

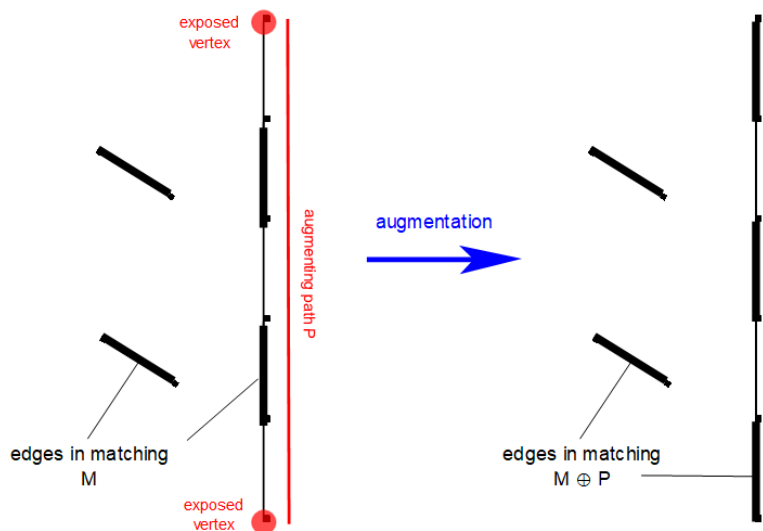
带花树

带花树是解决一般图的最大匹配的算法。

增广路

由于涉及到最大匹配，我们这里对增广路进行一个简短的介绍：

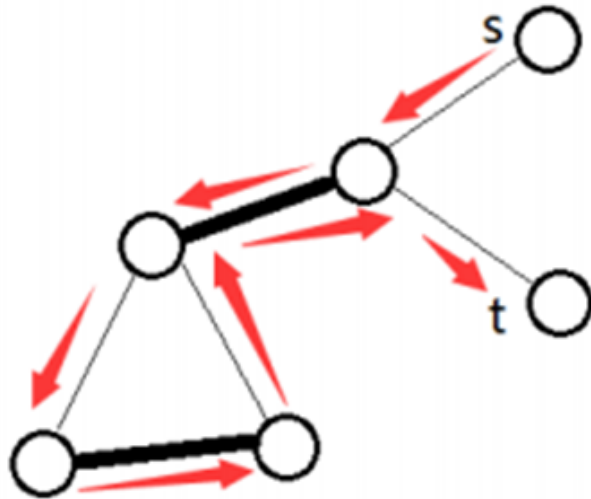
1. 如果在一个 M 匹配的图 G 中，有一个点 v 是孤立的指没有与其匹配的点。一条在 G 中的路径如果其边在 M 中交替出现，则称为交错路 \square alternating path \square 。一条增广路 \square augmenting path \square P 是一条开始并结束于不相同的孤立点的交错路。
2. 在增广路中不是匹配的边比是匹配的边多，因此增广路的边条数为奇数。
3. 一个匹配在增广路上的增广即为异或操作 $\square M1 = M \text{ xor } P \square$



一般图中增广路与二分图的区别

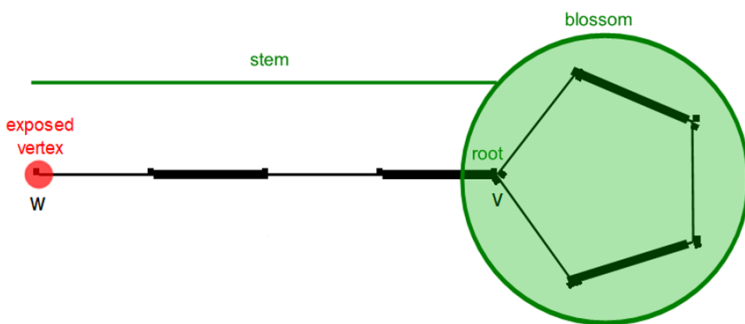
考虑下我们之前在二分图中找增广的过程：

1. 我们寻找增广路的时候，会将路径上的点黑白染色，匹配只存在于黑点白点之间。
2. 如果没有环或是只有偶环（二分图中只有偶环），那么每个点的颜色是确定的。
3. 但是如果出现了奇环，那么点的颜色不再确定，因为奇环顺时针走一圈和逆时针走一圈的染色结果是不同的。（如下图所示，模拟一次增广的染色，我们可以明显发现进入奇环的那个点的染色结果出现冲突）



