arXiv:2303.06102v1 [cs.DS] 10 Mar 2023

动态距离预言机的自举

Sebastian Forster * Gramoz Goranci † Yasamin Nazari* Antonis Skarlatos*

摘要

在全动态设置下设计近似全点对距离预言机，是动态图算法中的核心问题之一。尽管这一课题已得到广泛研究，直到最近，Forster、Goranci 和 Henzinger 才提出了首个在任何非平凡近似下突破更新时间 $O(\sqrt{n})$界限的结果 [SODA'21] 他们针对任意参数 $\rho \geq 1$实现了摊销更新时间，并使近似比中的因子为 $O(\log\ n)^{3\rho-2}$。

本文提出了首个更新时间和查询时间均较小且为多项式级的常数伸展全动态距离预言机。以往工作要么要求至少达到多对数近似，要么需要大得多的更新时间。我们的结果在伸展因子与更新时间之间提供了更精细的权衡。例如，对于常数参数 $\rho < 1$，我们可以实现常数伸展因子 $O(\frac{1}{\rho^2})^{4/\rho}$、摊销更新时间 $\tilde{O}(n^{\rho})$以及查询时间 $\tilde{O}(n^{\rho/8})$。我们的算法采用随机化方法，并假设对手是遗忘型对手。

我们构造中的一个核心技术思想，是设计一种从递减近似枢纽标记方案到全动态距离预言机的黑盒归约，这一思想本身或许也具有独立意义。随后，我们反复应用这一归约，将现有递减算法引导为全动态解决方案。

1 引言

全点对最短路径问题（APSP）是组合优化中的一项核心图问题。它有着广泛的应用，例如路线规划、导航系统和网络路由，并且从实践和理论两个角度都得到了广泛研究。在理论计算机科学中，这一问题因其对基础算法工具和定义发展的历史性贡献，以及作为解决其他问题的子程序而广受关注。

全点对最短路径（APSP）问题也已在动态设置下得到广泛研究。在这种情况下，底层图会发生边插入与删除（称为边更新），目标是快速报告任意源点—目标点顶点对之间最短路径的近似值。对于 APSP 问题的某些应用而言，动态设置或许更加贴近现实，例如在导航系统中，道路网络的交通状况会不断变化，因此路网连接统计也容易发生改变。一种朴素（但成本相当高）的处理更新方案，是在每次更新后运行一次精确的静态算法。然而，从直觉上说，人们会

*萨尔茨堡大学计算机科学系，奥地利萨尔茨堡。本研究由奥地利科学基金（FWF）资助，项目编号：P 32863-N。本项目获得了欧洲研究理事会（ERC）在欧盟地平线2020研究与创新计划下的资助（资助协议编号 947702）。

†奥地利维也纳维也纳大学计算机科学系

- 设 F 为阶段开始时 G 中存在的边集，E 为 G中当前的边集，D 为阶段内从 G 中删除的边集。

- 设 $I = E \setminus (F \setminus D)$ 为自阶段开始以来插入 G、且在该阶段内之后未被删除的边集；$U = \{v \in V \mid \exists e \in I\colon\ v \in e\}$ 为 I中边的端点集合。

- 令 H为辅助图，它由所有边 $(u, v) \in I$及其枢纽集合$S(t)(u)$ 和 $S(t)(v)$ 组成；此时，A 已处理完 t次边删除。具体而言，$V(H) = \{v \in V \mid v \in U$或 $(u \in U$和 $v \in S(t)(u))\}$，且 $E(H) = \{(u, v) \mid (u, v) \in I$或 $(v \in U$和 $u \in S(t)(v))\}$。注意，在任意固定时刻，$V(H)$ 的大小至多为 $\ell \cdot (2 + 2\gamma)$，而 $E(H)$ 的大小至多为 $\ell \cdot (1 + 2\gamma)$。

