Listen Learn Lead

Engineering Excellence for Empowerment

Nadar Saraswathi College of Engineering & Technology

NSCET E-LEARNING PRESENTATION

LISTEN … LEARN… LEAD…

COMPUTER SCIENCE AND ENGINEERING

II YEAR / IV SEMESTER

CS8451 – DESIGN AND ANALYSIS OF ALGORITHMS

S ARUL JOTHI M.E.,MISTE,
ASSISTANT PROFESSOR
Nadar Saraswathi College of Engineering & Technology,
Vadapudupatti, Annanji (PO), Theni – 625531.

UNIT 4-Iterative Improvement

- Iterative Improvement method
- Simplex method
- Maximum flow Problem
- Maximum matching in Bipartite Graphs
- The stable Marriage Problem

Lecture 01-Iterative Improvement method

- It starts with some feasible solution (a solution that satisfies all the constraints of the problem) and proceeds to improve it by repeated applications of some simple step.
- This step typically involves a small, localized change yielding a feasible solution with an improved value of the objective function. When no such change improves the value of the objective function, the algorithm returns the last feasible solution as optimal and stops.

Iterative Improvement method

The linear programming problem of optimizing a linear function of several variables subject to a set of linear constraints:

maximize (or minimize) $c_1x_1 + \dots + c_nx_n$

subject to $a_{i1}x_1 + \dots + a_{in}x_n \leq$ (or $\geq$ or $=$) b_i for $i = 1,\dots,m$

$x1 \geq 0,\dots,x_n \geq 0$.

Iterative Improvement method

A general method for solving linear programming problems,

EXAMPLE 1 Consider the following linear programming problem in two variables:

maximize $3x + 5y$

subject to $x + y \leq 4$

$x + 3y \leq 6$

$x \geq 0, y \geq 0.$

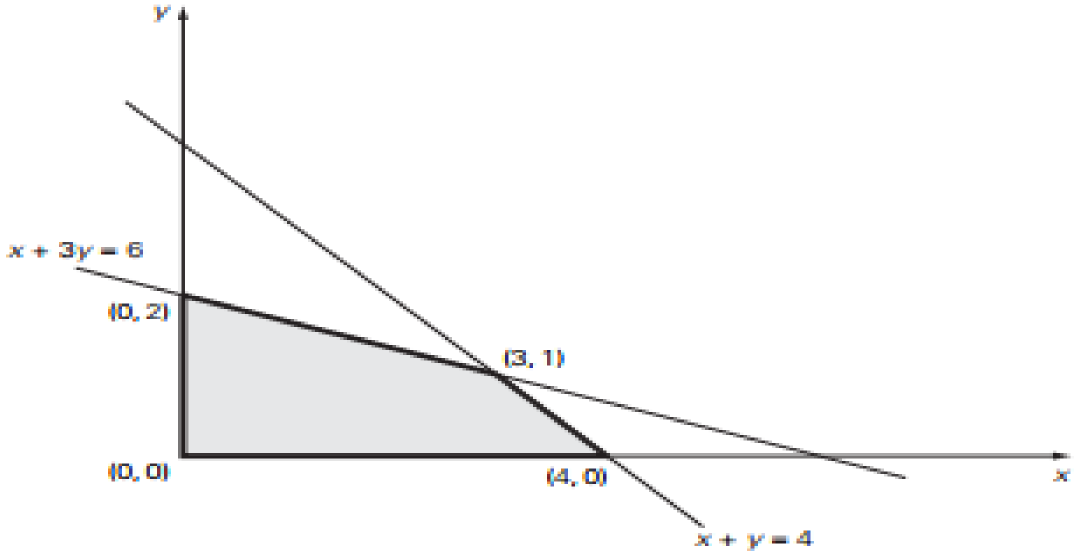
Iterative Improvement method

- feasible solution to this problem is any point (x, y) that satisfies all the constraints of the problem;
- Our task is to find an optimal solution, a point in the feasible region

  with the largest value of the objective function $z = 3x + 5y$.

Iterative Improvement method

- Feasible region of Problem



Iterative Improvement method

- Linear programming problems with the empty feasible region are called infeasible
- Obviously, infeasible problems do not have optimal solutions
- Another complication may arise if the problem's feasible region is unbounded, If the feasible region of a linear programming problem is unbounded,
- its objective function may or may not attain a finite optimal value on it.

