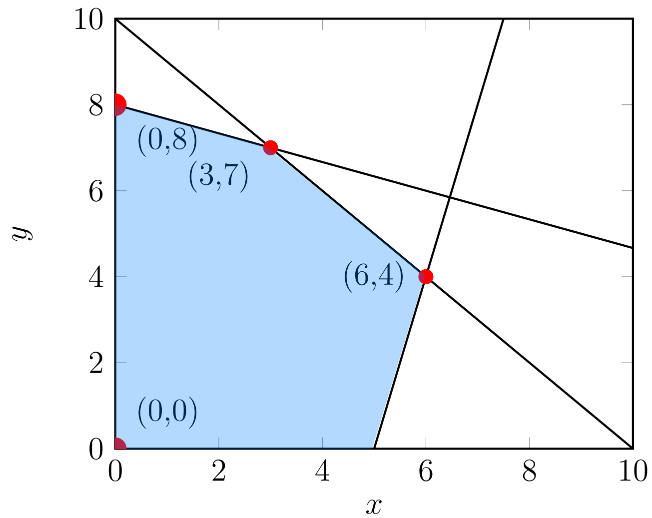
Rishad Rahman (Deepak's Recitation) April 16, 2015

6.046 Problem 8-1

Collaborators: *None*

(a)



(b)

$$
\begin{aligned}
\max 4x_1 + x_2 &= z \\
x_1 + x_2 &\le 10 \\
4x_1 - x_2 &\le 20 \\
x_1 + 3x_2 &\le 24 \\
x_1, x_2 &\ge 0
\end{aligned}
$$

$$
\begin{aligned}
z &= 4x_1 + x_2 \\
x_4 &= 10 - x_1 - x_2 \\
x_5 &= 20 - 4x_1 + x_2 \\
x_6 &= 24 - x_1 - 3x_2
\end{aligned}
$$

(c) Make x_2 basic, switching with x_6 to get

$$\begin{aligned} z &= 8 + \frac{11}{3}x_1 - \frac{1}{3}x_6 \\ x_2 &= 8 - \frac{1}{3}x_1 - \frac{1}{3}x_6 \\ x_4 &= 2 - \frac{2}{3}x_1 + \frac{1}{3}x_6 \\ x_5 &= 28 - \frac{13}{3}x_1 - \frac{1}{3}x_6 \end{aligned}$$

which has solution $z = 8$ at $(x_1, x_2) = (0, 8)$.

Make x_1 basic, switching with x_4 to get

$$\begin{aligned} z &= 19 - \frac{11}{2}x_4 + \frac{3}{2}x_6 \\ x_1 &= 3 - \frac{3}{2}x_4 + \frac{1}{2}x_6 \\ x_2 &= 7 + \frac{1}{2}x_4 - \frac{1}{2}x_6 \\ x_5 &= 15 + \frac{13}{2}x_4 - \frac{5}{2}x_6 \end{aligned}$$

which has solution $z = 19$ at $(x_1, x_2) = (3, 7)$. Make x_6 basic, switching with x_5 to get

$$\begin{aligned} z &= 28 - \frac{8}{5}x_4 - \frac{3}{5}x_5 \\ x_1 &= 6 - \frac{1}{5}x_4 - \frac{1}{5}x_5 \\ x_2 &= 4 - \frac{4}{5}x_4 + \frac{1}{5}x_5 \\ x_6 &= 6 + \frac{13}{5}x_4 - \frac{2}{5}x_5 \end{aligned}$$

which has solution $z = 28$ at $(x_1, x_2) = (6, 4)$ which is the optimal solution.

(d)

$$\begin{aligned} \min 10y_1 + 20y_2 + 24y_3 &= z \\ y_1 + 4y_2 + y_3 &\geq 4 \\ y_1 - y_2 + 3y_3 &\geq 1 \\ y_1, y_2, y_3 &\geq 0 \end{aligned}$$

which has optimal solution $z = 28$ as well.

