Lecture 8: Hamilton cycles, matchings, independent sets · 1MA020

Vilhelm Agdur[1]

[1] vilhelm.agdur@math.uu.se

15 November 2023

We begin by continuing to pursue consequences of the Ford-Fulkerson theorem, proving König's theorem.

König's theorem

In our last lecture, we proved the Ford-Fulkerson theorem relating minimum cuts and maximal flows, and used it to prove the Hall marriage theorem on matchings in bipartite graphs. We begin this lecture by proving another result we can derive from Ford-Fulkerson, namely König's theorem about vertex covers of bipartite graphs.

Definition 1. Let G be a finite simple graph. A *vertex cover* of G is a subset $S \subseteq V$ such that every edge has an endpoint in S. The *covering number* of G, denoted $\beta(G)$, is the minimum size of any vertex cover of G.

Example 2. A star graph has covering number 1, while a complete graph on n vertices has covering number $n-1$. A cycle graph on $2n$ vertices has covering number n.

Theorem 3 (König's theorem)**.** *Let G be a bipartite graph. Then the maximum cardinality of a matching on G equals $\beta(G)$, the minimum cardinality of a vertex cover of G.*

Proof. Let M be a maximal matching in G, and like in the proof of the marriage theorem, construct a flow network G' from G. As we saw in the proof of that theorem, this matching M corresponds to a maximal flow in G' of value $|M|$. By Ford-Fulkerson, this means there is a minimum cut S, T on G' of capacity $|M|$.

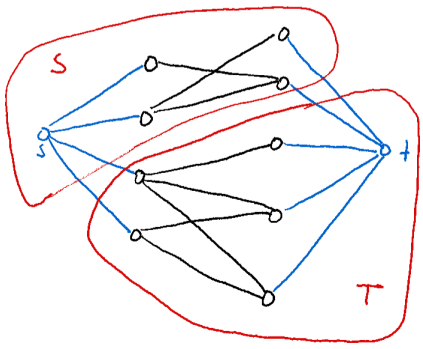

Figure 1: A non-trivial minimal s-t-cut in a flow network created from a bipartite graph.

Now, given this cut, let us construct a vertex cover. In particular, we let

$$C = (A \cap T) \cup (B \cap S).$$

That this C is a vertex cover of G is precisely the statement that there is no edge $a \to b$ with $a \in A \cap S$ and $b \in B \cap T$. This, however, is something we already saw is true for all minimum cuts in the previous proof – because if there were such an edge, it would contribute ∞ to the capacity of the cut. So C is indeed a vertex cover, and it is easy to convince oneself, looking at Figure 1, that $|C| = c(S,T) = |f| = |M|$. So what we have seen is that the size of a maximal matching upper bounds the minimum vertex cover size, that is, $\beta(G) \leq |M|$.

The other direction of the inequality in fact holds for all graphs, not just bipartite graphs – every edge of a matching has to be covered by a vertex in the cover, and no vertex can cover more than one edge of the matching at a time. So any vertex cover has to have at least as many vertices in it as a maximal matching has edges. □

Hamilton cycles

We studied, at the very beginning of the course, the notion of Eulerian circuits. The twin notion of Hamilton cycles appeared in our exercise session, but less us quickly repeat what we said about those.

Definition 4. Let $G = (V, E)$ be a finite simple graph. A *Hamilton cycle* in G is a cycle that visits every vertex of G.[2] If G admits a Hamilton cycle, we call it *Hamiltonian*.

[2] Recall that a *cycle* by definition is a walk that starts and ends at the same vertex, and does not reuse any vertices other than the one it started with.

Unlike for Eulerianity, there is no simple condition to determine whether a graph is Hamiltonian. In fact, computing whether a graph is Hamiltonian is an NP-Complete problem.

Remark 5. Given a Hamiltonian cycle C in G, we can easily construct a perfect matching of G by just picking every second edge in the cycle. We can of course not in general turn a perfect matching into a Hamilton cycle, so Hamiltonicity is a stronger condition than having a perfect matching.

That Hamiltonicity is an NP-Complete problem means we probably will never have any necessary and sufficient conditions for it. However, we can still prove results of the form "if condition so-and-so holds, G is Hamiltonian". Let us prove the most famous such result.[3]

[3] It is named after a different Dirac than the one of quantum mechanics fame.

