arXiv:2304.02315v1 [cs.DS] 5 Apr 2023

T确定性拥塞团中的拉普拉斯范式

Sebastian Forster* Tijn de Vos*

摘要

在本文中，我们将拉普拉斯范式的技术引入拥塞团，同时进一步将自己限制于确定性算法。具体而言，我们展示如何以精度 ϵ 在 $n^{o(1)}$ 对数$(1/\epsilon)$轮次内求解一个拉普拉斯系统。我们展示如何在现有的用于求解流问题的内点法中利用这一结果。我们得到一个用于最大权重为 U的加权有向图上最大流的 $m^{3/7+o_{(1)}}U^{1/7}$ 轮次算法，并且我们得到一个用于最大费用W的有向图上单位容量最小费用流的 $\tilde{O}(m^{3/7}($ $n^{0.158}+n^{o(1)}$多项式对数$W))$轮次算法。为此，我们给出一种在 $O($对数 n对数* $n)$轮次内计算欧拉定向的新颖例程，我们认为其可能具有独立的意义。

*计算机科学系，萨尔茨堡大学，奥地利。本研究得到奥地利科学基金会（FWF）资助：P32863-N。本项目获得了欧洲研究委员会（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）。

对于确定性设置，据我们所知，只有两种算法可供我们比较。首先，有一个平凡的 $O(n \log U)$ 轮次算法，它会使所有知识全局共享，从而使每个节点都能在本地求解该问题。

定理 2.2（预条件切比雪夫迭代）。 存在一个算法 *PreconCheby*，使得对于任意对称正半定矩阵 A、B 和 κ，其中

$$A \preccurlyeq B \preccurlyeq \kappa A$$

任何误差容限 $0 \leq \epsilon \leq 1/2$，*PreconCheby*(A, B, b, κ) 在精确算术模型中是一个对称线性算子，其满足：

1. 如果 Z 是实现此算子的矩阵，则 $(1-\epsilon)A^{\dagger} \preccurlyeq Z \preccurlyeq (1+\epsilon)A^{\dagger}$；

*2.*对于任意向量 b，*PreconCheby*(A, B, b, ϵ)需要 $O(\sqrt{\kappa}$对数$(1/\epsilon))$次迭代，每次迭代包括关于 A的一次矩阵-向量乘法、涉及 B 的一次求解，以及常数次向量运算。

这得出了以下关于使用谱稀疏化器进行拉普拉斯方程求解的推论。

推论 2.3. 设 G 是一个具有 n 个顶点的加权图，$\epsilon \in [0, 1/2]$ 是一个参数，且 $b \in \mathbb{R}^n$ 是一个向量。假设 H 是 G的一个 α-近似谱稀疏化器，其中 $\alpha \geq 1$满足一定条件。则存在一个算法 *LaplacianSolve*(G, H, b, ϵ)，输出一个向量 $y \in \mathbb{R}^n$，使得 $||x - y||_{L_G} \leq \epsilon ||x||_{L_G}$，其中 $x \in R^n$满足 $L_G x = b$。*LaplacianSolve*(G, H, b, ϵ)需要 $O(\sqrt{\alpha}$对数$(1/\epsilon))$次迭代，每次迭代包括一次与 L_G的矩阵-向量乘法、一次涉及 L_H的求解，以及常数次向量运算。

接下来，我们使用该定理在拥塞团中计算谱稀疏化器。

定理 3.3。存在一种确定性算法，给定带权重 $w: E \to \{1, 2, \ldots, U\}$ 的无向图 $G = (V, E)$，以及参数 $1 \le r \le \sqrt{\log n / \log\log n}$，该算法可在拥塞团中用 $O(\log n \log U n^{O(1/r^2)})$ 轮次计算 G 的 $\log(n)^{O(r^2)}$-近似谱稀疏化器 H，其 $|E(H)| \le O(n \log n \log U)$。算法结束时，每个节点都知道 H。

[3]对于 $U \subseteq V$，我们将 $G[U]$ 记作由 U诱导的诱导子图，即 $G[U] := (U, E \cap (U \times U))$。

两种算法的运行时间均为 $O(T\text{对数}(1/\Delta))$。

在拥塞团中，计算欧拉定向有如下结果 .

