Matchings and flows
ICTS-RRI Maths Circle, Bengaluru
10, 24 May 2025, 10:00 am to 1:00 pm

Session III: 10 May 2025

The questions we discussed in this session were listed at the end of the previous exploration sheet. The following section contains a report on what we did on 10 May. If you want to know what we will do on the 24th, please jump straight to the section named *Flows in networks.*

Report on the 10 May session

We started by discussing the relationship between maximum matchings and augmenting paths in graphs that are not necessarily bipartite. So I asked them what an augmenting path in a graph with a matching might be? This was not difficult. They said we can still say what we are used to saying: it is an alternating path that starts at a free vertex and ends at another free vertex. I wrote the following on the board.

> (Berge) A matching M is a maximum sized matching in a graph G if and only if there is no augmenting path in G wrt M.

I tried to impose on them the perennial confusion: what is *if* and what is *only if*? Suppose we have two statements statement A and statement B, and we say "A *if* B". What do we mean? And if we say "A *only if* B"? So "A *if*" means A is true whenever B is true, that is, "B implies A". And "A only if B" stands for "A implies B". With that out of the way, let us ask, which part of Berge's theorem is obvious (the *if* part, or the *only if* part). They said it is obvious that if M is a maximum sized matching then there cannot be an augmenting path in G wrt M. But is this part *if* or *only if*? Answer (confusingly): the *only if* part. What about the *if* part? Assume that M is not a maximum sized matching in G, then why must there be an augmenting path. This is not completely straightforward. Where does one start? I let them think this over. The thing is that it is tempting to go around looking for free vertices and augmenting paths that start from them (this worked in bipartite graphs). I gave them a hint. So, let us call M the red matching. What does it mean to say that it is not the maximum sized matching? Ans: There is a another matching M' that has more edges than M. I suggested that we call it the *blue* matching. The edges are blue or red (and some are both, someone suggested we call them purple); there are more blue edges than red. Ignore everything but the edges in the two matchings. What does it look like? Every vertex has degree 0, 1 or 2—alternating paths and alternating cycles. Can they all be cycles? No, there are more blue edges than red. Again, there was a temptation to go looking for free vertices. What would be an augmenting path in the graph that remains? Oh, the path must start and end with blue edges. Why must there be such a path? For the same reason: there are more blue edges than red. Are we done? Yes! Was this the *if* direction or the *only if* direction? ... never mind!

I reminded them of the game we ended with last time: two players take turns to mark vertices in the given graph. The rules: the first player can start anywhere; then onwards, each players must choose an unmarked vertex that neighbours the vertex that was just marked by their opponent. The player who cannot find such a vertex loses.

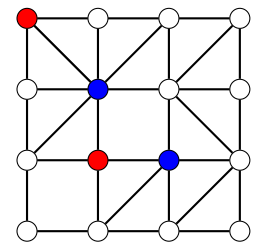

Figure 1: The players take turns to grow a path

Claim: Player II wins if and only if the graph has perfect matching.

Immediately: the only if direction is obvious. If the graph has perect matching, then Player II can keep picking the vertex that is matched to the vertex Player I picks. So I played (as Player I) the game on a graph similar to the one in fig. 1, and lost. What if the graph has no perfect matching? Now, we need to figure out how Player I can win. Where must he start? Is there a special vertex in the graph? Who knows. A triangle does not have a perfect matching, but no vertex in it is special in any sense. Hint: Fix a maximum sized matching M. What do we know? Ans: there is free vertex. And? ... there is no augmenting path. I left them to figure this out. They did.

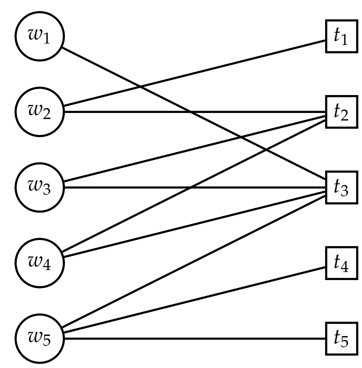

Figure 2: The graph for the office

Bipartite graphs. Moving ahead to where we began: workers and tasks in an office (fig. 2). Could we get more work done if workers were allowed to split their work day between task? When written with number, this amounts to assigning a non-negative weight x_e to each edge e so that the sum of the weights of the edges incident on each vertex is at most 1; to maximize the amount of work done, we must find an assignment such that the sum of the x_e is as large as possible. That is, for graph $G = (A, B, E)$ we have the following optimization problem (where we have to find an assignment x_e for each edge).

