In this lecture we will design approximation algorithms using linear programming. The key insight behind this approach is that the closely related integer programming problem is NP-hard (a proof is left to the reader). We can therefore reduce any NP-complete optimization problem to an integer program, "relax" it to a linear program by removing the integrality constraints, solve the linear program, and then "round" the LP solution to a solution to the original problem. We first describe the integer programming problem in more detail.

10.1 Integer Programming and LP relaxation

Definition 10.1.1 *An integer program is a linear program in which all variables must be integers.*

As in a linear program, the constraints in an integer program form a polytope. However, the feasible set is given by the set of all integer-valued points within the polytope, and not the entire polytope. Therefore, the feasible region is not a convex set. Moreover, the optimal solution may not be achieved at an extreme point of the polytope; it is found at an extreme point of the convex hull of all feasible integral points. (See Figure 10.1.1.)

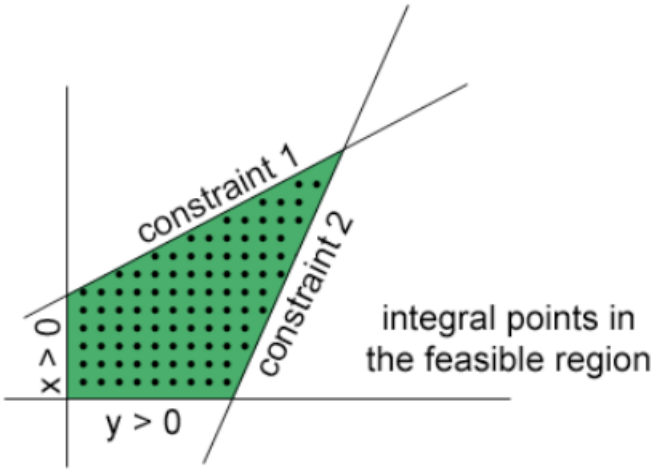

Figure 10.1.1: The feasible region of an integer linear program.

Usually, in designing LP-based approximation algorithms we follow the following approach:

1. Reduce the problem to an integer program.
2. Relax the integrality constraint, that is, allow variables to take on non-integral values.

3. Solve the resulting linear program to obtain a fractional optimal solution.

4. "Round" the fractional solution to obtain an integral feasible solution.

Note that the optimal solution to the LP is not necessarily integral. However, since the feasible region of the LP is larger than the feasible region of the IP, the optimal value of the former is no worse than the optimal value of the latter. This implies that the optimal value to the LP is a lower bound on OPT, the optimal value for the problem we started out with. While the rounded solution is not necessarily optimal for the original problem, since we start out with the optimal LP solution, we aim to show that the rounded solution is not too far from optimal.

These relationships between the different values are illustrated in Figure 10.1.2 below. The gap between the optimal LP value and the optimal integral solution is called the integrality gap of the linear program.

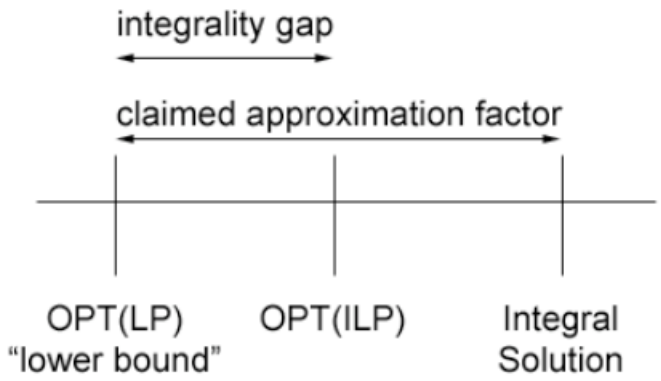

Figure 10.1.2: The relationship between the optimal LP and ILP values for minimization problems.

We now apply the linear programming approach to two problems: vertex cover and facility location.