定理 1.4. 存在一种确定性拥塞团算法，给定一个欧拉图$G=(V,E)$，可在 $O(\text{对数 } n\,\text{对数}^* n)$ 轮次内求得一个欧拉定向。

证明。 该算法由以下步骤组成。

1. 由于该图是欧拉图，每个节点都有偶数个邻居。每个节点在内部将其邻居两两配对，从而形成环。这意味着我们已经获得了一个环分解，其中每个节点仅掌握局部知识——这显而易见，因为迄今为止尚未进行任何通信。对于每个节点所参与的环，它都会跟踪一个状态（活跃/不活跃），初始状态设为活跃。

引理 4.2. 设 $G=(V,E)$ 是一个有 n 个顶点和 m 条边的图。设 $f\colon E\to\mathbb{R}_{\geq 0}$ 是一个 s-t 流函数， Δ 是一个实数，且满足 $1/\Delta$ 是 2 的幂，并且 $f(e)$ 是一个整数倍

算法 1: 流舍入(G, s, t, f, c, Δ)

输入： 图 $G = (V, E)$; 源点 $s \in V$和汇点 $t \in V$; ow 函数 $f\colon E \to \mathbb{R}_{\geq 0}$;
值 $\Delta \in \mathbb{R}_{\geq 0}$，使得$1/\Delta$ 是 2 的幂，并且 $f(e)$ 是.....的整数倍
Δ 对于每个 $e \in E$; （可选）成本函数 $c\colon E \to \mathbb{Z}_{\geq 0}$;

如果 f 的总流量不是整数，**那么**
    添加一条从 t 到 s 的边，其流值与总流值相同。
当 $\Delta < 1$**时**
    $E' \leftarrow \{(u, v) \in E\colon f(u, v)/\Delta$ 是 $odd\}$
    求 E' 的欧拉分解（忽略边的方向）
    对于 E' **的欧拉分解中的每个环，都**
        如果 环包含边 (t, s) **则**
            遍历该环，使得边 (t, s) 是一条前向边。
        否则如果 成本函数 c 存在 **则**
            遍历该环，使得正向边上的费用总和不超过
              比反向边上的费用总和。
        else
            任意遍历该环。
    对于每条边 (u, v) E' **执行**
        如果(u, v) 是相对于包含(u, v) 的路径遍历的前向边**那么**
            $f(u, v) \leftarrow f(u, v) + \Delta$
        else
            $f(u, v) \leftarrow f(u, v) - \Delta$
    $\Delta \leftarrow 2\Delta$
返回 f.

算法 2： MaxFlow(G_0, s, t, U, F)

输入： 有向图 $G_0 = (V, E_0, \vec{u})$，其中每个 $e \in E_0$ 具有两个非负整数容量 u_e^- 和 u_e^+；$|V| = n$ 和 $|E_0| = m$；源点 s 和汇点 t；最大整数容量 U；目标流值 $F \geq 0$；

/* 预处理边 */

添加 m 条无向边（t，s），其正向和反向容量为 $2U$ 至 G_0

/* 初始化 */

对于每个 $e =$（u，v）$\in E_0$ **执行**

将 e 替换为三条无向边（u，v）、（s，v）和（u，t），其容量为 u_e

令新图为 $G = (V, E)$

初始化 $\vec{f} \leftarrow \vec{0}$ 和 $\vec{y} \leftarrow \vec{0}$

/* *Progress* 步骤 */

$\vec{\tilde{f}}, \vec{\hat{f}}, \vec{\hat{y}} \leftarrow$ 增广(G, s, t, F)

通过令 $\rho_e \leftarrow$ 计算 $\frac{\tilde{f}_e}{\min\{u_e^+ - f_e, u_e^- + f_e\}}$

$\vec{f}, \vec{y} \leftarrow$ 修复 $\left(G, \vec{\tilde{f}}, \vec{\hat{y}}\right)$

$\eta \leftarrow \frac{1}{1} - \frac{1}{7}$ 对数 $_m U - O($对数 $_m$ 对数(mU), $\hat{\delta} \leftarrow \frac{1}{m^{\frac{1}{2}-\eta}}$

对于 $t = 1$ **到** $100 \cdot \frac{1}{\hat{\delta}} \cdot$ 对数 U **执行**

如果 $\|\vec{\rho}\|_3 \leq \frac{m^{\frac{1}{2}-\eta}}{33(1-\alpha)}$ **则**

$\delta \leftarrow \frac{1}{33(1-\alpha)\|\vec{\rho}\|_3}$

$\vec{\tilde{f}}, \vec{\hat{f}}, \vec{\hat{y}} \leftarrow$ 增广(G, s, t, F)

计算 $\vec{\rho}$，通过计 $\rho_e \leftarrow \frac{\tilde{f}_e}{\min\{u_e^+ - f_e, u_e^- + f_e\}}$

