寻找最小距离保持子图：参数化研究

Kirill Simonov 卑尔根大学信息学系，挪威
Farehe Soheil 波茨坦大学哈索·普拉特纳研究所（HPI），德国**Shaily Verma** 焦特布尔印度理工学院，印度

arXiv:2603.21442v1 [cs.DS] 22 Mar 2026

摘要

给定一个图 G 和一个顶点子集 S，距离保持器 是 G 的一个子图，它能够保持 S中顶点之间的最短路径。我们区分子集范围距离保持器和成对距离保持器：前者保持 S中所有顶点对之间的距离，后者仅保持输入中给定的 S中特定顶点对之间的距离。尽管已有大量研究致力于确定距离保持器以及更广义的图伸展子图的大小上下界，但寻找最小距离保持器的计算复杂性却相对鲜少受到关注。

我们研究相应的子集距离保持子图（SDP）和成对距离保持子图（PDP）问题，并首次探讨其计算复杂性。我们针对若干自然参数，系统刻画了问题的复杂性图景，这些参数包括终端数量、解的大小、顶点覆盖和树宽。我们的主要贡献如下：

- 即使在网格子图上，PDP 和 SDP 也是 NP 困难的。此外，以终端数量为参数时，这些问题在网格子图上是 W[1] 困难的，而在完整网格上则是固定参数可解的。PDP 在顶点覆盖数为 3 的图上是 NP 困难的，而以图的顶点覆盖为参数时，SDP 是固定参数可解的。因此，顶点覆盖参数可以区分这两种变体。以终端
- 数量和图的树宽为参数时，这两个问题都是固定参数可解的。

-

2012年ACM主题分类 计算理论 → 固定参数可处理性

关键词与短语 距离保持器、网格、顶点覆盖、参数化算法

数字对象标识符 10.4230/LIPIcs...

经费支持 *Farehe Soheil*：由 HPI人工智能基础研究生院（FAI）资助。

1 引言

由 Peleg 和 Schäffer 引入的图伸展子图 [35],旨在用更小的子图捕捉图的距离结构。图伸展子图可用于加速几乎所有依赖最短路径距离的算法 [15, 3], ，并在特定领域有多种应用，例如分布式计算 [3, 20] 或计算几何 [34, 19]。因此，寻找稀疏或低权重的伸展子图始终是值得追求的研究目标 [2, 9, 8, 31, 26, 14, 1]。

理想的稀疏化器类型是距离保持器：一个（最好较小的）子图，能够精确保留原图中的最短路径。很容易看出，如果目标是保留所有成对距离，就无法减少任何边。在这种情况下，为了保留每条边两个端点之间的距离，都必须将该边纳入子图。然而，如果只考虑一个指定顶点子集（称为终端点）之间的距离，问题便再次变得耐人寻味。形式化地说，给定图 G 及其终端集合 S，我们称子集距离保持器 为图 G 的一个子图，使得对于 S 中的任意一对顶点，其距离与



© Kirill Simonov、Farehe Soheil 和 Shaily Verma；依据知识共享许可协议 CC-BY 4.0 授权。莱布尼茨国际信息学会议录，Schloss Dagstuhl – Leibniz-Zentrum für Informatik，达格施图尔出版社，德国

整个图 G中的距离相同。如果只关心某些特定的终端点对，还可以进一步细化定义：给定图 G 和顶点对集合$\mathcal{P}$，成对距离保持子图 是 G 的一个子图，能够保持$\mathcal{P}$中每对顶点之间的距离。为使这两种变体的术语保持一致，我们将$\mathcal{P}$ 中的顶点称为终端点，并记作 $S = \bigcup \mathcal{P}$。

距离保持器最早由 Bollobás、Coppersmith 和 Elkin [9] 在保留所有足够大的成对距离这一背景下提出。上述更一般的定义则由 Coppersmith 和 Elkin [16]提出并加以研究。距离保持器与图伸展子图 [9, 16], 有类似的应用：在保持相关距离不变的同时，提供通用的图压缩。因此，它们在伸展子图相关文献中受到了广泛关注。此外，稀疏距离保持器已被广泛用于构造稀疏伸展子图 [36, 21, 30, 31] 以及距离预言机 [22, 23]。因此，寻找小型距离保持器对于更一般类型的伸展子图及其他对象也具有重要意义。

与一般图伸展子图的研究类似，距离保持器领域的大多数文献都专注于获得保持器规模的紧下界和上界。在他们最初的工作中，Coppersmith 和 Elkin [16] 证明，当 $|S| = O(n^{1/4})$（在子集情形下）或 $|\mathcal{P}| = O(n^{1/2})$（在成对情形下）时，存在一个仅含线性数量边的距离保持器，其中 n 是图中的顶点数。在过去二十年里，这些界得到了改进和推广 [8, 7, 31]。在相关研究方向上，Krauthgamer 等人研究了使用图 minor 而非子图 [32]来保持终端距离。他们证明，即使图是网格，子集距离保持 minor 也至少需要 $\Omega(|S|^2)$ 个顶点；而一个简单的上界表明，任意图都存在一个规模为 $O(|S|^4)$的 minor。

虽然现有界限根据图的大小 n 或终端数量 $|S|$为距离保持器的规模提供了有力保证，但探讨以下问题也很有意义：对于给定实例，最小的距离保持器是什么？ 这一算法问题是我们工作的核心关注点。形式化地，我们定义以下两个计算问题。在子集距离保持子图（SDP）问题中，给定一个图 G、一个终端子集 S 和一个整数 k，任务是判断是否存在一个 G 的子图，其边数至多为 k，并且保持 S中顶点之间的距离。类似地，我们定义成对距离保持子图（PDP）问题，此外还给定一组终端点对 $\mathcal{P}$。注意，PDP 是 SDP 的严格推广，因为集合 $\mathcal{P}$ 可以包含 S中的所有点对。

与针对界限的广泛研究相比，计算问题 PDP 和 SDP 受到的关注相对较少。Gajjar 和 Radhakrishnan 证明，SDP 因而也包括 PDP，在一般图上是 NP 困难的；他们还根据保持子图中的分支顶点数量 [26]，给出了区间图上的其他结果。从优化的角度来看，Chlamtáč 等人针对最小权两两距离保持器 [14]给出了紧的 $O(n^{3/5+\epsilon})$-因子近似算法。Abdolmaleki 等人则针对总距离上界与距离保持器规模之间的差值，给出了类似的近似结果 [1]。总体而言，尽管该问题在一般情形下具有 NP-困难性，如何计算最小距离保持器仍是一个悬而未决的问题。

在继续介绍我们的结果之前，我们还想指出 PDP/SDP 与其他研究充分的路径打包问题之间的相似性。PDP 可以理解为寻找 $|\mathcal{P}|$ 条最短路径，使其分别连接对应的终端对，并使所得路径集合的交点尽可能多（以使用的边总数衡量）。与之自然对应的是k-不相交最短路径问题，其任务是寻找 k 条连接对应终端对的最短路径，且这些路径顶点不相交，即彼此没有交点。Lochet 最近证明，该问题在以终端对数量为参数时是 W[1]-困难的，但存在 XP 算法 [33]。Bentert 等人研究了打包 k 条顶点不相交或边不相交的最短环这一相关问题，并证明该问题存在 FPT 算法

在平 anar 图，而该问题的一个推广在一般图上是 W[1]-困难的 [4].

我们的贡献 本文着重于从参数化复杂性的视角，根据实例的性质刻画计算最小距离保持器的可处理性。我们的结果总结于表 1。

我们的研究从终端数量 $|S|$入手：由于距离保持器是通过为每对终端选择一条最短路径得到的，因此 $|S|$ 的数值会自然地影响问题的复杂性。具体而言，终端数量／终端对数量出现在已知的存在性界中，这些界既涉及距离保持子图 [16, 8, 7, 31]，也涉及图 minor [32]。另一方面，相关的终端连通性问题在终端对数量为常数时通常是多项式时间可解的，例如 k-不相交最短路径 [33]。

根据已知的结构性观察 [32, 16], ，可以很快看出，当终端数量为常数时，PDP 和 SDP 都存在多项式时间算法。换言之，以 $|S|$ 为参数时，这些问题属于 XP。但能否进一步改进运行时间？我们的第一个主要结果给出了否定答案，即使对于平面嵌入受到相当严格限制的图——网格的诱导子图——也成立。

▶**定理 1.** SDP 以终端数量为参数是 $W[1]$-困难的，即使在网格的诱导子图上也是如此。

由于 SDP 是 PDP 的特例，PDP 的 W[1]-困难性可由定理 1 直接推出。请注意，我们的归约还表明，在网格的诱导子图上，PDP 和 SDP 都是 NP 困难的。

另一方面，我们的下一个结果表明，当给定图是完整网格时，这些问题变为固定参数可解。

▶ **定理 2.** PDP 存在一个运行时间为 $4^{|S|^2} \cdot |V(G)|^{O(1)}$ 的算法，当输入图为网格时。

根据包含关系，定理 2 推出 SDP 也有相同的结果。

网格的情形看似简单，因为实例仅由终端的位置决定，但我们发现这具有迷惑性：PDP 推广了直角斯坦纳树形问题，因此即使在网格上也是 NP 困难的 [37]。此外，网格上的保距子式规模已知界并不优于一般图上的界 [32],因此，在距离保持器这一背景下，网格仍是一个具有挑战性的特殊情形。