10.2 Vertex Cover revisited

We have already seen a factor of 2 approximation using maximum matchings for the lower bound. Today we will see another factor of 2 approximation based on Linear Programming. This time we consider a weighted version of the problem. Recall that we are given a graph $G = (V, E)$ with weights $w : V \to \Re^+$, and our goal is to find the minimum weight subset of vertices such that every edge is incident on some vertex in that subset.

We first reduce this problem to an integer program. We have one $\{0, 1\}$ variable for each vertex denoting whether or not this vertex is picked in the vertex cover. Call this variable x_v for vertex v. Then we get the following integer program. The first constraint essentially states that for each

edge we must pick at least one of its endpoints.

$$\begin{aligned} \text{Minimize } \sum_{v \in V} w_v x_v \text{ subject to} & \\ x_u + x_v \geq 1 & \qquad \forall (u,v) \in E \\ x_v \in \{0,1\} & \qquad \forall v \in V \end{aligned}$$

To obtain a linear program, we relax the last constraint to the following:

$$x_v \in [0,1] \qquad \forall v \in V$$

Next we use a standard LP solver to find the optimal solution to this LP. Let x_v^* denote this optimal fractional solution. By our previous argument, the following holds:

Proposition 10.2.1 *Val(x^*)$\leq$OPT, where OPT is the value of the optimal solution to the vertex cover instance.*

The example below illustrates that the optimal solution to the LP is not necessarily integral.

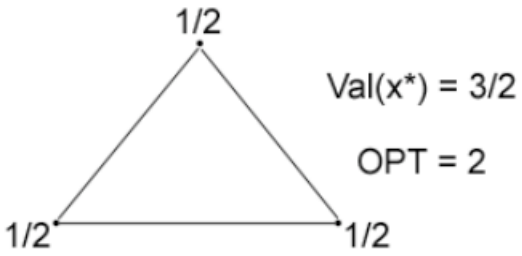

Figure 10.2.3: An example where the vertex cover LP has an integrality gap of 4/3. The optimal fractional solution sets $x_v = 1/2$ for all vertices, with a total cost of 3/2, while the optimal integral solution has cost 2.

It remains to round the fractional solution. For vertex cover, the obvious rounding works: for each $x_v^* \geq 1/2$, set $x_v = 1$ and include v in the vertex cover. For each $x_v^* < 1/2$, set $x_v = 0$ and don't include v in the vertex cover.

It is easy to see that this is a feasible solution and forms a vertex cover. Consider any edge $(u,v) \in E$. Then, by construction, $x_u^* + x_v^* \geq 1$. Therefore, at least one of x_u^* and x_v^* is at least 1/2, and is picked in the vertex cover. It remains to prove that the solution we obtain has small weight.

Lemma 10.2.2 $\sum_{v \in V} x_v w_v \leq 2 \sum_{v \in V} x_v^* w_v$

Proof: Recall that we set x_v to be 1 if and only if $x_v^* \geq 1/2$, and 0 otherwise. The lemma then follows by noting that $x_v \leq 2x_v^*$ for all v. ■

Finally, the weight of our vertex cover is exactly $\sum_{v \in V} x_v w_v$ because by definition $x_v = 1$ if and only if v is included in our vertex cover, and 0 otherwise. We therefore have the following theorem.

Theorem 10.2.3 *The above algorithm is a 2-approximation to weighted vertex cover.*

10.3 Facility location

The story behind the facility location problem is that a company wants to locate a number of warehouses such that the cost of opening these warehouses plus the cost of shipping its product to its retail stores is minimized. Formally, the facility location problem gives a collection of facilities and a collection of customers, and asks which facilities we should open to minimize the total cost. We accept a facility cost of f_i if we decide to open facility i, and we accept a routing cost of $c(i, j)$ if we decide to route customer j to facility i. Furthermore, the routing costs form a metric, that is, they are distance functions satisfying the triangle inequality.

