Q_j^{1/2}x + Q_j^{-1/2}q_j\|^2 + p_j - q_j'Q_j^{-1}q_j \leq 0, \quad j = 1, \dots, r, \end{aligned}$$

or, by neglecting the constant  $p_0 - q_0'Q_0^{-1}q_0$ ,

$$\begin{aligned} & \text{minimize} && \|Q_0^{1/2}x + Q_0^{-1/2}q_0\| \\ & \text{subject to} && \|Q_j^{1/2}x + Q_j^{-1/2}q_j\| \leq (q_j'Q_j^{-1}q_j - p_j)^{1/2}, \quad j = 1, \dots, r. \end{aligned}$$

By introducing an auxiliary variable  $x_{n+1}$ , the problem can be written as

$$\begin{aligned} & \text{minimize} && x_{n+1} \\ & \text{subject to} && \|Q_0^{1/2}x + Q_0^{-1/2}q_0\| \leq x_{n+1} \\ & && \|Q_j^{1/2}x + Q_j^{-1/2}q_j\| \leq (q_j'Q_j^{-1}q_j - p_j)^{1/2}, \quad j = 1, \dots, r. \end{aligned}$$

It can be seen that this problem has the second order cone form (6.18).

We finally note that the problem of this example is special in that it has no duality gap, assuming its optimal value is finite, i.e., there is no need for the interior point conditions of Prop. 6.1.2. This can be traced to the fact that linear transformations preserve the closure of sets defined by quadratic constraints (see e.g., BNO03], Section 1.5.2).

### Semidefinite Programming

Consider the space of symmetric  $n \times n$  matrices, viewed as the space  $\Re^{n^2}$  with the inner product

$$\langle X, Y \rangle = \text{trace}(XY) = \sum_{i=1}^n \sum_{j=1}^n x_{ij}y_{ij}.$$

Let  $D$  be the cone of matrices that are positive semidefinite, called the *positive semidefinite cone*.

The dual cone is

$$\hat{D} = \{Y \mid \text{trace}(XY) \geq 0, \forall X \in D\},$$

and it can be shown that  $\hat{\hat{D}} = D$ , i.e.,  $D$  is self-dual. Indeed, if  $Y \notin D$ , there exists a vector  $v \in \Re^n$  such that

$$0 > v'Yv = \text{trace}(vv'Y).$$

Hence the positive semidefinite matrix  $X = vv'$  satisfies  $0 > \text{trace}(XY)$ , so  $Y \notin \hat{D}$  and it follows that  $D \supset \hat{D}$ . Conversely, let  $Y \in D$ , and let  $X$  be any positive semidefinite matrix. We can express  $X$  as

$$X = \sum_{i=1}^n \lambda_i e_i e_i',$$

where  $\lambda_i$  are the nonnegative eigenvalues of  $X$ , and  $e_i$  are corresponding orthonormal eigenvectors. Then,

$$\text{trace}(XY) = \text{trace}\left(Y \sum_{i=1}^n \lambda_i e_i e_i'\right) = \sum_{i=1}^n \lambda_i e_i' Y e_i \geq 0.$$

It follows that  $Y \in \hat{D}$ , and  $D \subset \hat{D}$ .

Consider the space of symmetric  $n \times n$  matrices, viewed as the space  $\Re^{n^2}$  with the inner product

$$\langle X, Y \rangle = \text{trace}(XY) = \sum_{i=1}^n \sum_{j=1}^n x_{ij}y_{ij}.$$

Let  $D$  be the cone of positive semidefinite matrices, and note that as shown earlier  $D = \hat{D}$ , and that its interior is the set of positive definite matrices.

The semidefinite programming problem (SDP for short) is to minimize a linear function of a symmetric matrix over the intersection of an affine set with the positive semidefinite cone. It has the form

$$\begin{aligned} &\text{minimize} && \langle C, X \rangle \\ &\text{subject to} && \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m, \quad X \in D, \end{aligned} \tag{6.23}$$



where  $C, A_1, \dots, A_m$ , are given  $n \times n$  symmetric matrices, and  $b_1, \dots, b_m$ , are given scalars. It is seen to be a special case of the primal problem in the left-hand side of the duality relation (6.14).

The SDP is a fairly general problem. In particular, it can also be shown that a SOCP can be cast as a SDP (see Exercise 6.3). Thus SDP involves a more general structure than SOCP. This is consistent with the practical observation that the latter problem is generally more amenable to computational solution.

We can view the SDP as a problem with linear cost, linear constraints, and a convex set constraint (as in Section 5.3.3). Then, similar to the case of SOCP, it can be verified that the dual problem (6.13), as given by the right-hand side of the duality relation (6.14), takes the form

$$\begin{aligned} & \text{maximize} && b' \lambda \\ & \text{subject to} && C - (\lambda_1 A_1 + \dots + \lambda_m A_m) \in D, \end{aligned} \quad (6.24)$$

where  $b = (b_1, \dots, b_m)$  and the maximization is over the vector  $\lambda = (\lambda_1, \dots, \lambda_m)$ . By applying the duality result of Prop. 6.1.1, we have the following proposition.

**Proposition 6.1.3: (Semidefinite Duality Theorem)** Consider the primal SDP (6.23), and its dual problem (6.24).

- (a) If the optimal value of the primal problem is finite and there exists a primal-feasible solution, which is positive definite, then there is no duality gap, and the dual problem has an optimal solution.
- (b) If the optimal value of the dual problem is finite and there exist scalars  $\bar{\lambda}_1, \dots, \bar{\lambda}_m$  such that  $C - (\bar{\lambda}_1 A_1 + \dots + \bar{\lambda}_m A_m)$  is positive definite, then there is no duality gap, and the primal problem has an optimal solution.

**Example 6.1.4: (Minimizing the Maximum Eigenvalue)**

Given a symmetric  $n \times n$  matrix  $M(\lambda)$ , which depends on a parameter vector  $\lambda = (\lambda_1, \dots, \lambda_m)$ , we want to choose  $\lambda$  so as to minimize the maximum eigenvalue of  $M(\lambda)$ . We pose this problem as

$$\begin{aligned} & \text{minimize} && z \\ & \text{subject to} && \text{maximum eigenvalue of } M(\lambda) \leq z, \end{aligned}$$

or equivalently

$$\begin{aligned} & \text{minimize} && z \\ & \text{subject to} && zI - M(\lambda) \in D, \end{aligned}$$

where  $I$  is the  $n \times n$  identity matrix, and  $D$  is the semidefinite cone. If  $M(\lambda)$  is an affine function of  $\lambda$ ,

$$M(\lambda) = C + \lambda_1 M_1 + \cdots + \lambda_m M_m,$$

the problem has the form of the dual problem (6.24), with the optimization variables being  $(z, \lambda_1, \dots, \lambda_m)$ .

**Example 6.1.5: (Lower Bounds for Discrete Optimization Problems)**

Semidefinite programming has provided an effective means for deriving lower bounds to the optimal value of several types of discrete optimization problems. As an example, consider the following quadratic problem with quadratic equality constraints

$$\begin{aligned} &\text{minimize} && x'Q_0x + a_0'x + b_0 \\ &\text{subject to} && x'Q_ix + a_i'x + b_i = 0, \quad i = 1, \dots, m, \end{aligned} \tag{6.25}$$

where  $Q_0, \dots, Q_m$  are symmetric  $n \times n$  matrices,  $a_0, \dots, a_m$  are vectors in  $\mathbb{R}^n$ , and  $b_0, \dots, b_m$  are scalars.

This problem can be used to model broad classes of discrete optimization problems. To see this, consider an integer constraint that a variable  $x_i$  must be either 0 or 1. Such a constraint can be expressed by the quadratic equality  $x_i^2 - x_i = 0$ . Furthermore, a linear inequality constraint  $a_j'x \leq b_j$  can be expressed as the quadratic equality constraint  $y_j^2 + a_j'x - b_j = 0$ , where  $y_j$  is an additional variable.

Introducing a multiplier vector  $\lambda = (\lambda_1, \dots, \lambda_m)$ , the dual function is given by

$$q(\lambda) = \inf_{x \in \mathbb{R}^n} \{x'Q(\lambda)x + a(\lambda)'x + b(\lambda)\},$$

where

$$Q(\lambda) = Q_0 + \sum_{i=1}^m \lambda_i Q_i, \quad a(\lambda) = a_0 + \sum_{i=1}^m \lambda_i a_i, \quad b(\lambda) = b_0 + \sum_{i=1}^m \lambda_i b_i.$$

Let  $f^*$  and  $q^*$  be the optimal values of problem (6.25) and its dual, and note that by weak duality, we have  $f^* \geq q^*$ . By introducing an auxiliary scalar variable  $\xi$ , we see that the dual problem is to find a pair  $(\xi, \lambda)$  that solves the problem

$$\begin{aligned} &\text{maximize} && \xi \\ &\text{subject to} && q(\lambda) \geq \xi. \end{aligned}$$

The constraint  $q(\lambda) \geq \xi$  of this problem can be written as

$$\inf_{x \in \mathbb{R}^n} \{x'Q(\lambda)x + a(\lambda)'x + b(\lambda) - \xi\} \geq 0,$$

or equivalently, introducing a scalar variable  $t$ ,

$$\inf_{x \in \mathfrak{R}^n, t \in \mathfrak{R}} \{(tx)'Q(\lambda)(tx) + a(\lambda)'(tx)t + (b(\lambda) - \xi)t^2\} \geq 0,$$

or equivalently,

$$\inf_{x \in \mathfrak{R}^n, t \in \mathfrak{R}} \{x'Q(\lambda)x + a(\lambda)'xt + (b(\lambda) - \xi)t^2\} \geq 0,$$

or equivalently,

$$\begin{pmatrix} Q(\lambda) & \frac{1}{2}a(\lambda) \\ \frac{1}{2}a(\lambda)' & b(\lambda) - \xi \end{pmatrix} \in D, \quad (6.26)$$

where  $D$  is the positive semidefinite cone. Thus the dual problem is equivalent to the SDP of maximizing  $\xi$  over all  $(\xi, \lambda)$  satisfying the constraint (6.26), and its optimal value  $q^*$  is a lower bound to  $f^*$ .

## 6.2 ALGORITHMIC DESCENT - STEEPEST DESCENT

Most of the algorithms for minimizing a convex function  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  over a convex set  $X$  generate a sequence  $\{x_k\} \subset X$  and involve one or both of two principal ideas:

- (a) *Iterative descent*, whereby the generated sequence  $\{x_k\}$  satisfies

$$\phi(x_{k+1}) < \phi(x_k) \quad \text{if and only if } x_k \text{ is not optimal,}$$

where  $\phi$  is a *merit function*, that measures the progress of the algorithm towards optimality, and is minimized only at optimal points, i.e.,

$$\arg \min_{x \in X} \phi(x) = \arg \min_{x \in X} f(x).$$

Examples are  $\phi(x) = f(x)$  and  $\phi(x) = \min_{x^* \in X^*} \|x - x^*\|$ , where  $X^*$  is the set of minima of  $f$  over  $X$ , assumed nonempty and closed.

- (b) *Approximation*, whereby the generated sequence  $\{x_k\}$  is obtained by solving at each  $k$  an approximation to the original optimization problem, i.e.,

$$x_{k+1} \in \arg \min_{x \in X_k} F_k(x),$$

where  $F_k$  is a function that approximates  $f$  and  $X_k$  is a set that approximates  $X$ . These may depend on the prior iterates  $x_0, \dots, x_k$ , as well as other parameters. Key ideas here are that minimization of  $F_k$  over  $X_k$  should be easier than minimization of  $f$  over  $X$ , and that  $x_k$  should be a good starting point for obtaining  $x_{k+1}$  via some (possibly special purpose) method. Of course, the approximation of

$f$  by  $F_k$  and/or  $X$  by  $X_k$  should improve as  $k$  increases, and there should be some convergence guarantees as  $k \rightarrow \infty$ .

The methods to be discussed in this chapter revolve around combinations of these ideas, and are often directed towards solving dual problems of fairly complex primal optimization problems. Of course, an implicit assumption here is that there is special structure that favors the use of duality. We start with a discussion of the descent approach in this section, and we continue with it in Sections 6.3, 6.8, and 6.9. We discuss the approximation approach in Sections 6.4-6.7.

### Steepest Descent

A natural iterative descent approach to minimizing  $f$  over  $X$  is based on cost improvement: starting with a point  $x_0 \in X$ , construct a sequence  $\{x_k\} \subset X$  such that

$$f(x_{k+1}) < f(x_k), \quad k = 0, 1, \dots,$$

unless  $x_k$  is optimal for some  $k$ , in which case the method stops. For example, if  $X = \Re^n$  and  $d_k$  is a *descent direction* at  $x_k$ , in the sense that the directional derivative  $f'(x_k; d_k)$  is negative, we may effect descent by moving from  $x_k$  by a small amount along  $d_k$ . This suggests a descent algorithm of the form

$$x_{k+1} = x_k + \alpha_k d_k,$$

where  $d_k$  is a descent direction, and  $\alpha_k$  is a positive stepsize, which is small enough so that  $f(x_{k+1}) < f(x_k)$ .

For the case where  $f$  is differentiable and  $X = \Re^n$ , there are many popular algorithms based on cost improvement. For example, in the classical gradient method, we use  $d_k = -\nabla f(x_k)$ . Since for a differentiable  $f$  we have

$$f'(x_k; d) = \nabla f(x_k)'d,$$

it follows that

$$\frac{d_k}{\|d_k\|} = \arg \min_{\|d\| \leq 1} f'(x_k; d),$$

[assuming that  $\nabla f(x_k) \neq 0$ ]. Thus the gradient method uses the direction with greatest rate of cost improvement, and for this reason it is also called the method of *steepest descent*.

More generally, for minimization of a real-valued convex function  $f : \Re^n \mapsto \Re$ , let us view the steepest descent direction at  $x$  as the solution of the problem

$$\begin{aligned} &\text{minimize} && f'(x; d) \\ &\text{subject to} && \|d\| \leq 1. \end{aligned} \tag{6.27}$$

We will show that this direction is  $-g^*$ , where  $g^*$  is the vector of minimum norm in  $\partial f(x)$ .

Indeed, we recall from Prop. 5.4.8, that  $f'(x; \cdot)$  is the support function of the nonempty and compact subdifferential  $\partial f(x)$ ,

$$f'(x; d) = \max_{g \in \partial f(x)} d'g, \quad \forall x, d \in \mathfrak{R}^n. \quad (6.28)$$

Next we note that the sets  $\{d \mid \|d\| \leq 1\}$  and  $\partial f(x)$  are compact, and the function  $d'g$  is linear in each variable when the other variable is fixed, so by Prop. 5.5.3, we have

$$\min_{\|d\| \leq 1} \max_{g \in \partial f(x)} d'g = \max_{g \in \partial f(x)} \min_{\|d\| \leq 1} d'g,$$

and a saddle point exists. Furthermore, according to Prop. 3.4.1, for any saddle point  $(d^*, g^*)$ ,  $g^*$  maximizes the function  $\min_{\|d\| \leq 1} d'g = -\|g\|$  over  $\partial f(x)$ , so  $g^*$  is the unique vector of minimum norm in  $\partial f(x)$ . Moreover,  $d^*$  minimizes  $\max_{g \in \partial f(x)} d'g$  or equivalently  $f'(x; d)$  [by Eq. (6.28)] subject to  $\|d\| \leq 1$  (so it is a direction of steepest descent), and minimizes  $d'g^*$  subject to  $\|d\| \leq 1$ , so it has the form

$$d^* = -\frac{g^*}{\|g^*\|}$$

[except if  $0 \in \partial f(x)$ , in which case  $d^* = 0$ ]. In conclusion, for each  $x \in \mathfrak{R}^n$ , the opposite of the vector of minimum norm in  $\partial f(x)$  is the unique direction of steepest descent.

The steepest descent method has the form

$$x_{k+1} = x_k - \alpha_k g_k,$$

where  $g_k$  is the vector of minimum norm in  $\partial f(x_k)$ , and  $\alpha_k$  is a positive stepsize such that  $f(x_{k+1}) < f(x_k)$  (assuming that  $x_k$  is not optimal, which is true if and only if  $g_k \neq 0$ ).

One limitation of the steepest descent method is that it does not easily generalize to extended real-valued functions  $f$  because  $\partial f(x_k)$  may be empty for  $x_k$  at the boundary of  $\text{dom}(f)$ . Another limitation is that it requires knowledge of the set  $\partial f(x)$ , as well as finding the minimum norm vector on this set (a potentially nontrivial optimization problem). A third serious drawback of the method is that it may get stuck far from the optimum, depending on the stepsize rule. Somewhat surprisingly, this can happen even if the stepsize  $\alpha_k$  is chosen to minimize  $f$  along the halfline

$$\{x_k - \alpha g_k \mid \alpha \geq 0\}.$$

An example is given in Exercise 6.8. The difficulty in this example is that at the limit,  $f$  is nondifferentiable and has subgradients that cannot be

approximated by subgradients at the iterates, arbitrarily close to the limit. Thus, the steepest descent direction may undergo a large/discontinuous change as we pass to the convergence limit. By contrast, this would not happen if  $f$  were continuously differentiable at the limit, and in fact the steepest descent method has sound convergence properties when used for minimization of differentiable functions.

The limitations of steepest descent motivate alternative algorithmic approaches that are not based on cost function descent. We focus primarily on such approaches in the remainder of this chapter, as they are far more popular than the descent approach for convex nondifferentiable problems. This is in sharp contrast with differentiable problems, where algorithms based on steepest descent, Newton's method, and their variants are dominant. We will return to steepest descent and related approaches in Section 6.9, where we will discuss some special methods with optimal computational complexity.

### 6.3 SUBGRADIENT METHODS

The simplest form of a subgradient method for minimizing a real-valued convex function  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  over a closed convex set  $X$  is given by

$$x_{k+1} = P_X(x_k - \alpha_k g_k), \tag{6.29}$$

where  $g_k$  is a subgradient of  $f$  at  $x_k$ ,  $\alpha_k$  is a positive stepsize, and  $P_X(\cdot)$  denotes projection on the set  $X$ . Thus a single subgradient is required at each iteration, rather than the entire subdifferential. This is often a major advantage.

The following example shows how to compute a subgradient of functions arising in duality and minimax contexts, without computing the full subdifferential.

**Example 6.3.1: (Subgradient Calculation in Minimax Problems)**

Let

$$f(x) = \sup_{z \in Z} \phi(x, z), \tag{6.30}$$

where  $x \in \mathfrak{R}^n$ ,  $z \in \mathfrak{R}^m$ ,  $\phi : \mathfrak{R}^n \times \mathfrak{R}^m \mapsto (-\infty, \infty]$  is a function, and  $Z$  is a subset of  $\mathfrak{R}^m$ . We assume that  $\phi(\cdot, z)$  is convex and closed for each  $z \in Z$ , so  $f$  is also convex and closed. For a fixed  $x \in \text{dom}(f)$ , let us assume that  $z_x \in Z$  attains the supremum in Eq. (6.30), and that  $g_x$  is some subgradient of the convex function  $\phi(\cdot, z_x)$ , i.e.,  $g_x \in \partial\phi(x, z_x)$ . Then by using the subgradient inequality, we have for all  $y \in \mathfrak{R}^n$ ,

$$f(y) = \sup_{z \in Z} \phi(y, z) \geq \phi(y, z_x) \geq \phi(x, z_x) + g'_x(y - x) = f(x) + g'_x(y - x),$$

i.e.,  $g_x$  is a subgradient of  $f$  at  $x$ , so

$$g_x \in \partial\phi(x, z_x) \quad \Rightarrow \quad g_x \in \partial f(x).$$

We have thus obtained a convenient method for calculating a single subgradient of  $f$  at  $x$  at little extra cost, once a maximizer  $z_x \in Z$  of  $\phi(x, \cdot)$  has been found. On the other hand, calculating the entire subdifferential  $\partial f(x)$  may be much more complicated.

### Example 6.3.2: (Subgradient Calculation in Dual Problems)

Consider the problem

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{subject to } x \in X, \quad g(x) \leq 0, \end{aligned}$$

and its dual

$$\begin{aligned} & \text{maximize } q(\mu) \\ & \text{subject to } \mu \geq 0, \end{aligned}$$

where  $f: \mathfrak{R}^n \mapsto \mathfrak{R}$ ,  $g: \mathfrak{R}^n \mapsto \mathfrak{R}^r$  are given functions,  $X$  is a subset of  $\mathfrak{R}^n$ , and

$$q(\mu) = \inf_{x \in X} L(x, \mu) = \inf_{x \in X} \{f(x) + \mu'g(x)\}$$

is the dual function.

For a given  $\mu \in \mathfrak{R}^r$ , suppose that  $x_\mu$  minimizes the Lagrangian over  $x \in X$ ,

$$x_\mu \in \arg \min_{x \in X} \{f(x) + \mu'g(x)\}.$$

Then we claim that  $-g(x_\mu)$  is a subgradient of the negative of the dual function  $f = -q$  at  $\mu$ , i.e.,

$$q(\nu) \leq q(\mu) + (\nu - \mu)'g(x_\mu), \quad \forall \nu \in \mathfrak{R}^r.$$

This is a special case of the preceding example, and can also be verified directly by writing for all  $\nu \in \mathfrak{R}^r$ ,

$$\begin{aligned} q(\nu) &= \inf_{x \in X} \{f(x) + \nu'g(x)\} \\ &\leq f(x_\mu) + \nu'g(x_\mu) \\ &= f(x_\mu) + \mu'g(x_\mu) + (\nu - \mu)'g(x_\mu) \\ &= q(\mu) + (\nu - \mu)'g(x_\mu). \end{aligned}$$

Note that this calculation is valid for all  $\mu \in \mathfrak{R}^r$  for which there is a minimizing vector  $x_\mu$ , and yields a subgradient of the function

$$- \inf_{x \in X} \{f(x) + \mu'g(x)\},$$

regardless of whether  $\mu \geq 0$ .

An important characteristic of the subgradient method (6.29) is that the new iterate may not improve the cost for any value of the stepsize; i.e., for some  $k$ , we may have

$$f(P_X(x_k - \alpha g_k)) > f(x_k), \quad \forall \alpha > 0,$$

(see Fig. 6.3.1). However, if the stepsize is small enough, the distance of the current iterate to the optimal solution set is reduced (this is illustrated in Fig. 6.3.2). Essential for this is the following nonexpansion property of the projection<sup>†</sup>

$$\|P_X(x) - P_X(y)\| \leq \|x - y\|, \quad \forall x, y \in \mathfrak{R}^n. \quad (6.31)$$

Part (b) of the following proposition provides a formal proof of the distance reduction property and an estimate for the range of appropriate stepsizes.

---

<sup>†</sup> To show the nonexpansion property, note that from Prop. 1.1.9,

$$(z - P_X(x))'(x - P_X(x)) \leq 0, \quad \forall z \in X.$$

Since  $P_X(y) \in X$ , we obtain

$$(P_X(y) - P_X(x))'(x - P_X(x)) \leq 0.$$

Similarly,

$$(P_X(x) - P_X(y))'(y - P_X(y)) \leq 0.$$

By adding these two inequalities, we see that

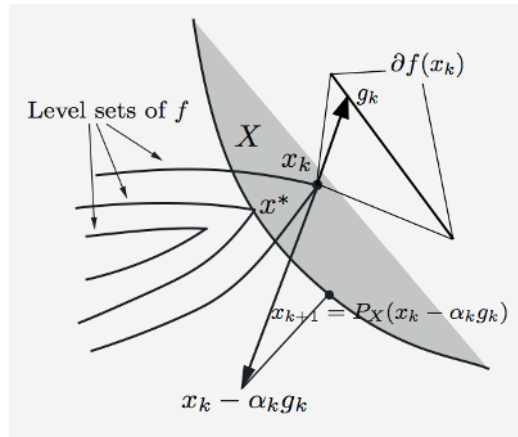
$$(P_X(y) - P_X(x))'(x - P_X(x) - y + P_X(y)) \leq 0.$$

By rearranging and by using the Schwarz inequality, we have

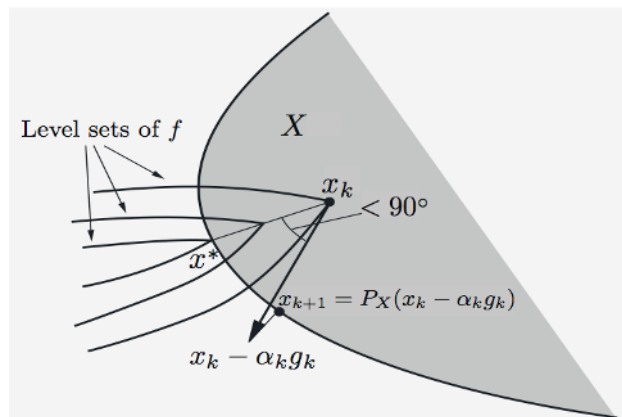
$$\|P_X(y) - P_X(x)\|^2 \leq (P_X(y) - P_X(x))'(y - x) \leq \|P_X(y) - P_X(x)\| \cdot \|y - x\|,$$

from which the nonexpansion property of the projection follows.





**Figure 6.3.1.** Illustration of how the iterate  $P_X(x_k - \alpha_k g_k)$  may not improve the cost function with a particular choice of subgradient  $g_k$ , regardless of the value of the stepsize  $\alpha$ .



**Figure 6.3.2.** Illustration of how, given a nonoptimal  $x_k$ , the distance to any optimal solution  $x^*$  is reduced using a subgradient iteration with a sufficiently small stepsize. The crucial fact, which follows from the definition of a subgradient, is that the angle between the subgradient  $g_k$  and the vector  $x^* - x_k$  is greater than  $90$  degrees. As a result, if  $\alpha_k$  is small enough, the vector  $x_k - \alpha_k g_k$  is closer to  $x^*$  than  $x_k$ . Through the projection on  $X$ ,  $P_X(x_k - \alpha_k g_k)$  gets even closer to  $x^*$ .

**Proposition 6.3.1:** Let  $\{x_k\}$  be the sequence generated by the subgradient method. Then, for all  $y \in X$  and  $k \geq 0$ :

(a) We have

$$\|x_{k+1} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k(f(x_k) - f(y)) + \alpha_k^2\|g_k\|^2.$$

(b) If  $f(y) < f(x_k)$ , we have

$$\|x_{k+1} - y\| < \|x_k - y\|,$$

for all stepsizes  $\alpha_k$  such that

$$0 < \alpha_k < \frac{2(f(x_k) - f(y))}{\|g_k\|^2}.$$

**Proof:** (a) Using the nonexpansion property of the projection [cf. Eq. (6.31)], we obtain for all  $y \in X$  and  $k$ ,

$$\begin{aligned} \|x_{k+1} - y\|^2 &= \|P_X(x_k - \alpha_k g_k) - y\|^2 \\ &\leq \|x_k - \alpha_k g_k - y\|^2 \\ &= \|x_k - y\|^2 - 2\alpha_k g'_k(x_k - y) + \alpha_k^2\|g_k\|^2 \\ &\leq \|x_k - y\|^2 - 2\alpha_k(f(x_k) - f(y)) + \alpha_k^2\|g_k\|^2, \end{aligned}$$

where the last inequality follows from the subgradient inequality.

(b) Follows from part (a). **Q.E.D.**

Part (b) of the preceding proposition suggests the stepsize rule

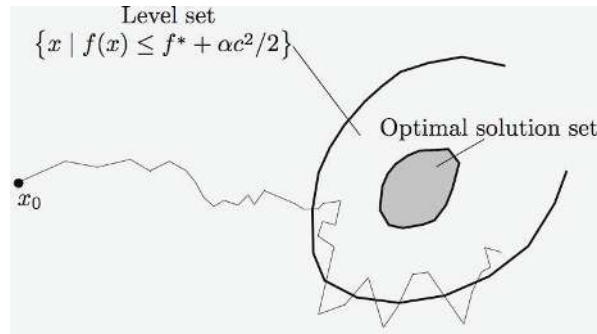
$$\alpha_k = \frac{f(x_k) - f^*}{\|g_k\|^2}, \quad (6.32)$$

where  $f^*$  is the optimal value. This rule selects  $\alpha_k$  to be in the middle of the range

$$\left(0, \frac{2(f(x_k) - f(x^*))}{\|g_k\|^2}\right)$$

where  $x^*$  is an optimal solution [cf. Prop. 6.3.1(b)], and reduces the distance of the current iterate to  $x^*$ .

Unfortunately, however, the stepsize (6.32) requires that we know  $f^*$ , which is rare. In practice, one must use some simpler scheme for selecting



**Figure 6.3.3.** Illustration of a principal convergence property of the subgradient method with a constant stepsize  $\alpha$ , and assuming a bound  $c$  on the subgradient norms  $\|g_k\|$ . When the current iterate is outside the level set

$$\left\{ x \mid f(x) \leq f^* + \frac{\alpha c^2}{2} \right\},$$

the distance to any optimal solution is reduced at the next iteration. As a result the method gets arbitrarily close to (or inside) this level set.

a stepsize. The simplest possibility is to select  $\alpha_k$  to be the same for all  $k$ , i.e.,  $\alpha_k \equiv \alpha$  for some  $\alpha > 0$ . Then, if the subgradients  $g_k$  are bounded ( $\|g_k\| \leq c$  for some constant  $c$  and all  $k$ ), Prop. 6.3.1(a) shows that for all optimal solutions  $x^*$ , we have

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha(f(x_k) - f^*) + \alpha^2 c^2,$$

and implies that the distance to  $x^*$  decreases if

$$0 < \alpha < \frac{2(f(x_k) - f^*)}{c^2}$$

or equivalently, if  $x_k$  is outside the level set

$$\left\{ x \mid f(x) \leq f^* + \frac{\alpha c^2}{2} \right\};$$

(see Fig. 6.3.3). Thus, if  $\alpha$  is taken to be small enough, the convergence properties of the method are satisfactory. Since a small stepsize may result in slow initial progress, it is common to use a variant of this approach whereby we start with moderate stepsize values  $\alpha_k$ , which are progressively reduced up to a small positive value  $\alpha$ , using some heuristic scheme. An example will be discussed at the end of the present section. Other possibilities for stepsize choice include a diminishing stepsize, whereby  $\alpha_k \rightarrow 0$ , and schemes that replace the unknown optimal value  $f^*$  in Eq. (6.32) with an estimate.

### Convergence Analysis

We will now discuss the convergence of the subgradient method

$$x_{k+1} = P_X(x_k - \alpha_k g_k).$$

Throughout our analysis in this section, we denote by  $\{x_k\}$  the corresponding generated sequence, we write

$$f^* = \inf_{x \in X} f(x), \quad X^* = \{x \in X \mid f(x) = f^*\}, \quad \bar{f} = \liminf_{k \rightarrow \infty} f(x_k),$$

and we assume the following:

**Assumption 6.3.1: (Subgradient Boundedness)** For some scalar  $c$ , we have

$$c \geq \sup_{k \geq 0} \{\|g\| \mid g \in \partial f(x_k)\}.$$

We note that Assumption 6.3.1 is satisfied if  $f$  is polyhedral, an important special case in practice (cf. Example 6.1.1). Furthermore, if  $X$  is compact, then Assumption 6.3.1 is satisfied [see Prop. 5.4.2(a)]. Similarly, Assumption 6.3.1 will hold if it can be ascertained somehow that  $\{x_k\}$  is bounded.

We will consider three different types of stepsize rules:

- (a) A constant stepsize.
- (b) A diminishing stepsize.
- (c) A dynamically chosen stepsize based on the value  $f^*$  [cf. Prop. 6.3.1(b)] or a suitable estimate.

We first consider the case of a constant stepsize rule.

**Proposition 6.3.2:** Assume that  $\alpha_k$  is fixed at some positive scalar  $\alpha$ .

- (a) If  $f^* = -\infty$ , then  $\bar{f} = f^*$ .
- (b) If  $f^* > -\infty$ , then

$$\bar{f} \leq f^* + \frac{\alpha c^2}{2}.$$

**Proof:** We prove (a) and (b) simultaneously. If the result does not hold, there must exist an  $\epsilon > 0$  such that

$$\bar{f} > f^* + \frac{\alpha c^2}{2} + 2\epsilon.$$

Let  $\hat{y} \in X$  be such that

$$\bar{f} \geq f(\hat{y}) + \frac{\alpha c^2}{2} + 2\epsilon,$$

and let  $k_0$  be large enough so that for all  $k \geq k_0$  we have

$$f(x_k) \geq \bar{f} - \epsilon.$$

By adding the preceding two relations, we obtain for all  $k \geq k_0$ ,

$$f(x_k) - f(\hat{y}) \geq \frac{\alpha c^2}{2} + \epsilon.$$

Using Prop. 6.3.1(a) for the case where  $y = \hat{y}$  together with the above relation and Assumption 6.3.1, we obtain for all  $k \geq k_0$ ,

$$\|x_{k+1} - \hat{y}\|^2 \leq \|x_k - \hat{y}\|^2 - 2\alpha\epsilon.$$

Thus we have

$$\begin{aligned} \|x_{k+1} - \hat{y}\|^2 &\leq \|x_k - \hat{y}\|^2 - 2\alpha\epsilon \\ &\leq \|x_{k-1} - \hat{y}\|^2 - 4\alpha\epsilon \\ &\dots \\ &\leq \|x_{k_0} - \hat{y}\|^2 - 2(k+1-k_0)\alpha\epsilon, \end{aligned}$$

which cannot hold for  $k$  sufficiently large – a contradiction. **Q.E.D.**

The next proposition gives an estimate of the number of iterations needed to guarantee a level of optimality up to the threshold tolerance  $\alpha c^2/2$  given in the preceding proposition. As can be expected, the number of necessary iterations depends on the distance

$$d(x_0) = \min_{x^* \in X^*} \|x_0 - x^*\|,$$

of the initial point  $x_0$  to the optimal solution set  $X^*$ .

**Proposition 6.3.3:** Assume that  $\alpha_k$  is fixed at some positive scalar  $\alpha$ , and that  $X^*$  is nonempty. Then for any positive scalar  $\epsilon$ , we have

$$\min_{0 \leq k \leq K} f(x_k) \leq f^* + \frac{\alpha c^2 + \epsilon}{2}, \quad (6.33)$$

where

$$K = \left\lceil \frac{d(x_0)^2}{\alpha\epsilon} \right\rceil.$$

**Proof:** Assume, to arrive at a contradiction, that Eq. (6.33) does not hold, so that for all  $k$  with  $0 \leq k \leq K$ , we have

$$f(x_k) > f^* + \frac{\alpha c^2 + \epsilon}{2}.$$

By using this relation in Prop. 6.3.1(a) with  $y \in X^*$  and  $\alpha_k = \alpha$ , we obtain for all  $k$  with  $0 \leq k \leq K$ ,

$$\begin{aligned} \min_{x^* \in X^*} \|x_{k+1} - x^*\|^2 &\leq \min_{x^* \in X^*} \|x_k - x^*\|^2 - 2\alpha(f(x_k) - f^*) + \alpha^2 c^2 \\ &\leq \min_{x^* \in X^*} \|x_k - x^*\|^2 - (\alpha^2 c^2 + \alpha\epsilon) + \alpha^2 c^2 \\ &= \min_{x^* \in X^*} \|x_k - x^*\|^2 - \alpha\epsilon. \end{aligned}$$

Summation of the above inequalities over  $k$  for  $k = 0, \dots, K$ , yields

$$\min_{x^* \in X^*} \|x_{K+1} - x^*\|^2 \leq \min_{x^* \in X^*} \|x_0 - x^*\|^2 - (K+1)\alpha\epsilon,$$

so that

$$\min_{x^* \in X^*} \|x_0 - x^*\|^2 - (K+1)\alpha\epsilon \geq 0,$$

which contradicts the definition of  $K$ . **Q.E.D.**

By letting  $\alpha = \epsilon/c^2$ , we see from the preceding proposition that we can obtain an  $\epsilon$ -optimal solution in  $O(1/\epsilon^2)$  iterations of the subgradient method. Note that the number of iterations is independent of the dimension  $n$  of the problem.

We next consider the case where the stepsize  $\alpha_k$  diminishes to zero, but satisfies  $\sum_{k=0}^{\infty} \alpha_k = \infty$  [for example,  $\alpha_k = \beta/(k + \gamma)$ , where  $\beta$  and  $\gamma$  are some positive scalars]. This condition is needed so that the method can “travel” infinitely far if necessary to attain convergence; otherwise, if

$$\min_{x^* \in X^*} \|x_0 - x^*\| > c \sum_{k=0}^{\infty} \alpha_k,$$

where  $c$  is the constant in Assumption 6.3.1, convergence to  $X^*$  starting from  $x_0$  is impossible.

**Proposition 6.3.4:** If  $\alpha_k$  satisfies

$$\lim_{k \rightarrow \infty} \alpha_k = 0, \quad \sum_{k=0}^{\infty} \alpha_k = \infty,$$

then  $\bar{f} = f^*$ .

**Proof:** Assume, to arrive at a contradiction, that the above relation does not hold, so there exists an  $\epsilon > 0$  such that

$$\bar{f} - 2\epsilon > f^*.$$

Then there exists a point  $\hat{y} \in X$  such that

$$\bar{f} - 2\epsilon > f(\hat{y}).$$

Let  $k_0$  be large enough so that for all  $k \geq k_0$ , we have

$$f(x_k) \geq \bar{f} - \epsilon.$$

By adding the preceding two relations, we obtain for all  $k \geq k_0$ ,

$$f(x_k) - f(\hat{y}) > \epsilon.$$

By setting  $y = \hat{y}$  in Prop. 6.3.1(a), and by using the above relation and Assumption 6.3.1, we have for all  $k \geq k_0$ ,

$$\|x_{k+1} - \hat{y}\|^2 \leq \|x_k - \hat{y}\|^2 - 2\alpha_k \epsilon + \alpha_k^2 c^2 = \|x_k - \hat{y}\|^2 - \alpha_k (2\epsilon - \alpha_k c^2).$$

Since  $\alpha_k \rightarrow 0$ , without loss of generality, we may assume that  $k_0$  is large enough so that

$$2\epsilon - \alpha_k c^2 \geq \epsilon, \quad \forall k \geq k_0.$$

Therefore for all  $k \geq k_0$  we have

$$\|x_{k+1} - \hat{y}\|^2 \leq \|x_k - \hat{y}\|^2 - \alpha_k \epsilon \leq \dots \leq \|x_{k_0} - \hat{y}\|^2 - \epsilon \sum_{j=k_0}^k \alpha_j,$$

which cannot hold for  $k$  sufficiently large. **Q.E.D.**

We now discuss the stepsize rule

$$\alpha_k = \gamma_k \frac{f(x_k) - f^*}{\|g_k\|^2}, \quad 0 < \underline{\gamma} \leq \gamma_k \leq \bar{\gamma} < 2, \quad \forall k \geq 0, \quad (6.34)$$

where  $\underline{\gamma}$  and  $\bar{\gamma}$  are some scalars. We first consider the case where  $f^*$  is known. We later modify the stepsize, so that  $f^*$  can be replaced by a dynamically updated estimate.

**Proposition 6.3.5:** Assume that  $X^*$  is nonempty. Then, if  $\alpha_k$  is determined by the dynamic stepsize rule (6.34),  $\{x_k\}$  converges to some optimal solution.

**Proof:** From Prop. 6.3.1(a) with  $y = x^* \in X^*$ , we have

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha_k(f(x_k) - f^*) + \alpha_k^2 \|g_k\|^2, \quad \forall x^* \in X^*, \quad k \geq 0.$$

By using the definition of  $\alpha_k$  [cf. Eq. (6.34)] and the fact  $\|g_k\| \leq c$  (cf. Assumption 6.3.1), we obtain

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \underline{\gamma}(2 - \bar{\gamma}) \frac{(f(x_k) - f^*)^2}{c^2}, \quad \forall x^* \in X^*, \quad k \geq 0.$$

This implies that  $\{x_k\}$  is bounded. Furthermore,  $f(x_k) \rightarrow f^*$ , since otherwise we would have  $\|x_{k+1} - x^*\| \leq \|x_k - x^*\| - \epsilon$  for some suitably small  $\epsilon > 0$  and infinitely many  $k$ . Hence for any limit point  $\bar{x}$  of  $\{x_k\}$ , we have  $\bar{x} \in X^*$ , and since the sequence  $\{\|x_k - x^*\|\}$  is decreasing, it converges to  $\|\bar{x} - x^*\|$  for every  $x^* \in X^*$ . If there are two distinct limit points  $\tilde{x}$  and  $\bar{x}$  of  $\{x_k\}$ , we must have  $\tilde{x} \in X^*$ ,  $\bar{x} \in X^*$ , and  $\|\tilde{x} - x^*\| = \|\bar{x} - x^*\|$  for all  $x^* \in X^*$ , which is possible only if  $\tilde{x} = \bar{x}$ . **Q.E.D.**

In most practical problems the optimal value  $f^*$  is not known. In this case we may modify the dynamic stepsize (6.34) by replacing  $f^*$  with an estimate. This leads to the stepsize rule

$$\alpha_k = \gamma_k \frac{f(x_k) - f_k}{\|g_k\|^2}, \quad 0 < \underline{\gamma} \leq \gamma_k \leq \bar{\gamma} < 2, \quad \forall k \geq 0, \quad (6.35)$$

where  $f_k$  is an estimate of  $f^*$ . We consider a procedure for updating  $f_k$ , whereby  $f_k$  is given by

$$f_k = \min_{0 \leq j \leq k} f(x_j) - \delta_k, \quad (6.36)$$

and  $\delta_k$  is updated according to

$$\delta_{k+1} = \begin{cases} \rho \delta_k & \text{if } f(x_{k+1}) \leq f_k, \\ \max\{\beta \delta_k, \delta\} & \text{if } f(x_{k+1}) > f_k, \end{cases} \quad (6.37)$$

where  $\delta$ ,  $\beta$ , and  $\rho$  are fixed positive constants with  $\beta < 1$  and  $\rho \geq 1$ .

Thus in this procedure, we essentially “aspire” to reach a target level  $f_k$  that is smaller by  $\delta_k$  over the best value achieved thus far [cf. Eq. (6.36)]. Whenever the target level is achieved, we increase  $\delta_k$  (if  $\rho > 1$ ) or we keep it at the same value (if  $\rho = 1$ ). If the target level is not attained at a given iteration,  $\delta_k$  is reduced up to a threshold  $\delta$ . This threshold guarantees that the stepsize  $\alpha_k$  of Eq. (6.35) is bounded away from zero, since from Eq. (6.36), we have  $f(x_k) - f_k \geq \delta$  and hence

$$\alpha_k \geq \underline{\gamma} \frac{\delta}{c^2}.$$



As a result, the method behaves similar to the one with a constant stepsize (cf. Prop. 6.3.2), as indicated by the following proposition.

**Proposition 6.3.6:** Assume that  $\alpha_k$  is determined by the dynamic stepsize rule (6.35) with the adjustment procedure (6.36)–(6.37). If  $f^* = -\infty$ , then

$$\inf_{k \geq 0} f(x_k) = f^*,$$

while if  $f^* > -\infty$ , then

$$\inf_{k \geq 0} f(x_k) \leq f^* + \delta.$$

**Proof:** Assume, to arrive at a contradiction, that

$$\inf_{k \geq 0} f(x_k) > f^* + \delta. \quad (6.38)$$

Each time the target level is attained [i.e.,  $f(x_k) \leq f_{k-1}$ ], the current best function value  $\min_{0 \leq j \leq k} f(x_j)$  decreases by at least  $\delta$  [cf. Eqs. (6.36) and (6.37)], so in view of Eq. (6.38), the target value can be attained only a finite number of times. From Eq. (6.37) it follows that after finitely many iterations,  $\delta_k$  is decreased to the threshold value and remains at that value for all subsequent iterations, i.e., there is an index  $\bar{k}$  such that

$$\delta_k = \delta, \quad \forall k \geq \bar{k}. \quad (6.39)$$

In view of Eq. (6.38), there exists  $\bar{y} \in X$  such that  $\inf_{k \geq 0} f(x_k) - \delta \geq f(\bar{y})$ . From Eqs. (6.36) and (6.39), we have

$$f_k = \min_{0 \leq j \leq k} f(x_j) - \delta \geq \inf_{k \geq 0} f(x_k) - \delta \geq f(\bar{y}), \quad \forall k \geq \bar{k},$$

so that

$$\alpha_k(f(x_k) - f(\bar{y})) \geq \alpha_k(f(x_k) - f_k) = \gamma_k \left( \frac{f(x_k) - f_k}{\|g_k\|} \right)^2, \quad \forall k \geq \bar{k}.$$

By using Prop. 6.3.1(a) with  $y = \bar{y}$ , we have

$$\|x_{k+1} - \bar{y}\|^2 \leq \|x_k - \bar{y}\|^2 - 2\alpha_k(f(x_k) - f(\bar{y})) + \alpha_k^2 \|g_k\|^2, \quad \forall k \geq 0.$$

By combining the preceding two relations and the definition of  $\alpha_k$  [cf. Eq. (6.35)], we obtain

$$\begin{aligned} \|x_{k+1} - \bar{y}\|^2 &\leq \|x_k - \bar{y}\|^2 - 2\gamma_k \left( \frac{f(x_k) - f_k}{\|g_k\|} \right)^2 + \gamma_k^2 \left( \frac{f(x_k) - f_k}{\|g_k\|} \right)^2 \\ &= \|x_k - \bar{y}\|^2 - \gamma_k(2 - \gamma_k) \left( \frac{f(x_k) - f_k}{\|g_k\|} \right)^2 \\ &\leq \|x_k - \bar{y}\|^2 - \underline{\gamma}(2 - \bar{\gamma}) \frac{\delta^2}{\|g_k\|^2}, \quad \forall k \geq \bar{k}, \end{aligned}$$

where the last inequality follows from the facts  $\gamma_k \in [\underline{\gamma}, \bar{\gamma}]$  and  $f(x_k) - f_k \geq \delta$  for all  $k$ . By summing the above inequalities over  $k$  and using Assumption 6.3.1, we have

$$\|x_k - \bar{y}\|^2 \leq \|x_{\bar{k}} - \bar{y}\|^2 - (k - \bar{k})\underline{\gamma}(2 - \bar{\gamma})\frac{\delta^2}{c^2}, \quad \forall k \geq \bar{k},$$

which cannot hold for sufficiently large  $k$  – a contradiction. **Q.E.D.**

## 6.4 POLYHEDRAL APPROXIMATION METHODS

In this section, we will discuss methods, which (like the subgradient method) calculate a single subgradient at each iteration, but use all the subgradients previously calculated to construct piecewise linear approximations of the cost function and/or the constraint set. In Sections 6.4.1 and 6.4.2, we focus on the problem of minimizing a convex function  $f : \Re^n \mapsto \Re$  over a closed convex set  $X$ , and we assume that at each  $x \in X$ , a subgradient of  $f$  can be computed. In Sections 6.4.3-6.4.5, we discuss various generalizations.

### 6.4.1 Outer Linearization - Cutting Plane Methods

Cutting plane methods are rooted in the representation of a closed convex set as the intersection of its supporting halfspaces. The idea is to approximate either the constraint set or the epigraph of the cost function by the intersection of a limited number of halfspaces, and to gradually refine the approximation by generating additional halfspaces through the use of subgradients.