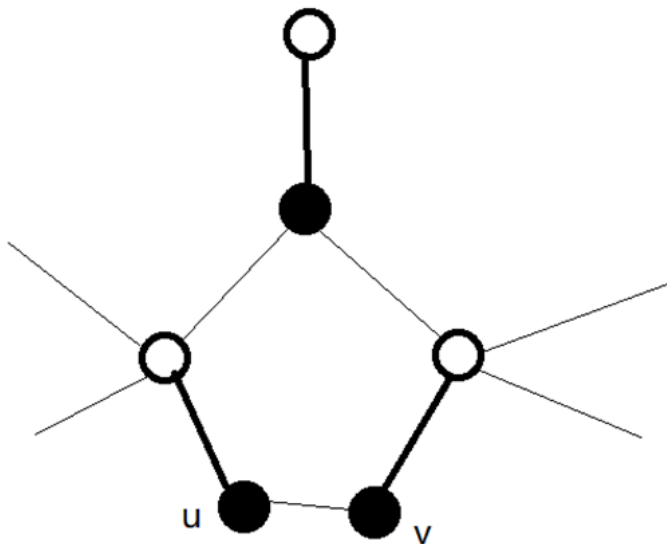
花与收缩的定义

在一个 M 匹配的图 G 中，一朵花 B 是一个图 G 中的包含 $2k+1$ 条边的奇环，其中 k 条边在 M 中，并存在从环上任意一个点 v (花根) 到一个孤立点 w 的交错路 (花茎)。



如何找到一朵花

1. 从一个孤立的点 w 开始遍历图。
2. 从孤立点 w 开始遍历，标记 w 为 S 型点 [out of M]
3. 交替地用 S 和 O 标记结点，保证无两个相邻节点有标记。
4. 如果找到两个相邻结点含有标记 O 那么我们就找到了一个奇环和一朵花。

