Introduction to Linear Programming (LP)

Speaker: David Wajc

Talk given as part of: Technion CS Dept. Algorithms Seminar (236813)

Introduction: Motivating Example

Bob has reached the conclusion that his weight is becoming an issue, as he keeps arriving late to classes he has to run across campus to get to.

He would like to put himself on a diet which will help him lose weight while still getting his RDA of important nutritional elements.

Bob wonders how few calories he can eat a day, in order to get at least the recommended dietary allowance (RDA), part of which is given below:

	Protein (g)	Calcium (mg)
RDA	50	800

The foods Bob is willing to eat and their nutritional values per serving are given below. Bob is unwilling to have more than 2 servings of milk a day.

Food	Serving size	Energy(kcal)	Protein (g)	Calcium (mg)
Soup Bowl	245 g	150	10	20
Milk	237 ml	120	10	360
Cereals ™	40 g	127	3	0
Shawarma	300 g	519	75	50
Apple	142 g	55	0	8

Extra constraint:
Milk/day ≤ 2

	Protein (g)	Calcium (mg)
RDA	50	800

	Food	Serving size	Energy(kcal)	Protein (g)	Calcium (mg)
x_1	**Soup Bowl**	245 g	150	10	20
x_2	**Milk**	237 ml	120	10	360
x_3	**Cereals ™**	40 g	127	3	0
x_4	**Shawarma**	300 g	519	75	50
x_5	**Apple**	142 g	55	0	8

Unsatisfactory menu:
$x = (1,5,3,0,2)$ - too much milk.
$x = (1,2,3,0,2)$ - does not provide enough protein.

Satisfactory menus:
$x = (1,2,3,1,2)$, total energy intake: 1400 kcal.
$x = (3,2,0,0,2.5)$, total energy intake: 827.5 kcal.

Talk Outline

- Introduction to Linear Programming
    1. Problem Definition and Terminology
    2. Geometric Interpretation
    3. More Motivating Examples
    4. Duality

- Integer Linear Programming

- Rounding Fractional LP in Approximation Algorithms :
    1. Rounding to approximate Set Cover
    2. Randomized Rounding to approximate Set Cover

Introduction: What is the Problem?

Find an n-dimensional vector x minimizing $\sum_{i=1}^{n} c_i x_i$ subject to constraints of the form:

$$\sum_{i=1}^{n} a_{j,i} x_i \geq b_j \text{ for } j \in \{1,\ldots,m\}$$

$$x_i \geq 0 \text{ for } i \in \{1,\ldots,n\}$$

The above is an LP in Canonical Form. It can be written succinctly as:

$$\min c^T x$$

$$\text{s.t. } Ax \geq b \text{ and } x \geq 0$$

Introduction: What is the Problem?

General Form: Find an n-dimensional vector, x, which minimizes (or maximizes) a linear function, subject to linear equality/inequality constraints.

Minimize/maximize $\sum_{i=1}^{n} c_i x_i$

subject to:

$$\sum_{i=1}^{n} a_{j,i} x_i \geq b_j \text{ for } j \in \{1, \ldots, p\}$$

$$\sum_{i=1}^{n} a_{j,i} x_i = b_j \text{ for } j \in \{p+1, \ldots, p+q\}$$

$$\sum_{i=1}^{n} a_{j,i} x_i \leq b_j \text{ for } j \in \{p+q+1, \ldots, m\}$$

Extra constraint:

Milk/day ≤ 2

	Protein (g)	Calcium (mg)
RDA	50	800

	Food	Serving size	Energy(kcal)	Protein (g)	Calcium (mg)
x_1	**Soup Bowl**	245 g	150	10	20
x_2	**Milk**	237 ml	120	10	360
x_3	**Cereals ™**	40 g	127	3	0
x_4	**Shawarma**	300 g	519	75	50
x_5	**Apple**	142 g	55	0	8

Our problem as an LP:

Minimize $150x_1 + 120x_2 + 127x_3 + 519x_4 + 55x_5$

Subject to
$$10x_1 + 10x_2 + 3x_3 + 75x_4 \geq 50$$
$$20x_1 + 360x_2 + 50x_4 + 8x_5 \geq 800$$
$$x_2 \leq 2$$
$$x_1 x_2, x_3, x_4, x_5 \geq 0$$

Introduction: Terminology

Terms from last lesson:

1. Objective function.
2. Feasible Solution.
3. Optimal Solution/Value.

Terms we haven't yet encountered:

4. A problem is infeasible if it has no feasible solution.
5. A problem is unbounded if it has feasible solutions, but no optimal solution.

Talk Outline

- Introduction to Linear Programming
  1. Problem Definition and Terminology
  2. Geometric Interpretation
  3. More Motivating Examples
  4. Duality

- Integer Linear Programming

- Rounding Fractional LP in Approximation Algorithms :
  1. Rounding to approximate Set Cover
  2. Randomized Rounding to approximate Set Cover

