COMP260 Spring 2014 Notes: February 4th

Andrew Winslow

In these notes, all graphs are undirected. We consider matching, covering, and packing in bipartite graphs, general graphs, and hypergraphs. We also introduce the concept of NP-hardness.

1 Matching in Bipartite Graphs

In lecture 2 (by Greg), we considered matchings and perfect matchings on graphs:

Definition 1. *A* matching *of a graph $G = (V, E)$ is a subset $E' \subseteq E$ such that every vertex $v \in V$ is an endpoint of at most one edge in E'.*

Definition 2. *A* perfect matching *of a graph $G = (V, E)$ is a subset $E' \subseteq E$ such that every vertex $v \in V$ is an endpoint of exactly one edge in E'.*

A characterization of the bipartite graphs with perfect matchings was also given:

Theorem 3 (Hall's Theorem)**.** *A bipartite graph $G = (V_1, V_2, E)$ has a perfect matching iff for every subset $V \in V_1$, $|V| \leq |\{v_2 \in V_2 : v_1 \in V_1, \{v_1, v_2\} \in E\}|$.*

For a moment I want to think about a related problem on graphs you've probably seen before:

Definition 4. *Given a graph $G = (V, E)$, a subset $V' \subseteq V$ is a* vertex cover of G *provided for every edge $\{v_1, v_2\} \in E$, either v_1 or v_2 is in V'.*

Problem 5 (Minimum vertex cover)**.** *Given a graph $G = (V, E)$, find a smallest vertex cover of G.*

Any set of vertices V' with $\{\{u, v\} : u \in V'\}$ (smallest or otherwise) is called a *vertex cover* of the graph. Intuitively, a cover is a dusting of vertices so that every edge in the graph is "covered" by some vertex, although you might also say that each edge is "hit" by some vertex. Note that vertex cover is a *minimization problem*, with the goal to find a vertex cover that is as small as possible. On the other hand, finding a large matching in a graph is a *maximization problem*:

Problem 6 (Maximum matching)**.** *Given a graph $G = (V, E)$, find a largest set of edges $E' \subseteq E$ such that E' is a matching on G.*

If a graph G has a perfect matching, then any solution to the maximum matching problem on G should be a perfect matching. Hall's Theorem says that determining whether a graph has a perfect matching is equivalent to some condition on every subset of one of the wo vertex partitions. But you also that it is possible to determine whether a matching is maximum via checking for an augmenting path:

Theorem 7. *For a bipartite graph $G = (V_1, V_2, E)$, a set of edges $E' \subseteq E$ is a maximum matching iff there is no path $v_1, v_2, \ldots, v_k$ in G with edges alternating in membership between $E - E'$ and E'.*

So Hall's Theorem and alternating paths are two ways to ideas related to finding a maximum matching of a graph. Here's another one, with a proof courtesy of [1]:

Theorem 8 (König's Theorem)**.** *Given a bipartite graph $G = (V_1, V_2, E)$, G has a maximum matching of size a if and only if G has a minimum vertex cover of size a.*

Proof. Denote the size of a maximum matching and minimum vertex cover of a graph G by $\alpha(G)$ and $\beta(G)$, respectively. First, observe that for any graph $G = (V, E)$, $\alpha(G) \leq \beta(G)$, since any vertex cover requires at least one endpoint of every edge in a matching. Now suppose, for the sake of contradiction, that G is a minimal graph with $\alpha(G) \neq \beta(G)$. Since G is minimal, G is connected. Since G is connected and $\alpha(G) \neq \beta(G)$, G is not a cycle or a path, and so G has at least one vertex u of degree 3 or more with a neighbor v.

If $\alpha(G - v) < \alpha(G)$, then since G is minimal, $G - v$ has a cover V' with $|V'| = \alpha(G - v) < \alpha(G)$. So $V' \cup \{v\}$ is a vertex cover of G and $|V' \cup \{v\}| \leq \alpha(G)$. So $\beta(G) \leq \alpha(G)$ and so $\beta(G) = \alpha(G)$, a contradiction. So $\alpha(G - v) \not< \alpha(G)$ and $\alpha(G - v) = \alpha(G)$. So there exists a maximum matching E' of G with no edges incident to v.

