CS7800: Advanced Algorithms

Soheil Behnezhad

Linear Programming (Part I)

Basics

Examples

Slides credit: Kevin Wayne

Linear Programming

- Optimize a linear function subject to linear inequalities.

$$
\begin{array}{llrcll}
(\mathrm{P}) & \max & \sum_{j=1}^{n} c_j x_j & & & \\
& \text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j & = & b_i & 1 \le i \le m \\
& & x_j & \ge & 0 & 1 \le j \le n
\end{array}
$$

$$
\begin{array}{llrcl}
(\mathrm{P}) & \max & c^T x & & \\
& \text{s.t.} & Ax & = & b \\
& & x & \ge & 0
\end{array}
$$

- Generalizes shortest paths, max flow, assignment problems, matching, multicommodity flow, MST, …
- LPs are used extensively in:
  - Designing poly-time algorithms.
  - Designing/analyzing approximation algorithms.

A sweet-spot between generality and solvability

Brewery Problem

- Small brewery produces ale and beer.
- Production limited by scarce resources: corn, hops, barley malt.
- Recipes for ale and beer require different proportions of resources.

Beverage	Corn (pounds)	Hops (ounces)	Malt (pounds)	Profit ($)
Ale (barrel)	5	4	35	13
Beer (barrel)	15	4	20	23
constraint	480	160	1190	

- How can brewer maximize profits?
  - Devote all resources to ale: 34 barrels of ale => $442
  - Devote all resources to beer: 32 barrels of beer => $736
  - 7.5 barrels of ale, 29.5 barrels of beer => $776
  - 12 barrels of ale, 28 barrels of beer => 800

Brewery Problem

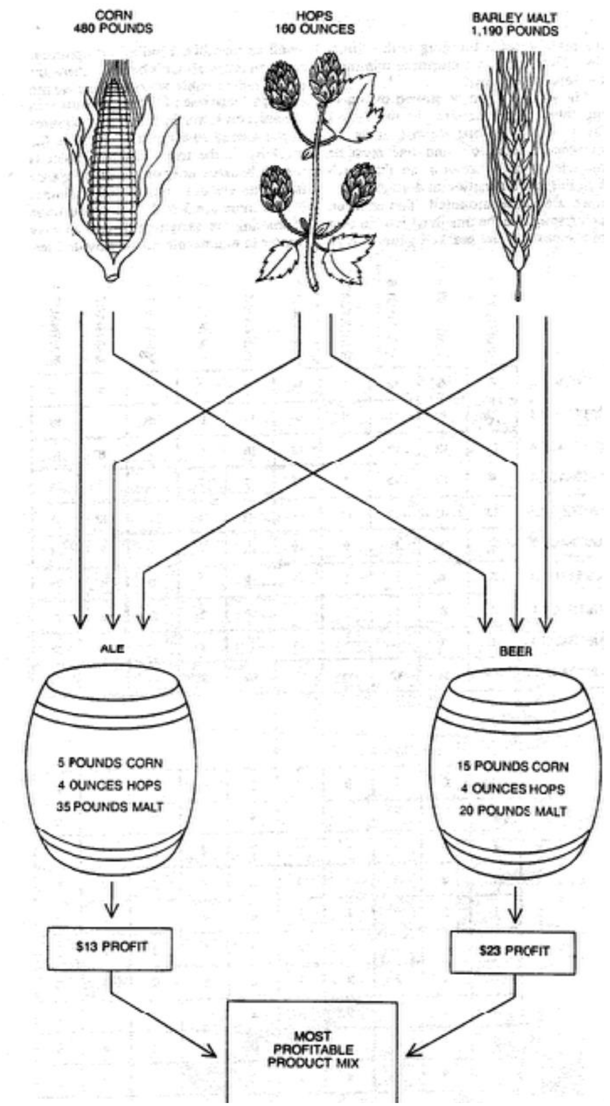
objective function

$$
\begin{array}{lccccccl}
 & \text{Ale} & & \text{Beer} & & & & \\
\max & 13A & + & 23B & & & & \text{Profit} \\
\text{s.t.} & 5A & + & 15B & \le & 480 & & \text{Corn} \\
 & 4A & + & 4B & \le & 160 & & \text{Hops} \\
 & 35A & + & 20B & \le & 1190 & & \text{Malt} \\
 & A & , & B & \ge & 0 & &
\end{array}
$$

constraint

decision variable

Beverage	Corn (pounds)	Hops (ounces)	Malt (pounds)	Profit ($)
Ale (barrel)	5	4	35	13
Beer (barrel)	15	4	20	23
constraint	480	160	1190	



