CS/ECE/ISYE524: Introduction to Optimization – Integer Optimization Models

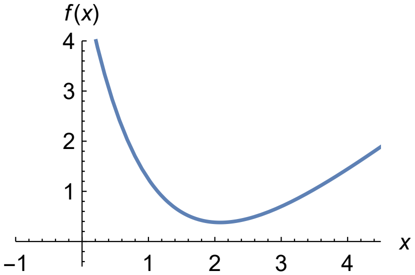
Jeff Linderoth

Department of Industrial and Systems Engineering
University of Wisconsin-Madison

April 1, 2024

Convex programs

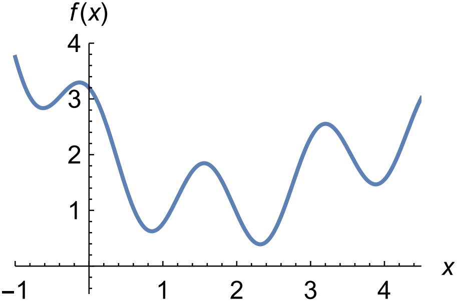
- In convex optimization, we have seen: LP, QP, QCQP, SOCP, SDP
- Can be efficiently solved
- Optimal cost can be bounded above and below
- Local optimum is global

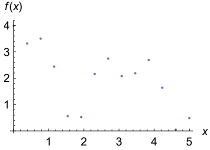


Nonconvex programs

- In general, cannot be efficiently solved
- Cost cannot be bounded easily
- Usually we can only guarantee local optimality
- Difficulty depends strongly on the instance

Continuous and Discrete problems:





Turn Focus To...

- Integer (linear) programs
  - A LP where some or all variables are discrete (boolean, integer, or general discrete-valued)
  - If all variables are integers, it's called IP or ILP
  - If variables are mixed, it's called MIP or MILP
- Nonconvex nonlinear programs
  - If continuous, it's called NLP
  - If discrete, it's called MINLP
- Approximation and relaxation
  - Can we solve solve a convex problem instead?
  - If not, can we approximate?

Discrete variables

Why are discrete variables sometimes necessary?

1. A decision variable is fundamentally discrete

- Whether a particular power plant is used or not $\{0, 1\}$
- Number of automobiles produced $\{0, 1, 2, ...\}$
- Dollar bill amount $\{\$1, \$5, \$10, \$20, \$50, \$100\}$

- But sometimes we can safely model a discrete variables as a continuous variable. Typically this is OK when the variable naturally has a large value and the quality of the solution would not suffer much from rounding down to the next integer.
- Of the examples above, the "automobile" variable may be like this.

Discrete variables

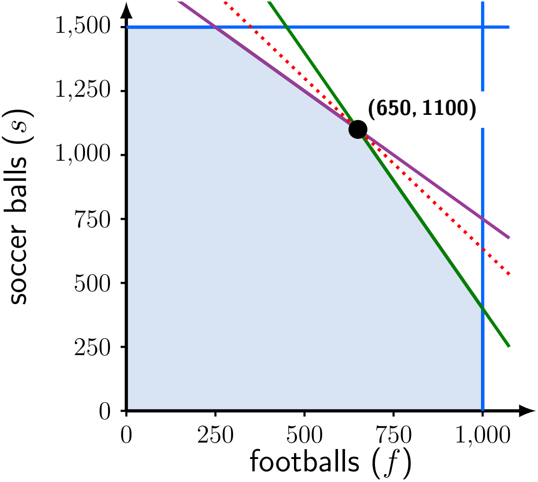
Why are discrete variables sometimes necessary?

2. Used to represent a logic constraint *algebraically*.

- "At most two of the three machines can run at once."
  $z_1 + z_2 + z_3 \leq 2$ (z_i is 1 if machine i is running, 0 otherwise)
- "If machine 1 is running, so is machine 2."
  $$z_1 \leq z_2$$
- Goal: (logic constraint) $\iff$ (LP with extra boolean variables)

