通过细粒度交叉组合得到 P 中问题的参数化下界

KlausHeeger ！ 柏林工业大学，算法学与计算复杂性，德国

André Nichterlein
柏林工业大学，算法学与计算复杂性，德国

Rolf Niedermeier
柏林工业大学，算法学与计算复杂性，德国

arXiv:2301.00797v2 [cs.DS] 6 Jan 2023

摘要

我们提出了一个通用框架，用于排除满足细粒度复杂性假设下具有多项式运行时间下界的问题，其参数化运行时间为 $O(\ell^\beta + n^\gamma)$ 的形式。我们的框架基于参数化复杂性中的交叉组合。我们有条件地排除以下问题的运行时间为 $O(\ell^{\gamma/(\gamma-1)-\varepsilon} + n^\gamma)$ 的形式，其中任意 $1 < \gamma < 2$ 和 $\varepsilon > 0$ 均成立：

- 最长公共（递增）子序列：给定两个由字母表 Σ（或 N）和 $\ell \in$ N 上的字符构成、长度为 n 的字符串，是否存在一个长度为 ℓ 的公共（递增）子序列，使其同时出现在两个字符串中？离散弗雷歇距离：给定两个列表，每个列表含有 n 个点，且 $k \in N$，这两个列表的弗雷歇距离是否至多为 k？此处，ℓ 是在一次最优遍历中，一个列表领先于另一个列表的点数最大值。平面运动规划：给定平面内一组 n 个互不相交的轴平行线段障碍物，以及一个线段机器人（称为杆），能否在不接触任何障碍物的情况下，将杆移动到指定目标？此处，ℓ 是任一线段邻域内的线段最大数量。

此外，对于以下问题，我们排除运行时间为 $O(\ell^{2\gamma/(\gamma-1)-\varepsilon} + n^\gamma)$ 的算法，其中任意 $1 < \gamma < 3$ 和 $\varepsilon > 0$：

- 负三角形：给定一个有 n 个顶点的边加权图，是否存在一个边权之和为负的三角形？这里，ℓ 是最大连通分量的阶数。三角形集合：给定一个有 n 个顶点的顶点着色图，对于每组三种颜色，是否都存在一个顶点分别具有这三种颜色的三角形？这里，ℓ 是最大连通分量的阶数。第二短路径：给定一个有 n 个顶点的边加权有向图、两个顶点 s 和 t，以及 $k \in$N，第二长的 s-t 路径长度是否至多为 k？这里，ℓ 是有向反馈顶点集数。

除了第二短路径之外，所有这些运行时间界都是紧的，也就是说，已知存在运行时间分别为 $O(\ell^{\gamma/(\gamma-1)} + n^\gamma)$（适用于任意 $1 < \gamma < 2$）和 $O(\ell^{2\gamma/(\gamma-1)} + n^\gamma)$（适用于任意 $1 < \gamma < 3$）的算法。我们的运行时间下界也意味着这些问题的核化算法存在下界。

2012年 ACM 学科分类计算理论 → 图算法分析；计算理论 → 参数化复杂性与精确算法

关键词与短语P 中的固定参数可处理性、核化、分解

经费支持*Klaus Heeger*：由德国研究联合会项目 NI 369/16 “FPTinP”资助。

致谢 谨以此文缅怀我们的同事、朋友和导师 Rolf Niedermeier。他不幸在本文完成之前离世。感谢匿名评审提出深思熟虑且富有建设性的意见。

1 引言

近年来，细粒度复杂性领域的许多研究结果表明，许多已有数十年历史、用于求解多项式时间可解问题的教科书算法实际上已是最优的。以最长公共子序列（LCS）为例：给定两个各含 n 个字符的输入字符串，任务是找出一个同时作为子序列出现在这两个输入字符串中的最长字符串。经典的 $O(n^2)$ 时间算法常在动态规划入门课程中讲授 [18]。Bringmann 和 Künnemann [14]以及 Abboud 等人 [1] 分别独立证明，对于任意 $\varepsilon > 0$，若有算法能在 $O(n^{2-\varepsilon})$ 时间内求解 LCS，就会推翻强指数时间假设（SETH）。近年来，许多多项式时间可解问题都已被证明存在此类条件下界 [51]。

规避此类下界的一种方法是“P 中的固定参数可处理性” [2, 34]。对于最长公共子序列，已有一种（相当古老的）参数化算法，其运行时间为 $O(kn + n\log n)$，其中 k 是最长公共子序列的长度 [39]。因此，如果 k 很小（例如 $O(n^{0.99})$），就能打破 $O(n^2)$障碍（而无需推翻 SETH）。一个自然的问题是，我们能否做得更好。正如 $k \leq n$所示，对于任意 $\varepsilon > 0$，运行时间为 $O(k^{1-\varepsilon}n)$的算法都会推翻 SETH。然而，没有明显的论据能够排除运行时间为 $O(k^2+n)$的情况。事实上，这类加法式运行时间不仅值得期待（同样，因为当 k 较小时，其速度甚至快于 $O(kn)$），在参数化算法学中也相当常见，这可以通过核化实现：对于最长公共子序列，问题在于是否存在可在线性时间内应用的数据缩减规则，能将输入缩减至大小为 $O(k)$。这样，我们只需应用教科书中的算法，就能在总运行时间为 $O(k^2 + n)$的情况下求解最长公共子序列。核化在参数化复杂性社区中得到了深入研究 [5, 31]，并且在最大匹配 [42]或最小割 [38]等多项式时间可解问题中也很有效。

Bringmann 和 Künnemann [15] 通过一项深入研究表明，这样一种 $O(k^2 + n)$ 时间算法（实际上，许多其他针对最长公共子序列的参数化算法也如此）将推翻 SETH。这也意味着，上文提到的这类核化算法很可能并不存在。Bringmann 和 Künnemann [15] 的结果基于精心设计的归约。

在这项工作中，我们采用不同的途径，为若干问题获得类似结果。我们提出了一个易于应用的通用框架，用于（有条件地）排除那些能够获得条件运行时间下界的问题存在运行时间为 $O(k^\beta + n^\gamma)$ 的算法。事实上，我们针对各种字符串问题（包括最长公共子序列问题）、图问题以及计算几何问题，证明了 β 与 γ之间存在紧确的权衡。这表明，平凡的权衡往往已是所能期望的最佳结果。

1.1 相关工作

细粒度复杂性是一个活跃的研究领域，已有数百篇相关论文。关于研究结果和所采用假设的概述，请参阅 Vassilevska Williams [51] 的综述。

过去几年里，许多研究致力于探索各种问题的“P 中的固定参数可处理性”，例如最大匹配 [19, 22, 30, 37, 40, 42, 44, 47],双曲性 [19, 29], 和直径 [2, 19]。这方面的参数化下界并不多见。某些线性时间归约可以用来排除任何有意义的固定参数可处理运行时间；这也称为一般问题困难性 [9]。通过使用各种精心设计的归约，Bringmann 和 Künnemann [15] 证明了最长公共子序列在七种不同参数下的参数化运行时间下界（假设 SETH 成立）。类似地，Duraj 等人 [24] 证明，求解最长

公共最长递增子序列所需时间为 $O((n\ell)^{1-\epsilon})$，其中 ℓ 是某个 $\epsilon > 0$ 的解大小，这将推翻 SETH。

Fluschnik 等人 [28] 针对子图检测问题（例如负权三角形和三角形集合），给出了严格核化（即不允许参数增大的核）的下界。从概念上说，他们采用了*diminisher*-框架 [17, 27]，该框架最初被开发出来，旨在基于假设 P $\neq$ NP 排除多项式大小的严格核。基本思路是针对一个 NP 困难问题的实例 I，迭代地应用一个缩减器（其算法通过增大实例规模来降低参数）和一个（假设存在的）严格核（用于缩减并控制实例规模）。经过多项迭代后，这个整体的多项式时间算法将返回一个与 I等价的常数规模实例，从而得出 P $=$ NP。Fluschnik 等人[28] 将相同的思路应用于多项式时间可解问题。相比之下，我们依赖并调整了 Bodlaender 等人 [11] 提出的组合框架；该框架旨在基于更强的假设 NP $\not\subseteq$ coNP / poly 排除（一般的）多项式大小核。

组合框架的运作方式如下。以 NP 困难问题负权团为例：给定一个边加权图 G 和一个整数 k，G 中是否包含一个负权 k团，也就是说，是否存在一个由 k 个顶点组成的团，使得团内各边的权重之和为负。

设 $(G_1, k), (G_2, k) \ldots, (G_t, k)$ 是若干个具有相同 k 的负权团实例。显然，将所有 G_i 做不交并得到的图 G 包含负权 k团，当且仅当某个 G_i 包含负权 k 团。此外，G 的最大连通分量的阶为 $\max_{i\in[t]}\{|V(G_i)|\}$。现在假设负权团具有大小为 $O(\ell^c)$ 的核，其中 c 为某个常数，ℓ 表示最大连通分量的阶。通过选取 $t = k^{c+1}$，可知对实例 (G, k) 进行核化后，得到的实例大小小于 ℓ，也就是说，其比特数少于 G 中编码的实例数。鉴于负权团的 NP 困难性，这种压缩似乎很难实现；事实上，它将意味着 NP $\subseteq$ coNP / poly [32],，进而导致多项式层次坍塌。

组合及其扩展形式——交叉组合 [12] 在参数化复杂性文献中得到了广泛应用。此外，为了排除规模受特定次数多项式约束的核，人们对组合框架 [20]进行了调整。

参数权衡。 对于我们的一些运行时间下界，我们有紧确的上界，这些上界可通过简单的分类论证推导得出。

▶ **观察 1.1** (民间定理)。 如果问题 P 存在一个 $\widetilde{O}(\ell^\beta n^\gamma)$时间算法[1],，那么对于每个 $\lambda > 0$，它都存在一个 $\widetilde{O}(\ell^{\beta+\frac{\gamma\cdot\beta}{\lambda}} + n^{\gamma+\lambda})$时间算法。

证明。 如果 $\ell \leq n^{\frac{\lambda}{\beta}}$，那么 $\widetilde{O}(\ell^\beta n^\gamma)$时间算法的运行时间为 $\widetilde{O}(n^{\gamma+\lambda})$时间。否则 $n \leq \ell^{\frac{\beta}{\lambda}}$。那么 $\widetilde{O}(\ell^\beta n^\gamma)$算法的运行时间为 $\widetilde{O}(\ell^{\beta+\frac{\gamma\cdot\beta}{\lambda}})$时间。 ◀

1.2 我们的结果与技术

我们提出一种基于组合的框架，用于建立参数化运行时间下界，并将该框架应用于最长公共子序列、最长公共（弱）递增子序列、离散弗雷歇距离、平面运动

[1] $\widetilde{O}$ 隐藏了多对数因子。

表1 可实现运行时间概览。表格上半部分列出了四个问题的结果：这些问题可在 $O(n^2)$ 时间内求解，但根据 SETH 或 3SUM 假设，对于任意 $\varepsilon > 0$，都无法在 $O(n^{2-\varepsilon})$ 时间内求解。下半部分列出了三个图问题的结果：根据全点对最短路径假设，这些问题不存在运行时间为 $O(n^{3-\varepsilon})$ 的算法。图1直观展示了参数化上界和下界。

problems	最长公共子序列	ℓ=解大小
	最长公共（弱）递增子序列	ℓ=解大小
	离散弗雷歇距离	ℓ=最大偏移
	平面运动规划	ℓ=任一线段附近的线段最大数量
results	上界	下界
	$O(n^2)$ [6, 7, 35, 53]	假设 SETH / 3SUM，则不存在 $O(n^{2-\varepsilon})$ [1, 13, 14, 33]
	$\widetilde{O}(\ell n)$ [39, 45, 50]	
	对每个 $\gamma > 1$，都有 $O(\ell^{\gamma/(\gamma-1)} + n^\gamma)$	对于任意 $\gamma < 2$ [15, 24]，都不存在 $O(\ell^{\gamma/(\gamma-1)-\varepsilon} + n^\gamma)$
	(观察 1.1)	(推论 4.6、4.11 和 4.13)
problems	负三角形	ℓ=最大分量大小
	三角形集合	ℓ=最大分量大小
	第二短路径（仅有下界）	ℓ=有向反馈顶点集大小
results	上界	下界
	$O(n^3/2^{\Theta(\log^{0.5} n)})$ [16, 54]	不存在 $O(n^{3-\varepsilon})$，假设全点对最短路 [52]
	$\widetilde{O}(\ell^2 n)$（民间定理）	
	每个 $\gamma > 1$ 对应一个 $O(\ell^{2\gamma/(\gamma-1)} + n^\gamma)$	不存在任何 $\gamma < 3$ 对应的 $O(\ell^{2\gamma/(\gamma-1)-\varepsilon} + n^\gamma)$
	(观察 1.1)	(推论 4.15 和 4.17 以及命题 4.18)

规划、负三角形和第二短路径（问题定义见第1.3节）。利用类似的思路，我们还为三角形集合问题建立了运行时间下界。对于除以有向反馈顶点集为参数的第二短路径之外的所有这些问题，都存在匹配的运行时间上界。具体结果和参数化方式概述见表1。此外，我们在图1中直观展示了运行时间上可行与不可行的权衡。

