# 最大团问题的分支限界算法

## 最大团问题的定义

最大团问题是指在一个简单无向图G = (V, E)中寻找顶点最多的一个子图C，使得。在本文中，我们使用下列定义：

- 补图——给定一个图G(V, E)，它的补图定义为 (V, )，其中。

- 极大团——图的一个团，该团不能通过添加图的顶点而扩大。

- 独立集——图G = (V,E)的子图S，其中。显然，图的最大团是其补图的最大独立集。

- 候选顶点——可以用来扩充当前团的顶点，即与当前团所有顶点都邻接的顶点。

- 颜色数——在着色过程中赋予顶点的自然数。

- 染色数——对图进行着色所需的最小颜色数。

分支限界算法是求解最大团问题最常用的算法。算法步骤一般分为分支部分与限界部分，其中分支部分选择候选顶点集中的一个顶点进行分支，将该顶点添加到当前团中，在新的节点（新的递归过程）中，候选集被更新为原候选集中与该顶点邻接的所有顶点，因此候选集中所有的顶点都邻接于当前团的所有顶点。而限界部分计算当前候选集中，最大团的大小的上界，如果该上界加上当前团的顶点数不大于目前找到的最大的团，则该分支不可能产生更好的团，因此需要裁减该分支。

本文主要工作在于研究目前最先进的几个分支限界算法，并在这个基础上，构造出效率更好的求解算法。

## 下界与代价函数

分支限界法算法中，分支部分保证了算法的完整性与精确性，而限界部分则决定了算法的效率。为了提高算法效率，关键在于估算出尽可能紧的代价函数，即团大小的上界，以及尽快找到一个大的下界，即一个尽可能大的团。

估计代价函数时，一个平凡的做法是以候选集的顶点个数作为代价函数，如果当前团大小加上候选集顶点个数不超过下界，则可以当前分支无需进入下一轮迭代。然而，这样的代价函数过于松散，未能精确估计团的上界，导致算法效率不高。

### 2.1 染色数

着色过程所需要的颜色数（大于或小于染色数），可以作为最大团问题的上界（命题1）。

**命题1**（Balas 和Yu 1986）[]任意图的最大团大小不大于该图的染色数。

这个命题可以从以下事实直接得出：一个具有k 个顶点的团至少需要用k 种颜色进行着色，因为这些顶点都是两两邻接的，因而团的任意两个顶点不能具有同一种颜色。

### 2.2 贪心着色

染色数问题所对应的判定问题，本身就属于NP难问题，因此使用染色数作为最大团问题的代价函数是不实际的。另一种方案则是使用成为贪心着色[]的方法，求出染色数的近似解，作为最大团问题的代价函数。贪心着色过程遵循如下原则：

- 如果(v1,，v2)∈E，则No[v1] ≠ No[v2]

- No[v] = 1，或者No[v] = k > 1，则存在顶点v1∈NR[v]，v2∈NR[v]，...，vk-1∈NR[v]，使得No[v1] = 1，No[v2] = 2，....，No[vk-1] = k-1。

其中，No[v]表示顶点v的颜色编号，NR[v]表示顶点v在子图R中的邻居。

显然，贪心着色过程所用到的颜色数可作为最大团问题的代价函数。对于顶点个数为n的图，贪心着色过程的时间复杂度为O(n2)。

### 2.3下界与分支顺序

好的分支顺序能够使算法快速更新下界，从而在后续迭代中得以裁剪更多的分支。贪心着色过程为分支顺序做了很好的准备——优先选择颜色编码大的顶点。因为团最多只能在每个颜色编码中选取其中一个顶点，而颜色编码最大的顶点，在所有其他颜色编码中都存在与之相邻的顶点。另外，这种分支顺序使得在迭代树的同一层中，只需要进行一次贪心着色过程，因为排除了所选择的顶点后，新一轮贪心着色过程不会产生新的着色方案。

## 2.4 顶点顺序对着色的影响

贪心着色过程根据既定的顺序，依次对顶点进行着色。不同的顺序对着色效果（即所拥得颜色数）有很大的影响。

### 重着色

### 初始化

### 三、动态排序

## 四、Maxsat推理