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

December 19, 2014

6
Convex Optimization Algorithms

Contents

In this supplementary chapter, we discuss several algorithmic approaches 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 algorithms that we discuss do not require differentiability for their application. We refer to general nonlinear programming textbooks for methods (e.g., [Ber99]) that rely on differentiability, such as gradient and Newton-type methods.

6.1 CONVEX OPTIMIZATION MODELS: AN OVERVIEW

We begin with a broad overview of some important types of convex optimization problems, and some of their principal characteristics. Convex optimization algorithms have a broad range of applications, 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 chapter we provide an overview and discuss some examples in Sections 6.1.1 and 6.1.2, respectively. In Sections 6.1.3 and 6.1.4, we discuss some additional structures involving a large number of additive terms in the cost, or a large number of constraints. These types of problems also arise often in the context of duality. Finally, in Section 6.1.5, we discuss an important technique, based on conjugacy and duality, whereby we can transform convex constrained optimization problems to equivalent unconstrained (or less constrained) ones.

6.1.1 Lagrange Dual Problems

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

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

where X is a set, $g(x) = \big(g_1(x), \ldots, g_r(x)\big)'$, and $f : X \mapsto \Re$ and $g_j : X \mapsto \Re$, $j = 1, \ldots, r$, are given functions. We refer to this as the *primal problem*, and we denote its optimal value by f^*.

The dual problem is

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

where the dual function q is given by

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

and L is the Lagrangian function defined by

$$L(x, \mu) = f(x) + \mu' g(x), \qquad x \in X,\ \mu \in \Re^r.$$

The dual optimal value is

$$q^* = \sup_{\mu \in \Re^r} q(\mu).$$

The weak duality relation $q^* \leq f^*$ is easily shown by writing for all $\mu \geq 0$, and $x \in X$ with $g(x) \leq 0$,

$$q(\mu) = \inf_{z \in X} L(z, \mu) \leq f(x) + \sum_{j=1}^{r} \mu_j g_j(x) \leq f(x),$$

so

$$q^* = \sup_{\mu \geq 0} q(\mu) \leq \inf_{x \in X,\, g(x) \leq 0} f(x) = f^*.$$

We state this formally as follows.

Proposition 6.1.1: (Weak Duality Theorem) For any feasible solutions x and μ of the primal and dual problems, respectively, we have $q(\mu) \leq f(x)$. Moreover, $q^* \leq f^*$.

Generally the solution process is simplified when strong duality holds. The following strong duality result has been shown in Prop. 5.3.1.

Proposition 6.1.2: (Convex Programming Duality - Existence of Dual Optimal Solutions) Consider the problem (6.1). Assume that f^* is finite, and that one of the following two conditions holds:

(1) There exists $\overline{x} \in X$ such that $g_j(\overline{x}) < 0$ for all $j = 1, \ldots, r$.

(2) The functions g_j, $j = 1, \ldots, r$, are affine, and there exists $\overline{x} \in \mathrm{ri}(X)$ such that $g(\overline{x}) \leq 0$.

Then $q^* = f^*$ and the set of optimal solutions of the dual problem is nonempty. Under condition (1) this set is also compact.

The following proposition gives necessary and sufficient conditions for optimality (see Prop. 5.3.2).

Proposition 6.1.3: (Optimality Conditions) Consider the problem (6.1). There holds $q^* = f^*$, and (x^*, μ^*) are a primal and dual optimal solution pair if and only if x^* is feasible, $\mu^* \geq 0$, and

$$x^* \in \arg\min_{x \in X} L(x, \mu^*), \qquad \mu_j^* g_j(x^*) = 0, \quad j = 1, \ldots, r. \tag{6.4}$$

Partially Polyhedral Constraints

The preceding results for the inequality-constrained problem (6.1) can be refined by making more specific assumptions regarding available polyhedral structure in the constraint functions and the abstract constraint set X. Let us first consider an extension of problem (6.1) where there are additional linear equality constraints:

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in X, \quad g(x) \leq 0, \quad Ax = b, \end{aligned} \tag{6.5}$$

where X is a convex set, $g(x) = \big(g_1(x), \ldots, g_r(x)\big)'$, $f : X \mapsto \Re$ and $g_j : X \mapsto \Re$, $j = 1, \ldots, r$, are convex functions, A is an $m \times n$ matrix, and $b \in \Re^m$. We can deal with this problem by simply converting the constraint $Ax = b$ to the equivalent set of linear inequality constraints

$$Ax \leq b, \qquad -Ax \leq -b, \tag{6.6}$$

with corresponding dual variables $\lambda^+ \geq 0$ and $\lambda^- \geq 0$. The Lagrangian function is

$$f(x) + \mu' g(x) + (\lambda^+ - \lambda^-)'(Ax - b),$$

and by introducing a dual variable

$$\lambda = \lambda^+ - \lambda^- \tag{6.7}$$

with no sign restriction, it can be written as

$$L(x, \mu, \lambda) = f(x) + \mu' g(x) + \lambda'(Ax - b).$$

The dual problem is

$$\begin{aligned} &\text{maximize} \quad \inf_{x \in X} L(x, \mu, \lambda) \\ &\text{subject to} \quad \mu \geq 0, \ \lambda \in \Re^m. \end{aligned}$$

The following is the standard duality result; see Prop. 5.3.5.

Proposition 6.1.4: (Convex Programming - Linear Equality and Nonlinear Inequality Constraints) Consider problem (6.5). Assume that f^* is finite, that there exists $\overline{x} \in X$ such that $A\overline{x} = b$ and $g(\overline{x}) < 0$, and that there exists $\tilde{x} \in \text{ri}(X)$ such that $A\tilde{x} = b$. Then $q^* = f^*$ and there exists at least one dual optimal solution.

In the special case of a problem with just linear equality constraints:

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in X, \quad Ax = b, \end{aligned} \tag{6.8}$$

the Lagrangian function is

$$L(x, \lambda) = f(x) + \lambda'(Ax - b),$$

and the dual problem is

$$\begin{aligned} &\text{maximize} \quad \inf_{x \in X} L(x, \lambda) \\ &\text{subject to} \quad \lambda \in \Re^m. \end{aligned}$$

The corresponding duality result is given as Prop. 5.3.3, and for the case where there are additional linear inequality constraints, as Prop. 5.3.4.

Discrete Optimization and Lower Bounds

The preceding propositions deal with situations where the most favorable form of duality ($q^* = f^*$) holds. However, duality can be useful even when there is duality gap, as often occurs in problems of the form (6.1) that 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

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

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

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

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 fixed at either 0 or 1 (i.e., 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 μ of this problem and the corresponding dual value

$$q(\mu) = \inf_{x \in \tilde{X}} \big\{ f(x) + \mu' g(x) \big\}, \tag{6.9}$$

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

Separable Problems - Decomposition

Let us now discuss an important problem structure that involves Lagrange duality, and arises frequently in applications. Consider the problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{n} f_i(x_i) \\ &\text{subject to} \quad a_j' x \leq b_j, \quad j = 1, \ldots, r, \end{aligned} \tag{6.10}$$

where $x = (x_1, \ldots, x_n)$, each $f_i : \Re \mapsto \Re$ 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 μ_j to the constraint $a_j' x \leq b_j$, we obtain the dual problem [cf. Eq. (6.2)]

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{n} q_i(\mu) - \sum_{j=1}^{r} \mu_j b_j \\ &\text{subject to} \quad \mu \geq 0, \end{aligned} \tag{6.11}$$

where

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

and $\mu = (\mu_1, \ldots, \mu_r)$. 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. (6.10). An example is when x has m components $x_1, \ldots, x_m$ of dimensions $n_1, \ldots, n_m$, respectively, and the problem has the form

$$\begin{aligned} &\text{minimize} \ \ \sum_{i=1}^m f_i(x_i) \\ &\text{subject to} \ \ \sum_{i=1}^m g_i(x_i) \le 0, \quad x_i \in X_i, \ \ i = 1, \ldots, m, \end{aligned} \tag{6.12}$$

where $f_i : \Re^{n_i} \mapsto \Re$ and $g_i : \Re^{n_i} \mapsto \Re^r$ are given functions, and X_i are given subsets of $\Re^{n_i}$. The advantage of convenient computation of the dual function value using decomposition extends to such problems as well. We may also note that when the components $x_1, \ldots, x_m$ are one-dimensional, and the functions f_i and sets X_i are convex, there is a particularly favorable strong duality result for the separable problem (6.12), even when the constraint functions g_i are nonlinear but consist of convex components $g_{ij} : \Re \mapsto \Re$, $j = 1, \ldots, r$; see Tseng [Tse09].

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

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

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.

6.1.2 Fenchel Duality and Conic Programming

We recall the Fenchel duality framework from Section 5.3.5. It involves the problem

$$\begin{aligned} &\text{minimize} \quad f_1(x) + f_2(Ax) \\ &\text{subject to} \;\; x \in \Re^n, \end{aligned} \tag{6.13}$$

where A is an $m \times n$ matrix, $f_1 : \Re^n \mapsto (-\infty, \infty]$ and $f_2 : \Re^m \mapsto (-\infty, \infty]$ are closed convex functions, and we assume that there exists a feasible solution. The dual problem, after a sign change to convert it to a minimization problem, can be written as

$$\begin{aligned} &\text{minimize} \quad f_1^\star(A'\lambda) + f_2^\star(-\lambda) \\ &\text{subject to} \;\; \lambda \in \Re^m, \end{aligned} \tag{6.14}$$

where $f_1^\star$ and $f_2^\star$ are the conjugate functions of f_1 and f_2. We denote by f^* and q^* the optimal primal and dual values. The following is given as Prop. 5.3.8.

Proposition 6.1.5: (Fenchel Duality)

(a) If f^* is finite and $\big(A \cdot \text{ri}\big(\text{dom}(f_1)\big)\big) \cap \text{ri}\big(\text{dom}(f_2)\big) \neq \emptyset$, then $f^* = q^*$ and there exists at least one dual optimal solution.

(b) There holds $f^* = q^*$, and (x^*, λ^*) is a primal and dual optimal solution pair if and only if

$$x^* \in \arg\min_{x \in \Re^n} \big\{ f_1(x) - x'A'\lambda^* \big\} \quad \text{and} \quad Ax^* \in \arg\min_{z \in \Re^n} \big\{ f_2(z) + z'\lambda^* \big\}. \tag{6.15}$$

An important problem structure, which can be analyzed as a special case of the Fenchel duality framework is the conic programming problem discussed in Section 5.3.6:

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \;\; x \in C, \end{aligned} \tag{6.16}$$

where $f : \Re^n \mapsto (-\infty, \infty]$ is a closed proper convex function and C is a closed convex cone in $\Re^n$.

Indeed, let us apply Fenchel duality with A equal to the identity and the definitions

$$f_1(x) = f(x), \qquad f_2(x) = \begin{cases} 0 & \text{if } x \in C, \\ \infty & \text{if } x \notin C. \end{cases}$$

The corresponding conjugates are

$$f_1^\star(\lambda) = \sup_{x\in\Re^n}\big\{\lambda'x - f(x)\big\}, \qquad f_2^\star(\lambda) = \sup_{x\in C}\lambda'x = \begin{cases} 0 & \text{if } \lambda \in C^*, \\ \infty & \text{if } \lambda \notin C^*, \end{cases}$$

where

$$C^* = \{\lambda \mid \lambda'x \leq 0,\ \forall\ x \in C\}$$

is the polar cone of C (note that $f_2^\star$ is the support function of C; cf. Example 1.6.1). The dual problem [cf. Eq. (6.14)] is

$$\begin{aligned} &\text{minimize} && f^\star(\lambda) \\ &\text{subject to} && \lambda \in \hat{C}, \end{aligned} \tag{6.17}$$

where $f^\star$ is the conjugate of f and $\hat{C}$ is the negative polar cone (also called the *dual cone* of C):

$$\hat{C} = -C^* = \{\lambda \mid \lambda'x \geq 0,\ \forall\ x \in C\}.$$

Note the symmetry between the primal and dual problems (6.16) and (6.17). The strong duality relation $f^* = q^*$ can be written as

$$\inf_{x\in C} f(x) = -\inf_{\lambda\in\hat{C}} f^\star(\lambda).$$

The following proposition translates the conditions of Prop. 6.1.5 to guarantee that there is no duality gap and that the dual problem has an optimal solution (cf. Prop. 5.3.9).

Proposition 6.1.6: (Conic Duality Theorem) Assume that the optimal value of the primal conic problem (6.16) is finite, and that $\text{ri}\big(\text{dom}(f)\big) \cap \text{ri}(C) \neq \varnothing$. Then, there is no duality gap and the dual problem (6.17) has an optimal solution.

Using the symmetry of the primal and dual problems, we also obtain that there is no duality gap and the primal problem (6.16) has an optimal solution if the optimal value of the dual conic problem (6.17) is finite and $\text{ri}\big(\text{dom}(f^\star)\big) \cap \text{ri}(\hat{C}) \neq \varnothing$. It is also possible to exploit polyhedral structure in f and/or C, using Prop. 5.3.6. Furthermore, we may derive primal and dual optimality conditions using Prop. 6.1.5(b).

Linear-Conic Problems

An important special case of conic programming, called *linear-conic problem*, arises when $\mathrm{dom}(f)$ is affine and f is linear over $\mathrm{dom}(f)$, 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

$$\begin{aligned} &\text{minimize} \quad c'x \\ &\text{subject to} \quad x - b \in S, \quad x \in C. \end{aligned} \tag{6.18}$$

To derive the dual problem, we note that

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

It can be seen that the dual problem (6.17), after discarding the superfluous term $c'b$ from the cost, can be written as

$$\begin{aligned} &\text{minimize} \quad b'\lambda \\ &\text{subject to} \quad \lambda - c \in S^\perp, \quad \lambda \in \hat{C}. \end{aligned} \tag{6.19}$$

Figure 6.1.1 illustrates the primal and dual linear-conic problems.

The following proposition translates the conditions of Prop. 6.1.6 to the linear-conic duality context.

Proposition 6.1.7: (Linear-Conic Duality Theorem) Assume that the primal problem (6.18) has finite optimal value. Assume further that either $(b + S) \cap \mathrm{ri}(C) \neq \emptyset$ or C is polyhedral. Then, there is no duality gap and the dual problem has an optimal solution.

Proof: Under the condition $(b + S) \cap \mathrm{ri}(C) \neq \emptyset$, the result follows from Prop. 6.1.6. For the case where C is polyhedral, the result follows from the more refined version of the Fenchel duality theorem, discussed at the end of Section 5.3.5. **Q.E.D.**

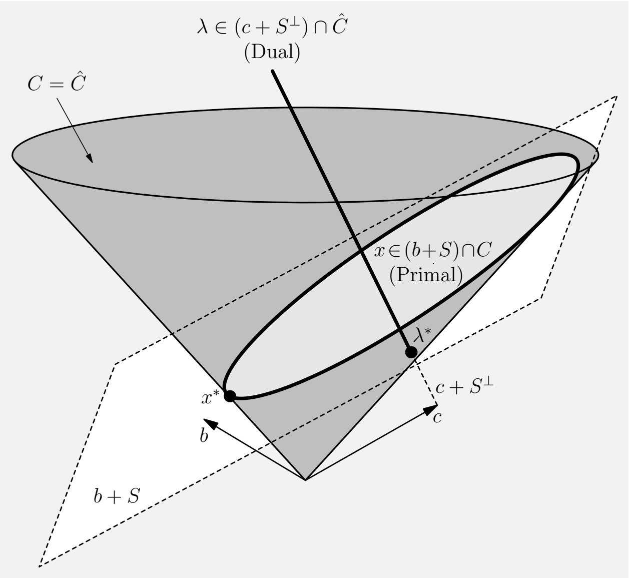

Figure 6.1.1. Illustration of primal and dual linear-conic problems for the case of a 3-dimensional problem, 2-dimensional subspace S, and a self-dual cone ($C = \hat{C}$); cf. Eqs. (6.18) and (6.19).

Special Forms of Linear-Conic Problems

The primal and dual linear-conic problems (6.18) and (6.19) 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:

$$\min_{Ax=b,\ x\in C} c'x \qquad \Longleftrightarrow \qquad \max_{c-A'\lambda\in\hat{C}} b'\lambda, \tag{6.20}$$

$$\min_{Ax-b\in C} c'x \qquad \Longleftrightarrow \qquad \max_{A'\lambda=c,\ \lambda\in\hat{C}} b'\lambda, \tag{6.21}$$

where $x \in \Re^n$, $\lambda \in \Re^m$, $c \in \Re^n$, $b \in \Re^m$, and A is an $m \times n$ matrix.

To verify the duality relation (6.20), let $\overline{x}$ be any vector such that $A\overline{x} = b$, and let us write the primal problem on the left in the primal conic form (6.18) as

$$\begin{aligned} &\text{minimize} \quad c'x \\ &\text{subject to} \quad x - \overline{x} \in \mathrm{N}(A), \quad x \in C, \end{aligned} \tag{6.22}$$

where $\mathrm{N}(A)$ is the nullspace of A. The corresponding dual conic problem (6.19) is to solve for μ the problem

$$\begin{aligned} &\text{minimize} \quad \overline{x}'\mu \\ &\text{subject to} \quad \mu - c \in \mathrm{N}(A)^\perp, \quad \mu \in \hat{C}. \end{aligned} \tag{6.23}$$

Since $\mathrm{N}(A)^\perp$ is equal to $\mathrm{Ra}(A')$, the range of A', the constraints of problem (6.23) can be equivalently written as $c-\mu \in -\mathrm{Ra}(A') = \mathrm{Ra}(A')$, $\mu \in \hat{C}$, or

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

for some $\lambda \in \Re^m$. Making the change of variables $\mu = c - A'\lambda$, the dual problem (6.23) can be written as

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

By discarding the constant $\overline{x}'c$ from the cost function, using the fact $A\overline{x} = b$, 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 (6.20). The duality relation (6.21) 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

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

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

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

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

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

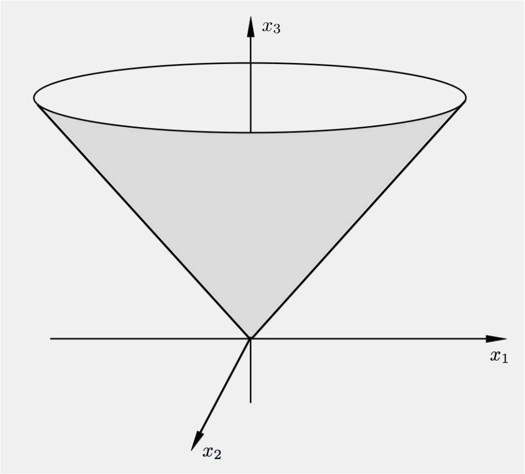

Figure 6.1.2. The second order cone in $\Re^3$:

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

Combining the last two relations, we have

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

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.

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

$$\begin{aligned} &\text{minimize} \quad c'x \\ &\text{subject to} \quad A_i x - b_i \in C_i, \ i = 1, \ldots, m, \end{aligned} \tag{6.24}$$

where $x \in \Re^n$, c is a vector in $\Re^n$, and for $i = 1, \ldots, m$, A_i is an $n_i \times n$ matrix, b_i is a vector in $\Re^{n_i}$, and C_i is the second order cone of $\Re^{n_i}$. It is seen to be a special case of the primal problem in the left-hand side of the duality relation (6.21), where

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

Thus from the right-hand side of the duality relation (6.21), and the self-duality relation $C = \hat{C}$, the corresponding dual linear-conic problem has the form

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^{m} b_i'\lambda_i \\ &\text{subject to} && \sum_{i=1}^{m} A_i'\lambda_i = c, \quad \lambda_i \in C_i,\ i = 1,\ldots,m, \end{aligned} \tag{6.25}$$

where $\lambda = (\lambda_1,\ldots,\lambda_m)$. By applying the duality result of Prop. 6.1.7, we have the following.

Proposition 6.1.8: (Second Order Cone Duality Theorem) Consider the primal SOCP (6.24), and its dual problem (6.25).

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

$$A_i\overline{x} - b_i \in \text{int}(C_i), \qquad i = 1,\ldots,m,$$

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 $\overline{\lambda} = (\overline{\lambda}_1,\ldots,\overline{\lambda}_m)$ such that

$$\overline{\lambda}_i \in \text{int}(C_i), \qquad i = 1,\ldots,m,$$

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

Note that while Prop. 6.1.7 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, to be discussed in Chapter 3. We refer to the literature for a more detailed description and analysis (see e.g., Ben-Tal and Nemirovski [BeT01], and Boyd and Vanderberghe [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 6.1.1: (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. This approach is also known as a set membership description of the uncertainty and has also been used in fields other than optimization, such as set membership estimation, and minimax control.

As an example, consider the problem

$$\begin{aligned} &\text{minimize } \ c'x \\ &\text{subject to } \ a_j'x \le b_j, \quad \forall\, (a_j, b_j) \in T_j, \quad j = 1, \ldots, r, \end{aligned} \tag{6.26}$$

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

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

$$a_j'x \le b_j, \ \ \forall\, (a_j, b_j) \in T_j \qquad \text{if and only if} \qquad g_j(x) \le 0,$$

where

$$g_j(x) = \sup_{(a_j, b_j) \in T_j} \{a_j'x - b_j\}. \tag{6.27}$$

Thus, the robust linear programming problem (6.26) is equivalent to

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

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 $g_j(x) \le 0$ can be expressed in terms of a second order cone. To see this, let

$$T_j = \big\{ (\overline{a}_j + P_j u_j, \overline{b}_j + q_j' u_j) \mid \|u_j\| \le 1,\, u_j \in \Re^{n_j} \big\}, \tag{6.28}$$

where P_j is a given $n \times n_j$ matrix, $\overline{a}_j \in \Re^n$ and $q_j \in \Re^{n_j}$ are given vectors, and $\overline{b}_j$ is a given scalar. Then, from Eqs. (6.27) and (6.28),

$$\begin{aligned} g_j(x) &= \sup_{\|u_j\| \le 1} \big\{ (\overline{a}_j + P_j u_j)'x - (\overline{b}_j + q_j' u_j) \big\} \\ &= \sup_{\|u_j\| \le 1} (P_j'x - q_j)'u_j + \overline{a}_j'x - \overline{b}_j, \end{aligned}$$

and finally

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

Thus,

$$g_j(x) \leq 0 \qquad \text{if and only if} \qquad (P_j'x - q_j, \overline{b}_j - \overline{a}_j'x) \in C_j,$$

where C_j is the second order cone of $\Re^{n_j+1}$; 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 (6.26) is a SOCP of the form (6.24).

Example 6.1.2: (Quadratically Constrained Quadratic Problems)

Consider the quadratically constrained quadratic problem

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

where $Q_0, \ldots, Q_r$ are symmetric $n \times n$ positive definite matrices, $q_0, \ldots, q_r$ are vectors in $\Re^n$, and $p_0, \ldots, p_r$ 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 $Q_j = 0$, $j = 1, \ldots, r$.

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

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

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

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

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

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

It can be seen that this problem has the second order cone form (6.24).

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. 6.1.8. 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

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

Let C be the cone of matrices that are positive semidefinite, called the *positive semidefinite cone*. The interior of C is the set of positive definite matrices.

The dual cone is

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

and it can be shown that $\hat{C} = C$, i.e., C is self-dual. Indeed, if $Y \notin C$, 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 $Y \notin \hat{C}$ and it follows that $C \supset \hat{C}$. Conversely, let $Y \in C$, 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 λ_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 $Y \in \hat{C}$, and $C \subset \hat{C}$.

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

$$\begin{aligned} &\text{minimize} && < D, X > \\ &\text{subject to} && < A_i, X >= b_i, \quad i = 1, \ldots, m, \quad X \in C, \end{aligned} \tag{6.29}$$

where D, $A_1, \ldots, A_m$, are given $n \times n$ symmetric matrices, and $b_1, \ldots, b_m$, are given scalars. It is seen to be a special case of the primal problem in the left-hand side of the duality relation (6.20).

The SDP is a fairly general problem. In particular, it can also be shown that a SOCP can be cast as a SDP (see the end-of-chapter exercises). 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 (6.19), as given by the right-hand side of the duality relation (6.20), takes the form

$$\begin{aligned} &\text{maximize} && b'\lambda \\ &\text{subject to} && D - (\lambda_1 A_1 + \cdots + \lambda_m A_m) \in C, \end{aligned} \tag{6.30}$$

where $b = (b_1, \ldots, b_m)$ and the maximization is over the vector $\lambda = (\lambda_1, \ldots, \lambda_m)$. By applying the duality result of Prop. 6.1.7, we have the following proposition.

Proposition 6.1.9: (Semidefinite Duality Theorem) Consider the primal problem (6.29), and its dual problem (6.30).

(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 $\overline{\lambda}_1, \ldots, \overline{\lambda}_m$ such that $D - (\overline{\lambda}_1 A_1 + \cdots + \overline{\lambda}_m A_m)$ is positive definite, then there is no duality gap, and the primal problem has an optimal solution.

Example 6.1.3: (Minimizing the Maximum Eigenvalue)

Given a symmetric $n \times n$ matrix $M(\lambda)$, which depends on a parameter vector $\lambda = (\lambda_1, \ldots, \lambda_m)$, we want to choose λ 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

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

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

$$M(\lambda) = M_0 + \lambda_1 M_1 + \cdots + \lambda_m M_m,$$

this problem has the form of the dual problem (6.30), with the optimization variables being $(z, \lambda_1, \ldots, \lambda_m)$.

Example 6.1.4: (Semidefinite Relaxation - Lower Bounds for Discrete Optimization Problems)

Semidefinite programming provides 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

$$\begin{aligned} &\text{minimize} \quad x'Q_0x + a_0'x + b_0 \\ &\text{subject to} \quad x'Q_ix + a_i'x + b_i = 0, \quad i = 1, \ldots, m, \end{aligned} \tag{6.31}$$

where $Q_0, \ldots, Q_m$ are symmetric $n \times n$ matrices, $a_0, \ldots, a_m$ are vectors in $\Re^n$, and $b_0, \ldots, b_m$ 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 $\lambda = (\lambda_1, \ldots, \lambda_m)$, the dual function is given by

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

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 (6.31) and its dual, and note that by weak duality, we have $f^* \geq q^*$. By introducing an auxiliary scalar variable ξ, we see that the dual problem is to find a pair (ξ, λ) that solves the problem

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

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

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

or equivalently, introducing a scalar variable t,

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

This relation can be equivalently written as

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

or

$$\begin{pmatrix} Q(\lambda) & \frac{1}{2}a(\lambda) \\ \frac{1}{2}a(\lambda)' & b(\lambda)-\xi \end{pmatrix} \in C, \tag{6.32}$$

where C is the positive semidefinite cone. Thus the dual problem is equivalent to the SDP of maximizing ξ over all (ξ,λ) satisfying the constraint (6.32), and its optimal value q^* is a lower bound to f^*.

6.1.3 Additive Cost Problems

In this section we focus on a structural characteristic that arises in several important contexts, including dual problems: a cost function that is the sum of a large number of components,

$$f(x) = \sum_{i=1}^{m} f_i(x), \tag{6.33}$$

where the functions $f_i : \Re^n \mapsto \Re$ are convex. Such functions can be minimized with special methods, called *incremental*, which exploit their additive structure (see Chapter 2).

An important special case is the cost function of the dual/separable problem (6.11); after a sign change to convert to minimization it takes the form (6.33). We provide a few more examples.

Example 6.1.5: (ℓ_1-Regularization)

Many problems in data analysis/machine learning involve an additive cost function, 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. Often a regularization function is added to the least squares objective, to induce desirable properties of the solution. Recently, nondifferentiable regularizarion functions have become increasingly important, as in the so called *ℓ_1-regularization problem*

$$\begin{aligned} &\text{minimize} \quad \sum_{j=1}^{m}(a_j'x-b_j)^2 + \gamma\sum_{i=1}^{n}|x_i| \\ &\text{subject to} \ \ (x_1,\ldots,x_n)\in\Re^n, \end{aligned}$$

(sometimes called the *lasso method*), which arises in statistical inference. Here a_j and b_j are given vectors and scalars, respectively, and γ is a positive scalar. The ℓ_1 regularization term affects the solution in a different way than a quadratic term (it tends to set a large number of components of x to 0; see the end-of-chapter references). There are several interesting variations of the ℓ_1-regularization approach, with many applications, for which we refer to the literature.

Example 6.1.6: (Maximum Likelihood Estimation)

We observe a sample of a random vector Z whose distribution $P_Z(\cdot\,; x)$ depends on an unknown parameter vector $x \in \Re^n$. For simplicity we assume that Z can take only a finite set of values, so that $P_Z(z; x)$ is the probability that Z takes the value z when the parameter vector has the value x. We wish to estimate x based on the given sample value z, by using the maximum likelihood method, i.e., by solving the problem

$$\begin{aligned} &\text{maximize} \quad P_Z(z; x) \\ &\text{subject to} \ \ x \in \Re^n. \end{aligned} \tag{6.34}$$

The cost function $P_Z(z; \cdot)$ of this problem may either have an additive structure or may be equivalent to a problem that has an additive structure. For example the event that $Z = z$ may be the union of a large number of disjoint events, so $P_Z(z; x)$ is the sum of the probabilities of these events. For another important context, suppose that the data z consists of m independent samples $y_1, \ldots, y_m$ drawn from a distribution $P_Y(\cdot\,; x)$, in which case

$$P_Z(z; x) = P_Y(y_1; x) \cdots P_Y(y_m; x).$$

Then the maximization (6.34) is equivalent to the additive cost minimization

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

where

$$f_i(x) = -\log P_Y(y_i; x).$$

In many applications the number of samples m is very large, in which case special methods that exploit the additive structure of the cost are recommended.

Example 6.1.7: (Minimization of an Expected Value - Stochastic Programming)

An important context where additive cost functions arise is the minimization of an expected value

$$\begin{aligned} &\text{minimize} \quad E\big\{F(x, w)\big\} \\ &\text{subject to} \ \ x \in X, \end{aligned} \tag{6.35}$$

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

For example, in *stochastic programming*, a classical model of two-stage optimization under uncertainty, a vector $x \in X$ is selected, a random event occurs that has m possible outcomes $w_1, \ldots, w_m$, 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 π_i is the corresponding probability. This is a problem with an additive cost function. Additive cost function problems also arise from problem (6.35) in a different way, when the expected value $E\big\{F(x,w)\big\}$ is approximated by an m-sample average

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} F(x, w_i),$$

where w_i are independent samples of the random variable w. The minimum of the sample average $f(x)$ is then taken as an approximation of the minimum of $E\big\{F(x,w)\big\}$.

Example 6.1.8: (Weber Problem in Location Theory)

A basic problem in location theory is to find a point x in the plane whose sum of weighted distances from a given set of points $y_1, \ldots, y_m$ is minimized. Mathematically, the problem is

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} w_i \|x - y_i\| \\ &\text{subject to} \quad x \in \Re^n, \end{aligned}$$

where $w_1, \ldots, w_m$ are given positive scalars. This problem descends from the famous Fermat-Torricelli-Viviani problem (see [BMS99] for an account of the history).

The structure of the additive cost function (6.33) often facilitates the use of a distributed computing system that is well-suited for the incremental approach. The following is an illustrative example.

Example 6.1.9: (Distributed Incremental Optimization – Sensor Networks)

Consider a network of m sensors where data are collected and are used to solve some inference problem involving a parameter vector x. If $f_i(x)$ represents an error penalty for the data collected by the ith sensor, the inference problem is of the form (6.33). While it is possible to collect all the data at a fusion center where the problem will be solved in centralized manner, it may be preferable to adopt a distributed approach in order to save in data communication overhead and/or take advantage of parallelism in computation. In such an approach the current iterate x_k is passed on from one sensor to another, with each sensor i performing an incremental iteration involving just its local component function f_i, and the entire cost function need not be known at any one location. We refer to Blatt, Hero, and Gauchman [BHG08], and Rabbat and Nowak [RaN04], [RaN05] for further discussion.

The approach of computing incrementally the values and subgradients of the components f_i in a distributed manner can be substantially extended to apply to general systems of asynchronous distributed computation, where the components are processed at the nodes of a computing network, and the results are suitably combined, as discussed by Nedić, Bertsekas, and Borkar [NBB01].

Let us finally note a generalization of the problem of this section, which arises when the functions f_i are convex and extended real-valued. This is essentially equivalent to constraining x to lie in the intersection of the domains of f_i, typically resulting in a problem of the form

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

where f_i are convex and real-valued and X_i are closed convex sets. Methods that are suitable for the unconstrained version of the problem where $X_i \equiv \Re^n$ can often be modified to apply to the constrained version, as we will see later.

6.1.4 Large Number of Constraints

Problems of the form

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad a_j'x \leq b_j, \quad j = 1, \ldots, r, \end{aligned} \tag{6.36}$$

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 (6.36) with

$$\begin{aligned} &\text{minimize} \quad f(x) + c\sum_{j=1}^r P(a_j'x - b_j) \\ &\text{subject to} \quad x \in \Re^n, \end{aligned} \tag{6.37}$$

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

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

An interesting alternative is to use

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

in which case it can be shown that the optimal solutions of problems (6.36) and (6.37) coincide when c is sufficiently large (see Section 6.1.5, as well as [Ber99], Section 5.4.5, [BNO03], Section 7.3). The cost function of the penalized problem (6.37) is of the additive form (6.33).

The idea of replacing constraints by penalties is more generally applicable. For example, the constraints in problem (6.36) could be nonlinear, or abstract of the form $x \in \cap_{j=1}^r X_j$. In the latter case the problem of minimizing a Lipschitz continuous function f over $\cap_{j=1}^r X_j$ may be replaced by unconstrained minimization of

$$f(x) + c \sum_{j=1}^r \text{dist}(x; X_j),$$

where $\text{dist}(x; X_j) = \inf_{y \in X_j} \|y - x\|$, and c is a penalty parameter that is larger than the Lipschitz constant of f (see Section 6.1.5).

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 Chapter 6.4.2. The ideas of outer and inner approximation can also be used to approximate nonpolyhedral convex constraint sets (in effect an infinite number of linear constraints) by polyhedral ones.

Network Optimization Problems

Problems with a large number of constraints also arise in problems involving a graph, and can often be handled with algorithms that take into account the graph structure. The following example is typical.

Example 6.1.10: (Optimal Routing in a Communication Network)

We are given a directed graph, which is viewed as a model of a data communication network. We are also given a set W of ordered node pairs $w = (i, j)$. The nodes i and j are referred to as the *origin* and the *destination* of w, respectively, and w is referred to as an OD pair. For each w, we are given a scalar r_w referred to as the *input traffic* of w. In the context of routing of data in a communication network, r_w (measured in data units/second) is the arrival rate of traffic entering and exiting the network at the origin and the destination of w, respectively. The routing objective is to divide each r_w among the many paths from origin to destination in a way that the resulting total arc flow pattern minimizes a suitable cost function. We denote:

P_w: A given set of paths that start at the origin and end at the destination of w. All arcs on each of these paths are oriented in the direction from the origin to the destination.

x_p: The portion of r_w assigned to path p, also called the *flow of path* p.

The collection of all path flows $\{x_p \mid p \in P_w,\, w \in W\}$ must satisfy the constraints

$$\sum_{p\in P_w} x_p = r_w, \qquad \forall\, w \in W, \tag{6.38}$$

$$x_p \geq 0, \qquad \forall\, p \in P_w,\ w \in W. \tag{6.39}$$

The total flow F_{ij} of arc (i, j) is the sum of all path flows traversing the arc:

$$F_{ij} = \sum_{\substack{\text{all paths } p\\ \text{containing } (i,j)}} x_p. \tag{6.40}$$

Consider a cost function of the form

$$\sum_{(i,j)} D_{ij}(F_{ij}). \tag{6.41}$$

The problem is to find a set of path flows $\{x_p\}$ that minimize this cost function subject to the constraints of Eqs. (6.38)-(6.40). We assume that D_{ij} is a convex and continuously differentiable function of F_{ij} with first derivative denoted by D'_{ij}. In data routing applications, the form of D_{ij} is often based on a queueing model of average delay (see [BeG92]).

The preceding problem is known as a *multicommodity network flow problem*. The terminology reflects the fact that the arc flows consist of several different commodities; in the present example, the different commodities are the data of the distinct OD pairs.

By expressing the total flows F_{ij} in terms of the path flows in the cost function (6.41) [using Eq. (6.40)], the problem can be formulated in terms of the path flow variables $\{x_p \mid p \in P_w,\, w \in W\}$ as

$$\begin{aligned} &\text{minimize} && D(x)\\ &\text{subject to} && \sum_{p\in P_w} x_p = r_w, \quad \forall\, w \in W,\\ & && x_p \geq 0, \quad \forall\, p \in P_w,\ w \in W, \end{aligned}$$

where

$$D(x) = \sum_{(i,j)} D_{ij} \left(\sum_{\substack{\text{all paths } p \\ \text{containing } (i,j)}} x_p \right)$$

and x is the vector of path flows x_p. There is a potentially huge number of variables as well as constraints in this problem. However, by judiciously taking into account the special structure of the problem, the constraint set can be simplified and the number of variables can be reduced to a manageable size, using algorithms that will be discussed later.

6.1.5 Exact Penalty Functions

In this section, we discuss a transformation that is often useful in the context of algorithmic solution of constrained convex optimization problems. In particular, we derive a form of equivalence between a constrained convex optimization problem, and a penalized problem that is less constrained or is entirely unconstrained. The motivation is that in some analytical contexts, it is useful to be able to work with an equivalent problem that is less constrained. Furthermore, some convex optimization algorithms do not have constrained counterparts, but can be applied to a penalized unconstrained problem.

We consider the problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in X, \qquad g(x) \le 0, \end{aligned} \tag{6.42}$$

where $g(x) = \big(g_1(x), \ldots, g_r(x)\big)$, X is a convex subset of $\Re^n$, and $f : \Re^n \to \Re$ and $g_j : \Re^n \to \Re$ are real-valued convex functions. We denote by f^* the primal optimal value, and by q^* the dual optimal value, i.e., $q^* = \sup_{\mu \ge 0} q(\mu)$, where

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

We assume that $-\infty < q^*$ and $f^* < \infty$.

We introduce a convex function $P : \Re^r \mapsto \Re$, called *penalty function*, which satisfies

$$P(u) = 0, \qquad \forall\ u \le 0, \tag{6.43}$$

$$P(u) > 0, \qquad \text{if } u_j > 0 \text{ for some } j = 1, \ldots, r. \tag{6.44}$$

We consider solving, in place of the original problem (6.42), the "penalized" problem

$$\begin{aligned} &\text{minimize} \quad f(x) + P\big(g(x)\big) \\ &\text{subject to} \quad x \in X, \end{aligned} \tag{6.45}$$

where the inequality constraints have been replaced by the extra cost $P\big(g(x)\big)$ for their violation.

Interesting examples of penalty functions are

$$P(u) = \frac{c}{2}\sum_{j=1}^{r}\big(\max\{0, u_j\}\big)^2,$$

and

$$P(u) = c\sum_{j=1}^{r}\max\{0, u_j\},$$

where c is a positive penalty parameter. A generic property is that P is monotone in the sense

$$u \le v \qquad \Rightarrow \qquad P(u) \le P(v). \tag{6.46}$$

To see this, we argue by contradiction. If there exist u and v with $u \le v$ and $P(u) > P(v)$, then by continuity of P, there must exist $\overline{u}$ close enough to u such that $\overline{u} < v$ and $P(\overline{u}) > P(v)$. Since P is convex, it is monotonically increasing along the halfline $\big\{\overline{u} + \alpha(\overline{u} - v) \mid \alpha \ge 0\big\}$, and since $P(\overline{u}) > P(v) \ge 0$, P takes positive values along this halfline. However, since $\overline{u} < v$, this halfline eventually enters the negative orthant, where P takes the value 0 by Eq. (6.43), a contradiction.

The convex conjugate function of P is given by

$$Q(\mu) = \sup_{u\in\Re^r}\big\{u'\mu - P(u)\big\},$$

and it can be seen that

$$Q(\mu) \ge 0, \qquad \forall\ \mu \in \Re^r,$$

$$Q(\mu) = \infty, \qquad \text{if } \mu_j < 0 \text{ for some } j = 1,\ldots,r.$$

Some interesting penalty functions P are shown in Fig. 6.1.3, together with their conjugates.

Consider the primal function of the original constrained problem,

$$p(u) = \inf_{x\in X,\, g(x)\le u} f(x), \qquad u \in \Re^r.$$

Since $-\infty < q^*$ and $f^* < \infty$ by assumption, we have $p(0) < \infty$ and $p(u) > -\infty$ for all $u \in \Re^r$ [since for any μ with $q(\mu) > -\infty$, we have $p(u) \ge q(\mu) - \mu'u > -\infty$ for all $u \in \Re^r$], so that p is proper (this will

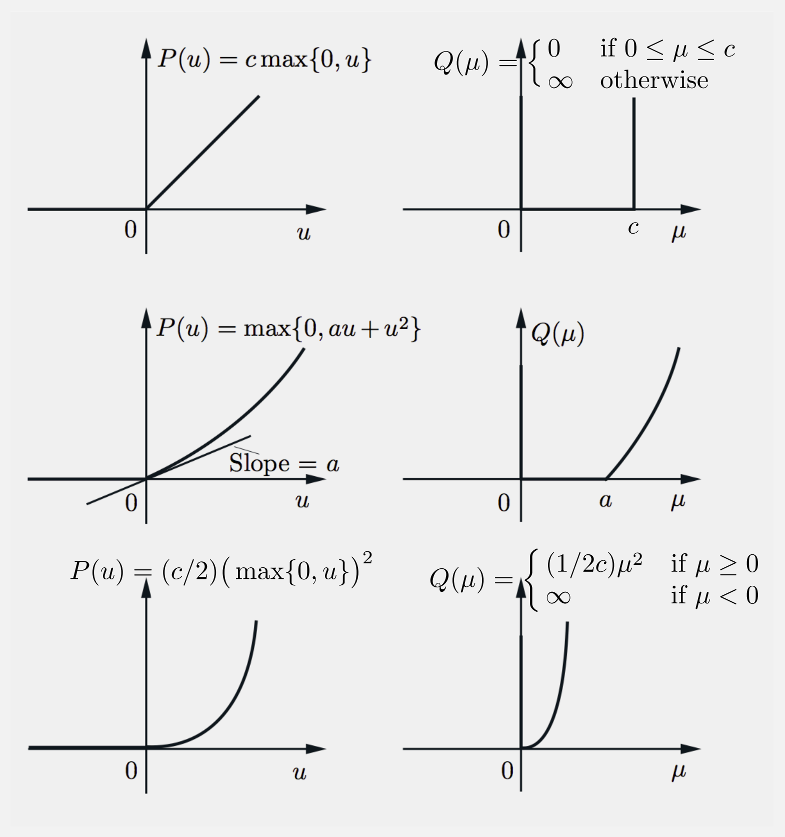

Figure 6.1.3. Illustration of conjugates of various penalty functions.

be needed for application of the Fenchel duality theorem). We have, using also the monotonicity relation (6.46),

$$\begin{aligned}
\inf_{x\in X}\big\{f(x) + P\big(g(x)\big)\big\} &= \inf_{x\in X}\ \inf_{u\in\Re^r,\, g(x)\le u}\big\{f(x) + P\big(g(x)\big)\big\} \\
&= \inf_{x\in X}\ \inf_{u\in\Re^r,\, g(x)\le u}\big\{f(x) + P(u)\big\} \\
&= \inf_{x\in X,\, u\in\Re^r,\, g(x)\le u}\big\{f(x) + P(u)\big\} \\
&= \inf_{u\in\Re^r}\ \inf_{x\in X,\, g(x)\le u}\big\{f(x) + P(u)\big\} \\
&= \inf_{u\in\Re^r}\big\{p(u) + P(u)\big\}.
\end{aligned}$$

We can now use the Fenchel duality theorem (Prop. 6.1.5) with the identifications $f_1 = p$, $f_2 = P$, and $A = I$. We use the conjugacy relation between the primal function p and the dual function q to write

$$\inf_{u\in\Re^r}\big\{p(u)+P(u)\big\}=\sup_{\mu\geq 0}\big\{q(\mu)-Q(\mu)\big\}, \tag{6.47}$$

so that

$$\inf_{x\in X}\big\{f(x)+P\big(g(x)\big)\big\}=\sup_{\mu\geq 0}\big\{q(\mu)-Q(\mu)\big\}; \tag{6.48}$$

see Fig. 6.1.4. Note that the conditions for application of the Fenchel Duality Theorem are satisfied since the penalty function P is real-valued, so that the relative interiors of $\mathrm{dom}(p)$ and $\mathrm{dom}(P)$ have nonempty intersection. Furthermore, as part of the conclusions of the Primal Fenchel duality theorem, it follows that the supremum over $\mu\geq 0$ in Eq. (6.48) is attained.

