第6章 图与网络分析

6.7 最大匹配问题

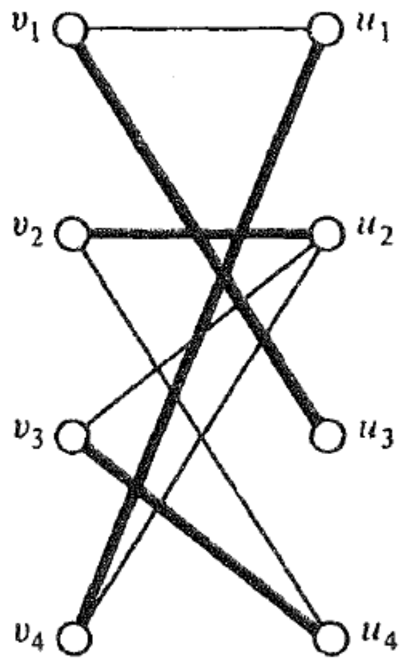
基本概念

- **图 $G=(V,E)$的对集 M**：M 是 E 的子集，且 M 中任意两条边均不相邻（都不共享顶点）。
- **M-饱和点 i**：$V(M)$中的顶点（$i \in V(M)$，匹配的点）。
- **M-非饱和点 i**：$V(M)$之外的顶点（$i \notin V(M)$，没有匹配的点）。
- **极大对集 M**：不存在另外一个对集M'，使得$M \subset M'$。
- **最大对集 M**：不存在另外一个对集M'，使得$|M'| > |M|$。
- **完美对集 M**：对集 M，匹配了 G 上所有的点。

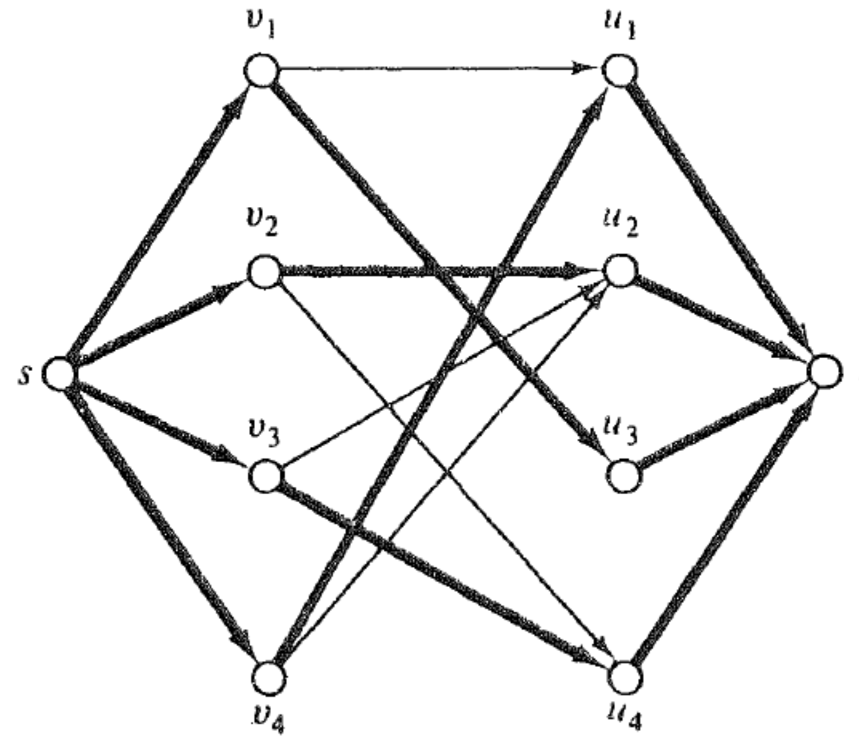
使用最大流算法求二分图上的最大匹配

- 给定二分图$G = (V, U, E)$，构造流网络。
- 增加一个源点 s，从 s 到 V 中每个顶点引一条有向边。
- 增加一个目标顶点 t，从 U 中每个顶点向 t 引一条有向边。
- E中的边均从 V 指向 U。
- 记得到的流网络为$G' = (V', E')$。G'中的每条边均为单位容量。
- 计算G'上从 s 到 t 的最大流。
- E 中的饱和边即构成 G 上的一个最大匹配。

例子



(a)



(b)

