

算法分析与设计基础 第十四周作业

徐浩博 软件02 2020010108

Problem 1 最长简单回路问题

首先,我们需要确定最长简单回路问题的形式化定义,我们将之定义为 $LONGEST_PATH = \{(G, k) : \text{在 } G=(V, E) \text{ 中存在长度至少为 } k \text{ 的简单回路}\}$.

我们先证明 $LONGEST_PATH$ 是NP问题. 对于一个certificate (v_1, \dots, v_k) , 必须要满足 $v_1 = v_k$ 才能保证是一个回路, 且必须保证该路径除了首尾没有其他重复的点, 这样才能是一个简单回路; 其次是 $v_{i-1}v_i (\forall 2 \leq i \leq k)$ 必须是图G中的一个边; 最后是我们将这个certificate的各边加起来查看长度是否不小于k. 综合以上三点, 我们在 $O(k)$ 时间内可以验证certificate, 即 $LONGEST_PATH$ 是NP问题.

其次我们要证明 $HamiltonLoop \leq_P LONGEST_PATH$. 首先, 哈密顿回路是NP完全的, 我们要将哈密顿回路归约到 $LONGEST_PATH$ 上. 假设一张图G, 我们要判定它是否有哈密顿回路, 它等价于判断 $(G, |V|) \in LONGEST_PATH$ 是否为真 (是否存在长度至少为 $|V|$ 的简单回路): 一方面, 如果有哈密顿回路, 那么可以选择该哈密顿回路作为简单回路, 长度为 $|V|$, 因此 $(G, |V|) \in LONGEST_PATH$; 另一方面, 如果 $(G, |V|) \in LONGEST_PATH$, 说明G有一条长度至少为 $|V|$ 的简单回路, 再由简单回路中节点不能重复, 因此长度 $\leq |V|$, 故必然有一条长度为 $|V|$ 的简单回路, 它必然将所有节点访问一遍且是简单回路, 那么它就是哈密顿回路, 即G中存在哈密顿回路. 因此定它是否有哈密顿回路, 它等价于判断 $LONGEST_PATH(G, |V|)$ 是否为真, 故哈密顿回路可以归约到 $LONGEST_PATH$, 而我们的归约算法是 $O(1)$ 的 (并未进行新的构造), 故 $HamiltonLoop \leq_P LONGEST_PATH$.

综合以上两点, $LONGEST_PATH$ 是NP完全的.

Problem 2 独立集

a. 独立集简写为IS(INDEPENDENT SET), 形式化定义为:

$$IS = \{(G = (V, E), k) : \exists V' \subseteq V, |V'| \geq k, s.t. \forall e = (v_i, v_j) \in E, \text{有 } v_i \notin V' \text{ or } v_j \notin V'\}$$

. 下证IS是NP完全的.

首先证IS是NP问题. 对于 (G, k) , 有某个certificate V' , 我们只需逐一验证E中的每条边e两个顶点是否同时在 V' 中, 并且需要保证 $|V'| \geq k$. 因此, 验证算法是 $O(|E|)$ 的复杂度, 即IS是NP问题.

其次, 我们要证明 $CLIQUE \leq_P IS$. $CLIQUE$ 问题是完全的, 如果要判定 (G, k) 是否满足 $CLIQUE$, 那么我们将G作如下变换: 构建 $E' = \{(v_i, v_j) : v_i \neq v_j \wedge (v_i, v_j) \notin E\}$, $G' = (V, E')$; 判定 (G, k) 是否满足 $CLIQUE$ 转化为判定 (G', k) 是否属于IS. 这两个问题等价: 如果 (G, k) 属于 $CLIQUE$, 那么必然存在 v_1, \dots, v_k 在G中两两之间均有边, 而在 G' 中这k个点两两之间必然无边, 因此令 $V' = \{v_1, \dots, v_k\}$, V' 满足任意一边的两个顶点必有一个不在 E' 内, 也即 (G', k) 属于IS; 反之, 如果 (G', k) 属于IS, 那么必然存在 v_1, \dots, v_k 在 G' 中两两之间无边, 因此在G中两两有边, 构成一个k度的团, 有 (G, k) 属于 $CLIQUE$. 而从G转换为 G' 的复杂度是 $O(\|V\|^2)$ 的, 为多项式复杂度, 因此有 $CLIQUE \leq_P IS$.

综合以上两点, IS是NP完全的.

b. 伪代码如下:

```

1 FIND-IS(V, E):
2     k = n
3     while TRUE:
4         if IS(V, E, k) == TRUE:
5             break
6         k -= 1
7
8     V' = V - v
9     E' = E - {each edge related to v}
10
11    for each v in V:
12        V' = V' - v
13        E' = E' - {each edge related to v}
14        if IS(V', E', k) == FALSE:
15            V' = V' + v
16            E' = E' + {each edge related to v}
17    return V'

```

简单来说, 我们枚举每个V中的节点, 尝试删去该节点及相关的边, 查看独立集的最大值是否改变, 如果改变, 则说明该节点在我们要求的独立集里. 显然, 该算法的复杂度为 $O(|V|)$.

