Maximum Matching Problem: A Randomized, Algebraic Approach

Daniel Alabi

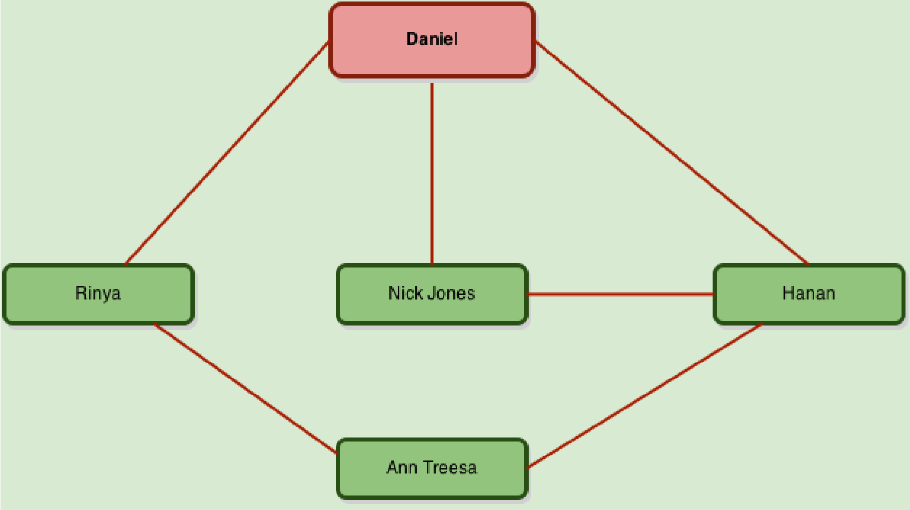
May 15, 2014

Outline

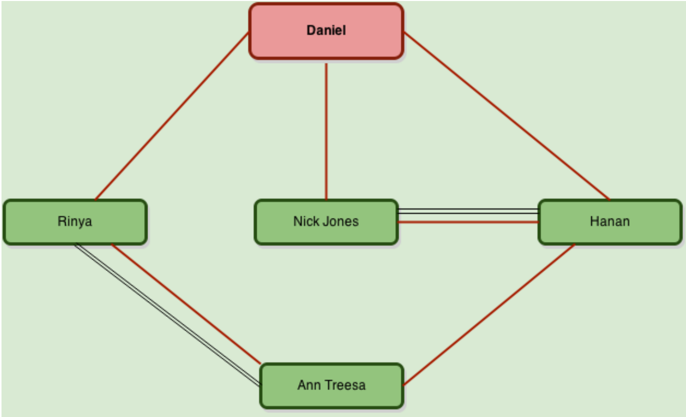
1. Maximum Matching
   - Toy Problem: Selecting Doubles for Badminton
   - Problem Formulation
   - A Combinatorial Approach

2. An Algebraic Approach
   - Detecting the Presence of a Perfect Matching
   - Obtaining a Perfect Matching
   - Rabin-Vazirani Theorem
   - Obtaining a Maximum Matching

