Relaxations for solving integer programs.

Consider an IP with arbitrary objective function c and arbitrary feasible region X:

$$z = \max\{c(x) \mid x \in X \subseteq \mathbb{Z}^n_+\}. \tag{1}$$

We typically solve it by finding lower bounds $\underline{z}$ and upper bounds $\overline{z}$ on z, i.e., $\underline{z} \leq z \leq \overline{z}$.

How do we find (good) bounds?

1. primal bounds. Any feasible solution $x^* \in X$ gives a lower bound $\underline{z} = c(x^*)$.
2. dual bounds. Upper bounds $\overline{z}$ often obtained by solving relaxations.

Matching bounds, i.e., $\underline{z} = \overline{z}$, imply that you have solved the problem.

Definition 1. *A problem (RP)*

$$z^R = \max\{f(x) \mid x \in T \subseteq \mathbb{R}^n_+\} \tag{2}$$

is a relaxation of problem IP (1) *if:*

1. $X \subseteq T$, *and*
2. $c(x) \leq f(x)$ *for all* $x \in X$.

Proposition 1. *If RP* (2) *is a relaxation of IP* (1), *then* $z \leq z^R$.

There are many types of *relaxations*, e.g.,

1. Linear programming relaxation;
2. Combinatorial relaxation;
3. Lagrangian relaxation.

Definition 2. *A <u>polyhedron</u> is a set of the form* $\{x \in \mathbb{R}^n \mid Ax \leq b\}$*, where* $A \in \mathbb{R}^{m \times n}$ *and* $b \in \mathbb{R}^m$*.*

Definition 3. *A set* $P \subseteq \mathbb{R}^n$ *is <u>bounded</u> if there exists* $d \in \mathbb{R}_{++}$ *such that* $P \subseteq [-d, d]^n$*.*

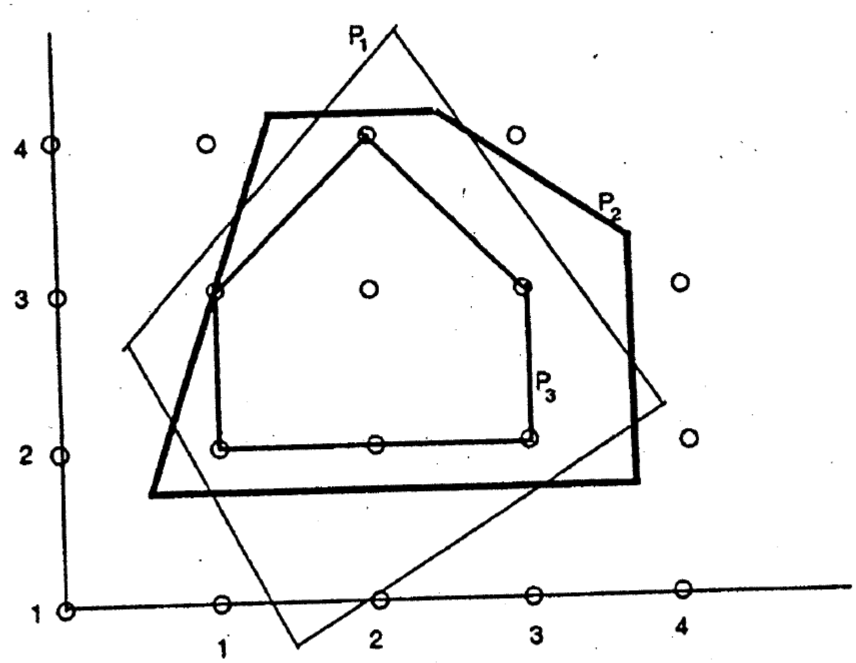
Definition 4. *A polyhedron that is bounded is called a <u>polytope</u>.*

Definition 5. *A polyhedron* $P \subseteq \mathbb{R}^{n+d}$ *is a <u>formulation</u> for a set* $X \subseteq \mathbb{Z}^n \times \mathbb{R}^d$ *if and only if* $X = P \cap \left(\mathbb{Z}^n \times \mathbb{R}^d\right)$*.*

Linear programming relaxation. For the integer program $\max\{c^T x \mid P \cap \mathbb{Z}^n\}$ with formulation $P = \{x \in \mathbb{R}^n_+ \mid Ax \leq b\}$, its <u>*linear programming relaxation*</u> is $z^{LP} = \max\{c^T x \mid x \in P\}$.

Definition 6. *Given a set* $X \subseteq \mathbb{Z}^n \times \mathbb{R}^d$ *and two formulations* P_1 *and* P_2 *for* X*,*

- P_1 *is a <u>stronger</u> formulation than* P_2 *if* $P_1 \subsetneq P_2$*;*
- P_1 *is <u>at least as strong</u> as* P_2 *if* $P_1 \subseteq P_2$*;*
- P_1 *and* P_2 *are <u>incomparable</u> if* $P_1 \not\subseteq P_2$ *and* $P_2 \not\subseteq P_1$*;*
- P_1 *is <u>ideal</u> or <u>perfect</u> if* $P_1 = \mathrm{conv}(X)$*.*



Branch and bound

Variants of branch and bound are the most common way to solve IPs. They are based on the following simple result for the general problem:

$$z = \max\{c(x) \mid x \in S\}.$$

Proposition 2. *Let $S = S_1 \cup \cdots \cup S_k$ be a decomposition of S into smaller sets, and let $z_i = \max\{c(x) \mid x \in S_i\}$ for $i = 1, \ldots, k$. Then, $z = \max\{z_i \mid i \in [k]\}$.*

Proposition 3. *Suppose we have lower and upper bounds for the subproblems: $\underline{z_i} \le z_i \le \overline{z_i}$. Then, we can get lower and upper bounds for the original problem:*

- $\underline{z} = \max\left\{\underline{z_i} \mid i \in [k]\right\}$*;*
- $\overline{z} = \max\{\overline{z_i} \mid i \in [k]\}$.

What is the typical decomposition $S = S_1 \cup \cdots \cup S_k$ in LP-based branch-and-bound algorithms?

Solving MIPs via a branch-and-bound approach

In what follows, N_i represents a node in the branch-and-bound tree with corresponding LP relaxation LP_i, and N_0 is the root node.

1. **Initialize.**
   - $\mathcal{L} := \{N_0\}$;
   - $\underline{z} := -\infty$;
   - $(x^*, y^*) := \emptyset$;
2. **Terminate?**
   - if $\mathcal{L} = \emptyset$, the solution (x^*, y^*) is optimal.
3. **Select node.**
   - choose a node N_i in $\mathcal{L}$ and delete it from $\mathcal{L}$;
4. **Bound.**
   - solve LP_i;
   - if LP_i is infeasible, go to step 2;
   - if LP_i is feasible, let (x^i, y^i) be an optimal solution of LP_i and let z_i its objective value;
5. **Prune.**
   - if $z_i \leq \underline{z}$ go to step 2;
   - if (x^i, y^i) is feasible to the MIP, update:
     - $\underline{z} := z_i$;
     - $(x^*, y^*) := (x^i, y^i)$;
     - go to step 2;
6. **Branch.**
   - from LP_i construct $k \geq 2$ linear programs $\mathrm{LP}_{i_1}, \ldots, \mathrm{LP}_{i_k}$ with smaller feasible regions whose union does not contain (x^i, y^i), but contains the solutions of LP_i with $x \in \mathbb{Z}^n$.
   - add the new nodes $N_{i_1}, \ldots, N_{i_k}$ to $\mathcal{L}$ and go to step 2;

Some may say this is not an "algorithm" since some steps are not clearly specified. Like what?