Lecture notes for Mar 6, 2023
Edmonds' Blossom Algorithm and Linear Programming

Chun-Hung Liu

March 6, 2023

Last time we proved an easy upper bound for the size of a maximum matching.

Proposition 1 *Let G be a graph. Then*

$$\max_{M} |M| \leq \min_{X \subseteq V(G)} \frac{|V(G)| + |X| - \text{oddcomp}_G(X)}{2},$$

where the maximum is over all matchings M in G.

Recall that last time we stated Edmonds' Blossom Algorithms for finding a maximum matching and proved the following properties.

Lemma 2 *During the entire process, the following properties are preserved.*

1. *T is a forest and $V(T)$ equals the set of marked vertices.*
2. *M is a matching of G.*
3. *For every $v \in S$, $s(v) =$outer, and if v is M-saturated, then the vertex matched to v by M is in $V(T)$.*
4. *For every marked edge, its both ends are in T.*
5. *For every edge e in M, either $e \in E(T)$, or its both ends are not in $V(T)$.*

6. $\bigcup_{(Z,z)\in\mathcal{F}} Z = V(G)$, *and if* (Z_1, z_1) *and* (Z_2, z_2) *are distinct members of* $\mathcal{F}$ *with* $Z_1 \cap Z_2 \neq \emptyset$, *then* $|Z_1| \geq 2 \leq |Z_2|$, $|Z_1 \cap Z_2| = 1$ *and* $Z_1 \cap Z_2 \subseteq \{z_1, z_2\}$.

7. *For every member* (Z, z) *of* $\mathcal{F}$, *if* $|Z| \geq 2$, *then* $Z \subseteq V(T)$, $T[Z]$ *is a connected subgraph in* T *rooted at* z, *and* $s(v) =$*outer for every* $v \in Z$.

8. *For every* $v \in V(T)$, *the following hold.*

   (a) $R(v)$ *is the root of the component of* T *containing* v.

   (b) *If* $R(v) \neq v$, *then* v *is* M*-saturated.*

   (c) *If* $b(v) = v \neq R(v)$, *then the edge between* v *and its parent is in* M.

   (d) *If* $a(v) \neq *$, *then* $va(v) \in E(G) - M$, *and* $s(a(v)) =$*outer.*

   (e) *If* $s(v)$ *is inner, then it has a parent in* T, *the edge between* v *and its parent is not in* M, *it has a unique child in* T, *and the edge between* v *and the child is an edge in* M.

   (f) *If* $v = b(u)$ *for some* $u \in V(G)$, *then* $s(v) =$*outer.*

   (g) *If* $b(v) \neq *$, *then there exists a unique path* W *of the form* $v_1 v_2 ... v_{|V(W)|}$ *satisfying* $v_1 = v$, $v_{|V(W)|} = b(v)$, *and* $v_{i+1} = a(v_i)$ *for every odd* i, *and* v_{i+1} *is the vertex matched to* v_i *by* M *for every even* i*; moreover,* W *is an* M*-alternating path in* G *from* v *to* $b(v)$, $V(W)$ *is contained in* Z *for some* $(Z, b(v)) \in \mathcal{F}$, *and all vertices of* W *are outer.*

   (h) *If* $b(v) \neq *$, *then there exists a unique path* W *of the form* $v_1 v_2 ... v_{|V(W)|}$ *satisfying* $v_1 = v$, $v_{|V(W)|} = b(v)$, *and* $v_{i+1} = a(v_i)$ *for every even* i, *and* v_{i+1} *is the vertex matched to* v_i *by* M *for every odd* i*; moreover,* W *is an* M*-alternating path in* G *from* v *to* $b(v)$, *and* $V(W)$ *is contained in* Z *for some* $(Z, b(v)) \in \mathcal{F}$, *and all vertices of* W *are outer.*

   (i) *If* $b(v) \neq v$, *then the path* P_v *defined in the algorithm is an* M*-alternating path from* v *to* $R(v)$ *such that the edge of* P_v *incident with* v *is in* M *and the edge of* P_v *incident with* $R(v)$ *is not in* M.