Iterative Improvement method

- (Extreme Point Theorem) Any linear programming problem with a nonempty bounded feasible region has an optimal solution; moreover, an optimal solution can always be found at an extreme point of the problem's feasible region.

Lecture 02

Simplex method

- Apply the simplex method to a linear programming problem, it has to be represented in a special form called the standard form.
- The standard form has the following requirements:
- It must be a maximization problem.
- All the constraints (except the non negativity constraints) must be in the form of linear equations with nonnegative right-hand sides.
- All the variables must be required to be nonnegative.

Iterative Improvement method-Simplex method

Thus, the general linear programming problem in standard form with m constraints and n unknowns ($n \geq m$) is

maximize $c1x1 + \ldots + c_n x_n$

subject to $ai1x1 + \ldots + a_{in} x_n = b_i$, where $bi \geq 0$ for $i = 1, 2, \ldots, m$

$x1 \geq 0, \ldots, xn \geq 0$.

Iterative Improvement method-Simplex method

- Any linear programming problem can be transformed into an equivalent problem in standard form.
- If a constraint is given as an inequality, it can be replaced by an equivalent equation by adding a slack variable representing the difference between the two sides of the original inequality.
- For example, the two inequalities of problem can be transformed

$$c = [c_1 \; c_2 \; \dots \; c_n], \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Iterative Improvement method-Simplex method

- For the general case of a problem with m equations in n unknowns ($n \geq m$), $n - m$ variables need to be set to zero to get a system of m equations in m unknowns.
- If the system obtained has a unique solution—as any nondegenerate system of linear equations with the number of equations equal to the number of unknowns does—we have a basic solution; its coordinates set to zero before solving the system are called nonbasic, and its coordinates obtained by solving the system are called basic

Iterative Improvement method-Simplex method

- If all the coordinates of a basic solution are nonnegative, the basic solution is called a basic feasible solution.
- the simplex method progresses through a series of adjacent extreme points (basic feasible solutions) with increasing values of the objective function.
- Each such point can be represented by a simplex tableau, a table storing the information about the basic feasible solution corresponding to the extreme point

Simplex Algorithm

Step 0 Initialization

Present a given linear programming problem in standard form and set up an initial tableau with nonnegative entries in the rightmost column and m other columns composing the m × m identity matrix.

Step 1

Optimality test If all the entries in the objective row (except, possibly,the one in the rightmost column, which represents the value of them are nonnegative—stop: the tableau represents an optimal solution whose basic variables' values are in the rightmost column and the remaining, nonbasic variables' values are zero

Simplex Algorithm

Step 2 Finding the entering variable Select a negative entry from among the first n elements of the objective row. (A commonly used rule is to select the negative entry with the largest absolute value, with ties broken arbitrarily.) Mark its column to indicate the entering variable and the pivot column.

Step 3 Finding the departing variable For each positive entry in the pivot column, calculate the θ-ratio by dividing that row's entry in the rightmost column by its entry in the pivot column. (If all the entries in the pivot column are negative or zero, the problem is unbounded—stop.)

Find the row with the smallest θ-ratio (ties may be broken arbitrarily),and mark this row to indicate the departing variable and the pivot row.

Simplex Algorithm

Step 4 Forming the next tableau Divide all the entries in the pivot row by its entry in the pivot column. Subtract from each of the other rows,including the objective row, the new pivot row multiplied by the entry in the pivot column of the row in question.

Replace the label of the pivot row by the variable’s name of the pivot column

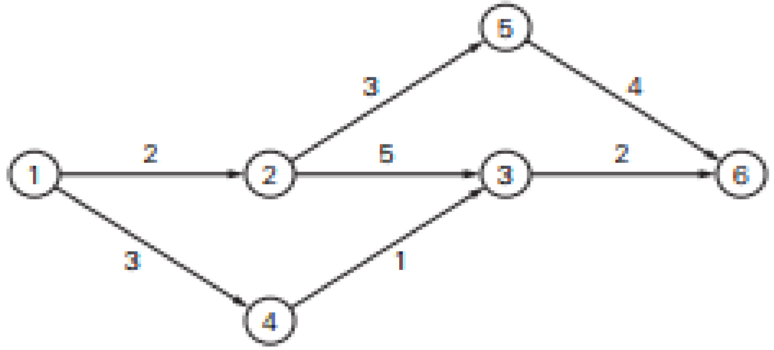
and go back to Step 1.

