CSC373S Lecture 22

- Our next application of IP/LP is the set cover problem, a generalization of vertex cover. We will consider the following (weighted) set cover problem.

  Given a collection of sets $\mathcal{C} = S_1, ..., S_n$ over a universe $U = \{u_1, ..., u_m\}$ with weight function $w : \mathcal{C} \rightarrow \Re^{\geq 0}$, find a subcollection $\mathcal{C}'$ such that $\cup_{i:S_i \in \mathcal{C}'} S_i = U$

  We can formulate this as a $\{0, 1\}$ IP :

  minimize $\sum_i w_i x_i$

  subj to $\sum_{i:u_j \in S_i} x_i \geq 1$ for all $u_j \in U$

  IP: $x_i \in \{0, 1\}$ (Intended meaning is that S_i in the cover

  LP relaxtion : $x_i \geq 0$

  Vertex cover can be seen as a special case of set cover, namely:

  A vertex v is a set $S_v = \{e_j : v \text{ is adjacent to edge } e_j\}$ so that the universe is the set of edges E and the sets in the collection are the vertices V.

  Note that this collection of sets has the property that each universe element appears in exactly two sets. This leads to what is called the f-frequency set cover problem where each element occurs in at most f sets. Note that vertex cover is essentially the 2-frequency set cover problem. (In the 2-frequency problem, we can also have have universe elements which only occur once but we would then choose the set containing such an element as part of our cover and remove the elements in that sset. We would continue to do so until all elements were in at least two sets.)

  In the f-frequency restriction, there are at most f different S_i in the summation $\sum_{i:u_j \in S_i}$ so we could rewrite this explicitly by stating:

  subj to $\sum_{1 \leq i \leq s(j)} x_{j_i} \geq 1$
  where $S_{j_1}, \ldots, S_{j_{s(j)}}$ are the sets containing u_j for some $s(j) \leq f$.

  Then solving the LP relaxation we round the LP opt $\{x_i^* | 1 \leq i \leq n\}$ by setting $\bar{x}_i = 1$ iff $x_i^* \geq \frac{1}{f}$.

- There is a very natural and efficient greedy algorithm for solving the weighted vertex cover problem with approximation h_d where $d = \max_i |S_i|$. But we want to use this problem to illustrate the concept of *randomized rounding*. The following randomized rounding algorithm will with high probability produce a cover that is within a factor $O(H_d)$ of the optimum. For simplicity we will just prove the factor

$O(\log m)$. (There is also a connection between a primal dual approach to solving the LP relaxation and the "natural" deterministic greedy algorithm that achieves approximation ratio H_d but we will not have time to discuss primal dual algorithms.) We consider the LP relaxation of the weighted set cover problem:

$min \sum_i w_i \cdot x_i$

subj to $\sum_{i:u_j \in S_i} x_i \geq 1$

$x_i \geq 0$ for LP

We solve this LP and find an optimal solution $\{x_1^*, ..., x_n^*\}$. We know that $0 \leq x_i^* \leq 1$ since in an optimal solution, each x_i^* is at most 1.

We now treat the x_i^* values as probabilities and choose S_i (to be in our set cover) with probability x_i^*.

Now this is a covering minimization problem and the chosen sets may not be a cover. So we will most likely have to repeat this process enough times to have a good probability that all elements are covered.

First, it is easy to calculate the expected cost of the sets selected by the LP optimum. $|calE_{\{x_i^*\}}[cost(\mathcal{C}')] = \sum w_i \cdot Prob[S_i \text{ is chosen}] = \sum w_i x_i^* = OPT_{LP}$.

Now we need to calculate the probability that a given $u_j = u$ is not covered. Lets say that u occurs in sets $S_1, ..., S_k$. The LP solution must satisfy the constraint : $\sum_{i:u \in S_i} x_i^* \geq 1$.

Under this constraint, we can maximize the probablity that u is not covered by setting $x_i^* = 1/k$ for $1 \leq i \leq k$. So the probability that u is not covered is at most $(1 - 1/k)^k \leq 1/e$.

Suppose now that we run the same randomized rounding algorithm $c \ln m$ times ($m = |U|$) for some constant c, each time adding sets (given by the rounded LP) to the set cover. While we may be adding the same set many times (and hence overcounting), the cost of the cover is now at most $(c \ln m) OPT_{LP}$.

The probability that u is not covered is now $\leq (1/e)^{c \ln m} = (1/m)^c$.

Let $E_1, \ldots E_m$ be a set of random events with Prob$[E_i] \leq p_i$. Then Prob[at least one E_i occurs] $\leq \sum_{i=1}^m p_i$. Letting E_i be the event that element u_i is not covered. Then by the union bound the probability that some $u \in U$ is not covered is $\leq |U|(1/m)^c = (1/m)^{c-1}$.

Using the Markov inequality we can also say that the expected cost is within $O(logm)OPT_{LP}$ with good probability so that we get both a cover and cost $O(logm)OPT_{LP}$

with good probability which certainly shows that with good probability we get a cover with cost $O(logm)OPT$ since $OPT_{LP} \leq OPT$.

- Another application of randomized rounding is used to show that Max-Sat (for arbitrary formulas) can be approximated with approximation factor $\frac{3}{4}$.

