最小割问题的并行化求解

杨树　18002710003

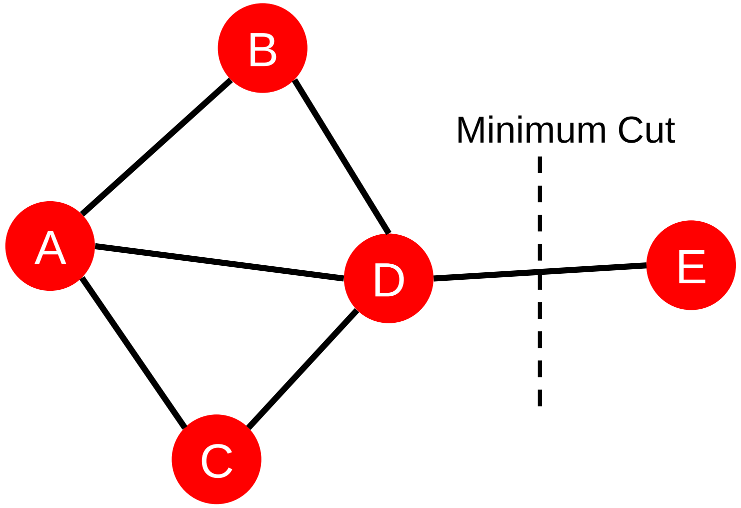
1 问题描述

最小割（Minimum Cut)

对 于 无 向 连 通 图 G=(V, E), 求 一 种 分 割 方 法

将 所 有 顶 点 划 分 成 互 补 且 不 相 交 的 两 个 子 集， 使 得 被 切 断 的 边 的 数 量 最 小； 如 果 边 有 权 值， 则 使 得 被 切 断 边 的 权 值 和 最 小

这里主要针对不带权的无向图求Global MinCut

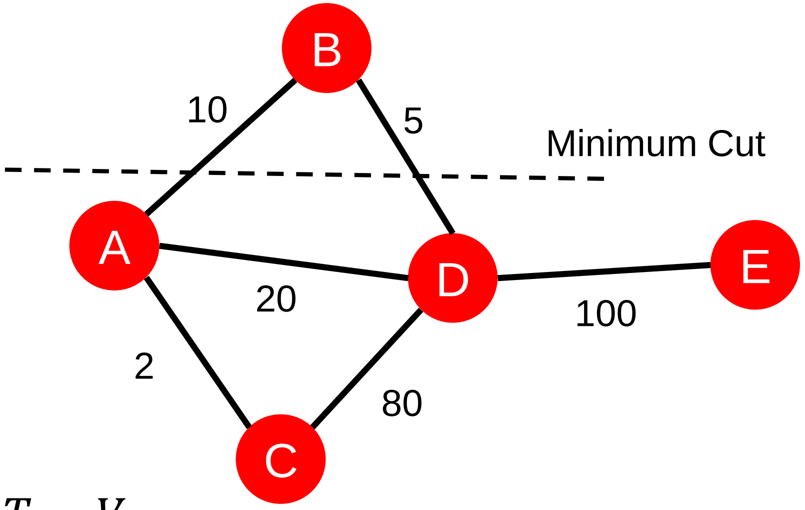


$S = \{A, B, C, D\}$

$T = \{E\}$

$S \cap T = \emptyset$ 且 $S \cup T = V$

CutSize = 1



$S = \{A, C, D, E\}$

$T = \{B\}$

$S \cap T = \emptyset$ 且 $S \cup T = V$

CutSize = 15

算法1　转换成最大流问题

- **Max-Flow Min-Cut Theorem** [Ford and Fulkerson, 1956]

  In every network, the maximum value of a feasible flow equals the minimum capacity of a source/sink cut.