The typical iteration of the simplest cutting plane method is to solve the problem

$$\begin{aligned} & \text{minimize } F_k(x) \\ & \text{subject to } x \in X, \end{aligned}$$

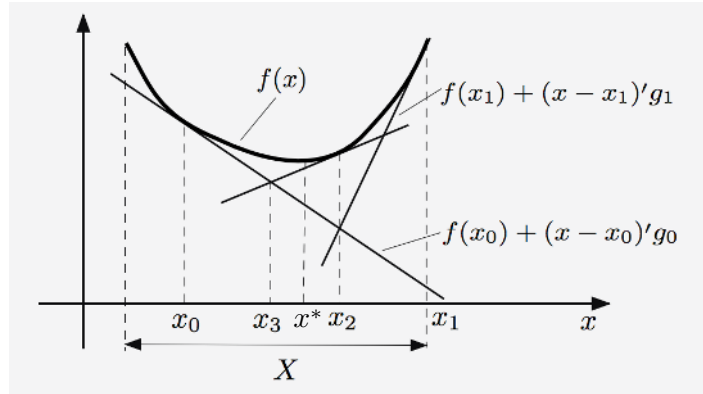
where the cost function  $f$  is replaced by a polyhedral approximation  $F_k$ , constructed using the points  $x_0, \dots, x_k$  generated so far and associated subgradients  $g_0, \dots, g_k$ , with  $g_i \in \partial f(x_i)$  for all  $i$ . In particular, for  $k = 0, 1, \dots$ ,

$$F_k(x) = \max\{f(x_0) + (x - x_0)'g_0, \dots, f(x_k) + (x - x_k)'g_k\} \quad (6.40)$$

and  $x_{k+1}$  minimizes  $F_k(x)$  over  $x \in X$ ,

$$x_{k+1} \in \arg \min_{x \in X} F_k(x); \quad (6.41)$$

see Fig. 6.4.1. We assume that the minimum of  $F_k(x)$  above is attained for all  $k$ . For those  $k$  for which this is not guaranteed, artificial bounds may



**Figure 6.4.1.** Illustration of the cutting plane method. With each new iterate  $x_k$ , a new hyperplane  $f(x_k) + (x - x_k)'g_k$  is added to the polyhedral approximation of the cost function.

be placed on the components of  $x$ , so that the minimization will be carried out over a compact set and consequently the minimum will be attained by Weierstrass' Theorem.

The following proposition establishes the associated convergence properties.

**Proposition 6.4.1:** Every limit point of a sequence  $\{x_k\}$  generated by the cutting plane method is an optimal solution.

**Proof:** Since for all  $i$ ,  $g_i$  is a subgradient of  $f$  at  $x_i$ , we have

$$f(x_i) + (x - x_i)'g_i \leq f(x), \quad \forall x \in X,$$

so from the definitions (6.40) and (6.41) of  $F_k$  and  $x_k$ , it follows that

$$f(x_i) + (x_k - x_i)'g_i \leq F_{k-1}(x_k) \leq F_{k-1}(x) \leq f(x), \quad \forall x \in X, \quad i < k. \quad (6.42)$$

Suppose that a subsequence  $\{x_k\}_K$  converges to  $\bar{x}$ . Then, since  $X$  is closed, we have  $\bar{x} \in X$ , and by using Eq. (6.42), we obtain for all  $k$  and all  $i < k$ ,

$$f(x_i) + (x_k - x_i)'g_i \leq F_{k-1}(x_k) \leq F_{k-1}(\bar{x}) \leq f(\bar{x}).$$

By taking the upper limit above as  $i \rightarrow \infty$ ,  $k \rightarrow \infty$ ,  $i < k$ ,  $i \in K$ ,  $k \in K$ , we obtain

$$\limsup_{\substack{i \rightarrow \infty, k \rightarrow \infty, i < k \\ i \in K, k \in K}} \{f(x_i) + (x_k - x_i)'g_i\} \leq \limsup_{k \rightarrow \infty, k \in K} F_{k-1}(x_k) \leq f(\bar{x}).$$

Since the subsequence  $\{x_k\}_K$  is bounded and the union of the subdifferentials of a real-valued convex function over a bounded set is bounded (cf. Prop. 5.4.2), it follows that the subgradient subsequence  $\{g_i\}_K$  is bounded. Therefore we have

$$\lim_{\substack{i \rightarrow \infty, k \rightarrow \infty, i < k \\ i \in K, k \in K}} (x_k - x_i)' g_i = 0, \quad (6.43)$$

while by the continuity of  $f$ , we have

$$f(\bar{x}) = \lim_{i \rightarrow \infty, i \in K} f(x_i). \quad (6.44)$$

Combining the three preceding relations, we obtain

$$\limsup_{k \rightarrow \infty, k \in K} F_{k-1}(x_k) = f(\bar{x}).$$

This equation together with Eq. (6.42) yields

$$f(\bar{x}) \leq f(x), \quad \forall x \in X,$$

showing that  $\bar{x}$  is an optimal solution. **Q.E.D.**

Note that the preceding proof goes through even when  $f$  is real-valued and lower-semicontinuous over  $X$  (rather than over  $\mathfrak{R}^n$ ), provided we assume that  $\{g_k\}$  is a bounded sequence [Eq. (6.43) then still holds, while Eq. (6.44) holds as an inequality, but this does not affect the subsequent argument]. Note also that the inequalities

$$F_{k-1}(x_k) \leq f^* \leq \min_{i \leq k} f(x_i), \quad k = 0, 1, \dots,$$

provide bounds to the optimal value  $f^*$  of the problem. In practice, the iterations are stopped when the upper and lower bound difference  $\min_{i \leq k} f(x_i) - F_{k-1}(x_k)$  comes within some small tolerance.

An important special case arises when  $f$  is polyhedral of the form

$$f(x) = \max_{i \in I} \{a_i'x + b_i\}, \quad (6.45)$$

where  $I$  is a finite index set, and  $a_i$  and  $b_i$  are given vectors and scalars, respectively. Then, any vector  $a_{i_k}$  that maximizes  $a_i'x_k + b_i$  over  $\{a_i \mid i \in I\}$  is a subgradient of  $f$  at  $x_k$  (cf. Example 5.4.4). We assume that the cutting plane method selects such a vector at iteration  $k$ , call it  $a_{i_k}$ . We also assume that the method terminates when

$$F_{k-1}(x_k) = f(x_k).$$

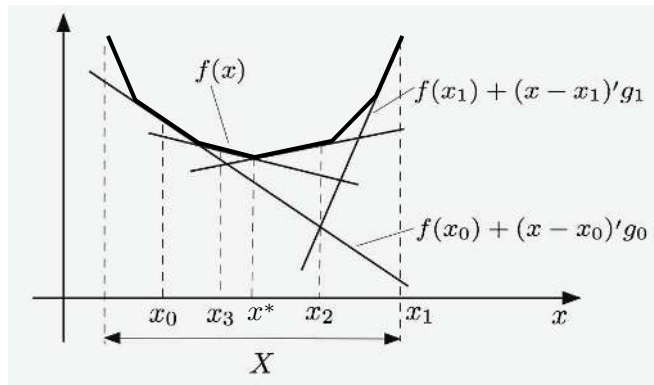
Then, since  $F_{k-1}(x) \leq f(x)$  for all  $x \in X$  and  $x_k$  minimizes  $F_{k-1}$  over  $X$ , we see that, upon termination,  $x_k$  minimizes  $f$  over  $X$  and is therefore optimal. The following proposition shows that the method converges finitely; see also Fig. 6.4.2.

**Proposition 6.4.2:** Assume that the cost function  $f$  is polyhedral of the form (6.45). Then the cutting plane method, with the subgradient selection and termination rules just described, obtains an optimal solution in a finite number of iterations.

**Proof:** If  $(a_{i_k}, b_{i_k})$  is equal to some pair  $(a_{i_j}, b_{i_j})$  generated at some earlier iteration  $j < k$ , then

$$f(x_k) = a'_{i_k} x_k + b_{i_k} = a'_{i_j} x_k + b_{i_j} \leq F_{k-1}(x_k) \leq f(x_k),$$

where the first inequality follows since  $a'_{i_j} x_k + b_{i_j}$  corresponds to one of the hyperplanes defining  $F_{k-1}$ , and the last inequality follows from the fact  $F_{k-1}(x) \leq f(x)$  for all  $x \in X$ . Hence equality holds throughout in the preceding relation, and it follows that the method terminates if the pair  $(a_{i_k}, b_{i_k})$  has been generated at some earlier iteration. Since the number of pairs  $(a_i, b_i)$ ,  $i \in I$ , is finite, the method must terminate finitely. **Q.E.D.**



**Figure 6.4.2.** Illustration of the finite convergence property of the cutting plane method in the case where  $f$  is polyhedral. What happens here is that if  $x_k$  is not optimal, a new cutting plane will be added at the corresponding iteration, and there can be only a finite number of cutting planes.

Despite the finite convergence property shown in Prop. 6.4.2, the cutting plane method has several drawbacks:

- (a) It can take large steps away from the optimum, resulting in large cost increases, even when it is close to (or even at) the optimum. For example, in Fig. 6.4.2,  $f(x_1)$  is much larger than  $f(x_0)$ . This phenomenon is referred to as *instability*, and has another undesirable effect, namely that  $x_k$  may not be a good starting point for the algorithm that minimizes  $F_k(x)$ .
- (b) The number of subgradients used in the cutting plane approximation  $F_k$  increases without bound as  $k \rightarrow \infty$  leading to a potentially large and difficult linear program to find  $x_k$ . To remedy this, one may occasionally discard some of the cutting planes. To guarantee convergence, it is essential to do so only at times when improvement in the cost is recorded, e.g.,  $f(x_k) \leq \min_{i < k} f(x_i) - \delta$  for some small positive  $\delta$ . Still one has to be judicious about discarding cutting planes, as some of them may reappear later.
- (c) The convergence is often slow. Indeed, for challenging problems, even when  $f$  is polyhedral, one should base termination on the upper and lower bounds

$$F_k(x_{k+1}) \leq \min_{x \in X} f(x) \leq \min_{0 \leq i \leq k+1} f(x_i),$$

rather than wait for finite termination to occur.

To overcome some of the limitations of the cutting plane method, a number of variants have been proposed, some of which are discussed in the present section. In Section 6.5 we will discuss proximal methods, which are aimed at limiting the effects of instability.

### Partial Cutting Plane Methods

In some cases the cost function has the form

$$f(x) + c(x),$$

where  $f : X \mapsto \Re$  and  $c : X \mapsto \Re$  are convex functions, but one of them, say  $c$ , is convenient for optimization, e.g., is quadratic. It may then be preferable to use a piecewise linear approximation of  $f$  only, while leaving  $c$  unchanged. This leads to a partial cutting plane algorithm, involving solution of the problems

$$\begin{aligned} &\text{minimize } F_k(x) + c(x) \\ &\text{subject to } x \in X, \end{aligned}$$

where as before

$$F_k(x) = \max\{f(x_0) + (x - x_0)'g_0, \dots, f(x_k) + (x - x_k)'g_k\} \quad (6.46)$$

with  $g_i \in \partial f(x_i)$  for all  $i$ , and  $x_{k+1}$  minimizes  $F_k(x)$  over  $x \in X$ ,

$$x_{k+1} \in \arg \min_{x \in X} \{F_k(x) + c(x)\}.$$

The convergence properties of this algorithm are similar to the ones shown earlier. In particular, if  $f$  is polyhedral, the method terminates finitely, cf. Prop. 6.4.2. The idea of partial piecewise approximation arises in a few contexts to be discussed in the sequel.

### Linearly Constrained Versions

Consider the case where the constraint set  $X$  is polyhedral of the form

$$X = \{x \mid c_i'x + d_i \leq 0, i \in I\},$$

where  $I$  is a finite set, and  $c_i$  and  $d_i$  are given vectors and scalars, respectively. Let

$$p(x) = \max_{i \in I} \{c_i'x + d_i\},$$

so the problem is to maximize  $f(x)$  subject to  $p(x) \leq 0$ . It is then possible to consider a variation of the cutting plane method, where both functions  $f$  and  $p$  are replaced by polyhedral approximations. The method is

$$x_{k+1} \in \arg \max_{P_k(x) \leq 0} F_k(x).$$

As earlier,

$$F_k(x) = \min \{f(x_0) + (x - x_0)'g_0, \dots, f(x_k) + (x - x_k)'g_k\},$$

with  $g_i$  being a subgradient of  $f$  at  $x_i$ . The polyhedral approximation  $P_k$  is given by

$$P_k(x) = \max_{i \in I_k} \{c_i'x + d_i\},$$

where  $I_k$  is a subset of  $I$  generated as follows:  $I_0$  is an arbitrary subset of  $I$ , and  $I_k$  is obtained from  $I_{k-1}$  by setting  $I_k = I_{k-1}$  if  $p(x_k) \leq 0$ , and by adding to  $I_{k-1}$  one or more of the indices  $i \notin I_{k-1}$  such that  $c_i'x_k + d_i > 0$  otherwise.

Note that this method applies even when  $f$  is a linear function. In this case there is no cost function approximation, i.e.,  $F_k = f$ , just outer approximation of the constraint set, i.e.,  $X \subset \{x \mid P_k(x) \leq 0\}$ .

The convergence properties of this method are very similar to the ones of the earlier method. In fact propositions analogous to Props. 6.4.1 and 6.4.2 can be formulated and proved. There are also versions of this method where  $X$  is a general closed convex set, which is iteratively approximated by a polyhedral set.

### Central Cutting Plane Methods

Let us discuss a method that is based on a somewhat different approximation idea. Like the preceding methods, it maintains a polyhedral approximation

$$F_k(x) = \max\{f(x_0) + (x - x_0)'g_0, \dots, f(x_k) + (x - x_k)'g_k\}$$

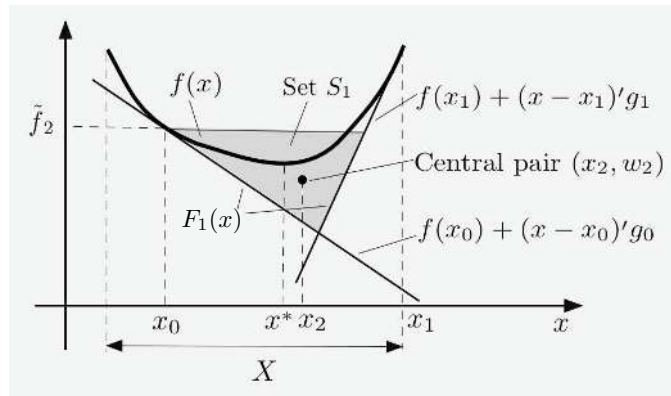
to  $f$ , but it generates the next vector  $x_{k+1}$  by using a different mechanism. In particular, instead of minimizing  $F_k$  as in Eq. (6.41), the method obtains  $x_{k+1}$  by finding a “central pair”  $(x_{k+1}, w_{k+1})$  within the subset

$$S_k = \{(x, w) \mid x \in X, F_k(x) \leq w \leq \tilde{f}_k\},$$

where  $\tilde{f}_k$  is the best upper bound to the optimal value that has been found so far,

$$\tilde{f}_k = \min_{i \leq k} f(x_i)$$

(see Fig. 6.4.3).



**Figure 6.4.3.** Illustration of the set

$$S_k = \{(x, w) \mid x \in X, F_k(x) \leq w \leq \tilde{f}_k\}$$

in the central cutting plane method.

There is a variety of methods for finding the central pair  $(x_{k+1}, w_{k+1})$ . Roughly, the idea is that the central pair should be “somewhere in the middle” of  $S_k$ . For example, consider the case where  $S_k$  is polyhedral with nonempty interior. Then  $(x_{k+1}, w_{k+1})$  could be the *analytic center* of  $S_k$ , where for any polyhedron

$$P = \{y \mid a'_p y \leq c_p, p = 1, \dots, m\}$$



with nonempty interior, its analytic center is defined as the unique maximizer of  $\sum_{p=1}^m \ln(c_p - a'_p y)$  over  $y \in P$ . Another possibility is the *ball center* of  $S$ , i.e., the center of the largest inscribed sphere in  $S_k$ ; for the generic polyhedron  $P$  with nonempty interior, the ball center can be obtained by solving the following problem with optimization variables  $(y, \sigma)$ :

$$\begin{aligned} & \text{maximize } \sigma \\ & \text{subject to } a'_p(y + d) \leq c_p, \quad \forall \|d\| \leq \sigma, \quad p = 1, \dots, m. \end{aligned}$$

It can be seen that this problem is equivalent to the linear program

$$\begin{aligned} & \text{maximize } \sigma \\ & \text{subject to } a'_p y + \|a_p\| \sigma \leq c_p, \quad p = 1, \dots, m. \end{aligned}$$

Central cutting plane methods have satisfactory convergence properties, even though they do not terminate finitely in the case of a polyhedral cost function  $f$ . They are closely related to the interior point methods to be discussed in Section 6.7, and they have benefited from advances in the practical implementation of these methods.

#### 6.4.2 Inner Linearization - Simplicial Decomposition

We now consider an *inner approximation* approach, whereby we approximate  $X$  with the convex hull of an ever expanding finite set  $X_k \subset X$  that consists of extreme points of  $X$  plus an arbitrary starting point  $x_0 \in X$ . The addition of new extreme points to  $X_k$  is done in a way that guarantees a cost improvement each time we minimize  $f$  over  $\text{conv}(X_k)$  (unless we are already at the optimum).

In this section, we assume a *differentiable* convex cost function  $f : \mathbb{R}^n \mapsto \mathbb{R}$  and a *bounded polyhedral constraint set*  $X$ . The method is then appealing under two conditions:

- (1) Minimizing a linear function over  $X$  is much simpler than minimizing  $f$  over  $X$ . (The method makes sense only if  $f$  is nonlinear.)
- (2) Minimizing  $f$  over the convex hull of a relative small number of extreme points is much simpler than minimizing  $f$  over  $X$ . (The method makes sense only if  $X$  has a large number of extreme points.)

Several classes of important large-scale problems, arising for example in communication and transportation networks, have structure that satisfies these conditions (see the end-of-chapter references).

At the typical iteration we have the current iterate  $x_k$ , and the finite set  $X_k$  that consists of the starting point  $x_0$  together with a finite collection of extreme points of  $X$  (initially  $X_0 = \{x_0\}$ ). We first generate  $\tilde{x}_{k+1}$  as an extreme point of  $X$  that solves the linear program

$$\begin{aligned} & \text{minimize } \nabla f(x_k)'(x - x_k) \\ & \text{subject to } x \in X. \end{aligned} \tag{6.47}$$

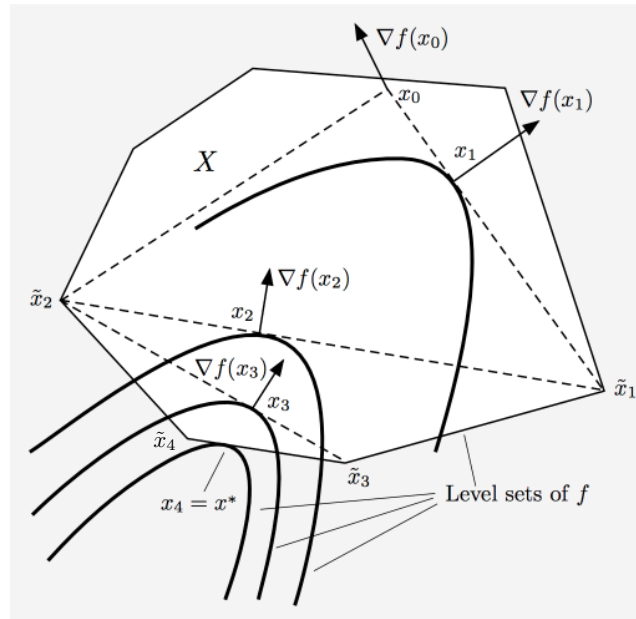
We then add  $\tilde{x}_{k+1}$  to  $X_k$ ,

$$X_{k+1} = \{\tilde{x}_{k+1}\} \cup X_k,$$

and we generate  $x_{k+1}$  as an optimal solution of the problem

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } x \in \text{conv}(X_{k+1}). \end{aligned} \tag{6.48}$$

The process is illustrated in Fig. 6.4.4.



**Figure 6.4.4.** Successive iterates of the simplicial decomposition method. For example, the figure shows how given the initial point  $x_0$ , and the calculated extreme points  $\tilde{x}_1, \tilde{x}_2$ , we determine the next iterate  $x_2$  as a minimizing point of  $f$  over the convex hull of  $\{x_0, \tilde{x}_1, \tilde{x}_2\}$ . At each iteration, a new extreme point of  $X$  is added, and after four iterations, the optimal solution is obtained.

For a convergence proof, note that there are two possibilities for the extreme point  $\tilde{x}_{k+1}$  that solves problem (6.47):

- (a) We have

$$0 \leq \nabla f(x_k)'(\tilde{x}_{k+1} - x_k) = \min_{x \in X} \nabla f(x_k)'(x - x_k),$$

in which case  $x_k$  minimizes  $f$  over  $X$ , since it satisfies the necessary and sufficient optimality condition of Prop. 1.1.8.

(b) We have

$$0 > \nabla f(x_k)'(\tilde{x}_{k+1} - x_k), \quad (6.49)$$

in which case  $\tilde{x}_{k+1} \notin \text{conv}(X_k)$ , since  $x_k$  minimizes  $f$  over  $x \in \text{conv}(X_k)$ , so that  $\nabla f(x_k)'(x - x_k) \geq 0$  for all  $x \in \text{conv}(X_k)$ .

Since case (b) cannot occur an infinite number of times ( $\tilde{x}_{k+1} \notin X_k$  and  $X$  has finitely many extreme points), case (a) must eventually occur, so the method will find a minimizer of  $f$  over  $X$  in a finite number of iterations.

Note that the essence of the preceding convergence proof is that  $\tilde{x}_{k+1}$  does not belong to  $X_k$ , unless the optimal solution has been reached. Thus it is not necessary that  $\tilde{x}_{k+1}$  solves exactly the linearized problem (6.47). Instead it is sufficient that  $\tilde{x}_{k+1}$  is an extreme point and that the condition (6.49) is satisfied. In fact an even more general procedure will work: it is not necessary that  $\tilde{x}_{k+1}$  be an extreme point of  $X$ . Instead it is sufficient that  $\tilde{x}_{k+1}$  be selected from a finite subset  $\tilde{X} \subset X$  such that  $\text{conv}(\tilde{X}) = X$ , and that the condition (6.49) is satisfied. These ideas may be used in variants of the simplicial decomposition method whereby problem (6.47) is solved inexactly.

There are a few other variants of the method. For example to address the case where  $X$  is an unbounded polyhedral set, one may augment  $X$  with additional constraints to make it bounded. There are extensions that allow for a nonpolyhedral constraint set, which is approximated by the convex hull of some of its extreme points in the course of the algorithm; see the literature cited at the end of the chapter. Finally, one may use variants, known as *restricted simplicial decomposition* methods, which allow discarding some of the extreme points generated so far. In particular, given the solution  $x_{k+1}$  of problem (6.48), we may discard from  $X_{k+1}$  all points  $\tilde{x}$  such that

$$\nabla f(x_{k+1})'(\tilde{x} - x_{k+1}) > 0,$$

while possibly adding to the constraint set the additional constraint

$$\nabla f(x_{k+1})'(x - x_{k+1}) \leq 0. \quad (6.50)$$

The idea is that the costs of the subsequent points  $x_{k+j}$ ,  $j > 1$ , generated by the method will all be no greater than the cost of  $x_{k+1}$ , so they will satisfy the constraint (6.50). In fact a stronger result can be shown: any number of extreme points may be discarded, as long as  $\text{conv}(X_{k+1})$  contains  $x_{k+1}$  and  $\tilde{x}_{k+1}$  [the proof is based on the theory of feasible direction methods (see e.g., [Ber99]) and the fact that  $\tilde{x}_{k+1} - x_{k+1}$  is a descent direction for  $f$ , so a point with improved cost can be found along the line segment connecting  $x_{k+1}$  and  $\tilde{x}_{k+1}$ ].

The simplicial decomposition method has been applied to several types of problems that have a suitable structure (large-scale multicommodity flow problems arising in communication and transportation network applications is an example; see the end-of-chapter references). Experience has

generally been favorable and suggests that the method requires a lot fewer iterations than the cutting plane method that uses an outer approximation of the constraint set. As an indication of this, we note that if  $f$  is linear, the simplicial decomposition method terminates in a single iteration, whereas the cutting plane method may require a very large number of iterations to attain the required solution accuracy.

### 6.4.3 Duality of Outer and Inner Linearization

We will now aim to explore the relation between outer and inner linearization, as a first step towards a richer class of approximation methods. In particular, we will show that given a closed proper convex function  $f : \mathbb{R}^n \mapsto (-\infty, \infty]$ , an outer linearization of  $f$  corresponds to an inner linearization of the conjugate  $f^*$  and reversely.

Consider an outer linearization of the epigraph of  $f$  defined by vectors  $y_0, \dots, y_k$  and corresponding hyperplanes that support the epigraph of  $f$  at points  $x_0, \dots, x_k$ :

$$F(x) = \max_{i=0, \dots, k} \{f(x_i) + (x - x_i)'y_i\}; \quad (6.51)$$

cf. Fig. 6.4.5. We will show that *the conjugate  $F^*$  of the outer linearization  $F$  can be described as an inner linearization of the conjugate  $f^*$  of  $f$ .*

Indeed, we have

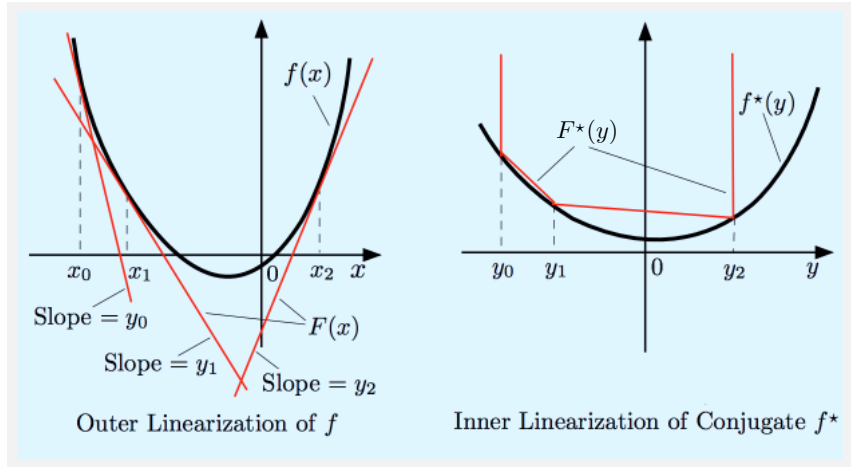
$$\begin{aligned} F^*(y) &= \sup_{x \in \mathbb{R}^n} \{y'x - F(x)\} \\ &= \sup_{x \in \mathbb{R}^n} \left\{ y'x - \max_{i=0, \dots, k} \{f(x_i) + (x - x_i)'y_i\} \right\} \\ &= \sup_{\substack{x \in \mathbb{R}^n, \xi \in \mathbb{R} \\ f(x_i) + (x - x_i)'y_i \leq \xi, \quad i=0, \dots, k}} \{y'x - \xi\}. \end{aligned}$$

By linear programming duality (cf. Prop. 5.2.1), the optimal value of the linear program in  $(x, \xi)$  of the preceding equation can be replaced by the dual optimal value, and we have with a straightforward calculation

$$F^*(y) = \inf_{\substack{\sum_{i=0}^k \alpha_i y_i = y, \quad \sum_{i=0}^k \alpha_i = 1 \\ \alpha_i \geq 0, \quad i=0, \dots, k}} \sum_{i=0}^k \alpha_i (f(x_i) - x_i' y_i),$$

where  $\alpha_i$  is the dual variable of the constraint  $f(x_i) + (x - x_i)'y_i \leq \xi$ . Since the hyperplanes defining  $F$  are supporting  $\text{epi}(f)$ , we have

$$x_i' y_i - f(x_i) = f^*(y_i), \quad i = 0, \dots, k,$$



**Figure 6.4.5.** Illustration of the conjugate  $F^*$  of an outer linearization  $F$  of a convex function  $f$  (here  $k = 2$ ). It is a piecewise linear, inner linearization of the conjugate  $f^*$  of  $f$ . Its break points are the “slopes”  $y_0, \dots, y_k$  of the supporting planes.

so we obtain

$$F^*(y) = \begin{cases} \inf_{\substack{\sum_{i=0}^k \alpha_i y_i = y, \\ \alpha_i \geq 0, i=0, \dots, k}} \sum_{i=0}^k \alpha_i f^*(y_i) & \text{if } y \in \text{conv}\{y_0, \dots, y_k\}, \\ \infty & \text{otherwise.} \end{cases} \quad (6.52)$$

Thus,  $F^*$  is a piecewise linear (inner) linearization of  $f^*$  with domain

$$\text{dom}(F^*) = \text{conv}\{y_0, \dots, y_k\},$$

and “break points” at  $y_i$ ,  $i = 0, \dots, k$ , with values equal to the corresponding values of  $f^*$ . In particular, the epigraph of  $F^*$  is the convex hull of  $k + 1$  vertical halflines corresponding to  $y_0, \dots, y_k$ :

$$\text{epi}(F^*) = \text{conv}\left(\left\{\{(y_i, w_i) \mid f^*(y_i) \leq w_i\} \mid i = 0, \dots, k\right\}\right)$$

(see Fig. 6.4.5).

Note that the inner linearization  $F^*$  is determined by  $y_0, \dots, y_k$ , and is independent of  $x_0, \dots, x_k$ . This indicates that the same is true of its conjugate  $F$ , and indeed, since

$$f(x_i) - y'_i x_i = -f^*(y_i),$$

from Eq. (6.51) we obtain

$$F(x) = \max_{i=0, \dots, k} \{y'_i x - f^*(y_i)\}.$$

However, not every function of the above form qualifies as an outer linearization within our framework: it is necessary that for every  $y_i$  there exists  $x_i$  such that  $y_i \in \partial f(x_i)$ , or equivalently that  $\partial f^*(y_i) \neq \emptyset$  for all  $i = 0, \dots, k$ . Similarly, not every function of the form (6.52) qualifies as an inner linearization within our framework: it is necessary that  $\partial f^*(y_i) \neq \emptyset$  for all  $i = 0, \dots, k$ .

#### 6.4.4 Generalized Simplicial Decomposition

We will now describe a generalization of the simplicial decomposition method, which applies to the problem

$$\begin{aligned} & \text{minimize } f(x) + c(x) \\ & \text{subject to } x \in \mathfrak{R}^n, \end{aligned} \tag{6.53}$$

where  $f : \mathfrak{R}^n \mapsto (-\infty, \infty]$  and  $c : \mathfrak{R}^n \mapsto (-\infty, \infty]$  are closed proper convex functions. This is the problem of the Fenchel duality context, and it contains as a special case the problem to which the ordinary simplicial decomposition method of Section 6.4.2 applies (where  $f$  is differentiable, and  $c$  is the indicator function of a closed convex set).

Here we start with some finite set  $X_0 \subset \text{dom}(c)$ . At the typical iteration, given a finite set  $X_k \subset \text{dom}(c)$ , we use the following three steps to compute vectors  $x_k, \tilde{x}_{k+1}$ , and a new set  $X_{k+1} = X_k \cup \{\tilde{x}_{k+1}\}$  to start the next iteration:

- (1) We obtain

$$x_k \in \arg \min_{x \in \mathfrak{R}^n} \{f(x) + C_k(x)\}, \tag{6.54}$$

where  $C_k$  is the polyhedral/inner linearization function whose epigraph is the convex hull of the finite collection of rays  $\{(\tilde{x}, w) \mid c(\tilde{x}) \leq w\}$ ,  $\tilde{x} \in X_k$ .

- (2) We obtain a subgradient  $g_k \in \partial f(x_k)$  such that

$$-g_k \in \partial C_k(x_k); \tag{6.55}$$

the existence of such a subgradient is guaranteed by the optimality condition of Prop. 5.4.7, applied to the minimization in Eq. (6.54), under suitable conditions.

- (3) We obtain  $\tilde{x}_{k+1}$  such that

$$-g_k \in \partial c(\tilde{x}_{k+1}),$$

and form  $X_{k+1} = X_k \cup \{\tilde{x}_{k+1}\}$ .

We assume that  $f$  and  $c$  are such that the steps (1)-(3) above can be carried out, and we will provide conditions guaranteeing that this is so. Note that step (3) is equivalent to finding

$$\tilde{x}_{k+1} \in \arg \min_{x \in \mathbb{R}^n} \{g'_k(x - x_k) + c(x)\}, \quad (6.56)$$

and that this is a linear programming problem in the important special case where  $c$  is polyhedral. Note also that problem (6.54) is a linearized version of the original problem (6.53), where  $c$  is replaced by  $C_k(x)$ , which is an inner linearization of  $c$ . To see this, note that if  $X_k = \{\tilde{x}_i \mid i \in I_k\}$ , where  $I_k$  is a finite index set,  $C_k$  is given by

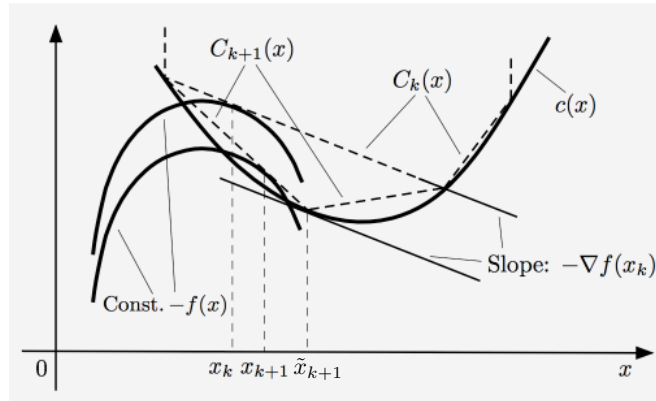
$$C_k(x) = \begin{cases} \inf_{\substack{\sum_{i \in I_k} \alpha_i \tilde{x}_i = x \\ \alpha_i \geq 0, \sum_{i \in I_k} \alpha_i = 1}} \sum_{i \in I_k} \alpha_i c(\tilde{x}_i) & \text{if } x \in \text{conv}(X_k), \\ \infty & \text{if } x \notin \text{conv}(X_k), \end{cases}$$

so the minimization (6.54) involves in effect the variables  $\alpha_i$ ,  $i \in I_k$ , and is equivalent to

$$\begin{aligned} & \text{minimize } f\left(\sum_{i \in I_k} \alpha_i \tilde{x}_i\right) + \sum_{i \in I_k} \alpha_i c(\tilde{x}_i) \\ & \text{subject to } \sum_{i \in I_k} \alpha_i = 1, \quad \alpha_i \geq 0, \quad i \in I_k. \end{aligned} \quad (6.57)$$

Let us note a few special cases where  $f$  is differentiable:

- (a) When  $c$  is the indicator function of a bounded polyhedral set  $X$ , and  $X_0 = \{x_0\}$ , the method reduces to the earlier simplicial decomposition method (6.47)-(6.48). Indeed, step (1) corresponds to the minimization (6.48), step (2) simply yields  $g_k = \nabla f(x_k)$ , and step (3), as implemented in Eq. (6.56), corresponds to solution of the linear program (6.47) that generates a new extreme point.
- (b) When  $c$  is polyhedral, the method can be viewed as essentially the special case of the earlier simplicial decomposition method (6.47)-(6.48) applied to the problem of minimizing  $f(x) + w$  subject to  $x \in X$  and  $(x, w) \in \text{epi}(c)$  [the only difference is that  $\text{epi}(c)$  is not bounded, but this is inconsequential if we assume that  $\text{dom}(c)$  is bounded, or more generally that the problem (6.54) has a solution]. In this case, the method terminates finitely, assuming that the vectors  $(\tilde{x}_{k+1}, c(\tilde{x}_{k+1}))$  obtained by solving the linear program (6.56) are extreme points of  $\text{epi}(c)$ .
- (c) When  $c$  is a general convex function, the method is illustrated in Fig. 6.4.6. The existence of a solution  $x_k$  to problem (6.54) is guaranteed



**Figure 6.4.6.** Illustration of successive iterates of the generalized simplicial decomposition method in the case where  $f$  is differentiable. Given the inner linearization  $C_k$  of  $c$ , we minimize  $f + C_k$  to obtain  $x_k$  (graphically, we move the graph of  $-f$  vertically until it touches the graph of  $C_k$ ). We then compute  $\tilde{x}_{k+1}$  as a point at which  $-\nabla f(x_k)$  is a subgradient of  $c$ , and we use it to form the improved inner linearization  $C_{k+1}$  of  $c$ . Finally, we minimize  $f + C_{k+1}$  to obtain  $x_{k+1}$  (graphically, we move the graph of  $-f$  vertically until it touches the graph of  $C_{k+1}$ ).

by the compactness of  $\text{conv}(X_k)$  and Weierstrass' Theorem, while step (2) yields  $g_k = \nabla f(x_k)$ . The existence of a solution to problem (6.56) must be guaranteed by some assumption such as coercivity of  $c$ .

Let us now consider the case where  $f$  is extended real-valued and nondifferentiable. Then, assuming that

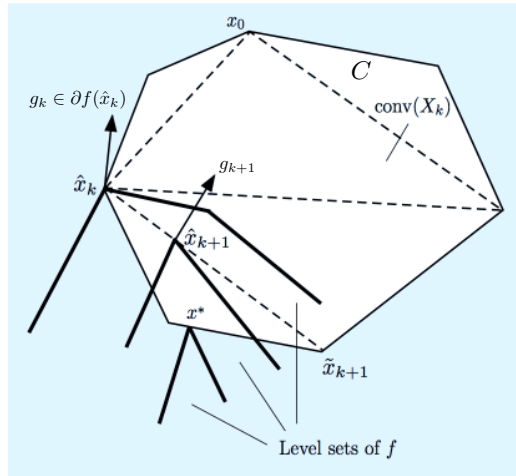
$$\text{ri}(\text{dom}(f)) \cap \text{conv}(X_0) \neq \emptyset,$$

the existence of the subgradient  $g_k$  is guaranteed by the optimality condition of Prop. 5.4.7, and the existence of a solution  $x_k$  to problem (6.54) is guaranteed by Weierstrass' Theorem. When  $c$  is the indicator function of a polyhedral set  $X$ , the condition of step (2) becomes

$$g'_k(\tilde{x} - x_k) \geq 0, \quad \forall \tilde{x} \in \text{conv}(X_k), \tag{6.58}$$

i.e.,  $-g_k$  is in the normal cone of  $\text{conv}(X_k)$  at  $x_k$ . The method is illustrated for this case in Fig. 6.4.7. It terminates finitely, assuming that the vector  $\tilde{x}_{k+1}$  obtained by solving the linear program (6.56) is an extreme point of  $X$ . The reason is that in view of Eq. (6.58), the vector  $\tilde{x}_{k+1}$  does not belong to  $X_k$  (unless  $x_k$  is optimal), so  $X_{k+1}$  is a strict enlargement of  $X_k$ . In the more general case where  $c$  is a general closed proper convex function,





**Figure 6.4.7.** Illustration of the generalized simplicial decomposition method for the case where  $f$  is nondifferentiable and  $c$  is the indicator function of a polyhedral set  $X$ . For each  $k$ , we compute a subgradient  $g_k \in \partial f(x_k)$  such that  $-g_k$  lies in the normal cone of  $\text{conv}(X_k)$  at  $x_k$ , and we use it to generate a new extreme point of  $X$ . Note that in contrast to the differentiable case, there may be multiple such subgradients.

the convergence of the method will be discussed later, in the context of a more general method.

Let us now address the calculation of a subgradient  $g_k \in \partial f(x_k)$  such that  $-g_k \in \partial C_k(x_k)$  [cf. Eq. (6.55)]. This may be a difficult problem as it may require knowledge of  $\partial f(x_k)$  as well as  $\partial C_k(x_k)$ . However, in special cases,  $g_k$  may be obtained simply as a byproduct of the minimization

$$x_k \in \arg \min_{x \in \mathbb{R}^n} \{f(x) + C_k(x)\}, \quad (6.59)$$

[cf. Eq. (6.54)]. In particular, consider the case where  $c$  is the indicator of a closed convex set  $X$ , and

$$f(x) = \max\{f_1(x), \dots, f_r(x)\},$$

where  $f_1, \dots, f_r$  are differentiable functions. Then the minimization (6.59) takes the form

$$\begin{aligned} &\text{minimize } z \\ &\text{subject to } f_j(x) \leq z, \quad j = 1, \dots, r, \quad x \in \text{conv}(X_k), \end{aligned} \quad (6.60)$$

where  $X_k$  is a polyhedral inner linearization to  $X$ . According to the optimality conditions of Prop. 5.3.2, the optimal solution  $(x_k, z^*)$  together

with dual optimal variables  $\mu_j^* \geq 0$ , satisfies

$$(x_k, z^*) \in \arg \min_{x \in \text{conv}(X_k), z \in \mathfrak{R}} \left\{ \left( 1 - \sum_{j=1}^r \mu_j^* \right) z + \sum_{j=1}^r \mu_j^* f_j(x) \right\}.$$

It follows that

$$\sum_{j=1}^r \mu_j^* = 1, \quad \mu_j^* \geq 0, \quad j = 1, \dots, r, \quad (6.61)$$

and

$$\left( \sum_{j=1}^r \mu_j^* \nabla f_j(x_k) \right)' (x - x_k) \geq 0, \quad \forall x \in \text{conv}(X_k). \quad (6.62)$$

From Eq. (6.61) and the analysis of Example 5.4.5, the vector

$$g_k = \sum_{j=1}^r \mu_j^* \nabla f_j(x_k) \quad (6.63)$$

is a subgradient of  $f$  at  $x_k$ . Furthermore, from Eq. (6.62), it follows that  $-g_k$  is in the normal cone of  $\text{conv}(X_k)$  at  $x_k$ .

In conclusion,  $g_k$ , as given by Eq. (6.63), is a suitable subgradient for determining a new extreme point via problem (6.56). Note an important advantage of this method over potential competitors: it involves solution of linear programs of the form (6.56) to generate new extreme points of  $X$ , and low-dimensional nonlinear programs of the form (6.60). When each  $f_j$  is twice differentiable, the latter programs can be solved by fast Newton-like methods, such as sequential quadratic programming (see e.g., [Ber82], [Ber99], [NoW06]).

### Dual/Cutting Plane Implementation

We now provide a dual implementation of generalized simplicial decomposition. The result is an outer linearization/cutting plane-type of method, which is mathematically equivalent to generalized simplicial decomposition. The idea is that the problem

$$\begin{aligned} & \text{minimize} && f(x) + c(x) \\ & \text{subject to} && x \in \mathfrak{R}^n, \end{aligned}$$

[cf. Eq. (6.53)] is in a form suitable for application of Fenchel duality (cf. Section 5.3.5, with the identifications  $f_1 = f$  and  $f_2 = c$ ). In particular, the dual problem is

$$\begin{aligned} & \text{minimize} && f_1^*(\lambda) + f_2^*(-\lambda) \\ & \text{subject to} && \lambda \in \mathfrak{R}^n, \end{aligned}$$

where  $f_1^*$  and  $f_2^*$  are the conjugates of  $f$  and  $c$ , respectively. The generalized simplicial decomposition algorithm (6.54)-(6.56) can alternatively be implemented by replacing  $f_2^*$  by a piecewise linear/cutting plane outer linearization, while leaving  $f_1^*$  unchanged, i.e., by solving at iteration  $k$  the problem

$$\begin{aligned} & \text{minimize} && f_1^*(\lambda) + F_{2,k}^*(-\lambda) \\ & \text{subject to} && \lambda \in \mathfrak{R}^n, \end{aligned} \tag{6.64}$$

where  $F_{2,k}^*$  is an outer linearization of  $f_2^*$  (the conjugate of  $C_k$ ). This problem is the (Fenchel) dual of problem (6.54):

$$\begin{aligned} & \text{minimize} && f(x) + C_k(x) \\ & \text{subject to} && x \in \mathfrak{R}^n, \end{aligned}$$

[or equivalently, the low-dimensional problem (6.57)].

Note that solutions of problem (6.64) are the subgradients  $g_k$  satisfying Eq. (6.55), while the associated subgradient of  $f_2^*$  at  $-g_k$  is the vector  $\tilde{x}_{k+1}$  generated by Eq. (6.56), as shown in Fig. 6.4.8. In fact, the function  $F_{2,k}^*$  has the form

$$F_{2,k}^*(-\lambda) = \max_{i \in I_{k-1}} \{f_2^*(-g_i) - \tilde{x}'_{i+1}(\lambda - g_i)\},$$

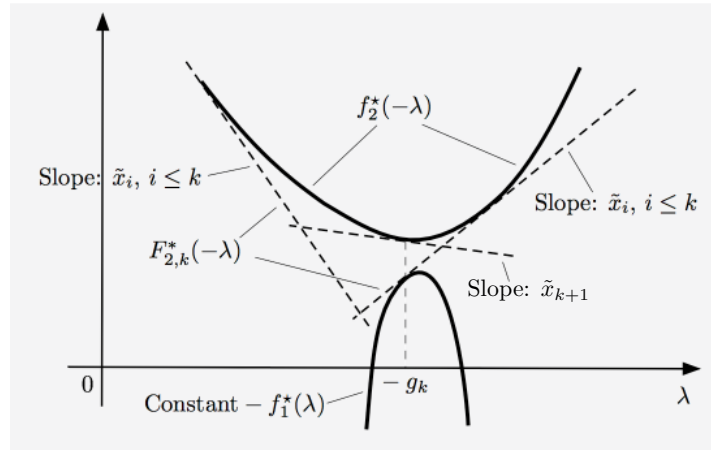
where  $g_i$  and  $\tilde{x}_{i+1}$  are vectors that can be obtained either by using the primal, the generalized simplicial decomposition method (6.54)-(6.56), or by using its dual, the cutting plane method based on solving the outer approximation problems (6.64). The ordinary cutting plane method, described in the beginning of Section 6.4.1, is obtained as the special case where  $f_1^*(\lambda) \equiv 0$ .

Whether the primal or the dual implementation is preferable depends on the structure of the functions  $f$  and  $c$ . When  $f$  (and hence also  $f_1^*$ ) is not polyhedral, the dual implementation may not be attractive, because it requires the  $n$ -dimensional nonlinear optimization (6.64) at each iteration, as opposed to the typically low-dimensional optimization (6.54). In the alternative case where  $f$  is polyhedral, both methods require the solution of linear programs.

### 6.4.5 Generalized Polyhedral Approximation

We will now consider a unified framework for polyhedral approximation, which combines the cutting plane and simplicial decomposition methods. We consider the problem

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m f_i(x_i) \\ & \text{subject to} && (x_1, \dots, x_m) \in S, \end{aligned} \tag{6.65}$$



**Figure 6.4.8.** Illustration of the cutting plane implementation of the generalized simplicial decomposition method. The ordinary cutting plane method, described in the beginning of Section 6.4.1, is obtained as the special case where  $f_1^*(x) \equiv 0$ . In this case,  $f$  is the indicator function of the set consisting of the just the origin, and the primal problem is to evaluate  $c(0)$ .

where  $(x_1, \dots, x_m)$  is a vector in  $\mathfrak{R}^{n_1 + \dots + n_m}$ , with components  $x_i \in \mathfrak{R}^{n_i}$ ,  $i = 1, \dots, m$ , and

$f_i : \mathfrak{R}^{n_i} \mapsto (-\infty, \infty]$  is a closed proper convex function for each  $i$ ,

$S$  is a subspace of  $\mathfrak{R}^{n_1 + \dots + n_m}$ .

