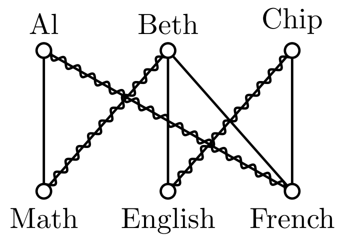
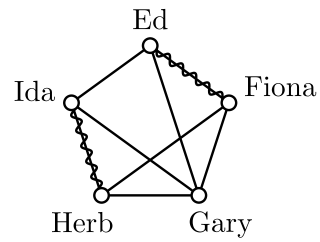
MATCHINGS

Reading: B&M 16.1-5.

Examples: Assignment Problem

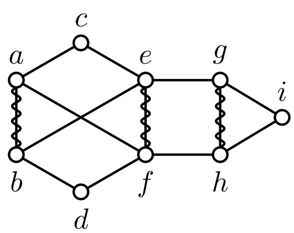


Roommate Problem

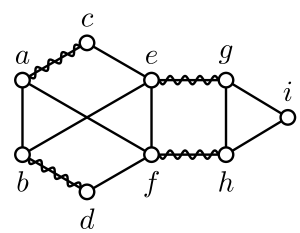


∘ *matching* M: set of independent (pairwise nonadjacent, no common vertex) edges.
∘ M-*saturated* vertex: incident with edge of M, otherwise M-*unsaturated*.
∘ *perfect* matching or 1-*factor*: saturates all vertices.

Examples:



M_1, maxim*al*, not maxim*um*

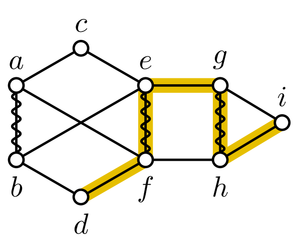


M_2, maxim*um*

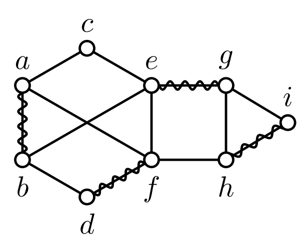
Notation: For $S \subseteq E(G)$ we also use S to mean $G[S]$ subgraph induced by S, S and ends of edges in S. For sets $S \triangle T = (S - T) \cup (T - S)$. For subgraphs $H \triangle J$ is subgraph induced by $E(H) \triangle E(J)$. Different from previous definition for spanning subgraphs.

∘ M-*alternating path*: edges alternately in, not in, M.
∘ M-*augmenting path* P: nontrivial M-alternating, ends M-unsaturated. Then $M \triangle P$ is a larger matching.

Examples:



M_1-augmenting path P_1

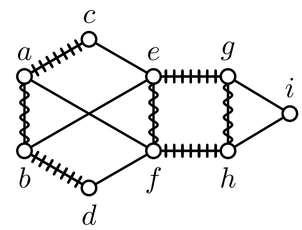


$M_1 \triangle P_1$

Berge's Theorem: A matching M has an M-alternating path ⇔ it is not maximum.

Proof: (⇒) If an M-augmenting path exists, we can find a larger matching than M.

(⇐) Suppose M is not maximum. Let M' be a matching that *is* maximum. Let $F = M \triangle M'$. The components of F are M-alternating paths or *even* cycles. Since $|M'| > |M|$, some component of M' has more M' edges than M edges: must be odd length path, starts and ends with M-unsaturated vertices, so M-augmenting path as required. ∎



$M_1 \triangle M_2$

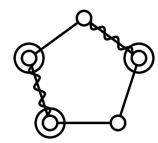
Corollary M1: Contrapositive: Matching M is maximum ⇔ no M-augmenting path exists.

Matchings in bipartite graphs

∘ $\alpha'(G)$ = size of maximum matching.

∘ *vertex cover* K: $K \subseteq V(G)$, every edge has at least one end in K.
∘ $\beta(G)$ = cardinality of minimum vertex cover.