$\vec{f}, \vec{y} \leftarrow$ 修复 $\left(G, \vec{\tilde{f}}, \vec{\hat{y}}\right)$

else

令 S^* 为边集，包含 $m^{4\eta}$ 条具有最大 $|\rho_e|$ 的边

$G \leftarrow$ Boosting $\left(G, S^*, U, \vec{f}, \vec{y}\right)$

$\vec{f} \leftarrow$ 流舍入$(G, \vec{f}, s, t)$

当 存在一条从 s 到 t 的增广路径，关于 $\vec{f}$ 为 G **做**

为其增广一条增广路径 $\vec{f}$

算法 3： 增广(G, s, t, F)

/* 增广步骤 */

1 对于每个 $e \in E$, 令 $r_e \leftarrow \frac{1}{(u_e^+ - f_e)^2} + \frac{1}{(u_e^- + f_e)^2}$ 和 $w_e \leftarrow \frac{1}{r_e}$

2 求解拉普拉斯线性方程组$\boldsymbol{L}(G)$ $\widetilde{\vec{\phi}} = F \cdot \vec{\chi}_{s,t}$ 其中$\vec{\chi}_{s,t}$是其条目为的向量 -1（分别为 1）在顶点 s（分别为 t）处，其余为 0

3 对于每个 $e = (u, v) \in E$, 令 $f_e \leftarrow \frac{\widetilde{\phi}_v - \widetilde{\phi}_u}{r_e}$ $\widehat{f_e \leftarrow f_e + \delta f_e}$ 和

4 对于每个 $v \in V$, 令 $\widehat{y}_v \leftarrow y_v + \delta \widetilde{\phi}_v$

5 返回 $\vec{\widetilde{f}}, \vec{\widehat{f}}, \vec{\widetilde{y}}$

~~算法 4:~~ ~~修复~~$\left(G, \vec{\widehat{f}}, \vec{\widetilde{y}}\right)$

/* 修正步骤 */

1 对于每个 $e = (u, v) \in E$, 令 $r_e \leftarrow \frac{1}{\left(u_e^+ - \widehat{f_e}\right)^2} + \frac{1}{\left(u_e^- + \widehat{f_e}\right)^2}$, $w_e \leftarrow \frac{1}{r_e}$ and $\theta_e \leftarrow w_e \left[\left(\widehat{y}_v - \widehat{y}_u\right) - \left(\frac{1}{u_e^+ - \widehat{f_e}} - \frac{1}{u_e^- - \widehat{f_e}} \right) \right]$

2 $\vec{f}' \leftarrow \vec{\widehat{f}} + \vec{\theta}$

3 设 $\vec{\widetilde{\delta}}$ be $\vec{\theta}$的残差向量

4 对于每个 $e \in E$, 令 $r_e \leftarrow \frac{1}{(u_e^+ - f_e')^2} + \frac{1}{(u_e^- + f_e')^2}$ 和 $w_e \leftarrow \frac{1}{r_e}$

5 求解拉普拉斯线性方程组$\boldsymbol{L}(G)$ $\vec{\phi}' = -\vec{\widetilde{\delta}}$

6 对于每个 $e = (u, v) \in E$, 令 $\theta_e' \leftarrow \frac{\phi_v' - \phi_u'}{r_e}$

7 对于每个 $e \in E$, $f_e \leftarrow f_e' + \theta_e'$

8 对于每个顶点 $v \in V$, $y_v \leftarrow \widehat{y}_v + \phi_v'$

9 返回 $\vec{f}, \vec{y}$

算法 6： 最小费用流$(G, \vec{\sigma}, W)$

输入： 有向图 $G_0 = (V_0, E_0, \vec{c}_0)$，其中每条边的容量为单位容量，费用为 $\vec{c}_0$；$|V_0| = n$ 和 $|E_0| = m$；整数需求向量 $\vec{\sigma}$；绝对最大费用 W；

$G = (P \cup Q, E)$, $\vec{b}$, $\vec{f}, \vec{y}, \vec{s}, \vec{v}, \vec{\mu}$, $c_\rho, c_T, \eta \leftarrow$ 初始化$(G_0, \vec{\sigma})$