框架。 我们调整交叉组合框架，以获得多项式时间可解问题的下界。举例来说，考虑负权三角形问题，即将负权团中的 k 固定为三。假设全点对最短路径假设成立，负权团无法在 $O(n^{3-\varepsilon})$ 时间内解决 [52]。与交叉组合框架的第一个区别在于，我们从一个负权三角形实例 G 开始，然后将其分解为许多小实例，具体如下：将 G 的顶点 $V(G)$ 划分为 z 个大小为 n/z 的集合 V_1、.....、V_z，其中 z 的选择取决于我们希望排除的运行时间（有关指定 z 的具体公式，请参见第 3 节中的引理 3.3 的证明）。然后，我们创建 z^3 个负权团实例：对于每个 (i、j、k)，取图 $G[V_i \cup V_j \cup V_k]$。显然， G 含有负权三角形，当且仅当创建的实例中至少有一个含有负权三角形。

接下来，我们按照上述说明，对负权团应用组合（不交并），得到一个具有 G' $n' = z^3 \cdot n/z = z^2 n$ 个顶点的实例。注意， G' 中最大连通分量的大小为 $3n/z$。因此，运行时间为

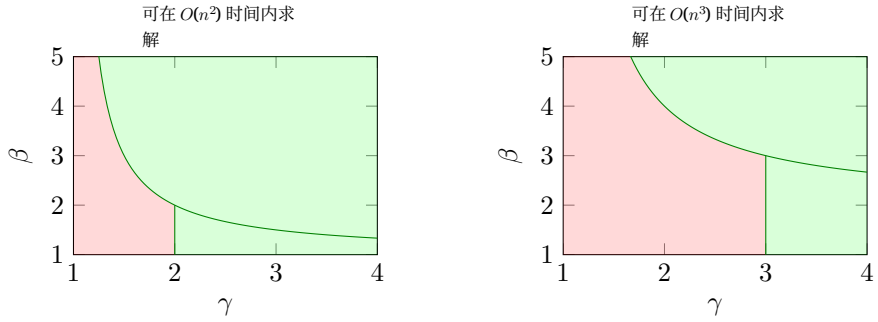


图1 关于运行时间为 $O(n^\gamma + \ell^\beta)$ 时可能（绿色）和不太可能（红色）的权衡概览。左：第一类可在 $O(n^2)$ 时间内求解的问题（表1上半部分）；右：第二类可在 $O(n^3)$ 时间内求解的问题（表1下半部分）。

$O(n^\gamma + \ell^\beta)$ 的负权三角形算法可在 $O(z^{2\gamma}n^\gamma + (3n/z)^\beta)$ 时间内求解 G'。通过仔细地将 z 选为 n、β 和 γ 的函数，我们可以得到，对于 γ 和 β 的各种组合，该算法的运行时间为 $O(n^{3-\varepsilon})$。

负权三角形能够按上述方式分解，这一性质并非该问题独有。事实上，早已有研究指出："许多问题（例如 SAT）都有一种简单的自归约，可以证明其'直接或'版本是困难的，前提是问题本身是困难的" [4]。我们的框架将这种分解概念形式化（定义见第2节），并对交叉组合的定义作出调整。我们还证明，正交向量、3-Sum 和负权 k-团等常用的"困难"问题都可以分解。因此，只需展示交叉组合，即可应用我们的框架并获得下界。

1.3 预备知识与符号约定。

问题定义。对于 $\ell \in \mathbb{N}$，我们设 $[\ell] := \{1, 2, \ldots, \ell\}$。

正交向量
输入：两个大小为n的集合 A，$B \subseteq \{0,1\}^d$，其中 $d \in \mathbb{N}$。**问题：**是否存在 $a \in A$ 和 $b \in B$，使得 a 和 b 正交，也就是说，对于每个 $i \in [d]$，i的第 a 个坐标或 i的第 b 个坐标为零。

我们将正交向量问题限制为满足 $d \leq O(\log n)$ 的实例，称为对数维度正交向量。

3-Sum
输入：一个包含 A 个 n 整数的数组。**问题：**是否存在 i、j、$h \in [n]$，使得 $A[i] + A[j] + A[h] = 0$?

负权 k团**输入：**一个边加权图 G，包含 n 个顶点。**问题：**图 G 是否包含一个负权的 k团?

最长公共子序列**输入：**两个字符串 x^1 和 x^2，长度为 n，字母表为 Σ 和 $k \in \mathbb{N}$。**问题：**判断 k 中是否存在长度为 x^1 且同时属于 x^2的公共子序列。

6 通过细粒度交叉组合得到的 P 中问题的参数化下界

最长公共（弱）递增子序列**输入：** 两个字符串 x^1 和 x^2，长度为 n，取自$\mathbb{N}$ 和 $k \in \mathbb{N}$。**问题：** 判断是否存在公共子序列 y，长度为 k，属于 x^1 和 x^2，且满足 $y[i] < y[i+1]$ $(y[i] \leq y[i+1])$ 对所有 $i \in [k-1]$均成立。

离散弗雷歇距离
输入： 平面中的两个点列表 $P = (p_1, \ldots, p_n)$ 和 $Q = (q_1, \ldots, q_n)$，以及 $k \in \mathbb{N}$。**问题：** P 和 Q 的弗雷歇距离是否至多为 k，也就是说，是否存在两个满射的非递减函数 f_P，f_Q：$[2n] \to [n]$，使得 $f_P(1) = 1 = f_Q(1)$，$f_P(2n) = n = f_Q(2n)$，并且 $\max^{i \in [2n]} \text{dist}(p_{f_P(i)}\ q_{f_Q(i)}) \leq k$?

平面运动规划
输入： 平面中一组 n 互不相交且互不接触的轴平行线段障碍物，以及一个线段机器人（杆或梯子）、一个给定的起点和一个给定的目标点。**问题：** 在允许平移和旋转的情况下，能否将杆从起点移动到目标点，同时不与障碍物发生碰撞?

第二短路径
输入： 一个 n-顶点边加权有向图 G，顶点 s 和 t，以及 $k \in N$。**问题：** 第二短 s-t-路径的长度是否至多为 k?

三角形集合**输入：** 一个顶点着色图 G，其上有 n 个顶点。**问题：** 对于每组三种颜色，是否存在 G 一个三角形，其顶点分别着三种颜色?

假设。 本文中的条件下界基于 SETH、3-Sum 假设和全点对最短路径假设（见 VassilevskaWilliams [51] 了解更多详情）。

▶ **假设 1** (SETH)。 对于每个 $\varepsilon > 0$，都存在一个 $k \in \mathbb{N}$，使得 k-SAT 无法在 $O(2^{(1-\varepsilon)n})$ 时间内求解，其中 n 是变量的数量。

▶ **假设 2** (3SUM)。 对于任意 $\varepsilon > 0$，包含 $\{-n^4, \ldots, n^4\}$ 中 n 个整数的 3-Sum 无法在 $O(n^{2-\varepsilon})$ 时间内求解。

▶**假设 3** (全点对最短路)。 对于任意 $\varepsilon > 0$，在具有多项式边权的 n 顶点图上，全源最短路径问题无法在 $O(n^{3-\varepsilon})$ 时间内求解。

参数化复杂性。 在上述许多问题中，n 不是输入规模，而是一个参数，且输入规模受 $n^{O(1)}$ 限制。参数化问题是由实例 (I, p) 构成的集合 $\in \Sigma^* \times \Sigma^*$，其中 Σ 表示一个有限字母表，I 表示经典实例，p 表示参数。核化 是一种多项式时间算法，它将任意实例 (I, p) 映射为一个等价实例 (I', p')（即核），使得对于某个可计算函数 f，有 $|I'| + p' \leq f(p)$。如果 f 是多项式，则 (I', p') 是一个多项式大小核。

本文将讨论范围限定如下：要么 $p = n$（p 是单个参数），要么 $p =$（n, ℓ）（p 是组合参数）。此外，n 和 ℓ 始终是非负整数；n 与输入规模相关，而 ℓ 则无关（ℓ 可视为“经典”参数）。

2 框架

我们的框架包含以下三个步骤（高层次的介绍请参见第1.2节）。

1. 从“困难”问题 P 的一个实例 $(I, n_{\mathcal{P}})$ 开始，将其分解为 P 的多个实例的析取：$t(I_1, n_1) \ldots, (I_t, n_t)$。在第 3 节中，我们为常用的困难问题 3-Sum 、正交向量和负权 k- 团提供了这类分解。2. 使用 OR 交叉组合，将 $(I_1, n_1) \ldots, (I_t, n_t)$ 组合成“目标”问题的一个实例 $(J, (n, \ell))$。这一步需要针对具体应用完成。3. 将假设存在的 $\widetilde{O}(n^{\gamma} + \ell^{\beta})$-时间算法应用于 J。如果 γ 和 β 的组合足够小，那么所得算法将比问题 P 的下界更快。

为了更正式地描述我们的框架，我们先定义分解和交叉组合。请注意，文中提及的所有问题都是参数化问题。

▶ **定义 2.1**（OR 分解）。对于问题 P，$\alpha > 1$ 的 α-OR 分解 是这样一种算法：给定 $\lambda > 0$ 和 P 的一个实例（I, n），该算法在某个 $\alpha' < \alpha$ 下，以$\widetilde{O}$（$n^{\alpha'}$）时间计算出 $t \in \widetilde{O}$（$n^{\alpha\lambda/(\alpha+\lambda)}$）个实例（$I_1, n_1$）.....（$I_t, n_t$），这些实例均属于 P，且满足

- （I, n）∈ 属于 P，当且仅当（I_i, n_i）∈ 属于 P（对某个
- $i \in [t]$成立），并且$n_i \in \widetilde{O}$（$n^{\alpha/(\alpha+\lambda)}$）对所有 $i \in [t]$均成立。

如果存在一个 α-OR 分解，我们就称问题$\mathcal{P}$ 是 α-*OR* 可分解的。对于某些问题，证明其可OR-分解性比证明其他问题更容易。因此，使用适当的归约来传递可OR-分解性可能是有益的（我们在第 3 节证明 3-Sum 是 2-OR 可分解时便采用了这种做法）。拟线性时间归约若不会使参数增加太多，也是一种选择。为此，我们称这样一种归约为参数保持归约：它将问题$\mathcal{P}$的一个实例（$I^{\mathcal{P}}$，$n^{\mathcal{P}}$）映射为问题$\mathcal{Q}$的一个实例（$I^{\mathcal{Q}}$，$n^{\mathcal{Q}}$），且满足 $n^{\mathcal{Q}} \in \widetilde{O}$（$n^{\mathcal{P}}$）。

▶**命题 2.2**。设 P 和 Q 是两个问题，且存在从 P 到 Q 以及从 Q 到 P 的准线性时间参数保持归约。那么对于任意 $\alpha > 1$，P 是α-可*OR*-分解的，当且仅当 Q 是α-可*OR*-分解的。

证明。假设 P 是 α-可OR-分解的（Q 是 α-可OR-分解的情形是对称的）。现在我们为 Q 给出一个 α-OR 分解。给定一个实例（$I^{\mathcal{Q}}$，$n^{\mathcal{Q}}$）和$\lambda > 0$，我们首先将（$I^{\mathcal{Q}}$，$n^{\mathcal{Q}}$）归约为一个实例（$I^{\mathcal{P}}$，$n^{\mathcal{P}}$）。随后，我们应用 P 的 α-OR 分解，得到 P 的实例（$I_1^{\mathcal{P}}$，$n_1^{\mathcal{P}}$）..... （$I_t^{\mathcal{P}}$，$n_t^{\mathcal{P}}$）。最后，我们将每个实例（$I_i^{\mathcal{P}}$，$n_i^{\mathcal{P}}$）归约为一个实例（$I_i^{\mathcal{P}}$，$n_i^{\mathcal{P}}$）。显然，这是 Q 的一个 α-OR 分解。◀

在我们框架的第二步中，我们引入细粒度 OR 交叉组合：

▶ **定义 2.3** (细粒度 OR 交叉组合)。对于 $\nu \geq 1, \mu \geq 0$，从问题 P 到问题 Q 的一个(ν, μ)-OR- 交叉组合是一个算法 A，它以 P 的 t 个实例$(I_1^{\mathcal{P}}, n_1^{\mathcal{P}}) .., (I_t^{\mathcal{P}}, n_t^{\mathcal{P}})$ 为输入，在 $\widetilde{O}(t \cdot n_{\max}^{\nu} + \sum_{i=1}^{t} |I_i^{\mathcal{P}}|)$ 时间内运行，且满足 $n_{\max} := \max_{i \in [t]} n_i^{\mathcal{P}}$，并计算出 Q 的一个实例$(I^{\mathcal{Q}}, (n^{\mathcal{Q}}, \ell^{\mathcal{Q}}))$，使得

1. $(I^{\mathcal{Q}}, (n^{\mathcal{Q}}, \ell^{\mathcal{Q}})) \in \mathcal{Q}$ 当且仅当$(I_i^{\mathcal{P}}, n_i^{\mathcal{P}}) \in \mathcal{P}$对于某个 $i \in [t]$，并且2. $n^{\mathcal{Q}} \in \widetilde{O}(t \cdot n_{\max}^{\nu})$ 和 $\ell^{\mathcal{Q}} \in \widetilde{O}(n_{\max}^{\mu})$。

我们称问题 $\mathcal{P}$ (ν，μ)-*OR*-交叉组合 到问题 $\mathcal{Q}$，如果存在一个从$\mathcal{P}$ 到$\mathcal{Q}$的(ν，μ)-OR 交叉组合。