Standard Form

- "Standard form" of an LP.
  - Input: real numbers a_{ij}, c_j, b_i.
  - Output: real numbers x_j.
  - $n =$ # of decision variables, $m =$ # of constraints.
  - Maximize linear objective function subject to linear constraints.

$$
\begin{array}{llrcll}
\text{(P)} & \max & \sum_{j=1}^{n} c_j x_j & & & \\
& \text{s. t.} & \sum_{j=1}^{n} a_{ij} x_j & = & b_i & 1 \le i \le m \\
& & x_j & \ge & 0 & 1 \le j \le n
\end{array}
$$

$$
\begin{array}{llrcl}
\text{(P)} & \max & c^T x & & \\
& \text{s. t.} & Ax & = & b \\
& & x & \ge & 0
\end{array}
$$

- Linear: No $x^2, xy, \arccos(x)$, etc.
- Programming: planning (predates computer programming).

Brewery: Converting to Standard Form

$$
\begin{array}{lrcrcr}
\max & 13A & + & 23B & & \\
\text{s.t.} & 5A & + & 15B & \le & 480 \\
 & 4A & + & 4B & \le & 160 \\
 & 35A & + & 20B & \le & 1190 \\
 & A & , & B & \ge & 0
\end{array}
$$

Brewery LP

$$
\begin{array}{llrcl}
(\mathrm{P}) & \max & c^T x & & \\
 & \text{s.t.} & Ax & = & b \\
 & & x & \ge & 0
\end{array}
$$

LP Standard form

- How do we convert the brewery LP to standard form?
  - Add a slack variable for each inequality.
  - Now a 5-dimensional problem.

$$
\begin{array}{lrcrcrcrcrcr}
\max & 13A & + & 23B & & & & & & & & \\
\text{s.t.} & 5A & + & 15B & + & S_C & & & & & = & 480 \\
 & 4A & + & 4B & & & + & S_H & & & = & 160 \\
 & 35A & + & 20B & & & & & + & S_M & = & 1190 \\
 & A & , & B & , & S_C & , & S_H & , & S_M & \ge & 0
\end{array}
$$