Selecting Doubles for Badminton Tournament

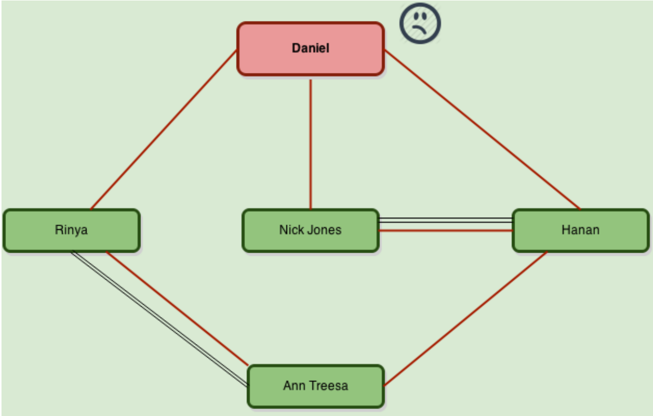


Selecting Doubles for Badminton Tournament



$$M = \{(Rinya, AnnTreesa), (NickJones, Hanan)\}$$

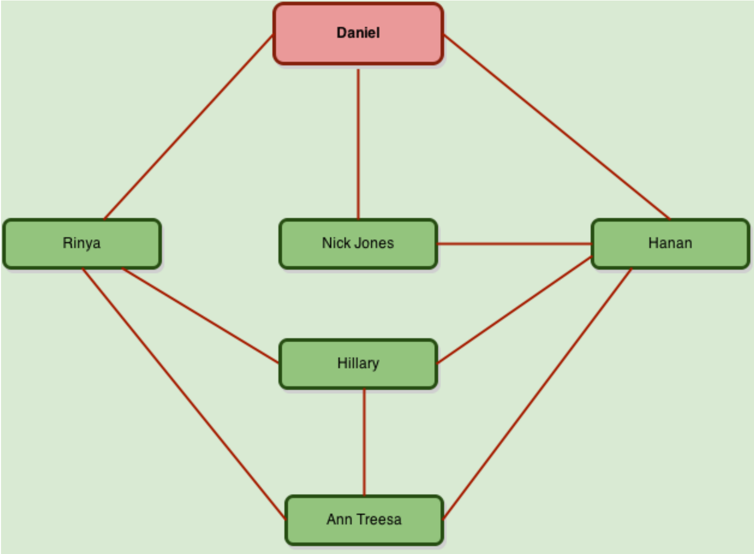
Selecting Doubles for Badminton Tournament



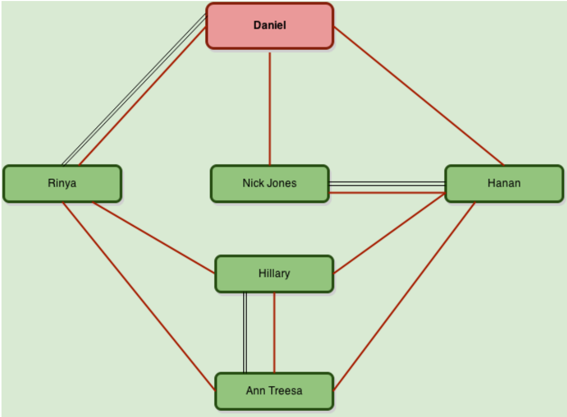
$$M = \{(Rinya, AnnTreesa), (NickJones, Hanan)\}$$

Maximum Matching of size 2

Selecting Doubles for Badminton Tournament

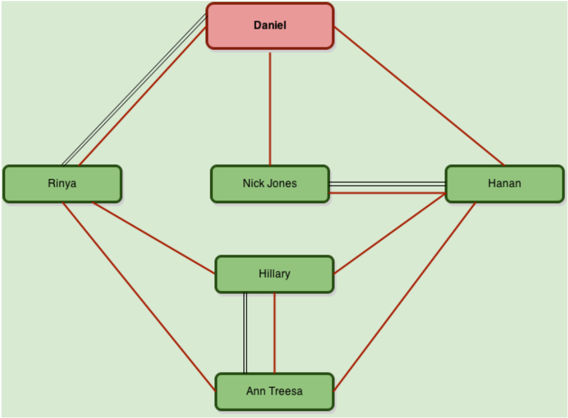


Selecting Doubles for Badminton Tournament



$$M = \{(Rinya, AnnTreesa), (NickJones, Hanan), (Hillary, AnnTreesa)\}$$

Selecting Doubles for Badminton Tournament



$$M = \{(Daniel, Rinya), (NickJones, Hanan), (Hillary, AnnTreesa)\}$$

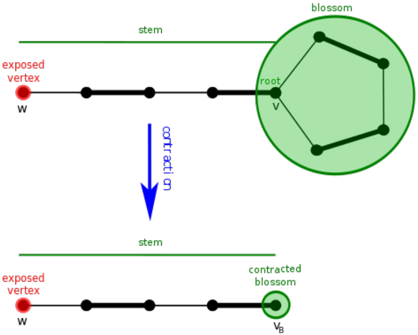
Maximum (Perfect) Matching of size 3

Formal Problem Statement

Given an undirected graph $G = (V, E)$, find a set M such that each vertex in V is incident to at most one edge in M and $|M|$ is maximized.

Blossom Algorithm [Edmonds, 1965]

Uses blossoms, contractions, and augmenting paths.



It's pretty fast! A modified version of the Blossom Algorithm by Micali and Vazirani runs in $O(|E||V|^{1/2})$.


I'M SO FAST BUT...
CODE ME NOT
imgflip.com

Outline

Two-Step Process:

1.

