Lecture 3

Basic Applications of LP*

Dantzig Presents LP George Dantzig developed Linear Programming during World War II and presented the ideas to a conference of eminent mathematicians and statisticians. Among the audience were Hotelling and von Neumann. In his book on the topic of LP, Dantzig recalls after finishing his speech asking for questions. Hotelling asks what the point of Dantzigs presentation has been pointing out the "world is not linear." Before Dantzig answers, von Neumann speaks up to say that if the axioms Dantzig has presented hold, then LP is an effective tool.

3.1 Max s-t Flow in a Directed Graph

Input: A di-graph:

$$G = (V, E)$$

Capacities:

$$\forall (u, v) \in E \quad c_{(u,v)} \geq 0$$

A source and a sink:

$$s, t \in V$$

Conservation of flow:

$$\text{flow into } (v \neq s, t) = \text{flow out of } v$$

A History of Flow The problem was originally studied by Tolstoy in the 1930's. Tolstoy was a mathematician in the Soviet Union studying how to optimally transport goods along the Soviet raliways from one city to another. In his formulation the vertices were cities and the edges were railroads connecting two cities. The capacity of each edge was the amount of goods the specific railroad could transport in a given day. The bottleneck was solely the capacities and not production or consumption on either end and there was no available storage at the intermediate cities.

*Lecturer: Ryan O'Donnell. Scribe: Will Devanny.

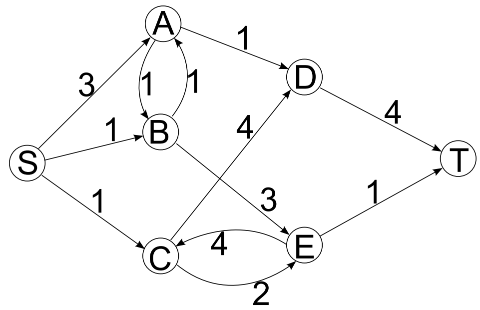

Figure 3.1: An input for the max s-t flow problem

The problem can be naturally set up as an LP by using a variable for the flow along each edge.

$$\begin{aligned} \max \quad & \sum_{v:(s,v)\in E} f_{(s,v)} - \sum_{v:(v,s)\in E} f_{(v,s)} \qquad (3.1) \\ \text{s.t.} \quad & \forall (u,v)\in E \quad f_{(u,v)} \geq 0 \\ & \forall (u,v)\in E \quad f_{(u,v)} \leq C_{(u,v)} \\ \forall v \neq s,t \quad & \sum_{u:(u,v)\in E} f_{(u,v)} = \sum_{w:(v,w)\in E} f_{(v,w)} \end{aligned}$$

We have to be careful with our objective function, Equation (3.1), to subtract any flow that might come back into the sink. In Figure 3.2, the results of a run of the LP on the example are shown.

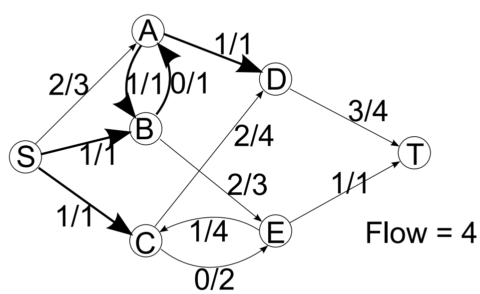

Figure 3.2: A solution to Figure 3.1

Remarks:

- Our solution has integral values.

  Theorem 3.1. *If the $C_{(u,v)}$ are integers then every b.f.s. optimal solution is integral.*

  This happens in general when the matrix A of the LP is unimodular.

- Is 4 the true optimal? Examine the cut created by $S = \{s, a\}$ and $T = V \setminus S$. The total capacity out of A is 4 and therefore $LPopt \leq 4$.

- Is that a coincidence? No.

  Theorem 3.2. *Max s-t flow = Min s-t cut in terms of the capacity graph.*

  This is an example of LP duality.

A History of Flow cont. Max flow was published in '54 again in the context of studying railroads by Ford and Fulkerson. They had heard about the problem from Ted Harris then working at Rand Corp. Harris orginally began studying flow in the USSR's railway system similar to Tolstoy years earlier. However Harris was looking at potential military applications of the min cut problem.

3.2 Max Perfect Matching in a Bipartite Graph

Dantzig studied max perfect matching during his time in the military. He had a group of people he wished to assign to an equal number of jobs. He knew a given person doing a given job would give the military some benefit. His goal was to give each person a job in such a way as to maximize the overall benefit. More formally we have a bipartite graph $G = (U \cup V, E)$ with some weight on the edges, $w_{u,v}\, \forall (u, v) \in E$. The weights are the value of person u doing job v.

