Lec # 23, 24 & 25

Definition 4.32 Let P be a set of vertices and $\overline{P}$ denote those vertices not in P (called the complement of P). A ***cut*** $(P, \overline{P})$ is the set of all arcs xy where x is a vertex from P and y is a vertex from $\overline{P}$. An $\boldsymbol{s-t}$ ***cut*** is a cut in which the source s is in P and the sink t is in $\overline{P}$.

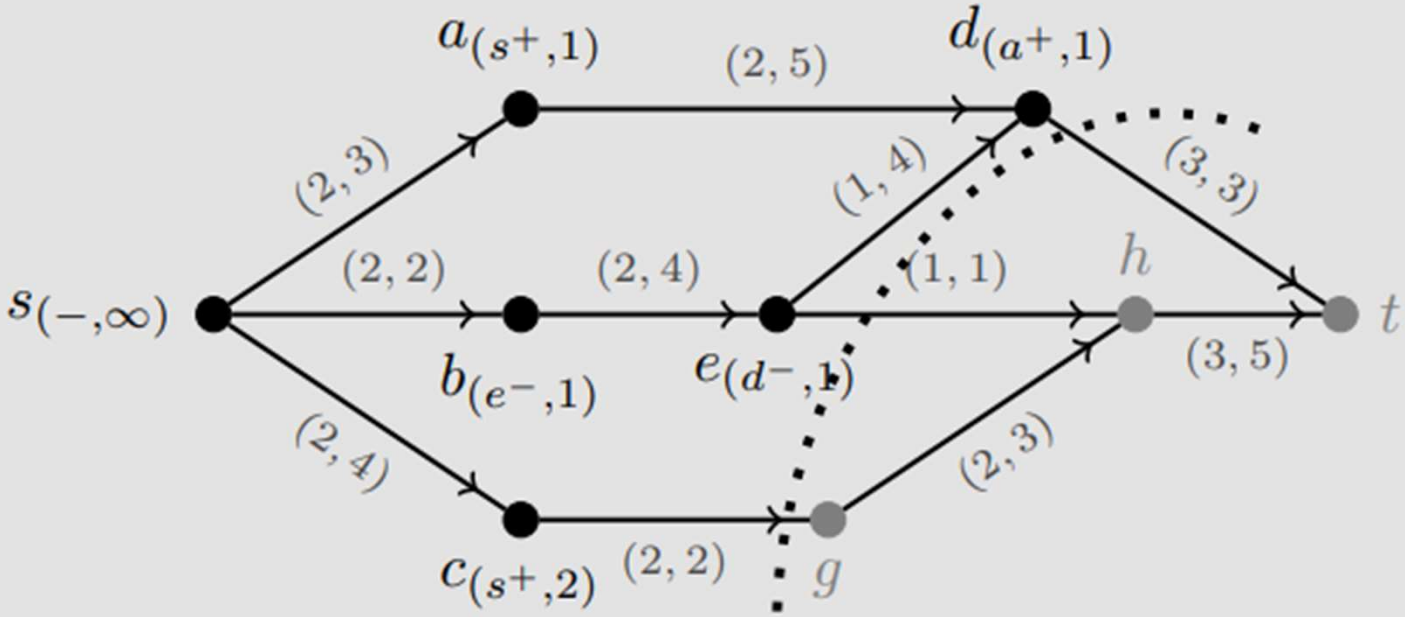
Definition 4.33 The ***capacity*** of a cut, $c(P, \overline{P})$, is defined as the sum of the capacities of the arcs that comprise the cut.

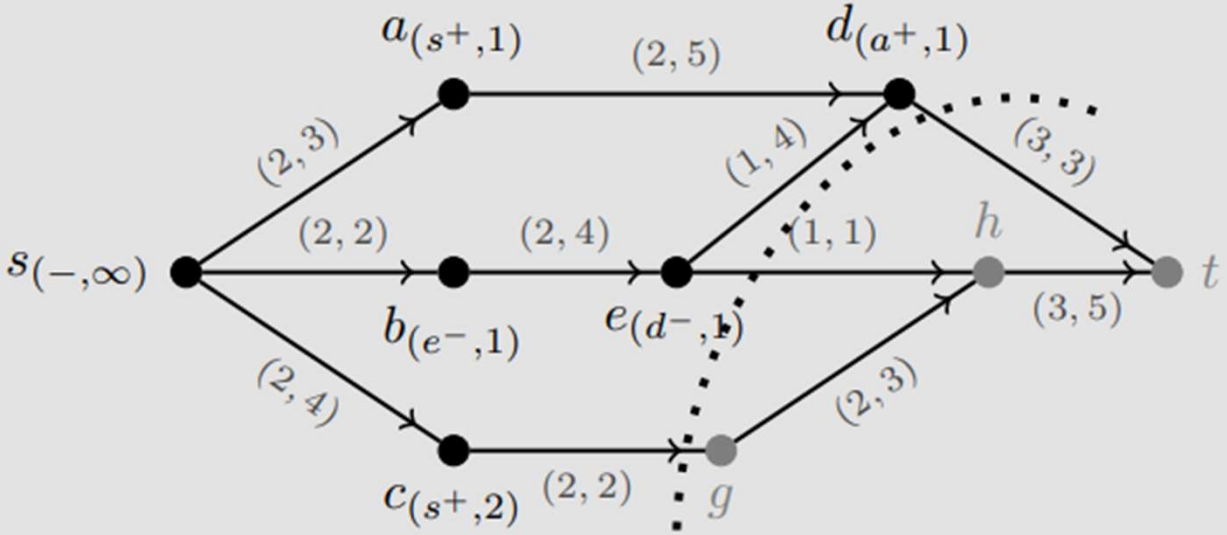
Theorem 4.34 (Max Flow–Min Cut) In any directed network, the value of a maximum $s-t$ flow equals the capacity of a minimum $s-t$ cut.