$$\begin{aligned} \underset{(x_e : e \in E)}{\text{maximize}} \quad & \sum_{v \in V} x_e \\ \text{subject to} \quad & \sum_{e: e \text{ is incident on } v} x_e \le 1, \quad \text{for all } v \in A \cup B, \\ & 0 \le x_e \quad \text{for all } e \in E. \end{aligned}$$

After I put down this formulation, I put down a bipartite graph where the size of the maximum matching was 4, but there were five vertices on both sides. So there were 5 workers and 5 tasks, but the skills were such that only four of the jobs could ever be done on any day. I asked them to come up with a fractional assignment where more than four jobs could be done. They tried for some time, and concluded that it was not possible in the given graph. Their argument essentially established that

$$\max \mathsf{size}(\text{fractional matching}) \leq |\text{Minimum vertex cover}|,$$

and since the $|\text{Maximum matching}| = |\text{Minimum vertex cover}|$, they concluded that we gain nothing by asking workers to multitask. Having just discussed fractional matchings, I asked them if there was any other factional quantity they could imagine. The Kőnig-Egerváry theorem was on the board. Soon enough they asked if there was something called a fractional vertex cover as well. Indeed! I wrote down

$$\begin{aligned} \text{minimize} \quad & \sum_{v \in V} y_v \\ \text{subject to} \quad & \sum_{v:e \text{ is incident on } v} y_v \leq 1, \quad \text{for all } e \in E, \\ & 0 \leq y_v \quad \text{for all } v \in V. \end{aligned}$$

I told them that we could think of y_v is the amount of water that needs to be stored at the intersections of roads so that the amount of water at the two ends of each street is adequate to fight any fire in the street (the needs of all the streets are covered). They realized that

$$|\text{Maximum matching}| \leq \max \mathsf{size}(\text{fractional vertex cover}),$$

and they saw that (thanks to the Kőnig-Egervary theorem) moving to fractions did not help us get a smaller vertex cover. I finally got to write down

$$\begin{aligned} |\text{Maximum matching}| &\leq \max \mathsf{size}(\text{fractional matching}) && (1) \\ &\leq \min \mathsf{size}(\text{fractional vertex cover}) && (2) \\ &\leq |\text{Minimum cover}|. && (3) \end{aligned}$$

(I told them that the second inequality too was within their reach, but did not wait for them to prove it.) By now everybody understood that the Kőnig-Egerváry theorem tells us that the quantities at the extreme left and the extreme right are equal, so all four quantities that appear in this sequence of inequalities are the same!

Back to general graphs. The arguments we used to establish that inequalities in eq. (1), eq. (2), eq. (3) hold for all graphs, but we don't have the equivalent of the Kőnig-Egerváry theorem to collapse everything into one. Let us consider a triangle. We computed the four quantities:

$$|\text{Maximum matching}| = 1 \tag{4}$$
$$\min \text{size}(\text{fractional vertex cover}) = 1.5 \tag{5}$$
$$\max \text{size}(\text{fractional matching}) = 1.5 \tag{6}$$
$$|\text{Minimum cover}| = 2. \tag{7}$$

Here the extreme left and the extreme right are apart by a factor of 2, but (i) the two quantitites in the middle are equal, and (as one of them observed), (ii) the two quantities in the middle are the average of the ones at the two extremes. I told them that the first (i) holds for all graphs and is a deep fact; on the other hand, I said, (ii) was just a coincidence, but I gave no example where (ii) does not hold (e.g., 4 vertices with all 6 edges, |maximum matching|=2, |min vertex cover|=3, size(max fractional matching)=size(min fractional cover)=2).

Bipartite graphs again. Finaly, we asked once again how we had found augmenting paths in bipartite graphs in order to increase the size of a given matching. The matching edges were directed from left to right, the non-matching edges were directed right to left, and then we tried to find a path from a free vertex on the left to a free vertex on the right. I asked if we could set this up as a physics experiment. Someone suggested placing pipes along edges and pushing water through them: it was hard to imagine pipes that would allow water to travel in only one direction. Someone suggested wires carrying current: this too has the same problem . . . but we place diodes on the wires and hook up a battery and a light bulb across a free vertex on the left and a free vertex on the right? Clearly, the bulb lights up iff there is an augmenting path. When we try to imagine how the electrical current might have found its way through the circuit, we see that the process exactly mirrors the Hungarian exploration . . . the exploration imitates electrical flows which was set up to imitate our matematical logic!