9. *For every member* (Z, z) *of* $\mathcal{F}$, Z *is a blossom based at* z.

Now we use them to prove the correctness of the algorithm.

1 Correctness of Edmonds' Blossom Algorithm

Lemma 3 *Let M be the matching of G when Edmonds' Blossom algorithm terminates. Let X be the set of inner vertices when Edmonds' Blossom algorithm terminates. Let Y be the set of outer vertices when Edmonds' Blossom algorithm terminates. Let $W = V(G) - (X \cup Y)$. Then the following hold.*

1. *The components of $G[Y]$ are exactly the odd components of $G - X$.*
2. *$M \cap E(G[Y])$ gives a near-perfect matching for each component of $G[Y]$.*
3. *$2|M| = |V(G)| + |X| - \text{oddcomp}_G(X)$.*
4. *M matches each vertex in X to a vertex in Y such that no two vertices in X are matched to vertices in the same component of $G[Y]$.*
5. *$M \cap E(G[W])$ is a perfect matching of $G[W]$.*

Proof. First, notice that $X \cup Y$ is the set of marked vertices and W is the set of unmarked vertices.

Note that no inner vertex is contained in Z for any member (Z, z) of $\mathcal{F}$ with $|Z| \geq 2$. By Properties 8(g) and 8(h) in Lemma 2, every member of $G[Z]$ is connected for every $(Z, z) \in \mathcal{F}$. So every component of $G - X$ is a union of members of $\mathcal{F}$. Moreover, all vertices in $G - (X \cup W)$ are outer, so there is no edge of $G - X$ between two different vertices in different members of $\mathcal{F}$ intersecting Y and there is no edge of $G - X$ between a vertex in Y and a vertex in W, for otherwise the shrinking step should merge these two members or the growing step should include more marked vertices. So by property 6 in Lemma 2, for every component C of $G - X$ intersecting Y, there exist members $(Z_{C,i}, z_{C,i})$ of $\mathcal{F}$ for $i \in [k]$ and some integer k such that $V(C) = \bigcup_{i=1}^{k} Z_{C,i}$, $Z_{C,j} \cap \bigcup_{i=1}^{j-1} Z_{C,i} = \{z_{C,j}\}$ for every $2 \leq j \leq k$, and all vertices in C are descendants of $z_{C,1}$. For each component C of $G - X$ intersecting Y, by property 9 in Lemma 2, $z_{C,1}$ is the unique non-M-saturated vertex in C, so $|V(C)|$ is odd, and $M \cap E(G[Y])$ gives a near-perfect matching for each component of $G[Y]$. This proves Statement 2 and proves that every component of $G - X$ intersecting Y is odd. And notice that every vertex in W is saturated by M. Since every M-saturated marked vertices is matched to a marked vertex by M, $M \cap E(G[W])$ is a perfect matching of $G[W]$, so every component of $G - X$ intersecting W is a component of $G[W]$ and is even. This proves Statements 1 and 5.

For every component C of $G-X$, if $z_{C,1}$ is not a root of a component of T, then let p_C be the parent of $z_{C,1}$. Since all vertices in C are descendants of $z_{C,1}$, $p_C \not\in V(C)$, so $p_C \in X$. Since p_C is inner, Property 8(e) in Lemma 2 implies that it has the unique child, so the number of components of $G-X$ intersecting Y equals $|X|$ plus the number of components of T. Note that for each component of T, its root is the unique non-M-saturated vertex. So the number of components of T equals $|V(G)|-2|M|$. Therefore, Statement 1 implies $\text{oddcomp}_G(X) = |X| + (|V(G)|-2|M|)$. This proves Statement 3.

