

图生成与分解器

李子龙

电子信息与电气工程学院

518070910095 F1903301

2020 年 12 月 24 日

Copyright © 2020 by LogCreative, All Rights Reserved.

To learn more about the author, please visit <https://github.com/LogCreative>.

(LC) No. 0211

1 问题重述

1. 开发一个图自动生成器

随机生成一个有向图，将图放置到指定文件中，每一行如下格式

- $\langle \text{节点编号} \rangle$: 节点
- $\langle \text{出发节点编号}, \text{结束节点编号}, \text{权重} \rangle$: 有向边

2. 开发一个图分解器

- 分割图文件

- 将上述图分为若干子图，每个子图中节点数不大于 n 。
- A 图分割后，每个子图可以用单独的文件保存：如 A_1, A_2, A_3, \dots
- 令子图之间的交互（即能够跨越子图边界的边）权重之和最小，我们将挑选若干自动生成的图，对比大家生成的权重之和值。在结果正确的前提下，计算权重之和越小，分数越高。

- 优化子图存储

上述图分割算法导致分割成的多个子图之间存在重复的节点，请设计一个方法，使

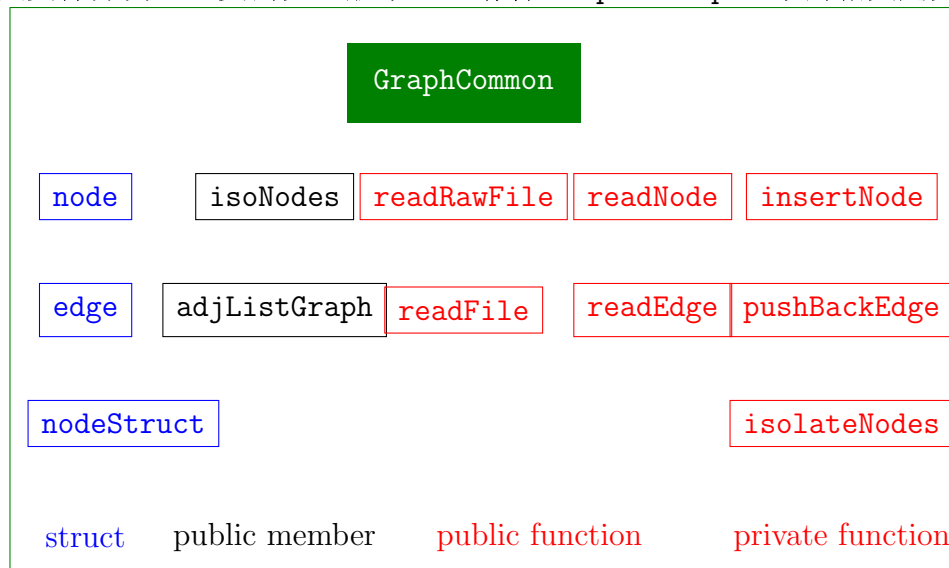
- 多个子图文件中分别载入程序后，不存在重复的节点
- 每个子图可以最多增加一个虚节点（如子图的文件名），代表外界（即其他子图）对该子图的引用
- 设计一个算法，将多个子图合并及删除虚节点后，检查与原图 A 一致。输出分割边的权重和。

- 子图上算法

- 指定一个点，列出计算所有经有向边可达的节点
- 指定两个点，输出最短路径
- 如果指定的节点不存在，报错即可

2 公共类 GraphCommon

在开始的无 UI 编译时期，存储为同一个公共类 GraphCommon，后面因为需要使用全局的非静态变量，将头文件分离。该类的完整版本还是请看 GraphDecomp.h 中的相关定义。



2.1 结构体

该类定义了三个小结构体：

- 节点类 node
- 有向边类 edge
- 节点结构 nodeStruct（用于计算邻接矩阵）

前两个都重载了输入输出运算符，以符合目标格式。

	输入	输出
node	<P1>	<1>
edge	<P1 P2 2.0>	<1,2,2.0>

前缀符号以及分割符可以在后面调整。值得一提的是，优化后子图的虚边定义为如下的格式：

<起始点,虚点符号,<文件名.终止节点,权重>>
<1,-1,<01.2,2.0>>

在本程序中，-1被定义虚节点符号。

节点结构中定义了三个成员：

- 邻接矩阵边存储
- 邻接矩阵列数值
- 节点发出边总权重

并通过两个私有函数同步更新这些数值。

2.2 公共成员

该类定义了两个公共成员：

- `set<int> isoNodes` 包含了所有的孤立节点(isolated nodes)，也就是完全不连通的部分。
- `map<int, vector<edge>> adjListGraph`，邻接表图，仅包含连通部分节点，对于有些连通节点可能为发出空边的集合，即 `adjListGraph[node] = vector<edge>()`；。

2.3 公共函数

该类定义了四个公共函数。

- `readNode` 通过输入文本流读取明确定义的节点信息
- `readEdge` 通过文本流读取有向边的信息。
- `readFile` 通过文本流读取文件信息。
- `readRawFile` 通过文本流读取特定格式的文件信息。

2.4 私有函数

该类定义了三个私有函数。

- `insertNode` 插入节点，包含了对虚节点的检查机制。
- `pushBackEdge` 向邻接表插入边，也包含了对节点是否为虚节点的转换检查机制。
- `isolateNodes` 将 `readNode` 后的集合去除根据 `readEdge` 所读取的连接边点，变为孤立节点的集合。

3 图生成器 GraphGen

图生成器的类 `GraphGen` 是 `GraphCommonGen` 的派生类。

本程序的图生成器有几个参数需要设置：

- 节点类型 `nodeType`: `continuous`连续编号的, `discrete`离散的。
- 边生成类型 `edgeType`: `Tree`树（不含环路）, `Graph`图（带有环路）。
- 连通图类型 `isoType`: `Single` 单个连通图, `Multi` 多个连通图。
- 节点编号增长量 `MAX_INCREASEMENT`: 在离散编号模式下，每次生成一个节点都会增长一个数字，这个数字不会超过最大增长量。
- 最大孩子数 `MAX_CHILD`: 每个节点的发出有向边个数不会超过最大孩子数。
- 最大连通子图数 `MAX_ISOGRAPH`: 在多个连通图生成模式下，每个图的连通子图数不会超过最大连通子图数。

- **节点行数占比 Node / Lines:** 在新文件模式下，仍然会生成随机个数的节点数，但是只输出占比量的节点行数，其余为有向边的行。

该程序将会根据上述参数，递增而随机地生成节点编号。然后通过层序遍历生成各个边，如果没有环路的限制，则有可能随机到一个环路节点上去。

随机数采用下面的代码生成：

```
1 srand((unsigned)time(0)*(++gseed));  
2 return 1.0 * rand() / RAND_MAX;
```

当然，这种方式依然不是特别特别随机，但已经足够。