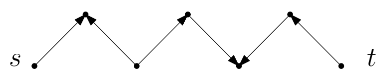
NDMI011: Combinatorics and Graph Theory 1

Lecture #7
Applications of networks

Irena Penev

1 The Ford-Fulkerson algorithm (again)

Recall that an (s,t)-*path* in a network (G, s, t, c) is a sequence $v_0, v_1, \dots, v_\ell$ of vertices of G such that $v_0 = s$, $v_\ell = t$, and for all $i \in \{0, \dots, \ell - 1\}$, we have that one of (v_i, v_{i+1}) and (v_{i+1}, v_i) belongs to $E(G)$. Note that an (s,t)-path may, but need not be, a directed (s,t)-path (see the figure below for an example).



Given a flow f in the network (G, s, t, c), an (s,t)-path $v_0, v_1, \dots, v_\ell$ in (G, s, t, c) is said to be an f-*augmenting path* if the following two conditions are satisfied (see Figure 1.1 for an example):

- for all $i \in \{1, \dots, \ell - 1\}$ such that $(v_i, v_{i+1}) \in E(G)$, we have that $f(v_i, v_{i+1}) < c(v_i, v_{i+1})$;
- for all $i \in \{1, \dots, \ell - 1\}$ such that $(v_{i+1}, v_i) \in E(G)$, we have that $f(v_{i+1}, v_i) > 0$.

In Lecture 6, we saw how, given a flow f in a network (G, s, t, c), one can either find an f-augmenting path, or determine that one does not exist. If one does not exist, then Lemma 2.5 from Lecture Notes 6 guarantees that the flow f is maximum.

The Ford-Fulkerson algorithm is the following. Its input is a network (G, s, t, c), and it proceeds as follows:

1. Set $f(e) := 0$ for all $e \in E(G)$.
2. While there exists an f-augmenting path in the network:

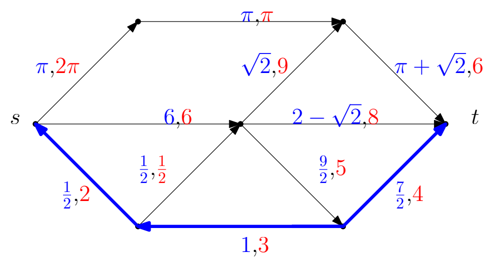


Figure 1.1: An f-augmenting path (edges in blue) in a network (G, s, t, c). (Flow is in blue and capacities are in red.)

(a) Find an f-augmenting path $v_0, \dots, v_\ell$ (with $v_0 = s$ and $v_\ell = t$).

(b) Set

- $\varepsilon_1 = \min\Big(\{c(v_i, v_{i+1}) - f(v_i, v_{i+1}) \mid 0 \le i \le \ell - 1, (v_i, v_{i+1}) \in E(G)\} \cup \{\infty\}\Big)$;
- $\varepsilon_2 = \min\Big(\{f(v_{i+1}, v_i) \mid 0 \le i \le \ell - 1, (v_{i+1}, v_i) \in E(G)\} \cup \{\infty\}\Big)$;
- $\varepsilon = \min\{\varepsilon_1, \varepsilon_2\}$.

(c) Update f as follows:

- $f(v_i, v_{i+1}) := f(v_i, v_{i+1}) + \varepsilon$ for all $i \in \{0, \dots, \ell - 1\}$ such that $(v_i, v_{i+1}) \in E(G)$;[1]
- $f(v_{i+1}, v_i) := f(v_{i+1}, v_i) - \varepsilon$ for all $i \in \{0, \dots, \ell - 1\}$ such that $(v_{i+1}, v_i) \in E(G)$.[2]

3. Return f.

In the previous lecture, we did not actually prove the correctness of the algorithm. For correctness, we would need the following two properties:

(1) the algorithm terminates for every input network (G, s, t, c);

(2) if, given an input network (G, s, t, c), the algorithm returns a flow f, then f is indeed a maximum flow in (G, s, t, c).

It is easy to see that (2) is satisfied. Indeed, the algorithm returns f only if there is no f-augmenting path in the input network (G, s, t, c), and in this case, Lemma 2.5 from Lecture Notes 6 guarantees that the f is a maximum flow in (G, s, t, c). Unfortunately, (1) may fail (we give an example at the

[1] So, for edges on our augmenting path directed with the flow, we increase the flow by ε.