We refer to this as an *extended monotropic program* (EMP for short).<sup>†</sup>

A classical example of EMP is single commodity network optimization problems where  $x_i$  represents the (scalar) flow of an arc of a directed graph, and  $S$  is the circulation subspace of the graph (see e.g., [Ber98]). Also problems involving general linear constraints and an additive convex cost function can be converted to EMP. In particular, the problem

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m f_i(x_i) && (6.66) \\ & \text{subject to} && Ax = b, && x_i \in X_i, && i = 1, \dots, m, \end{aligned}$$

---

<sup>†</sup> Monotropic programming, a class of problems introduced and extensively analyzed by Rockafellar in his book [Roc84], is the special case of problem (6.65) where each component  $x_i$  is one-dimensional (i.e.,  $n_i = 1$ ). The name “monotropic” means “turning in a single direction” in Greek, and captures the characteristic monotonicity property of convex functions of a single variable such as  $f_i$ .

where  $A$  is a given matrix and  $b$  is a given vector, is equivalent to

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m f_i(x_i) + \delta_Z(z) \\ & \text{subject to} && Ax - z = 0, \end{aligned}$$

where  $z$  is a vector of artificial variables, and  $\delta_Z$  is the indicator function of the set  $Z = \{z \mid z = b\}$ . This is an EMP where the constraint subspace is

$$S = \{(x, z) \mid Ax - z = 0\}.$$

When the functions  $f_i$  are linear, problem (6.66) reduces to a linear programming problem. When the functions  $f_i(x_i)$  are positive semidefinite quadratic, problem (6.66) reduces to a convex quadratic programming problem.

Note also that while the vectors  $x_1, \dots, x_m$  appear independently in the cost function

$$\sum_{i=1}^m f_i(x_i),$$

they may be coupled through the subspace constraint. For example, consider a cost function of the form

$$f(x) = \ell(x_1, \dots, x_m) + \sum_{i=1}^m f_i(x_i),$$

where  $\ell$  is a proper convex function of all the components  $x_i$ . Then, by introducing an auxiliary vector  $z \in \mathfrak{R}^{n_1 + \dots + n_m}$ , the problem of minimizing  $f$  over a subspace  $X$  can be transformed to the problem

$$\begin{aligned} & \text{minimize} && \ell(z) + \sum_{i=1}^m f_i(x_i) \\ & \text{subject to} && (x, z) \in S, \end{aligned}$$

where  $S$  is the subspace of  $\mathfrak{R}^{2(n_1 + \dots + n_m)}$

$$S = \{(x, x) \mid x \in X\}.$$

This problem is of the form (6.65).

Another problem that can be converted to the EMP format (6.65) is

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m f_i(x) \\ & \text{subject to} && x \in X, \end{aligned} \tag{6.67}$$

where  $f_i : \mathfrak{R}^n \mapsto (-\infty, \infty]$  are proper convex functions, and  $X$  is a subspace of  $\mathfrak{R}^n$ . This can be done by introducing  $m$  copies of  $x$ , i.e., auxiliary vectors  $z_i \in \mathfrak{R}^n$  that are constrained to be equal, and write the problem as

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m f_i(z_i) \\ & \text{subject to} && (z_1, \dots, z_m) \in S, \end{aligned}$$

where  $S$  is the subspace

$$S = \{(x, \dots, x) \mid x \in X\}.$$

It can thus be seen that convex problems with linear constraints can generally be formulated as EMP. We will see that these problems share a powerful and symmetric duality theory, which is similar to Fenchel duality and forms the basis for a symmetric and general framework for polyhedral approximation.

### The Dual Problem

To derive the appropriate dual problem, we introduce auxiliary vectors  $z_i \in \mathfrak{R}^{n_i}$  and we convert the EMP (6.65) to the equivalent form

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m f_i(z_i) \\ & \text{subject to} && z_i = x_i, \quad i = 1, \dots, m, \quad (x_1, \dots, x_m) \in S. \end{aligned} \tag{6.68}$$

We then assign a multiplier vector  $\lambda_i \in \mathfrak{R}^{n_i}$  to the constraint  $z_i = x_i$ , thereby obtaining the Lagrangian function

$$L(x_1, \dots, x_m, z_1, \dots, z_m, \lambda_1, \dots, \lambda_m) = \sum_{i=1}^m (f_i(z_i) + \lambda_i'(x_i - z_i)). \tag{6.69}$$

The dual function is

$$\begin{aligned} q(\lambda) &= \inf_{(x_1, \dots, x_m) \in S, z_i \in \mathfrak{R}^{n_i}} L(x_1, \dots, x_m, z_1, \dots, z_m, \lambda_1, \dots, \lambda_m) \\ &= \inf_{(x_1, \dots, x_m) \in S} \sum_{i=1}^m \lambda_i' x_i + \sum_{i=1}^m \inf_{z_i \in \mathfrak{R}^{n_i}} \{f_i(z_i) - \lambda_i' z_i\} \\ &= \begin{cases} \sum_{i=1}^m q_i(\lambda_i) & \text{if } (\lambda_1, \dots, \lambda_m) \in S^\perp, \\ -\infty & \text{otherwise,} \end{cases} \end{aligned}$$

where

$$q_i(\lambda_i) = \inf_{z_i \in \mathbb{R}^{n_i}} \{f_i(z_i) - \lambda_i' z_i\}, \quad i = 1, \dots, m,$$

and  $S^\perp$  is the orthogonal subspace of  $S$ .

Note that since  $q_i$  can be written as

$$q_i(\lambda_i) = - \sup_{z_i \in \mathbb{R}^{n_i}} \{\lambda_i' z_i - f_i(z_i)\},$$

it follows that  $-q_i$  is the conjugate of  $f_i$ , so by Prop. 1.6.1,  $-q_i$  is a closed proper convex function. The dual problem is

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^m q_i(\lambda_i) \\ & \text{subject to} && (\lambda_1, \dots, \lambda_m) \in S^\perp. \end{aligned} \tag{6.70}$$

Thus, with a change of sign to convert maximization to minimization, the dual problem becomes

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m f_i^*(\lambda_i) \\ & \text{subject to} && (\lambda_1, \dots, \lambda_m) \in S^\perp, \end{aligned} \tag{6.71}$$

where  $f_i^*$  is the conjugate of  $f_i$ , and has the same form as the primal. Furthermore, assuming that the functions  $f_i$  are closed, when the dual problem is dualized, it yields the primal problem, and the duality is fully symmetric.

Throughout our duality analysis of this section, we denote by  $f_{opt}$  and  $q_{opt}$  the optimal primal and dual values, and in addition to the convexity assumption on  $f_i$  made earlier, we will assume that appropriate conditions hold that guarantee the strong duality relation  $f_{opt} = q_{opt}$ .

Since the EMP problem can be viewed as a special case of the convex programming problem of Section 5.3, it is possible to obtain optimality conditions as special cases of the corresponding conditions (cf. Prop. 5.3.3). In particular, it can be seen that a pair  $(x, \lambda)$  satisfies the Lagrangian optimality condition of Prop. 5.3.3, applied to the Lagrangian (6.69), if and only if  $x_i$  attains the infimum in the equation

$$q_i(\lambda_i) = \inf_{z_i \in \mathbb{R}^{n_i}} \{f_i(z_i) - \lambda_i' z_i\}, \quad i = 1, \dots, m,$$

or equivalently,

$$\lambda_i \in \partial f_i(x_i), \quad i = 1, \dots, m. \tag{6.72}$$

Thus, by applying Prop. 5.3.3, we obtain the following.

**Proposition 6.4.3: (EMP Optimality Conditions)** There holds  $-\infty < q_{opt} = f_{opt} < \infty$  and  $(x_1^{opt}, \dots, x_m^{opt}, \lambda_1^{opt}, \dots, \lambda_m^{opt})$  are an optimal primal and dual solution pair of the EMP problem if and only if

$$(x_1^{opt}, \dots, x_m^{opt}) \in S, \quad (\lambda_1^{opt}, \dots, \lambda_m^{opt}) \in S^\perp,$$

and

$$x_i^{opt} \in \arg \min_{x_i \in \mathbb{R}^n} \{f_i(x_i) - x_i' \lambda_i^{opt}\}, \quad i = 1, \dots, m. \quad (6.73)$$

Note that by the Conjugate Subgradient Theorem (Prop. 5.4.3), the condition (6.73) of the preceding proposition is equivalent to either one of the following two subgradient conditions:

$$\lambda_i^{opt} \in \partial f_i(x_i^{opt}), \quad x_i^{opt} \in \partial f_i^*(\lambda_i^{opt}).$$

### General Polyhedral Approximation Scheme

The EMP formalism allows a broad and elegant algorithmic framework that combines elements of the cutting plane and simplicial decomposition methods of the preceding sections. In particular, problem (6.71) will be approximated, by using inner or outer linearization of some of the functions  $f_i$ . The optimal solution of the dual approximate problem will then be used to construct more refined inner and outer linearizations.

We introduce an algorithm that uses a fixed partition of the index set  $\{1, \dots, m\}$ :

$$\{1, \dots, m\} = I \cup \underline{I} \cup \bar{I}$$

that determines which of the functions  $f_i$  are outer approximated (set  $\underline{I}$ ) and inner approximated (set  $\bar{I}$ ).

For  $i \in \underline{I}$ , given a finite set  $\Lambda_i \subset \text{dom}(f_i^*)$  such that  $\partial f_i^*(\tilde{\lambda}) \neq \emptyset$  for all  $\tilde{\lambda} \in \Lambda_i$ , we consider the outer linearization of  $f_i$  corresponding to  $\Lambda_i$ :

$$\underline{f}_{i, \Lambda_i}(x_i) = \max_{\tilde{\lambda} \in \Lambda_i} \{\tilde{\lambda}' x_i - f_i^*(\tilde{\lambda})\},$$

or equivalently, as mentioned in Section 6.4.3,

$$\underline{f}_{i, \Lambda_i}(x_i) = \max_{\tilde{\lambda} \in \Lambda_i} \{f_i(x_{\tilde{\lambda}}) + \tilde{\lambda}'(x_i - x_{\tilde{\lambda}})\},$$

where for each  $\tilde{\lambda} \in \Lambda_i$ ,  $x_{\tilde{\lambda}}$  is such that  $\tilde{\lambda} \in \partial f_i(x_{\tilde{\lambda}})$ .



For  $i \in \bar{I}$ , given a finite set  $X_i \subset \text{dom}(f_i)$  such that  $\partial f_i(\tilde{x}) \neq \emptyset$  for all  $\tilde{x} \in X_i$ , we consider the inner linearization of  $f_i$  corresponding to  $X_i$  by

$$\bar{f}_{i,X_i}(x_i) = \begin{cases} \min \begin{cases} \sum_{\tilde{x} \in X_i} \alpha_{\tilde{x}} \tilde{x} = x_i, & \sum_{\tilde{x} \in X_i} \alpha_{\tilde{x}} f_i(\tilde{x}) \\ \sum_{\tilde{x} \in X_i} \alpha_{\tilde{x}} = 1, \alpha_{\tilde{x}} \geq 0, \tilde{x} \in X_i \end{cases} & \text{if } x_i \in \text{conv}(X_i), \\ \infty & \text{otherwise.} \end{cases}$$

As mentioned in Section 6.4.3, this is the function whose epigraph is the convex hull of the halflines  $\{(x_i, w) \mid f_i(x_i) \leq w\}$ ,  $x_i \in X_i$  (cf. Fig. 6.4.5).

We assume that at least one of the sets  $\underline{I}$  and  $\bar{I}$  is nonempty. At the start of the typical iteration, we have for each  $i \in \underline{I}$ , a finite subset  $\Lambda_i \subset \text{dom}(f_i^*)$ , and for each  $i \in \bar{I}$ , a finite subset  $X_i \subset \text{dom}(f_i)$ . The iteration is as follows:

**Typical Iteration:**

Find a primal-dual optimal solution pair  $(\hat{x}, \hat{\lambda}) = (\hat{x}_1, \hat{\lambda}_1, \dots, \hat{x}_m, \hat{\lambda}_m)$  of the EMP

$$\text{minimize} \quad \sum_{i \in \underline{I}} f_i(x_i) + \sum_{i \in \underline{I}} \underline{f}_{i,\Lambda_i}(x_i) + \sum_{i \in \bar{I}} \bar{f}_{i,X_i}(x_i) \quad (6.74)$$

subject to  $(x_1, \dots, x_m) \in S$ ,

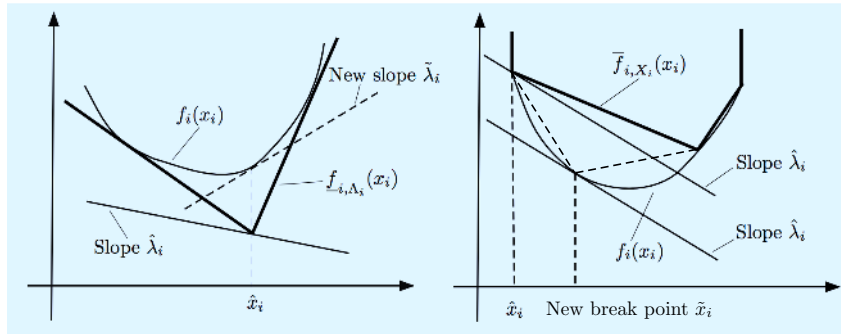
where  $\underline{f}_{i,\Lambda_i}$  and  $\bar{f}_{i,X_i}$  are the outer and inner linearizations of  $f_i$  corresponding to  $X_i$  and  $\Lambda_i$ , respectively. Then enlarge the sets  $X_i$  and  $\Lambda_i$  as follows (see Fig. 6.4.9):

- (a) For  $i \in \underline{I}$ , compute a subgradient  $\tilde{\lambda}_i \in \partial f_i(\hat{x}_i)$  and we add  $\tilde{\lambda}_i$  to the corresponding set  $\Lambda_i$ .
- (b) For  $i \in \bar{I}$ , compute a subgradient  $\tilde{x}_i \in \partial f_i^*(\hat{\lambda}_i)$  and we add  $\tilde{x}_i$  to the corresponding set  $X_i$ .

If there is no strict enlargement, i.e., for all  $i \in \underline{I}$  we have  $\tilde{\lambda}_i \in \Lambda_i$ , and for all  $i \in \bar{I}$  we have  $\tilde{x}_i \in X_i$ , the algorithm terminates.

We will show in a subsequent proposition that if the algorithm terminates,  $(\hat{x}_1, \dots, \hat{x}_m, \hat{\lambda}_1, \dots, \hat{\lambda}_m)$  is a primal and dual optimal solution pair. If there is strict enlargement and the algorithm does not terminate, we proceed to the next iteration, using the enlarged sets  $\Lambda_i$  and  $X_i$ .

Note that we implicitly assume that at each iteration, there exists a primal and dual optimal solution pair of problem (6.74). Furthermore, we assume that the enlargement step can be carried out, i.e., that  $\partial f_i(\hat{x}_i) \neq \emptyset$  for all  $i \in \underline{I}$  and  $\partial f_i^*(\hat{\lambda}_i) \neq \emptyset$  for all  $i \in \bar{I}$ . Sufficient assumptions may need to be imposed on the problem to guarantee that this is so.



**Figure 6.4.9.** Illustration of the enlargement step in the polyhedral approximation method, after we obtain a primal-dual optimal solution pair  $(\hat{x}, \hat{\lambda}) = (\hat{x}_1, \hat{\lambda}_1, \dots, \hat{x}_m, \hat{\lambda}_m)$ . Note that in the figure on the right, we use the fact

$$\tilde{x}_i \in \partial f_i^*(\tilde{\lambda}_i) \iff \hat{\lambda}_i \in \partial f_i(\tilde{x}_i)$$

(cf. the Conjugate Subgradient Theorem, Prop. 5.4.3). The enlargement step on the left (finding  $\tilde{\lambda}_i$ ) is also equivalent to  $\tilde{\lambda}_i$  satisfying  $\hat{x}_i \in \partial f_i^*(\tilde{\lambda}_i)$ , or equivalently, solving the optimization problem

$$\begin{aligned} &\text{maximize} && \{ \tilde{\lambda}'_i \hat{x}_i - f_i^*(\tilde{\lambda}_i) \} \\ &\text{subject to} && \tilde{\lambda}_i \in \mathfrak{R}^{n_i}. \end{aligned}$$

The enlargement step on the right (finding  $\tilde{x}_i$ ) is also equivalent to solving the optimization problem

$$\begin{aligned} &\text{maximize} && \{ \hat{\lambda}'_i x_i - f_i(x_i) \} \\ &\text{subject to} && x_i \in \mathfrak{R}^{n_i}. \end{aligned}$$

We refer to the preceding algorithm as the *generalized polyhedral approximation* or GPA algorithm. Note two prerequisites for the algorithm to be effective:

- (1) The (partially) linearized problem (6.74) must be easier to solve than the original problem (6.71). For example, problem (6.74) may be a linear program, while the original may be nonlinear (cf. the cutting plane method of Section 6.4.1); or it may effectively have much smaller dimension than the original (cf. the simplicial decomposition method of Section 6.4.2).
- (2) Finding the enlargement vectors ( $\tilde{\lambda}_i$  for  $i \in \underline{I}$ , and  $\tilde{x}_i$  for  $i \in \bar{I}$ ) must not be too difficult. This can be done by the differentiation  $\tilde{\lambda}_i \in \partial f_i(\hat{x}_i)$  for  $i \in \underline{I}$ , and  $\tilde{x}_i \in \partial f_i^*(\hat{\lambda}_i)$  or  $i \in \bar{I}$ . Alternatively, if this

is not convenient for some of the functions (e.g., because some of the  $f_i$  or the  $f_i^*$  are not available in closed form), one may calculate  $\lambda_i$  and/or  $\tilde{x}_i$  via the relations

$$\hat{x}_i \in \partial f_i^*(\tilde{\lambda}_i), \quad \hat{\lambda}_i \in \partial f_i(\tilde{x}_i);$$

(cf. the Conjugate Subgradient Theorem, Prop. 5.4.3). This involves solving optimization problems. For example, finding  $\tilde{x}_i$  such that  $\hat{\lambda}_i \in \partial f_i(\tilde{x}_i)$  for  $i \in \bar{I}$  is equivalent to solving the problem

$$\begin{aligned} & \text{maximize} && \{\hat{\lambda}'_i x_i - f_i(x_i)\} \\ & \text{subject to} && x_i \in \mathfrak{R}^{n_i}, \end{aligned}$$

and may be nontrivial (cf. Fig. 6.4.9).

The facility of solving the linearized problem (6.74) and carrying out the subsequent enlargement step may guide the choice of functions that are inner or outer linearized. If  $x_i$  is one-dimensional, which is typically true in separable-type problems, the enlargement step is typically quite easy.

There are two potential advantages of the GPA algorithm over the earlier cutting plane and simplicial decomposition methods, depending on the problem's structure:

- (a) The refinement process may be faster, because at each iteration, multiple cutting planes and break points are added (as many as one per function  $f_i$ ). As a result, in a single iteration, a more refined approximation may result, compared with classical methods where a single cutting plane or extreme point is added. Moreover, when the component functions  $f_i$  are scalar, adding a cutting plane/break point to the polyhedral approximation of  $f_i$  can be very simple, as it requires a one-dimensional differentiation or minimization for each  $f_i$ .
- (b) The approximation process may preserve some of the special structure of the cost function and/or the constraint set. For example if the component functions  $f_i$  are scalar, or have partially overlapping dependences, e.g.,

$$\begin{aligned} f(x_1, \dots, x_m) = & f_1(x_1, x_2) + f_2(x_2, x_3) + \dots \\ & + f_{m-1}(x_{m-1}, x_m) + f_m(x_m), \end{aligned}$$

the minimization of  $f$  by the classical cutting plane method leads to general/unstructured linear programming problems. By contrast, using separate outer approximation of the components functions leads to linear programs with special structure, which can be solved efficiently by specialized methods, such as network flow algorithms, or interior point algorithms that can exploit the sparsity structure of the problem.

The symmetric duality of the EMP can be exploited in the implementation of the GPA algorithm. In particular, the algorithm may be applied to the dual problem of problem (6.71):

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m f_i^*(\lambda_i) \\ & \text{subject to} && (\lambda_1, \dots, \lambda_m) \in S^\perp, \end{aligned} \quad (6.75)$$

where  $f_i^*$  is the conjugate of  $f_i$ . Then the inner (or outer) linearized index set  $\bar{I}$  of the primal becomes the outer (or inner, respectively) linearized index set of the dual. At each iteration, the algorithm solves the approximate dual EMP,

$$\begin{aligned} & \text{minimize} && \sum_{i \in I} f_i^*(\lambda_i) + \sum_{i \in \underline{I}} \bar{f}_{i, \Lambda_i}^*(\lambda_i) + \sum_{i \in \bar{I}} \underline{f}_{i, X_i}^*(\lambda_i) \\ & \text{subject to} && (\lambda_1, \dots, \lambda_m) \in S^\perp, \end{aligned} \quad (6.76)$$

which is simply the dual of the approximate primal EMP (6.74) [since the outer (or inner) linearization of  $f_i^*$  is the conjugate of the inner (or respectively, outer) linearization of  $f_i$ ]. Thus the algorithm produces mathematically identical results when applied to the primal or the dual EMP. The choice of whether to apply the algorithm in its primal or its dual form is a simply a matter of whether calculations with  $f_i$  or with their conjugates  $f_i^*$  are more or less convenient. In fact, when the algorithm makes use of both the primal solution  $(\hat{x}_1, \dots, \hat{x}_m)$  and the dual solution  $(\hat{\lambda}_1, \dots, \hat{\lambda}_m)$  in the enlargement step, the question of whether the starting point is the primal or the dual EMP becomes moot: it is best to view the algorithm as applied to the pair of primal and dual EMP, without designation of which is primal and which is dual.

### Termination and Convergence

Now let us show the optimality of the primal and dual solution pair obtained upon termination of the algorithm. We will use two basic properties of outer approximations. The first is that for closed proper convex functions  $f$  and  $\underline{f}$ ,

$$\underline{f} \leq f, \quad \underline{f}(x) = f(x) \quad \implies \quad \partial \underline{f}(x) \subset \partial f(x). \quad (6.77)$$

The second is that for an outer linearization  $\underline{f}_\Lambda$  of  $f$ ,

$$\tilde{\lambda} \in \Lambda, \quad \tilde{\lambda} \in \partial f(x) \quad \implies \quad \underline{f}_\Lambda(x) = f(x). \quad (6.78)$$

The first property follows from the definition of subgradients, whereas the second property follows from the definition of  $\underline{f}_\Lambda$ .

**Proposition 6.4.4: (Optimality at Termination)** If the GPA algorithm terminates at some iteration, the corresponding primal and dual solutions,  $(\hat{x}_1, \dots, \hat{x}_m)$  and  $(\hat{\lambda}_1, \dots, \hat{\lambda}_m)$ , form a primal and dual optimal solution pair of the EMP problem.

**Proof:** From Prop. 6.4.3 and the definition of  $(\hat{x}_1, \dots, \hat{x}_m)$  and  $(\hat{\lambda}_1, \dots, \hat{\lambda}_m)$  as a primal and dual optimal solution pair of the approximate problem (6.74), we have

$$(\hat{x}_1, \dots, \hat{x}_m) \in S, \quad (\hat{\lambda}_1, \dots, \hat{\lambda}_m) \in S^\perp.$$

We will show that upon termination, we have for all  $i$

$$\hat{\lambda}_i \in \partial f_i(\hat{x}_i), \quad (6.79)$$

which by Prop. 6.4.3 implies the desired conclusion.

Since  $(\hat{x}_1, \dots, \hat{x}_m)$  and  $(\hat{\lambda}_1, \dots, \hat{\lambda}_m)$  are a primal and dual optimal solution pair of problem (6.74), Eq. (6.79) holds for all  $i \notin \underline{I} \cup \bar{I}$  (cf. Prop. 6.4.3). We will complete the proof by showing that it holds for all  $i \in \underline{I}$  (the proof for  $i \in \bar{I}$  follows by a dual argument).

Indeed, let us fix  $i \in \underline{I}$  and let  $\tilde{\lambda}_i \in \partial f_i(\hat{x}_i)$  be the vector generated by the enlargement step upon termination. We must have  $\tilde{\lambda}_i \in \Lambda_i$ , since there is no strict enlargement upon termination. Since  $\underline{f}_{i, \Lambda_i}$  is an outer linearization of  $f_i$ , by Eq. (6.78), the fact  $\tilde{\lambda}_i \in \Lambda_i, \tilde{\lambda}_i \in \partial f_i(\hat{x}_i)$  implies

$$\underline{f}_{i, \Lambda_i}(\hat{x}_i) = f_i(\hat{x}_i),$$

which in turn implies by Eq. (6.77) that

$$\partial \underline{f}_{i, \Lambda_i}(\hat{x}_i) \subset f_i(\hat{x}_i).$$

By Prop. 6.4.3, we also have  $\hat{\lambda}_i \in \partial \underline{f}_{i, \Lambda_i}(\hat{x}_i)$ , so  $\hat{\lambda}_i \in \partial f_i(\hat{x}_i)$ . **Q.E.D.**

As in Sections 6.4.1, 6.4.2, convergence can be easily established in the case where the functions  $f_i, i \in \bar{I} \cup \underline{I}$ , are polyhedral, assuming that care is taken to ensure that the corresponding enlargements vectors  $\lambda_i$  are chosen from a finite set of extreme points. In particular, assume that:

- (a) All outer linearized functions  $f_i$  are real-valued and polyhedral, and all inner linearized functions  $f_i$  the conjugates  $f_i^*$  are real-valued and polyhedral.
- (b) The vectors  $\tilde{\lambda}_i$  and  $\tilde{x}_i$  added to the polyhedral approximations are elements of the finite representations of the corresponding  $f_i^*$  and  $f_i$ .

Then at each iteration there are two possibilities: either  $(\hat{x}, \hat{\lambda})$  is an optimal primal-dual pair for the original problem and the algorithm terminates, or the approximation of one of the  $f_i$ ,  $i \in \underline{I} \cup \bar{I}$ , will be refined/improved. Since there can be only a finite number of refinements, convergence in a finite number of iterations follows.

Other convergence results are possible, extending some of the analysis of Sections 6.4.1, 6.4.2. In particular, if the set  $\bar{I}$  is empty (no inner approximation) and the sequence  $\{\tilde{\lambda}_i^k\}$  is bounded for every  $i \in \underline{I}$ , then every limit point of  $\{\hat{x}^k\}$  is primal optimal. To see this, note that for all  $k, \ell \leq k-1$ , and  $(x_1, \dots, x_m) \in S$ , we have

$$\begin{aligned} \sum_{i \notin \underline{I}} f_i(\hat{x}_i^k) + \sum_{i \in \underline{I}} (f_i(\hat{x}_i^\ell) + (\hat{x}_i^k - \hat{x}_i^\ell)' \tilde{\lambda}_i^\ell) &\leq \sum_{i \notin \underline{I}} f_i(\hat{x}_i^k) + \sum_{i \in \underline{I}} f_{-i, \Lambda_i^{k-1}}(\hat{x}_i^k) \\ &\leq \sum_{i=1}^m f_i(x_i). \end{aligned}$$

Let  $\{\hat{x}^k\}_{\mathcal{K}}$  be a subsequence converging to a vector  $\bar{x}$ . By taking the limit as  $\ell \rightarrow \infty$ ,  $k \in \mathcal{K}$ ,  $\ell \in \mathcal{K}$ ,  $\ell < k$ , and using the closedness of  $f_i$ , we obtain

$$\sum_{i=1}^m f_i(\bar{x}_i) \leq \liminf_{k \rightarrow \infty, k \in \mathcal{K}} \sum_{i \notin \underline{I}} f_i(\hat{x}_i^k) + \liminf_{\ell \rightarrow \infty, \ell \in \mathcal{K}} \sum_{i \in \underline{I}} f_i(\hat{x}_i^\ell) \leq \sum_{i=1}^m f_i(x_i)$$

for all  $(x_1, \dots, x_m) \in S$ . It follows that  $\bar{x}$  is primal optimal, i.e., every limit point of  $\{\hat{x}^k\}$  is optimal. The preceding convergence argument also goes through even if the sequences  $\{\tilde{\lambda}_i^k\}$  are not assumed bounded, as long as the limit points  $\bar{x}_i$  belong to the relative interior of the corresponding functions  $f_i$  (this follows from the subgradient decomposition result of Prop. 5.4.1).

Exchanging the roles of primal and dual, we similarly obtain a convergence result for the case where  $\underline{I}$  is empty (no outer linearization): assuming that the sequence  $\{\tilde{x}_i^k\}$  is bounded for every  $i \in \bar{I}$ , every limit point of  $\{\hat{\lambda}^k\}$  is dual optimal.

We finally state a more general convergence result from Bertsekas and Yu [BeY09], which applies to the mixed case where we simultaneously use outer and inner approximation (both  $\bar{I}$  and  $\underline{I}$  are nonempty). The proof is more complicated than the preceding ones, and we refer to [BeY09] for a detailed analysis.

**Proposition 6.4.5:** Consider the GPA algorithm. Let  $(\hat{x}^k, \hat{\lambda}^k)$  be a primal and dual optimal solution pair of the approximate problem at the  $k$ th iteration, and let  $\tilde{\lambda}_i^k, i \in \underline{I}$  and  $\tilde{x}_i^k, i \in \bar{I}$  be the vectors generated at the corresponding enlargement step. Suppose that there exist convergent subsequences  $\{\hat{x}_i^k\}_{\mathcal{K}}, i \in \underline{I}$ ,  $\{\tilde{\lambda}_i^k\}_{\mathcal{K}}, i \in \bar{I}$ , such that the sequences  $\{\tilde{\lambda}_i^k\}_{\mathcal{K}}, i \in \underline{I}$ ,  $\{\tilde{x}_i^k\}_{\mathcal{K}}, i \in \bar{I}$ , are bounded. Then:

- (a) Any limit point of the sequence  $\{(\hat{x}^k, \hat{\lambda}^k)\}_{\mathcal{K}}$  is a primal and dual optimal solution pair of the original problem.
- (b) The sequence of optimal values of the approximate problems converges to the optimal value  $f_{opt}$ .

### Application to Generalized Simplicial Decomposition

Let us now show that the general polyhedral approximation scheme contains as a special case the algorithm of the preceding section for the problem

$$\begin{aligned} & \text{minimize} && f(x) + c(x) \\ & \text{subject to} && x \in \mathfrak{R}^n, \end{aligned} \tag{6.80}$$

where  $f : \mathfrak{R}^n \mapsto (-\infty, \infty]$  and  $c : \mathfrak{R}^n \mapsto (-\infty, \infty]$  are closed, proper, convex functions; cf. Section 6.4.4. As a consequence, it also contains as special cases the ordinary cutting plane and simplicial decomposition methods of Sections 6.4.1 and 6.4.2, respectively.

We recast the problem into the EMP

$$\begin{aligned} & \text{minimize} && f_1(x_1) + f_2(x_2) \\ & \text{subject to} && (x_1, x_2) \in S, \end{aligned}$$

where

$$f_1(x_1) = f(x_1), \quad f_2(x_2) = c(x_2), \quad S = \{(x_1, x_2) \mid x_1 = x_2\}.$$

The dual problem takes the form

$$\begin{aligned} & \text{minimize} && f_1^*(\lambda_1) + f_2^*(\lambda_2) \\ & \text{subject to} && (\lambda_1, \lambda_2) \in S^\perp, \end{aligned}$$

where  $f_1^*$  and  $f_2^*$  are the conjugates of  $f_1$  and  $f_2$ , respectively. Since

$$S^\perp = \{(\lambda_1, \lambda_2) \mid \lambda_1 = -\lambda_2\},$$

the dual problem is

$$\begin{aligned} & \text{minimize} && f_1^*(\lambda) + f_2^*(-\lambda) \\ & \text{subject to} && \lambda \in \mathfrak{R}^n. \end{aligned}$$

Let  $f_2$  be replaced by an inner linearization  $\bar{f}_{2,X}$  or by an outer linearization  $\underline{f}_{2,-\Lambda}$ , and let  $(\hat{\lambda}, -\hat{\lambda})$  be a dual optimal solution at the typical iteration. At the end of the iteration,  $X$  is enlarged to include a vector  $\tilde{x}$  such that  $-\hat{\lambda} \in \partial f_2(\tilde{x})$  in the case of inner linearization, or  $\Lambda$  is enlarged to include  $\hat{\lambda}$  in the case of outer linearization. A comparison with the development of Section 6.4.4 verifies that when inner (or outer) linearization of  $f_2$  is used, this method coincides with the generalized simplicial decomposition algorithm (or cutting plane algorithm, respectively) given there.

### Application to Network Optimization and Monotropic Programming

Network optimization problems involve a directed graph with set of nodes  $\mathcal{N}$  and set of arcs  $\mathcal{A}$ . A classical problem is to minimize a cost function

$$\sum_{a \in \mathcal{A}} f_a(x_a),$$

where  $f_a$  is a scalar closed proper convex function, and  $x_a$  is the flow of arc  $a \in \mathcal{A}$ . The minimization is over all flow vectors  $x = \{x_a \mid a \in \mathcal{A}\}$  that belong to the circulation subspace  $S$  of the graph (at each node, the sum of all incoming arc flows is equal to the sum of all outgoing arc flows).

The GPA method that uses inner linearization of all the functions  $f_a$  that are nonlinear is particularly attractive for this problem, because of the favorable structure of the corresponding approximate EMP:

$$\begin{aligned} & \text{minimize} && \sum_{a \in \mathcal{A}} \bar{f}_{a, X_a}(x_a) \\ & \text{subject to} && x \in S, \end{aligned}$$

where for each arc  $a$ ,  $\bar{f}_{a, X_a}$  is the inner approximation of  $f_a$ , corresponding to a finite set of break points  $X_a \subset \text{dom}(f_a)$ . By suitably introducing multiple arcs in place of each arc, we can recast this problem as a linear minimum cost network flow problem that can be solved using very fast polynomial algorithms. These algorithms, simultaneously with an optimal primal (flow) vector, yield a dual optimal (price differential) vector (see e.g., [Ber98], Chapters 5-7). Furthermore, because the functions  $f_a$  are scalar, the enlargement step is very simple.

Some of the preceding advantages of GPA method with inner linearization carry over to monotropic programming problems ( $n_i = 1$  for all  $i$ ), the key idea being the simplicity of the enlargement step. Furthermore, there are effective algorithms for solving the associated approximate primal and dual EMP, such as out-of-kilter methods [Roc84], [Tse01], and  $\epsilon$ -relaxation methods [Ber98], [TsB00].

## 6.5 PROXIMAL AND BUNDLE METHODS

As discussed in the preceding section, one of the drawbacks of the cutting plane method is the instability phenomenon, whereby the method can take large steps away from the current point, with significant deterioration of the cost function value. A way to limit the effects of this is to add to the polyhedral function approximation a quadratic term  $p_k(x)$  that penalizes



large deviations from some reference point  $y_k$ . Thus in this method,  $x_{k+1}$  is obtained as

$$x_{k+1} \in \arg \min_{x \in X} \{F_k(x) + p_k(x)\}, \quad (6.81)$$

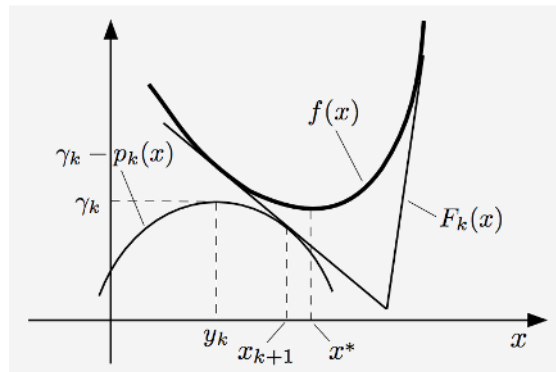
where similar to the cutting plane method,

$$F_k(x) = \max \{f(x_0) + (x - x_0)'g_0, \dots, f(x_k) + (x - x_k)'g_k\},$$

and

$$p_k(x) = \frac{1}{2c_k} \|x - y_k\|^2,$$

where  $c_k$  is a positive scalar parameter (cf. Fig. 6.5.1). We refer to  $p_k(x)$  as the *proximal term*, and to its center  $y_k$  as the *proximal center* (the method for choosing  $y_k$  will be described later; often  $y_k = x_k$ ). The purpose of the proximal term is to provide a measure of stability to the cutting plane method at the expense of solving a more difficult subproblem at each iteration (e.g., a quadratic versus a linear program, in the case where  $X$  is polyhedral).



**Figure 6.5.1.** Using a proximal term to reduce the effect of instability in the cutting plane method. The point  $x_{k+1}$  is the one at which the graph of the negative proximal term  $-p_k(x)$ , raised by some amount  $\gamma_k$ , just touches the graph of  $F_k$ . Then  $x_{k+1}$  tends to be closer to the proximal center  $y_k$ , with the distance  $\|x_{k+1} - y_k\|$  depending on the size of the proximal term, i.e., the penalty parameter  $c_k$ .

We can view iteration (6.81) as an approximate version of a general algorithm for minimizing a convex function. We first discuss this algorithm in the next section, and we then focus on approximations that use subgradients.



The degree of regularization is controlled by the parameter  $c_k$ . For small values of  $c_k$ ,  $x_{k+1}$  tends to stay close to  $x_k$  (a form of instability reduction), albeit at the expense of slower convergence. The convergence mechanism is illustrated in Fig. 6.5.3.

For another connection, let us consider two successive points  $x_k$  and  $x_{k+1}$  generated by the algorithm. The subdifferential of the function

$$f(x) + \frac{1}{2c_k} \|x - x_k\|^2$$

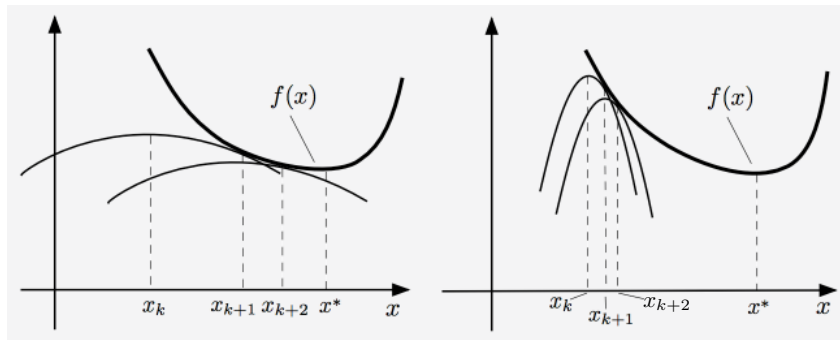
at  $x_{k+1}$  must contain 0 and is equal to

$$\partial f(x_{k+1}) + \frac{x_{k+1} - x_k}{c_k},$$

(cf. Prop. 5.4.6), so that

$$\frac{x_k - x_{k+1}}{c_k} \in \partial f(x_{k+1}). \quad (6.83)$$

Using this formula, we see that the move from  $x_k$  to  $x_{k+1}$  is “nearly” a subgradient step. In particular, while  $x_k - x_{k+1}$  is not a multiple of a vector in  $\partial f(x_k)$ , it is “close” to being one, if  $\partial f(x_k) \approx \partial f(x_{k+1})$ . We will make this more precise in Section 6.8, after we introduce a notion of “approximate subgradient.”



**Figure 6.5.3.** Illustration of the role of the parameter  $c_k$  in the convergence process of the proximal point algorithm. In the figure on the left,  $c_k$  is large, the graph of the quadratic term is “blunt,” and the method makes fast progress toward the optimal solution set  $X^*$ . In the figure on the right,  $c_k$  is small, the graph of the quadratic term is “pointed,” and the method makes slow progress.

### Convergence

The proximal point algorithm has excellent convergence properties, the most basic of which is the following.

**Proposition 6.5.1: (Convergence)** Let  $\{x_k\}$  be a sequence generated by the proximal point algorithm (6.82). Then, assuming that  $\sum_{k=0}^{\infty} c_k = \infty$ , we have

$$f(x_k) \downarrow f^*,$$

and if  $X^*$  is nonempty,  $\{x_k\}$  converges to some point in  $X^*$ .

**Proof:** We first note that since  $x_{k+1}$  minimizes  $f(x) + \frac{1}{2c_k}\|x - x_k\|^2$ , we have by setting  $x = x_k$ ,

$$f(x_{k+1}) + \frac{1}{2c_k}\|x_{k+1} - x_k\|^2 \leq f(x_k), \quad \forall k. \quad (6.84)$$

It follows that  $\{f(x_k)\}$  is monotonically nondecreasing. Hence  $f(x_k) \downarrow f_\infty$ , where  $f_\infty$  is either a scalar or  $-\infty$ , and satisfies  $f_\infty \geq f^*$ .

For any  $y \in \mathfrak{R}^n$ , we have

$$\begin{aligned} \|x_k - y\|^2 &= \|x_{k+1} - y + x_k - x_{k+1}\|^2 \\ &= \|x_{k+1} - y\|^2 + 2(x_{k+1} - y)'(x_k - x_{k+1}) + \|x_{k+1} - x_k\|^2, \end{aligned}$$

and from the subgradient relation (6.83),

$$f(x_{k+1}) + \frac{1}{c_k}(x_k - x_{k+1})'(y - x_{k+1}) \leq f(y).$$

Combining these two relations, we obtain

$$\begin{aligned} \|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2c_k(f(x_{k+1}) - f(y)) - \|x_{k+1} - x_k\|^2 \\ &\leq \|x_k - y\|^2 - 2c_k(f(x_{k+1}) - f(y)), \end{aligned} \quad (6.85)$$

and for any  $N \geq 0$ , by adding over  $k = 0, \dots, N$ , we have

$$\|x_{N+1} - y\|^2 + 2 \sum_{k=0}^N c_k (f(x_{k+1}) - f(y)) \leq \|x_0 - y\|^2, \quad \forall y \in \mathfrak{R}^n, N \geq 0,$$

so that

$$2 \sum_{k=0}^N c_k (f(x_{k+1}) - f(y)) \leq \|x_0 - y\|^2, \quad \forall y \in \mathfrak{R}^n, N \geq 0.$$

Taking the limit as  $N \rightarrow \infty$ , we have

$$2 \sum_{k=0}^{\infty} c_k (f(x_{k+1}) - f(y)) \leq \|x_0 - y\|^2, \quad \forall y \in \mathfrak{R}^n. \quad (6.86)$$

Assume to arrive at a contradiction that  $f_\infty > f^*$ , and let  $\hat{y}$  be such that  $f_\infty > f(\hat{y}) > f^*$ . Since  $\{f(x_k)\}$  is monotonically nondecreasing, we have

$$f(x_{k+1}) - f(\hat{y}) \geq f_\infty - f(\hat{y}) > 0.$$

Then in view of the assumption  $\sum_{k=0}^{\infty} c_k = \infty$ , Eq. (6.86) leads to a contradiction. Thus  $f_\infty = f^*$ .

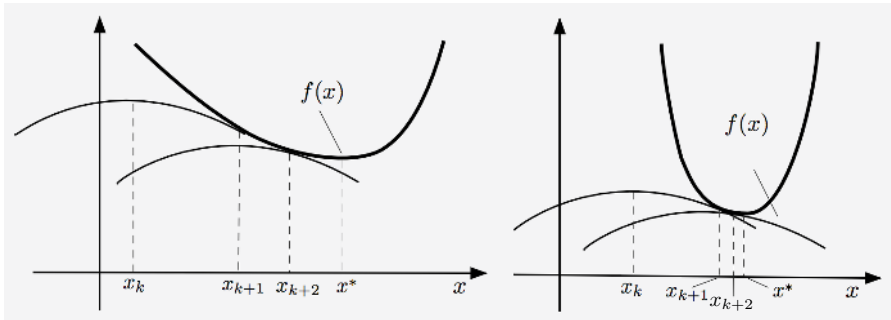
Consider now the case where  $X^*$  is nonempty, and let  $x^*$  be any point in  $X^*$ . Applying Eq. (6.85) with  $y = x^*$ , we have

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2c_k (f(x_{k+1}) - f(x^*)), \quad k = 0, 1, \dots \quad (6.87)$$

Thus  $\|x_k - x^*\|^2$  is monotonically nondecreasing, so  $\{x_k\}$  is bounded, and each of its limit points must belong to  $X^*$ , since  $\{f(x_k)\}$  monotonically decreases to  $f^*$  and  $f$  is closed. Also, by Eq. (6.87), the distance of  $x_k$  to each limit point is monotonically nondecreasing, so  $\{x_k\}$  must converge to a unique limit, which must be an element of  $X^*$ . **Q.E.D.**

### Rate of Convergence

The following proposition shows that the convergence rate of the algorithm depends on the magnitude of  $c_k$  and on the order of growth of  $f$  near the optimal solution set (see also Fig. 6.5.4).



**Figure 6.5.4.** Illustration of the role of the growth properties of  $f$  near  $X^*$  in the convergence rate of the proximal point algorithm. In the figure on the left,  $f$  grows slowly and the convergence is slow. In the figure on the right,  $f$  grows fast and the convergence is fast.

**Proposition 6.5.2: (Rate of Convergence)** Let  $\{x_k\}$  be a sequence generated by the proximal point algorithm (6.82), under the assumptions that  $\sum_{k=0}^{\infty} c_k = \infty$  and  $X^*$  is nonempty. Assume further that for some scalars  $\beta > 0$ ,  $\delta > 0$ , and  $\alpha \geq 1$ , we have

$$f^* + \beta(d(x))^\alpha \leq f(x), \quad \forall x \in \mathfrak{R}^n \text{ with } d(x) \leq \delta, \quad (6.88)$$

where

$$d(x) = \min_{x^* \in X^*} \|x - x^*\|.$$

(a) For all  $k$  sufficiently large, we have

$$d(x_{k+1}) + \beta c_k (d(x_{k+1}))^{\alpha-1} \leq d(x_k), \quad \text{if } x_{k+1} \notin X^*. \quad (6.89)$$

(b) Let  $1 < \alpha < 2$  and  $x_k \notin X^*$  for all  $k$ . Then if  $\inf_{k \geq 0} c_k > 0$ ,

$$\limsup_{k \rightarrow \infty} \frac{d(x_{k+1})}{(d(x_k))^{1/(\alpha-1)}} < \infty.$$

(c) Let  $\alpha = 2$  and  $x_k \notin X^*$  for all  $k$ . Then if  $\lim_{k \rightarrow \infty} c_k = \bar{c}$  with  $\bar{c} \in (0, \infty)$ ,

$$\limsup_{k \rightarrow \infty} \frac{d(x_{k+1})}{d(x_k)} \leq \frac{1}{1 + \beta \bar{c}},$$

while if  $\lim_{k \rightarrow \infty} c_k = \infty$ ,

$$\lim_{k \rightarrow \infty} \frac{d(x_{k+1})}{d(x_k)} = 0.$$

**Proof:** (a) Let  $\hat{x}_{k+1}$  denote the projection of  $x_{k+1}$  on  $X^*$ , let  $d = \hat{x}_{k+1} - x_{k+1}$ , and assume that  $d \neq 0$ . From the subgradient relation (6.83), we have

$$f(x_{k+1}) + \frac{1}{c_k} (x_k - x_{k+1})' (\hat{x}_{k+1} - x_{k+1}) \leq f(\hat{x}_{k+1}) = f^*.$$

Using the hypothesis,  $\{x_k\}$  converges to some point in  $X^*$ , so it follows that

$$\beta c_k (d(x_{k+1}))^\alpha \leq (x_{k+1} - x_k)' (\hat{x}_{k+1} - x_{k+1}),$$

for  $k$  sufficiently large. We add to both sides  $(x_{k+1} - \hat{x}_k)' (x_{k+1} - \hat{x}_{k+1})$ , yielding

$$(x_{k+1} - \hat{x}_k)' (x_{k+1} - \hat{x}_{k+1}) + \beta c_k (d(x_{k+1}))^\alpha \leq (x_k - \hat{x}_k)' (x_{k+1} - \hat{x}_{k+1}),$$

and we use the fact

$$\|x_{k+1} - \hat{x}_{k+1}\|^2 \leq (x_{k+1} - \hat{x}_k)'(x_{k+1} - \hat{x}_{k+1}),$$

which follows from the Projection Theorem (cf. Prop. 1.1.9), to obtain

$$\|x_{k+1} - \hat{x}_{k+1}\|^2 + \beta c_k (d(x_{k+1}))^\alpha \leq \|x_k - \hat{x}_k\| \|x_{k+1} - \hat{x}_{k+1}\|.$$

By dividing with  $\|x_{k+1} - \hat{x}_{k+1}\|$ , Eq. (6.89) follows.

