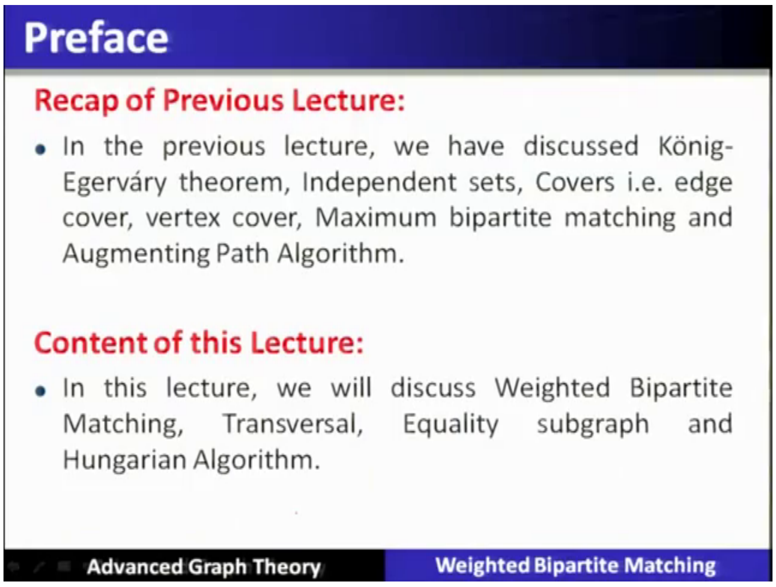
Advanced Graph Theory

Prof. Rajiv Mishra
Department of Computer Science and Engineering
Indian Institute of Technology, Patna

Lecture - 09
Weighted Bipartite Matching

(Refer Slide Time: 00:15)



Preface

Recap of Previous Lecture:

- In the previous lecture, we have discussed König-Egerváry theorem, Independent sets, Covers i.e. edge cover, vertex cover, Maximum bipartite matching and Augmenting Path Algorithm.

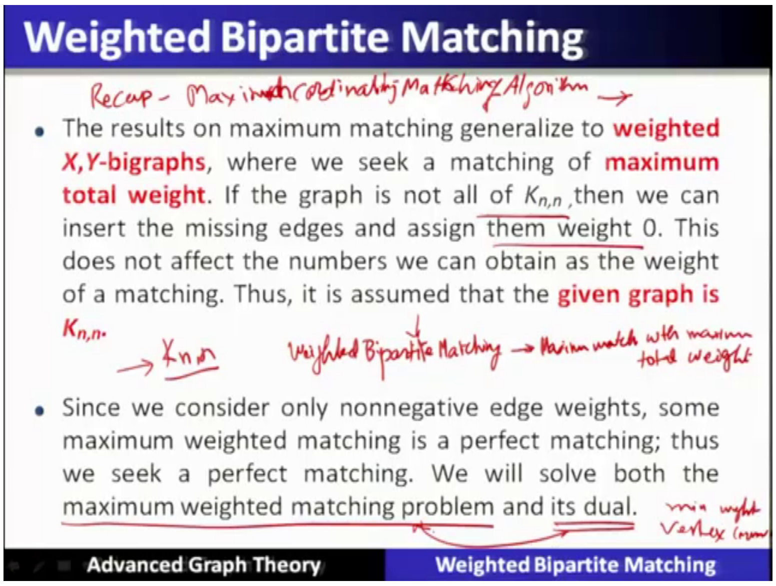
Content of this Lecture:

- In this lecture, we will discuss Weighted Bipartite Matching, Transversal, Equality subgraph and Hungarian Algorithm.

Advanced Graph Theory | Weighted Bipartite Matching

Weighted bipartite matching: recap of previous lecture we have discussed Konig Egervary theorem independent sets covers that is the vertex cover, edge cover, maximum bipartite matching, augmenting path algorithm. Content of this lecture in this lecture we will discuss weighted bipartite matching, transversal, equality subgraph and algorithm which is given for weighted bipartite matching is called Hungarian Algorithm Weighted bipartite matching.

(Refer Slide Time: 00:51)



Weighted Bipartite Matching

Recap - Maximum cardinality Matching Algorithm →

- The results on maximum matching generalize to **weighted X,Y-bigraphs**, where we seek a matching of **maximum total weight**. If the graph is not all of $K_{n,n}$,then we can insert the missing edges and assign them weight 0. This does not affect the numbers we can obtain as the weight of a matching. Thus, it is assumed that the **given graph is $K_{n,n}$.**

$\rightarrow K_{n,n}$ Weighted Bipartite Matching → Maximum match with maximum total weight

- Since we consider only nonnegative edge weights, some maximum weighted matching is a perfect matching; thus we seek a perfect matching. We will solve both the maximum weighted matching problem and its dual. min weight Vertex cover

Advanced Graph Theory | Weighted Bipartite Matching

In the last lecture we have seen the maximum bipartite matching or we can we have also seen the maximal cardinality. We have seen earlier the maximal cardinality, matching algorithm which we have seen in the last lectures we will generalize it.

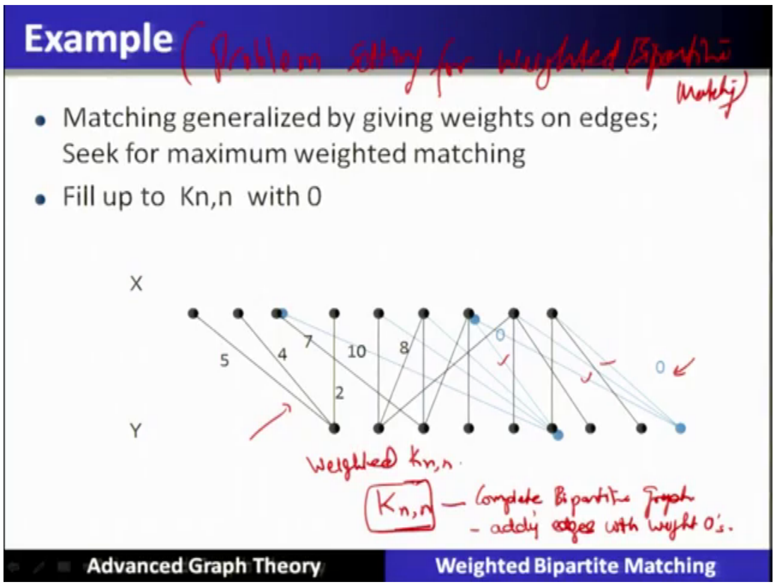
If you take a graph which is a weighted bipartite graph we seek a matching a maximum matching of the maximum total weight and that is why it is called as a weighted bipartite matching. So, the results on a maximum bipartite matching will now generalized to the weighted graphs and in this we seek the matching of a maximum total weight. Hence it is called a weighted bipartite matching.

So, here we are given a graph again of that particular complete bipartite graph K n. If the edges are not present then we will add the edges having the weight zeroes, to make it K n n having adding this particular extra edges of a weight 0, we are not going to change any anything in this particular problem setting that is the solution is not going to be changed here in this case so, we can do that..

Since we consider only the non negative edge weights on the edges some maximum weighted matching is a perfect matching and thus we will see the perfect matching and we will solve both the maximum bipartite matching and it is dual in this particular problem setting and we will see what it is dual is on a last lecture we have seen that dual of a maximum matching problem was the vertex cover..

So, it is a minimum weight vertex cover and that will be basically solving the problem of maximum weighted bipartite matching here in this particular problem setting.

(Refer Slide Time: 04:15)



Example (Problem setting for weighted Bipartite Matching)

- Matching generalized by giving weights on edges; Seek for maximum weighted matching
- Fill up to Kn,n with 0

weighted Kn,n.

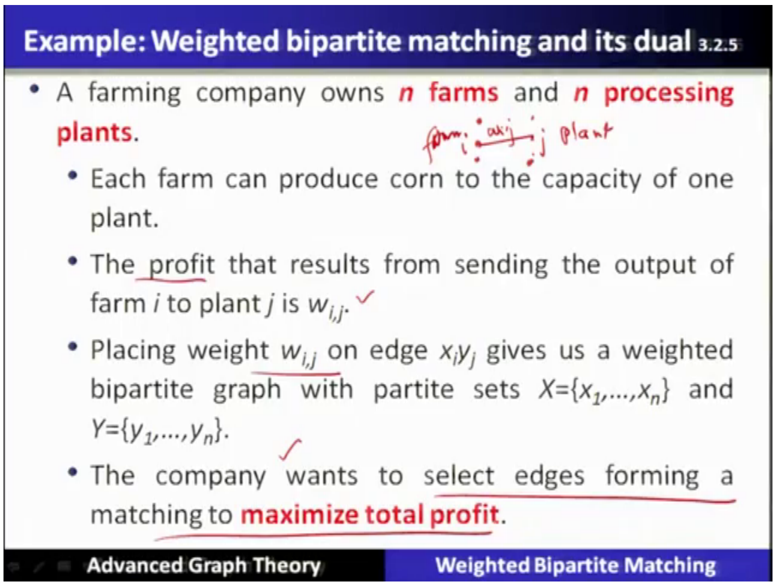
$\boxed{K_{n,n}}$ – Complete Bipartite Graph
- adding edges with weight 0's.

Advanced Graph Theory | Weighted Bipartite Matching

So, let us see the example of a problem setting. So, here this particular graph you see that it is K n n graph and these edges which were not there they were added with a weight 0 to make it as complete bipartite graph.

Now all other edges were having a weight. So, we have now obtained a weighted bipartite graph and now we will seek a perfect matching in this particular problem setting of a of a maximum total weight.

(Refer Slide Time: 05:35)



Example: Weighted bipartite matching and its dual 3.2.5

- A farming company owns n **farms** and n **processing plants**.
  - Each farm can produce corn to the capacity of one plant.
  - The profit that results from sending the output of farm i to plant j is $w_{i,j}$.
  - Placing weight $w_{i,j}$ on edge x_iy_j gives us a weighted bipartite graph with partite sets $X=\{x_1,\ldots,x_n\}$ and $Y=\{y_1,\ldots,y_n\}$.
  - The company wants to select edges forming a matching to **maximize total profit**.