2-D Geometric Example

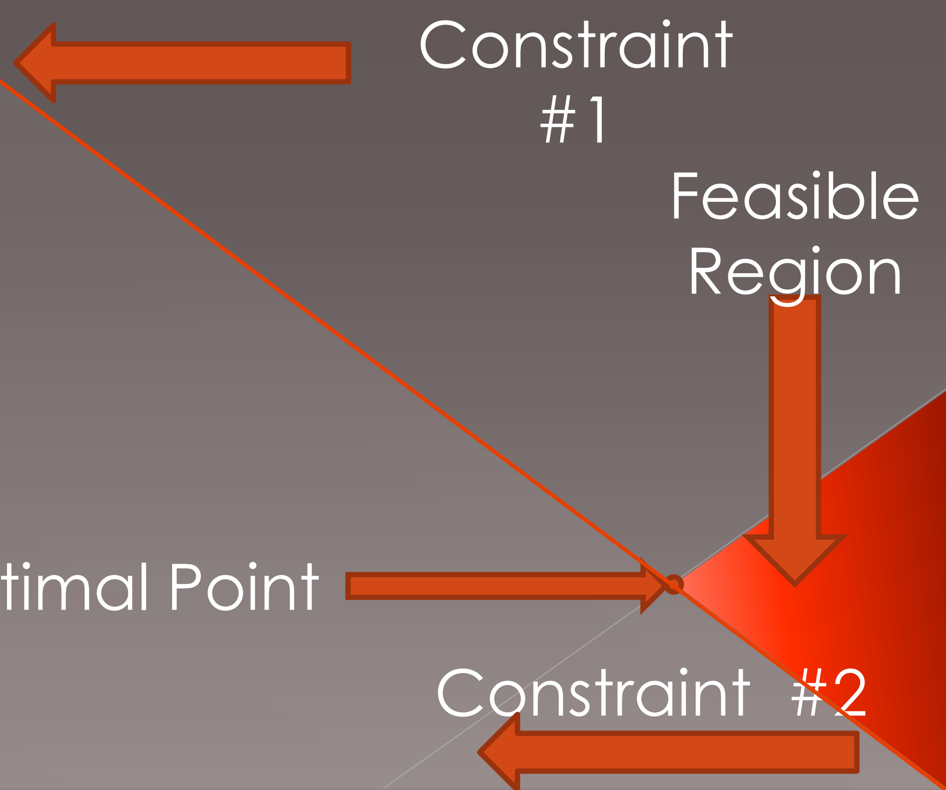
Consider the LP:

$$\min(x)$$

Subject to:

$$(18/25)x + y \geq 18$$

$$(-1/2)x + y \leq -15/2$$



LP - Geometric Interpretation:

- Find a point in n-dimensional space, x, within some convex polytope, which is furthest away from some half-plane.

- Each constraint defines a half-space in n-dimensional space. This is a face of the polytope, or Feasible Region.

$$\sum_{i \in [n]} a_{j,i} x_i \geq b_j \text{ for } j \in [m]$$

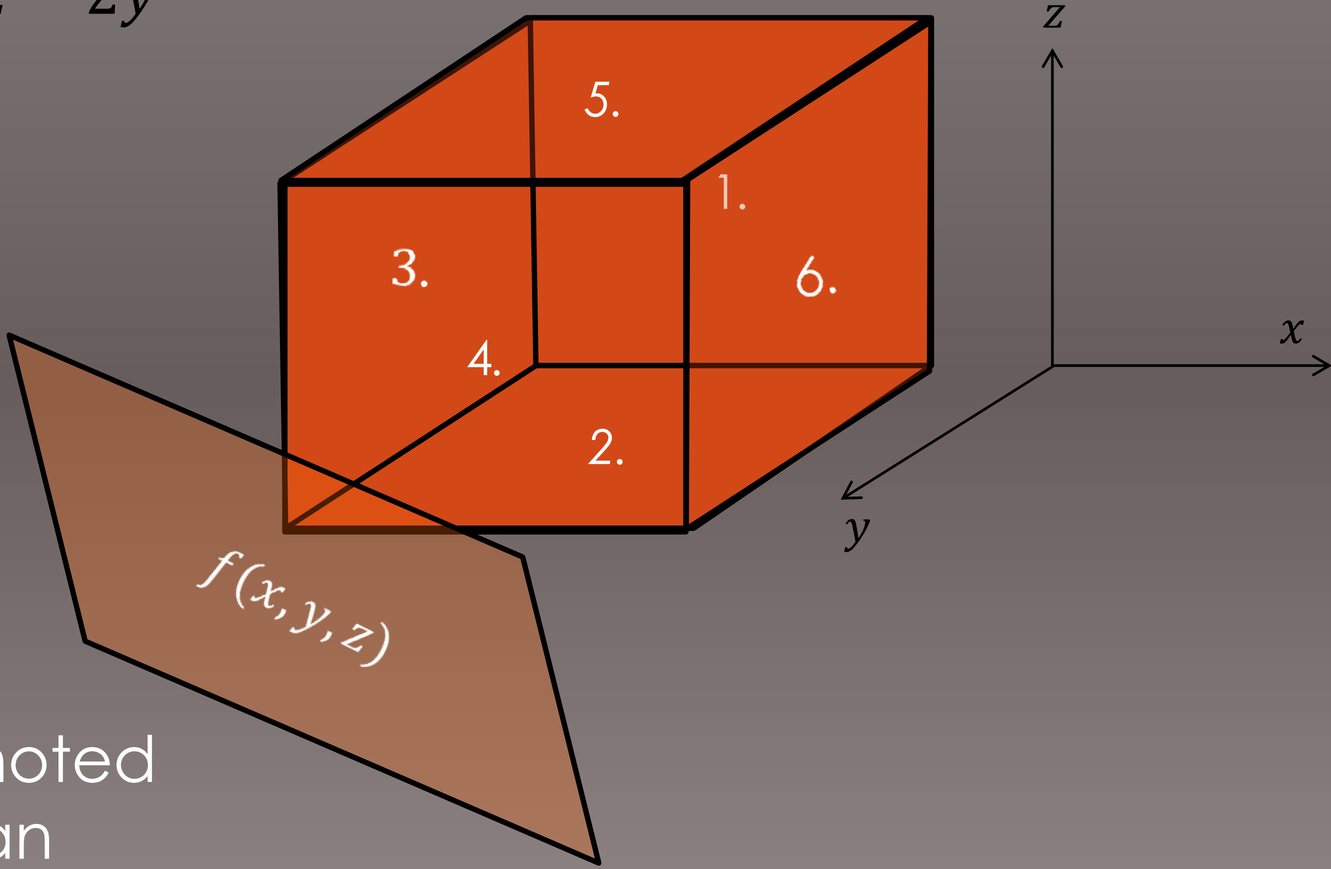
3-D Geometric Example

Consider the LP:

Maximize $x + z - 2y$

Subject to:

1. $y \geq 0$
2. $z \geq 0$
3. $x \geq -5/4$
4. $y \leq 1$
5. $z \leq 1$
6. $x \leq -1/4$



The plane denoted by $f(x, y, z)$ is an isosurface.