Theorem 6 (Dirac's theorem)**.** *Let $G = (V, E)$ be a simple graph on $n \geq 3$ vertices, such that every vertex has degree at least $\frac{n}{2}$. Then G is Hamiltonian.*

Proof. Assume G is a graph satisfying the conditions of the theorem. Let us begin by observing that this G cannot be disconnected – if it were, the vertices in the smallest component would have fewer than half of the other vertices to connect to, so they would have too low degree.

Now, consider a path P of maximum length in G, say

$$P = v_0\, e_1\, v_1\, \ldots\, v_{k-1}\, e_k\, v_k.$$

Since this path is maximal, it contains all neighbours of v_0 and of v_k, since otherwise it could be extended. By our degree condition, there are (counting with multiplicity) at least n such neighbours – and so by essentially the pigeonhole principle, there must be an edge $e_i = \{v_{i-1}, v_i\}$ on our path such that v_i is a neighbour of v_0 and v_{i-1} is a neighbour of v_k, as in Figure 2.

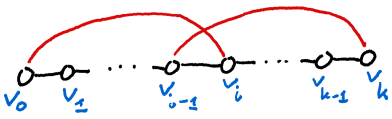

Figure 2: The path P along with the configuration of two edges that must exist by the pigeonhole principle.

Now we notice that this in fact lets us turn our path into a cycle – we start at v_0, walk until v_{i-1}, then use the edge to v_k, walk backwards to v_i, and finally use the edge to v_0. This gives us a cycle C.

It remains to see that this C is in fact a Hamilton cycle. So, suppose for contradiction that it is not, so that there is some vertex v not on the cycle. Since G is connected, we can in fact assume that this vertex is adjacent to some vertex of C. This, however, means we can create a longer path than P, by starting at v, walking onto C, and then following the cycle, as indicated in Figure 3. However, we assumed P was a maximal path, so we have a contradiction. So C must be a Hamilton cycle. □

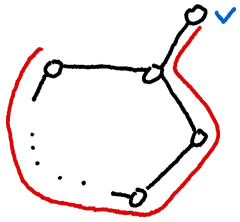

Figure 3: A longer path than P created from the cycle C, assuming it was not a Hamilton cycle.

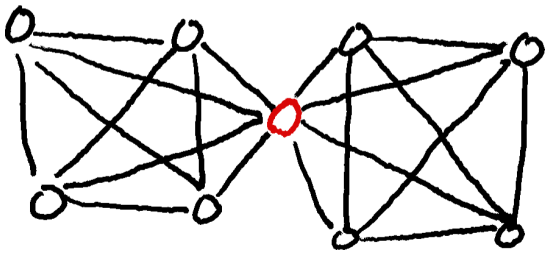

Figure 4: A graph used to show that the condition in Dirac's theorem is the best possible uniform lower bound.

Remark 7. This result is in fact the best possible result with a uniform lower bound on the degrees of the vertices – if we had picked any $k(n)$ less than $\frac{n}{2}$, there would be a counterexample. For example, consider the graph in Figure 4, where we have taken $k = \left\lfloor \frac{n-1}{2} \right\rfloor$ and glued together two copies of K_{k-1} by identifying two vertices. This graph has minimum degree k, but can of course not contain a Hamilton cycle since such a cycle would have to visit the glued-together vertex twice.

Let us consider one particular example of a graph family that is Hamiltonian.

Definition 8. The *d-dimensional cube graph* has as its vertex set $\{0,1\}^d$, that is, the set of binary strings of length d. Two vertices are neighbours if the corresponding binary strings differ in only one position. The first few are illustrated in Figure 5.

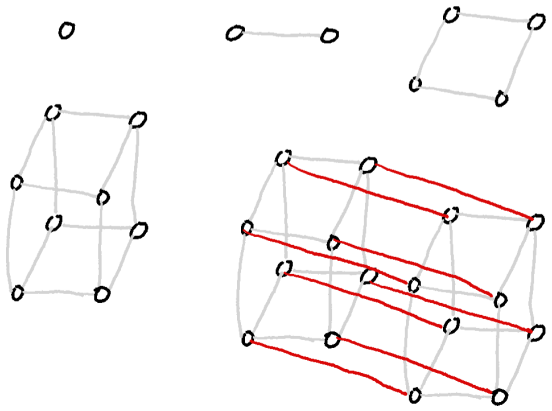

Figure 5: The d-dimensional cubes for $d = 0, 1, 2, 4$. In the four-dimensional case we have highlighted some edges in red, to illustrate the inductive structure of the graph family.

