Integer Programming (IP)

- An LP problem with an additional constraint that variables will only get an integral value, maybe from some range.
- BIP – binary integer programming: variables should be assigned only 0 or 1.
- Can model many problems.
- NP-hard to solve!

Example: Vertex Cover

Variables: for each $v \in V$, x_v - is v in the cover?

Minimize $\Sigma_v x_v$

Subject to: $x_i + x_j \geq 1 \quad \forall_{\{i,j\} \in E}$

$x_v \in \{0,1\}$

Weighted Vertex Cover

Input: Graph $G=(V,E)$ with non-negative weights $w(v)$ on the vertices.

Goal: Find a minimum-cost set of vertices S, such that all the edges are covered. An edge is covered iff at least one of its endpoints is in S.

Recall: Vertex Cover is NP-complete.

The best known approximation factor is $2- (\log \log |V|/2 \log|V|)$.

Weighted Vertex Cover

Variables: for each $v \in V$, $x(v)$ – is v in the cover?

Min $\Sigma_{v \in V} w(v)x(v)$

s.t.

$x(v) + x(u) \geq 1, \ \forall (u,v) \in E$

$x(v) \in \{0,1\} \quad \forall v \in V$

The LP Relaxation

This is **not** a linear program: the constraints of type $x(v) \in \{0,1\}$ are not linear. We got an LP with integrality constraints on variables – an **integer linear programs (IP)** that is NP-hard to solve.

However, if we replace the constraints $x(v) \in \{0,1\}$ by $x(v) \geq 0$ and $x(v) \leq 1$, we will get a linear program.

The resulting LP is called a **Linear Relaxation** of IP, since we relax the integrality constraints.

LP Relaxation of Weighted Vertex Cover

Min $\Sigma_{v \in V} w(v)x(v)$

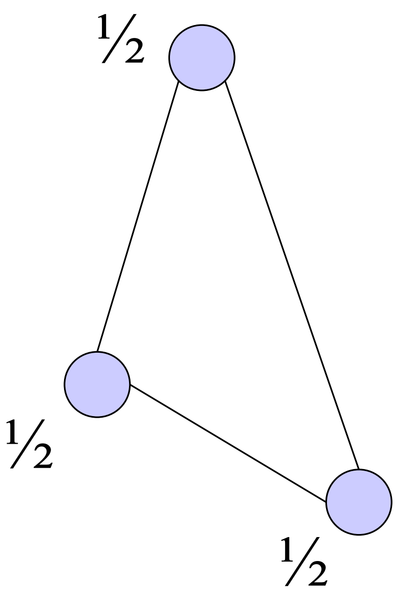
s.t.

$x(v) + x(u) \geq 1, \ \forall (u,v) \in E$

$x(v) \geq 0, \ \forall v \in V$

$x(v) \leq 1, \ \forall v \in V$

LP Relaxation of Weighted Vertex Cover - example



Consider the case of a 3-cycle in which all weights are 1.

An optimal VC has cost 2 (any two vertices)

An optimal relaxation has cost 3/2 (for all three vertices x(v)=1/2)

The LP and the IP are different problems. Can we still learn something about Integral VC?

Why LP Relaxation Is Useful ?

The optimal value of LP-solution provides a bound on the optimal value of the original optimization problem. OPT(LP) is always better than OPT(IP) (why?)

Therefore, if we find an integral solution within a factor r of OPT_{LP}, it is also an r-approximation of the original problem.

It can be done by 'wise' rounding.

2-approx. for weighted VC

1. Solve the LP-Relaxation.

2. Let S be the set of all the vertices v with $x(v) \geq 1/2$. Output S as the solution.

Analysis: The solution is feasible: for each edge $e=(u,v)$, either $x(v) \geq 1/2$ or $x(u) \geq 1/2$

The value of the solution is: $\Sigma_{v\in s} w(v) = \Sigma_{\{v|x(v) \geq 1/2\}} w(v) \leq \Sigma_{v\in V} w(v)2x(v) = 2OPT_{LP}$

Since $OPT_{LP} \leq OPT_{VC}$, the cost of the solution is $\leq 2OPT_{VC}$.

LP Duality

Consider LP: max $\mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} \le \mathbf{b}$, $\mathbf{x} \ge 0$
n variables, m constraints

How large can the optimum $\mathbf{c}^T\mathbf{x}$ be?