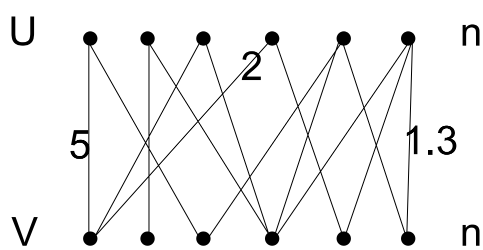

Figure 3.3: A bipartite graph on $2n$ vertices with associated weights

Our instinct in attacking this problem is to have a variable x_{uv} that is equal to 1 if we assign u job v and 0 if not:

$$\begin{aligned} \max \quad & \sum_{(u,v)\in E} w_{uv}x_{uv} \\ \text{s.t.} \quad & 0 \le x_{uv} \le 1 \\ & x_{uv} \in \mathbb{Z} && (3.2) \\ \forall v \in V \quad & \sum_{u:(u,v)\in E} x_{uv} = 1 && (3.3) \\ \forall u \in U \quad & \sum_{v:(u,v)\in E} x_{uv} = 1 && (3.4) \end{aligned}$$

Unfortunately Equation (3.2) isn't a linear constraint. We need to use the LP relaxation.

3.2.1 LP Relaxation

To form an LP relaxation of and IP, we drop the IP constraints. This enables us to solve the program efficiently. In the current problem we would remove constraint (3.2).

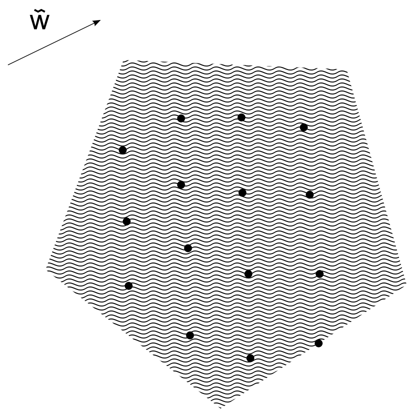

Figure 3.4: The feasible region of an LP with integer points inside

Remarks:

- The new LP is never unbounded, because we are inside the unit hypercube.
- If the LP is infeasible so is the original IP. Because if the feasible space of the LP is empty then it contains no integer points. The IP space inside of an LP can be seen in Figure 3.4.
- In general for relaxations, $Opt \le LPopt$. This holds even when the optimal value is infeasible ($-\infty \le c$).

For this problem, a lucky thing is true:

Theorem 3.3. *All extreme points are integral.*

Theorem 3.4 (Corrolary)**.** *If the LP is feasible so is the IP and IPopt = LPopt.*

Proof. By Contrapositive: If $\tilde{x}$ is feasible and non-integral then it is not an extreme point. $\tilde{x}$ not an extreme point means $\tilde{x} = \theta x^+ + (1-\theta)x^-$ for some $\theta \in [0, 1]$.

Suppose we have a feasible and non-integral solution $\tilde{x}$. Then there is a non-integral edge. If we look at one of its end vertices, that vertex must have another non-integral edge incident to it because of Equation (3.3) and Equation (3.4). Similarly we can travel along this other edge to the its opposite vertex and find another non-integral edge. Because the graph is finite and bipartite by repeating this process we will eventually end up with an even length cycle of non-integral edges, C, as seen in Figure 3.5.

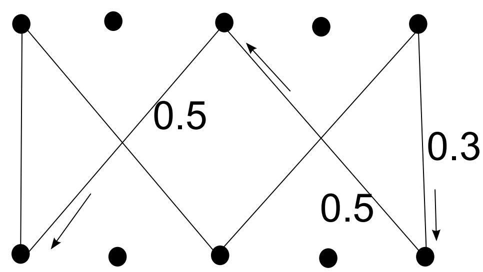

Figure 3.5: A non-integer cycle in a bipartite graph

Let $\epsilon = \min(\min_{(u,v)\in C} x_{uv}, \min_{(u,v)\in C} 1 - x_{uv})$. In words ϵ is the minimum distance from one of the weights on the cycle to an integer. Let x^+ be the same as $\tilde{x}$ but with ϵ added to the odd edges and $-\epsilon$ added to the even edges. Let x^- be the same as $\tilde{x}$ but with $-\epsilon$ added to the odd edges and $+\epsilon$ added to the even edges. We now have $\tilde{x} = \frac{1}{2}x^+ + \frac{1}{2}x^-$.