添加一个新的顶点 v_0 和无向边 (v_0, v)，对于每个 $v \in P$ 到 G

对 $i = 1$ **至** $c_T \cdot m^{1/2-3\eta}$ **执行**

  对于每个 $v \in P$ **执行**

    将边 (v_0, v) 对每个 $v \in P$ 的电阻设为 $r_{v_0 v} \leftarrow$ $\frac{m^{1+2\eta}}{a\ v\ ()}$，其中 $a(v) \leftarrow \sum_{u \in Q, e=(v,u) \in E} \nu_e + \nu_e$; /*$\bar{e} = (\bar{v}, u)$是 e 的伙伴边，即来自 Q的共享一个公共顶点的唯一边。 */

  对于 $j = 1$ **到** $m^{2\eta}$ **执行**

    当$\|\vec{\rho}\|_{\vec{v},3} > c_\rho \cdot m^{1/2-\eta}$**时执行**

      $\vec{\rho}, \vec{y}, \vec{s}, \vec{v} \leftarrow$ Perturbation$(G, \vec{\rho}, \vec{f}, \vec{y}, \vec{s}, \vec{v})$

    $\vec{f}, \vec{s}, \vec{\rho}, \widehat{\mu} \leftarrow$ Progress$(G, \vec{\sigma}, \vec{f}, \vec{v})$

Repairing$(G, \vec{f}, \vec{y})$

算法 7： 初始化$(G, \vec{\sigma})$

创建一个新顶点 v_{aux}，使用 $\sigma(v_{aux}) = 0$

对每个 $v \in V_0$ **执行**

  $t(v) \leftarrow \sigma(v) + \frac{1}{2}\deg_{in}^{G_0}(v) - \frac{1}{2}\deg_{out}^{G_0}(v)$

  如果 $t(v) > 0$ **则** 构造 $2t(v)$ 平行边(v, v_{aux})，费用为 $\|\vec{c}_0\|_1$

  否则如果 $t(v) < 0$ **则**

    构造 $|2t(v)|$ 平行边 (v_{aux}, v)，费用为 $\|\vec{c}_0\|_1$

令新图为 $G_1 = (V_1, E_1, \vec{c}_1)$

初始化二分图 $G = (P \cup Q, E, \vec{c})$，具有 $E \leftarrow \emptyset, P \leftarrow V_1$ 和 $Q \leftarrow \{e_{uv} \mid (u, v) \in E_1\}$，其中 e_{uv}是与边 $(u, v) \in E_1$对应的顶点。

对 每个 $(u, v) \in E_1$ **执行**

  令 $E \leftarrow E \cup \{(u, e_{uv}), (v, e_{uv})\}$，其中 $c(u, e_{uv}) = c_1(u, v)$ 和 $c(v, e_{uv}) = 0$，并设 $b(u) \leftarrow \sigma(u) + 度_{in}^{G_1}(u)$，$b(v) \leftarrow \sigma(v) + 度_{in}^{G_1}(v)$ 以及 $b(e_{uv}) \leftarrow 1$

对于每个 $v \in P$，设置 $y_v \leftarrow \|\vec{c}\|_\infty$；对于每个 $v \notin P$，设置 $y_v \leftarrow 0$

对于每个 $e = (u, v) \in E$，设置 $f_e \leftarrow \frac{1}{2}$、$s_e \leftarrow c_e + y_u - y_v$和 $v_e \leftarrow \frac{s_e}{2\|\vec{c}\|_\infty}$

设置 $\widehat{\mu} \leftarrow \|\vec{c}\|_\infty$，$c_\rho \leftarrow 400\sqrt{3}\cdot$对数$^{1/3} W$，$c_T \leftarrow 3c_\rho$对数$W$ 并且 $\eta \leftarrow \frac{1}{14}$

返回 G，$\vec{b}, \vec{f}, \vec{y}, \vec{s}, \vec{v}, \widehat{\mu}, c_\rho, c_T$和 η

算法 8: PERTURBATION$(G, \vec{\rho}, \vec{f}, \vec{y}, \vec{s}, \vec{v})$

对于每个 $v \in Q$ **执行** 令 $e = (u, v)$ 并 $e = (u, v)$

  $y_v \leftarrow y_v - s_e$　　$v_e \leftarrow 2v_e$

  $v_e \leftarrow v_e + \frac{v_e f_e}{f_e}$

算法 9:PROGRESS$(G, \vec{\sigma}, \vec{f}, \vec{v})$

对于每个 $e \in E$，令 $r_e \leftarrow \frac{v_e}{f_e^2}$

求解拉普拉斯线性方程组$\boldsymbol{L}(G)$ $\vec{\widehat{\phi}} = \vec{\sigma}$

对于每个 $e = (u, v) \in E$，令 $f_e \leftarrow \frac{\widehat{\phi}_v - \widehat{\phi}_u}{r_e}$ 并且 $\rho_e \leftarrow \frac{|\widehat{f_e}|}{f_e}$