First, we reduce this problem to an integer program. We let the variable x_i denote whether facility i is open, and let y_{ij} denote whether customer j is assigned to facility i. The following program then expresses the problem. The first constraint says that each customer should be assigned to at least one facility. The second says that is a customer is assigned to a facility, then that facility must be open.

$$\begin{aligned}
\text{minimize } \sum_i f_i x_i + \sum_{i,j} c(i,j) y_{ij} \text{ subject to} & \\
\sum_i y_{ij} \geq 1 & \qquad \forall j \\
x_i \geq y_{ij} & \qquad \forall i, j \\
x_i, y_{ij} \in \{0, 1\} & \qquad \forall i, j
\end{aligned}$$

To obtain a linear program, we relax the last constraint to $x_i, y_{ij} \in [0, 1]$.

For convenience, let $C_f(x)$ denote the total factory cost induced by x, i.e., $\sum_i f_i x_i$. Similarly, let $C_r(y)$ denote the total routing cost induced by y, $\sum_{i,j} c(i, j) y_{ij}$.

Let x^*, y^* be the optimal solution to this linear program. Since every feasible solution to the original ILP lies in the feasible region of this LP, the cost $C(x^*, y^*)$ is less than or equal to the optimal solution to the ILP. Since x^* and y^* are almost certainly non-integral, we need a way to round this solution to a feasible, integral solution without increasing the cost function much.

Note that the y_{ij} variables for a single j form a probability distribution over the facilities. The LP pays the expected routing cost over these facilities. If we could pick the closest facility over all those that j is connected to, our routing cost would be no more than in the LP solution. However, the closest facility is not necessarily the cheapest facility, and this is what makes this rounding process complicated.

To get around this problem, we first use a filtering technique that ensures that all facilities that j is connected to have small routing cost, and we pick the cheapest of these in our solution.

1. For each customer j, compute the average cost $\tilde{c}_j = \sum_i c(i, j) y^*_{ij}$.

2. For each customer j, let the S_j denote the set $\{i : c(i,j) \leq 2\tilde{c}_j\}$.

3. For all i and j: if $i \notin S_j$, then set $\tilde{y}_{ij} = 0$; else, set $\tilde{y}_{ij} = y^*_{ij} / \sum_{i \in S_j} y^*_{ij}$.

4. For each facility i, let $\tilde{x}_i = \min(2x^*_i, 1)$.

Lemma 10.3.1 *For all i and j, $\tilde{y}_{ij} \leq 2y^*_{ij}$.*

Proof: If we fix j and treat y^*_{ij} as a probability distribution, then we can show this by Markov's inequality. However, the proof of Markov's Inequality is simple enough to show precisely how it applies here:

$$\tilde{c}_j = \sum_i c(i,j) y^*_{ij} \geq \sum_{i \notin S_j} c(i,j) y^*_{ij} > \sum_{i \notin S_j} 2\tilde{c}_j y^*_{ij} \geq 2\tilde{c}_j \sum_{i \notin S_j} y^*_{ij}.$$

So, $1/2 \geq \sum_{i \notin S_j} y^*_{ij}$. For any fixed j, y^*_{ij} is a probability distribution, so $\sum_{i \in S_j} y^*_{ij} \geq 1/2$. Therefore, $\tilde{y}_{ij} = y^*_{ij} / \left(\sum_{i \in S_j} y^*_{ij} \right) \leq 2y^*_{ij}$. ■

Lemma 10.3.2 *$\tilde{x}, \tilde{y}$ is feasible, and $C(\tilde{x}, \tilde{y}) \leq 2C(x^*, y^*)$.*

Proof: For any fixed j, the elements $\tilde{y}_{ij}$ form a probability distribution. For every i and j, $\tilde{y}_{ij} \leq 2y^*_{ij}$ and thus $\tilde{x}_i \geq \sum_i \tilde{y}_{ij}$. It is clear that $0 \leq x_i, y_{ij} \leq 1$ for all i and j, so $\tilde{x}$ and $\tilde{y}$ are feasible solutions to the LP. ■

Now, given $\tilde{x}$ and $\tilde{y}$, we perform the following algorithm:

1. Pick the unassigned j that minimizes $\tilde{c}_j$.

2. Open factory i, where $i = \text{argmin}_{i \in S_j}(f_i)$.

3. Assign customer j to factory i.

4. For all j' such that $S_j \cap S_{j'} \neq \emptyset$, assign customer j' to factory i.

5. Repeat steps 1-4 until all customers have been assigned to a factory.

Let L be the set of facilities that we open in this way. We now show that the solution that this algorithm picks has reasonably limited cost.