Since u is degree 3 or more, there is an edge e incident to u, not incident to v, and not in E'. Since G is minimal, there exists a cover V'' of $G - e$ with $|V''| = |E'| = \alpha(G)$. Since E' has no edges incident to v, $|V''| = |E'|$, and V'' must have a vertex on each edge of E', v is not in V''. So V'' contains u and is a cover of G. So $\alpha(G) = \beta(G)$. □

The key to this proof is repeated abuse of the minimality of the counterexample to prove that there exists a cover of a slightly smaller graph with size $\alpha(G)$ that also suffices to cover G.

It is natural to wonder whether extending König's Theorem to general graphs is possible. However, think of even a simple case of the complete graph on n vertices, K_n. Every matching, including maximum matchings, has size at most $\lfloor n/2 \rfloor$. Every vertex cover, including minimum vertex covers, has size at least $n - 1$, since omitting any pair of vertices from the cover leaves the edge between them uncovered. So in this case the maximum matching and minimum vertex cover sizes differ by about $n/2$.

It should also be noted that König's Theorem and Hall's Theorem are actually equivalent, as are many other theorems (Menger's Theorem, Egerváry's statement, Birkhoff-Von Neumann Theorem, Dilworth's Theorem, Max-Flow Min-Cut Theorem), even though they may not look that way, e.g. the Birkhoff-Von Neumann Theorem is a statement about doubly stochastic matrices.

2 Edmonds' Algorithm for Maximum Matching

3 Minimum Edge Covers

Just like vertex covers are subsets of vertices where every edge of the graph is coincident to some vertex in the cover, edge covers are subsets of edges so that every vertex of the graph is coincident to some edge in the cover:

Definition 9. *Given a graph $G = (V, E)$, a subset $E' \subseteq E$ is an* edge cover of G *provided every vertex in V is an endpoint of some edge in E'.*

Homework 1. *Prove that every perfect matching is a minimum edge cover.*

Edge covers and matchings are sets of edges with opposite goals: covers must touch all the vertices, while matchings cannot share vertices.

Problem 10 (Minimum edge cover)**.** *Given a graph $G = (V, E)$, find a minimum edge cover.*

The minimum edge cover problem is the covering (minimization) equivalent of the packing (maximization) problem for edges known as the maximum matching problem. Somewhat weirdly, there is a simple way to turn any maximum matching into a minimum edge cover by adding extra edges to vertices not covered by edges in the maximum matching, giving the following result:

Theorem 11. *The minimum edge cover problem has a polynomial-time algorithm.*

Homework 2. *Prove that for any graph $G = (V, E)$ with maximum matching $E' \subseteq E$, there exists a minimum edge cover E'' with $E' \subseteq E''$ and $|E''| - |E'| = |V| - 2|E'|$. That is, every maximum matching can be extended into a minimum edge cover by adding one edge per vertex.*

4 Other Packing and Covering Problems

Many problems on graphs are either minimization problems where the goal is to use the fewest of something to cover the entire graph, or maximization problems where the goal is the pack as many things into the graph. Minimum vertex cover and edge cover are clearly covering problems, while maximum matching is a packing problem. Here we give two other well-known packing and covering problems on graphs:

Definition 12. *Given a graph $G = (V, E)$, a subset $V' \subseteq V$ is an* independent set *provided that for any pair of vertices $u, v \in V'$, $\{u, v\} \notin E$.*

Problem 13 (Maximum independent set)**.** *Given a graph $G = (V, E)$, find a maximum independent set.*

Definition 14. *Given a graph $G = (V, E)$, a subset $V' \subseteq V$ is a* dominating set *if for every vertex in V is either in V' or a neighbor of a vertex in V'.*

Problem 15 (Minimum dominating set)**.** *Given a graph $G = (V, E)$, find a minimum dominating set.*

There is evidence that both of these problems do not have efficient (polynomial-time) algorithms, but developing a proof has turned out to be harder and worth more[1] than originally thought. In Section 7 we return to the dominating set problem to prove a theorem that says that no polynomial-time algorithm for the minimum dominating set problem exists, assuming a widely believed conjecture. Before then, we need to build some machinery.