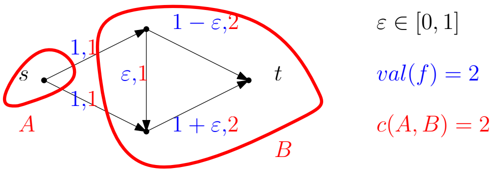
[2] So, for edges on our augmenting path directed against the flow, we decrease the flow by ε.

end of the section). The good news is that this is only possible if some of the capacities in the network are irrational. If all capacities are rational, then the algorithm terminates and correctly outputs a maximum flow. We first deal with the case when the capacities are integers.

Theorem 1.1. *Let (G, s, t, c) be a network in which all capacities are non-negative integers. Then, for input (G, s, t, c), the Ford-Fulkerson algorithm terminates and outputs a maximum flow, and furthermore, the output flow through each edge is a non-negative integer. In particular, some maximum flow in (G, s, t, c) has the property that flows through all edges are non-negative integers.*

Proof. If we begin with an integer flow (i.e. a flow f such that $f(e)$ is an integer for each edge e in our network) in the network (G, s, t, c), and we find an augmenting path, then since all capacities are integers, the number ε (defined as in the description of the Ford-Fulkerson algorithm) will be a positive integer; so, the updated flow will still be an integer flow, since the flow through an edge can either remain unchanged, or increase by ε, or decrease by ε. Now, the initial flow created by the Ford-Fulkerson algorithm for the network (G, s, t, c) is the zero-flow (and so in particular, an integer flow), and by what we just proved, after each iteration, the new flow is still an integer flow. The algorithm terminates because after each iteration, the value of the flow increases by a positive integer (namely, by the ε that we compute for that iteration), and the maximum value of the flow is bounded (e.g. by the sum of capacities), and so there can be only finitely many iterations. The fact that the algorithm returns a correct answer follows from its stopping criterion: the algorithm terminates and returns a flow f once there are no f-augmenting paths, and in this case, Lemma 2.5 from Lecture Notes 6 guarantees that f is a maximum flow. $\square$

Note that Theorem 1.1 does **not** state that every maximum flow in a network with integer capacities is an integer flow. It merely guarantees that at least one maximum flow in such a network is an integer flow.[3] For instance, the flow in the picture below is maximum for any value of $\varepsilon \in [0, 1]$, but only two values of ε (namely, $\varepsilon = 0$ and $\varepsilon = 1$) yield an integer flow.



[3]While the maximum value of a flow in a network is unique, there may be many (possibly infinitely many) flows in the network that have that value, and by definition, all such flows are maximum.

Theorem 1.1 is important for certain theoretical applications (see section 2 for an example), as well for certain practical applications.[4]

If we replace the word "integer" by the word "rational" in the statemnent of Theorem 1.1, we still get a correct statement.

Theorem 1.2. *Let (G, s, t, c) be a network in which all capacities are non-negative rational numbers. Then, for input (G, s, t, c), the Ford-Fulkerson algorithm terminates and outputs a maximum flow, and furthermore, the output flow through each edge is an non-negative rational number. In particular, some maximum flow in (G, s, t, c) has the property that flows through all edges are non-negative rational numbers.*

Proof. Let d be a positive integer such that all capacities in (G, s, t, c) are integer multiples of $\frac{1}{d}$.[5] Now the proof is completely analogous to that of Theorem 1.1, except that instead of integers, we have integer multiples of $\frac{1}{d}$ (for flows and capacities) throughout.[6] $\square$

The key point of the proof of Theorem 1.2 is that there exists some positive integer d such that in each iteration, the value of the flow increases by at least $\frac{1}{d}$, and so there cannot be infinitely many iterations. If (some of) our capacities are irrational, such a d need not exist. Let us give an example of this.[7] First, let $r = \frac{-1+\sqrt{5}}{2}$, and let the sequence $\{r_n\}_{n=0}^{\infty}$ be defined recursively as follows:

- $r_0 = 1$ and $r_1 = r$;
- $r_{n+2} = r_n - r_{n+1}$ for all integers $n \geq 0$.

It is easy to check that $r_n = r^n$ for all integers $n \geq 0$.[8] Let M be some large number (say, $M = 100$). We now consider the network flow below.