(b) From Eq. (6.89) and the fact  $\alpha < 2$ , the desired relation follows.

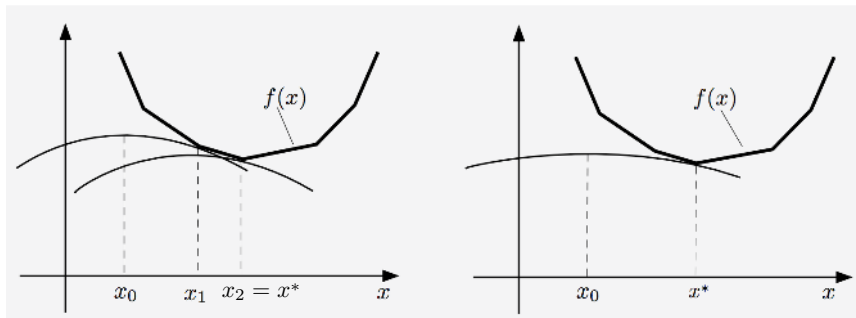
(c) For  $\alpha = 2$ , Eq. (6.89) becomes

$$(1 + \beta c_k) d(x_{k+1}) \leq d(x_k),$$

from which the result follows. **Q.E.D.**

Proposition 6.5.2 shows that as the growth order  $\alpha$  in Eq. (6.88) increases, the rate of convergence becomes slower. The threshold case is when  $\alpha = 2$ ; then the distance of the iterates to  $X^*$  decreases at least at the rate of a geometric progression if  $c_k$  remains bounded, and decreases even faster (superlinearly) if  $c_k \rightarrow \infty$ . Generally, the convergence is accelerated if  $c_k$  is increased with  $k$ , rather than kept constant; this is illustrated most clearly when  $\alpha = 2$  [cf. part (c) of Prop. 6.5.2]. When  $1 < \alpha < 2$ , the convergence rate is faster than that of a geometric progression (superlinear); see Prop. 6.5.2(b). When  $\alpha > 2$ , the convergence rate is slower than when  $\alpha = 2$  (sublinear); see Exercise 6.10.

The case where  $\alpha = 1$  allows for a cost function  $f$  that is polyhedral. Then the proximal point algorithm converges finitely (in fact in a single iteration for  $c_0$  sufficiently large), as illustrated in Fig. 6.5.5 and as shown in the following proposition.



**Figure 6.5.5.** Finite convergence of the proximal point algorithm when  $f(x)$  grows at a linear rate near the optimal solution set  $X^*$  (e.g.,  $f$  is polyhedral). In the figure on the right, we have convergence in a single iteration for a large enough value of  $c$ .

**Proposition 6.5.3: (Finite Convergence)** Assume that  $X^*$  is nonempty and that there exists a scalar  $\beta > 0$  such that

$$f^* + \beta d(x) \leq f(x), \quad \forall x \in \mathfrak{R}^n, \quad (6.90)$$

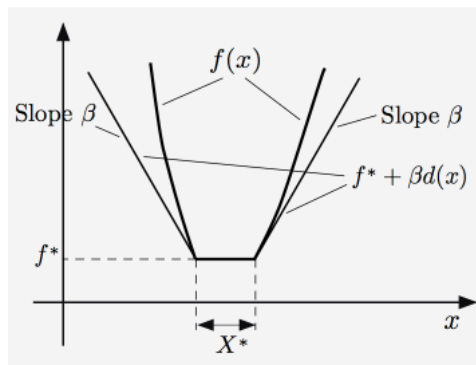
where  $d(x) = \min_{x^* \in X^*} \|x - x^*\|$ . Then if  $\sum_{k=0}^{\infty} c_k = \infty$ , the proximal point algorithm (6.82) converges to  $X^*$  finitely [that is, there exists  $\bar{k} > 0$  such that  $x_k \in X^*$  for all  $k \geq \bar{k}$ ]. Furthermore, if  $c_0 \geq d(x_0)/\beta$ , the algorithm converges in a single iteration (i.e.,  $x_1 \in X^*$ ).

**Proof:** The assumption (6.88) of Prop. 6.5.2 holds with  $\alpha = 1$  and all  $\delta > 0$ , so Eq. (6.89) becomes

$$d(x_{k+1}) + \beta c_k \leq d(x_k), \quad \text{if } x_{k+1} \notin X^*,$$

and holds for all  $k$ . If  $\sum_{k=0}^{\infty} c_k = \infty$  and  $x_k \notin X^*$  for all  $k$ , by adding the preceding inequality over all  $k$ , we obtain a contradiction. Hence we must have  $x_k \in X^*$  for  $k$  sufficiently large. Similarly, if  $c_0 \geq d(x_0)/\beta$ , we must have  $x_1 \in X^*$ . **Q.E.D.**

The condition (6.90) is illustrated in Fig. 6.5.6. It can be shown that the condition holds when  $f$  is a polyhedral function and  $X^*$  is nonempty (see Exercise 6.9).



**Figure 6.5.6.** Illustration of the condition

$$f^* + \beta d(x) \leq f(x), \quad \forall x \in \mathfrak{R}^n,$$

[cf. Eq. (6.90)].

It is also possible to prove the one-step convergence property of Prop. 6.5.3 with a simpler argument that does not rely on Prop. 6.5.2 and Eq.



(6.89). Indeed, assume that  $x_0 \neq X^*$ , let  $\hat{x}_0$  be the projection of  $x_0$  on  $X^*$ , and consider the function

$$\tilde{f}(x) = f^* + \beta d(x) + \frac{1}{2c_0} \|x - x_0\|^2. \quad (6.91)$$

Its subdifferential at  $\hat{x}_0$  is given by (cf. Prop. 5.4.6)

$$\begin{aligned} \partial \tilde{f}(\hat{x}_0) &= \left\{ \beta \gamma \frac{x_0 - \hat{x}_0}{\|x_0 - \hat{x}_0\|} + \frac{1}{c_0} (\hat{x}_0 - x_0) \mid \gamma \in [0, 1] \right\} \\ &= \left\{ \left( \frac{\beta \gamma}{d(x_0)} - \frac{1}{c_0} \right) (x_0 - \hat{x}_0) \mid \gamma \in [0, 1] \right\}. \end{aligned}$$

It follows that if  $c_0 \geq d(x_0)/\beta$ , then  $0 \in \partial \tilde{f}(\hat{x}_0)$ , so that  $\hat{x}_0$  minimizes  $\tilde{f}(x)$ . Since from Eqs. (6.90) and (6.91), we have

$$\tilde{f}(x) \leq f(x) + \frac{1}{2c_0} \|x - x_0\|^2, \quad \forall x \in \mathfrak{R}^n,$$

with equality when  $x = \hat{x}_0$ , it follows that  $\hat{x}_0$  minimizes  $f(x) + \frac{1}{2c_0} \|x - x_0\|^2$  over  $x \in X$ , and is therefore equal to the first iterate  $x_1$  of the proximal point algorithm.

### Gradient and Subgradient Interpretations

An interesting interpretation of the proximal point iteration is obtained by considering the function

$$\phi_c(z) = \inf_{x \in \mathfrak{R}^n} \left\{ f(x) + \frac{1}{2c} \|x - z\|^2 \right\} \quad (6.92)$$

for a fixed positive value of  $c$ . It is easily seen that

$$\inf_{x \in \mathfrak{R}^n} f(x) \leq \phi_c(z) \leq f(z), \quad \forall z \in \mathfrak{R}^n,$$

from which it follows that the set of minima of  $f$  and  $\phi_c$  coincide (this is also evident from the geometric view of the proximal minimization given in Fig. 6.5.7). The following proposition shows that  $\phi_c$  is a convex differentiable function, and calculates its gradient.

**Proposition 6.5.4:** The function  $\phi_c$  of Eq. (6.92) is convex and differentiable, and we have

$$\nabla \phi_c(z) = \frac{z - x_c(z)}{c} \quad \forall z \in \mathfrak{R}^n, \quad (6.93)$$

where  $x_c(z)$  is the unique minimizer in Eq. (6.92).

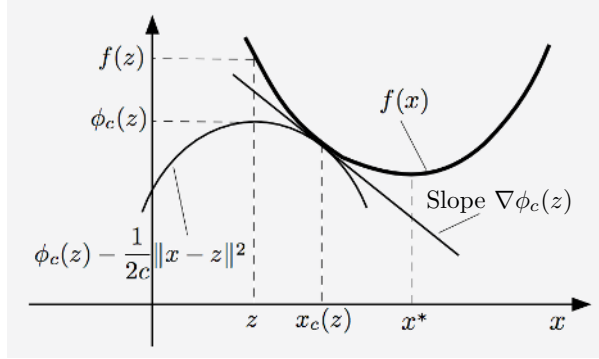


Figure 6.5.7. Illustration of the function

$$\phi_c(z) = \inf_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2c} \|x - z\|^2 \right\}.$$

We have  $\phi_c(z) \leq f(z)$  for all  $z \in \mathbb{R}^n$ , and at the set of minima of  $f$ ,  $\phi_c$  coincides with  $f$ . We also have

$$\nabla\phi_c(z) = \frac{z - x_c(z)}{c}.$$

For some geometric insight as to why this relation holds, consider the case where  $f$  is linear and note the definition of  $\phi_c$  in the figure.

**Proof:** We first note that  $\phi_c$  is convex, since it is obtained by partial minimization of  $f(x) + \frac{1}{2c}\|x - z\|^2$ , which is convex as a function of  $(x, z)$  (cf. Prop. 3.3.1). Furthermore,  $\phi_c$  is real-valued, since the infimum in Eq. (6.92) is attained.

Let us fix  $z$ , and for notational simplicity, denote  $\bar{x} = x_c(z)$ . To show that  $\phi_c$  is differentiable with the given form of gradient, we note that by the optimality condition of Prop. 5.4.7, we have  $y \in \partial\phi_c(z)$  if and only if  $z$  attains the minimum over  $v \in \mathbb{R}^n$  of

$$\phi_c(v) - y'v = \inf_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2c} \|x - v\|^2 \right\} - y'v.$$

Equivalently,  $y \in \partial\phi_c(z)$  if and only if  $(\bar{x}, z)$  attains the minimum over  $(x, v) \in \mathbb{R}^{2n}$  of the function

$$F(x, v) = f(x) + \frac{1}{2c} \|x - v\|^2 - y'v,$$

which is equivalent to  $(0, 0) \in \partial F(\bar{x}, z)$ , or

$$0 \in \partial f(\bar{x}) + \frac{\bar{x} - z}{c}, \quad y = \frac{z - \bar{x}}{c}. \tag{6.94}$$

[This last step is obtained by viewing  $F$  as the sum of the function  $f$  and the differentiable function

$$\frac{1}{2c}\|x - v\|^2 - y'v,$$

and by writing

$$\partial F(x, v) = \{(g, 0) \mid g \in \partial f(x)\} + \left\{ \frac{x - v}{c}, \frac{v - x}{c} - y \right\};$$

cf. Prop. 5.4.6]. Since the subgradient relation on the left side of Eq. (6.94) is satisfied by the definition of  $\bar{x}$ , we see that  $\phi_c$  has a unique subgradient at  $z$  with the given form. **Q.E.D.**

Using the gradient formula (6.93), we see that the proximal point iteration can be written as

$$x_{k+1} = x_k - c_k \nabla \phi_{c_k}(x_k),$$

so it is a gradient iteration for minimizing  $\phi_{c_k}$ . This interpretation provides insight into the working mechanism of the algorithm and has formed the basis for various acceleration schemes, particularly in connection with the Augmented Lagrangian method, a popular constrained minimization method to be discussed in Section 6.6.2 (see also the book [Ber82] and the references quoted there).

### 6.5.2 Proximal Cutting Plane Method

Let us consider minimization of a real-valued convex function  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$ , over a closed convex set  $X$ , by using the proximal point algorithm. Since  $f$  may not be differentiable, it is natural to try polyhedral approximation ideas for minimizing

$$f(x) + \frac{1}{2c_k}\|x - x_k\|^2$$

over  $X$  (assuming of course that at each  $x \in X$ , a subgradient of  $f$  can be computed). In particular, we may consider replacing the original function  $f$  with a simpler polyhedral approximation  $F_k$ , thereby simplifying the corresponding proximal minimization. A special advantage of this idea is that once a cutting plane has been constructed at some iteration, it can be used for approximation of  $f$  at all iterations of the current or the subsequent proximal minimizations. Thus one may perform the proximal minimizations approximately, and update  $x_k$  after any number of cutting plane iterations, while carrying over the computed cutting planes from one proximal minimization to the next. An extreme form of implementation

of this idea is to update  $x_k$  after a single cutting plane iteration, as in the following algorithm.

At the typical iteration, we perform a proximal point iteration, aimed at minimizing the current polyhedral approximation to  $f$  given by [cf. Eq. (6.46)]

$$F_k(x) = \max\{f(x_0) + (x - x_0)'g_0, \dots, f(x_k) + (x - x_k)'g_k\}, \quad (6.95)$$

i.e.,

$$x_{k+1} \in \arg \min_{x \in X} \left\{ F_k(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}, \quad (6.96)$$

where  $c_k$  is a positive scalar parameter. A subgradient  $g_{k+1}$  of  $f$  at  $x_{k+1}$  is then computed,  $F_{k+1}$  is accordingly updated, and the process is repeated. We call this the *proximal cutting plane method*.

The method terminates if  $x_{k+1} = x_k$ ; in this case, Eqs. (6.95) and (6.96) imply that

$$f(x_k) = F_k(x_k) \leq F_k(x) + \frac{1}{2c_k} \|x - x_k\|^2 \leq f(x) + \frac{1}{2c_k} \|x - x_k\|^2, \quad \forall x \in X,$$

so  $x_k$  is a point where the proximal point algorithm terminates, and it must therefore be optimal by Prop. 6.5.1. Note, however, that unless  $f$  and  $X$  are polyhedral, finite termination is unlikely.

The convergence properties of the method are easy to derive, based on what we already know. The idea is that  $F_k$  asymptotically converges to  $f$ , at least near the generated iterates, so asymptotically, the algorithm essentially becomes the proximal point algorithm, and inherits the corresponding convergence properties. Let us derive a finite convergence result for the polyhedral case.

**Proposition 6.5.5: (Finite Termination of the Proximal Cutting Plane Method)** Consider the proximal cutting plane method for the case where  $f$  and  $X$  are polyhedral, with

$$f(x) = \max_{i \in I} \{a_i'x + b_i\},$$

where  $I$  is a finite index set, and  $a_i$  and  $b_i$  are given vectors and scalars, respectively. Assume that the optimal solution set is nonempty and that the subgradient added to the cutting plane approximation at each iteration is one of the vectors  $a_i$ ,  $i \in I$ . Then the method terminates finitely with an optimal solution.

**Proof:** Since there are only finitely many vectors  $a_i$  to add, eventually the polyhedral approximation  $F_k$  will not change, i.e.,  $F_k = F_{\bar{k}}$  for all  $k > \bar{k}$ .

Thus, for  $k \geq \bar{k}$ , the method will become the proximal point algorithm for minimizing  $F_{\bar{k}}$ , so by Prop. 6.5.3, it will terminate with a point  $\bar{x}$  that minimizes  $F_{\bar{k}}$  subject to  $x \in X$ . But then, we will have concluded an iteration of the cutting plane method for minimizing  $f$  over  $X$ , with no new vector added to the approximation  $F_k$ , which implies termination of the cutting plane method, necessarily at a minimum of  $f$  over  $X$ . **Q.E.D.**

The proximal cutting plane method aims at increased stability over the ordinary cutting plane method, but it has some drawbacks:

- (a) There is a potentially difficult tradeoff in the choice of the parameter  $c_k$ . In particular, stability is achieved only by choosing  $c_k$  small, since for large values of  $c_k$  the changes  $x_{k+1} - x_k$  may be substantial. Indeed for a polyhedral function  $f$  and large enough  $c_k$ , the method finds the exact minimum of  $F_k$  over  $X$  in a single minimization (cf. Prop. 6.5.3), so it is identical to the ordinary cutting plane method, and fails to provide any stabilization. On the other hand, small values of  $c_k$  lead to slow rate of convergence.
- (b) The number of subgradients used in the approximation  $F_k$  may grow to be very large, in which case the quadratic program solved in Eq. (6.96) may become very time-consuming.

These drawbacks motivate algorithmic variants, called *bundle methods*, which we will discuss next. The main difference is that the proximal center  $x_k$  is updated only after making enough progress in minimizing  $f$  to ensure a certain measure of stability.

### 6.5.3 Bundle Methods

In the basic form of a bundle method, the iterate  $x_{k+1}$  is obtained by minimizing over  $X$  the sum of  $F_k$ , a cutting plane approximation to  $f$ , and a quadratic proximal term  $p_k(x)$ :

$$x_{k+1} \in \arg \min_{x \in X} \{F_k(x) + p_k(x)\}. \quad (6.97)$$

The proximal center of  $p_k$  need not be  $x_k$  (as in the proximal cutting plane method), but is rather one of the past iterates  $x_i$ ,  $i \leq k$ .

In one version of the method,  $F_k$  is given by

$$F_k(x) = \max\{f(x_0) + (x - x_0)'g_0, \dots, f(x_k) + (x - x_k)'g_k\}, \quad (6.98)$$

while  $p_k(x)$  is of the form

$$p_k(x) = \frac{1}{2c_k} \|x - y_k\|^2,$$

where  $y_k \in \{x_i \mid i \leq k\}$ . Following the computation of  $x_{k+1}$ , the new proximal center  $y_{k+1}$  is set to  $x_{k+1}$ , or is left unchanged ( $y_{k+1} = y_k$ )

depending on whether, according to a certain test, “sufficient progress” has been made or not. An example of such a test is

$$f(y_k) - f(x_{k+1}) \geq \beta \delta_k,$$

where  $\beta$  is a fixed scalar with  $\beta \in (0, 1)$ , and

$$\delta_k = f(y_k) - (F_k(x_{k+1}) + p_k(x_{k+1})).$$

Thus,

$$y_{k+1} = \begin{cases} x_{k+1} & \text{if } f(y_k) - f(x_{k+1}) \geq \beta \delta_k, \\ y_k & \text{if } f(y_k) - f(x_{k+1}) < \beta \delta_k, \end{cases} \quad (6.99)$$

and initially  $y_0 = x_0$ . In the parlance of bundle methods, iterations where  $y_{k+1}$  is updated to  $x_{k+1}$  are called *serious steps*, while iterations where  $y_{k+1} = y_k$  are called *null steps*.

The method terminates if  $x_{k+1} = y_k$ ; in this case, Eqs. (6.97) and (6.98) imply that

$$f(y_k) + p_k(y_k) = F_k(y_k) + p_k(y_k) \leq F_k(x) + p_k(x) \leq f(x) + p_k(x), \quad \forall x \in X,$$

so  $y_k$  is a point where the proximal point algorithm terminates, and must therefore be optimal. Of course, finite termination is unlikely, unless  $f$  and  $X$  are polyhedral. An important point, however, is that prior to termination, we have  $\delta_k > 0$ . Indeed, since

$$F_k(x_{k+1}) + p_k(x_{k+1}) \leq F_k(y_k) + p_k(y_k) = F_k(y_k),$$

and  $F_k(y_k) = f(y_k)$ , we have

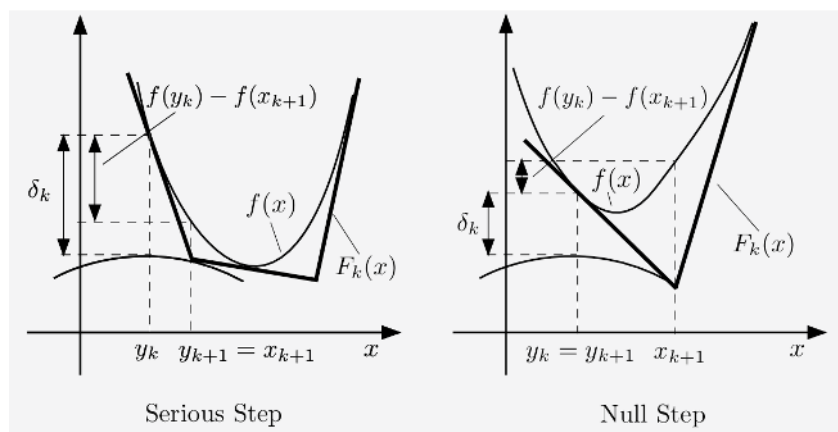
$$0 \leq f(y_k) - (F_k(x_{k+1}) + p_k(x_{k+1})) = \delta_k,$$

with equality only if  $x_{k+1} = y_k$ , i.e., when the algorithm terminates.

The scalar  $\delta_k$  is illustrated in Fig. 6.5.8. Since  $f(y_k) = F_k(y_k)$  [cf. Eq. (6.98)],  $\delta_k$  represents the reduction in the proximal objective  $F_k + p_k$  in moving from  $y_k$  to  $x_{k+1}$ . If the reduction in the true objective,

$$f(y_k) - f(x_{k+1}),$$

does not exceed a fraction  $\beta$  of  $\delta_k$  (or is even negative as in the right-hand side of Fig. 6.5.8), this indicates a large discrepancy between proximal and true objective, and an associated instability. As a result the algorithm foregoes the move from  $y_k$  to  $x_{k+1}$  with a null step [cf. Eq. (6.99)], but improves the cutting plane approximation by adding the new plane corresponding



**Figure 6.5.8.** Illustration of the test (6.99) for a serious or a null step in the bundle method. It is based on

$$\delta_k = f(y_k) - (F_k(x_{k+1}) + p_k(x_{k+1})),$$

the reduction in proximal cost, which is always positive, except at termination. A serious step is performed if and only if the reduction in true cost,  $f(y_k) - f(x_{k+1})$ , exceeds a fraction  $\beta$  of the reduction  $\delta_k$  in proximal cost.

to  $x_{k+1}$ . Otherwise, it performs a serious step, with the guarantee of true cost improvement afforded by the test (6.99).

The convergence analysis of the bundle method just presented follows the corresponding arguments for the cutting plane and the proximal point method. The idea is that the method makes “substantial” progress with every serious step. Furthermore, null steps cannot be performed indefinitely, for in this case, the polyhedral approximation to  $f$  will become increasingly accurate and the reduction in true cost will converge to the reduction in proximal cost. Then, since  $\beta < 1$ , the test for a serious step will be passed. In the case where  $f$  and  $X$  are polyhedral, the method converges finitely, similar to the case of the proximal point and proximal cutting plane algorithms (cf. Props. 6.5.3 and 6.5.5).

**Proposition 6.5.6: (Finite Termination of the Bundle Method)**

Consider the bundle method for the case where  $f$  and  $X$  are polyhedral, with

$$f(x) = \max_{i \in I} \{a'_i x + b_i\},$$

where  $I$  is a finite index set, and  $a_i$  and  $b_i$  are given vectors and scalars, respectively. Assume that the optimal solution set is nonempty and that the subgradient added to the cutting plane approximation at each iteration is one of the vectors  $a_i$ ,  $i \in I$ . Then the method terminates finitely with an optimal solution.

**Proof:** Since there are only finitely many vectors  $\alpha_i$  to add, eventually the polyhedral approximation  $F_k$  will not change, i.e.,  $F_k = F_{\bar{k}}$  for all  $k > \bar{k}$ . We note that  $F_k(x_{k+1}) = f(x_{k+1})$  for all  $k > \bar{k}$ , since otherwise a new cutting plane would be added to  $F_k$ . Thus, for  $k > \bar{k}$ ,

$$\begin{aligned} f(y_k) - f(x_{k+1}) &= f(y_k) - F_k(x_{k+1}) \\ &= f(y_k) - (F_k(x_{k+1}) + p_k(x_{k+1})) + p_k(x_{k+1}) \\ &= \delta_k + p_k(x_{k+1}) \\ &\geq \beta\delta_k. \end{aligned}$$

Therefore, according to Eq. (6.99), the method will perform serious steps for all  $k > \bar{k}$ , and become identical to the proximal cutting plane algorithm, which converges finitely by Prop. 6.5.5. **Q.E.D.**

### Discarding Old Subgradients

We mentioned earlier that one of the drawbacks of the cutting plane algorithms is that the number of subgradients used in the approximation  $F_k$  may grow to be very large. The monitoring of progress through the test (6.99) for serious/null steps can also be used to discard some of the accumulated cutting planes. For example, at the end of a serious step, upon updating the proximal center  $y_k$  to  $y_{k+1} = x_{k+1}$ , we may discard any subset of the cutting planes.

It may of course be useful to retain some of the cutting planes, particularly the ones that are “active” or “nearly active” at  $y_{k+1}$ , i.e., those  $i \leq k$  for which the linearization error

$$F_k(y_{k+1}) - (f(x_i) + (y_{k+1} - x_i)'g_i)$$

is 0 or close to 0, respectively. The essential validity of the method is maintained, by virtue of the fact that  $\{f(y_k)\}$  is a monotonically decreasing sequence, with “sufficiently large” cost reductions between proximal center updates.

An extreme possibility is to discard all past subgradients following a serious step from  $y_k$  to  $x_{k+1}$ . Then, after a subgradient  $g_{k+1}$  at  $x_{k+1}$  is calculated, the next iteration becomes

$$x_{k+2} = \arg \min_{x \in X} \left\{ f(x_{k+1}) + g'_{k+1}(x - x_{k+1}) + \frac{1}{2c_{k+1}} \|x - x_{k+1}\|^2 \right\}.$$



It can be seen that we have

$$x_{k+2} = P_X(x_{k+1} - c_{k+1}g_{k+1}),$$

where  $P_X(\cdot)$  denotes projection on  $X$ , so after discarding all past subgradients following a serious step, the next iteration is an ordinary subgradient iteration with stepsize equal to  $c_{k+1}$ .

Another possibility is (following the serious step) to replace all the cutting planes with a single cutting plane: the one obtained from the hyperplane that passes through  $(x_{k+1}, F_k(x_{k+1}))$  and separates the epigraphs of the functions  $F_k(x)$  and  $\gamma_k - \frac{1}{2c_k}(x - y_k)$ , where

$$\gamma_k = F_k(x_{k+1}) + \frac{1}{2c_k}(x - y_k),$$

(see Fig. 6.5.9). This is the cutting plane

$$F_k(x_{k+1}) + \hat{g}'_k(x - x_{k+1}), \quad (6.100)$$

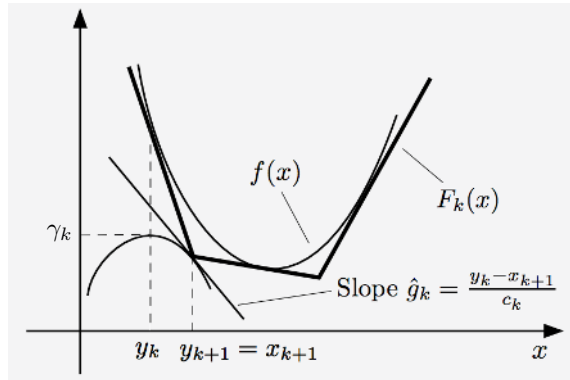
where  $\hat{g}_k$  is given by

$$\hat{g}_k = \frac{y_k - x_{k+1}}{c_k}. \quad (6.101)$$

The next iteration will then be performed with just two cutting planes: the one just given in Eq. (6.100) and a new one obtained from  $x_{k+1}$ ,

$$f(x_{k+1}) + g'_{k+1}(x - x_{k+1}),$$

where  $g_{k+1} \in \partial f(x_{k+1})$ .



**Figure 6.5.9.** Illustration of the cutting plane

$$F_k(x_{k+1}) + \hat{g}'_k(x - x_{k+1}),$$

where

$$\hat{g}_k = \frac{y_k - x_{k+1}}{c_k}.$$

The “slope”  $\hat{g}_k$  can be shown to be a convex combination of the subgradients that are “active” at  $x_{k+1}$ .

The vector  $\hat{g}_k$  is sometimes called an “aggregate subgradient,” because it can be shown to be a convex combination of the past subgradients  $g_0, \dots, g_k$ . This is evident from Fig. 6.5.9, and can also be verified by using quadratic programming duality arguments (see Exercise 6.18).

There are also many other variants of bundle methods, which aim at increased efficiency and the exploitation of special structure. We refer to the literature for related algorithms and their analyses.

## 6.6 DUAL PROXIMAL POINT ALGORITHMS

Let us recall the proximal point algorithm of Section 6.5.1:

$$x_{k+1} = \arg \min_{x \in \mathfrak{R}^n} \left\{ f(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}, \quad (6.102)$$

where  $f : \mathfrak{R}^n \mapsto (-\infty, \infty]$ ,  $x_0$  is an arbitrary starting point, and  $\{c_k\}$  is a positive scalar parameter sequence with  $\inf_{k \geq 0} c_k > 0$ . We will develop a dual implementation of the algorithm and its cutting plane/bundle versions, which parallels the duality between the simplicial decomposition and cutting plane methods, developed in Section 6.4.

We first note that the minimization above is in a form suitable for application of Fenchel duality (cf. Section 5.3.5) with the identifications

$$f_1(x) = f(x), \quad f_2(x) = \frac{1}{2c_k} \|x - x_k\|^2.$$

We can write the Fenchel dual problem as

$$\begin{aligned} &\text{minimize} && f_1^*(\lambda) + f_2^*(-\lambda) \\ &\text{subject to} && \lambda \in \mathfrak{R}^n, \end{aligned} \quad (6.103)$$

where  $f_1^*$  and  $f_2^*$  are the conjugate functions of  $f_1$  and  $f_2$ , respectively. We have

$$f_2^*(\lambda) = \sup_{x \in \mathfrak{R}^n} \{x' \lambda - f_2(x)\} = \sup_{x \in \mathfrak{R}^n} \left\{ x' \lambda - \frac{1}{2c_k} \|x - x_k\|^2 \right\} = x'_k \lambda + \frac{c_k}{2} \|\lambda\|^2,$$

where the last equality follows by noting that the supremum over  $x$  is attained at  $x = x_k + c_k \lambda$ . Denoting by  $f^*$  the conjugate of  $f$ ,

$$f_1^*(\lambda) = f^*(\lambda) = \sup_{x \in \mathfrak{R}^n} \{x' \lambda - f(x)\},$$

and substituting into Eq. (6.103), we see that the dual problem (6.103) can be written as

$$\begin{aligned} &\text{minimize} && f^*(\lambda) - x'_k \lambda + \frac{c_k}{2} \|\lambda\|^2 \\ &\text{subject to} && \lambda \in \mathfrak{R}^n. \end{aligned} \quad (6.104)$$

We now note that since  $f_2$  is real-valued, the relative interior condition of the Fenchel Duality Theorem [Prop. 5.3.8(a)] is satisfied, so there is no duality gap. Furthermore, both primal and dual problems have a unique solution, since they involve a closed, strictly convex, and coercive cost function.

Let  $\lambda_{k+1}$  be the unique optimal solution of problem (6.104). Then  $\lambda_{k+1}$  together with  $x_{k+1}$  satisfy the Lagrangian optimality conditions of Prop. 5.3.8. When applied to the primal problem, these conditions can be written as

$$x_{k+1} \in \arg \max_{x \in \mathfrak{R}^n} \{x' \lambda_{k+1} - f(x)\},$$

$$x_{k+1} \in \arg \min_{x \in \mathfrak{R}^n} \{x' \lambda_{k+1} - f_2(x)\} = \arg \min_{x \in \mathfrak{R}^n} \left\{ x' \lambda_{k+1} + \frac{1}{2c_k} \|x - x_k\|^2 \right\},$$

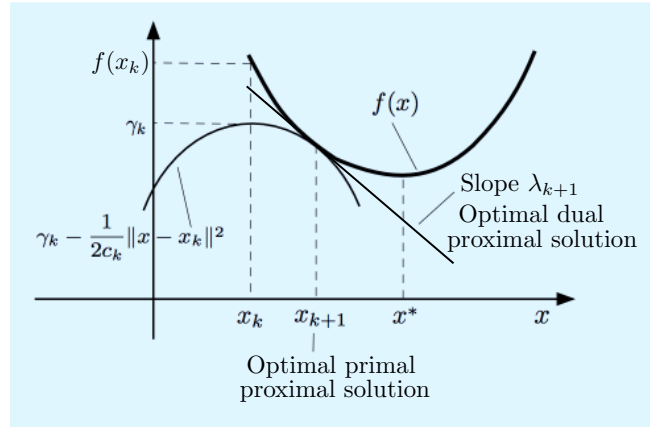
or equivalently,

$$\lambda_{k+1} \in \partial f(x_{k+1}), \quad x_{k+1} = x_k - c_k \lambda_{k+1}. \quad (6.105)$$

Similarly, when applied to the dual problem, the Lagrangian optimality condition of Prop. 5.3.8(b) can be written as

$$x_{k+1} \in \partial f^*(\lambda_{k+1}), \quad \lambda_{k+1} = \frac{x_k - x_{k+1}}{c_k}; \quad (6.106)$$

(cf. Fig. 6.6.1).



**Figure 6.6.1.** Visualization of the proximal iteration (6.102) and the solution  $\lambda_{k+1}$  of the corresponding dual problem (6.104) in terms of Fenchel duality [cf. Eq. (6.105)].

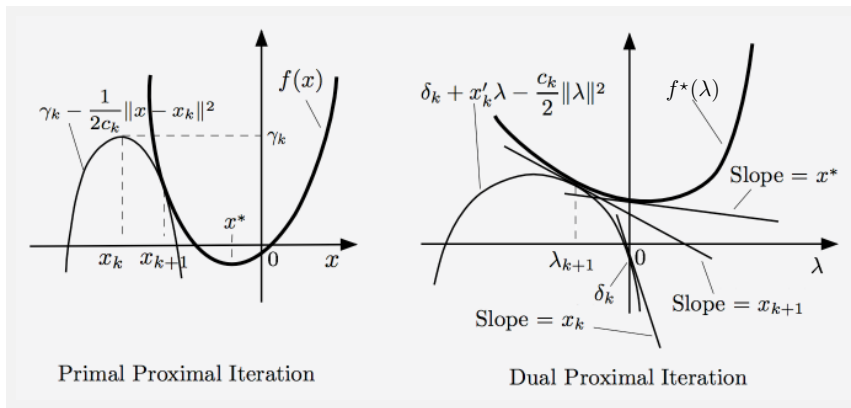
We thus obtain a dual (and mathematically equivalent) implementation of the proximal point algorithm. It is given by the iteration

$$\lambda_{k+1} = \arg \min_{\lambda \in \mathbb{R}^n} \left\{ f^*(\lambda) - x'_k \lambda + \frac{c_k}{2} \|\lambda\|^2 \right\}, \quad (6.107)$$

followed by

$$x_{k+1} = x_k - c_k \lambda_{k+1}, \quad (6.108)$$

(see Fig. 6.6.2). As  $x_k$  converges to a minimum  $x^*$  of  $f$ ,  $\lambda_k$  converges to 0. Thus the dual algorithm does not aim to minimize  $f^*$ , but rather to find a subgradient of  $f^*$  at 0, which [by Prop. 5.4.4(b)] is a minimum of  $f$ . In particular, we have  $\lambda_k \in \partial f(x_k)$ ,  $x_k \in \partial f^*(\lambda_k)$  [cf. Eqs. (6.105) and (6.106)], and as  $\lambda_k$  converges to 0 and  $x_k$  converges to a minimum  $x^*$  of  $f$ , we have  $0 \in \partial f(x^*)$  and  $x^* \in \partial f^*(0)$ .



**Figure 6.6.2.** Illustration of primal and dual proximal point algorithms. The primal algorithm aims to find  $x^*$ , a minimum of  $f$ . The dual algorithm aims to find  $x^*$  as a subgradient of  $f^*$  at 0, i.e., it aims to solve the (generalized) equation  $x^* \in \partial f^*(0)$  [cf. Prop. 5.4.4(b)].

The primal and dual implementations are mathematically equivalent and generate identical sequences  $\{x_k\}$ , assuming the same starting point  $x_0$ . Whether one is preferable over the other depends on which of the minimizations (6.102) and (6.107) is easier, i.e., whether  $f$  or its conjugate  $f^*$  has more convenient structure.

We will now discuss two applications of the dual proximal point algorithm. The first application leads to a popular general purpose algorithm for constrained minimization, the Augmented Lagrangian method. The second application is a dual implementation of the proximal cutting plane and bundle methods of Section 6.5.

### 6.6.1 Augmented Lagrangian Methods

We will now discuss a special case of the dual proximal point algorithm, where  $-f$  is the dual function  $q$  of a constrained optimization problem. In this case, because  $q$  is typically not available in closed form, the dual proximal approach is more convenient than the primal. The resulting methods are popular because they allow the solution of constrained optimization problems, through a sequence of easier unconstrained (or less constrained) optimizations, which can be performed with fast and reliable algorithms, such as Newton, Quasi-Newton, and conjugate gradient methods (see the literature cited at the end of the chapter).

Consider the constrained minimization problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && x \in X, \quad Ex = d, \end{aligned} \tag{6.109}$$

where  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  is a convex function,  $X$  is a closed convex set,  $E$  is an  $m \times n$  matrix, and  $d \in \mathfrak{R}^m$ .<sup>†</sup>

Consider also the corresponding primal and dual functions

$$p(v) = \inf_{x \in X, Ex=d+v} f(x), \quad q(\lambda) = \inf_{x \in X} L(x, \lambda),$$

where  $L(x, \lambda) = f(x) + \lambda'(Ex - d)$  is the Lagrangian function. We assume that  $p$  is closed, so that, except for sign changes,  $q$  and  $p$  are conjugates of each other [i.e.,  $-q(-\lambda)$  is the conjugate convex of  $p$ ; cf. Section 4.2.1].

Let us apply the proximal point algorithm to the dual problem of maximizing  $q$ . It has the form

$$\lambda_{k+1} = \arg \max_{\mu \in \mathfrak{R}^m} \left\{ q(\mu) - \frac{1}{2c_k} \|\mu - \lambda_k\|^2 \right\}.$$

In view of the conjugacy relation between  $q$  and  $p$  (taking also into account the required sign changes), it can be seen that the dual proximal point algorithm has the form

$$v_{k+1} = \arg \min_{v \in \mathfrak{R}^m} \left\{ p(v) + \lambda_k' v + \frac{c_k}{2} \|v\|^2 \right\}; \tag{6.110}$$

---

<sup>†</sup> We focus on linear equality constraints for convenience, but the analysis can be extended to convex inequality constraints as well. In particular, a linear inequality constraint of the form  $a_j'x \leq b_j$  can be converted to an equality constraint  $a_j'x + z_j = b_j$  by using a slack variable constraint  $z_j \geq 0$ , which can be absorbed into the set  $X$ . For convex inequality constraints, see Section 4.2.4 and the sources given at the end of the chapter. The book [Ber82] is a comprehensive reference on Augmented Lagrangian methods.

see Fig. 6.6.3. To implement this algorithm, we use the definition of  $p$  to write the above minimization as

$$\begin{aligned}
 & \min_{v \in \mathbb{R}^m} \left\{ \inf_{x \in X, Ex-d=v} \{f(x)\} + \lambda_k' v + \frac{c_k}{2} \|v\|^2 \right\} \\
 &= \min_{v \in \mathbb{R}^m} \inf_{x \in X, Ex-d=v} \left\{ f(x) + \lambda'(Ex-d) + \frac{c}{2} \|Ex-d\|^2 \right\} \quad (6.111) \\
 &= \inf_{x \in X} \left\{ f(x) + \lambda'(Ex-d) + \frac{c}{2} \|Ex-d\|^2 \right\} \\
 &= \inf_{x \in X} L_{c_k}(x, \lambda_k),
 \end{aligned}$$

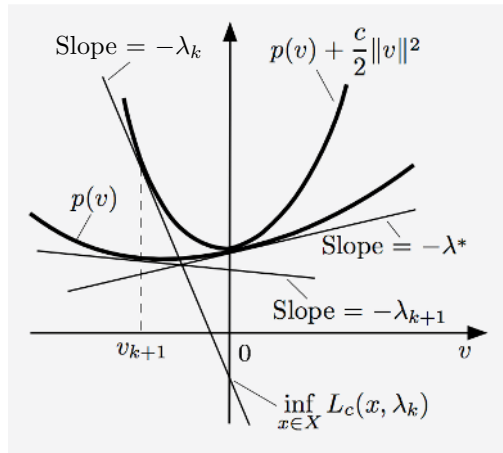
where for any  $c > 0$ ,  $L_c$  is the Augmented Lagrangian function

$$L_c(x, \lambda) = f(x) + \lambda'(Ex-d) + \frac{c}{2} \|Ex-d\|^2.$$

Furthermore, the minimizing  $v$  and  $x$  in Eq. (6.111) are related, and we have

$$v_{k+1} = Ex_{k+1} - d,$$

where  $x_{k+1}$  is any vector that minimizes  $L_{c_k}(x, \lambda_k)$  over  $X$  (we assume that such a vector exists - this is not guaranteed, and must be verified independently).



**Figure 6.6.3.** Illustration of the dual proximal minimization (6.110) and the update

$$\lambda_{k+1} = \lambda_k + c_k v_{k+1}$$

in the Augmented Lagrangian method. We have  $-\lambda_{k+1} \in \partial p(v_{k+1})$  based on the dual Lagrangian optimality conditions [cf. Eq. (6.106)].

Thus, the iteration of the dual algorithm [cf. Eq. (6.108), with a change of sign of  $\lambda_k$  inherited from the change of sign in Eq. (6.110)] takes the form  $\lambda_{k+1} = \lambda_k + c_k v_{k+1}$ , or

$$\lambda_{k+1} = \lambda_k + c_k(Ex_{k+1} - d), \quad (6.112)$$

where

$$x_{k+1} \in \arg \min_{x \in X} L_{c_k}(x, \lambda_k). \quad (6.113)$$

The algorithm (6.112)-(6.113) is known as the *Augmented Lagrangian method* or the *method of multipliers*. As we have seen, it is the special case of the dual proximal point algorithm applied to maximization of the dual function  $q$ .

The convergence properties of the Augmented Lagrangian method are derived from the corresponding properties of the proximal point algorithm (cf. Section 6.5.1). The sequence  $\{q(\lambda_k)\}$  converges to the optimal dual value, and  $\{\lambda_k\}$  converges to an optimal dual solution, provided such a solution exists (cf. Prop. 6.5.1). Furthermore, convergence in a finite number of iterations is obtained in the case of a linear programming problem (cf. Prop. 6.5.3).

We also claim that every limit point of the generated sequence  $\{x_k\}$  is an optimal solution of the primal problem (6.109). To see this, note that since  $\{\lambda_k\}$  converges to an optimal dual solution, from the update formula (6.112) we obtain

$$Ex_{k+1} - d \rightarrow 0, \quad c_k(Ex_{k+1} - d) \rightarrow 0.$$

Furthermore, we have

$$L_{c_k}(x_{k+1}, \lambda_k) = \min_{x \in X} \left\{ f(x) + \lambda_k'(Ex - d) + \frac{c_k}{2} \|Ex - d\|^2 \right\}.$$

The last two relations yield

$$\limsup_{t \rightarrow \infty} f(x_{k+1}) = \limsup_{t \rightarrow \infty} L_{c_k}(x_{k+1}, \lambda_k) \leq f(x), \quad \forall x \in X \text{ with } Ex = d,$$

so if  $x^* \in X$  is a limit point of  $\{x_k\}$ , we obtain

$$f(x^*) \leq f(x), \quad \forall x \in X \text{ with } Ex = d,$$

as well as  $Ex^* = d$  (in view of  $Ex_{k+1} - d \rightarrow 0$ ). Therefore any limit point  $x^*$  of the generated sequence  $\{x_k\}$  is an optimal solution of the primal problem (6.109).

Finally, let us consider the ‘‘penalized’’ dual function  $q_c$ , given by

$$q_c(\lambda) = \max_{y \in \mathbb{R}^m} \left\{ q(y) - \frac{1}{2c} \|y - \lambda\|^2 \right\}. \quad (6.114)$$

Then, according to Prop. 6.5.4,  $q_c$  is differentiable, and we have

$$\nabla q_c(\lambda) = \frac{y_c(\lambda) - \lambda}{c},$$

where  $y_c(\lambda)$  is the unique vector attaining the maximum in Eq. (6.114). Furthermore, the multiplier iteration (6.112) can be written as a gradient iteration:

$$\lambda_{k+1} = \lambda_k + c_k \nabla q_{c_k}(\lambda_k).$$

This interpretation motivates variations based on faster Newton or Quasi-Newton methods for maximizing  $q_c$ . There are many algorithms along this line, some of which involve inexact minimization of the Augmented Lagrangian to enhance computational efficiency. We refer to the literature cited at the end of the chapter for analysis of such methods.

### 6.6.2 Proximal Inner Linearization Methods

Let us recall the proximal cutting plane method (cf. Section 6.4.2) applied to minimizing a real-valued convex function  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$ , over a closed convex set  $X$ . The typical iteration involves a proximal minimization of the current cutting plane approximation to  $f$  given by

$$F_k(x) = \max\{f(x_0) + (x - x_0)'g_0, \dots, f(x_k) + (x - x_k)'g_k\} + \delta_X(x),$$

where  $g_i \in \partial f(x_i)$  for all  $i$  and  $\delta_X$  is the indicator function of  $X$ . Thus,

$$x_{k+1} \in \arg \min_{x \in \mathfrak{R}^n} \left\{ F_k(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}, \quad (6.115)$$

where  $c_k$  is a positive scalar parameter. A subgradient  $g_{k+1}$  of  $f$  at  $x_{k+1}$  is then computed,  $F_{k+1}$  is accordingly updated, and the process is repeated.

Similar to the earlier discussion, we may use Fenchel duality to implement the proximal minimization (6.115) in terms of conjugate functions [cf. Eq. (6.107)]. The dual to the proximal iteration (6.115) can be written as

$$\begin{aligned} &\text{minimize} && F_k^*(\lambda) - x'_k \lambda + \frac{c_k}{2} \|\lambda\|^2 \\ &\text{subject to} && \lambda \in \mathfrak{R}^n, \end{aligned} \quad (6.116)$$

where  $F_k^*$  is the conjugate of  $F_k$ . Once  $\lambda_{k+1}$ , the unique minimizer in the dual proximal iteration (6.116), is computed,  $x_k$  is updated via

$$x_{k+1} = x_k - c_k \lambda_{k+1}$$

[cf. Eq. (6.108)]. Then, a subgradient  $g_{k+1}$  of  $f$  at  $x_{k+1}$  is obtained either directly, or as a vector attaining the supremum in the conjugacy relation

$$f(x_{k+1}) = \sup_{\lambda \in \mathfrak{R}^n} \{x'_{k+1} \lambda - f^*(\lambda)\},$$



where  $f^*$  is the conjugate function of  $f$ :

$$g_{k+1} \in \arg \max_{\lambda \in \mathfrak{R}^n} \{x'_{k+1} \lambda - f^*(\lambda)\}.$$

Let us now discuss the details of the preceding computations, assuming for simplicity that there are no constraints, i.e.,  $X = \mathfrak{R}^n$ . According to Section 6.4.3,  $F_k^*$  is a piecewise linear, inner approximation of  $f^*$  (see Fig. 6.4.5). In particular,  $F_k^*$  is a piecewise linear (inner) approximation of  $f^*$  with domain

$$\text{dom}(F_k^*) = \text{conv}(\{g_0, \dots, g_k\}),$$

and “break points” at  $g_i$ ,  $i = 0, \dots, k$ , with values equal to the corresponding values of  $f^*$ .