Converting to Standard Form

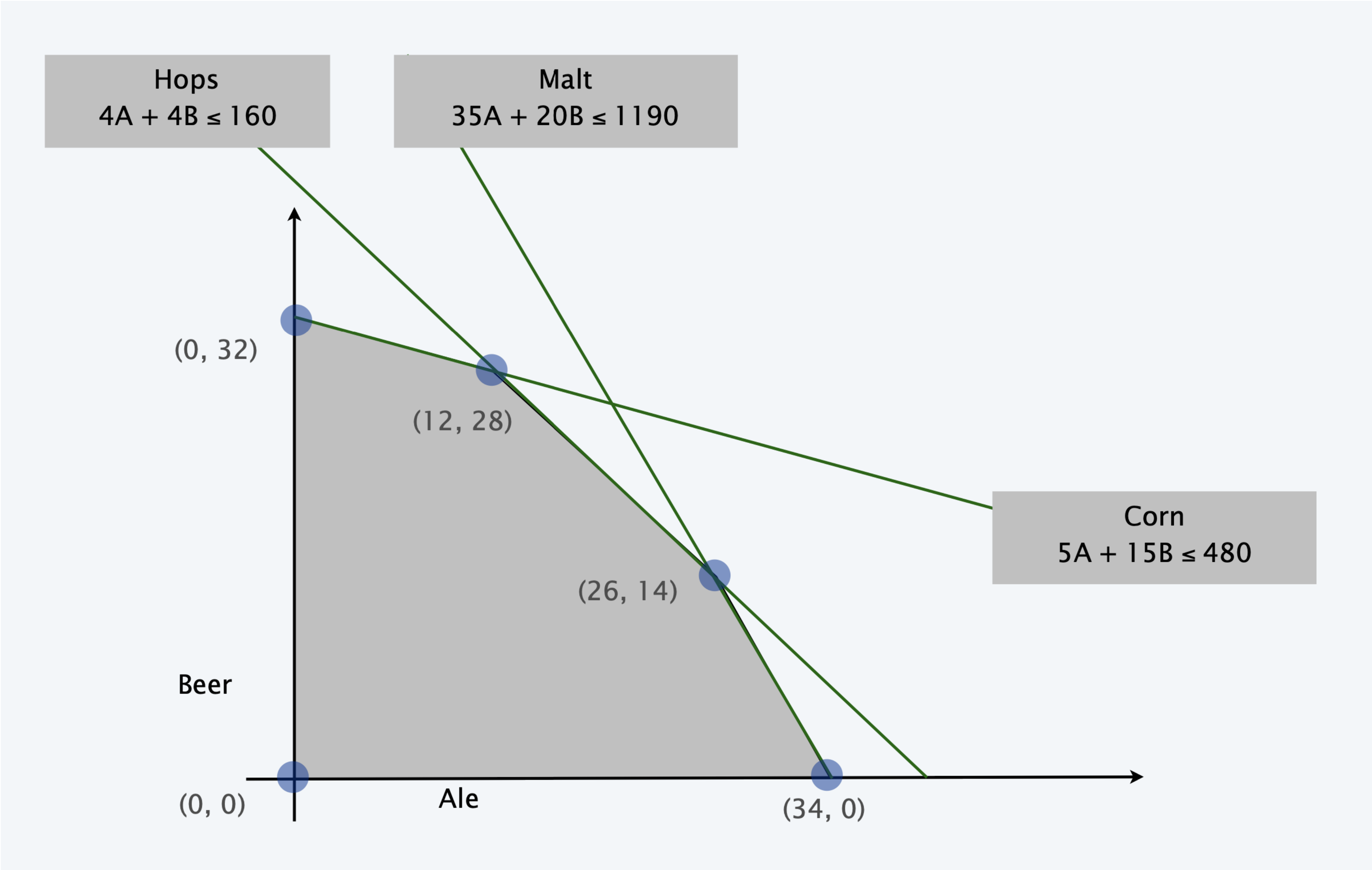
- It's easy to convert an LP to standard form.

$$
\begin{array}{llrcl}
(\mathrm{P}) & \max & c^T x & & \\
 & \mathrm{s.t.} & Ax & = & b \\
 & & x & \geq & 0
\end{array}
$$