Min-Cut Method

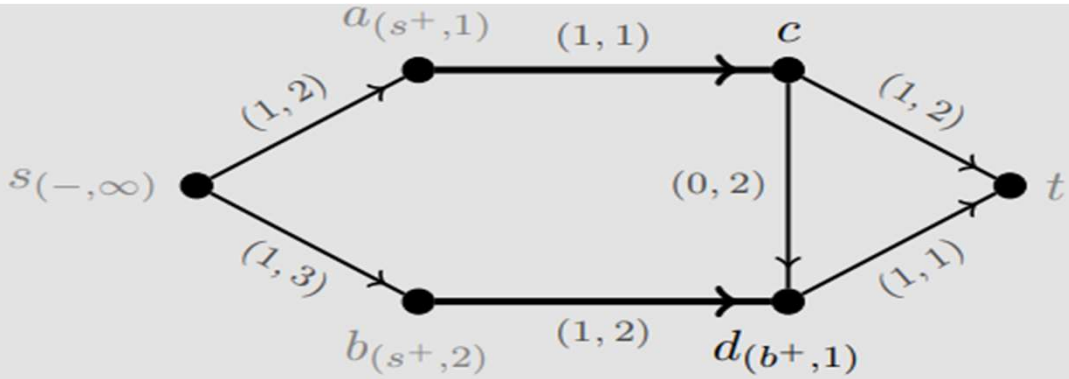
1. Let $G = (V, A, c)$ be a network with a designated source s and sink t and each arc is given a capacity c.
2. Apply the Augmenting Flow Algorithm.
3. Define an $s - t$ cut $(P, \overline{P})$ where P is the set of labeled vertices from the final implementation of the algorithm.
4. $(P, \overline{P})$ is a minimum $s - t$ cut for G.

Example 4.7 Use the Min-Cut Method to find a minimum $s-t$ cut for the network G_8 on page 189 and the network G_9 from Example 4.5.

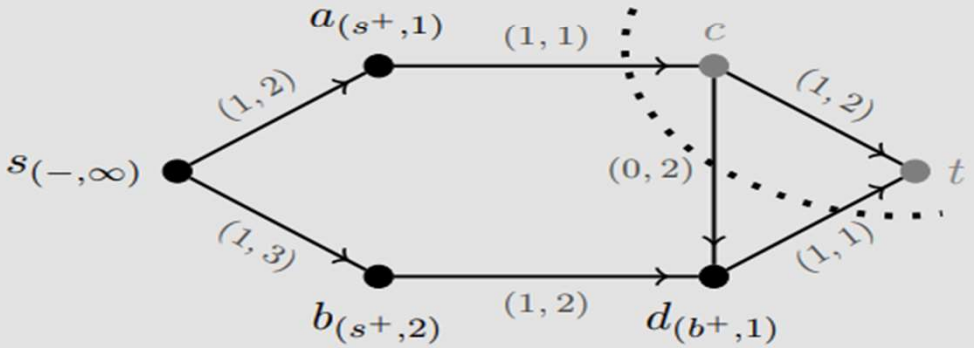




The Min-Cut Method sets $P = \{s, a, b, c, d, e\}$ and $\overline{P} = \{g, h, t\}$. The arcs in the cut are $\{dt, eh, cg\}$, making the capacity of this cut $c(P, \overline{P}) = 3 + 1 + 2 = 6$. Since we have found a flow and cut with the same value, we know the flow is maximum and the cut is minimum.