在确定了类网格图的可解性边界后，我们转而关注下一个自然问题：对于不含大型网格的图，我们能否高效计算距离保持器？具体而言，我们可以考虑有界树宽的图——回顾一下，当且仅当一个图不含某个常数规模的网格子式时，它才具有有界树宽；而树宽与网格子式最大规模之间存在多项式依赖关系 [12]。我们给出肯定答案，证明 SDP 和 PDP 均为固定参数可解，其参数为 $|S|$ 加上输入图的树宽。

▶ **定理 3.** PDP 可由一个运行时间为 $2^{O\left((\text{树宽}+|S|)^2\right)} \cdot |V(G)|^{O(1)}$的算法求解，其中树宽是 G 的树宽。

定理 3 的结果进而引出了一个问题：对于终端数量任意的情形，仅凭树宽是否足以设计一个固定参数可解算法？遗憾的是，我们接下来的结果对此给出了否定答案，即使对于 PDP 而言，参数的限制要严格得多也是如此。

▶**定理4**。 PDP 即使在顶点覆盖数为3的二分图上也是 *NP* 困难。

XX:4 寻找最小距离保持子图：参数化研究

表 1 无权无向图上成对距离保持器和子集距离保持器的参数化复杂性。

参数	PDP	SDP
终端数量	在网格子图上是 W[1]-困难的（定理 1）	
	网格上的固定参数可解（定理 2）	
树宽 +终端数量	FPT（定理 3）	
顶点覆盖	对 vc = 3 而言是 NP 困难的（定理 4）	FPT（定理 5）
解的大小	W[1]-困难（定理 6）	

另一方面，通过分支论证，我们发现，以图的顶点覆盖数为参数时，SDP 是固定参数可解的。

▶ **定理 5.** SDP 存在一个 $2^{O(2^{\mathrm{vc}})} \cdot |V(G)|^{O(1)}$-时间算法，其中 vc 是 G 的最小顶点覆盖的大小。

因此，有界顶点覆盖的情形清楚地划分了 SDP 和 PDP。

最后，我们可以探讨是否能够高效地计算解的大小较小的距离保持器。由于解的大小始终不小于终端数量，因此这是一个严格限制更强的参数，所以定理 1 的结果并不足够。通过一个来自团问题的简单归约，我们发现，在一般图中，以解的大小为参数时，SDP 和 PDP 都是 W[1]-困难的。

▶**定理 6.** SDP 以解的大小为参数时是 $W[1]$-困难的。

然而，显然，定理 6 的结果无法推广到与定理 1 相同的图类。事实上，如果解的大小有界，那么终端数量以及它们之间的距离也有界，因此只有每个终端周围小球内的顶点才与问题相关。在网格子图中，这些小球只包含有界数量的顶点，因此可以立即得到该问题的多项式核。

相关工作 关于伸展子图文献的全面综述，请参阅 Ahmed 等人 [2] 的研究。根据标准定义，距离保持器可解释为 (1,0)-伸展子图（无伸长、无加性误差）。

Bollobás 等人引入了 D 保持器，旨在保持所有高于阈值D的距离，并为若干变体建立了紧确的渐近界，包括 Steiner 版本和有向版本 [9]。随后，Coppersmith 和 Elkin 引入了如今的标准终端对变体：成对距离保持器和源点式／子集式距离保持器，并给出了一般上界及与之匹配的下界区域，说明在何种情况下可以实现线性规模的保持子图 [16]。Bodwin 和 Williams 证明了早期上界在带权实例中的紧确性，并为成对情形建立了改进的一般上界，进一步细化了上界对终端对数量的依赖关系 [8]。最近，Bodwin 为有向保持器以及无向无权图中的稠密终端对集合（以 Ruzsa–Szemerédi 函数表示）推导出了更多界，并改进了子集式距离保持器的下界 [7]。Kogan 和 Parter 将保持器与伸展子图和跳数集的研究联系起来，进一步表明，对保持器的理解有助于设计其他距离近似和距离稀疏化对象 [31]。

寻找最小 t-伸展子图的问题，即乘法失真为 t的伸展子图，已得到更广泛的研究，也包括从参数化复杂性的角度进行的研究。该问题的 NP 困难性由 Cai [11] 以及 Peleg 和 Schäffer [35]证明。Venkatesan 等人 [38]研究了多种图类上的最小 t-伸展子图问题，包括弦图、凸二分图和分裂图，并表明其难解性或存在多项式时间算法取决于 t 的取值。Brandes 和 Handke 证明了平面图上 $t \geq 5$ 的最小 t-伸展子图问题是 NP 困难的 [10],；Kobayashi 将这一难解性推广到 $t \in \{2, 3, 4\}$，并证明当参数为输入图与伸展子图之间的边差时，存在一个固定参数可解算法 [29]。Fomin 等人将这一研究视角扩展到有向情形，证明以 t 加上边差为参数时存在固定参数可解算法，并证明在相同参数下寻找最小 t 加性伸展子图是 W[1] 困难的 [25]。

论文结构。我们在第 2 节介绍必要的预备知识。随后，在第 3 节展示关于网格及网格子图的结果。第 4 节介绍以树宽和终端数量为参数的 FPT 算法。第 5 节继续讨论顶点覆盖的相关结果。第 6 节给出关于解的大小的困难性证明。最后，我们在第 7 节总结全文并提出开放问题。

2 预备知识

首先介绍本文中使用的标准记号和概念。本文讨论的所有图都是简单、无向且无权的。对于图 $G = (V, E)$，我们还用 $V(G)$ 和 $E(G)$ 分别表示其顶点集和边集。对于顶点子集 $X \subseteq V(G)$，我们将由 X 诱导的 G 的子图记作 $G[X]$，即 $V(G[X]) = X$ 且 $E(G[X]) = \{uv \in E(G) \mid u, v \in X\}$。对于顶点 $v \in V(G)$，令 $N_G(v)$ 表示其邻域，并令 $\delta_G(v)$ 表示在 G 中与 v 相 incident 的边集。对于边 $e \in E(G)$，$G - e$ 表示从 G 中删除 e 后得到的图，即 $(V, E \setminus \{e\})$。对于顶点 $u, v \in V(G)$，令 $d_G(u, v)$ 表示 G 中 u 与 v 之间最短路径的长度。当图 G 可由上下文确定时，我们省略下标。对于路径 π，$|\pi|$ 表示其长度。我们将端点为 u 和 v 的边记作 uv。

对于整数 $n \geq 1$，我们写作 $[n] := \{1, 2, \ldots, n\}$；对于集合 X，我们用 $|X|$ 表示其基数。形式上，$(\mathcal{F}_v)_{v \in V(G)}$ 表示一个集合向量，其中第 v 个分量是 $\mathcal{F}_v$。我们采用括号和下标记法，以清楚地强调索引集。

对于子图 $H \subseteq G$，H 的大小（记作 $|H|$）是 H 中边的数量。

我们还回顾参数化复杂性中的相关概念。更详尽的解释和细节请参见 [17]。

参数化复杂性。参数化问题是指每个实例都带有一个参数 k 的判定问题。若某个算法能在时间 $f(k) \cdot \text{poly}(|I|)$ 内解决该问题，其中 f 是某个可计算函数，则称该算法为固定参数可解（FPT）算法。如果一个问题是 $W[1]$-难的，那么它不太可能存在 FPT 算法。

从一个参数化问题 A 到另一个参数化问题 B 的参数化归约 是一种多项式时间变换，它将 A 的一个实例（I，k）映射为 B的一个实例（I'，k'），并满足：(i)（I，k）是一个是实例，当且仅当（I'，k'）是一个是实例；(ii) $k' \leq g(k)$，其中 g是某个可计算函数。它与经典多项式时间归约的关键区别在于，目标实例中的参数由原始参数的某个函数所界定。

许多固定参数可解算法利用有界搜索树，其中递归分支由参数引导，以确保分支总数和树的深度均被某个可计算函数 f 所界定，界限为$f(k)$。

有了上述记号，我们正式定义成对距离保持器问题和子集距离保持器问题。

点对距离保持子图（PDP）

输入： 一个无向图 $G=(V,\ E)$，一个无序终端对集合 $\mathcal{P}\subseteq\{\{u,\ v\}\ |u,\ v\in V,\ u\neq v\}$，以及一个整数 k。

问题： 是否存在一个子图 $H\subseteq G$，使得对于每个 $p=\{u,\ v\}\ \in\mathcal{P}$，$d_G(u,\ v)=d_H(u,\ v)$，且 $|E(H)|\leq k$?

