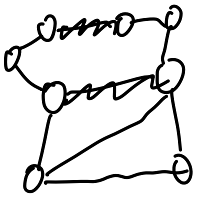
20211507 컴퓨터공학과 김동건

5.7 Yakety Yak. Consider the following model of cellphone conversations. We have an undirected graph $G = (V, E)$ where the vertices are people, and each edge indicates that two people are within range of each other. Whenever two people are talking, their neighbors must stay silent on that frequency to avoid interference. Thus a set of conversations consists of a set of edges $C \subset E$, where vertices in different edges in C cannot be neighbors of each other.

The *cellphone capacity* of G is the largest number of conversations that can take place simultaneously on one frequency, i.e., the size of the largest such set C. Now consider the following problem:



> CELLPHONE CAPACITY
>
> Input: A graph G and an integer k
>
> Question: Is there a set C of conversations with $|C| \geq k$?

Prove that $\overbrace{\text{CELLPHONE CAPACITY}}^{\text{C.C}}$ is NP-complete.

If we show that both ① C.C Problem is NP. and ② reduce np complete problem to C.C Problem.

Then, C.C is NP-complete.

① is trivial. Because if we have conversations, then we can check it is valid in Polynomial time.

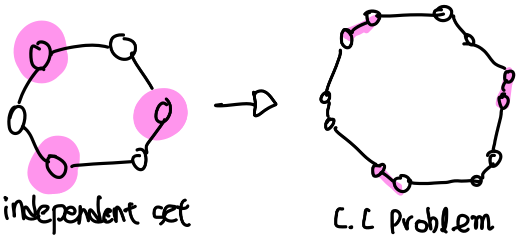
② is not easy. C.C Problem looks similar to independenset problem. So, Let's reduce independent set to C.C problem.

Independent set) Input: Graph G (V, E), k

C.C Problem) Input: Graph G' (V', E'), k'

Let's divide each vertex in independent set into two vertex, and add edge between them. ex) 

ex)



Claim is independen set is k in (V, E) ⟺ C.C problem's answer is k' in (V', E')

And, here we set k' = k.

proof of →)

If vertex u is independent set in G, then make conversation between u-u'.

proof of ←)

If x-y is conservation in G',

select one of them, and select one is u or u' then

select u in independent set.

∴ We can reduce it in Polynomial time.

So, C.C Problem is NP-complete.

5.8 Max-2-SAT reloaded. Give a simpler proof that Max-2-SAT is NP-complete by reducing to it from the unweighted version of Max Cut. Hint: replace each edge with a pair of clauses.

As many clauses as possible must be satiestied...

Claim: MAX CUT $\le$ MAX-2-SAT

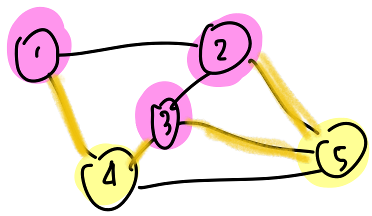
$\longrightarrow$ reduction

We already know that MAX-CUT is np-complete (in l 20)

And MAX-2-SAT is in np. ($\because$ If cut is provided, we can check that cut is max cut in Poly time)

So, If we can reduce, MAX-2-SAT is also np-complete.

ex)



max cut = 4

every edge (x, y), we can represent edge to 2clause

$(x \vee y) \wedge (\bar{x} \vee \bar{y})$. ($x$: pink, $\bar{x}$: yellow)

Then, If x and y is same side, only one of 2 clause is true. But if x and y is opposite site, 2 clause is true,

So, set the MAX-2-SAT value to $|E|$ + maxcut. ($|E|$ is number of edges in maxcut graph)

max-2-SAT value is $|E|$ + maxcut if and only if MAX-CUT is maxcut.

Now, we reduce MAX-CUT to Max-2-SAT.

5.11 Hypercoloring. A *hypergraph* $G = (V, E)$ is like a graph, except that each "edge" can be a set of more than 2 vertices. A hypergraph is *k-uniform* if every edge consists of k vertices. Finally, we say that a hypergraph is 2-*colorable* if we can color its vertices black and white in such a way that no edge is monochromatic; that is, every edge contains at least one black vertex and at least one white one.

Prove that HYPERGRAPH 2-COLORING is NP-complete, even for the special case of 3-uniform hypergraphs. (Don't reply that this is just NAE-3-SAT in disguise, since NAESAT allows variables to be negated.)

Hypergraph 2- coloring is np, because if we have coloring result, we can check it satisfy condition in polynomial time.

I will reduce $\underset{\text{(np-complete)}}{\text{NAE-3-SAT}}$ to $\underset{\text{(special case of hypergraph)}}{\text{3-uniform hypergraph 2-coloring}}$.

So, NAE-3-SAT $\le$ 3-uniformhypergraph 2-coloring $\le$ hyporgraph 2-coloring.

If reduce is success, That means hypergraph 2-coloring is np- complete.

Now let's reduce NAE-3-SAT to 3- uniform hypergraph.