The final labeling in the network G_9 from Example 4.5 gives $P = \{s, a, b, d\}$ and $\overline{P} = \{c, t\}$. The arcs in the cut set are $\{ac, dt\}$. Note, cd is not in the cut since the arc is in the wrong direction. The capacity of this cut is $c(P, \overline{P}) = 2$, and since we have found a flow and cut with the same value, we know the flow is maximum and the cut is minimum.



EX #:4.6
Problems: 4.1-4.6, 4.9, 4.10, 4.13, 4.14, 4.15,4.17, 4.20.

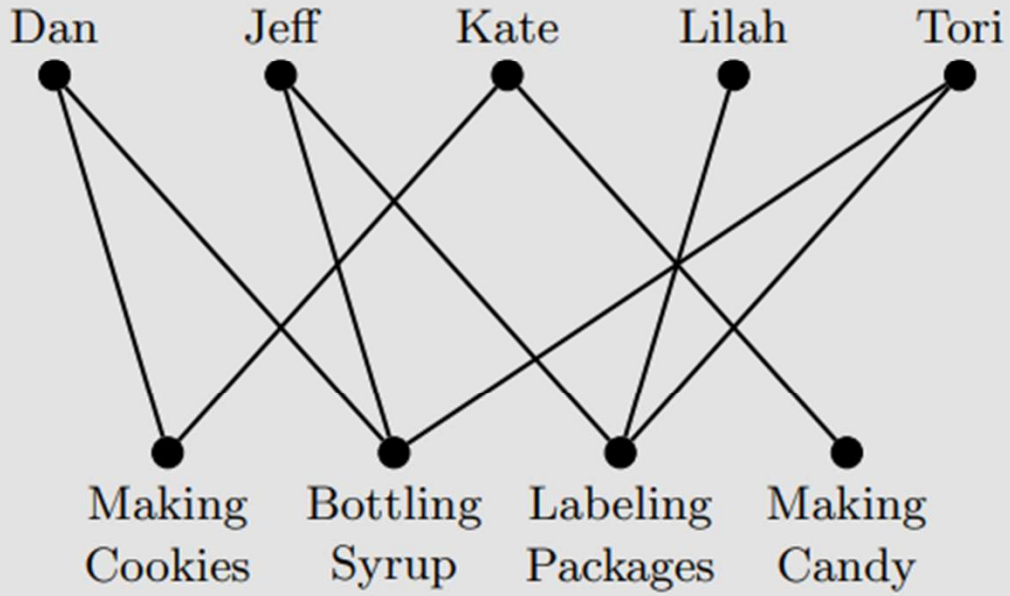
Matching in Bipartite Graph

Definition 5.1 Given a graph $G = (V, E)$, a ***matching*** M is a subset of the edges of G so that no two edges share an endpoint. The size of a matching, denoted $|M|$, is the number of edges in the matching.

Example 5.1 The Vermont Maple Factory just received a rush order for 6-dozen boxes of maple cookies, 3-dozen bags of maple candy, and 10-dozen bottles of maple syrup. Some employees have volunteered to stay late tonight to help finish the orders. In the chart below, each employee is shown along with the jobs for which he or she is qualified. Draw a graph to model this situation and find a matching.

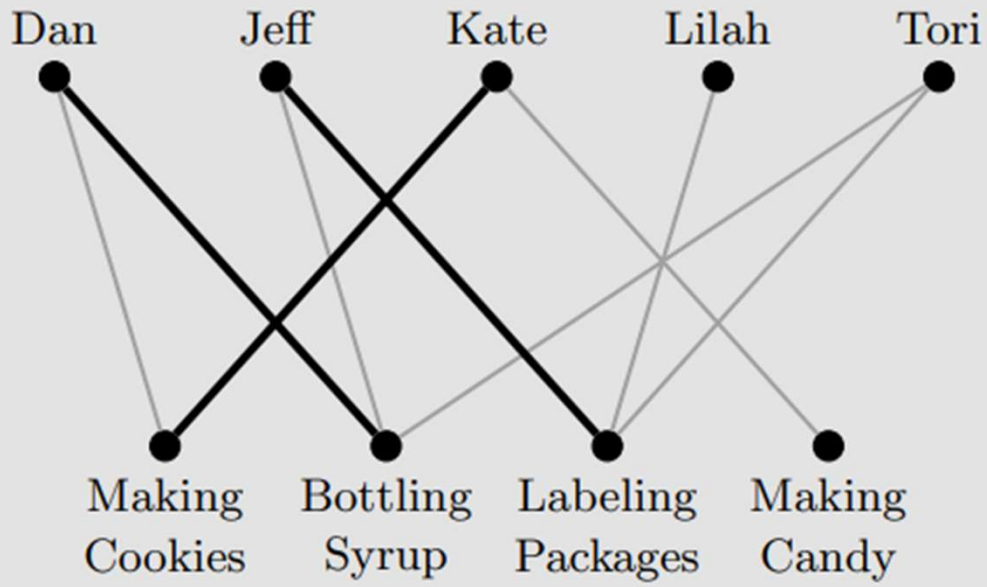
Employee	Task	
Dan	Making Cookies	Bottling Syrup
Jeff	Labeling Packages	Bottling Syrup
Kate	Making Candy	Making Cookies
Lilah	Labeling Packages	
Tori	Labeling Packages	Bottling Syrup

Solution: Model using a bipartite graph where X consists of the employees and Y consists of the tasks. We draw an edge between two vertices a and b if employee a is capable of completing the task b, creating G_1 below.



