26 Apr 2024

Integer Programming and Linear Programming (§11.6 in book)

Announcements

1. Problem Set 9: released this morning
   due next Thurs 11:59

2. CIS Student Hiring (for Fall TA positions)
   cis-student-hiring.coecis.cornell.edu
   Applications due Mon., 4/29, 11:55pm.

Integer Programming:

Given variables $x_1, \ldots, x_n$ (values in $\mathbb{Z}$) — integers

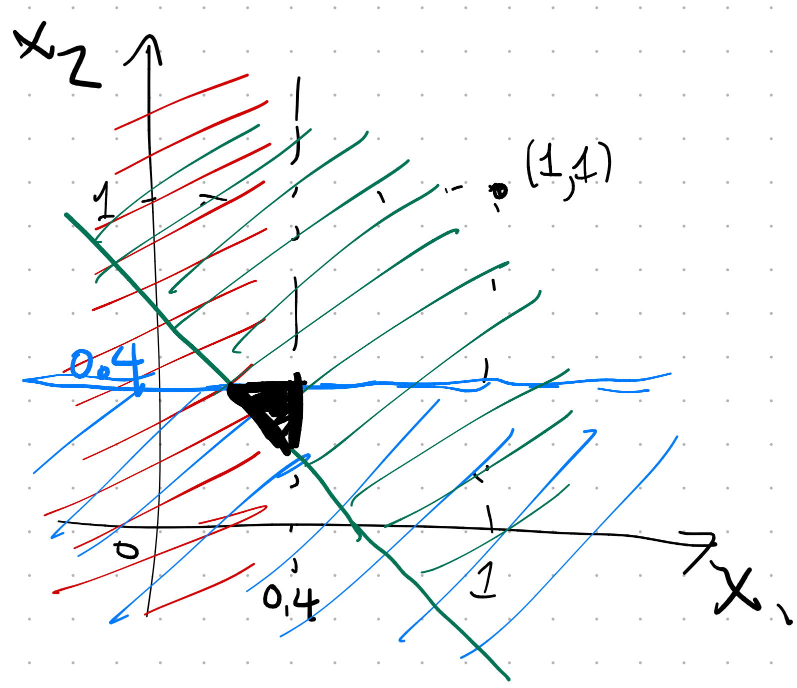
constraints $\vec{a}_i \cdot \vec{x} \le b_i \quad i = 1, 2, \ldots, m$

$\vec{a}_i$: coefficient vector of constraint i

b_i: right-hand side

Does there exist an integer solution $\vec{x} \in \mathbb{Z}^n$?

$$5x_1 \le 2$$
$$5x_2 \le 2$$
$$-2x_1 - 2x_2 \le -1$$



Infeasible integer program.

Linear Programming: does $\{\vec{a}_i \cdot \vec{x} \le b_i : i=1,\dots,m\}$ have a solution $\vec{x} \in \mathbb{R}^n$?

Integer Prog NP-Complete. (Later this lecture.)

Linear Prog $\in P$ (CS 6820)

(various ORIE courses)

VERTEX COVER $\le_p$ INTEGER PROG.

Vtx cover: $G = (V,E)$ undir graph.

$S \subseteq V$ is called a "vertex cover" if every edge of G has at least one endpoint in S.

Decision problem: Given (G, k) is there a vtx cover of size $\le k$?

Optimization problem: Given G what is the minimum size of a vertex cover?

To reduce vertex cover to integer prog. we start by making variables x_v for each vertex $v \in V$.

Intended interpretation: $x_v = 1$ means $v \in S$
$x_v = 0$ means $v \notin S$

Now write inequalities among $\{x_v\}$ s.t. finding a set S which is a vertex cover of size $\le k$ is equivalent to finding $\vec{x}$ that satisfies the inequalities.

x_v can only take values $\{0,1\}$ $\longrightarrow$ $x_v \geq 0 \qquad x_v \leq 1 \qquad \forall v$

edge e is covered $\longrightarrow$ $x_u + x_v \geq 1 \qquad \forall e=(u,v)$

$|S| \leq k$ $\longrightarrow$ $\sum_{v \in V} x_v \leq k$

FACT: $\{$ vectors $\vec{x} \in \mathbb{Z}^V$ satisfying the above $\}$

$$\cong$$

$\{$ vertex cover sets S of size $\leq k \}$

$\vec{x} \updownarrow$ $S = \{ v : x_v = 1 \}$

Int. Prog. and Lin. Prog. (optimization version)

$$\boxed{\begin{array}{ll} \text{maximize} & c \cdot x \\ \text{s.t.} & a_i \cdot x \leq b_i \ \forall i \end{array}} \quad \text{or} \quad \boxed{\begin{array}{ll} \text{minimize} & c \cdot x \\ \text{s.t.} & a_i \cdot x \geq b_i \ \forall i \end{array}}$$

Int. Prog. optimization is NP-Hard.

Lin. Prog. optimization is solvable in poly time.

A general strategy for designing approximation algorithms:

1. write the problem as an integer program.
2. RELAX to a linear program: same variables & constraints, but the variables can take real values.
3. Solve the LP.
4. Output a "nearby" integer point.

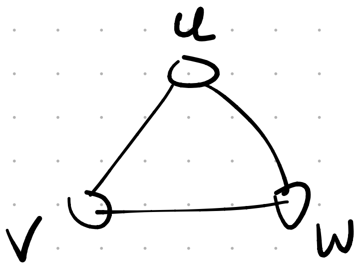
Applying to vertex cover.

$$[\text{VC-LP}] \quad \text{minimize} \quad \sum_{v \in V} x_v$$

$$\text{s.t.} \quad x_u + x_v \geq 1 \quad \forall e = (u,v)$$

$$0 \leq x_v \leq 1 \quad \forall v$$

What could go wrong when we solve this?



$$x_u = x_v = x_w = \frac{1}{2}.$$

LP SOLVER: OPT = 3.
(6 variables, each = $\frac{1}{2}$)

TRUTH: min VC size = 4.

A poly-time 2-approx to vertex cover by LP rounding:

1. Solve VC-LP to obtain an optimal fractional solution, $\vec{x}$
2. Round $\vec{x}$ to integer vector $\vec{y}$,

$$y_v = \begin{cases} 1 & \text{if } x_v \geq \frac{1}{2} \\ 0 & \text{if } x_v < \frac{1}{2} \end{cases}$$

3. Output $S = \{ v \mid y_v = 1 \}$.

CLAIM. S is always a vertex cover. Its size is always $\leq 2 \cdot$ (opt VC size)

Why vertex cover?

$\forall e = (u,v) \qquad x_u + x_v \geq 1$

At least one of x_u, x_v is $\geq \frac{1}{2}$.

$\implies$ at least one of $y_u, y_v = 1$.

$$\text{size}(S) = \sum_{v \in V} y_v$$

$$\leq 2 \cdot \sum_{v \in V} x_v$$

$$= 2 \cdot \text{OPT}(\text{VC-LP})$$

$$\leq 2 \cdot \text{OPT}(\text{VC-IP})$$