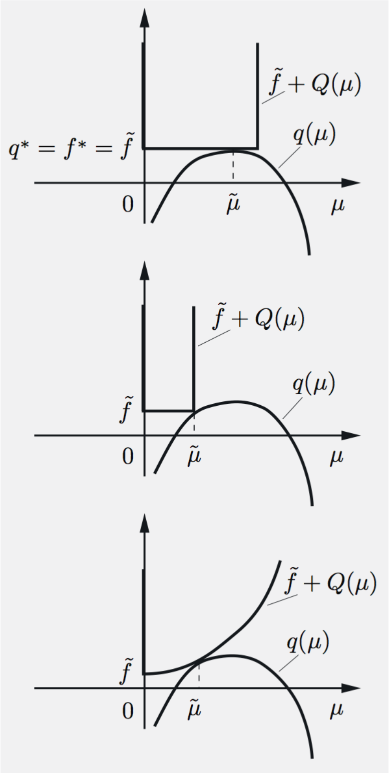

Figure 6.1.4. Illustration of the duality relation (6.48), and the optimal values of the penalized and the dual problem. Here f^* is the optimal value of the original problem, which is assumed to be equal to the optimal dual value q^*, while $\tilde{f}$ is the optimal value of the penalized problem,

$$\tilde{f}=\inf_{x\in X}\big\{f(x)+P\big(g(x)\big)\big\}.$$

The point of contact of the graphs of the functions $\tilde{f}+Q(\mu)$ and $q(\mu)$ corresponds to the vector $\tilde{\mu}$ that attains the maximum in the relation

$$\tilde{f}=\max_{\mu\geq 0}\big\{q(\mu)-Q(\mu)\big\}.$$

It can be seen from Fig. 6.1.4 that in order for the penalized problem (6.45) to have the same optimal value as the original constrained problem (6.42), the conjugate Q must be "sufficiently flat" so that it is minimized by some dual optimal solution μ^*, i.e., $0\in\partial Q(\mu^*)$ for some dual optimal solution μ^*, which by the Fenchel Inequality Theorem (Prop. 5.4.3), is equivalent to $\mu^* \in \partial P(0)$. This is part (a) of the following proposition. Parts (b) and (c) of the proposition deal with issues of equality of corresponding optimal solutions. The proposition assumes the convexity and other assumptions made in the early part in this section regarding problem (6.42) and the penalty function P.

Proposition 6.1.10:

(a) The penalized problem (6.45) and the original constrained problem (6.42) have equal optimal values if and only if there exists a dual optimal solution μ^* such that $\mu^* \in \partial P(0)$.

(b) In order for some optimal solution of the penalized problem (6.45) to be an optimal solution of the constrained problem (6.42), it is necessary that there exists a dual optimal solution μ^* such that

$$u'\mu^* \leq P(u), \qquad \forall\ u \in \Re^r. \tag{6.49}$$

(c) In order for the penalized problem (6.45) and the constrained problem (6.42) to have the same set of optimal solutions, it is sufficient that there exists a dual optimal solution μ^* such that

$$u'\mu^* < P(u), \qquad \forall\ u \in \Re^r \text{ with } u_j > 0 \text{ for some } j. \tag{6.50}$$

Proof: (a) We have using Eqs. (6.47) and (6.48),

$$p(0) \geq \inf_{u\in\Re^r} \big\{p(u) + P(u)\big\} = \sup_{\mu\geq 0}\big\{q(\mu) - Q(\mu)\big\} = \inf_{x\in X} \big\{f(x) + P\big(g(x)\big)\big\}.$$

Since $f^* = p(0)$, we have $f^* = \inf_{x\in X} \big\{f(x) + P\big(g(x)\big)\big\}$ if and only if equality holds in the above relation. This is true if and only if

$$0 \in \arg\min_{u\in\Re^r} \big\{p(u) + P(u)\big\},$$

which by Prop. 5.4.7, is true if and only if there exists some $\mu^* \in -\partial p(0)$ with $\mu^* \in \partial P(0)$. Since the set of dual optimal solutions is $-\partial p(0)$ (see Example 5.4.2), the result follows.

(b) If x^* is an optimal solution of both problems (6.42) and (6.45), then by feasibility of x^*, we have $P\big(g(x^*)\big) = 0$, so these two problems have equal optimal values. From part (a), there must exist a dual optimal solution $\mu^* \in \partial P(0)$, which is equivalent to Eq. (6.49), by the subgradient inequality.

(c) If x^* is an optimal solution of the constrained problem (6.42), then $P\big(g(x^*)\big) = 0$, so we have

$$f^* = f(x^*) = f(x^*) + P\big(g(x^*)\big) \geq \inf_{x\in X}\big\{f(x) + P\big(g(x)\big)\big\}.$$

The condition (6.50) implies the condition (6.49), so that by part (a), equality holds throughout in the above relation, showing that x^* is also an optimal solution of the penalized problem (6.45).

Conversely, if x^* is an optimal solution of the penalized problem (6.45), then x^* is either feasible [satisfies $g(x^*) \leq 0$], in which case it is an optimal solution of the constrained problem (6.42) [in view of $P\big(g(x)\big) = 0$ for all feasible vectors x], or it is infeasible in which case $g_j(x^*) > 0$ for some j. In the latter case, by using the given condition (6.50), it follows that there exists an $\epsilon > 0$ such that

$$\mu^{*\prime} g(x^*) + \epsilon < P\big(g(x^*)\big).$$

Let $\tilde{x}$ be a feasible vector such that $f(\tilde{x}) \leq f^* + \epsilon$. Since $P\big(g(\tilde{x})\big) = 0$ and $f^* = \min_{x \in X} \big\{ f(x) + \mu^{*\prime} g(x) \big\}$, we obtain

$$f(\tilde{x}) + P\big(g(\tilde{x})\big) = f(\tilde{x}) \leq f^* + \epsilon \leq f(x^*) + \mu^{*\prime} g(x^*) + \epsilon.$$

By combining the last two equations, we obtain

$$f(\tilde{x}) + P\big(g(\tilde{x})\big) < f(x^*) + P\big(g(x^*)\big),$$

which contradicts the hypothesis that x^* is an optimal solution of the penalized problem (6.45). This completes the proof. **Q.E.D.**

Note that in the case where the necessary condition (6.49) holds but the sufficient condition (6.50) does not, it is possible that the constrained problem (6.42) has optimal solutions that are not optimal solutions of the penalized problem (6.45), even though the two problems have the same optimal value.

To elaborate on Prop. 6.1.10, consider the penalty function

$$P(u) = c \sum_{j=1}^{r} \max\{0, u_j\},$$

where $c > 0$. The condition $\mu^* \in \partial P(0)$, or equivalently, $u'\mu^* \leq P(u)$ for all $u \in \Re^r$ [cf. Eq. (6.49)], is equivalent to

$$\mu_j^* \leq c, \qquad \forall\ j = 1, \ldots, r.$$

Similarly, the condition $u'\mu^* < P(u)$ for all $u \in \Re^r$ with $u_j > 0$ for some j [cf. Eq. (6.50)], is equivalent to

$$\mu_j^* < c, \qquad \forall\ j = 1, \ldots, r.$$

A General Exact Penalty Function

Let us finally discuss the case of a general Lipschitz continuous (not necessarily convex) cost function and an abstract constraint set $X \subset \Re^n$. The idea is to use a penalty that is proportional to the distance from X:

$$\mathrm{dist}(x; X) = \inf_{y \in X} \|x - y\|.$$

We have the following proposition.

Proposition 6.1.11: Let $f : Y \mapsto \Re$ be a function defined on a subset Y of $\Re^n$. Assume that f is Lipschitz continuous with constant L, i.e.,

$$\big|f(x) - f(y)\big| \le L\|x - y\|, \qquad \forall\, x, y \in Y.$$

Let also X be a nonempty closed subset of Y, and c be a scalar with $c > L$. Then x^* minimizes f over X if and only if x^* minimizes

$$F_c(x) = f(x) + c\,\mathrm{dist}(x; X)$$

over Y.

Proof: For a vector $x \in Y$, let $\hat{x}$ denote a vector of X that is at minimum distance from x. We have for all $x \in Y$,

$$F_c(x) = f(x) + c\|x - \hat{x}\| = f(\hat{x}) + \big(f(x) - f(\hat{x})\big) + c\|x - \hat{x}\| \ge f(\hat{x}) = F_c(\hat{x}),$$

with strict inequality if $x \neq \hat{x}$. Hence minima of F_c can only lie within X, while $F_c = f$ within X. This shows that x^* minimizes f over X if and only if x^* minimizes F_c over Y. **Q.E.D.**

The following proposition provides a generalization.

Proposition 6.1.12: Let $f : Y \mapsto \Re$ be a function defined on a subset Y of $\Re^n$, and let X_i, $i = 1, \ldots, m$, be closed subsets of Y with nonempty intersection. Assume that f is Lipschitz continuous over Y. Then there is a scalar $\overline{c} > 0$ such that for all $c \ge \overline{c}$, the set of minima of f over $\cap_{i=1}^m X_i$ coincides with the set of minima of

$$f(x) + c \sum_{i=1}^m \mathrm{dist}(x; X_i)$$

over Y.

Proof: Let L be the Lipschitz constant for f, and let $c_1, \ldots, c_m$ be scalars satisfying

$$c_k > L + c_1 + \cdots + c_{k-1}, \qquad \forall\, k = 1, \ldots, m,$$

where $c_0 = 0$. Define

$$F^k(x) = f(x) + c_1 \operatorname{dist}(x; X_1) + \cdots + c_k \operatorname{dist}(x; X_k), \qquad k = 1, \ldots, m,$$

and for $k = 0$, denote $F^0(x) = f(x)$, $c_0 = 0$. By applying Prop. 6.1.11, the set of minima of F^m over Y coincides with the set of minima of F^{m-1} over X_m, since c_m is greater than $L + c_1 + \cdots + c_{m-1}$, the Lipschitz constant for F^{m-1}. Similarly, for all $k = 1, \ldots, m$, the set of minima of F^k over $\cap_{i=k+1}^m X_i$ coincides with the set of minima of F^{k-1} over $\cap_{i=k}^m X_i$. Thus, for $k = 1$, we obtain that the set of minima of $f + c\sum_{i=1}^m \operatorname{dist}(\cdot; X_i) = F^m$ over Y coincides with the set of minima of $f = F^0$ over $\cap_{i=1}^m X_i$. **Q.E.D.**

Example 6.1.11: (Finding a Point in a Set Intersection)

As an example of the preceding analysis, consider a feasibility problem that arises in many contexts. It involves finding a point with certain properties within a set intersection $\cap_{i=1}^m X_i$, where each X_i is a closed convex set. Proposition 6.1.12 applies to this problem with $f(x) \equiv 0$, and can be used to convert the problem to one with an additive cost structure. In this special case of course, the penalty parameter c may be chosen to be any positive constant. We will revisit a more general version of this problem in Section 6.7.1.

6.2 ALGORITHMIC DESCENT - STEEPEST DESCENT

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

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

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

where ϕ 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 $x_0, \ldots, x_k$, 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 $k \to \infty$.

The methods to be discussed in this chapter revolve around these two ideas and their combinations, 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 and 6.10. We discuss the approximation approach in Sections 6.4-6.9.

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

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

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 α_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 the directional derivative at x_k is given by

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

it follows that

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

[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

$$\begin{aligned} &\text{minimize} \quad f'(x; d) \\ &\text{subject to} \quad \|d\| \le 1. \end{aligned} \tag{6.51}$$

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 $\partial f(x)$,

$$f'(x; d) = \max_{g \in \partial f(x)} d'g, \qquad \forall\, x, d \in \Re^n. \tag{6.52}$$

Next we note that the sets $\big\{d \mid \|d\| \le 1\big\}$ 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

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

and a saddle point exists. Furthermore, according to Prop. 3.4.1, for any saddle point (d^*, g^*), g^* maximizes the function $\min_{\|d\| \le 1} d'g = -\|g\|$ 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. (6.52)] subject to $\|d\| \le 1$ (so it is a direction of steepest descent), and minimizes $d'g^*$ subject to $\|d\| \le 1$, 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 $x \in \Re^n$, *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

$$x_{k+1} = x_k - \alpha_k g_k, \tag{6.53}$$

where g_k is the vector of minimum norm in $\partial f(x_k)$, and α_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 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 α_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 (see Section 6.10.1).

Gradient Projection

In the constrained case where X is a strict closed convex subset of $\Re^n$, the descent approach based on the iteration

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

becomes more complicated because it is not enough for d_k to be a descent direction at x_k. It must also be a *feasible direction* in the sense that $x_k + \alpha d_k$ must belong to X for small enough $\alpha > 0$. Generally, in the case where f is convex but nondifferentiable it is not easy to find feasible descent directions. However, in the case where f is differentiable there are several possibilities, including the *gradient projection method*, which has the form

$$x_{k+1} = P_X\big(x_k - \alpha \nabla f(x_k)\big), \tag{6.54}$$

where $\alpha > 0$ is a constant stepsize and $P_X(\cdot)$ denotes projection on X (see Fig. 6.2.1). Note that the projection is well defined since X is closed and convex (cf. Prop. 1.1.9).

Indeed, from the geometry of the projection theorem (cf. Fig. 6.2.1), we have

$$\nabla f(x_k)'(x_{k+1} - x_k) \leq 0,$$

and the inequality is strict unless $x_{k+1} = x_k$, in which case the optimality condition of Prop. 5.4.7 implies that x_k is optimal. Thus if x_k is not

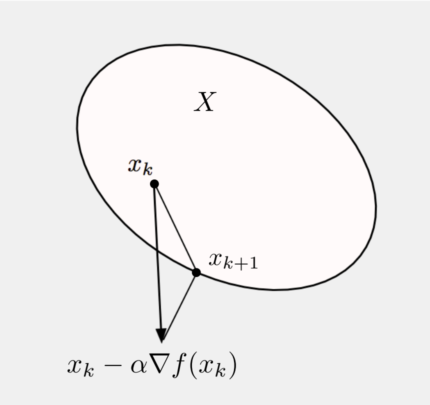

Figure 6.2.1. Illustration of the gradient projection iteration at x_k. We move from x_k along the along the direction $-\nabla f(x_k)$ and project $x_k - \alpha\nabla f(x_k)$ onto X to obtain x_{k+1}. We have

$$\nabla f(x_k)'(x_{k+1} - x_k) \leq 0,$$

and unless $x_{k+1} = x_k$, in which case x_k minimizes f over X, the angle between $\nabla f(x_k)$ and $(x_{k+1} - x_k)$ is strictly greater than 90 degrees, in which case

$$\nabla f(x_k)'(x_{k+1} - x_k) < 0.$$

optimal, $x_{k+1} - x_k$ defines a feasible descent direction at x_k. Based on this fact, we can show with some further analysis the descent property

$$f(x_{k+1}) < f(x_k)$$

when α is sufficiently small; see Section 6.10.1, where we will discuss the properties of the gradient projection method and some variations, and we will show that it has satisfactory convergence behavior under quite general conditions.

The difficulty in extending the cost improvement approach to nondifferentiable cost functions motivates alternative approaches. In one of the most popular algorithmic schemes, we abandon the idea of cost function descent, but aim to reduce the distance to the optimal solution set. This leads to the class of subgradient methods, discussed in the next section.

6.3 SUBGRADIENT METHODS

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

$$x_{k+1} = P_X(x_k - \alpha_k g_k), \tag{6.55}$$

where g_k is a subgradient of f at x_k, α_k is a positive stepsize, and $P_X(\cdot)$ denotes projection on the set X. Thus contrary to the steepest descent method (6.53), 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

$$f(x) = \sup_{z \in Z} \phi(x, z), \tag{6.56}$$

where $x \in \Re^n$, $z \in \Re^m$, $\phi : \Re^n \times \Re^m \mapsto (-\infty, \infty]$ is a function, and Z is a subset of $\Re^m$. 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 $x \in \mathrm{dom}(f)$, let us assume that $z_x \in Z$ attains the supremum in Eq. (6.56), 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 $y \in \Re^n$,

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

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

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

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 $\phi(x, \cdot)$ is found, any $g_x \in \partial\phi(x, z_x)$ is a subgradient of f at x. 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

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

and its dual

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

where $f : \Re^n \mapsto \Re$, $g : \Re^n \mapsto \Re^r$ are given (not necessarily convex) functions, X is a subset of $\Re^n$, and

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

is the dual function.

For a given $\mu \in \Re^r$, suppose that x_μ minimizes the Lagrangian over $x \in X$,

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

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

$$q(\nu) \le q(\mu) + (\nu - \mu)' g(x_\mu), \qquad \forall\, \nu \in \Re^r.$$

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

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

Note that this calculation is valid for all $\mu \in \Re^r$ for which there is a minimizing vector x_μ, and yields a subgradient of the function

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

regardless of whether $\mu \ge 0$.

An important characteristic of the subgradient method (6.55) is that the new iterate may not improve the cost for any value of the stepsize; i.e., for some k, we may have

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

(see Fig. 6.3.1). However, it turns out that 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). Part (b) of the following proposition provides a formal proof of the distance reduction property and an estimate for the range of appropriate stepsizes. Essential for this proof is the following nonexpansion property of the projection†

$$\|P_X(x) - P_X(y)\| \le \|x - y\|, \qquad \forall\, x, y \in \Re^n. \tag{6.57}$$

† To show the nonexpansion property, note that from the projection theorem (Prop. 1.1.9),

$$\big(z - P_X(x)\big)' \big(x - P_X(x)\big) \le 0, \qquad \forall\, z \in X.$$

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

$$\big(P_X(y) - P_X(x)\big)' \big(x - P_X(x)\big) \le 0.$$

Similarly,

$$\big(P_X(x) - P_X(y)\big)' \big(y - P_X(y)\big) \le 0.$$

By adding these two inequalities, we see that

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

Proposition 6.3.1: Let $\{x_k\}$ be the sequence generated by the subgradient method (6.55). Then, for all $y \in X$ and $k \geq 0$:

(a) We have

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

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

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

for all stepsizes α_k such that

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

Proof: (a) Using the nonexpansion property of the projection [cf. Eq. (6.57)], we obtain for all $y \in X$ and k,

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

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

$$\alpha_k = \frac{f(x_k) - f^*}{\|g_k\|^2}, \tag{6.58}$$

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

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

By rearranging this relation and by using the Schwarz inequality, we have

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

from which the nonexpansion property of the projection follows.

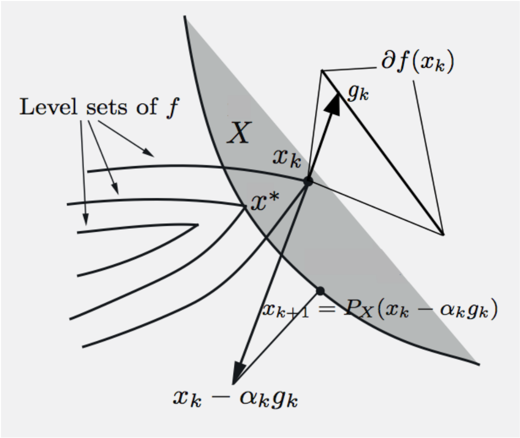

Figure 6.3.1. Illustration of how the iterate $P_X(x_k - \alpha g_k)$ may not improve the cost function with a particular choice of subgradient g_k, regardless of the value of the stepsize α.

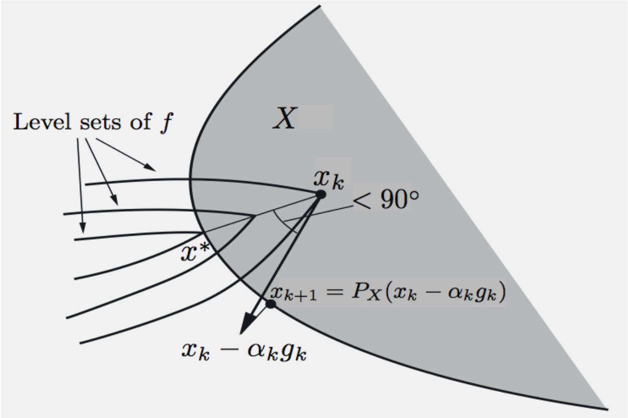

Figure 6.3.2. Illustration of how, given a nonoptimal x_k, the distance to any optimal solution x^* is reduced using a subgradient iteration with a sufficiently small stepsize. The crucial fact, which follows from the definition of a subgradient, is that the angle between the subgradient g_k and the vector $x^* - x_k$ is greater than 90 degrees. As a result, if α_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^*.

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 (6.58) requires that we know f^*, which is rare. In practice, one must use some simpler scheme for selecting a stepsize. The simplest possibility is to select α_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

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

and implies that the distance to x^* decreases if

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

or equivalently, if x_k is outside the level set

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

(see Fig. 6.3.3). Thus, if α 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 α_k, which are progressively reduced up to a small positive value α, using some heuristic scheme. Other possibilities for stepsize choice include a diminishing stepsize, whereby $\alpha_k \to 0$, and schemes that replace the unknown optimal value f^* in Eq. (6.58) with an estimate.

6.3.1 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

$$f^* = \inf_{x \in X} f(x), \qquad X^* = \bigl\{x \in X \mid f(x) = f^*\bigr\}, \qquad \overline{f} = \liminf_{k \to \infty} f(x_k),$$

and we assume the following:

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

$$c \geq \sup\bigl\{\|g\| \mid g \in \partial f(x_k)\bigr\}, \qquad \forall\, k \geq 0.$$

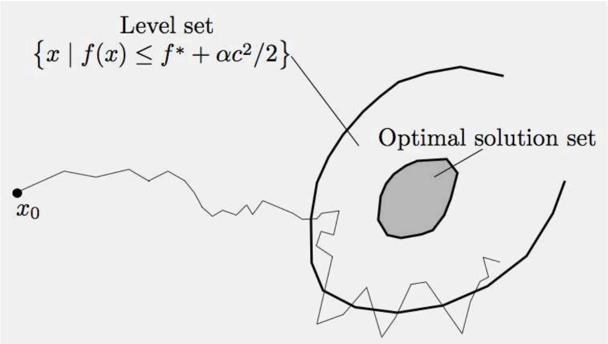

Figure 6.3.3. Illustration of a principal convergence property of the subgradient method with a constant stepsize α, and assuming a bound c on the subgradient norms $\|g_k\|$. When the current iterate is outside the level set

$$\left\{x \;\middle|\; f(x) \le f^* + \frac{\alpha c^2}{2}\right\},$$

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.

We note that Assumption 6.3.1 is satisfied if f is polyhedral, an important special case in practice, or if X is compact, [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 α_k is fixed at some positive scalar α.

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

(b) If $f^* > -\infty$, then

$$\overline{f} \le f^* + \frac{\alpha c^2}{2}.$$

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

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

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

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

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

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

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

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

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 α_k is fixed at some positive scalar α, and that X^* is nonempty. Then for any positive scalar ϵ, we have

$$\min_{0 \leq k \leq K} f(x_k) \leq f^* + \frac{\alpha c^2 + \epsilon}{2}, \tag{6.59}$$

where

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

Proof: Assume, to arrive at a contradiction, that Eq. (6.59) 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 $0 \leq k \leq K$,

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

Summation of the above inequalities over k for $k = 0, \ldots, K$, 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 ϵ-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 α_k diminishes to zero, but satisfies $\sum_{k=0}^{\infty} \alpha_k = \infty$ [for example, $\alpha_k = \beta/(k+\gamma)$, where β and γ 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 α_k satisfies

$$\lim_{k \to \infty} \alpha_k = 0, \qquad \sum_{k=0}^{\infty} \alpha_k = \infty,$$

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

Proof: Assume, to arrive at a contradiction, that there exists an $\epsilon > 0$ such that

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

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

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

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

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

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 $k \geq k_0$,

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

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

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

Therefore for all $k \geq k_0$ we have

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

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

We now discuss the stepsize rule

$$\alpha_k = \gamma_k \frac{f(x_k) - f^*}{\|g_k\|^2}, \qquad 0 < \underline{\gamma} \leq \gamma_k \leq \overline{\gamma} < 2, \qquad \forall\ k \geq 0, \tag{6.60}$$

where $\underline{\gamma}$ and $\overline{\gamma}$ 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 α_k is determined by the dynamic stepsize rule (6.60), $\{x_k\}$ converges to some optimal solution.

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

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

By using the definition of α_k [cf. Eq. (6.60)] and the fact $\|g_k\| \leq c$ (cf. Assumption 6.3.1), we obtain

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

This implies that $\{x_k\}$ is bounded. Furthermore, $f(x_k) \to f^*$, 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 $\overline{x}$ of $\{x_k\}$, we have $\overline{x} \in X^*$, and since the sequence $\{\|x_k - x^*\|\}$ is decreasing, it converges to $\|\overline{x} - x^*\|$ for every $x^* \in X^*$. If there are two distinct limit points $\tilde{x}$ and $\overline{x}$ of $\{x_k\}$, we must have $\tilde{x} \in X^*$, $\overline{x} \in X^*$, and $\|\tilde{x} - x^*\| = \|\overline{x} - x^*\|$ for all $x^* \in X^*$, which is possible only if $\tilde{x} = \overline{x}$. **Q.E.D.**

In most practical problems the optimal value f^* is not known. In this case we may modify the dynamic stepsize (6.60) by replacing f^* with an estimate. This leads to the stepsize rule

$$\alpha_k = \gamma_k \frac{f(x_k) - f_k}{\|g_k\|^2}, \qquad 0 < \underline{\gamma} \leq \gamma_k \leq \overline{\gamma} < 2, \qquad \forall\, k \geq 0, \tag{6.61}$$

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

$$f_k = \min_{0 \leq j \leq k} f(x_j) - \delta_k, \tag{6.62}$$

and δ_k is updated according to

$$\delta_{k+1} = \begin{cases} \rho\delta_k & \text{if } f(x_{k+1}) \leq f_k, \\ \max\big\{\beta\delta_k, \delta\big\} & \text{if } f(x_{k+1}) > f_k, \end{cases} \tag{6.63}$$

where δ, β, and ρ 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 δ_k over the best value achieved thus far [cf. Eq. (6.62)]. Whenever the target level is achieved, we increase δ_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, δ_k is reduced up to a threshold δ. This threshold guarantees that the stepsize α_k of Eq. (6.61) is bounded away from zero, since from Eq. (6.62), 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 α_k is determined by the dynamic stepsize rule (6.61) with the adjustment procedure (6.62)–(6.63). 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

$$\inf_{k\geq 0} f(x_k) > f^* + \delta. \tag{6.64}$$

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 δ [cf. Eqs. (6.62) and (6.63)], so in view of Eq. (6.64), the target value can be attained only a finite number of times. From Eq. (6.63) it follows that after finitely many iterations, δ_k is decreased to the threshold value and remains at that value for all subsequent iterations, i.e., there is an index $\overline{k}$ such that

$$\delta_k = \delta, \qquad \forall\ k \geq \overline{k}. \tag{6.65}$$

In view of Eq. (6.64), there exists $\overline{y} \in X$ such that $\inf_{k\geq 0} f(x_k) - \delta \geq f(\overline{y})$. From Eqs. (6.62) and (6.65), we have

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

so that

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

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

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

By combining the preceding two relations and the definition of α_k [cf. Eq. (6.61)], we obtain

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

where the last inequality follows from the facts $\gamma_k \in [\underline{\gamma}, \overline{\gamma}]$ 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

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

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

We will now consider various types of subgradient methods, which use approximate subgradients. As we will see, 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.

6.3.2 ϵ-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.

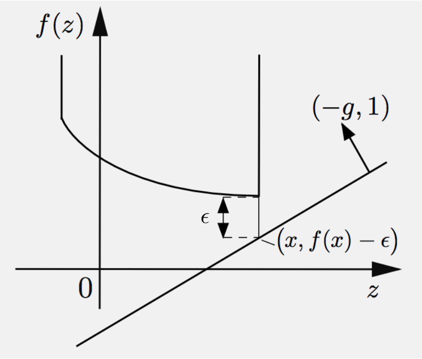

Figure 6.3.4. Illustration of an ϵ-subgradient of a convex function f. A vector g is an ϵ-subgradient at $x \in \text{dom}(f)$ if and only if there is a hyperplane with normal $(-g, 1)$, which passes through the point $\big(x, f(x) - \epsilon\big)$, and separates this point from the epigraph of f.

Given a proper convex function $f : \Re^n \mapsto (-\infty, \infty]$ and a scalar $\epsilon > 0$, we say that a vector g is an ϵ-*subgradient* of f at a point $x \in \text{dom}(f)$ if

$$f(z) \geq f(x) + (z - x)'g - \epsilon, \qquad \forall\ z \in \Re^n. \tag{6.66}$$

The ϵ-*subdifferential* $\partial_\epsilon f(x)$ is the set of all ϵ-subgradients of f at x, and by convention, $\partial_\epsilon f(x) = \emptyset$ for $x \notin \text{dom}(f)$. It can be seen that

$$\partial_{\epsilon_1} f(x) \subset \partial_{\epsilon_2} f(x) \qquad \text{if } 0 < \epsilon_1 < \epsilon_2,$$

and that

$$\cap_{\epsilon \downarrow 0} \partial_\epsilon f(x) = \partial f(x).$$

To interpret geometrically an ϵ-subgradient, note that the defining relation (6.66) can be written as

$$f(z) - z'g \geq \big(f(x) - \epsilon\big) - x'g, \qquad \forall\ z \in \Re^n.$$

Thus g is an ϵ-subgradient at x if and only if the epigraph of f is contained in the positive halfspace corresponding to the hyperplane in $\Re^{n+1}$ that has normal $(-g, 1)$ and passes through $\big(x, f(x) - \epsilon\big)$, as illustrated in Fig. 6.3.4.

Figure 6.3.5 illustrates the definition of the ϵ-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 dom(f). This follows by the Nonvertical Hyperplane Theorem (Prop. 1.5.8).

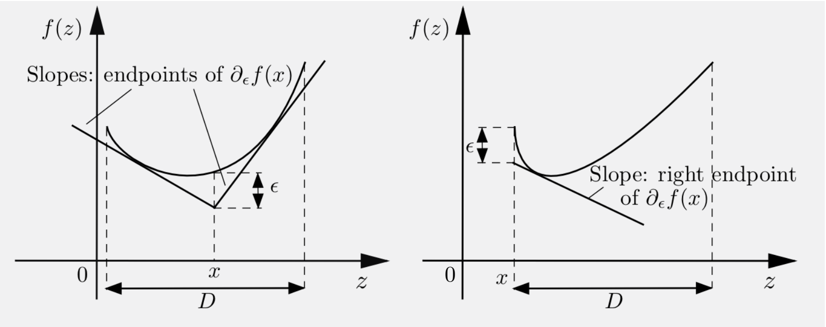
$f(z)$
Slopes: endpoints of $\partial_\epsilon f(x)$
ϵ
0
x
z
D
$f(z)$
ϵ
Slope: right endpoint of $\partial_\epsilon f(x)$
0
x
D
z

Figure 6.3.5. Illustration of the ϵ-subdifferential $\partial_\epsilon f(x)$ of a one-dimensional function $f : \Re \mapsto (-\infty, \infty]$, which is closed and convex, and has as effective domain an interval D. The ϵ-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 ∞.

The following example motivates the use of ϵ-subgradients in the context of duality and minimax problems. It shows that ϵ-subgradients may be computed more economically than subgradients, through an approximate minimization.

Example 6.3.3: (ϵ-Subgradient Calculation in Minimax and Dual Problems)

As in Example 6.3.1, let us consider the minimization of

$$f(x) = \sup_{z \in Z} \phi(x, z), \tag{6.67}$$

where $x \in \Re^n$, $z \in \Re^m$, Z is a subset of $\Re^m$, and $\phi : \Re^n \times \Re^m \mapsto (-\infty, \infty]$ 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. (6.67), 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 ϵ, we can then obtain an ϵ-subgradient at x, which we can use in turn within an ϵ-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. (6.67), 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 $y \in \Re^n$, 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 ϵ-subgradient of f at x, so

$$\phi(x, z_x) \geq \sup_{z \in Z} \phi(x, z) - \epsilon \text{ and } g_x \in \partial\phi(x, z_x) \qquad \Rightarrow \qquad g_x \in \partial_\epsilon f(x).$$

We now consider the class of *ϵ-subgradient methods* for minimizing a real-valued convex function $f : \Re^n \mapsto \Re$ over a closed convex set X, given by

$$x_{k+1} = P_X(x_k - \alpha_k g_k), \tag{6.68}$$

where g_k is an ϵ_k-subgradient of f at x_k, α_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 *ϵ-subgradient methods generally aim to converge to the ϵ-optimal set*, where $\epsilon = \lim_{k\to\infty} \epsilon_k$, rather than the optimal set, as subgradient methods do.

To get a sense of the convergence proof, 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 ϵ-subgradient method, we can show that for all $y \in X$ and $k \geq 0$

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

Using this inequality, one can essentially replicate the convergence analysis of Section 6.3.1. As an example, consider the case of constant α_k and ϵ_k: $\alpha_k \equiv \alpha$ for some $\alpha > 0$ and $\epsilon_k \equiv \epsilon$ for some $\epsilon > 0$. Then, if the ϵ-subgradients g_k are bounded, with $\|g_k\| \leq c$ for some constant c and all k, we obtain for all optimal solutions x^*,

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

where $f^* = \inf_{x \in X} f(x)$ is the optimal value. This implies that the distance to x^* decreases if

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

or equivalently, if x_k is outside the level set

$$\left\{ x \;\Big|\; f(x) \le f^* + \epsilon + \frac{\alpha c^2}{2} \right\}$$

(cf. Fig. 6.3.3). With analysis similar to the one for the subgradient case, we can also show that if $\alpha_k \to 0$, $\sum_{k=0}^{\infty} \alpha_k = \infty$, and $\epsilon_k \to \epsilon \ge 0$, we have

$$\liminf_{k \to \infty} f(x_k) \le f^* + \epsilon$$

(cf. Prop. 6.3.4).

6.3.3 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

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

where the functions $f_i : \Re^n \mapsto \Re$ are convex. We mentioned several contexts where cost functions of this type arise in Section 6.1.3. 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

$$x_{k+1} = \psi_{m,k}, \qquad (6.69)$$

where starting with $\psi_{0,k} = x_k$, we obtain $\psi_{m,k}$ after the m steps

$$\psi_{i,k} = P_X(\psi_{i-1,k} - \alpha_k g_{i,k}), \qquad i = 1, \ldots, m, \qquad (6.70)$$

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 ϵ-subgradient method (6.68). An important fact here is that *if two vectors x and $\overline{x}$ are "near" each other, then subgradients at $\overline{x}$ can be viewed as ϵ-subgradients at x*, with ϵ "small." In particular, if $g \in \partial f(\overline{x})$, we have for all $z \in \Re^n$,

$$\begin{aligned} f(z) &\geq f(\overline{x}) + g'(z - \overline{x}) \\ &\geq f(x) + g'(z - x) + f(\overline{x}) - f(x) + g'(x - \overline{x}) \\ &\geq f(x) + g'(z - x) - \epsilon, \end{aligned}$$

where

$$\epsilon = \max\big\{0, f(x) - f(\overline{x})\big\} + \|g\| \cdot \|\overline{x} - x\|.$$

Thus, we have $g \in \partial_\epsilon f(x)$, and ϵ is small when $\overline{x}$ is near x.

We now observe from Eq. (6.70) that the ith 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 α_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 ϵ_i-subgradient of f_i at x_k, where ϵ_i is small. In particular, if we ignore the projection operation in Eq. (6.70), 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 ϵ_i-subgradient of f_i at x_k, where ϵ_i is "small" (proportional to α_k). Let us also use the formula

$$\partial_{\epsilon_1} f_1(x) + \cdots + \partial_{\epsilon_m} f_m(x) \subset \partial_\epsilon f(x),$$

where $\epsilon = \epsilon_1 + \cdots + \epsilon_m$, to approximate the ϵ-subdifferential of the sum $f = \sum_{i=1}^m f_i$. (This relation follows from the definition of ϵ-subgradient.) Then, it can be seen that the incremental subgradient iteration can be viewed as an ϵ-subgradient iteration with $\epsilon = \epsilon_1 + \cdots + \epsilon_m$. The size of ϵ depends on the size of α_k, as well as the function f, and we generally have $\epsilon \to 0$ as $\alpha_k \to 0$. As a result, for the case where $\alpha_k \to 0$ 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 α_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

$$f^* = \inf_{x \in X} f(x), \qquad X^* = \big\{ x \in X \mid f(x) = f^* \big\},$$

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

where $\|\cdot\|$ denotes the standard Euclidean norm. In our analysis, we assume the following:

Assumption 6.3.2: (Subgradient Boundedness) We have

$$c_i \geq \sup_{k \geq 0} \big\{ \|g\| \mid g \in \partial f_i(x_k) \cup \partial f_i(\psi_{i-1,k}) \big\}, \qquad i = 1, \ldots, m,$$

for some scalars $c_1, \ldots, c_m$.

We note that Assumption 6.3.2 is satisfied if each f_i is real-valued and polyhedral. In particular, Assumption 6.3.2 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 $\Re^n$, 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 6.3.2 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 result, which parallels Prop. 6.3.1(a) for the (nonincremental) subgradient method.

Proposition 6.3.7: Let $\{x_k\}$ be the sequence generated by the incremental method (6.69), (6.70). Then for all $y \in X$ and $k \geq 0$, we have

$$\|x_{k+1} - y\|^2 \leq \|x_k - y\|^2 - 2\alpha_k \big(f(x_k) - f(y) \big) + \alpha_k^2 c^2, \tag{6.71}$$

where $c = \sum_{i=1}^m c_i$ and c_i are the scalars of Assumption 6.3.2.

Proof: Using the nonexpansion property of the projection, the subgradient boundedness (cf. Assumption 6.3.2), and the subgradient inequality for each component function f_i, we obtain for all $y \in X$,

$$\begin{aligned}\|\psi_{i,k} - y\|^2 &= \big\|P_X\left(\psi_{i-1,k} - \alpha_k g_{i,k}\right) - y\big\|^2 \\ &\le \|\psi_{i-1,k} - \alpha_k g_{i,k} - y\|^2 \\ &\le \|\psi_{i-1,k} - y\|^2 - 2\alpha_k g_{i,k}'(\psi_{i-1,k} - y) + \alpha_k^2 c_i^2 \\ &\le \|\psi_{i-1,k} - y\|^2 - 2\alpha_k\big(f_i(\psi_{i-1,k}) - f_i(y)\big) + \alpha_k^2 c_i^2, \qquad \forall\ i,k.\end{aligned}$$

By adding the above inequalities over $i = 1,\ldots,m$, we have for all $y \in X$ and k,

$$\begin{aligned}\|x_{k+1} - y\|^2 &\le \|x_k - y\|^2 - 2\alpha_k\sum_{i=1}^m\big(f_i(\psi_{i-1,k}) - f_i(y)\big) + \alpha_k^2\sum_{i=1}^m c_i^2 \\ &= \|x_k - y\|^2 - 2\alpha_k\left(f(x_k) - f(y) + \sum_{i=1}^m\big(f_i(\psi_{i-1,k}) - f_i(x_k)\big)\right) \\ &\quad + \alpha_k^2\sum_{i=1}^m c_i^2.\end{aligned}$$

By strengthening the above inequality, we have for all $y \in X$ and k, using also the fact $\psi_{0,k} = x_k$,

$$\begin{aligned}\|x_{k+1} - y\|^2 &\le \|x_k - y\|^2 - 2\alpha_k\big(f(x_k) - f(y)\big) \\ &\quad + 2\alpha_k\sum_{i=1}^m c_i||\psi_{i-1,k} - x_k|| + \alpha_k^2\sum_{i=1}^m c_i^2 \\ &\le \|x_k - y\|^2 - 2\alpha_k\big(f(x_k) - f(y)\big) \\ &\quad + \alpha_k^2\left(2\sum_{i=2}^m c_i\left(\sum_{j=1}^{i-1} c_j\right) + \sum_{i=1}^m c_i^2\right) \\ &= \|x_k - y\|^2 - 2\alpha_k\big(f(x_k) - f(y)\big) + \alpha_k^2\left(\sum_{i=1}^m c_i\right)^2 \\ &= \|x_k - y\|^2 - 2\alpha_k\big(f(x_k) - f(y)\big) + \alpha_k^2 c^2,\end{aligned}$$

where in the first inequality we use the relation

$$f_i(x_k) - f_i(\psi_{i-1,k}) \le \|\tilde g_{i,k}\| \cdot \|\psi_{i-1,k} - x_k\| \le c_i\|\psi_{i-1,k} - x_k\|$$

with $\tilde g_{i,k} \in \partial f_i(x_k)$, and in the second inequality we use the relation

$$\|\psi_{i,k} - x_k\| \le \alpha_k\sum_{j=1}^i c_j, \qquad i = 1,\ldots,m, \quad k \ge 0,$$

which follows from Eqs. (6.69), (6.70), and Assumption 6.3.2. **Q.E.D.**

Among other things, Prop. 6.3.7 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 α_k is sufficiently small [less than $2\big(f(x_k) - f(y)\big)/c^2$]. 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 6.3.8: Let $\{x_k\}$ be the sequence generated by the incremental method (6.69), (6.70), with the stepsize α_k fixed at some positive constant α.