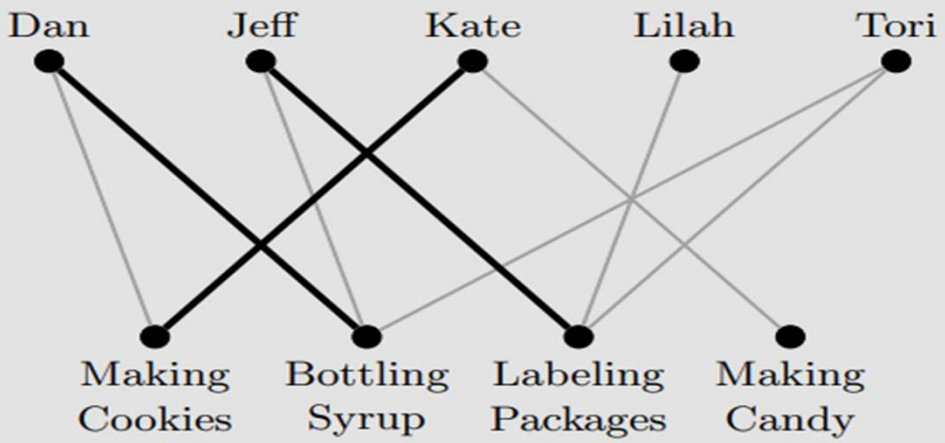
G_1

A matched edge, which is shown in bold below, represents the assignment of a task to an employee. One possible matching is shown below.



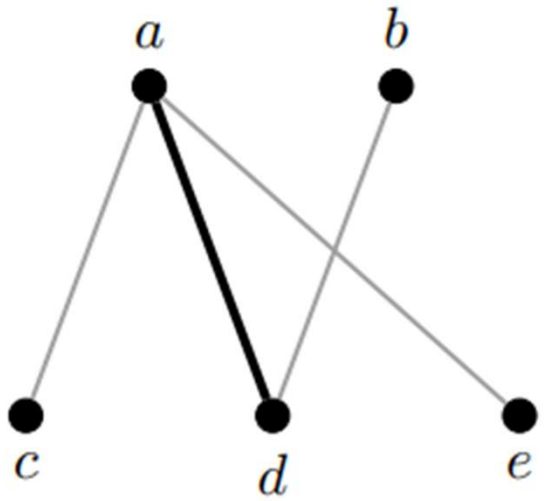
Definition 5.2 A vertex is ***saturated*** by a matching M if it is incident to an edge of the matching; otherwise, it is called ***unsaturated***.

A matched edge, which is shown in bold below, represents the assignment of a task to an employee. One possible matching is shown below.

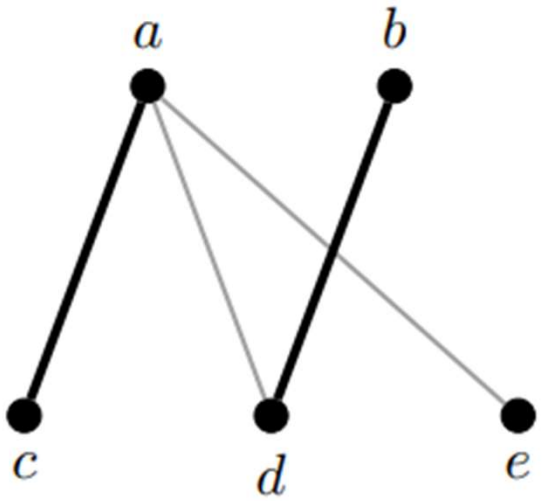


Definition 5.3 Given a matching M on a graph G, we say M is

- ***maximal*** if M cannot be enlarged by adding an edge.
- ***maximum*** if M is of the largest size amongst all possible matchings.
- ***perfect*** if M saturates every vertex of G.
- an ***X-matching*** if it saturates every vertex from the collection of vertices X (a similar definition holds for a Y-matching).