Iterate this process until we have all integer values. $\square$

Does this respect the value of *Opt*?

$$\frac{obj(x^+) + obj(x^-)}{2} = obj(\tilde{x})$$

So $obj(\tilde{x})$ is the average of $obj(x^+)$ and $obj(x^-)$. Because $obj(\tilde{x}) = Opt$ and neither $obj(x^+)$ nor $obj(x^-)$ is greater than Opt, $obj(x^+)$ and $obj(x^-)$ must both be equal to Opt.

3.3 Minimum Vertex Cover

Input:
Undirected graph:

$$G = (V, E)$$

Vertex costs:

$$\forall v \in V \quad c_v \geq 0$$

Output:

$$S \subseteq V \text{s.t.} \forall (u,v) \in E u \in S \text{ or } v \in S \text{ with } \min \sum_{v \in S} c_{uv}$$

Remarks:

- The problem is NP-Hard. So we do not expect to find an LP to solve the problem perfectly.
- The greedy algorithm tends not to work too well.

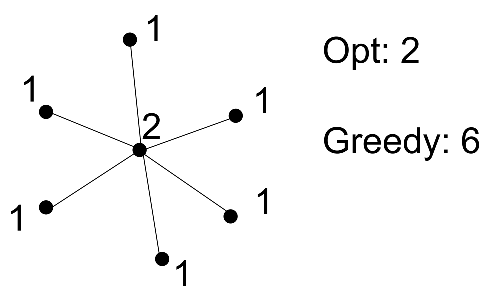

Figure 3.6: An example vertex cover problem

To phrase this as an IP, we will again use a variable x_v to be 1 if the vertex v is in the cover and 0 otherwise:

$$\begin{aligned} \min \quad & \sum_{v \in V} c_v x_v \\ \text{s.t.} \quad \forall v \in V \quad & 0 \leq x_v \leq 1 \\ \forall v \in V \quad & x_v \in \mathbb{Z} \qquad (3.5) \\ \forall (u,v) \in E \quad & x_u + x_v \geq 1 \end{aligned}$$

To relax this IP we throw out Equation (3.5). This LP will give us a fractional cover of the vertices.

Remarks:

- LPopt $\leq$ IPopt
- The LP is bounded, because we are again inside the unit cube.
- The LP is feasible. We can set all the variables to 1 or to do slightly better $\frac{1}{2}$.

3.3.1 LP Rounding

The idea is to use the optimal fractional solution to obtain a nicer integral solution.

Given a feasible $\tilde{x}$. We can define $S = S_{\tilde{x}} = v \in V : \tilde{x}_v \geq \frac{1}{2}$.

Fact: S is always a vertex cover. In the LP solution $x_u + x_v \geq 1$ implies at least one if the x's is greater than $\frac{1}{2}$.

Fact:

$$Cost(S) \leq 2LPCost(\tilde{x})$$

$$LPCost(\tilde{x}) = \sum_{v \in S} c_v \tilde{x}_v \geq \sum_{v \in S} c_v \frac{1}{2} = \frac{1}{2} Cost(S)$$

Corrolary: Let x^* be an optimal LP solution. Then $Cost(S_{x^*} \leq 2LPCost(x^{\&}) = 2LPopt \leq 2IPopt)$.

Remarks:

- This is called a factor 2 approximation algorithm.
- No better approximation is known.
- If $\mathsf{P} \neq \mathsf{NP}$ then we can't do better than 1.36.
- If the Unique Games Conjecture is true then we can't do better than $2 - \epsilon$.
- Every extreme point is half integral $(0, \frac{1}{2}, 1)$.

3.4 Simplex Algorithm Intro

The simplex algorithm is not in P, not good in theory, and no longer considered the best in practice. Interior point methods anecdotally do better on larger data sets. The simplex algorithm is considered good in smoothed analysis, a combination of average and worst case.

Theorem 3.5. *Solving LPs poly-time reduces to testing LP feasibility.*

Proof. Consider an LP:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \end{aligned}$$

Suppose we can test feasibility of the LP in poly-time.

$$\begin{array}{rl} \text{Add constraint } c^T x \geq 1000 & \quad \text{Feasible? No} \\ c^T x \geq 500 & \quad \text{Feasible? Yes} \\ c^T x \geq 750 & \quad \text{Feasible? No} \\ c^T x \geq 625 & \quad \text{Feasible? Yes} \\ \ldots & \quad \text{(binary search)} \end{array}$$

- How do we pick the starting number? Number 4 on the first homework gives a way to upper bound the size of a feasible solution.
- How do we know when to stop? We can similarly estimate the granularity of the solution.

$\square$