子集距离保持子图（SDP）

输入： 一个无向图 $G=(V,\ E)$，一个顶点子集 $S\subseteq V$，称为终端，以及一个整数 k。

问题： 是否存在一个子图 $H\subseteq G$，使得对于每一对顶点 $u,\ v\in S$,$d_G(u,\ v)=d_H(u,\ v)$，以及 $|E(H)|\leq k$,

从现在起，我们将成对距离保持器的一个实例记为 $(G,\ \mathcal{P})$，将子集距离保持器的一个实例记为 $(G,\ S)$。我们分别将成对距离保持器和子集距离保持器称为 PDP 和 SDP。

注意，SDP 是 PDP 的一个特殊情形，其中 $\mathcal{P}=\left\{\{u,\ v\}\ \middle|\ \forall u,\ v\in S,\ u\neq v\right\}$。关键区别在于，PDP 允许任意终端对集合，该集合可能编码复杂的底层结构；而 SDP 始终编码 S 上的一个团。

与斯坦纳树问题不同，后者只保持终端之间的连通性，PDP 会保持最短路径，并且可能包含环。

最后，我们记录一些关于最优距离保持器的简单观察，这些观察将在全文中使用。

▶**观察 7.** 设 H 是一个大小至多为 h 的 PDP （分别为SDP）实例的可行解（距离保持器）。若存在一条边 $e\in E(H)$ 不属于任意一对终端之间的最短路径，则 $H'=\left(V(H)\ E(H)\setminus\{e\}\right)$ 也是一个大小至多为 $h-1$ 的可行解。

根据观察 7，不失一般性，我们可以假设最优 PDP（分别为 SDP）中的每条边都属于一条终端之间的最短路径。

▶**观察 8.** 设 H 是一个大小至多为 h 的 PDP （分别为SDP）实例的可行解（距离保持器）。若一个非终端顶点 $v\in V(H)$ 不属于任何终端之间的最短路径，则 $H':=(V(H)\setminus\{v\}\ E(H)\setminus\ \delta_H(v))$ 也是一个大小至多为 h 的可行解。

因此，我们可以放心地移除那些不参与任何终端间最短路径的非终端顶点。

3 网格及网格的子图

本节专门介绍我们关于网格及网格子图的结果。我们首先证明，当参数为终端数量时，网格子图上的 W[1]-困难性（定理 1）；这是本节的主要结果。随后，我们展示了具有相同

参数的网格上的 FPT 算法（定理 2）。最后，我们通过从直线型斯坦纳树形图进行归约，观察到成对距离保持子图在网格上是 NP 困难的。

3.1 按终端数量参数化的 W[1]-困难性

在本节中，我们证明当以终端数量作为参数时，SDP 是 W[1]-困难的。我们给出一个从二部多色团（BMCC）问题出发的参数化归约；该问题在以解的大小 [17] 为参数时是 W[1]-困难的。随后，我们扩展这一归约，证明当以终端对数量作为参数时问题是 W[1]-困难的，即使输入图被限制为网格的诱导子图也是如此。

更正式地说，我们证明定理 1，并为方便起见在此重述。

▶**定理 1.** SDP 在以终端数量为参数时是 $W[1]$-困难的，即使是在网格的诱导子图上也是如此。

由于 SDP 是 PDP 的特殊情形，下面的推论可立即得到。

▶**推论 9.** PDP 以终端对的数量为参数时是 $W[1]$-困难的，即使在网格的诱导子图上也是如此。

我们首先提醒读者什么是二部多色团（BMCC）问题。

二部多色团（BMCC）

输入： 一个二分图 $G = (L \cup R, E)$、一个整数 k，以及将 L 和 R 各自划分为 k 个独立集所得的划分 $(L_1, L_2, \ldots, L_k)$ 和 $(R_1, R_2, \ldots, R_k)$。

问题： 是否存在一个集合 $X \subseteq L \cup R$，使得 $|X \cap L_i| = 1$ 且对于每个 $i \in [k]$ 都有 $|X \cap R_i| = 1$，并且 $G[X]$ 是二部团（二部团）？

已知以解的大小为参数的 BMCC 问题是 W[1]-困难的。这直接源于对多色团（MCC）问题的一种标准参数化归约；该问题是 W[1]-困难的 [17]。给定一个 MCC 实例，我们复制输入图的顶点集，并将每个顶点与其副本以及其邻居的副本相连。显然，BMCC 实例中的一个 $2k$ 色团对应于 MCC 实例中的一个 k 色团，反之亦然。

请注意，将顶点集划分为独立集可以视为一种适当的顶点着色，其中每个独立集对应一种不同的颜色。因此，问题的解是一个恰好包含每种颜色各一个顶点的二部团，这也解释了“多颜色二部团”这一名称。在整个归约过程中，我们用 颜色类 指代集合 L_i 和 R_j，其中 $i,\ j \in [k]$。不失一般性，假设对于每个$i,\ j \in [k]$ 和 $p \in \mathbb{N}$，$|L_i| = |R_j| = p$；否则，我们可以按需在较小的集合中加入孤立虚拟顶点。定理 1 的证明分两步进行。首先，我们在第 3.1.1 小节中描述一种核心归约，它在多项式时间内将一个实例 I_{BMCC} 映射为一个平面图 G' 上的 SDP 实例及其终端集合 S。然后，在单独的嵌入步骤中，我们说明如何修改该构造，使 G' 成为网格图的子图，同时保留相关的距离性质。

3.1.1 核心归约

在本小节中，我们首先介绍从 BMCC 到 SDP 的核心归约。然后，我们证明所构造图中关于最短路径的结构以及子集距离保持器的若干重要性质。最后，我们证明归约的正确性。

XX:8 寻找最小距离保持子图：参数化研究

构件构造。

设 $I_{\mathrm{BMCC}} = (G = (L \cup R,\ E)\ k)$ 是一个 BMCC 实例，其中有 $|L \cup R| = n$、 $L = \bigcup_{i=1}^{k} L_i$ 和 $R = \bigcup_{j=1}^{k} R_j$。我们构造一个 SDP 实例 $I_{\mathrm{SDP}} = (G',\ S)$，使得 $|S| = 4k$ 和$|V(G')|,\ |E(G')| \in O(n^2k + nk^2)$，具体如下。

我们创建两种构件来表示颜色类：对于每个 $i \in [k]$ ，创建一个垂直放置的 L_i-列构件，以及一个水平放置的 R_i-行构件。这些构件容纳原图的顶点，其边界由终端顶点定义。列构件与行构件之间的交互将对 G的边进行编码。

令 Δ、 α 和 ℓ 为长度参数，稍后确定其取值，使得：

$$\alpha \ll \ell \ll \Delta \quad \text{with} \quad \alpha, \ell, \Delta \in O(k + |V(G)| + |E(G)|).$$

对于每个 $i \in [k]$，我们通过向G'添加两个终端顶点 l_i^{top} 和 l_i^{bot}，来初始化 L_i-列构件。然后，对于每个顶点 $v \in L_i$，我们引入两个顶点 w_v^{top} 和 w_v^{bot}，并用一条长度为 ℓ 的路径将它们连接起来。接下来，我们分别用一条长度为 Δ 的路径连接 w_v^{top} 与 l_i^{top} ，以及 w_v^{bot} 与 l_i^{bot}。

类似地，对于每个 $i \in [k]$，我们通过向 G' 添加两个终端顶点 r_i^{left} 和 r_i^{right} ，来初始化 R_i-行构件。对于每个顶点 $u \in R_i$，我们引入两个顶点 w_u^{left} 和 w_u^{right}，并用一条长度为 ℓ 的路径将它们连接起来。我们分别用一条长度为Δ的路径连接 w_u^{left} 与 r_i^{left}，以及 w_u^{right} 与 r_i^{right} 。

我们将终端点 $\{l_i^{\mathrm{top}},\ l_i^{\mathrm{bot}},\ r_i^{\mathrm{left}},\ r_i^{\mathrm{right}} \mid i \in [k]\}$ 称为端点，因为它们充当行构件和列构件的端点。

为了确保能够控制最短路径并便于后续的网格嵌入，我们引入四个连接器顶点： c^{top}， c^{bot}， c^{left} 和 c^{right}. 。随后，我们通过一条边使构件端点与其对应的连接器相邻。也就是说，对于每个 $i \in [k]$:

- 我们通过一条边连接 c^{top} 和 l_i^{top} ，并通过一条边连接 c^{bot} 和 l_i^{bot}。
- 我们通过一条边连接 c^{left} 和 r_i^{left}，并通过一条边连接 c^{right} 和r_i^{right} 。

F最后，我们以环状方式连接连接器顶点： $c^{\mathrm{top}} - c^{\mathrm{left}} - c^{\mathrm{bot}} - c^{\mathrm{right}} - c^{\mathrm{t}}$
操作与
长度为 2Δ的路径。
I_{SDP} 的终端集合定义为

$$S = \{l_i^{\mathrm{top}}, l_i^{\mathrm{bot}}, r_i^{\mathrm{left}}, r_i^{\mathrm{right}} \mid i \in [k]\},$$

换言之，对于 G 中的每个颜色类， G'中都有两个终端点。至此，构件框架和 G'的顶点集便构造完成。构件示意图及构造概览请参见图 1。

接下来，我们将描述这些构件如何相互作用，以表示 G的边集。

对于 $u \in R$ 和 $v \in L$，令 π_u 为从 w_u^{left}到 w_u^{right}的水平路径，并令 π_v 为 G'中从 w_v^{top}到 w_v^{bot}的垂直路径。

在我们的构造中，每条水平路径 π_u 都与每条垂直路径 π_v相交。我们根据 G中的邻接关系来定义这些交点区域的局部图结构。

对于每一对 $(u, v) \in R \times L$，顶点 $z_{uv}^1, z_{uv}^2, z_{uv}^3$ 被定义为垂直路径 π_v 上沿自下而上的顺序连续排列的三个顶点。然后， π_u 与π_v 之间的交点定义如下:

- **情况** $uv \in E(G)$：在这种情况下， π_u 和 π_v 共享一段长度为 2 的公共线段。具体来说，它们在连续的三个顶点 $z_{uv}^1, z_{uv}^2, z_{uv}^3$ 处相交，并共享边 $z_{uv}^1 z_{uv}^2$ 和$z_{uv}^2 z_{uv}^3$。经过这段共享线段后， π_u 继续向 π_v 外延伸两条边。