Higher Dimension Example?

Talk Outline

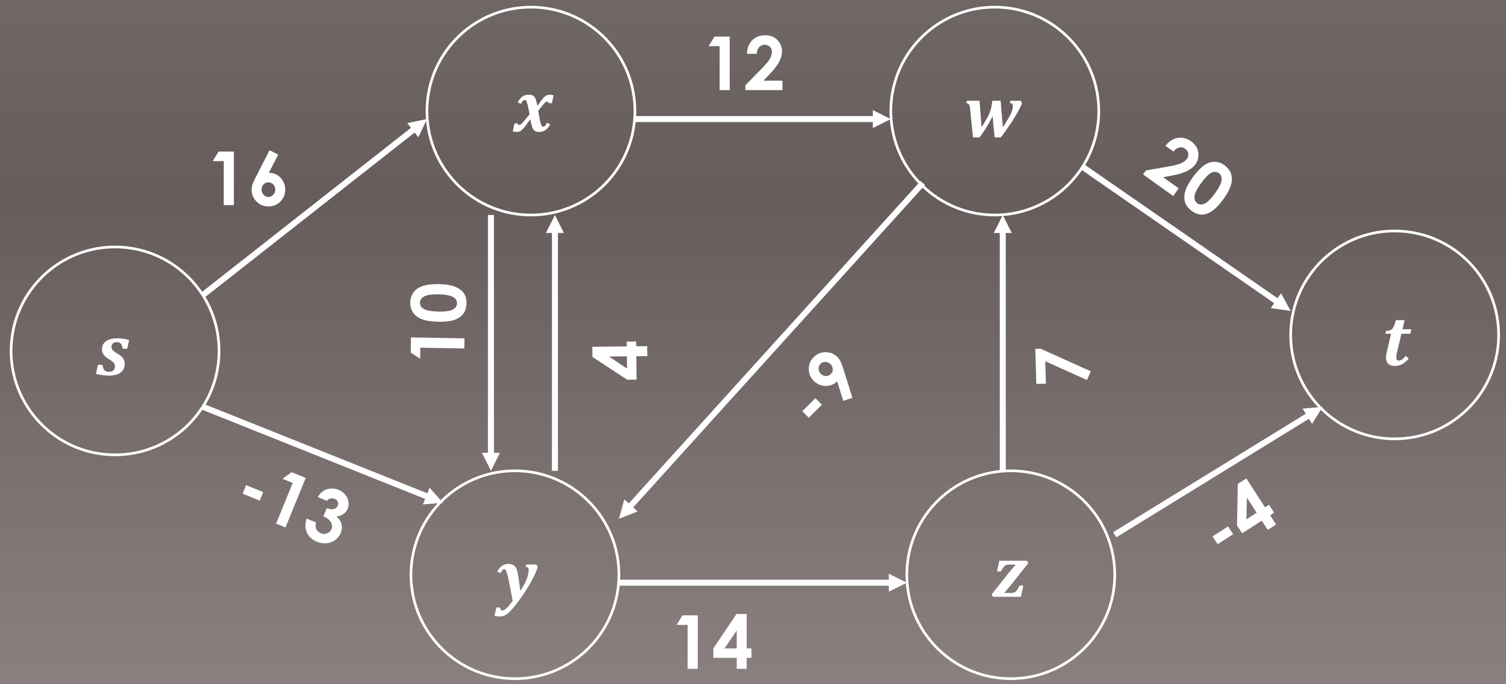
- Introduction to Linear Programming
  1. Problem Definition and Terminology
  2. Geometric Interpretation
  3. More Motivating Examples
  4. Duality

- Integer Linear Programming

- Rounding Fractional LP in Approximation Algorithms :
  1. Rounding to approximate Set Cover
  2. Randomized Rounding to approximate Set Cover

Single Source Shortest Path (SSSP)

SSSP (reminder): Given a weighted directed graph $G = (V, E)$ and vertex $s \in V$, find, for each $v \in V$, $dist(s, v)$: the minimal weight of a path from s to v.



Single Source Shortest Path (SSSP)

SSSP (reminder): Given a weighted directed graph $G = (V, E)$ and vertex $s \in V$, find, for each $v \in V$, $dist(s, v)$: the minimal weight of a path from s to v.

LP Formulation

Define d_v for each vertex v. The LP is:

$$\max \sum_{v \in V} d_v$$

$$\text{subject to } d_s = 0$$

$$d_v \le d_u + w(u, v) \quad \forall (u,v) \in E$$

SSSP – LP Formulation

Define d_v for each vertex v. The LP is:

$$\max \sum_{v \in V} d_v \quad \text{subject to } d_s = 0$$

$$d_v \le d_u + w(u,v) \qquad \forall (u,v) \in E$$

Proof of correctness:

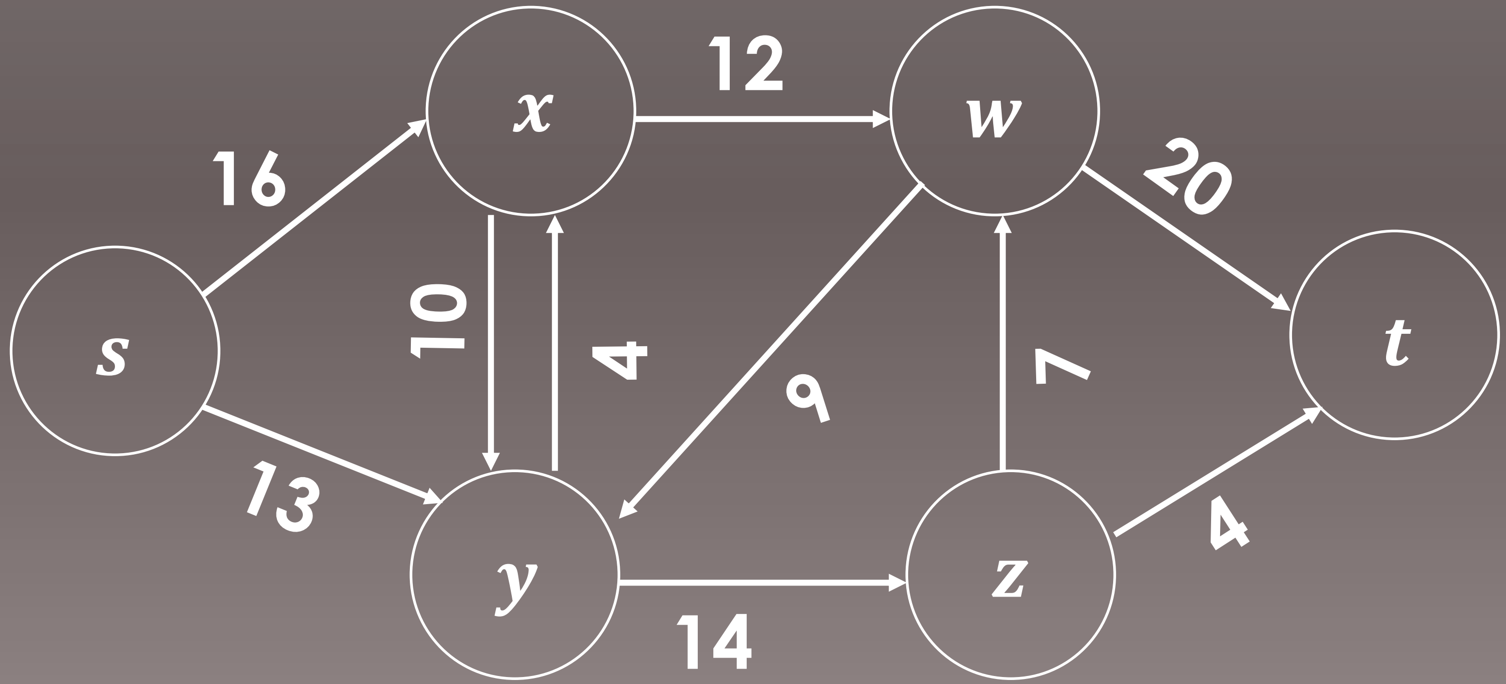
For every v and path p: $s = v_1 \to v_2 \to \cdots \to v_{k-1} \to v_k = v$
we have $d_v \le w(p) = \Sigma_{i=1}^{k-1} w(v_i, v_{i+1})$. $(\star)$

In particular, $d_v \le dist(s,v)$.
Therefore $\Sigma_v d_v \le \Sigma_v(dist(s,v))$.

Finally, if d: $d_v = dist(s,v)$ is a feasible solution, it is therefore the only optimal solution.

Maximum Flow

Max Flow(reminder): Given a directed capacitated graph $G = (V, E)$ and vertices s, t, find the maximum outgoing flow from s to t, subject to edge capacities and flow preservation in vertices $v \neq s, t$.



Maximum Flow LP Formulation

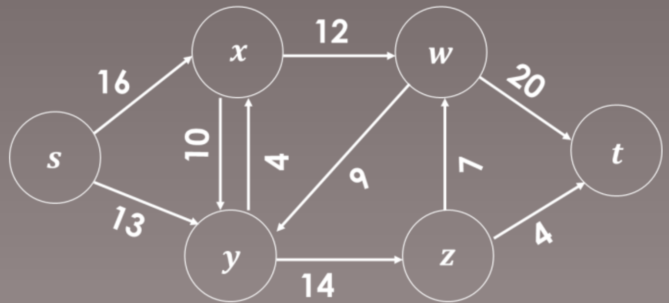
Define $f(u,v)$ for every edge (u,v). The LP:

$$\max \sum_{v \in adj(s)} f(s,v)$$

$$\text{subject to } \quad f(u,v) \le c(u,v) \qquad \forall (u,v) \in E$$

$$f(u,v) \ge 0 \qquad \forall (u,v) \in E$$

$$\sum_{v \in V} f(u,v) = \sum_{w \in V} f(w,u) \qquad \forall s,t \ne u \in V$$



Talk Outline

- Introduction to Linear Programming
  1. Problem Definition and Terminology
  2. Geometric Interpretation
  3. More Motivating Examples
  4. Duality

- Integer Linear Programming

- Rounding Fractional LP in Approximation Algorithms :
  1. Rounding to approximate Set Cover
  2. Randomized Rounding to approximate Set Cover

Back to Our Diet Problem's LP:

Minimize $150x_1 + 120x_2 + 127x_3 + 519x_4 + 55x_5$

Subject to
$$10x_1 + 10x_2 + 3x_3 + 75x_4 \geq 50$$
$$20x_1 + 360x_2 + 50x_4 + 8x_5 \geq 800$$
$$x_2 \leq 2$$
$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Best feasible solution found so far had value 827.5. Could we have done better? Can we give lower bounds on the objective function?