Advanced Graph Theory | Weighted Bipartite Matching

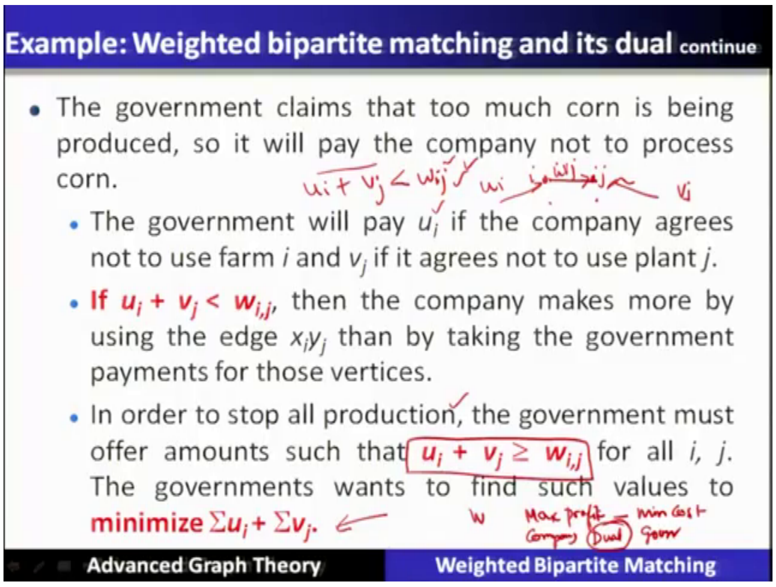
So, this kind of problem will arise in many practical situations one such situation we are going to see here in this particular problem setting that is a farming company will own let us say that n farms and n processing plant and each farm can produce the corn to the capacity of one plant.

So, the profit that results from sending the output of the farm to j we are going to represent it. So, this is let us say farm and this is the plant. So, let us say that if it is a farm i and it is sending it is produce to the plant j.

So, there will be an edge and the weights assigned to this edge is W i j and this particular W i j will indicate that the profit if the farm will send it is produce to the plant then the total profit earned will be mentioned as W i j. So, not only the edges we are going to add we are going to add it is weight also in this particular problem setting.

Now, this particular company wants to select those edges which will form a matching which will matching of a total of a total maximum profit.

(Refer Slide Time: 07:09)



Example: Weighted bipartite matching and its dual continue

- The government claims that too much corn is being produced, so it will pay the company not to process corn.
  $u_i + v_j < w_{ij}$ ✓ u_i $i \xrightarrow{w_{ij}} j$ v_j
  - The government will pay u_i if the company agrees not to use farm i and v_j if it agrees not to use plant j.
  - If $u_i + v_j < w_{i,j}$, then the company makes more by using the edge x_iy_j than by taking the government payments for those vertices.
  - In order to stop all production, the government must offer amounts such that $\boxed{u_i + v_j \geq w_{i,j}}$ for all i, j. The governments wants to find such values to minimize $\Sigma u_i + \Sigma v_j$. ←

W Max profit Company — $\boxed{\text{Dual}}$ — Min cost Govn

Advanced Graph Theory | Weighted Bipartite Matching

Furthermore the government claims that too much of corn is being produced. So, it will pay the company not to process the corn. The government will pay let us say unit u i if the company agrees not to use the farm i and v j if it agrees not to use plant j.

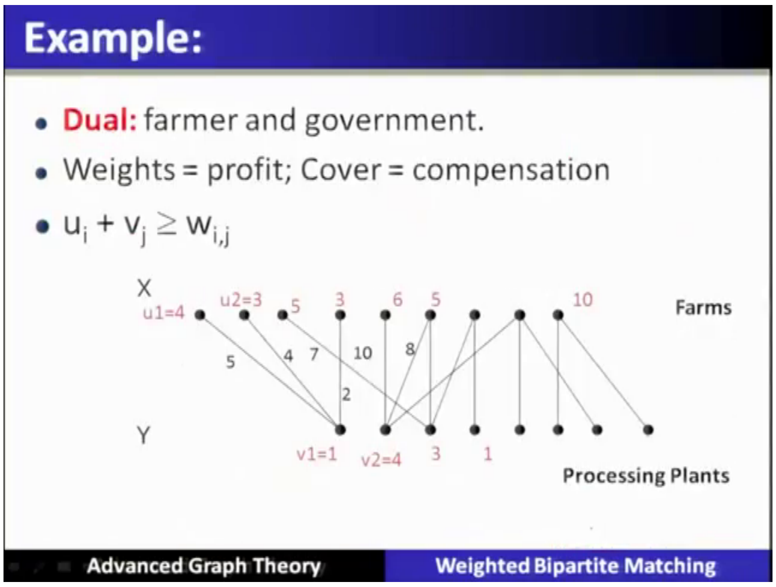
So, in the earlier problem setting we have seen that there is an edge of sending i to j. Now this particular i and j will also add and weights on the vertices. So, the government will pay u i if the company agrees not to use the farm i and it will use v j if it agrees not to use the plant j. So, it is u i and it is v j and this is W i j.

Now, there is a decision here in this case u i plus v j ; that means, to stop a production i and j should get a money and that is equivalent to u i plus v j and this particular money is compared to W i j now the plant who is making the profit if it is using the produce of a plant or a produce of a farm and manufacture it getting the profit of W i j , but if it is not doing it you will get u i plus v j since this particular value is less. So, it will try to the plant will try to make a profit company will make a profit and it will do the production..

So, in order to stop all the productions of a corn the government must offer the amounts which are basically at least W i j that means u i pus v j is at least W i j now then government wants this particular value is to be minimized. So, it is a dual problem; that means, the company wants to make the maximum profit and the government and the government wants to stop the production with the minimum cost bearing.

So, it is a dual problem and we are going to solve this duality problem using weighted bipartite matching. So, the algorithms and all those techniques which we are going to cover which will solve this kind of problems so all these things we will discuss again.

(Refer Slide Time: 09:57)



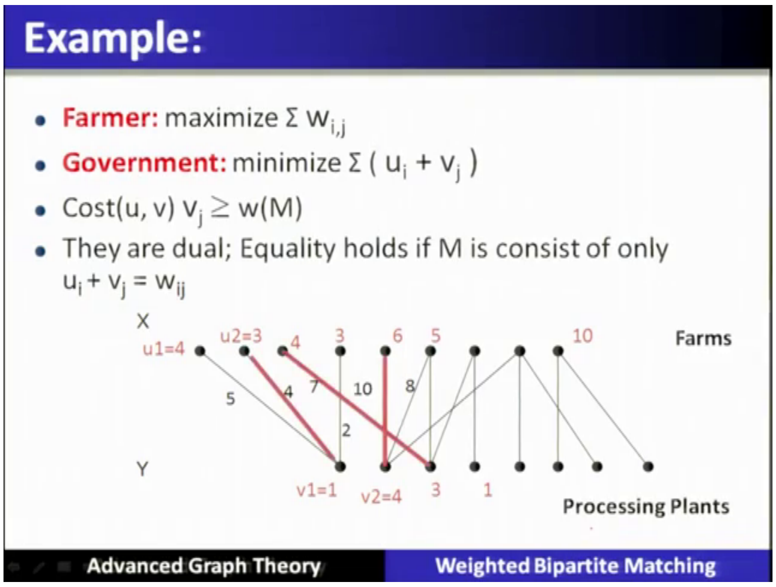
Example:

- **Dual:** farmer and government.
- Weights = profit; Cover = compensation
- $u_i + v_j \geq w_{i,j}$

X
u1=4
u2=3
5
3
6
5
10
Farms
5
4
7
10
8
2
Y
v1=1
v2=4
3
1
Processing Plants

Advanced Graph Theory Weighted Bipartite Matching

(Refer Slide Time: 10:00)



Example:

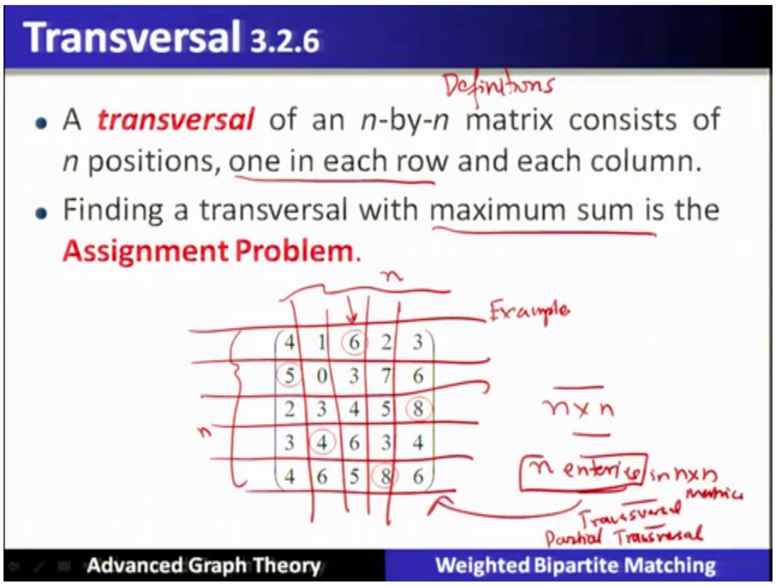
- **Farmer:** maximize $\Sigma\ w_{i,j}$
- **Government:** minimize $\Sigma\ (u_i + v_j)$
- Cost(u, v) $v_j \geq$ w(M)
- They are dual; Equality holds if M is consist of only $u_i + v_j = w_{ij}$

X
u1=4
u2=3
4
3
6
5
10
Farms
5
4
7
10
8
2
Y
v1=1
v2=4
3
1
Processing Plants

Advanced Graph Theory Weighted Bipartite Matching

After seeing the details of the algorithm.