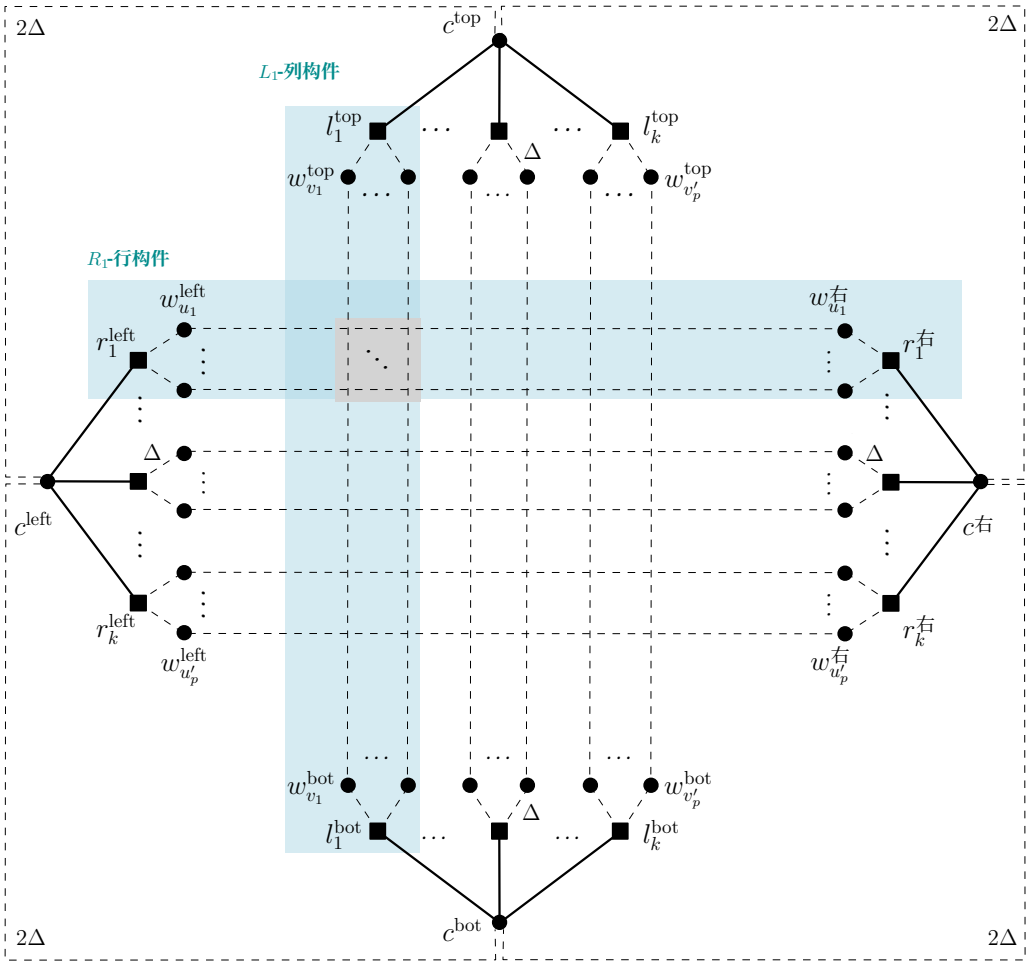


图 1 构造的总体概览。终端顶点以正方形表示，非终端顶点以圆形表示。实线表示一条边，虚线表示长度大于一的路径。对于 BMCC实例中的每个颜色类，我们都在 SDP实例中创建一个对应的构件。水平 R_k行构件和垂直 L_k列构件以浅蓝色突出显示。每个 L_i列构件在其端点处有两个终端点 l_i^{top} 和 l_i^{bot}，每个 R_i行构件也同样在其端点处有两个终端点 r_i^{left}和 r_i^{right} 。所有连接连接器顶点的路径长度均为 2Δ。在每个 L_i列构件中，对于每个 $v \in L_i$，w_v^{top}与 w_v^{bot}之间的路径长度为 ℓ，w_v^{top}与 l_i^{top} 之间的路径长度为 Δ，w_v^{bot}与l_i^{bot}之间的路径长度也为 Δ。类似地，在每个 R_i行构件中，对于每个$u \in R_i$，相同的长度性质均成立。注意，构件的端点通过一条边连接到相应的连接器顶点。如图所示，在 G 中，有 $u \in R_1$、$u' \in R_k$、$v \in L_1$和 $v' \in L_k$。灰色突出显示的区域表示 R_k行构件与 L_k列构件中的路径相交的区域，这对应于原图 G中 R_k 与 L_k 之间的边。

情况 $uv \notin E(G)$：在这种情况下，π_u 的路径经过特殊设计，以避免与 π_v共享边。具体来说，π_u 在 π_v之前恰好“向上走两条边”，与 π_v 仅在一个顶点 z_{uv}^2处相交，然后继续向 π_v外延伸一条边。这样可以确保当 $uv \notin E(G)$时，π_u 和 π_v 之间没有共享边。

我们将这个三元组 $z_{uv}^1, z_{uv}^2, z_{uv}^3$称为路径 π_u 和 π_v的局部交区域。如果 $uv \in E(G)$，那么 π_u 收集边 $z_{uv}^1 z_{uv}^2$ 和 $z_{uv}^2 z_{uv}^3$（因此 π_u 和 π_v 共享一条长度为 2 的线段）；而如果 $uv \notin E(G)$，那么 π_u 仅在中间顶点 z_{uv}^2处与 π_v 相交。

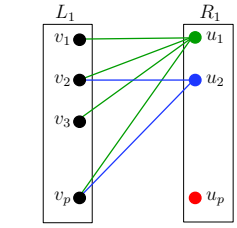
XX:10 寻找最小距离保持子图：参数化研究

固定 $v \in L$，并考虑水平路径 $\{\pi_u : u \in R\}$ 从上到下与 π_v 相交的顺序。对于该顺序中任意两条相邻的 水平路径 π_{u_1} （上方）和 π_{u_2} （下方），我们通过在 $z^3_{u_1 v}$ 和 $z^1_{u_2 v}$之间插入一条长度为 α 的路径来细分 π_v。等价地，沿着 π_v，局部交区域呈现为彼此不相交的三元组，并由长度为 α的内部顶点不相交子路径隔开。

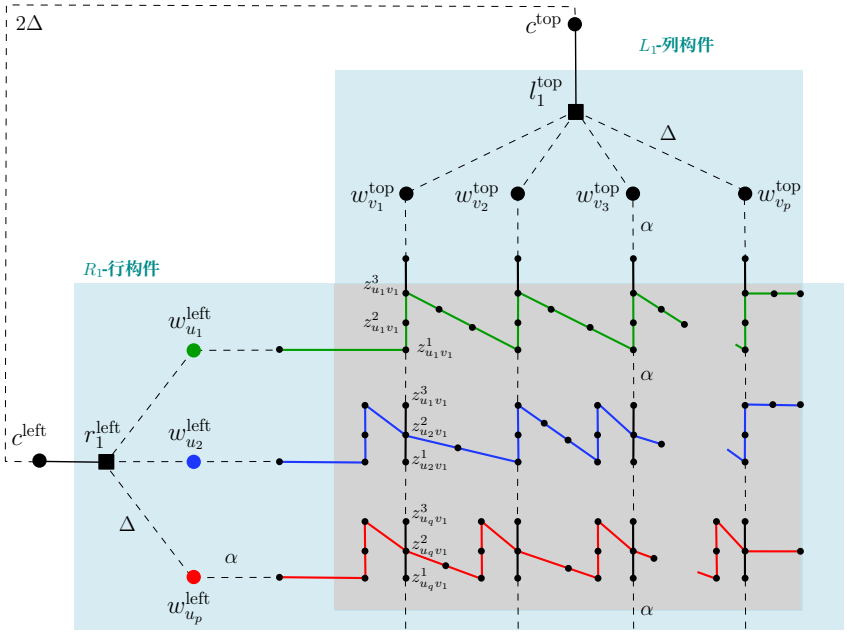
此外，在第一个局部交区域之前和最后一个局部交区域之后，我们通过插入一条长度为 α 的路径，将每个 π_u 和每个 π_v 细分。

构件交叉及其长度的示意图见图 2b。

关键的是，我们构造这些局部交点，使得每条水平路径 π_u 恰好前进五条边，就能与 π_v的一条垂直路径交叉或从其旁经过，而这与边 uv 是否存在于 G 中无关。此外，每条垂直 π_v 上的长度为 α 的路段确保最短路径不会因在不同的水平/垂直路径和构件之间切换而获益。请注意， G 中边的存在只会影响交点的拓扑结构 （共享边还是单顶点交叉），而不会影响每条垂直或水平路径的长度。



（a） $G[L_1, R_1]$ 与$L_1 = \{v_1, \ldots, v_p\}$ 和$R_1 = \{u_1, \ldots, u_p\}$。



（b） R_1行构件与 L_1列构件之间的局部交叉模式。

图 2 同一 L_1–R_1 交互的两种视图：（a）底层的导出二部图 $G[L_1, R_1]$，以及（b） G'中对应的局部交叉结构。请注意，所有水平路径都在各自对应的局部交区域与所有垂直路径相交。由于 u_1 与L_1的所有顶点相连，在 G中，其在 G' 中对应的水平路径与 L_1列构件中的所有垂直路径共享长度为 2 的路段。相反， u_p 与 L_1中的任何顶点都不相连，因此其水平路径仅在一个顶点处与每条垂直路径相交。顶点 u_2 的交互模式则是混合的：它与对应于 v_2 和 v_p 的垂直路径共享长度为 2 的路段，而与其他路径仅在单个顶点处相交。沿每条垂直路径，各局部交区域之间由长度为 α的路径隔开。此外，每条垂直和水平路径在起点处（以及终点处）都由 α 细分。

至此，图 G' 、终端集合 S 以及 SDP 实例$I_{\text{SDP}} = (G', S)$的构造完成。

为了证明该归约关于 I_{BMCC} 的规模是多项式的，首先我们设置参数 α、 ℓ 和 Δ。

首先注意，在每对相邻的局部交区域之间的每条垂直路径上都会插入 α，以防止水平路径通过使用垂直路径的子路径进行切换，从而缩短路径长度。由于每条水平路径经过每条垂直路径都需要付出 5 条边的代价，通过设置 $\alpha \gg 5$，我们确保切换到垂直路径的子路径会使距离变大。因此，$\alpha \in O(1)$ 已足够。