Let us now consider the dual proximal optimization of Eq. (6.107). It takes the form

$$\begin{aligned} & \text{minimize} && \sum_{i=0}^k \alpha_i (f^*(g_i) - x'_k g_i) + \frac{c_k}{2} \left\| \sum_{i=0}^k \alpha_i g_i \right\|^2 \\ & \text{subject to} && \sum_{i=0}^k \alpha_i = 1, \quad \alpha_i \geq 0, \quad i = 0, \dots, k. \end{aligned} \quad (6.117)$$

If  $(\alpha_0^k, \dots, \alpha_k^k)$  attains the minimum, we have

$$\lambda_{k+1} = \sum_{i=0}^k \alpha_i^k g_i, \quad x_{k+1} = x_k - c_k \sum_{i=0}^k \alpha_i^k g_i. \quad (6.118)$$

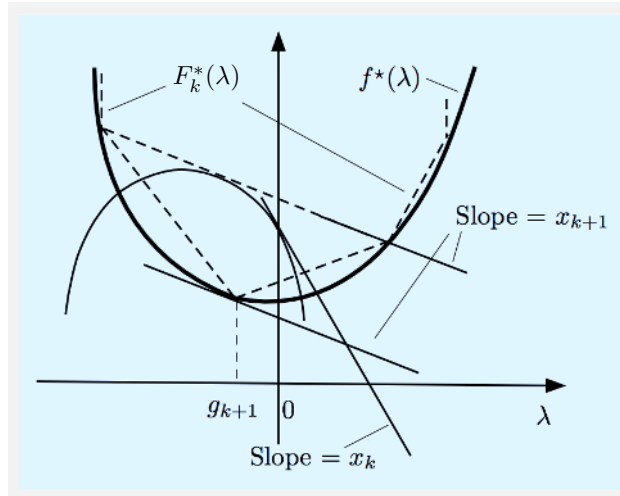
The next subgradient  $g_{k+1}$  may be obtained from the maximization

$$g_{k+1} \in \arg \max_{\lambda \in \mathfrak{R}^n} \{x'_{k+1} \lambda - f^*(\lambda)\}. \quad (6.119)$$

As Fig. 6.6.4 indicates,  $g_{k+1}$  provides a new break point and an improved inner approximation to  $f^*$  [equivalently,  $(g_{k+1}, f^*(g_{k+1}))$  is a new extreme point added to the Minkowski-Weyl representation of  $\text{epi}(F_k^*)$ ].

We refer to the algorithm defined by Eqs. (6.117), (6.118), (6.119), as the *proximal inner linearization algorithm*. Note that all the computations of the algorithm involve the conjugate  $f^*$  and not  $f$ . Thus, if  $f^*$  is more convenient to work with than  $f$ , the proximal inner linearization algorithm is preferable to the proximal cutting plane algorithm. The maximization (6.119) is often the most challenging part of the algorithm, and the key to its successful application.

Let us finally note that bundle versions of the algorithm are easily obtained by introducing a proximity control mechanism, and a corresponding test to distinguish between serious steps, where we update  $x_k$  via Eq. (6.118), and null steps, where we leave  $x_k$  unchanged, but simply add the extreme point  $(g_{k+1}, f^*(g_{k+1}))$  to the current inner approximation of  $f^*$ .



**Figure 6.6.4.** Illustration of an iteration of the proximal inner linearization algorithm. The proximal minimization determines the “slope”  $x_{k+1}$  of  $F_k^*$ , which then determines the next subgradient/break point  $g_{k+1}$  via the maximization

$$g_{k+1} \in \arg \max_{\lambda \in \mathbb{R}^n} \{x'_{k+1} \lambda - f^*(\lambda)\},$$

i.e.,  $g_{k+1}$  is a point at which  $x_{k+1}$  is a subgradient of  $f^*$ .

## 6.7 INTERIOR POINT METHODS

Let us consider inequality constrained problems of the form

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } x \in X, \quad g_j(x) \leq 0, \quad j = 1, \dots, r, \end{aligned} \tag{6.120}$$

where  $f$  and  $g_j$  are real-valued convex functions, and  $X$  is a closed convex set. The interior (relative to  $X$ ) of the set defined by the inequality constraints is

$$S = \{x \in X \mid g_j(x) < 0, j = 1, \dots, r\},$$

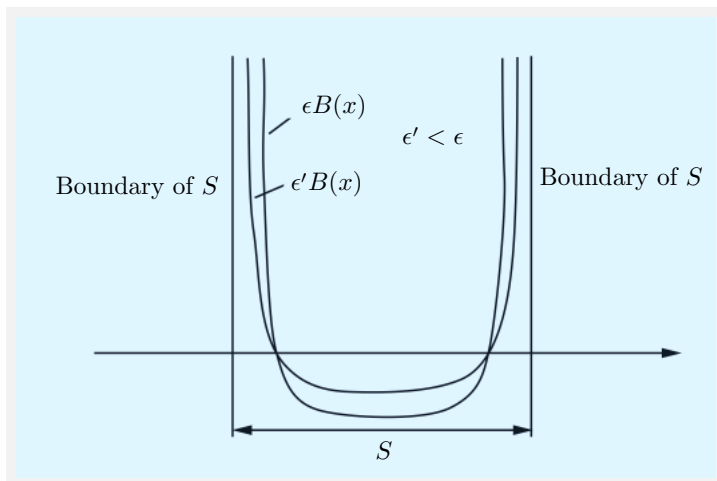
and is assumed to be nonempty.

In interior point methods, we add to the cost a function  $B(x)$  that is defined in the interior set  $S$ . This function, called the *barrier function*, is continuous and goes to  $\infty$  as any one of the constraints  $g_j(x)$  approaches 0 from negative values. The two most common examples of barrier functions are:

$$B(x) = - \sum_{j=1}^r \ln\{-g_j(x)\}, \quad \text{logarithmic,}$$

$$B(x) = -\sum_{j=1}^r \frac{1}{g_j(x)}, \quad \text{inverse.}$$

Note that both of these barrier functions are convex since the constraint functions  $g_j$  are convex. Figure 6.7.1 illustrates the form of  $B(x)$ .



**Figure 6.7.1** Form of a barrier function. The barrier term  $\epsilon B(x)$  goes to zero for all interior points  $x \in S$  as  $\epsilon \rightarrow 0$ .

The barrier method is defined by introducing a parameter sequence  $\{\epsilon_k\}$  with

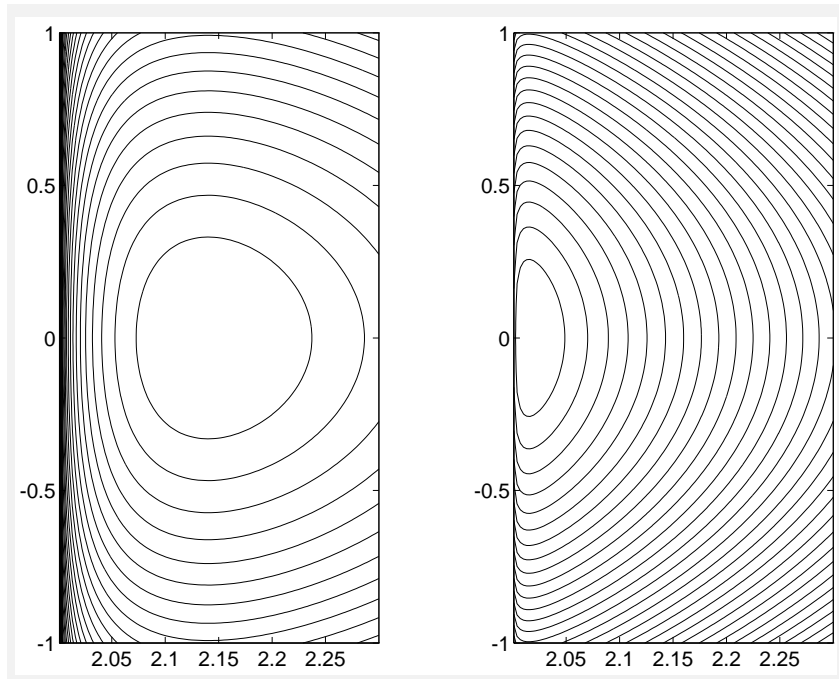
$$0 < \epsilon_{k+1} < \epsilon_k, \quad k = 0, 1, \dots, \quad \epsilon_k \rightarrow 0.$$

It consists of finding

$$x_k \in \arg \min_{x \in S} \{f(x) + \epsilon_k B(x)\}, \quad k = 0, 1, \dots \quad (6.121)$$

Since the barrier function is defined only on the interior set  $S$ , the successive iterates of any method used for this minimization must be interior points.

If  $X = \mathfrak{R}^n$ , one may use unconstrained methods such as Newton's method with the stepsize properly selected to ensure that all iterates lie in  $S$ . Indeed, Newton's method is often recommended for reasons that have to do with *ill-conditioning*, a phenomenon that relates to the difficulty of carrying out the minimization (6.121) (see Fig. 6.7.2 and sources such as [Ber99] for a discussion). Note that the barrier term  $\epsilon_k B(x)$  goes to zero for all interior points  $x \in S$  as  $\epsilon_k \rightarrow 0$ . Thus the barrier term becomes increasingly inconsequential as far as interior points are concerned, while progressively allowing  $x_k$  to get closer to the boundary of  $S$  (as it should if the solutions of the original constrained problem lie on the boundary



**Figure 6.7.2.** The convergence process of the barrier method for the two-dimensional problem

$$\begin{aligned} &\text{minimize } f(x) = \frac{1}{2}((x^1)^2 + (x^2)^2) \\ &\text{subject to } 2 \leq x^1, \end{aligned}$$

with optimal solution  $x^* = (2, 0)$ . For the case of the logarithmic barrier function  $B(x) = -\ln(x^1 - 2)$ , we have

$$x_k \in \arg \min_{x^1 > 2} \left\{ \frac{1}{2}((x^1)^2 + (x^2)^2) - \epsilon_k \ln(x^1 - 2) \right\} = \left( 1 + \sqrt{1 + \epsilon_k}, 0 \right),$$

so as  $\epsilon_k$  is decreased, the unconstrained minimum  $x_k$  approaches the constrained minimum  $x^* = (2, 0)$ . The figure shows the equal cost surfaces of  $f(x) + \epsilon B(x)$  for  $\epsilon = 0.3$  (left side) and  $\epsilon = 0.03$  (right side). As  $\epsilon_k \rightarrow 0$ , computing  $x_k$  becomes more difficult because of ill-conditioning (the equal cost surfaces become very elongated near  $x_k$ ).

of  $S$ ). Figure 6.7.2 illustrates the convergence process, and the following proposition gives the main convergence result.

**Proposition 6.7.1:** Every limit point of a sequence  $\{x_k\}$  generated by a barrier method is a global minimum of the original constrained problem (6.120).

**Proof:** Let  $\{\bar{x}\}$  be the limit of a subsequence  $\{x_k\}_{k \in K}$ . If  $\bar{x} \in S$ , we have  $\lim_{k \rightarrow \infty, k \in K} \epsilon_k B(x_k) = 0$ , while if  $\bar{x}$  lies on the boundary of  $S$ , we have  $\lim_{k \rightarrow \infty, k \in K} B(x_k) = \infty$ . In either case we obtain

$$\liminf_{k \rightarrow \infty} \epsilon_k B(x_k) \geq 0,$$

which implies that

$$\liminf_{k \rightarrow \infty, k \in K} \{f(x_k) + \epsilon_k B(x_k)\} = f(\bar{x}) + \liminf_{k \rightarrow \infty, k \in K} \{\epsilon_k B(x_k)\} \geq f(\bar{x}). \quad (6.122)$$

The vector  $\bar{x}$  is a feasible point of the original problem (6.120), since  $x_k \in S$  and  $X$  is a closed set. If  $\bar{x}$  were not a global minimum, there would exist a feasible vector  $x^*$  such that  $f(x^*) < f(\bar{x})$  and therefore also [since by the Line Segment Principle (Prop. 1.3.1)  $x^*$  can be approached arbitrarily closely through the interior set  $S$ ] an interior point  $\tilde{x} \in S$  such that  $f(\tilde{x}) < f(\bar{x})$ . We now have by the definition of  $x_k$ ,

$$f(x_k) + \epsilon_k B(x_k) \leq f(\tilde{x}) + \epsilon_k B(\tilde{x}), \quad k = 0, 1, \dots,$$

which by taking the limit as  $k \rightarrow \infty$  and  $k \in K$ , implies together with Eq. (6.122), that  $f(\bar{x}) \leq f(\tilde{x})$ . This is a contradiction, thereby proving that  $\bar{x}$  is a global minimum of the original problem. **Q.E.D.**

The idea of using a barrier function as an approximation to constraints has been used in several different ways, in methods that generate successive iterates lying in the interior of the constraint set. These methods are generically referred to as *interior point methods*, and have been extensively applied to linear, quadratic, and conic programming problems. The logarithmic barrier function has been central in many of these methods. In the next two sections we will discuss a few methods that are designed for problems with special structure. In particular, in Section 6.7.1 we will discuss in some detail primal-dual methods for linear programming, currently one of the most popular methods for solving linear programs. In Section 6.7.2 we will address briefly interior point methods for conic programming problems.

### 6.7.1 Primal-Dual Methods for Linear Programming

Let us consider the linear program

$$\begin{aligned} & \text{minimize } c'x \\ & \text{subject to } Ax = b, \quad x \geq 0, \end{aligned} \quad (\text{LP})$$

where  $c \in \Re^n$  and  $b \in \Re^m$  are given vectors, and  $A$  is an  $m \times n$  matrix of rank  $m$ . The dual problem, derived in Section 5.2, is given by

$$\begin{aligned} & \text{maximize } b'\lambda \\ & \text{subject to } A'\lambda \leq c. \end{aligned} \tag{DP}$$

As shown in Section 5.2, (LP) has an optimal solution if and only if (DP) has an optimal solution. Furthermore, when optimal solutions to (LP) and (DP) exist, the corresponding optimal values are equal.

Recall that the logarithmic barrier method involves finding for various  $\epsilon > 0$ ,

$$x(\epsilon) \in \arg \min_{x \in S} F_\epsilon(x), \tag{6.123}$$

where

$$F_\epsilon(x) = c'x - \epsilon \sum_{i=1}^n \ln x^i,$$

$x^i$  is the  $i$ th component of  $x$  and  $S$  is the interior set

$$S = \{x \mid Ax = b, x > 0\}.$$

We assume that  $S$  is nonempty and bounded.

Rather than directly minimizing  $F_\epsilon(x)$  for small values of  $\epsilon$  [cf. Eq. (6.123)], we will apply Newton's method for solving the system of optimality conditions for the problem of minimizing  $F_\epsilon(\cdot)$  over  $S$ . The salient features of this approach are:

- (a) Only one Newton iteration is carried out for each value of  $\epsilon_k$ .
- (b) For every  $k$ , the pair  $(x_k, \lambda_k)$  is such that  $x_k$  is an interior point of the positive orthant, that is,  $x_k > 0$ , while  $\lambda_k$  is an interior point of the dual feasible region, that is,

$$c - A'\lambda_k > 0.$$

(However,  $x_k$  need not be primal-feasible, that is, it need not satisfy the equation  $Ax = b$ .)

- (c) Global convergence is enforced by ensuring that the expression

$$P_k = x_k'z_k + \|Ax_k - b\|, \tag{6.124}$$

is decreased to 0, where  $z_k$  is the vector of slack variables

$$z_k = c - A'\lambda_k.$$

The expression (6.124) may be viewed as a *merit function*, and consists of two nonnegative terms: the first term is  $x_k'z_k$ , which is positive (since  $x_k > 0$  and  $z_k > 0$ ) and can be written as

$$x_k'z_k = x_k'(c - A'\lambda_k) = c'x_k - b'\lambda_k + (b - Ax_k)'\lambda_k.$$

Thus when  $x_k$  is primal-feasible ( $Ax_k = b$ ),  $x_k'z_k$  is equal to the duality gap, that is, the difference between the primal and the dual costs,  $c'x_k - b'\lambda_k$ . The second term is the norm of the primal constraint violation  $\|Ax_k - b\|$ . In the method to be described, neither of the terms  $x_k'z_k$  and  $\|Ax_k - b\|$  may increase at each iteration, so that  $P_{k+1} \leq P_k$  (and typically  $P_{k+1} < P_k$ ) for all  $k$ . If we can show that  $P_k \rightarrow 0$ , then asymptotically both the duality gap and the primal constraint violation will be driven to zero. Thus every limit point of  $\{(x_k, \lambda_k)\}$  will be a pair of primal and dual optimal solutions, in view of the duality relation

$$\min_{Ax=b, x \geq 0} c'x = \max_{A'\lambda \leq c} b'\lambda,$$

shown in Section 5.2.

Let us write the necessary and sufficient conditions for  $(x, \lambda)$  to be a primal and dual optimal solution pair for the problem of minimizing the barrier function  $F_\epsilon(x)$  subject to  $Ax = b$ . They are

$$c - \epsilon x^{-1} - A'\lambda = 0, \quad Ax = b, \quad (6.125)$$

where  $x^{-1}$  denotes the vector with components  $(x^i)^{-1}$ . Let  $z$  be the vector of slack variables

$$z = c - A'\lambda.$$

Note that  $\lambda$  is dual feasible if and only if  $z \geq 0$ .

Using the vector  $z$ , we can write the first condition of Eq. (6.125) as  $z - \epsilon x^{-1} = 0$  or, equivalently,  $XZ = \epsilon e$ , where  $X$  and  $Z$  are the diagonal matrices with the components of  $x$  and  $z$ , respectively, along the diagonal, and  $e$  is the vector with unit components,

$$X = \begin{pmatrix} x^1 & 0 & \cdots & 0 \\ 0 & x^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & x^n \end{pmatrix}, \quad Z = \begin{pmatrix} z^1 & 0 & \cdots & 0 \\ 0 & z^2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & z^n \end{pmatrix}, \quad e = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

Thus the optimality conditions (6.125) can be written in the equivalent form

$$XZe = \epsilon e, \quad (6.126)$$

$$Ax = b, \quad (6.127)$$

$$z + A'\lambda = c. \quad (6.128)$$

Given  $(x, \lambda, z)$  satisfying  $z + A'\lambda = c$ , and such that  $x > 0$  and  $z > 0$ , a Newton iteration for solving this system is

$$x(\alpha, \epsilon) = x + \alpha \Delta x, \quad (6.129)$$

$$\lambda(\alpha, \epsilon) = \lambda + \alpha\Delta\lambda,$$

$$z(\alpha, \epsilon) = z + \alpha\Delta z,$$

where  $\alpha$  is a stepsize such that  $0 < \alpha \leq 1$  and

$$x(\alpha, \epsilon) > 0, \quad z(\alpha, \epsilon) > 0,$$

and the Newton increment  $(\Delta x, \Delta\lambda, \Delta z)$  solves the linearized version of the system (6.126)-(6.128)

$$X\Delta z + Z\Delta x = -v, \tag{6.130}$$

$$A\Delta x = b - Ax, \tag{6.131}$$

$$\Delta z + A'\Delta\lambda = 0, \tag{6.132}$$

with  $v$  defined by

$$v = XZe - \epsilon e. \tag{6.133}$$

After a straightforward calculation, the solution of the linearized system (6.130)-(6.132) can be written as

$$\Delta\lambda = (AZ^{-1}XA')^{-1}(AZ^{-1}v + b - Ax), \tag{6.134}$$

$$\Delta z = -A'\Delta\lambda, \tag{6.135}$$

$$\Delta x = -Z^{-1}v - Z^{-1}X\Delta z.$$

Note that  $\lambda(\alpha, \epsilon)$  is dual feasible, since from Eq. (6.132) and the condition  $z + A'\lambda = c$ , we see that  $z(\alpha, \epsilon) + A'\lambda(\alpha, \epsilon) = c$ . Note also that if  $\alpha = 1$ , that is, a pure Newton step is used,  $x(\alpha, \epsilon)$  is primal feasible, since from Eq. (6.131) we have  $A(x + \Delta x) = b$ .

### Merit Function Improvement

We will now evaluate the changes in the constraint violation and the merit function (6.124) induced by the Newton iteration.

By using Eqs. (6.129) and (6.131), the new constraint violation is given by

$$Ax(\alpha, \epsilon) - b = Ax + \alpha A\Delta x - b = Ax + \alpha(b - Ax) - b = (1 - \alpha)(Ax - b). \tag{6.136}$$

Thus, since  $0 < \alpha \leq 1$ , the new norm of constraint violation  $\|Ax(\alpha, \epsilon) - b\|$  is always no larger than the old one. Furthermore, if  $x$  is primal-feasible ( $Ax = b$ ), the new iterate  $x(\alpha, \epsilon)$  is also primal-feasible.

The inner product

$$g = x'z \tag{6.137}$$



after the iteration becomes

$$\begin{aligned} g(\alpha, \epsilon) &= x(\alpha, \epsilon)'z(\alpha, \epsilon) \\ &= (x + \alpha\Delta x)'(z + \alpha\Delta z) \\ &= x'z + \alpha(x'\Delta z + z'\Delta x) + \alpha^2\Delta x'\Delta z. \end{aligned} \quad (6.138)$$

From Eqs. (6.131) and (6.135) we have

$$\Delta x'\Delta z = (Ax - b)'\Delta\lambda,$$

while by premultiplying Eq. (6.130) with  $e'$  and using the definition (6.133) for  $v$ , we obtain

$$x'\Delta z + z'\Delta x = -e'v = n\epsilon - x'z.$$

By substituting the last two relations in Eq. (6.138) and by using also the expression (6.137) for  $g$ , we see that

$$g(\alpha, \epsilon) = g - \alpha(g - n\epsilon) + \alpha^2(Ax - b)'\Delta\lambda. \quad (6.139)$$

Let us now denote by  $P$  and  $P(\alpha, \epsilon)$  the value of the merit function (6.124) before and after the iteration, respectively. We have by using the expressions (6.136) and (6.139),

$$\begin{aligned} P(\alpha, \epsilon) &= g(\alpha, \epsilon) + \|Ax(\alpha, \epsilon) - b\| \\ &= g - \alpha(g - n\epsilon) + \alpha^2(Ax - b)'\Delta\lambda + (1 - \alpha)\|Ax - b\|, \end{aligned}$$

or

$$P(\alpha, \epsilon) = P - \alpha(g - n\epsilon + \|Ax - b\|) + \alpha^2(Ax - b)'\Delta\lambda.$$

Thus if  $\epsilon$  is chosen to satisfy

$$\epsilon < \frac{g}{n}$$

and  $\alpha$  is chosen to be small enough so that the second order term  $\alpha^2(Ax - b)'\Delta\lambda$  is dominated by the first order term  $\alpha(g - n\epsilon)$ , the merit function will be improved as a result of the iteration.

### A General Class of Primal-Dual Algorithms

Let us consider now the general class of algorithms of the form

$$x_{k+1} = x(\alpha_k, \epsilon_k), \quad \lambda_{k+1} = \lambda(\alpha_k, \epsilon_k), \quad z_{k+1} = z(\alpha_k, \epsilon_k),$$

where  $\alpha_k$  and  $\epsilon_k$  are positive scalars such that

$$x_{k+1} > 0, \quad z_{k+1} > 0, \quad \epsilon_k < \frac{g_k}{n},$$

where  $g_k$  is the inner product

$$g_k = x_k' z_k + (Ax_k - b)' \lambda_k,$$

and  $\alpha_k$  is such that the merit function  $P_k$  is reduced. Initially we must have  $x_0 > 0$ , and  $z_0 = c - A' \lambda_0 > 0$  (such a point can often be easily found; otherwise an appropriate reformulation of the problem is necessary for which we refer to the specialized literature). These methods are generally called *primal-dual*, in view of the fact that they operate simultaneously on the primal and dual variables.

It can be shown that it is possible to choose  $\alpha_k$  and  $\epsilon_k$  so that the merit function is not only reduced at each iteration, but also converges to zero. Furthermore, with suitable choices of  $\alpha_k$  and  $\epsilon_k$ , algorithms with good theoretical properties, such as polynomial complexity and superlinear convergence, can be derived.

Computational experience has shown that with properly chosen sequences  $\alpha_k$  and  $\epsilon_k$ , and appropriate implementation, the practical performance of the primal-dual methods is excellent. The choice

$$\epsilon_k = \frac{g_k}{n^2},$$

leading to the relation

$$g_{k+1} = (1 - \alpha_k + \alpha_k/n)g_k$$

for feasible  $x_k$ , has been suggested as a good practical rule. Usually, when  $x_k$  has already become feasible,  $\alpha_k$  is chosen as  $\theta \tilde{\alpha}_k$ , where  $\theta$  is a factor very close to 1 (say 0.999), and  $\tilde{\alpha}_k$  is the maximum stepsize  $\alpha$  that guarantees that  $x(\alpha, \epsilon_k) \geq 0$  and  $z(\alpha, \epsilon_k) \geq 0$

$$\tilde{\alpha}_k = \min \left\{ \min_{i=1, \dots, n} \left\{ \frac{x_k^i}{-\Delta x^i} \mid \Delta x^i < 0 \right\}, \min_{i=1, \dots, n} \left\{ \frac{z_k^i}{-\Delta z^i} \mid \Delta z^i < 0 \right\} \right\}.$$

When  $x_k$  is not feasible, the choice of  $\alpha_k$  must also be such that the merit function is improved. In some works, a different stepsize for the  $x$  update than for the  $(\lambda, z)$  update has been suggested. The stepsize for the  $x$  update is near the maximum stepsize  $\alpha$  that guarantees  $x(\alpha, \epsilon_k) \geq 0$ , and the stepsize for the  $(\lambda, z)$  update is near the maximum stepsize  $\alpha$  that guarantees  $z(\alpha, \epsilon_k) \geq 0$ .

There are a number of additional practical issues related to implementation, for which we refer to the specialized literature. There are also more sophisticated implementations of the Newton/primal-dual idea. We refer to the research monographs by Wright [Wri97] and Ye [Ye97], and to other sources for a detailed discussion, as well as extensions to nonlinear/convex programming problems, such as quadratic programming.

### 6.7.2 Interior Point Methods for Conic Programming

We now discuss briefly interior point methods for the conic programming problems discussed in Section 6.1.2. Consider first the SOCP

$$\begin{aligned} & \text{minimize} && c'x \\ & \text{subject to} && A_i x - b_i \in C_i, \quad i = 1, \dots, m, \end{aligned} \quad (6.140)$$

where  $x \in \mathfrak{R}^n$ ,  $c$  is a vector in  $\mathfrak{R}^n$ , and for  $i = 1, \dots, m$ ,  $A_i$  is an  $n_i \times n$  matrix,  $b_i$  is a vector in  $\mathfrak{R}^{n_i}$ , and  $C_i$  is the second order cone of  $\mathfrak{R}^{n_i}$  [cf. Eq. (6.18)]. We approximate this problem with

$$\begin{aligned} & \text{minimize} && c'x + \epsilon_k \sum_{i=1}^m B_i(A_i x - b_i) \\ & \text{subject to} && x \in \mathfrak{R}^n, \end{aligned} \quad (6.141)$$

where  $B_i$  is a function defined in the interior of the second order cone  $C_i$ , and given by

$$B_i(y) = -\ln(y_{n_i}^2 - (y_1^2 + \dots + y_{n_i-1}^2)), \quad y \in \text{int}(C_i),$$

and  $\{\epsilon_k\}$  is a positive sequence that converges to 0. Thus we have  $B_i(A_i x - b_i) \rightarrow \infty$  as  $A_i x - b_i$  approaches the boundary of  $C_i$ .

Similar to Prop. 6.7.1, it can be shown that if  $x_k$  is an optimal solution of the approximating problem (6.141), then every limit point of  $\{x_k\}$  is an optimal solution of the original problem. For theoretical as well as practical reasons, the approximating problem (6.141) should not be solved exactly. In the most efficient methods, one or more Newton steps corresponding to a given value  $\epsilon_k$  are performed, and then the value of  $\epsilon_k$  is appropriately reduced. If the aim is to achieve a favorable polynomial complexity result, a single Newton step should be performed between successive reductions of  $\epsilon_k$ , and the subsequent reduction of  $\epsilon_k$  must be correspondingly small, according to an appropriate formula, which is designed to enable a polynomial complexity proof. An alternative, which is more efficient in practice, is to allow multiple Newton steps until an appropriate termination criterion is satisfied, and then reduce  $\epsilon_k$  substantially. When properly implemented, methods of this type require in practice a consistently small total number of Newton steps [a number typically no more than 50, regardless of dimension (!) is often reported]. This empirical observation is far more favorable than what is predicted by the theoretical complexity analysis. We refer to the book by Boyd and Vanderbergue [BoV04], and sources quoted there for further details.

There is a similar interior point method for the dual SDP involving the multiplier vector  $\lambda = (\lambda_1, \dots, \lambda_m)$ :

$$\begin{aligned} & \text{maximize} && b'\lambda \\ & \text{subject to} && C - (\lambda_1 A_1 + \dots + \lambda_m A_m) \in D, \end{aligned} \quad (6.142)$$

where  $D$  is the cone of positive semidefinite matrices [cf. Eq. (6.24)]. It consists of solving the problem

$$\begin{aligned} & \text{maximize} && b'\lambda + \epsilon_k \ln(\det(C - \lambda_1 A_1 - \dots - \lambda_m A_m)) \\ & \text{subject to} && \lambda \in \mathfrak{R}^m, \quad C - \lambda_1 A_1 - \dots - \lambda_m A_m \in \text{int}(D), \end{aligned} \tag{6.143}$$

where  $\{\epsilon_k\}$  is a positive sequence that converges to 0. Here, we should use a starting point such that  $C - \lambda_1 A_1 - \dots - \lambda_m A_m$  is positive definite, and Newton's method should ensure that the iterates keep  $C - \lambda_1 A_1 - \dots - \lambda_m A_m$  within the positive definite cone  $\text{int}(D)$ .

The properties of this method are similar to the ones of the preceding SOCP method. In particular, if  $x_k$  is an optimal solution of the approximating problem (6.143), then every limit point of  $\{x_k\}$  is an optimal solution of the original problem (6.142).

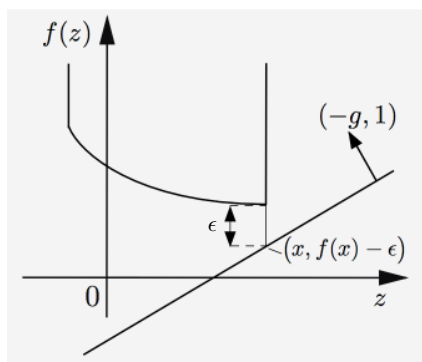
We finally note that there are primal-dual interior point methods for conic programming problems, which bear similarity with the one given in Section 6.7.1 for linear programming. Again, we refer to the specialized literature for further details and a complexity analysis.

## 6.8 APPROXIMATE SUBGRADIENT METHODS

In this section we consider various types of subgradient methods, which use approximate subgradients. There may be several different reasons for this approximation; for example, computational savings in the subgradient calculation, exploitation of special problem structure, or faster convergence.

### 6.8.1 $\epsilon$ -Subgradient Methods

Subgradient methods require the computation of a subgradient at the current point, but in some contexts, it may be necessary or convenient to use an approximation to a subgradient which we now introduce.



**Figure 6.8.1.** Illustration of an  $\epsilon$ -subgradient of a convex function  $f$ . A vector  $g$  is an  $\epsilon$ -subgradient at  $x \in \text{dom}(f)$  if and only if there is a hyperplane with normal  $(-g, 1)$ , which passes through the point  $(x, f(x) - \epsilon)$ , and separates this point from the epigraph of  $f$ .

Given a proper convex function  $f : \mathfrak{R}^n \mapsto (-\infty, \infty]$  and a scalar  $\epsilon > 0$ , we say that a vector  $g$  is an  $\epsilon$ -subgradient of  $f$  at a point  $x \in \text{dom}(f)$  if

$$f(z) \geq f(x) + (z - x)'g - \epsilon, \quad \forall z \in \mathfrak{R}^n. \quad (6.144)$$

The  $\epsilon$ -subdifferential  $\partial_\epsilon f(x)$  is the set of all  $\epsilon$ -subgradients of  $f$  at  $x$ , and by convention,  $\partial_\epsilon f(x) = \emptyset$  for  $x \notin \text{dom}(f)$ . It can be seen that

$$\partial_{\epsilon_1} f(x) \subset \partial_{\epsilon_2} f(x) \quad \text{if } 0 < \epsilon_1 < \epsilon_2,$$

and that

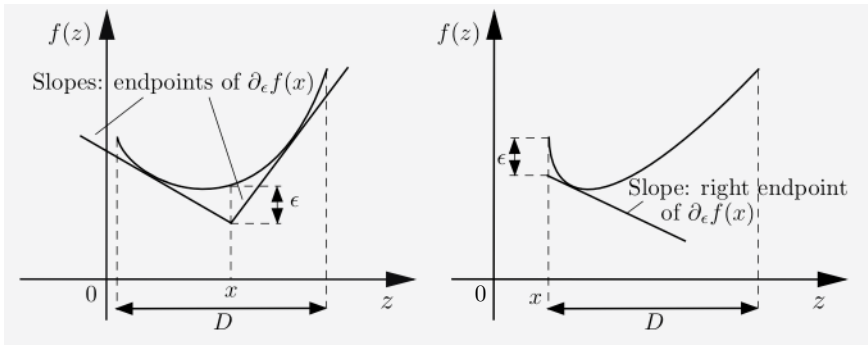
$$\bigcap_{\epsilon > 0} \partial_\epsilon f(x) = \partial f(x).$$

To interpret geometrically an  $\epsilon$ -subgradient, note that the defining relation (6.144) can be written as

$$f(z) - z'g \geq (f(x) - \epsilon) - x'g, \quad \forall z \in \mathfrak{R}^n.$$

Thus  $g$  is an  $\epsilon$ -subgradient at  $x$  if and only if the epigraph of  $f$  is contained in the positive halfspace corresponding to the hyperplane in  $\mathfrak{R}^{n+1}$  that has normal  $(-g, 1)$  and passes through  $(x, f(x) - \epsilon)$ , as illustrated in Fig. 6.8.1.

Figure 6.8.2 illustrates the definition of the  $\epsilon$ -subdifferential  $\partial_\epsilon f(x)$  for the case of a one-dimensional function  $f$ . The figure indicates that if  $f$  is closed, then [in contrast with  $\partial f(x)$ ]  $\partial_\epsilon f(x)$  is nonempty at all points of  $\text{dom}(f)$ . This follows by the Nonvertical Hyperplane Theorem (Prop. 1.5.8).



**Figure 6.8.2.** Illustration of the  $\epsilon$ -subdifferential  $\partial_\epsilon f(x)$  of a one-dimensional function  $f : \mathfrak{R} \mapsto (-\infty, \infty]$ , which is closed and convex, and has as effective domain an interval  $D$ . The  $\epsilon$ -subdifferential is an interval with endpoints corresponding to the slopes indicated in the figure. These endpoints can be  $-\infty$  (as in the figure on the right) or  $\infty$ .

The following example motivates the use of  $\epsilon$ -subgradients in the context of duality and minimax problems. It shows that  $\epsilon$ -subgradients may be computed more economically than subgradients, through an approximate minimization.

**Example 6.8.1: ( $\epsilon$ -Subgradient Calculation in Minimax and Dual Problems)**

As in Example 6.3.1, let us consider the minimization of

$$f(x) = \sup_{z \in Z} \phi(x, z), \quad (6.145)$$

where  $x \in \mathfrak{R}^n$ ,  $z \in \mathfrak{R}^m$ ,  $Z$  is a subset of  $\mathfrak{R}^m$ , and  $\phi : \mathfrak{R}^n \times \mathfrak{R}^m \mapsto (-\infty, \infty]$  is a function such that  $\phi(\cdot, z)$  is convex and closed for each  $z \in Z$ . We showed in Example 6.3.1 that if we carry out the maximization over  $z$  in Eq. (6.145), we can then obtain a subgradient at  $x$ . We will show with a similar argument, that if we carry out the maximization over  $z$  approximately, within  $\epsilon$ , we can then obtain an  $\epsilon$ -subgradient at  $x$ , which we can use in turn within an  $\epsilon$ -subgradient method.

Indeed, for a fixed  $x \in \text{dom}(f)$ , let us assume that  $z_x \in Z$  attains the supremum within  $\epsilon > 0$  in Eq. (6.145), i.e.,

$$\phi(x, z_x) \geq \sup_{z \in Z} \phi(x, z) - \epsilon = f(x) - \epsilon,$$

and that  $g_x$  is some subgradient of the convex function  $\phi(\cdot, z_x)$ , i.e.,  $g_x \in \partial\phi(x, z_x)$ . Then, for all  $y \in \mathfrak{R}^n$ , we have using the subgradient inequality,

$$f(y) = \sup_{z \in Z} \phi(y, z) \geq \phi(y, z_x) \geq \phi(x, z_x) + g'_x(y - x) \geq f(x) - \epsilon + g'_x(y - x),$$

i.e.,  $g_x$  is an  $\epsilon$ -subgradient of  $f$  at  $x$ , so

$$\phi(x, z_x) \geq \sup_{z \in Z} \phi(x, z) - \epsilon \text{ and } g_x \in \partial\phi(x, z_x) \quad \Rightarrow \quad g_x \in \partial_\epsilon f(x).$$

We now consider the class of  $\epsilon$ -subgradient methods for minimizing a real-valued convex function  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  over a closed convex set  $X$ , given by

$$x_{k+1} = P_X(x_k - \alpha_k g_k), \quad (6.146)$$

where  $g_k$  is an  $\epsilon_k$ -subgradient of  $f$  at  $x_k$ ,  $\alpha_k$  is a positive stepsize, and  $P_X(\cdot)$  denotes projection on  $X$ . Their convergence behavior and analysis are similar to those of subgradient methods, except that  $\epsilon$ -subgradient methods generally aim to converge to the  $\epsilon$ -optimal set, where  $\epsilon = \lim_{k \rightarrow \infty} \epsilon_k$ , rather than the optimal set, as subgradient methods do.

To get a sense of the convergence mechanism, note that there is a simple modification of the basic inequality of Prop. 6.3.1(a). In particular, if  $\{x_k\}$  is the sequence generated by the  $\epsilon$ -subgradient method, we have for all  $y \in X$  and  $k \geq 0$

$$\|x_{k+1} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k(f(x_k) - f(y) - \epsilon_k) + \alpha_k^2 \|g_k\|^2.$$

Using this inequality, one can essentially replicate the convergence analysis of Section 6.3. As an example, consider the case of constant  $\alpha_k$  and  $\epsilon_k$ :  $\alpha_k \equiv \alpha$  for some  $\alpha > 0$  and  $\epsilon_k \equiv \epsilon$  for some  $\epsilon > 0$ . Then, if the  $\epsilon$ -subgradients  $g_k$  are bounded, with  $\|g_k\| \leq c$  for some constant  $c$  and all  $k$ , we obtain for all optimal solutions  $x^*$ ,

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2\alpha(f(x_k) - f^* - \epsilon) + \alpha^2 c^2,$$

where  $f^* = \inf_{x \in X} f(x)$  is the optimal value. This implies that the distance to  $x^*$  decreases if

$$0 < \alpha < \frac{2(f(x_k) - f^* - \epsilon)}{c^2}$$

or equivalently, if  $x_k$  is outside the level set

$$\left\{ x \mid f(x) \leq f^* + \epsilon + \frac{\alpha c^2}{2} \right\}$$

(cf. Fig. 6.3.3). With analysis similar to the one for the subgradient case, we can also show that if  $\alpha_k \rightarrow 0$ ,  $\sum_{k=0}^{\infty} \alpha_k = \infty$ , and  $\epsilon_k \rightarrow \epsilon \geq 0$ , we have

$$\liminf_{k \rightarrow \infty} f(x_k) \leq f^* + \epsilon$$

(cf. Prop. 6.3.4).

## 6.8.2 Incremental Subgradient Methods

An interesting form of approximate subgradient method is an *incremental* variant, which applies to minimization over a closed convex set  $X$  of an additive cost function of the form

$$f(x) = \sum_{i=1}^m f_i(x), \quad (6.147)$$

where the functions  $f_i : \mathfrak{R}^n \mapsto \mathfrak{R}$  are convex. We mentioned several contexts where cost functions of this type arise in Section 6.1 [cf. Eq. (6.147)]. The idea of the incremental approach is to sequentially take steps along the subgradients of the component functions  $f_i$ , with intermediate adjustment of  $x$  after processing each component function.

Incremental methods are particularly interesting when the number of cost terms  $m$  is very large, and much larger than the dimension  $n$  (think here of  $n$  in the tens and hundreds, and  $m$  in the thousands and millions). Then a full subgradient step is very costly, and one hopes to make progress with approximate but much cheaper incremental steps.

In the incremental subgradient method, *an iteration is viewed as a cycle of  $m$  subiterations*. If  $x_k$  is the vector obtained after  $k$  cycles, the vector  $x_{k+1}$  obtained after one more cycle is

$$x_{k+1} = \psi_{m,k}, \quad (6.148)$$

where starting with  $\psi_{0,k} = x_k$ , we obtain  $\psi_{m,k}$  after the  $m$  steps

$$\psi_{i,k} = P_X(\psi_{i-1,k} - \alpha_k g_{i,k}), \quad i = 1, \dots, m, \quad (6.149)$$

with  $g_{i,k}$  being a subgradient of  $f_i$  at  $\psi_{i-1,k}$ .

The motivation for this method is faster convergence. In particular, we hope that far from the solution, a single cycle of the incremental subgradient method will be as effective as several (as many as  $m$ ) iterations of the ordinary subgradient method (think of the case where the components  $f_i$  are similar in structure).

One way to explain the convergence mechanism of the incremental method is to establish a connection with the  $\epsilon$ -subgradient method (6.146). An important fact here is that *if two vectors  $x$  and  $\bar{x}$  are “near” each other, then subgradients at  $\bar{x}$  can be viewed as  $\epsilon$ -subgradients at  $x$ , with  $\epsilon$  “small.”* In particular, if  $g \in \partial f(\bar{x})$ , we have for all  $z \in \mathfrak{R}^n$ ,

$$\begin{aligned} f(z) &\geq f(\bar{x}) + g'(z - \bar{x}) \\ &\geq f(x) + g'(z - x) + f(\bar{x}) - f(x) + g'(x - \bar{x}) \\ &\geq f(x) + g'(z - x) - \epsilon, \end{aligned}$$

where

$$\epsilon = \max\{0, f(x) - f(\bar{x})\} + \|g\| \cdot \|\bar{x} - x\|.$$

Thus, we have  $g \in \partial_\epsilon f(x)$ , and  $\epsilon$  is small when  $\bar{x}$  is near  $x$ .

We now observe from Eq. (6.149) that the  $i$ th step within a cycle of the incremental subgradient method involves the direction  $g_{i,k}$ , which is a subgradient of  $f_i$  at the corresponding vector  $\psi_{i-1,k}$ . If the stepsize  $\alpha_k$  is small, then  $\psi_{i-1,k}$  is close to the vector  $x_k$  available at the start of the cycle, and hence  $g_{i,k}$  is an  $\epsilon_i$ -subgradient of  $f_i$  at  $x_k$ , where  $\epsilon_i$  is small. In particular, if we ignore the projection operation in Eq. (6.149), we have

$$x_{k+1} = x_k - \alpha_k \sum_{i=1}^m g_{i,k},$$

where  $g_i$  is a subgradient of  $f_i$  at  $\psi_{i-1,k}$ , and hence an  $\epsilon_i$ -subgradient of  $f_i$  at  $x_k$ , where  $\epsilon_i$  is “small” (proportional to  $\alpha_k$ ). Let us also use the formula

$$\partial_{\epsilon_1} f_1(x) + \dots + \partial_{\epsilon_m} f_m(x) \subset \partial_\epsilon f(x),$$

where  $\epsilon = \epsilon_1 + \dots + \epsilon_m$ , to approximate the  $\epsilon$ -subdifferential of the sum  $f = \sum_{i=1}^m f_i$ . (This relation follows from the definition of  $\epsilon$ -subgradient.)



Then, it can be seen that the incremental subgradient iteration can be viewed as an  $\epsilon$ -subgradient iteration with  $\epsilon = \epsilon_1 + \dots + \epsilon_m$ . The size of  $\epsilon$  depends on the size of  $\alpha_k$ , as well as the function  $f$ , and we generally have  $\epsilon \rightarrow 0$  as  $\alpha_k \rightarrow 0$ . As a result, for the case where  $\alpha_k \rightarrow 0$  and  $\sum_{k=0}^{\infty} \alpha_k = \infty$ , the incremental subgradient method converges to the optimal value, similar to the ordinary subgradient method. If the stepsize  $\alpha_k$  is kept constant, convergence to a neighborhood of the solution can be expected. These results will be established more precisely and in greater detail in the analysis that follows.

### Convergence Analysis

Incremental subgradient methods have a rich theory, which includes convergence and rate of convergence analysis, optimization and randomization issues of the component order selection, and distributed computation aspects. Our analysis in this section is selective, and focuses on the case of a constant stepsize. We refer to the sources cited at the end of the chapter for a fuller discussion.

We use the notation

$$f^* = \inf_{x \in X} f(x), \quad X^* = \{x \in X \mid f(x) = f^*\},$$

$$d(x) = \inf_{x^* \in X^*} \|x - x^*\|,$$

where  $\|\cdot\|$  denotes the standard Euclidean norm. In our analysis, we assume the following:

**Assumption 6.8.1: (Subgradient Boundedness)** We have

$$c_i \geq \sup_{k \geq 0} \{\|g\| \mid g \in \partial f_i(x_k) \cup \partial f_i(\psi_{i-1,k})\}, \quad i = 1, \dots, m,$$

for some scalars  $c_1, \dots, c_m$ .

We note that Assumption 6.8.1 is satisfied if each  $f_i$  is real-valued and polyhedral. In particular, Assumption 6.8.1 holds for the dual of an integer programming problem, where for each  $i$  and all  $x$  the set of subgradients  $\partial f_i(x)$  is the convex hull of a finite number of points. More generally, since each component  $f_i$  is real-valued and convex over the entire space  $\mathbb{R}^n$ , the subdifferential  $\partial f_i(x)$  is nonempty and compact for all  $x$  and  $i$  (Prop. 5.4.1). If the set  $X$  is compact or the sequences  $\{\psi_{i,k}\}$  are bounded, then Assumption 6.8.1 is satisfied since the set  $\cup_{x \in B} \partial f_i(x)$  is bounded for any bounded set  $B$  (cf. Prop. 5.4.2).

The following is a key lemma, which parallels Prop. 6.3.1(a) for the (nonincremental) subgradient method.

**Lemma 6.8.1:** Let  $\{x_k\}$  be the sequence generated by the incremental method (6.148), (6.149). Then for all  $y \in X$  and  $k \geq 0$ , we have

$$\|x_{k+1} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k(f(x_k) - f(y)) + \alpha_k^2 c^2, \quad (6.150)$$

where  $c = \sum_{i=1}^m c_i$  and  $c_i$  are the scalars of Assumption 6.8.1.

