

Lecture 10: February 6

Scribe: Debtanu Pal

Lecturer: Rohit Gurjar

1 Maximum Matching Problem as a Linear Program

Definition 10.1. (*Maximum Matching Problem*) Given a graph $G(V, E)$, a matching $M \subseteq E$ is subset of edges such that every vertex has at most one edge incident from M . The objective of the maximum matching Problem is to find a **set of matching edges with maximum cardinality**.

1.1 Integer Programming Formulation of the Matching Problem

We can write an integer programming formulation of the maximum matching Problem. Consider a variable X_e for each edge $e \in E$ of the graph $G(V, E)$. Any matching $M \subseteq E$ can be represented by a point $X \in \mathbb{R}^E$ such that $X_e = 1$ if the edge $e \in M$; $X_e = 0$ otherwise. To restrict the solution set to only matching points, we will be adding one constraint for each of the vertices of the graph $G(V, E)$. These constraints will make sure that there is only one edge is incident to each of the vertices i.e., the set of edges obtained is indeed a matching. We can write the above conditions as an integer program as :

$$\begin{aligned} \max \quad & \sum_{e \in E} X_e \\ \forall e \in E \quad & X_e \in \{0, 1\} \\ \forall v \in V \quad & \sum_{e \in \delta(v)} X_e \leq 1. \end{aligned}$$

where, $\delta(v)$ is the set of edges incident on the vertex $v \in V$.

But as we know, integer programming is NP-hard, we need to find a better way (convert this into a linear program perhaps) to solve the maximum matching Problem.

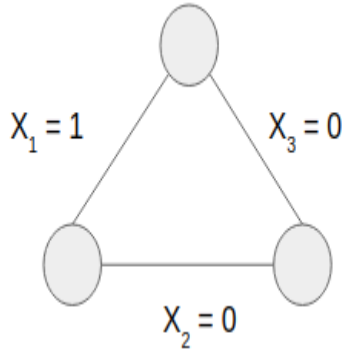
1.2 LP Relaxation

We can relax the conditions of the above integer program to obtain a linear program as follows. We convert the condition : $X_e \in \{0, 1\}$ to $0 \leq X_e \leq 1$, and hence obtain the linear program :

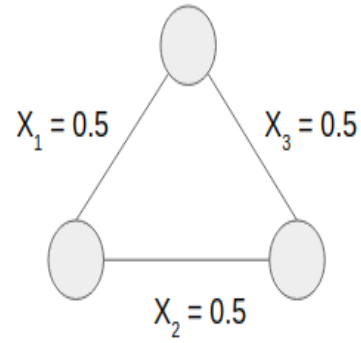
$$\begin{aligned} \max \quad & \sum_{e \in E} X_e \\ \forall e \in E \quad & 0 \leq X_e \leq 1 \\ \forall v \in V \quad & \sum_{e \in \delta(v)} X_e \leq 1. \end{aligned}$$

But, it is not necessary that the solution obtained by the LP is same as the solution to the maximum matching problem i.e., the integer program defined above. Consider for example a complete graph with 3 vertices:

The maximum matching for this graph will be of size 1 and hence the integer program will give a solution shown in the Figure 1a, where the optimal value is '1'. But, When we solve the relaxed linear program, we do not get this solution. Rather, we get a solution as shown in Figure 1b, where the optimal value is ' $\frac{3}{2}$ '. Here, each edge has a fractional value in Figure 1b, which does not really give us a matching.