- In some case, it isn't so obvious how to represent an optimization problem as an IP. Consider the Max Cut problem. We can think of a solution as a {0,1} choice about which vertices to (say) put into A in an (A, B) cut. We could have variables $y_i \in \{+1, -1\}$ with the intended meaning $y_i = 1$ (resp -1) iff vertex v_i in A (resp B).

  Then we would want to
  maximize $\sum_{1 \leq i < j \leq n} \frac{1}{2} w(i,j)(1 - y_i y_j)$
  subj to $y_i \in \{+1, -1\}$ i.e. $y_i^2 = 1$

  But obviously the objective (and the condition y_i^2) here is not a linear function and there doesnt appear to be any nice IP way to count the number or weight of edges in $A \times B$. This "quadratic program" leads to a different type of relaxation (semi definite programming SDP) which can be used to provide the best known approximation, namely $\approx .87856$. However, this result is beyond the scope of our discussion. (The same SDP approach provides the same approximation factor for Max-2-Sat.)

  Instead we will think of a cut as the edges in $A \times B$ and hence have a {0,1} variable $x_e = x_{(uv)}$ for every edge e with the intended meaning that $x_{(uv)} = 1$ iff (u, v) in the cut. We can assume that all edges are present by setting $w(e) = 0$ for any edge $e \notin E$. Now we need to find inequalities that will insure that the $\{x_e | x_e = 1\}$ defines a cut. This isnt so obvious but here is what works.

  max $\sum_{(i,j) \in E} w_{(i,j)} \cdot x_{(ij)}$

  subj to $x_{(ij)} \in \{0, 1\}$
  $x_{ij} + x_{jk} \geq x_{ik}; x_{ik} + x_{kj} \geq x_i j$, etc for every triangle (i, j, k) (all permutations)
  $x_{ij} + x_{ik} + x_{jk} \leq 2$

  You can think of these "triangle inequalities" as saying that the possible sizes of cut for each triangle are 0 or 2. These are called metric-cut inequalities.

  Clearly every cut must satisfy these constraints and conversely, every {0,1} solution of this IP defines a cut. This can be seen by the following argument:

  1. Define a relation $i \sim j$ if $x_{ij} = 0$ or $i = j$
  2. Show this is an equivalence relation: transitivity is the only thing to check, and by the triangle condition $x_{ij} = x_{ik} = 0$ implies $x_{jk} = 0$
  3. Show that there are at most 2 equivalence classes. This follows from the second trianlge condition; if i, j, k are in different classes, then $x_{ij} + x_{ik} + x_{jk} = 3$.

4. The equivalence classes are the cut.

- Another interesting IP formulation issue is for the (NP hard) makespan problem on unrelated machines. Early in the course we mentioned makespan problem on m *identical* machines. The goal is to schedule all n jobs on m identical so as to minimize the latest finishing time where each job J_j is described by a processing time p_j.

  In the IP formulation, the problem is:

  minimize t

  subj to $\sum_{1 \leq i \leq m} x_{i,j} = 1$ for each job J_j Each job is scheduled

  $\sum_{1 \leq j \leq n} p_j x_{i,j} \leq t$ The makespan is at most t

  $x_{i,j} \in 0, 1$ Integrality constraint

  Here the intended meaning is that $x_{i,j} = 1$ iff job J_j is scheduled on machine M_i.

  In the unrelated machines model, each job J_j has a (possibly different) processing time $p_{i,j}$ on machine M_i. Whereas the identical machines problem has a reasonably good greedy (4/3) approximation algorithm (sort so that $p_1 \geq p_2 ...$) and then schedule greedily, there is no known greedy algorithm for this problem that has an O(1) approximation.

  We have the same IP with $p_{i,j}$ now replacing p_j in the constraint above. It is easy to see that the LP relaxation of this IP has an unbounded integrality gap: consider one job with procesing time m, which has $OPT = m$ and $OPT_{LP} = 1$. The IP must set $x_{i,j} = 0$ if $p_{i,j} > t$ where as the fractional OPT does not have this constraint. So we want to say "for all (i, j): if $p_{i,j} > t$ then $x_{i,j} = 0$".

  But this isnt a linear constraint!

  Here a non-oblivious rounding is used to produce the best known 2-approximation for this problem. Namely, lets assume that all parameters are integers. Suppose we had a guess T for a bound on the makespan.

  We can obtain such guesses by using a binary search over the set of all possible makespan values. For each such guess T, we set up a linear system of constraints $LP(T)$ as above replacing t by T and removing any $x_{i,j}$ having $p_{i,j} > T$. Now we only want to test for and find a feasible fractional solution for $LP(T)$ (without any objective function) and using binary seacrh we find the smallest T for which $LP(T)$ has a fractional solution.

  Now there will be a more sophisticated "rounding procedure" that will allow us to construct an integral {0,1} solution which will achieve a makespan of $2T$ whereas $OPT \geq OPT_{LP} = T$.

  We wont do this now (or even later) but to do the rounding procedure we need results from LP theory. Letting n denote the number of LP variables and m the number of inequlaities, LP theory shows that there are computable optimal LP solutions (so called extreme points) such that any such solution has at most $n + m$

non zero variable values and (for LPs with integral coefficiants) at least $n - m$ of them are intgeral.