CS 591-06, Lecture 3

Jared Saia
University of New Mexico

Class Outline

- Review
- Proof of Hall's Theorem
- In-class Exercise
- Open Problems

Factoring a graph

Let G be a k-regular graph where k is an even number. We factor G as follows:

1. Construct an Euler-tour of G.
2. Orient the edges according to the direction of the tour. That is, if the tour enters v on edge e_1 and leaves on edge e_2, then e_1 is an in-edge to v and e_2 is an out-edge. Thus we have $d_{\text{in}} = d_{\text{out}} = k$.
3. Set up a bipartite matching problem, B_G, with a representative of each vertex in the graph on both sides. Add in all directed edges going from left to right. Note that each edge is represented in the matching problem exactly once.
4. Find a matching (which is guaranteed to exist by Hall's Theorem). The matched edges induce a 2-factor of the original graph. Remove these edges from B_G and repeat this step until there are no edges left.

Hall's Theorem

Statement of Hall's Marriage Theorem:

Let $G = (L, R)$ be a bipartite graph. Then:

- There is a perfect matching for G

if and only if

- $\forall L' \subseteq L,\ |L'| \leq |N(L')|$

where $|X|$ is the number of vertices in the set X and $N(X)$ is the set of neighbors of the set X.

Corollary

Last time we proved that for bipartite $G = (L, R)$,:

if

- G is k-regular,

then

- $\forall L' \subseteq L,\ |L'| \leq |N(L')|$

Proof: Consider any set $L' \subseteq L$, then

- number of edges between L and $N(L) = k * |L|$ (since k edges from each vertex in L)
- number of edges between L and $N(L) \leq k * |N(L)|$ (since k edges from each vertex in $N(L)$)
- Implies $|L'| \leq |N(L')|$

Proof of Hall's Theorem

We will now prove Hall's Theorem:

Let $G = (L, R)$ be a bipartite graph. Then

- There is a perfect matching for G

if and only if

- $\forall L' \subseteq L,\ |L'| \leq |N(L')|$

where $|X|$ is the number of vertices in the set X and $N(X)$ is the set of neighbors of the set X.

Proof of Hall's Theorem

Easy direction:

If

- there is a perfect matching for G,

then

- $\forall L' \subseteq L,\ |L'| \leq |N(L')|$

Easy Direction

if

- there is a perfect matching for G,

then

- $\forall L' \subseteq L,\ |L'| \leq |N(L')|$

Proof:

Assume there is a matching and that there is some set L' such that $|L'| > |N(L')|$. We know that every vertex in L' must be matched in the matching, hence we know that $N(L')$ contains at least $|L'|$ neighbors. This is a contradiction since we assumed that $|L'| > |N(L')|$.

Hard Direction

If

- $\forall L' \subseteq L,\ |L'| \leq |N(L')|$

then

- there is a perfect matching for G,

Proof of Hall's

To show: If for some graph $G = (L, R)$, $\forall L' \subseteq L,\ |L'| \leq |N(L')|$ then there is a perfect matching for G.

We will say that a graph is "slack" if it's the case that $\forall L' \subset L$, $|L'| \leq |N(L')| - 1$

Proof:

We will prove this by strong induction on the number of vertices in L.

Base case: If there is a single vertex the statement holds trivially.

Inductive Step

Inductive Step:

Case 1: The graph is slack.

Let l be some vertex in L, we know that l has some neighbor in R, let r be one of those neighbors. Let G' be the graph G with the vertices l and r removed. Since G is slack, G' satisfies Hall's condition (any subset of vertices on the left side of G' has a number of neighbors at least as large as the set). Thus, by the inductive hypothesis, there is a perfect matching for G'. If we add to this matching the edge between l and r, we then have a perfect matching for G.

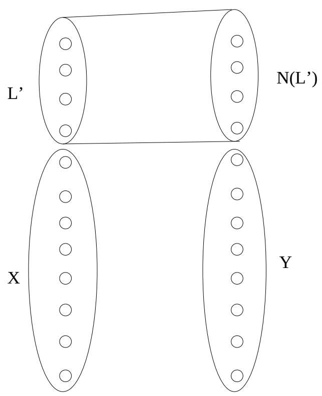
Proof of Hall's

Case 2: The graph is not slack.