Session IV: 24 May 2025

Flows in networks

Many of the ideas we encountered while discussing matchings in bipartite graphs generalize to flows in networks. To begin with, let us reinterpret the idea of maximum matching using network flows.

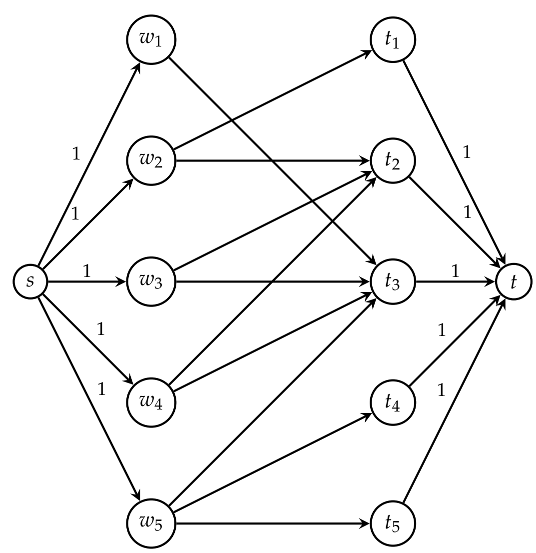

Figure 3: The network-flow graph for the office (the edges in the middle have infinite capacity)

We obtain the network in fig. 3. The quantities on the edges denote capacities (don't be alarmed by some edges with infinite capacity).

Exploration: flow versus cut

So here we have a network with directed edges where each edge e has a capacity $c_e \geq 0$. We wish to set up a flow in this network. What does that mean? To each edge we wish to assign a number f_e satisfying the following.

The capacity constraint: The flow on an edge should not exceed the edge's capacity, that is, for each edge e in the network

$$0 \leq f_e \leq c_e.$$

A *valid* flow must satisfy the capacity and conservation constraints.

Flow conservation: The idea is that at all vertices other than the source s and sink t, flow must be conserved. How should we write this down? Let $E^+(v)$ denote the edges entering v (these edges bring flow into v) and $E^-(v)$ denote the edges leaving v. Fill in the blanks below. (Note that we do not impose any conservation constraint for s and t.)

$$\sum_{\underline{\quad ? \quad}} f_e = \sum_{\underline{\quad ? \quad}} f_e \quad \text{(for all } v \in V \setminus \{s,t\}). \tag{8}$$

We say that a flow is *valid* if it obeys the capacity and conservation constraints.

Value of the flow: Let f denote a flow (so if $e_1, e_2, \ldots, e_m$ are the edges of the network, then you may think of f as m non-negative numbers $(f_{e_1}, f_{e_2}, \ldots, f_{e_m})$). The value of flow is the net flow leaving the source s, that is,

$$\mathsf{val}(f) = \sum_{e \in E^-(s)} f_e - \sum_{e \in E^+(s)} f_e.$$

The following statement should be intuitively obvious. How would you formally prove this using eq. (8)?

The net flow leaving s is equal to the net flow entering t.

So we could as well write

$$\mathsf{val}(f) = \sum_{e \in E^+(t)} f_e - \sum_{e \in E^-(t)} f_e.$$

Notice that we swapped the positions of E^+ and E^- in this case.

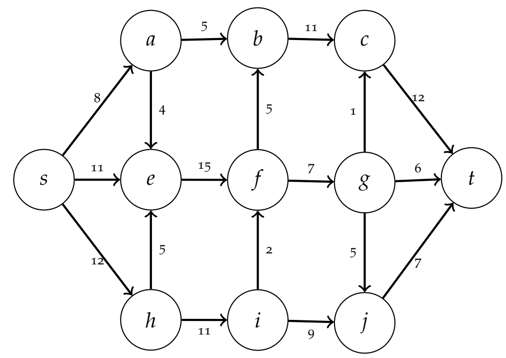

Figure 4: The values on the edges are the capacities.