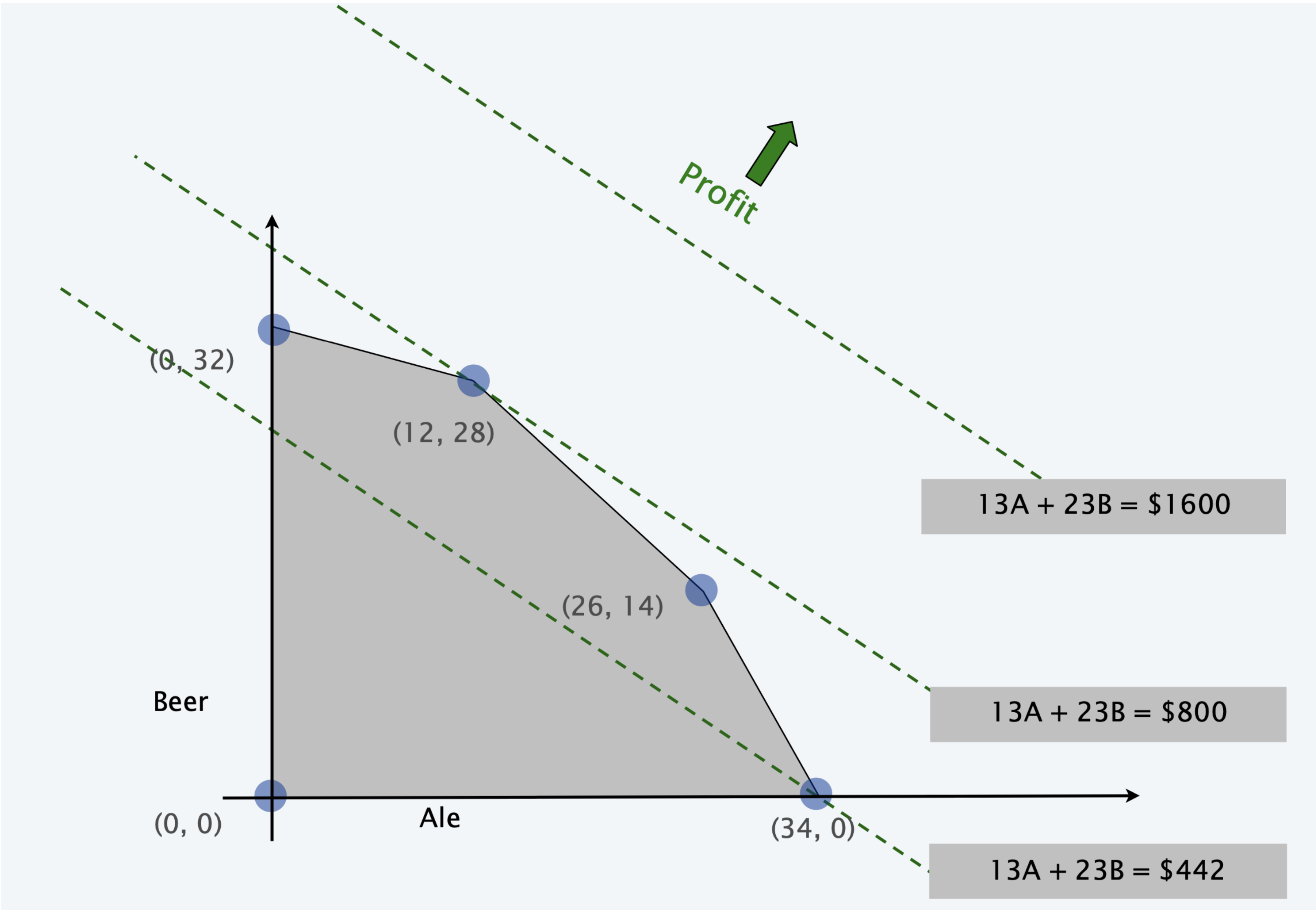
LP Standard form

- $\leq$ to =: $x + 2y - 3z \leq 17 \Rightarrow x + 2y - 3z + s = 17, s \geq 0$
- $\geq$ to =: $x + 2y - 3z \geq 17 \Rightarrow x + 2y - 3z - s = 17, s \geq 0$
- Min to max: $\min x + 2y - 3z \Rightarrow \max -x - 2y + 3z$
- Unrestricted to nonnegative:
  x unrestricted $\Rightarrow x = x^+ - x^-, x^+ \geq 0, x^- \geq 0$