Since $x_1, x_2, x_3, x_4, x_5 \geq 0$, we find that :

$$f(x) = 150x_1 + 120x_2 + 127x_3 + 519x_4 + 55x_5 \geq$$
$$10x_1 + 10x_2 + 3x_3 + 75x_4 \geq 50$$

Bob's Diet Problem – the LP:

Minimize $150x_1 + 120x_2 + 127x_3 + 519x_4 + 55x_5$

Subject to
$$10x_1 + 10x_2 + 3x_3 + 75x_4 \geq 50$$
$$20x_1 + 360x_2 + 50x_4 + 8x_5 \geq 800$$
$$x_2 \leq 2$$
$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Can we get better bounds?
Sure. Consider lower bounds given by linear combinations of the constraints:

$$f(x) = 150x_1 + 120x_2 + 127x_3 + 519x_4 + 55x_5 \geq \frac{1}{3}(20x_1 + 360x_2 + 0x_3 + 50x_4 + 8x_5) \geq \frac{800}{3}$$

$$f(x) = 150x_1 + 120x_2 + 127x_3 + 519x_4 + 55x_5 \geq 1 \cdot (20x_1 + 360x_2 + 0x_3 + 50x_4 + 8x_5) + 240 \cdot (0x_1 - 1x_2 + 0x_3 + 0x_4 + 0x_5) \geq 320$$

Bob's Diet Problem – the LP:

Minimize $150x_1 + 120x_2 + 127x_3 + 519x_4 + 55x_5$

Subject to

$$10x_1 + 10x_2 + 3x_3 + 75x_4 \geq 50$$
$$20x_1 + 360x_2 + 50x_4 + 8x_5 \geq 800$$
$$x_2 \leq 2$$
$$x_1, x_2, x_3, x_4, x_5 \geq 0$$

Our final bound:

$$f(x) = 150x_1 + 120x_2 + 127x_3 + 519x_4 + 55x_5 \geq$$
$$\frac{5}{4}(10x_1 + 10x_2 + 3x_3 + 75x_4 + 0x_5) +$$
$$\frac{55}{8}(20x_1 + 360x_2 + 0x_3 + 50x_4 + 8x_5) +$$
$$2367.5(-1x_2 + 0x_3 + 0x_4 + 0x_5) \geq 827.5$$

LP Duality

Recall, we're searching for a vector x, minimizing $\sum_{i=1}^{n} c_i x_i$

$$\text{subject to} \sum_{i=1}^{n} a_{j,i} x_i \geq b_j \ \text{for} \ j \in \{1, \ldots, m\}$$

$$x_i \geq 0 \ \text{for} \ i \in \{1, \ldots, n\}$$

We call the above the Primal LP.

Next, define a non-negative variable y_j for each constraint of the Primal LP.

The Dual LP: Find y Maximizing $\sum_{j=1}^{m} b_j y_j$

$$\text{subject to} \sum_{j=1}^{m} a_{j,i} y_j \leq c_i \ \text{for} \ i \in \{1, \ldots, n\}$$

$$y_j \geq 0 \ \text{for} \ j \in \{1, \ldots, m\}$$

Duality Theorems

Primal LP (P)	Dual LP (D)
$\min \sum_{i\in[n]} c_i \cdot x_i$ subject to $\sum_{i\in[n]} a_{j,i} x_i \geq b_j, \quad \forall j \in [m]$ $x_i \geq 0, \quad \forall i \in [n]$	$\max \sum_{j\in[m]} b_j \cdot y_j$ subject to $\sum_{j\in[m]} y_j a_{j,i} \leq c_i, \quad \forall i \in [n]$ $y_j \geq 0, \quad \forall j \in [m]$

Weak Duality Theorem: Let x and y be feasible solutions of (P) and (D), respectively. Then

$$\sum_{i\in[n]} c_i \cdot x_i \geq \sum_{j\in[m]} b_j \cdot y_j$$

Proof:

$$\sum_{i\in[n]} c_i \cdot x_i \geq \sum_{i\in[n]} \left(\sum_{j\in[m]} y_j a_{j,i} \right) \cdot x_i$$

$$= \sum_{j\in[m]} \left(\sum_{i\in[n]} a_{j,i} x_i \right) \cdot y_j \geq \sum_{j\in[m]} b_j \cdot y_j$$

Q.E.D.

Duality Theorems

Primal LP (P)	Dual LP (D)
$\min \sum_{i\in[n]} c_i \cdot x_i$ subject to $\sum_{i\in[n]} a_{j,i} x_i \geq b_j, \quad \forall j\in[m]$ $x_i \geq 0, \quad \forall i\in[n]$	$\max \sum_{j\in[m]} b_j \cdot y_j$ subject to $\sum_{j\in[m]} y_j a_{j,i} \leq c_i, \quad \forall i\in[n]$ $y_j \geq 0, \quad \forall j\in[m]$

Strong Duality Theorem: Let x^* and y^* be optimal solutions of (P) and (D), respectively. Then

$$\sum_{i\in[n]} c_i \cdot x_i^* = \sum_{j\in[m]} b_j \cdot y_j^*$$

We won't show a proof of this theorem here, but mention that we've seen examples of this fact:

1. The Diet Problem.
2. Maximum-Flow Vs. Minimum Cut.

Talk Outline

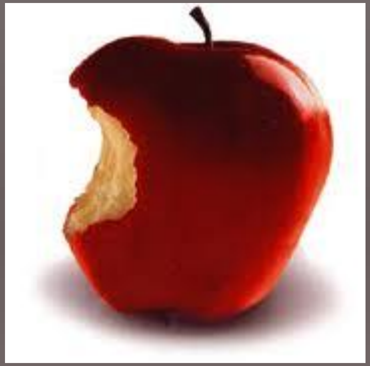
- Introduction to Linear Programming
  1. Problem Definition and Terminology
  2. Geometric Interpretation
  3. More Motivating Examples
  4. Duality

- Integer Linear Programming

- Rounding Fractional LP in Approximation Algorithms :
  1. Rounding to approximate Set Cover
  2. Randomized Rounding to approximate Set Cover

Back to our Diet Example

While half a serving of shawarma might mean something, what do we if the optimal diet requires we eat 0.5 apples a day? What do we do with the remaining 0.5 apple?