(a) If $f^* = -\infty$, then

$$\liminf_{k\to\infty} f(x_k) = f^*.$$

(b) If $f^* > -\infty$, then

$$\liminf_{k\to\infty} f(x_k) \leq f^* + \frac{\alpha c^2}{2},$$

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

$$\liminf_{k\to\infty} f(x_k) - \frac{\alpha c^2}{2} - 2\epsilon > f^*.$$

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

$$\liminf_{k\to\infty} f(x_k) - \frac{\alpha c^2}{2} - 2\epsilon \geq f(\hat{y}),$$

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

$$f(x_k) \geq \liminf_{k\to\infty} f(x_k) - \epsilon.$$

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 Prop. 6.3.7 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 $k \geq k_0$,

$$\|x_{k+1} - \hat{y}\|^2 \leq \|x_{k-1} - \hat{y}\|^2 - 4\alpha\epsilon \leq \cdots \leq \|x_{k_0} - \hat{y}\|^2 - 2(k+1-k_0)\alpha\epsilon,$$

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

$$\liminf_{k\to\infty} f(\psi_{i,k}) \leq f^* + \frac{\alpha c^2}{2}, \qquad \forall\, i = 1, \ldots, m. \tag{6.72}$$

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 6.3.9: Assume that X^* is nonempty. Let $\{x_k\}$ be the sequence generated by the incremental method (6.69), (6.70), with the stepsize α_k fixed at some positive constant α. Then for any positive scalar ϵ we have

$$\min_{0\leq k\leq K} f(x_k) \leq f^* + \frac{\alpha c^2 + \epsilon}{2}, \tag{6.73}$$

where K is given by

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

Proof: Assume, to arrive at a contradiction, that Eq. (6.73) 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. 6.3.7 with α_k replaced by α, we obtain for all k with $0 \leq k \leq K$,

$$\begin{aligned}\big(d(x_{k+1})\big)^2 &\leq \Big(d(x_k)\Big)^2 - 2\alpha\big(f(x_k) - f^*\big) + \alpha^2 c^2 \\ &\leq \big(d(x_k)\big)^2 - (\alpha^2 c^2 + \alpha\epsilon) + \alpha^2 c^2 \\ &= \big(d(x_k)\big)^2 - \alpha\epsilon.\end{aligned}$$

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

$$\big(d(x_{K+1})\big)^2 \leq d(x_0)^2 - (K+1)\alpha\epsilon,$$

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 (6.73) 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. (6.73) to hold is given by

$$N = mK = m \left\lfloor \frac{\big(d(x_0)\big)^2}{\alpha\epsilon} \right\rfloor.$$

The Role of the Order of Processing the Components

The error tolerance estimate $\alpha c^2/2$ of Prop. 6.3.8 and Eq. (6.72) 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 α, we can find a starting point for which the sequence $\{\psi_{i,k}\}$ generated by the method satisfies Eq. (6.72). 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

$$\liminf_{k\to\infty} f(\psi_{i,k}) = f^* + \frac{\beta\alpha c^2}{2}, \qquad \forall\ i = 1, \ldots, m, \tag{6.74}$$

where β 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

$$\liminf_{k\to\infty} f(\psi_{i,k}) = f^* + \frac{\alpha c^2}{2m}, \qquad \forall\ i = 1, \ldots, m. \tag{6.75}$$

Thus from Eqs. (6.74) and (6.75), we see that for a given stepsize α, the achievable range for the bound on the difference

$$\liminf_{k\to\infty} f(\psi_{i,k}) - f^*$$

corresponding to the incremental subgradient method with a fixed processing order is

$$\left[\frac{\alpha c^2}{2m}, \frac{\alpha c^2}{2} \right]. \tag{6.76}$$

By this we mean that there exists a choice of problem for which we can do no better that 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 (6.76), it can be seen that for a given stepsize α, 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.

6.3.4 Randomized Incremental Subgradient Methods

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

$$x_{k+1} = P_X\big(x_k - \alpha g(\omega_k, x_k)\big), \tag{6.77}$$

where ω_k is a random variable taking equiprobable values from the set $\{1, \ldots, m\}$, and $g(\omega_k, x_k)$ is a subgradient of the component f_{ω_k} at x_k. This simply means that if the random variable ω_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 6.3.3: For the randomized method (6.77):

(a) $\{\omega_k\}$ is a sequence of independent random variables, each uniformly distributed over the set $\{1, \ldots, m\}$. Furthermore, the sequence $\{\omega_k\}$ is independent of the sequence $\{x_k\}$.

(b) The set of subgradients $\big\{g(\omega_k, x_k) \mid k = 0, 1, \ldots\big\}$ is bounded, i.e., there is a positive constant c_0 such that with probability 1

$$\|g(\omega_k, x_k)\| \le c_0, \qquad \forall\ k \ge 0.$$

Note that if the set X is compact or the components f_i are polyhedral, then Assumption 6.3.3(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 6.3.10: (Supermartingale Convergence Theorem) Let Y_k, Z_k, and W_k, $k = 0, 1, 2, \ldots$, be three sequences of random variables and let $\mathcal{F}_k$, $k = 0, 1, 2, \ldots$, 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

$$E\big\{Y_{k+1} \mid \mathcal{F}_k\big\} \le Y_k - Z_k + W_k.$$

(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. 6.3.8 for the deterministic incremental method.

Proposition 6.3.11: Let $\{x_k\}$ be the sequence generated by the randomized incremental method (6.77).

(a) If $f^* = -\infty$, then with probability 1

$$\inf_{k\ge0} f(x_k) = f^*.$$

(b) If $f^* > -\infty$, then with probability 1

$$\inf_{k\ge0} f(x_k) \le f^* + \frac{\alpha m c_0^2}{2}.$$

Proof: By adapting Prop. 6.3.7 to the case where f is replaced by f_{ω_k}, we have

$$\|x_{k+1}-y\|^2 \le \|x_k-y\|^2 - 2\alpha\big(f_{\omega_k}(x_k) - f_{\omega_k}(y)\big) + \alpha^2 c_0^2, \quad \forall\, y \in X, \quad k \ge 0.$$

By taking the conditional expectation with respect to $\mathcal{F}_k = \{x_0, \ldots, x_k\}$, the method's history up to x_k, we obtain for all $y \in X$ and k,

$$\begin{aligned} E\big\{\|x_{k+1}-y\|^2 \mid \mathcal{F}_k\big\} &\le \|x_k-y\|^2 - 2\alpha E\big\{f_{\omega_k}(x_k) - f_{\omega_k}(y) \mid \mathcal{F}_k\big\} + \alpha^2 c_0^2 \\ &= \|x_k-y\|^2 - 2\alpha \sum_{i=1}^m \frac{1}{m}\big(f_i(x_k) - f_i(y)\big) + \alpha^2 c_0^2 \\ &= \|x_k-y\|^2 - \frac{2\alpha}{m}\big(f(x_k) - f(y)\big) + \alpha^2 c_0^2, \end{aligned} \tag{6.78}$$

where the first equality follows since ω_k takes the values $1, \ldots, m$ with equal probability $1/m$.

Now, fix a positive scalar γ, consider the level set L_γ 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

$$\hat{x}_{k+1} = \begin{cases} P_X\big(\hat{x}_k - \alpha g(\omega_k, \hat{x}_k)\big) & \text{if } \hat{x}_k \notin L_\gamma, \\ y_\gamma & \text{otherwise}, \end{cases}$$

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_γ, 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_γ.

Using Eq. (6.78) with $y = y_\gamma$, we have

$$E\big\{\|\hat{x}_{k+1} - y_\gamma\|^2 \mid \mathcal{F}_k\big\} \le \|\hat{x}_k - y_\gamma\|^2 - \frac{2\alpha}{m}\big(f(\hat{x}_k) - f(y_\gamma)\big) + \alpha^2 c_0^2,$$

or equivalently

$$E\big\{\|\hat{x}_{k+1} - y_\gamma\|^2 \mid \mathcal{F}_k\big\} \le \|\hat{x}_k - y_\gamma\|^2 - z_k, \tag{6.79}$$

where

$$z_k = \begin{cases} \frac{2\alpha}{m}\big(f(\hat{x}_k) - f(y_\gamma)\big) - \alpha^2 c_0^2 & \text{if } \hat{x}_k \notin L_\gamma, \\ 0 & \text{if } \hat{x}_k = y_\gamma. \end{cases}$$

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

$$\begin{aligned} z_k &= \frac{2\alpha}{m}\big(f(\hat{x}_k) - f(y_\gamma)\big) - \alpha^2 c_0^2 \\ &\ge \frac{2\alpha}{m}\left(-\gamma + 1 + \frac{\alpha m c_0^2}{2} + \gamma\right) - \alpha^2 c_0^2 \\ &= \frac{2\alpha}{m}. \end{aligned}$$

Since $z_k = 0$ for $\hat{x}_k \in L_\gamma$, we have $z_k \ge 0$ for all k, and by Eq. (6.79) and the Supermartingale Convergence Theorem (cf. Prop. 6.3.10), $\sum_{k=0}^{\infty} z_k < \infty$

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 $\gamma \to \infty$, 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

$$\begin{aligned} z_k &= \frac{2\alpha}{m}\big(f(\hat{x}_k) - f(y_\gamma)\big) - \alpha^2 c_0^2 \\ &\geq \frac{2\alpha}{m}\left(f^* + \frac{2}{\gamma} + \frac{\alpha m c_0^2}{2} - f^* - \frac{1}{\gamma}\right) - \alpha^2 c_0^2 \\ &= \frac{2\alpha}{m\gamma}. \end{aligned}$$

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 $\gamma \to \infty$, we obtain $\inf_{k\geq 0} f(x_k) \leq f^* + \alpha m c_0^2/2$. **Q.E.D.**

From Prop. 6.3.11(b), it can be seen that when $f^* > -\infty$, the randomized method (6.77) 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 (6.69), (6.70), with the same stepsize (cf. Prop. 6.3.8). 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. (6.74) and (6.75)]. Thus when randomization is used, one can afford to use a larger stepsize α 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. 6.3.9 for the nonrandomized method.

Proposition 6.3.12: Assume that the optimal set X^* is nonempty, let Assumption 6.3.3 hold, and let $\{x_k\}$ be the sequence generated by the randomized incremental method (6.77). Then, for any positive scalar ϵ, we have with probability 1

$$\min_{0 \le k \le N} f(x_k) \le f^* + \frac{\alpha m c_0^2 + \epsilon}{2}, \tag{6.80}$$

where N is a random variable with

$$E\{N\} \le \frac{m d(x_0)^2}{\alpha \epsilon}. \tag{6.81}$$

Proof: Define a new process $\{\hat{x}_k\}$ by

$$\hat{x}_{k+1} = \begin{cases} P_X\big(\hat{x}_k - \alpha g(\omega_k, \hat{x}_k)\big) & \text{if } \hat{x}_k \notin L_\gamma, \\ y_\gamma & \text{otherwise,} \end{cases}$$

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

$$L = \left\{ x \in X \;\middle|\; f(x) < f^* + \frac{\alpha m c_0^2 + \epsilon}{2} \right\},$$

the process $\{\hat{x}_k\}$ terminates at $\hat{y}$. Similar to the proof of Prop. 6.3.11 [cf. Eq. (6.78) with $y \in X^*$], for the process $\{\hat{x}_k\}$ we obtain for all k,

$$\begin{aligned} E\big\{d(\hat{x}_{k+1})^2 \mid \mathcal{F}_k\big\} &\le d(\hat{x}_k)^2 - \frac{2\alpha}{m}\big(f(\hat{x}_k) - f^*\big) + \alpha^2 c_0^2 \\ &= d(\hat{x}_k)^2 - z_k, \end{aligned} \tag{6.82}$$

where $\mathcal{F}_k = \{x_0, \ldots, x_k\}$ and

$$z_k = \begin{cases} \frac{2\alpha}{m}\big(f(\hat{x}_k) - f^*\big) - \alpha^2 c_0^2 & \text{if } \hat{x}_k \notin L, \\ 0 & \text{otherwise.} \end{cases}$$

In the case where $\hat{x}_k \notin L$, we have

$$z_k \ge \frac{2\alpha}{m}\left(f^* + \frac{\alpha m c_0^2 + \epsilon}{2} - f^*\right) - \alpha^2 c_0^2 = \frac{\alpha\epsilon}{m}. \tag{6.83}$$

By the Supermartingale Convergence Theorem (cf. Prop. 6.3.10), from Eq. (6.82) 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. (6.82), we obtain for all k,

$$\begin{aligned} E\big\{d(\hat{x}_{k+1})^2\big\} &\leq E\big\{d(\hat{x}_k)^2\big\} - E\{z_k\} \\ &\leq d(x_0)^2 - E\left\{\sum_{j=0}^{k} z_j\right\}, \end{aligned}$$

where in the last inequality we use the facts $\hat{x}_0 = x_0$ and

$$E\big\{d(x_0)^2\big\} = d(x_0)^2.$$

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. (6.83). **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. 6.3.9 that the deterministic method is guaranteed to reach the level set

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

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 (cf. Prop. 6.3.12), 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

$$\left\{x \;\middle|\; f(x) \leq f^* + \frac{\alpha m^2 c_0^2 + \epsilon}{2}\right\},$$

it must process a total of

$$N \leq \frac{m d(x_0)^2}{\alpha\epsilon}$$

component functions (this bound is essentially sharp, as shown in Exercise 6.17).

If in the randomized method we use the same stepsize α, then according to Prop. 6.3.12, we will reach with probability 1 the (much smaller) level set

$$\left\{x \;\Big|\; f(x) \leq f^* + \frac{\alpha m c_0^2 + \epsilon}{2}\right\}$$

after processing N component functions, where the expected value of N satisfies

$$E\{N\} \leq \frac{m d(x_0)^2}{\alpha\epsilon}.$$

Thus, for the same values of α and ϵ, 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 α 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 generally suggests that a randomized order of cost function component selection yields faster convergence than a cyclic order. Aside from random independent sampling of the component functions f_i, another randomization technique is to reshuffle randomly the order of the f_i after each cycle. For a large number of components m, practical experience suggests that this randomization scheme has similar convergence and rate of convergence behavior as the independent random sampling scheme. For small m, the practical performance of the order reshuffling scheme may be even better than random independent sampling. A plausible reason is that the order reshuffling scheme has the nice property of allocating exactly one computation slot to each component in an m-slot cycle. By comparison, choosing components by uniform sampling allocates one computation slot to each component *on the average*, but some components may not get a slot while others may get more than one. A nonzero variance in the number of slots that any fixed component gets within a cycle, may be detrimental to performance.

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 6.4.3-6.4.6, 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

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

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

$$F_k(x) = \max\big\{f(x_0) + (x - x_0)'g_0, \ldots, f(x_k) + (x - x_k)'g_k\big\} \tag{6.84}$$

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

$$x_{k+1} \in \arg\min_{x \in X} F_k(x); \tag{6.85}$$

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 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

$$f(x_i) + (x - x_i)'g_i \le f(x), \qquad \forall\ x \in X,$$

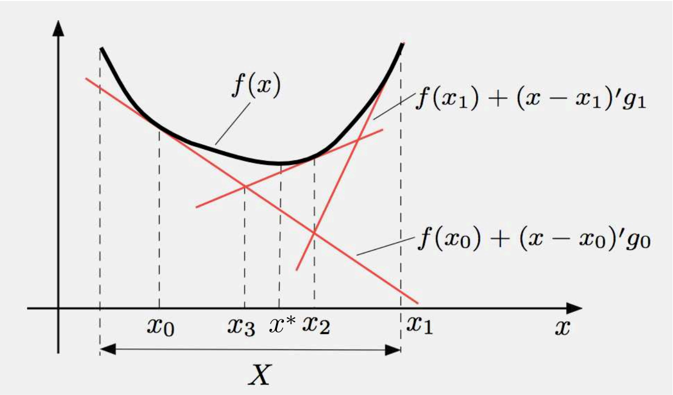

Figure 6.4.1. Illustration of the cutting plane method. With each new iterate x_k, a new hyperplane $f(x_k) + (x - x_k)'g_k$ is added to the polyhedral approximation of the cost function.

so from the definitions (6.84) and (6.85) of F_k and x_k, it follows that

$$f(x_i) + (x_k - x_i)'g_i \le F_{k-1}(x_k) \le F_{k-1}(x) \le f(x), \qquad \forall\ x \in X,\ i < k. \tag{6.86}$$

Suppose that a subsequence $\{x_k\}_K$ converges to $\overline{x}$. Then, since X is closed, we have $\overline{x} \in X$, and by using Eq. (6.86), we obtain for all k and all $i < k$,

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

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

$$\limsup_{\substack{i\to\infty,\, k\to\infty,\, i<k\\ i\in K,\, k\in K}} \big\{ f(x_i) + (x_k - x_i)'g_i \big\} \le \limsup_{k\to\infty,\, k\in K} F_{k-1}(x_k) \le f(\overline{x}).$$

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

$$\lim_{\substack{i\to\infty,\, k\to\infty,\, i<k\\ i\in K,\, k\in K}} (x_k - x_i)'g_i = 0, \tag{6.87}$$

while by the continuity of f, we have

$$f(\overline{x}) = \lim_{i\to\infty,\, i\in K} f(x_i). \tag{6.88}$$

Combining the three preceding relations, we obtain

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

This equation together with Eq. (6.86) yields

$$f(\overline{x}) \le f(x), \qquad \forall\ x \in X,$$

showing that $\overline{x}$ 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 $\Re^n$), provided we assume that $\{g_k\}$ is a bounded sequence [Eq. (6.87) then still holds, while Eq. (6.88) holds as an inequality, but this does not affect the subsequent argument]. Note also that the inequalities

$$F_{k-1}(x_k) \le f^* \le \min_{i\le k} f(x_i), \qquad k = 0, 1, \ldots,$$

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

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

$$f(x) = \max_{i\in I}\big\{a_i'x + b_i\big\}, \tag{6.89}$$

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 $F_{k-1}(x) \le f(x)$ 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 (6.89). Then the cutting plane method, with the subgradient selection and termination rules just described, obtains an optimal solution in a finite number of iterations.

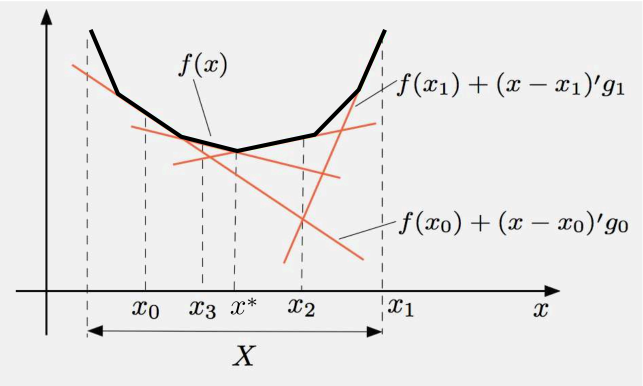

Figure 6.4.2. Illustration of the finite convergence property of the cutting plane method in the case where f is polyhedral. What happens here is that if x_k is not optimal, a new cutting plane will be added at the corresponding iteration, and there can be only a finite number of cutting planes.

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.**

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 $k \to \infty$ 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 pos-

itive δ. 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

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

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

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

where as before

$$F_k(x) = \max\bigl\{f(x_0) + (x - x_0)'g_0, \ldots, f(x_k) + (x - x_k)'g_k\bigr\} \tag{6.90}$$

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

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

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,

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

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., $X \subset \bigl\{x \mid P_k(x) \leq 0\bigr\}$.

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

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

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. (6.85), the method obtains x_{k+1} by finding a "central pair" (x_{k+1}, w_{k+1}) within the subset

$$S_k = \big\{(x,w) \mid x \in X,\ F_k(x) \le w \le \tilde f_k\big\},$$

where $\tilde f_k$ is the best upper bound to the optimal value that has been found so far,

$$\tilde f_k = \min_{i \le k} f(x_i)$$

(see Fig. 6.4.3).

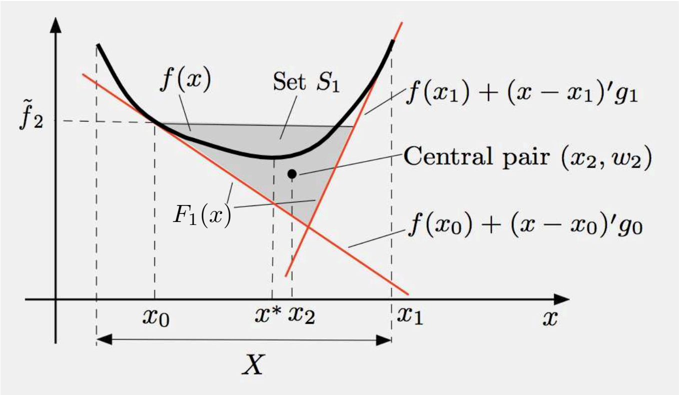

Figure 6.4.3. Illustration of the set

$$S_k = \Big\{(x,w) \mid x \in X,\ F_k(x) \le w \le \tilde f_k\Big\}$$

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 it 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

$$P = \{y \mid a_p'y \le c_p,\ p = 1,\ldots,m\}$$

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,σ):

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

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

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

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 6.9, 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 $f : \Re^n \mapsto \Re$ 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 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

$$\begin{aligned} &\text{minimize} \ \ \nabla f(x_k)'(x - x_k) \\ &\text{subject to} \ \ x \in X. \end{aligned} \tag{6.91}$$

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

$$\begin{aligned} &\text{minimize} \ \ f(x) \\ &\text{subject to} \ \ x \in \text{conv}(X_{k+1}). \end{aligned} \tag{6.92}$$

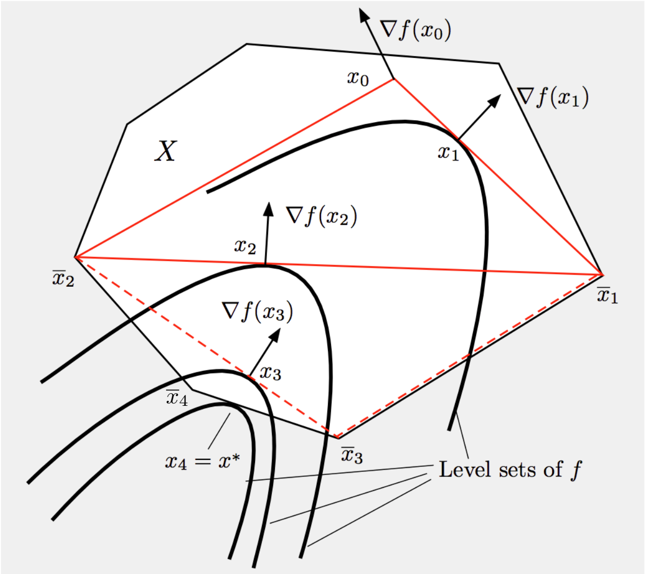

Figure 6.4.4. Successive iterates of the simplicial decomposition method. For example, the figure shows how given the initial point x_0, and the calculated extreme points $\tilde{x}_1$, $\tilde{x}_2$, we determine the next iterate x_2 as a minimizing point of f over the convex hull of $\{x_0, \tilde{x}_1,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

The process is illustrated in Fig. 6.4.4.

For a convergence proof, note that there are two possibilities for the extreme point $\tilde{x}_{k+1}$ that solves problem (6.91):

(a) We have

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

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

$$0 > \nabla f(x_k)'(\tilde{x}_{k+1} - x_k), \tag{6.93}$$

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 $\nabla f(x_k)'(x - x_k) \ge 0$ 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 (6.91). Instead it is sufficient that $\tilde{x}_{k+1}$ is an extreme point and that the condition (6.93) 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 (6.93) is satisfied. These ideas may be used in variants of the simplicial decomposition method whereby problem (6.91) 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 (6.92), 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

$$\nabla f(x_{k+1})'(x - x_{k+1}) \leq 0. \tag{6.94}$$

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 (6.94). 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 (an important example is large-scale multicommodity flow problems arising in communication and transportation network applications; see Example 6.1.10 and 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. Moreover simplicial decomposition does not exhibit the kind of instability phenomenon that is associated with the cutting plane method. In particular, once an optimal solution belongs to X_k, the method will terminate at the next iteration. By contrast, the cutting plane method, even after generating an optimal solution, it may move away from that solution.

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 $f : \Re^n \mapsto (-\infty, \infty]$, an outer linearization of f corresponds to an inner linearization of the conjugate $f^\star$ and reversely.

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

$$F(x) = \max_{i=0,\ldots,k} \big\{ f(x_i) + (x - x_i)' y_i \big\}; \tag{6.95}$$

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

Indeed, we have

$$\begin{aligned} F^\star(y) &= \sup_{x \in \Re^n} \big\{ y'x - F(x) \big\} \\ &= \sup_{x \in \Re^n} \left\{ y'x - \max_{i=0,\ldots,k} \big\{ f(x_i) + (x - x_i)' y_i \big\} \right\} \\ &= \sup_{\substack{x \in \Re^n,\, \xi \in \Re \\ f(x_i) + (x - x_i)' y_i \le \xi,\ i=0,\ldots,k}} \{ y'x - \xi \}. \end{aligned}$$

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

$$F^\star(y) = \inf_{\substack{\sum_{i=0}^k \alpha_i y_i = y,\ \sum_{i=0}^k \alpha_i = 1 \\ \alpha_i \ge 0,\ i=0,\ldots,k}} \sum_{i=0}^k \alpha_i \big(f(x_i) - x_i' y_i \big),$$

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

$$x_i' y_i - f(x_i) = f^\star(y_i), \qquad i = 0, \ldots, k,$$

so we obtain

$$F^\star(y) = \begin{cases} \inf_{\substack{\sum_{i=0}^k \alpha_i y_i = y,\ \sum_{i=0}^k \alpha_i = 1 \\ \alpha_i \ge 0,\ i=0,\ldots,k}} \sum_{i=0}^k \alpha_i f^\star(y_i) & \text{if } y \in \text{conv}\{y_0, \ldots, y_k\}, \\ \infty & \text{otherwise.} \end{cases} \tag{6.96}$$

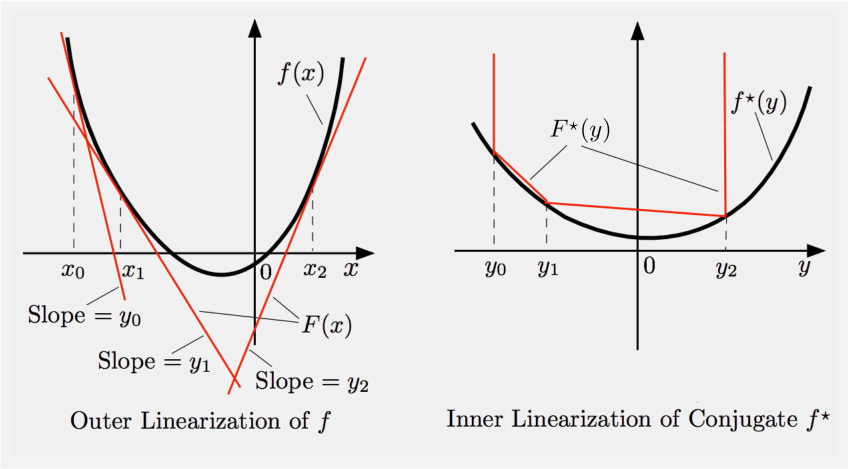

Figure 6.4.5. Illustration of the conjugate $F^\star$ of an outer linearization F of a convex function f (here $k = 2$). It is a piecewise linear, inner linearization of the conjugate $f^\star$ of f. Its break points are the "slopes" $y_0, \ldots, y_k$ of the supporting planes.

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

$$\mathrm{dom}(F^\star) = \mathrm{conv}\{y_0, \ldots, y_k\},$$

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

$$\mathrm{epi}(F^\star) = \mathrm{conv}\Big(\big\{\{(y_i, w_i) \mid f^\star(y_i) \le w_i\} \mid i = 0, \ldots, k\big\}\Big)$$

(see Fig. 6.4.5).

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

$$f(x_i) - y_i'x_i = -f^\star(y_i),$$

from Eq. (6.95) we obtain

$$F(x) = \max_{i=0,\ldots,k} \big\{y_i'x - f^\star(y_i)\big\}.$$

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 $\partial f^\star(y_i) \neq \emptyset$ for all $i = 0, \ldots, k$. Similarly, not every function of the form (6.96) qualifies as an inner linearization within our framework: it is necessary that $\partial f^\star(y_i) \neq \emptyset$ for all $i = 0, \ldots, k$.

6.4.4 Generalized Simplicial Decomposition

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

$$\begin{aligned} &\text{minimize} \ \ f(x) + c(x) \\ &\text{subject to} \ \ x \in \Re^n, \end{aligned} \tag{6.97}$$

where $f : \Re^n \mapsto (-\infty, \infty]$ and $c : \Re^n \mapsto (-\infty, \infty]$ 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). Note that here f need not be differentiable and/or real-valued.

We start the algorithm 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

$$x_k \in \arg \min_{x \in \Re^n} \big\{ f(x) + C_k(x) \big\}, \tag{6.98}$$

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

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

$$-g_k \in \partial C_k(x_k); \tag{6.99}$$

the existence of such a subgradient is guaranteed by the optimality condition of Prop. 5.4.7, applied to the minimization in Eq. (6.98), 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

$$\tilde{x}_{k+1} \in \arg \min_{x \in \Re^n} \big\{ g_k'(x - x_k) + c(x) \big\}, \tag{6.100}$$

and that this is a linear programming problem in the important special case where c is polyhedral. Note also that problem (6.98) is a linearized version of the original problem (6.97), 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 (6.98) involves in effect the variables α_i, $i \in I_k$, and is equivalent to

$$\begin{aligned} &\text{minimize} \quad f\left(\sum_{i\in I_k} \alpha_i \tilde{x}_i\right) + \sum_{i\in I_k} \alpha_i c(\tilde{x}_i) \\ &\text{subject to} \quad \sum_{i\in I_k} \alpha_i = 1, \quad \alpha_i \geq 0,\ i \in I_k. \end{aligned} \tag{6.101}$$

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 (6.91)-(6.92). Indeed, step (1) corresponds to the minimization (6.92), step (2) simply yields $g_k = \nabla f(x_k)$, and step (3), as implemented in Eq. (6.100), corresponds to solution of the linear program (6.91) 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 (6.91)-(6.92) 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 epi(c) is not bounded, but this is inconsequential if we assume that dom(c) is bounded, or more generally that the problem (6.98) has a solution]. In this case, the method terminates finitely, assuming that the vectors $\big(\tilde{x}_{k+1}, c(\tilde{x}_{k+1})\big)$ obtained by solving the linear program (6.100) are extreme points of 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 (6.98) is guaranteed by the compactness of conv(X_k) and Weierstrass' Theorem, while step (2) yields $g_k = \nabla f(x_k)$. The existence of a solution to problem (6.100) 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

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

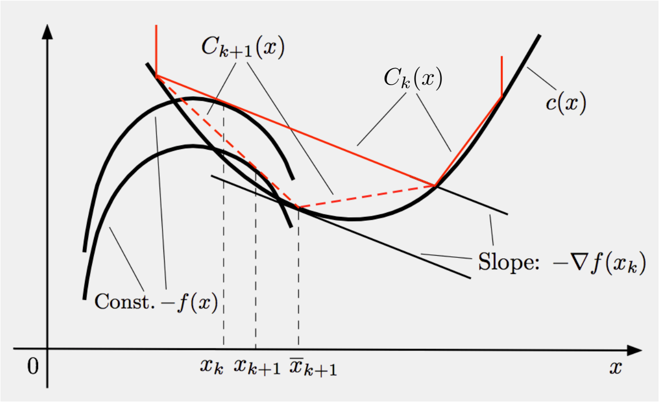

Figure 6.4.6. Illustration of successive iterates of the generalized simplicial decomposition method in the case where f is differentiable. Given the inner linearization C_k of c, we minimize $f + C_k$ to obtain x_k (graphically, we move the graph of $-f$ vertically until it touches the graph of C_k). 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}).

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 (6.98) is guaranteed by Weierstrass' Theorem. When c is the indicator function of a polyhedral set X, the condition of step (2) becomes

$$g_k'(\tilde{x} - x_k) \geq 0, \qquad \forall\ \tilde{x} \in \text{conv}(X_k), \tag{6.102}$$

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 (6.100) is an extreme point of X. The reason is that in view of Eq. (6.102), 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, 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. (6.99)]. 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

$$x_k \in \arg \min_{x \in \Re^n} \big\{ f(x) + C_k(x) \big\}, \tag{6.103}$$

[cf. Eq. (6.98)]. In particular, consider the case where c is the indicator of a closed convex set X, and

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

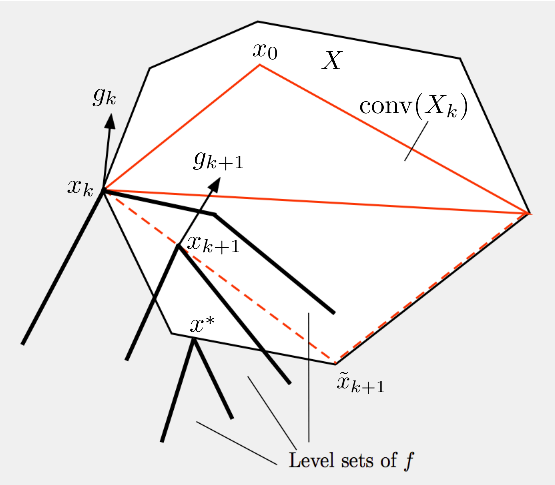

Figure 6.4.7. Illustration of the generalized simplicial decomposition method for the case where f is nondifferentiable and c is the indicator function of a polyhedral set X. For each k, we compute a subgradient $g_k \in \partial f(x_k)$ such that $-g_k$ lies in the normal cone of conv(X_k) at x_k, and 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.

where $f_1, \ldots, f_r$ are differentiable functions. Then the minimization (6.103) takes the form

$$\begin{aligned} &\text{minimize} \quad z \\ &\text{subject to} \quad f_j\,(x) \le z, \ j = 1, \ldots, r, \qquad x \in \text{conv}(X_k), \end{aligned} \tag{6.104}$$

where X_k is a polyhedral inner linearization to X. According to the optimality conditions of Prop. 6.1.3, the optimal solution (x_k, z^*) together with dual optimal variables $\mu_j^* \geq 0$, satisfies

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

and the complementary slackness conditions $f_j(\hat{x}^k) = z^*$ if $\mu_j^* > 0$. It follows that

$$\sum_{j=1}^r \mu_j^* = 1, \qquad \mu_j^* \geq 0, \quad j = 1, \ldots, r, \tag{6.105}$$

and

$$\left(\sum_{j=1}^r \mu_j^* \nabla f_j(x_k) \right)' (x - x_k) \geq 0, \qquad \forall\ x \in \text{conv}(X_k). \tag{6.106}$$

From Eq. (6.105) and the analysis of Example 5.4.5, the vector

$$g_k = \sum_{j=1}^{r} \mu_j^* \nabla f_j(x_k) \tag{6.107}$$

is a subgradient of f at x_k. Furthermore, from Eq. (6.106), it follows that $-g_k$ is in the normal cone of $\mathrm{conv}(X_k)$ at x_k.

We next consider a more general problem where there are additional inequality constraints defining the domain of f. This is the case where f is of the form

$$f(x) = \begin{cases} \max\big\{f_1(x), \ldots, f_r(x)\big\}, & \text{if } g_i(x) \leq 0, \ i = 1, \ldots, p, \\ \infty & \text{otherwise,} \end{cases} \tag{6.108}$$

with f_j and g_i being convex differentiable functions. Applications of this type include multicommodity flow problems with side constraints (the inequalities $g_i(x) \leq 0$, which are separate from the network flow constraints that comprise the set C; cf. [Ber98], Chapter 8, [LaP99]). The case where $r = 1$ and there are no side constraints is important in a variety of communication, transportation, and other resource allocation problems, and is one of the principal successful applications of simplicial decomposition; see e.g., [FlH95]. Side constraints and nondifferentiabilities in this context are often eliminated using barrier, penalty, or Augmented Lagrangian functions, but this can be awkward and restrictive. Our approach allows a more direct treatment.

As in the preceding case, we introduce additional dual variables $\nu_i^* \geq 0$ for the constraints $g_i(x) \leq 0$, and we write the Lagrangian optimality and complementary slackness conditions. Then Eq. (6.106) takes the form

$$\left(\sum_{j=1}^{r} \mu_j^* \nabla f_j(\hat{x}^k) + \sum_{i=1}^{p} \nu_i^* \nabla g_i(\hat{x}^k)\right)' (x - \hat{x}^k) \geq 0, \qquad \forall\, x \in \mathrm{conv}(X_k),$$

and it can be shown that the vector $\hat{\lambda}^k = \sum_{j=1}^{r} \mu_j^* \nabla f_j(\hat{x}^k) + \sum_{i=1}^{p} \nu_i^* \nabla g_i(\hat{x}^k)$ is a subgradient of f at $\hat{x}^k$, while $-\hat{\lambda}^k \in \partial H_k(\hat{x}^k)$ as required by Eq. (6.99).

Note an important advantage of this method over potential competitors: it involves solution of linear programs of the form (6.100) to generate new extreme points of X, and low-dimensional nonlinear programs of the form (6.104). 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

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

[cf. Eq. (6.97)] 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

$$\begin{aligned}&\text{minimize} \;\; f_1^\star(\lambda) + f_2^\star(-\lambda)\\ &\text{subject to} \;\; \lambda \in \Re^n,\end{aligned}$$

where $f_1^\star$ and $f_2^\star$ are the conjugates of f and c, respectively. The generalized simplicial decomposition algorithm (6.98)-(6.100) can alternatively be implemented by replacing $f_2^\star$ by a piecewise linear/cutting plane outer linearization, while leaving $f_1^\star$ unchanged, i.e., by solving at iteration k the problem

$$\begin{aligned}&\text{minimize} \;\; f_1^\star(\lambda) + F_{2,k}^\star(-\lambda)\\ &\text{subject to} \;\; \lambda \in \Re^n,\end{aligned} \tag{6.109}$$

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

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

[cf. problem (6.98) or equivalently, the low-dimensional problem (6.101)].

Note that solutions of problem (6.109) are the subgradients g_k satisfying Eq. (6.99), while the associated subgradient of $f_2^\star$ at $-g_k$ is the vector $\tilde{x}_{k+1}$ generated by Eq. (6.100), as shown in Fig. 6.4.8. In fact, the function $F_{2,k}^\star$ has the form

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

where g_i and $\tilde{x}_{i+1}$ are vectors that can be obtained either by using the primal, the generalized simplicial decomposition method (6.98)-(6.100), or by using its dual, the cutting plane method based on solving the outer approximation problems (6.109). The ordinary cutting plane method, described in the beginning of Section 6.4.1, is obtained as the special case where $f_1^\star(\lambda) \equiv 0$.

Whether the primal or the dual implementation is preferable depends on the structure of the functions f and c. When f (and hence also $f_1^\star$) is not polyhedral, the dual implementation may not be attractive, because it requires the n-dimensional nonlinear optimization (6.109) at each iteration, as opposed to the typically low-dimensional optimization (6.98). In the alternative case where f is polyhedral, both methods require the solution of linear programs.

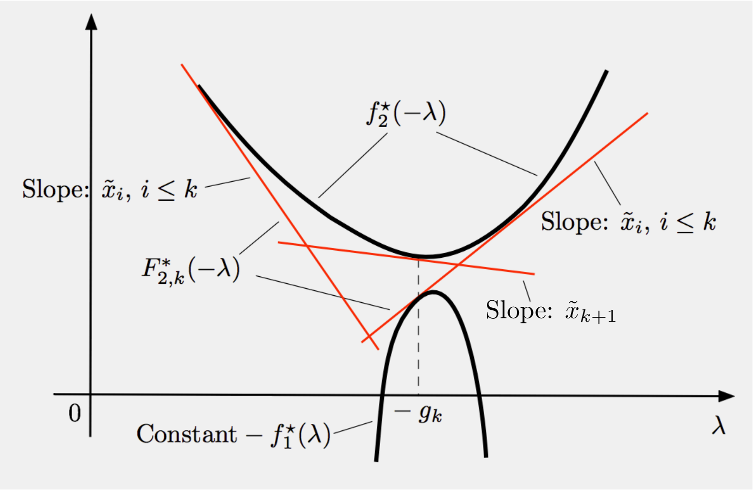
$f_2^\star(-\lambda)$
Slope: $\tilde{x}_i,\ i \le k$
Slope: $\tilde{x}_i,\ i \le k$
$F_{2,k}^\star(-\lambda)$
Slope: $\tilde{x}_{k+1}$
0
$-\ g_k$
Constant $-\ f_1^\star(\lambda)$
λ

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 $f_1^\star(x) \equiv 0$. In this case, f is the indicator function of the set consisting of just the origin, and the primal problem is to evaluate $c(0)$.

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

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^m f_i(x_i) \\ &\text{subject to} \quad (x_1,\ldots,x_m) \in S, \end{aligned} \tag{6.110}$$

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

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

S is a subspace of $\Re^{n_1+\cdots+n_m}$.

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

† Monotropic programming, a class of problems introduced and extensively analyzed by Rockafellar in his book [Roc84], is the special case of problem (6.110) 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.

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

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} f_i(x_i) \\ &\text{subject to} \quad Ax = b, \end{aligned} \tag{6.111}$$

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

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

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

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

When the functions f_i are linear, problem (6.111) reduces to a linear programming problem. When the functions $f_i(x_i)$ are positive semidefinite quadratic, problem (6.111) reduces to a convex quadratic programming problem.

Note also that while the vectors $x_1, \ldots, x_m$ 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

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

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

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

where S is the subspace of $\Re^{2(n_1+\cdots+n_m)}$

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

This problem is of the form (6.110).

Another problem that can be converted to the EMP format (6.110) is

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^m f_i(x) \\ &\text{subject to} && x \in X, \end{aligned} \tag{6.112}$$

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

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

where S is the subspace

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

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 $z_i \in \Re^{n_i}$ and we convert the EMP (6.110) to the equivalent form

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^m f_i(z_i) \\ &\text{subject to} && z_i = x_i, \quad i = 1,\ldots,m, \qquad (x_1,\ldots,x_m) \in S. \end{aligned} \tag{6.113}$$

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

$$L(x_1,\ldots,x_m,z_1,\ldots,z_m,\lambda_1,\ldots,\lambda_m) = \sum_{i=1}^m \big(f_i(z_i) + \lambda_i'(x_i - z_i)\big). \tag{6.114}$$

The dual function is

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

where

$$q_i(\lambda_i) = \inf_{z_i\in\Re^{n_i}} \big\{ f_i(z_i) - \lambda_i' z_i \big\}, \qquad i = 1,\ldots,m,$$

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

Note that since q_i can be written as

$$q_i(\lambda_i) = -\sup_{z_i\in\Re} \big\{ \lambda_i' z_i - f_i(z_i) \big\},$$

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

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^m q_i(\lambda_i) \\ &\text{subject to} && (\lambda_1,\ldots,\lambda_m)\in S^\perp. \end{aligned} \qquad (6.115)$$

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

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^m f_i^\star(\lambda_i) \\ &\text{subject to} && (\lambda_1,\ldots,\lambda_m)\in S^\perp, \end{aligned} \qquad (6.116)$$

where $f_i^\star$ 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, λ) satisfies the Lagrangian optimality condition of Prop. 5.3.3, applied to the Lagrangian (6.114), if and only if x_i attains the infimum in the equation

$$q_i(\lambda_i) = \inf_{z_i \in \Re^{n_i}} \big\{ f_i(z_i) - \lambda_i' z_i \big\}, \qquad i = 1, \ldots, m,$$

or equivalently,

$$\lambda_i \in \partial f_i(x_i), \qquad i = 1, \ldots, m. \tag{6.117}$$

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 $(x_1^{opt}, \ldots, x_m^{opt}, \lambda_1^{opt}, \ldots, \lambda_m^{opt})$ are an optimal primal and dual solution pair of the EMP problem if and only if

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

and

$$x_i^{opt} \in \arg \min_{x_i \in \Re^n} \big\{ f_i(x_i) - x_i' \lambda_i^{opt} \big\}, \quad i = 1, \ldots, m. \tag{6.118}$$

Note that by the Conjugate Subgradient Theorem (Prop. 5.4.3), the condition (6.118) of the preceding proposition is equivalent to either one of the following two subgradient conditions:

$$\lambda_i^{opt} \in \partial f_i(x_i^{opt}), \qquad x_i^{opt} \in \partial f_i^\star(\lambda_i^{opt}).$$

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 (6.116) 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 $\{1, \ldots, m\}$:

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

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 $\Lambda_i \subset \mathrm{dom}(f_i^\star)$ such that $\partial f_i^\star(\tilde\lambda) \neq \emptyset$ for all $\tilde\lambda \in \Lambda_i$, we consider the outer linearization of f_i corresponding to Λ_i:

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

or equivalently, as mentioned in Section 6.4.3,

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

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 $X_i \subset \mathrm{dom}(f_i)$ 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

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

As mentioned in Section 6.4.3, this is the function whose epigraph is the convex hull of the halflines $\big\{(x_i, w) \mid f_i(x_i) \leq w\big\}$, $x_i \in X_i$ (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 $\Lambda_i \subset \mathrm{dom}(f_i^\star)$, and for each $i \in \bar I$, a finite subset $X_i \subset \mathrm{dom}(f_i)$. The iteration is as follows:

Typical Iteration:

Find a primal-dual optimal solution pair $(\hat x, \hat\lambda) = (\hat x_1, \hat\lambda_1, \ldots, \hat x_m, \hat\lambda_m)$ of the EMP

$$\begin{aligned} &\text{minimize} \quad \sum_{i \in I} f_i(x_i) + \sum_{i \in \underline{I}} \underline{f}_{i,\Lambda_i}(x_i) + \sum_{i \in \bar I} \bar f_{i,X_i}(x_i) \\ &\text{subject to} \quad (x_1, \ldots, x_m) \in S, \end{aligned} \tag{6.119}$$

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 Λ_i, respectively. Then enlarge the sets X_i and Λ_i as follows (see Fig. 6.4.9):

(a) For $i \in \underline{I}$, we compute a subgradient $\tilde\lambda_i \in \partial f_i(\hat x_i)$ and we add $\tilde\lambda_i$ to the corresponding set Λ_i.

