浅析伪多项式复杂度的基于带势的Dijkstra算法的费用流

1.introduction

常见的费用流解法为用SPFA/Bellman_Ford 算法求解最小费用增广路，然后沿着该增广路增广，重复该过程直至增广路不存在。其复杂度为$\Omega(mf), O(nmf)$（其中f为流量，下同）

随机数据下其表现良好，但在网格图等特殊图中效率较低。如「NOIP2020联考 Day2 撤离」中，该算法难以得到超过80分。

此外，学术界已有$O(m^2 \log m \log f)$的弱多项式复杂度费用流解法。但其在流量较小时表现不佳。

本文将介绍伪多项式复杂度的基于带势的Dijkstra算法的费用流，其复杂度上为$O(m \log m * f)$ 。由于与f相关，其仍是伪多项式复杂度算法。

2.Algorithm

2.1 势函数

考虑不能直接使用Dijk算法的原因：残余网络上可能存在负边权。为使用Dijk算法，引入势函数Φ,对原图中的边$(u, v, cap, cost)$，建新边$(u, v, cap, cost')$,其中$cost' = cost + \Phi(u) - \Phi(v)$.我们需要通过选择适当的势函数$\Phi$,保证对于所有$cap > 0$的边都有$cost' \geq 0$.

Φ的具体选择将在2.3中讨论。

2.2 算法

考虑一条s到t的最短路径P,
$\sum_{e\in P} cost(e) = \sum_{e'\in P} cost'(e) - \Phi(from(e)) + \Phi(to(e)) = \Phi(t) - \Phi(s) + \sum_{e'\in P} cost'(e)$

因此运行Dijk算法后，求得的最短路$dist(t) + \Phi(t) - \Phi(s)$即为真正的最短路长度。沿该最短路增广即可。

重复该过程直至不存在增广路即可。复杂度为$O(m \log mf)$

2.3 势函数的选取与正确性证明

一种精妙的选取方式是，取该点之前的**最短路长度之和**为Φ.

正确性证明：只需证对于所有边$(u, v, w, cost')$,$cost' \geq 0$总成立。

2.3.1 对于原本就在残余网络中且增广后仍有容量剩余的边：

$\because dist(u) + cost' \geq dist(v), cost' = cost + \Phi(u) - \Phi(v)$

$\therefore (dist(u) + \Phi(u)) - (dist(v) + \Phi(v)) + cost \geq 0$

那么我们动态调整$\Phi(u) := \Phi(u) + dist(u)$即可满足条件。

2.3.2 对于原本没有容量，增广后获得了容量的边：

获得容量$\Leftrightarrow$ 反向边被增广$\Leftrightarrow dist(u) = dist(v) - cost'$.

显然是2.3.1的子集。

Q.E.D

对于势函数的更新我们直接用Dijk的结果即可。注意第一次求势函数时，若图中有负权边，则需用SPFA求出势函数初值，否则同样用Dijk算法。