(M2) If M matching, K vertex cover, then $|M| \le |K|$ since K contains at least one end of each $e \in M$. Hence $\alpha'(G) \le \beta(G)$. Also if $|M| = |K|$, then M is maximum, K is minimum. For bipartite graphs will show $\alpha' = \beta$, not true in general..

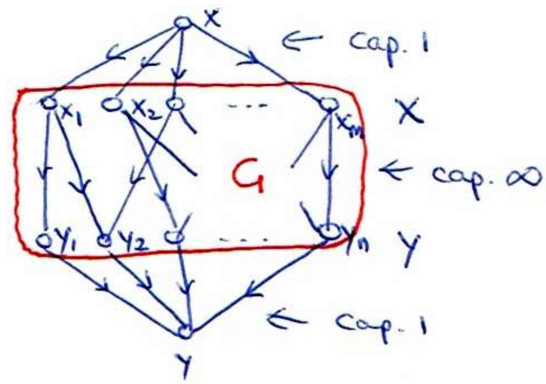


C_5 has $\alpha' = 2 < \beta = 3$.

Bipartite matching ↔ network flow

Given bipartite $G(X, Y)$, construct flow network:
Feasible integer flow in (D, c)
↕
Matching M in G

So $\alpha'(G)$ = value of max. xy-flow = capacity of min. xy-cut. So need to examine xy-cuts in D. Infinite capacity edges will not appear in min. cuts.

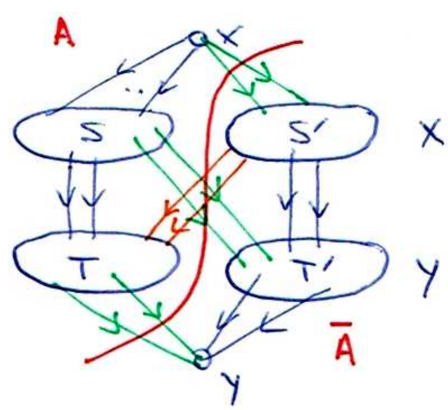


Given xy-cut $\delta^+ A$, let $S = X \cap A$, $S' = X - A$,
$T = Y \cap A$, $T' = Y - A$ $(A = \{x\} \cup S \cup T)$.

(1) If $[S, T'] \neq \emptyset$ then $c(\delta^+ A) = \infty$.

(2) So $\delta^+ A$ has finite capacity
⇔ $[S, T'] = \emptyset$ (in G)
⇔ $K = S' \cup T$ is a vertex cover (in G).

For vertex cover K in G, can take $A = \{x\} \cup (X - K) \cup (Y \cap K)$ in D. So 1-1 correspondence $A \leftrightarrow K$ with $c(\delta^+ A) = |S'| + |T| = |K|$.



(3) Since finite cap. xy-cuts exist (e.g., $\delta^+ x$),
$\alpha'(G)$ = min. cap. of xy-cut
= min. finite cap. of xy-cut
= $\min\{|K| \mid K \text{ a vertex cover}\}$
= $\beta(G)$ – **Kőnig-Egerváry Theorem** true for **bipartite** graphs.

(4) And $\alpha'(G) = $ cap. of min. xy-cut $= \min\limits_{S\subseteq X, T\subseteq Y: N(S)\subseteq T} |S'| + |T| = \min\limits_{S\subseteq X} \min\limits_{T\subseteq Y: N(S)\subseteq T} |S'| + |T|$

$= \min\limits_{S\subseteq X} |S'| + |N(S)|$ since $N(S)$ is smallest T with $N(S) \subseteq T$

$= \min\limits_{S\subseteq X} |X| - |S| + |N(S)| = |X| - \max\limits_{S\subseteq X} \underbrace{(|S| - |N(S)|)}_{\textit{excess} \text{ of } S}$

positive excess means S has too many vertices to match them all to $N(S)$

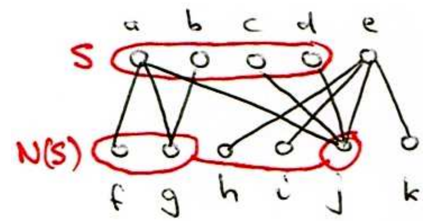
– **Kőnig-Ore formula** true for **bipartite** graphs.

