Topics on Computing and Mathematical Sciences I Graph Theory (3) Trees and Matchings I

Yoshio Okamoto

Tokyo Institute of Technology

April 23, 2008

Today's contents

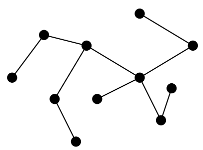
1. Trees
   - Trees, forests
   - Characterizations
   - Spanning trees, exchangeability
2. Matchings
   - Matchings
   - Maximum matchings, alternating paths, weak duality

Forests and trees

Definition (Forest)
A forest is a graph containing no cycle

Definition (Tree)
A tree is a connected graph containing no cycle
(A tree is a connected forest)

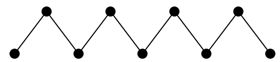
Example of a tree



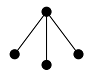
Special trees

Examples of trees

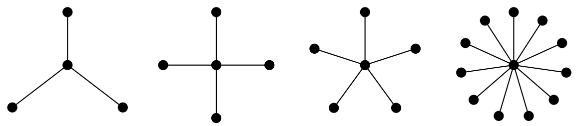
- Paths



- Claws ($\simeq K_{1,3}$)



- Stars ($\simeq K_{1,t}$)

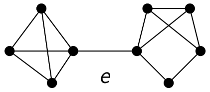


Cut edge

$G = (V, E)$ a graph

Definition (Cut edge)

An edge $e \in E$ is a cut edge of G if the deletion of e from G increases the number of connected components



Proposition 3.1

An edge $e \in E$ is a cut edge of $G \iff e$ does not belong to any cycle

Corollary 3.2

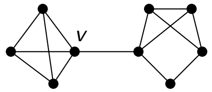
In a forest, every edge is a cut edge

Cut vertex

$G = (V, E)$ a graph

Definition (Cut vertex)

A vertex $v \in V$ is a cut vertex of G if the deletion of v from G increases the number of connected components



Is it true??

In a forest, every vertex is a cut vertex

Leaves in a tree

Definition (Leaf)

A leaf of a graph is a vertex of degree one

Lemma 3.3 (At least two leaves in a tree)

Every n-vertex tree has at least two leaves, when $n \geq 2$

Proof idea.

Extremality: Consider a maximal path! □

Lemma 3.4 (Deleting a leaf from a tree leaves a tree)

$T = (V, E)$ a tree, $v \in V$ a leaf of $T \implies T - v$ a tree

Characterizations of trees

Theorem 3.5 (Characterizations of trees)

$G = (V, E)$ an n-vertex graph ($n \geq 1$); The following are equivalent

1. G is connected and has no cycle (i.e., G is a tree)
2. G is connected and has $n-1$ edges
3. G has $n-1$ edges and no cycle
4. G has exactly one u, v-path for each $u, v \in V$

Proof idea.

[(1)⇒(2)] Induction on n, with Lem 3.3
[(2)⇒(3)] Delete edges from cycles of G, use Prop 1.7
[(3)⇒(1)] Use (1)⇒(2) for each connected component
[(1)⇒(4)] By contradiction
[(4)⇒(1)] By contradiction □

Spanning subgraphs and spanning trees

$G = (V, E)$ a graph; $H = (W, F)$ a subgraph of G

Definition (Spanning subgraph)

H spans G if $W = V$; H is a spanning subgraph of G

Remark:

- Spanning path $\equiv$ Hamiltonian path
- Spanning cycle $\equiv$ Hamiltonian cycle

Proposition 3.6 (A connected graph contains a spanning tree)

G connected $\iff$ G contains a spanning tree

Proof idea.

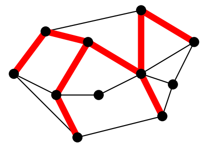
Repeat deleting edges from cycles, resulting in a spanning tree □

Repairing your spanning tree

If G is connected, it is enough for you to maintain a spanning tree T for wandering around G

Question

If an edge of T is damaged by the evil, is it possible to repair the spanning tree by adding another edge?



Exchangeability of spanning trees (1)

G a connected graph, T, T' spanning trees of G

Proposition 3.7 (Exchangeability I)

$\forall\, e \in E(T) \setminus E(T'), \exists\, e' \in E(T') \setminus E(T)$:
$T - e + e'$ is a spanning tree of G

Exchangeability of spanning trees (2)

G a connected graph, T, T' spanning trees of G

Proposition 3.8 (Exchangeability II)

$\forall\, e \in E(T) \setminus E(T'), \exists\, e' \in E(T') \setminus E(T)$:
$T' + e - e'$ is a spanning tree of G

Lemma 3.9 (Fundamental cycles)

Adding one edge to a tree forms exactly one cycle

Exchangeability of spanning trees (3)

G a connected graph, T, T' spanning trees of G

Exercise (Simultaneous exchangeability)

$\forall\, e \in E(T) \setminus E(T')$, $\exists\, e' \in E(T') \setminus E(T)$:
$T{-}e{+}e'$ and $T'{+}e{-}e'$ are spanning trees of G

Computing a minimum-cost spanning tree (1)

Problem MINIMUM COST SPANNING TREE

Input: G a connected graph, $w\colon E(G) \to \mathbb{R}_+$ a non-neg edge weight
Output: A spanning tree of G with minimum total edge weight

Well-known fact (Kalaba '60)

Start with an arbitrary spanning tree of G, and keep exchanging to a spanning tree with smaller total weight. Then you get an optimum.