(a) Maximum matching integer program solution



(b) Solution from the relaxed linear program

Figure 1: Difference between integer program and the relaxed LP

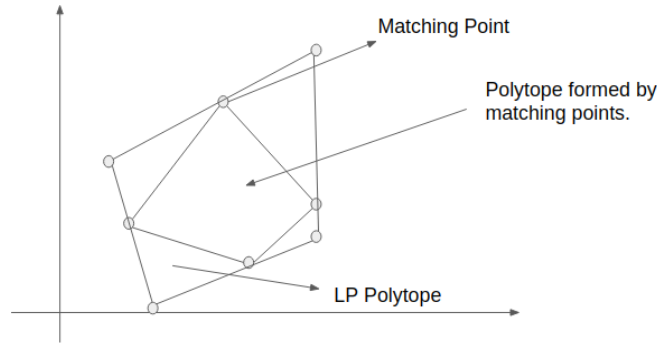


Figure 2: Convex hull of matching points

1.3 LP Polytope

It is to be noted that the convex hull of the matching points gives us a polytope – called matching polytope (see figure 2). Note that as the matching points are feasible for the relaxed linear program, those points are contained inside the LP polytope. And hence, the matching polytope is contained inside the LP polytope.

Now, as we already noted, integer programming is NP-hard and so we cannot solve it. The LP relaxation does not guarantee us an integer solution always, hence we cannot necessarily get a matching out of it. But there are some special cases (special kinds of graphs) where the linear program is guaranteed to give an integer solution (or the fractional solution can be converted to an integer solution without reducing the value of the objective function). One such case is that of **bipartite graphs**.

2 Maximum Matching as LP for Bipartite Graphs

The relaxed linear program that we obtained was :

$$\begin{aligned} \max \quad & \sum_{e \in E} X_e \\ \forall e \in E \quad & 0 \leq X_e \leq 1 \\ \forall v \in V \quad & \sum_{e \in \delta(v)} X_e \leq 1. \end{aligned}$$

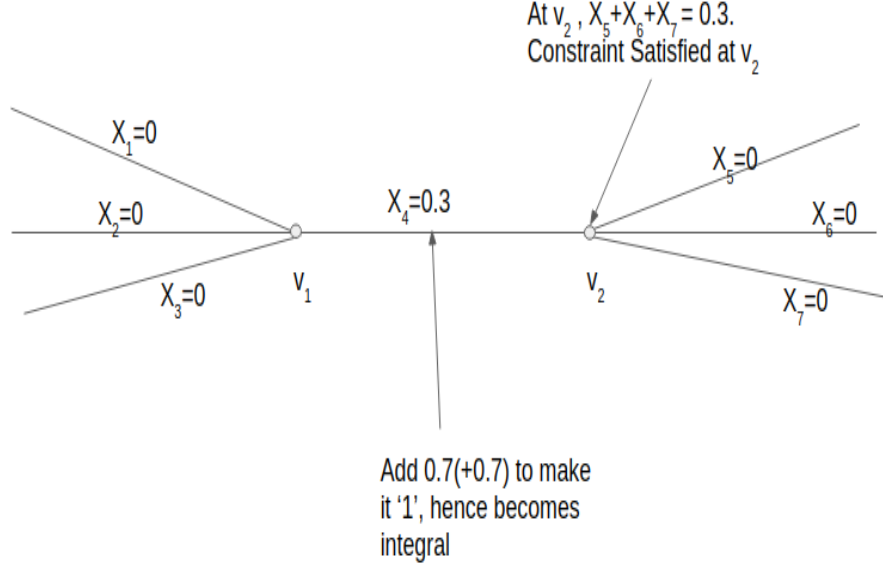


Figure 3: Fractional to Integral value Case-1

Claim 10.2. *The relaxed LP is correct (i.e., gives an integer solution or the solution can be converted to an integral solution) for bipartite graphs.*

Claim 10.3. *The relaxed LP is correct when there are no odd length cycles in the Graph.*

The claims 10.2 and 10.3 are equivalent. This is obvious because we know that a graph is bipartite, if and only if it has no odd length cycles. So, if the LP is correct for either bipartite graphs or for graphs with odd length cycles, it must be correct for the other also.