(Refer Slide Time: 10:04)



Transversal 3.2.6

- A ***transversal*** of an *n*-by-*n* matrix consists of *n* positions, one in each row and each column.
- Finding a transversal with maximum sum is the **Assignment Problem.**

4	1	6	2	3
5	0	3	7	6
2	3	4	5	8
3	4	6	3	4
4	6	5	8	6

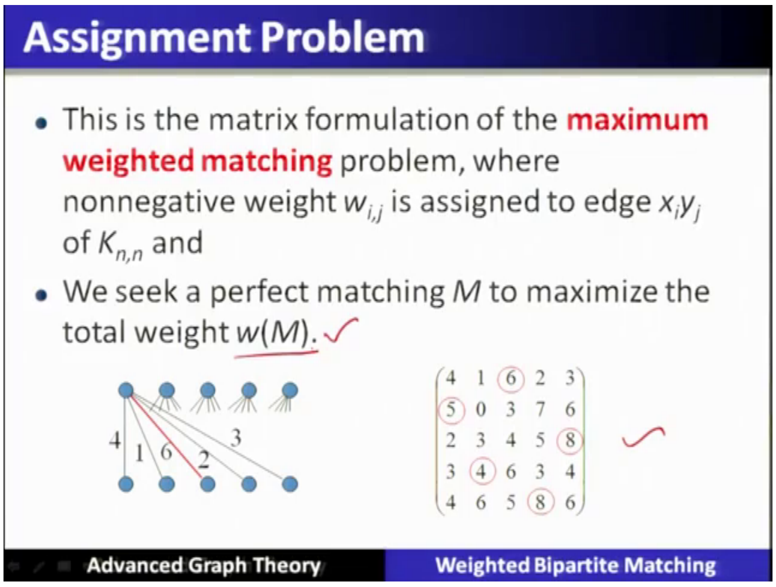
Definitions
n
Example
n
n x n
n entries in nxn matrix
Transversal
Partial Transversal
Advanced Graph Theory
Weighted Bipartite Matching

Let us see some of the definitions. So, transversal of n by n matrix consists of n positions, one in each row and one in each column so it is a n by n matrix is called transversal. If we can identify n positions; that means, if it is let us say in this example in every row there is only one particular element selected, in this row also there is one element selected, in this row also one element and so on.

So, in each row one element is selected similarly in column also one element is selected. So, in this way this particular matrix will have n entries year marked in n by n matrix and this particular setting in which we could identify n different positions in an n by n matrix then it is called a transversal.

If not n positions are identified, but following that every row or every column should basically fix up one position and not all n positions are identified then it is called a partial transversal these two terms which we are going to use it in this algorithm or in this discussion. So, finding a transversal with a maximum sum is nothing, but an assignment problem.

(Refer Slide Time: 12:01)



Assignment Problem

- This is the matrix formulation of the **maximum weighted matching** problem, where nonnegative weight $w_{i,j}$ is assigned to edge x_iy_j of $K_{n,n}$ and
- We seek a perfect matching M to maximize the total weight $\underline{w(M)}$. ✓

$$\begin{pmatrix} 4 & 1 & \boxed{6} & 2 & 3 \\ \boxed{5} & 0 & 3 & 7 & 6 \\ 2 & 3 & 4 & 5 & \boxed{8} \\ 3 & \boxed{4} & 6 & 3 & 4 \\ 4 & 6 & 5 & \boxed{8} & 6 \end{pmatrix}$$ ✓

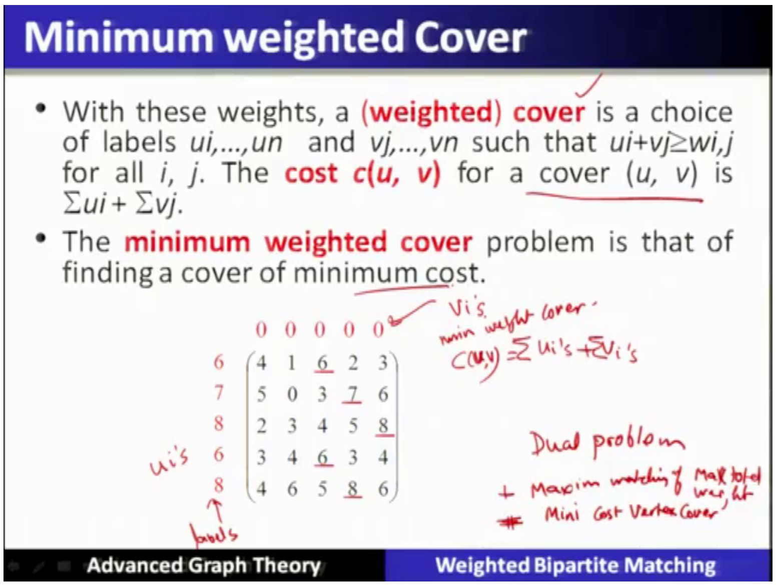
Advanced Graph Theory

Weighted Bipartite Matching

So, this is the matrix formulation of a maximum weighted bipartite matching where the nonnegative weights W i j assigned to the edges of complete bipartite graph K n, n and this will pose this will formulate the assignment problem. So, we seek in this particular problem setting a maximum matching or a perfect matching to maximize the total weight that is W of this M.

So, this particular transversal we have to identify so that when these particular n positions if you sum that will be having the maximum total weight, and that also can be represented in this particular graph.

(Refer Slide Time: 12:54)



Minimum weighted Cover

- With these weights, a **(weighted) cover** is a choice of labels *ui,…,un* and *vj,…,vn* such that *ui+vj≥wi,j* for all *i*, *j*. The **cost c(u, v)** for a cover (*u*, *v*) is Σ*ui* + Σ*vj*.
- The **minimum weighted cover** problem is that of finding a cover of minimum cost.

$$\begin{array}{c|ccccc} & 0 & 0 & 0 & 0 & 0 \\ \hline 6 & 4 & 1 & \underline{6} & 2 & 3 \\ 7 & 5 & 0 & 3 & \underline{7} & 6 \\ 8 & 2 & 3 & 4 & 5 & \underline{8} \\ 6 & 3 & 4 & \underline{6} & 3 & 4 \\ 8 & 4 & 6 & 5 & \underline{8} & 6 \end{array}$$

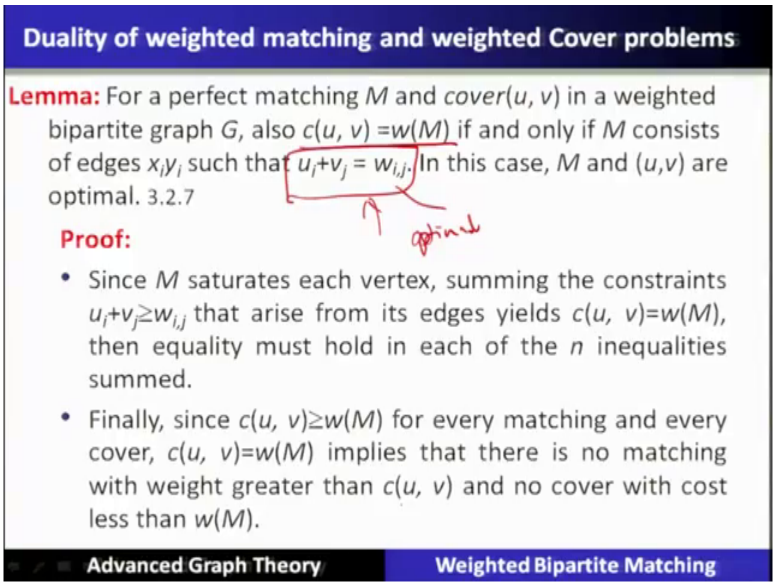
Advanced Graph Theory | Weighted Bipartite Matching

So, it is a matrix version of a weighted bipartite matching is called an assignment problem. Now we are solving the dual problem ; that means, when we find out the maximum matching of maximum total weight. Then we are solving minimum cost vertex cover also so both are dual problem..

If we solve maximum weighted matching then we are going to solve the minimum weighted cover or if we are solving the minimum weighted cover problem or given then we are going to solve the maximum matching of a maximum total weight. So, with this with these weights or weighted cover is a choice of these particular label.

So, these are the labels which we call them as u i s and these labels on this columns is called v i s and if you sum them that means, all u i s plus all v i s then it will form a cover and that is called a c u v cover. And this particular c u v cover is nothing, but summation of all u i s and summation of v i s we want to find out the this particular cover of a minimum weight that is finding a cost of a minimum cost minimum weight cover in this particular problem setting.

(Refer Slide Time: 14:55)



Duality of weighted matching and weighted Cover problems

Lemma: For a perfect matching M and $cover(u, v)$ in a weighted bipartite graph G, also $c(u, v) = w(M)$ if and only if M consists of edges $x_i y_i$ such that $\boxed{u_i+v_j = w_{i,j}}$. In this case, M and (u,v) are optimal. 3.2.7

optimal

Proof:

- Since M saturates each vertex, summing the constraints $u_i+v_j \geq w_{i,j}$ that arise from its edges yields $c(u, v)=w(M)$, then equality must hold in each of the n inequalities summed.
- Finally, since $c(u, v) \geq w(M)$ for every matching and every cover, $c(u, v)=w(M)$ implies that there is no matching with weight greater than $c(u, v)$ and no cover with cost less than $w(M)$.

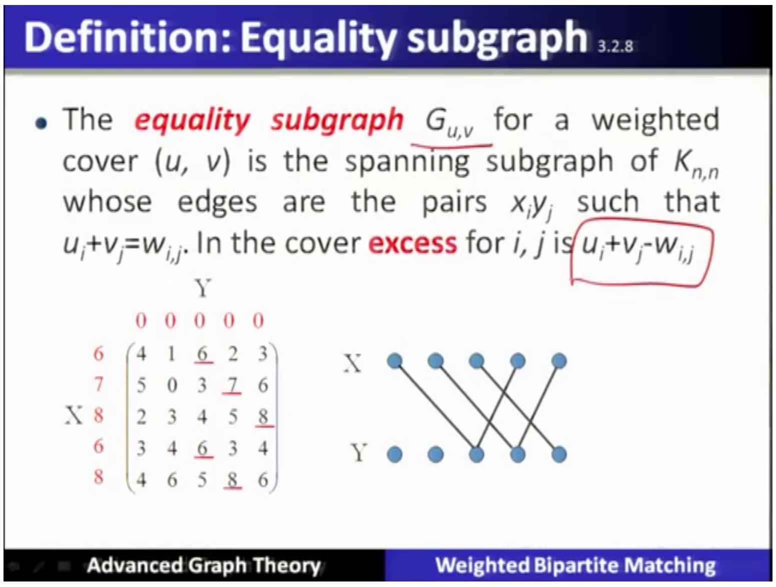
Advanced Graph Theory | Weighted Bipartite Matching

This duality of weighted matching and a weighted cover problem is expressed using a lemma. So, let us see the lemma for a perfect matching M and a cover u v in a weighted bipartite matching G, also is represented as c of u v is equal to the W of M if and only if M consists of the edges X i Y i such that u i plus v j is equal to W i j..