Well-known algorithms (in textbooks)

- Prim's method: Tree growing
  - Naive implementation: $O(n^2)$
- Kruskal's method: Greedy forest merging
  - Naive implementation: $O(m \log n)$

Computing a minimum-cost spanning tree (2)

Problem MINIMUM COST SPANNING TREE

Input: G a connected graph, $w\colon E(G) \to \mathbb{R}_+$ a non-neg edge weight
Output: A spanning tree of G with minimum total edge weight

Best algorithms: Current status

- $O(m\alpha(m, n))$ (Chazelle '00)
- $O((m{+}n \log L) \log\log L)$ ($L = \max_{e \in E(G)} w(e)$) (Johnson '77)
- Expected $O(m{+}n)$ (Karger, Klein, Tarjan '95)
- O(min # comparisons needed to determine an optimum) (Pettie, Ramachandran '02)

Open Problem

Develop an $O(m{+}n)$ algorithm for MINIMUM COST SPANNING TREE, or prove this is impossible

Today's contents

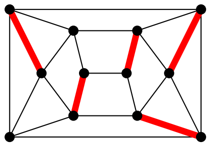
Matchings

$G = (V, E)$ a graph

Definition (Matching)

An edge subset $M \subseteq E$ is a matching of G if no two edges of M share a common vertex

Example

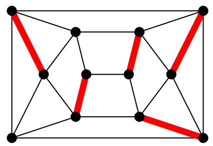


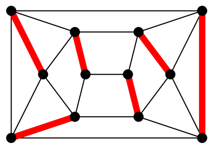
Saturation

$G = (V, E)$ a graph; $M \subseteq E$ a matching of G

Definition (Saturation)

A vertex $v \in V$ is M-saturated if v is incident to some edge of M;
Otherwise, v is M-unsaturated;
We say M saturates X ($X \subseteq V$) if every vertex $v \in X$ is M-saturated





Definition (Perfect matching)

M is perfect if it saturates V

Maximum matchings and maximal matchings

$G = (V, E)$ a graph; $M \subseteq E$ a matching of G

Definition (Maximum matching)

M is a maximum matching of G if $\forall$ matchings M' of G, $|M| \geq |M'|$

Definition (Maximal matching)

M is a maximal matching of G if $M \cup \{e\}$ is not a matching of G for any $e \in E \setminus M$

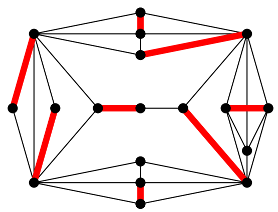
Note: M maximum $\substack{\Rightarrow \\ \not\Leftarrow}$ M maximal

Notation

$\alpha'(G)$ = the size of a maximum matching of G
(called the matching number of G)

Is it a maximum matching?

How can we certify this is a maximum matching?



"Exhibiting one large matching" is not enough for *proving* the optimality

Alternating paths and augmenting paths

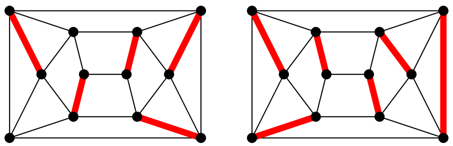
$G = (V, E)$ a graph; $M \subseteq E$ a matching of G

Definition (Alternating path)

A path of G is an M-alternating path if it alternates between edges in M and edges not in M

Definition (Augmenting path)

An M-alternating path of G is an M-augmenting path if its endpoints are M-unsaturated



Berge's characterization of maximum matchings

$G = (V, E)$ a graph; $M \subseteq E$ a matching of G

Theorem 3.10 (Characterization of maximum matchings, Berge '57)

M is a maximum matching of $G \iff G$ has no M-augmenting path

Proof idea.

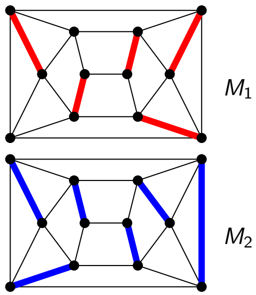
Consider the contrapositive of each direction

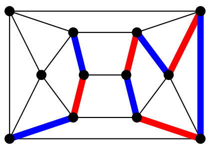
[$\Rightarrow$] If G has an M-augmenting path, then augment M

[$\Leftarrow$]

- Let M' be a matching s.t. $|M'| > |M|$
- Consider the symmetric difference $M \triangle M'$ $(= (M \cup M') \setminus (M \cap M'))$
- Each component of $(V, M \triangle M')$ is either a path or a cycle
- One of the paths is an M-augmenting path □

Symmetric difference of two matchings: a picture





$M_1 \triangle M_2$

Each connected component of $(V, M_1 \triangle M_2)$ is either a path or an even cycle