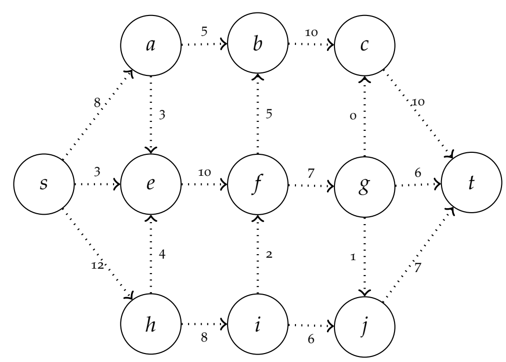

Figure 5: Here is a valid flow for the above network. What is the value of the flow? Is the net flow leaving s equal to the net flow entering t?

Cut: An s-t cut in a network is a set C of edges of the network that causes t to become unreachable from s. The capacity of a cut is the sum of the capacities of the edges in the cut:

$$\mathsf{cap}(C) = \sum_{e \in C} c_e.$$

Can you find a cut of capacity 4 in fig. 3? What about fig. 5? You should think of C as as a bottleneck in network; intuitively, if the bottleneck has small capacity then we ought not to be able to push much flow from s to to t. Can you derive the following from eq. (8)?

Suppose f is a valid flow and C is an s-t cut. Then,

$$\mathsf{val}(f) \leq \mathsf{cap}(C). \tag{9}$$

Hint: Remove the edges in C from G, and let S be the set of vertices reachable from S (of course, $s \in S$ and $t \notin S$). What do the equations in eq. (8) corresponding to $v \in S \setminus \{s\}$ say?

In the flow of network fig. 5, can we push additional flow along the route s, e, f, i, j, g, c, t. Note that (f, i) and (j, g) are not edges in the original graph; so you will have to think about cancelling some flow along them, followed by a little *adjust madi!*

Max-flow and min-cut

The primary goal of modeling problems using flows in networks is to consider valid flows of maximum value. Let $\mathsf{maxflow}(G)$ represent the maximum value of a valid flow in G. Similarly, let $\mathsf{mincut}(G)$ represent the minimum capacity of an s-t cut in G. What does eq. (9) tell us about the relationship between $\mathsf{maxflow}(G)$ and $\mathsf{mincut}(G)$?

In the max-flow for the network of figure fig. 3, what is min-cut represent? In fig. 5, using trial and error, compute the max-flow and the min-cut. How can you be sure that your answer is correct? We will develop an algorithm to find the maximum flow in networks. As in the case of matching in bipartite graphs, this will involve repeatedly improving the flow using *augmentation*.

Exploration: The matrix rounding problem

Consider the following data on the number (in millions) of doses of vaccines supplied to various states (this is not real data, I made it up). [1]

	Covishield	Covaxin	Sputnik	Moderna	Total
Assam	1.5857	2.6348	1.7678	3.9568	9.9451
Bihar	1.3566	2.6251	1.3340	1.2731	6.5889
Chattisgarh	5.5412	4.6178	6.1583	2.5724	18.8897
Total	8.4835	9.8777	9.2602	7.8023	

[1] In August 2021, I was asked to give a talk to undergraduate students. I chose to speak on flows in networks, using the transport of vaccines as my example. The slides from that talk are available here [click for PDF].

A newspaper wishes to publish this information, but it wants to whole numbers. It tries rounding up all figures so that information about the number of doses of each vaccine supplied to each state are close to the actual figure.

Round up	Covishield	Covaxin	Sputnik	Moderna	Total
Assam	2	3	2	4	11
Bihar	2	3	2	2	9
Chattisgarh	6	5	7	3	21
Total	10	11	11	9	

Rounding all entries up does not work. The trouble is that last row and the last column are not always close to the original.

Try rounding all entries down. Does that work?

The general matrix rounding problem: We are given a matrix of real numbers. We wish to round each entry to an integer.

- Every entry in the new table must differ from the corresponding entry of the original table by *less than* one. So 1.1 can be rounded to 2 or 1, but 1.0 can be rounded only to 1.
- The rounding should be done in such a way that all row sums and column sums also differ from their corresponding original values by less than 1.

Try to round the vaccine data above so that it satisfies the above conditions.

It was not a matter of luck that you were able to consistently round the vaccine data. We will see a theorem that says this can always be done. It has to do with network flows! The network below somehow represents the problem we have to solve. Flows are injected at various nodes and they are being sucked out at other nodes. Can you guess what they represent? The values on the edges are the actual flows; each edge has capacity 1 except the edge at the bottom which has a large capacity say 100. What we wish to do with this network may not be clear now, but give it a try. We will talk about it!