定理

- 定理：记G'上的最大流为f^*，流值为$|f^*|$。G上的最大匹配为M^*。则$|f^*| = |M^*|$。
- 证明：首先证$|f^*| \geq |M^*|$。
- 给定最大匹配M^*，令G'上M^*中的边的流值为1，s到M^*匹配的V一侧点的各条边上流值为1，M^*匹配的U一侧点到t的各条边上流值为1，则构造了一个流值为$|M^*|$的流f。
- 因此，显然有$|f^*| \geq |M^*|$。
- 再证$|f^*| \leq |M^*|$。
- 设f^*为G'上的最大流。
- 由整流定理，G'上每条边上的流值为整数。由于每条边的容量均为1，因此G'上每条边的流值不是0就是1。

证明

- 再由流守恒约束，V中每个顶点最多有一条出去的边流值为1。同理，U中每个顶点最多有一条进来的边流值为1。
- 记$M = \{e \in E \mid e\text{上的流值} > 0\}$，因此$M$中的任何两条边均不共享顶点，即，$M$是一个匹配，且$|f^*| = |M|$。
- 因此，显然有$|f^*| \le |M|$。□

基本概念

- M -交错路：边在对集 M 和 $E \setminus M$ 中交错出现的路。
- M -增广路：起点和终点都不在 $V(M)$中的 M -交错路。

定理 6.8.1（Berge, 1957）图 G 中的一个匹配 M 是最大匹配当且仅当 G 不包含 M-增广路。

通过增广路求二分图上的最大匹配

- 从图 $G = (S, T, E)$的任意一个匹配 M 开始，比如空集。
- 由 S 的一个未被匹配的顶点出发，用一个系统方法搜索一条 M-增广路 P。
- 若 P 存在，则通过交换 P 在 M 和不在 M 中的边，便得到一个其基数增加 1 的匹配。
- 然后从新的匹配开始，继续迭代，直到不存在 M-增广路，则当前的匹配就是 G 的最大匹配。