Lecture 4
Maximal Flow Problem

- The important problem of maximizing the flow of a material through a transportation network (pipeline system, communication system, electrical distribution system, and so on).
- Assume that the transportation network in question can be represented by a connected weighted digraph with n vertices numbered from 1 to n and a set of edges E, with the following properties:
- It contains exactly one vertex with no entering edges; this vertex is called the source and assumed to be numbered 1
- It contains exactly one vertex with no leaving edges; this vertex is called the sink and assumed to be numbered n.

Maximal Flow Problem

- The weight u_{ij} of each directed edge (i, j) is a positive integer, called the edge capacity. (This number represents the upper bound on the amount of the material that can be sent from i to j through a link represented by the edge.)
- A digraph satisfying these properties is called a flow network or simply a network.
- Example of a network graph



Maximal Flow Problem

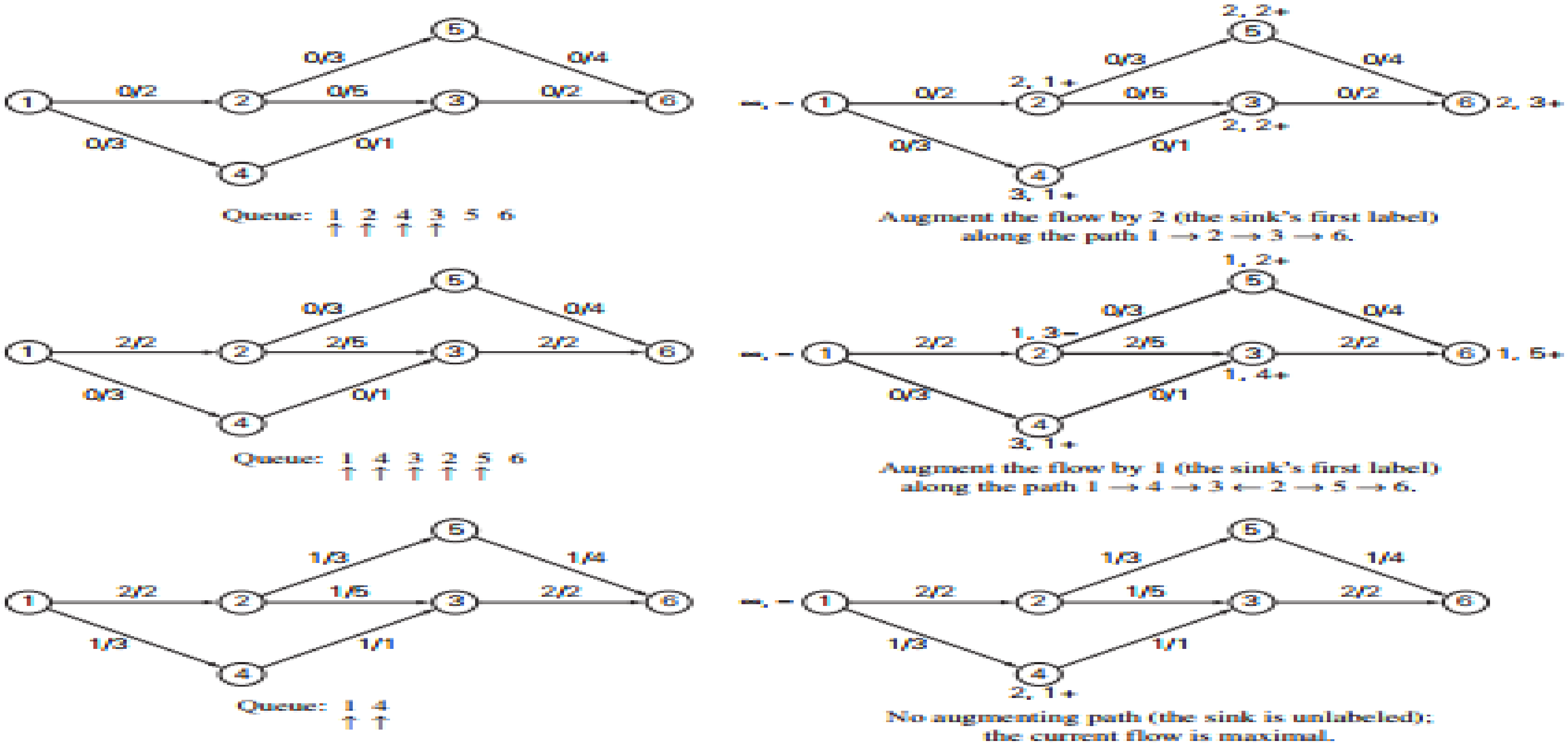
- The total amount of the material entering an intermediate vertex must be equal to the total amount of the material leaving the vertex. This condition is called the flow-conservation requirement.
- Let denote the amount sent through edge (i, j) by x_{ij} ,then for any intermediate vertex i, the flow conservation requirement can be expressed by the following equality constraint