- 给定源点和汇点(source and sink)，使用求解最大流的相关算法进行求解。

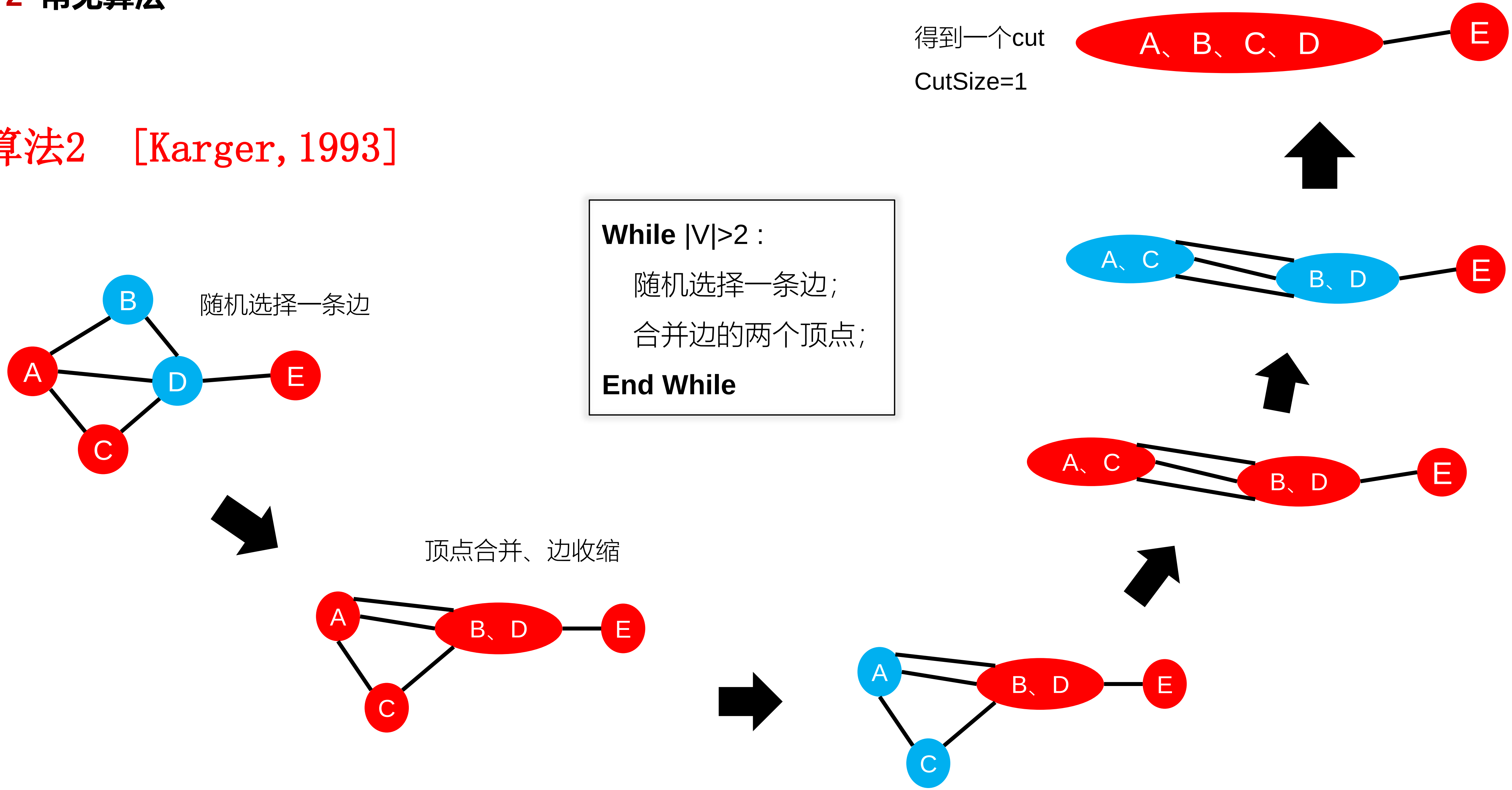
  比较好的一种算法时间复杂度为 $\mathcal{O}(\,|V|^2\,|E|\,)$ [Goldberg and Tarjan, 1988]