Corollary M3, Hall's Theorem: Bipartite $G(X, Y)$ has a matching saturating all of X

$\Leftrightarrow \alpha'(G) = |X|$

$\Leftrightarrow$ every $S \subseteq X$ has nonpositive excess ($\emptyset$ has excess of 0)

$\Leftrightarrow |N(S)| \geq |S|$ for all $S \subseteq X$.



Corollary M4 (Kőnig): For $k \geq 1$, every k-regular bipartite graph $G(X, Y)$ has a perfect matching. Graph does **not** need to be simple. For most matching results can just use underlying simple graph; not this one since underlying simple graph may not be regular.

Proof: From the bipartite degree-sum formula, $\sum_{x\in X} d(x) = k|X| = \sum_{y\in Y} d(y) = k|Y|$, so $|X| = |Y|$. So it is enough to find a matching saturating X. For any $S \subseteq X$, we have

$k|S| = \#$ edges out of $S \leq \#$ edges into $N(S) = k|N(S)|$

so that $|N(S)| \geq |S|$ for all $S \subseteq X$, so by Hall's Theorem the required matching exists. ∎

Corollary M5: For $k \geq 0$, the edges of a k-regular bipartite graph can be partitioned into k perfect matchings. Later will connect this to edge-colourings.

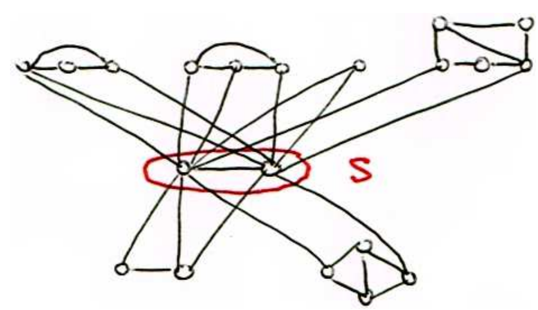
Matchings in general graphs

Look at what restricts size of maximum matching.

Example: $G - S$ has 4 *odd* components.

At most 2 odd components have vertex matched to vertex of S.

∴ At least two unmatched vertices.



- *defect* of M is $\text{def}(M) =$ number of M-unsaturated vertices $= n - 2|M|$.
- *shortfall* of $S \subseteq V(G)$ is $\text{shf}(S) = c_{\text{odd}}(G - S) - |S|$. My terminology: how much S falls short of helping all odd components get matched. May be positive, 0 or negative. But empty set has nonnegative shortfall so maximum always nonnegative.

(M6) For any matching M and $S \subseteq V(G)$, $\text{def}(M) \geq \text{shf}(S)$.

(M7) For any $S \subseteq V(G)$, $\text{shf}(S) \equiv |V(G)| \bmod 2$. Both even or both odd.

Proof: $|V(G)| = \overbrace{|S| - c_{\text{odd}}(G - S)}^{-\text{shf}(S)} + \overbrace{\sum_{\text{odd } C} (|V(C)| + 1)}^{\text{even}} + \overbrace{\sum_{\text{even } C} |V(C)|}^{\text{even}}$. ∎

Berge's Formula, 1958: For any G,

the minimum defect of any matching = the maximum shortfall of any $S \subseteq V(G)$.

Hence $\alpha'(G) = \frac{1}{2}\left(|V(G)| - \min_{M \text{ matching}} \operatorname{def}(M)\right) = \frac{1}{2}\left(|V(G)| - \max_{S\subseteq V(G)} (c_{\text{odd}}(G-S) - |S|)\right)$.