二分图上最大匹配的标号算法

输入： 二分图 $G = (S, T, E)$。

输出： G 的最大匹配 M。

1 $M \leftarrow \varnothing$。

2 对 S 中所有不在 M 中的顶点标号"$\varnothing$"，然后将这些顶点都加入 Q。/* Q 是已标号但未检查的顶点的集合 */

3 while $Q \neq \varnothing$ do

4 　　从 Q 中取出一个顶点 k，然后将 k 从 Q 中删除。

5 　　if $k \in S$ then

6 　　　　对每条同 k 关联的边$(k, j) \notin M$，若 j 尚未被标号，则给点 j 标号 k，并将 j 加入 Q。

7 　　else /* $k \in T$ */

二分图上最大匹配的标号算法

if k 在 M 中 then /* 此时有 $k \in T \cap V(M)$ */

  设(i, k)是关联于i、属于M的边（此时$i \in S$）。

  给点 i 标号 k，并将 i 加入 Q。

else /* 此时有 $k \in T \setminus V(M)$ */

  终止在 k 的一条增广路被找到。从 k 开始，

  反向追踪标号找到这条增广路 P，路的起

  始顶点有标号“$\varnothing$”。

  用增广路 P 更新 M。

  删除 G 上所有的标号。重新对 S 中所有不

  在 M 中的顶点标号“$\varnothing$”，然后将这些顶

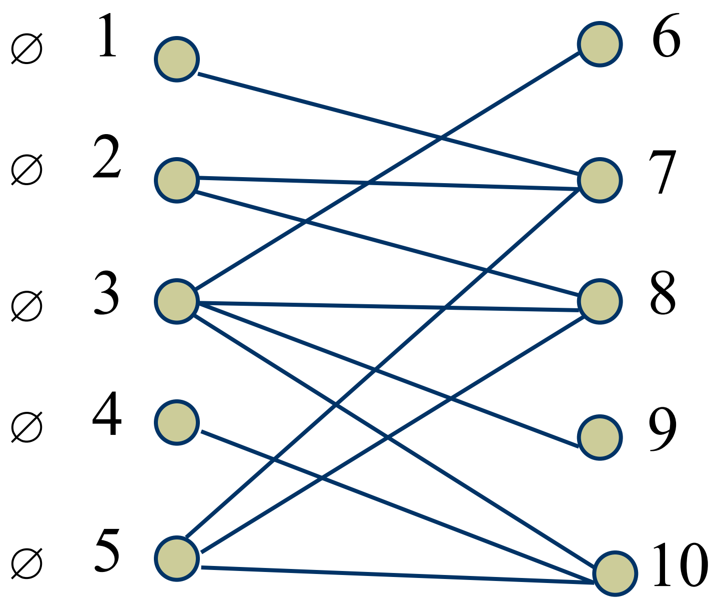
  点都加入 Q。

endif

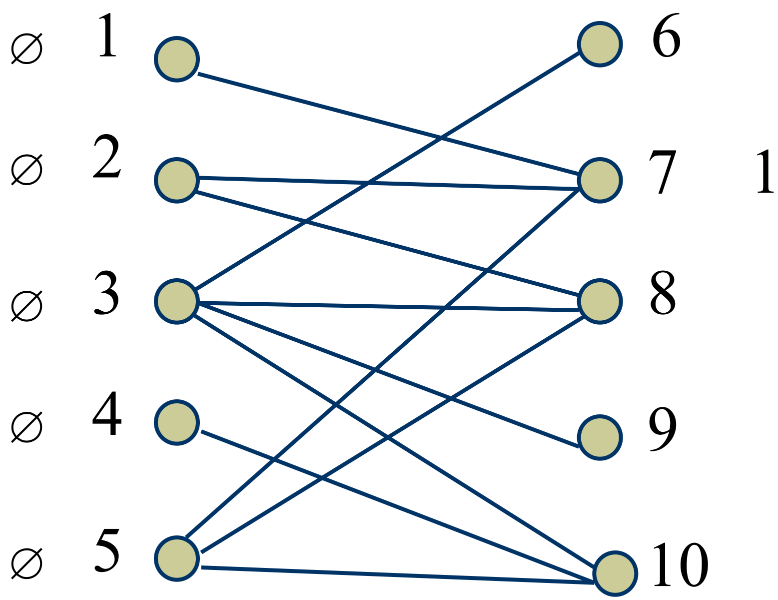
二分图上最大匹配的标号算法

```
        endif
    endwhile
    return M。
```

