Optimization Methods		Date:	*23-02-2018*
Instructor: *Sujit Prakash Gujar*		Scribes:	*Avinash Vadlamudi* *Srijan Singh* *Vedant Sareen*

Lecture 12: Integer Programming - An Introduction to Integer Programming

1 Recap

1. In the previous lecture, we saw the pseudo-code of the ellipsoid method for mimimizing convex functions.
2. We also discussed the Travelling Salesman problem, and formulated it as a linear program.
   - We realised that the program would take exponential time to solve, due to the presence of $2^n - 2$ subtour constraints, where n is the number of vertices.
   - However, we observed that a relaxed version of the problem may be solved in polynomial time by using a mincut-separating oracle along with the ellipsoid method.
   - We further saw that it is possible to have an infinite number of constraints too, for example:

     $$a^T \times a \geq 0 \; \forall \; a \in \mathbb{R}^n$$

     Such problems cannot be solved with the naive oracle that is typically used, and it may be possible to create specialised oracles to tackle such problems.
3. We had a cursory look at Generative Adversarial Networks, which are used to learn the probability distribution from which the training examples are assumed to be sampled.
   - GANs are inspired by zero sum games, and model the same between a Generator and a Discriminator, with the reward of one network being equal to the loss of the other.
4. We discussed the concept of Nash Equilibrium.
   - We remarked that we can prove the existence of Nash Equilibrium for two player zero sum games by modelling it as a linear program.
   - We saw worst case optimal strategies, and found that if (x^*, y^*) form the N.E. of a two player zero sum game, then they are also worst case optimal.

2 Introduction to Integer Programming

Till now, we looked at linear programs, where all requirements, including the bounds on the range of values that the decision variables can take, are represented by linear relationships.

Now, we move on to integer programs, where the decision variables are only permitted to take integral values.

Definition 2.1. *Integer Programming*: An integer programming problem is a mathematical optimization or feasibility program in which some or all of the variables are restricted to be integers.

We will look at integer programming as an extension to linear programming, with the objective function and the constraints (except the integer constraints) being linear.

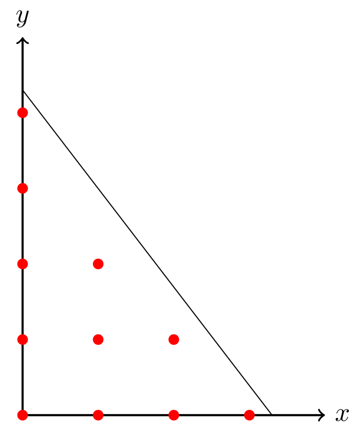

Figure 1: Solution space of a dummy IP problem

2.1 Representing an Integer Programs

Just like a linear programming problem, there are two main forms of representing an integer programming problem:

1. *Standard Form* :

$$\begin{aligned} \text{Maximize} : & \ c^T x \\ \text{Subject to} : & \ Ax \leq b \\ & \ x \geq 0 \\ & \ x \in \mathbb{Z}^n \end{aligned}$$

2. *Equational Form* :

$$\begin{aligned} \text{Maximize} : & \ c^T x \\ \text{Subject to} : & \ Ax = b \\ & \ x \geq 0 \\ & \ x \in \mathbb{Z}^n \end{aligned}$$

   Here, we assume that A contains only rational numbers. As we can rationalize A, we assume b also to be a vector of integers i.e. $b \in \mathbb{Z}^m$. Note that if we consider A to be real, we cannot put entries of b as integers.

2.2 Difficulty in solving Integer Programs

Integer programs are generally more complex than linear programs as the feasible region is not convex.

- In Figure 1 we have a bounded feasible region with the red points being the feasible integral solutions.
- As we can see, we need to check all the integral points inside the region which are close to the boundary for finding an integral optimal solution.
- Similarly, for a hyperplane in n dimensions, we may have an exponential number of integral points that must be checked.

2.3 Totally Unimodular Matrix

Definition 2.2. *Unimodular Matrix*: A unimodular matrix M is a square integer matrix having determinant 1 or -1.