5 NP-hardness

For a moment, let's forget about problems phrased like "Let G be a graph. Find a minimum ..." or "Let G be a graph. Find a maximum ..." and instead only consider problems of the form "Let G be a graph and a an integer. Does there exist a ... of size at most a?" and "Let G be a graph and a an integer. Does there exist a ... of size at least a?" These problems only ask for a decision, "Yes" or "No", and we call them *decision problems*. Consider the following subset of decision problems:

Definition 16. *A problem is said to be* in the complexity class NP *if there exists a polynomial-time algorithm to decide whether a solution is valid.*

Notice that NP is the class of decision problems for which another decision problem (the problem of determining whether a given solution is valid) has a polynomial-time algorithm. This is a little subtle, but here are some examples:

Problem 17 (Independent set)**.** *Given a graph $G = (V, E)$ and integer a, does G have an independent set of size at least a?*

Problem 18 (Dominating set)**.** *Given a graph $G = (V, E)$ and integer a, does G have a dominating set of size at most a?*

Problem 19 (Vertex cover)**.** *Given a graph $G = (V, E)$ and integer a, does G have a vertex cover of size at most a?*

For each of these problems, a proposed solution is given as a set of vertices. Determining whether the set is an independent set, dominating set, or vertex cover can be done in polynomial-time by checking various edges and vertices in the graph. So each of these problems is in the complexity class NP.

A problem's membership in NP is an upper bound on its difficulty – it means that the problem has some nice structure and can be solved using some bounded amount of time and space on a computer. Membership in the class NP in particular means that the problem has a polynomial-time algorithm on a special kind of computer that makes lucky guesses. There are currently 495 known complexity classes[2] with many kinds of restrictions, and showing relationships between these classes is an active area of research.

[1] $1,000,000 (`http://en.wikipedia.org/wiki/Millennium_Prize_Problems`)

[2] `https://complexityzoo.uwaterloo.ca/Complexity_Zoo`

After seeing that maximum matching and minimum edge cover problems had polynomial-time algorithms, you may be wondering whether problems like minimum vertex cover (or maximum independent set, or minimum dominating set...) also have polynomial-time algorithms. The truth is that we don't know.

What we do know is that these problems are all related in a the following way: given an instance A of one problem, this instance can be transformed into an instance of another problem A', so that the answer to A is "Yes" if and only if the answer to A' is "Yes. Moreover, the transformation takes only polynomial time. Such a transformation is called a *polynomial-time many-one reduction*.

Ok...so why is *this* useful? Many, many problems are known to be polynomial-time many-one reducible to each other (notice this reducibility is transitive). Problems like independent set, dominating set, and vertex cover are all known to be polynomial-time many-one reducible to each other, and the hardest problems in the class NPare also reducible to them. It is conjectured that the hardest problems in the class NPdo not have polynomial-time algorithms (called the P $\neq$ NP conjecture). If they don't, then neither do independent set, dominating set, and vertex cover, since these hardest problems can be turned into independent set, dominating set, and vertex cover problems in polynomial time. So constructing a polynomial-time many-one reduction from one of these problems to a new problem shows the new problem also likely does not have a polynomial-time algorithm.

We call these problems that are as hard as the hardest problems in NP NP-*hard* problems. Suppose we are told that the independent set problem (Problem 17) is NP-hard:

Theorem 20. *The independent set problem is* NP*-hard.*

This means that if the P $\neq$ NP conjecture holds, there is no polynomial-time algorithm for solving the independent set problem. Neat. Now let's use this to show that vertex cover problem is also NP-hard. Recall that this means that there is a polynomial-time many-one reduction from some known NP-hard problem to the vertex cover problem. Let's use the independent set problem, and preface this with a little result about independent sets and vertex covers in a graph:

Lemma 21. *Given a graph $G = (V, E)$, a subset $V' \subseteq V$ is an independent set iff $V - V'$ is a vertex cover.*

Proof. First, suppose $V' \subseteq V$ is an independent set. Consider an edge $\{v_1, v_2\} \in E$. If V' is an independent set, then either $v_1 \notin V'$ or $v_2 \notin V'$. So either $v_1 \in V - V'$ or $v_2 \in V - V'$. So for every edge $\{v_1, v_2\} \in E$, either $v_1 \in V - V'$ or $v_2 \in V - V'$. So $V - V'$ is a vertex cover.