例子

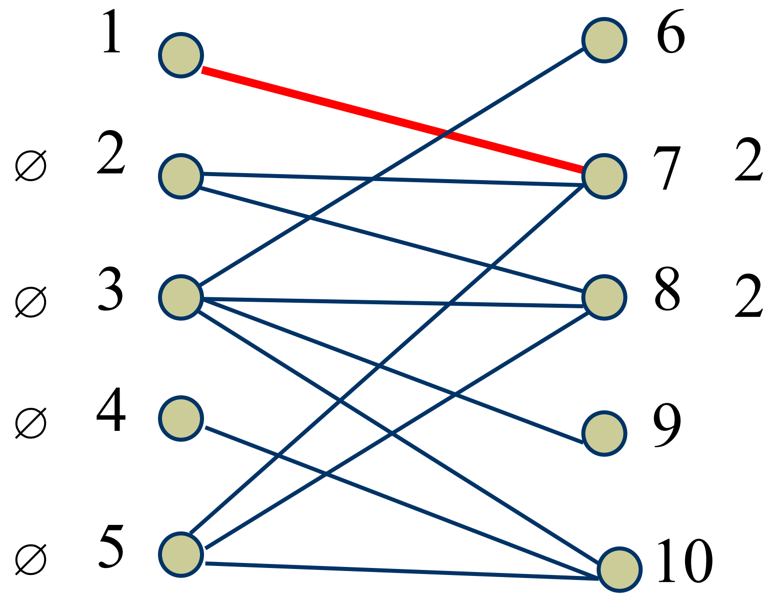


例子



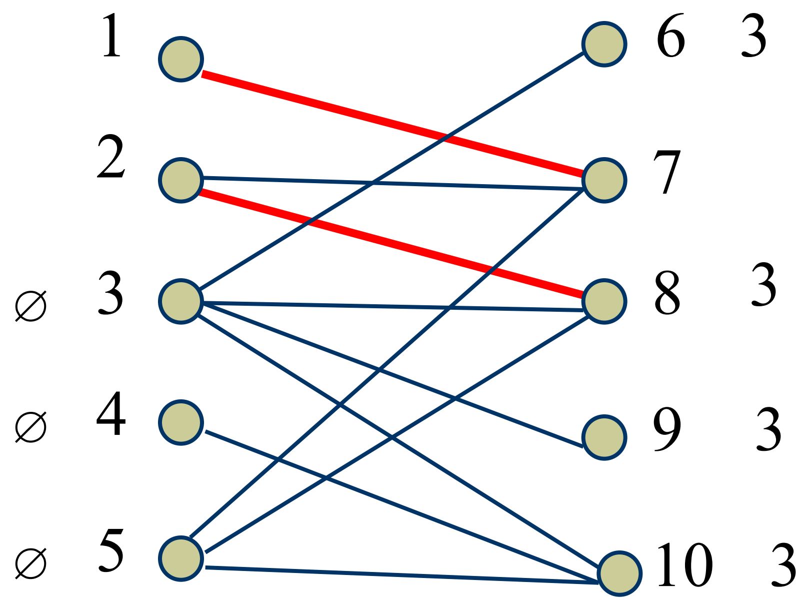
找到一条增广路(1, 7)。更新*M*。

例子



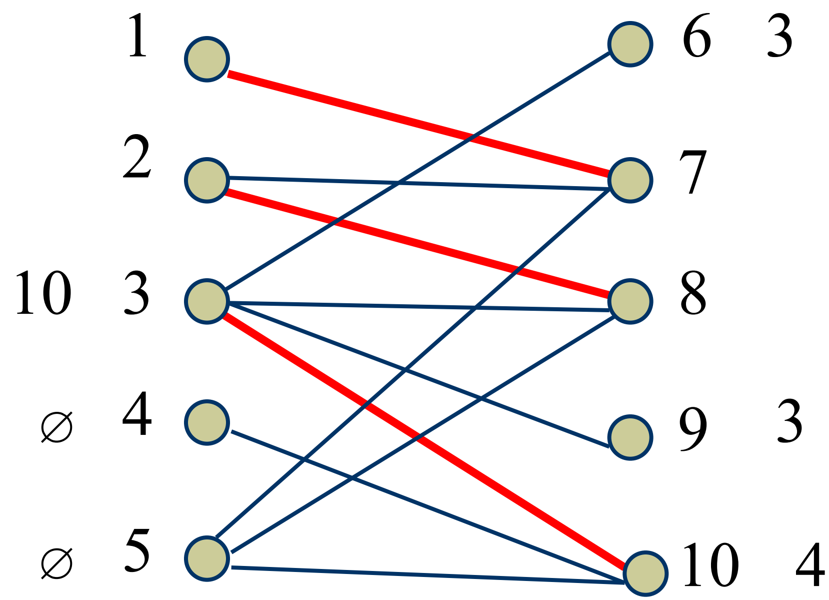
找到一条增广路(2, 8)。更新*M*。

例子



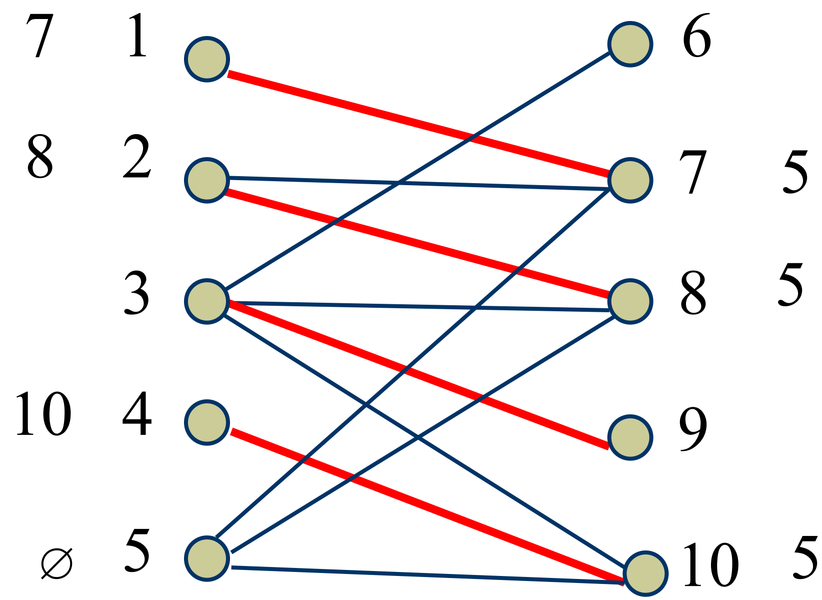
找到一条增广路(3, 10)。更新*M*。

例子



