Stable Marriage: Matchings and Arborescences

Hrishik Koley

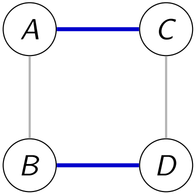
LIMIT Online Camp 2026

What is a matching?

Idea

A matching is a set of edges in a graph such that no two chosen edges touch the same vertex.

- Think of pairing people with projects, teams with rooms, or cards with envelopes.
- Each object can be used in at most one pair.
- The blue edges below form a matching.

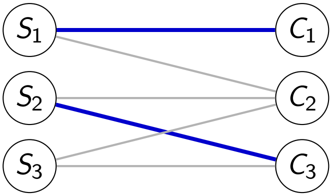


Matchings in bipartite graphs

Bipartite setting

Many matching problems have two groups: a left side and a right side. Edges show which pairs are allowed.

- Left side: students.
- Right side: clubs.
- An edge means a student is interested in that club.
- A matching chooses non-conflicting assignments.



Complete matching

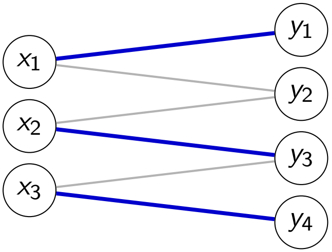
Formal definition

Let $G = (X \cup Y, E)$ be a bipartite graph. A matching $M \subseteq E$ is complete from X to Y if

$$\forall x \in X, \ \exists y \in Y \text{ such that } xy \in M.$$

Since M is a matching, each vertex of X is incident to exactly one edge of M.

- Complete matching is also called a matching that saturates X.
- Vertices in Y may remain unmatched.
- If $|X| = 3$ and $|Y| = 4$, a complete matching uses exactly 3 edges.

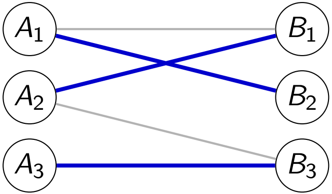


Perfect matching

Definition

A perfect matching matches every vertex in the graph exactly once.

- Nobody is left out on either side.
- In a bipartite graph, both sides must have the same number of vertices.
- Perfect matching is stronger than complete matching.



Hall's Marriage Theorem

Neighbourhood

Let $G = (V, E)$ be a graph. For $S \subseteq V$, define

$$N(S) = \{y \in V : \text{there exists } x \in S \text{ with } xy \in E\}.$$

Neighbourhood in Bipartite Graphs

Let $G = (X \cup Y, E)$ be a graph. For $S \subseteq X$, define

$$N(S) = \{y \in Y : \text{there exists } x \in S \text{ with } xy \in E\}.$$

Theorem

There is a matching that saturates X if and only if $|N(S)| \geq |S| \quad$ for every subset $S \subseteq X$.
If $|X| = |Y|$, a matching that saturates X is a perfect matching.

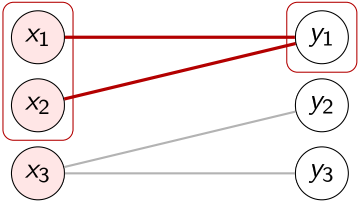
Hall fails: no perfect matching

Violation

Take $S = \{x_1, x_2\}$. Then

$$N(S) = \{y_1\}, \qquad |N(S)| = 1 < 2 = |S|.$$

- The two vertices x_1 and x_2 both have only one possible partner.
- At most one of them can be matched to y_1.
- So there is no complete matching, and hence no perfect matching.

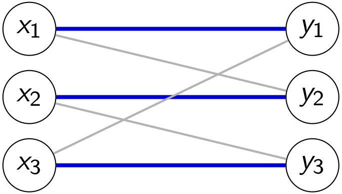


Hall holds: perfect matching exists

Checking the condition

Here $X = \{x_1, x_2, x_3\}$ and $Y = \{y_1, y_2, y_3\}$.