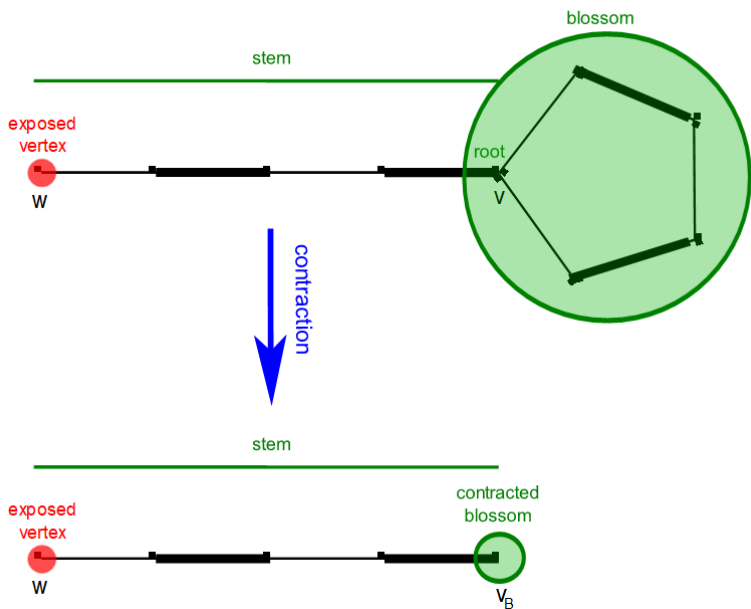


收缩的定义

定义收缩图 G' 是图 G 将所有花 B 缩为点后的图。

定义收缩匹配 M' 是在 G' 中的匹配 M

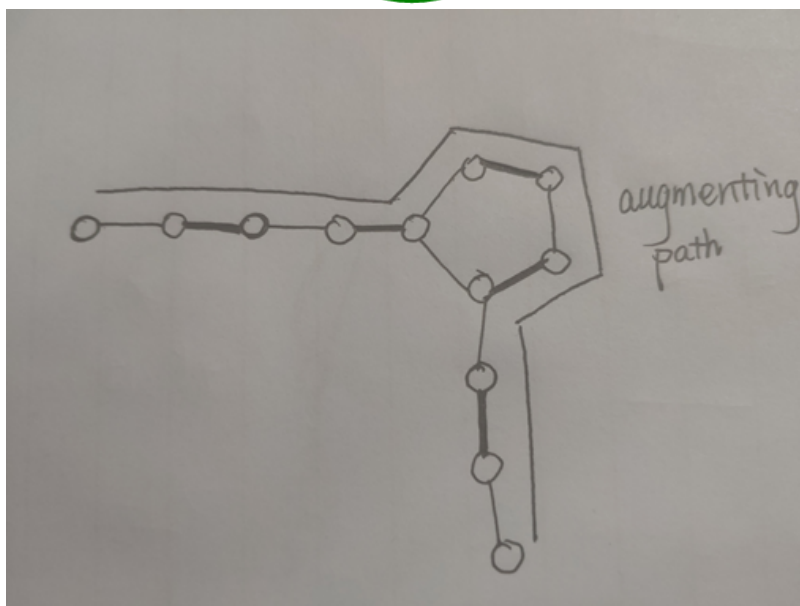
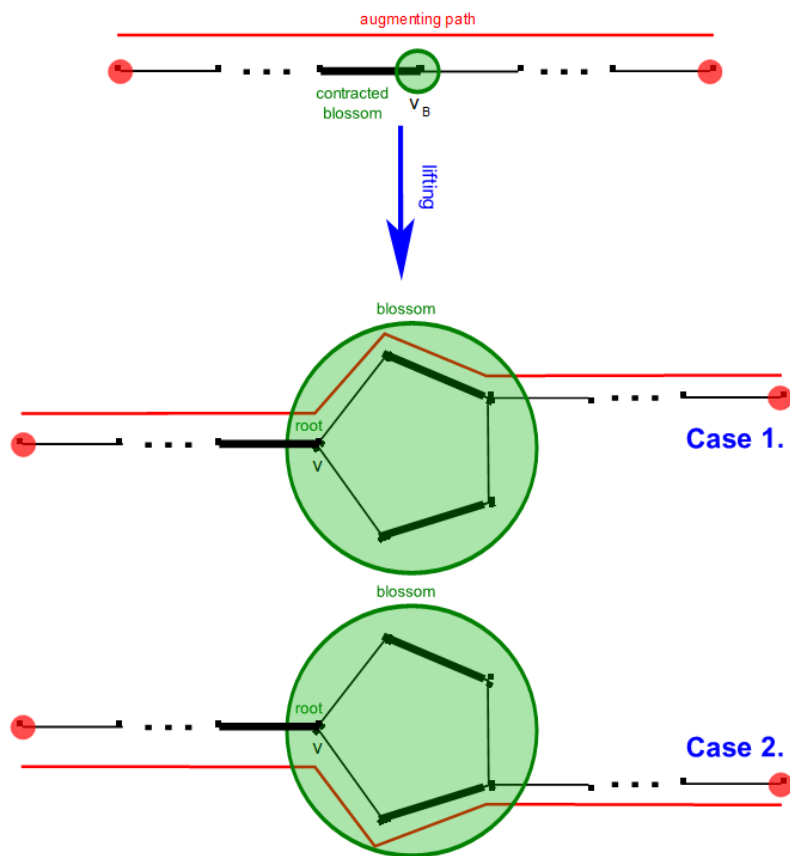
G' 含有 M' 的增广路当且仅当 G 含有 M 的增广路，且任何 M' 在图 G' 的增广路 P' 都可以通过展开收缩的花还原 G 中的匹配 M 。因此如果存在任意一条增广路 P' 通过收缩的花 V_B 都可以找到合适的增广路通过花 B 。因此我们可以将一朵花缩成一个点。