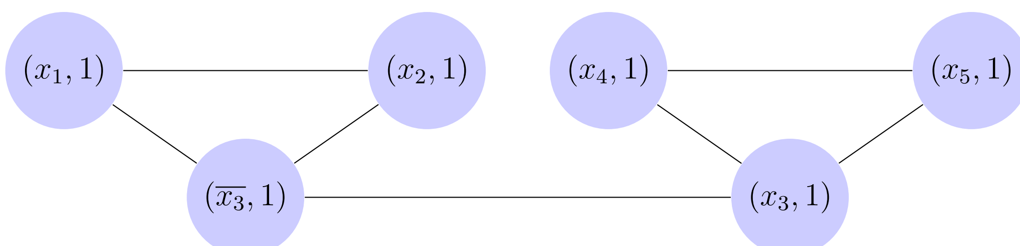
Rishad Rahman (Deepak's Recitation) April 24, 2015

6.046 Problem 8-2

Collaborators: *None*

(a) TRIPLE-SAT $\in$ NP because guessing three solutions and verifying them is linear time. Now we will show SAT $\leq_P$ TRIPLE-SAT. Suppose we had the SAT formula ϕ_x which uses the literals x_i. We make two copies ϕ_y and ϕ_z where we have the same formula except $x_i \to y_i$ in the former and $x_i \to z_i$ in the latter. Consider $\phi = \phi_x \vee \phi_y \vee \phi_z$. I claim that ϕ has at least three distinct satisfying assignments iff ϕ_x is satisfiable.
$\Rightarrow$ A solution exists which means at least one of ϕ_x, ϕ_y, ϕ_z must be true but since each is a copy of ϕ_x, we get a satisfying assignment for ϕ_x.
$\Leftarrow$ We construct three solutions for ϕ using the fact that ϕ_x is satisfiable. Use the satisfying assignment for x_i and the complement for y_i and z_i. Thus ϕ_x is true and so ϕ is satisfiable. We can do this for each copy by symmetry hence we get three solutions. Making three copies is linear, and constructing ϕ is also linear so we have a polynomial time reduction and TRIPLE-SAT is NP-Complete.

(b) We will use a reduction from 3SAT. Suppose our 3SAT is $\phi = c_1 \wedge c_2 ... \wedge c_n$ with $c_i = y_{i_1} \vee y_{i_2} \vee y_{i_3}$ and $y_j \in \{x_j, \overline{x_j}\}$. We create gadgets such that for each clause c_i we have literal vertices $y_{i_1}, y_{i_2}, y_{i_3}$ all with weight 1 and mutually connected. Do this for all clauses then connect all x_i with $\overline{x_i}$. For example $\phi = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_3 \vee x_4 \vee x_5)$ is represented by



We claim that this graph has a profit $\geq n$ iff ϕ is satisfiable.
$\Rightarrow$ This forces each gadget to have exactly one donut shop. Make the literals which get shops true to get the satisfying assignment. x is connected with $\overline{x}$ we never get a contradictory assignment and each clause gets at least one true assignment hence ϕ is satisfiable.
$\Leftarrow$ From each clause, pick a literal that is true and give it a donut shop in the clause's gadget. Since ϕ is satisfiable we never have a contradictory assignment hence x and $\overline{x}$ don't get shops at the same time and because we are picking one literal from each clause, each gadget gets exactly one donut shop so mutual neighbors don't get shops. Hence we have a profit of n. This completes the reduction.

If we can find the maximum total profit, that means we instantaneously know if for k there exists U' such that $\sum_{u \in U'} p(u) \geq k$ since we can take $U' = U$ if $k \leq$ the maximum profit, otherwise there is no solution. Hence this would solve DONUT in polynomial time and therefore any NP problem in polynomial time which would imply P=NP.

(c) The decision problem is to find whether for the specified inputs and a given k is there a schedule such that the profit is $\geq k$. This is NP since guessing the start time of each task (if we use the task), checking if there are conflicting tasks, checking if the deadline is met for each task, and checking if the profit is $\geq k$ are all polynomial. We will now show a reduction from SUBSET-SUM. If the set of numbers is $a_1, a_2, ...a_n$ and the target sum is t, then $\forall i$ we create task i with profit a_i, processing time a_i, and deadline t. We claim that there is schedule with profit $\geq t$ iff there is a subset with sum t.
$\Rightarrow$ Let the profits=processing times of the tasks we choose correspond to the numbers in our subset, S. Since the deadlines for the tasks are all t we must have that the total processing time $\displaystyle\sum_{a_i \in S} a_i \leq t$. But the profit $\displaystyle\sum_{a_i \in S} a_i \geq t$ hence we must have $\displaystyle\sum_{a_i \in S} a_i = t$.
$\Leftarrow$ We choose task i if $a_i \in S$. Since $\displaystyle\sum_{a_i \in S} a_i = t$ and the processing time and profit for task i is a_i we have that the total processing time is $t \leq t$ and the total profit is $t \geq t$ which completes the reduction.