我们来证明算法正确性: 首先, 删去节点只可能会使IS最大值k变小 (当然可能不变), 原因是我们考虑它的补图 G' (具体在a.中已定义), 删去节点显然只会使得团的最大值减小, 不可能变大, 故IS最大值也只会变小不会变大. 在此基础上, 我们遍历删去不会使IS最大值k变小的节点. 理论上, 我们经过一轮遍历只会剩下k个点, 这是因为如果保留超过k个节点, 假设为 v_1, \dots, v_{k+m} , 而 v_{k+m} 被保留下来的条件是删去该节点后IS最大值减小为k-1, 那么事实上 v_{k+m} 与另外k-1个点组成IS后, 剩下的m个节点肯定会于我们的遍历过程中被删除, 产生了矛盾, 于是我们经过一遍遍历后肯定只会剩下k个节点, 而这也是我们要求的 V' .

```

1 FIND-IS1(V, E):
2     V' = ∅
3     let all elements in array visited[] be 0
4     for each v in V:
5         if visited[v] == 0:
6             visited[v] = 0
7             dfs(v, -1)
8     return
9
10 dfs(v, father):

```

```

11     if father is not in V':
12         add v to V'
13     for each u in V linked to v:
14         dfs(u, v)

```

我们要证明这样一个结论：该图G每个连通块均是一个圈。我们来讨论每个连通块，由握手定理 $v = \frac{1}{2} \sum_{i=1}^n \text{degree}(v_i) = n$ ，我们取一个生成树，显然生成树的边数为 $n-1$ ；而V中只有一条边，因此生成树中有且只有两个度数为1的节点（如果大于2，则必然有节点不能通过新增边而满足度数为2；如果小于2，则树中的节点度数已经达到2，端点之间无法连边，不满足边数为 n ），剩余的一条边只能连在这两个点间，于是此连通块构成了一个圈。若干圈拼成了G。对于独立集来说，各个连通块之间选择V'中的节点是独立的。对每个圈，用dfs交错取点放进V'，直到不能再取为止，显然，这样的独立集是最大独立集。在该算法中，由于所有节点最多访问一次，所以复杂度 $O(|V|)$ 。

d. 我们首先证明最大独立集数等于 $|V| - |M|$ ，其中M是最大匹配。我们由二分图的Konig定理（证明见下一段）：二分图最大匹配数等于最小点覆盖数。而最小点覆盖是指最少用多少标记点使得每一条边至少都有一个端点被标记里。反向考虑：点覆盖就是用点拆掉所有边，因此剩下来点的构成独立集；而最小点覆盖就意味着用最少的点拆边，剩下来最多的点构成独立集。因此用最小点覆盖数 $|M|$ 拆边后，剩下最多的点构成独立集，剩下的点数就为 $|V| - |M|$ 。

Konig定理：最大匹配数=最小覆盖数首先最小覆盖数 \geq 最大匹配数，因为最大匹配中的边必须由 $|M|$ 个点来覆盖，因此最小覆盖数 $\geq |M|$ 。下面要证最小覆盖数 \leq 最大匹配数，我们将二分图分为左L右R两个部分，进行如下构造：对于每个L中未匹配的节点，我们寻找它们的增广路，标记寻找增广路过程中经过的节点，取L中被标记的点和R中未标记的点之并为K。首先明确，所有左部未匹配节点都被标记（对它们找增广路肯定包含自己）；所有右部未匹配节点都未被标记（否则找到未匹配节点则存在一条完整的增广路，与M是最大匹配矛盾），且M中每个匹配中节点要么同时被标记，要么同时没被标记（根据寻找增广路的过程得知）。下证K是一个覆盖，讨论所有E中的e：如果e属于M，则要么e两个节点同时被标记，此时被右节点覆盖，要么同时没被标记，此时被左节点覆盖；如果e左端属于M右端不属于M，假设左端点被标记则根据寻找增广路的过程，右端点也被标记，与右端点未匹配则一定未被标记矛盾，故左端点一定未被标记，则左端点在K内，e被覆盖；如果e左端不属于M，则未匹配节点均被标记，左端点属于K，e被覆盖。综上，K是E的一个覆盖，而对于每个M中的边，如果端点同时被标记则左端点在K内，否则右端点在K内，因此 $|M| \leq |K| \leq$ 最小覆盖数。综合以上两点有最大匹配数=最小覆盖数。

结合上面的分析，我们可以获得最大独立集：首先利用最大流Ford-Fulkerson或者匈牙利等算法获得最大匹配，然后将最大匹配中每一个匹配任选一个点加入V'中，再将没有匹配上的所有点均加入V'中，构成的就是最大独立集。复杂度 $O(|V||E|)$ 。