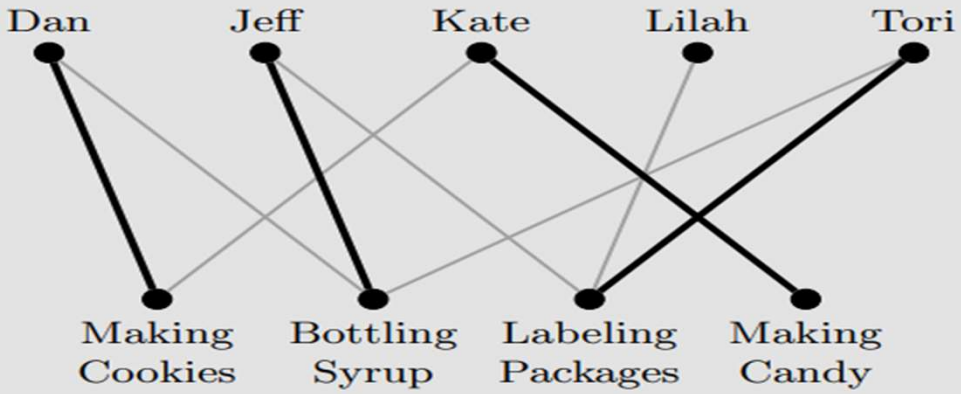
Maximal Matching
of G_2



Maximum Matching
of G_2

Example 5.2 Determine and find the proper type of matching for the Vermont Maple Factory graph G_1 from Example 5.1.

Solution: Since we need the tasks to be completed but do not need every employee to be assigned a task, we must find an X-matching where X consists of the vertices representing the tasks. An example of such a matching is shown below.



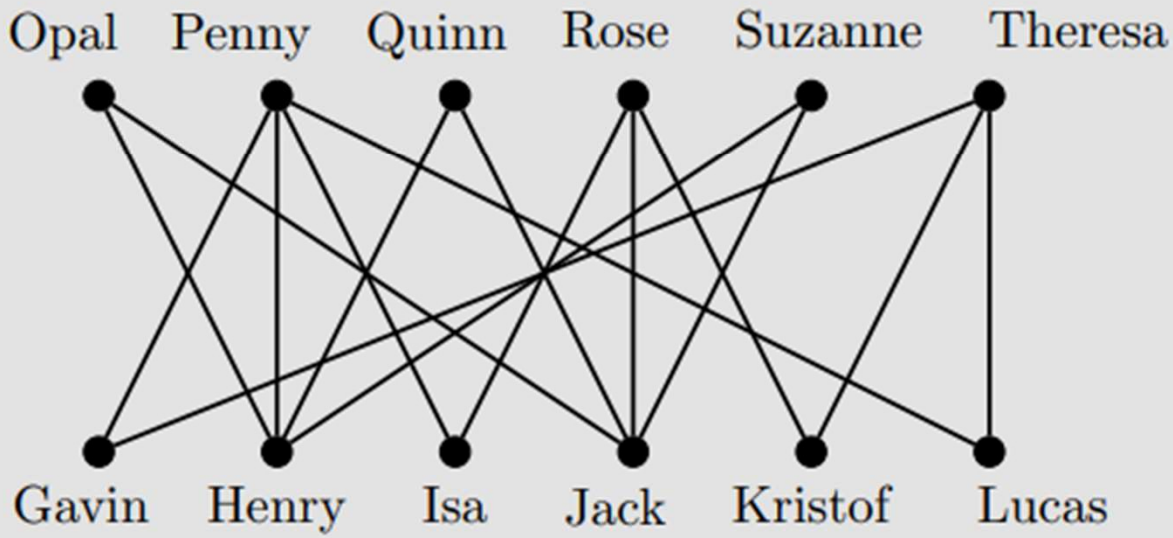
Note that all tasks are assigned to an employee, but not all employees have a task (Lilah is not matched with a task). In addition, this is not the only matching possible. For example, we could have Lilah labeling packages and Tori bottling syrup with Jeff having no task to complete.

Theorem 5.4 (Hall's Marriage Theorem) Given a bipartite graph $G = (X \cup Y, E)$, there exists an X-matching if and only if $|S| \leq |N(S)|$ for any $S \subseteq X$.