In this case M and u v are the optimal when u i plus v j is equal to W i j. Let us see the details since M saturates every vertex summing up the constraints u i plus v j is at least W i j that arises from the edges which will yield c u comma v and W of M ; that means, the equality must hold in each of n equalities which are summed up finally, when c u v that is the cover is at least the weight of the matching for every matching and every cover.

When this cover is equal to the weight of a matching which will imply that there is no matching with a weight greater than c u v and there is no cover with the cost less than W M that is this particular situation when it will arise this is an optimal situation ; that means, when we reach to this particular situation that will tell us that we have identified a maximum weighted matching with a cost of minimum vertex cover. Similarly minimum vertex cover which we have obtained with a cost of a maximum weighted matching both are equal.

(Refer Slide Time: 17:02)



Definition: Equality subgraph 3.2.8

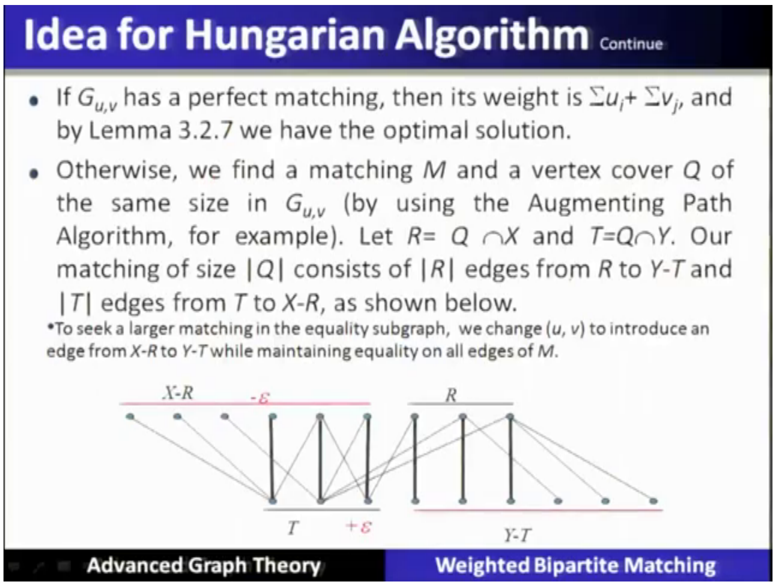
- The ***equality subgraph*** $G_{u,v}$ for a weighted cover (u, v) is the spanning subgraph of $K_{n,n}$ whose edges are the pairs x_iy_j such that $u_i+v_j=w_{i,j}$. In the cover **excess** for i, j is $\boxed{u_i+v_j-w_{i,j}}$

$$\begin{array}{cc|ccccc} & & & & Y & & \\ & & 0 & 0 & 0 & 0 & 0 \\ \hline & 6 & 4 & 1 & \underline{6} & 2 & 3 \\ & 7 & 5 & 0 & 3 & \underline{7} & 6 \\ X & 8 & 2 & 3 & 4 & 5 & \underline{8} \\ & 6 & 3 & 4 & \underline{6} & 3 & 4 \\ & 8 & 4 & 6 & 5 & \underline{8} & 6 \end{array}$$

Advanced Graph Theory | Weighted Bipartite Matching

Let us see more definitions equality subgraph equality subgraph is represented as G u, v for the weighted cover u, v is the spanning subgraph of a bipartite complete graph K n, n whose edges are the pairs X i Y j such that u i plus v j is equal to W i j. So, in the cover excess for i and j is nothing, but u i plus v j minus W i j is an excess. We will see this particular representation further.

(Refer Slide Time: 17:42)



Idea for Hungarian Algorithm Continue

- If $G_{u,v}$ has a perfect matching, then its weight is $\Sigma u_i + \Sigma v_j$, and by Lemma 3.2.7 we have the optimal solution.
- Otherwise, we find a matching M and a vertex cover Q of the same size in $G_{u,v}$ (by using the Augmenting Path Algorithm, for example). Let $R= Q \cap X$ and $T=Q\cap Y$. Our matching of size $|Q|$ consists of $|R|$ edges from R to Y-T and $|T|$ edges from T to X-R, as shown below.

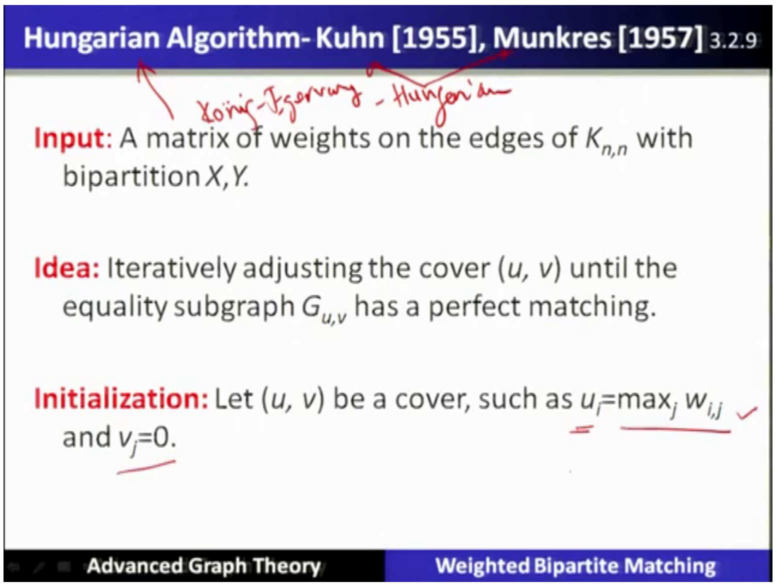
•To seek a larger matching in the equality subgraph, we change (u, v) to introduce an edge from X-R to Y-T while maintaining equality on all edges of M.

Advanced Graph Theory | Weighted Bipartite Matching

Now, in this particular graph which is called basically the equality subgraph. We have to seek for perfect matching and if this particular perfect matching with a weight equal to the c u v is equal to the so the matching of that weight if we have obtained then it is basically a optimal solution..

Otherwise we find matching M and a cover Q of same size by using augmenting path algorithm and we will seek a larger matching in the equality subgraph and change the matchings to get a maximum matching and reiterate so that we will basically obtain that maximum matching of a given graph and this is done through an algorithm.

(Refer Slide Time: 18:34)



Hungarian Algorithm- Kuhn [1955], Munkres [1957] 3.2.9

Konig-Egervary - Hungarian

Input: A matrix of weights on the edges of $K_{n,n}$ with bipartition X,Y.

Idea: Iteratively adjusting the cover (u, v) until the equality subgraph $G_{u,v}$ has a perfect matching.

Initialization: Let (u, v) be a cover, such as $u_i=\max_j w_{i,j}$ ✓ and $v_j=0$.

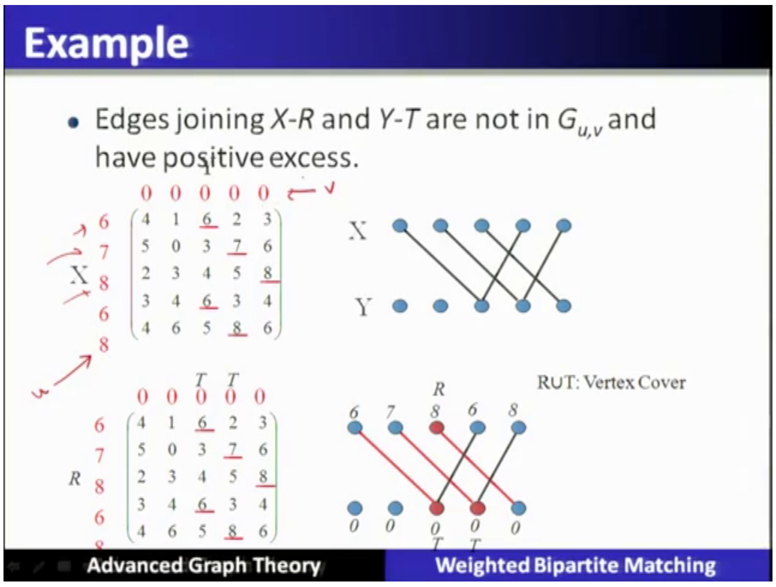
Advanced Graph Theory | Weighted Bipartite Matching

And that algorithm is called Hungarian algorithm which is named in the honour of Konig Egervary who was Hungarian and this algorithm was identified by Kuhn in 1955 and Munkres in 1957 both they have given together this independently this algorithms and this algorithm is basically named as Hungarian why because Konig Egervary theorem on which this algorithm is based.

So, in the honour of Konig Egervary who was Hungarian the algorithm was named as Hungarian algorithm. Input to this algorithm is a matrix of the weights on the edges of K n, n with the bipartition that we have already told. Now the idea of this algorithm is to iteratively adjust the cover until the equality subgraph has a perfect matching.

So, duality property will ensure that a perfect matching or a matching of a maximum weight has been achieved. So, initialization in this algorithm we will do that u v be the cover such that u i; that means, all the u i s are nothing, but the maximum of a particular row, in this particular matrix having these weights W i j and v j ; that means, on the columns the cover will become 0. Let us see this.

(Refer Slide Time: 20:20)



Example

- Edges joining $X\text{-}R$ and $Y\text{-}T$ are not in $G_{u,v}$ and have positive excess.

