A linear program is called a *network flow* problem if there is a directed graph $G = (N, A)$, where $N = \{1, \ldots, n\}$ is a set of *nodes* and A is a set of directed arcs connecting the nodes in N. That is, an arc is an ordered pair (i, j). Define x_{ij} to be the flow on arc (i, j). Then a minimum-cost network-flow problem is a linear program of the form

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = b_i \\ & 0 \leq x_{ij} \leq u_{ij}, \quad \forall i, j, \end{aligned}$$

where c_{ij} are cost coefficients, u_{ij} are upper bounds on the flows (equal to zero if no arc (i, j) exists) and b_i are *source/sink* quantities satisfying $\sum_{i=1}^{n} b_i = 0$. Network-flow problems can be solved even more efficiently than general linear programs using specialized algorithms.

NonDifferentiable Optimization

In nondifferentiable-optimization problems, the gradient $\nabla \mathbf{f}(\mathbf{x})$ may not always exist everywhere on the constraint set X. For example, $f(x)$ may be a scalar continuous, piecewise linear function of x of the form

$$f(x) = \begin{cases} -x & x \geq 0 \\ 3x & x < 0. \end{cases}$$

This function has a corner point at $x = 0$ where the derivative does not exist. In such cases, we have the following necessary and sufficient condition for optimality in the unconstrained, convex case:

PROPOSITION C.14 *If f is concave, and $X = \Re^n$, then $\mathbf{x}^*$ is a global maximum if and only if $0 \in \partial f(\mathbf{x}^*)$.*

Note that for the example given above, f is concave and $\partial f(0) = [-1, 3]$, so zero is contained in the subdifferential at x^* and hence $x^* = 0$ is a maximum. Also observe that if f is differentiable at x^*, the above condition reduces to (C.4).