By Property 8(e) in Lemma 2, every inner vertex is matched by M to its unique child. So M matches each vertex in X to a vertex in Y. Then Statement 4 follows from Statements 2. ■

Lemma 4 *The matching M output by the algorithm is a maximum matching in G.*

Proof. It immediately follows from Proposition 1 and Statement 3 of Lemma 3. ■

Theorem 5 *Edmonds' Blossom Algorithm outputs a maximum matching in time $O(|V(G)|^3)$.*

Proof. It outputs a maximum matching by Lemma 4. Since the maximum matching has size at most $|V(G)|$, augmenting cases can only happen at most $|V(G)|$ times. And each growing case takes time $O(1)$ and increases $|V(T)|$. And each shrinking case decreases $|\mathcal{F}|$ and takes time $O(|V(G)| + \sum_{v\in Y} \deg_G(v))$, where Y is the set of vertices that were inner and turns outer in this shrinking case. Since both $|V(T)|$ and $|\mathcal{F}|$ are bounded by $|V(G)|$, and every vertex can turn outer at most once, we know that between any two consecutive augmenting cases, it takes time $O(|V(G)|^2 + |E(G)|) = O(|V(G)|^2)$. Since augmenting cases can only happen at most $|V(G)|$ times, and each augmenting case takes time $O(|V(G)|)$, the total running time is $O(|V(G)|^3)$. ■

2 Applications of the blossom algorithm

Corollary 6 (Berge-Tutte formula) *Let G be a graph. Then*

$$\max_M |M| = \min_{X\subseteq V(G)} \frac{|V(G)|+|X|-\text{oddcomp}_G(X)}{2},$$

where the maximum is over all matchings M in G.

Proof. It immediately follows from Proposition 1 and Statement 3 of Lemma 3. ∎

Another corollary of Edmonds' blossom algorithm is to obtain the Gallai-Edmonds decomposition, which is a strengthening of Berge-Tutte formula. The *Gallai-Edmonds decomposition* of a graph G is a tuple (X, Y, W) such that

- Y is the set of vertices that are unsaturated by at least one maximum matching,
- X be the set of vertices not in Y but adjacent in G to some vertex in Y, and
- $W = V(G) - (X \cup Y)$.

We show that Edmonds' blossom algorithm also finds the Gallai-Edmonds decomposition.

Corollary 7 *Let G be a graph. Let Y be the set of vertices that are unsaturated by at least one maximum matching. Let X be the set of vertices not in Y but adjacent in G to some vertex in Y. Let $W = V(G)-(X\cup Y)$. (That is, (X, Y, W) is the Gallai-Edmonds decomposition.) Then the following hold.*

1. *If M is the matching when Edmonds' Blossom Algorithm terminates, and X_M, Y_M, W_M are the corresponding sets X, Y, W described in Lemma 3, then $X = X_M$, $Y = Y_M$ and $W = W_M$.*
2. *Every maximum matching of G contains a perfect matching of $G[W]$ and a near-perfect matching of each component of $G[Y]$ and matches each vertex in X to a vertex in Y such that different vertices in X are matched to vertices in different components of $G[Y]$.*
3. *Every connected component of $G[Y]$ has a near-perfect matching.*
4. $$\max_M |M| = \frac{|V(G)| + |X| - \mathrm{oddcomp}_G(X)}{2},$$
   where the maximum is over all matchings M in G.

Proof. Let M be the matching of G when Edmonds' Blossom algorithm terminates. Let X_M be the set of inner vertices when Edmonds' Blossom algorithm terminates. Let Y_M be the set of outer vertices when Edmonds' Blossom algorithm terminates. Let $W_M = V(G) - (X_M \cup Y_M)$.

By Lemma 4, M is a maximum matching. So every maximum matching has size $|M|$. Hence by Statements 1 and 3 of Lemma 3, every maximum matching of G has size $\frac{|V(G)|+|X_M|-\text{oddcomp}_G(X_M)}{2}$ and the components of $G[Y_M]$ are exactly the odd components of $G - X_M$, so every maximum matching contains a perfect matching of the union of the even component of $G - X_M$ (i.e. $G[W_M]$) and a near perfect matching of each odd component of $G - X_M$ (i.e. each component of $G[Y_M]$) and matches each vertex in X_M to a vertex in an odd component (i.e. in Y_M) such that different vertices in X_M are matched to different odd components of $G - X_M$ (i.e. different components of $G[Y_M]$).