(b) For $i \in \bar I$, we compute a subgradient $\tilde x_i \in \partial f_i^\star(\hat\lambda_i)$ 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, $(\hat{x}_1, \ldots, \hat{x}_m, \hat{\lambda}_1, \ldots, \hat{\lambda}_m)$ 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 Λ_i and X_i.

Note that we implicitly assume that at each iteration, there exists a primal and dual optimal solution pair of problem (6.119). 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 $\partial f_i^\star(\hat{\lambda}_i) \neq \emptyset$ for all $i \in \bar{I}$. Sufficient assumptions may need to be imposed on the problem to guarantee that this is so.

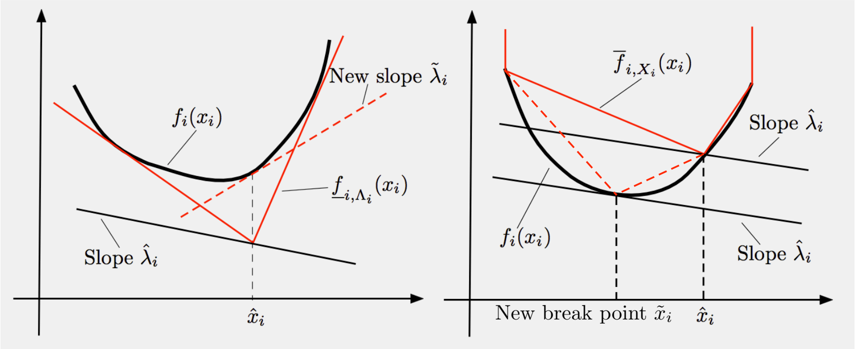
New slope $\tilde{\lambda}_i$
$f_i(x_i)$
$\underline{f}_{i,\Lambda_i}(x_i)$
Slope $\hat{\lambda}_i$
$\hat{x}_i$
$\overline{f}_{i,X_i}(x_i)$
Slope $\hat{\lambda}_i$
$f_i(x_i)$
Slope $\hat{\lambda}_i$
New break point $\tilde{x}_i$
$\hat{x}_i$

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

$$\tilde{x}_i \in \partial f_i^\star(\hat{\lambda}_i) \qquad \Longleftrightarrow \qquad \hat{\lambda}_i \in \partial f_i(\tilde{x}_i)$$

(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 $\hat{x}_i \in \partial f_i^\star(\tilde{\lambda}_i)$, or equivalently, solving the optimization problem

$$\begin{array}{ll} \text{maximize} & \left\{ \lambda_i' \hat{x}_i - f_i^\star(\lambda_i) \right\} \\ \text{subject to} & \lambda_i \in \Re^{n_i}. \end{array}$$

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

$$\begin{array}{ll} \text{maximize} & \left\{ \hat{\lambda}_i' x_i - f_i(x_i) \right\} \\ \text{subject to} & x_i \in \Re^{n_i}. \end{array}$$

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 (6.119) must be easier to solve than the original problem (6.116). For example, problem (6.119) 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 $\tilde x_i \in \partial f_i^\star(\hat\lambda_i)$ 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 $f_i^\star$ are not available in closed form), one may calculate λ_i and/or $\tilde x_i$ via the relations

$$\hat x_i \in \partial f_i^\star(\tilde\lambda_i), \qquad \hat\lambda_i \in \partial f_i(\tilde x_i);$$

(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

$$\begin{array}{ll} \text{maximize} & \big\{\hat\lambda_i' x_i - f_i(x_i)\big\} \\ \text{subject to} & x_i \in \Re^{n_i}, \end{array}$$

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

The facility of solving the linearized problem (6.119) 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, as is often 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.,

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

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 (6.116):

$$\begin{aligned}&\text{minimize} && \sum_{i=1}^m f_i^\star(\lambda_i)\\ &\text{subject to} && (\lambda_1,\dots,\lambda_m)\in S^\perp,\end{aligned}\tag{6.120}$$

where $f_i^\star$ 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,

$$\begin{aligned}&\text{minimize} && \sum_{i\in I} f_i^\star(\lambda_i)+\sum_{i\in \underline{I}}\bar f_{i,\Lambda_i}^\star(\lambda_i)+\sum_{i\in \bar I}\underline{f}_{i,X_i}^\star(\lambda_i)\\ &\text{subject to} && (\lambda_1,\dots,\lambda_m)\in S^\perp,\end{aligned}\tag{6.121}$$

which is simply the dual of the approximate primal EMP (6.119) [since the outer (or inner) linearization of $f_i^\star$ 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 simply a matter of whether calculations with f_i or with their conjugates $f_i^\star$ 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 any closed proper convex functions f and $\underline{f}$, we have

$$\underline{f} \leq f, \quad \underline{f}(x) = f(x) \qquad \Longrightarrow \qquad \partial \underline{f}(x) \subset \partial f(x). \tag{6.122}$$

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

$$\tilde{\lambda} \in \Lambda, \quad \tilde{\lambda} \in \partial f(x) \qquad \Longrightarrow \qquad \underline{f}_\Lambda(x) = f(x). \tag{6.123}$$

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, $(\hat{x}_1, \ldots, \hat{x}_m)$ and $(\hat{\lambda}_1, \ldots, \hat{\lambda}_m)$, form a primal and dual optimal solution pair of the EMP problem.

Proof: From Prop. 6.4.3 and the definition of $(\hat{x}_1, \ldots, \hat{x}_m)$ and $(\hat{\lambda}_1, \ldots, \hat{\lambda}_m)$ as a primal and dual optimal solution pair of the approximate problem (6.119), we have

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

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

$$\hat{\lambda}_i \in \partial f_i(\hat{x}_i), \tag{6.124}$$

which by Prop. 6.4.3 implies the desired conclusion.

Since $(\hat{x}_1, \ldots, \hat{x}_m)$ and $(\hat{\lambda}_1, \ldots, \hat{\lambda}_m)$ are a primal and dual optimal solution pair of problem (6.119), Eq. (6.124) 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. (6.123), 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. (6.122) that

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

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 enlargement vectors $\tilde{\lambda}_i$ 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 $f_i^\star$ 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 $f_i^\star$ 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, let $(\hat{x}^k, \hat{\lambda}^k)$ be the primal and dual pair generated at iteration k, and let $\tilde{\lambda}_i^k \in \partial f_i(\hat{x}_i^k)$ for $i \in \underline{I}$, and $\tilde{x}_i^k \in \partial f_i^\star(\hat{\lambda}_i^k)$ for $i \in \bar{I}$ be the vectors used for the corresponding enlargements. 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 we can easily show that every limit point of $\{\hat{x}^k\}$ is primal optimal. To see this, note that for all k, $\ell \leq k-1$, and $(x_1, \ldots, x_m) \in S$, we have

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

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

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

for all $(x_1, \ldots, x_m) \in S$. 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 [BeY11], 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 [BeY11] 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 kth 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 $\{\hat{x}_i^k\}_{\mathcal{K}}, i \in \underline{I}$, $\{\hat{\lambda}_i^k\}_{\mathcal{K}}, i \in \bar{I}$, 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

$$\begin{aligned} &\text{minimize} \quad f(x) + c(x) \\ &\text{subject to} \quad x \in \Re^n, \end{aligned} \tag{6.125}$$

where $f : \Re^n \mapsto (-\infty, \infty]$ and $c : \Re^n \mapsto (-\infty, \infty]$ 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

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

where

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

The dual problem takes the form

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

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

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

the dual problem is

$$\begin{array}{ll} \text{minimize} & f_1^\star(\lambda) + f_2^\star(-\lambda) \\ \text{subject to} & \lambda \in \Re^n. \end{array}$$

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 Λ 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 $x = \big\{x_a \mid a \in \mathcal{A}\big\}$ 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:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{a \in \mathcal{A}} \bar f_{a,X_a}(x_a) \\ \text{subject to} & x \in S, \end{array}$$

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 $X_a \subset \mathrm{dom}(f_a)$. 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 ϵ-relaxation methods [Ber98], [TsB00].

6.4.6 Simplicial Decomposition for Conic Programming

We will now aim to extend the range of applications of the generalized polyhedral approximation approach for EMP by allowing conic-like constraint sets. Our motivation is that the algorithmic framework of the preceding subsection is not well-suited for the case where some of the component functions of the cost are indicator functions of cones. There are two main reasons for this:

(1) The enlargement procedure requires the minimization of a polyhedral approximation function over a cone, which may not have a solution.

(2) The inner linearization method approximates a cone (an unbounded set) by the convex hull of a finite number of points (a compact set). It seems evident that a cone generated by a finite number of directions should provide a more effective approximation.

Motivated by these concerns, we extend our generalized polyhedral approximation approach so that it applies to the problem of minimizing the sum $\sum_{i=1}^{m} f_i(x_i)$ of convex extended real-valued functions f_i, subject to $(x_1, \ldots, x_m)$ being in the intersection of a given subspace and the Cartesian product of closed convex cones.

For simplicity, we focus on the pure simplicial decomposition approach (and by duality on the pure cutting plane approach). It is straightforward to extend our algorithms to the mixed case, where some of the component functions are inner linearized while others are outer linearized.

As a first step, we recall the conjugacy relation between inner and outer linearization for a closed proper convex function f and its conjugate $f^\star$, which we discussed earlier. In particular, for a given finite set of vectors X, the inner linearization $\overline{f}_X$ of f is the polyhedral function whose epigraph is the convex hull of the union of the vertical halflines corresponding to

$x \in X$:

$$\mathrm{epi}(\overline{f}_X) = \mathrm{conv}\Big(\cup_{x\in X} \big\{(x,w) \mid f(x) \le w\big\}\Big) \tag{6.126}$$

(see Fig. 6.4.5). The conjugate of $\overline{f}_X$ is an outer linearization of $f^\star$ defined by hyperplanes that support the epigraph of f at points λ_x, $x \in X$, such that $x \in \partial f^\star(\lambda_x)$ for each $x \in X$. It is given by

$$\overline{f}^\star_X(\lambda) = \max_{x\in X}\big\{f^\star(\lambda_x) + (\lambda - \lambda_x)'x\big\}. \tag{6.127}$$

Note that $\overline{f}^\star_X$ is a real-valued polyhedral function, and that its values $\overline{f}^\star_X(\lambda)$ do not depend on the vectors λ_x, as long as $x \in \partial f^\star(\lambda_x)$ for all $x \in X$ (cf. Fig. 6.4.5).

When the preceding definition of inner linearization is specialized to the indicator function $\delta(\cdot \mid C)$ of a closed convex set C, it amounts to approximating C with the convex hull of the finite number of points $X \subset C$. The conjugate of $\delta(\cdot \mid C)$ is the support function of C, and its outer linearization is the support function of the convex hull of X.

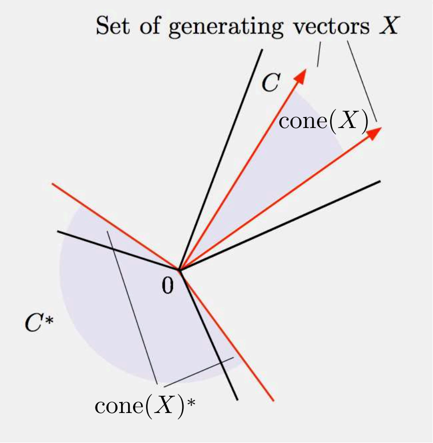

Figure 6.4.10. Illustration of cone(X) as an inner linearization of a cone C and its polar cone(X)* as an outer linearization of the polar

$$C^* = \{y \mid y'x \le 0,\ \forall\ x \in C\}.$$

If C is a closed convex cone, we will be using alternative and more symmetric outer and inner linearizations, which are based on generated cones rather than convex hulls. In particular, given a finite subset $X \subset C$, we view cone(X) as an inner linearization of C and its polar cone(X)* as an outer linearization of the polar C^* (see Fig. 6.4.10). This type of linearization has a twofold advantage: a cone is approximated by a cone (rather than by a compact set), and outer and linear linearizations yield convex functions of the same type as the original (indicator functions of cones).

Duality and Optimality Conditions

We now introduce a version of the EMP problem, generalized to include cone constraints. It is given by

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} f_i(x_i) + \sum_{i=m+1}^{r} \delta(x_i \mid C_i) \\ &\text{subject to} \ \ (x_1, \ldots, x_r) \in S, \end{aligned} \tag{6.128}$$

where $(x_1, \ldots, x_r)$ is a vector in $\Re^{n_1 + \cdots + n_r}$, with components $x_i \in \Re^{n_i}$, $i = 1, \ldots, r$, and

> $f_i : \Re^{n_i} \mapsto (-\infty, \infty]$ is a closed proper convex function for each i.
>
> S is a subspace of $\Re^{n_1 + \cdots + n_r}$.
>
> $C_i \subset \Re^{n_i}$, $i = m+1, \ldots, r$, is a closed convex cone, and $\delta(x_i \mid C_i)$ denotes the indicator function of C_i.

According to the EMP duality theory, the dual problem is

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} f_i^\star(\lambda_i) + \sum_{i=m+1}^{r} \delta(\lambda_i \mid C_i^\star) \\ &\text{subject to} \ \ (\lambda_1, \ldots, \lambda_r) \in S^\perp, \end{aligned} \tag{6.129}$$

and has the same form as the primal problem (6.128). Furthermore, since f_i is assumed closed proper and convex, and C_i is assumed closed convex, we have $f_i^{\star\star} = f_i$ and $(C_i^\star)^\star = C$, where $f_i^{\star\star}$ is the conjugate of $f_i^\star$ and $(C_i^\star)^\star$ is the polar of $C_i^\star$. Thus when the dual problem is dualized, it yields the primal problem, and the duality is fully symmetric.

To derive an appropriate dual problem, we introduce auxiliary vectors $z_i \in \Re^{n_i}$ and we convert problem (6.128) to the equivalent form

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} f_i(z_i) + \sum_{i=m+1}^{r} \delta(z_i \mid C_i) \\ &\text{subject to} \ \ z_i = x_i, \quad i = 1, \ldots, r, \qquad (x_1, \ldots, x_r) \in S. \end{aligned} \tag{6.130}$$

Then we assign a multiplier vector $\lambda_i \in \Re^{n_i}$ to the constraint $z_i = x_i$, and obtain the dual problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} f_i^\star(\lambda_i) + \sum_{i=m+1}^{r} \delta(\lambda_i \mid C_i^*) \\ &\text{subject to} \ \ (\lambda_1, \ldots, \lambda_r) \in S^\perp. \end{aligned} \tag{6.131}$$

which has the same form as the primal problem (6.128). (We leave the verification of this as an exercise for the reader.)

To state the associated optimality conditions, we first provide some preliminary duality relations for cones. We say that (x, λ) is a *dual pair with respect to the closed convex cones C and $C^\star$* if

$$x = P_C(x + \lambda) \qquad \text{and} \qquad \lambda = P_{C^\star}(x + \lambda).$$

The following result shows that there is a (necessarily unique) representation of a vector y as $y = x + \lambda$, where (x, λ) is a dual pair with respect to C and $C^\star$.

Proposition 6.4.6: (Cone Decomposition Theorem) Let C be a nonempty closed convex cone in $\Re^n$ and $C^\star$ be its polar cone.

(a) Any vector y can be written as $y = P_C(y) + P_{C^\star}(y)$.

(b) The following conditions are equivalent:

(i) (x, λ) is a dual pair with respect to C and $C^\star$.

(ii) $x \in C$, $\lambda \in C^\star$, and $x \perp \lambda$.

Proof: (a) We denote $\xi = y - P_C(y)$, and we will show that $\xi = P_{C^\star}(y)$. Indeed, by the projection theorem (Prop. 1.1.9), we have

$$\xi'\big(z - P_C(y)\big) \leq 0, \qquad \forall\ z \in C. \tag{6.132}$$

Since C is a cone, we have $(1/2)P_C(y) \in C$ and $2P_C(y) \in C$, so by taking $z = (1/2)P_C(y)$ and $z = 2P_C(y)$ in Eq. (6.132), it follows that

$$\xi' P_C(y) = 0. \tag{6.133}$$

By combining Eqs. (6.132) and (6.133), we obtain $\xi' z \leq 0$ for all $z \in C$, implying that $\xi \in C^\star$. Moreover, since $P_C(y) \in C$, we have

$$(y - \xi)'(z - \xi) = P_C(y)'(z - \xi) = P_C(y)' z \leq 0, \qquad \forall\ z \in C^\star,$$

where the second equality follows from Eq. (6.133). Thus ξ satisfies the necessary and sufficient condition for being the projection $P_{C^\star}(y)$.

(b) Suppose that property (i) holds, i.e., x and λ are the projections of $x + \lambda$ on C and $C^\star$, respectively. Then we have, using also Eq. (6.133),

$$x \in C, \qquad \lambda \in C^\star, \qquad \big((x + \lambda) - x\big)' x = 0,$$

or

$$x \in C, \qquad \lambda \in C^\star, \qquad \lambda' x = 0,$$

which is property (ii).

Conversely, suppose that property (ii) holds. Then, since $\lambda \in C^\star$, we have $\lambda' z \le 0$ for all $z \in C$, and hence

$$\big((x+\lambda)-x\big)'(z-x) = \lambda'(z-x) = \lambda' z \le 0, \qquad \forall\ z \in C,$$

where the second equality follows from the fact $x \perp \lambda$. Thus x satisfies the necessary and sufficient condition for being the projection $P_C(x+\lambda)$. Similarly, we show that λ is the projection $P_{C^\star}(x+\lambda)$. **Q.E.D.**

Let us denote by f_{opt} and q_{opt} the optimal primal and dual values. Assuming that strong duality holds ($q_{opt} = f_{opt}$), we can use the optimality conditions, whereby (x^{opt}, λ^{opt}) form an optimal primal and dual solution pair if and only if they satisfy the standard primal feasibility, dual feasibility, and Lagrangian optimality conditions (cf. Prop. 5.3.3). By working out these conditions similar to our earlier analysis, we obtain the following proposition.

Proposition 6.4.7: (Optimality Conditions) We have $-\infty < q_{opt} = f_{opt} < \infty$, and $x^{opt} = (x_1^{opt}, \ldots, x_r^{opt})$ and $\lambda^{opt} = (\lambda_1^{opt}, \ldots, \lambda_r^{opt})$ are optimal primal and dual solutions, respectively, of problems (6.128) and (6.131) if and only if

$$(x_1^{opt}, \ldots, x_r^{opt}) \in S, \qquad (\lambda_1^{opt}, \ldots, \lambda_r^{opt}) \in S^\perp, \tag{6.134}$$

$$x_i^{opt} \in \partial f_i^\star(\lambda_i^{opt}), \qquad i = 1, \ldots, m, \tag{6.135}$$

$(x_i^{opt}, \lambda_i^{opt})$ is a dual pair with respect to C_i and $C_i^\star$, $i = m+1, \ldots, r$. (6.136)

Note that the condition $x_i^{opt} \in \partial f_i^\star(\lambda_i^{opt})$ of the preceding proposition is equivalent to

$$\lambda_i^{opt} \in \partial f_i(x_i^{opt})$$

(cf. the Conjugate Subgradient Theorem, Prop. 5.4.3). Thus the optimality conditions are fully symmetric, consistently with the symmetric form of the primal and dual problems (6.128) and (6.131).

Simplicial Decomposition for Conical Constraints

We will now introduce our algorithm, whereby problem (6.128) is approximated by using inner linearization of some of the functions f_i and of all the cones C_i. The optimal primal and dual solution pair of the approximate problem is then used to construct more refined inner linearizations. The algorithm uses a fixed subset $\bar{I} \subset \{1, \ldots, m\}$, which corresponds to functions f_i that are inner linearized. For notational convenience, we denote by I the complement of $\bar{I}$ in $\{1, \ldots, m\}$, so that

$$\{1, \ldots, m\} = I \cup \bar{I},$$

and we also denote†

$$I_c = \{m+1, \ldots, r\}.$$

At the typical iteration of the algorithm, we have for each $i \in \bar{I}$, a finite set X_i such that $\partial f_i(x_i) \neq \varnothing$ for all $x_i \in X_i$, and for each $i \in I_c$ a finite set $X_i \subset C_i$. The iteration is as follows.

Typical Iteration:

Step 1: (Approximate Problem Solution) Find a primal and dual optimal solution pair

$$(\hat{x}, \hat{\lambda}) = (\hat{x}_1, \ldots, \hat{x}_r, \hat{\lambda}_1, \ldots, \hat{\lambda}_r)$$

of the problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i \in I} f_i(x_i) + \sum_{i \in \bar{I}} \bar{f}_{i,X_i}(x_i) + \sum_{i \in I_c} \delta\big(x_i \mid \text{cone}(X_i)\big) \\ &\text{subject to} \quad (x_1, \ldots, x_r) \in S, \end{aligned} \qquad (6.137)$$

where $\bar{f}_{i,X_i}$ are the inner linearizations of f_i corresponding to X_i.

Step 2: (Enlargement and Test for Termination) Enlarge the sets X_i as follows (see Fig. 6.4.11):

(a) For $i \in \bar{I}$, we add any subgradient $\tilde{x}_i \in \partial f_i^\star(\hat{\lambda}_i)$ to X_i.

(b) For $i \in I_c$, we add the projection $\tilde{x}_i = P_{C_i}(\hat{\lambda}_i)$ to X_i.

If there is no strict enlargement for all $i \in \bar{I}$, i.e., we have $\tilde{x}_i \in X_i$, and moreover $\tilde{x}_i = 0$ for all $i \in I_c$, the algorithm terminates. Otherwise, we proceed to the next iteration, using the enlarged sets X_i.

Note that we implicitly assume that at each iteration, there exists a primal and dual optimal solution pair of problem (6.137). The algorithm

† We allow $\bar{I}$ to be empty, so that none of the functions f_i is inner linearized. In this case the portions of the subsequent algorithmic descriptions and analysis that refer to the functions f_i with $i \in \bar{I}$ should be simply omitted. Also, there is no loss of generality in using $I_c = \{m+1, \ldots, r\}$, since the indicator functions of the cones that are not linearized, may be included within the set of functions f_i, $i \in I$.

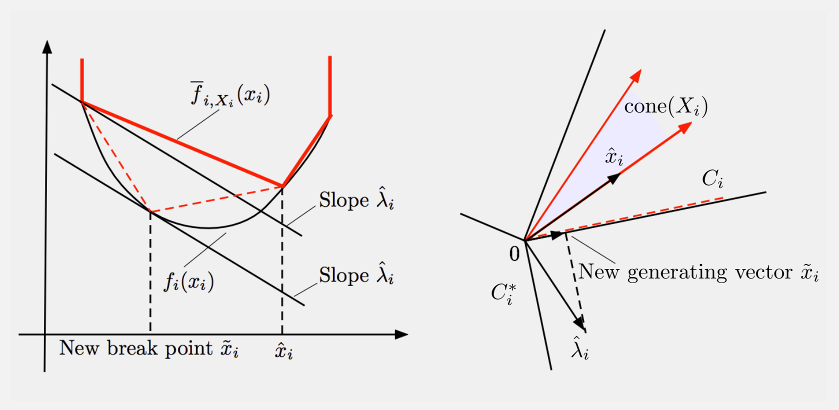

Figure 6.4.11. Illustration of the enlargement step of the algorithm, after we obtain a primal and dual optimal solution pair $(\hat{x}_1,\ldots,\hat{x}_r,\hat{\lambda}_1,\ldots,\hat{\lambda}_r)$. The enlargement step on the left [finding $\tilde{x}_i$ with $\tilde{x}_i \in \partial f_i^\star(\hat{\lambda}_i)$ for $i \in \bar{I}$] is also equivalent to finding $\tilde{x}_i$ satisfying $\hat{\lambda}_i \in \partial f_i(\tilde{x}_i)$, or equivalently, solving the optimization problem

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

The enlargement step on the right, for $i \in I_c$, is to add to X_i the vector $\tilde{x}_i = P_{C_i}(\hat{\lambda}_i)$, the projection on C_i of $\hat{\lambda}_i$.

for finding such a pair is left unspecified. Furthermore, we assume that the enlargement step can be carried out, i.e., that $\partial f_i^\star(\hat{\lambda}_i) \neq \emptyset$ for all $i \in \bar{I}$. Sufficient assumptions may need to be imposed on the problem to guarantee that this is so. Regarding the termination condition $\tilde{x}_i = 0$ for $i \in I_c$, note that it is simpler and weaker than the alternative conditions $\tilde{x}_i \in X_i$ or $\tilde{x}_i \in \text{cone}(X_i)$, which imply that $\tilde{x}_i = 0$ [if $P_{C_i}(\hat{\lambda}_i) = \tilde{x}_i \in \text{cone}(X_i)$, then $P_{C_i}(\hat{\lambda}_i) = P_{\text{cone}(X_i)}(\hat{\lambda}_i)$, while by the optimality conditions for problem (6.137), $P_{\text{cone}(X_i)}(\hat{\lambda}_i) = 0$ and hence $\tilde{x}_i = P_{C_i}(\hat{\lambda}_i) = 0$].

As an illustration of the algorithm, we apply it to the problem

$$\begin{aligned}&\text{minimize} \quad f(x)\\ &\text{subject to} \ \ x \in C,\end{aligned} \tag{6.138}$$

where $f : \Re^n \mapsto (-\infty,\infty]$ is a closed proper convex function and C is a closed convex cone. We reformulate this problem into our basic form (6.128) as

$$\begin{aligned}&\text{minimize} \quad f(x_1) + \delta(x_2 \mid C)\\ &\text{subject to} \ \ (x_1,x_2) \in S \overset{\text{def}}{=} \big\{(x_1,x_2) \mid x_1 = x_2\big\}.\end{aligned} \tag{6.139}$$

Note that primal and dual optimal solutions of this problem have the form (x^*, x^*) and $(\lambda^*, -\lambda^*)$, respectively, since

$$S^\perp = \{(\lambda_1, \lambda_2) \mid \lambda_1 + \lambda_2 = 0\}.$$

By transcribing our algorithm to this special case, we see that if $\hat{x}^k$ and $\hat{\lambda}^k$ are generated by $\hat{x}^k \in \arg\min_{x \in \mathrm{cone}(X^k)} f(x)$ and

$$\hat{\lambda}^k \in \partial f(\hat{x}^k), \qquad -\hat{\lambda}^k \in N_{\mathrm{cone}(X^k)}(\hat{x}^k), \tag{6.140}$$

then $(\hat{x}^k, \hat{x}^k)$ and $(\hat{\lambda}^k, -\hat{\lambda}^k)$ are optimal primal and dual solutions of the corresponding approximate problem of the algorithm [since Eq. (6.140) is the optimality condition (6.135) for problem (6.137)].

Convergence Analysis

We now discuss the convergence properties of the algorithm of this section. We will show that when the algorithm terminates, it does so at an optimal solution.

Proposition 6.4.8: (Optimality at Termination) If the algorithm of this section terminates at some iteration, the corresponding primal and dual solutions, $(\hat{x}_1, \ldots, \hat{x}_r)$ and $(\hat{\lambda}_1, \ldots, \hat{\lambda}_r)$, form a primal and dual optimal solution pair of problem (6.128).

Proof: We will verify that upon termination, the three conditions of Prop. 6.4.7 are satisfied for the original problem (6.128). From the definition of $(\hat{x}_1, \ldots, \hat{x}_r)$ and $(\hat{\lambda}_1, \ldots, \hat{\lambda}_r)$ as a primal and dual optimal solution pair of the approximate problem (6.137), we have

$$(\hat{x}_1, \ldots, \hat{x}_r) \in S, \qquad (\hat{\lambda}_1, \ldots, \hat{\lambda}_r) \in S^\perp,$$

thereby satisfying the first condition (6.134). Upon termination $P_{C_i}(\hat{\lambda}_i) = 0$, so $\hat{\lambda}_i \in C_i^*$. Also from the optimality conditions of Prop. 6.4.7, applied to the approximate problem (6.137), we have that for all $i \in I_c$, $(\hat{x}_i, \hat{\lambda}_i)$ is a dual pair with respect to $\mathrm{cone}(X_i)$ and $\mathrm{cone}(X_i)^*$, so that by Prop. 6.4.6(b), $\hat{x}_i \perp \hat{\lambda}_i$ and $\hat{x}_i \in C_i$. Thus by Prop. 6.4.6(b), $(\hat{x}_i, \hat{\lambda}_i)$ is a dual pair with respect to C_i and C_i^*, and the optimality condition (6.136) is satisfied.

Finally, we show that upon termination, we have

$$\hat{x}_i \in \partial f_i^\star(\hat{\lambda}_i), \qquad \forall\, i \in I \cup \bar{I}, \tag{6.141}$$

which by Prop. 6.4.7 will imply the desired conclusion. Since $(\hat{x}_1, \ldots, \hat{x}_r)$ and $(\hat{\lambda}_1, \ldots, \hat{\lambda}_r)$ are a primal and dual optimal solution pair of problem (6.137), Eq. (6.141) holds for all $i \in I$ (cf. Prop. 6.4.7). We will complete the proof by showing that it holds for all $i \in \bar{I}$.

Indeed, let us fix $i \in \bar{I}$ and let $\tilde{x}_i \in \partial f_i^\star(\hat{\lambda}_i)$ be the vector generated by the enlargement step upon termination, so that $\tilde{x}_i \in X_i$. Since $\bar{f}_{i,X_i}$ is an inner linearization of f_i, $\overline{f}^\star_{i,X_i}$ is an outer linearization of $f_i^\star$ of the form (6.127):

$$\overline{f}^\star_{i,X_i}(\lambda) = \max_{x \in X_i} \big\{ f^\star(\lambda_x) + (\lambda - \lambda_x)'x \big\},$$

where the vectors λ_x can be any vectors such that $x \in \partial f_i^\star(\lambda_x)$. Therefore, the fact $\tilde{x}_i \in X_i, \tilde{x}_i \in \partial f_i^\star(\hat{\lambda}_i)$ implies that

$$\overline{f}^\star_{i,X_i}(\hat{\lambda}_i) = f_i^\star(\hat{\lambda}_i),$$

which in turn shows that

$$\partial \overline{f}^\star_{i,X_i}(\hat{\lambda}_i) \subset \partial f_i^\star(\hat{\lambda}_i).$$

By Eq. (6.135), we also have $\hat{x}_i \in \partial \overline{f}^\star_{i,X_i}(\hat{\lambda}_i)$, so $\hat{x}_i \in \partial f_i^\star(\hat{\lambda}_i)$. Thus Eq. (6.141) is shown for $i \in \bar{I}$, and all the optimality conditions of Prop. 6.4.7 are satisfied for the original problem (6.128). **Q.E.D.**

The next proposition is a convergence result that is similar to the one we showed earlier, for the case of pure inner linearization.

Proposition 6.4.9: (Convergence) Consider the algorithm of this section, under the strong duality condition $-\infty < q_{opt} = f_{opt} < \infty$. Let $(\hat{x}^k, \hat{\lambda}^k)$ be the primal and dual optimal solution pair of the approximate problem (6.137), generated at the kth iteration, and let $\tilde{x}_i^k$, $i \in \bar{I}$, be the vectors generated at the corresponding enlargement step. Consider a subsequence $\{\hat{\lambda}^k\}_\mathcal{K}$ that converges to a vector $\hat{\lambda}$. Then:

(a) $\hat{\lambda}_i \in C_i^*$ for all $i \in I_c$.

(b) If the subsequences $\{\tilde{x}_i^k\}_\mathcal{K}$, $i \in \bar{I}$, are bounded, $\hat{\lambda}$ is dual optimal, and the optimal value of the inner approximation problem (6.137) converges monotonically from above to f^{opt}, while the optimal value of the dual problem of (6.137) converges monotonically from below to $-f^{opt}$.

Proof: (a) Let us fix $i \in I_c$. Since $\tilde{x}_i^k = P_{C_i}(\hat{\lambda}_i^k)$, the subsequence $\{\tilde{x}_i^k\}_\mathcal{K}$ converges to $\tilde{x}_i = P_{C_i}(\hat{\lambda}_i)$. We will show that $\tilde{x}_i = 0$, which implies that $\hat{\lambda}_i \in C_i^*$.

Denote $X_i^\infty = \cup_{k=0}^\infty X_i^k$. Since $\hat\lambda_i^k \in \text{cone}(X_i^k)^*$, we have $x_i'\hat\lambda_i^k \le 0$ for all $x_i \in X_i^k$, so that $x_i'\hat\lambda_i \le 0$ for all $x_i \in X_i^\infty$. Since $\tilde x_i$ belongs to the closure of X_i^∞, it follows that $\tilde x_i'\hat\lambda_i \le 0$. On the other hand, since $\tilde x_i = P_{C_i}(\hat\lambda_i)$, by the projection theorem for cones we have $\tilde x_i'(\hat\lambda_i - \tilde x_i) = 0$, which together with $\tilde x_i'\hat\lambda_i \le 0$, implies that $\|\tilde x_i\|^2 \le 0$, or $\tilde x_i = 0$.

(b) From the definition of $\overline{f}^\star_{i,X_i^k}$ [cf. Eq. (6.127)], we have for all $i \in \bar I$ and $k, \ell \in \mathcal{K}$ with $\ell < k$,

$$f_i^\star(\hat\lambda_i^\ell) + (\hat\lambda_i^k - \hat\lambda_i^\ell)'\tilde x_i^\ell \le \overline{f}^\star_{i,X_i^k}(\hat\lambda_i^k).$$

Using this relation and the optimality of $\hat\lambda^k$ for the kth approximate dual problem to write for all $k, \ell \in \mathcal{K}$ with $\ell < k$

$$\begin{aligned}\sum_{i\in I} f_i^\star(\hat\lambda_i^k) + \sum_{i\in\bar I}\big(f_i^\star(\hat\lambda_i^\ell) + (\hat\lambda_i^k - \hat\lambda_i^\ell)'\tilde x_i^\ell\big) &\le \sum_{i\in I} f_i^\star(\hat\lambda_i^k) + \sum_{i\in\bar I}\overline{f}^\star_{i,X_i^k}(\hat\lambda_i^k)\\ &\le \sum_{i\in I} f_i^\star(\lambda_i) + \sum_{i\in\bar I}\overline{f}^\star_{i,X_i^k}(\lambda_i),\end{aligned}$$

for all $(\lambda_1,\ldots,\lambda_m)$ such that there exist $\lambda_i \in \text{cone}(X_i^k)^*$, $i \in I_c$, with $(\lambda_1,\ldots,\lambda_r) \in S$. Since $C_i^* \subset \text{cone}(X_i^k)^*$, it follows that

$$\begin{aligned}\sum_{i\in I} f_i^\star(\hat\lambda_i^k) + \sum_{i\in\bar I}\big(f_i^\star(\hat\lambda_i^\ell) + (\hat\lambda_i^k - \hat\lambda_i^\ell)'\tilde x_i^\ell\big) &\le \sum_{i\in I} f_i^\star(\lambda_i) + \sum_{i\in\bar I}\overline{f}^\star_{i,X_i^k}(\lambda_i)\\ &\le \sum_{i=1}^m f_i^\star(\lambda_i),\end{aligned}\tag{6.142}$$

for all $(\lambda_1,\ldots,\lambda_m)$ such that there exist $\lambda_i \in C_i^*$, $i \in I_c$, with $(\lambda_1,\ldots,\lambda_r) \in S$, where the last inequality holds since $\overline{f}^\star_{i,X_i^k}$ is an outer linearization of $f_i^\star$.

By taking limit inferior in Eq. (6.142), as $k, \ell \to \infty$ with $k, \ell \in \mathcal{K}$, and by using the lower semicontinuity of $f_i^\star$, which implies that

$$f_i^\star(\hat\lambda_i) \le \liminf_{\ell\to\infty,\,\ell\in\mathcal{K}} f_i^\star(\hat\lambda_i^\ell), \qquad i \in I_c,$$

we obtain

$$\sum_{i=1}^m f_i^\star(\hat\lambda_i) \le \sum_{i=1}^m f_i^\star(\lambda_i) \tag{6.143}$$

for all $(\lambda_1,\ldots,\lambda_m)$ such that there exist $\lambda_i \in C_i^*$, $i \in I_c$, with $(\lambda_1,\ldots,\lambda_r) \in S$. We have $\hat\lambda \in S$ and $\hat\lambda_i \in C_i^*$ for all $i \in I_c$, from part (a). Thus Eq. (6.143) implies that $\hat\lambda$ is dual optimal. The sequence of optimal values of the dual approximation problem [the dual of problem (6.137)] is monotonically nondecreasing (since the outer approximation is monotonically refined) and converges to $-f^{opt}$ since $\hat{\lambda}$ is dual optimal. This sequence is the opposite of the sequence of optimal values of the primal approximation problem (6.137), so the latter sequence is monotonically nonincreasing and converges to f^{opt}. **Q.E.D.**

We have dealt so far with cases where the constraint set involves the intersection of compact sets and cones, which can be inner linearized separately. However, we can also deal with vector sums of compact sets and cones, which again can be linearized separately.† In particular the problem

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

where X is a compact set and C is a closed convex cone, can be written as

$$\begin{aligned} &\text{minimize} \quad f(x_1) + \delta(x_2|X) + \delta(x_3|C) \\ &\text{subject to} \quad x_1 = x_2 + x_3, \end{aligned}$$

which is of the form (6.128) with $S = \big\{(x_1, x_2, x_3) \mid x_1 = x_2 + x_3\big\}$.

We finally note that for efficient implementation, the projection on the cones C_i required at the enlargement step, should be done with an algorithm that takes advantage of the special structure of these cones. In the case of a polyhedral cone, the problem of projection is a quadratic programming problem. In the case of special nonpolyhedral cones, such as the second order and semidefinite cones, there are efficient specialized algorithms for which we refer to the literature (see e.g., Fukushima, Luo, and Tseng [FLT02]).

6.5 PROXIMAL METHODS

In this section we introduce a general approximation approach for "regularizing" (i.e., improving the structure) of convex optimization problems and algorithms. As one motivation, let us recall that one of the drawbacks of the cutting plane method for minimizing a convex function $f : \Re^n \mapsto \Re$ over a convex set X 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 introduce a quadratic term $p_k(x)$, called "proximal term," that penalizes

† Note that by the Minkowski-Weyl Theorem, any polyhedral set can be decomposed as the vector sum of the convex hull of a finite number of points and a cone generated by a finite number of points; see Prop. 2.3.2.

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

$$x_{k+1} \in \arg\min_{x\in X}\big\{F_k(x)+p_k(x)\big\}, \tag{6.144}$$

where similar to the cutting plane method,

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

and

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

where c_k is an adjustable positive scalar parameter (cf. Fig. 6.5.1). The method for choosing the "proximal center" 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).

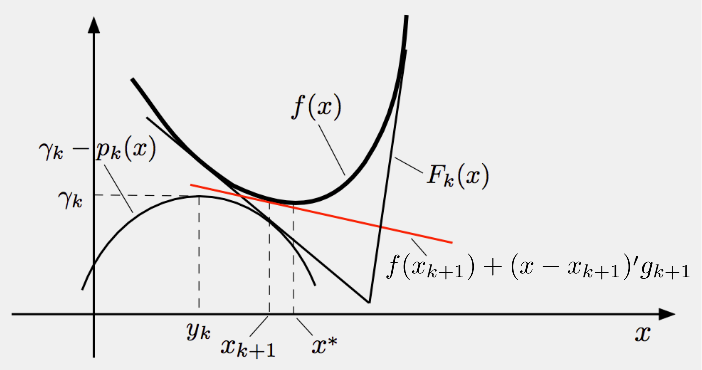

Figure 6.5.1. Using a proximal term to reduce the effect of instability in the cutting plane method. The point x_{k+1} is the one at which the graph of the negative proximal term $-p_k(x)$, raised by some amount γ_k, just touches the graph of F_k. Then x_{k+1} tends to be closer tor y_k, with the distance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

We can view iteration (6.144) as an approximate version of a general algorithm for minimizing a convex function. This algorithm embodies ideas of great significance in optimization, and will be discussed in the next section. We will return to cutting plane approximations that use subgradients in Sections 6.5.2 and 6.5.3.

6.5.1 Proximal Algorithm

Consider the minimization of a closed proper convex function $f : \Re^n \mapsto (-\infty, \infty]$, let f^* denote the optimal value

$$f^* = \inf_{x \in \Re^n} f(x),$$

and let X^* denote the set of minima of f (which could be empty),

$$X^* = \arg\min_{x \in \Re^n} f(x).$$

We may view this as a general constrained optimization problem where the constraint is $x \in \mathrm{dom}(f)$. We consider the algorithm

$$x_{k+1} \in \arg\min_{x \in \Re^n} \left\{ f(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}, \tag{6.145}$$

where x_0 is an arbitrary starting point and c_k is a positive scalar parameter. This is known as the *proximal algorithm*. Its chief utility is regularization: the quadratic term $\|x - x_k\|^2$ makes the function that is minimized in the iteration (6.145) strictly convex and coercive. This guarantees that x_{k+1} is well-defined as the unique point attaining the minimum in Eq. (6.145); see Fig. 6.5.2.

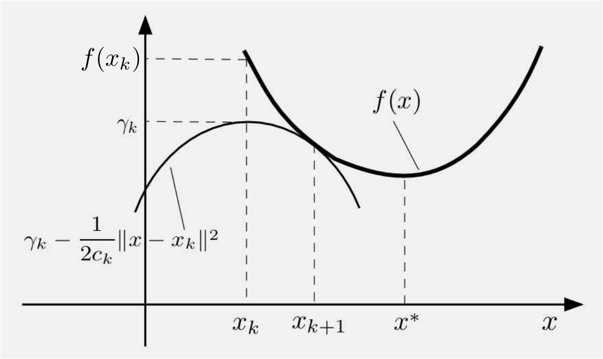

Figure 6.5.2. Geometric view of the proximal algorithm (6.145). The minimum of $f(x) + \frac{1}{2c_k}\|x - x_k\|^2$ is attained at the unique point x_{k+1} at which the graph of the quadratic function $-\frac{1}{2c_k}\|x - x_k\|^2$, raised by the amount

$$\gamma_k = f(x_{k+1}) + \frac{1}{2c_k}\|x_{k+1} - x_k\|^2,$$

just touches the graph of f.