▶ **定理 2.4**。设 $\alpha > \nu \geq 1$、$\gamma > 1$ 和 $\mu > 0$ 满足 $\alpha > \nu \cdot \gamma$。设 P 是一个参数为 $n_{\mathcal{P}}$ 的 α-*OR* 可分解问题，它（ν, μ）-*OR* 交叉组合到一个参数为 $n_{\mathcal{Q}}$ 和 $\ell_{\mathcal{Q}}$ 的问题 Q。如果存在一个用于 Q 的 $\widetilde{O}(n^{\gamma}{}_{\mathcal{Q}} + \ell^{\beta}{}_{\mathcal{Q}})$-时间算法，并且

$$0 < \beta < \frac{\gamma \cdot (\alpha - \nu)}{(\gamma - 1) \cdot \mu},$$

那么，对于某个 $\varepsilon > 0$，P 可以在 $O(n^{\alpha-\varepsilon}{}_{\mathcal{P}})$ 时间内求解。

证明。设 $(I_{\mathcal{P}}, n_{\mathcal{P}})$ 是 $\mathcal{P}$的一个实例。我们求解 $(I_{\mathcal{P}}, n_{\mathcal{P}})$ 的算法分以下步骤运行：

1. 应用 α-OR 分解（其中 λ 如下所述），得到实例 $(I_1, n_1) \ldots, (I_t, n_t)$，满足 $\max_{i \in [t]} n_i \leq q := n^{\alpha/(\alpha+\lambda)}$ P 且 $t := n^{\alpha\lambda/(\alpha+\lambda)}$ P $= q^{\lambda}$。注意，根据定义，此步骤的运行时间为某个 $\alpha' < \alpha$ 对应的 $\widetilde{O}(n_1'$ P.) 时间。n_t2. 对 $(I_1,$ $n_{\mathcal{Q}}$ $=$ $\widetilde{O}\ q^{\lambda+\nu} + n^{\alpha'}$ $\ell_{\mathcal{Q}} \in \widetilde{O}\ q^{\mu}$) , $(I_t,$) 应用 (ν, μ)-OR 交叉组合，为 Q 计算实例 $(I_{\mathcal{Q}}, (n_{\mathcal{Q}}, \ell_{\mathcal{Q}}))$。注意，运行时间为 ，且属于 $\widetilde{O}(t \cdot q^{\nu} + \sum_{i=1}^{t} |I_i^{\mathcal{P}}|)$ (P)。此外，( )。 3. 应用运行时间为 $\widetilde{O}(n^{\gamma}$Q $+ \ell^{\beta}$Q) 的算法。这需要 $\widetilde{O}(q^{(\lambda+\nu)\gamma} + q^{\mu \cdot \beta})$ 时间。

还需证明这三个步骤的运行时间均为某个 $\varepsilon > 0$ 下的 $O(n^{\alpha-\varepsilon}{}_{\mathcal{P}})$。为此，我们现在说明 $\lambda = \beta\mu/\gamma - \nu$。注意到 $\lambda + \nu = \beta\mu/\gamma < \mu \cdot \beta$，因此只需证明第三步的运行时间为某个 $\epsilon > 0$ 下的 $\widetilde{O}(n^{\alpha-\epsilon}{}_{\mathcal{P}})$。最后一步的运行时间为 $\widetilde{O}(q^{\mu \cdot \beta}) = \widetilde{O}(n^{\alpha\mu \cdot \beta/(\alpha+\lambda)}{}_{\mathcal{P}})$。其指数为

$$\frac{\alpha\mu\beta}{\alpha + \lambda} = \frac{\alpha\mu\beta}{\alpha + \beta\mu/\gamma - \nu} = \frac{\alpha\beta\gamma\mu}{\gamma(\alpha - \nu) + \beta\mu} = \frac{\alpha\gamma\mu}{\gamma(\alpha - \nu)/\beta + \mu}.$$

根据假设，我们有 $\beta < {}^{\gamma \cdot (\alpha - \nu)\ (\gamma - 1) \cdot \mu}$，因此其指数为

$$\frac{\alpha\gamma\mu}{\gamma(\alpha - \nu)/\beta + \mu} < \frac{\alpha\gamma\mu}{\gamma(\alpha - \nu)/\left(\frac{\gamma \cdot (\alpha - \nu)}{(\gamma - 1) \cdot \mu}\right) + \mu} = \frac{\alpha\gamma\mu}{(\gamma - 1) \cdot \mu + \mu} = \alpha.$$

还有一件事需要处理：我们必须确保 $\lambda > 0$。当 $\beta \to 0$, λ 变为负数时，情况并非总是如此。不过，对于任意 $\beta' > \beta$，一个运行时间为 $\widetilde{O}(n^{\gamma}{}_{\mathcal{Q}} + \ell^{\beta}{}_{\mathcal{Q}})$ 的算法，也意味着存在一个运行时间为 $\widetilde{O}(n^{\gamma}{}_{\mathcal{Q}} + \ell^{\beta'}{}_{\mathcal{Q}})$ 的算法。因此，我们只需选取某个更大的 β'，使得相应的 λ' 大于 0。为此，令 $\beta_{\max} := {}^{\gamma \cdot (\alpha - \nu)\ (\gamma - 1) \cdot \mu}$ 为 β 的上界。注意到 $\alpha > \nu \cdot \gamma$ 意味着

$$\lambda_{\max} := \frac{\beta_{\max}\mu}{\gamma} - \nu = \frac{\frac{\gamma \cdot (\alpha - \nu)}{(\gamma - 1) \cdot \mu}\mu}{\gamma} - \nu = \frac{\alpha - \nu}{\gamma - 1} - \nu = \frac{\alpha - \nu - \nu \cdot (\gamma - 1)}{\gamma - 1} = \frac{\alpha - \nu \cdot \gamma}{\gamma - 1} > 0$$

因此，我们可以选择 $\beta < \beta' < \beta_{\max}$ 使得 $\lambda' := \frac{\beta'\mu}{\gamma} - \nu > 0$。 ◀

注意，如果对于$\mathcal{P}$ 存在 $\Omega(n^{\alpha})$ 的（条件）运行时间下界，那么定理 2.4 将（有条件地）排除形如 $\widetilde{O}(n_{\mathcal{P}} + \ell^{\beta})$ 的运行时间，其中任意 $\beta \in \mathbb{R}$ 均满足 $\lim_{\gamma \to 1} {}^{\gamma \cdot (\alpha - \nu)\ (\gamma - 1) \cdot \mu} = \infty$。这种运行时间也排除了线性时间可计算的多项式大小核。更确切地说，我们得到如下结论。

▶ **推论 2.5.** 设 $\alpha > \nu \geq 1$、$\gamma > 1$ 和 $\mu > 0$ 满足 $\alpha > \nu \cdot \gamma$。设 P 是一个参数为 $n_{\mathcal{P}}$ 的 α-*OR* 可分解问题，且该问题可将(ν, μ)-*OR*-交叉组合到参数为 $n_{\mathcal{Q}}$ 和 $\ell_{\mathcal{Q}}$的问题 Q。假设存在一个 $(n^{\xi}{}_{\mathcal{Q}})$ 算法来判定 Q，并且该算法的

运行时间的上界 $n_{\mathcal{Q}}$ 受输入规模所界定。若存在一个针对 Q 的 $\widetilde{O}(\ell^{\beta}\ _{\mathcal{Q}})$-规模、 $\widetilde{O}(n^{\gamma}\ _{\mathcal{Q}})$-时间核，其中 $\gamma > 1$、 $\beta \in \mathbb{R}$为某些参数，并且

$$0 < \beta < \frac{\gamma \cdot (\alpha - \nu)}{(\gamma - 1) \cdot \mu \cdot \xi},$$

那么，对于某个 $\varepsilon > 0$，可以在 $O(n^{\alpha-\varepsilon}\ _{\mathcal{P}})$ 时间内求解 P。

证明。 一个大小为 $\widetilde{O}(\ell^{\beta}\ _{\mathcal{Q}})$、运行时间为 $\widetilde{O}(n^{\gamma}\ _{\mathcal{Q}})$ 的核，加上一个运行时间为 $\widetilde{O}(n^{\xi}\ _{\mathcal{Q}})$、用于求解 $\mathcal{Q}$的算法，可得一个运行时间为 $\widetilde{O}(n^{\gamma}\ _{\mathcal{Q}}\ ^{+\ell^{\beta\xi}}\ _{\mathcal{Q}})$ 的算法来求解$\mathcal{Q}$。推论由定理 2.4 得出。◀

我们应用该框架的一般方法如下。首先，选取一个问题$\mathcal{P}$，在某个假设下，该问题无法在 $O(n^{\alpha-\varepsilon})$ 时间内针对 $\varepsilon > 0$ 求解。然后，为$\mathcal{P}$ 构造一个α-分解，接着将其进行 (1,1)-OR 交叉组合，映射到目标问题。

3 个 OR-可分解问题

为了应用该框架，我们首先需要一些 OR-可分解问题。我们将发现，细粒度复杂性中的三个基本问题——正交向量、3-Sum 和负权 3-团——都是 OR-可分解问题。这些问题也为我们提供了运行时间下界：注意，除非 SETH 和 3-Sum 假设分别不成立，否则前两个问题无法在 $O(n^{2-\varepsilon})$ 时间内求解 [51]。此外，我们还利用负权 3-团（=负三角形）无法在 $O(n^{3-\varepsilon})$ 时间内求解这一事实，除非全点对最短路径假设不成立 [52]。

我们将利用这一点：“许多问题（例如 SAT）都有一种简单的自归约，可以证明在假设问题本身困难的情况下，其‘直接或’版本也是困难的” [4]。这种自归约基于将实例划分为许多较小的实例，其中至少有一个包含所需的小结构（即一对正交向量、三个和为 0 的数，或一个负三角形）。

正交向量。 现在我们证明正交向量是 2-OR 可分解的。

▶**引理 3.1**。 *以向量数量为参数的正交向量问题是 2-OR 可分解的。*

证明。设 (I, n) 是正交向量的一个实例，且 $\lambda > 0$。令 $\epsilon := \frac{2}{2+\lambda}$。将 A 划分为 $z := \mathsf{d}n^{1-\epsilon}\mathsf{e}$ 个集合 $A_1, \ldots, A_z$，每个集合至多包含 $\mathsf{d}n^{\epsilon}\mathsf{e}$ 个向量。对称地，将 B 划分为 $B_1, \ldots, B_z$，每个集合至多包含 $\mathsf{d}n^{\epsilon}\mathsf{e}$ 个向量。不失一般性，我们假设 $|A_i| = |B_j| = \mathsf{d}n^{\epsilon}\mathsf{e} =: n'$（例如，可以通过添加全 1 向量实现这一点）。对于每一对 $(i, j) \in [z]^2$，构造一个正交向量实例 $(I_{(i,j)}, n') = ((A_i, B_j)\ n')$。我们声称，这构成了一个 2-或分解。

每个实例中的向量数量为 $n'\ I_{(i,j)}\ n' = O(n^{2/(2+\lambda)})$。生成的实例数量为 $z^2 = O(n^{2\cdot(1-\epsilon)}) = O(n^{2\cdot(1-2/(2+\lambda))}) = O(n^{(4+2\lambda-4)/(2+\lambda)}) = O(n^{2\lambda/(2+\lambda)})$。 ~ ~计算分解的运行时间为 $O(z^2 \cdot n') = O(n^{2-\epsilon})$。

剩下要证明的是，对于某个 $(i, j) \in [z]^2$，(I, n) 是“是”实例，当且仅当 $(I_{(i,j)}, n')$ 是“是”实例。首先假设 (I, n) 是“是”实例。那么存在某个 $a \in A$ 和 $b \in B$，使得 a 和 b 正交。令 $i \in [z]$ 满足 $a \in A_i$，并令 $j \in [z]$ 满足 $b \in B_j$。那么 a 和 b 是 $(I_{(i,j)}, n')$ 中的正交向量，这表明 $(I_{(i,j)}, n')$ 是“是”实例。

最后，假设存在 $(i^*, j^*) \in [z]^2$，使得 $(I_{(i^*,j^*)}n')$ 是一个“是”实例。那么存在 $a \in A_{i^*}$ 和 $b \in B_{j^*}$，二者正交。因此， a 和 b 是 I 中的正交向量，这意味着 (I, n) 是一个“是”实例。◀

►注 3.2。注意，上述分解不会改变维度 d。因此，即使是限制版本的正交向量问题，只要满足 $d \in O(\log n)$，也都是 2-OR 可分解的。此外，我们可以假设所有构造出的正交向量实例都具有相同数量的向量和相同的维度。

负权重 k-团。现在我们证明，负权重 k-团是 k-可OR-分解的。

► **引理 3.3**。对于任意 $k \geq 3$，以顶点数量为参数的负权重 k-团是 k-可*OR*-分解的。

证明。证明思路沿用引理 3.1。设 $(I = (G, w)\ n)$ 是负权重 k-团的一个实例，且 $\lambda > 0$。令 $\epsilon := \frac{k}{k+\lambda}$。将顶点集合 $V(G)$ 划分为 $z := \mathsf{d}n^{1-\epsilon}\mathsf{e}$ 个集合 $V_1, \ldots, V_z$，每个集合至多包含 $\mathsf{d}n^{\epsilon}\mathsf{e}$ 个顶点。不失一般性，假设对所有 $i \in [z]$，$|V_i| = \mathsf{d}n^{\epsilon}\mathsf{e}$（例如，可以通过添加孤立顶点来实现）。令 $n_{(i_1,\ldots,i_k)} := |\{i_1, \ldots, i_k\}| \cdot |V_1|$。对于每个元组 $(i_1, \ldots, i_k) \in [z]^k$，构造负权重 k-团的一个实例 $(I_{(i_1,\ldots,i_k)} = G[A_{i_1} \cup \cdots \cup A_{i_k}], n_{(i_1,\ldots,i_k)})$。我们声称，这构成了一个 k-OR 分解。