Two-Step Process:

1. Characterize a graph algebraically using a Tutte matrix T

Two-Step Process:

1. Characterize a graph algebraically using a Tutte matrix T
2. Obtain T^{-1} and use to get edges in a Maximum Matching

Tutte Matrix

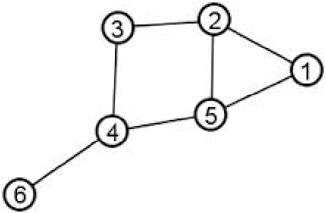
A skew-symmetric representation of a graph with indeterminates (formal variables) as entries.

Tutte Matrix

A skew-symmetric representation of a graph with indeterminates (formal variables) as entries.

$$T_{i,j} = \begin{cases} x_{ij} & : \{i,j\} \in E \text{ and } i > j \\ -x_{ji} & : \{i,j\} \in E \text{ and } i < j \\ 0 & : \{i,j\} \notin E \end{cases}$$

Tutte Matrix



$$
\begin{pmatrix}
0 & x_{12} & 0 & 0 & x_{15} & 0 \\
-x_{12} & 0 & x_{23} & 0 & x_{25} & 0 \\
0 & -x_{23} & 0 & x_{34} & 0 & 0 \\
0 & 0 & -x_{34} & 0 & x_{45} & x_{46} \\
-x_{15} & -x_{25} & 0 & -x_{45} & 0 & 0 \\
0 & 0 & 0 & -x_{46} & 0 & 0
\end{pmatrix}
$$

Tutte's Theorem

Tutte's Theorem

Let G be a graph and let T be its Tutte matrix.
Then, $det(T) \neq 0 \iff$ there exists a perfect matching in G

Awesome! So we just have to calculate $det(T)$

Tutte's Theorem

Let G be a graph and let T be its Tutte matrix.
Then, $det(T) \neq 0 \iff$ there exists a perfect matching in G

Awesome! So we just have to determine $det(T)$. But...
calculating $det(T)$ is expensive.

Tutte's Theorem

Let G be a graph and let T be its Tutte matrix.
Then, $det(T) \neq 0 \iff$ there exists a perfect matching in G

Awesome! So we just have to determine $det(T)$. But...
calculating $det(T)$ is expensive.

$$det(T) = \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \prod_i A_{i\sigma(i)}$$

Tutte's Theorem

Let G be a graph and let T be its Tutte matrix.
Then, $det(T) \not\equiv 0 \iff$ there exists a perfect matching in G

Awesome! So we just have to determine $det(T)$. But...
calculating $det(T)$ is expensive.

$$det(T) = \sum_{\sigma \in S_n} \text{sgn}(\sigma) \prod_i A_{i\sigma(i)}$$

$det(T)$ is a polynomial in the indeterminates in the matrix. $det(T)$ could have a superpolynomial number of terms. Very INEFFICIENT.


Randomization

Schwartz-Zippel Lemma (Restated)

Suppose $det(T) \not\equiv 0$; set each variable in T to an element in $\{1, \ldots, n^2\}$ uniformly at random.
Then, $\Pr[det(T) = 0] \leq 1/n$.

Schwartz-Zippel Lemma (Restated)

Suppose $det(T) \not\equiv 0$; set each variable in T to an element in $\{1, \ldots, n^2\}$ uniformly at random.
Then, $\Pr[det(T) = 0] \leq 1/n$.

Now, our algorithm is randomized (Monte Carlo).

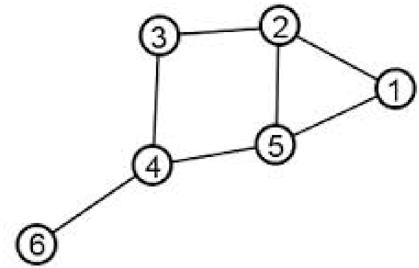
Algorithm for Detecting a Perfect Matching

Schwartz-Zippel Lemma (Restated)

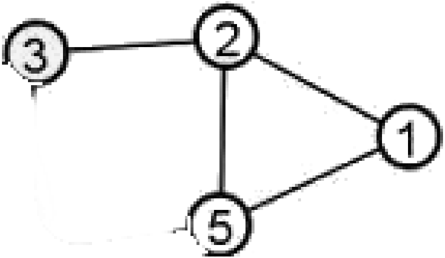
Suppose $det(T) \not\equiv 0$; set each variable in T to an element in $\{1, \ldots, n^2\}$ uniformly at random.
Then, $\Pr[det(T) = 0] \leq 1/n$.