Proposition 10.4. *Given any **fractional feasible solution** for the relaxed linear program for the maximum matching in a bipartite graph, we can **convert it into an integral feasible solution** for which the value of the objective function is greater than or equal to the value of the objective function for the fractional solution, i.e., **integral optimal** = **fractional optimal**.*

Proof. We will describe a procedure for converting any fractional solution to an integral solution, while maintaining feasibility. For a given solution, let us call an edge fractional, if the corresponding edge variable takes a fractional value ($0 < X_e < 1$) in the solution. We assume that there is at least one fractional edge. We can divide our procedure in two cases:-

1. When there are no neighbouring fractional edges for a fractional edge:

Let us say we have an edge $e \in E$, which has some non-zero fractional value α . Note that, if none of its neighbouring edge variables has a fractional value, then all its neighbours must have value **0**. Because, if any of them is **1**, then it will violate the constraint " $\sum_{e \in \delta(v)} X_e \leq 1$ " for one of end points v of the edge e .

In this case, we can just increment the value of this edge variable by $1 - \alpha$, so that the edge value becomes equal to 1. And, since all the neighbouring edges are 0 anyway, no constraints are violated. Moreover, we will have our objective function value increased by $1 - \alpha$. Refer to Figure 3 for an example.

2. When there are neighbouring fractional edges for every fractional edge. Start with an arbitrary fractional edge and start walking along neighboring fractional edges. There are two sub-cases: (a) when the walk results in a cycle and (b) when the walk results in a path.

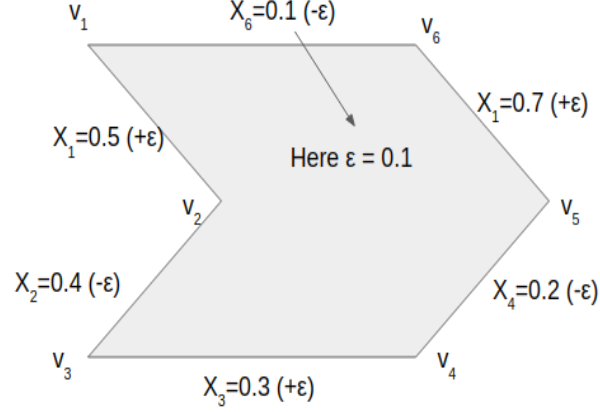


Figure 4: Fractional to Integral value Case-2(a)

(a) Fractional Neighbours forming a cycle.

We will consider even length cycles here, since if there are cycles in a bipartite graph they must be of even length. Moreover, during the procedure, we will understand why it will not work for odd length cycles.

Consider an even length cycle of vertices $V' = \{v_1, v_2, \dots, v_k, v_1\}$ with all edge variables having fractional values. Define ϵ to be the minimum amongst all values in the cycle. Without loss of generality, let's say the edge (v_1, v_2) has value $X_{(v_1, v_2)} = \epsilon$. Subtract the value ϵ from $X_{(v_i, v_{i+1})}$ add ϵ to $X_{(v_{i+1}, v_{i+2})}$ for all odd i between 1 and k (consider $v_{k+1} = v_1$). This keeps all the constraints corresponding to the vertices v_i , for $1 \leq i \leq k$, satisfied. See figure 4. After each such iteration, at least one variable in the cycle is guaranteed to become integral. Continue this procedure, until all edge variables become integral or no more cycles exist in the graph. This procedure preserves correctness since, all edge variables X_e are still between 0 and 1, and the constraint $\sum_{e \in \delta(v)} X_e \leq 1$ corresponding to each vertex $v \in V$ is also satisfied after the completion of this procedure.

Since at least one edge variable in the cycle becomes integral on every iteration, it will be a finite process. The objective function value will not decrease after the procedure (it will remain the same, since we added/subtracted ϵ an equal number of times to/from edge variables).