Geometry of LPs: Back to Brewery

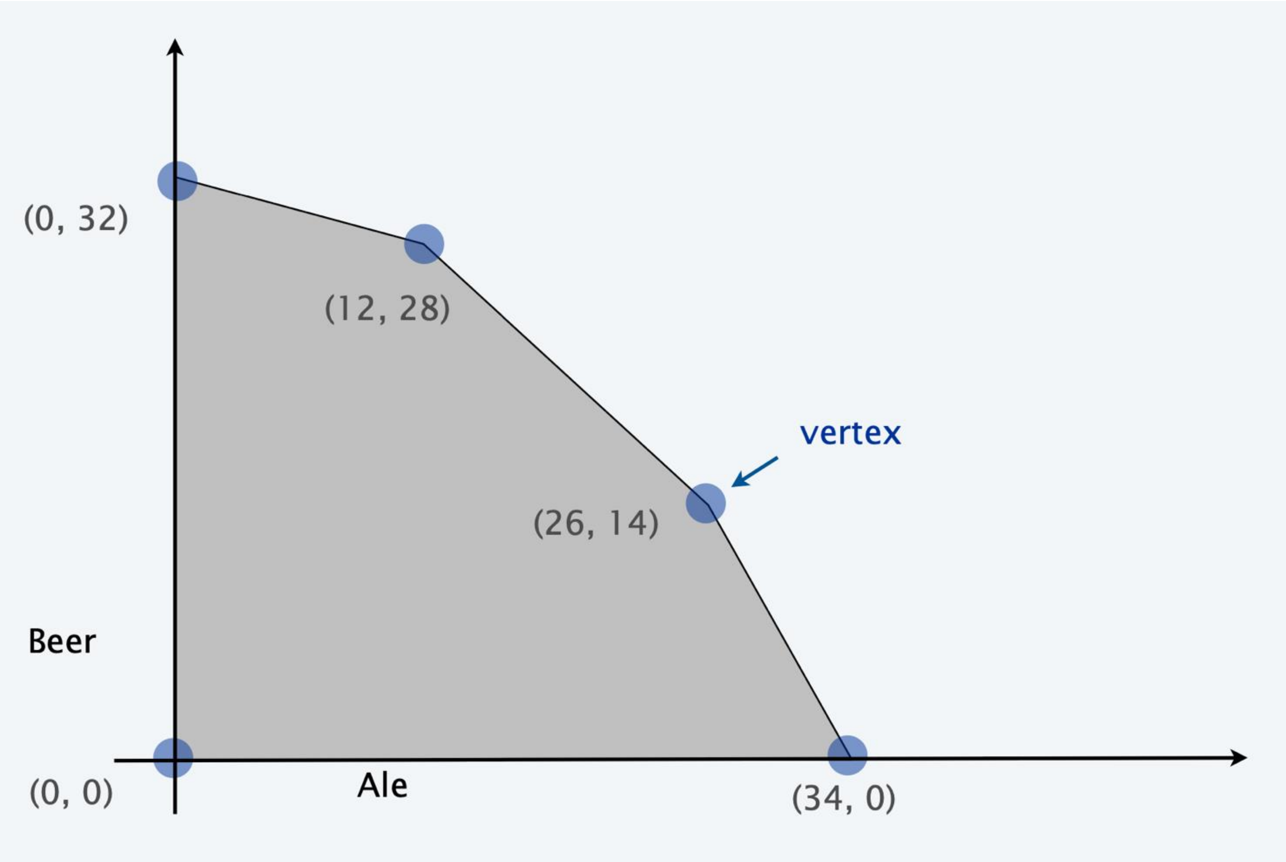


Geometry of LPs: Back to Brewery



Geometry of LPs: Back to Brewery

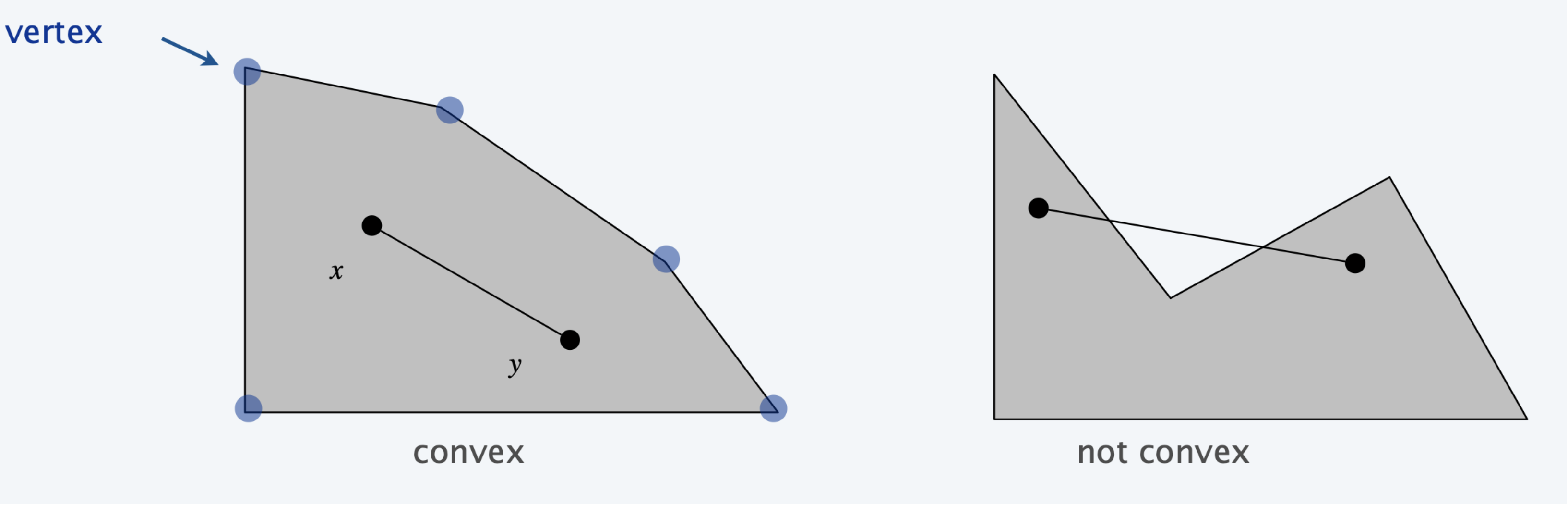
- Brewery problem observation: Regardless of objective function coefficients, an optimal solution occurs at a vertex.