每个实例 $(I_{(i,j)},\ n_{(i_1,\ldots,i_k)})$ 至多有 $O(k \cdot n^{\epsilon}) = O(n^{k/(k+\lambda)})$ 个顶点（注意，k 在此处是常数）。所创建实例的数量为 $z^k = O(n^{k\cdot(1-\epsilon)}) = O(n^{k\cdot(1-k/(k+\lambda))})$ $= O\ n^{(k^2+k\lambda-k^2)/(k+\lambda)}\ = O\ n^{k\lambda/(k+\lambda)}$ ( ) ( )。此外，所有所创建实例的大小总和，因此运行时间也为 $\widetilde{O}(z^k \cdot k \cdot n^{2\epsilon}) = \widetilde{O}(n^{k-k\epsilon+2\epsilon}) = \widetilde{O}(n^{k-(k-2)\epsilon}) = \widetilde{O}(n^{k'})$，其中 $k' < k$ 为某个常数。

还需证明，对于某个 $(i_1, \ldots, i_k) \in [z]^k$，当且仅当 $(I_{(i_1,\ldots,i_k)}\ n_{(i_1,\ldots,i_k)})$ 是一个“Yes”实例时，(I, n) 也是一个“Yes”实例。首先假设 (I, n) 是一个“Yes”实例。因此，G 包含一个负权团 $C = \{v_1, \ldots, v_k\}$。令 i_j 满足 $v_j \in V_{i_j}$。那么，C 是 $(I_{(i_1,\ldots,i_k)}\ n_{(i_1,\ldots,i_k)})$ 中的一个负权 k- 团。

最后，假设存在 $(i_1、\ldots\ldots、i_k) \in [n]^k$ 使得 $(I_{(i_1,\ldots,i_k)}\ n_{(i_1,\ldots,i_k)})$ 是一个“是”实例。那么在 $G[V_{i_1} \cup \ldots\ldots \cup V_{i_k}]$ 中存在一个团。由于 G 包含 $G[V_{i_1} \cup \ldots\ldots \cup V_{i_k}]$，因此 G 也包含一个负权 k-团。◄

3-Sum。证明 3-Sum 是 2-OR 可分解的还需要一些工作。

►**引理 3.4**。以数字数量为参数的 *3-Sum* 是 *2-OR* 可分解的。

我们不直接考虑 3-Sum，而是考虑其一个称为卷积 3-Sum 的变体；已有研究表明，在准线性时间拉斯维加斯归约下，它与 3-Sum 等价 [43]。

卷积 3-Sum**输入：**一个数组 A，包含 n 个整数。**问题：**是否存在 i、$j \in [n]$，使得 $A[i] + A[j] = A[i + j]$?

为简单起见，我们不会直接处理卷积 3-Sum，而是处理它的“多色”版本：

多色卷积 3-Sum**输入：**三个数组 A、B和 C，每个数组包含 n 个整数。**问题：**是否存在 i、$j \in [n]$，使得 $A[i] + B[j] = C[i + j]$?

首先，我们展示如何将卷积 3-Sum 归约为多色卷积 3-Sum，反之亦然。

▶**引理 3.5.** 存在一种线性时间参数保持归约，可将以数字数量为参数的卷积 *3-Sum* 归约为同样以数字数量为参数的多色卷积 *3-Sum*，反之亦然。

证明。 设 (A, n) 是一个 卷积 3-Sum实例。我们通过如下设置构造一个实例 $((A, B, C)\ n)$，它属于 多色卷积 3-Sum： $B[i]: = A[i]$ 以及 $C[i]: = A[i]$。注意，对于 $i, j \in [n]$，有 $A[i] + A[j] = A[i + j]$ 当且仅当 $A[i] + B[i] = A[i] + A[j] = C[i + j]$，这意味着这两个实例是等价的。

现在考虑一个多色卷积 3-Sum 的实例 $(I = (A, B, C)\ n)$。设 $z_{\max} := \max^{i\in[n]} \max\{A[i], B[i], C[i]\} + 1$。我们按如下方式构造一个卷积 3-Sum 的实例 $(A', n' = 4n)$。数组 A' 的长度为 $4n$，其中

$$A'[i] := \begin{cases} -5z_{\max} & 1 \le i \le n \\ A[i-n] + z_{\max} & n+1 \le i \le 2n, \\ B[i-2n] + 3z_{\max} & 2n+1 \le i \le 3n, \\ C[i-3n] + 4z_{\max} & 3n+1 \le i \le 4n\,. \end{cases}$$

首先，注意到，$A[i] + B[j] = C[i + j]$ 是（I, n）的一个解，这意味着（A',n'）也有一个解，因为 $A'[i+n] + A'[j+2n] = A[i] + z_{\max} + B[j] + 3z_{\max} = C[i+j] + 4z_{\max} = A'[3n+i+j]$。其次，考虑一个解 $A'[i'] + A'[j'] = A'[i'+j']$。那么 $i', j' > n$，否则 $A'[i'] + A'[j'] < 0$ 且 $A'[i'] + A'[j'] \neq -5z_{\max}$。此外，$i', j' \le 3n$，否则 $A'[i'] + A'[j'] > 5z_{\max} > A'[i'+j']$。进一步地，不失一般性，我们有 $n+1 \le i' \le 2n$ 和 $2n+1 \le j' \le 3n$。因此，我们有 $A[i'-n] + B[j'-2n] = A'[i'] - z_{\max} + A'[j'] - 3z_{\max} = A[i'+j'] - 4z_{\max} = C[i'+j'-3n]$。◀

接下来，我们证明多色卷积 3-Sum 是 2-OR 可分解的。

▶**引理 3.6.** 以数字个数为参数的多色卷积 *3-Sum* 是 *2-OR* 可分解的。

证明。证明的思路与引理 3.1 基本相同。对于一个数组 A，我们用 $A[i, j]$ 表示由条目 $(A[i], A[i + 1], \ldots, A[j])$ 组成的数组。设 $(I = (A, B, C)\ n)$ 是一个多色卷积 3-Sum 实例，且 $\lambda > 0$。令 $\epsilon := \frac{2}{2+\lambda}$，并令 $q := \mathsf{d}n^{\epsilon}\mathsf{e}$。将 A 划分为 $z := \mathsf{d}n^{1-\epsilon}\mathsf{e}$ 个数组 $A_1 = A[1, q], A_2 = A[q+1, 2q], \ldots, A_z = A[(z-1)q+1, n]$。为简便起见，假设 $n = z \cdot q$。对称地，将 B 划分为 z 个数组 $B_1 = B[1, q], B_2 = B[q + 1, 2q], \ldots, B_z = B[(z - 1)q + 1, n]$。对于 $(\ell, p) \in [z]^2$，我们用 $C_{(\ell,p)}$ 表示区间 $C[(\ell + p - 2)q + 2, (\ell + p)q]$。对于每一对 $(\ell, p) \in [z]^2$，构造一个卷积 3-Sum 实例 $(I_{(\ell,p)} = (A_\ell, B_p, C_{(\ell,p)})\ q)$。我们声称，这构成了一个 2-或分解。

每个实例 $(I_{(\ell,p)},\ q)$ 至多有 $O(q) = O(n^{\epsilon}) = O(n^{2/(2+\lambda)})$。创建的实例数量为 $z^2 = O(n^{2\cdot(1-\epsilon)}) = O(n^{2\cdot(1-2/(2+\lambda))}) = O(n^{(4+2\lambda-4)/(2+\lambda)}) = O(n^{2\lambda/(2+\lambda)})$。$\tilde{}$ $\tilde{}$计算分解可在 $O(z^2 \cdot q) = O(n^{2-\epsilon})$ 时间内完成。

接下来要证明，对于某个 $(\ell,\ p) \in [z]^2$，$(I,\ n)$ 是一个“Yes”实例，当且仅当 $(I_{(\ell,p)},\ q)$ 是一个“Yes”实例。首先假设 $(I,\ n)$ 是一个“Yes”实例。那么存在某个 $i = (\ell - 1)q + i^*, j = (p - 1)q + j^*$，使得 $A[i] + B[j] = C[i + j]$。注意到 $A[i] = A_\ell[i^*], B[j] = B_p[j^*]$，并且 $C[i+j] = C_{(\ell,p)}[i^* + j^*]$。因此，$(I_{(\ell,p)},\ q)$ 是一个“Yes”实例。

最后，假设存在 $(\ell,\ p) \in [z]^2$，使得 $(I_{(\ell,p)},\ q)$ 是一个“是”实例。那么 $A_\ell[i^*] + B_p[j^*] = C_{(\ell,p)}[i^* + j^*]$ 对于某个 $i^*,\ j^* \in [q]$。因此，$A[(\ell - 1)q + i^*] + B[(p - 1)q + j^*] = C_{(\ell,p)}[i^* + j^*] = C[(\ell + p - 2)q + i^* + j^*]$，这意味着 $(I,\ n)$ 是一个“是”实例。◀

3-Sum 的 2-OR 可分解性现在可由 3-Sum 与卷积 3-Sum 之间的准线性时间保参数归约，以及卷积 3-Sum 与多色卷积 3-Sum 之间的引理 3.5 得出。

引理 3.4 的证明。 由引理 3.6、3-SUM 与 卷积 3-Sum [43], 卷积 3-Sum 与 多色卷积 3-Sum的等价性，以及命题 2.2 可得。◀

应用框架。 上述结果使我们的框架更易于应用。要应用定理 2.4，我们只需提供一个从上述三个问题之一出发的合适 OR 交叉组合。因此，我们得到以下结论。

▶**命题 3.7。** 设 Q 是一个具有参数 n_Q 和 ℓ_Q的问题。若以 n 为参数的正交向量问题（分别为 *3-Sum*）可通过$(1,1)$-*OR* 交叉组合归约到 Q，则对于任意 $2>\gamma>1$ 和 $\beta<\gamma/\gamma-1$，若存在一个运行时间为 $O(\ell^{\beta}{}_Q+n^{\gamma}{}_Q)$ 的算法来求解 Q，便可分别推翻 *SETH* 和 *3-Sum* 假设。

如果按顶点数参数化的负三角形问题能以 $(1,1)$-*OR*-交叉组合到 Q，那么对于任意 $3>\gamma>1$ 和 $\beta<2\gamma/\gamma-1$，存在一个针对 Q 的 $O(\ell^{\beta}{}_Q+n^{\gamma}{}_Q)$ 时间算法，就会推翻全点对最短路径假设。

4 应用

现在，我们将定理 2.4 中的框架应用于若干不同领域的问题，例如字符串问题（第 4.1 节）、计算几何（第 4.2 节）和子图同构（第 4.3 节）。

4.1 字符串问题

首先来看最长公共子序列，它可以通过一个运行时间为 $O(n^2)$ 的算法求解 [18]。假设 SETH 成立，则不存在能以 $O(n^{2-\varepsilon})$ 时间求解最长公共子序列的算法，其中任意 $\varepsilon>0$ [1, 14]。不过，最长公共子序列可以在 $O(kn+n\log n)$ 时间内求解，其中 k 是最长公共子序列的长度 [39]。Bringmann 和 Künnemann [15] 证明，在 SETH 假设下，这一运行时间基本上是最优的。这里我们使用自己的框架重新证明这一结论。关于最长公共子序列更全面的文献综述，请参阅 Bringmann 和 Künnemann [15] 。

字母表 Σ 上的一个字符串 是 Σ^*中的一个元素。我们通过 $x[i]$访问字符串 x 的第 i个元素。字符串 x 的子序列 是一个字符串 y，满足存在一个单射且严格递增的函数 f，使得对于所有 i，$y[i]=x[f(i)]$。两个字符串 x 和 x' 的公共子序列 是一个同时为 x 和 x'的子序列的字符串。对于两个字符串 x 和 y，我们用 $x\circ y$表示它们的拼接（即以 x 开头、以 y结尾的字符串）。

▶**引理 4.1.** 以向量个数为参数的对数维正交向量问题可通过 $(1,1)$-*OR*-交叉组合归约为最长公共子序列问题，其中参数为输入字符串的长度以及最长公共子序列的长度 k。

证明。 设 $(I_1^{\mathrm{OV}},\ n_1)\ldots,(I_t^{\mathrm{OV}},\ n_t)$ 是对数维度的正交向量实例。记 d_i 为 I_i^{OV}的维度。Abboud、Backurs 和 Williams [1] 给出了一个从正交向量到最长公共子序列的归约：给定一个正交向量实例，其中包含 n 个维度为 d的向量，可构造

在 $n \cdot d^{O(1)}$ 时间内构造一个等价的最长公共子序列实例，其字符串长度为 $n \cdot d^{O(1)}$。我们将此归约应用于每个正交向量实例 (I_i^{OV}, n_i)，得到一个实例 $I_i^{\mathrm{LCS}} = ((x_i^1, x_i^2), (n_i^{\mathrm{LCS}}, k_i))$，其中 $k_i = O(n_i \cdot d_i^{O(1)})$ 且 $n_i^{\mathrm{LCS}} = n_i \cdot d_i^{O(1)}$。我们假设对每个 i, j，都有 $k_i = k_j$（可通过向字符串 x_i^1 和 x_i^2 追加适当长度的相同序列来实现），并令 $k := k_i$。此外，我们假设对 $i \neq j$，I_i^{LCS} 和 I_j^{LCS} 所用的字母表互不相交。定义 $x^1 := x_1^1 \circ x_2^1 \circ \cdots \circ x_t^1$ 以及 $x^2 := x_t^2 \circ x_{t-1}^2 \circ \cdots \circ x_1^2$。OR 交叉组合构造实例 (x^1, x^2, k)（参数为 (n^{LCS}, k)，其中 $n^{\mathrm{LCS}} = \sum_{i=1}^{t} n_i^{\mathrm{LCS}}$）。