How to find a maximum matching

Berge's theorem gives a way to find a maximum matching

Algorithm

1. M an arbitrary matching of G (for example, $M = \emptyset$)
2. Until $\exists$ an M-augmenting path, augment M to obtain a larger matching

Difficulty in the algorithm above

Unclear how to find an M-augmenting path, or certify one doesn't exist
(Berge's characterization is not seemingly a good characterization)

But, actually there are ways to do this; We don't discuss it here

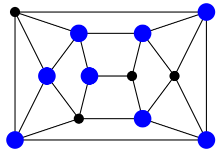
Vertex covers

$G = (V, E)$ a graph

Definition (Vertex cover)

A vertex subset $C \subseteq V$ is a vertex cover of G if every edge is incident to a vertex of C

Example



Minimum vertex covers and minimal vertex covers

$G = (V, E)$ a graph; $C \subseteq V$ a vertex cover of G

Definition (Minimum vertex cover)

C is a minimum vertex cover of G if $|C| \leq |C'|$ for all vertex covers C' of G

Definition (Minimal vertex cover)

C is a minimal vertex cover of G if $C \setminus \{v\}$ is not a vertex cover of G for any $v \in C$

Notation

$\beta(G)$ = the size of a minimum vertex cover of G

(called the covering number of G)

Weak duality

$G = (V, E)$ a graph

Proposition 3.11

$M \subseteq E$ a matching of G,
$C \subseteq V$ a vertex cover of G $\Rightarrow |M| \leq |C|$

Proof idea.

Double counting: Count $\{(e, v) \in M \times C \mid e \text{ incident to } v\}$ □

Corollary 3.12 (Weak duality)

For any graph G, $\alpha'(G) \leq \beta(G)$

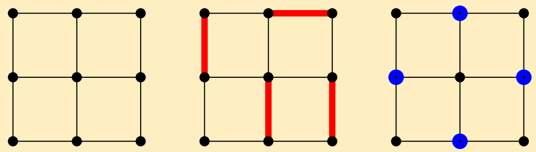
How to certify the optimality of a matching

You find a matching of size k, then you prove $\alpha'(G) \geq k$

You find a vertex cover of size k, then you prove $\alpha'(G) \leq \beta(G) \leq k$

Then, you may conclude that $\alpha'(G) = k$

Example



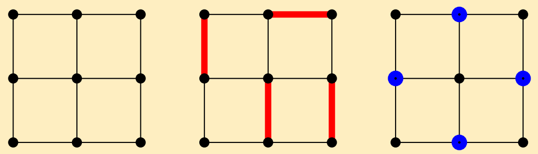
G $\alpha'(G) \geq 4$ $\alpha'(G) \leq 4$

An innocent question

Is it always possible to find a matching and a vertex cover of the same size?

If it is the case, we can always certify the optimality of a matching by exhibiting such a vertex cover

Example



G $\alpha'(G) \geq 4$ $\alpha'(G) \leq 4$

Not necessarily the case, but true for bipartite graphs

Next lecture

We should answer the following questions

- Is it always possible to find a matching and a vertex cover of the same size for bipartite graphs?
  - Answer: Yes (Kőnig-Egerváry theorem)
- Is it possible to certify the non-existence of a perfect matching in a (non-bipartite) graph easily?
  - Answer: Yes (Tutte's theorem)

Today's contents

Open problem: Erdős-Sós conjecture

Conjecture (Erdős, Sós '63)

$e(G) > (k-1)n(G)/2 \Longrightarrow G$ contains *all* trees with k edges

Known facts

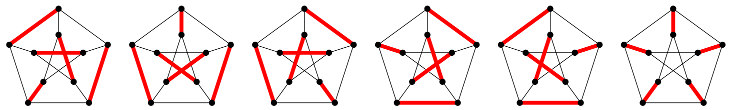
- Such a graph contains $K_{1,k}$ (Easy)
- Such a graph contains P_{k+1} (Erdős, Gallai '59)
- Such a graph contains all trees w/ diameter ≤ 4 (McLennan '05)
- Theorem holds when G contains no C_4 (Saclé, Woźniak '97)
- and many others...
- Theorem holds for large n
  (Ajtai, Komlós, Simonovits, Szemerédi, in preparation?)

Also refer to Exercise 3.4

Open problem: The Fulkerson Conjecture

Conjecture (Fulkerson '71)

In each 3-regular graph without a cut-edge, there exist (not necessarily distinct) six perfect matchings that together cover each edge precisely twice



- This is related to the half-integrality of certain high-dimensional polyhedra (polyhedral combinatorics)
- The conejcture is also called the Cycle Double Cover Conjecture (because of an equivalent form of the conjecture)
- See also Cornuéjols' book ('01, Conjecture 1.32 there).

From graphs to simplicial complexes

Fact

Given a graph G, the family F of matchings satisfies the following properties

- $\emptyset \in F$
- $M \in F$, $M' \subseteq M \Rightarrow M' \in F$

In general, a family F satisfying the properties above is called an (abstract) simplicial complex
The simplicial complex above is called the matching complex of G

Open problem

Determine the topological type of the matching complex of G

Well-studied when $G = K_n, K_{n,m}$, but not yet completely understood;
See a survey by Wachs ('03)