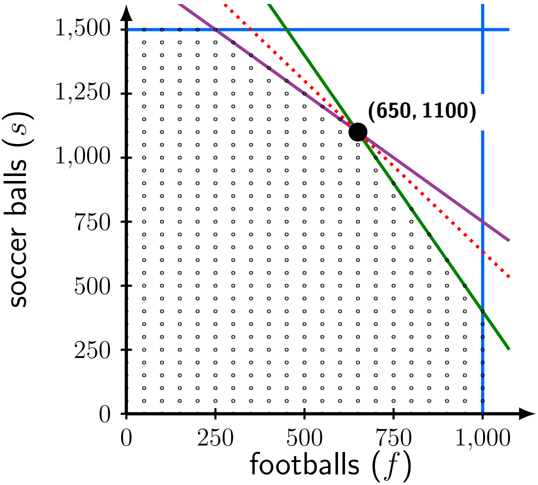
- Boolean variables take the values 0 or 1. Also known as "binary variables" or "zero-one" variables.

Return to Top Brass



$$
\begin{aligned}
\max_{f,s} \quad & 12f + 9s \\
\text{s.t.} \quad & 4f + 2s \le 4800 \\
& f + s \le 1750 \\
& 0 \le f \le 1000 \\
& 0 \le s \le 1500
\end{aligned}
$$

Return to Top Brass



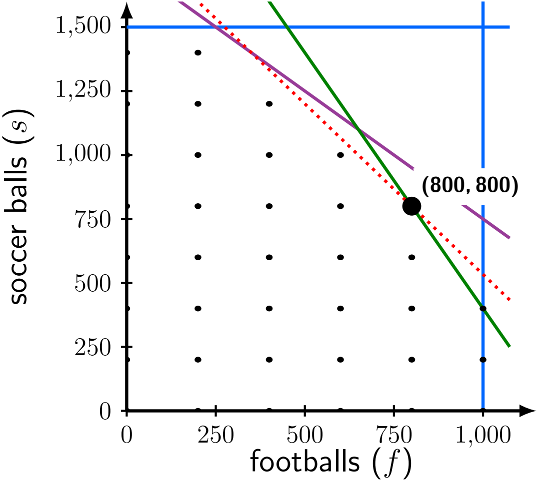
$$\max_{f,s} \quad 12f + 9s$$

$$\begin{aligned} \text{s.t.} \quad & 4f + 2s \leq 4800 \\ & f + s \leq 1750 \\ & 0 \leq f \leq 1000 \\ & 0 \leq s \leq 1500 \end{aligned}$$

f and s are multiples of 50

Same solution!

Return to Top Brass

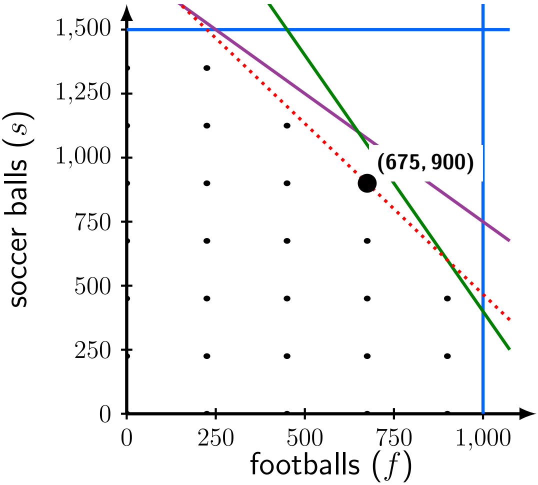


$$\max_{f,s} \quad 12f + 9s$$
$$\text{s.t.} \quad 4f + 2s \leq 4800$$
$$f + s \leq 1750$$
$$0 \leq f \leq 1000$$
$$0 \leq s \leq 1500$$

f and s are multiples of 200

Boundary solution!

Return to Top Brass



$$\max_{f,s} \quad 12f + 9s$$

$$\text{s.t.} \quad 4f + 2s \le 4800$$

$$f + s \le 1750$$

$$0 \le f \le 1000$$

$$0 \le s \le 1500$$

f and s are multiples of 225

Interior solution!