现在我们证明该归约的正确性。首先，假设对于某个 $i \in [t]$，(I_i^{OV}, n_i) 是一个“是”实例。那么，x_i^1 和 x_i^2 包含一个长度为 $k_i = k$的子序列 y。这个子序列 y也是 x^1 和 x^2的子序列，因此(x^1, x^2, k)是一个“是”实例。反过来，假设(x^1, x^2, k)是一个“是”实例，也就是说，x^1 和 x^2 包含一个长度为 k的子序列 y。令 $i \in [t]$满足 y 的首字母包含在 x_i^1中。我们声称，y 中的所有字母都包含在 x_i^1中。由于 y的首字母 $y[1]$ 包含在 x_i^1中，当 $j < i$时，y中的字母都不可能包含在 x_j^1 中。对于 $j > i$，注意到（由于 x_j^2 中的任意字母只会出现在 x_j^2 和 x_j^1中）x_j^1 中的任意字母只会在 x_2 中的 x_i^2 之前出现，因此也必须在 y中的 $y[1]$之前出现，这与 $y[1]$ 是 y的首字母相矛盾。因此，y 包含在 x_i^1 和 x_i^2中。由此，(x_i^1, x_i^2, k)是一个“是”实例，这意味着(I_i, n_i)是一个“是”实例。

计算 OR 交叉组合可在 $t \cdot \max_{i \in t} n_i \cdot (\max_{i \in [t]} d_i)^{O(1)} = \widetilde{O}(t \max^{i \in [t]} n_i)$ 时间内完成，并且需要 $k = O(\max_{i \in t} n_i \cdot (\max_{i \in t} d_i)^{O(1)}) = \widetilde{O}(\max_{i=1} n_i)$（这里我们利用了 $d_i \in O(\log n_i)$）。此外，我们有 $n^{\mathrm{LCS}} = \sum_{i=1}^{t} n_i^{\mathrm{LCS}} = O(\sum_{i=1}^{t} n_i \cdot d_i^{O(1)}) = \widetilde{O}(\sum_{i=1}^{t})$。因此，这是一个 (1,1)-OR 交叉组合。◀

我们指出，在上述组合中使用非恒定大小字母表可能是必要的，因为它排除了多项式大小的线性时间可计算核。相比之下，以解大小为参数、字母表大小恒定的最长公共子序列问题存在一个线性时间可计算的多项式大小核（注意，基于 SETH 的 $O(n^{2-\epsilon})$ 下界对于二进制字母表同样成立 [15]）：

▶命题 *4.2*。 *</style}*以解大小为参数的恒定字母表大小最长公共子序列问题 k 存在一个可在线性时间内计算的多项式大小核。

证明。 考虑以下可直接导出多项式核的归约规则。A字符串 的子串 x 是一个字符串 y，满足 $y = (x[i], x[i+1], \ldots, x[j])$ 其中 $i < j$。

▶ **归约规则 4.3**。 **</style}若对于某个 N 和 {**1,2}，字符串 x^i 包含一个长度至少为 $(k+1)^t$、仅含 t 种不同字母的子串，且不包含一个长度为 $(k+1)^{t-1}$、至多含 $t-1$种不同字母的子串，则我们可以删除 x'中除前 $(k+1)^t$个字母以外的所有字母。

首先，我们证明该归约规则是安全的。设 $I = (x^1, x^2, k)$ 是最长公共子序列的一个实例。我们不失一般性地假设该归约规则可应用于 x^1，即 x^1 包含一个长度为 $(k+1)^t$ 且仅含 t 种不同字母的子串，而 x' 中仅含 $t' < t$ 种不同字母的所有子串长度至多为 $(k+1)^{t'}$。设 $I_{\mathrm{red}} = (x_{\mathrm{red}}^1, x^2, k)$ 是应用归约规则 4.3 后得到的实例。显然，如果 I 是一个“否”实例，那么 I_{red} 也同样是一个“否”实例。

现在假设 I 是一个“是”实例，也就是说，x^1 和 x^2 含有一个长度为 k 的公共子序列 z。根据假设，x' 不含有长度为 $(k+1)^{t-1}$、且至多包含 $t-1$ 种不同字母的子串。因此，在 x' 的前 $(k+1)^{t-1}-1$ 个字母中，这 t 种字母各至少出现一次。更一般地，对于任意 $i \in [k]$，每种字母都至少出现一次

在 $x'[(i-1)\cdot(k+1)^{t-1}+1],\ldots,x'[i\cdot(k+1)^{t-1}]$ 中。因此，由这些出现过的 t 种字母组成的任意长度为 k 的字符串，都是 x' 的子序列。因此， z 也是 x^1_{red} 的子序列（因而也是 x^1_{red} 和 x^2 的公共子序列），这表明 I_{red} 也是一个“是”实例。接下来，我们分析穷举应用归约规则 4.3 所需的时间。

▷ 命题 4.4。对于大小为常数的字母表，可以在线性时间内穷尽地应用归约规则 4.3。

证明。我们从左向右处理 x^1 和 x^2 。对于字母表的每个子集 S ，我们都维护一个变量，记录当前字母之前来自 S 的连续字母数量。如果对于某个字母，该数量超过 $(k+1)^{|S|}$ ，就删除这个字母。注意，每当我们找到一个长度为 $(k+1)^{|S|}$ 且只包含来自 S的字母的字符串时，该字符串不包含长度为 $(k+1)^{|S|-1}$ 且只包含来自 $S\setminus\{\ell\}$ 的字母的子串，其中某个 $\ell\in S$如若不然，存储的 $S\setminus\{\ell\}$ 的数量就会超过 $(k+1)^{|S|-1}$。

由于字母表的大小为常数，上述过程显然可以在线性时间内完成。 ◁

彻底应用归约规则 4.3 后，实例的规模显然为$\mathcal{O}(k^{|\Sigma|})$。由于 $|\Sigma|$ 是常数，因此这是一个多项式规模的核。◀

与最长公共子序列类似，我们为相关问题最长公共弱递增子序列和最长公共递增子序列推导出相似的困难性结果。这两个问题都能在略低于二次的时间内求解 [7, 23]。对于最长公共递增子序列，我们的下界已由 Duraj 等人证明 [24]。关于最长公共子序列的更全面文献综述，请参阅 Duraj [23], Duraj 等人 [24] 。

▶ **引理 4.5**。 对数维正交向量$(1,1)$-*OR*-交叉组合到最长公共（弱）递增子序列，其中参数为字符串长度 n 以及最长公共（弱）递增子序列的长度 k 。

证明。证明与引理 4.1 的证明类似。设 $(I_1^{\mathrm{OV}},n_1)\ldots,(I_t^{\mathrm{OV}},n_t)$ 是对数维度正交向量问题的实例。我们用 d_i 表示 I_i^{OV} 的维度。Polak [48] 和 Duraj、Künnemann 与 Polak [24] 给出了从正交向量到最长公共（弱）递增子序列的归约：给定一个包含 n 个维度为 d 的向量的正交向量实例，可在 $\widetilde{O}(n\cdot d^{O(1)})$ 时间内构造一个长度为 $\widetilde{O}(n\cdot d^{O(1)})$ 的最长公共（弱）递增子序列实例。我们将此归约应用于每个正交向量实例 (I_i^{OV},n_i)，得到一个实例 $I_i^{\mathrm{LCIS}}=(x_i^1,x_i^2,k_i)$，其中 $k_i=n_i\cdot d_i^{O(1)}$ ，且其字符串长度为 $\widetilde{O}(n\cdot d^{O(1)})$。我们假设对每个 i,j ，都有 $k_i=k_j$ （这可以通过向字符串 x_i^1 和 x_i^2 追加适当长度的相同递增数列来实现），并令 $k:=k_i$。设 $C_{\max}$ 为所有实例 $I_1^{\mathrm{LCIS}},\ldots,I_t^{\mathrm{LCIS}}$ 中出现的最大数。我们通过将实例 I_i^{LCIS} 中的每个数增加 $i\cdot C_{\max}$ 来修改该实例。定义 $x^1:=x_1^1\circ x_2^1\circ\cdots\circ x_t^1$，以及 $x^2:=x_t^2\circ x_{t-1}^2\circ\cdots\circ x_1^2$。OR 交叉组合构造实例 (x^1,x^2,k)。

可以仿照引理 4.1 证明该归约的正确性。

该 OR 交叉组合可在 $\widetilde{O}(t\cdot\max_{i\in t}n_i\cdot(\max_{i\in[t]}d_i)^{O(1)})=\widetilde{O}(t\max^{i\in[t]}n_i)$ 时间内计算，并且需要 $k=\max_{i\in t}n_i\cdot(\max_{i\in t}d_i)^{O(1)}=\widetilde{O}(\max_{i=1}n_i)$（这里我们利用了$d_i\in O(\log n_i)$）。因此，它是一个 (1,1)-OR 交叉组合。◀

结合命题 3.7、引理 4.1 和引理 4.5，我们得到以下结果：

▶ **推论 4.6**。除非 *SETH* 不成立，否则不存在一个 $\widetilde{O}(n^\gamma + k^\beta)$时间的算法，能够求解以解大小 k 为参数的最长公共子序列或最长公共（弱）递增子序列问题，其中 $1 < \gamma < 2$ 和 $\beta < {}^{\gamma}_{\gamma-1}$。

除非 *SETH* 不成立，否则对于以解大小为参数的最长公共子序列或最长公共（弱）递增子序列问题，不存在运行时间为 $\widetilde{O}(n^\gamma)$、核大小为 $\widetilde{O}(k^\beta)$的核，其中 $1 < \gamma < 2$ *and* $\beta < {}_2{}^{\gamma}{}_{\cdot(\gamma-1)}$。

我们指出，这些运行时间下界是紧的；紧的上界可由已知的最长公共子序列 [39]或最长公共（弱）递增子序列 [45]算法的运行时间得出，前者的运行时间为 $O(kn+n\log\ n)$，后者为 $O(nk\log\log\ n+n\log\ n)$，再结合观察 1.1 即可。

4.2 计算几何

现在我们转向计算几何中的问题。我们用 dist(p, q) 表示平面上两点 p and q 之间的欧几里得距离。首先介绍一些与离散弗雷歇距离相关的定义。

对于点列表 P，一条经过 P 的游程是一个满射、非递减函数 $f_P\colon [2n] \to [n]$，满足 $f_P(1) = 1$ 且 $f_P(2n) = n$。离散弗雷歇距离的输入包含两个点列表 $P = (p_1, \ldots, p_n)$ 和 $Q = (q_1, \ldots, q_n)$。一对经过 P 和 Q 的游程 (f_P, f_Q) 称为一次遍历。我们将 $\max^{i\in[2n]}$ dist$(p_{f_P}(i)\ q_{f_Q}(i))$ 称为该遍历的宽度。若存在某个 $\ell \in [2n]$ 和 $i^* \in [n]$，使得 $f_P(\ell) = p_i$ 且 $f_Q(\ell) = i^*$，并且对于任意 $\ell' \in [j - i]$，都有 $f_P(\ell + \ell') = i + \ell'$ 且 $f_Q(\ell + \ell') = i^*$（或者 $f_P(\ell) = i^*$ 且$f_Q(\ell) = i$，并且对于任意 $\ell' \in [j - i]$，都有 $f_P(\ell + \ell') = i^*$ 且 $f_Q(\ell + \ell') = i + \ell'$），则称遍历 (f_P, f_Q) 遍历了 P的子列表 $p_i, p_{i+1}, \ldots, p_j$（或 Q的子列表 $q_i, q_{i+1}, \ldots, q_j$）。

离散弗雷歇距离可在 $O(n^2 \cdot \log\log\ n/\log^2 n)$ 时间内求解 [6]。Bring- mann [13] 给出了从正交向量[2] 到离散弗雷歇距离的线性时间归约，表明在假设 SETH 成立的情况下，对于任意 $\epsilon > 0$，离散弗雷歇距离都无法在 $O(n^{2-\epsilon})$ 时间内求解。我们利用这一归约，为参数 $\max_{i\in[2n]} |f_P(i) - f_Q(i)|$构造了一个 (1,1)-OR-组合，即最优解中$P$ 可能领先 Q （反之亦然）的最大时间步数。我们将参数 $\max_{i\in[2n]} |f_P(i) - f_Q(i)|$ 称为遍历 (f_P, f_Q) 的最大偏移 。关于离散弗雷歇距离的更全面文献综述，请参阅 Agarwal 等人 [6]。

▶ **引理 4.7.** 正交向量问题存在一种 $(1,1)$-*OR* 交叉组合，可归约到以输入点列表的长度和解中的最大偏移为参数的离散弗雷歇距离问题。

我们的 OR 交叉组合基于 Bringmann 将正交向量归约到离散弗雷歇距离的归约 [13]，该归约具有以下性质。

▶ **命题 4.8** (Bringmann [13, III.A 节])。给定一个正交向量问题的实例 $((A, B)\ n)$，可以在 $O(n)$ 时间内计算出一个离散弗雷歇距离问题的实例 $(P_A, P_B, 1)$，使得