Definition 2.3. *Totally Unimodular Matrix*: A Totally Unimodular matrix is a matrix for which every square non-singular submatrix is unimodular.

It can be seen that since each element of a matrix is also its square submatrix, the value of each element of a Totally Unimodular matrix is 0, 1 or -1.

Lemma 2.1. Let A be a totally unimodular matrix and $\bar{A}$ be a matrix obtained by appending a unit vector e as a new column to A, then $\bar{A}$ is also totally unimodular.

Proof.

$$\bar{A} = [\ A \mid e\ ]$$
$$\text{Where } e^T = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 & 0 & 0 \end{bmatrix}$$

Let Q be a square submatrix of $\bar{A}$.

1. If Q is a submatrix of A, then $\det(Q) \in \{1, -1, 0\}$.
2. If Q has last column from e and all the entries of last column of Q are 0, then $\det(Q) = 0$.
3. If last column of Q has one entry with 1, then expand the $\det(Q)$ using laplace expansion on last column to get a matrix of size $(n-1) \times (n-1)$ as the minor, which is also a submatrix of A. This implies $\det(Q) \in \{1, -1, 0\}$.

Hence, we conclude that if A is totally unimodular and e is unit vector, then $\bar{A}$ is also totally unimodular.

□

- By applying above-mentioned lemma recursively, we conclude that $\bar{A} = [\ A \mid I\ ]$ is also a totally unimodular matrix.
- We know that in standard form, the constraints of an LP are represented as

$$Ax \leq b$$

- The same LP can be written in equational form as

$$\bar{A}x = b$$

  Where $\bar{A} = [\ A \mid I\ ]$
- Therefore, in a standard form, if A is a totally unimodular matrix, then the corresponding matrix $\bar{A}$ in equational form is also totally unimodular.

3 Relating LP and IP

Lemma 3.1. Consider a linear program

$$\begin{aligned} \text{Maximize} : &\ c^T x \\ \text{Subject to} : &\ Ax \leq b \\ &\ x \geq 0 \\ &\ b \in \mathbb{Z}^m \end{aligned}$$

where A is totally unimodular. If the given LP has an optimal solution, then it also has an integral optimal solution.

Proof. 1. Convert LP to equational form by adding slack variables.

$$\bar{A} = [A \mid I_m]$$
$$\bar{A}x = b$$

2. The LP can now be solved using simplex method to get the optimal solution x^* with basis B.

$$x^* = (x_B^*, x_N^*)$$
$$\text{where } x_N^* = (0, \ldots, 0)$$
$$\text{and } x_B^* = \bar{A}_B^{-1} b$$

Some points to note here:

- Since an optimal solution exists, $\bar{A}_B$ must be non-singluar i.e. its inverse must exist.
- Since $\bar{A}_B$ is a square submatrix of $\bar{A}$, $det(\bar{A}_B) = 1 \; or -1$.
- Since
$$\bar{A}_B^{-1} = \frac{adj(\bar{A}_B)}{|\bar{A}_B|}$$
$\bar{A}_B^{-1}$ must have only integral values.
- Since $b \in \mathbb{Z}^m$, the optimal x values must also be integral.

Hence, proved.

□

4 Example Problems

4.1 Maximum Matching in Bipartite Graph

Definition 4.1. *Matching*: A matching or independent edge set in a graph is a set of edges without common vertices.

Definition 4.2. *Bipartite Graph*: A bipartite graph, also called a bigraph, is a set of graph vertices decomposed into two disjoint sets such that no two graph vertices within the same set are adjacent.

Therefore, the *maximum matching problem* refers to finding the largest possible matching.

Lemma 4.1. Let $G = (X \cup Y, E)$ be a bipartite graph. The incidence matrix A of G is totally unimodular.

Proof. We will prove the claim by induction on the size of square submatrices of A.

1. Let Q be a $l \times l$ submatrix of A.
2. *Base case:* Let $l = 1$. Since every entry of A is either 0 or 1, $det(Q) \in \{0, 1\} \subset \{-1, 0, 1\}$.
3. *Inductive Hypothesis:* Assume that $\forall Q'$ such that Q' is a $(l-1) \times (l-1)$ submatrix of A, $det(Q') \in \{1, 0, -1\}$.