We can divide each variables into two vertices (ex) $\overset{\text{variable}}{u} \rightarrow \underbrace{u, \bar{u}}_{\text{vertex}}$)

Now, we have to add constrain that u and $\bar{u}$ have different color.

To do this, add 3 edges p, q, r and make edges $\{p, q, r\}$, $\{p, u, \bar{u}\}$, $\{q, u, \bar{u}\}$, $\{r, u, \bar{u}\}$.

Finally, each clause, we add edges represent clause.

ex) $\underset{\text{Clause}}{(x \vee \bar{y} \vee z)} \rightarrow \underset{\text{edge}}{\{x, y, z\}}$

It implies that each clause have two colors.

So, we reduce it in Polynomial time.

$\therefore$ 3- uniformhypergraph 2-coloring is np-complete

so hypergraph 2-coloring is np-complete.

5.14 INDEPENDENT SET direct from SAT. Using the same gadgets as shown in Figure 5.14 for the reduction NAE-3-SAT ≤ MAX CUT, it is possible to reduce 3-SAT directly to INDEPENDENT SET. Explain how.

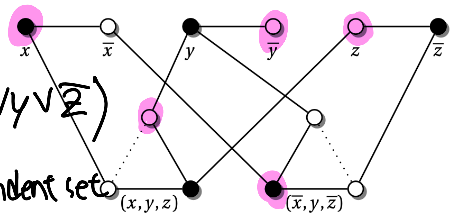
3-SAT ≤ independent set

$\longrightarrow$ reduction

ex) $(x \vee y \vee z) \wedge (\bar{x} \vee y \vee \bar{z})$

$x = 0$
$y = 1$
$z = 0$

●: independent set



Claim: 3-SAT with n variables, m clauses is satisfiable $\Leftrightarrow$ independent set is n+m in graph G made by Figure 5.14's method

Proof of $\rightarrow$)

Let's assume 3-SAT is satisfiable. Prepare empty set S and satisfying assignments.

Each variables, If variable x is true, add $\bar{x}$ to set S
else variable x is false, add x to set S

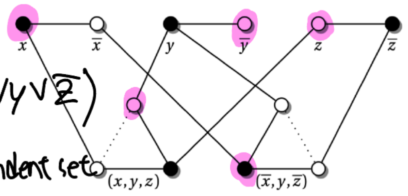
And each clauses, (=triangles) there is at least one true variables. Add only one of them, to set S.

Now, S is independent set size n+m.

ex) $(x \vee y \vee z) \wedge (\bar{x} \vee y \vee \bar{z})$

$x = 0$
$y = 1$
$z = 0$

●: independent set



Proof of $\leftarrow$)

We have independent set S size n+m in G. Each pair of vertex that represent u, $\bar{u}$. only one of them is in S. If u is in S, then set u = false, else set u = true.

And each triplet of vertex that represent clauses,

only one of them is in S. if that is u

It represent u is true.

So,

only if independent set size of $n+m$, 3 sat is satisfiable.

5.27 Augmented matchings. If we apply the reduction from MAX BIPARTITE MATCHING $\overset{\le}{\text{to}}$ MAX FLOW discussed in Section 3.8, a partial matching corresponds to a flow of one unit along each edge in the matching, as well as along the edges that attach these edges to s and t. The Ford–Fulkerson algorithm then looks for augmenting paths, i.e., paths from s to t along edges of nonzero capacity in the residual graph. Show that adding flow along an augmenting path is the same as flipping the edges along an alternating path, and increasing the size of the matching. Hint: consider Figure 5.25.

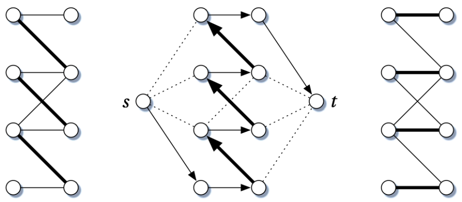


FIGURE 5.25: Increasing flow along an augmenting path corresponds to flipping edges along an alternating path, which increases the size of a partial matching.

Argumenting path is consists of

$(s \to \text{left}_1)$, $\left(\begin{matrix}\text{left}_1 \to \text{right}_1 \to \text{left}_2 \\ \to \text{right}_2 \to \cdots \to \text{right}_k\end{matrix}\right)$, $(\text{right}_k \to t)$ edges.

every edges $\text{right}_i \to \text{left}_{i+1}$ (reverse edge) is in matching. And left_1, right_k isn't in matching. So, If we add flow through this argumenting path, Reverse edges are deleted in matching. And every edges $\text{left}_i \to \text{right}_i$ are added to new matching.

Here, there are $(k-1)$ reverse edges, and k forward edges. So, it increases matching size by one.

And it is also same with flippig edges along alternating path because $\left(\begin{matrix}\text{left}_1 \to \text{right}_1 \to \text{left}_2 \\ \to \text{right}_2 \to \cdots \to \text{right}_k\end{matrix}\right)$ is alternating path in Bipartite graph.