Now suppose $V' \subseteq V$ and $V - V'$ is a vertex cover. Then for every edge $\{v_1, v_2\} \in E$, either $v_1 \in V - V'$ or $v_2 \in V - V'$ (or both). So v_1 and v_2 are not both in V'. So V' is an independent set. □

Theorem 22. *The vertex cover problem is* NP*-hard.*

Proof. Let $A = \langle G, a \rangle$ be an instance of the independent set problem.[3] Let $G = (V, E)$. We perform a polynomial-time reduction of A to the instance $A' = \langle G, |V| - a \rangle$ of the vertex cover problem.

Suppose the solution to A is "Yes". Then there is an independent set $V' \subseteq V$ of G with $|V'| \geq a$ and by Lemma 21, the set $V - V'$ is a vertex cover of G with $|V - V'| \leq a$. So the solution to A' is "Yes".

Now suppose the solution to A' is "Yes". Then there's a vertex cover $V' \subseteq V$ of G with $|V'| \leq a$ and by Lemma 21, the set $V - V'$ is an independent set of G with $|V - V'| \geq a$.

So the reduction of A to A' is polynomial-time and many-one reduction from instances of the independent set problem to instances of vertex cover. Then since the independent set problem is NP-hard (Theorem 20), the vertex cover problem is NP-hard. □

The vertex cover problem also remains NP-hard if you only consider instances where the graphs are 3-regular:

Definition 23. *A graph is* 3-regular *provided every vertex has degree 3.*

Problem 24 (3-Regular vertex cover)**.** *Given a 3-regular graph G and integer a, does G have a vertex cover of size at most a?*

Theorem 25. *The 3-regular vertex cover problem is* NP*-hard.*

6 Set Cover and Covering-Hitting Duality

Let's leave graphs for a moment and talk about sets.

Definition 26. *Given a set $U = \{1, 2, \ldots, n\}$ and a set of sets $S = \{S_1, S_2, \ldots, S_m\}$ with each $S_i \subseteq U$, a* set cover of U *is a subset $S' \subseteq S$ such that $\bigcup_{S_i \in S'} S_i = U$.*

Normally the U set is called the *universe*.

Problem 27 (Set cover)**.** *Given a set U, set of sets S with each $S_i \subseteq U$, and an integer a, is there a set cover of U with size at most a?*

All I want to observe for now is that there's a duality between sets and elements of the universe. Think about each set as an element and each element as a set, where an element (formerly a set) is in a set (formerly an element) if the former set contained the former element. Then a set contains an element in the primal if and only if the corresponding element "hits" (lies in) the corresponding set in the dual. So in the dual, the set cover problem becomes another problem:

Definition 28. *Given a set $U = \{1, 2, \ldots, n\}$ and a set of sets $S = \{S_1, S_2, \ldots, S_m\}$ with each $S_i \subseteq U$, a* hitting set of S *is a subset $U' \subseteq U$ such that for every $S_i \in S$, there exists an element $u \in U'$ such that $u \in S_i$.*

[3] The $\langle \ldots \rangle$ notation is commonly used to specify an instance of a problem.

Problem 29 (Hitting set)**.** *Given a set U, set of sets S with each $S_i \subseteq U$, and an integer a, is there a hitting set of size at most a?*

This is because "covering" all the elements in the primal becomes a matter of "hitting" all the sets in the dual.

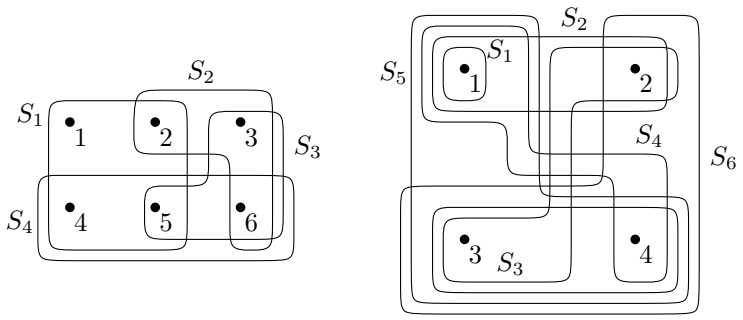

Figure 1: A primal instance of set cover (left) and corresponding instance of hitting set (right).