请记住，对于每个 $i \in [k]$，$|R_i| = |L_i| = p$，并令 $n := |V(G)|$。总之，由于有 n 条路径和 $2k$ 个构件，$\ell \in O(n+k)$ 对每条垂直路径都成立。不失一般性，我们可以在每条水平路径的首尾延伸，使其达到该长度。因此，令$\ell \in O(n+k)$ 即可。最后，我们设 $\Delta \in O(n+k) \gg \ell$。

现在我们对 $|S|$ 以及 G' 的大小进行界定。根据构造，

$$|S| = \left|\{l_i^{\text{top}}, l_i^{\text{bot}}, r_i^{\text{left}}, r_i^{\text{right}} : i \in [k]\}\right| = 4k.$$

在每个构件中，有 p 条路径，每条长度为 $2\Delta + \ell \in O(n+k)$，因此每个构件包含 $p \cdot O(n+k) = O(n(n+k))$ 个顶点和边。由于共有 $2k$ 个构件，所有构件的总规模为 $O(k \cdot n(n+k)) = O(n^2k + nk^2)$。根据 $\Delta \in O(n+k)$，我们得到 $|V(G')|,\ |E(G')| \in O(n^2k + nk^2)$。

G'中最短路径的结构。

在证明归约的正确性之前，我们先证明一些重要性质，用以描述 S中终端对之间在 G' 中的最短路径结构。

如下文所述，按照构造，我们假设 α 足够大，使得终端之间的任何最短路径都不会因绕行多条路径或多个构件而获益。此外，我们假设 $\ell < \Delta$，使得穿过单个行／列构件（其端点之间的长度为 $2\Delta + \ell$）严格短于绕过连接器环的一半（长度为 4Δ）。

▶ **观察 10.** 对于任意两个与同一连接器相邻的终端 s、$t \in S$，每条s–t 最短路径都会经过连接 s 和 t 与它们共同相邻的连接器的两条边。

上述观察可直接由构造得出，因为每个终端恰好通过一条边与一个连接器顶点相邻，且 $d_{G'}(s,t) = 2$.（参见图 1）

如果两个连接器顶点在连接器—连接器环上相邻，我们就称它们为连续的 。于是有以下命题：

▶ **命题 11.** 对于任意两个分别与相邻连接器相邻的终端 s、$t \in S$，每一条最短的 s–t 路径都会经过连接器环中位于这两个相邻连接器之间的部分。

证明。 考虑分别与 s 和 t 相邻的 c^{top} 和 c^{left} 连接器相邻的终端。对于任意 i，$j \in [k]$，沿连接器环中位于它们各自相邻连接器之间的部分行进，得到路径

$$l_i^{\text{top}} \to c^{\text{top}} \to c^{\text{left}} \to r_j^{\text{left}},$$

其长度为 $2\Delta + 2$。任何其他经过构件的连接这对终端的路径，都必须从垂直方向转向水平方向，因此长度至少为 $2\Delta + \alpha + \alpha$（因为每条水平路径和垂直路径在起点和终点都由 α 细分），严格大于前者。由对称性，同理可得：

$$d_{G'}(l_i^{\text{top}}, r_j^{\text{right}}) = d_{G'}(l_i^{\text{bot}}, r_j^{\text{left}}) = d_{G'}(l_i^{\text{bot}}, r_j^{\text{right}}) = 2\Delta + 2.$$

◀

▶ **命题 12.** 对于任意两组终端 $s = l_i^{top}$，$t = l_i^{bot}$或 $s = r_i^{left}$，$t = r_i^{right}$，其中 $i \in [k]$，每条最短的 s–t 路径都会完整经过对应列或行构件内部的一条完整内部路径。

证明. 对于每个 $i \in [k]$，在构件内部经过一条完整路径可得

$$l_i^{\text{top}} \to (\text{one vertical path in } L_i) \to l_i^{\text{bot}}, \quad \text{and} \quad r_i^{\text{left}} \to (\text{one horizontal path in } R_i) \to r_i^{\text{right}}$$

both 其长度为 $2\Delta + \ell$。我们对 L_i 列构件中的终端证明该命题；th e case 对于 R_i 行构件，情况是对称的。

将上述路径记为 π，并令 π^* 为 G' 中一条最短的 $l_i^{\text{top}} \to l_i^{\text{bot}}$ 路径。

假设 π^* 在不同的垂直路径之间切换，而不是沿着一条完整路径前进。根据构造，每次这样的切换要么（i）前进两条水平边，同时（在最佳情况下）仅取得一个单位的垂直进展；要么（ii）前进两条水平边，同时（在最佳情况下）仅取得两个单位的垂直进展。这两种情况都会使路径长度至少增加一条边，因此 $|\pi^*| \geq |\pi| + 1$，这就产生了矛盾。所以，在不同的垂直路径之间切换，无法得到比沿单条垂直路径前进更短的 l_i^{top}–l_i^{bot} 路径。

对于水平路径，情况更糟，因为它们之间的垂直子路径长度为 $\alpha \gg 5$，因此在不同的水平路径之间切换会使路径长度至少增加 $\alpha - 5$ 条边；由于 α 足够大，这就产生了矛盾。

M此外 b由……构成 tructi，一个ll vertical paths between l_i^{top} an以及 l_i^{bot} have th相同的 leng第 $2\Delta + \ell$。最后，任何不经过任何构件的路径都会到达一个连接器顶点，接着沿连接器环走过一半，到达对面的连接器顶点，然后进入另一个终端点；这样的路径长度至少为 $4\Delta + 2$，严格大于 $2\Delta + \ell$，因为 $\ell < \Delta$。

因此，命题成立：在 L_i列构件内部走完一条完整路径（类似地，在 R_i行构件内部走完一条完整路径）是唯一的最短路径，其长度为 $2\Delta + \ell$。◀

▶ **命题 13。** 对于任意两个终端点 $s = l_i^{top}$，$t = l_j^{bot}$ 或 $s = r_i^{left}$，$t = r_j^{right}$，其中 $i, j \in [k]$，每条 s–t 最短路径都会完整经过两个对应列构件或行构件之一内部的一条完整内部路径。

证明。 特别地，对于所有 $i \neq j$，

$$d_{G'}(l_i^{\text{top}}, l_j^{\text{bot}}) \leq 2\Delta + \ell + 2 \quad \text{via} \quad l_i^{\text{top}} \to (\text{traverse } L_i\text{-gadget}) \to l_i^{\text{bot}} \to c^{\text{bot}} \to l_j^{\text{bot}},$$
$$\text{or} \quad l_i^{\text{top}} \to c^{\text{top}} \to l_j^{\text{top}} \to (\text{traverse } L_j\text{-gadget}) \to l_j^{\text{bot}},$$

同理，

$$d_{G'}(r_i^{\text{left}}, r_j^{\text{right}}) \leq 2\Delta + \ell + 2 \quad \text{via} \quad r_i^{\text{left}} \to (\text{traverse } R_i\text{-gadget}) \to r_i^{\text{right}} \to c^{\text{right}} \to r_j^{\text{right}}$$
$$\text{or} \quad r_i^{\text{left}} \to c^{\text{left}} \to r_j^{\text{left}} \to (\text{traverse } R_j\text{-gadget}) \to r_j^{\text{right}}.$$

根据对称性，(l_i^{bot}，l_j^{top}) 和 (r_i^{right}，r_j^{left}) 这两对也同样成立。

证明与命题 12 类似：图中所示路径给出了上界，而任何替代路径要么绕过连接器环（代价为 $\geq 4\Delta + 2$），要么绕行经过额外的构件 / 缓冲区（代价为 $\geq 2\Delta + \ell + 2$，且对于水平构件至少还要经过一段 α），因此在既定参数范围内都不可能更短。◀

有了上述关于 S 中终端点对之间最短路径的所有性质，我们现在陈述 I_{SDP} 的子集保持器的两个重要性质。

▶ **Co推论 14.** 设 H 是该 I_{SDP} 实例的距离保持器。那么，根据观察 *10* 和命题 *11*，H 必须包含：

- 连接器—连接器环上的所有边，以及连接器顶点与构件端点 $S = \{r_i^{left}, r_i^{right}, l_i^{top},$
- $l_i^{bot} \mid i \in [k]\}$ 之间的所有边。

（术语。）如果 H 的任何真子图都不再是 (G', S) 的子集距离保持器，那么子集距离保持器 H 就是极小的 。

▶ **引理 15**。 设 $H \subseteq G'$ 是 I_{SDP}的一个极小子集距离保持器。那么对于每个$i \in [k]$， H 包含

- $l_i^{top} \to l_i^{bot}$列构件内部恰好一条 L_i内部路径的边，以及 R_i行构件内部恰好一条
- $r_i^{left} \to r_i^{right}$ 内部路径的边。

证明。 固定 $i \in [k]$。由于 H 保持终端距离，因此它必须保持 $d_{G'}(l_i^{\mathrm{top}}, l_i^{\mathrm{bot}}) = 2\Delta + \ell$。根据命题 12，每条最短 $l_i^{\mathrm{top}} \to l_i^{\mathrm{bot}}$路径都是通过 L_i内部的一条完整内部路径得到的，因此 H 必须包含至少一条此类内部路径的边。对 r_i^{left}和 r_i^{right} 也可应用相同的论证。

