15.1 Introduction

In the last lecture we saw how to formulate optimization problems as Integer Linear Programs. We also noted (using reduction from 3-SAT), that unless $P = NP$, ILPs can not be solved in polynomial time. Therefore, we relax the integrality condition on the variables of the program to obtain a fractional solution of the *LP relaxation* in polynomial time and then try to obtain an approximate integral solution from this.
We obtained a 2-approximation algorithm for the Weighted Vertex Cover problem by using a very simple procedure to round the fractional solution of the corresponding linear program to get a feasible integral solution. There are mainly 3 techniques for rounding LP solutions:

1. Direct rounding
2. Randomized rounding
3. Rounding based on the Basic solution structure

In this lecture we will obtain approximation algorithms for the following 2 problems by direct rounding of their LP relaxation solution:

- The Metric Uncapacitated Facility Location Problem
- The $1|r_j|\sum C_j$ non-preemptive scheduling problem

15.2 Metric Uncapacitated Facility Location

The Metric Uncapacitated Facility Location problem (UFL) is as follows: we are given a set $V = \{1, \ldots, n\}$ of terminals with distances between them, c_{ij}. We will assume that c_{ij} satisfy symmetry and triangle inequality. For each terminal i in V, there is an associated facility opening cost, f_i. Our goal is to choose a subset $F \subseteq V$ of terminals to open as facilities so as to minimize the following cost function:

$$\text{cost}(F) = \sum_{i \in F} f_i + \sum_{j \in V} c(j, F) \tag{15.2.1}$$

where, $c(j, F) = \min_{i \in F} c_{ij}$, is the connection cost for terminal j. Let $i(j) = \text{argmin}_{i \in F} c_{ij}$, denote the facility that terminal j connects to. Since we are not placing any capacity constraints on the terminals, this version is called the uncapacitated facility location problem. To write the integer linear program formulation of this problem, we first introduce the following variables:

$$y_i = \begin{cases} 1 & \text{if } i \in F \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if terminal } j \text{ is assigned to facility } i \\ 0 & \text{otherwise} \end{cases}$$

The UFL problem can now be written as the following integer program

$$\text{minimize } Z(x,y) = \sum_{i \in V} f_i y_i + \sum_{i,j \in V} c_{ij} x_{ij}$$

subject to

$$\begin{aligned} \sum_{i \in V} x_{ij} &= 1 && \forall j \in V \\ x_{ij} &\leq y_i && \forall i,j \in V \\ x_{ij} &\in \{0,1\} && \forall i,j \in V \\ y_i &\in \{0,1\} && \forall i \in V \end{aligned}$$

The first constraint ensures that each terminal is assigned to atleast one facility and the second constraint ensures that if terminal j is assigned to facility i, then there must be a facility open at i. Remaining constraints are the integrality constraints for x_{ij} and y_i. It is easy to see that we can drop the integrality constraints on either the x_{ij}'s or on the y_i's and still get an integral solution. Let Z^*_{ILP} be the optimal value of the above integer program, UFL-IP. We will now relax the integrality conditions to $x_{ij}, y_i \geq 0$. Let Z^*_{LP} be the optimal value of this linear program relaxation, UFL-LP. Then,

$$Z^*_{LP} \;\leq\; Z^*_{ILP} \;=\; OPT$$

We have the following LP-rounding theorem due to Shmoys et al. [1]:

Theorem 15.2.1 *Given any feasible fractional solution (x,y) to the UFL-LP, there exists an integer feasible solution $(\hat{x},\hat{y})$ such that $Z(\hat{x},\hat{y}) \leq 4Z(x,y)$.*

This immediately gives a 4-approximation for the metric UFL problem. In Section 15.2.1, we will first show how to convert a fractional solution (x,y) to an integer solution $(\hat{x},\hat{y})$ while increasing the objective function value by a factor of atmost 6. In Section 15.2.2, we will use the ideas of Section 15.2.1 to obtain the 4-approximation.