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

$$\frac{x_k - x_{k+1}}{c_k} \in \partial f(x_{k+1}). \tag{6.146}$$

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 $\partial f(x_k) \approx \partial f(x_{k+1})$. Indeed, the proximal algorithm bears a connection to subgradient methods, as well as to polyhedral approximation methods, which will be become apparent in what follows, as variants of the proximal minimization idea are developed. For the moment we focus on the convergence properties of the basic method.

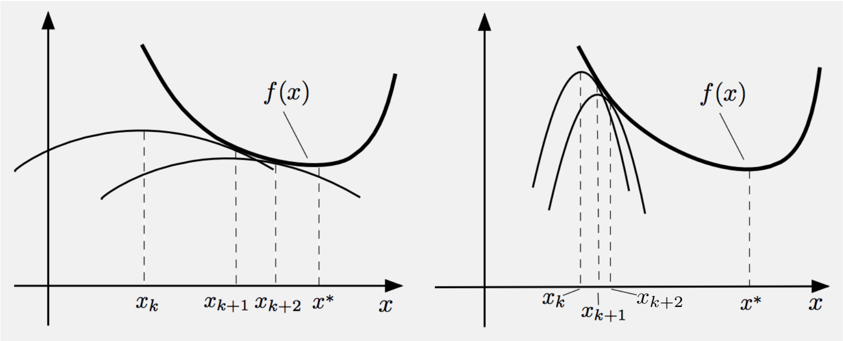

Figure 6.5.3. Illustration of the role of the parameter c_k in the convergence process of the proximal algorithm. In the figure on the left, c_k is large, the graph of the quadratic term is "blunt," and the method makes fast progress toward the optimal solution set X^*. In the figure on the right, c_k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

Convergence

The proximal 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 algorithm (6.145). 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 $x = x_k$,

$$f(x_{k+1}) + \frac{1}{2c_k}\|x_{k+1} - x_k\|^2 \leq f(x_k), \qquad \forall\ k. \tag{6.147}$$

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

For any $y \in \Re^n$, 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 (6.146),

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

Combining these two relations, we obtain

$$\begin{aligned}\|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2c_k\big(f(x_{k+1}) - f(y)\big) - \|x_{k+1} - x_k\|^2 \\ &\leq \|x_k - y\|^2 - 2c_k\big(f(x_{k+1}) - f(y)\big),\end{aligned} \tag{6.148}$$

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

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

so that

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

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

$$2\sum_{k=0}^{\infty} c_k\big(f(x_{k+1}) - f(y)\big) \le \|x_0 - y\|^2, \qquad \forall\ y \in \Re^n. \tag{6.149}$$

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

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

Then in view of the assumption $\sum_{k=0}^{\infty} c_k = \infty$, Eq. (6.149) 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. (6.148) with $y = x^*$, we have

$$\|x_{k+1} - x^*\|^2 \le \|x_k - x^*\|^2 - 2c_k\big(f(x_{k+1}) - f(x^*)\big), \qquad k = 0, 1, \ldots. \tag{6.150}$$

Thus $\|x_k - x^*\|^2$ is monotonically nonincreasing, so $\{x_k\}$ is bounded, and each of its limit points must belong to X^*, since $\big\{f(x_k)\big\}$ monotonically decreases to f^* and f is closed. Also, by Eq. (6.150), the distance of x_k to each limit point is monotonically nonincreasing, so $\{x_k\}$ converges 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).

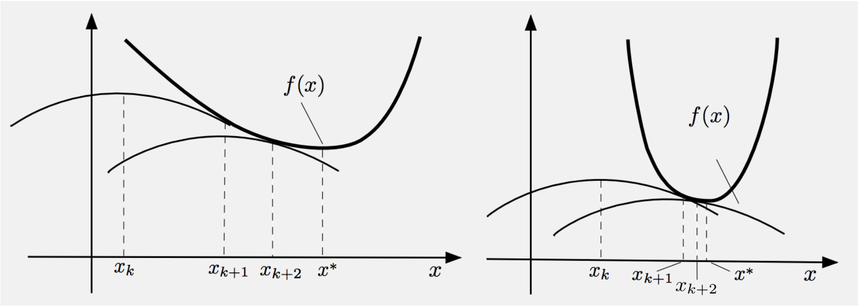

Figure 6.5.4. Illustration of the role of the growth properties of f near X^* in the convergence rate of the proximal algorithm. In the figure on the left, f grows slowly and the convergence is slow. In the figure on the right, f grows fast and the convergence is fast.

Proposition 6.5.2: (Rate of Convergence) Let $\{x_k\}$ be a sequence generated by the proximal algorithm (6.145), 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

$$f^* + \beta\big(d(x)\big)^{\alpha} \leq f(x), \qquad \forall\, x \in \Re^n \;\text{ with } d(x) \leq \delta, \tag{6.151}$$

where

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

(a) For all k sufficiently large, we have

$$d(x_{k+1}) + \beta c_k \big(d(x_{k+1})\big)^{\alpha-1} \leq d(x_k). \tag{6.152}$$

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

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

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

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

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

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

Proof: (a) Since Eq. (6.152) clearly holds when $x_{k+1} \in X^*$, we assume that $x_{k+1} \notin X^*$ and we denote by $\hat{x}_{k+1}$ the projection of x_{k+1} on X^*. From the subgradient relation (6.146), 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 from Eq. (6.151) that

$$f^* + \beta\big(d(x_{k+1})\big)^{\alpha} \leq f(x_{k+1}),$$

for k sufficiently large. Combining the preceding two relations, we obtain

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

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

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

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/App. B), to obtain

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

Dividing with $\|x_{k+1} - \hat{x}_{k+1}\|$ (which is nonzero), Eq. (6.152) follows.

(b) From Eq. (6.152) and the fact $\alpha < 2$, the desired relation follows.

(c) For $\alpha = 2$, Eq. (6.152) 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 α in Eq. (6.151) 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 $c_k \to \infty$. 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 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.

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

$$f^* + \beta d(x) \leq f(x), \qquad \forall\ x \in \Re^n, \tag{6.153}$$

where $d(x) = \min_{x^* \in X^*} \|x - x^*\|$. Then if $\sum_{k=0}^\infty c_k = \infty$, the proximal algorithm (6.145) converges to X^* finitely [that is, there exists $\overline{k} > 0$ such that $x_k \in X^*$ for all $k \geq \overline{k}$]. Furthermore, if $c_0 \geq d(x_0)/\beta$, the algorithm converges in a single iteration (i.e., $x_1 \in X^*$).

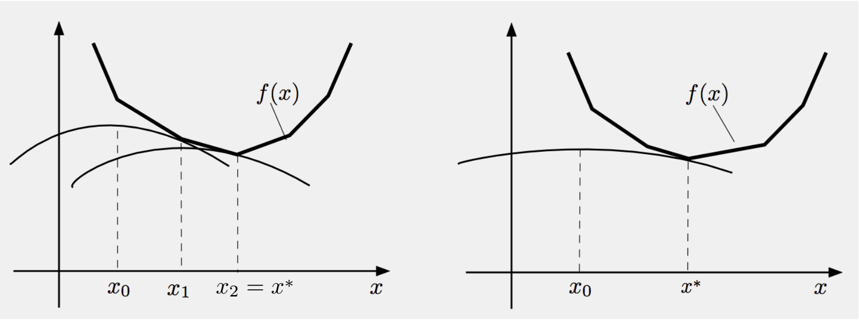

Figure 6.5.5. Finite convergence of the proximal algorithm when $f(x)$ grows at a linear rate near the optimal solution set X^* (e.g., f is polyhedral). In the figure on the right, we have convergence in a single iteration for a large enough value of c.

Proof: The assumption (6.151) of Prop. 6.5.2 holds with $\alpha = 1$ and all $\delta > 0$, so Eq. (6.152) becomes

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

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 (6.153) 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).

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

$$\tilde{f}(x) = f^* + \beta d(x) + \frac{1}{2c_0}\|x - x_0\|^2. \tag{6.154}$$

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

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

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. (6.153) and (6.154), we have

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

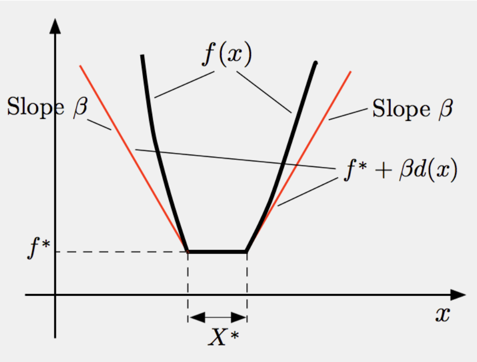

Figure 6.5.6. Illustration of the condition

$$f^* + \beta d(x) \le f(x), \qquad \forall\, x \in \Re^n,$$

[cf. Eq. (6.153)].

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 algorithm.

From the preceding discussion and graphical illustrations, it can be seen that the rate of convergence of the proximal algorithm is improved by choosing large values of c. However, the corresponding regularization effect is reduced as c is increased, and this may adversely affect the proximal minimizations. In practice, it is often suggested to start with a moderate value of c, and gradually increase this value in subsequent proximal minimizations. How fast c can increase depends on the method used to solve the corresponding proximal minimization problems. If a fast Newton-like method is used, a fast rate of increase of c (say by a factor 5-10) may be possible, resulting in very few proximal minimizations. If instead a relatively slow first order method is used, it may be best to keep c constant at a moderate value, which is usually determined by trial and error.

Gradient and Subgradient Interpretations

We have already interpreted the proximal algorithm as an approximate subgradient method [cf. Eq. (6.146)]. For another interesting interpretation, consider the function

$$\phi_c(z) = \inf_{x \in \Re^n} \left\{ f(x) + \frac{1}{2c}\|x - z\|^2 \right\}, \tag{6.155}$$

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

$$\inf_{x \in \Re^n} f(x) \le \phi_c(z) \le f(z), \qquad \forall\, z \in \Re^n,$$

from which it follows that the set of minima of f and ϕ_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 ϕ_c is a convex differentiable function, and calculates its gradient.

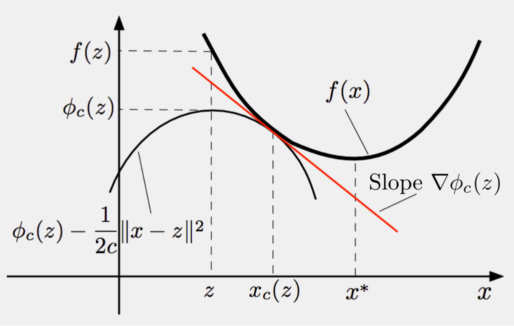

Figure 6.5.7. Illustration of the function

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

We have $\phi_c(z) \leq f(z)$ for all $z \in \Re^n$, and at the set of minima of f, ϕ_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 ϕ_c in the figure.

Proposition 6.5.4: The function ϕ_c of Eq. (6.155) is convex and differentiable, and we have

$$\nabla\phi_c(z) = \frac{z - x_c(z)}{c} \qquad \forall\ z \in \Re^n, \tag{6.156}$$

where $x_c(z)$ is the unique minimizer in Eq. (6.155).

Proof: We first note that ϕ_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, ϕ_c is real-valued, since the infimum in Eq. (6.155) is attained.

Let us fix z, and for notational simplicity, denote $\overline{z} = x_c(z)$. To show that ϕ_c is differentiable with the given form of gradient, we note that by the optimality condition of Prop. 5.4.7, we have $v \in \partial\phi_c(z)$ if and only if z

attains the minimum over $y \in \Re^n$ of

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

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

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

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

$$0 \in \partial f(\overline{z}) + \frac{\overline{z} - z}{c}, \qquad v = \frac{z - \overline{z}}{c}. \tag{6.157}$$

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

$$\frac{1}{2c}\|x - y\|^2 - v'y,$$

and by writing

$$\partial F(x, y) = \big\{(g, 0) \mid g \in \partial f(x)\big\} + \left\{ \frac{x - y}{c}, \frac{y - x}{c} - v \right\};$$

cf. Prop. 5.4.6.] The right side of Eq. (6.157) uniquely defines v, so that v is the unique subgradient of ϕ_c at z, and it has the form $v = (z - \overline{z})/c$, as required by Eq. (6.156). From the left side of Eq. (6.157), we also see that $v = \nabla\phi_c(z) \in \partial f\big(x_c(z)\big)$. **Q.E.D.**

Using the gradient formula (6.156), we see that the proximal 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 ϕ_{c_k} (which has the same minima as f, as noted earlier). This interpretation provides insight into the 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 $f : \Re^n \mapsto \Re$, over a closed convex set X, by using the proximal 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 future iterations. Thus one may perform the proximal minimizations approximately, and update the proximal center 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 iteration, aimed at minimizing the current polyhedral approximation to f given by [cf. Eq. (6.90)]

$$F_k(x) = \max\big\{f(x_0) + (x - x_0)'g_0, \ldots, f(x_k) + (x - x_k)'g_k\big\}, \tag{6.158}$$

i.e.,

$$x_{k+1} \in \arg\min_{x \in X} \left\{ F_k(x) + \frac{1}{2c_k}\|x - x_k\|^2 \right\}, \tag{6.159}$$

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. (6.158) and (6.159) imply that

$$f(x_k) = F_k(x_k) \le F_k(x) + \frac{1}{2c_k}\|x - x_k\|^2 \le f(x) + \frac{1}{2c_k}\|x - x_k\|^2, \qquad \forall\, x \in X,$$

so x_k is a point where the proximal 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 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., $F_k = F_{\overline{k}}$ for all $k > \bar{k}$. Thus, for $k \geq \overline{k}$, the method will become the proximal algorithm for minimizing $F_{\overline{k}}$, so by Prop. 6.5.3, it will terminate with a point $\overline{x}$ that minimizes $F_{\overline{k}}$ 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. (6.159) 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 $p_k(x)$:

$$x_{k+1} \in \arg\min_{x \in X} \big\{ F_k(x) + p_k(x) \big\}. \tag{6.160}$$

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

$$F_k(x) = \max\big\{ f(x_0) + (x - x_0)'g_0, \ldots, f(x_k) + (x - x_k)'g_k \big\}, \tag{6.161}$$

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 β is a fixed scalar with $\beta \in (0, 1)$, and

$$\delta_k = f(y_k) - \big(F_k(x_{k+1}) + p_k(x_{k+1}) \big).$$

Thus,

$$y_{k+1} = \begin{cases} x_{k+1} & \text{if } f(y_k) - f(x_{k+1}) \geq \beta \delta_k, \\ y_k & \text{if } f(y_k) - f(x_{k+1}) < \beta \delta_k, \end{cases} \tag{6.162}$$

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. (6.160) and (6.161) 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 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

$$0 \leq f(y_k) - \big(F_k(x_{k+1}) + p_k(x_{k+1})\big) = \delta_k,$$

with equality only if $x_{k+1} = y_k$, i.e., when the algorithm terminates.

The scalar δ_k is illustrated in Fig. 6.5.8. Since $f(y_k) = F_k(y_k)$ [cf. Eq. (6.161)], δ_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 β of δ_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. (6.162)], but improves the cutting plane approximation by adding the new plane corresponding to x_{k+1}. Otherwise, it performs a serious step, with the guarantee of true cost improvement afforded by the test (6.162).

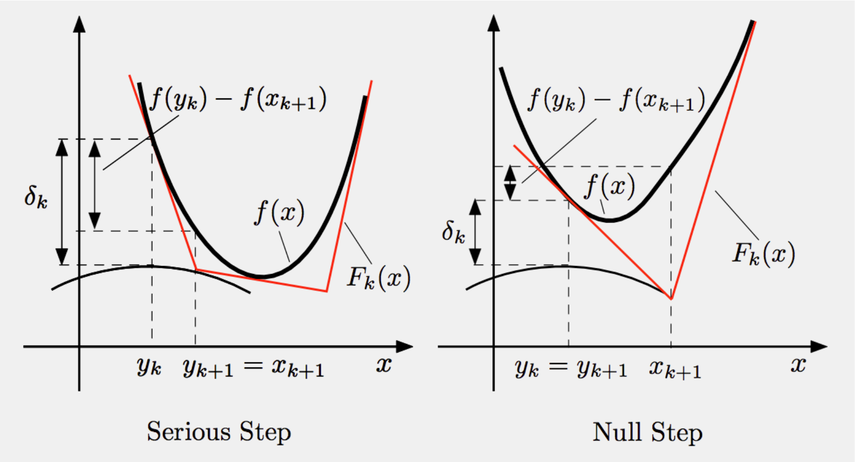

Figure 6.5.8. Illustration of the test (6.162) for a serious or a null step in the bundle method. It is based on

$$\delta_k = f(y_k) - \big(F_k(x_{k+1}) + p_k(x_{k+1})\big),$$

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 β of the reduction δ_k in proximal cost.

The convergence analysis of the bundle method just presented follows the corresponding arguments for the cutting plane and the proximal 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 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

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

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., $F_k = F_{\overline{k}}$ 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 $k > \bar{k}$,

$$\begin{aligned} f(y_k) - f(x_{k+1}) &= f(y_k) - F_k(x_{k+1}) \\ &= f(y_k) - \big(F_k(x_{k+1}) + p_k(x_{k+1})\big) + p_k(x_{k+1}) \\ &= \delta_k + p_k(x_{k+1}) \\ &\geq \beta \delta_k. \end{aligned}$$

Therefore, according to Eq. (6.162), 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 (6.162) 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 $y_{k+1} = x_{x+1}$, 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

$$F_k(y_{k+1}) - \big(f(x_i) + (y_{k+1} - x_i)'g_i\big)$$

is 0 or close to 0, respectively. The essential validity of the method is maintained, by virtue of the fact that $\big\{f(y_k)\big\}$ 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 $\big(x_{k+1}, F_k(x_{k+1})\big)$ and separates the epigraphs of the functions $F_k(x)$ and $\gamma_k - \frac{1}{2c_k}\|x - y_k\|^2$, where

$$\gamma_k = F_k(x_{k+1}) + \frac{1}{2c_k}\|x_{k+1} - y_k\|^2,$$

(see Fig. 6.5.9). This is the cutting plane

$$F_k(x_{k+1}) + \hat{g}'_k(x - x_{k+1}), \tag{6.163}$$

where $\hat{g}_k$ is given by

$$\hat{g}_k = \frac{y_k - x_{k+1}}{c_k}. \tag{6.164}$$

The next iteration will then be performed with just two cutting planes: the one just given in Eq. (6.163) and a new one obtained from x_{k+1},

$$f(x_{k+1}) + g'_{k+1}(x - x_{k+1}),$$

where $g_{k+1} \in \partial f(x_{k+1})$.

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 $g_0, \ldots, g_k$. 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.

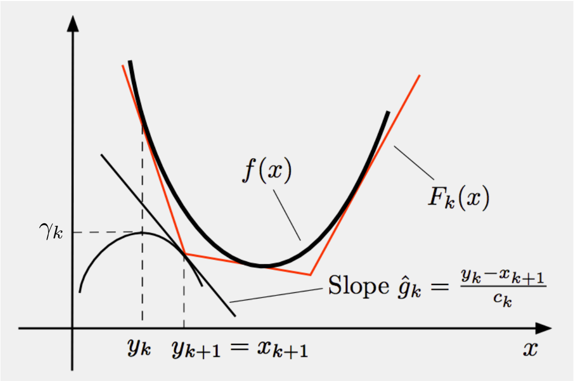

Figure 6.5.9. Illustration of the cutting plane

$$F_k(x_{k+1}) + \hat{g}_k'(x - x_{k+1}),$$

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}.

6.6 DUAL PROXIMAL ALGORITHMS

In this section, we will develop an equivalent dual implementation of the proximal algorithm, based on Fenchel duality. We will then apply it to the cutting plane/bundle setting, taking advantage of the duality between the simplicial decomposition and cutting plane methods, developed in Section 6.4.3. We will also apply it in a special way to obtain a popular constrained optimization algorithm, the Augmented Lagrangian method.

We recall the proximal algorithm of Section 6.5.1:

$$x_{k+1} = \arg\min_{x \in \Re^n} \left\{ f(x) + \frac{1}{2c_k}\|x - x_k\|^2 \right\}, \tag{6.165}$$

where $f : \Re^n \mapsto (-\infty, \infty]$, 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 note that the minimization above is in a form suitable for application of Fenchel duality (cf. Section 5.3.5) with the identifications

$$f_1(x) = f(x), \qquad f_2(x) = \frac{1}{2c_k}\|x - x_k\|^2.$$

We can write the Fenchel dual problem as

$$\begin{array}{ll} \text{minimize} & f_1^\star(y) + f_2^\star(-y) \\ \text{subject to} & y \in \Re^n, \end{array} \tag{6.166}$$

where $f_1^\star$ and $f_2^\star$ are the conjugate functions of f_1 and f_2, respectively. We have

$$f_2^\star(y) = \sup_{x\in\Re^n}\big\{x'y - f_2(x)\big\} = \sup_{x\in\Re^n}\left\{x'y - \frac{1}{2c_k}\|x-x_k\|^2\right\} = x_k'y + \frac{c_k}{2}\|y\|^2,$$

where the last equality follows by noting that the supremum over x is attained at $x = x_k + c_k y$. Denoting by $f^\star$ the conjugate of f,

$$f_1^\star(y) = f^\star(y) = \sup_{x\in\Re^n}\big\{x'y - f(x)\big\},$$

and substituting into Eq. (6.166), we see that the dual problem (6.166) can be written as

$$\begin{aligned}&\text{minimize} \quad f^\star(y) - x_k'y + \frac{c_k}{2}\|y\|^2\\ &\text{subject to} \quad y\in\Re^n.\end{aligned} \tag{6.167}$$

We also note that since f_2 and $f_2^\star$ are real-valued, the relative interior condition of the Fenchel duality theorem [Prop. 6.1.5(a)] is satisfied, so there is no duality gap. In fact both primal and dual problems have a unique solution, since they involve a closed, strictly convex, and coercive cost function.

Let y_{k+1} be the unique solution of problem (6.167). Then y_{k+1} together with x_{k+1} satisfy the necessary and sufficient optimality conditions of Prop. 6.1.5(b). When applied to the primal problem, these conditions can be written as

$$x_{k+1} \in \arg\max_{x\in\Re^n}\big\{x'y_{k+1} - f(x)\big\},$$

$$x_{k+1} \in \arg\min_{x\in\Re^n}\big\{x'y_{k+1} - f_2(x)\big\} = \arg\min_{x\in\Re^n}\left\{x'y_{k+1} + \frac{1}{2c_k}\|x-x_k\|^2\right\},$$

or equivalently,

$$y_{k+1}\in\partial f(x_{k+1}), \qquad x_{k+1} = x_k - c_k y_{k+1}; \tag{6.168}$$

see Fig. 6.6.1. Similarly, when applied to the dual problem, the necessary and sufficient optimality conditions of Prop. 6.1.5(b) can be written as

$$x_{k+1}\in\partial f^\star(y_{k+1}), \qquad y_{k+1} = \frac{x_k - x_{k+1}}{c_k}. \tag{6.169}$$

We thus obtain a dual implementation of the proximal algorithm. In this algorithm, instead of solving the Fenchel primal problem involved in the proximal iteration (6.165), we first solve the Fenchel dual problem (6.167), and then obtain the optimal primal Fenchel solution x_{k+1} using the optimality condition (6.168):

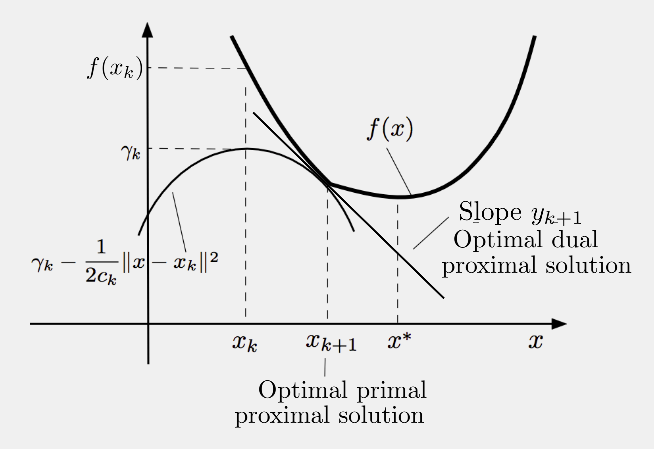

Figure 6.6.1. Illustration of the optimality conditions (6.168).

Dual Implementation of the Proximal Algorithm:

Find

$$y_{k+1} = \arg\min_{y\in\Re^n}\left\{f^\star(y) - x_k'y + \frac{c_k}{2}\|y\|^2\right\}, \tag{6.170}$$

and then set

$$x_{k+1} = x_k - c_k y_{k+1}. \tag{6.171}$$

The dual algorithm is illustrated in Fig. 6.6.2. Note that as x_k converges to a minimum x^* of f, y_k converges to 0. Thus the dual iteration (6.170) does not aim to minimize $f^\star$, but rather to find a subgradient of $f^\star$ at 0, which [by Prop. 5.4.4(a)] minimizes f. In particular, we have $y_k \in \partial f(x_k)$, $x_k \in \partial f^\star(y_k)$ [cf. Eqs. (6.168) and (6.169)], and as y_k converges to 0 and x_k converges to a minimum x^* of f, we have $0 \in \partial f(x^*)$ and $x^* \in \partial f^\star(0)$.

The primal and dual implementations of the proximal algorithm 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 (6.165) and (6.170) is easier, i.e., whether f or its conjugate $f^\star$ has more convenient structure.

6.6.1 Proximal Inner Linearization Methods

In Section 6.5.2 we saw that the proximal algorithm can be combined with outer linearization to yield the proximal cutting plane algorithm. In this section we use a dual combination, involving the dual proximal algorithm

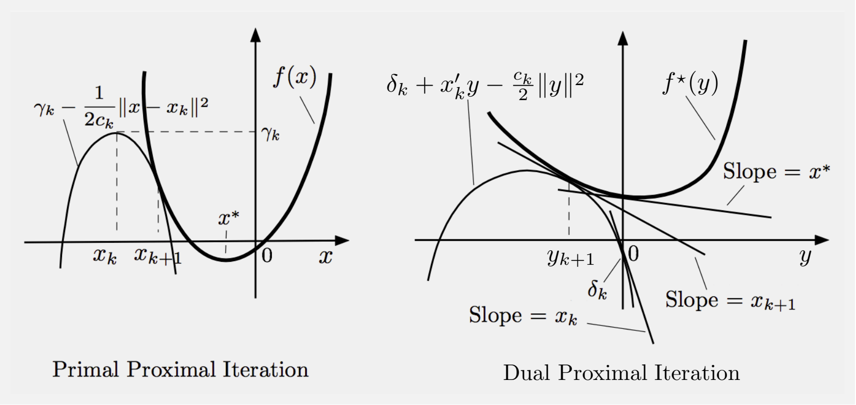

Figure 6.6.2. Illustration of primal and dual proximal algorithms. The primal algorithm aims to find x^*, a minimum of f. The dual algorithm aims to find x^* as a subgradient of $f^\star$ at 0, i.e., it aims to solve the (generalized) equation $x^* \in \partial f^\star(0)$ (cf. Prop. 5.4.4/COT).

(6.170)-(6.171) and inner linearization (the dual of outer linearization). This yields a new method, which is connected to the proximal cutting plane algorithm of Section 6.5.2 by Fenchel duality (see Fig. 6.6.3).

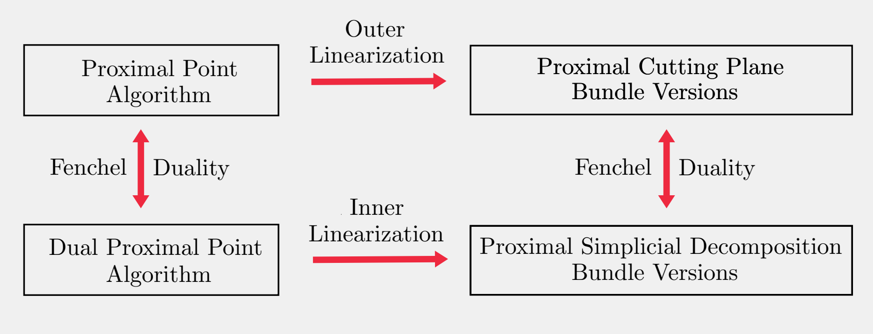
Proximal Point Algorithm
Outer Linearization
Proximal Cutting Plane Bundle Versions
Fenchel Duality
Fenchel Duality
Dual Proximal Point Algorithm
Inner Linearization
Proximal Simplicial Decomposition Bundle Versions

Figure 6.6.3. Relations of the proximal and proximal cutting plane methods, and their duals. The dual algorithms are obtained by application of the Fenchel duality theorem.

Let us recall the proximal cutting plane method applied to minimizing a real-valued convex function $f : \Re^n \mapsto \Re$, 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 δ_X is the indicator function of X. Thus,

$$x_{k+1} \in \arg\min_{x \in \Re^n} \left\{ F_k(x) + \frac{1}{2c_k} \|x - x_k\|^2 \right\}, \tag{6.172}$$

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 preceding subsection, we may use Fenchel duality to implement the proximal minimization (6.172) in terms of conjugate functions [cf. Eq. (6.170)]. The Fenchel dual of the minimization involved in the proximal iteration (6.172) can be written as

$$\begin{aligned} &\text{minimize} \quad F_k^\star(y) - x_k' y + \frac{c_k}{2}\|y\|^2 \\ &\text{subject to} \quad y \in \Re^n, \end{aligned} \tag{6.173}$$

where $F_k^\star$ is the conjugate of F_k. Once y_{k+1}, the unique minimizer in the dual proximal iteration (6.173), is computed, x_k is updated via

$$x_{k+1} = x_k - c_k y_{k+1}$$

[cf. Eq. (6.171)]. 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

$$f(x_{k+1}) = \sup_{y \in \Re^n} \bigl\{ x_{k+1}' y - f^\star(y) \bigr\},$$

where $f^\star$ is the conjugate function of f:

$$g_{k+1} \in \arg\max_{y \in \Re^n} \bigl\{ x_{k+1}' y - f^\star(y) \bigr\}.$$

We will now discuss the details of the preceding computations, assuming for simplicity that there are no constraints, i.e., $X = \Re^n$. According to Section 6.4.3, $F_k^\star$ is a piecewise linear, inner approximation of $f^\star$. In particular, $F_k^\star$ is a piecewise linear (inner) approximation of $f^\star$ with domain

$$\text{dom}(F_k^\star) = \text{conv}\bigl(\{g_0, \ldots, g_k\}\bigr),$$

and "break points" at g_i, $i = 0, \ldots, k$, with values equal to the corresponding values of $f^\star$.

Let us consider the dual proximal optimization of Eq. (6.170). It takes the form

$$\begin{aligned} &\text{minimize} \quad \sum_{i=0}^k \alpha_i \bigl(f^\star(g_i) - x_k' g_i \bigr) + \frac{c_k}{2} \left\| \sum_{i=0}^k \alpha_i g_i \right\|^2 \\ &\text{subject to} \quad \sum_{i=0}^k \alpha_i = 1, \quad \alpha_i \geq 0, \ i = 0, \ldots, k. \end{aligned} \tag{6.174}$$

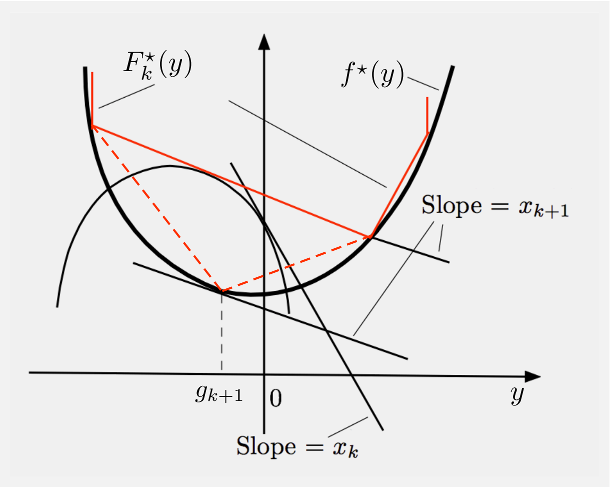

Figure 6.6.4. Illustration of an iteration of the proximal inner linearization algorithm. The proximal minimization determines the "slope" x_{k+1} of $F_k^\star$, which then determines the next subgradient/break point g_{k+1} via the maximization

$$g_{k+1} \in \arg\max_{y\in\Re^n} \bigl\{x_{k+1}'y - f^\star(y)\bigr\},$$

i.e., g_{k+1} is a point at which x_{k+1} is a subgradient of $f^\star$.

If $(\alpha_0^k, \ldots, \alpha_k^k)$ attains the minimum, we have

$$y_{k+1} = \sum_{i=0}^{k} \alpha_i^k g_i, \qquad x_{k+1} = x_k - c_k \sum_{i=0}^{k} \alpha_i^k g_i. \tag{6.175}$$

The next subgradient g_{k+1} may be obtained from the maximization

$$g_{k+1} \in \arg\max_{y\in\Re^n} \bigl\{x_{k+1}'y - f^\star(y)\bigr\}. \tag{6.176}$$

As Fig. 6.6.4 indicates, g_{k+1} provides a new break point and an improved inner approximation to $f^\star$ [equivalently, $\bigl(g_{k+1}, f^\star(g_{k+1})\bigr)$ is a new extreme point added to the Minkowski-Weyl representation of epi$(F_k^\star)$].

We refer to the algorithm defined by Eqs. (6.174)-(6.176), as the *proximal inner linearization algorithm*. Note that all the computations of the algorithm involve the conjugate $f^\star$ and not f. Thus, if $f^\star$ is more convenient to work with than f, the proximal inner linearization algorithm is preferable to the proximal cutting plane algorithm. The maximization (6.176) 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. (6.175), and null steps, where we leave x_k unchanged, but simply add the extreme point $\big(g_{k+1}, f^\star(g_{k+1})\big)$ to the current inner approximation of $f^\star$.

6.6.2 Augmented Lagrangian Methods

In this section, we will develop an equivalent dual implementation of the proximal algorithm, based on Fenchel duality. We will then apply it to the cutting plane/bundle setting, taking advantage of the duality between the simplicial decomposition and cutting plane methods, developed in Section 6.4.3. We will also apply it in a special way to obtain a popular constrained optimization algorithm, the Augmented Lagrangian method.

Consider the constrained minimization problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in X, \quad Ex = d, \end{aligned} \tag{6.177}$$

where $f : \Re^n \mapsto \Re$ is a convex function, X is a closed convex set, E is an $m \times n$ matrix, and $d \in \Re^m$.†

Consider also the corresponding primal and dual functions

$$p(v) = \inf_{x \in X,\, Ex - d = v} f(x), \qquad q(\lambda) = \inf_{x \in X} L(x, \lambda),$$

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 algorithm to the dual problem of maximizing q. It has the form

$$\lambda_{k+1} = \arg \max_{\lambda \in \Re^m} \left\{ q(\lambda) - \frac{1}{2c_k} \|\lambda - \lambda_k\|^2 \right\}.$$

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 algorithm has the form

$$v_{k+1} = \arg \min_{v \in \Re^m} \left\{ p(v) + {\lambda_k}' v + \frac{c_k}{2} \|v\|^2 \right\}; \tag{6.178}$$

† 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 $a_j'x \le b_j$ can be converted to an equality constraint $a_j'x + z_j = b_j$ by using a slack variable constraint $z_j \ge 0$, which can be absorbed into the set X. The book [Ber82] is a comprehensive reference on Augmented Lagrangian methods.

see Fig. 6.6.5. To implement this algorithm, we use the definition of p to write the above minimization as

$$\begin{aligned}
&\min_{v\in\Re^m}\left\{\inf_{x\in X,\,Ex-d=v}\{f(x)\}+{\lambda_k}'v+\frac{c_k}{2}\|v\|^2\right\}\\
&\qquad=\min_{v\in\Re^m}\inf_{x\in X,\,Ex-d=v}\left\{f(x)+\lambda'(Ex-d)+\frac{c}{2}\|Ex-d\|^2\right\}\\
&\qquad=\inf_{x\in X}\left\{f(x)+\lambda'(Ex-d)+\frac{c}{2}\|Ex-d\|^2\right\}\\
&\qquad=\inf_{x\in X}L_{c_k}(x,\lambda_k),
\end{aligned}\tag{6.179}$$

where for any $c>0$, L_c is the Augmented Lagrangian function

$$L_c(x,\lambda)=f(x)+\lambda'(Ex-d)+\frac{c}{2}\|Ex-d\|^2.$$

The minimizing v and x in Eq. (6.179) 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 either assumed or verified independently).

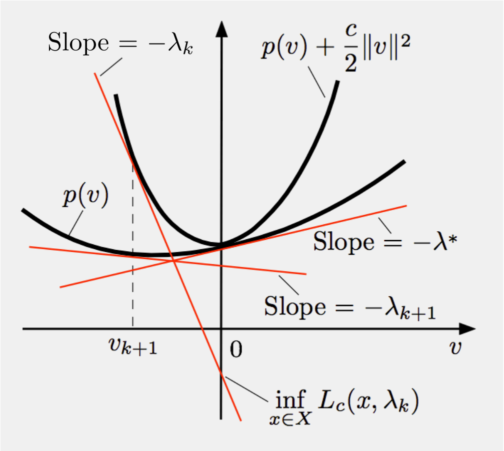

Figure 6.6.5. Illustration of the dual proximal minimization (6.178) and the update

$$\lambda_{k+1}=\lambda_k+c_kv_{k+1}$$

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. (6.169)].

Thus, the iteration of the dual algorithm [cf. Eq. (6.171), with a change of sign of λ_k inherited from the change of sign in Eq. (6.178)] takes the form $\lambda_{k+1} = \lambda_k + c_k v_{k+1}$, or

$$\lambda_{k+1} = \lambda_k + c_k(Ex_{k+1} - d), \tag{6.180}$$

where

$$x_{k+1} \in \arg\min_{x\in X} L_{c_k}(x, \lambda_k). \tag{6.181}$$

The algorithm (6.180)-(6.181) 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 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 algorithm (cf. Section 6.5.1). The sequence $\big\{q(\lambda_k)\big\}$ 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).

Assuming that $\{\lambda_k\}$ converges to an optimal dual solution, we also claim that every limit point of the generated sequence $\{x_k\}$ is an optimal solution of the primal problem (6.177). To see this, note that from the update formula (6.180) we obtain

$$Ex_{k+1} - d \to 0, \qquad c_k\big(Ex_{k+1} - d\big) \to 0.$$

Furthermore, we have

$$L_{c_k}\big(x_{k+1}, \lambda_k\big) = \min_{x\in X}\left\{f(x) + \lambda_k'(Ex - d) + \frac{c_k}{2}\|Ex - d\|^2\right\}.$$

The preceding relations yield

$$\limsup_{k\to\infty} f(x_{k+1}) = \limsup_{k\to\infty} L_{c_k}\big(x_{k+1}, \lambda_k\big) \le f(x), \quad \forall\ x \in X \text{ with } Ex = d,$$

so if $x^* \in X$ is a limit point of $\{x_k\}$, we obtain

$$f(x^*) \le f(x), \qquad \forall\ x \in X \text{ with } Ex = d,$$

as well as $Ex^* = d$ (in view of $Ex_{k+1} - d \to 0$). Therefore any limit point x^* of the generated sequence $\{x_k\}$ is an optimal solution of the primal problem (6.177). However, there is no guarantee that $\{x_k\}$ has a limit point, and indeed the dual sequence $\{\lambda_k\}$ will converge to a dual optimal solution, if one exists, even if the primal problem (6.177) does not have an optimal solution.

Finally, let us consider the "penalized" dual function q_c, given by

$$q_c(\lambda) = \max_{y \in \Re^m} \left\{ q(y) - \frac{1}{2c} \|y - \lambda\|^2 \right\}. \tag{6.182}$$

Then, according to Prop. 6.5.4, q_c is differentiable, and we have

$$\nabla q_c(\lambda) = \frac{y_c(\lambda) - \lambda}{c}, \tag{6.183}$$

where $y_c(\lambda)$ is the unique vector attaining the maximum in Eq. (6.182). Since $y_{c_k}(\lambda_k) = \lambda_{k+1}$, we have using Eqs. (6.180) and (6.183),

$$\nabla q_{c_k}(\lambda_k) = \frac{\lambda_{k+1} - \lambda_k}{c_k} = E x_{k+1} - d, \tag{6.184}$$

and the multiplier iteration (6.180) 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 (whose maxima coincide with the ones of q, for any $c > 0$). 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.

6.7 INCREMENTAL PROXIMAL METHODS

In this section we will consider incremental variants of the proximal algorithm, which like the incremental subgradient methods of Section 6.3.3, apply to additive costs of the form

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

where the functions $f_i : \Re^n \mapsto \Re$ are real-valued and 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

$$x_{k+1} = \psi_{m,k}, \tag{6.185}$$

where starting with $\psi_{0,k} = x_k$, we obtain $\psi_{m,k}$ after the m proximal steps

$$\psi_{i,k} = \arg\min_{x\in X} \left\{ f_i(x) + \frac{1}{2\alpha_k}\|x - \psi_{i-1,k}\|^2 \right\}, \qquad i = 1,\ldots,m, \tag{6.186}$$

where α_k is a positive parameter.

We will discuss shortly schemes to adjust α_k. We will see that α_k plays a role analogous to the stepsize in incremental subgradient methods (see the subsequent Prop. 6.7.1). In this connection, we note that for convergence of the method, it is essential that $\alpha_k \to 0$, as illustrated in the following example.

Example 6.7.1:

Consider the unconstrained scalar case ($X = \Re$) 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

$$\psi_{1,k} = \arg\min_{x\in\Re} \left\{ |x| + \frac{1}{2\alpha}|x - x_k|^2 \right\},$$

$$\psi_{2,k} = \arg\min_{x\in\Re} \left\{ |x-1| + \frac{1}{2\alpha}|x - \psi_{1,k}|^2 \right\},$$

$$\psi_{3,k} = x_{k+1} = \arg\min_{x\in\Re} \left\{ |x+1| + \frac{1}{2\alpha}|x - \psi_{2,k}|^2 \right\},$$

and generates the sequences $\psi_{1,k} = 0$, $\psi_{2,k} = \alpha$, $\psi_{3,k} = x_k = 0$, $k = 0,1,\ldots$. Thus, starting at the optimal solution $x_0 = 0$ and using a constant parameter $\alpha_k \equiv \alpha$, the method oscillates proportionately to α.

The following proposition suggests the similarity between incremental proximal and incremental subgradient methods.

Proposition 6.7.1: The incremental proximal iteration (6.186) can be written as

$$\psi_{i,k} = P_X(\psi_{i-1,k} - \alpha_k g_{i,k}), \qquad i = 1,\ldots,m, \tag{6.187}$$

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. (6.187) 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 Prop. 6.3.7. We denote

$$c_i = \sup_{k\geq 0}\max\big\{\|\hat g_{i,k}\|,\,\|g_{i,k}\|\big\},\qquad i=1,\ldots,m, \tag{6.188}$$

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. (6.187). In the following proposition the scalars c_i are assumed finite (this replaces Assumption 6.3.2).