为证明唯一性，假设 H 包含 L_i内部两条不同 $l_i^{\mathrm{top}} \to l_i^{\mathrm{bot}}$路径 π_1 、 π_2的边。从 H 中删除仅属于这两条路径之一的所有边，即 π_1，得到 H'。终端之间的距离仍保持不变：终端对 $(l_i^{\mathrm{top}}, l_i^{\mathrm{bot}})$ 仍由 π_2 连接（根据命题 12）。对于任何其他最短路径使用了 π_1的终端对，仍可使用 π_2，且长度不变，因为根据构造， L_i 中的所有内部路径具有相同的长度和相同的端点。

因此， H' 仍然是一个距离保持器，这与 H的极小性矛盾。因此，每个列构件中恰好包含一条内部路径。行构件的情形对称。◀

归约的正确性。

确定所有相关的最短路径后，我们现在证明该归约的正确性。

▶ **引理 16**。 设 $I_{\mathrm{BMCC}} = \big(G = (L \cup R, E)\ k\big)$ 是一个 BMCC 实例，其中 $L = \bigcup_{i=1}^{k} L_i$ 且 $R = \bigcup_{j=1}^{k} R_j$。设 $I_{\mathrm{SDP}} = (G', S, k')$ 是如上构造的 SDP 实例。那么， I_{BMCC}有一个包含 $2k$ 个顶点的二部团，当且仅当 I_{SDP} 有一个大小至多为 $k' = 8\Delta + 4k + 4k\Delta + 2k\ell - 2k^2$的子集距离保持器。

证明。 为证明正向蕴含，假设 I_{BMCC} 有一个解 $X \subseteq L \cup R$ ，其中 $|X| = 2k$个顶点构成一个二部团 C。我们按如下方式构造一个子集距离保持器 $H \subseteq G'$ ，用于 I_{SDP} 。

对于每个 $i \in [k]$，令 $v_i^* \in L_i \cap X$ 和 $u_i^* \in R_i \cap X$ 为二部团中选出的顶点。此外，令 $\pi_{u_i^*}$ 为水平路径，它在 R_i行构件中经由 $w_{u_i^*}^{\mathrm{left}}$和 $w_{u_i^*}^{\mathrm{right}}$ 连接 r_i^{left} 与 r_i^{right} ；令 $\pi_{v_j^*}$ 为垂直路径，它在 L_j列构件中经由 $w_{v_j^*}^{\mathrm{top}}$ 和 $w_{v_j^*}^{\mathrm{bot}}$ 连接 l_j^{top} 与 l_j^{bot}。

对于所有 i， $j \in [k]$，我们将 $\pi_{u_i^*}$ 和 $\pi_{v_j^*}$ 中的所有边都纳入 H。此外，我们将连接器环上的所有边，以及连接连接器顶点与构件端点的边，都纳入 H。

为说明 H 保留了 S 中所有终端对之间的距离，注意，根据观察 10 和命题 11，连接器环与连接器—端点边（记作 cx）保留了位于同一侧或相邻两侧的终端之间的所有最短路径，其中c 为连接器，且 $x \in S$。根据命题 12 和命题 13，位于相对两侧的终端之间的最短路径，是通过完整遍历单个行／列构件实现的。令 F 为连接器顶点及终端集合 S 上的诱导子图，即 F 恰好包含连接器环以及连接器与终端之间的边（参见图 1）。由于 H 包含子图 F，并且对于 i， $j \in [k]$，它具有用于遍历行构件和列构件 R_i 与 L_j 的 $\pi_{u_i^*}$ 和 $\pi_{v_j^*}$，因此所有终端距离都在 H 中得到保留。

根据构造，连接器—连接器环总共贡献 $4 \cdot 2\Delta = 8\Delta$ 条边，而连接器与构件端点之间的连接总共为 H贡献 $4k$ 条边。对于每个

$i,\ j \in [k]$，由于 $u_i^* v_j^* \in E$（因为它们构成一个二部团），$\pi_{u_i^*}$在每个 L_j-列构件中都与垂直路径 $\pi_{v_j^*}$共享两条边，且 $j \in [k]$。

令 $E^{\cap}$ 为路径 $\pi_{u_i^*}$ 和 $\pi_{v_j^*}$ 之间所有共享边的集合，其中遍历所有 $i,\ j \in [k]$。于是 $|E^{\cap}| = 2k^2$，因为每一对 $\{i,\ j\}$ 都会贡献两条共享边。

此外，每个 $\pi_{u_i^*}$（同理，$\pi_{v_j^*}$）都有 $2\Delta + \ell$ 条边。将所有 $\pi_{u_i^*}$ 和 $\pi_{v_j^*}$ 路径的长度相加时，$E^{\cap}$ 会被计数两次。

因此，H 中的边总数为

$$|E(H)| = 8\Delta + 4k + 2k(2\Delta + \ell) - |E^{\cap}| = 8\Delta + 4k + 4k\Delta + 2k\ell - 2k^2.$$

对于反向证明，假设 I_{SDP} 有一个至多包含 $k' = 8\Delta + 4k + 4k\Delta + 2k\ell - 2k^2$ 条边的子集距离保持器 $H \subseteq G'$。不失一般性，我们可以假设 H 按包含关系极小，否则可以移除冗余边。我们证明 I_{BMCC} 有一个解，其中 $2k$ 个顶点构成一个二部团。

根据推论 14，H 必须包含连接器—连接器环上的所有边，以及将连接器顶点连接到构件端点（或终端点）的边，这恰好就是上文定义的子图 F。因此，H中已经有 $8\Delta + 4k$ 条边。

因此，除了子图 F贡献的边之外，H 至多还能包含 $4k\Delta + 2k\ell - 2k^2$ 条额外的边。根据引理 15，对于每个 $i,\ j \in [k]$在 R_i行构件中恰好包含一条水平路径 $\pi_{u_i^*}$，并在 L_j列构件中恰好包含一条垂直路径 $\pi_{v_j^*}$，它们连接相应的构件端点。每条这样的路径至少贡献 $2\Delta + \ell$ 条边，因此这些路径总共至少贡献 $4k\Delta + 2k\ell$ 条边。预算 k' 中的 $-2k^2$ 项表明，这些水平路径与垂直路径之间至少必须有 $2k^2$ 条共享边。H中有k 条水平路径和 k 条垂直路径，总共有 k^2 对；根据构造，每对路径最多共享两条边。因此，在 H中，每条水平路径$\pi_{u_i^*}$ 都必须与每条垂直路径 $\pi_{v_j^*}$共享两条边，以满足 $i \in [k]$。根据 G'的构造，只有当对每个$i,\ j \in [k]$时，边 $u_i^* v_j^*$存在于 G中，才有可能实现这一点。因此，集合 $X = \{u_i^*, v_j^* \mid i, j \in [k]\}$ 在 G中构成一个含有 $2k$ 个顶点的二部团。◀

3.1.2 嵌入网格

本小节将说明如何将第 3.1.1 小节中介绍的平面归约嵌入网格的一个子图中，同时保留必要的距离性质。该嵌入对于将一般平面图的困难性结果转移到类似网格的结构至关重要。

我们使用（整数）网格图 Γ，其顶点集为 $\mathbb{Z}^2$，且两个顶点相邻当且仅当它们的 ℓ_1-距离为 1。我们的嵌入构造出一个有限子图 $G^{\square} \subseteq \Gamma$。注意，网格的一个子图总可以经过二次细分，变成网格中大小为原来两倍的诱导子图。这一修改会使所有距离加倍，但不会改变任意两个原始顶点之间的最短路径集合。因此，只需证明一种构造网格的（不一定是诱导）子图的归约即可。

请记住，在 BMCC 问题中，起初给定一个二分图 $G = (R \cup L,\ E)$，其两部分为 $R = \bigcup_{i=1}^{k} R_i$ 和 $L = \bigcup_{i=1}^{k} L_i$，满足对于每个 $i,\ j \in [k]$，都有 $|L_i| = |R_j| = p$。在下文中，我们假设 p 和 k 均为 2 的幂；否则，可以根据需要向较小的集合添加虚拟孤立顶点。

设 G' 为核心归约（第 3.1.1 小节）中构造的平面图，并用 $G^{\square}$表示我们将在下文构造的嵌入网格的图。

首先，我们简要回顾一下构造 G' 的主要特征，这些特征与嵌入步骤相关。

图 G' 包含 k 个行构件和 k 个列构件，因此总共有 $2k$ 个构件。 G' 中的构件端点 恰好就是终端顶点

$$\{\, l_i^{\mathrm{top}},\, l_i^{\mathrm{bot}},\, r_i^{\mathrm{left}},\, r_i^{\mathrm{right}} \mid i \in [k] \,\},$$

即，每个 L_i 列构件的两个端点以及每个 R_i行构件的两个端点。

对于每个 $u \in R$， G' 的构造都会引入两个非终端顶点 w_u^{left} 和 w_u^{right}；对于每个 $v \in L$，它都会引入非终端点 w_v^{top} 和 w_v^{bot}；我们将这些顶点统称为 w-顶点。

在每个 R_i 行构件中， r_i^{left} 与 r_i^{right} 之间相关的遍历通过（平面）路径实现，这些路径旨在沿水平方向延伸；而在每个 L_i列构件中， l_i^{top} 与 l_i^{bot} 之间的遍历则通过（平面）路径实现，这些路径旨在沿垂直方向延伸。