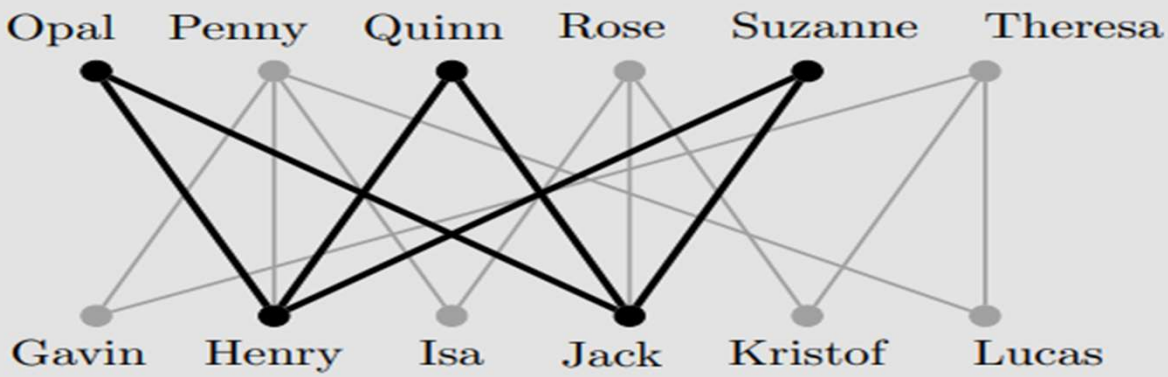
Example 5.3 In a small town there are 6 boys and 6 girls whose parents wish to pair into marriages where the only requirement is that a girl must like her future spouse (pretty low standards in my opinion). The table below lists the girls and the boys she likes. Find a pairing with as many marriages occurring as possible.

Girls	Boys She Likes			
Opal	Henry	Jack		
Penny	Gavin	Isa	Henry	Lucas
Quinn	Henry	Jack		
Rose	Kristof	Isa	Jack	
Suzanne	Henry	Jack		
Theresa	Gavin	Lucas	Kristof	

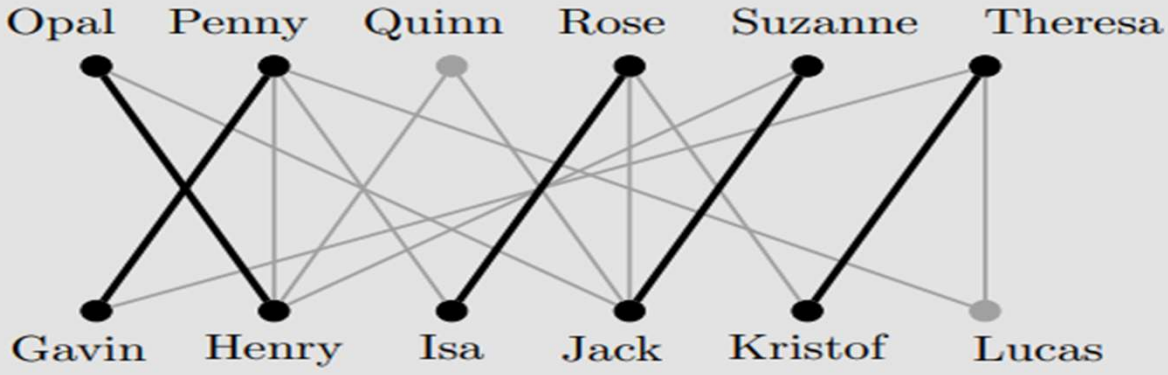
Solution: The information from the table will be displayed using a bipartite graph, where X consists of the girls and Y consists of the boys.



Notice that Opal, Quinn, and Suzanne all only like the same two boys (Henry and Jack), so at most two of these girls can be matched, as shown below in the following graph.



This means at most 5 marriages are possible; one such solution is shown below.



Corollary 5.5 Every k-regular bipartite graph has a perfect matching for all $k > 0$.

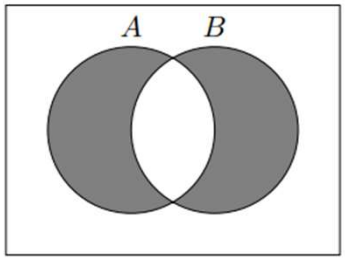
Definition 5.6 Given a matching M of a graph G, a path is called

- ***M-alternating*** if the edges in the path alternate between edges that are part of M and edges that are not part of M.
- ***M-augmenting*** if it is an M-alternating path and both endpoints of the path are unsaturated by M, implying both the starting and ending edges of the path are not part of M.

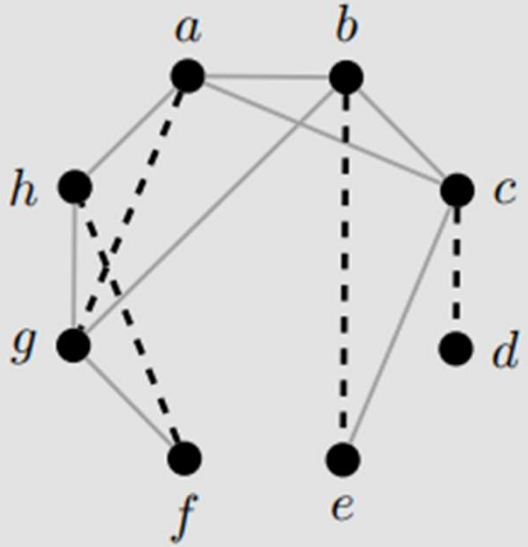
Theorem 5.7 (Berge's Theorem) A matching M of a graph G is maximum if and only if G does not contain any M-augmenting paths.