Consider a vector $\mathbf{y}$ of m variables.
If we demand that $\mathbf{y} \ge 0$ then $\mathbf{y}^T A\mathbf{x} \le \mathbf{y}^T\mathbf{b}$
If we demand that $\mathbf{c}^T \le \mathbf{y}^T A$ then $\mathbf{c}^T\mathbf{x} \le \mathbf{y}^T A\mathbf{x}$

So $\mathbf{c}^T\mathbf{x} \le \mathbf{y}^T A\mathbf{x} \le \mathbf{y}^T\mathbf{b}$
How small can $\mathbf{y}^T\mathbf{b}$ be?

minimize $\mathbf{b}^T\mathbf{y}$ s.t. $A^T\mathbf{y} \ge \mathbf{c}$, $\mathbf{y} \ge 0$ (called the dual LP)

Duality

Primal: maximize $\mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} \le \mathbf{b}, \mathbf{x} \ge 0$
Dual: minimize $\mathbf{b}^T\mathbf{y}$ s.t. $A^T\mathbf{y} \ge \mathbf{c}, \mathbf{y} \ge 0$

- In the primal, **c** is cost function and **b** was in the constraint. In the dual, their roles are swapped.
- Inequality sign is changed and maximization turned to minimization.

Dual:

minimize $2x + 3y$

s.t $x + 2y \ge 4,$
$2x + 5y \ge 1,$
$x - 3y \ge 2,$
$x, y \ge 0$

Primal:

maximize $4p + q + 2r$

s.t $p + 2q + r \le 2,$
$2p + 5q - 3r \le 3,$
$p, q, r \ge 0$

Duality – general form

Primal	max $c^T x$	min $b^T y$	Dual
	$\le b_i$	≥ 0	
Constraints	$\ge b_i$	≤ 0	Variables
	$= b_i$	unconstrained	
	≤ 0	$\le c_i$	
Variables	≥ 0	$\ge c_i$	Constraints
	unconstrained	$= c_i$	

max $\mathbf{c}^T\mathbf{x}$ s.t. $A\mathbf{x} \le \mathbf{b}$, $\mathbf{x} \ge 0$

If $\mathbf{y} \ge 0$ then $\mathbf{y}^T A\mathbf{x} \le \mathbf{y}^T\mathbf{b}$

The Duality Theorem

Let P,D be an LP and its dual.
If one has optimal solution so does the other, and their values are the same.

We only saw $\mathbf{c}^T\mathbf{x} \leq \mathbf{y}^T\mathbf{b}$ (weak duality)
The duality thm: $\mathbf{c}^T\mathbf{x} = \mathbf{y}^T\mathbf{b}$ (proof not here)

Simple Example

- Diet problem: minimize $2x + 3y$
  subject to $x+2y \geq 4$,
  $x \geq 0, y \geq 0$

Peanut Butter

Steak

- Dual problem: maximize $4p$
  subject to $p \leq 2$,
  $2p \leq 3$,
  $p \geq 0$
- Dual: the problem faced by a pharmacist who sells synthetic protein, trying to compete with peanut butter and steak

Simple Example

- The pharmacist wants to maximize the price p, subject to constraints:
  - synthetic protein must not cost more than protein available in foods.
  - price must be non-negative
  - revenue to druggist will be 4p
- Solution: $p = 3/2$ → objective value = $4p = 6$
- Not coincidence that it's equal the minimal cost in original problem.

What's going on?

- Notice: feasible sets completely different for primal and dual, but nonetheless an important relation between them.
- Duality theorem says that in the competition between the grocery and the pharmacy the result is always a tie.
- Optimal solution to primal tells consumer what to do.
- Optimal solution to dual fixes the natural prices at which economy should run.

Duality Theorem

Druggist's max revenue = Consumers min cost

Practical Use of Duality:

- Sometimes simplex algorithm (or other algorithms) will run faster on the dual than on the primal.
- Can be used to bound how far you are from optimal solution.
- Interplay between primal and dual can be used in designing algorithms
- Important implications for economists.