- 对于源点和汇点的选择，最Naive的方式就是遍历所有的顶点对，则总时间复杂度为 $\mathcal{O}(\,|V|^4\,|E|\,)$

  若该图是稠密的，即 $|E| \approx |V|^2$，则时间复杂度为 $\mathcal{O}(\,|V|^6\,)$

2 常见算法

算法2 [Karger, 1993]



While |V|>2 :

随机选择一条边;

合并边的两个顶点;

End While

算法2 [Karger, 1993]

- 记 $|V| = n$
- 算法只需要迭代 $n-2$ 次即可得到一个Cut
- 可以证明得到，该算法得到的 Cut 是 Minimum Cut 的概率 $P \geq \frac{2}{n(n-1)}$
- 重复执行该算法 $\binom{n}{2}\ln n$ 次，可将算法失败（得到的不是全局最小割）的概率降至 $\frac{1}{n}$ 以下

$$\left(1-\frac{2}{n(n-1)}\right)^{\binom{n}{2}\ln n} \leq e^{-\ln n} = \frac{1}{n}$$

- 时间复杂度为 $O(n^4 \ln n)$

算法3 [Karger and Stein, 1996]

- 记 $|V| = n$
- 可以证明得到，该算法得到的 Cut 是 Minimum Cut 的概率 $P \geq \frac{1}{\ln n}$
- 重复执行该算法 $(\ln n)^2$ 次，可将算法失败（得到的不是全局最小割）的概率降至 $\frac{1}{n}$ 以下

$$\left(1 - \frac{1}{\ln n}\right)^{(\ln n)^2} \leq e^{-\ln n} = \frac{1}{n}$$

- 时间复杂度为 $O(n^2(\ln n)^3)$

```
FastMinCut (G) {
    If |V| < 6
        return minCut (G, 2)
    Else
        t = 1 + |V|/√2
        graph_1 = minCut (G, t)
        graph_2 = minCut (G, t)
        return min (FastMinCut(graph_1), FastMinCut(graph_2))
}
```

时间复杂度 $O(n^2 \ln n)$

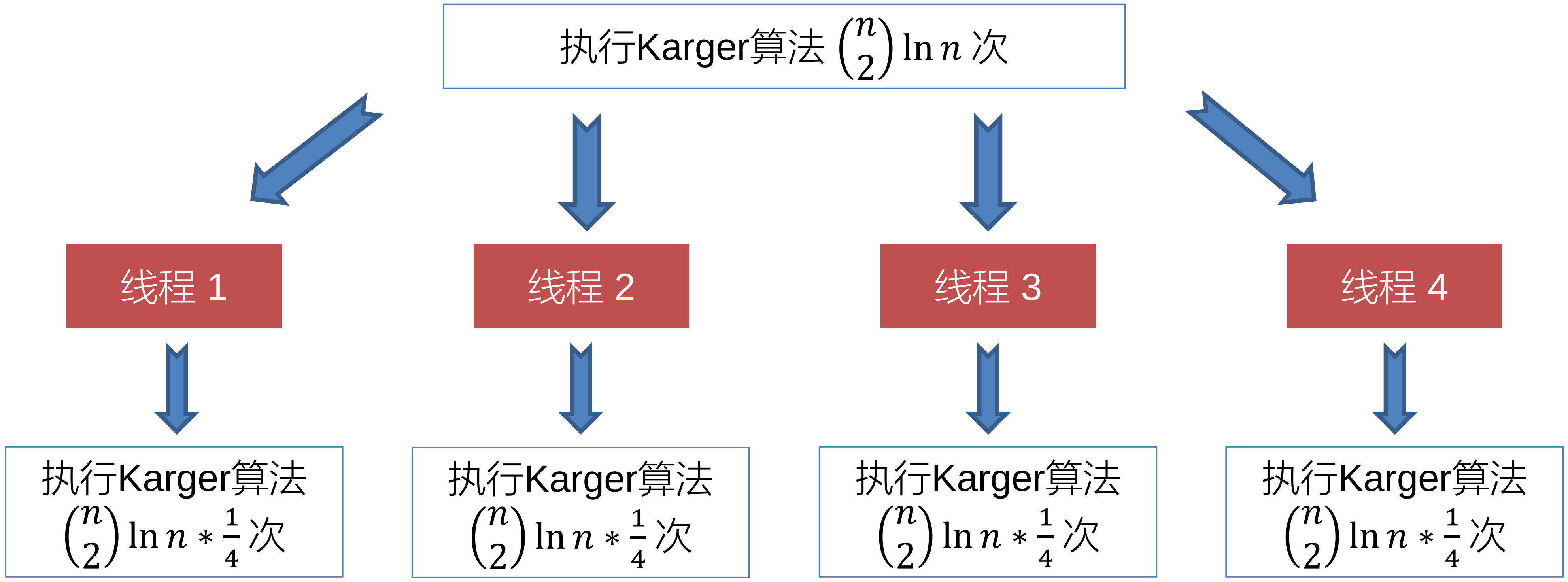
时间复杂度 $O(n^2)$

```
minCut (G, t) {
    While |V| > t
        随机选择一对顶点
        合并
    End While
    return G
}
```