We can now detect the presence of a perfect matching in a graph pretty quickly ($O(n^{2.373})$).

$G \Rightarrow G - \{i, j\}$



G



$G - \{4, 6\}$

Rabin, Vazirani

Let $G = (V, E)$ be an undirected simple graph with a perfect matching and T be its associated Tutte matrix. Then $(T^{-1})_{i,j} \neq 0$ if and only if $G - \{i, j\}$ has a perfect matching.

Rabin, Vazirani

Let $G = (V, E)$ be an undirected simple graph with a perfect matching and T be its associated Tutte matrix. Then $(T^{-1})_{i,j} \neq 0$ if and only if $G - \{i, j\}$ has a perfect matching.

Adjoint Formula

For any non-singular $n \times n$ matrix A, we have $(A^{-1})_{i,j} = \frac{(adjA)_{i,j}}{det(A)}$, where the value of $(adjA)_{i,j}$ (called the *adjoint*) is the determinant of A after the ith row and jth column have both been deleted.

Rabin, Vazirani

Let $G = (V, E)$ be an undirected simple graph with a perfect matching and T be its associated Tutte matrix. Then $(T^{-1})_{i,j} \neq 0$ if and only if $G - \{i, j\}$ has a perfect matching.

Proof:

Rabin, Vazirani

Let $G = (V, E)$ be an undirected simple graph with a perfect matching and T be its associated Tutte matrix. Then $(T^{-1})_{i,j} \neq 0$ if and only if $G - \{i, j\}$ has a perfect matching.

Proof:

$(T^{-1})_{i,j} \neq 0 \Rightarrow G - \{i, j\}$ **has a perfect matching.**

Rabin, Vazirani

Let $G = (V, E)$ be an undirected simple graph with a perfect matching and T be its associated Tutte matrix. Then $(T^{-1})_{i,j} \neq 0$ if and only if $G - \{i, j\}$ has a perfect matching.

Proof:

$(T^{-1})_{i,j} \neq 0 \Rightarrow G - \{i, j\}$ **has a perfect matching.**

- Suppose G has a perfect matching and $(T^{-1})_{i,j} \neq 0$.

Rabin, Vazirani

Let $G = (V, E)$ be an undirected simple graph with a perfect matching and T be its associated Tutte matrix. Then $(T^{-1})_{i,j} \neq 0$ if and only if $G - \{i, j\}$ has a perfect matching.

Proof:

$(T^{-1})_{i,j} \neq 0 \Rightarrow G - \{i, j\}$ **has a perfect matching.**

Suppose G has a perfect matching and $(T^{-1})_{i,j} \neq 0$.

– Then by the adjoint formula $(adj T)_{i,j} \neq 0$ which implies that $G - \{i, j\}$ has a perfect matching.

Rabin, Vazirani

Let $G = (V, E)$ be an undirected simple graph with a perfect matching and T be its associated Tutte matrix. Then $(T^{-1})_{i,j} \neq 0$ if and only if $G - \{i, j\}$ has a perfect matching.

Proof:

$G - \{i, j\}$ **has a perfect matching** $\Rightarrow (T^{-1})_{i,j} \neq 0$.

Rabin, Vazirani

Let $G = (V, E)$ be an undirected simple graph with a perfect matching and T be its associated Tutte matrix. Then $(T^{-1})_{i,j} \neq 0$ if and only if $G - \{i, j\}$ has a perfect matching.

Proof:

$G - \{i, j\}$ **has a perfect matching** $\Rightarrow (T^{-1})_{i,j} \neq 0$**.**

Suppose $G - \{i, j\}$ has a perfect matching M.

Then if vertices i and j are added back to $G - \{i, j\}$, (i, j) will not be incident on any edge in M.

– So $M' = M \cup \{(i, j)\}$ will be a perfect matching for G.

Proof:

$G - \{i, j\}$ **has a perfect matching** $\Rightarrow (T^{-1})_{i,j} \neq 0$**.**

Suppose $G - \{i, j\}$ has a perfect matching M.

Then if vertices i and j are added back to $G - \{i, j\}$, (i, j) will not be incident on any edge in M.

So $M' = M \cup \{(i, j)\}$ will be a perfect matching for G.