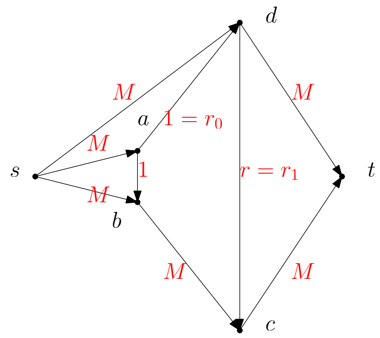
[4]Consider, for example, a network that models a transportation network of trucks, where the capacity of a truck is the number of containers that it can carry. Certainly, we would want a maximum flow that is an integer flow. (A truck should not transport $\frac{7}{3}$ or $\sqrt[3]{\pi}$ containers!)

[5]To see that d exists, we can first write all capacities in (G, s, t, c) as fractions, and then we take d to be the least common multiple of the denominators of the capacities.

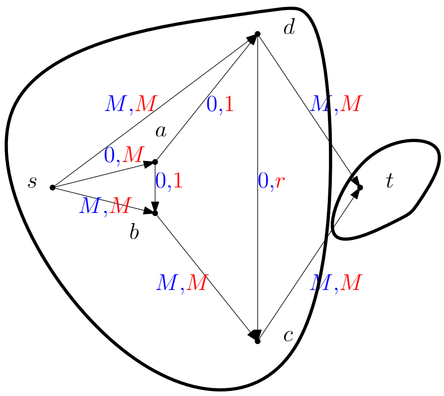
[6]Check this!

[7]We give only describe the construction. If you'd like a challenge, prove that it actually works. (It's a slightly messy induction.)

[8]This formula can be obtained using, for example, generating functions. Correctness is easily verified by induction.

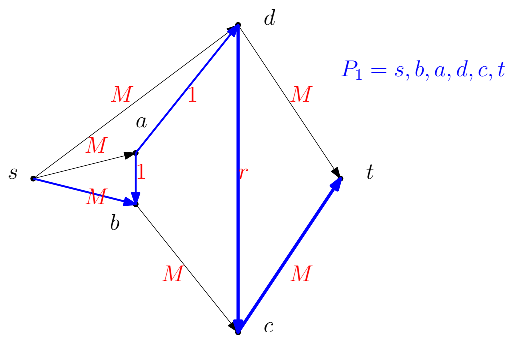


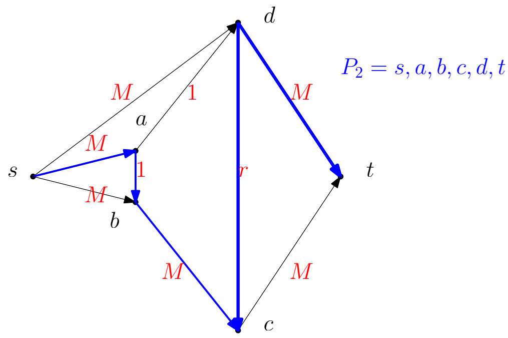
The maximum value of a flow in this network is $2M$, as certified by the flow represented below, and the cut $(\{s, a, b, c, d\}, \{t\})$ of capacity $2M$.

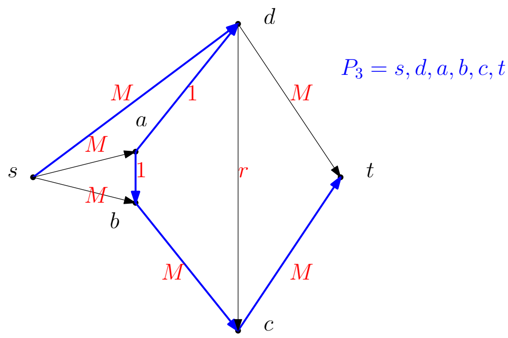


We note that the flow above can easily be obtained in two iterations of the Ford-Fulkerson algorithm: we start with the zero flow, then we choose the augmenting path s, d, t (with $\varepsilon = M$), and then we choose the augmenting path s, b, c, t (again with $\varepsilon = M$). However, if we choose "bad" augmenting paths, the algorithm may continue forever, as we describe below.

