# CSCI 570 - HW 12

December 2, 2022

1. [20 points] A variation of the satisfiability problem is the MIN 2-SAT problem. The goal in the MIN 2-SAT problem is to find a truth assignment that minimizes the number of satisfied clauses. Give a 2 approximation algorithm that you can find for the problem.

Answer:

We can assume for simplicity that no clause contains a variable as well as the complement of the variable. (Such clauses are satisfied regardless of the truth assignment used. So, if they were present ,we must always include them in the set of true clauses anyway, and it really comes down to minimizing the true clauses among the rest. It can be easily shown that the approximation ratio will consequently hold regardless)

We will say that two clauses have a ***literal conflict*** if one of them contains a literal $x$ and other contains $\neg x$. Now we make a series of claims that lead to our algorithm:

1. There exists a truth assignment to variables for which a set of clauses $C$ all evaluate to false if and only if $C$ has no pair with a literal conflict.

   Proof: If a truth assignment makes all the clauses false, then there cannot be a pair of clauses with literal conflict since, if the two contained some $x$ and $\neg x$ respectively, one of them must be true no matter whether $x$ is set to true or false.
   On the other hand, if there were no conflicts, we can simply gather all the literals in this set of clauses, for instance, say $\{x_1, x_2, \neg x_5, x_7\}$ and no variables will have both literal in the set, so we can determine an assignment (in the example above, $\{x_1 = x_2 = x_7 = F, x_5 = T\}$)

to make each literal in the former set false, thereby making the set of clauses containing those literals all false.

2. For a graph $G = \{V, E\}$, a set $S$ is a vertex cover if and only if $V \setminus S$ is an independent set.
   Proof: Consider two vertices $u$ and $v$ in $V\S$. There cannot be an edge between $u$ and $v$ since it will be uncovered as $u$ and $v$ are not in the vertex cover. Thus, $V\S$ has no edges, i.e., is an independent set. Similarly, if $V\S$ is an independent set, every edge in the graph has an endpoint in S, making it a vertex cover.

Now, we give our 2-approximation algorithm as follows:
Let $I$ be an instance of MINSAT consisting of the clause set $C$ and variable set $X$. Construct a graph $G$ where each clause is a vertex. Add an edge between two vertices if the corresponding clauses have a literal conflict. Compute a minimum vertex cover $S$ of $G$ using a 2-approx algorithm (We know that this is possible; for details of the algortihm, check section 11.4 in the textbook.) and output this $S$ as the minimum set of clauses that can be true.
The proof of why this algorithm works needs one final argument:

1. S is a vertex cover of G constructed as above, if and only if there exists a truth assignment that can set all clauses in $V \setminus S$ to false.
   Proof: If S is a vertex cover of G, then $V \setminus S$ is an independent set by result two above. Since $V \setminus S$ has no edges, there are no literal conflicts between the clauses in this set (by construction of G), hence they can all be set to false by result 1 above. The other direction of the proof follows similarly.

It follows that the minimum set of clauses that can be true corresponds to the minimum vertex cover of G. Since we used a 2-approx solution for the vertex cover, our algorithm is a 2-approx algorithm for MINSAT.

2. [20 points] Write down the problem of finding a Min-$s$-$t$-Cut of a directed network with source $s$ and sink $t$ as an Integer Linear Program and explain your program.

$$
\begin{aligned}
\text{minimize} \sum_{(u,v)\in E} c(u,v).x_{(u,v)} \\
\text{subject to} : x_v - x_u + x_{(u,v)} \geq 0 \quad \forall (u,v) \in E \\
x_u \in \{0,1\} \quad \forall u \in V : u \neq s, u \neq t \\
x_{(u,v)} \in \{0,1\} \quad \forall (u,v) \in E \\
x_s = 1 \\
x_t = 0
\end{aligned}
$$

The variable $x_u$ indicates if the vertex $u$ is on the side of $s$ in the cut. That is, $x_u = 1$ if and only if $u$ is on the side of $s$. Setting $x_s = 1$ and $x_t = 0$ ensures that $s$ and $t$ are separated. Likewise, the variable $x_{(u,v)}$ indicates if the edge $(u,v)$ crosses the cut. The first constraint ensures that if $u$ is on the side of $s$ and $v$ is on the side of $t$, then the edge $(u,v)$ must be included in the cut. For completeness, one should argue that with the above correspondence (that is, $x_{(u,v)}$ indicating if an edge crosses the cut), every min-s-t-cut corresponds to a feasible solution and vice versa.

Grading (20pt):

- 15 pt: Correctly write LP
- 5 pt: Correctly explain LP

3. [10 points] 720 students have pre-enrolled for the "Analysis of Algorithms" class in Fall. Each student must attend one of the 16 discussion sections, and each discussion section $i$ has capacity for $D_i$ students. The aggregate happiness level of students assigned to a discussion section $i$ is proportionate to $\alpha_i(D_i - S_i)$, where $\alpha_i$ is a parameter reflecting how well the air-conditioning sysem works for the room used for section $i$ (the higher the

better), and $S_i$ is the actual number of students assigned to that section. We want to find out how many students to assign to each section in order to maximize total student happiness. Express the problem as a integer linear program problem.