Hence if $y \in Y$, the maximum matching of G that does not saturate y shows that y must be in an odd component of $G - X_M$ (i.e. in Y_M). So $Y \subseteq Y_M$. And for every $y \in Y_M$, the path P_y mentioned in Edmonds' Blossom Algorithm is defined, and $M \Delta E(P_y)$ is a matching with size equal to $|M|$ that does not saturate y, so $M \Delta E(P_y)$ is a maximum matching that does not saturate y, and hence $y \in Y_M$. So $Y_M \subseteq Y$. Therefore, $Y_M = Y$.

Since $Y = Y_M$ are exactly the vertices contained in the odd components of $G - X_M$, $X = N_G(Y) = N_G(Y_M) = X_M$. So $W = V(G) - (X \cup Y) = V(G) - (X_M \cup Y_M) = W_M$. Therefore, Statement 1 of this corollary holds.

Recall that every maximum matching contains a perfect matching of $G[W_M] = G[W]$ and a near perfect matching of each component of $G[Y_M] = G[Y]$ and matches each vertex in $X_M = X$ to a vertex in $Y_M = Y$ such that different vertices in $X_M = X$ are matched to different components of $G[Y_M] = G[Y]$. So Statement 2 of this corollary holds. And Statements 3 and 4 of this corollary follows from Statements 2 and 3 of Lemma 3, respectively. ■

Corollary 8 *Given a graph G, the Gallai-Edmonds decomposition of G can be found in $O(|V(G)|^3)$ time.*

Proof. It immediately follows from Theorem 5 and Corollary 7. ■

3 Linear programming

Many problems we have considered can be formulated as optimization problems involving matrices and vectors.

For example, given a subset S of $V(G)$, we can define a 0-1 vector x whose each entry corresponds to a vertex of G such that a vertex v is in S if and only if the v-th entry of x (denoted by x_v) is 1. So a subset S is a stable set in G if and only if the corresponding vector x satisfies $x_u + x_v \leq 1$ for every edge $uv \in E(G)$. Equivalently, S is a stable set in G if and only if the corresponding vector x satisfies $Ax \leq 1$, where A is the edge-vertex incident matrix of G. Hence the independence number of G is the maximum $1^T x$ over all 0-1 vectors x satisfying $Ax \leq 1$, where A is the edge-vertex incident matrix of G.

Therefore, finding the independence number of a graph is a special case of the optimization problem of the form $\max_x c^T x$ subject to $Ax \leq b$ and x is a 0-1 vector, for some matrix A and vectors b and c. Such an optimization problem is called an *integer programming problem*.

Example: Examples of combinatorial problems that can be formulated as $\max_x c^T x$ subject to $Ax \leq b$ and x is a 0-1 vector include:

1. As we have seen, the independence number of a graph is the case that A is the edge-vertex incident matrix of G and b and c are equal to 1.

2. The maximum size of a matching is the case that A is the vertex-edge incidence matrix of A, and b and c are equal to 1. (Note that b has $|V(G)|$ entries and c has $|E(G)|$ entries.)

As we have shown that it is NP-hard to find the independence number, integer programming is NP-hard in general. However, some special case of integer programming problems can be solved in polynomial time, such as the one that formulates the maximum size of a matching.

If we do not restrict x to be a 0-1 vector, then we obtain the problem $\max_x c^T x$ subject to $Ax \leq b$. This kind of problem is called a *linear programming problem*.

Linear Programming
Input: An $m \times n$ matrix A over real numbers, a column vector $b \in \mathbb{R}^m$ and a column vector $c \in \mathbb{R}^n$.
Output: Find $\max_{x \in \mathbb{R}^n, Ax \leq b} c^T x$, or conclude that the set $\{x \in \mathbb{R}^n : Ax \leq b\}$ is empty, or conclude that the maximum does not exist (i.e. for every $\alpha \in \mathbb{R}^n$, there exists $x_0 \in \mathbb{R}^n$ with $Ax_0 \leq b$ such that $c^T x_0 > \alpha$).