在之后每个阶段开始时，C 存储 F，E和 H，并将更新计数器设为 0。此外，C 在当前图 G上初始化递减近似枢纽标记方案 A，并在 H 上初始化全动态距离预言机 B；此时 H 是一个初始为空、包含 $\ell \cdot (2 + 2\mu)$ 个顶点的“草图”图。图 H可视为负责维护使用插入边的路径的估计值。

每当发生对 G 的更新时，C 首先通过更新计数器检查自阶段开始以来的更新次数是否超过 ℓ。如果超过，C 就开始一个新的

[2]这一细微的技术细节确保 B 无需处理顶点插入。[3]对最初 ℓ 次更新的特殊处理，只是为了在运行时间分析中进行严谨的摊还分析论证所必需的技术措施。

最后，对 H 所做的所有更改都会由运行在 H 上的全动态距离预言机 B 处理，其中边权变化通过先删除再重新插入来模拟。

现在，对于任意一对顶点 s 和 t 之间的近似距离查询，返回以下结果：

$$\delta_C(s,t) = \min\left(\min_{p\in S(s)\cap V(H), q\in S(t)\cap V(H)} \left(\delta(s,p) + \delta_{\mathcal{B}}(p,q) + \delta(t,q)\right), \delta_{\mathcal{A}}(s,t)\right).$$

每当 $S(s) \cap V(H) = \emptyset$ 或 $S(t) \cap V(H) = \emptyset$ 时，我们令内部项 $\min(\cdot) = \infty$。

[4]注意，t 是该阶段中仅由 A 处理的更新次数。

正确性。为证明该算法的正确性，我们需要证明 $\mathrm{dist}_G(s,t) \le \delta_C(s,t) \le \alpha\beta \cdot \mathrm{dist}_G(s,t)$。下界 $\mathrm{dist}_G(s,t) \le \delta_C(s,t)$ 显然成立，因为对于 C 返回的每个近似距离，相应路径使用的边来自 G，或者使用递减近似枢纽标记方案给出的距离估计，而这些估计绝不会低估实际距离。为证明上界，考虑图 G 中从 s 到 t 的一条最短路径 π，并令 G_A 表示由 A 维护的图（即，$G_{\mathcal{A}}$ 的边集为 $E(G_{\mathcal{A}}) = F \setminus D$）。如果路径 π 仅包含集合 $F \setminus D$ 中的边，那么 $\delta_C(s,t) \le \delta_{\mathcal{A}}(s,t) \le \alpha \cdot \mathrm{dist}_{G_{\mathcal{A}}}(s,t) = \alpha \cdot \mathrm{dist}_G(s,t)$，结论即得。否则，令 $(u_1,v_1)\ldots,(u_j,v_j) \in I$ 表示 π 中自当前阶段开始以来插入的边，按它们在 π 上出现的顺序排列。此外，令 $p_0 \in S(s) \cap S(u_1)$ 表示“证明” $\delta_{\mathcal{A}}(s,u_1)$ 的顶点，即 $\delta_{\mathcal{A}}(s,u_1) = \delta(s,p_0) + \delta(u_1,p_0)$。类似地，令 $p_j \in S(v_j) \cap S(t)$ 表示“证明” $\delta_A(v_j,t)$ 的顶点，并且对每个 $1 \le i \le j-1$，令 $p_i \in S(v_i) \cap S(u_{i+1})$ 表示“证明” $\delta_{\mathcal{A}}(v_i,u_{i+1})$ 的顶点。根据近似枢纽标签方案的定义，这些顶点必定存在。此外，根据 H 的构造，边 (p_0,u_1) 和 (p_j,t) 已插入 H，因为 $u_1 \in U$ 和 $p_0 \in S(u_1)$，且 $v_j \in U$ 和 $p_j \in S(v_j)$ 分别成立。因此，顶点 p_0 和 p_j 属于 $V(H)$，并且和 $\delta(s,p_0) + \delta_{\mathcal{B}}(p_0,p_j) + \delta(t,p_j)$ 参与内部项 $\min(\cdot)$。因此，为分析所述伸展因子上界，我们按如下方式进行：