Definition 5.8 Let A and B be two sets. Then the ***symmetric difference*** $A\triangle B$ is all those elements in exactly one of A and B; that is, $A\triangle B = (A - B) \cup (B - A)$.

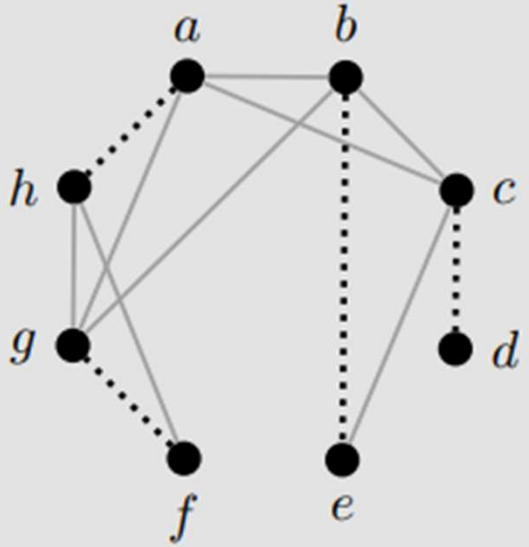
$$A\triangle B = A \cup B - A \cap B.$$



Example 5.4 Below are two different matchings of a graph G. Find $M_1 \triangle M_2$.

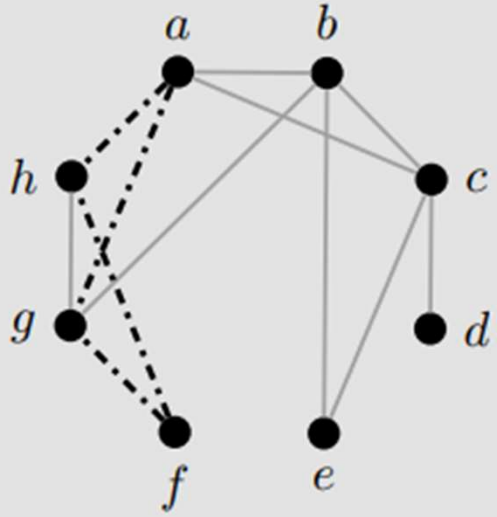


M_1



M_2

Solution: Note that the symmetric difference will only show those edges involved in exactly one of the two matchings. Thus if an edge is in both matchings or left unmatched in both matchings, it does not appear in $M_1 \triangle M_2$.



$M_1 \triangle M_2$

Lemma 5.9 Let M_1 and M_2 be two matchings in a graph G. Then every component of $M_1 \triangle M_2$ is either a path or an even cycle.

Theorem 5.7 (Berge's Theorem, restated) A matching M of a graph G is not maximum if and only if G contains some M-augmenting path.

Augmenting Path Algorithm

Input: Bipartite graph $G = (X \cup Y, E)$.

Steps:

1. Find an arbitrary matching M.
2. Let U denote the set of unsaturated vertices in X.
3. If U is empty, then M is a maximum matching; otherwise, select a vertex x from U.
4. Consider y in $N(x)$.
5. If y is also unsaturated by M, then add the edge xy to M to obtain a larger matching M'. Return to Step (2) and recompute U. Otherwise, go to Step (6).
6. If y is saturated by M, then find a maximal M-alternating path from x using xy as the first edge.

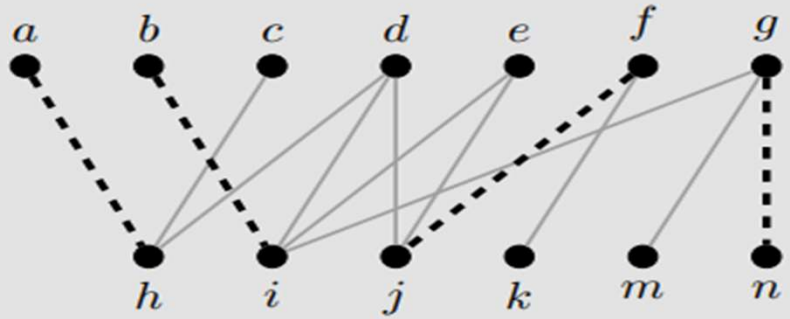
(a) If this path is M-augmenting, then switch edges along that path to obtain a larger matching M'; that is, remove from M the matched edges along the path and add the unmatched edges to create M'. Return to Step (2) and recompute U.

(b) If the path is not M-augmenting, return to Step (4), choosing a new vertex from $N(x)$.