**Proof:** Using the nonexpansion property of the projection, the subgradient boundedness (cf. Assumption 6.8.1), and the subgradient inequality for each component function  $f_i$ , we obtain for all  $y \in X$ ,

$$\begin{aligned} \|\psi_{i,k} - y\|^2 &= \|P_X(\psi_{i-1,k} - \alpha_k g_{i,k}) - y\|^2 \\ &\leq \|\psi_{i-1,k} - \alpha_k g_{i,k} - y\|^2 \\ &\leq \|\psi_{i-1,k} - y\|^2 - 2\alpha_k g'_{i,k}(\psi_{i-1,k} - y) + \alpha_k^2 c_i^2 \\ &\leq \|\psi_{i-1,k} - y\|^2 - 2\alpha_k(f_i(\psi_{i-1,k}) - f_i(y)) + \alpha_k^2 c_i^2, \quad \forall i, k. \end{aligned}$$

By adding the above inequalities over  $i = 1, \dots, m$ , we have for all  $y \in X$  and  $k$ ,

$$\begin{aligned} \|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2\alpha_k \sum_{i=1}^m (f_i(\psi_{i-1,k}) - f_i(y)) + \alpha_k^2 \sum_{i=1}^m c_i^2 \\ &= \|x_k - y\|^2 - 2\alpha_k \left( f(x_k) - f(y) + \sum_{i=1}^m (f_i(\psi_{i-1,k}) - f_i(x_k)) \right) \\ &\quad + \alpha_k^2 \sum_{i=1}^m c_i^2. \end{aligned}$$

By strengthening the above inequality, we have for all  $y \in X$  and  $k$ , using also the fact  $\psi_{0,k} = x_k$ ,

$$\begin{aligned} \|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2\alpha_k(f(x_k) - f(y)) \\ &\quad + 2\alpha_k \sum_{i=1}^m c_i \|\psi_{i-1,k} - x_k\| + \alpha_k^2 \sum_{i=1}^m c_i^2 \\ &\leq \|x_k - y\|^2 - 2\alpha_k(f(x_k) - f(y)) \\ &\quad + \alpha_k^2 \left( 2 \sum_{i=2}^m c_i \left( \sum_{j=1}^{i-1} c_j \right) + \sum_{i=1}^m c_i^2 \right) \end{aligned}$$

$$\begin{aligned}
&= \|x_k - y\|^2 - 2\alpha_k(f(x_k) - f(y)) + \alpha_k^2 \left( \sum_{i=1}^m c_i \right)^2 \\
&= \|x_k - y\|^2 - 2\alpha_k(f(x_k) - f(y)) + \alpha_k^2 c^2,
\end{aligned}$$

where in the first inequality we use the relation

$$f_i(x_k) - f_i(\psi_{i-1,k}) \leq \|\tilde{g}_{i,k}\| \cdot \|\psi_{i-1,k} - x_k\| \leq c_i \|\psi_{i-1,k} - x_k\|$$

with  $\tilde{g}_{i,k} \in \partial f_i(x_k)$ , and in the second inequality we use the relation

$$\|\psi_{i,k} - x_k\| \leq \alpha_k \sum_{j=1}^i c_j, \quad i = 1, \dots, m, \quad k \geq 0,$$

which follows from Eqs. (6.148), (6.149), and Assumption 6.8.1. **Q.E.D.**

Among other things, Lemma 6.8.1 guarantees that given the current iterate  $x_k$  and some other point  $y \in X$  with lower cost than  $x_k$ , the next iterate  $x_{k+1}$  will be closer to  $y$  than  $x_k$ , provided the stepsize  $\alpha_k$  is sufficiently small [less than  $2(f(x_k) - f(y))/c^2$ ]. In particular, for any optimal solution  $x^* \in X^*$ , any  $\epsilon > 0$ , and any  $\alpha_k \leq \epsilon/c^2$ , either

$$f(x_k) \leq f^* + \epsilon,$$

or else

$$\|x_{k+1} - x^*\| < \|x_k - x^*\|.$$

As in the case of the (nonincremental) subgradient method, for a constant stepsize rule, convergence can be established to a neighborhood of the optimum, which shrinks to 0 as the stepsize approaches 0. Convergence results for diminishing stepsize, and dynamic stepsize rules, which parallel Props. 6.3.4-6.3.6 can also be similarly established (see the sources cited at the end of the chapter).

**Proposition 6.8.1:** Let  $\{x_k\}$  be the sequence generated by the incremental method (6.148), (6.149), with the stepsize  $\alpha_k$  fixed at some positive constant  $\alpha$ .

(a) If  $f^* = -\infty$ , then

$$\liminf_{k \rightarrow \infty} f(x_k) = f^*.$$

(b) If  $f^* > -\infty$ , then

$$\liminf_{k \rightarrow \infty} f(x_k) \leq f^* + \frac{\alpha c^2}{2},$$

where  $c = \sum_{i=1}^m c_i$ .

**Proof:** We prove (a) and (b) simultaneously. If the result does not hold, there must exist an  $\epsilon > 0$  such that

$$\liminf_{k \rightarrow \infty} f(x_k) - \frac{\alpha c^2}{2} - 2\epsilon > f^*.$$

Let  $\hat{y} \in X$  be such that

$$\liminf_{k \rightarrow \infty} f(x_k) - \frac{\alpha c^2}{2} - 2\epsilon \geq f(\hat{y}),$$

and let  $k_0$  be large enough so that for all  $k \geq k_0$ , we have

$$f(x_k) \geq \liminf_{k \rightarrow \infty} f(x_k) - \epsilon.$$

By adding the preceding two relations, we obtain for all  $k \geq k_0$ ,

$$f(x_k) - f(\hat{y}) \geq \frac{\alpha c^2}{2} + \epsilon.$$

Using Lemma 6.8.1 for the case where  $y = \hat{y}$  together with the above relation, we obtain for all  $k \geq k_0$ ,

$$\|x_{k+1} - \hat{y}\|^2 \leq \|x_k - \hat{y}\|^2 - 2\alpha\epsilon.$$

This relation implies that for all  $k \geq k_0$ ,

$$\|x_{k+1} - \hat{y}\|^2 \leq \|x_{k-1} - \hat{y}\|^2 - 4\alpha\epsilon \leq \dots \leq \|x_{k_0} - \hat{y}\|^2 - 2(k+1-k_0)\alpha\epsilon,$$

which cannot hold for  $k$  sufficiently large – a contradiction. **Q.E.D.**

The preceding proposition involves only the iterates at the end of cycles. However, by shifting the starting index in each cycle and repeating the preceding proof, we see that

$$\liminf_{k \rightarrow \infty} f(\psi_{i,k}) \leq f^* + \frac{\alpha c^2}{2}, \quad \forall i = 1, \dots, m. \quad (6.151)$$

The next proposition gives an estimate of the number  $K$  of cycles needed to guarantee a given level of optimality up to the threshold tolerance  $\alpha c^2/2$  given in the preceding proposition.

**Proposition 6.8.2:** Assume that  $X^*$  is nonempty. Let  $\{x_k\}$  be the sequence generated by the incremental method (6.148), (6.149), with the stepsize  $\alpha_k$  fixed at some positive constant  $\alpha$ . Then for any positive scalar  $\epsilon$  we have

$$\min_{0 \leq k \leq K} f(x_k) \leq f^* + \frac{\alpha c^2 + \epsilon}{2}, \quad (6.152)$$

where  $K$  is given by

$$K = \left\lceil \frac{d(x_0)^2}{\alpha\epsilon} \right\rceil.$$

**Proof:** Assume, to arrive at a contradiction, that Eq. (6.152) does not hold, so that for all  $k$  with  $0 \leq k \leq K$ , we have

$$f(x_k) > f^* + \frac{\alpha c^2 + \epsilon}{2}.$$

By using this relation in Lemma 6.8.1 with  $\alpha_k$  replaced by  $\alpha$ , we obtain for all  $k$  with  $0 \leq k \leq K$ ,

$$\begin{aligned} (d(x_{k+1}))^2 &\leq (d(x_k))^2 - 2\alpha(f(x_k) - f^*) + \alpha^2 c^2 \\ &\leq (d(x_k))^2 - (\alpha^2 c^2 + \alpha\epsilon) + \alpha^2 c^2 \\ &= (d(x_k))^2 - \alpha\epsilon. \end{aligned}$$

Summation of the above inequalities over  $k$  for  $k = 0, \dots, K$ , yields

$$(d(x_{K+1}))^2 \leq d(x_0)^2 - (K+1)\alpha\epsilon,$$

so that

$$d(x_0)^2 - (K+1)\alpha\epsilon \geq 0,$$

which contradicts the definition of  $K$ . **Q.E.D.**

Note that the estimate (6.152) involves only the iterates obtained at the end of cycles. Since every cycle consists of  $m$  subiterations, the total number  $N$  of component functions that must be processed in order for Eq. (6.152) to hold is given by

$$N = mK = m \left\lceil \frac{(d(x_0))^2}{\alpha\epsilon} \right\rceil.$$

### The Role of the Order of Processing the Components

The error tolerance estimate  $\alpha c^2/2$  of Prop. 6.8.1 and Eq. (6.151) is an upper bound, and assumes the *worst* possible order of processing the components  $f_i$  within a cycle. One question that arises is whether this bound is sharp, in the sense that there exists a problem and a processing order, such that for each stepsize  $\alpha$ , we can find a starting point for which the sequence  $\{\psi_{i,k}\}$  generated by the method satisfies Eq. (6.151). Exercise 6.17 provides an example where the bound is satisfied within a constant that is independent of the problem data, i.e., for an unfavorable processing order and starting point, the method satisfies

$$\liminf_{k \rightarrow \infty} f(\psi_{i,k}) = f^* + \frac{\beta\alpha c^2}{2}, \quad \forall i = 1, \dots, m, \quad (6.153)$$

where  $\beta$  is a positive constant that is fairly close to 1. Thus, there is not much room for improvement of the worst-order error tolerance estimate  $\alpha c^2/2$ .

On the other hand, suppose that we are free to choose the *best* possible order of processing the components  $f_i$  within a cycle. Would it then be possible to lower the tolerance estimate  $\alpha c^2/2$ , and by how much? We claim that with such an optimal choice, it is impossible to lower the tolerance estimate by more than a factor of  $m$ . To see this, consider the case where all the  $f_i$  are the one-dimensional functions  $f_i(x) = (c/m)|x|$ . Then, because all functions  $f_i$  are identical, the order of processing the components is immaterial. If we start at  $x_0 = (\alpha c)/2m$ , then it can be seen that the method oscillates between  $x_0$  and  $-x_0$ , and the corresponding function value is

$$f(x_0) = f(-x_0) = \sum_{i=1}^m \frac{c}{m} \left| \frac{\alpha c}{2m} \right| = \frac{\alpha c^2}{2m}.$$

Since  $f^* = 0$ , this example shows that there exists a problem and a starting point such that

$$\liminf_{k \rightarrow \infty} f(\psi_{i,k}) = f^* + \frac{\alpha c^2}{2m}, \quad \forall i = 1, \dots, m. \quad (6.154)$$

Thus from Eqs. (6.153) and (6.154), we see that for a given stepsize  $\alpha$ , the achievable range for the bound

$$\liminf_{k \rightarrow \infty} f(\psi_{i,k}) - f^*$$

corresponding to the incremental subgradient method with a fixed processing order is

$$\left[ \frac{\alpha c^2}{2m}, \frac{\alpha c^2}{2} \right]. \quad (6.155)$$

By this we mean that there exists a choice of problem for which we can do no better than the lower end of the above range, even with optimal processing order choice; moreover, for all problems and processing orders, we will do no worse than the upper end of the above range.

From the bound range (6.155), it can be seen that for a given stepsize  $\alpha$ , there is significant difference in the performance of the method with the best and the worst processing orders. Unfortunately, it is difficult to find the best processing order for a given problem. In the next section, we will show that, remarkably, *by randomizing the order, we can achieve the lower tolerance error estimate*

$$\frac{\alpha c^2}{2m}$$

*with probability 1.*

### 6.8.3 Subgradient Methods with Randomization

It can be verified that the convergence analysis of the preceding subsection goes through assuming any order for processing the component functions  $f_i$ , as long as each component is taken into account exactly once within a cycle. In particular, at the beginning of each cycle, we could reorder the components  $f_i$  by either shifting or reshuffling and then proceed with the calculations until the end of the cycle. However, the order used can significantly affect the rate of convergence of the method. Unfortunately, determining the most favorable order may be very difficult in practice. A popular technique for incremental methods is to reshuffle randomly the order of the functions  $f_i$  at the beginning of each cycle. A variation of this method is to pick randomly a function  $f_i$  at each iteration rather than to pick each  $f_i$  exactly once in every cycle according to a randomized order. In this section, we analyze this type of method for the case of a constant stepsize.

We focus on the randomized method given by

$$x_{k+1} = P_X(x_k - \alpha g(\omega_k, x_k)), \quad (6.156)$$

where  $\omega_k$  is a random variable taking equiprobable values from the set  $\{1, \dots, m\}$ , and  $g(\omega_k, x_k)$  is a subgradient of the component  $f_{\omega_k}$  at  $x_k$ . This simply means that if the random variable  $\omega_k$  takes a value  $j$ , then the vector  $g(\omega_k, x_k)$  is a subgradient of  $f_j$  at  $x_k$ . Throughout this section we assume the following.

**Assumption 6.8.2:** For the randomized method (6.156):

- (a)  $\{\omega_k\}$  is a sequence of independent random variables, each uniformly distributed over the set  $\{1, \dots, m\}$ . Furthermore, the sequence  $\{\omega_k\}$  is independent of the sequence  $\{x_k\}$ .
- (b) The set of subgradients  $\{g(\omega_k, x_k) \mid k = 0, 1, \dots\}$  is bounded, i.e., there is a positive constant  $c_0$  such that with probability 1

$$\|g(\omega_k, x_k)\| \leq c_0, \quad \forall k \geq 0.$$

Note that if the set  $X$  is compact or the components  $f_i$  are polyhedral, then Assumption 6.8.2(b) is satisfied. The proofs of several propositions in this section rely on the Supermartingale Convergence Theorem as stated for example in Bertsekas and Tsitsiklis [BeT96], p. 148.

**Proposition 6.8.3: (Supermartingale Convergence Theorem)**

Let  $Y_k$ ,  $Z_k$ , and  $W_k$ ,  $k = 0, 1, 2, \dots$ , be three sequences of random variables and let  $\mathcal{F}_k$ ,  $k = 0, 1, 2, \dots$ , be sets of random variables such that  $\mathcal{F}_k \subset \mathcal{F}_{k+1}$  for all  $k$ . Suppose that:

- (1) The random variables  $Y_k$ ,  $Z_k$ , and  $W_k$  are nonnegative, and are functions of the random variables in  $\mathcal{F}_k$ .
- (2) For each  $k$ , we have

$$E\{Y_{k+1} \mid \mathcal{F}_k\} \leq Y_k - Z_k + W_k.$$

- (3) There holds, with probability 1,  $\sum_{k=0}^{\infty} W_k < \infty$ .

Then, we have  $\sum_{k=0}^{\infty} Z_k < \infty$ , and the sequence  $Y_k$  converges to a nonnegative random variable  $Y$ , with probability 1.

The following proposition parallels Prop. 6.8.1 for the deterministic incremental method.

**Proposition 6.8.4:** Let  $\{x_k\}$  be the sequence generated by the randomized incremental method (6.156).

- (a) If  $f^* = -\infty$ , then with probability 1

$$\inf_{k \geq 0} f(x_k) = f^*.$$

- (b) If  $f^* > -\infty$ , then with probability 1

$$\inf_{k \geq 0} f(x_k) \leq f^* + \frac{\alpha m c_0^2}{2}.$$

**Proof:** By adapting Lemma 6.8.1 to the case where  $f$  is replaced by  $f_{\omega_k}$ , we have

$$\|x_{k+1} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha(f_{\omega_k}(x_k) - f_{\omega_k}(y)) + \alpha^2 c_0^2, \quad \forall y \in X, \quad k \geq 0.$$

By taking the conditional expectation with respect to  $\mathcal{F}_k = \{x_0, \dots, x_k\}$ ,



the method's history up to  $x_k$ , we obtain for all  $y \in X$  and  $k$ ,

$$\begin{aligned} E\{\|x_{k+1} - y\|^2 \mid \mathcal{F}_k\} &\leq \|x_k - y\|^2 - 2\alpha E\{f_{\omega_k}(x_k) - f_{\omega_k}(y) \mid \mathcal{F}_k\} + \alpha^2 c_0^2 \\ &= \|x_k - y\|^2 - 2\alpha \sum_{i=1}^m \frac{1}{m} (f_i(x_k) - f_i(y)) + \alpha^2 c_0^2 \\ &= \|x_k - y\|^2 - \frac{2\alpha}{m} (f(x_k) - f(y)) + \alpha^2 c_0^2, \end{aligned} \tag{6.157}$$

where the first equality follows since  $\omega_k$  takes the values  $1, \dots, m$  with equal probability  $1/m$ .

Now, fix a positive scalar  $\gamma$ , consider the level set  $L_\gamma$  defined by

$$L_\gamma = \begin{cases} \left\{ x \in X \mid f(x) < -\gamma + 1 + \frac{\alpha m c_0^2}{2} \right\} & \text{if } f^* = -\infty, \\ \left\{ x \in X \mid f(x) < f^* + \frac{2}{\gamma} + \frac{\alpha m c_0^2}{2} \right\} & \text{if } f^* > -\infty, \end{cases}$$

and let  $y_\gamma \in X$  be such that

$$f(y_\gamma) = \begin{cases} -\gamma & \text{if } f^* = -\infty, \\ f^* + \frac{1}{\gamma} & \text{if } f^* > -\infty. \end{cases}$$

Note that  $y_\gamma \in L_\gamma$  by construction. Define a new process  $\{\hat{x}_k\}$  as follows

$$\hat{x}_{k+1} = \begin{cases} P_X(\hat{x}_k - \alpha g(\omega_k, \hat{x}_k)) & \text{if } \hat{x}_k \notin L_\gamma, \\ y_\gamma & \text{otherwise,} \end{cases}$$

where  $\hat{x}_0 = x_0$ . Thus the process  $\{\hat{x}_k\}$  is identical to  $\{x_k\}$ , except that once  $x_k$  enters the level set  $L_\gamma$ , the process terminates with  $\hat{x}_k = y_\gamma$  (since  $y_\gamma \in L_\gamma$ ). We will now argue that  $\{\hat{x}_k\}$  (and hence also  $\{x_k\}$ ) will eventually enter each of the sets  $L_\gamma$ .

Using Eq. (6.157) with  $y = y_\gamma$ , we have

$$E\{\|\hat{x}_{k+1} - y_\gamma\|^2 \mid \mathcal{F}_k\} \leq \|\hat{x}_k - y_\gamma\|^2 - \frac{2\alpha}{m} (f(\hat{x}_k) - f(y_\gamma)) + \alpha^2 c_0^2,$$

or equivalently

$$E\{\|\hat{x}_{k+1} - y_\gamma\|^2 \mid \mathcal{F}_k\} \leq \|\hat{x}_k - y_\gamma\|^2 - z_k, \tag{6.158}$$

where

$$z_k = \begin{cases} \frac{2\alpha}{m} (f(\hat{x}_k) - f(y_\gamma)) - \alpha^2 c_0^2 & \text{if } \hat{x}_k \notin L_\gamma, \\ 0 & \text{if } \hat{x}_k = y_\gamma. \end{cases}$$

The idea of the subsequent argument is to show that as long as  $\hat{x}_k \notin L_\gamma$ , the scalar  $z_k$  (which is a measure of progress) is strictly positive and bounded away from 0.

(a) Let  $f^* = -\infty$ . Then if  $\hat{x}_k \notin L_\gamma$ , we have

$$\begin{aligned} z_k &= \frac{2\alpha}{m} (f(\hat{x}_k) - f(y_\gamma)) - \alpha^2 c_0^2 \\ &\geq \frac{2\alpha}{m} \left( -\gamma + 1 + \frac{\alpha m c_0^2}{2} + \gamma \right) - \alpha^2 c_0^2 \\ &= \frac{2\alpha}{m}. \end{aligned}$$

Since  $z_k = 0$  for  $\hat{x}_k \in L_\gamma$ , we have  $z_k \geq 0$  for all  $k$ , and by Eq. (6.158) and the Supermartingale Convergence Theorem (cf. Prop. 6.8.3),  $\sum_{k=0}^{\infty} z_k < \infty$  implying that  $\hat{x}_k \in L_\gamma$  for sufficiently large  $k$ , with probability 1. Therefore, in the original process we have

$$\inf_{k \geq 0} f(x_k) \leq -\gamma + 1 + \frac{\alpha m c_0^2}{2}$$

with probability 1. Letting  $\gamma \rightarrow \infty$ , we obtain  $\inf_{k \geq 0} f(x_k) = -\infty$  with probability 1.

(b) Let  $f^* > -\infty$ . Then if  $\hat{x}_k \notin L_\gamma$ , we have

$$\begin{aligned} z_k &= \frac{2\alpha}{m} (f(\hat{x}_k) - f(y_\gamma)) - \alpha^2 c_0^2 \\ &\geq \frac{2\alpha}{m} \left( f^* + \frac{2}{\gamma} + \frac{\alpha m c_0^2}{2} - f^* - \frac{1}{\gamma} \right) - \alpha^2 c_0^2 \\ &= \frac{2\alpha}{m\gamma}. \end{aligned}$$

Hence,  $z_k \geq 0$  for all  $k$ , and by the Supermartingale Convergence Theorem, we have  $\sum_{k=0}^{\infty} z_k < \infty$  implying that  $\hat{x}_k \in L_\gamma$  for sufficiently large  $k$ , so that in the original process,

$$\inf_{k \geq 0} f(x_k) \leq f^* + \frac{2}{\gamma} + \frac{\alpha m c_0^2}{2}$$

with probability 1. Letting  $\gamma \rightarrow \infty$ , we obtain  $\inf_{k \geq 0} f(x_k) \leq f^* + \alpha m c_0^2 / 2$ .

**Q.E.D.**

From Prop. 6.8.4(b), it can be seen that when  $f^* > -\infty$ , the randomized method (6.156) with a fixed stepsize has a better error bound (by a factor  $m$ , since  $c^2 \approx m^2 c_0^2$ ) than the one of the nonrandomized method (6.148), (6.149), with the same stepsize (cf. Prop. 6.8.1). In effect, the randomized method achieves in an expected sense the error tolerance of the nonrandomized method with the best processing order [compare with the discussion in the preceding subsection and Eqs. (6.153) and (6.154)].

Thus when randomization is used, one can afford to use a larger stepsize  $\alpha$  than in the nonrandomized method. This suggests a rate of convergence advantage in favor of the randomized method.

A related result is provided by the following proposition, which parallels Prop. 6.8.2 for the nonrandomized method.

**Proposition 6.8.5:** Assume that the optimal set  $X^*$  is nonempty, let Assumption 6.8.2 hold, and let  $\{x_k\}$  be the sequence generated by the randomized incremental method (6.156). Then, for any positive scalar  $\epsilon$ , we have with probability 1

$$\min_{0 \leq k \leq N} f(x_k) \leq f^* + \frac{\alpha m c_0^2 + \epsilon}{2}, \quad (6.159)$$

where  $N$  is a random variable with

$$E\{N\} \leq \frac{m d(x_0)^2}{\alpha \epsilon}. \quad (6.160)$$

**Proof:** Define a new process  $\{\hat{x}_k\}$  by

$$\hat{x}_{k+1} = \begin{cases} P_X(\hat{x}_k - \alpha g(\omega_k, \hat{x}_k)) & \text{if } \hat{x}_k \notin L_\gamma, \\ y_\gamma & \text{otherwise,} \end{cases}$$

where  $\hat{x}_0 = x_0$  and  $\hat{y}$  is some fixed vector in  $X^*$ . The process  $\{\hat{x}_k\}$  is identical to  $\{x_k\}$ , except that once  $x_k$  enters the level set

$$L = \left\{ x \in X \mid f(x) < f^* + \frac{\alpha m c_0^2 + \epsilon}{2} \right\},$$

the process  $\{\hat{x}_k\}$  terminates at  $\hat{y}$ . Similar to the proof of Prop. 6.8.4 [cf. Eq. (6.157) with  $y \in X^*$ ], for the process  $\{\hat{x}_k\}$  we obtain for all  $k$ ,

$$\begin{aligned} E\{d(\hat{x}_{k+1})^2 \mid \mathcal{F}_k\} &\leq d(\hat{x}_k)^2 - \frac{2\alpha}{m}(f(\hat{x}_k) - f^*) + \alpha^2 c_0^2 \\ &= d(\hat{x}_k)^2 - z_k, \end{aligned} \quad (6.161)$$

where  $\mathcal{F}_k = \{x_0, \dots, x_k\}$  and

$$z_k = \begin{cases} \frac{2\alpha}{m}(f(\hat{x}_k) - f^*) - \alpha^2 c_0^2 & \text{if } \hat{x}_k \notin L, \\ 0 & \text{otherwise.} \end{cases}$$

In the case where  $\hat{x}_k \notin L$ , we have

$$z_k \geq \frac{2\alpha}{m} \left( f^* + \frac{\alpha m c_0^2 + \epsilon}{2} - f^* \right) - \alpha^2 c_0^2 = \frac{\alpha \epsilon}{m}. \quad (6.162)$$

By the Supermartingale Convergence Theorem (cf. Prop. 6.8.3), from Eq. (6.161) we have

$$\sum_{k=0}^{\infty} z_k < \infty$$

with probability 1, so that  $z_k = 0$  for all  $k \geq N$ , where  $N$  is a random variable. Hence  $\hat{x}_N \in L$  with probability 1, implying that in the original process we have

$$\min_{0 \leq k \leq N} f(x_k) \leq f^* + \frac{\alpha m c_0^2 + \epsilon}{2}$$

with probability 1. Furthermore, by taking the total expectation in Eq. (6.161), we obtain for all  $k$ ,

$$\begin{aligned} E\{d(\hat{x}_{k+1})^2\} &\leq E\{d(\hat{x}_k)^2\} - E\{z_k\} \\ &\leq d(x_0)^2 - E\left\{\sum_{j=0}^k z_j\right\}, \end{aligned}$$

where in the last inequality we use the facts  $\hat{x}_0 = x_0$  and

$$E\{d(x_0)^2\} = d(x_0)^2.$$

Therefore

$$d(x_0)^2 \geq E\left\{\sum_{k=0}^{\infty} z_k\right\} = E\left\{\sum_{k=0}^{N-1} z_k\right\} \geq E\left\{\frac{N\alpha\epsilon}{m}\right\} = \frac{\alpha\epsilon}{m} E\{N\},$$

where the last inequality above follows from Eq. (6.162). **Q.E.D.**

### Comparison of Deterministic and Randomized Methods

Let us now compare the estimate of the above proposition with the corresponding estimate for the deterministic incremental method. We showed in Prop. 6.8.2 that the deterministic method is guaranteed to reach the level set

$$\left\{x \mid f(x) \leq f^* + \frac{\alpha c^2 + \epsilon}{2}\right\}$$

after no more than  $d(x_0)^2/(\alpha\epsilon)$  cycles, where  $c = \sum_{i=1} c_i$ . To compare this estimate with the one for the randomized method, we note that  $c_i \leq c_0$ , so that  $c$  can be estimated as  $mc_0$ , while each cycle requires the processing of  $m$  component functions. Thus, the deterministic method, in order to reach the level set

$$\left\{x \mid f(x) \leq f^* + \frac{\alpha m^2 c_0^2 + \epsilon}{2}\right\},$$

it must process a total of

$$N \leq \frac{md(x_0)^2}{\alpha\epsilon}$$

component functions (this bound is essentially sharp, as shown in Exercise 6.17).

If in the randomized method we use the same stepsize  $\alpha$ , then according to Prop. 6.8.5, we will reach with probability 1 the (much smaller) level set

$$\left\{ x \mid f(x) \leq f^* + \frac{\alpha mc_0^2 + \epsilon}{2} \right\}$$

after processing  $N$  component functions, where the expected value of  $N$  satisfies

$$E\{N\} \leq \frac{md(x_0)^2}{\alpha\epsilon}.$$

Thus, for the same values of  $\alpha$  and  $\epsilon$ , the bound on the number of component functions that must be processed in the deterministic method is the same as the bound on the expected number of component functions that must be processed in the randomized method. However, the error term  $\alpha m^2 c_0^2$  in the deterministic method is  $m$  times larger than the corresponding error term in the randomized method. Similarly, if we choose the stepsize  $\alpha$  in the randomized method to achieve the same error level (in cost function value) as in the deterministic method, then the corresponding expected number of iterations becomes  $m$  times smaller. Practical computational experience appears to be consistent with this finding.

#### 6.8.4 Incremental Proximal Methods

We will now consider incremental variants of the proximal point algorithm, which like the incremental subgradient methods of the preceding section, apply to additive costs of the form

$$f(x) = \sum_{i=1}^m f_i(x),$$

where the functions  $f_i : \mathfrak{R}^n \mapsto \mathfrak{R}$  are convex. We wish to minimize  $f$  over a nonempty closed convex set  $X$ .

The idea is to take proximal steps using single component functions  $f_i$ , with intermediate adjustment of the proximal center. In particular, we view an iteration as a cycle of  $m$  subiterations. If  $x_k$  is the vector obtained after  $k$  cycles, the vector  $x_{k+1}$  obtained after one more cycle is

$$x_{k+1} = \psi_{m,k}, \tag{6.163}$$

where starting with  $\psi_{0,k} = x_k$ , we obtain  $\psi_{m,k}$  after the  $m$  proximal steps

$$\psi_{i,k} = \arg \min_{x \in X} \left\{ f_i(x) + \frac{1}{2\alpha_k} \|x - \psi_{i-1,k}\|^2 \right\}, \quad i = 1, \dots, m, \tag{6.164}$$

where  $\alpha_k$  is a positive parameter.

We will discuss shortly schemes to adjust  $\alpha_k$ , and we will see that it plays a role analogous to the stepsize in incremental subgradient methods (see the subsequent Prop. 6.8.6). In this connection, we note that for convergence of the method, it is essential that  $\alpha_k \rightarrow 0$ , as illustrated in the following example.

**Example 6.8.2:**

Consider the unconstrained scalar case ( $X = \mathfrak{R}$ ) and the cost function

$$|x| + |x - 1| + |x + 1|.$$

Then starting at  $x_0 = 0$  and  $\alpha_k \equiv \alpha$ , with  $\alpha \in (0, 1]$ , the method takes the form

$$\begin{aligned} \psi_{1,k} &= \arg \min_{x \in \mathfrak{R}} \left\{ |x| + \frac{1}{2\alpha} |x - x_k|^2 \right\}, \\ \psi_{2,k} &= \arg \min_{x \in \mathfrak{R}} \left\{ |x - 1| + \frac{1}{2\alpha} |x - \psi_{1,k}|^2 \right\}, \\ \psi_{3,k} = x_{k+1} &= \arg \min_{x \in \mathfrak{R}} \left\{ |x + 1| + \frac{1}{2\alpha} |x - \psi_{2,k}|^2 \right\}, \end{aligned}$$

and generates the sequences  $\psi_{1,k} = 0$ ,  $\psi_{2,k} = \alpha$ ,  $\psi_{3,k} = x_k = 0$ ,  $k = 0, 1, \dots$ . Thus, starting at the optimal solution  $x_0 = 0$  and using a constant parameter  $\alpha_k \equiv \alpha$ , the method oscillates proportionately to  $\alpha$ .

The following proposition suggests the similarity between incremental proximal and subgradient methods.

**Proposition 6.8.6:** The incremental proximal iteration (6.164) can be written as

$$\psi_{i,k} = P_X(\psi_{i-1,k} - \alpha_k g_{i,k}), \quad i = 1, \dots, m, \quad (6.165)$$

where  $g_{i,k}$  is some subgradient of  $f_i$  at  $\psi_{i,k}$ .

**Proof:** We use the formula for the subdifferential of the sum of the three functions  $f_i$ ,  $(1/2\alpha_k)\|x - \psi_{i-1,k}\|^2$ , and the indicator function of  $X$  (Prop. 5.4.6), together with the condition that 0 should belong to this subdifferential at the optimum  $\psi_{i,k}$ . We obtain that

$$\psi_{i,k} = \arg \min_{x \in X} \left\{ f_i(x) + \frac{1}{2\alpha_k} \|x - \psi_{i-1,k}\|^2 \right\}$$

if and only if

$$\frac{1}{\alpha_k}(\psi_{i-1,k} - \psi_{i,k}) \in \partial f_i(\psi_{i,k}) + N_X(\psi_{i,k}),$$

where  $N_X(\psi_{i,k})$  is the normal cone of  $X$  at  $\psi_{i,k}$ . This is true if and only if

$$\psi_{i-1,k} - \psi_{i,k} - \alpha_k g_{i,k} \in N_X(\psi_{i,k}),$$

for some  $g_{i,k} \in \partial f_i(\psi_{i,k})$ , which in turn is true if and only if Eq. (6.165) holds (cf. Prop. 5.4.7). **Q.E.D.**

Note the difference between the incremental subgradient and proximal iterations. In the former case *any* subgradient of  $f_i$  at  $\psi_{i,k-1}$  is used, while in the latter case a *particular* subgradient at the *next*  $\psi_{i,k}$  is used. It turns out that for convergence purposes this difference is relatively inconsequential: we will show that much of the analysis of the preceding section for incremental subgradient methods carries through with suitable modifications to the incremental proximal case. To this end, we provide the following analog of the crucial Lemma 6.8.1. We denote

$$c_i = \sup_{k \geq 0} \max \{ \|\hat{g}_{i,k}\|, \|g_{i,k}\| \}, \quad i = 1, \dots, m, \quad (6.166)$$

where  $\hat{g}_{i,k}$  is the subgradient of minimum norm in  $\partial f_i(x_k)$  and  $g_{i,k}$  is the subgradient of Eq. (6.165). In the following proposition the scalars  $c_i$  are assumed finite (this replaces Assumption 6.8.1).

**Lemma 6.8.2:** Let  $\{x_k\}$  be the sequence generated by the incremental proximal method (6.163), (6.164). Then for all  $y \in X$  and  $k \geq 0$ , we have

$$\|x_{k+1} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k(f(x_{k+1}) - f(y)) + \alpha_k^2 c^2, \quad (6.167)$$

where  $c = \sum_{i=1}^m c_i$  is assumed finite.

**Proof:** Using the nonexpansion property of the projection, and the subgradient inequality for each component function  $f_i$ , we obtain for all  $y \in X$ ,

$i = 1, \dots, m$ , and  $k \geq 0$ ,

$$\begin{aligned}
 \|\psi_{i,k} - y\|^2 &= \|P_X(\psi_{i-1,k} - \alpha_k g_{i,k}) - y\|^2 \\
 &\leq \|\psi_{i-1,k} - \alpha_k g_{i,k} - y\|^2 \\
 &\leq \|\psi_{i-1,k} - y\|^2 - 2\alpha_k g'_{i,k}(\psi_{i-1,k} - y) + \alpha_k^2 \|g_{i,k}\|^2 \\
 &\leq \|\psi_{i-1,k} - y\|^2 - 2\alpha_k g'_{i,k}(\psi_{i,k} - y) + \alpha_k^2 c_i^2 \\
 &\quad + 2\alpha_k g'_{i,k}(\psi_{i,k} - \psi_{i-1,k}) \\
 &\leq \|\psi_{i-1,k} - y\|^2 - 2\alpha_k (f_i(\psi_{i,k}) - f_i(y)) + \alpha_k^2 c_i^2 \\
 &\quad + 2\alpha_k g'_{i,k}(\psi_{i,k} - \psi_{i-1,k}).
 \end{aligned}$$

We note that since  $\psi_{i,k}$  is the projection on  $X$  of  $\psi_{i-1,k} - \alpha_k g_{i,k}$ , we have

$$g'_{i,k}(\psi_{i,k} - \psi_{i-1,k}) \leq 0,$$

which combined with the preceding inequality yields

$$\|\psi_{i,k} - y\|^2 \leq \|\psi_{i-1,k} - y\|^2 - 2\alpha_k (f_i(\psi_{i,k}) - f_i(y)) + \alpha_k^2 c_i^2, \quad \forall i, k.$$

By adding the above inequalities over  $i = 1, \dots, m$ , we have for all  $y \in X$  and  $k$ ,

$$\begin{aligned}
 \|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2\alpha_k \sum_{i=1}^m (f_i(\psi_{i,k}) - f_i(y)) + \alpha_k^2 \sum_{i=1}^m c_i^2 \\
 &= \|x_k - y\|^2 - 2\alpha_k \left( f(x_{k+1}) - f(y) + \sum_{i=1}^m (f_i(\psi_{i,k}) - f_i(x_{k+1})) \right) \\
 &\quad + \alpha_k^2 \sum_{i=1}^m c_i^2.
 \end{aligned}$$

By strengthening the above inequality, we have for all  $y \in X$  and  $k$ , using also the fact  $\psi_{m,k} = x_{k+1}$ ,

$$\begin{aligned}
 \|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2\alpha_k (f(x_{k+1}) - f(y)) \\
 &\quad + 2\alpha_k \sum_{i=1}^m c_i \|\psi_{i,k} - x_{k+1}\| + \alpha_k^2 \sum_{i=1}^m c_i^2 \\
 &\leq \|x_k - y\|^2 - 2\alpha_k (f(x_{k+1}) - f(y)) \\
 &\quad + \alpha_k^2 \left( 2 \sum_{i=1}^{m-1} c_i \left( \sum_{j=i+1}^m c_j \right) + \sum_{i=1}^m c_i^2 \right) \\
 &= \|x_k - y\|^2 - 2\alpha_k (f(x_{k+1}) - f(y)) + \alpha_k^2 \left( \sum_{i=1}^m c_i \right)^2 \\
 &= \|x_k - y\|^2 - 2\alpha_k (f(x_{k+1}) - f(y)) + \alpha_k^2 c^2,
 \end{aligned}$$



where in the first inequality we use the relation

$$f_i(x_{k+1}) - f_i(\psi_{i,k}) \leq \|\hat{g}_{i,k+1}\| \cdot \|\psi_{i,k} - x_{k+1}\| \leq c_i \|\psi_{i,k} - x_{k+1}\|,$$

which follows from the subgradient inequality and the definition (6.166) of  $c_i$ , while in the second inequality we use the relation

$$\|\psi_{i,k} - x_{k+1}\| \leq \alpha_k \sum_{j=i+1}^m \|g_{j,k}\| \leq \alpha_k \sum_{j=i+1}^m c_j, \quad i = 1, \dots, m, \quad k \geq 0,$$

which follows from Eqs. (6.163)-(6.165), and the definition (6.166) of  $c_i$ . **Q.E.D.**

Among other things, Lemma 6.8.2 guarantees that given the current iterate  $x_k$  and some other point  $y \in X$  having lower cost than  $x_k$ , the next iterate  $x_{k+1}$  will be closer to  $y$  than  $x_k$ , provided the stepsize  $\alpha_k$  is sufficiently small [less than  $2(f(x_{k+1}) - f(y))/c^2$ ]. In particular, for any optimal solution  $x^* \in X^*$ , any  $\epsilon > 0$ , and any  $\alpha_k \leq \epsilon/c^2$ , either

$$f(x_{k+1}) \leq f^* + \epsilon,$$

or else

$$\|x_{k+1} - x^*\| < \|x_k - x^*\|.$$

Using Lemma 6.8.2, we can provide convergence results for the incremental proximal method, which parallels the corresponding results for the incremental subgradient method. For a constant stepsize,  $\alpha_k \equiv \alpha$ , convergence can be established to a neighborhood of the optimum, which shrinks to 0 as  $\alpha \rightarrow 0$ . In particular, Prop. 6.8.1 holds verbatim, and the convergence rate estimate of Prop. 6.8.2 also holds. Furthermore, a convergence result for a diminishing stepsize, which parallels Prop. 6.3.4 can also be similarly established. Also, randomization in the order of selecting the components  $f_i$  in the proximal iteration can be introduced, with an analysis that parallels the one of Section 6.8.3.

We finally note that while some cost function components may be well suited for a proximal iteration, others may not be because the minimization in Eq. (6.164) is inconvenient. With this in mind, we may consider combinations of subgradient and proximal methods for problems of the form

$$\begin{aligned} \text{minimize} \quad & F(x) \stackrel{\text{def}}{=} \sum_{i=1}^m F_i(x) \\ \text{subject to} \quad & x \in X, \end{aligned}$$

where for all  $i$ ,

$$F_i(x) = f_i(x) + h_i(x),$$

$f_i : \mathfrak{R}^n \mapsto \mathfrak{R}$  and  $h_i : \mathfrak{R}^n \mapsto \mathfrak{R}$  are real-valued convex functions, and  $X$  is a nonempty closed convex set. An incremental algorithm for this problem may iterate on the components  $f_i$  with a proximal iteration, and on the components  $h_i$  with a subgradient iteration. By choosing all the  $f_i$  or all the  $h_i$  to be identically zero, we obtain as special cases the subgradient and proximal iterations, respectively.

The convergence and rate of convergence properties of such combined methods have been analyzed in Bertsekas [Ber10a] and [Ber10b], where it is shown that the convergence behavior is similar to the one described earlier for the incremental subgradient method. This includes convergence within a certain error bound for a constant stepsize, exact convergence to an optimal solution for an appropriately diminishing stepsize, and improved convergence rate/iteration complexity when randomization is used to select the cost component for iteration. However, the combined incremental subgradient/proximal methods offer greater flexibility, and may exploit the special structure of problems where the functions  $f_i$  are suitable for a proximal iteration, while the components  $h_i$  are not and thus may be preferably treated with a subgradient iteration.

## 6.9 OPTIMAL ALGORITHMS AND COMPLEXITY

In this section we focus on some computational complexity issues relating to optimization problems of the form

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && x \in X, \end{aligned}$$

where  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  is convex and  $X$  is a closed convex set. We denote by  $f^*$  the optimal value, and we assume throughout that it is finite. In this section, we will discuss algorithms that have good performance guarantees, in the sense that they require a relatively low number of iterations to achieve a given optimal solution tolerance.

Given some  $\epsilon > 0$ , suppose we want to estimate the number of iterations required by a particular algorithm to obtain a solution with cost that is within  $\epsilon$  of the optimal. If we can show that a sequence  $\{x_k\}$  generated by a method has the property that for any  $\epsilon > 0$ , we have

$$\inf_{k \leq N(\epsilon)} f(x_k) \leq f^* + \epsilon,$$

where  $N(\epsilon)$  is a function that depends on  $\epsilon$ , as well as the problem data and the starting point  $x_0$ , we say that the method has *iteration complexity*  $O(N(\epsilon))$ .

It is generally thought that if  $N(\epsilon)$  does not depend on the dimension  $n$  of the problem, then the algorithm holds an advantage for problems of large dimension. This view favors simple gradient/subgradient-like methods over sophisticated Newton-like methods whose overhead per iteration increases fast with  $n$ . In this section, we will focus on algorithms with iteration complexity that is independent of  $n$ , and all our subsequent references to complexity estimates implicitly assume this.

As an example, we mention the subgradient method for which an  $O(1/\epsilon^2)$  iteration complexity result can be shown (cf., the discussion following Prop. 6.8.2). It turns out that when applied to *differentiable* problems, the subgradient method has better performance guarantees. For unconstrained problems ( $X = \Re^n$ ) with Lipschitz continuous cost gradient, its iteration complexity can be estimated as  $O(1/\epsilon)$ , as we will show shortly. Furthermore, we will discuss a variant that employs an intricate extrapolation device, and has the even better iteration complexity of  $O(1/\sqrt{\epsilon})$ , for both unconstrained and constrained problems. It can be shown that  $O(1/\sqrt{\epsilon})$  is a sharp estimate, i.e., it is the best that we can expect across the class of problems with convex cost functions with Lipschitz continuous gradient (see the end-of-chapter references).

We finally note that the algorithms of this section assume just Lipschitz continuity of the gradient of  $f$ , and ignore any other special structure, such as when  $f$  is the sum of a large number of components. Thus, for special classes of problems, they may not be competitive with methods that exploit structure, such as the incremental methods of the preceding section.

### Gradient Projection Methods for Differentiable Minimization

Let  $f : \Re^n \mapsto \Re$  be a differentiable convex function that we want to minimize over a closed convex set  $X$ . We assume that  $f$  has Lipschitz continuous gradient, i.e., for some constant  $L$ ,

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|, \quad \forall x, y \in X. \quad (6.168)$$

We will consider the gradient projection method

$$x_{k+1} = P_X(x_k - \alpha \nabla f(x_k)), \quad (6.169)$$

where  $\alpha > 0$  is a constant stepsize. This is the specialization of the subgradient method of Section 6.3, for the case where  $f$  is differentiable. We will show that the method has the convergence property  $f(x_k) \rightarrow f^*$  for any starting point  $x_0$ , provided  $\alpha$  is sufficiently small. Furthermore, its iteration complexity is  $O(1/\epsilon)$ . To this end, we will need the following lemma:

**Lemma 6.9.1:** Let  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  be a continuously differentiable function, with gradient satisfying the Lipschitz condition (6.168). Then for all  $x, y \in \mathfrak{R}^n$ , we have

$$f(y) - f(x) - \nabla f(x)'(y - x) \leq \frac{L}{2} \|y - x\|^2, \quad (6.170)$$

**Proof:** Let  $t$  be a scalar parameter and let  $g(t) = f(x + t(y - x))$ . The chain rule yields  $(dg/dt)(t) = \nabla f(x + t(y - x))'(y - x)$ . Thus, we have

$$\begin{aligned} f(y) - f(x) &= g(1) - g(0) \\ &= \int_0^1 \frac{dg}{dt}(t) dt \\ &= \int_0^1 y' \nabla f(x + t(y - x)) dt \\ &\leq \int_0^1 y' \nabla f(x) dt + \left| \int_0^1 (y - x)' (\nabla f(x + t(y - x)) - \nabla f(x)) dt \right| \\ &\leq \int_0^1 y' \nabla f(x) dt + \int_0^1 \|y - x\| \cdot \|\nabla f(x + t(y - x)) - \nabla f(x)\| dt \\ &\leq y' \nabla f(x) + \|y - x\| \int_0^1 Lt \|y - x\| dt \\ &= y' \nabla f(x) + \frac{L}{2} \|y - x\|^2 \end{aligned}$$

thereby proving Eq. (6.170). **Q.E.D.**

From the projection theorem and the definition (6.169) of  $x_{k+1}$ , we have

$$(x_k - \alpha \nabla f(x_k) - x_{k+1})'(x_k - x_{k+1}) \leq 0,$$

so that

$$\nabla f(x_k)'(x_{k+1} - x_k) \leq -\frac{1}{\alpha} \|x_{k+1} - x_k\|^2.$$

Using this relation together with Eq. (6.170), we have

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \nabla f(x_k)'(x_{k+1} - x_k) + \frac{L}{2} \|x_{k+1} - x_k\|^2 \\ &\leq f(x_k) - \left( \frac{1}{\alpha} - \frac{L}{2} \right) \|x_{k+1} - x_k\|^2, \end{aligned} \quad (6.171)$$

so the gradient projection method (6.169) reduces the cost function value at each iteration, provided the stepsize lies in the range  $\alpha \in (0, \frac{2}{L})$ .