Special case: $\min_M \operatorname{def}(M) = 0 \Leftrightarrow \max_S \operatorname{shf}(S) = 0 \Leftrightarrow \operatorname{shf}(S) \le 0 \ \forall\ S \subseteq V(G)$ $\quad (\operatorname{shf}(\emptyset) \ge 0)$

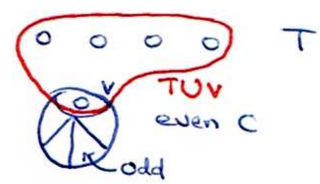
i.e. G has a perfect matching $\Leftrightarrow c_{\text{odd}}(G-S) \le |S| \ \forall\ S \subseteq V(G)$

– **Tutte's 1-Factor Theorem, 1947**.

Berge originally proved his formula using Tutte's Theorem. We prove both together.
B&M call this Tutte-Berge Formula/Theorem.

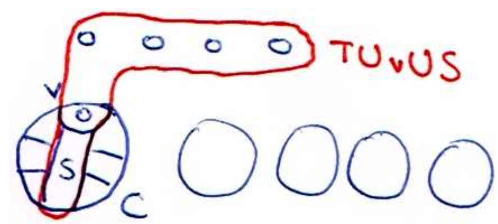
Proof of Berge's Formula: (This proof based on West, Eur. J. Combin. 2011. Similar proof in Kotlov, arXiv:math/0011204v1, 2000.) By induction on $|V(G)|$. True if $|V(G) = 1$. Consider a maxim*al* set of maxim*um* shortfall (which is ≥ 0), T. Enough to find matching M with $\operatorname{def}(M) = \operatorname{shf}(T)$.

(a) All components of $G-T$ are odd: If not, take an even component C and $v \in V(C)$. Then $c_{\text{odd}}(G-(T\cup v)) = c_{\text{odd}}(G-T) + c_{\text{odd}}(C-v) \ge c_{\text{odd}}(G-T)+1$ and $|T\cup v| = |T|+1$, giving $\operatorname{shf}(T\cup v) \ge \operatorname{shf}(T)$, contradicting maximality assumptions for T. Will use 'v' to denote *set* of single vertex v to simplify notation, should not cause any confusion.



(b) If C is a component of $G-T$ and $v \in V(C)$ then $C-v$ has a perfect matching M_{C-v}: We claim that $\operatorname{shf}_{C-v}(S) \le 0 \ \forall\ S \subseteq V(C-v)$. We have

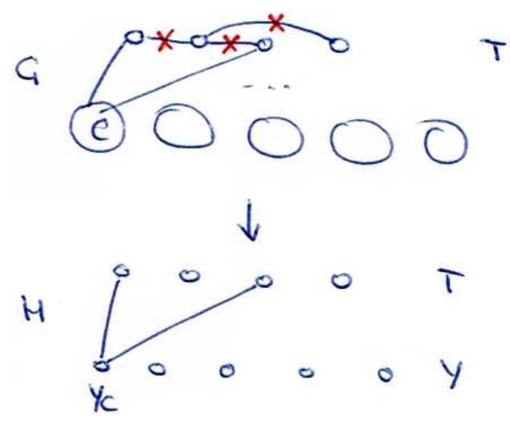
$$\begin{aligned}\operatorname{shf}_G(T\cup v\cup S) &= c_{\text{odd}}(G-T-v-S) - |T\cup v\cup S| \\ &= (c_{\text{odd}}(G-T)-1) + c_{\text{odd}}(C-v-S)) - |T| - |S| - 1 \\ &= (c_{\text{odd}}(G-T)-|T|) + (c_{\text{odd}}(C-v-S)-|S|) - 2 \\ &= \operatorname{shf}_G(T) + \operatorname{shf}_{C-v}(S) - 2.\end{aligned}$$



Since T is maximal of maximum shortfall, $\operatorname{shf}_G(T\cup v\cup S) < \operatorname{shf}_G(T)$, so $\operatorname{shf}_{C-v}(S) \le 1$. But since $C-v$ is even, by Observation M7 $\operatorname{shf}_{C-v}(S)$ is even, so $\operatorname{shf}_{C-v}(S) \le 0$. Thus, by induction (using the special case) $C-v$ has a perfect matching.