1. P_A 和 P_B 的弗雷歇距离为 *1*，当且仅当$((A, B)\ n)$ 是一个“是”实例；**2.** P_A从 $s_A :=$ $(-\frac{1}{3}, 15)$ 处开始，并且 $s_{\bar{A}}$只出现在 P_A的起点处，

[2] Bringmann [13] 从可满足性问题进行归约，但他的归约实际上是先将可满足性问题归约到正交向量问题，再将正交向量问题归约到离散弗雷歇距离问题。

3. P_B从 s_B:=*(*$-\frac{1}{3}$*, 0)*处开始；*4.* P_B 中的每个点到 c_A := *(0, 1 3)*的距离至多为 *1*；*5.* P_A 中的每个点到 c_B := s_B的距离至多为 *1*；*6.* ℓ_A:=*(*$-\frac{4}{3}$*, 0)*到 $Q \setminus \{s_B\}$中的所有点的距离都大于 *1*；*7.* ℓ_B:=*(*$-\frac{4}{3}$*, 1 5)*到 $P_A \setminus \{s_A\}$中的所有点的距离都大于 *1*。*8.* P_B 中的一个点到 s_A的距离大于 *1*。

引理 4.7 的证明。设 $(I_1^{\mathrm{OV}} = (A_1, B_1)\ n_1) \ldots, (I_t^{\mathrm{OV}} = (A_t, B_t)\ n_t)$ 是正交向量实例。为简化记号，我们假设 $I_1^{\mathrm{OV}}, \ldots, I_t^{\mathrm{OV}}$ 的维度都相同，均为 d，且存在某个 $n \in \mathbb{N}$，使得对于所有 $i \in [t]$，都有 $n_i = n$。此外，我们假设 I_t^{OV} 的不同实例中的任意两个向量都不正交。这可以通过将维度增加 $2 \log t$ 来实现，并利用新增维度为每个 $a \in A_i$ 添加 i 的二进制编码及其“反码”（即将 0 替换为 1，反之亦然）。对于每个 $b \in B_i$，我们附加 i 的二进制编码的“反码”以及 i 的二进制编码。

我们将命题 4.8 中的归约应用于每个实例 (I_i^{OV}, n)，得到一个离散弗雷歇距离实例 $I_i^{\mathrm{DFD}} = (P_A^i, P_B^i, 1)$。我们令 $n_i^{\mathrm{DFD}} := |P_A^i|$为 P_A^i的长度。对于 $i \in [t]$，令 $s_A^i := s_A, s_B^i := s_B, c_A^i := c_A, \ell_A^i := \ell_A, \ell_B^i := \ell_B$（其中 $s_A, s_B, c_A, c_B, \ell_A$和 ℓ_B 的定义与命题 4.8 中相同）。此外，令 I_{t+1}^{DFD}为将命题 4.8 应用于任意一个正交向量问题的“No”实例所得的实例。由此，我们构造一个离散弗雷歇距离实例 $(I^{\mathrm{DFD}} = (P_A, P_B, 1), (2t + \sum_{i=1}^{t+1} n_i^{\mathrm{DFD}}, 2 \max_{i \in [t+1]} n_i^{\mathrm{DFD}}))$，其参数为点列表的长度和最大偏移，构造如下。令 $P_A := \bigcirc_{i \in [t]} (P_A^i \circ \ell_A^i \circ c_A^i) \circ P_A^{t+1}$，并令 $P_B := \bigcirc_{i \in [t+1]} (P_B^i \circ \ell_B^i) \circ c_B$。为确保 P_A 和 P_B 的点数相同，我们在 P_B末尾追加适当数量的 c_B 副本。

直观而言，该构造的思路如下：构造中的不对称性使我们能够在某个 $i \in [t+1]$ 下处于 P_B 中的 $s_B^i = c_B$ 时，遍历 P_A 的任意部分。然而，反过来则不成立，因此，尽可能遍历 P_B 是理想的。为了遍历 ℓ_B^i，我们必须在某个 $i \in [t+1]$ 下处于 s_A^j 或 ℓ_A^j 中。我们无法停留在 s_A^j 中，因为根据性质8， P_{B_i} 中至少有一个点到 s_A^i 的距离大于一。因此有两种选择：遍历包含 s_A^j, ℓ_B^i，其中 $i, j \in [t+1]$。由于性质8迫使我们离开 s_A-点，采用此选项时会得到 $j > i$。因此，若只采用这一选项，最终会陷入僵局，因为最后一个 $\ell_{B_{t+1}}$ 无法遍历。为避免这种情况并在 P_B 中“领先”，我们必须在某个时候采用第二种选择：遍历包含 ℓ_A^i, ℓ_B^i，其中某个 $i \in [t]$。我们证明，在这种情况下， P_{A_i} 和 P_{B_i} 必须同时遍历，也就是说， I_i^{DFD} 是一个是实例。第二种选择的好处是，我们可以先遍历 (ℓ_A^i, ℓ_B^i)，再遍历 (c_A^i, s_B^{i+1})，然后遍历 $P_{B_{i+1}}$。这样一来，我们在 P_B 中便“领先”，之后就可以在遍历剩余部分时采用第一种选择。

现在我们证明 OR 交叉组合的正确性。

▷ 命题 4.9. 若存在某个 $i^* \in [t]$，使得存在彼此正交的 $a \in A_{i^*}$ 和 $b \in B_{i^*}$，则存在一个遍历 T，其对应于 (P_A, P_B)，宽度至多为 1，最大偏移为 $2 \max_{i \in [t+1]} n_i^{\mathrm{DFD}}$。

证明。我们定义如下遍历：从（s_A^1， s_B^1）开始。对于每个 $i < i^*$，我们定义以下遍历 T_i（先遍历 $P_A^i \circ \ell_A^i \circ c_A^i$，之后遍历 $P_B^i \circ \ell_B^i$）：遍历 P_A，直到到达 P_A^i的末尾。根据性质 5，到目前为止，该遍历的宽度为 1。之后，前往（ℓ_A^i， s_B^i），接着前往（c_A^i， s_B^i）。然后遍历 P_B^i。根据性质 4，该遍历的宽度仍为 1。我们继续前往（s_A^{i+1}， ℓ_B^i）和（s_A^{i+1}， s_B^{i+1}）。

根据性质1，存在一个宽度为 1 的（$P_A^{i^*}$，$P_B^{i^*}$）的遍历。我们继续进行此遍历。我们继续前往（$\ell_A^{i^*}$，$\ell_B^{i^*}$）和（$c_A^{i^*}$，$s_B^{i^*+1}$）。然后，我们遍历 $P_B^{i^*+1}$。接着是（$s_A^{i^*+1}$，$\ell_B^{i^*+1}$）和（$s_A^{i^*+1}$，$s_B^{i^*+2}$）。因此，我们现在位于 $P_A^{i^*+1}$的起点，在 P_A中，以及 $P_B^{i^*+2}$的起点，在 P_B中。

之后，对于 $t \geq i > i^*$，我们按照第一步中的方式遍历 P_A^i 和 P_B^{i+1}：首先，遍历 P_A^i，直到 P_A^i的末尾。根据性质 5，到目前为止，这次遍历的宽度为 1。之后，前往 (ℓ_A^i，s_B^{i+1})，接着前往 (c_A^i，s_B^{i+1})。然后遍历 P_B^{i+1}。根据性质 4，这次遍历的宽度仍为 1。接下来前往 (s_A^{i+1}，ℓ_B^{i+1}) 和 (s_A^{i+1}，s_B^{i+2})。

最后，我们到达 s_A^{t+1}并来到 P_B^{t+1}的末尾。我们前往 (s_A^{t+1}，ℓ_B^{t+1})，接着前往 (s_A^{t+1}，c_B)，然后遍历 P_A^{t+1}。根据性质 5，构造出的遍历宽度仍为 1。

因此，我们找到了一个遍历 (P_A, P_B)，其宽度至多为 1。容易验证，该遍历的最大偏移为 $2\max_{i\in[t+1]} n_i^{\mathrm{DFD}}$。◁

▷ 断言4.10. 如果不存在这样的 $i^* \in [t]$，使得存在正交的 $a \in A_{i^*}$ 和 $b \in B_{i^*}$，那么 P_A 和 P_B 的弗雷歇距离大于 1。

证明。 反设存在一个宽度为 1 的遍历 T。首先，我们证明对于每个 T<style id='9}，在到达 s_B^i的第一个点 P_B^i之前，都不会到达 $i \in [t+1]$ 的第一个点 s_A^i。 P_A^i。

反设 s_B^i 在 s_A^i 之前到达，且 i 是最小的。显然 $i > 1$，因为 T 以 (s_A^1,s_B^1) 开始。根据 i 的选取，我们知道 s_A^{i-1} 在 s_B^{i-1} 之前或同时到达。注意，在到达 s_B^i 之前（而 s_B^i 在 s_A^i 之前到达），必须先到达 ℓ_B^{i-1}。根据性质 7， P_A 中与 ℓ_B^{i-1} 的距离至多为 1 的点只有 ℓ_A^j 和 s_A^j，其中 $j \in [t]$。根据 i 的选取可知，(s_A^{i-1}, ℓ_B^{i-1}) 或 ($\ell_A^{i-1}, \ell_B^{i-1}$) 是 T 的一部分。我们分别证明这两种情况都会导致矛盾：由于 s_A^{i-1} 在 s_B^{i-1} 之前或同时到达，根据性质8，(s_A^{i-1}, ℓ_B^{i-1}) 不属于 T。注意，($\ell_A^{i-1}, \ell_B^{i-1}$) 不可能属于 T，因为其直接前驱由 P_{A_i} 和 P_{B_i} 中对应的末点组成：由于 s_A^{i-1} 在 s_B^{i-1} 之前到达，这将意味着 P_{A_i} 和 P_{B_i} 的弗雷歇距离至多为 1，这与性质1以及 P_{A_i} 和 P_{B_i} 不包含正交向量相矛盾。

因此，对于每个 $j \in [t+1]$，都有 T在 s_B^j之前到达 s_A^j。性质1意味着，不存在宽度为 1 的遍历，既能到达 P_B^{t+1} 的最后一个点，又能到达 P_A^{t+1}的最后一个点。因此， T必须在仅部分遍历 P_A^{t+1}的情况下到达 P_B^{t+1}的最后一个点。之后， T 必须移动到 ℓ_B^{t+1}，这意味着此时 T 位于 s_A^{t+1} （因为所有其他点到 ℓ_B^{t+1} 的距离都大于 1）。然而，根据性质8，P_B^{t+1} 中存在某个点，其到 s_A^{t+1} 的距离大于 1，这与 T 在停留于 s_A^{t+1} 的同时穿过 P_B^{t+1} 相矛盾。◁

~显然可以在 $O(t\max^{i\in[t]} n_i)$ 时间内计算出该 OR 交叉组合。根据断言4.9和4.10，正确性成立。根据命题 4.9，所构造实例的最大偏移为$O(n)$。因此，它是一个 (1，1)-OR-组合。◀

结合命题 3.7 和引理 4.7，可得以下结论：

▶ **推论 4.11**。 除非 *SETH* 不成立，否则不存在一个运行时间为 $\widetilde{O}(n^\gamma + \ell^\beta)$ 的算法，用于求解以最大偏移 ℓ 为参数的离散弗雷歇距离问题，其中参数为 $1 < \gamma < 2$ 和 $\beta < \frac{\gamma}{\gamma-1}$。

除非 *SETH* 不成立，否则不存在一个运行时间为 $\widetilde{O}(n^\gamma)$、大小为 $\widetilde{O}(\ell^\beta)$ 的核，用于求解以最大偏移为参数的离散弗雷歇距离问题，其中参数为 $1 < \gamma < 2$ 和 $\beta < 2^{\gamma\cdot(\gamma-1)}$。

很容易将用于计算离散弗雷歇距离的 $O(n^2)$ 算法扩展为在 $O(n\ell)$ 时间内运行，其中 ℓ 是某个最优解的最大偏移在所有解中的最小值

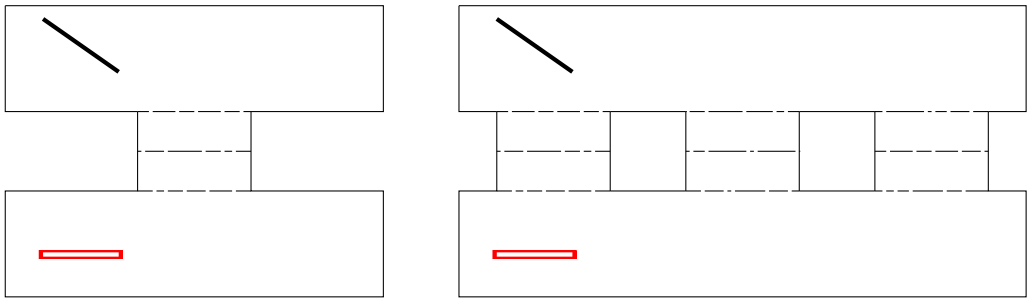


图 2 左：Gajentaan 和 Overmars [33] 通过从 3-Sum 归约构造出的平面运动规划实例示意图。右：将三个 3-Sum 实例进行 OR 交叉组合，得到一个平面运动规划实例的示例。杆的目标位置被红色区域包围。