Note that **this procedure would not work for odd length cycles**, because we cannot add/subtract ϵ same number of times to/from the edge variables in this case. See figure 5. Here, before completing the procedure, it is not sure what should be done for X_5 . If we add ϵ to X_5 , the constraint at v_1 gets violated as $X_5 + X_1$ will become $(0.3 + (\epsilon = 0.2)) + (0.7 + (\epsilon = 0.2)) = 1.4$ (considering no more edges are incident at v_1). If we subtract, the objective function value will have decreased since we basically did $(\text{objective value} + (-3\epsilon) + (2\epsilon)) = (\text{objective value} - \epsilon)$. So, that is not suitable as well. Hence, this procedure does not work for odd length cycles.

(b) Fractional neighbours forming a path.

Consider such a path with vertex sequence (v_1, v_2, \dots, v_k) . The edges (v_1, v_2) and (v_{k-1}, v_k) are the terminal edges in the sense that the only fractional edge incident on v_1 (v_k) is the one connecting it to v_2 (v_{k-1}). Note that any integral edges adjacent to the path must have value 0. We modify the solution as follows: for each odd i between 1 and k , increase the value $X_{(v_i, v_{i+1})}$ by $\epsilon > 0$ and decrease the value $X_{(v_{i+1}, v_{i+2})}$ by ϵ . We choose ϵ to be the minimum value $\min\{X_{(v_{i+1}, v_{i+2})} : 1 \leq i \leq k \text{ is odd}\}$. This ensures that none of the values becomes negative when we subtract ϵ . Moreover, due to alternate $+/-\epsilon$ on the path, all the vertex constraints remain valid. Now, we need to see the change in the objective value. One can observe that the objective value increases if k is even and remains same otherwise.

See figure 6 for example. We want to make the value of X_5 integral. Here, we can do 2 things:

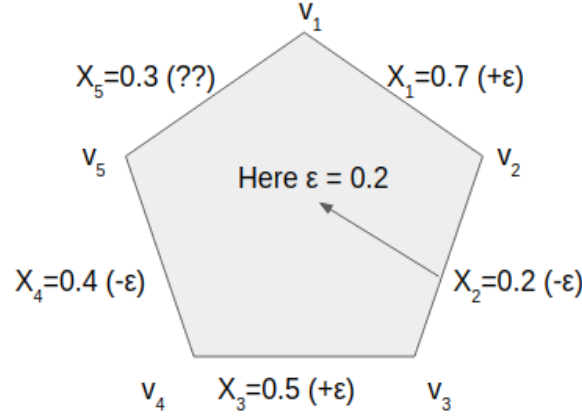


Figure 5: Problem with odd length cycles in Case 2(a)

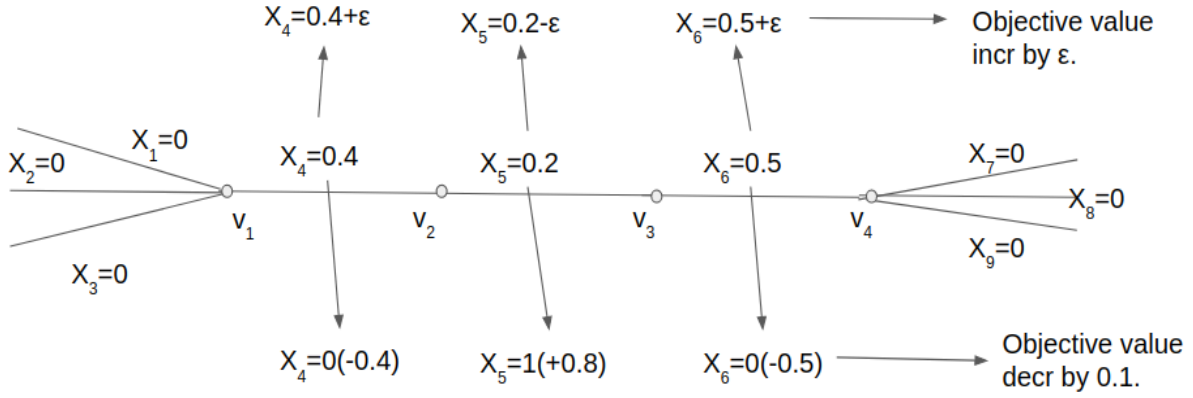


Figure 6: Fractional to integral values Case-2(b)