Proposition 6.7.2: Let $\{x_k\}$ be the sequence generated by the incremental proximal method (6.185), (6.186). Then for all $y\in X$ and $k\geq 0$, we have

$$\|x_{k+1}-y\|^2 \leq \|x_k-y\|^2 - 2\alpha_k\big(f(x_{k+1})-f(y)\big)+\alpha_k^2c^2, \tag{6.189}$$

where $c=\sum_{i=1}^m c_i$ is assumed finite.

Proof: Using the nonexpansion property of the projection[cf. Eq. (6.57)], and the subgradient inequality for each component function f_i, we obtain for all $y \in X$, $i = 1, \ldots, m$, and $k \geq 0$,

$$\begin{aligned}
\|\psi_{i,k} - y\|^2 &= \big\|P_X\left(\psi_{i-1,k} - \alpha_k g_{i,k}\right) - y\big\|^2 \\
&\leq \|\psi_{i-1,k} - \alpha_k g_{i,k} - y\|^2 \\
&\leq \|\psi_{i-1,k} - y\|^2 - 2\alpha_k g_{i,k}'(\psi_{i-1,k} - y) + \alpha_k^2 \|g_{i,k}\|^2 \\
&\leq \|\psi_{i-1,k} - y\|^2 - 2\alpha_k g_{i,k}'(\psi_{i,k} - y) + \alpha_k^2 c_i^2 \\
&\quad + 2\alpha_k g_{i,k}'(\psi_{i,k} - \psi_{i-1,k}) \\
&\leq \|\psi_{i-1,k} - y\|^2 - 2\alpha_k \big(f_i(\psi_{i,k}) - f_i(y)\big) + \alpha_k^2 c_i^2 \\
&\quad + 2\alpha_k g_{i,k}'(\psi_{i,k} - \psi_{i-1,k}).
\end{aligned}$$

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

$$\|\psi_{i,k} - y\|^2 \leq \|\psi_{i-1,k} - y\|^2 - 2\alpha_k \big(f_i(\psi_{i,k}) - f_i(y)\big) + \alpha_k^2 c_i^2, \qquad \forall\ i, k.$$

By adding the above inequalities over $i = 1, \ldots, m$, we have for all $y \in X$ and k,

$$\begin{aligned}
\|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2\alpha_k \sum_{i=1}^m \big(f_i(\psi_{i,k}) - f_i(y)\big) + \alpha_k^2 \sum_{i=1}^m c_i^2 \\
&= \|x_k - y\|^2 - 2\alpha_k \left(f(x_{k+1}) - f(y) + \sum_{i=1}^m \big(f_i(\psi_{i,k}) - f_i(x_{k+1})\big) \right) \\
&\quad + \alpha_k^2 \sum_{i=1}^m c_i^2.
\end{aligned}$$

By strengthening the above inequality, we have for all $y \in X$ and k, using also the fact $\psi_{m,k} = x_{k+1}$,

$$\begin{aligned}
\|x_{k+1} - y\|^2 &\leq \|x_k - y\|^2 - 2\alpha_k \big(f(x_{k+1}) - f(y)\big) \\
&\quad + 2\alpha_k \sum_{i=1}^m c_i ||\psi_{i,k} - x_{k+1}|| + \alpha_k^2 \sum_{i=1}^m c_i^2 \\
&\leq \|x_k - y\|^2 - 2\alpha_k \big(f(x_{k+1}) - f(y)\big) \\
&\quad + \alpha_k^2 \left(2 \sum_{i=1}^{m-1} c_i \left(\sum_{j=i+1}^m c_j \right) + \sum_{i=1}^m c_i^2 \right) \\
&= \|x_k - y\|^2 - 2\alpha_k \big(f(x_{k+1}) - f(y)\big) + \alpha_k^2 \left(\sum_{i=1}^m c_i \right)^2 \\
&= \|x_k - y\|^2 - 2\alpha_k \big(f(x_{k+1}) - f(y)\big) + \alpha_k^2 c^2,
\end{aligned}$$

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 (6.188) of c_i, while in the second inequality we use the relation

$$\|\psi_{i,k} - x_{k+1}\| \leq \alpha_k \sum_{j=i+1}^{m} \|g_{j,k}\| \leq \alpha_k \sum_{j=i+1}^{m} c_j, \qquad i = 1, \ldots, m, \quad k \geq 0,$$

which follows from Eqs. (6.185)-(6.187), and the definition (6.188) of c_i. **Q.E.D.**

Among other things, Prop. 6.7.2 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 α_k is sufficiently small [less than $2\big(f(x_{k+1}) - f(y)\big)/c^2$]. 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 Prop. 6.7.2, we can provide convergence results for the incremental proximal method, which parallel 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 $\alpha \to 0$. In particular, Prop. 6.3.8 holds verbatim, and the convergence rate estimate of Prop. 6.3.9 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 6.3.4.

Incremental Augmented Lagrangian Methods

We will now revisit the Augmented Lagrangian methodology of Section 6.6.2, in the context of large-scale separable problems and incremental proximal methods. Consider the separable constrained minimization problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} f_i(x_i) \\ &\text{subject to} \quad x_i \in X_i, \quad i = 1, \ldots, m, \quad \sum_{i=1}^{m} (E_i x_i - d_i) = 0, \end{aligned} \tag{6.190}$$

where $f_i : \Re^{n_i} \mapsto \Re$ are convex functions (n_i is a positive integer, which may depend on i), X_i are nonempty closed convex sets, E_i are given $r \times n_i$ matrices, and $d_i \in \Re^r$ are given vectors. For simplicity and consistency with Section 6.6.2, we focus on linear equality constraints, but the analysis can be extended to separable problems with convex inequality constraints as well.

Similar to our discussion of separable problems in Section 6.1.1, the dual function is given by

$$q(\lambda) = \inf_{x_i \in X_i,\, i=1,\dots,m} \left\{ \sum_{i=1}^m \big(f_i(x_i) + \lambda'(E_i x_i - d_i) \big) \right\},$$

and by decomposing the minimization over the components of x, can be expressed in the additive form

$$q(\lambda) = \sum_{i=1}^m q_i(\lambda),$$

where

$$q_i(\lambda) = \inf_{x_i \in X_i} \big\{ f_i(x_i) + \lambda'(E_i x_i - d_i) \big\}.$$

Thus the dual problem,

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=1}^m q_i(\lambda) \\ \text{subject to} & \lambda \in \Re^r, \end{array}$$

has a suitable form for application of incremental methods, assuming that the dual function components q_i are real-valued.† In particular, the incremental proximal algorithm (6.185)-(6.186) updates the current vector λ_k to a new vector λ_{k+1} after a cycle of m subiterations:

$$\lambda_{k+1} = \psi_{m,k}, \tag{6.191}$$

where starting with $\psi_{0,k} = \lambda_k$, we obtain $\psi_{m,k}$ after the m proximal steps

$$\psi_{i,k} = \arg\max_{\lambda \in \Re^r} \left\{ q_i(\lambda) - \frac{1}{2\alpha_k} \|\lambda - \psi_{i-1,k}\|^2 \right\}, \qquad i = 1, \dots, m, \tag{6.192}$$

† The function q_i is real-valued for example if X_i is compact. If instead q_i has the form

$$q_i(\lambda) = \begin{cases} \tilde{q}_i(\lambda) & \text{if } \lambda \in \Lambda_i, \\ -\infty & \text{if } \lambda \notin \Lambda_i, \end{cases}$$

where $\tilde{q}_i : \Re^r \mapsto \Re$ is a real-valued concave function and Λ_i is a closed convex set, then a constrained form of the incremental proximal algorithm applies, which can deal incrementally with the dual constraint set $\Lambda = \cap_{i=1}^m \Lambda_i$ (see Section 6.7.2).

where α_k is a positive parameter.

We now recall the Fenchel duality relation between proximal and Augmented Lagrangian minimization discussed in Section 6.6.2. Based on that relation, the proximal incremental update (6.192) can be written in terms of the data of the primal problem as

$$\psi_{i,k} = \psi_{i-1,k} + \alpha_k(E_{i,k}x_{i,k} - d_i), \tag{6.193}$$

where $x_{i,k}$ is obtained with the minimization

$$x_{i,k} \in \arg\min_{x_i \in X_i} L_{\alpha_k,i}(x_i, \psi_{i-1,k}), \tag{6.194}$$

and $L_{\alpha_k,i}$ is the "incremental" Augmented Lagrangian function

$$L_{\alpha_k,i}(x_i, \lambda) = f_i(x_i) + \lambda'(E_i x_i - d_i) + \frac{\alpha_k}{2}\|E_i x_i - d_i\|^2. \tag{6.195}$$

Note that this algorithm allows decomposition within the Augmented Lagrangian framework, which is not possible in the standard method of Section 6.6.2 [cf. Eq. (6.179)], since the addition of the penalty term

$$\frac{c}{2}\left\|\sum_{i=1}^{m}(E_i x_i - d_i)\right\|^2$$

to the Lagrangian function destroys its separability.

6.7.1 Incremental Subgradient-Proximal Methods

The incremental proximal method of the preceding section has some advantages over the incremental subgradient method of Section 6.3.3 [cf. Eqs. (6.69)-(6.70)], chief among which is the greater stability that characterizes the proximal method. On the other hand while some cost function components may be well suited for a proximal iteration, others may not be because the minimization in Eq. (6.186) is inconvenient. With this in mind, we may consider combinations of subgradient and proximal methods for problems of the form

$$\begin{aligned} &\text{minimize} \quad F(x) \overset{\text{def}}{=} \sum_{i=1}^{m} F_i(x) \\ &\text{subject to} \ \ x \in X, \end{aligned}$$

where for all i,

$$F_i(x) = f_i(x) + h_i(x),$$

$f_i : \Re^n \mapsto \Re$ and $h_i : \Re^n \mapsto \Re$ 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 incremental subgradient and proximal iterations, respectively. To this end, we may consider several incremental algorithms that involve a combination of a proximal and a subgradient iteration. One such algorithm has the form

$$z_k = \arg\min_{x \in X} \left\{ f_{i_k}(x) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}, \tag{6.196}$$

$$x_{k+1} = P_X\big(z_k - \alpha_k g_{i_k}(z_k)\big), \tag{6.197}$$

where i_k is the index of the component chosen for iteration and $g_{i_k}(z_k)$ is an arbitrary subgradient of h_{i_k} at z_k. Note that the iteration is well-defined because the minimum in Eq. (6.196) is uniquely attained since f_i is continuous and $\|x - x_k\|^2$ is real-valued, strictly convex, and coercive, while the subdifferential $\partial h_i(z_k)$ is nonempty since h_i is real-valued. Note also that by choosing all the f_i or all the h_i to be identically zero, we obtain as special cases the incremental subgradient and incremental proximal iterations, respectively.

The iterations (6.196) and (6.197) maintain both sequences $\{z_k\}$ and $\{x_k\}$ within the constraint set X, but it may be convenient to relax this constraint for either the proximal or the subgradient iteration, thereby requiring a potentially simpler computation. This leads to the algorithm

$$z_k = \arg\min_{x \in \Re^n} \left\{ f_{i_k}(x) + \frac{1}{2\alpha_k} \|x - x_k\|^2 \right\}, \tag{6.198}$$

$$x_{k+1} = P_X\big(z_k - \alpha_k g_{i_k}(z_k)\big), \tag{6.199}$$

where the restriction $x \in X$ has been omitted from the proximal iteration, and the algorithm

$$z_k = x_k - \alpha_k g_{i_k}(x_k), \tag{6.200}$$

$$x_{k+1} = \arg\min_{x \in X} \left\{ f_{i_k}(x) + \frac{1}{2\alpha_k} \|x - z_k\|^2 \right\}, \tag{6.201}$$

where the projection onto X has been omitted from the subgradient iteration. It is also possible to use different stepsize sequences in the proximal and subgradient iterations.

The convergence and rate of convergence properties of such combined methods are similar to the ones derived earlier for the pure subgradient and pure proximal versions (see Bertsekas [Ber10a] and [Ber10b], where it is shown that their 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.

We now illustrate the incremental methods of this subsection in the context of some important applications.

Weighted Least Squares and Regularization

Let us consider weighted least squares problems, involving minimization of a sum of quadratic component functions $f_i(x)$ that correspond to errors between data and the output of a model that is parameterized by a vector $x \in \Re^n$. Often a convex regularization function $R(x)$ is added to the least squares objective, to induce desirable properties of the solution. This gives rise to problems of the form

$$\begin{array}{ll} \text{minimize} & R(x) + \dfrac{1}{2}\displaystyle\sum_{i=1}^{m} w_i (c_i' x - d_i)^2 \\ \text{subject to} & x \in \Re^n, \end{array} \tag{6.202}$$

where c_i and d_i are given vectors and scalars, respectively, and w_i are positive weights. Typically, either $R(x) \equiv 0$ (which corresponds to no regularization) or R has an additive form, $R(x) = \sum_{j=1}^{n} R_j(x^j)$, where $x^1, \ldots, x^n$ are the n components of x. Then if either m is very large or the data (c_i, d_i) become available sequentially over time, it makes sense to consider incremental methods.

The classical type of regularization involves a quadratic function R (as in classical regression and the LMS method),

$$R(x) = \frac{\gamma}{2}\|x\|^2,$$

where γ is a positive scalar. Then there are several possibilities for applying incremental gradient, or proximal methods or combinations thereof, since differentiation and minimization operations involving quadratic functions can typically be done in closed form.

A popular alternative to quadratic regularization, which is well suited for some applications, is the use of a nondifferentiable regularization function. Our incremental methods may still apply because what is important is that R has a simple form that facilitates the use of proximal algorithms, such as for example a separable form, so that the proximal iteration on

R is simplified through decomposition. As an example, consider the case where $R(x)$ is given in terms of the ℓ_1-norm:

$$R(x) = \gamma\|x\|_1 = \gamma\sum_{j=1}^{n}|x^j|, \tag{6.203}$$

γ is a positive scalar and x^j is the jth coordinate of x. Then the proximal iteration

$$z_k = \arg\min_{x\in\Re^n}\left\{\gamma\,\|x\|_1 + \frac{1}{2\alpha_k}\|x - x_k\|^2\right\}$$

decomposes into the n one-dimensional minimizations

$$z_k^j = \arg\min_{x^j\in\Re}\left\{\gamma\,|x^j| + \frac{1}{2\alpha_k}|x^j - x_k^j|^2\right\}, \qquad j = 1,\ldots,n,$$

and can be done in closed form

$$z_k^j = \begin{cases} x_k^j - \gamma\alpha_k & \text{if } \gamma\alpha_k \le x_k^j, \\ 0 & \text{if } -\gamma\alpha_k < x_k^j < \gamma\alpha_k, \\ x_k^j + \gamma\alpha_k & \text{if } x_k^j \le -\gamma\alpha_k, \end{cases} \qquad j = 1,\ldots,n. \tag{6.204}$$

We refer to the literature for a discussion of a broad variety of applications in estimation and signal processing problems, where nondifferentiable regularization functions play an important role.

We now note that the incremental algorithms of this section are well-suited for solution of problems of the form (6.202)-(6.203). For example, the kth incremental iteration may consist of selecting a data pair (c_{i_k}, d_{i_k}) and performing a proximal iteration of the form (6.204) to obtain z_k, followed by a gradient iteration on the component $\frac{1}{2}w_i(c_{i_k}'x - d_{i_k})^2$, starting at z_k:

$$x_{k+1} = z_k - \alpha_k w_i c_{i_k}(c_{i_k}' z_k - d_{i_k}).$$

This algorithm is a special case of the algorithms of this section with $f_i(x)$ being $\gamma\|x\|_1$ (we use m copies of this function) and $h_i(x) = \frac{1}{2}w_i(c_i'x - d_i)^2$. It can be viewed as an incremental version of a popular class of algorithms in signal processing, known as iterative shrinkage/thresholding.

Finally, let us note that as an alternative, the proximal iteration (6.204) could be replaced by a proximal iteration on $\gamma\,|x^j|$ for some selected index j, with all indexes selected cyclically in incremental iterations. Randomized selection of the data pair (c_{i_k}, d_{i_k}) is also interesting, particularly in contexts where the data has a natural stochastic interpretation.

Iterated Projection Algorithms

A feasibility problem that arises in many contexts involves finding a point with certain properties within a set intersection $\cap_{i=1}^m X_i$, where each X_i is a closed convex set. For the case where m is large and each of the sets X_i has a simple form, incremental methods that make successive projections on the component sets X_i have a long history. We may consider the following generalization of the classical feasibility problem,

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in \cap_{i=1}^m X_i, \end{aligned} \tag{6.205}$$

where $f : \Re^n \mapsto \Re$ is a convex cost function, and the method

$$x_{k+1} = P_{X_{i_k}}\big(x_k - \alpha_k g(x_k)\big), \tag{6.206}$$

where the index i_k is chosen from $\{1, \ldots, m\}$ according to a randomized rule, and $g(x_k)$ is a subgradient of f at x_k. The incremental approach is particularly well-suited for problems of the form (6.205) where the sets X_i are not known in advance, but are revealed as the algorithm progresses. While the problem (6.205) does not involve a sum of component functions, it may be converted into one that does by using an exact penalty function. In particular, consider the problem

$$\begin{aligned} &\text{minimize} \quad f(x) + \gamma \sum_{i=1}^m \text{dist}(x; X_i) \\ &\text{subject to} \quad x \in \Re^n, \end{aligned} \tag{6.207}$$

where γ is a positive penalty parameter. Then for f Lipschitz continuous and γ sufficiently large, problems (6.205) and (6.207) are equivalent, as shown in Prop. 6.1.12.

Regarding algorithmic solution, from Prop. 6.1.12, it follows that we may consider in place of the original problem (6.205) the additive cost problem (6.207) for which our algorithms apply. In particular, let us consider an algorithm that involves a proximal iteration on one of the functions $\gamma\,\text{dist}(x; X_i)$ followed by a subgradient iteration on f. A key fact here is that the proximal iteration involving the distance function $\text{dist}(\cdot; X_{i_k})$,

$$z_k = \arg\min_{x \in \Re^n} \left\{ \gamma\,\text{dist}(x; X_{i_k}) + \frac{1}{2\alpha_k}\|x - x_k\|^2 \right\}, \tag{6.208}$$

consists of a projection on X_{i_k} of x_k, followed by an interpolation. This is shown in the following proposition.

Proposition 6.7.3: Let z_k be the vector produced by the proximal iteration (6.208). If $x_k \in X_{i_k}$, then $z_k = x_k$, while if $x_k \notin X_{i_k}$,

$$z_k = \begin{cases} (1-\beta_k)x_k + \beta_k P_{X_{i_k}}(x_k) & \text{if } \beta_k < 1, \\ P_{X_{i_k}}(x_k) & \text{if } \beta_k \geq 1, \end{cases} \tag{6.209}$$

where

$$\beta_k = \frac{\alpha_k \gamma}{\text{dist}(x_k; X_{i_k})}.$$

Proof: The case $x_k \in X_{i_k}$ is evident, so assume that $x_k \notin X_{i_k}$. From the nature of the cost function in Eq. (6.208) we see that z_k is a vector that lies in the line segment between x_k and $P_{X_{i_k}}(x_k)$. Hence there are two possibilities: either

$$z_k = P_{X_{i_k}}(x_k), \tag{6.210}$$

or $z_k \notin X_{i_k}$ in which case by setting to 0 the gradient at z_k of the cost function in Eq. (6.208) yields

$$\gamma \frac{z_k - P_{X_{i_k}}(z_k)}{\Big\| z_k - P_{X_{i_k}}(z_k) \Big\|} = \frac{1}{\alpha_k}(x_k - z_k).$$

This equation implies that x_k, z_k, and $P_{X_{i_k}}(z_k)$ lie on the same line, so that $P_{X_{i_k}}(z_k) = P_{X_{i_k}}(x_k)$ and

$$z_k = x_k - \frac{\alpha_k \gamma}{\text{dist}(x_k; X_{i_k})}\big(x_k - P_{X_{i_k}}(x_k)\big) = (1-\beta_k)x_k + \beta_k P_{X_{i_k}}(x_k). \tag{6.211}$$

By calculating and comparing the value of the cost function in Eq. (6.208) for each of the possibilities (6.210) and (6.211), we can verify that (6.211) gives a lower cost if and only if $\beta_k < 1$. **Q.E.D.**

Let us now consider the problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} \big(f_i(x) + h_i(x)\big) \\ &\text{subject to} \quad x \in \cap_{i=1}^m X_i. \end{aligned}$$

As noted earlier, using Prop. 6.1.12, we can convert this problem to the unconstrained minimization problem

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{m} \big(f_i(x) + h_i(x) + \gamma \text{dist}(x; X_i)\big) \\ &\text{subject to} \quad x \in \Re^n, \end{aligned}$$

where γ is sufficiently large. The algorithm (6.200)-(6.201) applied to this problem, yields the iteration

$$y_k = x_k - \alpha_k g_{i_k}(x_k), \qquad z_k = \arg\min_{x\in\Re^n}\left\{ f_{i_k}(x) + \frac{1}{2\alpha_k}\|x-y_k\|^2 \right\}, \tag{6.212}$$

$$x_{k+1} = \begin{cases} (1-\beta_k)z_k + \beta_k P_{X_{i_k}}(z_k) & \text{if } \beta_k < 1, \\ P_{X_{i_k}}(z_k) & \text{if } \beta_k \geq 1, \end{cases} \tag{6.213}$$

where i_k is the index of the component chosen for iteration and $g_{i_k}(x_k)$ is an arbitrary subgradient of h_{i_k} at x_k, and

$$\beta_k = \frac{\alpha_k \gamma}{\text{dist}(z_k; X_{i_k})}, \tag{6.214}$$

[cf. Eq. (6.209)].

Let us finally note that our incremental methods also apply to the case where f has an additive form:

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^m f_i(x) \\ &\text{subject to} \ \ x \in \cap_{i=1}^m X_i. \end{aligned}$$

In this case the interpolated projection iterations (6.209) on the sets X_i are followed by subgradient or proximal iterations on the components f_i. A related problem for which our methods are well-suited is

$$\begin{aligned} &\text{minimize} \quad f(x) + c\sum_{j=1}^r \max\big\{0, g_j(x)\big\} \\ &\text{subject to} \ \ x \in \cap_{i=1}^m X_i, \end{aligned}$$

which is obtained by replacing convex inequality constraints of the form $g_j(x) \leq 0$ with the nondifferentiable penalty terms $c \max\big\{0, g_j(x)\big\}$, where $c > 0$ is a penalty parameter (cf. Section 6.1.5). Then a possible incremental method at each iteration, would either do a subgradient iteration on f, or select one of the violated constraints (if any) and perform a subgradient iteration on the corresponding function g_j, or select one of the sets X_i and do an interpolated projection on it.

6.7.2 Incremental Constraint Projection-Proximal Methods

In this subsection we consider incremental algorithms for problems of the form

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \ \ x \in \cap_{i=1}^m X_i, \end{aligned}$$

with a focus on the case where the number m of components X_i in the constraint set $\cap_{i=1}^m X_i$ is large, so that iterations involving a single component are desirable. The approach of the preceding subsection [cf. the algorithm (6.212)-(6.214)] was to replace the problem by an equivalent penalized unconstrained problem. The reason was that our earlier subgradient-proximal algorithms require that all cost component functions must be real-valued, so we may not replace the constraint components X_i by their (extended real-valued) indicator functions $\delta(\cdot \mid X_i)$. In this section, we summarize a different approach, which instead treats the constraint components directly.

For generality, let us consider the case where the cost function f is given as the expected value

$$f(x) = E\big[f_w(x)\big],$$

where $f_w : \Re^n \mapsto \Re$ is a function of x involving a random variable v. The case $f(x) = \sum_{i=1}^m f_i(x)$ is a special case where w is uniformly distributed over the subset of integers $\{1, \ldots, m\}$.

A natural modification of the subgradient projection and proximal methods, is to select the constraints X_i randomly, and process them sequentially in conjunction with sample component functions $f_w(\cdot)$. This motivates algorithms, which at iteration k, select cost function components f_{w_k}, and constraint components X_{v_k}, with $\{v_k\}$ being a sequence of random variables taking values in $\{1, \ldots, m\}$, and $\{w_k\}$ being a sequence of random variables, generated by some probabilistic process (for example w_k may be independent identically distributed with the same distribution as w). In particular, two algorithms based on these component sampling schemes are:

(a) The subgradient projection-like algorithm

$$x_{k+1} = P_{v_k}\big(x_k - \alpha_k g(w_k, x_k)\big), \tag{6.215}$$

where $g(w_k, x_k)$ is any subgradient of f_{w_k} at x_k and P_{v_k} denotes the projection onto the set X_{v_k}.

(b) The proximal-like algorithm

$$x_{k+1} = \arg \min_{x \in X_{v_k}} \left\{ f_{w_k}(x) + \frac{1}{2\alpha_k}\|x - x_k\|^2 \right\}. \tag{6.216}$$

Note that by using individual constraint set components X_{v_k}, the projection P_{v_k} in Eq. (6.215), and the proximal minimization in Eq. (6.216) may be greatly simplified. For example, when X is a polyhedral set, it can be represented as the intersection of a finite number of halfspaces. Then the algorithms (6.215) and (6.216) involve successive projections onto or minimizations over halfspaces, which are often easier to implement and computationally inexpensive.

By combining the preceding subgradient projection and proximal incremental iterations, we also obtain the following algorithm that involves random optimality updates and random feasibility updates, of the form

$$z_k = x_k - \alpha_k g(w_k, \bar{x}_k), \qquad x_{k+1} = z_k - \beta_k \left(z_k - P_{v_k} z_k\right), \tag{6.217}$$

where $\bar{x}_k$ is a random variable "close" to x_k such as

$$\bar{x}_k = x_k, \qquad \text{or} \qquad \bar{x}_k = x_{k+1}, \tag{6.218}$$

and $\{\beta_k\}$ is a sequence of positive scalars. In the case where $\bar{x}_k = x_k$, the kth iteration is a subgradient projection step and takes the form of Eq. (6.215). In the case where $\bar{x}_k = x_{k+1}$, the corresponding iteration is a proximal step and takes the form of Eq. (6.216) [cf. Prop. 6.7.1].

The convergence analysis of the algorithm (6.217)-(6.218) requires that the stepsize α_k diminishes to 0, as in earlier incremental methods. The stepsize β_k should either be constant [e.g., $\beta_k = 1$, in which case the second iteration in Eq. (6.217) takes the form $x_{k+1} = P_{v_k} z_k$] or converge to 0 more slowly than α_k. In this way the algorithm tends to operate on two different time scales: the convergence to the feasible set, which is controlled by β_k, is faster than the convergence to the optimal solution, which is controlled by α_k. We refer to Wang and Bertsekas [WaB12], [WaB13] for this analysis, under a variety of sampling schemes for w_k and v_k, both cyclic and randomized, as well as to the earlier paper by Nedić [Ned11], which first proposed and analyzed constraint projection algorithms.

6.8 GENERALIZED PROXIMAL ALGORITHMS AND EXTENSIONS

The proximal algorithm admits several extensions, which may be particularly well-suited for specialized application domains, such as inference and signal processing. Moreover the algorithm, with some unavoidable limitations, applies to nonconvex problems as well. In this section, we will illustrate some of main ideas and possibilities, without providing a substantial convergence analysis.

A general form of the algorithm for minimizing a closed proper extended real-valued function $f : \Re^n \mapsto (-\infty, \infty]$ is

$$x_{k+1} \in \arg \min_{x \in \Re^n} \big\{ f(x) + D_k(x, x_k) \big\}, \tag{6.219}$$

where $D_k : \Re^{2n} \mapsto (-\infty, \infty]$ is a regularization term that replaces the quadratic

$$\frac{1}{2c_k} \|x - x_k\|^2$$

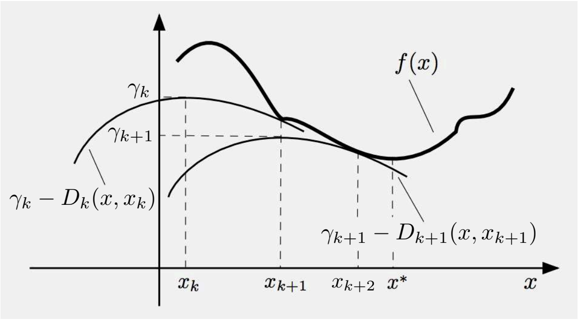

Figure 6.8.1. Illustration of the generalized proximal algorithm (6.219). The regularization term need not be quadratic, and the cost function f need not be convex.

in the proximal algorithm.

The algorithm (6.219) can be graphically interpreted similar to the proximal algorithm, as illustrated in Fig. 6.8.1. The figure and the convergence and rate of convergence results of the preceding section provide some qualitative guidelines about the kind of behavior that may be expected from the algorithm. This behavior, however, may be complicated and/or unreliable, particularly when the cost function f is nonconvex.

An important question is whether the minimum in Eq. (6.219) is attained for all k; this is not automatically guaranteed, even if D_k is continuous and coercive, because f is not assumed convex [take for example $f(x) = -\|x\|^3$ and $D_k(x, x_k) = \|x - x_k\|^2$]. To simplify the presentation, we will implicitly assume the attainment of the minimum throughout our discussion; it is guaranteed for example if f is closed, proper, convex, and $D_k(\cdot, x_k)$ is closed, coercive, and its effective domain intersects with $\mathrm{dom}(f)$ for all k (cf. Prop. 3.2.3).

We will now introduce two conditions on D_k that guarantee some sound behavior for the algorithm. The first is that x_k minimizes $D_k(\cdot, x_k)$:

$$D_k(x, x_k) \geq D_k(x_k, x_k), \qquad \forall\, x \in \Re^n,\ k = 0, 1, \ldots. \tag{6.220}$$

With this condition we are assured that the algorithm has a cost improvement property. Indeed, we have

$$\begin{aligned} f(x_{k+1}) &\leq f(x_{k+1}) + D_k(x_{k+1}, x_k) - D_k(x_k, x_k) \\ &\leq f(x_k) + D_k(x_k, x_k) - D_k(x_k, x_k) \\ &= f(x_k), \end{aligned} \tag{6.221}$$

where the first inequality follows from Eq. (6.220), and the second inequality follows from the definition (6.219) of the algorithm. The condition

(6.220) also guarantees that the algorithm stops at a global minimum of f, i.e.,

$$x^* \in \arg\min_{x\in\Re^n} f(x) \qquad \Rightarrow \qquad x^* \in \arg\min_{x\in\Re^n} \big\{f(x) + D_k(x, x^*)\big\}.$$

However, for the algorithm to be reliable, an additional condition is required to guarantee that it produces strict cost improvement when outside of some set of "desirable points" X^* (such as the global or the local minima of f). One such condition is that the algorithm can stop only at points of X^*, i.e.,

$$x_k \in \arg\min_{x\in\Re^n} \{f(x) + D_k(x, x_k)\} \qquad \Rightarrow \qquad x_k \in X^*, \tag{6.222}$$

in which case, the second inequality in the calculation of Eq. (6.221) is strict, and we have

$$f(x_{k+1}) < f(x_k), \qquad \text{if } x_k \notin X^*. \tag{6.223}$$

A set of assumptions guaranteeing the condition (6.222) are:

(a) f is convex and X^* is the set of global minima of f.

(b) $D_k(\cdot, x_k)$ satisfies Eq. (6.220), and is convex and differentiable at x_k.

(c) We have

$$\text{ri}\big(\text{dom}(f)\big) \cap \text{ri}\big(\text{dom}(D_k(\cdot, x_k))\big) \neq \emptyset. \tag{6.224}$$

To see this, note that if

$$x_k \in \arg\min_{x\in\Re^n} \big\{f(x) + D_k(x, x_k)\big\},$$

by the Fenchel duality theorem (Prop. 6.1.5), there exists a dual optimal solution λ^* such that $-\lambda^*$ is a subgradient of $D_k(\cdot, x_k)$ at x_k, so that $\lambda^* = 0$ [by Eq. (6.220)], and also λ^* is a subgradient of f at x_k, so that x_k minimizes f. Note that the condition (6.222) may fail if $D_k(\cdot, x_k)$ is not differentiable; for example, if

$$f(x) = \tfrac{1}{2}\|x\|^2, \qquad D_k(x, x_k) = \tfrac{1}{c}\|x - x_k\|,$$

then for any $c > 0$, the points $x_k \in [-1/c, 1/c]$ minimize $f(\cdot) + D_k(\cdot, x_k)$. Simple examples can also be constructed to show that the relative interior condition is essential to guarantee the condition (6.222).

We summarize the preceding discussion in the following proposition:

Proposition 6.8.1: Under the conditions (6.220) and (6.222), and assuming that the minimum of $f(x) + D_k(x, x_k)$ over x is attained for every k, the algorithm

$$x_{k+1} \in \arg \min_{x \in \Re^n} \big\{ f(x) + D_k(x, x_k) \big\}$$

improves strictly the value of f at each iteration where $x_k \notin X^*$, and may only stop (i.e., $x_{k+1} = x_k$) if $x_k \in X^*$.

Of course, cost improvement is a reassuring property for the algorithm (6.219), but it does not guarantee convergence. Thus, even under the assumptions of the preceding proposition, the convergence of the algorithm to a global minimum is not a foregone conclusion. This is true even if f is assumed convex and has a nonempty set of minima X^*, and $D_k(\cdot, x_k)$ is also convex.

If f is not convex, there may be additional serious difficulties. First, the global minimum in Eq. (6.219) may be hard to compute, since the cost function $f(\cdot) + D_k(\cdot, x_k)$ may not be convex. Second, the algorithm may converge to local minima or stationary points of f that are not global minima; see Fig. 6.8.2. As this figure indicates, convergence to a global minimum is facilitated if the regularization term $D_k(\cdot, x_k)$ is relatively “flat.” The algorithm may also not converge at all, even if f has local minima and the algorithm is started near or at a local minimum of f (see Fig. 6.8.3).

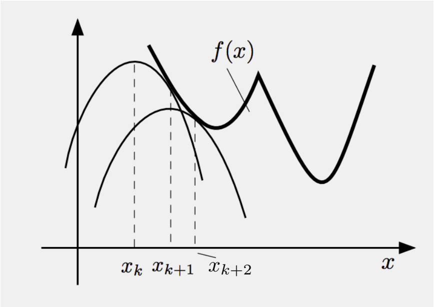

Figure 6.8.2. Illustration of a case where the generalized proximal algorithm (6.219) converges to a local minimum that is not global. In this example convergence to the global minimum would be attained if the regularization term $D_k(\cdot, x_k)$ were sufficiently “flat.”

In this section, we will not provide a convergence analysis of the generalized proximal algorithm (6.219). We will instead illustrate the algorithm with some examples.

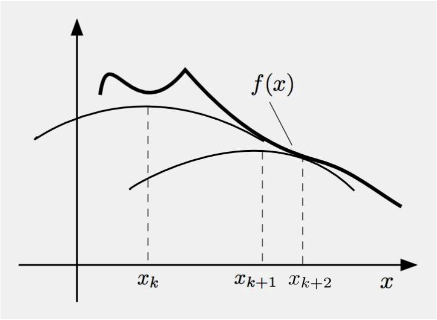

Figure 6.8.3. Illustration of a case where the generalized proximal algorithm (6.219) diverges even when started at a local minimum of f.

Example 6.8.1: (Nonquadratic Regularization with Bregman Distance)

Let $\phi : \Re^n \mapsto (-\infty, \infty]$ be a convex function, which is differentiable within an open set containing $\text{dom}(f)$, and define

$$D_k(x, y) = \frac{1}{c_k}\big(\phi(x) - \phi(y) - \nabla\phi(y)'(x - y)\big), \qquad \forall\ x, y \in \text{dom}(f),$$

where c_k is a positive penalty parameter. This function, with some additional conditions discussed in Exercise 6.19, is known as the *Bregman distance function*. Its use in connection with proximal minimization was suggested by Censor and Zenios [CeZ92], and followed up by several other authors; see the end-of-chapter references. Note that for $\phi(x) = \frac{1}{2}\|x\|^2$, we have

$$D_k(x, y) = \frac{1}{c_k}\left(\frac{1}{2}\|x\|^2 - \frac{1}{2}\|y\|^2 - y'(x - y)\right) = \frac{1}{2c_k}\|x - y\|^2,$$

so the quadratic regularization term of the proximal algorithm is included as a special case.

It can be seen that because of the convexity of ϕ, the condition (6.220) holds (cf. Prop. 1.1.7). Furthermore, because of the differentiability of $D_k(\cdot, x_k)$ (which follows from the differentiability of ϕ), the condition (6.222) holds as well when f is convex.

Example 6.8.2: (Majorization-Minimization Algorithm)

An equivalent version of the generalized proximal algorithm (6.219) (known as *majorization-minimization* algorithm) is obtained by absorbing the cost function into the regularization term. This leads to the algorithm

$$x_{k+1} \in \arg\min_{x \in \Re^n} M_k(x, x_k), \tag{6.225}$$

where $M_k : \Re^{2n} \mapsto (-\infty, \infty]$ satisfies the conditions

$$M_k(x, x) = f(x), \qquad \forall\, x \in \Re^n,\ k = 0, 1, \ldots, \tag{6.226}$$

$$M_k(x, x_k) \geq f(x_k), \qquad \forall\, x \in \Re^n,\ k = 0, 1, \ldots. \tag{6.227}$$

By defining

$$D_k(x, y) = M_k(x, y) - M_k(x, x),$$

we have

$$M_k(x, x_k) = f(x) + D_k(x, x_k),$$

so the algorithm (6.225) can be written in the generalized proximal format (6.219). Moreover the condition (6.227) is equivalent to the condition (6.220), which guarantees cost improvement [strict if we also assume the condition

$$x_k \in \arg\min_{x \in \Re^n} M_k(x, x_k) \qquad \Rightarrow \qquad x_k \in X^*,$$

where X^* is the set of desirable points for convergence, cf. Eq. (6.222) and Prop. 6.8.1]. The following example illustrates the majorization-minimization algorithm.

Example 6.8.3: (Regularized Least Squares)

Consider the problem of unconstrained minimization of the function

$$f(x) = R(x) + \|Ax - b\|^2,$$

where A is an $m \times n$ matrix, b is a vector in $\Re^m$, and $R : \Re^n \mapsto \Re$ is a nonnegative-valued convex regularization function. Let D be any symmetric matrix such that $D - A'A$ is positive definite (for example D may be a sufficiently large multiple of the identity). Let us define

$$M(x, y) = R(x) + \|Ay - b\|^2 + 2(x - y)'A'(Ay - b) + (x - y)'D(x - y),$$

and note that M satisfies the condition $M(x, x) = f(x)$ [cf. Eq. (6.226)], as well as the condition $M(x, x_k) \geq f(x_k)$ for all x and k [cf. Eq. (6.227)] in view of the calculation

$$\begin{aligned} M(x, y) - f(x) &= \|Ay - b\|^2 - \|Ax - b\|^2 \\ &\quad + 2(x - y)'A'(Ay - b) + (x - y)'D(x - y) \\ &= (x - y)'(D - A'A)(x - y). \end{aligned} \tag{6.228}$$

When D is the identity matrix I, by scaling A, we can make the matrix $I - A'A$ positive definite, and from Eq. (6.228), we have

$$M(x, y) = R(x) + \|Ax - b\|^2 - \|Ax - Ay\|^2 + \|x - y\|^2.$$

The majorization-minimization algorithm for this form of M has been used extensively in signal processing applications.

Example 6.8.4: (Expectation-Maximization Algorithm)

Let us discuss an algorithm that has many applications in problems of statistical inference, and show that it is a special case of the generalized proximal algorithm (6.219). We observe a sample of a random vector Z whose distribution $P_Z(\cdot;x)$ depends on an unknown parameter vector $x \in \Re^n$. For simplicity we assume that Z can take only a finite set of values, so that $P_Z(z;x)$ is the probability that Z takes the value z when the parameter vector has the value x. We wish to estimate x based on the given sample value z, by using the maximum likelihood method, i.e., by solving the problem

$$\begin{aligned} &\text{maximize} && \log P_Z(z;x) \\ &\text{subject to} && x \in \Re^n. \end{aligned} \tag{6.229}$$

Note that in the maximum likelihood method, it is often preferable to minimize the logarithm of $P_Z(z;x)$ rather than $P_Z(z;x)$ itself, because $P_Z(z;x)$ may be a product of probabilities corresponding to independent observations (see Example 6.1.6).

In many important practical settings the distribution $P_Z(\cdot;x)$ is known indirectly through the distribution $P_W(\cdot;x)$ of another random vector W that is related to Z via $Z = g(W)$, where g is some function (see Fig. 6.8.4). For example, Z may be a subvector of W (in statistical methodology, W and Z are said to be the "complete" and the "incomplete" data, respectively). Of course, one may calculate $P_Z(\cdot;x)$ from $P_W(\cdot;x)$ by using the formula

$$P_Z(z;x) = \sum_{\{w \mid g(w)=z\}} P_W(w;x),$$

but this may be inconvenient because the cost function of problem (6.229) involves the logarithm of a sum with a potentially large number of terms. An alternative that often involves more convenient calculations using the logarithm of $P_W(w;x)$ is the *expectation-maximization* method (EM for short), which we now describe.

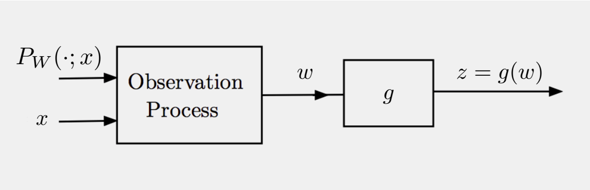

Figure 6.8.4. Illustration of the estimation context of the EM method. We are given data z and we want to estimate the parameter vector x. The EM method uses calculations involving the distribution $P_W(\cdot;x)$ of the "complete" obervation vector w, which defines z via $z = g(w)$.

We consider the following problem, which is equivalent to the maximum likelihood problem (6.229):

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in \Re^n, \end{aligned} \tag{6.230}$$

where

$$f(x) = -\log P_Z(z; x). \tag{6.231}$$

We will derive a suitable regularization term $D(x, y)$, so that the regularized cost function

$$f(x) + D(x, y)$$

is conveniently computed for all x and y, using the known distribution $P_W(\cdot; x)$. The EM algorithm is the generalized proximal algorithm (6.219) with f and D as described above, and aims to minimize f.

To derive D, we use the definition of conditional probability to write for any value w

$$P_W(w; x) = P_Z(z; x)\, P_W(w \mid z; x),$$

so that we have

$$\log P_W(w; x) = \log P_Z(z; x) + \log P_W(w \mid z; x), \qquad \forall\ w.$$

For any parameter value y (not necessarily equal to x), we take expected value in the above relation with respect to the conditional distribution $P_W(\cdot \mid z; y)$ and we obtain for all x and y,

$$E\big\{\log P_W(w; x) \mid z; y\big\} = \log P_Z(z; x) + E\big\{\log P_W(w \mid z; x) \mid z; y\big\}.$$

from which by using the definition (6.231) of f and by denoting

$$D(x, y) = -E\big\{\log P_W(w \mid z; x) \mid z; y\big\}, \tag{6.232}$$

we obtain

$$f(x) + D(x, y) = -E\big\{\log P_W(w; x) \mid z; y\big\}. \tag{6.233}$$

We now make two observations:

(a) $D(x, y)$ is a legitimate regularization function in that it satisfies the critical condition

$$D(x, y) \geq D(y, y), \qquad \forall\ x, y,$$

[cf. Eq. (6.220)], which guarantees the cost improvement property (6.223). This follows from a general property of probability distributions, known as Gibbs' inequality.†

† Gibbs' inequality within our context states that given a probability distribution $p = (p_1, \ldots, p_m)$ with positive components, the expression

$$F(q) = -\sum_{i=1}^{m} p_i \log q_i$$

is minimized uniquely over all probability distributions $q = (q_1, \ldots, q_m)$ by the distribution p. The proof is obtained by noting that F is a strictly convex function within the relative interior of the unit simplex, and by applying the optimality conditions for convex programming of Section 5.3.