解 [8]。结合观察 1.1 可知，推论 4.11 中的运行时间下界是紧的。

现在，我们针对计算几何中的另一个问题——平面运动规划——给出下界；该下界并非基于正交向量的困难性，而是基于 3-Sum 的困难性。平面运动规划可以在 $O(n^2)$ 时间内求解 [53]。假设 3-Sum 猜想成立，对于任意 $\epsilon > 0$ [33]，平面运动规划都无法在 $O(n^{2-\epsilon})$ 时间内求解。若两个线段之间的距离至多为杆的长度，我们就称其中一条线段位于另一条线段的邻域内。

▶ **引理 4.12**。 *3-Sum* $(1,1)$*-OR-*交叉组合归约为平面运动规划，其中参数为任一线段邻域内的线段最大数量。

证明概要。设 $I_1^{\text{3-Sum}}, \ldots, I_t^{\text{3-Sum}}$ 是 3-Sum 的实例。我们用 n_i 表示 $I_1^{\text{3-Sum}}$ 中数字的数量。Gajentaan 和 Overmars [33] 提出了一种从 3-Sum 到平面运动规划的归约：给定一个包含 n 个数字的 3-Sum 实例，该归约可在 $O(n)$ 时间内构造一个平面运动规划实例，其中杆最初位于一个较大的“上方”矩形中，并须通过一条狭窄通道到达一个“下方”矩形（图 2 以图示给出了证明）。我们将此归约应用于每个 3-Sum 实例 $I_i^{\text{3-Sum}}$，得到一个包含 $\widetilde{O}(n_i)$ 条线段的实例 I_i^{PMP}。基于这些实例，我们如下构造一个平面运动规划实例：将杆在每个 I_i^{PMP} 中起始所在的大矩形对齐。狭窄通道彼此相邻地复制（如果较大的起始矩形不够宽，就将其加宽。

该归约的正确性显而易见。

除边界框外，实例 $I_i^{\text{3-Sum}}$中的任意线段到杆的距离都至多为杆的长度。将边界框的线段拆分成许多不长于杆的小线段后，我们可以得到，对于每条线段 s，距离其不超过杆长度的线段至多有 $O(n)$ 条。◀

结合命题 3.7 和引理 4.12，可得：

▶ **推论 4.13**。 除非 *3SUM* 假设不成立，否则不存在一个运行时间为 $\widetilde{O}(n^\gamma + \ell^\beta)$ 的算法来解决平面运动规划问题，其中参数为任意线段在 $\beta < \frac{\gamma}{\gamma-1}$ 中的邻域内所拥有的线段的最大数量 ℓ。

除非 *3SUM* 假设不成立，否则对于以任一线段在其邻域内拥有的最大线段数 ℓ 为参数的平面运动规划，不存在运行时间为 $\widetilde{O}(n^\gamma)$、规模为 $\widetilde{O}(\ell^\beta)$ 的核。

根据 Sifrony 和 Sharir 提出的运行时间为 $O(K^2 \log n)$ 的算法（其中 K 是距离小于杆长的线段对数量）、观察 $K^2 \leq n \cdot \ell$ 以及观察 1.1，该运行时间下界是紧的。

4.3 图问题

现在我们转向图问题。首先，考虑以连通分量的最大规模为参数的 Minimum Weight k-团。

▶ **命题 4.14.** *Minimum Weight k-团 $(1,1)$-OR-交叉组合到以连通分量的最大规模为参数的 Minimum Weight k- 团。*

证明。 设 $G_1, \ldots, G_t$ 是 Minimum Weight k-团的实例。该交叉组合只需计算 G 的 $G_1, \ldots, G_t$ 的不交并。

显然，该交叉组合是正确的，运行时间为线性时间，且连通分量的最大规模不超过 $G_1, \ldots, G_s$ 的最大规模。◀

由于负三角形是最小权重 k-团的特殊情形，结合命题 3.7 和命题 4.14 可得以下下界：

▶ **推论 4.15.** 除非全点对最短路径假设不成立，否则不存在运行时间为 $\widetilde{O}(n^\gamma + \ell^\beta)$ 的算法，来求解以 ℓ （$1 < \gamma < 3$ 和 $\beta < 2 \cdot \frac{\gamma}{\gamma - 1}$ 的连通分量最大大小）为参数的负三角形问题。

除非全点对最短路径假设不成立，否则不存在运行时间为 $\widetilde{O}(n^\gamma)$、大小为 $\widetilde{O}(\ell^\beta)$ 的核，来求解以 ℓ （$1 < \gamma < 3$ 和 $\beta < 2 \cdot 3^{\gamma \cdot (\gamma - 1)}$ 的连通分量最大大小）为参数的负三角形问题。

运行时间为 $O(\ell^2 n)$ 的算法（其中 ℓ 是连通分量的最大大小）显然存在。结合观察 1.1 可知，运行时间下界是紧的。

接下来，我们讨论第二短路径，该问题可在 $O(mn)$ [46] 时间内求解，或者在 $O(Mn^\omega)$ 时间内求解（其中 M 是最大边权） [36]。如果图是无向的 [41]，或者目标是寻找第二短游走 [26],，则存在一个准线性时间算法。对于无权有向图，该问题可在 $O(m\sqrt{n})$ 时间内求解 [49]。可以在 $O(\frac{m}{\epsilon})$ 时间内计算出一个 ϵ-近似 [10]。

▶ **引理 4.16。** 负三角形 $(1,1)$-OR-交叉组合到以有向反馈顶点数为参数的第二短路径问题。

证明。设 $(I_1^{\mathrm{NT}}, n_1), \ldots, (I_t^{\mathrm{NT}}, n_t)$ 为负三角形问题的实例。不失一般性，我们假设每个负三角形实例都具有相同数量的 n 个顶点，即对所有 $i \in [t]$，均有 $n_i = n$，并且所有实例中的边权绝对值最大值相同。Vassilevska Williams 和 Williams [52] 给出了一个从负三角形问题到第二短路径问题的线性时间归约：给定一个负三角形问题实例 $(I_i^{\mathrm{NT}} = (G_i, w_i), n)$，按如下方式构造一个第二短路径问题实例 I_i^{2SP}：对于每个顶点 $v \in V(G_i)$，添加三个顶点 a^v、b^v 和 c^v。此外，添加两个顶点 s^i 和 t^i，以及一条路径 $s^i, p_1^i, p_2^i, \ldots, p_n^i, t^i$，其所有边的权重均为 0。此外，对于每个 $i \in [n]$，$u, v \in V(G_i)$，添加边 (p_i, a^v)、(a^v, b^u)、(b^u, c^v) 和 (c^v, p_i)。所有这些边的权重均为正（取决于 $E(G_i)$ 中的边权）。对所有 $i, j \in [t]$ 和 $j \in [\ell]$，令 s^i 与 s^j、t^i 与 t^j、p_ℓ^i 与 p_ℓ^j 分别合并，便得到一个第二短路径问题实例，其有向反馈顶点集大小为 n，即 $\{p_1, \ldots, p_n\}$。由于所构造的实例等价于析取

I_1^{2SP}、...、I_t^{2SP}（离开 s，p_1，...，p_n，t 多次不会有任何好处，因为离开这条路径的所有边权重均为正），因此我们得到了一个 (1，1)-OR 交叉组合。◀

结合命题 3.7 和引理 4.16，可得：

▶ **推论 4.17**。除非全点对最短路径假设不成立，否则不存在 $\widetilde{O}(n^\gamma + \ell^\beta)$时间的算法，可用于以有向反馈顶点集为参数的第二短路径问题 ℓ，其中参数为最大规模 ℓ 的连通分量，针对 $1 < \gamma < 3$ 和 $\beta < 2\gamma/(\gamma-1)$。

除非全点对最短路径假设不成立，否则不存在大小为 $\widetilde{O}(\ell^\beta)$、时间为 $\widetilde{O}(n^\gamma)$ 的核，可用于以有向反馈顶点集为参数的第二短路径问题；该问题以最大规模为 ℓ 的连通分量为参数，针对 $1 < \gamma < 3$ 和 $\beta < 3^{2\gamma}/(\gamma-1)$。

与本文研究的其他问题不同，我们尚不清楚第二短路径问题的运行时间下界是否是紧的。

4.4 三角形集合

最后，我们考虑三角形集合问题。显然，三角形集合可以在 $O(n^3)$ 时间内解决，但假设 SETH、3-Sum 猜想、或全点对最短路径猜想 [3]成立，则不存在 $O(n^{3-\epsilon})$ 时间算法。对于这个问题，我们无法直接应用我们的框架，即找到一个从 OR 可分解问题到三角形集合的 OR 交叉组合。不过，通过将分解和组合合并为一个步骤，我们仍然可以用非常相似的方式得到一个下界。区别在于，我们在以下结果的证明中使用的三角形集合分解不满足“OR”性质。

▶**命题 4.18**。除非正交向量、*3-Sum* 或全点对最短路径假设不成立，否则不存在一个在参数化意义下、以 ℓ 和 $1 < \gamma < 3$ 的连通分量最大规模为参数的 *Negative Triangle* $\widetilde{O}(n^\gamma+\ell^\beta)$时间算法 $\beta < 2\cdot\gamma/(\gamma-1)$。

除非正交向量、*3-Sum* 或全点对最短路径假设不成立，否则对于以最大连通分量大小 ℓ 为参数的三角形集合问题，不存在 $O(n^\gamma)$ 时间、$\widetilde{O}(\ell^\beta)$ 大小的核，其中 $1 < \gamma < 3$ 和 $\beta < 2\cdot 3^\gamma\cdot(\gamma-1)$。

证明。固定 $1 < \gamma < 3$。设 G 是三角形集合的一个实例。将 $V(G)$ 划分为 $z := n^{\lambda/3+\lambda}$ 个集合 V_1，...，V_z，每个集合大小为 $q := n^{3/(3+\lambda)}$，其中 $\lambda := \beta/\gamma - 1$。对于每个 $(i,j,k) \in [z]^3$，令 $G_{(i,j,k)}$ 为由 $V_i \cup V_j \cup V_k$诱导的图。最后，令 H 为所有 $G_{(i,j,k)}$的并集。注意，H 对应于我们框架中的 OR 分解的输出。显然，$\sim G$ 有三角形集合，当且仅当 H 有。此外，H 可在 $O(q^2\cdot z^3) = \widetilde{O}(n^{3(\lambda+2)/(3+\lambda)})$ 时间内计算得出（注意，每个 z^3 图 $G_{(i,j,k)}$ 的大小为 $O(q^2)$，因为一个含有 q个顶点的图可能有 $O(q^2)$条边）。

现在假设存在一个用于三角形集合的 $\widetilde{O}(n^\gamma + \ell^\beta)$ 算法，其运行时间为 $\beta < 2\cdot\gamma/(\gamma-1)$，并将该算法应用于 H。这显然解决了 H。这一步的运行时间为 $\widetilde{O}((z^3\cdot q)^\gamma + q^\beta) = \widetilde{O}(n^{\gamma\cdot(3\cdot(\lambda+1)/(3+\lambda))} + n^{\beta\cdot 3/(3+\lambda)})$ 时间。注意，$\gamma\cdot 3\cdot(\lambda+1)/(3+\lambda) = 3\cdot\beta/\gamma/2+\beta/\gamma < 3$。此外，还有

$$\beta\cdot 3/(3+\lambda) = 3\cdot\frac{\beta}{2+\beta/\gamma} = 3\cdot\frac{\beta\gamma}{2\gamma+\beta} = 3\cdot\frac{\gamma}{2\gamma/\beta+1} < 3\cdot\frac{\gamma}{2\gamma/\left(\frac{2\gamma}{(\gamma-1)}\right)+1} = 3\cdot\frac{\gamma}{(\gamma-1)+1} = 3$$

其中，我们在不等式中使用了假设 $\beta < 2\cdot\gamma/(\gamma-1)$。因此，我们可以在 $O(n^{3-\epsilon})$ 时间内解决三角形集合问题，其中 $\epsilon > 0$。

关于核的结论可由以下观察得出：一个针对三角形集合问题、规模为 $O(\ell^\beta)$ 且运行时间为 $O(n^\gamma)$ 的核，可直接推出一个运行时间为 $O(n^\gamma + \ell^{3\cdot\beta})$ 的三角形集合算法。◀

由于 Triangle Collection 存在一个平凡的 $O(n\ell^2)$ 时间算法，因此由观察 1.1 可知，该运行时间下界是紧的。

5 结论

我们提出了一个框架，用于将条件运行时间下界推广为参数化运行时间下界，并将其应用于各种问题。除了将该框架应用于更多问题这一显而易见的任务外，未来工作还面临其他挑战。例如，我们能否得到“AND-困难”问题，从而使用类似于那些用于排除压缩 [21] 的 AND-交叉组合？此外，该框架能否调整为适用于动态问题、计数问题或枚举问题？