- set $X_5 = 1$ and to preserve the constraints at v_2 and v_3 we must then set $X_4 = 0$ and $X_6 = 0$. But this reduces the value of the objective function by 0.1 and hence, should not be done.
- Set $\epsilon = X_5$ (basically the value of the edge variable you want to make integral). Then set $X_5 = 0$ and add the value of ϵ from its adjacent edges namely X_4 and X_6 . This will preserve the constraints at v_2 and v_3 . Moreover, it increases the value of the objective function by a value of ϵ (here 0.2).

We repeat this procedure till no cases like 2(a) or 2(b) is left i.e., all cases are reduced to Case 1.

□

Proof of Claim 10.2 and Claim 10.3. From the above procedure, it is clear that solving the LP either gives an all integral solution (all edge variable values are integers) or it gives a solution which can be converted to an all integral solution with the guarantee that it is also optimum (objective function value for the all integral solution is never less than original solution to the LP). □

3 Dual Formulation of the Relaxed Linear Program

From the relaxed LP in Section 1.2, we can remove the $X_e \leq 1$ constraint since it is redundant. This condition is already taken care of by the second constraint i.e $\forall v \in V \sum_{e \in \delta(v)} X_e \leq 1$. We will formulate the dual LP using this new LP as the Primal LP.

The refined relaxed LP is as follows :-

$$\begin{aligned} \max \quad & \sum_{e \in E} X_e \\ \forall v \in V \quad & \sum_{e \in \delta(v)} X_e \leq 1 \\ \forall e \in E \quad & X_e \geq 0 \end{aligned}$$

The dual can be formulated as follows:

$$\begin{aligned} \min \quad & \sum_{v \in V} y_v \\ \forall e = (u, v) \in E, \quad & (y_u + y_v) \geq 1 \\ & y_v \geq 0 \end{aligned}$$

According to the rules of formulating the dual here the dual LP should have one variable corresponding to each constraint of the Primal LP (one for each vertex precisely named $y_v, \forall v \in V$), and one constraint corresponding to each variable of the Primal LP (one for each edge). And, the objective function becomes a minimizing function.

3.1 Correspondence of the Dual LP with the Vertex Cover Problem

Definition 10.5 (Vertex Cover Problem). *Given a graph $G(V, E)$, a **vertex cover** is any set of vertices which covers (includes) all the edges in the set E . More formally a set $C \subseteq V$ is a vertex cover if for each edge $(u, v) \in E$, $u \in C$ or $v \in C$.*

*The **Minimum vertex cover** problem aims at finding the **Vertex Cover with the minimum number of vertices**.*

Now, if we look at the dual LP carefully, we see that any integer solution to this LP corresponds to the **Vertex Cover Problem**. Here is how:-

Let us consider that y_i can take only integer values and we have the graph $G(V, E)$. This LP gives a value to each vertex (y_i), such that the value is non-negative and the second constraint ($\forall e(u, v) \in E, (y_u + y_v) \geq 1$) makes sure that for each edge at least one of the vertices has the value 1. Any feasible solution to this LP will hence cover all the edges.

But since the objective is to minimize the sum over all $(y_i)_{i \in V}$, the integer solution will be such that minimum number of y_i 's are set to 1 satisfying all the constraints. The set C of vertices x for which $y_x = 1$ in an optimal integer solution, is therefore precisely a minimum vertex cover of the graph $G(V, E)$.

But, one point that is to be noted is that the solution to the dual LP does not necessarily give a vertex cover. See figure 7, a complete graph with 3 vertices. The minimum vertex cover for this graph is of size two ($\{v_1, v_2\}$), whereas solving the LP gives the objective function value as $\frac{3}{2}$ with $y_{v_1} = y_{v_2} = y_{v_3} = 0.5$. Hence,

Optimal(dual LP) \leq Minimum vertex cover for general graphs.