Max Flow LP and its dual

Consider the max st-flow LP (add an arc from t to s):

$$\max\ f_{ts}\quad s.t.$$

$$f_{uv} \le c_{uv}\quad \forall uv \in E$$

$$\sum_{uv\in E} f_{uv} - \sum_{vu\in E} f_{vu} \le 0\quad \forall v \in V$$

$$f_{uv} \ge 0$$

$$\min \sum_{uv\in E} c_{uv} d_{uv}\quad s.t.$$

$$d_{uv} - p_u + p_v \ge 0\quad \forall uv \in E$$

$$p_s - p_t \ge 1$$

$$d_{uv} \ge 0\,, p_u \ge 0$$

IP version of dual = min st-cut

$$\min \sum_{uv \in E} c_{uv} d_{uv} \quad s.t.$$

$$d_{uv} - p_u + p_v \geq 0 \quad \forall uv \in E$$

$$p_s - p_t \geq 1$$

$$d_{uv} \in \{0,1\}, p_u \in \{0,1\}$$

Consider optimal solution (d^*, p^*): $p_s^* = 1, p_t^* = 0$
p^* naturally defines a cut: $S = \{v\colon\ p_v^* = 1\}, T = \{v\colon\ p_v^* = 0\}$
For $u \in S, v \in T$: $d_{uv}^* = 1$ for other uv can have $d_{uv}^* = 0$
So objective function is capacity of an st-cut!
Minimum achieved at the minimum st-cut.

Back to LP Dual – still min-cut?

$$\min \sum_{uv \in E} c_{uv} d_{uv} \quad s.t.$$

$$d_{uv} - p_u + p_v \geq 0 \quad \forall uv \in E$$

$$p_s - p_t \geq 1$$

$$0 \leq d_{uv} \leq 1, 0 \leq p_u \leq 1$$

Dropping the upper bounds $d_{uv} \leq 1, p_u \leq 1$ cannot increase the objective value. We're back at the dual of the max-flow LP.

Can the objective function be improved when dropping
the integrality constraints? In general – yes.
This specific matrix has a special property called total unimodularity
Such LPs have integral optimal solutions.
So optimum of dual LP remains value of min st-cut
By duality theorem: max-flow = min-cut

Complementary Slackness

Primal: $\max \sum_{j=1}^{n} c_j \cdot x_j \quad s.t. \quad \forall i \ \sum_{j=1}^{n} A_{ij} \cdot x_j \le b_i, x \ge 0$

Dual: $\min \sum_{i=1}^{m} b_i \cdot y_i \quad s.t. \quad \forall j \ \sum_{i=1}^{m} A_{ij} \cdot y_i \ge c_j, y \ge 0$

so $\forall j \ c_j x_j \le (A^T y)_j x_j$ and $\forall i \ b_i y_i \ge (Ax)_i y_i$

for optimal solutions $c^T x^\star = b^T y^\star$ so

$\sum_{j=1}^{n} c_j \cdot x_j^\star = \sum_{j=1}^{n} \left(\sum_{i=1}^{m} A_{ij} \cdot y_i^\star \right) \cdot x_j^\star = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} A_{ij} \cdot x_j^\star \right) \cdot y_i^\star = \sum_{i=1}^{m} b_i \cdot y_i^\star$

so $\forall j \ c_j x_j^\star = (A^T y)_j x_j^\star$ and $\forall i \ b_i y_i^\star = (Ax)_i y_i^\star$

hence, $\forall j$ either $x_j^\star = 0$ or $\sum_{i=1}^{m} A_{ij} \cdot y_i^\star = c_j$ and

$\forall i$ either $y_i^\star = 0$ or $\sum_{j=1}^{n} A_{ij} \cdot x_j^\star = b_i$

either a variable is zero or the corresponding constraint in the dual is tight.

Weighted Vertex Cover (again)

Min $\Sigma_{v \in V} w_v \cdot x_v$

s.t.

$x_v + x_u \geq 1, \ \forall (u,v) \in E$

$x_v \geq 0, \ \forall v \in V$

$x_v \leq 1, \ \forall v \in V$

Max $\Sigma_{(u,v) \in E} 1 \cdot y_{uv}$

s.t.

$\Sigma_{u:(u,v) \in E} y_{uv} \leq w_v \quad \forall v \in V$

$y_e \geq 0, \ \forall e \in E$

Solve the relaxed dual problem. Let y* be the solution. Complementary slackness tells us that if a dual constraint is not tight then corresponding x_v is zero. So set x_v to 0 unless constraint is tight.