(b) The regularized cost function $f(x)+D(x,y)$ of Eq. (6.233) can be expressed in terms of the logarithm of the distribution $P_W(w;x)$. To see this, we note the expression for the conditional distribution

$$P_W(w \mid z;y)=\begin{cases} \frac{P_W(w;y)}{\sum_{\{\overline{w}\mid g(\overline{w})=z\}} P_W(\overline{w};y)} & \text{if } g(w)=z,\\ 0 & \text{if } g(w)\neq z,\end{cases}$$

which can be used to write

$$\begin{aligned} f(x)+D(x,y)&=-E\big\{\log P_W(w;x)\mid z;y\big\}\\ &=-\frac{\sum_{\{w\mid g(w)=z\}} P_W(w;y)\log P_W(w;x)}{\sum_{\{\overline{w}\mid g(\overline{w})=z\}} P_W(\overline{w};y)}.\end{aligned}$$

Since the denominator is positive and independent of x, we see that the minimization of $f(x)+D(x,y)$ over x for fixed y is equivalent to minimization over x of the numerator

$$-\sum_{\{w\mid g(w)=z\}} P_W(w;y)\log P_W(w;x).$$

Changing minimization to maximization, we can now summarize the EM algorithm. It is given by

$$x_{k+1}\in\arg\max_{x\in\Re^n}\sum_{\{w\mid g(w)=z\}} P_W(w;x_k)\log P_W(w;x),$$

and involves the logarithm of $P_W(\cdot;x)$ rather than the logarithm of $P_Z(\cdot;x)$. It is equivalent to the generalized proximal algorithm with a regularization term $D(x,y)$ given by Eq. (6.232). There are a lot of questions regarding the convergence of the algorithm, because there is no guarantee of convexity of f, D, or the regularized cost function $f(\cdot)+D(\cdot,y)$. These questions must be addressed separately in the context of specific applications. A general analysis is given in the paper by Tseng [Tse04], which we have followed in this example.

6.9 INTERIOR POINT METHODS

In this section we will develop an approximation approach that is different from the linearization and regularization approaches of the preceding sections. This approach is based on approximating the indicator function of the constraint set by an "interior" penalty, which is added to the cost function. We focus on inequality-constrained problems of the form

$$\begin{aligned} &\text{minimize} \ \ f(x)\\ &\text{subject to} \ \ x\in X,\qquad g_j(x)\le 0,\quad j=1,\ldots,r,\end{aligned} \tag{6.234}$$

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

$$S = \bigl\{x \in X \mid g_j(x) < 0,\ j = 1, \ldots, r\bigr\},$$

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 ∞ as any one of the constraints $g_j(x)$ approaches 0 from negative values. The two most common examples of barrier functions are:

$$B(x) = -\sum_{j=1}^{r} \ln\bigl\{-g_j(x)\bigr\}, \qquad \text{logarithmic},$$

$$B(x) = -\sum_{j=1}^{r} \frac{1}{g_j(x)}, \qquad \text{inverse}.$$

Note that both of these barrier functions are convex since the constraint functions g_j are convex. Figure 6.9.1 illustrates the form of $B(x)$.

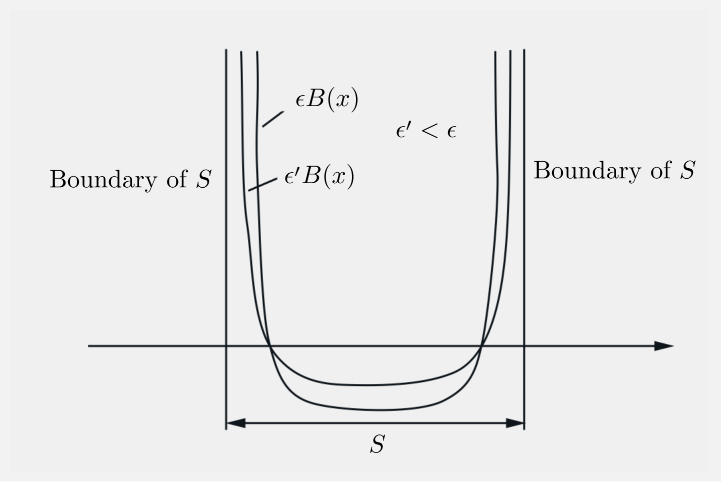

Figure 6.9.1 Form of a barrier function. The barrier term $\epsilon B(x)$ goes to zero for all interior points $x \in S$ as $\epsilon \to 0$.

The barrier method is defined by introducing a parameter sequence $\{\epsilon_k\}$ with

$$0 < \epsilon_{k+1} < \epsilon_k, \quad k = 0, 1, \ldots, \qquad \epsilon_k \to 0.$$

It consists of finding

$$x_k \in \arg\min_{x \in S} \bigl\{f(x) + \epsilon_k B(x)\bigr\}, \qquad k = 0, 1, \ldots \tag{6.235}$$

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 $X = \Re^n$, 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 (6.235) (see Fig. 6.9.2 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 $\epsilon_k \to 0$. 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 of S). Figure 6.9.2 illustrates the convergence process, and the following proposition gives the main convergence result.

Proposition 6.9.1: Every limit point of a sequence $\{x_k\}$ generated by a barrier method is a global minimum of the original constrained problem (6.234).

Proof: Let $\{\overline{x}\}$ be the limit of a subsequence $\{x_k\}_{k\in K}$. If $\overline{x} \in S$, we have $\lim_{k\to\infty,\, k\in K} \epsilon_k B(x_k) = 0$, while if $\overline{x}$ lies on the boundary of S, we have $\lim_{k\to\infty,\, k\in K} B(x_k) = \infty$. In either case we obtain

$$\liminf_{k\to\infty} \epsilon_k B(x_k) \geq 0,$$

which implies that

$$\liminf_{k\to\infty,\, k\in K} \big\{ f(x_k) + \epsilon_k B(x_k) \big\} = f(\overline{x}) + \liminf_{k\to\infty,\, k\in K} \big\{ \epsilon_k B(x_k) \big\} \geq f(\overline{x}). \tag{6.236}$$

The vector $\overline{x}$ is a feasible point of the original problem (6.234), since $x_k \in S$ and X is a closed set. If $\overline{x}$ were not a global minimum, there would exist a feasible vector x^* such that $f(x^*) < f(\overline{x})$ 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 $f(\tilde{x}) < f(\overline{x})$. We now have by the definition of x_k,

$$f(x_k) + \epsilon_k B(x_k) \leq f(\tilde{x}) + \epsilon_k B(\tilde{x}), \qquad k = 0, 1, \ldots,$$

which by taking the limit as $k \to \infty$ and $k \in K$, implies together with Eq. (6.236), that $f(\overline{x}) \leq f(\tilde{x})$. This is a contradiction, thereby proving that $\overline{x}$ 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

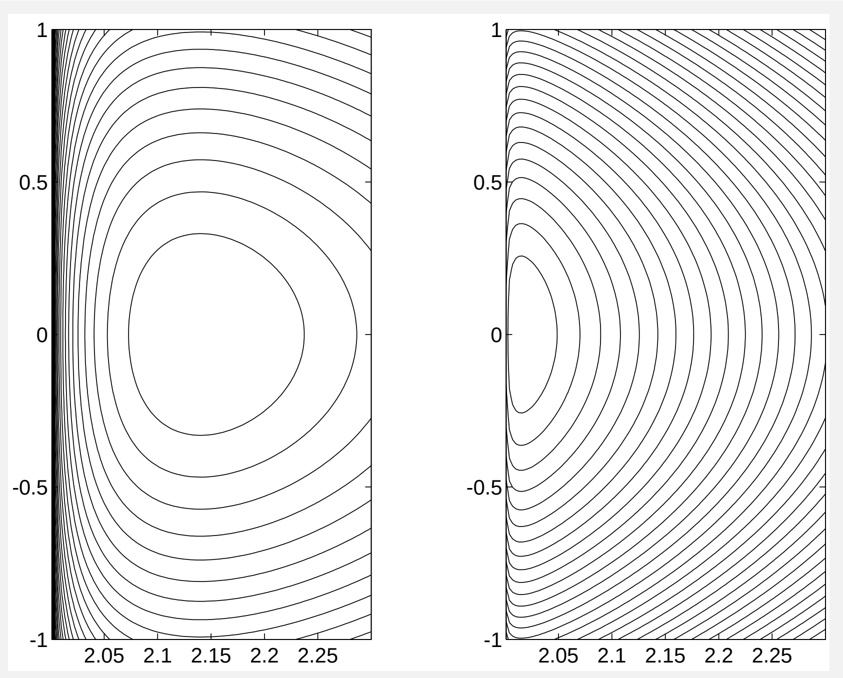

Figure 6.9.2. The convergence process of the barrier method for the two-dimensional problem

$$\text{minimize } \ f(x) = \tfrac{1}{2}\Big((x^1)^2 + (x^2)^2\Big)$$
$$\text{subject to } \ 2 \le x^1,$$

with optimal solution $x^* = (2, 0)$. For the case of the logarithmic barrier function $B(x) = -\ln(x^1 - 2)$, we have

$$x_k \in \arg\min_{x^1>2} \Big\{\tfrac{1}{2}\Big((x^1)^2 + (x^2)^2\Big) - \epsilon_k \ln(x^1 - 2)\Big\} = \Big(1 + \sqrt{1 + \epsilon_k}\,, 0\Big),$$

so as ϵ_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 $\epsilon_k \to 0$, computing x_k becomes more difficult because of ill-conditioning (the equal cost surfaces become very elongated near x_k).

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 6.9.1 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 6.9.2 we will address briefly interior point methods for conic programming problems.

6.9.1 Primal-Dual Methods for Linear Programming

Let us consider the linear program

$$\begin{aligned}&\text{minimize } \ c'x\\ &\text{subject to } \ Ax = b, \qquad x \geq 0,\end{aligned} \tag{LP}$$

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 $\epsilon > 0$,

$$x(\epsilon) \in \arg\min_{x \in S} F_\epsilon(x), \tag{6.237}$$

where

$$F_\epsilon(x) = c'x - \epsilon \sum_{i=1}^{n} \ln x^i,$$

x^i is the ith component of x and S is the interior set

$$S = \big\{x \mid Ax = b,\, x > 0\big\}.$$

We assume that S is nonempty and bounded.

Rather than directly minimizing $F_\epsilon(x)$ for small values of ϵ [cf. Eq. (6.237)], 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 ϵ_k.

(b) For every k, the pair (x_k, λ_k) is such that x_k is an interior point of the positive orthant, that is, $x_k > 0$, while λ_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

$$P_k = x_k{}' z_k + \|Ax_k - b\|, \tag{6.238}$$

is decreased to 0, where z_k is the vector of slack variables

$$z_k = c - A'\lambda_k.$$

The expression (6.238) 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 $P_{k+1} \le P_k$ (and typically $P_{k+1} < P_k$) for all k. If we can show that $P_k \to 0$, 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

$$\min_{Ax=b,\, x\ge 0} c'x = \max_{A'\lambda \le c} b'\lambda,$$

shown in Section 5.2.

Let us write the necessary and sufficient conditions for (x, λ) 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

$$c - \epsilon x^{-1} - A'\lambda = 0, \qquad Ax = b, \tag{6.239}$$

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 λ is dual feasible if and only if $z \ge 0$.

Using the vector z, we can write the first condition of Eq. (6.239) 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 (6.239) can be written in the equivalent form

$$XZe = \epsilon e, \tag{6.240}$$

$$Ax = b, \tag{6.241}$$

$$z + A'\lambda = c. \tag{6.242}$$

Given (x, λ, z) satisfying $z + A'\lambda = c$, and such that $x > 0$ and $z > 0$, a Newton iteration for solving this system is

$$x(\alpha, \epsilon) = x + \alpha\Delta x, \tag{6.243}$$

$$\lambda(\alpha, \epsilon) = \lambda + \alpha\Delta\lambda,$$

$$z(\alpha, \epsilon) = z + \alpha\Delta z,$$

where α is a stepsize such that $0 < \alpha \leq 1$ and

$$x(\alpha, \epsilon) > 0, \qquad z(\alpha, \epsilon) > 0,$$

and the Newton increment $(\Delta x, \Delta\lambda, \Delta z)$ solves the linearized version of the system (6.240)-(6.242)

$$X\Delta z + Z\Delta x = -v, \tag{6.244}$$

$$A\Delta x = b - Ax, \tag{6.245}$$

$$\Delta z + A'\Delta\lambda = 0, \tag{6.246}$$

with v defined by

$$v = XZe - \epsilon e. \tag{6.247}$$

After a straightforward calculation, the solution of the linearized system (6.244)-(6.246) can be written as

$$\Delta\lambda = \big(AZ^{-1}XA'\big)^{-1}\big(AZ^{-1}v + b - Ax\big), \tag{6.248}$$

$$\Delta z = -A'\Delta\lambda, \tag{6.249}$$

$$\Delta x = -Z^{-1}v - Z^{-1}X\Delta z.$$

Note that $\lambda(\alpha, \epsilon)$ is dual feasible, since from Eq. (6.246) 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. (6.245) we have $A(x + \Delta x) = b$.

Merit Function Improvement

We will now evaluate the changes in the constraint violation and the merit function (6.238) induced by the Newton iteration.

By using Eqs. (6.243)and (6.245), the new constraint violation is given by

$$Ax(\alpha,\epsilon)-b = Ax+\alpha A\Delta x-b = Ax+\alpha(b-Ax)-b = (1-\alpha)(Ax-b). \tag{6.250}$$

Thus, since $0<\alpha\le 1$, 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

$$g = x'z \tag{6.251}$$

after the iteration becomes

$$\begin{aligned} g(\alpha,\epsilon) &= x(\alpha,\epsilon)'z(\alpha,\epsilon) \\ &= (x+\alpha\Delta x)'(z+\alpha\Delta z) \\ &= x'z+\alpha(x'\Delta z+z'\Delta x)+\alpha^2\Delta x'\Delta z. \end{aligned} \tag{6.252}$$

From Eqs. (6.245) and (6.249) we have

$$\Delta x'\Delta z = (Ax-b)'\Delta\lambda,$$

while by premultiplying Eq. (6.244) with e' and using the definition (6.247) for v, we obtain

$$x'\Delta z+z'\Delta x = -e'v = n\epsilon - x'z.$$

By substituting the last two relations in Eq. (6.252) and by using also the expression (6.251) for g, we see that

$$g(\alpha,\epsilon) = g-\alpha(g-n\epsilon)+\alpha^2(Ax-b)'\Delta\lambda. \tag{6.253}$$

Let us now denote by P and $P(\alpha,\epsilon)$ the value of the merit function (6.238) before and after the iteration, respectively. We have by using the expressions (6.250) and (6.253),

$$\begin{aligned} P(\alpha,\epsilon) &= g(\alpha,\epsilon)+\|Ax(\alpha,\epsilon)-b\| \\ &= g-\alpha(g-n\epsilon)+\alpha^2(Ax-b)'\Delta\lambda+(1-\alpha)\|Ax-b\|, \end{aligned}$$

or

$$P(\alpha,\epsilon) = P-\alpha\big(g-n\epsilon+\|Ax-b\|\big)+\alpha^2(Ax-b)'\Delta\lambda.$$

Thus if ϵ is chosen to satisfy

$$\epsilon < \frac{g}{n}$$

and α 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

$$x_{k+1} = x(\alpha_k, \epsilon_k), \qquad \lambda_{k+1} = \lambda(\alpha_k, \epsilon_k), \qquad z_{k+1} = z(\alpha_k, \epsilon_k),$$

where α_k and ϵ_k are positive scalars such that

$$x_{k+1} > 0, \qquad z_{k+1} > 0, \qquad \epsilon_k < \frac{g_k}{n},$$

where g_k is the inner product

$$g_k = x_k{}' z_k + (Ax_k - b)' \lambda_k,$$

and α_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 α_k and ϵ_k so that the merit function is not only reduced at each iteration, but also converges to zero. Furthermore, with suitable choices of α_k and ϵ_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 α_k and ϵ_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, α_k is chosen as $\theta\tilde{\alpha}_k$, where θ is a factor very close to 1 (say 0.999), and $\tilde{\alpha}_k$ is the maximum stepsize α that guarantees that $x(\alpha, \epsilon_k) \geq 0$ and $z(\alpha, \epsilon_k) \geq 0$

$$\tilde{\alpha}_k = \min\left\{ \min_{i=1,\ldots,n} \left\{ \frac{x_k^i}{-\Delta x^i} \;\middle|\; \Delta x^i < 0 \right\}, \min_{i=1,\ldots,n} \left\{ \frac{z_k^i}{-\Delta z^i} \;\middle|\; \Delta z^i < 0 \right\} \right\}.$$

When x_k is not feasible, the choice of α_k must also be such that the merit function is improved. In some works, a different stepsize for the x update than for the (λ, z) update has been suggested. The stepsize for the x update is near the maximum stepsize α that guarantees $x(\alpha, \epsilon_k) \geq 0$, and the stepsize for the (λ, z) update is near the maximum stepsize α 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.

6.9.2 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

$$\begin{aligned} &\text{minimize} \quad c'x \\ &\text{subject to} \quad A_i x - b_i \in C_i, \ i = 1, \ldots, m, \end{aligned} \tag{6.254}$$

where $x \in \Re^n$, c is a vector in $\Re^n$, and for $i = 1, \ldots, m$, A_i is an $n_i \times n$ matrix, b_i is a vector in $\Re^{n_i}$, and C_i is the second order cone of $\Re^{n_i}$ [cf. Eq. (6.24)]. We approximate this problem with

$$\begin{aligned} &\text{minimize} \quad c'x + \epsilon_k \sum_{i=1}^{m} B_i(A_i x - b_i) \\ &\text{subject to} \quad x \in \Re^n, \end{aligned} \tag{6.255}$$

where B_i is a function defined in the interior of the second order cone C_i, and given by

$$B_i(y) = -\ln\big(y_{n_i}^2 - (y_1^2 + \cdots + y_{n_i-1}^2)\big), \qquad y \in \text{int}(C_i),$$

and $\{\epsilon_k\}$ is a positive sequence that converges to 0. Thus we have $B_i(A_i x - b_i) \to \infty$ as $A_i x - b_i$ approaches the boundary of C_i.

Similar to Prop. 6.9.1, it can be shown that if x_k is an optimal solution of the approximating problem (6.255), 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 (6.255) should not be solved exactly. In the most efficient methods, one or more Newton steps corresponding to a given value ϵ_k are performed, and then the value of ϵ_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 ϵ_k, and the subsequent reduction of ϵ_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 ϵ_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 Vanderberghe [BoV04], and sources quoted there for further details.

There is a similar interior point method for the dual SDP involving the multiplier vector $\lambda = (\lambda_1, \ldots, \lambda_m)$:

$$\begin{aligned} &\text{maximize} \quad b'\lambda \\ &\text{subject to} \quad C - (\lambda_1 A_1 + \cdots + \lambda_m A_m) \in D, \end{aligned} \tag{6.256}$$

where D is the cone of positive semidefinite matrices [cf. Eq. (6.30)]. It consists of solving the problem

$$\begin{aligned} &\text{maximize} \quad b'\lambda + \epsilon_k \ln\big(\det(C - \lambda_1 A_1 - \cdots - \lambda_m A_m)\big) \\ &\text{subject to} \quad \lambda \in \Re^m, \quad C - \lambda_1 A_1 - \cdots - \lambda_m A_m \in \text{int}(D), \end{aligned} \tag{6.257}$$

where $\{\epsilon_k\}$ is a positive sequence that converges to 0. Here, we should use a starting point such that $C - \lambda_1 A_1 - \cdots - \lambda_m A_m$ is positive definite, and Newton's method should ensure that the iterates keep $C - \lambda_1 A_1 - \cdots - \lambda_m A_m$ 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 (6.257), then every limit point of $\{x_k\}$ is an optimal solution of the original problem (6.256).

We finally note that there are primal-dual interior point methods for conic programming problems, which bear similarity with the one given in Section 6.9.1 for linear programming. Again, we refer to the specialized literature for further details and a complexity analysis.

6.10 GRADIENT PROJECTION - OPTIMAL COMPLEXITY ALGORITHMS

In this section we focus on problems of the form

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

where $f : \Re^n \mapsto \Re$ is convex and X is a closed convex set. We will discuss algorithms that have good performance guarantees, in the sense that they require a relatively small number of iterations to achieve a given optimal solution tolerance. These algorithms rely in part on differentiability properties of f, and cost function descent ideas.

6.10.1 Gradient Projection Methods

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,

$$\big\|\nabla f(x) - \nabla f(y)\big\| \le L\,\|x-y\|, \qquad \forall\ x, y \in X, \tag{6.258}$$

and that the optimal value $f^* = \inf_{x\in X} f(x)$ is finite. We will consider the gradient projection method

$$x_{k+1} = P_X\big(x_k - \alpha\nabla f(x_k)\big), \tag{6.259}$$

where $\alpha > 0$ is a constant stepsize. This method, briefly discussed in Section 6.2, is the specialization of the subgradient method of Section 6.3, for the case where f is differentiable.

Let us introduce the linear approximation of f based on the gradient at x, given by

$$\ell(y;x) = f(x) + \nabla f(x)'(y-x), \qquad \forall\ x, y \in \Re^n, \tag{6.260}$$

An interesting observation is that the gradient projection iterate can be alternatively written in terms of $\ell(x;x_k)$ as

$$\begin{aligned} x_{k+1} &= P_X\big(x_k - \alpha\nabla f(x_k)\big)\\ &= \arg\min_{x\in X} \big\|x - \big(x_k - \alpha\nabla f(x_k)\big)\big\|^2\\ &= \arg\min_{x\in X} \big\{\|x-x_k\|^2 + 2\alpha\nabla f(x_k)'(x-x_k) + \alpha^2\|\nabla f(x_k)\|^2\big\}\\ &= \arg\min_{x\in X} \big\{f(x_k) + 2\alpha\nabla f(x_k)'(x-x_k) + \|x-x_k\|^2\big\}\\ &= \arg\min_{x\in X} \left\{\ell(x;x_k) + \frac{1}{2\alpha}\|x-x_k\|^2\right\}. \end{aligned} \tag{6.261}$$

Thus x_{k+1} can be viewed as the result of a proximal iteration on $\ell(x;x_k)$, and indeed the gradient projection method and the proximal algorithm coincide when the cost function f is linear.

We will now show that the gradient projection method has the convergence property $f(x_k) \to f^*$ for any starting point x_0, provided α is sufficiently small (this is a stronger property than what can be proved for the subgradient method, which requires a diminishing stepsize for convergence, cf. Section 6.3.1). To this end, we show the following proposition, which will be used on several occasions in the analysis of this section.

Proposition 6.10.1: Let $f : \Re^n \mapsto \Re$ be a continuously differentiable function, with gradient satisfying the Lipschitz condition (6.258). Then for all $x, y \in X$, we have

$$f(y) \leq \ell(y;x) + \frac{L}{2}\|y-x\|^2, \tag{6.262}$$

Proof: Let t be a scalar parameter and let $g(t) = f\big(x + t(y-x)\big)$. The chain rule yields $(dg/dt)(t) = \nabla f\big(x + t(y-x)\big)'(y-x)$. Thus, we have

$$\begin{aligned}
f(y) - f(x) &= g(1) - g(0) \\
&= \int_0^1 \frac{dg}{dt}(t)\,dt \\
&= \int_0^1 (y-x)'\nabla f\big(x+t(y-x)\big)\,dt \\
&\leq \int_0^1 (y-x)'\nabla f(x)\,dt + \left|\int_0^1 (y-x)'\big(\nabla f\big(x+t(y-x)\big) - \nabla f(x)\big)\,dt\right| \\
&\leq \int_0^1 (y-x)'\nabla f(x)\,dt + \int_0^1 \|y-x\|\cdot\|\nabla f\big(x+t(y-x)\big) - \nabla f(x)\|dt \\
&\leq (y-x)'\nabla f(x) + \|y-x\|\int_0^1 Lt\|y-x\|\,dt \\
&= (y-x)'\nabla f(x) + \frac{L}{2}\|y-x\|^2
\end{aligned}$$

thereby proving Eq. (6.262). **Q.E.D.**

With the preceding proposition, we can show a basic cost improvement inequality, which will be the key to the convergence analysis. From the projection theorem and the definition (6.259) of x_{k+1}, we have

$$\big(x_k - \alpha\nabla f(x_k) - x_{k+1}\big)'(x - x_{k+1}) \leq 0, \qquad \forall\, x \in X, \tag{6.263}$$

so that by setting $x = x_{k+1}$,

$$\nabla f(x_k)'(x_{k+1} - x_k) \leq -\frac{1}{\alpha}\big\|x_{k+1} - x_k\big\|^2.$$

Using this relation together with Eq. (6.262), we obtain

$$\begin{aligned}
f(x_{k+1}) &\leq \ell(x_{k+1};x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\
&= f(x_k) + \nabla f(x_k)'(x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 \qquad (6.264) \\
&\leq f(x_k) - \left(\frac{1}{\alpha} - \frac{L}{2}\right)\|x_{k+1} - x_k\|^2,
\end{aligned}$$

so the gradient projection method (6.259) reduces the cost function value at each iteration, provided the stepsize lies in the range $\alpha \in \big(0, \frac{2}{L}\big)$.

Moreover, we can show that any limit point of $\{x_k\}$ is an optimal solution. Indeed, if $\bar{x}$ is the limit of a subsequence $\{x_k\}_{\mathcal{K}}$, from Eq. (6.264), we have $f(x_k) \to f(\bar{x})$ and $\|x_{k+1} - x_k\| \to 0$, implying that $\{x_{k+1}\}_{\mathcal{K}} \to \bar{x}$. Taking the limit in Eq. (6.263), as $k \to \infty$, $k \in \mathcal{K}$, it follows that

$$\big(\bar{x} - \alpha \nabla f(\bar{x}) - \bar{x}\big)'(x - \bar{x}) \leq 0, \qquad \forall\, x \in X,$$

or

$$\nabla f(\bar{x})'(x - \bar{x}) \geq 0, \qquad \forall\, x \in X.$$

This implies that $\bar{x}$ minimizes f over X.

We now turn to estimating the number of iterations needed to attain the optimal cost f^* within a given tolerance. Let X^* be the set of minima of f over X, and denote

$$d(x) = \inf_{x^* \in X^*} \|x - x^*\|, \qquad x \in \Re^n.$$

We first consider the case where the stepsize is $\alpha = 1/L$ at all iterations, and then adapt the proof for a more practical stepsize selection method that does not require knowledge of L.

Proposition 6.10.2: Let $f : \Re^n \mapsto \Re$ be a convex differentiable function and X be a closed convex set. Assume that ∇f satisfies the Lipschitz condition (6.258), and that the set of minima X^* of f over X is nonempty. Let $\{x_k\}$ be a sequence generated by the gradient projection method (6.259) with stepsize $\alpha = 1/L$. Then $\lim_{k\to\infty} d(x_k) = 0$, and

$$f(x_k) - f^* \leq \frac{L d(x_0)^2}{2k}, \qquad k = 1, 2, \ldots. \tag{6.265}$$

Proof: Using Eq. (6.262), we have

$$f(x_{k+1}) \leq \ell(x_{k+1}; x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2. \tag{6.266}$$

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. (6.266),

$$\begin{aligned} f(x_{k+1}) &\le \ell(x_{k+1}; x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &\le \ell(x^*; x_k) + \frac{L}{2}\|x^* - x_k\|^2 - \frac{L}{2}\|x^* - x_{k+1}\|^2 \\ &\le f(x^*) + \frac{L}{2}\|x^* - x_k\|^2 - \frac{L}{2}\|x^* - x_{k+1}\|^2, \end{aligned}$$

so denoting $e_k = f(x_k) - f^*$, we have

$$\begin{aligned} \frac{L}{2}\|x^* - x_{k+1}\|^2 &\le \frac{L}{2}\|x^* - x_k\|^2 - e_{k+1} \\ &\le \frac{L}{2}\|x^* - x_0\|^2 - (e_1 + \cdots + e_{k+1}) \\ &\le \frac{L}{2}d(x_0)^2 - (k+1)e_{k+1}, \end{aligned}$$

where the last inequality uses the fact $e_0 \ge e_1 \ge \cdots \ge e_{k+1}$ [cf. Eq. (6.264)]. This proves Eq. (6.265). Also the preceding relation together with the fact $d(x_{k+1}) \le \|x^* - x_{k+1}\|$ implies that $d(x_k) \to 0$. **Q.E.D.**

The preceding proposition shows that the gradient projection method requires k iterations to achieve the optimal value within an $O(1/k)$ tolerance [cf. Eq. (6.265)]. This was proved for the case where we can choose the stepsize as $\alpha = 1/L$, which is a little unrealistic since L is generally unknown. However, there is a practical procedure for selecting and adjusting α so that $\lim_{k\to\infty} d(x_k) = 0$ and a similar $O(1/k)$ error bound holds, even if the value of L is unknown. The key is that α should be adjusted to a value α_k at iteration k, so that the following analog of Eq. (6.266) holds

$$f(x_{k+1}) \le \ell(x_{k+1}; x_k) + \frac{1}{2\alpha_k}\|x_{k+1} - x_k\|^2, \tag{6.267}$$

which also implies that $f(x_{k+1}) \le f(x_k)$.† In particular, we may use some arbitrary initial stepsize $\alpha_0 > 0$, and generate iterates according to

$$x_{k+1} = P_X\big(x_k - \alpha_k \nabla f(x_k)\big), \tag{6.268}$$

† To see this, note that Eqs. (6.260) and (6.267) imply that

$$f(x_{k+1}) \le f(x_k) + \nabla f(x_k)'(x_{k+1} - x_k) + \frac{1}{2\alpha_k}\|x_{k+1} - x_k\|^2 \le f(x_k),$$

where the last inequality holds by Eq. (6.261).

as long as the condition (6.267) is satisfied. As soon as Eq. (6.267) is violated at some iteration k, we reduce α_k by a certain factor, and repeat the iteration as many times as necessary for Eq. (6.267) to hold. Because the condition (6.267) will be satisfied as soon as $\alpha_k \leq 1/L$, if not earlier, only a finite number of stepsize reductions will be needed during the entire algorithm. It can be seen that with Eq. (6.267) in place of Eq. (6.266), the proof of the preceding proposition carries through for the gradient projection method (6.268) that uses the stepsize procedure just described in place of the constant stepsize $\alpha = 1/L$.

Iteration Complexity Issues

Let us now consider some general computational complexity issues relating to the optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X, \end{array}$$

where $f : \Re^n \mapsto \Re$ is convex and X is a closed convex set. We assume that there exists an optimal solution. We will be interested in algorithms that have good performance guarantees, in the sense that they require a relatively low number of iterations (in the worst case) to achieve a given optimal solution tolerance, and we will evaluate the performance of gradient projection within this context.

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 ϵ of the optimal. If we can show that any 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 ϵ, as well as the problem data and the starting point x_0, we say that the method has *iteration complexity* $O\big(N(\epsilon)\big)$.

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 iter-

† The reader should be warned that it may not be safe to speculate with confidence on the relative advantages of the various gradient and subgradient methods of this and the next section, and to compare them with Newton-like methods, based on the complexity estimates that we provide. The reason is that our analysis does not take into account the special structure that is typically present in large-scale problems, while our complexity estimates involve unknown constants, whose size may affect the comparisons between various methods.

ation 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. 6.3.3). On the other hand, Prop. 6.10.2, shows that the gradient projection method has iteration complexity $O(1/\epsilon)$, when applied to *differentiable* problems with Lipschitz continuous cost gradient. The following example shows that this estimate cannot be improved.

Example 6.10.1:

Consider the unconstrained minimization of the scalar function f given by

$$f(x) = \begin{cases} \frac{c}{2}|x|^2 & \text{if } |x| \leq \epsilon, \\ c\epsilon|x| - \frac{c\epsilon^2}{2} & \text{if } |x| > \epsilon, \end{cases}$$

with $\epsilon > 0$ (cf. Fig. 6.10.1). Here the constant in the Lipschitz condition (6.258) 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 ϵ-neighborhood of $x^* = 0$ is $|x_0|/\epsilon$. The number of iterations to get to within ϵ of the optimal cost $f^* = 0$, is also proportional to $1/\epsilon$.

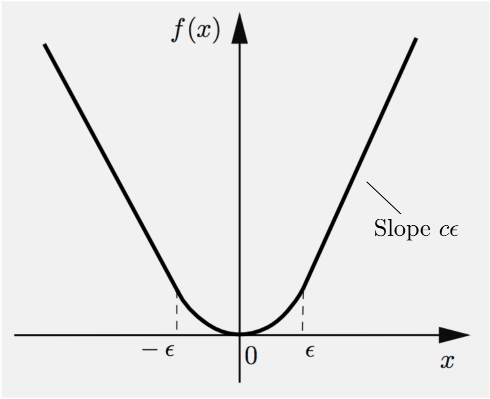

Figure 6.10.1. The scalar cost function f of Example 6.10.1. It is quadratic for $|x| \leq \epsilon$ and linear for $|x| > \epsilon$.

In the next section, we will discuss a variant of the gradient projection method that employs an intricate extrapolation device, and has the improved iteration complexity of $O\big(1/\sqrt{\epsilon}\big)$. It can be shown that $O\big(1/\sqrt{\epsilon}\big)$ 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). This is what we mean by calling this variant of gradient projection an "optimal" algorithm. On the other hand this algorithm ignores any special problem structure, such as when f is the sum of a large number of components, so for special classes of problems, it may not be competitive with methods that exploit structure, such as the incremental methods of the preceding section.

6.10.2 Gradient Projection with Extrapolation

We will now discuss a method for improving the iteration complexity of the gradient projection method. A closer examination of the preceding Example 6.10.1 suggests that while a stepsize less that $2/c$ 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

$$x_{k+1} = x_k - \alpha\nabla f(x_k) + \beta(x_k - x_{k-1}),$$

and adds the extrapolation term $\beta(x_k - x_{k-1})$ to the gradient increment, where $x_{-1} = x_0$ and β 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 as follows:

$$\begin{aligned} y_k &= x_k + \beta(x_k - x_{k-1}), && \text{(extrapolation step)},\\ x_{k+1} &= y_k - \alpha\nabla f(y_k), && \text{(gradient step)}. \end{aligned} \tag{6.269}$$

When applied to the function of the preceding Example 6.10.1, the method converges to the optimum, and reaches a neighborhood of the optimum more quickly: it can be verified that for a starting point $x_0 >> 1$ and $x_k > \epsilon$, it has the form $x_{k+1} = x_k - \epsilon_k$, with $\epsilon \leq \epsilon_k < \epsilon/(1-\beta)$. However, the method still has an $O(1/\epsilon)$ iteration complexity, since for $x_0 >> 1$, the number of iterations needed to obtain $x_k < \epsilon$ is $O\big((1-\beta)/\epsilon\big)$. This can be seen by verifying that

$$x_{k+1} - x_k = \beta(x_k - x_{k-1}) - \epsilon,$$

so when $x_0 >> 1$, we have approximately $x_{k+1} - x_k \approx \epsilon/(1-\beta)$.

It turns out that a better iteration complexity is possible with a similar scheme that involves a reversal of the order of gradient iteration and extrapolation, and more vigorous extrapolation. In this scheme the constant extrapolation factor β is replaced with a variable factor β_k that converges to 1 at a properly selected rate. Unfortunately, it is very difficult to obtain strong intuition about the mechanism by which this remarkable phenomenon occurs.

An Optimal Algorithm for Differentiable Cost

We will consider a constrained version of the gradient/extrapolation method (6.269) for the problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in X, \end{aligned} \tag{6.270}$$

where $f : \Re^n \mapsto \Re$ is convex and differentiable, and X is a closed convex set. We assume that f has Lipschitz continuous gradient [cf. Eq. (6.258)], and we denote

$$d(x) = \inf_{x^* \in X^*} \|x - x^*\|, \qquad x \in \Re^n,$$

where X^* is the set of minima of f over X.

The method has the form

$$\begin{aligned} y_k &= x_k + \beta_k(x_k - x_{k-1}), && \text{(extrapolation step)}, \\ x_{k+1} &= P_X\big(y_k - \alpha\nabla f(y_k)\big), && \text{(gradient projection step)}, \end{aligned} \tag{6.271}$$

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 β_k, the method has iteration complexity $O\big(1/\sqrt{\epsilon}\big)$. We will assume that

$$\beta_k = \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}}, \qquad k = 0, 1, \ldots \tag{6.272}$$

where the sequence $\{\theta_k\}$ satisfies $\theta_0 = \theta_1 \in (0,1]$, and

$$\frac{1-\theta_{k+1}}{\theta_{k+1}^2} \le \frac{1}{\theta_k^2}, \qquad \theta_k \le \frac{2}{k+2}, \qquad k = 0, 1, \ldots \tag{6.273}$$

One possible choice is

$$\beta_k = \begin{cases} 0 & \text{if } k = 0, \\ \frac{k-1}{k+2} & \text{if } k = 1, 2, \ldots, \end{cases} \qquad \theta_k = \begin{cases} 1 & \text{if } k = -1, \\ \frac{2}{k+2} & \text{if } k = 0, 1, \ldots \end{cases}$$

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 6.10.3: Let $f : \Re^n \mapsto \Re$ be a convex differentiable function and X be a closed convex set. Assume that ∇f satisfies the Lipschitz condition (6.258), and that the set of minima X^* of f over X is nonempty. Let $\{x_k\}$ be a sequence generated by the algorithm (6.271), where $\alpha = 1/L$ and β_k satisfies Eqs. (6.272)-(6.273). Then $\lim_{k\to\infty} d(x_k) = 0$, and

$$f(x_k) - f^* \le \frac{2L}{(k+1)^2} d(x_0)^2, \qquad k = 1, 2, \ldots.$$

Proof: We introduce the sequence

$$z_k = x_{k-1} + \theta_{k-1}^{-1}(x_k - x_{k-1}), \qquad k = 0, 1, \ldots, \tag{6.274}$$

where $x_{-1} = x_0$, so that $z_0 = x_0$. We note that by using Eqs. (6.271), (6.272), z_k can also be rewritten as

$$z_k = x_k + \theta_k^{-1}(y_k - x_k), \qquad k = 1, 2, \ldots, \tag{6.275}$$

Fix $k \geq 0$ and $x^* \in X^*$, and let

$$y^* = (1 - \theta_k)x_k + \theta_k x^*.$$

Using Eq. (6.262), we have

$$f(x_{k+1}) \leq \ell(x_{k+1}; y_k) + \frac{L}{2}\|x_{k+1} - y_k\|^2, \tag{6.276}$$

where

$$\ell(u; w) = f(w) + \nabla f(w)'(u - w), \qquad \forall\ u, w \in \Re^n,$$

[cf. Eq. (6.260)]. 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 $y \in X$ [cf. Eq. (6.261)], 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. (6.276), we obtain

$$\begin{aligned}
f(x_{k+1}) &\leq \ell(y^*; y_k) + \frac{L}{2}\|y^* - y_k\|^2 - \frac{L}{2}\|y^* - x_{k+1}\|^2 \\
&= \ell\big((1 - \theta_k)x_k + \theta_k x^*; y_k\big) + \frac{L}{2}\|(1 - \theta_k)x_k + \theta_k x^* - y_k\|^2 \\
&\quad - \frac{L}{2}\|(1 - \theta_k)x_k + \theta_k x^* - x_{k+1}\|^2 \\
&= \ell\big((1 - \theta_k)x_k + \theta_k x^*; y_k\big) + \frac{\theta_k^2 L}{2}\|x^* + \theta_k^{-1}(x_k - y_k) - x_k\|^2 \\
&\quad - \frac{\theta_k^2 L}{2}\|x^* + \theta_k^{-1}(x_k - x_{k+1}) - x_k\|^2 \\
&= \ell\big((1 - \theta_k)x_k + \theta_k x^*; y_k\big) + \frac{\theta_k^2 L}{2}\|x^* - z_k\|^2 \\
&\quad - \frac{\theta_k^2 L}{2}\|x^* - z_{k+1}\|^2 \\
&\leq (1 - \theta_k)\ell(x_k; y_k) + \theta_k \ell(x^*; y_k) + \frac{\theta_k^2 L}{2}\|x^* - z_k\|^2 \\
&\quad - \frac{\theta_k^2 L}{2}\|x^* - z_{k+1}\|^2,
\end{aligned}$$

where the last equality follows from Eqs. (6.274) and (6.275), and the last inequality follows from the convexity of $\ell(\cdot;y_k)$. Using the inequality

$$\ell(x_k;y_k) \le f(x_k),$$

we have

$$f(x_{k+1}) \le (1-\theta_k)f(x_k)+\theta_k\ell(x^*;y_k)+\frac{\theta_k^2L}{2}\|x^*-z_k\|^2-\frac{\theta_k^2L}{2}\|x^*-z_{k+1}\|^2.$$

Finally, by rearranging terms, we obtain

$$\frac{1}{\theta_k^2}\big(f(x_{k+1})-f^*\big)+\frac{L}{2}\|x^*-z_{k+1}\|^2$$
$$\le \frac{1-\theta_k}{\theta_k^2}\big(f(x_k)-f^*\big)+\frac{L}{2}\|x^*-z_k\|^2-\frac{f^*-\ell(x^*;y_k)}{\theta_k}.$$

By adding this inequality for $k=0,1,\ldots,$ while using the inequality

$$\frac{1-\theta_{k+1}}{\theta_{k+1}^2} \le \frac{1}{\theta_k^2},$$

we obtain

$$\frac{1}{\theta_k^2}\big(f(x_{k+1})-f^*\big)+\sum_{i=0}^{k}\frac{f^*-\ell(x^*;y_i)}{\theta_k} \le \frac{L}{2}\|x^*-z_0\|^2.$$

Using the facts $x_0=z_0$, $f^*-\ell(x^*;y_i)\ge 0$, and $\theta_k\le 2/(k+2)$, and taking the minimum over all $x^*\in X^*$, we obtain

$$f(x_{k+1})-f^* \le \frac{2L}{(k+2)^2}d(x_0)^2,$$

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

$$f(x_{k+1}) \le \ell(x_{k+1};y_k)+\frac{1}{2\alpha}\|x_{k+1}-y_k\|^2, \tag{6.277}$$

[cf. Eq. (6.276)] is satisfied. As soon as this condition is violated at some iteration, α is reduced by a certain factor, and the iteration is repeated as many times as necessary for Eq. (6.277) to hold. Once α is reduced below the level $1/L$, the test (6.277) 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 $O\big(\sqrt{L/\epsilon}\big)$.

6.10.3 Nondifferentiable Cost – Smoothing

The preceding analysis applies to differentiable cost functions. However, it can be applied to cases where f is real-valued and 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 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 ϵ-approximation whose gradient is Lipschitz continuous with constant $L = O(1/\epsilon)$. By applying the optimal method given earlier, we obtain an ϵ-optimal solution with iteration complexity $O(\sqrt{L/\epsilon}) = O(1/\epsilon)$.

