

高级算法: 作业三

xueshi hu

Section 1

问题描述: 请介绍一个问题 (包括问题背景和定义), 并介绍该问题的求解方案, 包括 (1) 该问题到一般模型的编码, 包括 SAT, MaxSAT, CSP, MIP 等; (2) 针对该问题 (原问题形式) 设计一个局部搜索算法, 需要给出基本数据结构和伪代码。此外, 针对算法给出一定的理论分析, 或给出某些实现要点, 二者取其一 (当然都做也可以, 更好。)。以小论文 (语言不限, 可中文可英文) 的形式提交, 字数没有限制。

Section 2

图着色 (Graph Coloring) 问题指的是, 对于一个无向图 $G(E, V)$, 是否存在指定数量的颜色将所有的顶点染色, 其对于任意一个边, 其两个顶点的颜色不同。其中取相同颜色的顶点可以构成独立集 (Independent set)。

使用 SAT 对其进行编码, 上课讲解过一种编码方法是基于点的, 简单复述一遍, 加入共有 k 种颜色, x_{ij} 表示顶点 i 取第 j 个颜色, 由于每个顶点至少需要一个颜色, 构造子句 $\bigvee_{j=1}^k x_{ij}$, 相邻的边的颜色不可以冲突, 对于边 mn 颜色 i , 构造子句 $\neg(x_{mi} \wedge x_{ni})$

下面提出一种新的基于边的编码方法:

- 对于边 i , 如果顶点编号较小的那个的颜色为 m , 另一个点的颜色为 n , 那么变量 x_{imn} 为真
- 每个点的颜色不能互相冲突, 顶点 i 的两个临边不可以将其染色为 m 和 n , 可以构造 $\neg(M \wedge N)$, 如果顶点 i 的临边的个数为 N_{ei} 这种子句的数量为 $\binom{2}{N_{ei}} k(k-1)$, 其中 $\binom{2}{N_{ei}}$ 表示选择的两个边可能性, $k(k-1)$ 表示一共可以选择的颜色种类。
- 所有的边最少存在一种染色模式, 可以构造子句 $\bigvee_{m=1}^k \bigvee_{n=1}^k x_{imn}$

Section 3

下面考虑使用禁忌算法 (Tabu) 和混合进化 (Hybrid Revolutionary) 实现图着色算法, 代码在

图着色问题, local search 的基本方法是, 首先对于所有的点使用给定的颜色进行随机着色, 然后冲突最多的点, 将其颜色进行替换。Tabu 的思想在于让 local search 不要走回去, 在图着色问题中间, 对于刚刚修改颜色的点, 不要在修改颜色。

下面简单分析其实现:

- 初始化数据结构, 关键的为:

Table 1: Tabu Search Experiment Results

序号	数据	染色数	时间 (s)	禁忌步长
1	DSJC125.1	5	0.013845	10
2	DSJC250.1	8	0.252603	10
3	DSJC250.5	28	95.3894	10
4	DSJC250.9	72	NULL	10
5	DSJC500.1	13	0.187932	10
6	DSJC500.1	12	NULL	10
7	DSJC250.9	73	NULL	10
8	DSJC250.9	100	0.055752	10
8	DSJC250.9	75	NULL	10
9	DSJC250.9	80	0.115334	10
10	DSJC500.5	51	6.62945	10
11	DSJC500.5	50	251.354	10
12	DSJC500.5	49	NULL	10
13	DSJC1000.1	22	0.920005	10
14	DSJC1000.1	25	0.020005	10
15	DSJC1000.1	21	36.9668	10

- color_conflict_num : 二维数组, 记录所有的顶点, 使用不同颜色的冲突数量
- tabu_tenure : 记录一个顶点发生修改的时间点
- 每步, 遍历整个 color_conflict_num 找到没有被禁忌的最大值, 更新数据结构

相对朴素的 tabu search, 存在两个简单的点需要加以说明:

- tabu tenure 的实现, 一种方法是, 存储每一个顶点被 tabu 的时间, 然后每一步减少 1, 直到不再被禁忌。但是, 更加高效的实现方法是采用类似于时间戳的方法, 记录当前的迭代次数, 每次更新一个节点的时候, 在数组 tabu_tenure 中间记录修改的迭代次数, 通过检查两者的差值来确定时间。
- 当被禁忌的节点可以导致冲突数量减少, 而没有被禁忌的节点会导致冲突数量增加的时候, 此时, 无视禁忌。

初步测试结果在表格1中。

Section 4

混合进化基于禁忌算法, 其思想源自于生物, 创建一个种群, 杂交, 去掉不良个体, 如此迭代。在图着色问题中间, 首先利用禁忌算法计算出来一组解, 从中间随机选择两个解构造的新的解, 构造方法是交替选择其中一个解的染色节点, 对于构造的解, 运行禁忌算法, 将得到的解替换掉最差的解。

Table 2: HEA Search Experiment Results

序号	数据	染色数	时间	禁忌不长	种群大小
1	DSJC125.1	5	0.063239	10	5
2	DSJC250.1	8	0.723432	10	5
3	DSJC500.5	52	10.3452	10	5
4	DSJC500.5	51	33.7604	10	5
5	DSJC500.5	50	109.607	10	5
5	DSJC500.5	49	NULL	10	4
5	DSJC500.5	49	NULL	10	10
6	DSJC1000.1	25	1.69291	10	5

说明一下”杂交”的过程:

1. 将禁忌算法的结果表示为二维数组 solution，那么 solution[i] 表示颜色全部染成 i 的节点
2. 对于”亲本”，solution1 和 solution2，交替选择，构造子代 offspring，也就是当 $i \% 2 == 0$ 成立的时候，offspring[i] 节点来自于 solution1[i]，否则来自于 solution2[i]
3. 上面的构造过程中，当一个顶点被放到 solution[i] 之后，就不能放到其他的 solution 中间，这会导致一些节点无法染色，对于这些点，采取随机染色。

初步测试结果在表格2中。

两者实现的代码部署在 github 上: <https://github.com/Martins3/OperationalResearchLab>