Announcements and Outline

Announcements

- No lecture this Friday
- Question: How was the test?

Today’s outline

- Continue LP relaxation of an IP and rounding to an integral solution.
- Integrality gap for the basic IP/LP for the vertex cover problem.
- Set Cover
- A ring routing problem
- The makespan problem in the unrelated machines model
Review of the IP formulation for vertex cover and its LP relaxation

- We discuss weighted vertex cover as a \{0, 1\} IP.

- We will discuss its LP relaxation and “naïve” and why this provides a 2-approximation.

- The integrality gap of this IP/LP relaxation is $2 - \frac{1}{n}$. Adding some additional inequalities (say corresponding to odd cycles) does not help to improve the integrality gap.

- In general there will be many IP formulations for a given problem. An integrality gap pertains to one (or a class of) IP/LP relaxations.

- Despite considerable effort, there is no known polynomial time algorithm that achieves a $2 - \epsilon$ approximation for any $\epsilon > 0$ even for the unweighted case.
An IP for weighted vertex cover

**IP formula for weighted vertex cover**

\[
\begin{align*}
\text{minimize} & \quad \sum_{j=1}^{n} w_j \cdot x_j \\
\text{s.t.} & \quad x_i + x_j \geq 1 \quad \text{for each edge } (i,j) \in E \\
x_i & \in \{0, 1\} \quad \text{for each } i = 1, \ldots, n
\end{align*}
\]

- The intended meaning is that \( x_i = 1 \) iff vertex \( v_i \) is in the cover.
- The LP relaxation is to relax the integrality condition \( x_i \in \{0, 1\} \) to \( x_i \geq 0 \).
- In this problem it follows that an optimal LP solution also satisfies \( x_i \leq 1 \).
Rounding an LP optimal solution

- Suppose $x^*$ is an LP optimum.

- We can apply a “naïve” rounding (naive in the sense that the rounding ignores the input) to the fractional solution by setting $x'_i = 1$ iff $x^*_i \geq 1/2$

- Claim: $x'$ is an integral solution to the IP and hence $V' = \{ v_i \mid x'_i = 1 \}$ is a vertex cover. Why?

Claim

The weight of the cover $V'$ is at most twice the weight of an optimal cover.

Proof.

- Because the LP is a relaxation, it also allows IP solutions. Thus, $OPT_{LP} \leq OPT_{IP}$.
- Then we have $w(V') \leq 2 \cdot OPT_{LP} \leq 2 \cdot OPT_{IP}$.
**The integrality gap**

- For the complete (unweighted) graph on \( n \) nodes, the optimal IP value is \( n - 1 \), whereas the LP optimum value is \( n/2 \).

- For a given IP/LP formulation of a minimization problem, the **integrality gap** is the worst case (over all input instances) ratio of an integral optimum value to a fractional (LP) optimum value.

- The approximation analysis show that this ratio is at most 2 and the previous example shows that it is at least \( (2 - \frac{1}{n}) \).

- For the \( n \) node cycle, the optimum IP solution is \( \lceil n/2 \rceil \) and the LP OPT is \( n/2 \). A naive rounding would be (at best) a 2-approximation.

- For any known polynomial time approach to add additional constraints, the integrality gap essentially remains at 2.

**Informal but commonly used claim**

The integrality gap provides a limit to obtaining an approximation using a particular IP/LP formulation of a problem.
**Set cover**

### The set cover problem

- Given a collection of (possibly weighted) sets \( C = \{S_1, \ldots, S_m\} \) where \( S_i \subseteq U \).
- **Goal:** Find a minimal size (weight) subcollection \( C' \) that covers all the elements in the universe \( U \).

- Set cover generalizes vertex cover and turns out to be NP-hard to approximate better than \( H_n \) where \( n = |U| \). Recall that

\[
H_n = \sum_{k=1}^{n} \frac{1}{k} \approx \ln n
\]

- There is a natural greedy algorithm that will achieve an approximation of \( H_d \) where \( d = \max_i |S_i| \).
- Unless all problems in NP can be computed in time \( n^{O(\log \log n)} \), it is not possible to approximate the set cover problem to within a factor \((1 - \epsilon) \ln n\) for any \( \epsilon > 0 \).
Set cover and the $f$-frequency set cover problem

- We can express the set cover problem as an IP.

**IP formula for set cover**

$$\text{minimize} \quad \sum_{j=1}^{n} w_j \cdot x_j$$

$$\text{s.t.} \quad \sum_{i \colon e \in S_i} x_i \geq 1 \quad \text{for each element } e \in U$$

$$x_i \in \{0, 1\} \quad \text{for each } i = 1, \ldots, n$$

- In the $f$-frequency set cover problem, we assume that every element occurs in at most $f$ sets.

- Vertex cover is a 2-frequency set cover problem. Why? What are the elements and what are the sets?
For the $f$-frequency set cover problem, we can relax it to an LP in the same way that we represented and provided an approximation for the vertex cover problem.