Convexity

- Convex set: If two points x and y are in the set, then so is $\lambda x + (1 - \lambda)y$ for any $0 \leq \lambda \leq 1$.

- Vertex: A point x in the set that can't be written as a strict convex combination of two distinct points in the set.



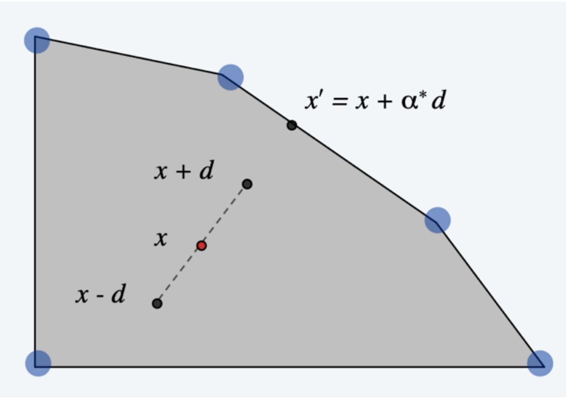
- Observation: The feasible region of an LP is a convex set.

A vertex is optimal

- Theorem: If there is an optimal solution to (P) then there is one that is a vertex.

$$
\begin{array}{llrcl}
\text{(P)} & \max & c^T x & & \\
& \text{s.t.} & Ax & = & b \\
& & x & \geq & 0
\end{array}
$$

- Intuition: Start from an optimal solution, move in a direction that does not decrease the objective value (one such direction must exist) until you reach a boundary. Repeat.



Examples: One Shortest Path

- Input: Weighted directed graph G=(V, E), weight function $\ell : E \rightarrow R$ and two vertices $s, t \in V$.
- Output: Length of the shortest path from s to t.

Examples: One Shortest Path

- Input: Weighted directed graph G=(V, E), weight function $\ell: E \to R$ and two vertices $s, t \in V$.
- Output: Length of the shortest path from s to t.

$$
\begin{array}{lll}
\text{maximize} & dist(t) & \\
\text{subject to} & dist(s) = 0 & \\
 & dist(v) - dist(u) \leq \ell(u\to v) & \text{for every edge } u\to v
\end{array}
$$

- This LP is feasible iff the graph has no negative cycles.
- The shortest path distances form a feasible solution to the LP.
- Counterintuitively, this maximization LP solves a minimization problem.
- Is the LP solution unique?

Examples: One Shortest Path, Take two

- Input: Weighted directed graph G=(V, E), weight function $\ell: E \rightarrow R$ and two vertices $s, t \in V$.
- Output: Length of the shortest path from s to t.