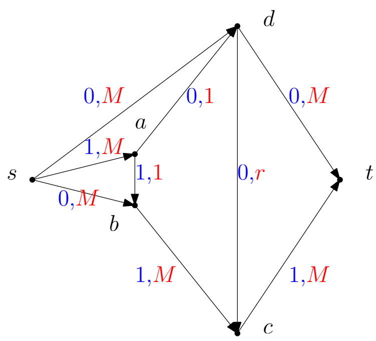
Let P_1 be the s,t-path s, b, a, d, c, t; let P_2 be the s,t-path s, a, b, c, d, t; and let P_3 be the s,t-path s, d, a, b, c, t.







We start with the zero flow f_0, and then we use the augmenting path s, a, b, c, t (with $\varepsilon = 1$), thus obtaining the flow f_1, represented below.



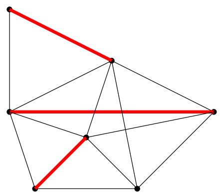
From now on, we cyclically select augmenting paths P_1, P_2, P_3. It can be then shown by induction that the algorithm never terminates,[9] and furthermore, the value of the flows that the algorithm produces converges to $1+2\sum_{n=2}^{\infty} r_n = 3$, whereas the maximum flow in our network has value $2M > 3$.[10]

[9]This is, essentially, because ε tends to zero as we keep iterating. Recall that in the case of rational capacities (see Theorem 1.2), we could always find an integer $d \geq 1$ such that in each iteration, we had $\varepsilon \geq \frac{1}{d}$. This need not be the case if (some of) our capacities are irrational.

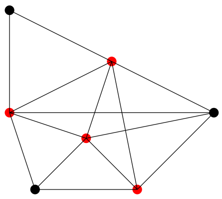
[10]If you want a bit of a challenge, try to prove (by induction) that this is indeed correct.

2 Matchings and transversals

A *matching* in a graph G is a set of edges $M \subseteq E(G)$ such that every vertex of G is incident with at most one edge in M. An example of a matching in a graph is given below (edges of the matching are in red).



A *vertex cover* of a graph G is any set C of vertices of G such that every edge of G has at least one endpoint in C. An example of a vertex cover in a graph is given below (vertices of the vertex cover are in red).



The Kőnig-Egerváry theorem. *The maximum size of a matching in a bipartite graph is equal to the minimum size of a vertex cover in that graph.*

Proof. Let G be a bipartite graph with bipartition (A, B). Clearly, it suffices to prove the following two statements:

(a) for every matching M and every vertex cover C of G, we have that $|M| \leq |C|$;[11]

(b) there exist a matching M and a vertex cover C of G such that $|M| = |C|$.

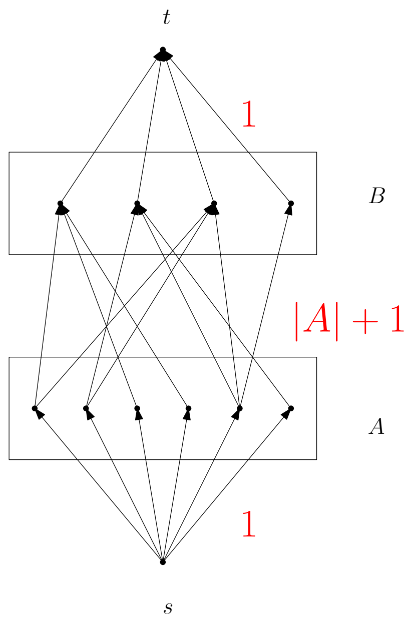
We begin by proving (a). Fix a matching M and a vertex cover C in G. Clearly, every edge of M has at least one endpoint in C. Since no two edges of M share an endpoint, we deduce that $|M| \leq |C|$. This proves (a).

It remains to prove (b). Let s and t be two new vertices, i.e. $s \neq t$ and $s, t \notin V(G)$. We now form a network (G', s, t, c) as follows:

- $V(G') = V(G) \cup \{s, t\}$;
- $E(G') = \{(s, a) \mid a \in A\} \cup \{(a, b) \mid a \in A, b \in B, ab \in E(G)\} \cup \{(b, t) \mid b \in B\}$;

[11] In fact, (a) holds for all graphs, not just bipartite ones. However, there are (non-bipartite) graphs for which (b) fails.

- $c(a, b) = |A| + 1$ for all $(a, b) \in E(G')$, with $a \in A$ and $b \in B$;
- $c(s, a) = 1$ for all $a \in A$;
- $c(b, t) = 1$ for all $b \in B$.