We solve the LP optimally to obtain a solution $x^*$ and then round by setting

$$x'_i = 1 \text{ iff } x^*_i \geq 1/f$$

Similar to the vertex cover approximation, this yields an $f$-approximation algorithm for the $f$-frequency set cover problem.
Ring routing: another IP/LP with naive rounding

The call routing problem

- There is an $n$ node bi-directional ring network $G = (V, E)$ upon which calls must be routed, where

$$V = \{0, 1, \ldots, n - 1\}$$
$$E = \{(i, i + 1 \mod n)\} \cup \{(i, i - 1 \mod n)\}$$

- Calls $c_j$ are pairs $(s_j, f_j)$ originating at node $s_j$ and terminating at node $f_j$.
- Each call can be routed in a clockwise or counter-clockwise direction.
- The load $L_e$ on any directed edge is the maximum number of calls routed on this directed edge.
- **Goal:** Minimize $\max_{e \in E} L_e$.

We can achieve a 2-approximation by an LP relaxation of a natural IP followed by a naive rounding just as in the vertex cover example.
The ring routing example continued

- To formulate this problem as an IP, for each call we will introduce variables $x_j$ and $y_j$ that indicate the direction of call $c_j$.
- (You can also use just one indicator variable to represent the direction but I think it might be easier to think in terms of two such variables.)

**IP formula for ring routing**

minimize $L$

subject to:

1. $\sum_{j : c_j \text{ routed clockwise}} x_j \leq L$
   
   for each edge $(i, i + 1) \mod n$

2. $\sum_{j : c_j \text{ routed clockwise}} y_j \leq L$
   
   for each edge $(i, i - 1) \mod n$

3. $x_j + y_j = 1$
   
   for each $i = 1, \ldots, n$

4. $x_j, y_j \in \{0, 1\}$
   
   for each $i = 1, \ldots, n$
Recall: the makespan problem

The makespan problem for the identical machines model

- The input consists of $n$ jobs $J = \{J_1, \ldots, J_n\}$ that are to be scheduled on $m$ identical machines.
- Each job $J_k$ is described by a processing time (or load) $p_k$.
- **Goal:** Minimize the latest finishing time (maximum load) over all machines.

Theorem 1. The makespan of the assignment computed by GREEDY LOAD BALANCE is at most twice the makespan of the optimal assignment.

Proof: Fix an arbitrary input, and let $OPT$ denote the makespan of its optimal assignment. The approximation bound follows from two trivial observations. First, the makespan of any assignment (and therefore of the optimal assignment) is at least the duration of the longest job. Second, the makespan of any assignment is at least the total duration of all the jobs divided by the number of machines.

$$OPT \geq \max_j T[j]$$

and

$$OPT \geq \frac{1}{m} \sum_j T[j]$$

Now consider the assignment computed by GREEDY LOAD BALANCE. Suppose machine $i$ has the largest total running time, and let $j$ be the last job assigned to machine $i$. Our first trivial observation implies that $T[j] \leq OPT$. To finish the proof, we must show that $Total[i] - T[j] \leq OPT$. Job $j$ was assigned to machine $i$ because it had the smallest finishing time, so $Total[i] - T[j] \leq Total[k]$ for all $k$. (Some values $Total[k]$ may have increased since job $j$ was assigned, but that only helps us.) In particular, $Total[i] - T[j]$ is less than or equal to the average finishing time over all machines. Thus,

$$Total[i] - T[j] \leq \frac{1}{m} \sum_j T[j] \leq OPT$$

by our second trivial observation. We conclude that the makespan $Total[i]$ is at most $2 \cdot OPT$.

2

Theorem 2. The makespan of the assignment computed by SORTED GREEDY LOAD BALANCE is at most $\frac{3}{2}$ times the makespan of the optimal assignment.
Makespan for the unrelated machines model

The input consists of $n$ jobs $\mathcal{J} = \{J_1, \ldots, J_n\}$ and $m$ machines $M_1, \ldots, M_m$.

Each job $J_j$ is represented by a vector $\langle p_{1j}, p_{2j}, \ldots, p_{mj} \rangle$ where $p_{ij}$ represents the processing time of job $J_j$ on machine $i$.

Without loss of generality, we assume $m \leq n$.

Goal: Minimize the latest finishing time (maximum load) over all machines.

We will sketch a 2-approximation IP/LP with non naïve rounding algorithm.

This is the best known poly-time approximation.

It is known that it is NP-hard to achieve better than $\frac{3}{2}$-approximation even for the special case of the restrictive machines model for which every $p_{ij}$ is either some $p_j$ or $\infty$. 
In the IP formulation, the problem is:

\[
\text{minimize } t \\
\text{s.t. } \sum_{i=1}^{m} x_{ij} = 1 \quad \text{for each job } J_j \\
\sum_{j=1}^{n} p_{ij}x_{ij} \leq t \quad \text{for each machine } M_i \\
x_{ij} \in \{0, 1\} \quad \text{for } i = 1, \ldots, m \text{ and } j = 1, \ldots, n
\]

- The intended meaning is that \( x_{ij} = 1 \) iff job \( J_j \) is scheduled on machine \( M_i \).
- The LP relaxation is that \( 0 \leq x_{ij} \). The condition \( x_{ij} \leq 1 \) is implied.
- The integrality gap is unbounded! Why? How do we get around this unbounded integrality gap?