Mixed-Integer Linear Programs

$$
\begin{aligned}
\underset{x}{\text{maximize}} \quad & c^{\mathsf{T}} x \\
\text{subject to:} \quad & Ax \leq b \\
& x \geq 0 \\
& x_i \in S_i
\end{aligned}
$$

where S_i can be:

- The real numbers, $\mathbb{R}$
- The integers, $\mathbb{Z}$
- Boolean / Binary $\{0, 1\}$
- A discrete set, $\{v_1, v_2, \ldots, v_k\}$

Mixed-integer programs

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & c^\mathsf{T} x \\ \text{subject to:} \quad & Ax \le b \\ & x \ge 0 \\ & x_i \in S_i \end{aligned}$$

The solution can be

- Same as the LP version
- On a boundary
- In the interior

Rounding

Back to the standard IP formulation:

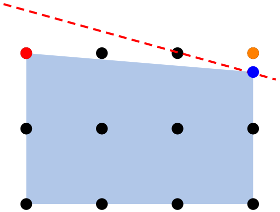
$$\begin{aligned} \underset{x}{\text{maximize}} \quad & c^\mathsf{T} x \\ \text{subject to:} \quad & Ax \leq b \\ & x \in \mathbb{Z} \end{aligned}$$

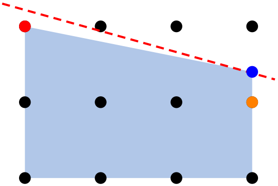
Idea:

- Solve the problem for $x \in \mathbb{R}$ instead (a regular LP).
- Round each x_i in the solution to the nearest integer.
- This *does not* work in general! (But in some cases it does.)

Rounding

- If LP solution is already integral, then it is also the exact solution to the original IP. (e.g. min cost flow problems)
- Rounding can lead to an infeasible point
- Rounding can produce a point far from the optimal point





true optimum (●), relaxed optimum (●), rounded (●)

Convex relaxation

$$\underset{x \in S}{\text{minimize}} \quad f(x)$$

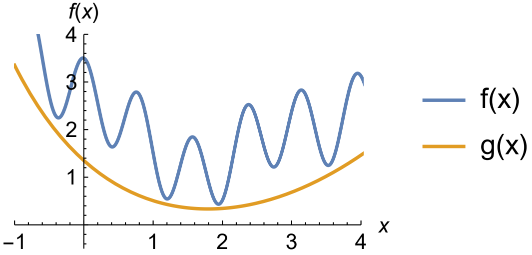
Two ideas we will discuss:

1. *Function relaxation:* if f is troublesome, bound it with a function that is easier to work with, e.g. a convex function.
2. *Constraint relaxation:* If S is troublesome, find a bigger set that is easier to work with, e.g. a convex set.

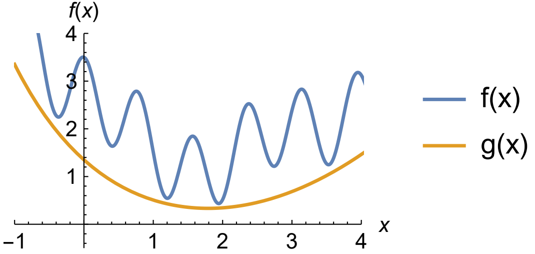
Function relaxation

$$f_{\text{opt}} = \underset{x \in S}{\text{minimize}} \;\; f(x)$$

Suppose we can find g such that $g(x) \leq f(x)$ for all x.
In other words g is a *lower bound* on f.



Function relaxation



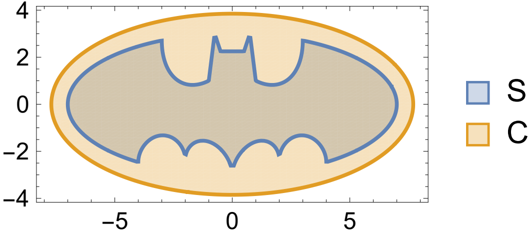
- Solve $g_{\text{opt}} = \min_{x \in S} g(x)$ and let $\hat{x}$ be the corresponding x.
- We have the bounds: $g_{\text{opt}} = g(\hat{x}) \leq f_{\text{opt}} \leq f(\hat{x})$.
- If $f(\hat{x}) = g_{\text{opt}}$ then the bound is tight and $f_{\text{opt}} = f(\hat{x})$.

Pick a convex g so that g_{opt} and $\hat{x}$ are easy to find!