15.2.1 LP-rounding Algorithm

The algorithm to obtain an integer solution from the fractional solution proceeds in 2 stages: *filtering* and *rounding*. Since in the fractional solution, (x,y), we may have nonzero x_{ij} where c_{ij} are 'large', we filter out such large distances in the first stage. We begin by defining the fractional connnection cost for terminal j, Δ_j as:

$$\Delta_j = \sum_{i \in V} c_{ij} x_{ij}$$

The following Lemma is proved by applying the filtering technique due to Lin & Vitter [2]:

Lemma 15.2.2 *Given a fractional solution (x, y) to UFL-LP, we can find another feasible fractional solution (x', y') such that*

1. $Z(x', y') \leq 2Z(x, y)$
2. *If* $x'_{ij} > 0$ *then* $c_{ij} \leq 2\Delta_j$.

Proof: For each terminal j, define the ball B_j to be the set of facilities whose distance from j is less than or equal to $2\Delta_j$. That is,

$$B_j = \{i : c_{ij} \leq 2\Delta_j\}$$

We will now filter out the long edges by setting $x'_{ij} = 0$ if $i \notin B_j$. To satisfy the constraints $\sum_{i \in V} x'_{ij} = 1$, we scale up x_{ij} inside B_j to get

$$x'_{ij} = \begin{cases} \frac{x_{ij}}{\sum_{i \in B_j} x_{ij}} & \text{if } i \in B_j \\ 0 & \text{if } i \notin B_j \end{cases}$$

Since $\sum_i x_{ij} = 1$, we can imagine the x_{ij}'s , for a fixed j, as defining a probability mass function over the set of facilities. By definition, $\Delta_j = \sum_i c_{ij} x_{ij}$, is analogous to the expected distance of a facility from terminal j under this distribution. By Markov's inequality,

$$\begin{aligned} \sum_{i \notin B_j} x_{ij} &= Pr\{c_{ij} > 2\Delta_j\} \\ &< \frac{1}{2} \end{aligned}$$

Therefore we have to scale the x_{ij} inside B_j by atmost 2 in the worst case. To satisfy the second set of constraints in UFL-LP, we have to scale up the y_i's. We set $y'_i = \min(1, 2y_i)$. It is straightforward to see that (x', y') defined as above is a feasible solution to UFL-LP. The second condition of Lemma 15.2.2 is satisfied by construction. All that remains to be proven is that $Z(x', y') \leq 2Z(x, y)$. Let $F(x, y) = \sum_{i \in V} f_i y_i$ be the facility opening cost and $C(x, y) = \sum_{i,j \in V} c_{ij} x_{ij}$ be the total connection cost for the feasible solution (x, y) so that $Z(x, y) = F(x, y) + C(x, y)$. Then,

$$\begin{aligned} \Delta'_j &= \sum_{i \in V} c_{ij} x'_{ij} \\ &\leq \sum_{i \in B_j} 2\Delta_j x'_{ij} \\ &= 2\Delta_j \end{aligned}$$

Therefore, $C(x', y') = \sum_{j \in V} \Delta'_j \leq 2 \sum_{j \in V} \Delta_j = 2C(x, y)$. Also, $F(x', y') = \sum_{i \in V} f_i y'_i \leq \sum_{i \in V} 2f_i y_i = 2F(x, y)$. Combining the two, $Z(x', y') \leq 2Z(x, y)$. ■

The solution (x', y') obtained after the filtering process ensures that all the candidate facilities for any terminal j are 'close' to it. If we could pick atleast one facility from each B_j without substantially increasing the facility location cost then we would be done. This, however, may not be possible. We will instead use the triangle inequality to round the y_i''s and obtain a 6-approximate solution.

The rounding stage proceeds as follows: first pick the terminal j with the smallest connection cost, Δ_j, under the fractional solution (x, y). For this terminal, we open the facility $i(j) \in B_j$ with the smallest facility opening cost. Note that since $\sum_{i \in B_j} y_i' \geq \sum_{i \in B_j} x_{ij}' = 1$,

$$\begin{aligned}\sum_{i \in B_j} f_i y_i' &\geq \sum_{i \in B_j} f_{i(j)} y_i' \\ &\geq f_{i(j)} \sum_{i \in B_j} y_i' \\ &\geq f_{i(j)}\end{aligned}$$