3 Karger-Stein 算法的并行化

一、由于算法需要多次执行，所以很适合并行，可以将循环展开，利用多线程进行处理

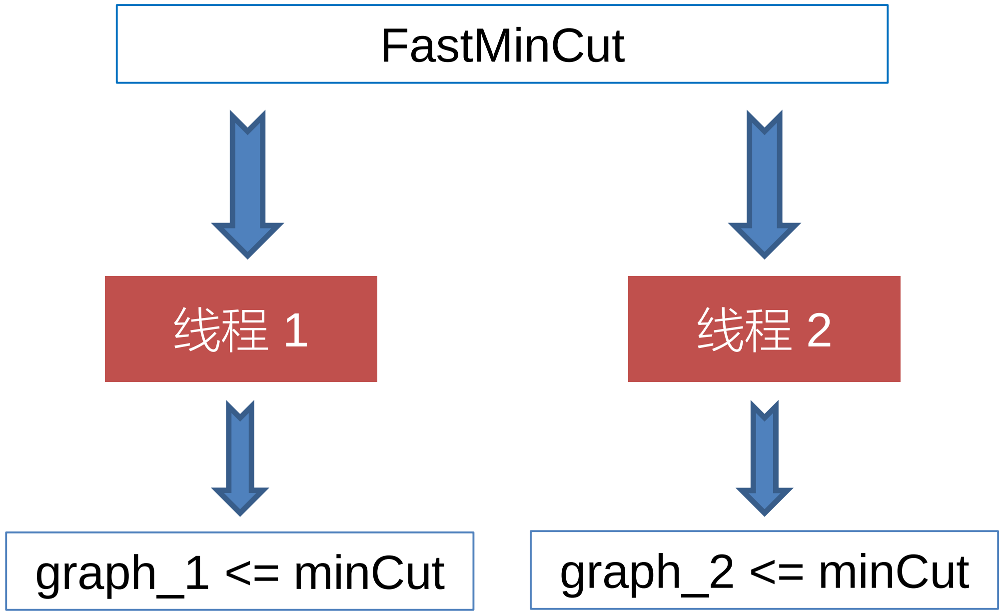
例如：Karger算法



3 Karger-Stein 算法的并行化

二、每次执行 FastMinCut 算法时，都会重复调用两次 minCut 函数，没有相关性所以能并行执行

```
FastMinCut (G) {
   If |V| < 6
      return minCut (graph, 2)
   Else
      t = 1 + |V|/√2
      graph_1 = minCut (graph, t)
      graph_2 = minCut (graph, t)
      return min (FastMinCut(graph_1), FastMinCut(graph_2))
}
```



参考文献

1. Lester R Ford and Delbert R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399-404, 1956.
2. Andrew V Goldberg and Robert E Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921-940, 1988.
3. David R Karger. Global min-cuts in rnc, and other ramifications of a simple min-out algorithm. *In Proceedings of the fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 1993:21-30.
4. David R Karger and Clifford Stein. A new approach to the minimum cut problem. Journal of the ACM (JACM), 43(4):601-640, 1996.

Thanks

杨树 18002271003