Let f be a maximum flow in (G', s, t, c), and let R be a cut of minimum capacity. By Theorem 1.1, we may assume that $f(e)$ is an integer for all $e \in E(G')$. By the Max-flow min-cut theorem, we know that $val(f) = c(R)$. It now suffices to produce a matching of size $val(f)$ and vertex cover of size $c(R)$.

First, we claim that $f(e) \in \{0, 1\}$ for all $e \in E(G')$. Clearly, it suffices to show that $f(e) \leq 1$ for all $e \in E(G')$.[12] For all $a \in A$, we have that $f(s, a) \leq c(s, a) = 1$; and for all $b \in B$, we have that $f(b, t) \leq c(b, t) = 1$. Now, fix $a \in A$ and $b \in B$ such that $ab \in E(G)$. The inflow into a is at most 1,[13] and so the outflow is at most one. So, $f(a, b) \leq 1$. This proves that $f(e) \in \{0, 1\}$ for all $e \in E(G')$, as we had claimed.

[12] This is because, for all $e \in E(G')$, $f(e)$ is a non-negative integer, and so if $f(e) \leq 1$, then $f(e) \in \{0, 1\}$.

[13] This is because (s, a) is the only edge in G' with head a, and $f(s, a) \leq c(s, a) = 1$.

Now, let $M = \{ab \in E(G) \mid a \in A, b \in B, f(a,b) = 1\}$. Then[14]

$$\begin{aligned} |M| &= |\{(a,b) \in E(G') \mid a \in A, b \in B, f(a,b) = 1\}| \\ &= |\{e \in S_{G'}(A \cup \{s\}, B \cup \{t\}) \mid f(e) = 1\}| \\ &\overset{(*)}{=} f(A \cup \{s\}, B \cup \{t\}) \\ &\overset{(**)}{=} val(f), \end{aligned}$$

where (*) follows from the fact that $f(e) \in \{0,1\}$ for all $e \in E(G)$, and (**) follows from Lemma 2.3 from Lecture Notes 6. Let us check that M is a matching in G. Suppose otherwise. Then one of the following holds:

(i) there exist $a \in A$ and $b_1, b_2 \in B$ (with $b_1 \neq b_2$) such that $ab_1, ab_2 \in M$;

(ii) there exist $a_1, a_2 \in A$ (with $a_1 \neq a_2$) and $b \in B$ such that $a_1b, a_2b \in M$.

Suppose first that (i) holds. Then $f(a,b_1) = f(a,b_2) = 1$, and so the outflow from a is at least 2. On the other hand, the inflow into a is at most 1,[15] a contradiction. Suppose now that (ii) holds. then $f(a_1,b) = f(a_2,b) = 1$, and so the inflow into b is at least 2. On the other hand, the outflow from b is at most 1,[16] a contradiction. This proves that M is indeed a matching.

It remains to produce a vertex cover of size $c(R)$. Let C be the set of all vertices in $V(G) = A \cup B$ that are incident with at least one edge of R. Our goal is to show that C is a vertex cover of size at most $c(R)$. First, note that $\{(s,a) \mid a \in A\}$ is a cut in (G', s, t, c) of capacity $|A|$, and so $c(R) \leq |A|$. Since every edge from A to B has capacity $|A| + 1 > c(R)$, we deduce that R does not contain any edges from A to B; then $R = \{(s,a) \mid a \in A \cap C\} \cup \{(b,t) \mid b \in B \cap C\}$. It follows that

$$\begin{aligned} c(R) &= \Big(\sum_{a \in A \cap C} \underbrace{c(s,a)}_{=1} \Big) + \Big(\sum_{b \in B \cap C} \underbrace{c(b,t)}_{=1} \Big) \\ &= |A \cap C| + |B \cap C| \\ &= |C|. \end{aligned}$$

It remains to show that C is a vertex cover of G. Fix adjacent vertices $a \in A$ and $b \in B$; we must show that at least one of a, b belongs to C. Suppose otherwise. It then follows from the construction of C that R contains none of

[14] $S_{G'}(A \cup \{s\}, B \cup \{t\})$ is the set of all edges from $A \cup \{s\}$ to $B \cup \{t\}$ in the oriented graph G'; note that all edges in $S_{G'}(A \cup \{s\}, B \cup \{t\})$ are in fact from A to B.

[15] This is because (s,a) is the only edge in G' with head a, and $f(s,a) \leq c(s,a) = 1$.