更重要的是，回顾核心归约可知，终端到w 的距离在 G' 中的每个构件内都是一致的：对于每个 $i \in [k]$、每个 $u \in R_i$和每个 $v \in L_i$,

$$d_{G'}\big(r_i^{\mathrm{left}}, w_u^{\mathrm{left}}\big) = d_{G'}\big(r_i^{\mathrm{right}}, w_u^{\mathrm{right}}\big) = \Delta \qquad \text{and} \qquad d_{G'}\big(l_i^{\mathrm{top}}, w_v^{\mathrm{top}}\big) = d_{G'}\big(l_i^{\mathrm{bot}}, w_v^{\mathrm{bot}}\big) = \Delta.$$

这种一致性至关重要：它确保最短终端到终端路径经过某个构件时，在构件内部选择哪条水平／垂直路径都不会影响最终的终端距离。然而，在网格图中，我们无法用一条长边实现长度为Δ 的邻接关系。因此，在 $G^{\square}$的构造中，为了实现这种一致性，我们只将端点到w 的连接路径构造成根到叶长度相等、嵌入网格的二叉树；我们不 会将整个行／列构件嵌入为二叉树。

具体而言，对于每个构件端点 $t \in S$ 及其对应的、位于同一侧的 w顶点集合 W_t，我们在网格中用一棵有根二叉树替换 G' 中终端到w 的连接；该树的根为t，叶节点恰为 W_t。这意味着，与同一端点对应的所有 w顶点到该端点的距离都相等。下文中唯一因侧而异的是叶节点的布局（间距），以及由此产生的嵌入区域高度／宽度界；我们将分别讨论行构件和列构件。

为避免平面构造与网格嵌入之间产生歧义，本节始终采用以下约定：不带撇号的符号（例如 π_u、 π_v）表示 G' 中的路径，而它们在 $G^{\square}$ 中对应的网格嵌入则用带上标 $\square$ 的相同符号表示（例如 $\pi_u^{\square}$、 $\pi_v^{\square}$）。此外，本节始终将 $\Delta' \gg 6kp$ 设为长度参数。

终端点到 w 的嵌入。

行构件。 令 $W^{\mathrm{left}} = \{w_u^{\mathrm{left}} \in G' \mid u \in R\}$ 和 $W^{\mathrm{right}} = \{w_u^{\mathrm{right}} \in G' \mid u \in R\}$ 分别为 G' 中行构件左端和右端的所有 w^{left}-顶点和 w^{right}-顶点的集合。

我们将 W^{left} 的所有顶点放置在第一坐标为 0 的网格列上，即位于形如 $(0,\ y)$ 的点处。将 W^{left} 按从上到下的顺序排列为 $(w_{u_1}^{\mathrm{left}}、\ \ldots\ldots、\ w_{u_{kp}}^{\mathrm{left}})$，并为它们赋予坐标 $(0,\ y_1)\ldots\ldots,\ (0,\ y_{kp})$，其中 $y_1 = 6kp$ 满足 $y_1 > y_2 > \cdots > y_{kp}$，且对每个 $i \in [kp-1]$ 都有 $y_i - y_{i+1} = 6$，等价地

$$d_\Gamma\big((0, y_i), (0, y_{i+1})\big) = 6.$$

令 $\beta_1 = \Delta' + 4kp$；类似地，我们将 W^{right} 的所有顶点放置在第一坐标为 β_1 的网格列上，即放在形式为 $(\beta_1,\ y)$ 的点上。将 W^{right} 按从上到下的顺序排列为 $(w_{u_1}^{\mathrm{right}}、\ldots\ldots、w_{u_{kp}}^{\mathrm{right}})$，并为它们分配坐标 $(\beta_1,\ y_1)\ldots\ldots,\ (\beta_1,\ y_{kp})$。注意， W^{right} 中顶点的 y 坐标与 W^{left} 中顶点的坐标相同；也就是说，对于 $i \in [kp]$， $w_{u_i}^{\mathrm{left}}$ 和 $w_{u_i}^{\mathrm{right}}$ 被放置在相同的网格行上。

这样，由于 p 是 2 的幂，并且按上述从上到下的顺序，对应于任意固定的 R_i-行构件的顶点在 W^{left} 中构成一个恰好包含 p 个连续顶点的连续块（W右中也同样如此），我们可以分别处理每个这样的连续块。特别地，对于每个 $i \in [k]$，我们用一棵包含在高度为 $\lceil \log(6p) \rceil$、宽度为 $6(p-1)$ 的网格区域内的有根二叉树，来嵌入 r_i^{left} 与 W^{left} 中第 i 个连续块的 p 个顶点之间的连接；该树的根为 r_i^{left}，叶节点恰好是这些 p w^{left} 个顶点。类似地，我们用一棵根为 r_i^{right}、高度和宽度相同的有根二叉树，来嵌入 r_i^{right} 与 W^{right} 中第 i 个连续块之间的连接。
图 3a 展示了上述针对两个终端点 r_i^{left} 和 r_i^{right} 的嵌入示例，每个终端点均带有 8 w 个顶点。

因此，对于每个 $i \in [k]$ 以及所有 $u,\ u' \in R_i$,

$$d_{G\square}\big(r_i^{\text{left}}, w_u^{\text{left}}\big) = d_{G\square}\big(r_i^{\text{left}}, w_{u'}^{\text{left}}\big) = 6(p-1)/2 + \lceil \log(6p) \rceil \qquad \text{and}$$
$$d_{G\square}\big(r_i^{\text{right}}, w_u^{\text{right}}\big) = d_{G\square}\big(r_i^{\text{right}}, w_{u'}^{\text{right}}\big) = 6(p-1)/2 + \lceil \log(6p) \rceil. \tag{1}$$

注意，由于所有 r_i^{left} 个顶点（同样地，所有 r_i^{right} 个顶点）都是高度相同的二叉树的根，并且其对应的 w 个顶点位于同一网格列中，因此所有 r_i^{left} 个顶点（同样地，所有 r_i^{right} 个顶点）都位于同一网格列中。

我们用一条垂直直线网格路径连接所有 r_i^{left} 个顶点（同样地，连接所有 r_i^{right} 个顶点）。

列构件。令 $W_{\text{上}} = \{w_v^{\text{top}} \in G' \mid v \in L\}$ 和 $W^{\text{bot}} = \{w_v^{\text{bot}} \in G' \mid v \in L\}$ 分别表示 G' 中列构件上端和底端的所有 w^{top} 个顶点及 w^{bot} 个顶点的集合。

令 $x_1 = 1 + \Delta'/2$，以及 $\beta_2 = 3 + y_1 + \Delta'/2$。我们将 W^{top} 的所有顶点放置在第二坐标为 β_2 的网格行上，即放置在形如 $(x,\ \beta_2)$ 的点处。将 W^{top} 从左到右排序为 $(w_{v_1}^{\text{top}}$、……、$w_{v_{kp}}^{\text{top}})$，我们为它们分配坐标 $(x_1,\ \beta_2)$……、$(x_{kp},\ \beta_2)$，使得 $x_1 < x_2 < \cdots < x_{kp}$，并且对每个 $i \in [kp-1]$ 都有 $x_{i+1} - x_i = 4$；等价地，

$$d_\Gamma\big((x_i, \beta_2), (x_{i+1}, \beta_2)\big) = 4$$

令 $\beta_3 = \beta_2 - \Delta' - 6kp$；类似地，我们将 W^{bot} 的所有顶点放置在第二坐标为 β_3 的网格行上，即放置在形如 $(x,\ \beta_3)$ 的点处。将 W^{bot} 从左到右排序为 $(w_{v_1}^{\text{bot}} \ \ldots\ w_{v_{kp}}^{\text{bot}})$ x_{kp} $x_1\ \beta_3\ \ldots\ x_{kp}\ \beta_3$ W^{bot} $x_1 < x_2 <$ <、，，我们为它们分配坐标 (，)，(，)，使得 $\cdots$。注意，中的顶点的坐标与 W^{top} 中顶点的坐标相同，即对于 $i \in [kp]$，$w_{v_i}^{\text{top}}$ 和 $w_{v_i}^{\text{bot}}$ 被放置在同一网格列上。

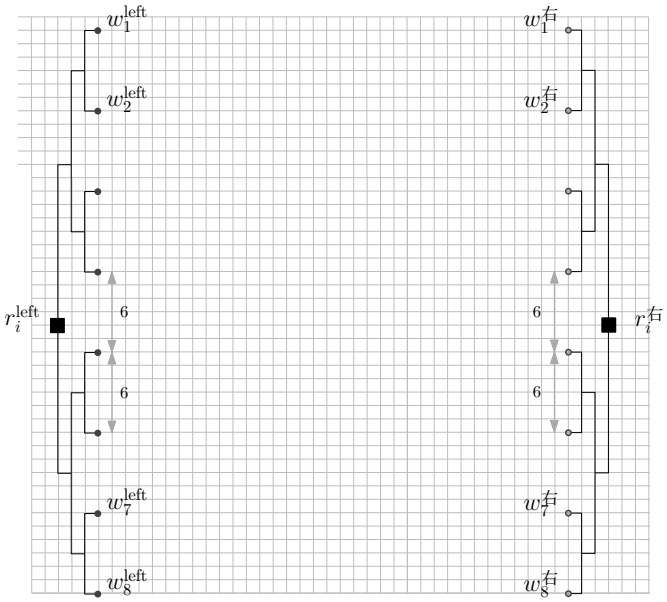
与行构件的论证相同，对应于任意固定L_i列构件的顶点，在 W^{top} 中构成一个恰好包含 p 个连续顶点的连续块（在 W^{bot}中也类似）。对于每个 $i \in [k]$，我们使用一棵包含在高度为 $\lceil \log(4p) \rceil$、宽度为 $4(p-1)$ 的网格区域中的有根二叉树，来嵌入 l_i^{top} 与 W^{top} 中第 i 个块的 p 个顶点之间的连接；该树的根为 l_i^{top}，叶节点恰好是这些 p w^{top}个顶点。类似地，我们使用一棵根为 l_i^{bot}、高度和宽度相同的有根二叉树，来嵌入 l_i^{bot}与 W^{bot} 中第 i 个块之间的连接。
图 3b 展示了上述嵌入的一个示例，其中有两个终端点 l_i^{top} 和 l_i^{bot}，每个终端点各有 8 w个顶点。
因此，对于每个 $i \in [k]$ 以及所有 $v,\ v' \in L_i$,

$$d_{G\square}\big(l_i^{\text{top}}, w_v^{\text{top}}\big) = d_{G\square}\big(l_i^{\text{top}}, w_{v'}^{\text{top}}\big) = 4(p-1)/2 + \lceil \log(4p) \rceil \qquad \text{and}$$
$$d_{G\square}\big(l_i^{\text{bot}}, w_v^{\text{bot}}\big) = d_{G\square}\big(l_i^{\text{bot}}, w_{v'}^{\text{bot}}\big) = 4(p-1)/2 + \lceil \log(4p) \rceil.$$