Integer Linear Programming

Definition:
LP with all coordinates in x restricted to be integers.

The Bad News:
Hard. Even determining whether a feasible solution exists is NP-hard.

The Good News:
We can use Integer LPs and their LP-Relaxation (the equivalent Fractional Linear Program) to devise efficient approximation algorithms.

Vertex Cover

Definition: Let $G = (V, E)$ be an undirected graph, with cost function $c: V \to R^+$.
We say that $U \subseteq V$, is a Vertex Cover if every edge in E has an endpoint in U.

Easy: Take $U = V$.

(NP-)Hard: Find a smallest cost Vertex Cover.

Integer LP for Vertex Cover

Define x_v for each vertex v. The LP is:

$$\min \sum_{v \in V} c(v) \cdot x_v$$

$$\text{subject to } x_u + x_v \geq 1, \quad \forall \{u, v\} \in E$$

$$x_v \in \{0,1\}, \quad \forall v \in V$$

Set Cover

Definition: Let $S_1, \dots, S_n \subseteq U = \cup S_i$ be sets of elements, with cost function $c: \{S_i\}_{i=1}^{n} \to R^+$.
We say that $S' \subseteq \{S_1, S_2, \dots, S_n\}$ is a Set Cover if every element of U is contained in a set in S'.

Easy: Take $S' = \{S_1, \dots, S_n\}$

(NP-)Hard: Find a smallest cost Set Cover.

Integer LP for Set Cover

Define x_i for each set S_i. The LP is:

$$\min \sum_{i \in [n]} c(S_i) \cdot x_i$$

$$\text{subject to } \sum_{j: i \in S_j} x_j \geq 1, \quad \forall i \in U$$

$$x_j \in \{0,1\}, \quad \forall j \in [n]$$

Integer LP
Checking for Feasible Solutions

Let us consider the (NP-hard) decision problem of Set Cover, SC.

$$\text{SC} = \{S_1, S_2, ..., S_n, c, k \mid S_1, S_2, ..., S_n \text{ has a Set Cover of cost} \le k\}$$

Modified Integer LP for Set Cover

Define x_i for each set i. The LP is:

Unnecessary →

$$\min \sum_{i \in [n]} c(S_i) \cdot x_i \qquad \sum_{i \in [n]} c(S_i) \cdot x_i \le k$$

$$\text{subject to} \sum_{j : i \in S_j} x_j \ge 1, \quad \forall i \in U$$

$$x_j \in \{0,1\}, \quad \forall j \in [n]$$

Integer LP vs. LP- Set Cover

Integer LP	Fractional LP
$\min \sum_{i\in[n]} c(S_i)\cdot x_i$	$\min \sum_{i\in[n]} c(S_i)\cdot x_i$
s.t. $\sum_{j:i\in S_j} x_j \geq 1, \quad \forall i\in U$	s.t. $\sum_{j:i\in S_j} x_j \geq 1, \quad \forall i\in U$
$x_j \in \{0,1\}, \ \forall j\in[n]$	$1\geq x_j \geq 0, \ \forall j\in[n]$

(Unnecessary)

Denote: $IOPT$ = Optimal value of Integer LP.

OPT = Optimal value of Fractional LP.

Question: IOPT vs. OPT?

Answer: $IOPT \geq OPT$, as the fractional LP's feasible points contain all the feasible points of the integer LP.

Talk Outline

- Introduction to Linear Programming
  1. Problem Definition and Terminology
  2. Geometric Interpretation
  3. More Motivating Examples
  4. Duality

- Integer Linear Programming

- Rounding Fractional LP in Approximation Algorithms :
  1. Rounding to approximate Set Cover
  2. Randomized Rounding to approximate Set Cover

Set Cover (Deterministic) Rounding

Define: $f(i) = |\{j \mid i \in S_j\}|$.
$f = \max\{f(i) \mid i \in U\}$.

We'll show a simple f-approximation to the minimum Set Cover:

Algorithm:

1. Find x^*, an optimal solution to the LP-relaxation.
2. Pick all sets S for which $x_S^* \geq 1/f$ in the solution.

Set Cover (Deterministic) Rounding

Define: $f(i) = |\{j \mid i \in S_j\}|$.
$f = \max\{f(i) \mid i \in U\}$.

Algorithm:

1. Find x^*, an optimal solution to the LP-relaxation.
2. Pick all sets S for which $x_S^* \geq 1/f$ in the solution.

Feasible Solution:

Since every element i appears in at most f sets, one of the sets S that contain it must have $x_S^* \geq 1/f$. Therefore the algorithm returns a feasible solution.

Deterministic Rounding (cont.)

Define: $f(i) = |\{j \mid i \in S_j\}|$.

$f = \max\{f(i) \mid i \in \cup\}$.

Algorithm:

1. Find x^*, an optimal solution to the LP-relaxation.
2. Pick all sets S for which $x_S^* \geq 1/f$ in the solution.

Approximation Guarantee

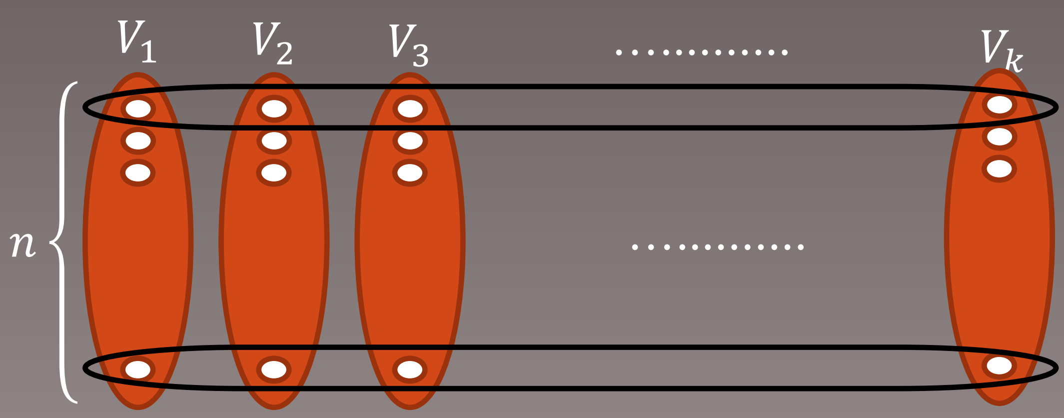
As $OPT \leq IOPT$, and we have at most multiplied every coordinate of x^* by f, we have obtained a solution of cost$\leq f \cdot OPT \leq f \cdot IOPT$.