Answer:
Our variables will be the $S_i$. Our objective function is:

$$\begin{aligned}
\text{maximize} \sum_{i=1}^{16} \alpha_i (D_i - S_i) \\
\text{subject to: } D_i - S_i \geq 0 \text{ for } 0 < i \leq 16 \\
S_i \geq 0 \text{ for } 0 < i \leq 16 \\
\sum_{i=1}^{16} S_i = 720
\end{aligned}$$

Grading:

- Objective function (4 points)
- Each constraint (2 points)

4. [16 points] A set of $n$ space stations need your help in building a radar system to track spaceships traveling between them. The $i^{th}$ space station is located in 3D space at coordinates $(x_i, y_i, z_i)$. The space stations never move. Each space station $i$ will have a radar with power $r_i$, where $r_i$ is to be determined. You want to figure how powerful to make each space station's radar transmitter, so that whenever any spaceship travels in a straight line from one station to another, it will always be in radar range of either the first space station (its origin) or the second space station (its destination). A radar with power $r$ is capable of tracking space ships anywhere in the sphere with radius $r$ centered at itself. Thus, a space ship is within radar range through its strip from space station $i$ to space station $j$ if every point along the line from $(x_i, y_i, z_i)$ to $(x_j, y_j, z_j)$ falls within either the sphere of radius $r_i$ centered at $(x_i, y_i, z_i)$ or the sphere of radius $r_j$ centered at $(x_j, y_j, z_j)$. The cost of each radar transmitter is proportional to its power, and you want to

minimize the total cost of all of the radar transmitters. You are given all of the $(x_1, y_1, z_1), ..., (x_n, y_n, z_n)$ values, and your job is to choose values for $r_1, ..., r_n$. Express this problem as a linear program.

(a) Describe your variables for the linear program (3 pts).
Answer:
$r_i$=the power of the $i^{th}$ radar transmitter., i=1,2,...n (3 pts)

(b) Write out the objective function (5 pts).
Answer:
Minimize $r_1 + r_2 + ... + r_n$ or $\sum_{i}^{n} r_i$
Defining the objective function without mentioning $r_i$: -3 pts

(c) Describe the set of constraints for LP. You need to specify the number of constraints needed and describe what each constraint represents (8 pts).
Answer:
$r_i + r_j \geq \sqrt{((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2)}$. Or, $r_i + r_j \geq d_{i,j}$ for each pair of stations $i$ and $j$, where $d_{i,j}$ is the distance from station $i$ to station $j$ (6 pts).
We need $\sum_{i=1}^{n-1} i = (n^2 - n)/2$ constraints of inequality (The number of constraints is due to the number of unique paths between each pair of space stations) (2pts).

5. (Ungraded) Given a graph G and two vertex sets $A$ and $B$, let $E(A, B)$ denote the set of edges with one endpoint in $A$ and one endpoint in $B$. The Max Equal Cut problem is defined as follows
**Instance** Graph $G(V, E), V = 1, 2, ..., 2n$.
**Question** Find a partition of $V$ into two $n$-vertex sets $A$ and $B$, maximizing the size of $E(A, B)$.
Provide a factor $\frac{1}{2}$-approximation algorithm for solving the Max Equal Cut problem.

Answer:
Start with empty sets $A$, $B$, and perform $n$ iterations:
In iteration $i$, pick vertices $2i - 1$ and $2i$, and place one of them in $A$ and the other in $B$, according to which choice maximizes $|E(A, B)|$ at that point.

Explanation: In a particular iteration, when we have cut $(A, B)$ and we want to add $u$ and $v$, suppose $u$ has $N_{A_u}$, $N_{B_u}$ neighbours in $A$, $B$ respectively and suppose $v$ has $N_{A_v}$, $N_{B_v}$ neighbours in $A$, $B$ respectively. Then, adding $u$ to $A$ and $v$ to $B$ adds $N_{B_u} + N_{A_v}$ edges to the cut, whereas doing the other way round adds $N_{B_v} + N_{A_u}$ edges to the cut. Since the sum of these two options is nothing but the total number of edges being added to this partial subgraph, the bigger of the two must be at least half the total number of edges being added to this partial subgraph (a subtle point to consider is whether $u$ and $v$ have an edge between them and it can be seen that the argument holds in either case). Since this is true for each iteration, at the end, when all the nodes (and edges) are added, our algorithm is bound to add at least half of the total $|E|$ edges. Naturally since the max equal cut capacity $OPT \leq |E|$, our solution is a $\frac{1}{2}$-approximation.