3.2 Proving König's Theorem using LP Duality

We have the following results till now:-

$$\text{size(Maximum Matching)} \leq \text{Optimal(Primal LP)} \quad \{\text{from Section 1}\} \tag{1}$$

$$\text{Optimal(Dual LP)} \leq \text{size(Vertex Cover Problem)} \quad \{\text{from Section 3.1}\} \tag{2}$$

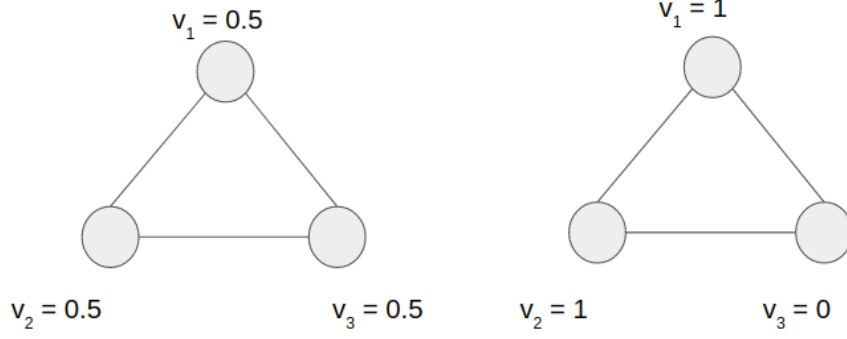


Figure 7: Solution for Vertex cover LP and the Actual Vertex Cover

$$\text{Optimal(Primal LP)} \leq \text{Optimal(Dual LP)} \quad \{\text{from Weak Duality}\} \quad (3)$$

Theorem 10.6. *size(Maximum Matching) \leq size(Minimum Vertex Cover)*

Proof. Combine the equations 1, 2, 3. □

Theorem 10.7 (König's Theorem). *In a bipartite graph, the size of the maximum matching is equal to the size of the minimum vertex cover.*

Proof. From section 2, we have that for a Bipartite graph,

$$\text{size(Maximum Matching)} = \text{Optimal(Primal LP)}$$

Again, from the strong Duality theorem we have,

$$\text{Optimal(Primal LP)} = \text{Optimal(Dual LP)}$$

We want to show :

$$\text{size(Maximum Matching)} = \text{size(Minimum Vertex Cover)}$$

for bipartite graphs.

Hence, we need to prove that :

$$\text{Optimal(Dual LP)} = \text{size(Minimum Vertex Cover)}$$

for bipartite graphs.

Claim 10.8. *Any optimal solution of the vertex cover LP (dual LP) is integral for bipartite Graphs. Hence, for bipartite graphs $\text{Optimal(Dual LP)} = \text{size(Minimum Vertex Cover)}$.*

Proof. Let U_1, U_2 be the two parts of the bipartition of the vertices. Let $y^* = \{y_1, y_2, \dots, y_k\}$, $k = |U_1 \cup U_2|$ be an optimal solution to the vertex cover LP chosen such that y^* has the maximum number of integer components. It does not make much sense that y^* would have any component > 1 (the objective function will not be minimum). So assume $\forall i \in \{1, 2, \dots, k\}$, $y_i \leq 1$. Let F be the set of fractional vertices $F = \{v \in U_1 \cup U_2 \mid 0 < y_v < 1\}$. If $F = \emptyset$, we are done. Otherwise without loss of generality, we assume that $|F \cap U_1| \geq |F \cap U_2|$. Let $\epsilon = \min\{y_i \mid i \in F \cap U_1\}$. Then subtract ϵ from all vertices in $F \cap U_1$ and add ϵ to all vertices of $F \cap U_2$.