7. Stop repeating Steps (2)–(4) when all vertices from U have been considered.

Output: Maximum matching for G.

Example 5.5 Apply the Augmenting Path Algorithm to the bipartite graph G_6 below, where $X = \{a, b, c, d, e, f, g\}$ and $Y = \{h, i, j, k, m, n\}$, with an initial matching shown as dashed lines.

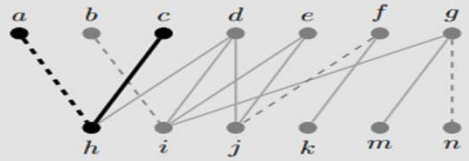


G_6

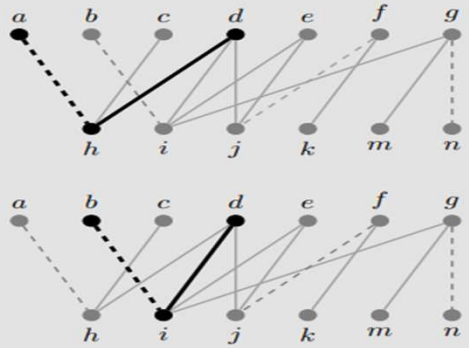
Solution: During each step shown below, the path under consideration will be in bold, with the matching shown as dashed lines throughout.

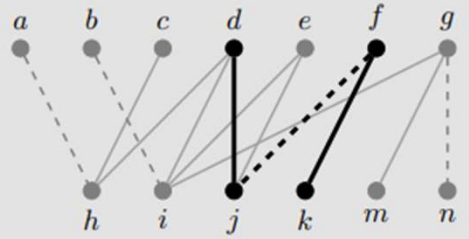
Step 1: The unsaturated vertices from X are $U = \{c, d, e\}$.

Step 2: Choose c. The only neighbor of c is h, which is saturated by M. Form an M-alternating path starting with the edge ch. This produces the path $c\,h\,a$, as shown on the next page, which is not augmenting.

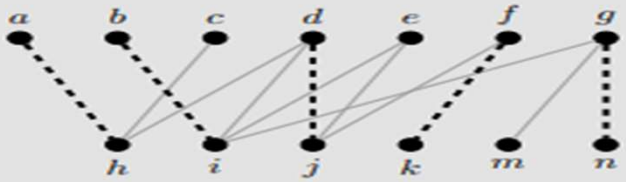


Step 3: Choose a new vertex from U, say d. Then $N(d) = \{h, i, j\}$. Below are the alternating paths originating from d.



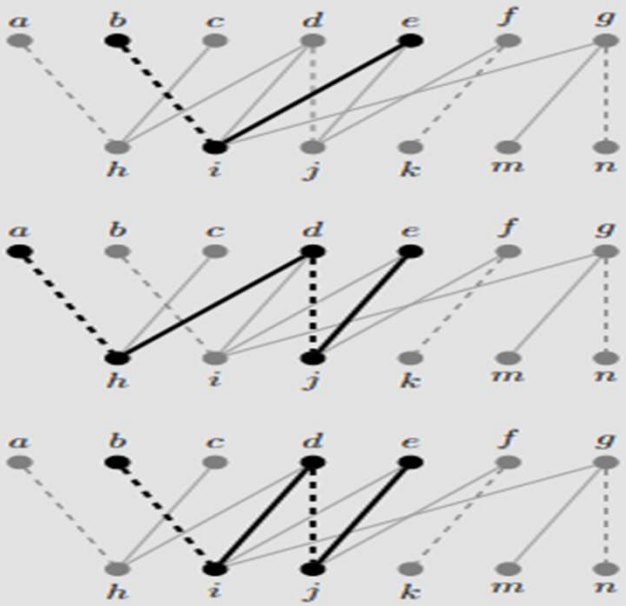


Note that the last path $(d\,j\,f\,k)$ is M-augmenting. Form a new matching M' by removing edge fj from M and adding edges dj and fk, as shown in the following graph.



Step 4: Recalculate $U = \{c, e\}$. We must still check c since it is possible for the change in matching to modify possible alternating paths from a previously reviewed vertex; however, the path obtained is $c\,h\,a$, the same as from Step 2.

Step 5: Check the paths from e. The alternating paths are shown below.



None of these paths are augmenting. Thus no M'-augmenting paths exist in G and so M' must be maximum by Berge's Theorem.

Output: The maximum matching $M' = \{ah, bi, dj, fk, gn\}$ from Step 3.

Definition 5.10 A ***vertex cover*** Q for a graph G is a subset of vertices so that every edge of G has at least one endpoint in Q.

Theorem 5.11 (König-Egerváry Theorem) For a bipartite graph G, the size of a maximum matching of G equals the size of a minimum vertex cover for G.