- Each single vertex in X has at least two neighbours.
- Any two vertices in X have all three vertices of Y as neighbours.
- All three vertices in X also have three neighbours.
- Therefore $|N(S)| \geq |S|$ for every $S \subseteq X$.



Blue edges give a perfect matching.

Proof sketch of Hall's theorem

Why Hall's condition is necessary

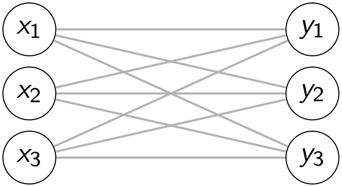
If a matching saturates X, then every set $S \subseteq X$ is matched to $|S|$ distinct vertices in $N(S)$. Hence $|N(S)| \geq |S|$.

Why Hall's condition is sufficient

Start with a largest possible matching. If some $x_0 \in X$ is unmatched, follow alternating paths: first use an unused edge, then a matched edge, and repeat.

- Let S be the left-side vertices reached from x_0.
- All neighbours of S must already be matched back into $S \setminus \{x_0\}$.
- Thus $|N(S)| \leq |S| - 1$, contradicting Hall's condition.
- Therefore no left-side vertex is unmatched; a complete matching exists.

Stable matching problem



Setting

We have two groups X and Y of the same size.
Each vertex ranks all vertices on the other side.

- Example: students and projects pairings.
- Classical Example: girls rank boys, boys rank girls, and we try to find the stable perfectly matched boy-girl pairs.
- We want a perfect matching between X and Y.
- But we also want the matching to be stable.

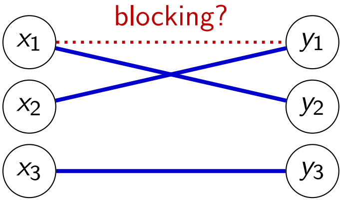
What is a stable matching?

Blocking pair

In a matching M, a pair (x, y) with $xy \notin M$ is a blocking pair if

x prefers y to its partner in M, and y prefers x to its partner in M.

- A matching is stable if it has no blocking pair.
- A blocking pair is a pair that would rather be together than stay with their assigned partners.



Gale–Shapley: deferred acceptance

Algorithm idea

One side, say X, makes proposals. The other side, Y, keeps only its best current proposal and rejects the rest.

1. Start with everyone unmatched.
2. While some $x \in X$ is unmatched and has not proposed to everyone:
   - x proposes to the highest-ranked $y \in Y$ it has not yet proposed to.
   - y tentatively keeps its favourite among its current partner and new proposers, and rejects the rest.
3. When proposals stop, the tentative pairs become the final matching.

Note:

Acceptances are temporary; the ones being proposed are always on the look out for better proposals and may switch later to a better proposer.

Example: preference lists

X proposes

Vertex	Preference order
x_1	$y_1 \succ y_2 \succ y_3$
x_2	$y_1 \succ y_3 \succ y_2$
x_3	$y_2 \succ y_1 \succ y_3$

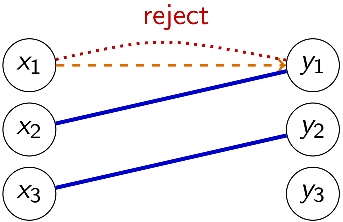
Y responds

Vertex	Preference order
y_1	$x_2 \succ x_1 \succ x_3$
y_2	$x_1 \succ x_3 \succ x_2$
y_3	$x_3 \succ x_2 \succ x_1$

- The symbol $a \succ b$ means "is preferred to."
- We will let x_1, x_2, x_3 propose in this example.

Example: first round of proposals

- x_1 proposes to y_1.
- x_2 proposes to y_1.
- x_3 proposes to y_2.
- y_1 prefers x_2 to x_1, so y_1 keeps x_2 and rejects x_1.
- y_2 keeps x_3 for now.