$$\sum_{j:(j,i)\in E} x_{ji} = \sum_{j:(i,j)\in E} x_{ij} \quad \text{for } i = 2, 3, \ldots, n-1,$$

Maximal Flow Problem

- This quantity, the total outflow from the source—or, equivalently, the total inflow into the sink—is called the value of the flow.
- flow is an assignment of real numbers xij to edges (i, j) of a given network that satisfy flow-conservation constraints and the capacity constraints
- find a path from source to sink along which some additional flow can be sent. Such a path is called flow augmenting.
- If no flow-augmenting path can be found, concluded that the current flow is optimal. This general template for solving the maximum-flow problem is called the augmenting-path method, also known as the Ford-Fulkerson method

Maximal Flow Problem-Example



Maximal Flow Problem

Max-Flow Min-Cut Theorem

The value of a maximum flow in a network is equal to the capacity of its minimum cut.

Proof: First, let x be a feasible flow of value v and let C(X, X) be a cut of capacity c in the same network

$$v = \sum_{i \in X,\, j \in \bar{X}} x_{ij} - \sum_{j \in \bar{X},\, i \in X} x_{ji}.$$

– Thus, the value of any feasible flow in a network cannot exceed the capacity of any cut in that network.

$$v \leq \sum_{i \in X,\, j \in \bar{X}} x_{ij} \leq \sum_{i \in X,\, j \in \bar{X}} u_{ij},$$

$$v \leq c.$$

Lecture 05- Maximal Flow Problem

```
Algorithm Ford –Fulkerson(G)
%%input:Graph G
%%output:Max-Flow
Begin
for every edge(i,j)Ɛ E do
 Flow=0
      End for
While (there exists an augumenting path P in residual graph) do
  Find max-flow in an augumenting path P
     for all(i,j) of P do
          update max-flow using P and modify residual network
      End for
End while
Return max-flow
 End
```

Maximal Flow Problem

Complexity analysis:

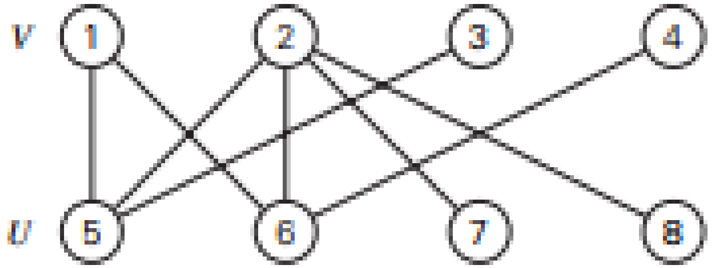
- The time required to find a shortest augmenting path by breadth-first search is in $O(n + m) = O(m)$ for networks represented by their adjacency lists, the time efficiency of the shortest-augmenting-path algorithm is in $O(nm^2)$.
- Faster algorithms of this kind have worst-case efficiency close to $O(nm)$.