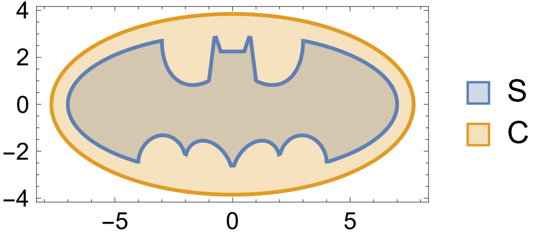
Constraint relaxation

$$f_{\mathsf{opt}} = \underset{x \in S}{\text{minimize}} \; f(x)$$

Suppose we can find some set C such that $S \subseteq C$.
In other words, C is a *superset* of S.



Constraint relaxation



- Solve $h_{\text{opt}} = \min_{x \in C} f(x)$ and let $\tilde{x}$ be the optimal x.
- We have the bound: $h_{\text{opt}} = f(\tilde{x}) \leq f_{\text{opt}} \leq f(x)$ for $x \in S$.
- If $\tilde{x} \in S$ then the bound is tight and $f_{\text{opt}} = f(\tilde{x})$.

Pick a convex C so that h_{opt} and $\tilde{x}$ are easy to find!

Common relaxations

1. Boolean constraint:

$$x \in \{0,1\} \quad \Longrightarrow \quad 0 \le x \le 1$$

If x_{opt} is 0 or 1, relaxation is exact.

2. Convex equality:

$$f(x) = 0 \quad \Longrightarrow \quad f(x) \le 0$$

If $f(x_{\text{opt}}) = 0$, relaxation is exact.

3. A constraint you don't like:

$$x \neq 3 \quad \Longrightarrow \quad \text{just remove the constraint!}$$

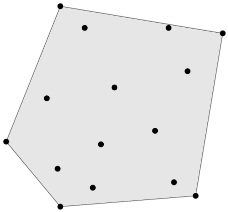
If $x_{\text{opt}} \neq 3$, relaxation is exact.

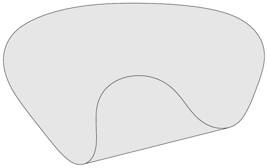
Convex hull

The **convex hull** of a set S, written $\text{conv}(S)$ is the smallest convex set that contains S.

Equivalent definitions:

- The set of all affine combinations of all points in S
- The intersection of all convex sets containing S





Common examples

- Facility location
  - locating warehouses, services, etc.
- Scheduling/sequencing
  - scheduling airline crews. (Obviously can't have a crew assigned to two flights at the same time.)
- Multicommodity flows
  - transporting many different goods across a network
- Traveling salesman problems
  - routing deliveries

Knapsack problem

My knapsack holds at most 15 kg. I have the following items:

item number	**1**	**2**	**3**	**4**	**5**
weight	12 kg	2 kg	4 kg	1 kg	1 kg
value	\$4	\$2	\$10	\$2	\$1

How can I maximize the value of the items in my knapsack?

$$\text{Let } z_i = \begin{cases} 1 & \text{knapsack contains item } i \\ 0 & \text{otherwise} \end{cases}$$

Knapsack problem

My knapsack holds at most 15 kg. I have the following items:

item number	**1**	**2**	**3**	**4**	**5**
weight	12 kg	2 kg	4 kg	1 kg	1 kg
value	\$4	\$2	\$10	\$2	\$1

How can I maximize the value of the items in my knapsack?

$$\begin{aligned} \underset{z}{\text{maximize}} \quad & 4z_1 + 2z_2 + 10z_3 + 2z_4 + z_5 \\ \text{subject to:} \quad & 12z_1 + 2z_2 + 4z_3 + z_4 + z_5 \leq 15 \\ & z_i \in \{0, 1\} \quad \text{for all } i \end{aligned}$$

notebook: Knapsack.ipynb

General (0,1) knapsack

- weights $w_1, \ldots, w_n$ and limit W.
- values $v_1, \ldots, v_n$
- decision variables $z_1, \ldots, z_n$

$$\begin{aligned} \underset{z}{\text{maximize}} \quad & \sum_{i=1}^{n} v_i z_i \\ \text{subject to:} \quad & \sum_{i=1}^{n} w_i z_i \le W \\ & z_i \in \{0,1\} \quad \text{for } i = 1, \ldots, n \end{aligned}$$