Now, take a look at figure 8. Let $F' = (U_1 \cup U_2) \setminus F$. We can divide the constraint into 4 types :

1. Constraints for edges $(u, v) \in E$ s.t $u \in F \cap U_1$ and $v \in F \cap U_2$.

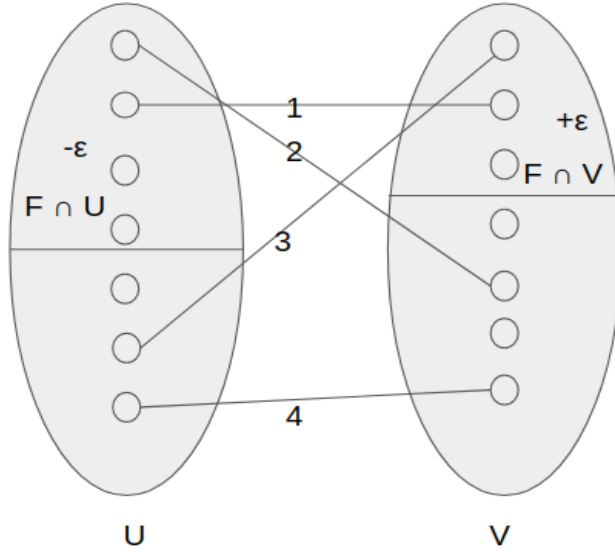


Figure 8: König's Theorem Proof

2. Constraints for edges $(u, v) \in E$ s.t $u \in F \cap U_1$ and $v \in F' \cap U_2$.
3. Constraints for edges $(u, v) \in E$ s.t $u \in F' \cap U_1$ and $v \in F \cap U_2$.
4. Constraints for edges $(u, v) \in E$ s.t $u \in F' \cap U_1$ and $v \in F' \cap U_2$.

After adding and subtracting ϵ to/from $F \cap U_2$ and $F \cap U_1$, constraints of the type 4 are not affected, since these are the vertices to/from which nothing was added/subtracted. Constraints of 1 are also not affected since these constraints correspond to edges $(u, v), u \in F \cap U_1, v \in F \cap U_2$ such that we assigned $y_u \leftarrow y_u - \epsilon$ and $y_v \leftarrow y_v + \epsilon$. Hence, the value of $y_u + y_v$ remains intact, and the constraint which was satisfied earlier is still satisfied.

Consider constraint of type 3. Here each constraint corresponds to an edge (u, v) s.t. $u \in U_1 \cap F', v \in F \cap U_2$. here y_u remains intact, but we assign $y_v = y_v + \epsilon$. Hence, the value of $y_u + y_v$ increases. $y_u + y_v$ will still remain greater than 1 and hence the constraint will be satisfied.

Finally, we consider the constraint of type 2, where each constraint corresponds to edge $(u, v), u \in F \cap U_1, v \in U_2 \cap F'$. Since, the vertex in U_1 was fractional (as $u \in F \cap U_1$) and the vertex in U_2 was not ($v \in F' \cap U_2$), then y_v must have been equal to 1 (otherwise, $y_u + y_v$ would have been less than 1, not satisfying the constraint). But now since we know $y_v = 1$ and $\epsilon \leq y_u$, even after setting $y_u = y_u - \epsilon$, The value of $y_u + y_v$ remains greater than or equal to 1. Hence, still satisfying the constraint.

At each iteration, at least one y_i is guaranteed to become integral. Moreover, the objective value $\sum_v y_v$ decreases or remains same in the process because $|F \cap U_1| \geq |F \cap U_2|$. This means that there is always an optimal solution to the vertex cover LP that is integral. And as we have seen that an all integral solution to this LP gives a minimum vertex cover (in Section 3.1), therefore :

$$\text{Optimal(Dual LP)} = \text{size(Minimum Vertex Cover)}.$$

Hence, combining all the previous results, we have :

$$\text{size(Maximum Matching)} = \text{size(Minimum Vertex Cover)}.$$

Hence, we have been able to prove König's theorem using LP duality.

□

□