7 Dominating Set and Set Cover

Set cover is a really classic problem, in part because set cover and its variants are closely related to many other popular problems (kind of like linear programming, which we might see later in the semester). Take the dominating set problem for instance. Consider an instance of the dominating set and the possibility of expressing the instance as an instance of the set cover problem (Figure 2).

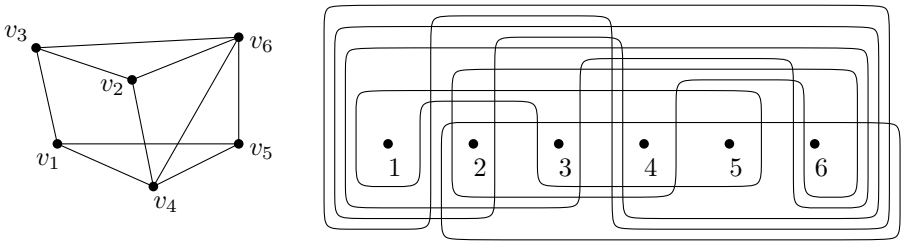

Figure 2: An instance of dominating set (left) and corresponding instance of set cover (right). Each set in the set cover instance corresponds to a vertex and its neighbors. For instance, the set $S_3 = \{1, 2, 3, 6\}$ corresponds to v_3 at its neighbors v_1, v_2, and v_6.

Any instance of dominating set (specified by a graph $G = (V, E)$) can be expressed as a set cover instance with $U = \{1, 2, \ldots, |V|\}$ and $S = \{\{u : \{v, u\} \in E\} \cup \{v\} : v \in V\}$. So the dominating set problem is a special case of the set cover problem. Now consider the other direction (Figure 3).

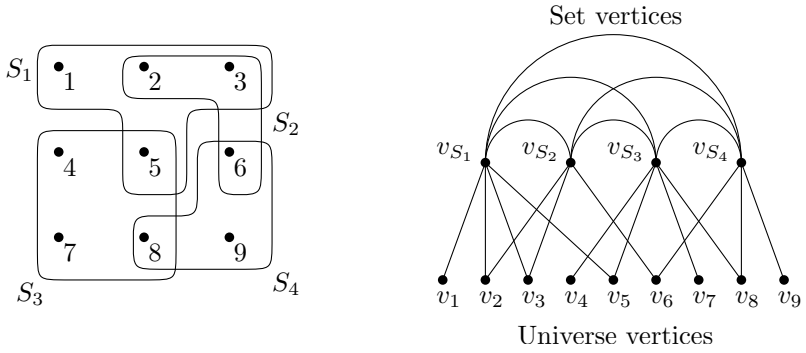

Figure 3: An instance of set cover (left) and corresponding instance of dominating set (right). Each set in the set cover instance corresponds to a set of edges in the dominating set instance between a *set vertex* and a set of *universe vertices*.

Here the correspondence isn't quite so easy. We make a vertex for each set (called the *set vertices*), a vertex for each element of the universe (called the *universe vertices*), an edge between each set and every element the set contains (encoding the sets) and an edge between every pair of set vertices.

Since set vertices are connected in a clique, any vertex is sufficient to dominate them all. Moreover, putting a universe vertex v in the dominating set is a bad idea, since selecting a set vertex connected to v covers a superset of those vertices dominated by v, and the goal is to find a dominating set that is small.

8 Hypergraph Covers and Set Covers

You have likely not seen hypergraphs before:

Definition 30. *A* hypergraph $G = (V, E)$ *consists of a set of* vertices V *and a set of* hyperedges E. *Each hyperedge $e \in E$ is a set of vertices $e \subseteq V$. Two vertices $v_1, v_2 \in V$ are* connected *if there exists a hyperedge $e \in E$ such that $v_1, v_2 \in e$.*

If each hyperedge in a hypergraph G has the same cardinality k, then G is said to be *k-uniform*. Regular graphs are exactly the 2-uniform hypergraphs. Consider the minimum edge cover problem on hypergraphs:

Definition 31. *Given a hypergraph $G = (V, E)$, a subset $E' \subseteq E$ is a* hyperedge cover of G *provided that for every vertex $v \in V$, there exists some edge $e \in E'$ such that $v \in e$.*

Problem 32 (Hypergraph edge cover)**.** *Given a hypergraph $G = (V, E)$ and an integer a, does there exist a hyperedge cover of G with at most a hyperedges?*

Recall that the minimum edge cover problem (on graphs, i.e. 2-uniform hypergraphs) has a polynomial-time algorithm where a maximum matching is extended to a minimum edge cover. What about general hypergraphs? Unfortunately, no.