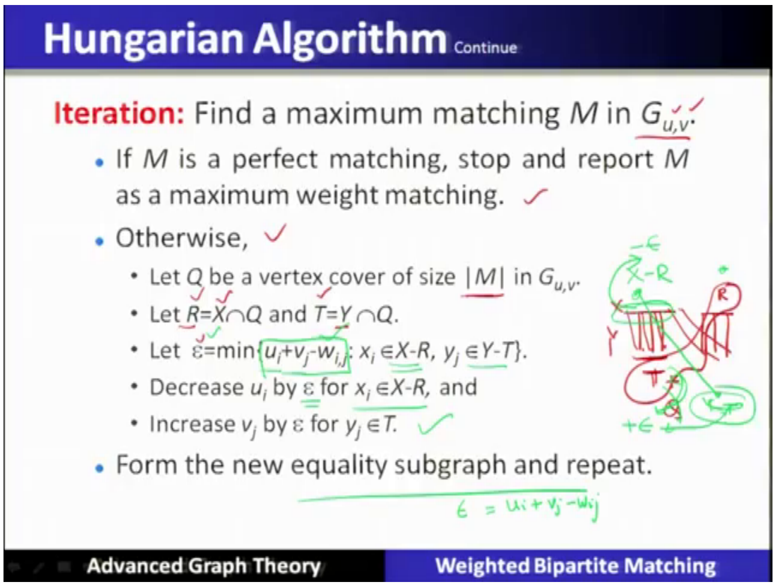
$$\begin{array}{c|ccccc} & 0 & 0 & 0 & 0 & 0 \\ \hline 6 & 4 & 1 & \underline{6} & 2 & 3 \\ 7 & 5 & 0 & 3 & \underline{7} & 6 \\ 8 & 2 & 3 & 4 & 5 & \underline{8} \\ 6 & 3 & 4 & \underline{6} & 3 & 4 \\ 8 & 4 & 6 & 5 & \underline{8} & 6 \end{array}$$

$$\begin{array}{c|ccccc} & 0 & 0 & \overset{T}{0} & \overset{T}{0} & 0 \\ \hline 6 & 4 & 1 & \underline{6} & 2 & 3 \\ 7 & 5 & 0 & 3 & \underline{7} & 6 \\ R\ 8 & 2 & 3 & 4 & 5 & \underline{8} \\ 6 & 3 & 4 & \underline{6} & 3 & 4 \\ 8 & 4 & 6 & 5 & \underline{8} & 6 \end{array}$$

Advanced Graph Theory | Weighted Bipartite Matching

So, here we can see u i s u i j; that means, in this particular row this is the maximum one. So, in every row the maximum value is being taken up as the cover u, similarly in all the columns it is initialized to be 0. So, these particular values will give the cover.

(Refer Slide Time: 20:51)



Hungarian Algorithm Continue

Iteration: Find a maximum matching M in $G_{u,v}$.

- If M is a perfect matching, stop and report M as a maximum weight matching.
- Otherwise,
  - Let Q be a vertex cover of size $|M|$ in $G_{u,v}$.
  - Let $R=X\cap Q$ and $T=Y\cap Q$.
  - Let $\varepsilon=\min\{\boxed{u_i+v_j-w_{i,j}}: x_i\in X\text{-}R,\ y_j\in Y\text{-}T\}$.
  - Decrease u_i by ε for $x_i\in X\text{-}R$, and
  - Increase v_j by ε for $y_j\in T$.
- Form the new equality subgraph and repeat.

$\epsilon = u_i + v_j - w_{ij}$

Advanced Graph Theory | Weighted Bipartite Matching

So, having initialised the cover now we have to see the iteration of Hungarian algorithm. Now in the iteration we find maximum matching M in that equality subgraph G u, v u and v which we have initialized and formed. Now if M is a perfect matching then we stop and report M as the maximum weighted matching..

Otherwise what you will do since q, is the vertex cover of size M in G u, v. So, let us find out the set of vertices of this vertex cover on the X partite set as R and the set of vertex cover on why it is called as T R and T. Then the next step in this algorithm is to find out an epsilon so if let us say that it is X and this is Y let me explain you.

So, here that is a T, and there it is basically R, and all other notes are basically touching them like this. So, T plus R together will form a vertex cover that is called Q here we have to find out an edge this way this is Y minus T and this is X minus R.

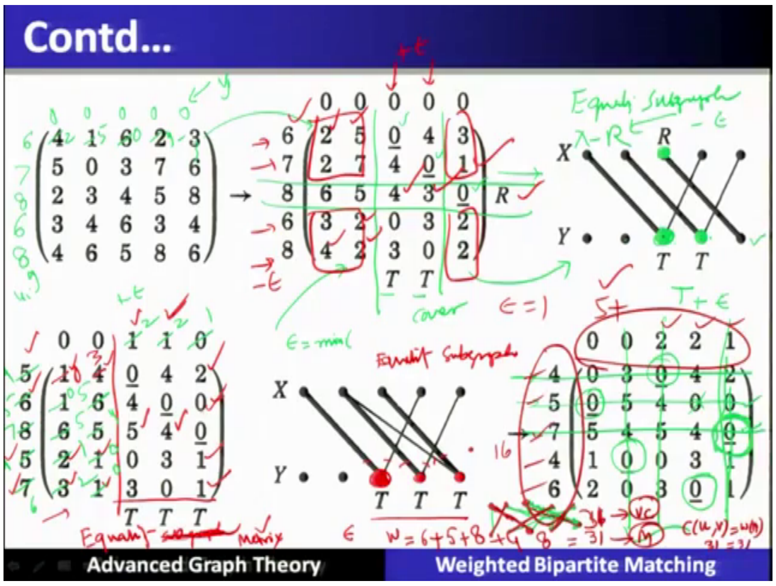
We have to find out an edge which is crossing from X minus R and touching another vertex in Y minus T and we will find out adding this particular edge is possible if you decrease this value by minus epsilon and increase here the value of u by increase the value of v j by epsilon here we have to increase and here we have to decrease then only this new edge can be entered into a matching.

So, epsilon will be ascertained with u i plus v j minus W i j for these values that is for X minus R these elements and for Y minus T these elements and we will find out the gap that is u i plus v j minus W i j. We will find out an epsilon and then we will reduce u i s which are there in X minus R; that means, these sets with these values minus epsilon..

Similarly we are going to increase the values of Y minus T by an epsilon. So, epsilon is to be increased here in T now then we will form a new equality subgraph in this particular modifications by epsilon and then you will repeat this iterations.

Let us take an example we have initialized to u and v then what we have to do? Then we have to find out equality subgraph in equality subgraph this is going to be reduced with an excess weights.

(Refer Slide Time: 24:48)


Contd...

Let us see the entire thing in this particular setting. So, here in this particular example let us say this is the K n, n and the edge weights are given and let us initialized in this particular row it is 6, it is 7, then it is 8, then it is 6, then it is 8, and on the columns is going to be initialized.

So, this is the initialization of v j is 0 and u i is the maximum of the edge weights in every row. Now we will find out the equality subgraph. So, xs values we have to basically say 6 minus 4 is 2; 6 minus 1 is 5; 6 minus 6 is 0; 6 minus 2 is 4; 6 minus 3 is 3.

So, just see that 2 5 0 4 and 3; that means, whatever we have done the modification is already given in this particular matrix. Similarly all other column rows and columns of the remaining matrix can be done. Then in this particular equality subgraph we have to find out a cover. So, for that we can draw the equality subgraph..

So, in this particular equality subgraph; that means, wherever zeroes are there; that means, in the first in the first row so wherever 0 is there 123 so it is at this end so we are going to add an edge similarly this edge will be added second row, and 1 2 3 4, 1 2 3 4.

So, this particular edge is added, similarly this edge will be added, and then this edge will be added and finally, this edge will be added. So, from this particular graph we will obtain a equality subgraph of all the elements are basically placed with a 0 values. Now we have to find out a cover.

So, in this cover we can see that this particular vertex will cover these 2 edges, this vertex will cover 2 edges. So, these two 1 2 3 4 edges are covered by these two vertices and this particular vertex will cover this edge. So, all the areas are covered with 3 different vertices. So, the cover on Y side we call it as T the cover on X side we call it as R.

So, we can draw like this. So, the remaining part so the remaining part what we are going to see we have to find out we have to find out the epsilon is the minimum value of these remaining part. So, these remaining will be from X minus R and from T.

So, here we are in T we are going to increase the epsilon value and here we are going to decrease with the epsilon value and epsilon we have to find out of them. So, from this particular column from this particular column this column and this column if you see what is the minimum value so minimum value is 1. So, epsilon here becomes 1 so 1 will be reduced from these values.

So, they will become 1, this will become 4, this is not going to be touched so this will become 2 and 1 will become 0..

Similarly 3 will become 2, and 2 will become 1, 4 will become 3, and 2 will become 1, similarly 2 will become 1 and this 2 will become 1. So, it will be increased the values to T.

So, here in this particular two T's that is 4 will become 5, and 3 will become 4, and that is all. So, this will become the equality subgraph and also here on these particular values also decreased minus epsilon and this here will be increased with epsilon. So, here it will become 1, and here also it will become 1 wherever T's are there 2 T's are there so two ones will be there..

And here 6 will become 5 to be decreased 7 will become 6 this is untouched why because R is there and 6 will become 5 and 8 will become 7 so this is all completely explained. So, this becomes an equality subgraph in this equality subgraph again we have to draw a equality subgraph..

So, equality matrix it is called and we have to draw equality subgraph ; that means, all zeroes wherever zeroes are there we have to place an edge. So, just see that only this edge is extra appearing and all other zeroes are same. So, one more 0 is appearing here why because when one will become 0 So, only this particular extra edge will be there ; that means, from second to this particular edge is represented over here..

Now in this equality subgraph again we have to find out the cover the vertex cover this particular vertex we will cover these 2 edges, these vertex will cover these 2 edges, this vertex will cover these 2 edges. So, you see that from R all the vertices basically will be covering so 3 vertex will cover all the edges so this will become T.

Now, we have to process as we have processed here. So, there will be 3 T's. Now we have to find out the epsilon and so epsilon if you see it will be again 1. So, 1 minus 1 is 0, 4 minus 1 is 3; then here 1 will become 0; 6 will become 5 6 will become 5 4 1 0 0 2 and here also 4 5 7 4 and 6.

Similarly here it will be all T's are there then plus epsilon will be added 1 plus 1 is 2; 1 plus 1 is 2; 0 plus 1 is 1. So, we have obtained this particular graph and in this particular graph we will do the further processing let us see the steps yeah this will be your final graph.

So, in this final graph we can see that we have to add using this particular equality subgraph we will obtain the matching let us see go ahead. Now wherever zeroes are there we will mark in the first row 1 and 2; so 1 and 3 then second one is first 1, then last 2, then third one is the last, then fourth one is 2 and 3, fifth one is 2 and 4. Now we have to find out the matching or the covers.

So, we can basically see that that this particular vertex this particular vertex is only having 1 edge. So, this particular edge definitely will be there in the matching so having contained this particular edge in the matching. So, this particular edge will be not there in the matching..

And once this particular element is there in the matching then entire row; that means, all other 0. So, this 0 is eliminated and this column is also eliminated that means, this row and this column has now 1 0 included this 0.