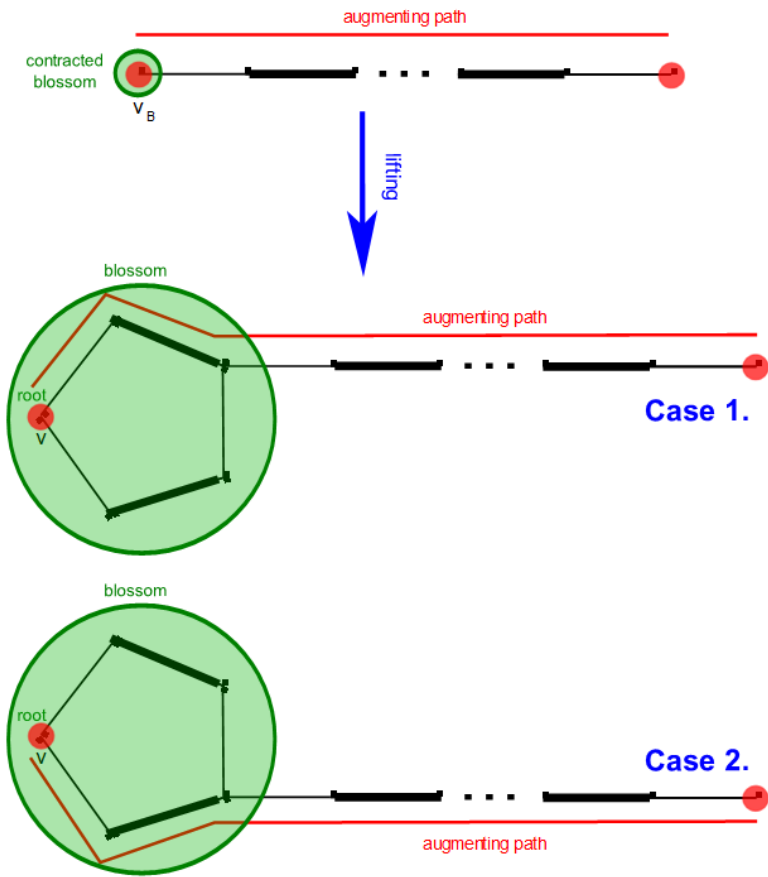
收缩的正确性证明

如以下几图所示，如果 P' 在图 G' 中经过 $u \rightarrow V_B \rightarrow w$ ，这条增广路可以在 G 中配替换为 $u \rightarrow (u \rightarrow \dots \rightarrow w) \rightarrow w$ ，其中 u' 与 w' 在花 B 中。路径 $u' \rightarrow w'$ 的选择需保证构成的新增广路仍然是交替的。（其中 u' 即在花 B 中也在匹配 M 中

(花根 $w' \rightarrow w$ 是一条增广路)



若 P' 在 V_B 结束，这条增广路可以在 G 中被替换为 $u \rightarrow (u \rightarrow \dots \rightarrow v)$ 其中 u' 与 v' 在花 B 中。路径 $u' \rightarrow v'$ 的选择需保证构成的新增广路仍然是交替的。
 (其中 v' 是孤立的 (花根 $u \rightarrow u'$ 是一条增广路))



From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: <https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:hotpot:%E5%B8%A6%E8%8A%B1%E6%A0%91&rev=1590138005>

Last update: 2020/05/22 17:00