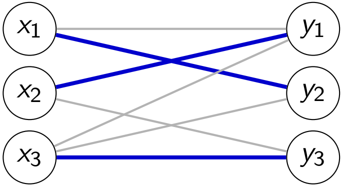
Tentative pairs: $x_2 y_1$ and $x_3 y_2$.

Example: later rounds and result

- x_1 is rejected, so it proposes next to y_2.
- y_2 prefers x_1 to x_3, so y_2 switches to x_1.
- x_3 then tries y_1, but y_1 keeps x_2.
- Finally x_3 proposes to y_3, which accepts.

Final matching

$$M = \{x_1y_2,\ x_2y_1,\ x_3y_3\}.$$



Blue edges are the output of Gale–Shapley.

Why Gale–Shapley is stable

Important monotonicity

Proposers move down their preference lists, while receivers' tentative partners only improve over time.

- Suppose the final matching has a blocking pair (x, y).
- Since x prefers y to its final partner, x must have proposed to y earlier.
- When y rejected x or later dropped x, it chose someone it preferred to x.
- After that, y's tentative partner could only get better.
- Thus y cannot prefer x to its final partner, a contradiction.

Conclusion

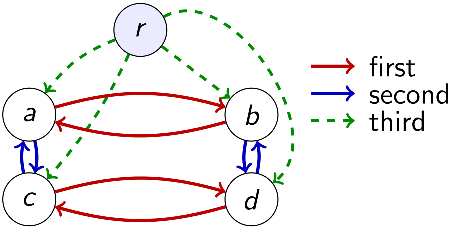
With equal-sized sets and complete preference lists, Gale–Shapley produces a stable perfect matching.

Arborescences: the input graph

Directed graph with a root

Let $G = (V \cup \{r\}, E)$ be a directed graph. The vertex r is the root and has no incoming edge.

- Every vertex $v \in V$ ranks its incoming edges.
- Red, blue, and green show first, second, and third choices.
- We will build directed trees rooted at r.

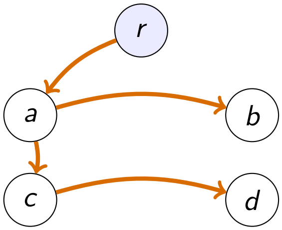


Definition: arborescence

r-arborescence

An r-arborescence is a directed tree rooted at r such that every $v \in V$ has exactly one incoming edge.

- Every vertex is reachable from r.
- Each non-root vertex has one parent.
- A vertex cares only about the incoming edge it receives.



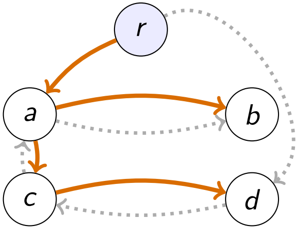
Here a gets its worst edge, c its second choice, and b, d their best edges.

Comparing two arborescences

Voting rule

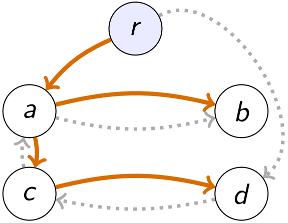
To compare arborescences A and A', each vertex votes for the one where it receives the better incoming edge.

- Solid edges form A.
- Dotted edges form A'.
- More votes means "more popular."



solid: A
dotted: A'

Example: A' is more popular than A



Vertex	Better edge	Vote
a	(c, a) over (r, a)	A'
b	same edge (a, b)	tie
c	(d, c) over (a, c)	A'
d	(c, d) over (r, d)	A

Vote count

A' gets 2 votes and A gets 1 vote.

Conclusion

A' is more popular than A.

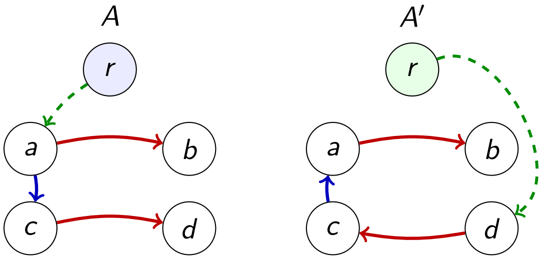
Popular arborescences

Definition

An arborescence A is popular if there is no arborescence A' that is more popular than A, that is majority of the vertices prefer their edges in A over any other arborescence.