Lemma 10.3.3 *$C_f(L) \leq 2C_f(x^*)$ and $C_r(L) \leq 6C_r(y^*)$.*

Proof: For any two customers j_1 and j_2 that were picked in Step 1, $S_{j_1} \cap S_{j_2} = \emptyset$.

Consider the facility cost incurred by one execution of Steps 1 through 4. Let j be the customer chosen in Step 1, and let i be the facility chosen in Step 2. Since $\tilde{x}$ is part of a feasible solution, $1 \leq \sum_{k \in S_j} \tilde{x}_k$. So, $f_i \leq f_i \sum_{k \in S_j} \tilde{x}_k$; and since f_i is chosen to be minimal, $f_i \leq \sum_{k \in S_j} f_k \tilde{x}_k$. Facility i is the only member of S_j that the algorithm can open.

Let J be the set of all customers selected in Step 1. Considering the above across the algorithm's whole execution yields

$$C_f(L) \leq \sum_{j \in J} \sum_{k \in S_j} f_k \tilde{x}_k = \sum_i f_i \tilde{x}_i = C_f(\tilde{x}) \leq 2C_f(x^*).$$

Consider now the routing cost C_r, and let $C_r(j)$ for a customer j denote the cost of routing j in L. If j was picked in Step 1, then its routing cost is $c(i,j)$ for some facility $i \in S_j$; so $C_r(j) \leq 2\tilde{c}_j$.

Now, suppose instead that j' was not picked in Step 1. By the algorithm, there is some j that picked in Step 1 such that $S_j \cap S_{j'} \neq \emptyset$. Suppose that facility i' is in this intersection, and say that facility i is the facility to which customers j and j' are routed. Now, at long last, we use the fact that $c(i,j)$ forms a metric: we know that $C_r(j') \leq c(i',j') + c(i',j) + c(i,j)$. Because i is in both S_j and $S_{j'}$, we know by their definition that $c(i',j') \leq 2\tilde{c}_{j'}$ and that $c(i',j) + c(i,j) \leq 4\tilde{c}_j$. The customer j' was not picked in Step 1, and customer j was, so $\tilde{c}_j \leq \tilde{c}_{j'}$, and thus, $C_r(j') \leq 6\tilde{c}_{j'}$.

Now, $\tilde{c}_j$ was the routing cost of customer j in the y^* LP solution. So, $C_r(L) \leq 6C_r(y^*)$. ■

This lemma yields the following as a corollary:

Theorem 10.3.4 *This algorithm is a 6-approximation to Facility Location.*

Notice that, in the preceding construction, we picked S_j to be all i such that the cost $c(i,j) \leq 2\tilde{c}_j$. The constant 2 is actually not optimal for this algorithm. Suppose we replace it with α, some parameter of the construction. If we redo the above arithmetic, we find that $C_f(L) \leq (1/(1-\alpha))C_f(x^*)$ and that $C_r(L) \leq (3/\alpha)C_r(y^*)$. Thus, if we let $\alpha = 3/4$ instead of $1/2$, this method yields a 4-approximation. If we let α be a variable in the actual computed values of $C_f(x^*)$ and $C_r(y^*)$, we would get a somewhat better approximation.

Note that the integrality gap of the facility location LP is actually quite a bit smaller than 4. There are several better rounding algorithms known based on the same lower bound that lead to improved approximations.