Define $x_v = \begin{cases} 1 \text{ if } \sum_{(u,v) \in E} y^*_{uv} = w_v \\ 0 \text{ otherwise} \end{cases}$

Weighted Vertex Cover (analysis)

$$x_v = \begin{cases} 1 \text{ if } \sum_{(u,v)\in E} y^*_{uv} = w_v \\ 0 \text{ otherwise} \end{cases}$$

Does the vector x define a vertex cover?

Suppose not. Then $x_s = x_t = 0$ for some edge (s,t).
Then $\sum_{(u,s)\in E} y^*_{us} < w_s$ and $\sum_{(u,t)\in E} y^*_{ut} < w_t$.
But y^*_{st} only appears in these two constraints, so we can increase y^*_{st} without violating any constraint, contradicting optimality of y^*.

Weighted Vertex Cover (analysis)

Y* is optimal solution to dual problem. Dual objective is $\Sigma_{(u,v)\in E} 1 \cdot y_{uv}$

$$x_v = \begin{cases} 1 \text{ if } \sum_{(u,v)\in E} y^*_{uv} = w_v \\ 0 \text{ otherwise} \end{cases}$$

$$\sum_{v\in V} w_v x_v \leq \sum_{v\in V} \sum_{(u,v)\in E} y^*_{uv} = 2 \sum_{(u,v)\in E} y^*_{uv} = 2OPT_{LP} \leq 2OPT$$

Every edge counted twice

Linear Programming -Summary

- Of great practical importance:
  - LPs model important practical problems
    - production, manufacturing, network design, flow control, resource allocation.
  - solving an LP is often an important component of solving or approximating the solution to an **integer linear programming problem.**
- Can be solved in poly-time, the simplex algorithm works very well in practice.
- Use packages, you really do not want to roll your own code here.

Randomized Algorithms

Textbook:

Randomized Algorithms, by Rajeev Motwani and Prabhakar Raghavan.

Randomized Algorithms

- **A Randomized Algorithm** uses a random number generator.
  - its behavior is determined not only by its input but also by the values chosen by RNG.
  - It is impossible to predict the output of the algorithm.
  - Two executions can produce different outputs.

Why Randomized Algorithms?

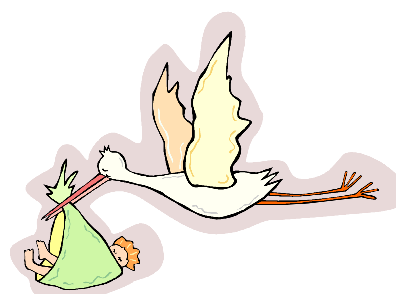
- Efficiency
- Simplicity
- Reduction of the impact of bad cases!
- Fighting an adversary.

Types of Randomized Algorithms

- **Las Vegas** algorithms
  - Answers are always correct, running time is random
  - In analysis: bound *expected* running time
- **Monte Carlo** algorithms
  - Running time is fixed, answers may be incorrect
  - In analysis: bound error probabilities