- Popularity is a majority-vote idea.
- It is a global tree chosen from local preferences.
- A popular arborescence need not always exist.
- As in the above example $A' \succ A$, that is A' is more popular than A.

Popular arborescence need not exist



Majority comparisons can form cycles, so a popular arborescence need not exist.

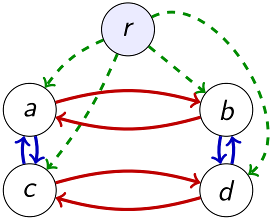
- In a comparison between A and A', a, c prefer A' over A, d prefers A over A', and b remains unbothered. So $A' \succ A$.
- In a comparison between A' and A'', a, d prefer A'' over A', b prefers A' over A'', and c remains unbothered. So $A'' \succ A'$.
- In a comparison between A'' and A, b, d prefer A over A'', and a, c prefer A'' over A. So A is as popular as A''.

Liquid democracy motivation

Liquid democracy

Voters may vote directly or delegate their vote to another voter they trust.

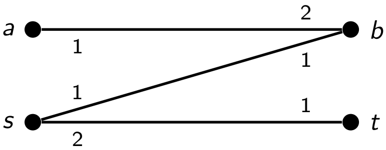
- Delegations are transitive: $u \to v$ and $v \to w$ means u is represented through w.
- Delegation cycles are forbidden.
- A delegation tree is naturally an arborescence rooted at direct representatives.
- Popularity asks whether a majority prefers another delegation tree.



r: direct vote
edges: delegates
ranks: trust

Popular matchings: definition

Gärdenfors (1975) introduced popularity as a majority-vote notion for matchings.



Numbers are preference ranks: smaller is better.

Pairwise election

For matchings M and N, each vertex votes for the matching giving it the better partner.

- Same partner means no vote.
- Being unmatched is worse than any acceptable partner.

Let $\phi(M, N)$ be the number of vertices that prefer M to N. A matching M is popular if

$$\phi(N, M) \leq \phi(M, N) \qquad \text{for every matching } N.$$

Gärdenfors theorem

Theorem [Gärdenfors 1975]

In the stable marriage setting, every stable matching is popular.

- Therefore a popular matching always exists whenever a stable matching exists.
- With complete preference lists, Gale–Shapley gives a stable matching, hence also a popular matching.

Why this is plausible

If another matching N defeated a stable matching M, then a majority of vertices would improve by switching from M to N.

Consequence

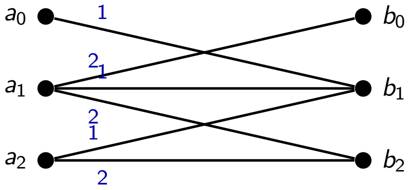
Such an improvement can be traced through alternating paths/cycles and yields a blocking pair for M, contradicting stability.

One-sided preferences in bipartite graphs

House-allocation model

Let $G = (A \cup B, E)$ be bipartite. Vertices in A are agents and vertices in B are houses.

- Each agent $a \in A$ strictly ranks its acceptable houses $N(a)$.
- Houses are indifferent and do not vote.
- For a matching M, write $M(a)$ for the house matched to a, or $\perp$ if a is unmatched.
- Every acceptable house is preferred to $\perp$.

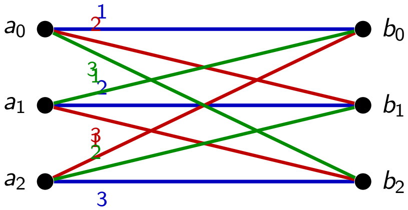


Popularity here

M is popular if no matching N gets more agent votes than M.