We will consider the smoothing technique for the special class of convex functions $f_0 : \Re^n \mapsto \Re$ of the form

$$f_0(x) = \max_{u \in U} \big\{ u'Ax - \phi(u) \big\}, \tag{6.278}$$

where U is a convex and compact subset of $\Re^m$, $\phi : U \mapsto \Re$ 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 (6.278) is quite broad. We introduce a function $p : \Re^m \mapsto \Re$ 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 σ, 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

$$f_\epsilon(x) = \max_{u \in U} \big\{ u'Ax - \phi(u) - \epsilon p(u) \big\}, \qquad x \in \Re^n, \tag{6.279}$$

and note that f_ϵ is a uniform approximation of f_0 in the sense that

$$f_\epsilon(x) \leq f_0(x) \leq f_\epsilon(x) + p^*\epsilon, \qquad \forall\, x \in \Re^n, \tag{6.280}$$

where

$$p^* = \max_{u \in U} p(u).$$

The following proposition shows that f_ϵ is also smooth and its gradient is Lipschitz continuous with Lipschitz constant that is proportional to $1/\epsilon$.

Proposition 6.10.4: For all $\epsilon > 0$, the function f_ϵ is convex and differentiable over $\Re^n$, 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. (6.279). Furthermore, we have

$$\big\|\nabla f_\epsilon(x) - \nabla f_\epsilon(y)\big\| \leq \frac{\|A\|^2}{\epsilon\sigma}\|x - y\|, \qquad \forall\ x, y \in \Re^n.$$

Proof: We first note that the maximum in Eq. (6.279) is uniquely attained in view of the strong convexity of p (which implies that p is strictly convex). Furthermore, f_ϵ is equal to $f^\star(A'x)$, where $f^\star$ is the conjugate of the function $\phi(u) + \epsilon p(u) + \delta_U(u)$, with δ_U being the indicator function of U. It follows that f_ϵ 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 $x, y \in \Re^n$. From the subgradient inequality, we have

$$\phi(y) - \phi(x) \geq g_x'\big(u_\epsilon(y) - u_\epsilon(x)\big), \qquad \phi(x) - \phi(y) \geq g_y'\big(u_\epsilon(x) - u_\epsilon(y)\big),$$

so by adding these two inequalities, we obtain

$$(g_x - g_y)'\big(u_\epsilon(x) - u_\epsilon(y)\big) \geq 0. \tag{6.281}$$

By using the optimality condition for the maximization (6.279), we have

$$\Big(Ax - g_x - \epsilon\nabla p\big(u_\epsilon(x)\big)\Big)'\big(u_\epsilon(y) - u_\epsilon(x)\big) \leq 0,$$

$$\Big(Ay - g_y - \epsilon\nabla p\big(u_\epsilon(y)\big)\Big)'\big(u_\epsilon(x) - u_\epsilon(y)\big) \leq 0,$$

where g_x and g_y are subgradients of ϕ at $u_\epsilon(x)$ and $u_\epsilon(y)$, respectively. Adding the two inequalities, and using the convexity of ϕ and the strong convexity of p, we obtain

$$\begin{aligned}(x-y)'A'\big(u_\epsilon(x)-u_\epsilon(y)\big) &\geq \Big(g_x - g_y + \epsilon\Big(\nabla p\big(u_\epsilon(x)\big) - \nabla p\big(u_\epsilon(y)\big)\Big)'\\ &\qquad \big(u_\epsilon(x)-u_\epsilon(y)\big)\\ &\geq \epsilon\Big(\nabla p\big(u_\epsilon(x)\big) - \nabla p\big(u_\epsilon(y)\big)\Big)'\big(u_\epsilon(x)-u_\epsilon(y)\big)\\ &\geq \epsilon\sigma\big\|u_\epsilon(x)-u_\epsilon(y)\big\|^2,\end{aligned}$$

where for the second inequality we used Eq. (6.281), and for the third inequality we used a standard property of strongly convex functions (see the Exercises for Chapter 1). Thus,

$$\begin{aligned}\big\|\nabla f_\epsilon(x)-\nabla f_\epsilon(y)\big\|^2 &= \big\|A'\big(u_\epsilon(x)-u_\epsilon(y)\big)\big\|^2\\ &\leq \|A'\|^2\,\big\|u_\epsilon(x)-u_\epsilon(y)\big\|^2\\ &\leq \frac{\|A'\|^2}{\epsilon\sigma}(x-y)'A'\big(u_\epsilon(x)-u_\epsilon(y)\big)\\ &\leq \frac{\|A'\|^2}{\epsilon\sigma}\|x-y\|\,\big\|A'\big(u_\epsilon(x)-u_\epsilon(y)\big)\big\|\\ &= \frac{\|A\|^2}{\epsilon\sigma}\|x-y\|\,\big\|\nabla f_\epsilon(x)-\nabla f_\epsilon(y)\big\|,\end{aligned}$$

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. (6.278), and $F : \Re^n \mapsto \Re$ is convex and differentiable, with gradient satisfying the Lipschitz condition

$$\big\|\nabla F(x) - \nabla F(y)\big\| \leq L\,\|x-y\|, \qquad \forall\, x,y \in X. \tag{6.282}$$

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. (6.280)], and has Lipschitz continuous gradient with Lipschitz constant $L+L_\epsilon = O(1/\epsilon)$. Thus, by applying the algorithm (6.271) and by using Prop. 6.10.3, we see that we can obtain a solution $\tilde{x} \in X$ such that $f(\tilde{x}) \leq f^* + p^*\epsilon$ with

$$O\big(\sqrt{(L+\|A\|^2/\epsilon\sigma)/\epsilon}\,\big) = O(1/\epsilon)$$

iterations.

6.10.4 Proximal Gradient Methods

We will now discuss briefly a method that combines the gradient projection and proximal algorithms, and contains both as special cases. The method applies to the special class of problems given by

$$\begin{aligned} &\text{minimize} && f(x) + h(x) \\ &\text{subject to} && x \in X, \end{aligned} \tag{6.283}$$

where $f : \Re^n \mapsto \Re$ and $h : \Re^n \mapsto \Re$ are convex functions, and X is a closed convex set. In addition h is assumed differentiable and Lipschitz continuous with Lipschitz constant L.

The proximal gradient method is given by

$$x_{k+1} \in \arg\min_{x \in X} \left\{ f(x) + \ell(x; x_k) + \frac{1}{2\alpha} \|x - x_k\|^2 \right\}, \tag{6.284}$$

where α is a positive scalar and $\ell(x; x_k)$ is the linear approximation of h at x_k, given by

$$\ell(x; x_k) = h(x_k) + \nabla h(x_k)'(x - x_k), \qquad \forall\, x \in \Re^n,$$

[cf. Eq. (6.260)]. It can be seen that if $h(x) \equiv 0$, the method reduces to the proximal algorithm, while if $f(x) \equiv 0$, the method reduces to the gradient projection method [cf. Eq. (6.261)].

A key fact about the proximal gradient method is that it improves the cost function value at each iteration (unless the current iterate is optimal). Indeed, by using the inequality

$$h(y) \leq \ell(y; x) + \frac{L}{2} \|y - x\|^2, \qquad \forall\, x, y \in X,$$

(cf. Prop. 6.10.1) we have for all $\alpha \in (0, 1/L]$,

$$\begin{aligned} f(x_{k+1}) + h(x_{k+1}) &\leq f(x_{k+1}) + \ell(x_{k+1}; x_k) + \frac{L}{2} \|x_{k+1} - x_k\|^2 \\ &\leq f(x_{k+1}) + \ell(x_{k+1}; x_k) + \frac{1}{2\alpha} \|x_{k+1} - x_k\|^2 \\ &\leq f(x_k) + \ell(x_k; x_k) \\ &= f(x_k) + h(x_k), \end{aligned}$$

where the last inequality follows from the definition (6.284) of the algorithm. Actually, by using the result of Exercise 6.19(b) we can show that

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

so by strengthening the last inequality of the preceding calculation, we obtain the stronger estimate

$$f(x_{k+1}) + h(x_{k+1}) \leq f(x_k) + h(x_k) - \frac{1}{2\alpha}\|x_k - x_{k+1}\|^2.$$

The preceding cost reduction property is the key to the convergence analysis of the algorithm, see e.g., the papers by Beck and Teboulle [BeT09], [BeT10], which also provide a survey of the applications of the algorithm, and give extensive references to the relevant literature.

Note that the proximal gradient method can be written as

$$z_k = x_k - \alpha\nabla h(x_k),$$

$$x_{k+1} \in \arg\min_{x\in X}\left\{f(x) + \frac{1}{2\alpha}\|x - z_k\|^2\right\}.$$

In this form the method bears a resemblance with one of the incremental subgradient-proximal methods of Section 6.7.1, applied to the case of two component functions [cf. Eqs. (6.200)-(6.201)]. However, there are some important differences. The proximal gradient method critically relies of the differentiability of h for its validity with a constant stepsize, and does not readily extend to the case where the cost function is the sum of more than two components. Still, however, the motivation for the proximal gradient and the incremental subgradient-proximal methods is similar: they both use linearization to facilitate the handling of components that complicate the application of the proximal algorithm.

Finally, let us note that there are versions of the proximal gradient algorithm that use extrapolation, similar to the one for the optimal algorithm of Section 6.10.2, and have complexity that is comparable to the one of that algorithm (see [BeT09], [BeT10]).

6.11 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]).

One may view ϵ-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 [NeB10]. The first proposal of an incremental subgradient method was in the paper by 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, and established the superiority of the randomized over the cyclic selection rule for component selection. Reference [NeB01b] provides additional convergence rate results for a constant stepsize, including a linear convergence rate to a neighborhood of the solution set under a strong convexity condition.

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].

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 Bertsekas [Ber83], Tsitsiklis, Bertsekas, and Athans [TBA86], and Bertsekas and Tsitsiklis [BeT89]). This analysis applies also to the incremental proximal algorithms of Section 6.7.

Cutting plane methods were introduced by Cheney and Goldstein [ChG59], and by Kelley [Kel60]. For analysis of related methods, see Ruszczynski [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 [BeY11], which contains a detailed convergence analysis. The polyhedral approximation method for conical constraints of Section 6.4.6 is also new.

The proximal 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], for both quadratic and nonquadratic proximal terms. A generalization of the proximal 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 Lemarechal [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.

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 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.

The incremental subgradient-proximal algorithms of Section 6.7.1, including the incremental constraint projection algorithm (6.212)-(6.214), were first proposed and analyzed in Bertsekas [Ber10a], [Ber10b]. The incremental Augmented Lagrangian scheme of Section 6.7 is new. The incremental constraint projection algorithm of Section 6.7.2 was proposed and analyzed by Wang and Bertsekas [WaB13]. Similar algorithms that involve incremental constraint projections, but not incremental cost function iterations, were proposed and analyzed earlier by Nedić [Ned11].

There are proximal-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 methods involving nonquadratic augmented Lagrangians (see also Kort and Bertsekas [KoB76], and Bertsekas [Ber82]). The use of the Bregman distance function has been the focus of much attention. 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 [Eck93], [Eck98], Iusem, Svaiter, and Teboulle [IST94], Iusem and Teboulle [IuT95], Auslender, Cominetti, and Haddou [AHR97], Polyak and Teboulle [PoT97], Iusem [Ius99], Facchinei and Pang [FaP03], Auslender and Teboulle [AuT03], Cruz Neto et. al. [CFI07], and Yin et. al. [YOG08].

Interior point methods date to the work of 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 Vanderberghe [WSV00] contains a collection of survey articles on semidefinite programming. The book by

Boyd and Vanderberghe [BoV04] describes many applications, and contains a lot of material and references.

The ideas of the iteration complexity analysis and algorithms of Section 6.10.2 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. 6.10.4 for the more general case, where p is convex but not necessarily differentiable, and analyzes several important special cases.

There have been several proposals of combinations of gradient and proximal methods for minimizing the sum of two functions (or more generally, finding a zero of the sum of two nonlinear operators). These methods have a long history, dating to the splitting algorithms of Lions and Mercier [LiM79], Passty [Pas79], and Spingarn [Spi85], and have received renewed attention more recently (see Beck and Teboulle [BeT09], [BeT10], and the references they give to specialized algorithms).

EXERCISES

6.1 (Minimizing the Sum or the Maximum of Norms [LVB98])

Consider the problems

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{p} \|F_i x + g_i\| \\ &\text{subject to} \quad x \in \Re^n, \end{aligned} \tag{6.285}$$

and

$$\begin{aligned} &\text{minimize} \quad \max_{i=1,\ldots,p} \|F_i x + g_i\| \\ &\text{subject to} \quad x \in \Re^n, \end{aligned}$$

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_∞ Approximation [LVB98])

Consider the complex l_1 approximation problem

$$\begin{aligned} &\text{minimize} \quad \|Ax - b\|_1 \\ &\text{subject to} \quad x \in \mathcal{C}^n, \end{aligned}$$

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 (6.285) and derive the corresponding dual problem. Repeat for the complex l_∞ approximation problem

$$\begin{aligned} &\text{minimize} \quad \|Ax - b\|_\infty \\ &\text{subject to} \quad x \in \mathcal{C}^n. \end{aligned}$$

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 (6.24).

(a) Partition the $n_i \times (n+1)$ matrices $(A_i \quad b_i)$ as

$$(A_i \quad b_i) = \begin{pmatrix} D_i & d_i \\ p_i' & q_i \end{pmatrix}, \qquad i = 1, \ldots, m,$$

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

$$A_i x - b_i \in C_i \qquad \text{if and only if} \qquad \|D_i x - d_i\| \le p_i' x - q_i,$$

so we can write the SOCP (6.24) as

$$\begin{aligned} &\text{minimize} \quad c'x \\ &\text{subject to} \ \ \|D_i x - d_i\| \le p_i' x - q_i, \ i = 1, \ldots, m. \end{aligned}$$

(b) Similarly partition λ_i as

$$\lambda_i = \begin{pmatrix} \mu_i \\ \nu_i \end{pmatrix}, \qquad i = 1, \ldots, m,$$

where $\mu_i \in \Re^{n_i - 1}$ and $\nu_i \in \Re$. Show that the dual problem (6.25) can be written in the form

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^m (d_i' \mu_i + q_i \nu_i) \\ &\text{subject to} \quad \sum_{i=1}^m (D_i' \mu_i + \nu_i p_i) = c, \quad \|\mu_i\| \le \nu_i, \ i = 1, \ldots, m. \end{aligned} \tag{6.286}$$

(c) Show that the primal and dual interior point conditions for strong duality (Prop. 6.1.8) hold if there exist primal and dual feasible solutions $\overline{x}$ and $(\overline{\mu}_i, \overline{\nu}_i)$ such that

$$\|D_i \overline{x} - d_i\| < p_i' \overline{x} - q_i, \qquad i = 1, \ldots, m,$$

and

$$\|\overline{\mu}_i\| < \overline{\nu}_i, \qquad i = 1, \ldots, m,$$

respectively.

6.5 (Monotropic-Conic Problems)

Consider the problem

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

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 $\Re^{n_1+\cdots+n_m}$, respectively. Show that a dual problem is

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{m} q_i(\lambda_i) \\ &\text{subject to} \quad \lambda \in \hat{C} + S^{\perp}, \end{aligned}$$

where $\lambda = (\lambda_1, \dots, \lambda_m)$, $\hat{C}$ is the dual cone of C, and

$$q_i(\lambda_i) = \inf_{z_i \in \Re} \big\{ f_i(z_i) - \lambda_i' z_i \big\}, \qquad i = 1, \dots, m.$$

6.6

Let $f : \Re^n \mapsto \Re$ be a convex function, and let $\{f_k\}$ be a sequence of convex functions $f_k : \Re^n \mapsto \Re$ with the property that $\lim_{k\to\infty} f_k(x_k) = f(x)$ for every $x \in \Re^n$ and every sequence $\{x_k\}$ that converges to x. Then, for any $x \in \Re^n$ and $y \in \Re^n$, and any sequences $\{x_k\}$ and $\{y_k\}$ converging to x and y, respectively, we have

$$\limsup_{k\to\infty} f_k'(x_k; y_k) \le f'(x; y). \tag{6.287}$$

Furthermore, if f is differentiable over $\Re^n$, then it is continuously differentiable over $\Re^n$.

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,

$$f_k'(x_k; y_k) \le \frac{f_k(x_k + \alpha y_k) - f_k(x_k)}{\alpha} < f'(x; y) + \epsilon,$$

so by taking the limit as $k \to \infty$,

$$\limsup_{k\to\infty} f_k'(x_k; y_k) \leq f'(x; y) + \epsilon.$$

Since this is true for all $\epsilon > 0$, we obtain $\limsup_{k\to\infty} f_k'(x_k; y_k) \leq f'(x; y)$.

If f is differentiable at all $x \in \Re^n$, 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 $y \in \Re^n$,

$$\limsup_{k\to\infty} \nabla f(x_k)'y = \limsup_{k\to\infty} f'(x_k; y) \leq f'(x; y) = \nabla f(x)'y.$$

By replacing y with $-y$ in the preceding argument, we obtain

$$-\liminf_{k\to\infty} \nabla f(x_k)'y = \limsup_{k\to\infty} \big(-\nabla f(x_k)'y\big) \leq -\nabla f(x)'y.$$

Therefore, we have $\nabla f(x_k)'y \to \nabla f(x)'y$ for every y, which implies that $\nabla f(x_k) \to \nabla f(x)$. Hence, $\nabla f(\cdot)$ is continuous.

6.7 (Danskin's Theorem)

Let Z be a compact subset of $\Re^m$, and let $\phi : \Re^n \times Z \mapsto \Re$ be continuous and such that $\phi(\cdot, z) : \Re^n \mapsto \Re$ is convex for each $z \in Z$.

(a) The function $f : \Re^n \mapsto \Re$ given by

$$f(x) = \max_{z\in Z} \phi(x, z) \tag{6.288}$$

is convex and has directional derivative given by

$$f'(x; y) = \max_{z\in Z(x)} \phi'(x, z; y), \tag{6.289}$$

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. (6.288)

$$Z(x) = \left\{ \overline{z} \;\middle|\; \phi(x, \overline{z}) = \max_{z\in Z} \phi(x, z) \right\}.$$

In particular, if $Z(x)$ consists of a unique point $\overline{z}$ and $\phi(\cdot, \overline{z})$ is differentiable at x, then f is differentiable at x, and $\nabla f(x) = \nabla_x \phi(x, \overline{z})$, where $\nabla_x \phi(x, \overline{z})$ is the vector with components

$$\frac{\partial \phi(x, \overline{z})}{\partial x_i}, \qquad i = 1, \ldots, n.$$

(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

$$\partial f(x) = \text{conv}\big\{\nabla_x \phi(x, z) \mid z \in Z(x)\big\}, \qquad \forall\, x \in \Re^n.$$

Solution: (a) We note that since ϕ is continuous and Z is compact, the set $Z(x)$ is nonempty by Weierstrass' Theorem and f is finite. For any $z \in Z(x)$, $y \in \Re^n$, 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 α 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

$$f'(x; y) \geq \sup_{z \in Z(x)} \phi'(x, z; y), \qquad \forall\ y \in \Re^n. \tag{6.290}$$

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 $\overline{z} \in Z$. Without loss of generality, we assume that the entire sequence $\{z_k\}$ converges to $\overline{z}$. We have

$$\phi(x_k, z_k) \geq \phi(x_k, z), \qquad \forall\ z \in Z,$$

so by taking the limit as $k \to \infty$ and by using the continuity of ϕ, we obtain

$$\phi(x, \overline{z}) \geq \phi(x, z), \qquad \forall\ z \in Z.$$

Therefore, $\overline{z} \in Z(x)$. We now have

$$\begin{aligned} f'(x; y) &\leq \frac{f(x+\alpha_k y) - f(x)}{\alpha_k} \\ &= \frac{\phi(x+\alpha_k y, z_k) - \phi(x, \overline{z})}{\alpha_k} \\ &\leq \frac{\phi(x+\alpha_k y, z_k) - \phi(x, z_k)}{\alpha_k} \\ &\leq -\phi'(x+\alpha_k y, z_k; -y) \\ &\leq \phi'(x+\alpha_k y, z_k; y), \end{aligned} \tag{6.291}$$

where the last inequality follows from the fact $-f'(x; -y) \leq f'(x; y)$. 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

$$\limsup_{k \to \infty} \phi'(x + \alpha_k y, z_k; y) \leq \phi'(x, \overline{z}; y). \tag{6.292}$$

We take the limit in inequality (6.291) as $k \to \infty$, and we use inequality (6.292) to conclude that

$$f'(x; y) \leq \phi'(x, \overline{z}; y).$$

This relation together with inequality (6.290) proves Eq. (6.289).

For the last statement of part (a), if $Z(x)$ consists of the unique point $\overline{z}$, Eq. (6.289) and the differentiability assumption on ϕ yield

$$f'(x;y) = \phi'(x,\overline{z};y) = y'\nabla_x\phi(x,\overline{z}), \qquad \forall\, y \in \Re^n,$$

which implies that $\nabla f(x) = \nabla_x\phi(x,\overline{z})$.

(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 $\overline{z} \in Z(x)$ and $y \in \Re^n$, we have

$$\begin{aligned} f(y) &= \max_{z\in Z} \phi(y,z) \\ &\geq \phi(y,\overline{z}) \\ &\geq \phi(x,\overline{z}) + \nabla_x\phi(x,\overline{z})'(y-x) \\ &= f(x) + \nabla_x\phi(x,\overline{z})'(y-x). \end{aligned}$$

Therefore, $\nabla_x\phi(x,\overline{z})$ is a subgradient of f at x, implying that

$$\text{conv}\big\{\nabla_x\phi(x,z) \mid z \in Z(x)\big\} \subset \partial f(x).$$

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 $\big\{\nabla_x\phi(x,z) \mid z \in Z(x)\big\}$ is compact. By Prop. 1.2.2, it follows that $\text{conv}\big\{\nabla_x\phi(x,z) \mid z \in Z(x)\big\}$ is compact. If $d \in \partial f(x)$ while $d \notin \text{conv}\big\{\nabla_x\phi(x,z) \mid z \in Z(x)\big\}$, by the Strict Separation Theorem (Prop. 1.5.3), there exists $y \neq 0$, and $\gamma \in \Re$, such that

$$d'y > \gamma > \nabla_x\phi(x,z)'y, \qquad \forall\, z \in Z(x).$$

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, $\partial f(x) \subset \text{conv}\big\{\nabla_x\phi(x,z) \mid z \in Z(x)\big\}$ 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

$$\big\{(x_1, x_2) \mid x_1 > |x_2| > (9/16)^2 |x_1|\big\}.$$

Verify computationally that it converges to the nonoptimal point $(0, 0)$ (cf. Fig. 6.11.1). What happens if a subgradient method with a constant stepsize is used instead? Check computationally.

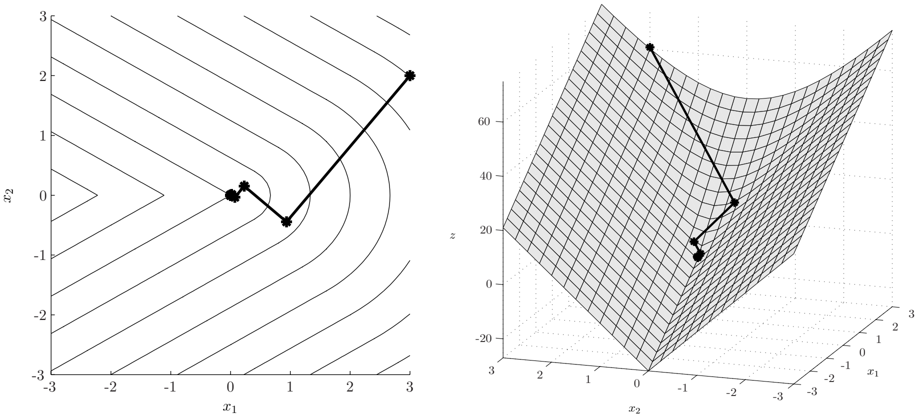

Figure 6.11.1. Level sets and steepest descent path for the function of Exercise 6.8. The method converges to the nonoptimal point $(0, 0)$.

6.9 (Growth Condition for Polyhedral Functions)

Let $f : \Re^n \mapsto (-\infty, \infty]$ 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

$$f^* + \beta d(x) \leq f(x), \qquad \forall\, x \notin X^*,$$

where $d(x) = \min_{x^* \in X^*} \|x - x^*\|$. *Hint*: Complete the details of the following argument:

Assume first that f is linear within $\text{dom}(f)$. Then, there exists $a \in \Re^n$ 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 $\sum_{j \in J_x} \gamma_j \geq \overline{\gamma}$, where $\overline{\gamma} = 1/\max_{j \in J} \|c_j\|$. 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

$$\beta = \overline{\gamma} \min_{j \in J} a'c_j.$$

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

$$f(x) = \max_{i \in I} \{a_i'x + b_i\}, \qquad \forall\ x \in \text{dom}(f),$$

where I is a finite set, and a_i and b_i are some vectors and scalars, respectively. Let

$$Y = \big\{(x, z) \mid z \geq f(x),\, x \in \text{dom}(f)\big\},$$

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

$$Y^* = \big\{(x, z) \mid x \in X^*,\, z = f^*\big\},$$

and that its minimal value is f^*.

Applying the result already shown to the function g, we have for some $\beta > 0$

$$f^* + \beta \hat{d}(x,z) \le g(x,z), \qquad \forall\ (x,z) \notin Y^*,$$

where

$$\hat{d}(x,z) = \min_{(x^*,z^*)\in Y^*} \big(\|x-x^*\|^2 + |z-z^*|^2\big)^{1/2} = \min_{x^*\in X^*} \big(\|x-x^*\|^2 + |z-f^*|^2\big)^{1/2}.$$

Since

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

we have

$$f^* + \beta d(x) \le g(x,z), \qquad \forall\ (x,z) \notin Y^*,$$

and by taking the infimum of the right-hand side over z for any fixed x, we obtain

$$f^* + \beta d(x) \le f(x), \qquad \forall\ x \notin X^*.$$

6.10 (Sublinear Convergence Rate in the Proximal Minimization Algorithm)

Consider the proximal algorithm under the assumptions of Prop. 6.5.2. Assume further that $\alpha > 2$. Show that

$$\limsup_{k\to\infty} \frac{d(x_{k+1})}{d(x_k)^{2/\alpha}} < \infty$$

which is known as *sublinear convergence*. *Hint*: Show first that

$$f(x_{k+1}) - f^* \le \frac{d(x_k)^2}{2c_k}, \qquad \forall\ k.$$

6.11 (Partial Proximal Minimization Algorithm [BeT94])

For each $c > 0$, let ϕ_c be the real-valued convex function on $\Re^n$ 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 $\{1,\ldots,n\}$. For any $z \in \Re^n$, consider a vector $\overline{z}$ satisfying

$$\overline{z} \in \arg\min_{x\in X} \left\{ f(x) + \frac{1}{2c}\sum_{i\in I} |x_i - z_i|^2 \right\},$$

and let $\tilde{z}$ be the vector with components

$$\tilde{z}_i = \begin{cases} z_i & \forall\, i \in I, \\ \overline{z}_i & \forall\, i \notin I. \end{cases}$$

(a) Show that

$$\tilde{z} \in \arg \min_{\{x \mid x_i = z_i,\, i \in I\}} \phi_c(x),$$

$$\overline{z} \in \arg \min_{x \in X} \left\{ f(x) + \frac{1}{2c}\|x - \tilde{z}\|^2 \right\}.$$

(b) Interpret $\overline{z}$ 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 $\phi_c(\overline{z}) \leq f(\overline{z}) \leq \phi_c(z) \leq f(z)$.

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 $f : \Re^n \mapsto (-\infty, \infty]$ be a closed proper convex function with conjugate convex function denoted by $f^\star$. For each $x \in \Re^n$, define

$$f_{c,\lambda}(x) = \inf_{u \in \Re^n} \left\{ f(x - u) + \lambda' u + \frac{c}{2}\|u\|^2 \right\}, \qquad (6.293)$$

where c is a positive scalar, and λ is a vector in $\Re^n$. Use the Fenchel duality theorem to show that

$$f_{c,\lambda}(x) = \sup_{y \in \Re^n} \left\{ x'y - f^\star(y) - \frac{1}{2c}\|y - \lambda\|^2 \right\}. \qquad (6.294)$$

Show also that $f_{c,\lambda}$ approximates f in the sense that

$$\lim_{c \to \infty} f_{c,\lambda}(x) = f(x), \qquad \forall\, x, \lambda \in \Re^n.$$

Furthermore, $f_{c,\lambda}$ is convex and differentiable as a function of x for fixed c and λ, 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. (6.293).

(ii) As the vector y that attains the supremum in Eq. (6.294).

6.13 (Smoothing and the Augmented Lagrangian Method I)

This exercise provides an application of the smoothing technique of the preceding exercise. Let $f : \Re^n \mapsto (-\infty, \infty]$ be a closed proper convex function with conjugate convex function denoted by $f^\star$. Let $F : \Re^n \mapsto \Re$ be another convex function, and let X be a closed convex set. Consider the problem

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

and the equivalent problem

$$\begin{aligned} &\text{minimize} \quad F(x) + f(x-u) \\ &\text{subject to} \quad x \in X,\ u = 0. \end{aligned}$$

Apply the Augmented Lagrangian method to the latter problem, and show that it takes the form

$$x_{k+1} \in \arg\min_{x \in X} \big\{ F(x) + f_{c_k,\lambda_k}(x) \big\},$$

where $f_{c,\lambda}$ is the smoothed function of the preceding exercise; the multiplier update is obtained from the equations

$$u_{k+1} \in \arg\min_{u \in \Re^n} \left\{ f(x_{k+1} - u) + {\lambda_k}'u + \frac{c_k}{2}\|u\|^2 \right\}, \qquad \lambda_{k+1} = \lambda_k + c_k u_{k+1}.$$

Alternatively, λ_{k+1} is given by

$$\lambda_{k+1} = \arg\max_{y \in \Re^n} \left\{ x_{k+1}'y - f^\star(y) - \frac{1}{2c_k}\|y - \lambda_k\|^2 \right\} = \nabla f_{c_k,\lambda_k}(x_{k+1}),$$

where $f^\star$ 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 $\Re^m$, respectively, and $\phi : \Re^n \times Z \mapsto \Re$ is a function such that $\phi(\cdot, z) : \Re^n \mapsto \Re$ is convex for each $z \in Z$, and $-\phi(x, \cdot) : Z \mapsto \Re$ is convex and closed for each $x \in \Re^n$. Consider the problem

$$\begin{aligned} &\text{minimize} \quad \sup_{z \in Z} \phi(x, z) \\ &\text{subject to} \quad x \in X, \end{aligned}$$

where X is a nonempty closed convex subset of $\Re^n$. Consider also the equivalent problem

$$\begin{aligned} &\text{minimize} \quad H(x, y) \\ &\text{subject to} \quad x \in X,\ y = 0, \end{aligned}$$

where H is the function

$$H(x,y) = \sup_{z\in Z}\big\{\phi(x,z) - y'z\big\}, \qquad x \in \Re^n,\ y \in \Re^m.$$

Apply the Augmented Lagrangian method to this problem, and show that it takes the form

$$x_{k+1} \in \arg\min_{x\in X} f^\star_{c_k,\lambda_k}(x),$$

where $f^\star_{c,\lambda} : \Re^n \mapsto \Re$ is the differentiable function given by

$$f^\star_{c,\lambda}(x) = \min_{y\in\Re^m}\left\{H(x,y) - \lambda' y + \frac{c}{2}\|y\|^2\right\}, \qquad x \in \Re^n.$$

The multiplier update is obtained from the equations

$$y_{k+1} \in \arg\min_{y\in\Re^m}\left\{H(x_{k+1},y) - \lambda_k' y + \frac{c_k}{2}\|y\|^2\right\}, \qquad \lambda_{k+1} = \lambda_k - c_k y_{k+1}.$$

6.15

Consider the scalar function $f(x) = |x|$. Show that for $x \in \Re$ and $\epsilon > 0$, we have

$$\partial_\epsilon f(x) = \begin{cases} \left[-1, -1 - \frac{\epsilon}{x}\right] & \text{for } x < -\frac{\epsilon}{2}, \\ [-1,1] & \text{for } x \in \left[-\frac{\epsilon}{2}, \frac{\epsilon}{2}\right], \\ \left[1 - \frac{\epsilon}{x}, 1\right] & \text{for } x > \frac{\epsilon}{2}. \end{cases}$$

6.16 (Subgradient Methods with Low Level Errors [NeB10])

Consider the problem of minimizing a convex function $f : \Re^n \to \Re$ over a closed convex set X, and assume that the optimal solution set, denoted X^*, is nonempty. Consider the iteration

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

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

$$\|e_k\| \le \beta, \qquad k = 0, 1, \ldots,$$

where β is some positive scalar. Assume that for some $\gamma > 0$, we have

$$f(x) - f^* \le \gamma \min_{x^*\in X^*}\|x - x^*\|, \qquad \forall\, x \in X,$$

where $f^* = \min_{x\in X} f(x)$, and that for some $c > 0$, we have

$$\|g\| \le c, \qquad \forall\, g \in \partial f(x_k),\ k = 0, 1, \ldots$$

(these assumptions are satisfied if f is a polyhedral function). Assuming $\beta < \gamma$, show that if α_k is equal to some constant α for all k, then

$$\liminf_{k\to\infty} f(x_k) \le f^* + \frac{\alpha\gamma(c+\beta)^2}{2(\gamma-\beta)}, \tag{6.295}$$

while if

$$\alpha_k \to 0, \qquad \sum_{k=0}^{\infty} \alpha_k = \infty,$$

then $\liminf_{k\to\infty} f(x_k) = f^*$. Use the example of Exercise 6.15 to show that the estimate (6.295) is sharp.

6.17 (Sharpness of the Error Tolerance Estimate)

Consider the unconstrained optimization of the two-dimensional function

$$f(x_1, x_2) = \sum_{i=1}^{M} c_0\big(|x_1+1| + 2|x_1| + |x_1-1| + |x_2+1| + 2|x_2| + |x_2-1|\big),$$

where c_0 is a positive constant, by using the incremental subgradient method with a constant stepsize α. Show that there exists a component processing order such that when the method starts a cycle at the point $\overline{x} = (\overline{x}_1, \overline{x}_2)$, where $\overline{x}_1 = \overline{x}_2 = \alpha M c_0$ with $\alpha M c_0 \le 1$, it returns to $\overline{x}$ at the end of the cycle. Use this example to show that starting from $\overline{x}$, we have

$$\liminf_{k\to\infty} f(\psi_{i,k}) \ge f^* + \frac{\beta\alpha c^2}{2}, \qquad \forall\ i = 1, \ldots, m,$$

for some constant β (independent of c_0 and M), where $c = mc_0$ and $m = 8M$ [cf. Eq. (6.74)].

Solution: Consider the incremental subgradient method with the stepsize α and the starting point $\overline{x} = (\alpha MC_0, \alpha MC_0)$, and the following component processing order:

M components of the form $|x_1|$ [endpoint is $(0, \alpha MC_0)$],

M components of the form $|x_1+1|$ [endpoint is $(-\alpha MC_0, \alpha MC_0)$],

M components of the form $|x_2|$ [endpoint is $(-\alpha MC_0, 0)$],

M components of the form $|x_2+1|$ [endpoint is $(-\alpha MC_0, -\alpha MC_0)$],

M components of the form $|x_1|$ [endpoint is $(0, -\alpha MC_0)$],

M components of the form $|x_1-1|$ [endpoint is $(\alpha MC_0, -\alpha MC_0)$],

M components of the form $|x_2|$ [endpoint is $(\alpha MC_0, 0)$], and

M components of the form $|x_2-1|$ [endpoint is $(\alpha MC_0, \alpha MC_0)$].

With this processing order, the method returns to $\overline{x}$ 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 $4MC_0 + 2\alpha M^2C_0^2$. The optimal function value is $f^* = 4MC_0$, so that

$$\liminf_{k\to\infty} f(\psi_{i,k}) \geq f^* + 2\alpha M^2C_0^2.$$

Since $m = 8M$ and $mC_0 = C$, we have $M^2C_0^2 = C^2/64$, implying that

$$2\alpha M^2C_0^2 = \frac{1}{16}\frac{\alpha C^2}{2},$$

and therefore

$$\liminf_{k\to\infty} f(\psi_{i,k}) \geq f^* + \frac{\beta\alpha C^2}{2},$$

with $\beta = 1/16$.

6.18 (Aggregate Subgradients)

Show that the aggregate subgradient of Eq. (6.164) 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 $\Re^n$, and let $\phi : \Re^n \mapsto \Re$ be a convex function that is differentiable over an open set that contains X. Define the function

$$D(x,y) = \phi(x) - \phi(y) - \nabla\phi(y)'(x-y), \qquad \forall\ y \in X,\ x \in \Re^n.$$

(a) Show that if ϕ is strongly convex in the sense that

$$\big(\nabla\phi(x) - \nabla\phi(y)\big)'(x-y) \geq \|x-y\|^2, \qquad \forall\ x,y \in X.$$

then D has the property

$$D(x,y) \geq \frac{1}{2}\|x-y\|^2, \qquad \forall\ x,y \in X,$$

with equality holding in the case where

$$\phi(x) = \frac{1}{2}\|x\|^2.$$

Abbreviated proof: Add the relations

$$D(x,y) = \phi(x) - \phi(y) - \nabla\phi(y)'(x-y),$$

$$D(y,x) = \phi(y) - \phi(x) - \nabla\phi(x)'(y-x),$$

and use the strong convexity property of D to show that

$$2D(x,y) = D(x,y) + D(y,x) \geq \|x-y\|^2.$$

(b) Let $F : X \mapsto (-\infty, \infty]$ be a closed proper convex function, and for any $y \in X$, define

$$y^+ = \arg\min_{x\in X}\big\{F(x) + D(x,y)\big\}.$$

Show that

$$F(y^+) + D(y^+,y) + D(x,y^+) \leq F(x) + D(x,y), \qquad \forall\ x,y \in X.$$

Abbreviated proof: Use the optimality condition of Prop. 5.4.7 to obtain

$$F(y^+) \leq F(x) + \nabla_x D(y^+,y)'(x-y^+).$$

Then by writing $\nabla_x D(y^+,y) = \nabla\phi(y^+) - \nabla\phi(y)$ and rearranging terms,

$$F(y^+) - \nabla\phi(y)'(y^+ - y) - \nabla\phi(y^+)'(x-y^+) \leq F(x) - \nabla\phi(y)'(x-y).$$

Add $\phi(x) - \phi(y)$ to both sides.

(c) Let $f : \Re^n \mapsto \Re$ be a convex differentiable function, and denote

$$\ell(y;x) = f(x) + \nabla f(x)'(y-x), \qquad \forall\ x,y \in \Re^n,$$

[cf. Eq. (6.260)]. Use part (b) with

$$F(y) = \alpha\,\ell(y;x_k), \qquad D(x,y) = \frac{1}{2}\|x-y\|^2,$$

to show that for all $x \in X$, we have

$$\ell(x_{k+1};x_k) + \frac{1}{2\alpha}\|x_{k+1}-x_k\|^2 \leq \ell(x;x_k) + \frac{1}{2\alpha}\|x-x_k\|^2 - \frac{1}{2\alpha}\|x-x_{k+1}\|^2,$$

where x_{k+1} is generated by the gradient projection iteration

$$x_{k+1} = P_X\big(x_k - \alpha\nabla f(x_k)\big).$$

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.

[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.

[BeT10] Beck, A., and Teboulle, M., 2010. "Gradient-Based Algorithms with Applications to Signal Recovery Problems, in Convex Optimization in Signal Processing and Communications (Y. Eldar and D. Palomar, eds.), Cambridge University Press, pp. 42-88.

[BeY11] Bertsekas, D. P., and Yu, H., 2011 "A Unifying Polyhedral Approximation Framework for Convex Optimization," SIAM J. on Optimization, Vol. 21, pp. 333-360.

[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.

[Ber83] Bertsekas, D. P., 1983. "Asynchronous Distributed Computation of Fixed Points," Math. Programming, Vol. 27, pp. 107-120.

[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; Math. Programming, Vol. 129, 2011, pp. 163-195.

[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.

[CFI07] Cruz Neto, J. X., Ferreira, A. N., Iusem, A. N., and Monteiro, R. D. C., 2007. "Dual Convergence of the Proximal Point Method with Bregman Distances for Linear Programming," Optimization Methods and Software, Vol. 22, pp. 339-360.

[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.

[Eck93] Eckstein, J., 1993. "Nonlinear Proximal Point Algorithms Using Bregman Functions, with Applications to Convex Programming," Math. Operations Res., Vol. 18, pp. 202-226.

[Eck98] Eckstein, J., 1998. "Approximate Iterations in Bregman-Function-Based Proximal Algorithms," Math. Programming, Vol. 83, pp. 113-123.

[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.

[FLT02] Fukushima, M., Luo, Z. Q., and Tseng, P., 2002. "Smoothing Functions for Second-Order-Cone Complementarity Problems," SIAM Journal on Optimization, Vol. 12, pp. 436-460.

[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.

[FlH95] Florian, M. S., and Hearn, D., 1995. "Network Equilibrium Models and Algorithms," Handbooks in OR and MS, Ball, M. O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L., (Eds.), Vol. 8, North-Holland, Amsterdam, pp. 485-550.

[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.

[LaP99] Larsson, T., and Patricksson, M., 1999. "Side Constrained Traffic Equilibrium Models - Analysis, Computation and Applications," Transportation Research, Vol. 33, pp. 233-264.

[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.

[LiM79] Lions, P. L., and Mercier, B., 1979. "Splitting Algorithms for the Sum of Two Nonlinear Operators," SIAM J. on Numerical Analysis, Vol. 16, pp. 964-979.

[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.

[NeB10] Nedić, A., and Bertsekas, D. P., 2010. “The Effect of Deterministic Noise in Subgradient Methods,” Math. Programming, Ser. A, Vol. 125, pp. 75-99.

[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.

[Ned11] Nedić, A., 2011. “Random algorithms for convex minimization problems,” Mathematical Programming, Ser. B, Vol. 129, pp. 225-253.

[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.

[Pas79] Passty, G. B., 1979. "Ergodic Convergence to a Zero of the Sum of Monotone Operators in Hilbert Space," J. Math. Anal. Appl., Vol. 72, pp. 383-390.

[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.

[Spi85] Spingarn, J. E., 1985. “Applications of the Method of Partial Inverses to Convex Programming: Decomposition,” Math. Programming, Vol. 32, pp. 199-223.

[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. Progamming, 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.

[Tse04] Tseng, P., 2004. “An Analysis of the EM Algorithm and Entropy-

Like Proximal Point Methods," Math. Operations Research, Vol. 29, pp. 27-44.

[Tse08] Tseng, P., 2008. "On Accelerated Proximal Gradient Methods for Convex-Concave Optimization," Report, Math. Dept., Univ. of Washington.

[Tse09] Tseng, P., 2009. "Some Convex Programs Without a Duality Gap," Math. Programming, Vol. 116, pp. 553-578.

[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 Vanderberghe, L., (eds), 2000. Handbook of Semidefinite Programming, Kluwer, Boston.

[WaB12] Wang, M., and Bertsekas, D. P., 2012. "Incremental Constraint Projection Methods for Variational Inequalities", Lab. for Information and Decision Systems Report LIDS-P-2898, MIT.

[WaB13] Wang, M., and Bertsekas, D. P., 2013. "Incremental Constraint Projection-Proximal Methods for Nonsmooth Convex Optimization," Lab. for Information and Decision Systems Report LIDS-P-2907, MIT.

[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.

[YOG08] Yin, W., Osher, S., Goldfarb, D., and Darbon, J., 2008. "Bregman Iterative Algorithms for ℓ_1-Minimization with Applications to Compressed Sensing," SIAM J. Imaging Sciences, Vol. 1, pp. 143-168.

[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.