Theorem 9. *The d-dimensional cube is Hamiltonian for all d.*

Exercise 1. Prove this. □

Remark 10. Notice that Theorem 9 gives us an ordering of the length-d bit strings suh that going from one to the next always only requires us to change one of the bits. This ordering is known as the *Gray code*, and it was originally invented to help in converting from analog to digital signals for colour television in the fifties.

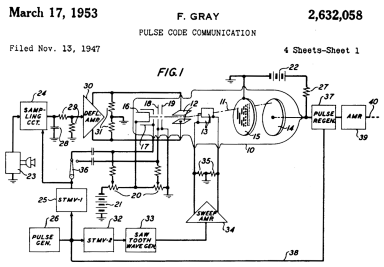

Figure 6: The original patent for the device using a Gray code – this supposedly happens in the thingy labelled by 15?

Independent sets

Another notion we encountered during the exercises is that of an *independent set*. Let us repeat the definition again here.

Definition 11. An *independent set* in a graph $G = (V, E)$ is a subset $I \subseteq V$ of the vertices such that no two elements of I are adjacent.

We saw that this concept in fact connects to the concept of matchings if we consider the line graph of a graph.

Definition 12. Given a graph $G = (V, E)$, the *line graph* $L(G)$ of G has as its vertices the *edges* of G, and there is an edge between e and e' whenever they share an endpoint in G. This is illustrated in Figure 7.

Remark 13. A matching on a graph G is precisely the same thing as an independent set in its line graph $L(G)$.

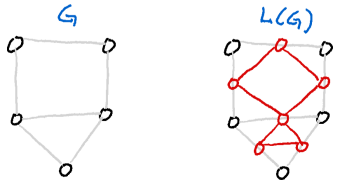

Figure 7: A graph G on the left, and on the right we have its line graph $L(G)$ drawn on top of G in red.

In the exercises, we sketched out an argument as to why finding an independent set is NP-Complete. Let us now actually go through the details of the proof.

Our method will be to show that we can encode an instance of 3-SAT as a problem of finding an independent set of a certain size in a graph. This will prove that independent set is NP-Complete, because the Cook-Levin theorem tells us that 3-SAT is.

So first, let us define what an instance of 3-SAT is.

Definition 14. An instance of 3-SAT is the problem of determining whether a given logical formula consisting of a conjunction of clauses of length 3 is satisfiable. So, we have some set of logical variables $x_1, x_2, \ldots, x_n$. A literal is x_i or $\neg x_i$ for some $i \in [n]$, and a clause of length 3 is a disjunction of three literals, e.g. $x_4 \vee \neg x_{10} \vee x_3$. We then take the conjunction of some number of such clauses, getting a formula like

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_2 \vee x_4)\,,$$

and what we want to determine is whether we can assign each variable a true or false value in such a way that the entire formula is true.

Theorem 15. *The problem of determining whether a graph contains an independent set of size k is NP-Complete.*

Proof. We prove this by associating to each instance of 3-SAT an instance of the independent set problem, and showing that the 3-SAT formula is satisfiable if and only if there is an independent set of the right size in the graph.

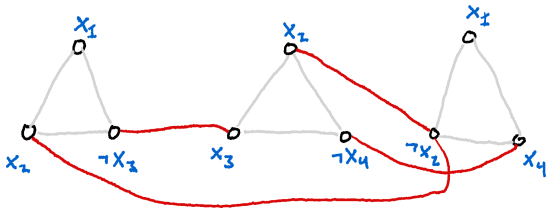

Figure 8: The graph corresponding to the 3-SAT formula $(x_1 \vee x_2 \vee \neg x_3) \wedge (x_2 \vee x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_2 \vee x_4)$.

In particular, given a formula, we create a graph as follows: For each clause, draw a triangle, and label its vertices with the literals of the clause. Then, for each literal, draw an edge between its vertex and any vertex labelled with the negation of the literal. An example of this process is given in Figure 8.

We claim that this graph has an independent set of size equal to the number of clauses, i.e. the number of triangles we drew, if and only if the corresponding formula is satisfiable. If there is an independent set of that size, it is clear that it must contain one vertex from each

triangle, which corresponds to picking one literal from each clause. Assigning the variables so that each of these literals becomes true gives a satisfying assignment.

In the other direction, any satisfying assignment must of course make at least one literal per clause true. Picking a vertex corresponding to one of those literals from each clause gives us an independent set of the desired size. □

Exercises