A one-sided instance

Do popular matchings always exist in the 1-sided preferences setting?



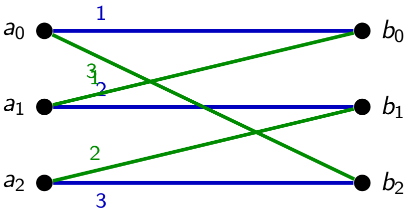
Preferences

Every agent has the same ranking:

$$b_0 \succ b_1 \succ b_2.$$

Matching	Edges
blue	$a_0 b_0,\ a_1 b_1,\ a_2 b_2$
red	$a_0 b_1,\ a_1 b_2,\ a_2 b_0$
green	$a_0 b_2,\ a_1 b_0,\ a_2 b_1$

Comparison: green versus blue



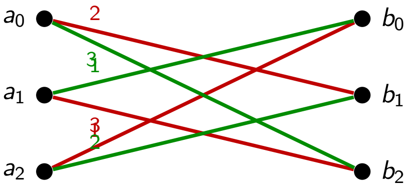
Agent	blue	green	Vote
a_0	b_0 (rank 1)	b_2 (rank 3)	blue
a_1	b_1 (rank 2)	b_0 (rank 1)	green
a_2	b_2 (rank 3)	b_1 (rank 2)	green

Vote count

green gets 2 votes and blue gets 1 vote.

Hence green $\succ$ blue.

Comparison: red versus green



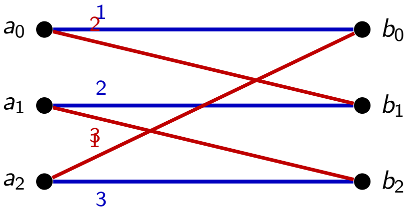
Agent	red	green	Vote
a_0	b_1 (rank 2)	b_2 (rank 3)	red
a_1	b_2 (rank 3)	b_0 (rank 1)	green
a_2	b_0 (rank 1)	b_1 (rank 2)	red

Vote count

red gets 2 votes and green gets 1 vote.

Hence red $\succ$ green.

Comparison: blue versus red



Agent	blue	red	Vote
a_0	b_0 (rank 1)	b_1 (rank 2)	blue
a_1	b_1 (rank 2)	b_2 (rank 3)	blue
a_2	b_2 (rank 3)	b_0 (rank 1)	red

Vote count

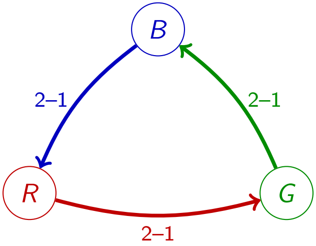
blue gets 2 votes and red gets 1 vote.

Hence blue $\succ$ red.

No popular matching in the example

The three comparisons give a majority cycle:

$$\text{blue} \succ \text{red} \succ \text{green} \succ \text{blue}.$$



Certificate of nonexistence

In fact, every matching in this instance is beaten by another matching.

- If M is not perfect, extend it by matching an unmatched agent to an unused house. The new matching beats M.
- If M is perfect, rotate the houses upward: the agent with b_1 gets b_0, the agent with b_2 gets b_1, and the agent with b_0 gets b_2. Two agents improve and one gets worse.

Hence this one-sided bipartite instance has no popular matching.

References

D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

P. Gärdenfors. Match making: assignments based on bilateral preferences. *Behavioural Science*, 20:166–173, 1975.

D. J. Abraham, R. W. Irving, T. Kavitha, and K. Mehlhorn. Popular matchings. *SIAM Journal on Computing*, 37(4):1030–1045, 2007.

P. Hall. On representatives of subsets. *Journal of the London Mathematical Society*, 10:26–30, 1935.

J. Kleinberg and É. Tardos. *Algorithm Design*. Pearson/Addison–Wesley, 2006.

Thank you!

Questions?