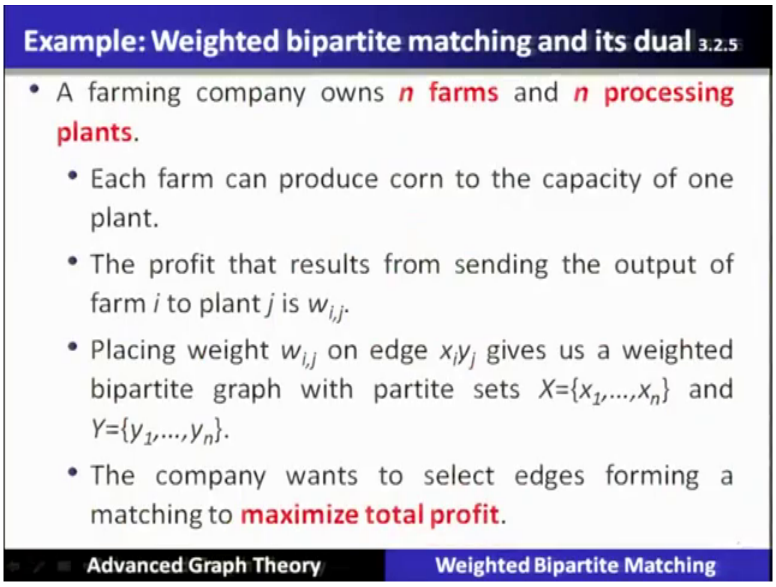
Similarly, if you take this particular row we have two zeroes we have to select one of them and this particular column we have to select one of them. So, in this particular row this 0 is selected. So, if 0 is selected then this particular column will go out and this column this row will also go out. So, this will not be there this will also will not be there in the matching so this one. So, there is only one 0 remains. So, here this 0 also will be there in the matching.

So, if this 0 is in the matching so this 0 will be eliminated. Now there comes this particular column this 0 is included in the matching. So, having included this particular matching only this particular matching remains. So, if you see the diagram the matching. So, 1 will be matched with 2, then 2 will be matched to one this edge then third will be matched to the last edge..

Third one will be matched to the to the last edge then fourth will be matched to the second one and then sixth that is the last one will be matched to last, but 1 this particular. So, this will become the matching and we can find out the total weight, total weight of this particular matching will be 0 that is 1 2 3 1 to third is 6 plus 5 plus it is 8, then it is 4, then it is 8. So, it comes out to be 31..

So, we have identified the maximum matching is in Hungarian algorithm and we have also estimated the cost that comes out to be 31. So, if we add the vertex cover which is obtained this is 5 and plus 16 plus 5 that becomes to be 31 is given by the minimum vertex cover and this is the matchings. So, here when the problem converges to the optimal then basically the vertex cover is equal to the to the to the matching and both are 31 here in this particular example.

(Refer Slide Time: 36:56)



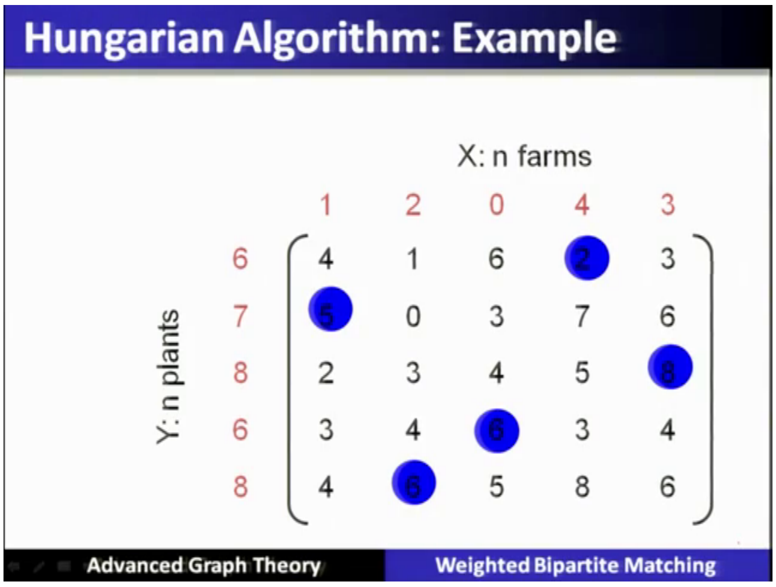
Example: Weighted bipartite matching and its dual 3.2.5

- A farming company owns ***n* farms** and ***n* processing plants**.
    - Each farm can produce corn to the capacity of one plant.
    - The profit that results from sending the output of farm *i* to plant *j* is $w_{i,j}$.
    - Placing weight $w_{i,j}$ on edge x_iy_j gives us a weighted bipartite graph with partite sets $X=\{x_1,\ldots,x_n\}$ and $Y=\{y_1,\ldots,y_n\}$.
    - The company wants to select edges forming a matching to **maximize total profit**.

Advanced Graph Theory | Weighted Bipartite Matching

Let us take a example we have started that is the farming company will own n plants and n processing units. Let us see this dual formulated problem how we are going to solve using Hungarian method.

(Refer Slide Time: 37:13)



Hungarian Algorithm: Example

X: n farms

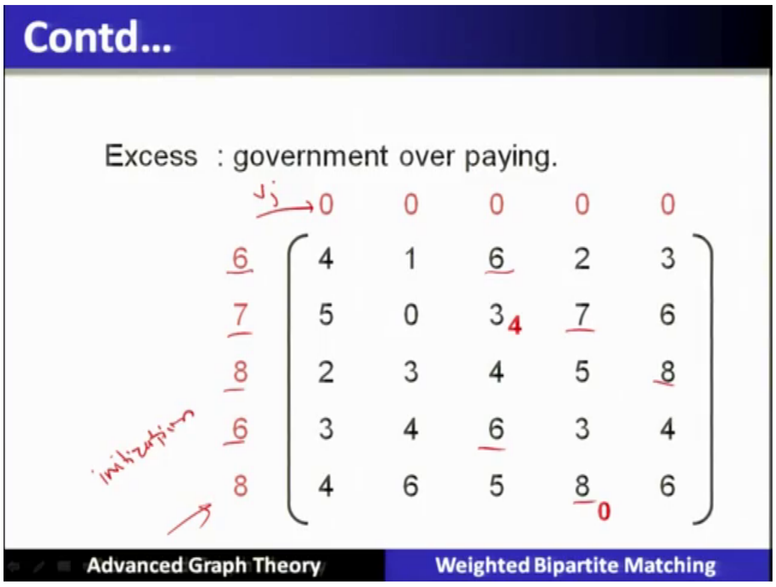
Y: n plants

$$\begin{array}{c|ccccc} & 1 & 2 & 0 & 4 & 3 \\ \hline 6 & 4 & 1 & 6 & \boxed{2} & 3 \\ 7 & \boxed{5} & 0 & 3 & 7 & 6 \\ 8 & 2 & 3 & 4 & 5 & \boxed{8} \\ 6 & 3 & 4 & \boxed{6} & 3 & 4 \\ 8 & 4 & \boxed{6} & 5 & 8 & 6 \end{array}$$

Advanced Graph Theory | Weighted Bipartite Matching

Let us represent this particular graph G u v having n farms and n plants and let us take the all these particular steps of this particular graph. So, this is basically the initialization.

(Refer Slide Time: 37:32)



Contd...

Excess : government over paying.

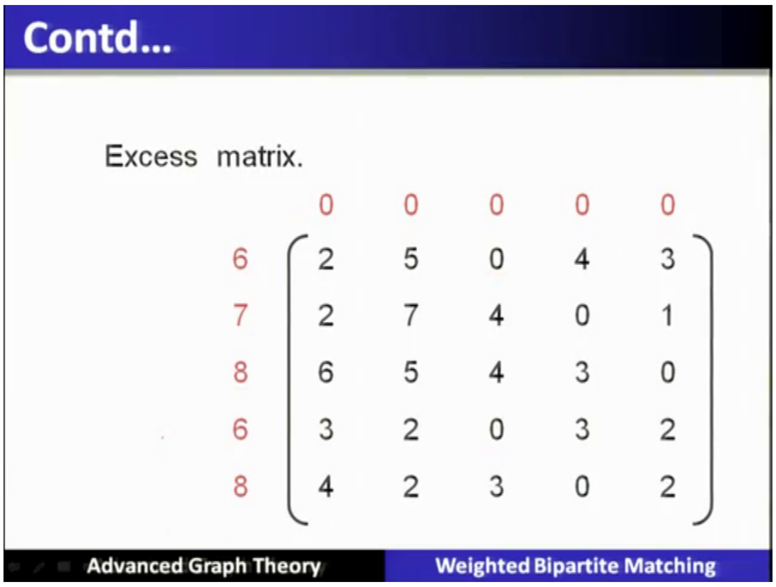
$$\begin{array}{c|ccccc} v_j \rightarrow & 0 & 0 & 0 & 0 & 0 \\ \hline \underline{6} & 4 & 1 & \underline{6} & 2 & 3 \\ \underline{7} & 5 & 0 & 3_{4} & \underline{7} & 6 \\ \underline{8} & 2 & 3 & 4 & 5 & \underline{8} \\ \underline{6} & 3 & 4 & \underline{6} & 3 & 4 \\ 8 & 4 & 6 & 5 & \underline{8}_{0} & 6 \end{array}$$

initialization

Advanced Graph Theory | Weighted Bipartite Matching

So, here on the vertex side v j s we are initializing in to zero and here you see that this is the maximum. So, maximum is kept over here and it is 7 7 is the maximum, 8 is the maximum, 6 is the maximum, 8 is the maximum. So, this is the initialization part of the algorithm which we have completed.

(Refer Slide Time: 38:06)



Contd...