参考文献

1 Amir Abboud、Arturs Backurs 和 Virginia Vassilevska Williams。LCS及其他序列相似性度量的紧确困难性结果。载于第56届 IEEE 计算机科学基础年会（FOCS 2015）论文集，第59–78页。IEEE计算机学会，2015年。URL https://doi.org/10.1109/FOCS.2015.14。2 Amir Abboud、Virginia Vassilevska Williams 和 Joshua R. Wang。稀疏图中半径和直径问题的近似算法与固定参数次二次算法。载于第27届 ACM-SIAM 离散算法年会（SODA 2016）论文集，第377–391页。SIAM，2016年。URL http://dx.doi.org/10.1137/1.9781611974331.ch28。3 Amir Abboud、Virginia Vassilevska Williams 和 Huacheng Yu。匹配三角形问题及基于一个极受关注的猜想证明困难性。SIAM 计算杂志，47 (3):1098–1122，2018年。URL https://doi.org/10.1137/15M1050987。4 Amir Abboud、Karl Bringmann、Danny Hermelin 和 Dvir Shabtay。基于 SETH 的子集和与双准则路径下界。ACM算法汇刊，18(1): 6:1–6:22，2022年。URL https://doi.org/10.1145/3450524。5 Faisal N. Abu-Khzam、Sebastian Lamm、Matthias Mnich、Alexander Noe、Christian Schulz 和 Darren Strash。实用数据约简的最新进展。CoRR，abs/2012.12594，2020年。URL https://arxiv.org/abs/2012.12594。6 Pankaj K. Agarwal、Rinat Ben Avraham、Haim Kaplan 和 Micha Sharir。次二次时间内计算离散弗雷歇距离。SIAM 计算杂志，43(2): 429–449，2014年。URL https://doi.org/10.1137/130920526。7 Anadi Agrawal 和 Pawel Gawrychowski。最长公共递增子序列问题的更快次二次算法。载于第31届国际算法与计算研讨会（ISAAC 2020）论文集，LIPIcs第181卷，第4:1–4:12页。Schloss Dagstuhl - Leibniz信息学中心，2020年。URL https://doi.org/10.4230/LIPIcs.ISAAC.2020.4。8 Jérémy Barbay。离散弗雷歇距离的自适应计算。载于第25届字符串处理与信息检索国际研讨会（SPIRE 2018），计算机科学讲义第11147卷，第50–60页。施普林格，2018年。doi: 10.1007/ 978-3-030-00479-8_5。URL https://doi.org/10.1007/978-3-030-00479-8_5。

22 参考文献

9 Matthias Bentert、Till Fluschnik、André Nichterlein 和 Rolf Niedermeier。三角形枚举的参数化方面。《计算机与系统科学杂志》，103:61–77，2019年。URL https://doi.org/10.1016/j.jcss.2019.02.004。10 Aaron Bernstein。一般图中近似替代路径和K条最短简单路径的近乎最优算法。载于第二十一届 ACM-SIAM 离散算法年会（SODA）论文集，第742–755页。SIAM，2010年。URL https://doi.org/10.1137/1.9781611973075.61。11 Hans L. Bodlaender、Rodney G. Downey、Michael R. Fellows 和 Danny Hermelin。关于不存在多项式核的问题。《计算机与系统科学杂志》，75(8): 423–434，2009年。URL https://doi.org/10.1016/j.jcss.2009.04.001。12 Hans L. Bodlaender、Bart M. P. Jansen 和 Stefan Kratsch。通过交叉组合证明核化下界。《SIAM 离散数学杂志》，28(1):277–305，2014年。URLhttps://doi.org/10.1137/120880240。13 Karl Bringmann。遛狗为何如此耗时：除非 SETH 不成立，否则弗雷歇距离不存在强次二次算法。载于第55届 IEEE 计算机科学基础年会（FOCS 2014）论文集，第661–670页。IEEE 计算机学会，2014年。URL https://doi.org/10.1109/FOCS.2014.76。14 Karl Bringmann 和 Marvin Künnemann。字符串问题与动态时间规整的二次条件下界。载于第56届 IEEE 计算机科学基础年会（FOCS 2015）论文集，第79–97页。IEEE计算机学会，2015年。URL https://doi.org/10.1109/FOCS.2015.15。15 Karl Bringmann 和 Marvin Künnemann。最长公共子序列的多变量细粒度复杂性。载于第二十九届 ACM-SIAM 离散算法年会（SODA 2018）论文集，第1216–1235页。SIAM，2018年。URLhttps://doi.org/10.1137/1.9781611975031.79。16 Timothy M. Chan 和 R. Ryan Williams。确定性全点对最短路、正交向量及其他问题：快速实现 Razborov-Smolensky 去随机化。ACM算法汇刊，17 (1):2:1–2:14，2021年。URL https://doi.org/10.1145/3402926。17 Yijia Chen、Jörg Flum 和 Moritz Müller。核化及其他预处理过程的下界。《计算系统理论》，48(4):803–839，2011年。18 Thomas H. Cormen、Charles E. Leiserson、Ronald L. Rivest 和 Clifford Stein。《算法导论》，第3版。麻省理工学院出版社，2009年。ISBN 978-0-262-03384-8。URL http://mitpress.mit.edu/books/introduction-algorithms。19 David Coudert、Guillaume Ducoffe 和 Alexandru Popa。有界团宽图若干类别上的完全多项式 FPT 算法。载于 ACM算法汇刊，15(3):33:1–33:57，2019年。URL https://doi.org/10.1145/3310228。20 Holger Dell 和 Dieter van Melkebeek。除非多项式时间层级坍缩，否则可满足性问题不允许非平凡稀疏化。《ACM 学报》，61(4):23:1–23:27，2014年。URL https://doi.org/10.1145/2629620。21 Andrew Drucker。经典与量子实例压缩的新界限。SIAM 计算杂志，44(5):1443–1479，2015年。URL https://doi.org/10.1137/130927115。22 Guillaume Ducoffe。有界团宽图上近线性时间的最大匹配算法。载于第16届参数化与精确计算国际研讨会（IPEC 2021）论文集，LIPIcs第214卷，第15:1–15:17页。Schloss Dagstuhl - Leibniz信息学中心，2021年。URL https://doi.org/10.4230/LIPIcs.IPEC.2021.15。

23 Lech Duraj。最长公共递增子序列问题的亚二次算法。载于第37届国际计算机科学理论方面研讨会（STACS 2020）论文集，LIPIcs第154卷，第41:1–41:18页。Schloss Dagstuhl - Leibniz信息学中心，2020年。URL https://doi.org/10.4230/LIPIcs.STACS.2020.41。24 Lech Duraj、Marvin Künnemann和Adam Polak。最长公共递增子序列的紧条件下界。Algorithmica，81(10):3968–3992，2019年。URL https://doi.org/10.1007/s00453-018-0485-7。25 Thomas Eiter和Heikki Mannila。计算离散弗雷歇距离。技术报告，1994年。26 David Eppstein。寻找k条最短路径。SIAM计算杂志，28(2):652–673，1998年。URL https://doi.org/10.1137/S0097539795290477。27 Henning Fernau、Till Fluschnik、Danny Hermelin、Andreas Krebs、Hendrik Molter和Rolf Niedermeier。可缩减参数化问题与严格多项式核化。可计算性，9(1):1–24，2020年。URL https://doi.org/10.3233/COM-180220。28 Till Fluschnik、George B. Mertzios和André Nichterlein。寻找常数规模子图的核化下界。载于第14届欧洲可计算性会议（CiE 2018）论文集，计算机科学讲义第10936卷，第183–193页。施普林格，2018年。URL https://doi.org/10.1007/978-3-319-94418-0_19。29 Till Fluschnik、Christian Komusiewicz、George B. Mertzios、André Nichterlein、Rolf Niedermeier和Nimrod Talmon。何时可以在线性时间内计算图双曲性？Algorithmica，81(5):2016–2045，2019年。URL https://doi.org/10.1007/s00453-018-0522-6。30 Fedor V. Fomin、Daniel Lokshtanov、Michal Pilipczuk、Saket Saurabh和Marcin Wrochna。针对低树宽图和矩阵的完全多项式时间参数化计算。ACM算法汇刊，14(3):34:1–34:45，2018年。URLhttps://doi.org/10.1145/3186898。31 Fedor V. Fomin、Daniel Lokshtanov、Saket Saurabh和Meirav Zehavi。核化：参数化预处理理论。剑桥大学出版社，2019年。doi: 10.1017/ 9781107415157。32 Lance Fortnow和Rahul Santhanam。NP的实例压缩与简洁PCP不可行。《计算机与系统科学杂志》，77(1):91–106，2011年。33 Anka Gajentaan和Mark H. Overmars。关于计算几何中一类 $O(n^2)$问题。计算几何，5:165–185，1995年。URL https://doi.org/10.1016/0925-7721(95)00022-2。34 Archontia C. Giannopoulou、George B. Mertzios和Rolf Niedermeier。多项式固定参数算法：区间图上最长路径的案例研究。理论计算机科学，689:67–95，2017年。URL https://doi.org/10.1016/j.tcs.2017.05.017。35 Szymon Grabowski。用于序列相似性问题的新制表法和稀疏动态规划技术。离散应用数学，212:96–103，2016年。URLhttps://doi.org/10.1016/j.dam.2015.10.040。36 Fabrizio Grandoni和Virginia Vassilevska Williams。更快的替代路径与距离敏感性预言机。ACM算法汇刊，16(1):15:1–15:25，2020年。URL https://doi.org/10.1145/3365835。37 Falko Hegerfeld和Stefan Kratsch。关于最大匹配的自适应算法。载于第46届自动机、语言与编程国际研讨会（ICALP 2019）论文集，LIPIcs第132卷，第71:1–71:16页。Schloss Dagstuhl -

24 参考文献

Leibniz信息学中心，2019年。URL https://doi.org/10.4230/LIPIcs.ICALP.2019.71。38 Monika Henzinger、Alexander Noe、Christian Schulz和Darren Strash。实践中寻找所有全局最小割。载于第28届年度欧洲算法研讨会（ESA 2020）论文集，LIPIcs第173卷，第59:1–59:20页。Schloss Dagstuhl - Leibniz信息学中心，2020年。URL https://doi.org/10.4230/LIPIcs.ESA.2020.59。39 Daniel S. Hirschberg。最长公共子序列问题的算法。《ACM 学报》，24(4):664–675，1977年。URL https://doi.org/10.1145/322033.322044。40 Yoichi Iwata、Tomoaki Ogasawara和Naoto Ohsaka。关于树深度对完全多项式FPT算法的作用。载于第35届计算机科学理论方面研讨会（STACS 2018）论文集，LIPIcs第96卷，第41:1–41:14页。Schloss Dagstuhl - Leibniz信息学中心，2018年。41 Naoki Katoh、Toshihide Ibaraki和Hisashi Mine。K条最短简单路径的高效算法。Networks，12(4):411–427，1982年。URL https://doi.org/10.1002/net.3230120406。42 Tomohiro Koana、Viatcheslav Korenwein、André Nichterlein、Rolf Niedermeier和Philipp Zschoche。真实世界图上最大匹配的数据约简：理论与实验。ACM实验算法学期刊，26:1.3:1–1.3:30，2021年。URLhttps://doi.org/10.1145/3439801。43 Tsvi Kopelowitz、Seth Pettie和Ely Porat。来自3SUM猜想的更高下界。载于第二十七届年度ACM-SIAM离散算法研讨会（SODA）论文集，第1272–1287页。SIAM，2016年。URL https://doi.org/10.1137/1.9781611974331.ch89。44 Stefan Kratsch和Florian Nelles。模分解上的高效自适应参数化算法。载于第26届年度欧洲算法研讨会（ESA 2018）论文集，LIPIcs第112卷，第55:1–55:15页。Schloss Dagstuhl - Leibniz信息学中心，2018年。URL https://doi.org/10.4230/LIPIcs.ESA.2018.55。45 Martin Kutz、Gerth Stølting Brodal、Kanela Kaligosi和Irit Katriel。计算最长公共递增子序列的更快算法。离散算法期刊，9(4):314–325，2011年。URL https://doi.org/10.1016/j.jda.2011.03.013。46 Eugene L. Lawler。计算离散优化问题中 K 优解的过程及其在最短路径问题中的应用。管理科学，18:401–405，1971/72年。ISSN 0025-1909。doi: 10.1287/mnsc.18.7.401。URL https://doi.org/10.1287/mnsc.18.7.401。47 George B. Mertzios、André Nichterlein和Rolf Niedermeier。线性时间数据约简对最大匹配的作用。Algorithmica，82(12):3521–3565，2020年。URLhttps://doi.org/10.1007/s00453-020-00736-0。48 Adam Polak。为什么很难在最长公共弱递增子序列问题上击败$O(n^2)$？信息处理快报，132:1–5，2018年。URL https://doi.org/10.1016/j.ipl.2017.11.007。49 Liam Roditty和Uri Zwick。无权有向图中的替代路径与k条最短简单路径。ACM算法汇刊，8(4):33:1–33:11，2012年。URL https://doi.org/10.1145/2344422.2344423。50 Shmuel Sifrony和Micha Sharir。二维多边形空间中杆的全新高效运动规划算法。Algorithmica，2:367–402，1987年。URL https://doi.org/10.1007/BF01840368。

51 Virginia Vassilevska Williams。关于算法与复杂性中的一些细粒度问题。载于《*2018*年里约热内卢国际数学家大会论文集》，第 3447–3487 页。World Scientific，2018年。52 Virginia Vassilevska Williams 和 R. Ryan Williams。路径、矩阵与三角形问题之间的次立方等价性。*ACM* 学报，65(5):27:1–27:38，2018年。网址 https://doi.org/10.1145/3186893。53 Gert Vegter。可见性图：用于可见性问题和运动规划的数据结构。载于《第二届斯堪的纳维亚算法理论研讨会（*Swat 1990*）论文集》，第 447 卷，计算机科学讲义，第 97–110 页。施普林格，1990年。网址 https://doi.org/10.1007/3-540-52846-6_81。54 R. Ryan Williams。通过电路复杂性加速全点对最短路径计算。*SIAM* 计算杂志，47(5):1965–1985，2018年。网址 https://doi.org/10.1137/15M1024524。