

Answer to diameter,maximal minimal path length Graph structure in the Web

Chunwei Yan

2012 年 10 月 26 日

1 前期假设

1.1 exploding nodes

作者通过 SCC 算法，得到了 SCC 的规模 $explode_3$ 。为了预测 IN 和 OUT 中节点的数目，提出了节点爆炸的概念。

在利用 BFS 算法遭遇到一个节点后，有时会把下一步搜索的候选节点极大增加 (如增长到 1 亿)，我们称这种情况为爆炸。

IN 和 SCC 中的节点会在入链 BFS 中爆炸，OUT 和 SCC 中的节点会在出链 BFS 中爆炸。可以得到

$$IN + SCC = explode_1$$

$$OUT + SCC = explode_2$$

如此，能够得到 IN 和 OUT 的数目。

$$\begin{cases} IN = explode_1 - explode_3 \\ OUT = explode_2 - explode_3 \\ SCC = explode_3 \end{cases}$$

作者然后利用了一个实验进行了验证，实验分为两种情况：

1. 在随机选择的起始节点运用 BFS 算法，其中前链和后链均爆炸的一部分节点包含于 SCC 算法生成的 SCC 节点集合中
2. SCC 集合中的任何一个节点在入链 BFS 算法中能够触及的节点数目均相同

1.2 SCC

作者在 3.3 章节专门介绍了对 SCC 直径的判断方法。

似乎是利用了抽样调查，从 SCC 取样 136 个节点，利用 BFS 测试，取得最大深度。得到一张表：

Measure	Minimum depth	Average depth	Maximum depth
In-links	475	482	503
Out-links	430	434	444

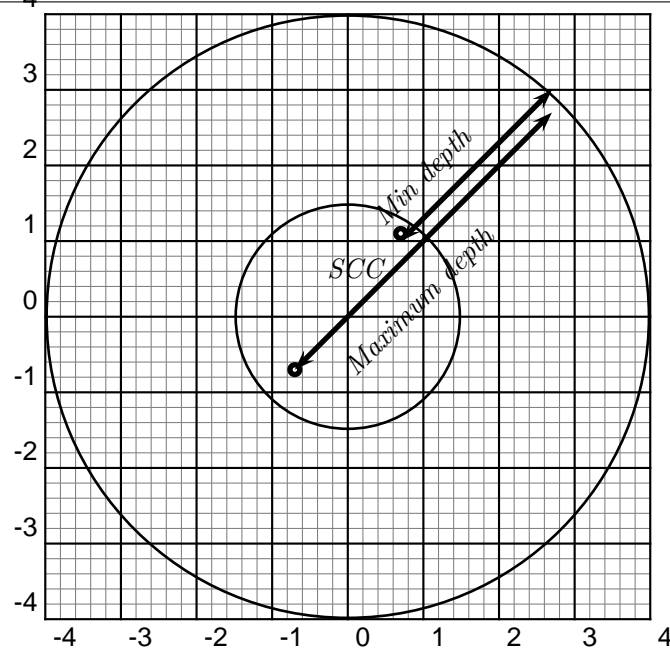


Figure1

Figure1 是一个比较理想的状态，通过图形，可以很简单判断：

$$diameterOfSCC = MaximumDepth - MinimumDepth$$

$$\max(503 - 475, 444 - 430) = 28$$

这就是课件中 SCC 直径 28 的由来。