In this case, there is some subset, L', of L such that $|L'| = |N(L')|$ (and $L' \neq L$). Since $|L'| < |L|$, and Hall's condition holds for the vertices in this set, we can use the inductive hypothesis to say there is a perfect matching for the vertices of L'.

Now let $X = L - L'$, and let $Y = R - N(L')$. We will now show that the condition for Hall's theorem holds for the graph (X, Y). Consider any $X' \subseteq X$. We will show that the set of neighbors of X' which are in Y is at least as large as the number of vertices in X'.

Proof of Hall's



Proof of Hall's

By Hall's, we know that for any X' which is a subset of X:

$$|X' \cup L'| \leq |N(X' \cup L')|$$

and we know that:

$$N(X' \cup L') = N(L') \cup (N(X') \cap Y)$$

We also know by assumption (since the graph was not slack) that:

$$|L'| = |N(L')|$$

Proof of Hall's

For any $X' \subseteq X$, the following holds:

$$\begin{aligned}
|X'| + |L'| &= |X' \cup L'| &(1)\\
&\leq |N(X' \cup L')| &(2)\\
&= ||N(L') \cup (N(X') \cap Y)| &(3)\\
&= |N(L')| + |(N(X') \cap Y)| &(4)\\
&= |L'| + |(N(X') \cap Y)| &(5)
\end{aligned}$$

Cancelling $|L'|$ from the first and last equation then gives that $|X'| \leq |N(X') \cap Y|$. So Hall's condition holds for the graph (X, Y), which means by the i.h. that there is a perfect matching of X which uses only vertices in Y. Combining this with the perfect matching of L' which uses vertices in $N(L')$ gives a perfect matching of all vertices in L.

Matching Algorithm

For a bipartite graph G with edge set E:

Defn: A *chain* is a sequence of vertices and edges $v_1, e_1, v_2, e_2, ..., e_l, v_l$ such that the edge $e_i = (v_{i-1}, v_i)$.

Let M be some (possibly non-maximum) matching on G.

Defn: A chain $v_1, e_1, v_2, e_2, \ldots, e_l, v_l$ is called *M-alternating* if the edges $e_2, e_4, \ldots$ are in M.

Defn: A chain $v_1, e_1, v_2, e_2, \ldots, e_l, v_l$ is called *M-augmenting* if it is M-alternating and the vertices v_1 and v_l are not incident to any edges in M.

Theorem: A Matching M is maximum for G if and only if the graph contains no M-augmenting chains.

Making a graph Δ'-regular

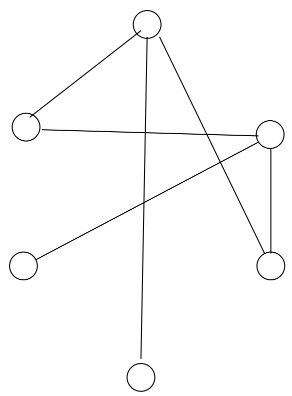
The following algorithm takes a graph with degree Δ and adds edges to it to get a graph of degree $\Delta' = 2\lceil\Delta/2\rceil$

Algorithm *Regular*

1. While there exists a vertex with degree less than $\Delta' - 1$, add a self loop to that vertex.
2. While there exist two distinct vertices of degree $\Delta' - 1$, add an arbitrarily directed edge between them.

Excercise

Make the following graph 4 regular as specified in Algorithm *Regular*



Direct Migration

Input: A demand graph G with degree Δ

Output: A migration plan taking $3/2\lceil\Delta/2\rceil$ time steps.

Algorithm *Direct Migration*:

1. Make G $2\lceil\Delta/2\rceil$ regular using Algorithm *Regular*
2. Factor G into $\lceil\Delta/2\rceil$ 2-factors using Algorithm *2-factor*
3. Take up to three time steps to send the objects associated with each 2-factor.

Indirect Migration

Input: A demand graph G with degree Δ and n vertices.

Output: A migration plan using $n/3$ bypass nodes and taking $2\lceil\Delta/2\rceil$ time steps.

Algorithm *Indirect Migration*:

1. Make G $2\lceil\Delta/2\rceil$ regular using Algorithm *Regular*
2. Factor G into $\lceil\Delta/2\rceil$ 2-factors using Algorithm *2-factor*
3. Use $n/3$ bypass nodes to take *two* time steps to send the objects associated with each 2-factor.