Randomized Algorithms

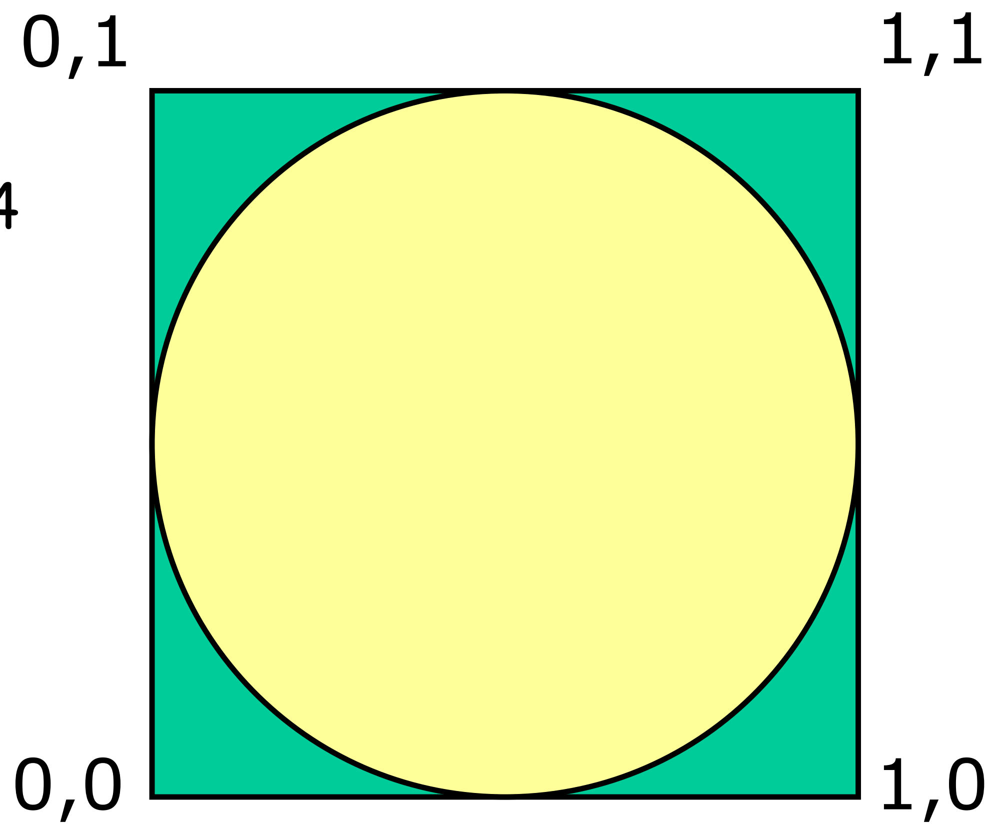
- Where do random numbers come from?
  - Sources of Entropy: physical phenomena, user's mouse movements, keystrokes, atmospheric noise, lava lamps.
- Pseudo-random generators: take a few "good" random bits and generate a lot of "fake" random bits.
  - Most often used in practice
  - Output of pseudorandom generator should be "indistinguishable" from true random



We will see
(up to random decisions):

1. A randomized approximation Algorithm for determining the value of Π.
2. A Randomized algorithm for the selection problem.
3. A randomized data structure.
4. Analysis of random walk on a graph.
5. A randomized graph algorithm.

Determining π



Square area = 1
Circle area = $\pi/4$

The probability that a random point in the square is in the circle = $\pi/4$

π = 4 * points in circle/points

Determining π

```
def findPi (points):
    incircle = 0

    for i in 1 to points:
        x = random() // float in [0,1]
        y = random()
        if (x - ½)² + (y - ½)² < 0.25)
                incircle = incircle + 1

    return 4.0 * incircle / points
```

Note : a point is in the circle if its distance from (½, ½) $<$ r

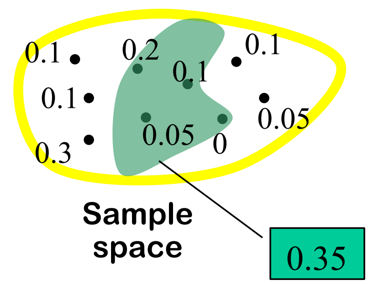
Determining π - Results

n	Output π
1	0.0
2	4.0
4	3.0
64	3.0625
1024	3.1640625
16384	3.1279296
131072	3.1376647
1048576	3.1411247

Real π = 3.14159265

If we wait long enough will it produce an arbitrarily accurate value?

Basic Probability Theory (a short recap)



- Sample Space Ω:
  - Set of possible outcome points
- Event $A \subseteq \Omega$:
  - A subset of outcomes
- $\Pr[A]$: probability of an event
  - For every event A: $\Pr[A] \in [0,1]$
  - If $A \cap B = \emptyset$ then $\Pr[A \cup B] = \Pr[A] + \Pr[B]$
  - $\Pr[\Omega] = 1$

Basic Probability Theory (a short recap)

- Random Variable X:
  - Function from event space to $\mathbb{R}$
- Example:
    - $\Omega=\{v=(g_1,g_2,\ldots g_n) \mid g_i\in[0,100]\}$
  - Events:
    - $A=\{v=(g_1,g_2,\ldots g_n) \mid \forall i: g_i\in[60,100]\}$
    - $B=\{v=(g_1,g_2,\ldots g_n) \mid \exists i,j,k : g_i,g_j,g_k \in[60,100]\}$
  - Random variables
    - X_i – 1 if student i passed, 0 if not
    - $X=X_1+\ldots+X_n$ – number of passing students
    - Y – Average grade

Possible grades for entire class

Everyone passed

At least three passed