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{u \to v} \ell(u\to v) \cdot x(u\to v) & \\
\text{subject to} & \displaystyle\sum_{u} x(u\to t) - \sum_{w} x(t\to w) = 1 & \\
 & \displaystyle\sum_{u} x(u\to v) - \sum_{w} x(v\to w) = 0 & \text{for every vertex } v \neq s, t \\
 & x(u\to v) \geq 0 & \text{for every edge } u\to v
\end{array}
$$

- Instead of variables on the vertices, this LP variables $x(u \rightarrow v)$ that intuitively indicate which edges belong to the shortest path from s to t.
- This LP has an integral solution. This is a special property of this specific LP. Generally, LPs with integer coefficients don’t necessarily have integral solutions.

Examples: Single Source Shortest Path

- Input: Weighted directed graph G=(V, E), weight function $\ell: E \to R$ and vertex $s \in V$.
- Output: Length of the shortest path from s to every other vertex.

Examples: Single Source Shortest Path

- Input: Weighted directed graph G=(V, E), weight function $\ell: E \to R$ and vertex $s \in V$.
- Output: Length of the shortest path from s to every other vertex.

- The problem with our previous maximization LP for one shortest path is that it maximizes $dist(t)$ and so the distance to vertices not on the path from s to t could be incorrect.
- The following modification to the LP makes sure that all distances are right:

$$\begin{array}{lll} \text{maximize} & \sum_{v} dist(v) & \\ \text{subject to} & dist(s) = 0 & \\ & dist(v) - dist(u) \leq \ell(u \to v) & \text{for every edge } u \to v \end{array}$$

- Again, we are using maximization to solve a minimization problem!

Examples: Single Source Shortest Path

- Input: Weighted directed graph G=(V, E), weight function $\ell: E \to R$ and vertex $s \in V$.
- Output: Length of the shortest path from s to every other vertex.

- We can also modify the second LP to find a shortest path tree

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{u\to v} \ell(u\to v)\cdot x(u\to v) & \\
\text{subject to} & \displaystyle\sum_{u} x(u\to v) - \sum_{w} x(v\to w) = 1 & \text{for every vertex } v \neq s \\
 & x(u\to v) \geq 0 & \text{for every edge } u\to v
\end{array}
$$

Examples: Max Flow

- Input: Weighted directed graph G=(V, E), non-negative capacity function $c: E \rightarrow R^{\geq 0}$ and special vertices $s, t \in V$.
- Output: Value of maximum flow from s to v.

Examples: Max Flow

- Input: Weighted directed graph G=(V, E), non-negative capacity function $c: E \rightarrow R^{\geq 0}$ and special vertices $s, t \in V$.
- Output: Value of maximum flow from s to v.

$$
\begin{array}{lll}
\text{maximize} & \displaystyle\sum_w f(s\to w) - \sum_u f(u\to s) & \\
\text{subject to} & \displaystyle\sum_w f(v\to w) - \sum_u f(u\to v) = 0 & \text{for every vertex } v \neq s,t \\
& f(u\to v) \leq c(u\to v) & \text{for every edge } u\to v \\
& f(u\to v) \geq 0 & \text{for every edge } u\to v
\end{array}
$$

Examples: Min Cut

- Input: Weighted directed graph G=(V, E), non-negative capacity function $c: E \rightarrow R^{\geq 0}$ and special vertices $s, t \in V$.
- Output: Value of the s, t cut with minimum capacity.

Examples: Min Cut

- Input: Weighted directed graph G=(V, E), non-negative capacity function $c: E \rightarrow R^{\geq 0}$ and special vertices $s, t \in V$.
- Output: Value of the (s, t)-cut with minimum capacity.

$$
\begin{array}{lrl}
\text{minimize} & \displaystyle\sum_{u \to v} c(u \to v) \cdot x(u \to v) & \\
\text{subject to} & x(u \to v) + S(v) - S(u) \geq 0 & \text{for every edge } u \to v \\
 & x(u \to v) \geq 0 & \text{for every edge } u \to v \\
 & S(s) = 1 & \\
 & S(t) = 0 &
\end{array}
$$

Examples: Maximum Bipartite Matching

- Input: Unweighted undirected bipartite graph G=(V, E)
- Output: Size of maximum matching in G.

Examples: Maximum Non-Bipartite Matching

- Input: Unweighted undirected general graph G=(V, E)
- Output: Size of maximum matching in G.