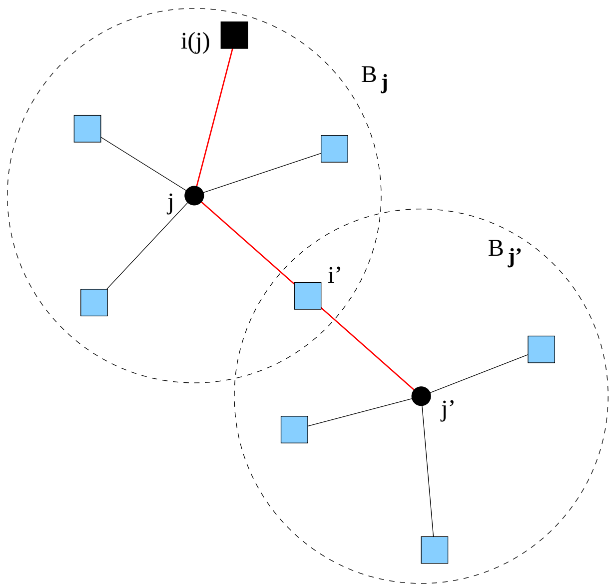

Figure 15.2.1: Clustering j and j' with $B_j \cap B_{j'} \neq \emptyset$

We can therefore open the facility $i(j)$ without increasing the facility opening cost by not opening any other facility in B_j. But there may be a terminal j' such that $i(j) \notin B_{j'}$ and $B_j \cap B_{j'} \neq \emptyset$. We assign such terminals to $i(j)$ (Figure 15.2.1). Let $i' \in B_j \cap B_{j'}$. Since $\Delta_j \leq \Delta_{j'}$, using triangle inequality,

$$\begin{aligned}c_{j'i(j)} &\leq c_{j'i'} + c_{i'j} + c_{ji(j)} \\ &\leq 2\Delta_{j'} + 2\Delta_j + 2\Delta_j \\ &\leq 6\Delta_{j'}\end{aligned}$$

We repeat the above rounding and clustering procedure till all the terminals have been assigned to some facility. Let $(\hat{x}, \hat{y})$ be the integer solution so obtained. The facility location cost does not increase by rounding (x', y') to $(\hat{x}, \hat{y})$. Therefore $F(\hat{x}, \hat{y}) \leq F(x', y') \leq 2F(x, y)$. For the total connection cost,

$$
\begin{aligned}
C(\hat{x}, \hat{y}) &= \sum_{j \in V} c_{i(j)j} \\
&\leq \sum_{j \in V} 6\Delta_j \\
&\leq 6C(x, y)
\end{aligned}
$$

Combining, $Z(\hat{x}, \hat{y}) = F(\hat{x}, \hat{y}) + C(\hat{x}, \hat{y}) \leq 2F(x, y) + 6C(x, y) < 6Z(x, y)$.

15.2.2 Improving to 4-approximation

In Section 15.2.1, during the filtering stage we chose the radius of the ball B_j around terminal j to be $2\Delta_j$. If we reduce this radius, the mass x_{ij} enclosed in B_j decreases. To satisfy the constraints, we will have to scale the x_{ij}'s (and hence the y_i's) by a larger factor, thereby increasing the facility opening cost. The total connection cost goes down because the candidate facilities are closer to the terminals. In this section we will derive the approximation ratio when the radius of ball B_j is $(1 + \alpha)\Delta_j$ and optimize the parameter α to obtain an improved approximation guarantee.
Again by using Markov's inequality, the mass enclosed in the ball B_j of radius $(1 + \alpha)\Delta_j$ is

$$
\begin{aligned}
\sum_{i \in B_j} x_{ij} &= Pr\{c_{ij} \leq (1 + \alpha)\Delta_j\} \\
&\geq \frac{\alpha}{1 + \alpha}
\end{aligned}
$$