Theorem 33. *The hypergraph edge cover problem is* NP*-hard.*

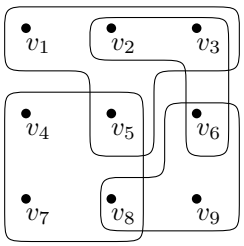

Figure 4: A hypergraph $G = (V, E)$, with $V = \{v_1, v_2, \ldots, v_9\}$ and $E = \{\{v_1, v_2, v_3, v_5\}, \{v_4, v_5, v_7, v_8\}, \{v_2, v_3, v_6\}, \{v_6, v_8, v_9\}\}$.

Is this obvious? Think about a drawing of a hypergraph (Figure 4), where the hyperedges are blobs surrounding the vertices that they contain. The hyperedges are (literally[4]) sets, and finding a minimum hyperedge covering means finding the smallest set of sets so that all vertices belong to at least one set. This is literally[5] the set cover problem.

Theorem 34. *The hypergraph edge cover problem is* NP*-hard.*

Proof. Let $\langle U, S, a\rangle$ be an instance of the set cover problem, with $U = \{1, 2, \ldots, n\}$, $S = \{S_1, S_2, \ldots, S_m\}$, and k an integer. Construct an instance $\langle G, a\rangle$ the minimum hypergraph edge cover problem in the following way: let $G = (V, E)$, where $V = \{v_i : i \in U\}$ and $E = \{\{v_i : i \in S_i\} : S_i \in S\}$. This can clearly be done in time linear in the size of $\langle U, S, k\rangle$.

If the answer to $\langle U, S, a\rangle$ is "Yes", then there exists a set cover of size at most a for U. So there exists a set of at most a hyperedges (those corresponding to the sets in the set cover), such that every vertex is contained in at least one edge. So there exists a hyperedge cover of size at most a, and the answer to $\langle G, a\rangle$ is "Yes".

If the answer to $\langle G, a\rangle$ is "Yes", then there exists a hyperedge cover of size at most a, such that every vertex is contained in at least one edge. So there exists a set of sets in S (those corresponding to the hyperedges in the hyperedge cover) such that every element of the universe lies in at least one set. So the answer to $\langle U, S, a\rangle$ is "Yes". □

Now consider the restriction of the hypergraph edge cover problem where all hyperedges have the same bounded cardinality:

Problem 35 (k-uniform hypergraph edge cover problem)**.** *Given a k-uniform hypergraph $G = (V, E)$ and an integer a, does there exist a hyperedge cover of G with at most a edges?*

This is equivalent to the set cover problem where each set has the same size k. We have shown this problem for $k = 2$ has a polynomial-time algorithm (Theorem 11),

[4]Using the meaning of "literally" that means "actually true".

[5]Same thing.

but what about for more general k? Say, $k = 3$? What's so special about 2-uniform hypergraphs, anyway?

Homework 3. *Prove that the* 3*-uniform hypergraph edge cover problem is* NP*-hard (hint: use Theorem 25).*

And what about the hypergraph vertex cover problem:

Definition 36. *Given a hypergraph $G = (V, E)$, a subset $V' \subseteq V$ is a* vertex cover of G *provided for every edge $e \in E$, there exists a vertex $v \in V'$ such that $v \in e$.*

Problem 37 (Hypergraph vertex cover)**.** *Given a hypergraph $G = (V, E)$ and an integer a, does there exist vertex cover of G with size at most a?*

And what of the complexity of this problem? Observe that the hypergraph vertex cover problem is equivalent to the hitting set problem, which is equivalent to the set cover problem, which is equivalent to the hypergraph edge cover problem. That is, Problem 37 $\equiv$ Problem 29 $\equiv$ Problem 27 $\equiv$ Problem 32. So the hypergraph vertex cover and hypergraph edge cover problems are equivalent. Keen.

References

[1] R. Rizzi, A short proof of König's matching theorem, Journal of Graph Theory, 33(3), 138–139, 2000.