We can now prove our main convergence rate estimate for differentiable functions. Let  $X^*$  be the set of unconstrained minima of  $f$ , and let

$$d(x) = \inf_{x^* \in X^*} \|x - x^*\|, \quad x \in \mathfrak{R}^n.$$

**Proposition 6.9.1:** Let  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  be a convex differentiable function and  $X$  be a closed convex set. Assume that  $\nabla f$  satisfies the Lipschitz condition (6.168), and that the set of minima  $X^*$  of  $f$  over  $\mathfrak{R}^n$  is nonempty. Let  $\{x_k\}$  be a sequence generated by the gradient projection method (6.169) with stepsize  $\alpha = 1/L$ . Then  $\lim_{k \rightarrow \infty} d(x_k) = 0$ , and

$$f(x_k) - f^* \leq \frac{Ld(x_0)^2}{2k}, \quad k = 1, 2, \dots$$

**Proof:** Let us denote

$$\ell(u; w) = f(w) + \nabla f(w)'(u - w), \quad \forall u, w \in \mathfrak{R}^n.$$

Using Eq. (6.170), we have

$$f(x_{k+1}) \leq \ell(x_{k+1}; x_k) + \frac{L}{2} \|x_{k+1} - x_k\|^2. \quad (6.172)$$

From the result of Exercise 6.19(c), we have for all  $x \in X$

$$\ell(x_{k+1}; x_k) + \frac{L}{2} \|x_{k+1} - x_k\|^2 \leq \ell(x; x_k) + \frac{L}{2} \|x - x_k\|^2 - \frac{L}{2} \|x - x_{k+1}\|^2.$$

Thus, letting  $x = x^*$ , where  $x^* \in X^*$  satisfies  $\|x_0 - x^*\| = d(x_0)$ , we obtain using also Eq. (6.172),

$$\begin{aligned} f(x_{k+1}) &\leq \ell(x_{k+1}; x_k) + \frac{L}{2} \|x_{k+1} - x_k\|^2 \\ &\leq \ell(x^*; x_k) + \frac{L}{2} \|x^* - x_k\|^2 - \frac{L}{2} \|x^* - x_{k+1}\|^2 \\ &\leq f(x^*) + \frac{L}{2} \|x^* - x_k\|^2 - \frac{L}{2} \|x^* - x_{k+1}\|^2, \end{aligned}$$

so denoting  $e_k = f(x_k) - f^*$ , we have

$$\begin{aligned} \frac{L}{2} \|x^* - x_{k+1}\|^2 &\leq \frac{L}{2} \|x^* - x_k\|^2 - e_{k+1} \\ &\leq \frac{L}{2} \|x^* - x_0\|^2 - (e_1 + \dots + e_{k+1}) \\ &\leq \frac{L}{2} d(x_0)^2 - (k+1)e_{k+1}, \end{aligned}$$

where the last inequality uses the fact  $e_0 \geq e_1 \geq \dots \geq e_{k+1}$  [cf. Eq. (6.171)].  
**Q.E.D.**

Note that there is a procedure for selecting the stepsize  $\alpha$  to attain the same iteration complexity, even if the value of  $L$  is unknown. In particular, we may use some arbitrary trial stepsize  $\alpha > 0$ , and generate iterates according to

$$x_{k+1} = P_X(x_k - \alpha \nabla f(x_k)),$$

as long as the condition

$$f(x_{k+1}) \leq \ell(x_{k+1}; x_k) + \frac{1}{2\alpha} \|x_{k+1} - x_k\|^2 \quad (6.173)$$

is satisfied [cf. Eq. (6.172)]. As soon as this condition is violated at some iteration, we reduce  $\alpha$  by a certain factor, and repeat the iteration as many times as is necessary for Eq. (6.173) to hold. Then it can be seen that the condition (6.173) will be satisfied if  $\alpha \leq 1/L$ . Thus, once  $\alpha$  is reduced below the level  $1/L$ , the test (6.173) will be passed, and no further reductions will be necessary. Furthermore, the preceding proof, with slight modification to account for the finite number of reductions of  $\alpha$ , shows that the algorithm has  $O(1/\epsilon)$  iteration complexity.

The following example shows that the complexity estimate  $O(1/\epsilon)$  cannot be improved.

**Example 6.9.1:**

Consider the unconstrained minimization of the scalar function  $f$  given by

$$f(x) = \begin{cases} \frac{c}{2}|x|^2 & \text{if } |x| \leq \epsilon, \\ c\epsilon|x| - \frac{c\epsilon^2}{2} & \text{if } |x| > \epsilon, \end{cases}$$

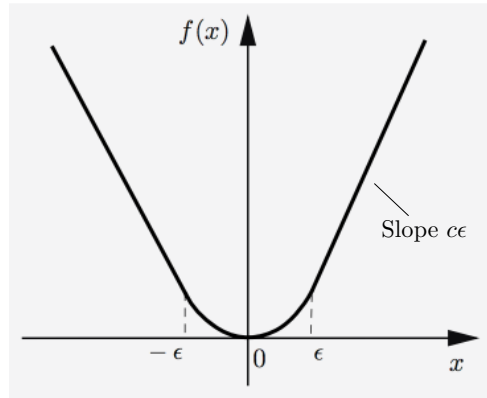
where  $\epsilon \in (0, 1)$  (cf. Fig. 6.9.1). Here the constant in the Lipschitz condition (6.168) is  $L = c$ , and for any  $x_k > \epsilon$ , the gradient iteration with stepsize  $\alpha = 1/L$  takes the form

$$x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k) = x_k - \frac{1}{c} c \epsilon = x_k - \epsilon.$$

Thus, the number of iterations to get within an  $\epsilon$ -neighborhood of  $x^* = 0$  is  $|x_0|/\epsilon$ . The number of iterations to get to within  $\epsilon$  of the optimal cost  $f^* = 0$  is also proportional to  $1/\epsilon$ .

A closer examination of the preceding example suggests that while a stepsize less than  $2/c$  is necessary within the region where  $|x| \leq \epsilon$  to ensure that the method converges, a larger stepsize outside this region would accelerate convergence. An acceleration scheme, known as the *heavy-ball* method or gradient method with *momentum*, has the form

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}),$$



**Figure 6.9.1.** The scalar cost function  $f$  of Example 6.9.1. It is quadratic for  $|x| \leq \epsilon$  and linear for  $|x| > \epsilon$ .

and adds the extrapolation term  $\beta(x_k - x_{k-1})$  to the gradient increment, where  $x_{-1} = x_0$  and  $\beta$  is a scalar with  $0 < \beta < 1$  (see the end-of-chapter references). A variant of this scheme with similar properties separates the extrapolation and the gradient steps:

$$\begin{aligned} y_k &= x_k + \beta(x_k - x_{k-1}), && \text{(extrapolation step),} \\ x_{k+1} &= y_k - \alpha \nabla f(y_k), && \text{(gradient step).} \end{aligned} \quad (6.174)$$

When applied to the function of the preceding example, the method converges to the optimum, and reaches a neighborhood of the optimum more quickly: it can be verified that for a starting point  $x_0 \gg 1$  and  $x_k > \epsilon$ , it has the form  $x_{k+1} = x_k - \epsilon_k$ , with  $0 < \epsilon_k < \epsilon/(1 - \beta)$ . However, the method still has an  $O(1/\epsilon)$  iteration complexity, since for  $x_0 \gg 1$ , the number of iterations needed to obtain  $x_k < \epsilon$  is  $O((1 - \beta)/\epsilon)$ .

It turns out that a better iteration complexity is possible with more vigorous extrapolation. We will show next that what is needed is to replace the constant extrapolation factor  $\beta$  with a variable factor  $\beta_k$  that converges to 1 at a properly selected rate.

### Gradient Projection Method with Extrapolation

We will consider a constrained version of the gradient/extrapolation method (6.174) for the problem

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in X, \end{aligned} \quad (6.175)$$

where  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  is convex and differentiable, and  $X$  is a closed convex set. We assume that  $f$  has Lipschitz continuous gradient [cf. Eq. (6.168)],

and we denote

$$d(x) = \inf_{x^* \in X^*} \|x - x^*\|, \quad x \in \mathfrak{R}^n,$$

where  $X^*$  is the set of minima of  $f$  over  $X$ .

The method has the form

$$\begin{aligned} y_k &= x_k + \beta_k(x_k - x_{k-1}), & (\text{extrapolation step}), \\ x_{k+1} &= P_X(y_k - \alpha \nabla f(y_k)), & (\text{gradient projection step}), \end{aligned} \quad (6.176)$$

where  $P_X(\cdot)$  denotes projection on  $X$ ,  $x_{-1} = x_0$ , and  $\beta_k \in (0, 1)$ . The following proposition shows that with proper choice of  $\beta_k$ , the method has iteration complexity  $O(1/\sqrt{\epsilon})$ . We will assume that

$$\beta_k = \frac{\theta_k(1 - \theta_{k-1})}{\theta_{k-1}}, \quad k = 0, 1, \dots \quad (6.177)$$

where the sequence  $\{\theta_k\}$  satisfies  $\theta_0 = \theta_1 \in (0, 1]$ , and

$$\frac{1 - \theta_{k+1}}{\theta_{k+1}^2} \leq \frac{1}{\theta_k^2}, \quad \theta_k \leq \frac{2}{k+2}, \quad k = 0, 1, \dots \quad (6.178)$$

One possible choice is

$$\beta_k = \begin{cases} 0 & \text{if } k = 0, \\ \frac{k-1}{k+2} & \text{if } k = 1, 2, \dots, \end{cases} \quad \theta_k = \begin{cases} 1 & \text{if } k = -1, \\ \frac{2}{k+2} & \text{if } k = 0, 1, \dots \end{cases}$$

We will also assume a stepsize  $\alpha = 1/L$ , and we will show later how the proof can be extended for the case where the constant  $L$  is not known.

**Proposition 6.9.2:** Let  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  be a convex differentiable function and  $X$  be a closed convex set. Assume that  $\nabla f$  satisfies the Lipschitz condition (6.168), and that the set of minima  $X^*$  of  $f$  over  $X$  is nonempty. Let  $\{x_k\}$  be a sequence generated by the algorithm (6.176), where  $\alpha = 1/L$  and  $\beta_k$  satisfies Eqs. (6.177)-(6.178). Then  $\lim_{k \rightarrow \infty} d(x_k) = 0$ , and

$$f(x_k) - f^* \leq \frac{2L}{(k+1)^2} d(x_0)^2, \quad k = 1, 2, \dots$$

**Proof:** We introduce the sequence

$$z_k = x_{k-1} + \theta_{k-1}^{-1}(x_k - x_{k-1}), \quad k = 0, 1, \dots, \quad (6.179)$$



where  $x_{-1} = x_0$ , so that  $z_0 = x_0$ . We note that by using Eqs. (6.176), (6.177),  $z_k$  can also be rewritten as

$$z_k = x_k + \theta_k^{-1}(y_k - x_k), \quad k = 1, 2, \dots, \quad (6.180)$$

Fix  $k \geq 0$  and  $x^* \in X^*$ , and let

$$y^* = (1 - \theta_k)x_k + \theta_k x^*.$$

Using Eq. (6.170), we have

$$f(x_{k+1}) \leq \ell(x_{k+1}; y_k) + \frac{L}{2} \|x_{k+1} - y_k\|^2, \quad (6.181)$$

where we use the notation

$$\ell(u; w) = f(w) + \nabla f(w)'(u - w), \quad \forall u, w \in \mathfrak{R}^n.$$

Since  $x_{k+1}$  is the projection of  $y_k - (1/L)\nabla f(y_k)$  on  $X$ , it minimizes

$$\ell(y; y_k) + \frac{L}{2} \|y - y_k\|^2$$

over  $y \in X$ , so from the result of Exercise 6.19(c), we have

$$\ell(x_{k+1}; y_k) + \frac{L}{2} \|x_{k+1} - y_k\|^2 \leq \ell(y^*; y_k) + \frac{L}{2} \|y^* - y_k\|^2 - \frac{L}{2} \|y^* - x_{k+1}\|^2.$$

Combining this relation with Eq. (6.181), we obtain

$$\begin{aligned} f(x_{k+1}) &\leq \ell(y^*; y_k) + \frac{L}{2} \|y^* - y_k\|^2 - \frac{L}{2} \|y^* - x_{k+1}\|^2 \\ &= \ell((1 - \theta_k)x_k + \theta_k x^*; y_k) + \frac{L}{2} \|(1 - \theta_k)x_k + \theta_k x^* - y_k\|^2 \\ &\quad - \frac{L}{2} \|(1 - \theta_k)x_k + \theta_k x^* - x_{k+1}\|^2 \\ &= \ell((1 - \theta_k)x_k + \theta_k x^*; y_k) + \frac{\theta_k^2 L}{2} \|x^* + \theta_k^{-1}(x_k - y_k) - x_k\|^2 \\ &\quad - \frac{\theta_k^2 L}{2} \|x^* + \theta_k^{-1}(x_k - x_{k+1}) - x_k\|^2 \\ &= \ell((1 - \theta_k)x_k + \theta_k x^*; y_k) + \frac{\theta_k^2 L}{2} \|x^* - z_k\|^2 \\ &\quad - \frac{\theta_k^2 L}{2} \|x^* - z_{k+1}\|^2 \\ &\leq (1 - \theta_k)\ell(x_k; y_k) + \theta_k \ell(x^*; y_k) + \frac{\theta_k^2 L}{2} \|x^* - z_k\|^2 \\ &\quad - \frac{\theta_k^2 L}{2} \|x^* - z_{k+1}\|^2, \end{aligned}$$

where the last equality follows from Eqs. (6.179) and (6.180), and the last inequality follows from the convexity of  $\ell(\cdot; y_k)$ . Using the inequality

$$\ell(x_k; y_k) \leq f(x_k),$$

we have

$$f(x_{k+1}) \leq (1 - \theta_k)f(x_k) + \theta_k \ell(x^*; y_k) + \frac{\theta_k^2 L}{2} \|x^* - z_k\|^2 - \frac{\theta_k^2 L}{2} \|x^* - z_{k+1}\|^2.$$

Finally, by rearranging terms, we obtain

$$\begin{aligned} \frac{1}{\theta_k^2} (f(x_{k+1}) - f^*) + \frac{L}{2} \|x^* - z_{k+1}\|^2 \\ \leq \frac{1 - \theta_k}{\theta_k^2} (f(x_k) - f^*) + \frac{L}{2} \|x^* - z_k\|^2 - \frac{f^* - \ell(x^*; y_k)}{\theta_k}. \end{aligned}$$

By adding this inequality for  $k = 0, 1, \dots$ , while using the inequality

$$\frac{1 - \theta_{k+1}}{\theta_{k+1}^2} \leq \frac{1}{\theta_k^2},$$

we obtain

$$\frac{1}{\theta_k^2} (f(x_{k+1}) - f^*) + \sum_{i=0}^k \frac{f^* - \ell(x^*; y_i)}{\theta_i} \leq \frac{L}{2} \|x^* - z_0\|^2.$$

Using the facts  $x_0 = z_0$ ,  $f^* - \ell(x^*; y_i) \geq 0$ , and  $\theta_k \leq 2/(k+2)$ , and taking the minimum over all  $x^* \in X^*$ , we obtain

$$f(x_{k+1}) - f^* \leq \frac{2L}{(k+2)^2} d(x_0)^2,$$

from which the desired result follows. **Q.E.D.**

We note a variation of the algorithm that does not require knowledge of  $L$ . Similar, to the case of the gradient projection method (without extrapolation), this variant uses some arbitrary trial stepsize  $\alpha > 0$ , as long as the condition

$$f(x_{k+1}) \leq \ell(x_{k+1}; y_k) + \frac{1}{2\alpha} \|x_{k+1} - y_k\|^2, \quad (6.182)$$

[cf. Eq. (6.181)] is satisfied. As soon as this condition is violated at some iteration,  $\alpha$  is reduced by a certain factor, and the iteration is repeated as many times as necessary for Eq. (6.182) to hold. Once  $\alpha$  is reduced below the level  $1/L$ , the test (6.182) will be passed, and no further reductions will be necessary. The preceding proof can then be modified to show that the variant has iteration complexity  $O(\sqrt{L/\epsilon})$ .

### Nondifferentiable Cost – Smoothing

The preceding analysis applies to differentiable cost functions. However, it can be applied to the case where  $f$  is convex but nondifferentiable by using a smoothing technique to convert the nondifferentiable problem to a differentiable one. In this way an iteration complexity of  $O(1/\epsilon)$  can be attained, which is much faster than the  $O(1/\epsilon^2)$  complexity of the subgradient method. The idea is to replace a nondifferentiable convex cost function by a smooth  $\epsilon$ -approximation whose gradient is Lipschitz continuous with constant  $L = O(1/\epsilon)$ . By applying the optimal method given earlier, we obtain an  $\epsilon$ -optimal solution with iteration complexity  $O(\sqrt{L/\epsilon}) = O(1/\epsilon)$ .

We will consider the smoothing technique for the class of convex functions  $f_0 : \mathfrak{R}^n \mapsto \mathfrak{R}$  of the form

$$f_0(x) = \max_{u \in U} \{u'Ax - \phi(u)\}, \quad (6.183)$$

where  $U$  is a convex and compact subset of  $\mathfrak{R}^m$ ,  $\phi : U \mapsto \mathfrak{R}$  is convex and continuous over  $U$ , and  $A$  is an  $m \times n$  matrix. Note that  $f_0$  is just the composition of the matrix  $A$  and the conjugate function of

$$\tilde{\phi}(u) = \begin{cases} \phi(u) & \text{if } u \in U, \\ \infty & \text{if } u \notin U, \end{cases}$$

so the class of convex functions  $f_0$  of the form (6.183) is quite broad. We introduce a function  $p : \mathfrak{R}^m \mapsto \mathfrak{R}$  that is strictly convex and differentiable. Let  $u_0$  be the unique minimum of  $p$  over  $U$ , i.e.,

$$u_0 = \arg \min_{u \in U} p(u)$$

We assume that  $p(u_0) = 0$  and that  $p$  is strongly convex over  $U$  with modulus of strong convexity  $\sigma$ , i.e., that

$$p(u) \geq \frac{\sigma}{2} \|u - u_0\|^2$$

(cf. the Exercises of Chapter 1). An example is the quadratic function  $p(u) = \frac{\sigma}{2} \|u - u_0\|^2$ , but there are also other functions of interest (see the paper by Nesterov [Nes05] for some other examples, which also allow  $p$  to be nondifferentiable and to be defined only on  $U$ ).

For a parameter  $\epsilon > 0$ , consider the function

$$f_\epsilon(x) = \max_{u \in U} \{u'Ax - \phi(u) - \epsilon p(u)\}, \quad x \in \mathfrak{R}^n, \quad (6.184)$$

and note that  $f_\epsilon$  is a uniform approximation of  $f_0$  in the sense that

$$f_\epsilon(x) \leq f_0(x) \leq f_\epsilon(x) + p^*\epsilon, \quad \forall x \in \mathfrak{R}^n, \quad (6.185)$$

where

$$p^* = \max_{u \in U} p(u).$$

The following proposition shows that  $f_\epsilon$  is also smooth and its gradient is Lipschitz continuous with Lipschitz constant that is proportional to  $1/\epsilon$ .

**Proposition 6.9.3:** For all  $\epsilon > 0$ , the function  $f_\epsilon$  is convex and differentiable over  $\mathfrak{R}^n$ , with gradient given by

$$\nabla f_\epsilon(x) = A'u_\epsilon(x),$$

where  $u_\epsilon(x)$  is the unique vector attaining the maximum in Eq. (6.184). Furthermore, we have

$$\|\nabla f_\epsilon(x) - \nabla f_\epsilon(y)\| \leq \frac{\|A\|^2}{\epsilon\sigma} \|x - y\|, \quad \forall x, y \in \mathfrak{R}^n.$$

**Proof:** We first note that the maximum in Eq. (6.184) is uniquely attained in view of the strong convexity of  $p$  (which implies that  $p$  is strictly convex). Furthermore,  $f_\epsilon$  is equal to  $f^*(A'x)$ , where  $f^*$  is the conjugate of the function  $\phi(u) + \epsilon p(u) + \delta_U(u)$ , with  $\delta_U$  being the indicator function of  $U$ . It follows that  $f_\epsilon$  is convex, and it is also differentiable with gradient  $\nabla f_\epsilon(x) = A'u_\epsilon(x)$  by the Conjugate Subgradient Theorem (Prop. 5.4.3).

Consider any vectors  $x, y \in \mathfrak{R}^n$ . By using the optimality condition for the maximization (6.184), we have

$$\left( Ax - g_x - \epsilon \nabla p(u_\epsilon(x)) \right)' (u_\epsilon(y) - u_\epsilon(x)) \leq 0,$$

$$\left( Ay - g_y - \epsilon \nabla p(u_\epsilon(y)) \right)' (u_\epsilon(x) - u_\epsilon(y)) \leq 0,$$

where  $g_x$  and  $g_y$  are subgradients of  $\phi$  at  $u_\epsilon(x)$  and  $u_\epsilon(y)$ , respectively. Adding the two inequalities, and using the convexity of  $\phi$  and the strong convexity of  $p$ , we obtain

$$\begin{aligned} (x - y)' A' (u_\epsilon(x) - u_\epsilon(y)) &\geq \left( g_x - g_y + \epsilon \left( \nabla p(u_\epsilon(x)) - \nabla p(u_\epsilon(y)) \right) \right)' \\ &\quad (u_\epsilon(x) - u_\epsilon(y)) \\ &\geq \epsilon \left( \nabla p(u_\epsilon(x)) - \nabla p(u_\epsilon(y)) \right)' (u_\epsilon(x) - u_\epsilon(y)) \\ &\geq \epsilon\sigma \|u_\epsilon(x) - u_\epsilon(y)\|^2, \end{aligned}$$

where for the second inequality we have used the fact

$$(g_x - g_y)' (u_\epsilon(x) - u_\epsilon(y)) \geq 0$$

(cf. the Exercises of Chapter 5), and for the third inequality we used a standard property of strongly convex functions (see the Exercises for Chapter 1). Thus,

$$\begin{aligned}
\|\nabla f_\epsilon(x) - \nabla f_\epsilon(y)\|^2 &= \|A'(u_\epsilon(x) - u_\epsilon(y))\|^2 \\
&\leq \|A'\|^2 \|u_\epsilon(x) - u_\epsilon(y)\|^2 \\
&\leq \frac{\|A'\|^2}{\epsilon\sigma} (x - y)' A'(u_\epsilon(x) - u_\epsilon(y)) \\
&\leq \frac{\|A'\|^2}{\epsilon\sigma} \|x - y\| \|A'(u_\epsilon(x) - u_\epsilon(y))\| \\
&= \frac{\|A\|^2}{\epsilon\sigma} \|x - y\| \|\nabla f_\epsilon(x) - \nabla f_\epsilon(y)\|,
\end{aligned}$$

from which the result follows. **Q.E.D.**

We now consider the minimization over a closed convex set  $X$  of the function

$$f(x) = F(x) + f_0(x),$$

where  $f_0$  is given by Eq. (6.183), and  $F : \mathfrak{R}^n \mapsto \mathfrak{R}$  is convex and differentiable, with gradient satisfying the Lipschitz condition

$$\|\nabla F(x) - \nabla F(y)\| \leq L \|x - y\|, \quad \forall x, y \in X. \quad (6.186)$$

We replace  $f$  with the smooth approximation

$$\tilde{f}(x) = F(x) + f_\epsilon(x),$$

and note that  $\tilde{f}$  uniformly differs from  $f$  by at most  $p^*\epsilon$  [cf. Eq. (6.185)], and has Lipschitz continuous gradient with Lipschitz constant  $L + L_\epsilon = O(1/\epsilon)$ . Thus, by applying the algorithm (6.176) and by using Prop. 6.9.2, we see that we can obtain a solution  $\tilde{x} \in X$  such that  $f(\tilde{x}) \leq f^* + p^*\epsilon$  with

$$O(\sqrt{(L + \|A\|^2/\epsilon\sigma)/\epsilon}) = O(1/\epsilon)$$

iterations.

## 6.10 NOTES, SOURCES, AND EXERCISES

Subgradient methods were first introduced in the Soviet Union in the middle 60s by Shor; the works of Ermoliev and Poljak were also particularly influential. Description of these works can be found in many sources, including Shor [Sho85], Ermoliev [Erm83], and Polyak [Pol87]. An extensive

bibliography for the early period of the subject is given in the edited volume by Balinski and Wolfe [BaW75]. There are many works dealing with analysis of subgradient methods. The convergence analysis of Section 6.3 is based on the paper by Nedić and Bertsekas [NeB01a]. There are several variations of subgradient methods that aim to accelerate the convergence of the basic method (see e.g., [CFM75], [Sho85], [Min86], [Str97], [LPS98], [Sho98], [ZLW99], [GZL02]).

Cutting plane methods were introduced by Cheney and Goldstein [ChG59], and by Kelley [Kel60]. For analysis of related methods, see Ruszczyński [Rus89], Lemaréchal and Sagastizábal [LeS93], Mifflin [Mif96], Burke and Qian [BuQ98], Mifflin, Sun, and Qi [MSQ98], and Bonnans et al. [BGL09]. Variants of cutting plane methods were introduced by Elzinga and Moore [ElM75]. More recent proposals, some of which relate to interior point methods, are described in the textbook by Ye [Ye97], and the survey by Goffin and Vial [GoV02]. The simplicial decomposition method was introduced by Holloway [Hol74]; see also Hohenbalken [Hoh77], Hearn, Lawphongpanich, and Ventura [HLV87], and Ventura and Hearn [VeH93]. Some of these references describe applications to communication and transportation networks (see also the textbook [Ber99], Examples 2.1.3 and 2.1.4). A simplicial decomposition method for minimizing a nondifferentiable convex function over a polyhedral set, based on concepts of ergodic sequences of subgradients and a conditional subgradient method, is given by Larsson, Patriksson, and Stromberg (see [Str97], [LPS98]).

Extended monotropic programming and its duality theory were developed in Bertsekas [Ber06]. The corresponding material on generalized simplicial decomposition (Section 6.4.4) and generalized polyhedral approximation (Section 6.4.5) is new, and is based on joint research of the author with H. Yu; see [BeY09], which contains a detailed convergence analysis.

The proximal point algorithm was introduced by Martinet [Mar70], [Mar72]. The finite termination of the method when applied to linear programs was shown independently by Polyak and Tretyakov [PoT74] and Bertsekas [Ber75a]. The rate of convergence analysis given here is due to Kort and Bertsekas [KoB76], and has been extensively discussed in the book [Ber82]. A generalization of the proximal point algorithm, applying to a broader class of problems, has been extensively developed by Rockafellar [Roc76a], [Roc76b], and together with its special cases, has been analyzed by many authors (see e.g., Luque [Luq84], Guler [Gul91], Eckstein and Bertsekas [EcB92], Pennanen [Pen02]). For a textbook discussion, see Facchinei and Pang [FaP03].

Bundle methods are currently one of the principal classes of methods for solving dual problems. Detailed presentations are given in the textbooks by Hiriart-Urrutu and Lemaréchal [HiR93], and Bonnans et al. [BGL06], which give many references; see also [FeK00], [MSQ98], [ZhL02]. The term “bundle” has been used with a few different meanings in convex algorithmic optimization, with some confusion resulting. To our knowledge, it was

first introduced in the 1975 paper by Wolfe [Wol75] to describe a collection of subgradients used for calculating a descent direction in the context of a specific algorithm of the descent type - a context with no connection to cutting planes or proximal minimization. It subsequently appeared in related descent method contexts through the 1970's and early 1980's. Starting in the middle 1980's, the context of the term "bundle method" gradually shifted, and it is now commonly associated with the stabilized proximal cutting plane methods that we have described in Section 6.5.2.

There are proximal point-type algorithms that use nonquadratic proximal terms, and find application in specialized contexts. They were introduced by several authors, starting to our knowledge with the paper by Kort and Bertsekas [KoB72], and the thesis by Kort [Kor75], in the context of broad classes of methods involving nonquadratic augmented Lagrangians (see also Kort and Bertsekas [KoB76], and Bertsekas [Ber82]). There has been much work in this area, directed at obtaining additional classes of methods, sharper convergence results, and an understanding of the properties that enhance computational performance; see Censor and Zenios [CeZ92], [CeZ97], Guler [Gul92], Chen and Teboulle [ChT93], [ChT94], Tseng and Bertsekas [TsB93], Bertsekas and Tseng [BeT94], Eckstein [Eck94], Iusem, Svaiter, and Teboulle [IST94], Iusem and Teboulle [IuT95], Auslender, Cominetti, and Haddou [AHR97], Polyak and Teboulle [PoT97], Iusem [Ius99], Facchinei and Pang [FaP03], and Auslender and Teboulle [AuT03]. These methods, when combined with the cutting plane approximation approach yield corresponding bundle-type algorithms.

The Augmented Lagrangian method was independently proposed in the papers by Hestenes [Hes69], Powell [Pow69], and Haarhoff and Buys [HaB70] in a nonlinear programming context where convexity played no apparent role. The papers contained little analysis, and did not suggest any relation to duality and the proximal point algorithm. These relations were analyzed by Rockafellar [Roc73], [Roc76a]. An extensive development and analysis of Augmented Lagrangian and related methods is given in the author's research monograph [Ber82], together with many references; see also the survey papers [Ber76], [Roc76b], [Ius99]. The textbook [BeT89] contains several applications of Augmented Lagrangians to classes of large-scale problems with special structure.

Interior point methods date to the work of M. R. Frisch in the middle 50's [Fri56]. They achieved a great deal of popularity in the early 80's when they were systematically applied to linear programming problems. The research monographs by Wright [Wri97] and Ye [Ye97] are devoted to interior point methods for linear, quadratic, and convex programming. More recently interior point methods were adapted and analyzed for conic programming, starting with the research monograph by Nesterov and Nemirovskii [NeN94]. This development had a strong influence in convex optimization practice, as conic programming became established as a field with a strong algorithmic methodology and extensive applications, ranging from discrete

optimization, to control theory, communications, and machine learning. The book by Wolkowicz, Saigal, and Vanderbergue [WSV00] contains a collection of survey articles on semidefinite programming. The book by Boyd and Vanderbergue [BoV04] describes many applications, and contains a lot of material and references.

One may view  $\epsilon$ -subgradient methods in the context of subgradient methods that involve errors in the calculation of the subgradient. Such methods have a long history; see e.g., Ermoliev [Erm69], [Erm83], Nurminskii [Nur74], Polyak [Pol87], and for more recent treatments, Correa and Lemarechal [CoL94], Solodov and Zavriev [SoZ98], and Nedić and Bertsekas [NeB07]. The first proposal of an incremental subgradient method was in Kibardin [Kib79], a work that remained unknown in the Western literature until about 2005. Our material on incremental subgradient methods is based on Nedić and Bertsekas [NeB01a], [NeB01b], [BNO03], which also provided the first analysis of incremental subgradient methods with randomization. Incremental gradient methods for differentiable cost functions have a long history in the area of neural network training; see [BeT96] and [Ber99] for textbook accounts of this methodology and references. For related works, some of which apply to differentiable problems only, see Davidon [Dav76], Luo [Luo91], Gaivoronski [Gai94], Grippo [Gri94], Luo and Tseng [LuT94], Mangasarian and Solodov [MaS94], Bertsekas and Tsitsiklis [BeT96], [BeT00], Bertsekas [Ber96], [Ber97], Kaskavelis and Caramanis [KaC98], Solodov [Sol98], Tseng [Tse98], Ben-Tal, Margalit, and Nemirovski [BMN01], Zhao, Luh, and Wang [ZLW99], Rabbat and Nowak [RaN05], Blatt, Hero, and Gauchman [BHG07]. The incremental proximal algorithms of Section 6.8.4 are considered and analyzed here for the first time, to our knowledge (see also [Ber10a], [Ber10b]).

A distributed asynchronous implementation of incremental subgradient methods, with and without randomization, was given by Nedić, Bertsekas, and Borkar [NBB01]. In the randomized distributed version, the multiple processors, whenever they become available, select at random a component  $f_i$ , calculate the subgradient  $g_i$ , and execute the corresponding incremental subgradient step. The algorithm is asynchronous in that different processors use slightly differing copies of the current iterate  $x_k$ ; this contributes to the efficiency of the method, because there is no waiting time for processors to synchronize at the end of iterations. Despite the asynchronism, convergence can be shown thanks to the use of a diminishing stepsize, similar to related distributed asynchronous gradient method analyses for differentiable optimization problems (see Tsitsiklis, Bertsekas, and Athans [TBA86], and Bertsekas and Tsitsiklis [BeT89]). This analysis applies also to the incremental proximal algorithms of Section 6.8.4.

The ideas of the iteration complexity analysis and algorithms of Section 6.9 have a long history, beginning in the late 70's. In this connection, we mention the works of Nemirovskii and Yudin [NeY83], and Nesterov [Nes83]. The focus on convergence rate analysis and optimal algorithms is



also characteristic of the work of Polyak (see e.g., the textbook [Pol87]), who among others, proposed the heavy-ball method [Pol64]. The optimal gradient projection/extrapolation method of this section stems from the ideas of Nesterov [Nes83], [Nes04], [Nes05] (see also Beck and Teboulle [BeT09], Lu, Monteiro, and Yuan [LMY08], and Tseng [Tse08]). We follow the analysis of Tseng [Tse08], who proposed and analyzed more general methods that also apply to important classes of nondifferentiable cost functions.

Smoothing for nondifferentiable optimization was first suggested by the author in [Ber75b], [Ber77], [Ber82], as an application of the Augmented Lagrangian methodology (see Exercises 6.12-6.14). It has been discussed by several other authors, including Polyak [Pol79], Papavassilopoulos [Pap81], and Censor and Zenios [CeZ92]. The idea of using smoothing in conjunction with a gradient method to construct optimal algorithms is due to Nesterov [Nes05]. In his work he proves the Lipschitz property of Prop. 6.9.3 for the more general case, where  $p$  is convex but not necessarily differentiable, and analyzes several important special cases.

---

## E X E R C I S E S

---

### 6.1 (Minimizing the Sum or the Maximum of Norms [LVB98])

Consider the problems

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^p \|F_i x + g_i\| \\ & \text{subject to} && x \in \mathfrak{R}^n, \end{aligned} \tag{6.187}$$

and

$$\begin{aligned} & \text{minimize} && \max_{i=1, \dots, p} \|F_i x + g_i\| \\ & \text{subject to} && x \in \mathfrak{R}^n, \end{aligned}$$

where  $F_i$  and  $g_i$  are given matrices and vectors, respectively. Convert these problems to second order cone form and derive the corresponding dual problems.

### 6.2 (Complex $l_1$ and $l_\infty$ Approximation [LVB98])

Consider the complex  $l_1$  approximation problem

$$\begin{aligned} & \text{minimize} && \|Ax - b\|_1 \\ & \text{subject to} && x \in \mathcal{C}^n, \end{aligned}$$

where  $\mathcal{C}^n$  is the set of  $n$ -dimensional vectors whose components are complex numbers. Show that it is a special case of problem (6.187) and derive the corresponding dual problem. Repeat for the complex  $l_\infty$  approximation problem

$$\begin{aligned} &\text{minimize} && \|Ax - b\|_\infty \\ &\text{subject to} && x \in \mathcal{C}^n. \end{aligned}$$

### 6.3

The purpose of this exercise is to show that the SOCP can be viewed as a special case of SDP.

- (a) Show that a vector  $x \in \Re^n$  belongs to the second order cone if and only if the matrix

$$x_n I + x_1 v_1 v_1' + \cdots + x_{n-1} v_{n-1} v_{n-1}'$$

is positive semidefinite, where  $v_i$  is the vector of  $\Re^n$  whose components are all equal to 0 except for the  $(i + 1)$ st component which is equal to 1. *Hint:* We have that for any positive definite  $n \times n$  matrix  $A$ , vector  $b \in \Re^n$ , and scalar  $d$ , the matrix

$$\begin{pmatrix} A & b \\ b' & c \end{pmatrix}$$

is positive definite if and only if  $c - b' A^{-1} b > 0$ .

- (b) Use part (a) to show that the primal SOCP can be written in the form of the dual SDP.

### 6.4 (Explicit Form of a Second Order Cone Problem)

Consider the SOCP (6.18).

- (a) Partition the  $n_i \times (n + 1)$  matrices  $(A_i \ b_i)$  as

$$(A_i \ b_i) = \begin{pmatrix} D_i & d_i \\ p_i' & q_i \end{pmatrix}, \quad i = 1, \dots, m,$$

where  $D_i$  is an  $(n_i - 1) \times n$  matrix,  $d_i \in \Re^{n_i - 1}$ ,  $p_i \in \Re^n$ , and  $q_i \in \Re$ . Show that

$$A_i x - b_i \in C_i \quad \text{if and only if} \quad \|D_i x - d_i\| \leq p_i' x - q_i,$$

so we can write the SOCP (6.18) as

$$\begin{aligned} &\text{minimize} && c' x \\ &\text{subject to} && \|D_i x - d_i\| \leq p_i' x - q_i, \quad i = 1, \dots, m. \end{aligned}$$

(b) Similarly partition  $\lambda_i$  as

$$\lambda_i = \begin{pmatrix} \mu_i \\ \nu_i \end{pmatrix}, \quad i = 1, \dots, m,$$

where  $\mu_i \in \Re^{n_i-1}$  and  $\nu_i \in \Re$ . Show that the dual problem (6.19) can be written in the form

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^m (d'_i \mu_i + q_i \nu_i) \\ & \text{subject to} && \sum_{i=1}^m (D'_i \mu_i + \nu_i p_i) = c, \quad \|\mu_i\| \leq \nu_i, \quad i = 1, \dots, m. \end{aligned} \tag{6.188}$$

(c) Show that the primal and dual interior point conditions for strong duality (Prop. 6.1.2) hold if there exist primal and dual feasible solutions  $\bar{x}$  and  $(\bar{\mu}_i, \bar{\nu}_i)$  such that

$$\|D_i \bar{x} - d_i\| < p'_i \bar{x} - q_i, \quad i = 1, \dots, m,$$

and

$$\|\bar{\mu}_i\| < \bar{\nu}_i, \quad i = 1, \dots, m,$$

respectively.

### 6.5 (Monotropic-Conic Problems)

Consider the problem

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^m f_i(x_i) \\ & \text{subject to} && x \in S \cap C, \end{aligned}$$

where  $x = (x_1, \dots, x_m)$  with  $x_i \in \Re^{n_i}$ ,  $i = 1, \dots, m$ , and  $f_i : \Re^{n_i} \mapsto (-\infty, \infty]$  is a proper convex function for each  $i$ , and  $S$  and  $C$  are a subspace and a cone of  $\Re^{n_1 + \dots + n_m}$ , respectively. Show that a dual problem is

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^m q_i(\lambda_i) \\ & \text{subject to} && \lambda \in \hat{C} + S^\perp, \end{aligned}$$

where  $\lambda = (\lambda_1, \dots, \lambda_m)$ ,  $\hat{C}$  is the dual cone of  $C$ , and

$$q_i(\lambda_i) = \inf_{z_i \in \Re^{n_i}} \{f_i(z_i) - \lambda'_i z_i\}, \quad i = 1, \dots, m.$$

6.6

Let  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  be a convex function, and let  $\{f_k\}$  be a sequence of convex functions  $f_k : \mathfrak{R}^n \mapsto \mathfrak{R}$  with the property that  $\lim_{k \rightarrow \infty} f_k(x_k) = f(x)$  for every  $x \in \mathfrak{R}^n$  and every sequence  $\{x_k\}$  that converges to  $x$ . Then, for any  $x \in \mathfrak{R}^n$  and  $y \in \mathfrak{R}^n$ , and any sequences  $\{x_k\}$  and  $\{y_k\}$  converging to  $x$  and  $y$ , respectively, we have

$$\limsup_{k \rightarrow \infty} f'_k(x_k; y_k) \leq f'(x; y). \tag{6.189}$$

Furthermore, if  $f$  is differentiable over  $\mathfrak{R}^n$ , then it is continuously differentiable over  $\mathfrak{R}^n$ .

**Solution:** From the definition of directional derivative, it follows that for any  $\epsilon > 0$ , there exists an  $\alpha > 0$  such that

$$\frac{f(x + \alpha y) - f(x)}{\alpha} < f'(x; y) + \epsilon.$$

Hence, using also the equation

$$f'(x; y) = \inf_{\alpha > 0} \frac{f(x + \alpha y) - f(x)}{\alpha},$$

we have for all sufficiently large  $k$ ,

$$f'_k(x_k; y_k) \leq \frac{f_k(x_k + \alpha y_k) - f_k(x_k)}{\alpha} < f'(x; y) + \epsilon,$$

so by taking the limit as  $k \rightarrow \infty$ ,

$$\limsup_{k \rightarrow \infty} f'_k(x_k; y_k) \leq f'(x; y) + \epsilon.$$

Since this is true for all  $\epsilon > 0$ , we obtain  $\limsup_{k \rightarrow \infty} f'_k(x_k; y_k) \leq f'(x; y)$ .

If  $f$  is differentiable at all  $x \in \mathfrak{R}^n$ , then using the continuity of  $f$  and the part of the proposition just proved, we have for every sequence  $\{x_k\}$  converging to  $x$  and every  $y \in \mathfrak{R}^n$ ,

$$\limsup_{k \rightarrow \infty} \nabla f(x_k)'y = \limsup_{k \rightarrow \infty} f'(x_k; y) \leq f'(x; y) = \nabla f(x)'y.$$

By replacing  $y$  with  $-y$  in the preceding argument, we obtain

$$-\liminf_{k \rightarrow \infty} \nabla f(x_k)'y = \limsup_{k \rightarrow \infty} (-\nabla f(x_k)'y) \leq -\nabla f(x)'y.$$

Therefore, we have  $\nabla f(x_k)'y \rightarrow \nabla f(x)'y$  for every  $y$ , which implies that  $\nabla f(x_k) \rightarrow \nabla f(x)$ . Hence,  $\nabla f(\cdot)$  is continuous.

### 6.7 (Danskin's Theorem)

Let  $Z$  be a compact subset of  $\mathfrak{R}^n$ , and let  $\phi : \mathfrak{R}^n \times Z \mapsto \mathfrak{R}$  be continuous and such that  $\phi(\cdot, z) : \mathfrak{R}^n \mapsto \mathfrak{R}$  is convex for each  $z \in Z$ .

(a) The function  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  given by

$$f(x) = \max_{z \in Z} \phi(x, z) \quad (6.190)$$

is convex and has directional derivative given by

$$f'(x; y) = \max_{z \in Z(x)} \phi'(x, z; y), \quad (6.191)$$

where  $\phi'(x, z; y)$  is the directional derivative of the function  $\phi(\cdot, z)$  at  $x$  in the direction  $y$ , and  $Z(x)$  is the set of maximizing points in Eq. (6.190)

$$Z(x) = \left\{ \bar{z} \mid \phi(x, \bar{z}) = \max_{z \in Z} \phi(x, z) \right\}.$$

In particular, if  $Z(x)$  consists of a unique point  $\bar{z}$  and  $\phi(\cdot, \bar{z})$  is differentiable at  $x$ , then  $f$  is differentiable at  $x$ , and  $\nabla f(x) = \nabla_x \phi(x, \bar{z})$ , where  $\nabla_x \phi(x, \bar{z})$  is the vector with components

$$\frac{\partial \phi(x, \bar{z})}{\partial x_i}, \quad i = 1, \dots, n.$$

(b) If  $\phi(\cdot, z)$  is differentiable for all  $z \in Z$  and  $\nabla_x \phi(x, \cdot)$  is continuous on  $Z$  for each  $x$ , then

$$\partial f(x) = \text{conv} \left\{ \nabla_x \phi(x, z) \mid z \in Z(x) \right\}, \quad \forall x \in \mathfrak{R}^n.$$

**Solution:** (a) We note that since  $\phi$  is continuous and  $Z$  is compact, the set  $Z(x)$  is nonempty by Weierstrass' Theorem and  $f$  is finite. For any  $z \in Z(x)$ ,  $y \in \mathfrak{R}^n$ , and  $\alpha > 0$ , we use the definition of  $f$  to obtain

$$\frac{f(x + \alpha y) - f(x)}{\alpha} \geq \frac{\phi(x + \alpha y, z) - \phi(x, z)}{\alpha}.$$

Taking the limit as  $\alpha$  decreases to zero, we obtain  $f'(x; y) \geq \phi'(x, z; y)$ . Since this is true for every  $z \in Z(x)$ , we conclude that

$$f'(x; y) \geq \sup_{z \in Z(x)} \phi'(x, z; y), \quad \forall y \in \mathfrak{R}^n. \quad (6.192)$$

To prove the reverse inequality and that the supremum in the right-hand side of the above inequality is attained, consider a sequence  $\{\alpha_k\}$  of positive scalars that converges to zero and let  $x_k = x + \alpha_k y$ . For each  $k$ , let  $z_k$  be a vector in  $Z(x_k)$ . Since  $\{z_k\}$  belongs to the compact set  $Z$ , it has a subsequence

converging to some  $\bar{z} \in Z$ . Without loss of generality, we assume that the entire sequence  $\{z_k\}$  converges to  $\bar{z}$ . We have

$$\phi(x_k, z_k) \geq \phi(x_k, z), \quad \forall z \in Z,$$

so by taking the limit as  $k \rightarrow \infty$  and by using the continuity of  $\phi$ , we obtain

$$\phi(x, \bar{z}) \geq \phi(x, z), \quad \forall z \in Z.$$

Therefore,  $\bar{z} \in Z(x)$ . We now have

$$\begin{aligned} f'(x; y) &\leq \frac{f(x + \alpha_k y) - f(x)}{\alpha_k} \\ &= \frac{\phi(x + \alpha_k y, z_k) - \phi(x, \bar{z})}{\alpha_k} \\ &\leq \frac{\phi(x + \alpha_k y, z_k) - \phi(x, z_k)}{\alpha_k} \\ &\leq -\phi'(x + \alpha_k y, z_k; -y) \\ &\leq \phi'(x + \alpha_k y, z_k; y), \end{aligned} \tag{6.193}$$

where the last inequality follows from the fact  $-\phi'(x; -y) \leq \phi'(x; y)$ . We apply the result of Exercise 6.6 to the functions  $f_k$  defined by  $f_k(\cdot) = \phi(\cdot, z_k)$ , and with  $x_k = x + \alpha_k y$ , to obtain

$$\limsup_{k \rightarrow \infty} \phi'(x + \alpha_k y, z_k; y) \leq \phi'(x, \bar{z}; y). \tag{6.194}$$

We take the limit in inequality (6.193) as  $k \rightarrow \infty$ , and we use inequality (6.194) to conclude that

$$f'(x; y) \leq \phi'(x, \bar{z}; y).$$

This relation together with inequality (6.192) proves Eq. (6.191).

For the last statement of part (a), if  $Z(x)$  consists of the unique point  $\bar{z}$ , Eq. (6.191) and the differentiability assumption on  $\phi$  yield

$$f'(x; y) = \phi'(x, \bar{z}; y) = y' \nabla_x \phi(x, \bar{z}), \quad \forall y \in \mathbb{R}^n,$$

which implies that  $\nabla f(x) = \nabla_x \phi(x, \bar{z})$ .

