1.2.2 Linear programming

Another important class of optimization problems is *linear programming*, in which the objective and all constraint functions are linear:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, m. \end{array} \tag{1.5}$$

Here the vectors $c, a_1, \ldots, a_m \in \mathbf{R}^n$ and scalars $b_1, \ldots, b_m \in \mathbf{R}$ are problem parameters that specify the objective and constraint functions.

Solving linear programs

There is no simple analytical formula for the solution of a linear program (as there is for a least-squares problem), but there are a variety of very effective methods for solving them, including Dantzig's simplex method, and the more recent interior-point methods described later in this book. While we cannot give the exact number of arithmetic operations required to solve a linear program (as we can for least-squares), we can establish rigorous bounds on the number of operations required to solve a linear program, to a given accuracy, using an interior-point method. The complexity in practice is order $n^2 m$ (assuming $m \geq n$) but with a constant that is less well characterized than for least-squares. These algorithms are quite reliable, although perhaps not quite as reliable as methods for least-squares. We can easily solve problems with hundreds of variables and thousands of constraints on a small desktop computer, in a matter of seconds. If the problem is sparse, or has some other exploitable structure, we can often solve problems with tens or hundreds of thousands of variables and constraints.

As with least-squares problems, it is still a challenge to solve extremely large linear programs, or to solve linear programs with exacting real-time computing requirements. But, like least-squares, we can say that solving (most) linear programs is a mature technology. Linear programming solvers can be (and are) embedded in many tools and applications.

Using linear programming

Some applications lead directly to linear programs in the form (1.5), or one of several other standard forms. In many other cases the original optimization problem does not have a standard linear program form, but can be transformed to an equivalent linear program (and then, of course, solved) using techniques covered in detail in chapter 4.

As a simple example, consider the *Chebyshev approximation problem*:

$$\text{minimize} \quad \max_{i=1,\ldots,k} |a_i^T x - b_i|. \tag{1.6}$$

Here $x \in \mathbf{R}^n$ is the variable, and $a_1, \ldots, a_k \in \mathbf{R}^n$, $b_1, \ldots, b_k \in \mathbf{R}$ are parameters that specify the problem instance. Note the resemblance to the least-squares problem (1.4). For both problems, the objective is a measure of the size of the terms $a_i^T x - b_i$. In least-squares, we use the sum of squares of the terms as objective, whereas in Chebyshev approximation, we use the maximum of the absolute values.