$\delta \leftarrow \min\left\{\frac{1}{8\|\vec{\rho}\|_{\vec{v}}^4}, \frac{1}{8}\right\}$

更新 $f_e' \leftarrow (1 - \delta)f_e + \delta \widehat{f_e}$和 $s_e' \leftarrow s_e - \frac{\delta}{1-\delta}(\widehat{\phi}_v - \widehat{\phi}_u)$

对于每个 $e \in E$，令 $f_e^\# \leftarrow \frac{(1-\delta) f_e s_e}{s_e'}$

获取 ow 向量 $\vec{\sigma}'$ 对应于残差 $\vec{f}' - \vec{f}^\#$

对于每个 $e \in E$，令 $r_e \leftarrow \frac{s_e'^2}{(1-\delta) f_e s_e}$

求解拉普拉斯线性方程组$\boldsymbol{L}(G)$ $\vec{\widetilde{\phi}} = \vec{\sigma}'$

对于每个 $e = (u, v) \in E$，令 $f_e \leftarrow \frac{\widetilde{\phi}_v - \widetilde{\phi}_u}{r_e}$

更新 $f_e \leftarrow f_e^\# + \widetilde{f_e}$和 $s_e \leftarrow s_e' - \frac{s_e'}{f_e^\# s_e}\widetilde{}$

算法 10： REPAIRING(G, $\vec{f}, \vec{y}$)

设 $\vec{b}^+$ 为对应于当前流的需求向量 $\vec{f}$

对于每个 $v \in P \cup Q$，将 $b_v^{\leq}$ ← 设为 $\min(b_v, b_v^+)$

对于每个 $v \in P \cup Q$，若 $f(E(v)) > b_v^{\leq}$，则设 $\vec{f}$ 在 $E(v)$ 上，使得 $f(E(v)) = b_v^{\leq}$，并令该所得向量为 $\vec{f}^{\leq}$

添加源点 s 并将汇点 t 添加到 G，并将 s 与每个 $v \in P$ 通过 $f_{sv}^{\leq} \leftarrow f^{\leq}(E(v))$ 连接，并将每个 $v \in Q$ 连接到 t，使用 $f_{vt}^{\leq} \leftarrow f^{\leq}(E(v))$，在 G 中

$\vec{M} \leftarrow$ 流舍入(G, $\vec{f}^{\leq}, s, t$)

移除 s，t 以及相关坐标 乐彩 $\vec{f}^{\leq}$ 和 $\vec{M}$ 来自 G

对于 $i = 1$ 到 $\widetilde{O}(m^{3/7})$ **执行**

- 构建图 $\overrightarrow{G}_M = (P \cup Q, E_M, \widetilde{c}_M)$ 使用 G, M 和 $\vec{\widetilde{c}}$ 使得对于每个 $e = (u, v) \in E$, $\widetilde{c}_e = c_e - y_u - y_v$ 和
  $E_M = \{(u, v) \in E \mid u \in P, v \in Q\} \cup \{(u, v) \mid u \in Q, v \in P, M_{uv} \neq 0\}$,
  $$\widetilde{c}_M(u, v) = \begin{cases} \widetilde{c}_{uv}, & u \in P, v \in Q \\ -\widetilde{c}_{uv}, & u \in Q, v \in P \end{cases}$$
- 集合 $F_M \leftarrow \{v \in P \cup Q \mid M(v) < b_v\}$
- 计算一条最短路径 π 在 $\overrightarrow{G}_M$ 从 $P \cap F_M$ 到 $Q \cap F_M$
  /* $\mathcal{D}_{\overrightarrow{G}_M}(P, u)$ 是从 P 到 u 的距离，位于 $\overrightarrow{G}_M$ */
  /* 可到达的边 $\overrightarrow{G}_M$ 从 $P \cap F_M$ 具有非负权重 $\widetilde{c}_e$ */
  对于 $u \in P \cup Q$ **执行此操作**
  - **如果** u 可以从 P 到达在 $\overrightarrow{G}_M$ **然后**
    - **如果** $u \in P$ **则**
      - $y_u \leftarrow y_u - \mathrm{D}_{\overrightarrow{G}_M}(P, u)$
    - **else**
      - $y_u \leftarrow y_u + \mathrm{D}_{\overrightarrow{G}_M}(P, u)$
- 增广 $\vec{M}$ 使用增广路径 π

返回 $\vec{M}$