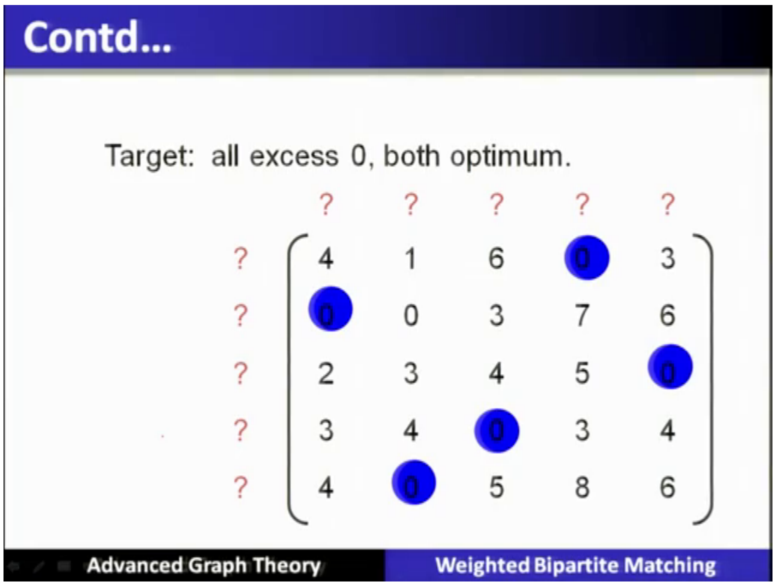
Excess matrix.

$$\begin{array}{c|ccccc} & 0 & 0 & 0 & 0 & 0 \\ \hline 6 & 2 & 5 & 0 & 4 & 3 \\ 7 & 2 & 7 & 4 & 0 & 1 \\ 8 & 6 & 5 & 4 & 3 & 0 \\ 6 & 3 & 2 & 0 & 3 & 2 \\ 8 & 4 & 2 & 3 & 0 & 2 \end{array}$$

Advanced Graph Theory | Weighted Bipartite Matching

Now, this particular matrix which we have obtained is called an excess matrix.

(Refer Slide Time: 38:12)



Contd...

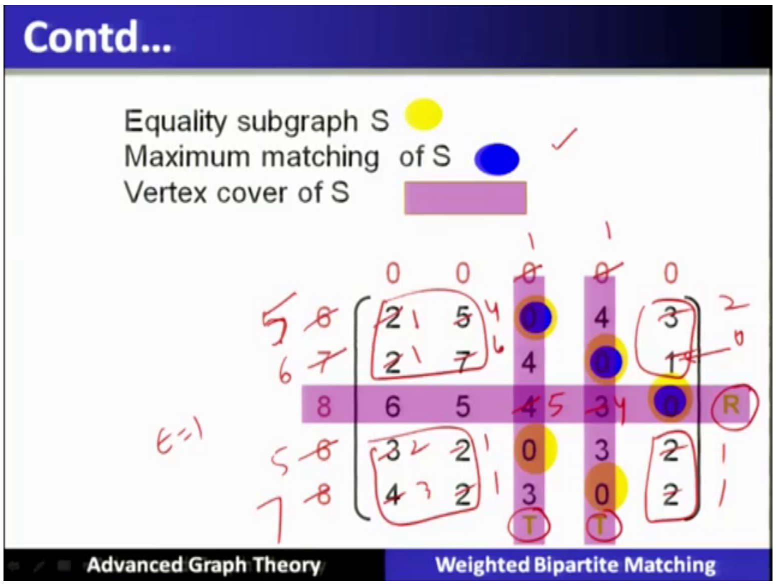
Target: all excess 0, both optimum.

$$\begin{array}{c|ccccc} & ? & ? & ? & ? & ? \\ \hline ? & 4 & 1 & 6 & 0 & 3 \\ ? & 0 & 0 & 3 & 7 & 6 \\ ? & 2 & 3 & 4 & 5 & 0 \\ ? & 3 & 4 & 0 & 3 & 4 \\ ? & 4 & 0 & 5 & 8 & 6 \end{array}$$

Advanced Graph Theory | Weighted Bipartite Matching

Now, we have to identify the matchings.

(Refer Slide Time: 38:21)



Contd...

Equality subgraph S ●

Maximum matching of S ● ✓

Vertex cover of S ▬

$$\begin{array}{cc|ccccc|c} & & 0 & 0 & \overset{1}{\cancel{0}} & \overset{1}{\cancel{0}} & 0 & \\ \hline 5 & \cancel{8} & \cancel{2}\,1 & \cancel{5}\,4 & 0 & 4 & \cancel{3}\,2 & \\ 6 & \cancel{7} & \cancel{2}\,1 & \cancel{7}\,6 & 4 & 0 & \cancel{1}\,0 & \\ & 8 & 6 & 5 & \cancel{4}\,5 & \cancel{3}\,4 & 0 & R \\ 5 & \cancel{8} & \cancel{3}\,2 & \cancel{2}\,1 & 0 & 3 & \cancel{2}\,1 & \\ 7 & \cancel{8} & \cancel{4}\,3 & \cancel{2}\,1 & 3 & 0 & \cancel{2}\,1 & \\ \hline & & & & T & T & & \end{array}$$

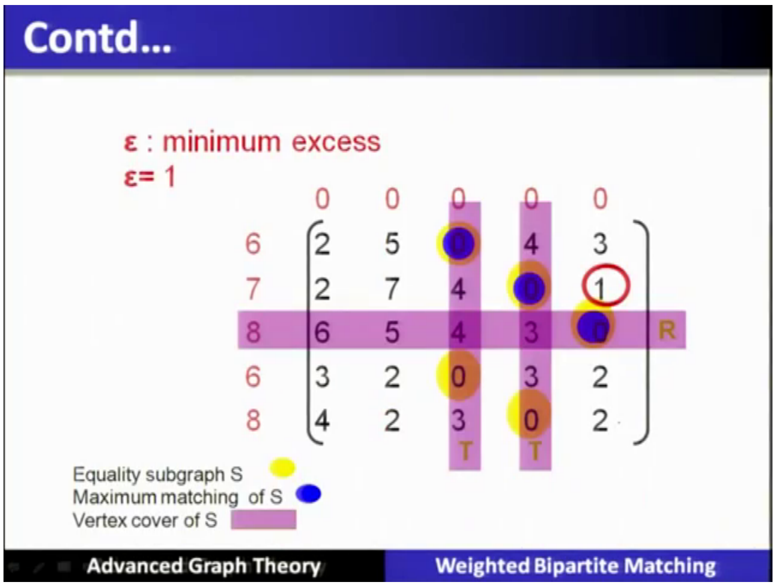
$\epsilon = 1$

Advanced Graph Theory | Weighted Bipartite Matching

In this particular matching we have seen this is a partial transversal why because not all columns and not all rows are being basically covered in this particular matching. So, we have to go to the next step and in this particular step we have to identify T and R. So, the remaining part we have to identify the epsilon and epsilon is 1.

So, if epsilon is 1; then we have to increase this particular value or we have to increase this value of T and we have to decrease this value by 1. And also these values are decreased. So, this becomes 0. So, one edge will be entered and here these values will be increased.

(Refer Slide Time: 39:36)



Contd...

ε : minimum excess
ε= 1

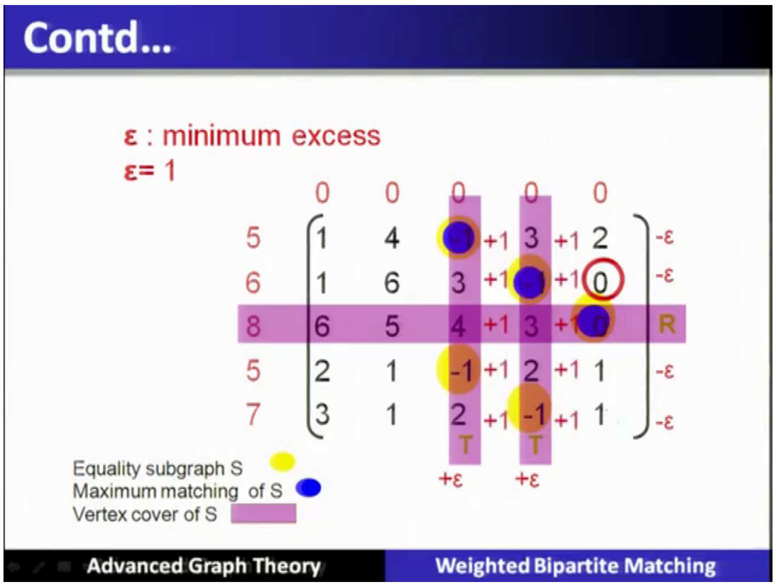
$$\begin{array}{c|ccccc|c}
 & 0 & 0 & 0 & 0 & 0 & \\
\hline
6 & 2 & 5 & 0 & 4 & 3 & \\
7 & 2 & 7 & 4 & 0 & 1 & \\
8 & 6 & 5 & 4 & 3 & 0 & R \\
6 & 3 & 2 & 0 & 3 & 2 & \\
8 & 4 & 2 & 3 & 0 & 2 & \\
 & & & T & T & &
\end{array}$$

Equality subgraph S ●
Maximum matching of S ●
Vertex cover of S ▨

Advanced Graph Theory | Weighted Bipartite Matching

So, same thing whatever we have seen earlier.

(Refer Slide Time: 39:38)



Contd...

ε : minimum excess
ε= 1

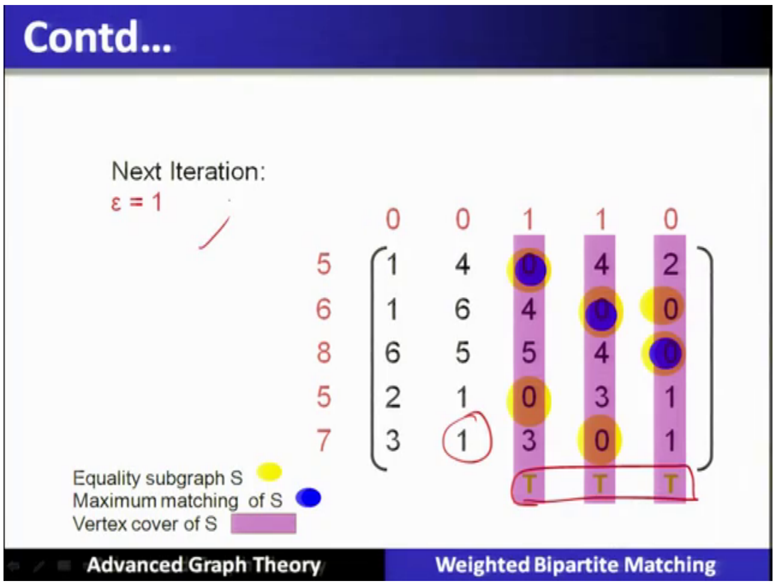
$$\begin{array}{c|ccccccc|c}
 & 0 & 0 & 0 & & 0 & & 0 & \\
\hline
5 & 1 & 4 & -1 & +1 & 3 & +1 & 2 & -\varepsilon \\
6 & 1 & 6 & 3 & +1 & -1 & +1 & 0 & -\varepsilon \\
8 & 6 & 5 & 4 & +1 & 3 & +1 & 0 & R \\
5 & 2 & 1 & -1 & +1 & 2 & +1 & 1 & -\varepsilon \\
7 & 3 & 1 & 2 & +1 & -1 & +1 & 1 & -\varepsilon \\
 & & & T & & T & & & \\
 & & & +\varepsilon & & +\varepsilon & & &
\end{array}$$

Equality subgraph S ●
Maximum matching of S ●
Vertex cover of S ▨

Advanced Graph Theory | Weighted Bipartite Matching

This is shown in 2 3 more steps.