[16] This is because (b,t) is the only edge in G' with tail b, and $f(b,t) \leq c(b,t) = 1$.

the edges (s,a), (a,b), and (b,t) of G', and consequently, s,a,b,t is a directed path from s to t in $G' \setminus R$, contrary to the fact that R is a cut in (G',s,t,c). This proves that C is indeed a vertex cover of G. This completes the proof of (b). $\square$

Given a bipartite graph G with bipartition (A,B),

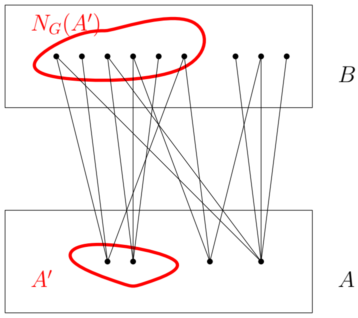
- an A-*saturating matching* in G is a matching M in G such that every vertex of A is incident with some edge in M;
- a B-*saturating matching* in G is a matching M in G such that every vertex of B is incident with some edge in M.

For a graph G and a set $A \subseteq V(G)$, we denote by $N_G(A)$ the set of all vertices in $V(G) \setminus A$ that have a neighbor in A. As a corollary of the Kőnig-Egerváry theorem, we obtain the following.

Hall's theorem (graph theoretic formulation). *Let G be a bipartite graph with bipartition (A,B). Then the following are equivalent:*

(a) all sets $A' \subseteq A$ satisfy $|A'| \leq |N_G(A')|$;

(b) G has an A-saturating matching.



Proof. Suppose first that (b) holds; we must prove that (a) holds. Fix an A-saturating matching M in G, and fix $A' \subseteq A$. Since M is an A-saturating matching, and since A' is a stable set,[17] we know that precisely $|A'|$ edges in M are incident with a vertex in A', and each of those edges has another endpoint in B. No two edges in M share an endpoint, and it follows that exactly $|A'|$ vertices in B are incident with an edge of M that has an endpoint in A'; let B' be the set of all such vertices of B. But clearly, $B' \subseteq N_G(A')$, and so $|N_G(A')| \geq |B'| = |A'|$. This proves (a).

Suppose, conversely, that (a) holds; we must prove that (b) holds. Since all edges of G are between A and B, it suffices to show that G has a matching of size at least $|A|$.[18] By the Kőnig-Egerváry theorem, it is enough to show

[17] A *stable set* (or *independent set*) is a set of pairwise non-adjacent vertices.
[18] Note that any matching in G of size at least $|A|$ is in fact of size precisely $|A|$.

that any vertex cover of G is of size at least $|A|$. Let C be a vertex cover of G. Then there can be no edges between $A \setminus C$ and $B \setminus C$, and we deduce that $N_G(A \setminus C) \subseteq B \cap C$, and consequently, $|N_G(A \setminus C)| \leq |B \cap C|$. Now we have the following:

$$\begin{aligned}
|A| &= |A \cap C| + |A \setminus C| \\
&\leq |A \cap C| + |N_G(A \setminus C)| \qquad \text{by (a)} \\
&\leq |A \cap C| + |B \cap C| \\
&= |C|.
\end{aligned}$$

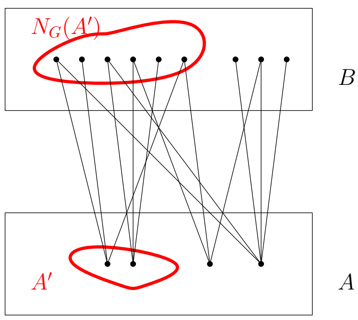
This completes the proof of (b). □

The *degree* of a vertex v in a graph G, denoted by $d_G(v)$, is the number of edges of G that v is incident with.

Corollary 2.1. *Let G be a bipartite graph with bipartition (A, B). Assume that G has at least one edge and that for all $a \in A$ and $b \in B$, we have that $d_G(a) \geq d_G(b)$. Then G has an A-saturating matching.*

Proof. We first check that $d_G(a) \geq 1$ for all $a \in A$. Suppose otherwise, and fix some $a_0 \in A$ such that $d(a_0) = 0$. Now, since G has at least one edge, and since every edge of G has one endpoint in A and the other one in B, we see that some vertex $b_0 \in B$ is incident with at least one edge, and so $d_G(b_0) \geq 1$. But now $d_G(a_0) < d_G(b_0)$, a contradiction. This proves that $d_G(a) \geq 1$ for all $a \in A$, as we had claimed.