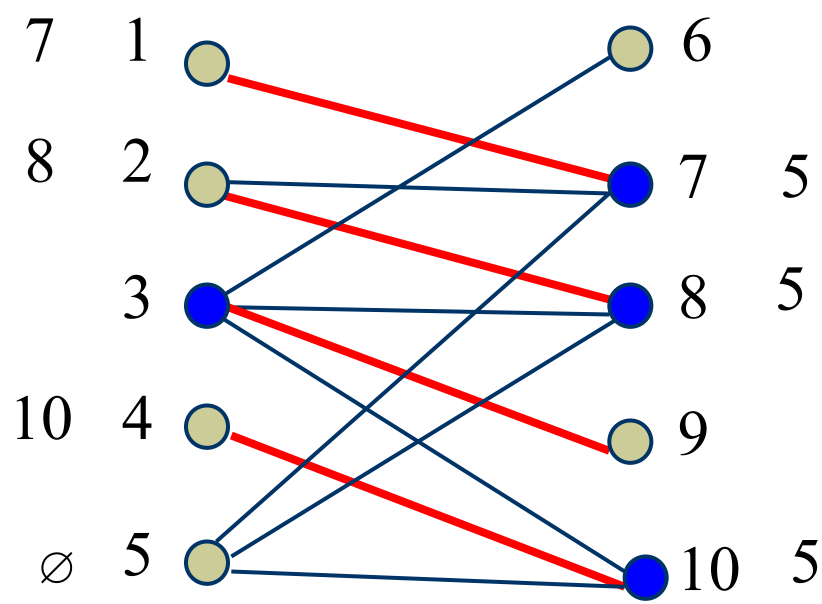
找到一条增广路(4, 10, 3, 9)。更新*M*。

例子



找不到增广路，结束。

例子



{红边}为最大匹配，{蓝色顶点}为顶点覆盖。

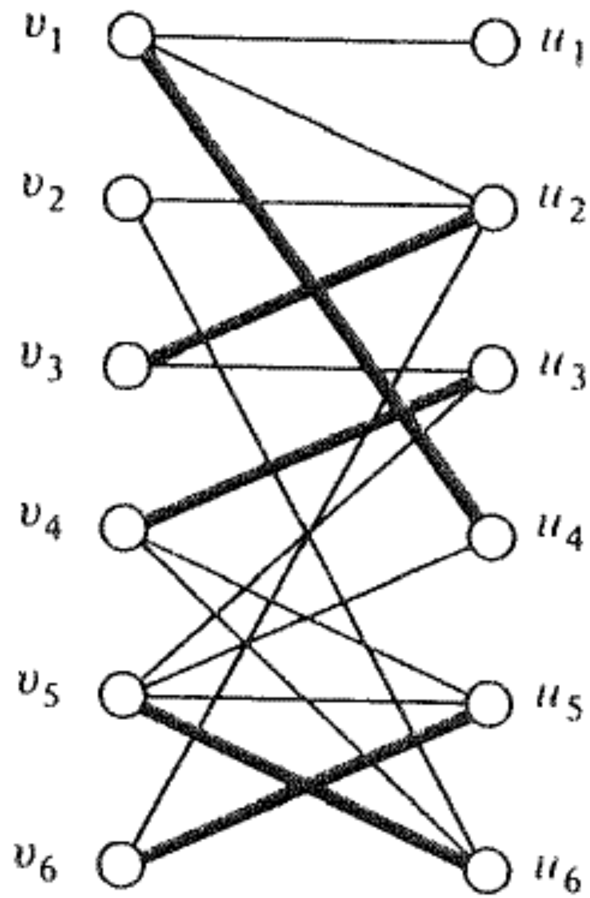
时间复杂度分析

- 令$|S| = m$，$|T| = n$，假设 $m \le n$。
- 找一条增广路（或判断不能找到）标号算法最多进行 O(mn) 次检查（因为最多有这么多条边）。
- 初始匹配最多被增广 m 次。
- 所以，总的计算量为 O(m^2n)。