(Refer Slide Time: 39:41)



Contd...

Next Iteration:

ε = 1

$$\begin{array}{c|ccccc} & 0 & 0 & 1 & 1 & 0 \\ \hline 5 & 1 & 4 & 0 & 4 & 2 \\ 6 & 1 & 6 & 4 & 0 & 0 \\ 8 & 6 & 5 & 5 & 4 & 0 \\ 5 & 2 & 1 & 0 & 3 & 1 \\ 7 & 3 & 1 & 3 & 0 & 1 \\ & & & T & T & T \end{array}$$

Equality subgraph S ●

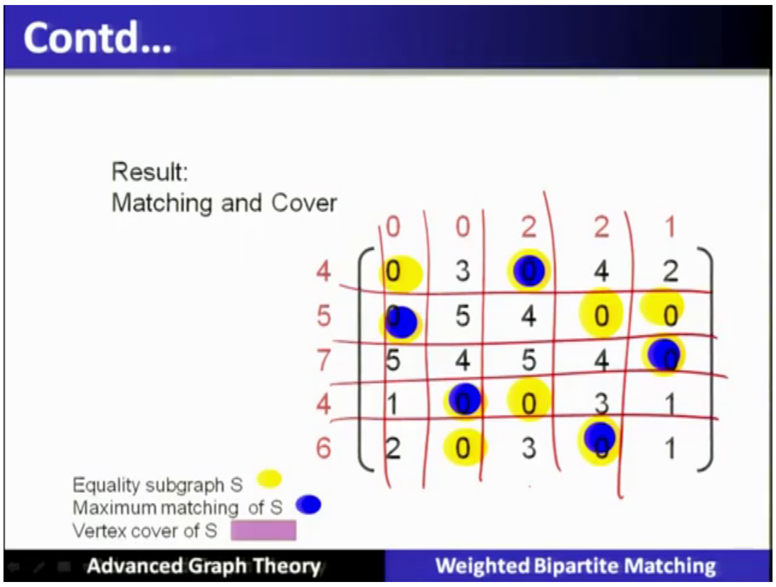
Maximum matching of S ●

Vertex cover of S ▨

Advanced Graph Theory | Weighted Bipartite Matching

Now, we have identified the next particular T and in the equality matrix we have to identify the epsilon and it will go to the next steps.

(Refer Slide Time: 39:55)



Contd...

Result:

Matching and Cover

$$\begin{array}{c|ccccc} & 0 & 0 & 2 & 2 & 1 \\ \hline 4 & 0 & 3 & 0 & 4 & 2 \\ 5 & 0 & 5 & 4 & 0 & 0 \\ 7 & 5 & 4 & 5 & 4 & 0 \\ 4 & 1 & 0 & 0 & 3 & 1 \\ 6 & 2 & 0 & 3 & 0 & 1 \end{array}$$

Equality subgraph S ●

Maximum matching of S ●

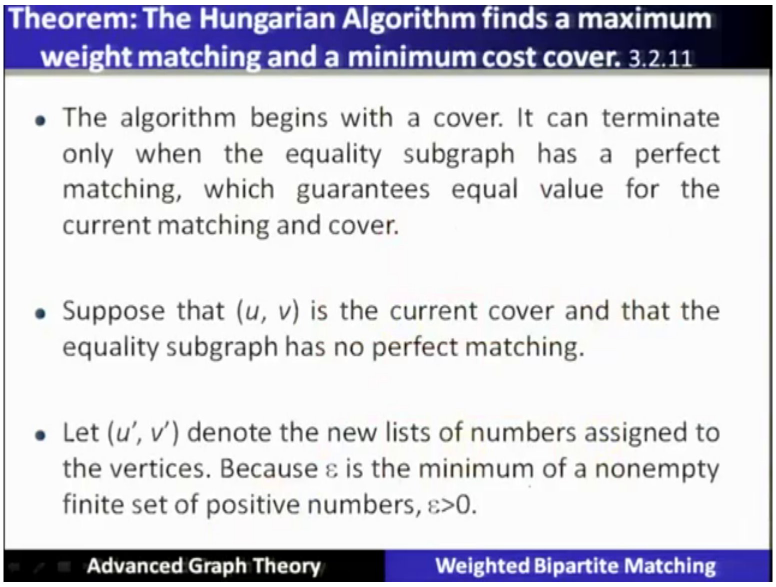
Vertex cover of S ▨

Advanced Graph Theory | Weighted Bipartite Matching

So, in the next step all these values will become 0, and in this particular equality graph we have identified that you see that in a every row in a every column there is a there is a 0 which is identified.

Now, we have to identify among these zeros all zeros in one different row and one different columns are being placed so that is basically the solution of an assignment problem or a Hungarian problem.

(Refer Slide Time: 40:26)

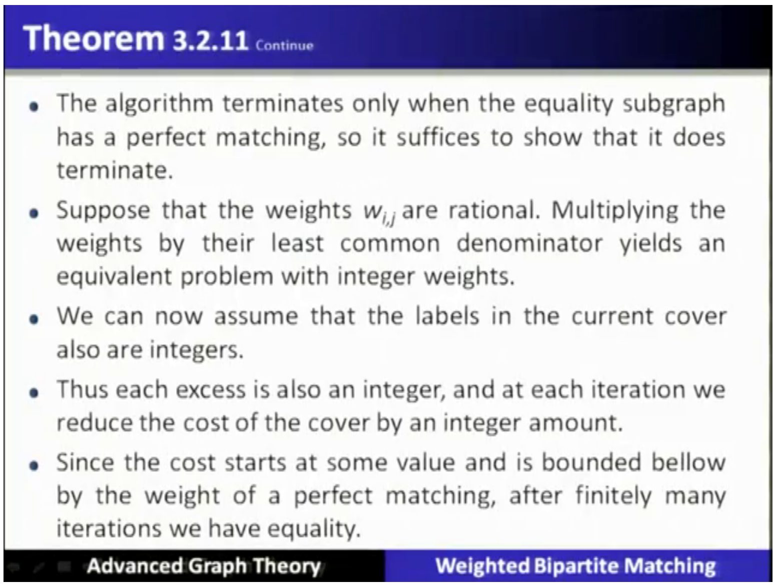


Theorem: The Hungarian Algorithm finds a maximum weight matching and a minimum cost cover. 3.2.11

- The algorithm begins with a cover. It can terminate only when the equality subgraph has a perfect matching, which guarantees equal value for the current matching and cover.
- Suppose that (u, v) is the current cover and that the equality subgraph has no perfect matching.
- Let (u', v') denote the new lists of numbers assigned to the vertices. Because ε is the minimum of a nonempty finite set of positive numbers, $\varepsilon>0$.

Advanced Graph Theory Weighted Bipartite Matching

(Refer Slide Time: 40:27)



Theorem 3.2.11 Continue

- The algorithm terminates only when the equality subgraph has a perfect matching, so it suffices to show that it does terminate.
- Suppose that the weights $w_{i,j}$ are rational. Multiplying the weights by their least common denominator yields an equivalent problem with integer weights.
- We can now assume that the labels in the current cover also are integers.
- Thus each excess is also an integer, and at each iteration we reduce the cost of the cover by an integer amount.
- Since the cost starts at some value and is bounded bellow by the weight of a perfect matching, after finitely many iterations we have equality.

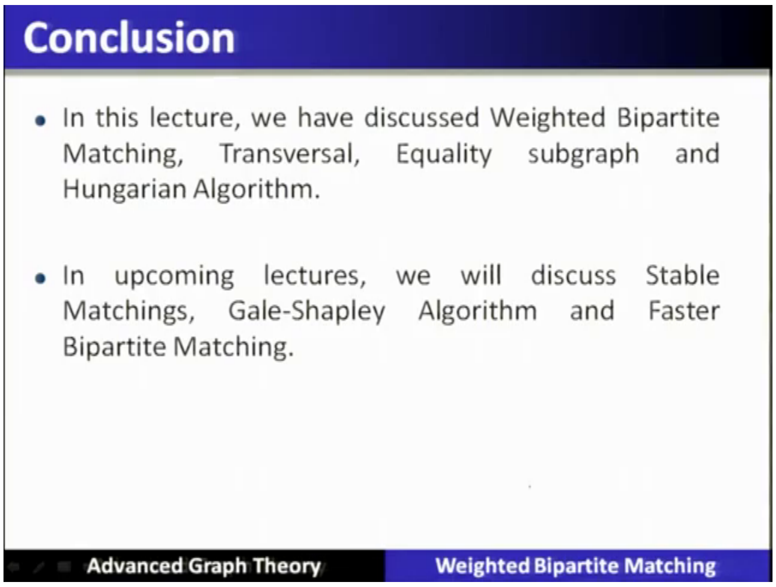
Advanced Graph Theory Weighted Bipartite Matching

So, whatever we have explained is basically is also proved in this particular theorem. So, let us see the details of this theorem this algorithm terminates only by an equality subgraph has a perfect matching. So, it suffices to show that it does terminate..

Suppose weights i j are rational then multiplying the weights by the least common denominator is an equivalent problem in the integer weights. We can now assume that the labels in the current cover are also integer thus excess is also an integer.

So, we convert any other given problem in our problem setting of the non negative edge weights and that to integer as weights and we can run this particular algorithm other problem settings as well.

(Refer Slide Time: 41:17)



Conclusion

- In this lecture, we have discussed Weighted Bipartite Matching, Transversal, Equality subgraph and Hungarian Algorithm.
- In upcoming lectures, we will discuss Stable Matchings, Gale-Shapley Algorithm and Faster Bipartite Matching.

Advanced Graph Theory Weighted Bipartite Matching

So, conclusion in this lecture we have discussed weighted bipartite matching transversal equality subgraph Hungarian algorithm in upcoming lectures. We will discuss stable matchings, Gale-shapley algorithm and a faster bipartite matching.

Thank you.