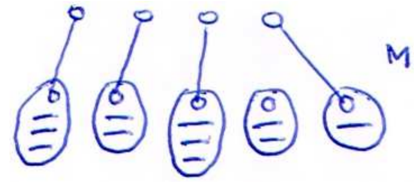
(c) Now can find matching of defect $\operatorname{shf}(T)$ as long as can match every vertex of T to a component of $G-T$. To do this, use bipartite matching! Construct a bipartite graph H from G by (1) deleting all edges inside T, and (2) contracting every component C of $G-T$ to a single vertex y_C; let Y be the set of such y_C's. Then H has a matching M_H saturating T: Let $S \subseteq T$, then

$$\begin{aligned}|Y| - |T| &= c_{\text{odd}}(G-T) - |T| = \operatorname{shf}_G(T) \\ &\ge \operatorname{shf}_G(T-S) = c_{\text{odd}}(G-(T-S)) - |T-S| \\ &\ge |Y - N_H(S)| - |T| + |S| \\ &= |Y| - |T| + |S| - |N_H(S)|\end{aligned}$$



because for every $y_C \notin N_H(S)$, C is an odd component of $G-(T-S)$ (may also be other odd components from S and components of $G-T$ adjacent to S). Thus, $|N_H(S)| \ge |S|$ for all such S, and M_H exists by Hall's Theorem.

(d) Now for each component C of $G-T$:
if M_H contains edge $t_C y_C$, let $t_C v_C$ be the corresponding edge in G,
otherwise let v_C be any vertex of C.
Let $M = \bigcup_{\text{comps } C \text{ of } G-T} M_{C-v_C} \cup \{t_C v_C \mid t_C y_C \in M_H\}$. Then $\text{def}(M) =$ (since all vertices of T covered by M, just worry about $G-T$) $c_{\text{odd}}(G-T)$ (vertices in $G-T$ missed by first term) $-|T|$ (vertices of $G-T$ covered by second term) $= \text{shf}(T)$, as required. ∎



Also gives structure for maximum matchings. If refine a bit more we get Gallai-Edmonds Theorem.
Set of maximum shortfall often called a *barrier* and value of min defect/max shortfall is *deficiency* of G.

Algorithmic material not covered in class, included here in case you are interested

---

Bipartite matching algorithm (Egerváry's algorithm)

Idea: Translate flow augmentation into direct operation on matching.

Rough outline: Search for M-augmenting paths by building search forest:
- start with all M-unsaturated vertices in X;
- go down ($X \to Y$) on non-M edges;
- go back up ($Y \to X$) on M edges (add automatically);
- hoping to get to an M-unsaturated vertex in Y.

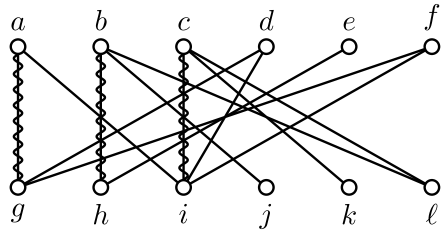
Augment and repeat, until no M-augmenting path can be found. Then use current search forest F to find vertex cover that proves M is maximum.

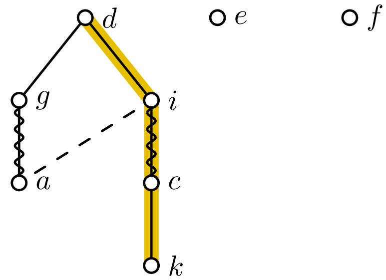
Exact algorithm

let $M = M_0$ ($M_0 = \emptyset$, or chosen greedily);
while find-alt-path-forest(M, F) returns an M-augmenting path P
    augment M using P;
maximum matching $M^* = M$;
minimum vertex cover $K^* = S' \cup T$ where
    S' = vertices of X not in F,
    T = vertices of Y in F.