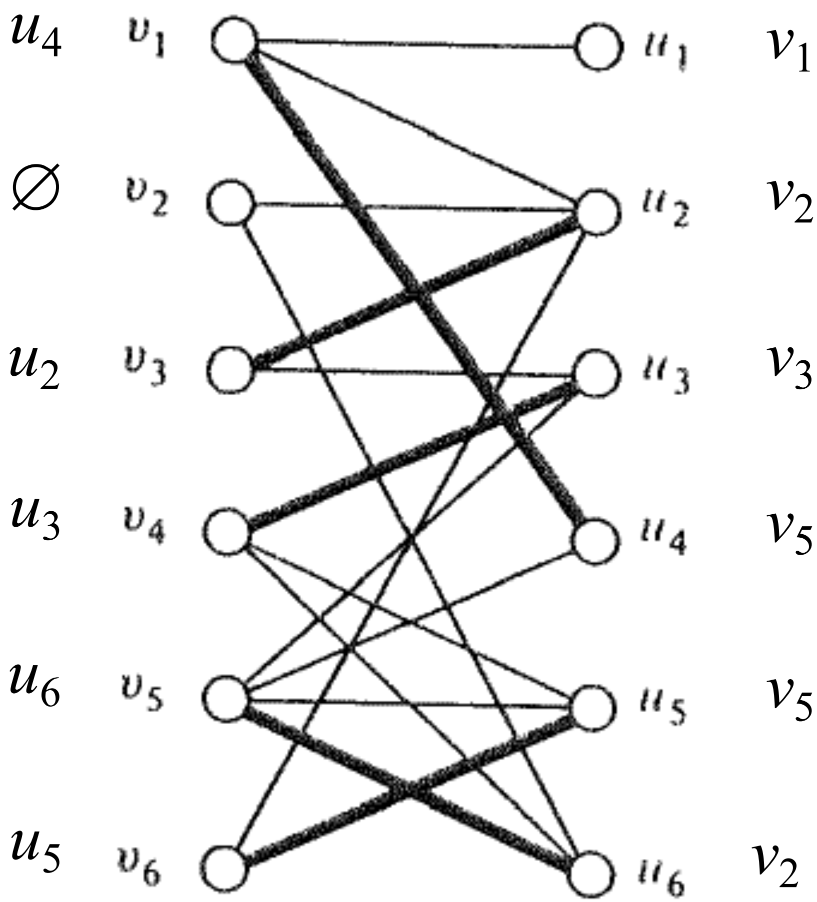
解释

- 从S中未匹配的顶点开始，标号找M-增广路的过程，实际上是一个从S中未匹配的顶点开始进行类广度优先搜索的过程。
- 该过程与标准的广度优先搜索不完全相同。
- 设搜索树的根位于第1层。区别仅在于，在搜索过程中，奇数层顶点（在S一侧）按广度优先展开；偶数层顶点（在T一侧）按M中的（唯一一条）边顺延（而不是按广度优先展开）。