Deterministic Rounding (cont.)

Tightness of Analysis:

We define the following hypergraph $G = (V, H)$.
$V = V_1 \cup V_2 \cup \cdots \cup V_k$, with the V_i's disjoint. $|V_i| = n$.

H is all n^k hyper-edges containing exactly one vertex from each V_i.
As in Vertex Cover, vertices are sets, elements are hyperedges and inclusion corresponds to incidence. Each element (hyperedge) appears in $f = k$ sets. Each set (vertex) has cost 1.

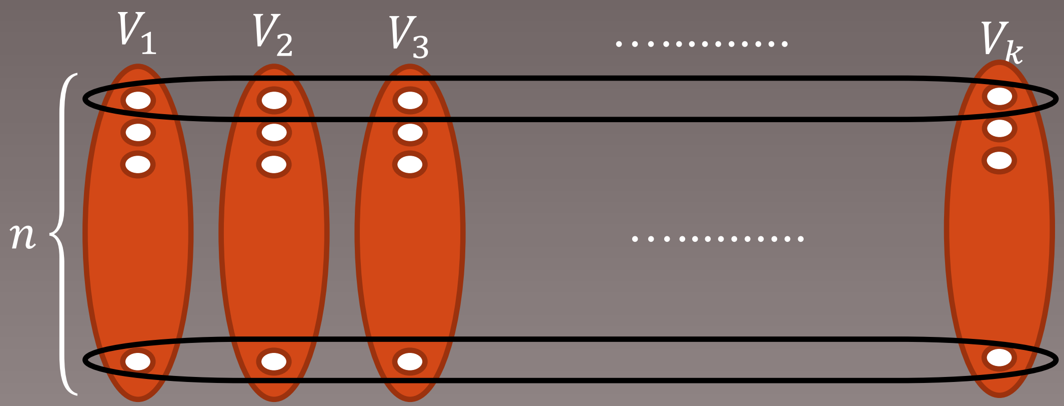


Deterministic Rounding (cont.)

Tightness of Analysis (cont.):

Notice that a fractional solution costs at least n. Therefore, a possible optimal solution can pick every set (vertex) to the extent of $1/k$, giving an optimal fractional solution of cost n.

Using this optimal fractional solution, the above algorithm returns a cover of cost kn (all the vertices), whereas a feasible integral solution of cost n exists; for example, all vertices in V_1.



Talk Outline

- Introduction to Linear Programming
    1. Problem Definition and Terminology
    2. Geometric Interpretation
    3. More Motivating Examples
    4. Duality

- Integer Linear Programming

- Rounding Fractional LP in Approximation Algorithms :
    1. Rounding to approximate Set Cover
    2. Randomized Rounding to approximate Set Cover

Randomized Approximation

Definition: We say a polynomial time algorithm, A, is a c –factor randomized approximation algorithm for a problem π if for every problem instance i, A returns a solution x such that

$$\Pr[x\ is\ a\ feasible\ solution\ with\ cost \leq\ c \cdot IOPT(i)] \geq ½$$

Probability – Reminders

Markov's Inequality

Let X be some random variable and $\mu = E[X]$. Then

$$\Pr[X \geq c \cdot \mu] \leq 1/c$$

One line proof:

$$E[X] = \Sigma_x \Pr[X = x] \cdot x \geq \Pr\big[X \geq c \cdot E[X]\big] \cdot c \cdot E[X]$$

Union Bound (Boole's Inequality)

Let $A_1, A_2, \ldots, A_k$ be events. Then

$$\Pr[A_1 \vee A_2 \vee \cdots \vee A_k] \leq \sum_i \Pr[A_i]$$

One line proof:

Induction &

$$\Pr[A_1 \vee A_2] = \Pr[A_1] + \Pr[A_2] - \Pr[A_1 \wedge A_2] \leq \Pr[A_1] + \Pr[A_2]$$

Randomized Rounding

Randomized Approximation: a Recipe:

1. Find an optimal solution to the LP-relaxation, x^*, and consider its entries as probabilities.

2. Use these probabilities to build an integer solution, x.

3. Show $\Pr[x \; infeasible] \leq {}^1/_4$.

4. Show $\Pr[cost \; of \; x > c \cdot IOPT] \leq {}^1/_4$.

5. By Union Bound, we find that
$\Pr[x \; infeasible \; or \; cost \; of \; x > c \cdot IOPT] \leq$ ½, and therefore
$\Pr[x \; is \; feasible, with \; cost \leq \; c \cdot IOPT] \geq$ ½.

Set Cover
(Randomized) Rounding

Algorithm:

1. Find x^*, an optimal solution to the LP-relaxation.
2. Initialize an empty covering, C.
3. Repeat $O(\log n)$ times:
   1. For every set S not yet in C, add it to C with probability x_S^*.

We will now wish to show:

1. $\Pr[C\ infeasible] \leq {}^1/_4$.
2. $\Pr[Cost\ C \geq \Omega(\log n) \cdot IOPT] \leq {}^1/_4$.

Putting both results together, we find that
$\Pr[C\ is\ a\ feasible\ cover\ of\ cost < O(\log n) \cdot IOPT]\ \geq ½$
and thus the above is an $O(\log n)$ –factor randomized approximation algorithm.