The scaling factor is therefore $\frac{1+\alpha}{\alpha}$ in the worst case. For the filtered solution (x', y'), we have $F(x', y') \leq \frac{1+\alpha}{\alpha} F(x, y)$ and $\Delta'_j \leq (1 + \alpha)\Delta_j$. By following the same rounding and clustering method as in Section 15.2.1,

$$
\begin{aligned}
F(\hat{x}, \hat{y}) &\leq F(x', y') \\
&\leq \frac{1 + \alpha}{\alpha} F(x, y)
\end{aligned}
$$

and

$$
\begin{aligned}
C(\hat{x}, \hat{y}) &\leq \sum_{j \in V} 3\Delta'_j \\
&\leq \sum_{j \in V} 3(1 + \alpha)\Delta_j \\
&\leq 3(1 + \alpha)C(x, y)
\end{aligned}
$$

The approximation guarantee in terms of α is $\max\left\{1 + \frac{1}{\alpha}, 3(1 + \alpha)\right\}$. This is minimised when $\alpha = \frac{1}{3}$. For this value of α, $F(\hat{x}, \hat{y}) \leq 4F(x, y)$ and $C(\hat{x}, \hat{y}) \leq 4C(x, y)$, yielding a 4-approximation.

15.3 Non-preemptive $1|r_j|\sum C_j$ Scheduling

We have earlier seen approximation algorithms for the $P||\max C_j$ scheduling problem, where the aim was to minimize the makespan of a collection of jobs. In the $1|r_j|\sum C_j$ scheduling problem, we only have a single machine and there is an associated release time r_j associated with each job j before which it can not be scheduled. Here, the goal is to minimize the average completion time (or, equivalently, the sum of the completion times). In the non-preemptive version, we further enforce the constraint that once a job is scheduled, it gets to run till completion.

The non-preemptive $1|r_j|\sum C_j$ scheduling problem is NP-hard. However, if we relax the problem by allowing jobs to be preempted and then resumed at any later point of time, then it is known that the $SRPT$ (Shortest Remaining Processing Time) policy, which schedules at each point of time the job with the least remaining processing time, is optimal. A common technique for getting near optimal non-preemptive schedules is to first obtain a solution to the relaxed preemptive version and then convert it to a non-preemptive schedule. We will now give an algorithm to convert a preemptive schedule for a single machine to a non-preemptive schedule with the average completion time at most twice as that of the preemptive schedule. Since we can solve the preemptive version optimally, this will imply a 2-approximation for the non-preemptive $1|r_j|\sum C_j$ problem.

Let C_j^P be the completion time of job j with processing time p_j in a given preemptive schedule. Also let $C_1^P \leq C_2^P \leq \ldots \leq C_n^P$. We create a new schedule by starting the jth job at time $C_j^P + \sum_{k=1}^{j-1} p_k$ and running it till completion. This non-preemptive schedule does not violate the release time constraints. The completion time for the job j in this non-preemptive schedule, C_j^N, will be

$$\begin{aligned} C_j^N &= C_j^P + \sum_{k=1}^{j} p_k \\ &\leq C_j^P + C_j^P \\ &= 2C_j^P \end{aligned}$$

where the inequality follows from the fact that since jobs $1, \ldots, j$ finished before time C_j^P, this must atleast be the sum of the processing times of these jobs. If we start with the preemptive schedule given by SRPT, then by the above conversion, we get a 2-approximate non-preemptive scheduling algorithm.

15.4 Conclusion

In this lecture we saw 2 applications of direct rounding of LP solutions to obtain approximation algorithms: the uncapacitated facility location problem and the $1|r_j|\sum C_j$ non-preemptive scheduling problem. The ideas used for obtaining the integer solution for the UFL problem (filtering and rounding) extend easily to the metric and non-metric K-median problem. Similarly, the technique of converting a preemptive schedule to non-preemptive schedule extends to the $P|r_j|\sum C_j$ scheduling problem. In this case, however, even obtaining the optimal preemptive schedule is NP-hard but a 2-approximate schedule can be obtained.

In the next few lectures we will see the use of other LP rounding techniques, namely, randomized rounding and rounding based on the structure of the basic solution of the linear program.

References

[1] David B. Shmoys, Éva Tardos and Karen Aardal. *Approximation algorithms for facility location problems.* In 29th Annual ACM Symposium on Theory of Computing, pages 265–274, 1997.

[2] Jyh-Han Lin and Jeffrey S. Vitter. *ϵ-approximations with minimum packing constraint violation.* In 24th Annual ACM Symposium on Theory of Computing, pages 771–782, 1992.