(b) By part (a), we have

$$f'(x; y) = \max_{z \in Z(x)} \nabla_x \phi(x, z)' y,$$

while by Prop. 5.4.8,

$$f'(x; y) = \max_{z \in \partial f(x)} d' y.$$

For all  $\bar{z} \in Z(x)$  and  $y \in \mathbb{R}^n$ , we have

$$\begin{aligned} f(y) &= \max_{z \in Z} \phi(y, z) \\ &\geq \phi(y, \bar{z}) \\ &\geq \phi(x, \bar{z}) + \nabla_x \phi(x, \bar{z})'(y - x) \\ &= f(x) + \nabla_x \phi(x, \bar{z})'(y - x). \end{aligned}$$

Therefore,  $\nabla_x \phi(x, \bar{z})$  is a subgradient of  $f$  at  $x$ , implying that

$$\text{conv}\{\nabla_x \phi(x, z) \mid z \in Z(x)\} \subset \partial f(x).$$

To prove the reverse inclusion, we use a hyperplane separation argument. By the continuity of  $\nabla_x \phi(x, \cdot)$  and the compactness of  $Z$ , we see that  $Z(x)$  is compact, and therefore also the set  $\{\nabla_x \phi(x, z) \mid z \in Z(x)\}$  is compact. By Prop. 1.2.2, it follows that  $\text{conv}\{\nabla_x \phi(x, z) \mid z \in Z(x)\}$  is compact. If  $d \in \partial f(x)$  while  $d \notin \text{conv}\{\nabla_x \phi(x, z) \mid z \in Z(x)\}$ , by the Strict Separation Theorem (Prop. 1.5.3), there exists  $y \neq 0$ , and  $\gamma \in \mathfrak{R}$ , such that

$$d'y > \gamma > \nabla_x \phi(x, z)'y, \quad \forall z \in Z(x).$$

Therefore, we have

$$d'y > \max_{z \in Z(x)} \nabla_x \phi(x, z)'y = f'(x; y),$$

contradicting Prop. 5.4.8. Therefore,  $\partial f(x) \subset \text{conv}\{\nabla_x \phi(x, z) \mid z \in Z(x)\}$  and the proof is complete.

### 6.8 (Failure of the Steepest Descent Method [Wol75])

Consider the minimization of the two-dimensional function

$$f(x_1, x_2) = \begin{cases} 5(9x_1^2 + 16x_2^2)^{1/2} & \text{if } x_1 > |x_2|, \\ 9x_1 + 16|x_2| & \text{if } x_1 \leq |x_2|, \end{cases}$$

using the steepest descent method, which moves from the current point in the opposite direction of the minimum norm subgradient (or gradient in the case where the function is differentiable at the point), with the stepsize determined by cost minimization along that direction. Suppose that the algorithm starts anywhere within the set

$$\{(x_1, x_2) \mid x_1 > |x_2| > (9/16)^2 |x_1|\}.$$

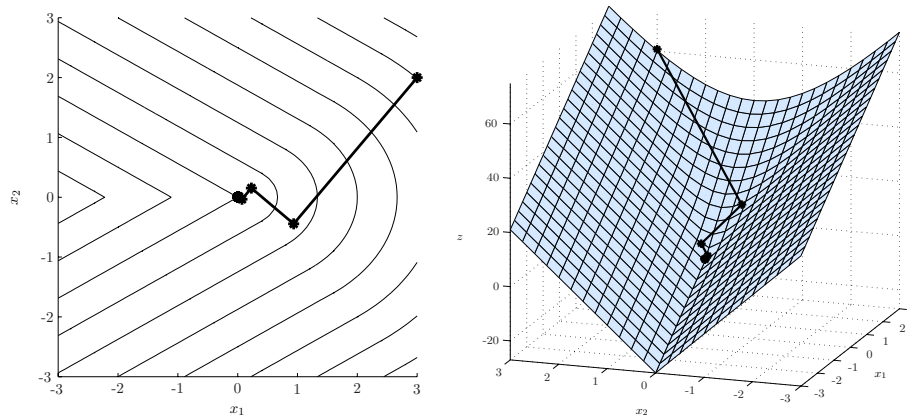
Verify computationally that it converges to the nonoptimal point  $(0, 0)$  (cf. Fig. 6.10.1). What happens if a subgradient method with a constant stepsize is used instead? Check computationally.

### 6.9 (Growth Condition for Polyhedral Functions)

Let  $f : \mathfrak{R}^n \mapsto (-\infty, \infty]$  be a polyhedral function, and assume that  $X^*$ , the set of minima of  $f$ , is nonempty. Show that there exists a scalar  $\beta > 0$  such that

$$f^* + \beta d(x) \leq f(x), \quad \forall x \notin X^*,$$

where  $d(x) = \min_{x^* \in X^*} \|x - x^*\|$ . *Hint:* Complete the details of the following argument:



**Figure 6.10.1.** Level sets and steepest descent path for the function of Exercise 6.8. The method converges to the nonoptimal point  $(0, 0)$ .

Assume first that  $f$  is linear within  $\text{dom}(f)$ . Then, there exists  $a \in \mathfrak{R}^n$  such that for all  $x, \hat{x} \in \text{dom}(f)$ , we have

$$f(x) - f(\hat{x}) = a'(x - \hat{x}).$$

For any  $x \in X^*$ , let  $S_x$  be the cone of vectors  $d$  that are in the normal cone  $N_{X^*}(x)$  of  $X^*$  at  $x$ , and are also feasible directions in the sense that  $x + \alpha d \in \text{dom}(f)$  for a small enough  $\alpha > 0$ . Since  $X^*$  and  $\text{dom}(f)$  are polyhedral sets, there exist only a finite number of possible cones  $S_x$  as  $x$  ranges over  $X^*$ . Thus, there is a finite set of nonzero vectors  $\{c_j \mid j \in J\}$ , such that for any  $x \in X^*$ ,  $S_x$  is either equal to  $\{0\}$ , or is the cone generated by a subset  $\{c_j \mid j \in J_x\}$ , where  $J = \cup_{x \in X^*} J_x$ . In addition, for all  $x \in X^*$  and  $d \in S_x$  with  $\|d\| = 1$ , we have

$$d = \sum_{j \in J_x} \gamma_j c_j,$$

for some scalars  $\gamma_j \geq 0$  with  $\sum_{j \in J_x} \gamma_j \geq \bar{\gamma}$ , where  $\bar{\gamma} = 1/\max_{j \in J} \|c_j\|$ . Also we can show that for all  $j \in J$ , we have  $a'c_j > 0$ , by using the fact  $c_j \in S_x$  for some  $x \in X^*$ .

For  $x \in \text{dom}(f)$  with  $x \notin X^*$ , let  $\hat{x}$  be the projection of  $x$  on  $X^*$ . Then the vector  $x - \hat{x}$  belongs to  $S_{\hat{x}}$ , and we have

$$f(x) - f(\hat{x}) = a'(x - \hat{x}) = \|x - \hat{x}\| \frac{a'(x - \hat{x})}{\|x - \hat{x}\|} \geq \beta \|x - \hat{x}\|,$$

where

$$\beta = \bar{\gamma} \min_{j \in J} a'c_j.$$

Since  $J$  is finite, we have  $\beta > 0$ , and this implies the desired result for the case where  $f$  is linear within  $\text{dom}(f)$ .



Assume now that  $f$  is of the form

$$f(x) = \max_{i \in I} \{a_i'x + b_i\}, \quad \forall x \in \text{dom}(f),$$

where  $I$  is a finite set, and  $a_i$  and  $b_i$  are some vectors and scalars, respectively. Let

$$Y = \{(x, z) \mid z \geq f(x), x \in \text{dom}(f)\},$$

and consider the function

$$g(x, z) = \begin{cases} z & \text{if } (x, z) \in Y, \\ \infty & \text{otherwise.} \end{cases}$$

Note that  $g$  is polyhedral and linear within  $\text{dom}(g)$ , that its set of minima is

$$Y^* = \{(x, z) \mid x \in X^*, z = f^*\},$$

and that its minimal value is  $f^*$ .

Applying the result already shown to the function  $g$ , we have for some  $\beta > 0$

$$f^* + \beta \hat{d}(x, z) \leq g(x, z), \quad \forall (x, z) \notin Y^*,$$

where

$$\hat{d}(x, z) = \min_{(x^*, z^*) \in Y^*} (\|x - x^*\|^2 + |z - z^*|^2)^{1/2} = \min_{x^* \in X^*} (\|x - x^*\|^2 + |z - f^*|^2)^{1/2}.$$

Since

$$\hat{d}(x, z) \geq \min_{x^* \in X^*} \|x - x^*\| = d(x),$$

we have

$$f^* + \beta d(x) \leq g(x, z), \quad \forall (x, z) \notin Y^*,$$

and by taking the infimum of the right-hand side over  $z$  for any fixed  $x$ , we obtain

$$f^* + \beta d(x) \leq f(x), \quad \forall x \notin X^*.$$

### 6.10 (Sublinear Convergence Rate in the Proximal Minimization Algorithm)

Consider the proximal point algorithm under the assumptions of Prop. 6.5.2. Assume further that  $\alpha > 2$ . Show that

$$\limsup_{k \rightarrow \infty} \frac{d(x_{k+1})}{d(x_k)^{2/\alpha}} < \infty$$

which is known as *sublinear convergence*. *Hint:* Show first that

$$f(x_{k+1}) - f^* \leq \frac{d(x_k)^2}{2c_k}, \quad \forall k.$$

**6.11 (Partial Proximal Minimization Algorithm [BeT94])**

For each  $c > 0$ , let  $\phi_c$  be the real-valued convex function on  $\mathfrak{R}^n$  defined by

$$\phi_c(z) = \min_{x \in X} \left\{ f(x) + \frac{1}{2c} \|x - z\|^2 \right\},$$

where  $f$  is a convex function over the closed convex set  $X$ . Let  $I$  be a subset of the index set  $\{1, \dots, n\}$ . For any  $z \in \mathfrak{R}^n$ , consider a vector  $\bar{z}$  satisfying

$$\bar{z} \in \arg \min_{x \in X} \left\{ f(x) + \frac{1}{2c} \sum_{i \in I} |x_i - z_i|^2 \right\},$$

and let  $\tilde{z}$  be the vector with components

$$\tilde{z}_i = \begin{cases} z_i & \forall i \in I, \\ \bar{z}_i & \forall i \notin I. \end{cases}$$

(a) Show that

$$\begin{aligned} \tilde{z} &\in \arg \min_{\{x | x_i = z_i, i \in I\}} \phi_c(x), \\ \bar{z} &\in \arg \min_{x \in X} \left\{ f(x) + \frac{1}{2c} \|x - \tilde{z}\|^2 \right\}. \end{aligned}$$

(b) Interpret  $\bar{z}$  as the result of a block coordinate descent step corresponding to the components  $z_i, i \notin I$ , followed by a proximal minimization step, and show that  $\phi_c(\bar{z}) \leq f(\bar{z}) \leq \phi_c(z) \leq f(z)$ .

**6.12 (Smoothing of Nondifferentiabilities [Ber75b], [Ber77], [Ber82], [Pap81], [Pol79])**

A simple and effective technique to handle nondifferentiabilities in the cost or the constraints of optimization problems is to replace them by smooth approximations and then use gradient-based algorithms. This exercise develops a general technique for deriving such approximations. Let  $f : \mathfrak{R}^n \mapsto (-\infty, \infty]$  be a closed proper convex function with conjugate convex function denoted by  $f^*$ . For each  $x \in \mathfrak{R}^n$ , define

$$f_{c,\lambda}(x) = \inf_{u \in \mathfrak{R}^n} \left\{ f(x - u) + \lambda' u + \frac{c}{2} \|u\|^2 \right\}, \tag{6.195}$$

where  $c$  is a positive scalar, and  $\lambda$  is a vector in  $\mathfrak{R}^n$ . Use the Fenchel duality theorem to show that

$$f_{c,\lambda}(x) = \sup_{y \in \mathfrak{R}^n} \left\{ x'y - f^*(y) - \frac{1}{2c} \|y - \lambda\|^2 \right\}. \tag{6.196}$$

Show also that  $f_{c,\lambda}$  approximates  $f$  in the sense that

$$\lim_{c \rightarrow \infty} f_{c,\lambda}(x) = f(x), \quad \forall x, \lambda \in \mathfrak{R}^n.$$

Furthermore,  $f_{c,\lambda}$  is convex and differentiable as a function of  $x$  for fixed  $c$  and  $\lambda$ , and  $\nabla f_{c,\lambda}(x)$  can be obtained in two ways:

- (i) As the vector  $\lambda + cu$ , where  $u$  attains the infimum in Eq. (6.195).
- (ii) As the vector  $y$  that attains the supremum in Eq. (6.196).

### 6.13 (Smoothing and the Augmented Lagrangian Method I)

This exercise provides an application of the smoothing technique of the preceding exercise. Let  $f : \mathfrak{R}^n \mapsto (-\infty, \infty]$  be a closed proper convex function with conjugate convex function denoted by  $f^*$ . Let  $F : \mathfrak{R}^n \mapsto \mathfrak{R}$  be another convex function, and let  $X$  be a closed convex set. Consider the problem

$$\begin{aligned} & \text{minimize} && F(x) + f(x) \\ & \text{subject to} && x \in X, \end{aligned}$$

and the equivalent problem

$$\begin{aligned} & \text{minimize} && F(x) + f(x - u) \\ & \text{subject to} && x \in X, u = 0. \end{aligned}$$

Apply the Augmented Lagrangian method to the latter problem, and show that it takes the form

$$x_{k+1} \in \arg \min_{x \in X} \{F(x) + f_{c_k, \lambda_k}(x)\},$$

where  $f_{c, \lambda}$  is the smoothed function of the preceding exercise; the multiplier update is obtained from the equations

$$u_{k+1} \in \arg \min_{u \in \mathfrak{R}^n} \left\{ f(x_{k+1} - u) + \lambda_k' u + \frac{c_k}{2} \|u\|^2 \right\}, \quad \lambda_{k+1} = \lambda_k + c_k u_{k+1}.$$

Alternatively,  $\lambda_{k+1}$  is given by

$$\lambda_{k+1} = \arg \max_{y \in \mathfrak{R}^n} \left\{ x_{k+1}' y - f^*(y) - \frac{1}{2c_k} \|y - \lambda_k\|^2 \right\} = \nabla f_{c_k, \lambda_k}(x_{k+1}),$$

where  $f^*$  is the conjugate convex function of  $f$ .

### 6.14 (Smoothing and the Augmented Lagrangian Method II)

This exercise provides an alternative smoothing technique to the one of the preceding exercise. It applies to general convex/concave minimax problems. Let  $Z$  be a nonempty convex subset of  $\mathfrak{R}^m$ , respectively, and  $\phi : \mathfrak{R}^n \times Z \mapsto \mathfrak{R}$  is a function such that  $\phi(\cdot, z) : \mathfrak{R}^n \mapsto \mathfrak{R}$  is convex for each  $z \in Z$ , and  $-\phi(x, \cdot) : Z \mapsto \mathfrak{R}$  is convex and closed for each  $x \in \mathfrak{R}^n$ . Consider the problem

$$\begin{aligned} & \text{minimize} && \sup_{z \in Z} \phi(x, z) \\ & \text{subject to} && x \in X, \end{aligned}$$

where  $X$  is a nonempty closed convex subset of  $\mathfrak{R}^n$ . Consider also the equivalent problem

$$\begin{aligned} & \text{minimize} && H(x, y) \\ & \text{subject to} && x \in X, y = 0, \end{aligned}$$

where  $H$  is the function

$$H(x, y) = \sup_{z \in Z} \{ \phi(x, z) - y'z \}, \quad x \in \mathfrak{R}^n, y \in \mathfrak{R}^m.$$

Apply the Augmented Lagrangian method to this problem, and show that it takes the form

$$x_{k+1} \in \arg \min_{x \in X} h_{c_k, \lambda_k}(x),$$

where  $f_{c, \lambda}^* : \mathfrak{R}^n \mapsto \mathfrak{R}$  is the differentiable function given by

$$f_{c, \lambda}^*(x) = \min_{y \in \mathfrak{R}^m} \left\{ H(x, y) - \lambda'y + \frac{c}{2} \|y\|^2 \right\}, \quad x \in \mathfrak{R}^n.$$

The multiplier update is obtained from the equations

$$y_{k+1} \in \arg \min_{y \in \mathfrak{R}^m} \left\{ H(x_{k+1}, y) - \lambda'_k y + \frac{c_k}{2} \|y\|^2 \right\}, \quad \lambda_{k+1} = \lambda_k - c_k y_{k+1}.$$

### 6.15

Consider the scalar function  $f(x) = |x|$ . Show that for  $x \in \mathfrak{R}$  and  $\epsilon > 0$ , we have

$$\partial_\epsilon f(x) = \begin{cases} [-1, -1 - \frac{\epsilon}{x}] & \text{for } x < -\frac{\epsilon}{2}, \\ [-1, 1] & \text{for } x \in [-\frac{\epsilon}{2}, \frac{\epsilon}{2}], \\ [1 - \frac{\epsilon}{x}, 1] & \text{for } x > \frac{\epsilon}{2}. \end{cases}$$

### 6.16 (Subgradient Methods with Low Level Errors [NeB07])

Consider the problem of minimizing a convex function  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$  over a closed convex set  $X$ , and assume that the optimal solution set, denoted  $X^*$ , is nonempty. Consider the iteration

$$x_{k+1} = P_X(x_k - \alpha_k(g_k + e_k)),$$

where for all  $k$ ,  $g_k$  is a subgradient of  $f$  at  $x_k$ , and  $e_k$  is an error such that for all  $k$ , we have

$$\|e_k\| \leq \beta, \quad k = 0, 1, \dots,$$

where  $\beta$  is some positive scalar. Assume that for some  $\gamma > 0$ , we have

$$f(x) - f^* \leq \gamma \min_{x^* \in X^*} \|x - x^*\|, \quad \forall x \in X,$$

where  $f^* = \min_{x \in X} f(x)$ , and that for some  $c > 0$ , we have

$$\|g\| \leq c, \quad \forall g \in \partial f(x_k), \quad k = 0, 1, \dots$$

(these assumptions are satisfied if  $f$  is a polyhedral function). Assuming  $\beta < \gamma$ , show that if  $\alpha_k$  is equal to some constant  $\alpha$  for all  $k$ , then

$$\liminf_{k \rightarrow \infty} f(x_k) \leq f^* + \frac{\alpha\gamma(c + \beta)^2}{2(\gamma - \beta)}, \quad (6.197)$$

while if

$$\alpha_k \rightarrow 0, \quad \sum_{k=0}^{\infty} \alpha_k = \infty,$$

then  $\liminf_{k \rightarrow \infty} f(x_k) = f^*$ . Use the example of Exercise 6.15 to show that the estimate (6.197) is sharp.

### 6.17 (Sharpness of the Error Tolerance Estimate)

Consider the unconstrained optimization of the two-dimensional function

$$f(x_1, x_2) = \sum_{i=1}^M c_0 (|x_1 + 1| + 2|x_1| + |x_1 - 1| + |x_2 + 1| + 2|x_2| + |x_2 - 1|),$$

where  $c_0$  is a positive constant, by using the incremental subgradient method with a constant stepsize  $\alpha$ . Show that there exists a component processing order such that when the method starts a cycle at the point  $\bar{x} = (\bar{x}_1, \bar{x}_2)$ , where  $\bar{x}_1 = \bar{x}_2 = \alpha M c_0$  with  $\alpha M c_0 \leq 1$ , it returns to  $\bar{x}$  at the end of the cycle. Use this example to show that starting from  $\bar{x}$ , we have

$$\liminf_{k \rightarrow \infty} f(\psi_{i,k}) \geq f^* + \frac{\beta\alpha c^2}{2}, \quad \forall i = 1, \dots, m,$$

for some constant  $\beta$  (independent of  $c_0$  and  $M$ ), where  $c = mc_0$  and  $m = 8M$  [cf. Eq. (6.153)].

**Solution:** Consider the incremental subgradient method with the stepsize  $\alpha$  and the starting point  $\bar{x} = (\alpha M C_0, \alpha M C_0)$ , and the following component processing order:

- $M$  components of the form  $|x_1|$  [endpoint is  $(0, \alpha M C_0)$ ],
- $M$  components of the form  $|x_1 + 1|$  [endpoint is  $(-\alpha M C_0, \alpha M C_0)$ ],
- $M$  components of the form  $|x_2|$  [endpoint is  $(-\alpha M C_0, 0)$ ],
- $M$  components of the form  $|x_2 + 1|$  [endpoint is  $(-\alpha M C_0, -\alpha M C_0)$ ],
- $M$  components of the form  $|x_1|$  [endpoint is  $(0, -\alpha M C_0)$ ],
- $M$  components of the form  $|x_1 - 1|$  [endpoint is  $(\alpha M C_0, -\alpha M C_0)$ ],
- $M$  components of the form  $|x_2|$  [endpoint is  $(\alpha M C_0, 0)$ ], and
- $M$  components of the form  $|x_2 - 1|$  [endpoint is  $(\alpha M C_0, \alpha M C_0)$ ].

With this processing order, the method returns to  $\bar{x}$  at the end of a cycle. Furthermore, the smallest function value within the cycle is attained at points

$(\pm\alpha MC_0, 0)$  and  $(0, \pm\alpha MC_0)$ , and is equal to  $4MC_0 + 2\alpha M^2 C_0^2$ . The optimal function value is  $f^* = 4MC_0$ , so that

$$\liminf_{k \rightarrow \infty} f(\psi_{i,k}) \geq f^* + 2\alpha M^2 C_0^2.$$

Since  $m = 8M$  and  $mC_0 = C$ , we have  $M^2 C_0^2 = C^2/64$ , implying that

$$2\alpha M^2 C_0^2 = \frac{1}{16} \frac{\alpha C^2}{2},$$

and therefore

$$\liminf_{k \rightarrow \infty} f(\psi_{i,k}) \geq f^* + \frac{\beta \alpha C^2}{2},$$

with  $\beta = 1/16$ .

### 6.18 (Aggregate Subgradients)

Show that the aggregate subgradient of Eq. (6.101) in the bundle method can be expressed as a convex combination of past subgradients  $g_i$ , which are “active” in the sense that

$$F_k(x_{k+1}) = f(x_i) + (x_{k+1} - x_i)' g_i.$$

*Hint:* Use quadratic programming duality in conjunction with the proximal optimization problem that defines  $x_{k+1}$ .

### 6.19 (Bregman Distance)

Let  $X$  be a closed convex subset of  $\mathfrak{R}^n$ , and let  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  be a convex function that is differentiable over an open set that contains  $X$ , and is strongly convex in the sense that

$$(\nabla f(x) - \nabla f(z))'(x - z) \geq \|x - z\|^2, \quad \forall x, z \in X.$$

Define a corresponding “distance” function

$$D(x, z) = f(x) - f(z) - \nabla f(z)'(x - z), \quad \forall z \in X, x \in \mathfrak{R}^n.$$

(a) For each  $z \in X$ , show that  $D(\cdot, z)$  is strongly convex over  $X$ , and in fact

$$D(x, z) \geq \frac{1}{2} \|x - z\|^2,$$

with equality holding in the case where

$$f(x) = \frac{1}{2} \|x\|^2.$$

- (b) Let  $\psi : X \mapsto (-\infty, \infty]$  be a closed proper convex function, and for any  $z \in X$ , define

$$z^+ = \arg \min_{x \in X} \{ \psi(x) + D(x, z) \}.$$

Show that

$$\psi(z^+) + D(z^+, z) + D(x, z^+) \leq \psi(x) + D(x, z), \quad \forall x, z \in X.$$

*Hint:* Use the optimality condition of Prop. 5.4.7 to obtain

$$\psi(z^+) \leq \psi(x) + \nabla_x D(z^+, z)'(x - z^+).$$

Then by writing  $\nabla_x D(z^+, z) = \nabla f(z^+) - \nabla f(z)$  and rearranging terms,

$$\psi(z^+) - \nabla f(z)'(z^+ - z) - \nabla f(z^+)'(x - z^+) \leq \psi(x) - \nabla f(z)'(x - z).$$

Add  $f(x) - f(z)$  to both sides.

- (c) Use part (b) with  $\psi(z) = \frac{1}{L}\ell(z; x_k)$  and  $D(x, z) = \frac{1}{2}\|x - z\|^2$  (cf. the notation of the proof of Prop. 6.9.1 and the similar notation of Prop. 6.9.2), to show that

$$\ell(x_{k+1}; x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 \leq \ell(x; x_k) + \frac{L}{2}\|x - x_k\|^2 - \frac{L}{2}\|x - x_{k+1}\|^2.$$

## References

- [AuT03] Auslender, A., and Teboulle, M., 2003. *Asymptotic Cones and Functions in Optimization and Variational Inequalities*, Springer-Verlag, New York, NY.
- [BGL09] Bonnans, F., Gilbert, J. C., Lemaréchal, C., and Sagastizábal, C. A., 2009. *Numerical Optimization: Theoretical and Practical Aspects*, Springer, NY.
- [BFG07] Blatt, D., Hero, A. O., Gauchman, H., 2007. “A Convergent Incremental Gradient Method with a Constant Step Size,” *SIAM J. on Optimization*, Vol. 18, pp. 29-51.
- [BMN01] Ben-Tal, A., Margalit, T., and Nemirovski, A., 2001. “The Ordered Subsets Mirror Descent Optimization Method and its Use for the Positron Emission Tomography Reconstruction,” in *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, Eds., Butnariu, D., Censor, Y., and Reich, S., Elsevier Science, Amsterdam, Netherlands.
- [BNO03] Bertsekas, D. P., with Nedić, A., and Ozdaglar, A. E., 2003. *Convex Analysis and Optimization*, Athena Scientific, Belmont, MA.
- [BaW75] Balinski, M., and Wolfe, P., (Eds.), 1975. *Nondifferentiable Optimization*, Math. Programming Study 3, North-Holland, Amsterdam.
- [BeT89] Bertsekas, D. P., and Tsitsiklis, J. N., 1989. *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, N. J; republished by Athena Scientific, Belmont, MA, 1997.
- [BeT94] Bertsekas, D. P., and Tseng, P., 1994. “Partial Proximal Minimization Algorithms for Convex Programming,” *SIAM J. on Optimization*, Vol. 4, pp. 551-572.
- [BeT96] Bertsekas, D. P., and Tsitsiklis, J. N., 1996. *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA.
- [BeT00] Bertsekas, D. P., and Tsitsiklis, J. N., 2000. “Gradient Convergence in Gradient Methods,” *SIAM J. on Optimization*, Vol. 10, pp. 627-642.



- [BeT09] Beck, A., and Teboulle, M., 2009. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems, SIAM J. on Imaging Sciences, Vol. 2, pp. 183-202.
- [BeY09] Bertsekas, D. P., and Yu, H., 2009. "A Unifying Polyhedral Approximation Framework for Convex Optimization," Lab. for Info. and Decision Systems Report LIDS-P-2820, MIT, Cambridge, MA.
- [Ber75a] Bertsekas, D. P., 1975. "Necessary and Sufficient Conditions for a Penalty Method to be Exact," Math. Programming, Vol. 9, pp. 87-99.
- [Ber75b] Bertsekas, D. P., 1975. "Nondifferentiable Optimization Via Approximation," Math. Programming Study 3, Balinski, M., and Wolfe, P., (Eds.), North-Holland, Amsterdam, pp. 1-25.
- [Ber76] Bertsekas, D. P., 1976. "Multiplier Methods: A Survey," Automatica, Vol. 12, pp. 133-145.
- [Ber77] Bertsekas, D. P., 1977. "Approximation Procedures Based on the Method of Multipliers," J. Opt. Th. and Appl., Vol. 23, pp. 487-510.
- [Ber82] Bertsekas, D. P., 1982. Constrained Optimization and Lagrange Multiplier Methods, Academic Press, N. Y; republished by Athena Scientific, Belmont, MA, 1997.
- [Ber96] Bertsekas, D. P., 1996. "Incremental Least Squares Methods and the Extended Kalman Filter," SIAM J. on Optimization, Vol. 6, pp. 807-822.
- [Ber97] Bertsekas, D. P., 1997. "A New Class of Incremental Gradient Methods for Least Squares Problems," SIAM J. on Optimization, Vol. 7, pp. 913-926.
- [Ber98] Bertsekas, D. P., 1998. Network Optimization: Continuous and Discrete Models, Athena Scientific, Belmont, MA.
- [Ber99] Bertsekas, D. P., 1999. Nonlinear Programming: 2nd Edition, Athena Scientific, Belmont, MA.
- [Ber06] Bertsekas, D. P., 2006. "Extended Monotropic Programming and Duality," Lab. for Information and Decision Systems Report 2692, MIT, March 2006, corrected in Feb. 2010; a version appeared in JOTA, 2008, Vol. 139, pp. 209-225.
- [Ber10a] Bertsekas, D. P., 2010. "Incremental Proximal Methods for Large Scale Convex Optimization," Lab. for Information and Decision Systems Report LIDS-P-2847, MIT, August 2010.
- [Ber10b] Bertsekas, D. P., 2010. "Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: A Survey", Lab. for Information and Decision Systems Report LIDS-P-2848, MIT, August 2010.
- [BuQ98] Burke, J. V., and Qian, M., 1998. "A Variable Metric Proximal

- Point Algorithm for Monotone Operators,” *SIAM J. on Control and Optimization*, Vol. 37, pp. 353-375.
- [CFM75] Camerini, P. M., Fratta, L., and Maffioli, F., 1975. “On Improving Relaxation Methods by Modified Gradient Techniques,” *Math. Programming Studies*, Vol. 3, pp. 26-34.
- [CeZ92] Censor, Y., and Zenios, S. A., 1992. “The Proximal Minimization Algorithm with D-Functions,” *J. Opt. Theory and Appl.*, Vol. 73, pp. 451-464.
- [CeZ97] Censor, Y., and Zenios, S. A., 1997. *Parallel Optimization: Theory, Algorithms, and Applications*, Oxford University Press, N. Y.
- [ChG59] Cheney, E. W., and Goldstein, A. A., 1959. “Newton’s Method for Convex Programming and Tchebycheff Approximation,” *Numer. Math.*, Vol. I, pp. 253-268.
- [ChT93] Chen, G., and Teboulle, M., 1993. “Convergence Analysis of a Proximal-Like Minimization Algorithm Using Bregman Functions,” *SIAM J. on Optimization*, Vol. 3, pp. 538-543.
- [ChT94] Chen, G., and Teboulle, M., 1994. “A Proximal-Based Decomposition Method for Convex Minimization Problems,” *Math. Programming*, Vol. 64, pp. 81-101.
- [CoL94] Correa, R., and Lemarechal, C., 1994. “Convergence of Some Algorithms for Convex Minimization,” *Math. Programming*, Vol. 62, pp. 261-276.
- [Dav76] Davidon, W. C., 1976. “New Least Squares Algorithms,” *J. Opt. Theory and Appl.*, Vol. 18, pp. 187-197.
- [EcB92] Eckstein, J., and Bertsekas, D. P., 1992. “On the Douglas-Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators,” *Math. Programming*, Vol. 55, pp. 293-318.
- [Eck94a] Eckstein, J., 1994. “Nonlinear Proximal Point Algorithms Using Bregman Functions, with Applications to Convex Programming,” *Math. Operations Res.*, Vol. 18, pp. 202-226.
- [ElM75] Elzinga, J., and Moore, T. G., 1975. “A Central Cutting Plane Algorithm for the Convex Programming Problem,” *Math. Programming*, Vol. 8, pp. 134-145.
- [Erm83] Ermoliev, Yu. M., 1983. “Stochastic Quasigradient Methods and Their Application to System Optimization,” *Stochastics*, Vol. 9, pp. 1-36.
- [FaP03] Facchinei, F., and Pang, J. S., 2003. *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Vol. II, Springer-Verlag, N. Y.
- [FeK00] Feltenmark, S., and Kiwiel, K. C., 2000. “Dual Applications of Proximal Bundle Methods, Including Lagrangian Relaxation of Nonconvex

- Problems.” *SIAM J. Optimization*, Vol. 10, 697-721.
- [Fri56] Frisch, M. R., 1956. “La Resolution des Problemes de Programme Lineaire par la Methode du Potential Logarithmique,” *Cahiers du Seminaire D’Econometrie*, Vol. 4, pp. 7-20.
- [GZL02] Guan, X. H., Zhai, Q. Z., and Lai, F., 2002. “New Lagrangian Relaxation Based Algorithm for Resource Scheduling with Homogeneous Subproblems,” *J. Opt. Theory and Appl.*, Vol. 113, pp. 6582
- [Gai94] Gaivoronski, A. A., 1994. “Convergence Analysis of Parallel Backpropagation Algorithm for Neural Networks,” *Optimization Methods and Software*, Vol. 4, pp. 117-134.
- [GoV02] Goffin, J. L., and Vial, J. P., 2002. “Convex Nondifferentiable Optimization: A Survey Focussed on the Analytic Center Cutting Plane Method,” *Optimization Methods and Software*, Vol. 17, pp. 805-867.
- [Gri94] Grippo, L., 1994. “A Class of Unconstrained Minimization Methods for Neural Network Training,” *Optimization Methods and Software*, Vol. 4, pp. 135-150.
- [Gul91] Guler, O., 1991. “On the Convergence of the Proximal Point Algorithm for Convex Minimization,” *SIAM J. Control Optim.*, Vol. 29, pp. 403-419.
- [Gul92] Guler, O., 1992. “New Proximal Point Algorithms for Convex Minimization,” *SIAM J. on Optimization*, Vol. 2, pp. 649-664.
- [HaB70] Haarhoff, P. C., and Buys, J. D., 1970. “A New Method for the Optimization of a Nonlinear Function Subject to Nonlinear Constraints,” *Computer J.*, Vol. 13, pp. 178-184.
- [HLV87] Hearn, D. W., Lawphongpanich, S., and Ventura, J. A., 1987. “Restricted Simplicial Decomposition: Computation and Extensions,” *Math. Programming Studies*, Vol. 31, pp. 119-136.
- [Hes69] Hestenes, M. R., 1969. “Multiplier and Gradient Methods,” *J. Opt. Th. and Appl.*, Vol. 4, pp. 303-320.
- [HiL93] Hiriart-Urruty, J.-B., and Lemarechal, C., 1993. *Convex Analysis and Minimization Algorithms*, Vols. I and II, Springer-Verlag, Berlin and N. Y.
- [Hoh77] Hohenbalken, B. von, 1977. “Simplicial Decomposition in Nonlinear Programming,” *Math. Programming*, Vol. 13, pp. 49-68.
- [Hol74] Holloway, C. A., 1974. “An Extension of the Frank and Wolfe Method of Feasible Directions,” *Math. Programming*, Vol. 6, pp. 14-27.
- [IST94] Iusem, A. N., Svaiter, B., and Teboulle, M., 1994. “Entropy-Like Proximal Methods in Convex Programming,” *Math. Operations Res.*, Vol. 19, pp. 790-814.

- [Ius99] Iusem, A. N., 1999. "Augmented Lagrangian Methods and Proximal Point Methods for Convex Minimization," *Investigacion Operativa*.
- [KaC98] Kaskavelis, C. A., and Caramanis, M. C., 1998. "Efficient Lagrangian Relaxation Algorithms for Industry Size Job-Shop Scheduling Problems," *IIE Transactions on Scheduling and Logistics*, Vol. 30, pp. 1085–1097.
- [Kel60] Kelley, J. E., 1960. "The Cutting-Plane Method for Solving Convex Programs," *J. Soc. Indust. Appl. Math.*, Vol. 8, pp. 703-712.
- [Kib79] Kibardin, V. M., 1979. "Decomposition into Functions in the Minimization Problem," *Automation and Remote Control*, Vol. 40, pp. 1311-1323.
- [KoB72] Kort, B. W., and Bertsekas, D. P., 1972. "A New Penalty Function Method for Constrained Minimization," *Proc. 1972 IEEE Confer. Decision Control*, New Orleans, LA, pp. 162-166.
- [KoB76] Kort, B. W., and Bertsekas, D. P., 1976. "Combined Primal-Dual and Penalty Methods for Convex Programming," *SIAM J. on Control and Optimization*, Vol. 14, pp. 268-294.
- [Kor75] Kort, B. W., 1975. "Combined Primal-Dual and Penalty Function Algorithms for Nonlinear Programming," Ph.D. Thesis, Dept. of Engineering-Economic Systems, Stanford Univ., Stanford, Ca.
- [LMY08] Lu, Z., Monteiro, R. D. C., and Yuan, M., 2008. "Convex Optimization Methods for Dimension Reduction and Coefficient Estimation in Multivariate Linear Regression," Report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.
- [LPS98] Larsson, T., Patriksson, M., and Stromberg, A.-B., 1998. "Ergodic Convergence in Subgradient Optimization," *Optimization Methods and Software*, Vol. 9, pp. 93-120.
- [LVB98] Lobo, M. S., Vandenberghe, L., Boyd, S., and Lebret, H., 1998. "Applications of Second-Order Cone Programming," *Linear Algebra and Applications*, Vol. 284, pp. 193-228.
- [LeS93] Lemaréchal, C., and Sagastizábal, C., 1993. "An Approach to Variable Metric Bundle Methods," in *Systems Modelling and Optimization*, Proc. of the 16th IFIP-TC7 Conference, Compiègne, Henry, J., and Yvon, J.-P., (Eds.), *Lecture Notes in Control and Information Sciences* 197, pp. 144-162.
- [LuT94] Luo, Z. Q., and Tseng, P., 1994. "Analysis of an Approximate Gradient Projection Method with Applications to the Backpropagation Algorithm," *Optimization Methods and Software*, Vol. 4, pp. 85-101.
- [Luq84] Luque, F.J., 1984. "Asymptotic Convergence Analysis of the Proximal Point Algorithm," *SIAM J. on Control and Optimization*, Vol. 22, pp.

277-293.

[Luo91] Luo, Z. Q., 1991. "On the Convergence of the LMS Algorithm with Adaptive Learning Rate for Linear Feedforward Networks," *Neural Computation*, Vol. 3, pp. 226-245.

[MSQ98] Mifflin, R., Sun, D., and Qi, L., 1998. "Quasi-Newton Bundle-Type Methods for Nondifferentiable Convex Optimization," *SIAM J. on Optimization*, Vol. 8, pp. 583-603.

[MaS94] Mangasarian, O. L., and Solodov, M. V., 1994. "Serial and Parallel Backpropagation Convergence Via Nonmonotone Perturbed Minimization," *Optimization Methods and Software*, Vol. 4, pp. 103-116.

[Mar70] Martinet, B., 1970. "Regularisation d' Inequations Variationelles par Approximations Successives," *Rev. Francaise Inf. Rech. Oper.*, Vol. 4, pp. 154-159.

[Mar72] Martinet, B., 1972. "Determination Approchee d'un Point Fixe d'une Application Pseudocontractante," *C. R. Acad. Sci. Paris*, Vol. 274A, pp. 163-165.

[Mif96] Mifflin, R., 1996. "A Quasi-Second-Order Proximal Bundle Algorithm," *Math. Programming*, Vol. 73, pp. 51-72.

[Min86] Minoux, M., 1986. *Mathematical Programming: Theory and Algorithms*, Wiley, N. Y.

[NBB01] Nedić, A., Bertsekas, D. P., and Borkar, V. S., 2001. "Distributed Asynchronous Incremental Subgradient Methods," in *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, Butnariu, D., Censor, Y., and Reich, S., (Eds.), Elsevier Science, Amsterdam, Netherlands.

[NeB01a] Nedić, A., and Bertsekas, D. P., 2001. "Incremental Subgradient Methods for Nondifferentiable Optimization," *SIAM J. on Optim.*, Vol. 12, pp. 109-138.

[NeB01b] Nedić, A., and Bertsekas, D. P., 2001. "Convergence Rate of Incremental Subgradient Algorithms," in *Stochastic Optimization: Algorithms and Applications*, Uryasev, S., and Pardalos, P. M., (Eds.), Kluwer Academic Publishers, Dordrecht, Netherlands, pp. 223-264.

[NeB07] Nedić, A., and Bertsekas, D. P., 2007. "The Effect of Deterministic Noise in Subgradient Methods," *Lab. for Information and Decision Systems Report*, MIT, *Math. Programming*, to appear.

[NeN94] Nesterov, Y., and Nemirovskii, A., 1994. *Interior Point Polynomial Algorithms in Convex Programming*, SIAM, *Studies in Applied Mathematics* 13, Phila., PA.

- [NeW88] Nemhauser, G. L., and Wolsey, L. A., 1988. *Integer and Combinatorial Optimization*, Wiley, N. Y.
- [NeY83] Nemirovsky, A., and Yudin, D. B., 1983. *Problem Complexity and Method Efficiency*, Wiley, N. Y.
- [Nes83] Nesterov, Y., 1983. "A Method for Unconstrained Convex Minimization Problem with the Rate of Convergence  $O(1/k^2)$ ", *Doklady AN SSSR* 269, pp. 543-547; translated as *Soviet Math. Dokl.*
- [Nes04] Nesterov, Y., 2004. *Introductory Lectures on Convex Optimization*, Kluwer Academic Publisher, Dordrecht, The Netherlands.
- [Nes05] Nesterov, Y., 2005. "Smooth Minimization of Nonsmooth Functions," *Math. Programming*, Vol. 103 pp. 127-152.
- [Nur74] Nurminskii, E. A., 1974. "Minimization of Nondifferentiable Functions in Presence of Noise," *Kibernetika*, Vol. 10, pp. 59-61.
- [OrR70] Ortega, J. M., and Rheinboldt, W. C., 1970. *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, N. Y.
- [Pap81] Papavassilopoulos, G., 1981. "Algorithms for a Class of Nondifferentiable Problems," *J. Opt. Th. and Appl.*, Vol. 34, pp. 41-82.
- [Pen02] Pennanen, T., 2002. "Local Convergence of the Proximal Point Algorithm and Multiplier Methods without Monotonicity," *Mathematics of Operations Research*, Vol. 27, pp. 170-191.
- [PoT74] Poljak, B. T., and Tretjakov, N. V., 1974. "An Iterative Method for Linear Programming and its Economic Interpretation," *Matecon*, Vol. 10, pp. 81-100.
- [PoT97] Polyak, R., and Teboulle, M., 1997. "Nonlinear Rescaling and Proximal-Like Methods in Convex Optimization," *Math. Programming*, Vol. 76, pp. 265-284.
- [Pol64] Poljak, B. T., 1964. "Some Methods of Speeding up the Convergence of Iteration Methods," *Z. Vyčisl. Mat. i Mat. Fiz.*, Vol. 4, pp. 1-17.
- [Pol79] Poljak, B. T., 1979. "On Bertsekas' Method for Minimization of Composite Functions," *Internat. Symp. Systems Opt. Analysis*, Benoussan, A., and Lions, J. L., (Eds.), pp. 179-186, Springer-Verlag, Berlin and N. Y.
- [Pol87] Polyak B. T., *Introduction to Optimization*, Optimization Software Inc., N.Y., 1987.
- [Roc73b] Rockafellar, R. T., 1973. "The Multiplier Method of Hestenes and Powell Applied to Convex Programming," *J. Opt. Th. and Appl.*, Vol. 12, pp. 555-562.
- [Pow69] Powell, M. J. D., 1969. "A Method for Nonlinear Constraints in Minimizing Problems," in *Optimization*, Fletcher, R., (Ed.), Academic

Press, N. Y, pp. 283-298.

[RaN05] Rabbat M. G. and Nowak R. D., "Quantized incremental algorithms for distributed optimization," *IEEE Journal on Select Areas in Communications*, Vol. 23, No. 4, 2005, pp. 798-808.

[Roc73] Rockafellar, R. T., 1973. "A Dual Approach to Solving Nonlinear Programming Problems by Unconstrained Optimization," *Math. Programming*, pp. 354-373.

[Roc76a] Rockafellar, R. T., 1976. "Monotone Operators and the Proximal Point Algorithm," *SIAM J. on Control and Optimization*, Vol. 14, pp. 877-898.

[Roc76b] Rockafellar, R. T., 1976. "Solving a Nonlinear Programming Problem by Way of a Dual Problem," *Symp. Matematica*, Vol. 27, pp. 135-160.

[Roc84] Rockafellar, R. T., 1984. *Network Flows and Monotropic Optimization*, Wiley, N. Y.; republished by Athena Scientific, Belmont, MA, 1998.

[Rus89] Ruszczynski, A., 1989. "An Augmented Lagrangian Decomposition Method for Block Diagonal Linear Programming Problems," *Operations Res. Letters*, Vol. 8, pp. 287-294.

[Sho85] Shor, N. Z., 1985. *Minimization Methods for Nondifferentiable Functions*, Springer-Verlag, Berlin.

[Sho98] Shor, N. Z., 1998. *Nondifferentiable Optimization and Polynomial Problems*, Kluwer Academic Publishers, Dordrecht, Netherlands.

[SoZ98] Solodov, M. V., and Zavriev, S. K., 1998. "Error Stability Properties of Generalized Gradient-Type Algorithms," *J. Opt. Theory and Appl.*, Vol. 98, pp. 663-680.

[Sol98] Solodov, M. V., 1998. "Incremental Gradient Algorithms with Stepsizes Bounded Away from Zero," *Computational Optimization and Applications*, Vol. 11, pp. 23-35.

[Str97] Stromberg, A-B., 1997. *Conditional Subgradient Methods and Ergodic Convergence in Nonsmooth Optimization*, Ph.D. Thesis, Univ. of Linkoping, Sweden.

[TBA86] Tsitsiklis, J. N., Bertsekas, D. P., and Athans, M., 1986. "Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms," *IEEE Trans. on Aut. Control*, Vol. AC-31, pp. 803-812.

[TsB93] Tseng, P., and Bertsekas, D. P., 1993. "On the Convergence of the Exponential Multiplier Method for Convex Programming," *Math. Programming*, Vol. 60, pp. 1-19.

[TsB00] Tseng, P., and Bertsekas, D. P., 2000. "An Epsilon-Relaxation Method for Separable Convex Cost Generalized Network Flow Problems,"

Math. Programming, Vol. 88, pp. 85-104.

[Tse98] Tseng, P., 1998. "Incremental Gradient(-Projection) Method with Momentum Term and Adaptive Stepsize Rule," *SIAM J. on Optimization*, Vol. 8, pp. 506-531.

[Tse01] Tseng, P., 2001. "An Epsilon Out-of-Kilter Method for Monotropic Programming," *Math. Oper. Res.*, Vol. 26, pp. 221-233.

[Tse08] Tseng, P., 2008. "On Accelerated Proximal Gradient Methods for Convex-Concave Optimization," Report, Math. Dept., Univ. of Washington.

[VeH93] Ventura, J. A., and Hearn, D. W., 1993. "Restricted Simplicial Decomposition for Convex Constrained Problems," *Math. Programming*, Vol. 59, pp. 71-85.

[WSV00] Wolkowicz, H., Saigal, R., and Vanderbergue, L., (eds), 2000. *Handbook of Semidefinite Programming*, Kluwer, Boston.

[Wol75] Wolfe, P., 1975. "A Method of Conjugate Subgradients for Minimizing Nondifferentiable Functions," *Math. Programming Study 3*, Balinski, M., and Wolfe, P., (Eds.), North-Holland, Amsterdam, pp. 145-173.

[Wri97] Wright, S. J., 1997. *Primal-Dual Interior Point Methods*, SIAM, Phila., PA.

[Ye97] Ye, Y., 1997. *Interior Point Algorithms: Theory and Analysis*, Wiley Interscience, N. Y.

[ZLW99] Zhao, X., Luh, P. B., and Wang, J., 1999. "Surrogate Gradient Algorithm for Lagrangian Relaxation," *J. Opt. Theory and Appl.*, Vol. 100, pp. 699-712.

[ZhL02] Zhao, X., and Luh, P. B., 2002. "New Bundle Methods for Solving Lagrangian Relaxation Dual Problems," *J. Opt. Theory and Appl.*, Vol. 113, pp. 373-397.