$$
\begin{aligned}
\delta_C(s,t) &\le \delta(s,p_0) + \delta_{\mathcal{B}}(p_0,p_j) + \delta(t,p_j) \\
&\quad \text{(stretch guarantee of } \mathcal{B}) \\
&\le \delta(s,p_0) + \beta \cdot \mathrm{dist}_H(p_0,p_j) + \delta(t,p_j) \\
&\quad \text{(triangle inequality)} \\
&\le \delta(s,p_0) + \beta \cdot \mathrm{dist}_H(p_0,u_1) + \sum_{1 \le i \le j-1} \beta \cdot (\mathrm{dist}_H(u_i,v_i) + \mathrm{dist}_H(v_i,p_i) + \mathrm{dist}_H(p_i,u_{i+1})) \\
&\quad + \beta \cdot \big(\mathrm{dist}_H(u_j,v_j) + \mathrm{dist}_H(v_j,p_j)\big) + \delta(t,p_j) \\
&\quad (\mathrm{dist}_H \le w_H) \\
&\le \delta(s,p_0) + \beta \cdot w_H(p_0,u_1) + \sum_{1 \le i \le j-1} \beta \cdot (w_H(u_i,v_i) + w_H(v_i,p_i) + w_H(p_i,u_{i+1})) \\
&\quad + \beta \cdot \big(w_H(u_j,v_j) + w_H(v_j,p_j)\big) + \delta(t,p_j)
\end{aligned}
$$

根据 H 的构造，π 的边 (u_i,v_i) 以及对应的边 (p_{i-1},u_i) 和 (v_i,p_i) 已被加入 H，因为 $(u_i,v_i) \in I$，$u_i \in U$和 $p_{i-1} \in S(u_i)$，而 $v_i \in U$ 和 $p_i \in S(v_i)$ 分别成立。因此，根据 $w_H(\cdot)$ 的定义，我们可以用 $w_G(u_i,v_i)$ 替换 $w_H(u_i,v_i)$，用 $\delta(u_i,p_{i-1})$ 替换 $w_H(p_{i-1},u_i)$，并用 $\delta(v_i,p_i)$ 替换 $w_H(v_i,p_i)$。因此，我们有：

$$
\begin{aligned}
\delta_C(s,t) &\le \delta(s,p_0) + \beta \cdot \delta(u_1,p_0) + \sum_{1 \le i \le j-1} \beta \cdot (w_G(u_i,v_i) + \delta(v_i,p_i) + \delta(u_{i+1},p_i)) \\
&\quad + \beta \cdot \big(w_G(u_j,v_j) + \delta(v_j,p_j)\big) + \delta(t,p_j) \\
&\quad (\pi \text{ is a shortest path}) \\
&= \delta(s,p_0) + \beta \cdot \delta(u_1,p_0) + \sum_{1 \le i \le j-1} \beta \cdot (\mathrm{dist}_G(u_i,v_i) + \delta(v_i,p_i) + \delta(u_{i+1},p_i)) \\
&\quad + \beta \cdot \big(\mathrm{dist}_G(u_j,v_j) + \delta(v_j,p_j)\big) + \delta(t,p_j) \\
&\quad (\beta \ge 1)
\end{aligned}
$$

[5]我们可以假设，只要 $\delta(\cdot,\cdot)$ 有有限值，它就不超过 nW，因为 G 中的最大距离至多为 nW。同样，我们可以使用值 $nW+1$ 来代替 ∞。[6]我们乘以 2，因为边权变化是通过先删除再重新插入来模拟的。

$$t_{\mathcal{B}_{i+1}}(n, m, W) = \frac{T_{\mathcal{A}}(n, m, W)}{\ell} + t_{\mathcal{B}_i}(n, \ell(1 + 2\mu), nW) \cdot (2 + 4\mu)$$