Now, suppose that G does not have an A-saturating matching. Then by Hall's theorem, there exists some $A' \subseteq A$ such that $|A'| > |N_G(A')|$.



Note that every edge in G has at least one endpoint in $(A \setminus A') \cup N_G(A')$,[19]

[19]Indeed, if some edge of G had neither endpoint in $(A \setminus A') \cup N_G(A')$, then one of its endpoints would be in A' and the other one would be in $B \setminus N_G(A')$, a contradiction.

and so

$$\begin{aligned}|E(G)| &\leq \sum_{v \in (A \setminus A') \cup N_G(A')} d_G(v) \\ &\leq \Big(\sum_{a \in A \setminus A'} d_G(a) \Big) + \Big(\sum_{b \in N_G(A')} d_G(b) \Big).\end{aligned}$$

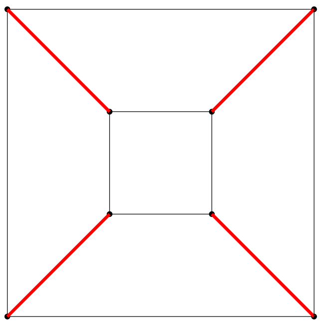
Now, since $A' \subseteq A$ and $N_G(A') \subseteq B$, we know that for all $a \in A'$ and $b \in N_G(A')$, we have that $d_G(a) \geq d_G(b)$. Furthermore, by our choice of A', we have that $|A'| > |N_G(A')|$. Since $d_G(a) \geq 1$ for all $a \in A$, we now deduce that $\sum\limits_{a \in A'} d_G(a) > \sum\limits_{b \in N_G(A')} d_G(b)$, and it follows that

$$\begin{aligned}|E(G)| &\leq \Big(\sum_{a \in A \setminus A'} d_G(a) \Big) + \Big(\sum_{b \in N_G(A')} d_G(b) \Big). \\ &< \Big(\sum_{a \in A \setminus A'} d_G(a) \Big) + \Big(\sum_{a \in A'} d_G(a) \Big) \\ &= \sum_{a \in A} d_G(a).\end{aligned}$$

But this is impossible since, obviously, $|E(G)| = \sum\limits_{a \in A} d_G(a)$. □

For a non-negative integer k, a graph G is *k-regular* if it all its vertices are of degree k. G is *regular* if there exists some non-negative integer k such that G is k-regular.

A *perfect matching* in a graph G is a matching M such that every vertex of G is incident with an edge in M. An example of a perfect matching is shown below (edges of the perfect matching are in red).



Obviously, not all graphs have perfect matchings. For instance, no graph with an odd number of vertices has a perfect matching. (There are also many graphs that have an even number of vertices, and yet do not have a perfect matching.)

Corollary 2.2. *Every regular bipartite graph that has at least one edge has a perfect matching.*

Proof. Let G be a k-regular ($k \geq 0$) bipartite graph with bipartition (A, B), and assume that G has at least one edge. By Corollary 2.1, G has an A-saturating matching. Now, since G has at least one edge, we see that $k \geq 1$. Further, since G is k-regular, we have that $|E(G)| = k|A|$ and $|E(G)| = k|B|$, and so $k|A| = k|B|$; since $k \neq 0$, it follows that $|A| = |B|$. Consequently, any A-saturating matching of G is a perfect matching. Since G has an A-saturating matching, it follows that G has a perfect matching. $\square$

For a graph G, and a set $S \subsetneqq V(G)$, let $\mathrm{odd}_G(S)$ be the number of odd components (i.e. components with an odd number of vertices) of the graph $G \setminus S$. The following theorem gives a necessary and sufficient condition for a graph to have a perfect matching.

Tutte's theorem. *Let G be a graph. Then the following are equivalent:*

(a) for all sets $S \subsetneqq V(G)$, we have that $odd_G(S) \leq |S|$;

(b) G has a perfect matching.

Proof. Omitted. $\square$

We complete this section by giving another formulation of Hall's theorem. We first need a definition. Suppose X and I are sets, and $\{A_i\}_{i \in I}$ is a family of (not necessarily distinct) subsets of X.[20] A *transversal* (or a *system of distinct representatives*) for $(X, \{A_i\}_{i \in I})$ is an injective (i.e. one-to-one) function $f : I \to X$ such that for all $i \in I$, we have that $f(i) \in A_i$.