– Since G has a perfect matching, T^{-1} exists and by the adjoint formula, $(T^{-1})_{i,j} = \frac{(adjT)_{i,j}}{det(T)}$.

Proof:

$G - \{i, j\}$ **has a perfect matching** $\Rightarrow (T^{-1})_{i,j} \neq 0$**.**

Suppose $G - \{i, j\}$ has a perfect matching M.

Then if vertices i and j are added back to $G - \{i, j\}$, (i, j) will not be incident on any edge in M.

So $M' = M \cup \{(i, j)\}$ will be a perfect matching for G.

Since G has a perfect matching, T^{-1} exists and by the adjoint formula, $(T^{-1})_{i,j} = \frac{(adjT)_{i,j}}{det(T)}$.

G has a perfect matching M'; so $\det(T) \neq 0$.

– Since $G - \{i, j\}$ has a perfect matching M, $(adjT)_{i,j} \neq 0$.

Proof:

$G - \{i,j\}$ **has a perfect matching** $\Rightarrow (T^{-1})_{i,j} \neq 0$**.**

Suppose $G - \{i,j\}$ has a perfect matching M.

Then if vertices i and j are added back to $G - \{i,j\}$, (i,j) will not be incident on any edge in M.

So $M' = M \cup \{(i,j)\}$ will be a perfect matching for G.

Since G has a perfect matching, T^{-1} exists and by the adjoint formula, $(T^{-1})_{i,j} = \frac{(adjT)_{i,j}}{det(T)}$.

G has a perfect matching M'; so $\det(T) \neq 0$.

Since $G - \{(i,j)\}$ has a perfect matching M, $(adjT)_{i,j} \neq 0$.

– Thus, $(T^{-1})_{i,j} \neq 0$

Algorithm for Finding a Perfect Matching

Algorithm *FindPerfectMatching*(G)

$M = \emptyset$
while G is not empty
    $T =$ Tutte matrix of G
    compute T^{-1}
    Find (i, j) such that $\{i, j\} \in G.E$ and $(T^{-1})_{i,j} \neq 0$
    $M = M \cup \{(i, j)\}$
    $G = G - \{i, j\}$
return M

Theorem [Lovasz]

Let T be the Tutte matrix of a graph G and let m be the size of the maximum matching in G, then $\text{rank(T)} = 2m$.

Algorithm for obtaining a Maximum Matching

Algorithm $FindMaxMatching(G)$

1. $G' = G$ with $n - 2m$ new "dummy" vertices added to G
2. Connect each dummy vertex to all the vertices in G
3. $N = FindPerfectMatching(G')$
4. $M = N$ with dummy vertices and edges removed
5. **return** M

Takes $O(n^3 \log^2 n)$ time. Not bad.

Improvements to the Rabin-Vazirani Algorithm

1. Use Gaussian elimination to reduce the amount of matrix inversions you do.

Improvements to the Rabin-Vazirani Algorithm

Algorithm *FindPerfectMatching*(G)

$M = \emptyset$
while G is not empty
    $T =$ Tutte matrix of G
    compute T^{-1}
    Find (i, j) such that $\{i, j\} \in G.E$ and $(T^{-1})_{i,j} \neq 0$
    $M = M \cup \{(i, j)\}$
    $G = G - \{i, j\}$
return M

Improvements to the Rabin-Vazirani Algorithm

1. Use Gaussian elimination to reduce the amount of matrix inversions you do.
2. The algorithm is parallelizable (it belongs in NC).

Improvements to the Rabin-Vazirani Algorithm

1. Use Gaussian elimination to reduce the amount of matrix inversions you do.
2. The algorithm is easily parallelizable (it belongs in NC).
3. You can also make the algorithm Las Vegas with expected running time $O(n^4 \log^2 n)$.

I implemented Rabin-Varizani's algorithm. Check it out here: http://github.com/alabid/math_comps_code

References I

Rabin, Vazirani
Maximum Matchings in General Graphs through Randomization
Journal of Algorithms 10, 551-567, 1989.

Peterson, Loui
The General Maximum Matching Algorithm of Micali and Vazirani
Algorithmica, 511-533, 1988

Ivan, Virza, Yuen
Algebraic Algorithms for Matching

Thanks for listening! W(h)oosah!

Thanks for listening! W(h)oosah!
Questions?