**有向图中的最优保护路径的算法**

**1、背景**

假设存在已知连通**有向图1**，拥有顶点和边，相邻的顶点之间有且只有一条边进行连接（每条边都为双向，并且双向的权值相同），从任意顶点出发都能访问任意其他顶点。

假设选定任意顶点S、D，使用迪杰斯特拉(Dijkstra)算法计算S->D的最短路径Path1，而且Path1经过与S相邻的顶点E（即S与E相邻，且S->D最短路径Path1经过E）。

我们需要提前为Path1计算一条合适的保护路径Path1\_LFA，这条保护路径只用于保护当顶点E发生故障时，S可以通过保护路径到达D。

Path1\_LFA的计算思路，是将被保护的E顶点从有向图中去除，形成新的**有向图2**，并计算出新的S->D最短路径Path2，在Path2经过的顶点中，寻找P/Q顶点（允许P/Q是同一个节点），使得P/Q满足下面的条件：

1. 在**有向图1**中 S到P的最短路径（S->P）不经过E。

满足这个条件的P顶点可能存在多个，是一个集合。

1. 在**有向图1**中 Q到D的最短路径（Q->D）不经过E。

满足这个条件的Q顶点可能存在多个，是一个集合。

最终的保护路径Path1\_LFA可以被描述为：S->P…Q->D，这样的保护路径PQ组合可能存在多个，可以按照下面的优先级进行筛选：

1. 在**有向图1**中，P->Q的最短路径不经过E顶点的优于经过E顶点。
2. 在Path2路径上，P->Q所经过的顶点数量越少越优。

**2、现有解决方案**

目前Path1\_LFA保护路径的计算:

1）计算出Path2

2）计算P集合和Q集合；

3）根据Path2 计算出精简的P集合和Q集合。

4）在3）中的P/Q集合基础上进行逐个计算比较选取最优

当前的做法，只计算某一个目的顶点D时可以解决，但是如果对有向图中所有的顶点计算保护路径时（S->D1......Dn），会遇到性能瓶颈。

**3、问题**

请大家结合自身知识，结合图论及其算法领域的最新成果，调研业内的现状和进展，给出自己的理解和拟解决方案,实现有向图中的高性能保护路径计算。

要求：请您以Word输出整体运作方案，并将其中要点以PPT形式进行输出，在极致挑战环节进行宣讲。