Randomized Rounding

Claim 1: $\Pr[\boldsymbol{C\ infeasible}] \leq {}^{\mathbf{1}}/_{\mathbf{4}}$:

Let e be some uncovered element, appearing in k sets. Denote these sets' probabilities by $p_1, p_2, \ldots, p_k$.

$$\Pr[e\ not\ picked\ in\ this\ iteration\ of\ 3] = \Pi_{i=1}^{k}(1 - p_i)$$

Since e is fractionally covered in x^*, we have $\Sigma p_i \geq 1$.

Lemma: Let $p_1 + p_2 + \cdots + p_k \geq 1$ and $1 \geq p_i \geq 0\ \forall i$, then

$$\Pi_{i=1}^{k}(1 - p_i) \leq \left(1 - \frac{1}{k}\right)^k.$$

Therefore, for any element e

$$\Pr[e\ not\ covered\ by\ C\ in\ one\ iteration] =$$

$$\Pi_{i=1}^{k}(1 - p_i) \leq \left(1 - \frac{1}{k}\right)^k \leq 1/e.$$

Proof of $(\star)$: Let $p_1 + p_2 + \cdots + p_k \geq 1$ and $1 \geq p_i \geq 0\ \forall i$,

then $f(p) = \Pi_{i=1}^{k}(1 - p_i) \leq \left(1 - \frac{1}{k}\right)^k$.

Consider $p = (p_1, p_2, \dots, p_k)$ which maximizes $f(p)$, subject to the above constraints.

1. As $1 \geq p_i\ \forall i$, every term in the product, $f(p)$, is positive.

2. If $\Sigma p_i > 1$, we can decrease some $p_i > 0$ and still have $\Sigma p_i \geq 1$, $1 \geq p_i \geq 0\ \forall i$, but increased $f(p)$. Therefore $\Sigma p_i = 1$.

3. Finally, if $p \neq (\frac{1}{k}, \frac{1}{k}, \dots, \frac{1}{k})$, there exist some $p_i > 1/k$ and $p_j < 1/k$. Replacing these by $(p_i + p_j)/2$ increases $f(p)$, a contradiction.

Therefore $\max(f(p)) = \left(1 - \frac{1}{k}\right)^k$.

QED.

Randomized Rounding

Claim 1: $\Pr[\boldsymbol{C\ infeasible}] \leq {}^{\mathbf{1}}/_{\mathbf{4}}$ (continued):
Picking up where we left off, we recall that for every element e

$$\Pr[e\ not\ covered\ by\ C\ in\ one\ iteration] \leq 1/e$$

If we repeat this $\ln(4n) = O(\log n)$ iterations, we find that for every element e:

$$\Pr[e\ not\ covered\ by\ C\ in\ all\ iterations] \leq \left(\frac{1}{e}\right)^{\ln(4n)} = \frac{1}{4n}$$

Therefore, by taking a Union Bound over all elements, we find that

$$\Pr[C\ infeasible] =$$
$$\Pr[one\ of\ the\ n\ elements\ in\ uncovered\ by\ C] \leq$$
$$n \cdot 1/4n = 1/4$$

Thus finishing our proof of claim 1.

Set Cover (Randomized) Rounding

Algorithm:

1. Find x^*, an optimal solution to the LP-relaxation.
2. Initialize an empty covering, C.
3. Repeat $\ln(4n) = O(\log n)$ times:
   1. For every set S not yet in C, add it to C with probability x_S^*.

We will now wish to show:

1. $\Pr[C\ infeasible] \leq 1/4$.
2. $\Pr[Cost\ C \geq 4\ln(4n) \cdot IOPT] \leq 1/4$.

Putting both results together, we find that
$\Pr[C\ is\ a\ feasible\ cover\ of\ cost < 4\ln(4n) \cdot IOPT] \geq ½$
and thus the above is an $O(\log n)$ –factor randomized approximation algorithm.

Randomized Rounding

Claim 2: $\Pr[\boldsymbol{cost(C) \geq 4\, ln(4n) \cdot IOPT}] \leq {}^{1}/_{4}$:

The expected cost of the addition to C in one iteration is at most $\text{cost}(x^*) = OPT$:

$$E[added\ cost] \leq \Sigma_S x_S^* \cdot c(s) = cost(x^*) = OPT$$

Repeating the process $\ln(4n)$ times gives us a C of expected cost at most

$$\ln(4n) \cdot cost(x) = \ln(4n) \cdot OPT \leq \ln(4n) \cdot IOPT.$$

Now, by Markov's inequality,

$$\Pr[cost(C) \geq 4\ln(4n) \cdot IOPT] \leq {}^{1}/_{4}$$

and claim 2 follows.

Set Cover
(Randomized) Rounding

Algorithm:

1. Find x^*, an optimal solution to the LP-relaxation.
2. Initialize an empty covering, C.
3. Repeat $\ln(4n)$ times:
   1. For every set S not yet in C, add it to C with probability x_S^*.

We will now wish to show:

1. $\Pr[C\ infeasible] \leq {}^1/_4$.
2. $\Pr[Cost\ C \geq 4\ln(4n) \cdot IOPT] \leq {}^1/_4$.

Putting both results together, we find that
$\Pr[C\ is\ a\ feasible\ cover\ of\ cost \leq 4\ln(4n) \cdot IOPT] \geq ½$
and thus the above is an $O(\log n)$ –factor randomized approximation algorithm.

Talk Outline

- Introduction to Linear Programming
  1. Problem Definition and Terminology
  2. Geometric Interpretation
  3. More Motivating Examples
  4. Duality

- Integer Linear Programming

- Rounding Fractional LP in Approximation Algorithms :
  1. Rounding to approximate Set Cover
  2. Randomized Rounding to approximate Set Cover

Questions?

Thank You.