Complexity of optimization problems

Generic optimization problem: For a (possibly infinite) set C and function $f : C \to \mathbb{R}$

$$\max_{x \in C} f(x) \qquad \text{(MAX-}f\text{)}$$
$$\min_{x \in C} f(x) \qquad \text{(MIN-}f\text{)}$$

C and f are encoded in some (possibly implicit) way by a string. The computational problem is to either find the optimal value or to find an optimal solution. We focus on finding optimal values for concreteness.

Example: MAX-CLIQUE

$$\max_{U \subseteq V \text{ is a clique of } G=(V,E)} |U| \qquad (MAX-CLIQUE)$$

MAX-CLIQUE (finding the optimal value) "reduces" to instances G, k of CLIQUE: try all values of k (or, even better, binary search to find optimal k). Obviously, CLIQUE reduces to MAX-CLIQUE.

One could say that CLIQUE is the decision version of MAX-CLIQUE and they are computationally equivalent up to polynomial time reductions.

More generally, given optimization problem MAX-f above, its decision version is

$$\text{D-}f = \{\langle C, f, k \rangle : (\exists x \in C) f(x) \geq k\}$$

MAX-f and D-f can be equivalent in many specific cases via the same reduction argument (binary search).

Example: 0-1-IP (binary linear integer programming)

$$\max \sum c_i x_i$$
$$\text{s.t.} Ax \leq b$$
$$x_i \in \{0,1\}$$

(where $Ax \leq b$ is a shorthand for $(\forall j = 1, \ldots, m) \sum_i a_{ji} x_i \leq b_j$). $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n$. Without loss of generality we can assume that A, b, c are integral. The decision version as above is

$$\text{D-0-1-IP} = \{\langle A, b, c, k\rangle : (\exists x \in \{0,1\}^n) Ax \leq b, c^T x \geq k\}.$$

But the inequality $c^T x \geq k$ is just another linear inequality that can be appended to A, b to get the following computationally equivalent version:

$$\text{D-0-1-IP'} = \{\langle \bar{A}, \bar{b}\rangle : (\exists x \in \{0,1\}^n) \bar{A} x \leq \bar{b}\}.$$

Claim: D-0-1-IP is NP-complete. It is clearly in NP (the certificate is some $x \in \{0,1\}^n$). To see the completeness, many combinatorial optimization problems can be reduced in polynomial time to D-0-1-IP. For example, CLIQUE $\leq_p$ D-0-1-IP: It is helpful to see first how to write MAX-CLIQUE as an equivalent binary integer program:

$$\max \sum_{v \in V} x_v$$
$$\text{s.t.} \qquad (\forall (i,j) \in V \times V \setminus E) \quad x_i + x_j \leq 1$$
$$x_i \in \{0,1\}$$

So, a polynomial time mapping reduction that shows CLIQUE $\leq_p$ D-0-1-IP is to map $G = (V, E)$ and k to a matrix A with a row for every $(i,j) \in V \times V \setminus E$. Row at (i,j) has length $|V|$ and has a one at positions i and j and zeros otherwise. Similarly $b_{(i,j)} = 1$ and $c_v = 1$ for all $v \in V$. Value k is the same.

Example: LINEAR-IP

$$\max c^T x$$
$$\text{s.t.} Ax \leq b$$
$$x_i \in \mathbb{Z}$$

$$\text{D-L-IP} = \{\langle A, b, c, k\rangle : (\exists x \in \mathbb{Z}^n) Ax \leq b, c^T x \geq k\}.$$

Clearly D-0-1-IP $\leq_p$ D-L-IP. One can show with some work that D-L-IP $\in$ NP, to conclude that D-L-IP is NP-complete.

Example: LINEAR-PROGRAMMING

$$\begin{aligned} &\max c^T x \\ \text{s.t.} &Ax \leq b \\ &x_i \in \mathbb{Q} \end{aligned}$$

Decision version can be show to be in P. One way is via the ellipsoid algorithm.