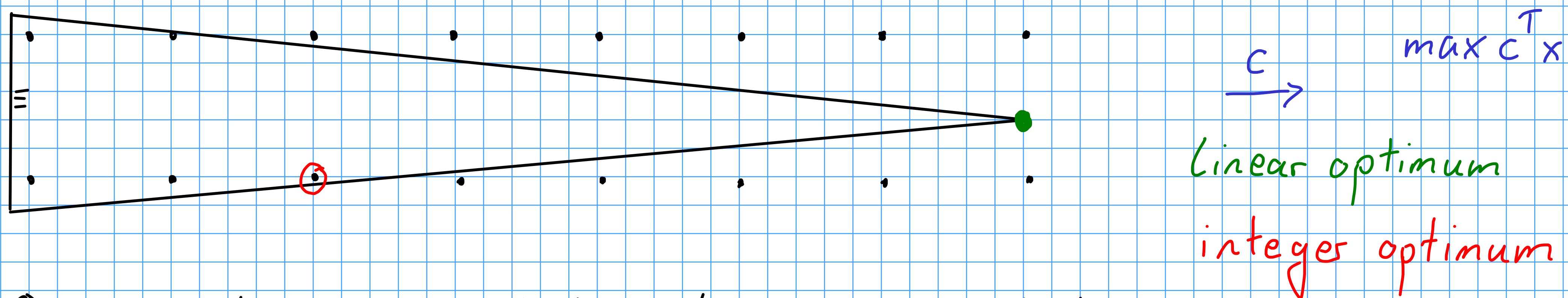
HW8: Exercise 15-8 from AMPL book.

Reading: Vanderbei 23.1-3

The Hardness of Integer Programming

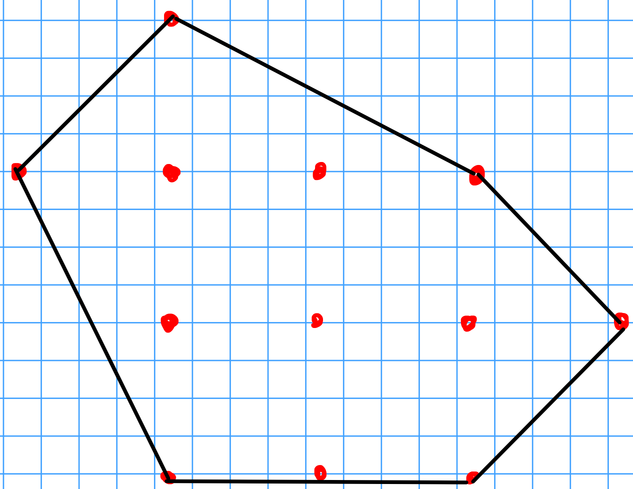
Solving IPs or MIPs as LPs (a "relaxation") and rounding fractional values to integers works well only for special applications. In general, finding an optimal or even any integer solution from a given optimal relaxed/fractional solution is just as hard as the original integer problem.



One can have an arbitrarily large gap between the optima.

Geometry of "Easy" and "Hard" Integer Programs

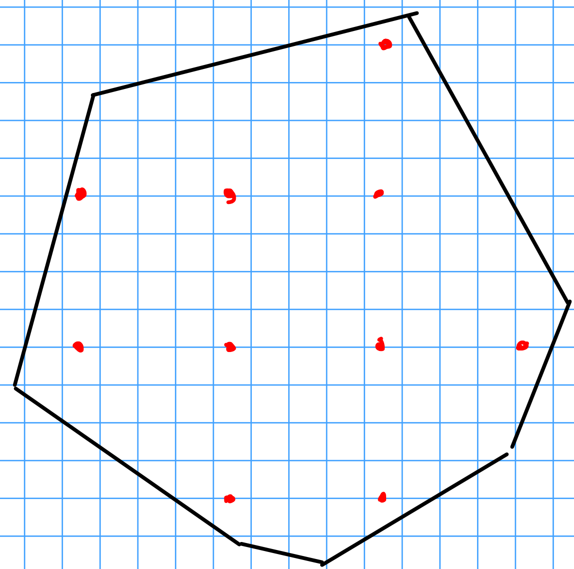
Best case



all vertices are integral

any obj. function leads to an opt. vertex, so an opt. integral solution can be found solving an LP

Bad case



not even a single integral point is on the boundary

for all obj. functions there will be a gap between LP and IP optimum

Best Case Example: Assignment Problem

$$\max \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i = 1,\dots,n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j = 1,\dots,n$$

$$x_{ij} \in \{0,1\} \quad \forall i,j = 1,\dots,n$$

$x_{ij} = 1$ if i and j are assigned to each other,
$x_{ij} = 0$ else

The assignment problem can be solved using $x_{ij} \geq 0$, i.e., as an LP, because the vertices are precisely the feasible 0/1-solutions.

Reason: "total unimodularity" of constraint matrix & integral right-hand sides

possible project topic

Bad Case Example: Selection / Knapsack Problem

$$\max \sum_{i=1}^{n} c_i \cdot x_i$$

NP-hard

x_i denotes whether item i is selected or not

$$\sum_{i=1}^{n} a_i \cdot x_i \leq b$$

$$x_i \in \{0,1\} \quad \forall i=1,\dots,n$$

There exist weakly-polynomial algorithms for all LPs, but integer programs can be NP-hard even for a single main constraint.

Binary (0,1) - Modeling Techniques

Let $N = \{1, \dots, n\}$ be a set of items and consider a selection problem with $x_i \in \{0,1\}$ based on whether item i is selected.

- select at most k items from a set $S \subseteq N$ $\quad \sum_{i \in S} x_i \leq k$
- select at least or exactly k items from S $\quad \sum_{i \in S} x_i \overset{=}{\geq} k$
- if you select item i, do not select item j $\quad x_i + x_j \leq 1$
- if you do not select i, do select j $\quad x_i + x_j \geq 1$
- if you select i, then also select j $\quad x_i \leq x_j$
- select either item i or j, but not both $\quad x_i + x_j = 1$
- select both item i and j, or none $\quad x_i = x_j$

Logical Conditions: Either-Or

Problem: A scheduling problem in which n tasks have to be completed sequentially (i.e., for each pair of tasks $i \neq j$ either i precedes j or vice versa)

Model Let x_i and d_i be start time and duration of task i

variables   parameter

$$x_j \geq x_i + d_i \quad \text{or} \quad x_i \geq x_j + d_j \qquad \forall i \neq j$$