Maximum matching in Bipartite Graphs

- A matching in a graph is a subset of its edges with the property that no two edges share a vertex.
- A maximum matching—more precisely, a maximum cardinality matching—is a matching with the largest number of edges.
- The maximum-matching problem is the problem of finding a maximum matching in a given graph
- In a bipartite graph, all the vertices can be partitioned into two disjoint sets V and U,not necessarily of the same size, so that every edge connects a vertex in one of these sets to a vertex in the other set

Maximum matching in Bipartite Graphs

- A graph is bipartite if its vertices can be colored in two colors so that every edge has its vertices colored in different colors; such graphs are also said to be 2-colorable
- Example for Bipartite graph



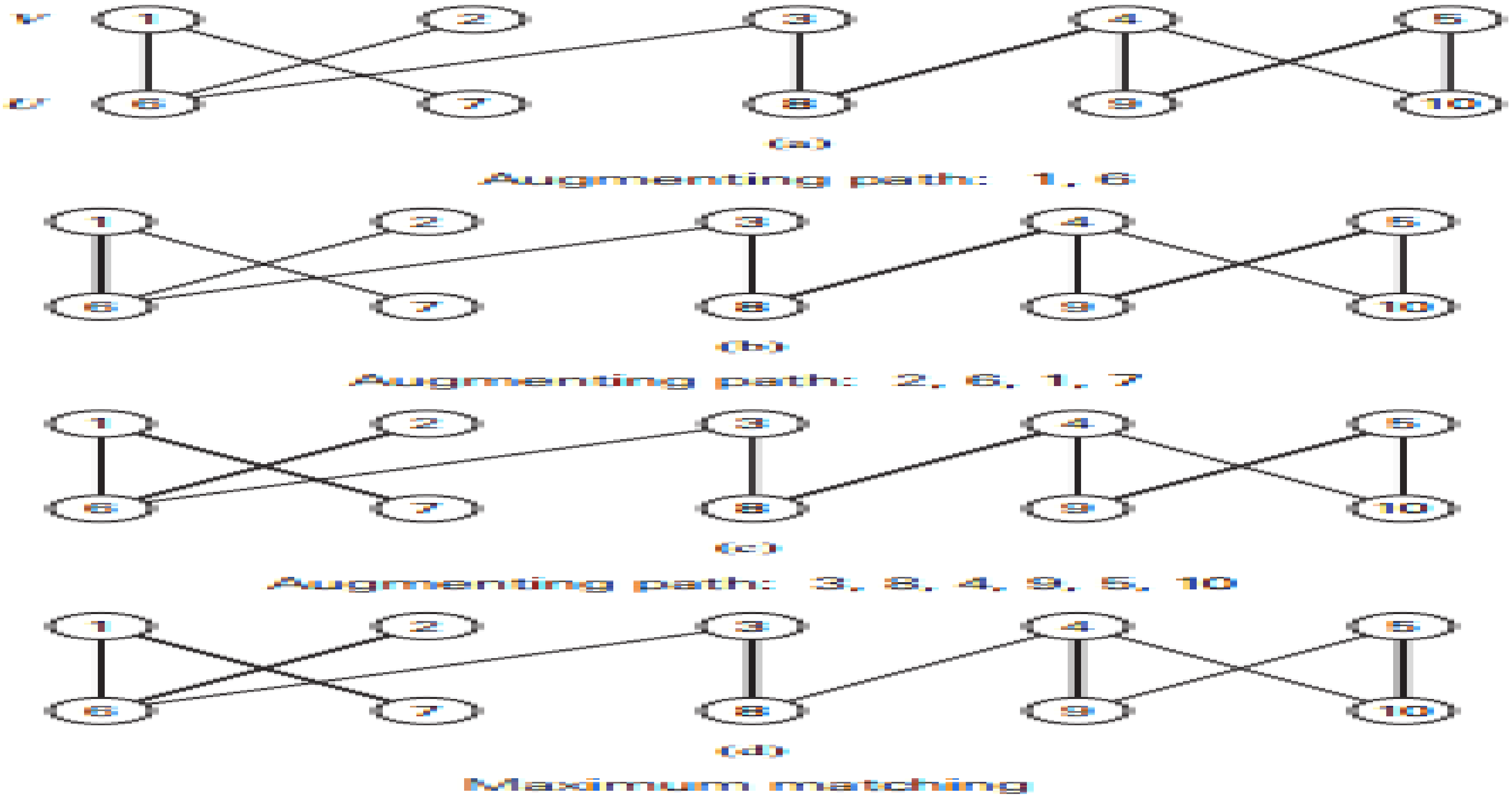
Maximum matching in Bipartite Graphs

- Let M be a matching in a bipartite graph G =V, U, E.
- Obviously, if every vertex in either V or U is matched (has a mate), i.e., serves as an endpoint of an edge in M, this cannot be done and M is a maximum matching.
- improving the current matching, both V and U must contain unmatched (also called free) vertices
- For example, for the matching Ma = {(4, 8), (5, 9)} in the graph in Figure vertices 1, 2, 3, 6, 7, and 10 are free, and vertices 4, 5, 8,and 9 are matched.

Maximum matching in Bipartite Graphs

- The path's edges in the odd-numbered positions and deleting from it the path's edges in the even-numbered positions yields a matching with one more edge than in M. Such a matching adjustment is called augmentation.
- obtain the matching Md = {(1, 7), (2, 6), (3, 8), (4, 9), (5, 10)}
- matching Md is not only a maximum matching but also perfect, i.e., a matching that matches all the vertices of the graph.

Maximum matching in Bipartite Graphs



Maximum matching in Bipartite Graphs

```
Algorithm bipartite _matching(G)
%%Input:Bipartite graph G
%%Output:Maximum matching M
Begin
    M=null
    Find augmenting path P
    while(P exists) then
        M=M+P
        Find next augmenting path
    End while
End
```

Maximum matching in Bipartite Graphs

Complexity Analysis:

- $n = |V| + |U|$ is the number of vertices in the graph.
- The time spent on each iteration is in $O(n + m)$, where $m = |E|$ is the number of edges in the graph. Hence, the time efficiency of the algorithm is in $O(n(n + m))$.