   Consider the $l \times l$ matrix Q:

   - If a column in Q has all zeroes, $det(Q) = 0$.
   - If a column in Q has one 1, $det(Q)$ can be calculated by using laplace expansion over that column, leading to a $(l-1) \times (l-1)$ submatrix Q' as minor. Hence, $det(Q) = \pm det(Q') \in \{1, 0, -1\}$.

- If all columns in Q have two ones:
  - For every edge considered in Q, both the vertices are being considered.
  - Therefore, if we add all the rows corresponding to the vertices in X, we would obtain a row consisting of all ones $[1, 1, \ldots, 1, 1]$.
  - The same can be done for all the vertices in Y, thus obtaining another row consisting of all ones.
  - After applying the above-mentioned row operations, we are left with two identical rows in Q, both consisting of all ones. Hence, $det(Q) = 0$.

4. Since both the base case and the inductive step have been performed, by mathematical induction, the statement holds for all possible natural values of l.

Hence, A is totally unimodular.

□

IP Formulation of Maximum Matching Problem

- The IP formulation of the maximum matching problem can be written as follows:

$$
\begin{aligned}
Maximize : & \sum_{e \in E} x_e && (1)\\
Subject\ to : & \forall v,\ \sum_{e} x_e \leq 1,\ where\ e\ is\ incident\ on\ v && (2)\\
& Ax \leq 1 && (3)\\
& x_e \in \{0, 1\} && (4)
\end{aligned}
$$

- Here A is the incidence matrix.
- This formulation can be relaxed into an LP by replacing $x_e \in \{0, 1\}$ with $x_e \in [0, 1]$, or $0 \leq x_e \leq 1$.
- Clearly, the condition $x_e \leq 1$ is redundant due to condition (2), thus assuring A is still totally unimodular for a bipartite graph.

4.2 Minimum Vertex Cover in Bipartite Graph

Definition 4.3. *Vertex Cover*: A vertex cover of a graph is a set of vertices such that each edge of the graph is incident to at least one vertex of the set.

Therefore, the *minimum vertex cover problem* refers to finding the smallest possible vertex cover.

- The IP formulation of the minimum vertex cover problem can be written as follows:

$$
\begin{aligned}
Minimize : & \sum_{v \in X \cup Y} y_v && (5)\\
Subject\ to : & \forall e(u, v) \in E,\ y_u + y_v \geq 1 && (6)\\
& A^T y \geq 1 && (7)\\
& y_v \in \{0, 1\} && (8)
\end{aligned}
$$

- This formulation can be relaxed into an LP by replacing $y_v \in \{0, 1\}$ with $y_v \in [0, 1]$, or $0 \leq y_v \leq 1$.

- Clearly, the condition $y_v \leq 1$ is redundant due to condition (6), thus assuring A is still totally unimodular for a bipartite graph.

It must be noted that the LP relaxation of the minimum vertex cover problem is the dual of the LP relaxation of the maximum matching problem for bipartite graph.

- In the original integer programs of both the problems the constraints matrices were totally unimodular. Hence, the optimal objective value of integer program is the same as LP relaxation in both the cases.
- The above mentioned points imply that both the original IPs should have same optimal objective value due to strong duality on LP relaxation.
- Hence, number of edges in maximum matching is equal to size of minimum vertex cover.

Theorem 4.2. *König's theorem:* In any bipartite graph, the number of edges in a maximum matching equals the number of vertices in a minimum vertex cover.

5 Summary

1. We looked at a new class of problems called Integer Programming problems, and got an intuition of why these problems take an exponential time to be solved.
2. We discussed the concept of Totally Unimodular Matrices, and used it to relate LP and IP by proving that if the constraint matrix A is totally unimodular, and if the LP has an optimal solution, then it also has an integral optimal solution.
3. We saw two example problems - Maximum Matching and Minimum Vertex Cover in Bipartite Graphs, and found that their LP relaxed formulations are dual of each other.