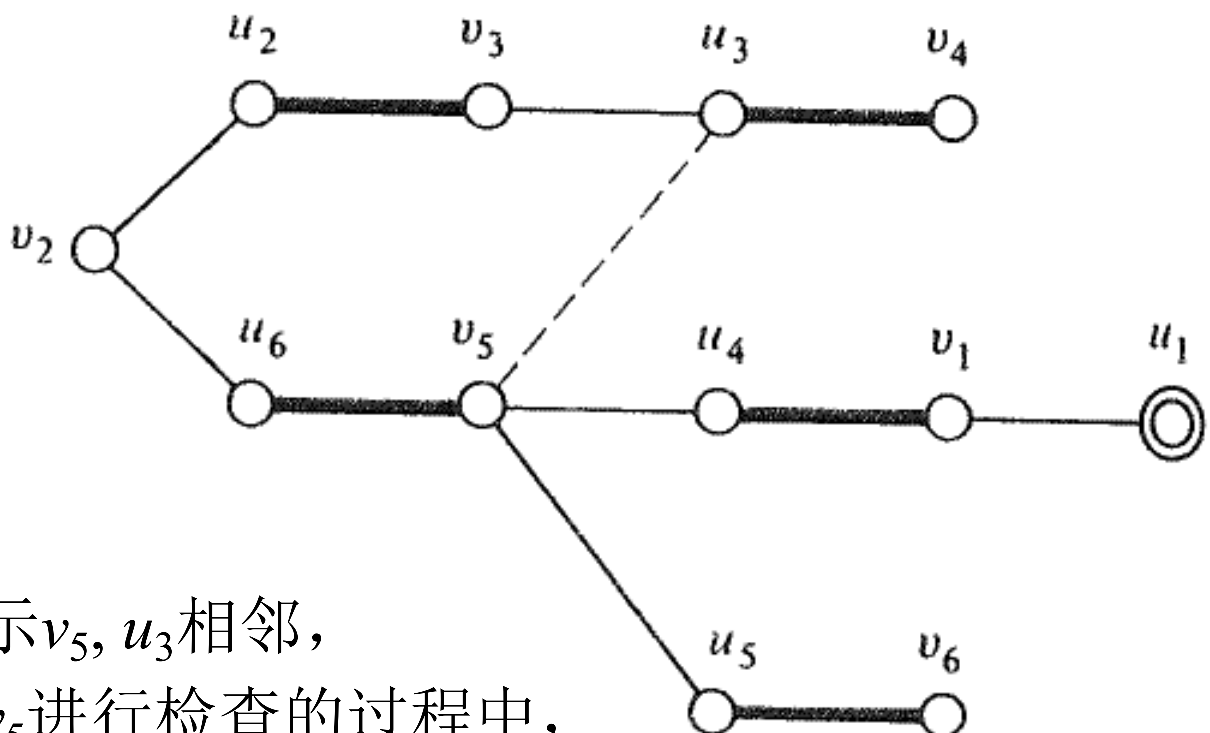
解释



标号，找增广路

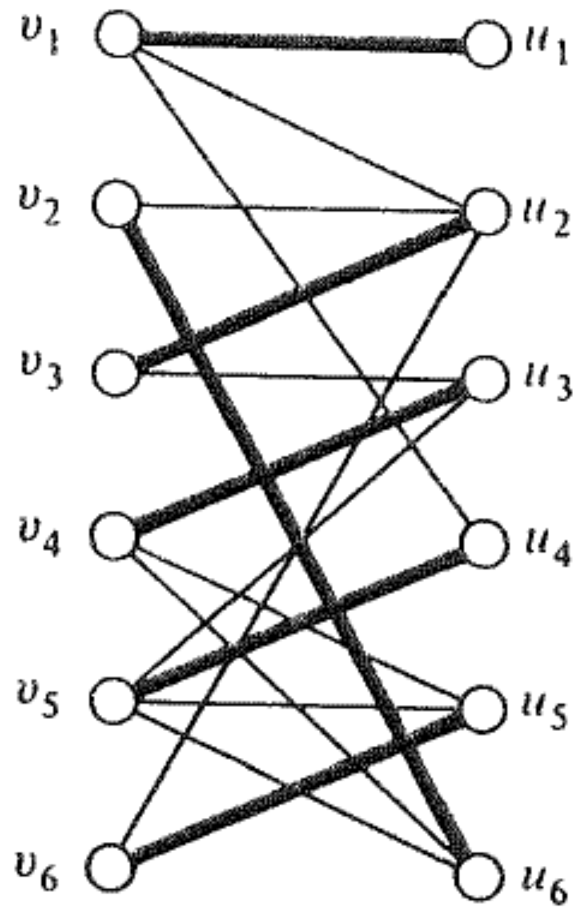

u_4
$\varnothing$
u_2
u_3
u_6
u_5
v_1
v_2
v_3
v_4
v_5
v_6
u_1
u_2
u_3
u_4
u_5
u_6
v_1
v_2
v_3
v_5
v_5
v_2

找增广路过程中形成的搜索树



虚线表示v_5, u_3相邻，
但在对v_5进行检查的过程中，
u_3已经标号，因此从v_5不能
对u_3标号。

增广，得到一个更大的匹配



顶点覆盖

- **顶点覆盖 K**：K 是 $V(G)$的子集，且 G 的每条边都至少有一个端点在 K 中。
- **最小顶点覆盖 K**：不存在另外一个覆盖 K'，使得$|K'|<|K|$。

定理 6.8.3（König，1931） 在二分图中，最大基数对集的边数等于最小顶点覆盖的点数。

谢谢大家