Stable marriage problem

- Consider a set Y = {m1, m2,...,m_n} of n men and a set X = {w1, w2,...,w_n} of n women.
- Each man has a preference list ordering the women as potential marriage partners with no ties allowed. Similarly, each woman has a preference list of the men, also with no ties.
- A marriage matching M is a set of n (m, w) pairs whose members are selected

from disjoint n-element sets Y and X in a one-one fashion, i.e., each man m from Y is paired with exactly one woman w from X and vice versa

Stable marriage problem

(a)Men preference list(b) Women preference list(c)Ranking Matrix

men's preferences

	1st	2nd	3rd
Bob:	Lea	Ann	Sue
Jim:	Lea	Sue	Ann
Tom:	Sue	Lea	Ann

(a)

women's preferences

	1st	2nd	3rd
Ann:	Jim	Tom	Bob
Lea:	Tom	Bob	Jim
Sue:	Jim	Tom	Bob

(b)

ranking matrix

	Ann	Lea	Sue
Bob	[2,3]	1,2	3,3
Jim	3,1	[1,3]	2,1
Tom	3,2	2,1	[1,2]

(c)

Stable marriage problem

- A pair (m, w), where m ∈ Y, w ∈ X, is said to be a blocking pair for a marriage matching M if man m and woman w are not matched in M but they prefer each other to their mates in M.
- A marriage matching M is called stable if there is no blocking pair for it;
- otherwise, M is called unstable
- The stable marriage problem is to find a stable marriage matching for men's and women's given preferences.

Stable marriage Algorithm

Input: A set of n men and a set of n women along with rankings of the women by each man and rankings of the men by each woman with no ties allowed in the rankings

Output: A stable marriage matching

Step 0 Start with all the men and women being free.

Step 1 While there are free men, arbitrarily select one of them and do the following: Proposal The selected free man m proposes to w, the next woman on his preference list.Response If w is free, she accepts the proposal to be matched with m. If she is not free, she compares m with her current mate. If she prefers m to him, she accepts m's proposal, making her former mate free; otherwise, she simply rejects m's proposal, leaving m free.

Step 2 Return the set of n matched pairs.

Stable marriage problem

Complexity Analysis:

The stable marriage algorithm terminates after no more than n^2 iterations with a stable marriage output.

This problem always has a solution that can be found by the Gale-Shapley algorithm.

THANK YOU