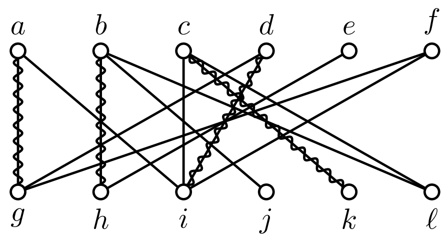
```
find-alt-path-forest(M, F) {
    let F = all M-unsaturated vertices in X;
    root each component of F;
    construct F by modified local tree search as follows:
        while there is xy with x ∈ X ∩ V(F), y ∈ Y − V(F) {
            add xy to F;        # necessarily xy ∉ M
            if y is incident with an edge yx' of M
                add yx' to F;
            else
                return M-augmenting path from root of y's component to y;
        }
    return nothing;
}
```

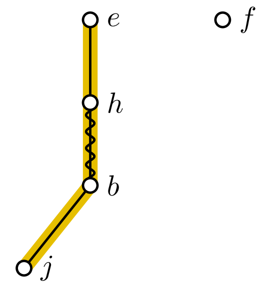
Example: Start with obvious vertical matching. Search forest F of alternating paths contains M-aug. path *dick*.



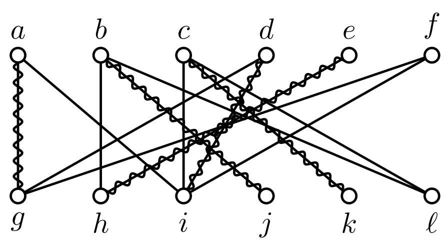


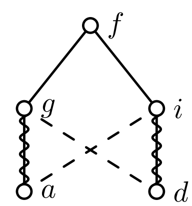
Augment, construct new search forest F, contains M-aug. path *ehbj*.



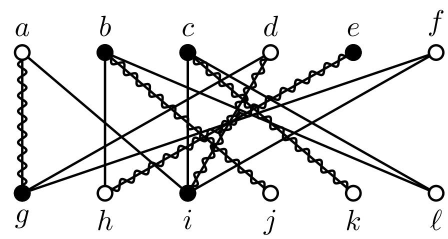


Augment, construct new search forest F. (Dashed edges go to an already used vertex.) Now no M-aug. path.



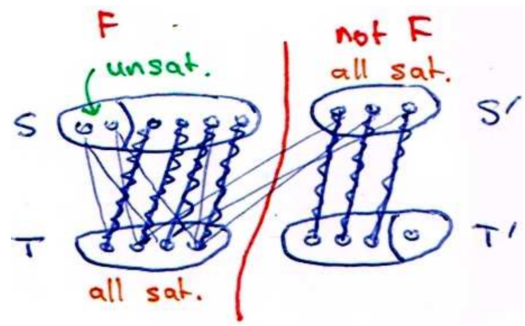


So current matching. size 4, is maximum M^*. Min. vertex cover K^* is S' (X not in F) $\cup$ T (Y in F) $= \{b, c, e\} \cup \{g, i\} = \{b, c, e, g, i\}$. Notice every edge of M^* has exactly one end in K^*.



Final situation: $S = X \cap V(F)$, $S' = X - S$, $T = Y \cap V(F)$, $T' = Y - T$.

- In S get precisely (a) all M-unsat. vertices of X, and (b) all ends of M-edges from T.
- In T get all ends of non-M-edges from S.
- So in S' all vertices are M-sat., by an M-edge from T'.
- Since no M-aug. path, all vertices of T are M-sat.
- So no M- or non-M-edges in $[S,T']$; only non-M-edges in $[S',T]$. So $K = S' \cup T$ is a vertex cover covering every edge of M exactly once: $|K| = |M|$.



Matching algorithm for general graphs: Egerváry's algorithm can be modified to deal with issues caused by odd cycles. Result is *Edmond's algorithm* using *blossoms*.