Hall's theorem (combinatorial formulation). *Let X and I be finite sets, and let $\{A_i\}_{i \in I}$ be a family of (not necessarily distinct) subsets of X. Then the following are equivalent:*

(a) all sets $J \subseteq I$ satisfy $|J| \leq |\bigcup_{j \in J} A_j|$;

(b) $(X, \{A_i\}_{i \in I})$ has a transversal.

Proof. HW. $\square$

3 Extending Latin rectangles

For positive integers r and n, with $r \leq n$, an $r \times n$ *Latin rectangle* is an $r \times n$ array (or matrix) whose entries are numbers $1, \dots, n$, and in which each number $1, \dots, n$ occurs at most once in each row and each column. One 2×4 Latin rectangle is represented below.

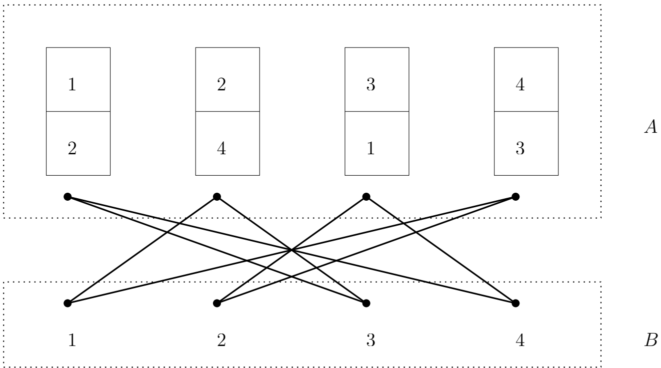
[20] Technically, we have that $A : I \to \mathscr{P}(X)$; for $i \in I$, we write A_i instead of $A(i)$.

1	2	3	4
2	4	1	3

Theorem 3.1. *Let r and n be positive integers such that $r < n$. Then every $r \times n$ Latin rectangle can be extended to an $n \times n$ Latin square.*[21]

Proof. Let $L = \begin{bmatrix} \mathbf{a}_1 & \dots & \mathbf{a}_n \end{bmatrix}$ be an $r \times n$ Latin rectangle.[22] Obviously, it suffices to show that we can extend L to an $(r+1) \times n$ Latin rectangle by adding a row of length n to the bottom of L, for then the result will follow immediately by an easy induction.

Let $A = \{\mathbf{a}_1, \dots, \mathbf{a}_n\}$ and $B = \{1, \dots, n\}$, and let G be the bipartite graph with bipartition (A, B) in which $\mathbf{a}_i \in A$ and $j \in B$ are adjacent if and only if j is not an entry of the column $\mathbf{a}_i$. For instance, for the Latin rectangle from the beginning of the section, we would get the following bipartite graph:



Each column of L has r entries, and consequently, there are $n - r$ values in B that do not appear in it. So, for all $\mathbf{a}_i \in A$, we have that $d_G(\mathbf{a}_i) = n - r$. Now, fix $j \in B$. We know that j appears exactly once in each row of L, and L has r rows. Consequently, j appears exactly r times in L, and since it cannot appear more than once in any column, we see that it appears in precisely r columns of L. Thus, j fails to appear in precisely $n - r$ columns of L, and consequently, $d_G(j) = n - r$. We have now shown that is $(n - r)$-regular. So, G is a regular bipartite graph, and (since $r < n$) it has at least one edge. Corollary 2.2 now implies that G has a perfect matching, call it M. Now, for each $i \in \{1, \dots, n\}$, let j_i be the (unique) element of $\{1, \dots, n\}$ such that $\mathbf{a}_i j_i \in M$. We now add the row $\begin{bmatrix} j_1 & \dots & j_n \end{bmatrix}$ to the bottom of L, and we thus obtain an $(r+1) \times n$ Latin square, which is what we needed. $\square$

[21] This means that, for any $r \times n$ Latin rectangle, it is possible to add $n - r$ rows of length n to the bottom of the Latin rectangle that we started with and thus obtain an $n \times n$ Latin square.

[22] This means that $\mathbf{a}_1, \dots, \mathbf{a}_n$ are the columns of our Latin rectangle, in that order.