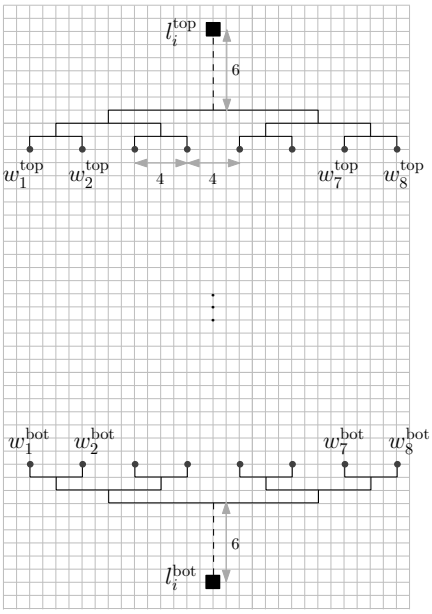
为了使终端点到 w 的距离与行构件中的距离相等（公式 (1)），对于与列构件对应的二叉树，我们通过长度适当的直线垂直网格路径连接每个根。这样，所有 l_i^{top} -顶点（同样，所有 l_i^{bot}-顶点）都位于同一网格行。

我们用一条水平直线网格路径连接所有 l_i^{top} -顶点（同样，也连接所有 l_i^{bot}-顶点）。

与“行构件”情况的推理相同，所有 r_i^{top} -顶点（同样，所有 r_i^{bot}-顶点）都放置在同一网格行上。



(a) 连接关系的二叉树嵌入示例 r_i^{left}与 r_i^{right} 及其对应的 w-顶点之间。



(b) l_i^{top} 与 l_i^{bot}及其对应的 w-顶点之间连接关系的二叉树嵌入示例。

图 3 用于实现网格嵌入中终端点到 w 连接的二叉树嵌入。在此示例中，$|L_i| = |R_i| = 8$。在水平树中，所有叶节点到各自终端点 r_i^{left}和 r_i^{right} 的距离均为 23。为了使垂直树中根到叶节点的距离也为 23，我们附加了长度为 6 的直线垂直路径，将根连接到树的其余部分。这样就使终端点到 w 的共同距离相等。

如上所述嵌入终端点到 w 的连接后，最后，我们用长度为 $1.5\Delta'$ 的简单网格路径连接以下各对构件端点，从而完成 $G^{\square}$的主框架：

$$r_1^{\text{left}} \to l_1^{\text{top}}, \quad l_k^{\text{top}} \to r_1^{\text{right}}, \quad r_k^{\text{right}} \to l_k^{\text{bot}}, \quad l_1^{\text{bot}} \to r_k^{\text{left}}$$

请注意，根据二叉树嵌入和公式 (1)，上述任意一对端点之间的网格距离至多为 $\Delta' + 6kp + 2\lceil \log(6kp) \rceil$。根据假设 $\Delta' \gg 6kp$，因此我们可以用预期长度为 $1.5\Delta'$的网格路径连接上述各对端点。整体生成的网格嵌入概览见图4。

构件的主框架就位后，我们现在将每个行／列构件的内部结构嵌入对应的网格区域。在此之前，我们先介绍一种简单的方法，用长度相同的网格路径来实现 G' 中长度为 Δ' 的路径。为方便起见，

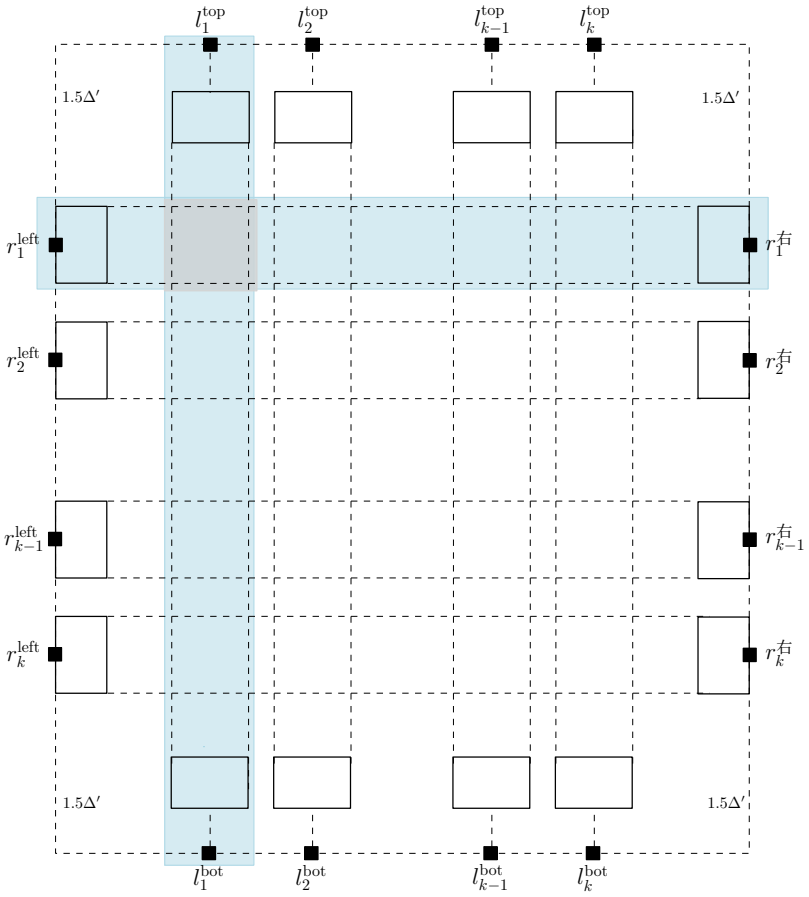


图4 G' 的网格嵌入总体示意图。终端顶点用方形表示，非终端顶点用圆形表示。虚线表示长度大于 1 的路径。水平 R_k 行和垂直 L_k 列构件以浅蓝色突出显示。每个矩形表示一棵以终端点为根、以对应一侧的 w 顶点为叶节点的二叉树：左、上、右和下。上方和下方的二叉树通过适当长度的路径连接到各自对应的根/终端点，以使距离相等。所有 r_i^{left} 顶点（同样，所有 r_i^{right} 顶点）均通过垂直直线网格路径连接。所有 l_i^{top} 顶点（同样，所有 l_i^{bot} 顶点）均通过水平直线网格路径连接。所有 $r_1^{\text{left}} \to l_1^{\text{top}}$，$l_k^{\text{top}} \to r_1^{\text{right}}$，$r_k^{\text{right}} \to l_k^{\text{bot}}$ 和 $l_1^{\text{bot}} \to r_k^{\text{left}}$ 均通过长度为 $1.5\Delta'$ 的网格路径连接。

假设 Δ' 是 4 的倍数，否则我们最多将其增加 3，这不会影响渐近界，也能保留该归约。

▶ **定义 17** (向右水平之字形路径)。 从一个网格顶点 a 出发、长度为Δ'的向右水平之字形路径，是通过恰好重复 $\Delta'/4$ 次模式$(\rightarrow,\uparrow,\rightarrow,\downarrow)$（每个方向各走一条边）所得到的任意简单网格路径。

该路径的总长度为 Δ'，其另一个端点恰好位于 a右侧 $\Delta'/2$ 个单位处（具有相同的 y坐标），也就是说，它们在 Γ 中的 ℓ_1距离为 $\Delta'/2$。类似地，向左水平之字形路径重复模式（$\leftarrow$，$\uparrow$，$\leftarrow$，$\downarrow$）。同样，长度为 Δ'的向下垂直之字形路径 （向上路径则相反）通过恰好重复 $\Delta'/4$ 次模式（$\downarrow$，$\rightarrow$，$\downarrow$，$\leftarrow$）（向上路径则重复（$\uparrow$，$\rightarrow$，$\uparrow$，$\leftarrow$））得到；其总长度为 Δ'，端点恰好位于起点下方（向上路径则位于上方） $\Delta'/2$ 个单位处。

为了将垂直路径嵌入 $G^{\square}$中，对于每个 $v \in L$，我们首先在 w_v^{top}上连接一条长度为 Δ' 的向下垂直之字形路径，并将其端点记为 t_v^{top}。对称地，我们在 w_v^{bot}上连接一条长度为 Δ' 的向上垂直之字形路径，并将其端点记为 t_v^{bot}。于是，

$$d_{G^{\square}}\big(w_v^{\mathrm{top}}, t_v^{\mathrm{top}}\big) = d_{G^{\square}}\big(w_v^{\mathrm{bot}}, t_v^{\mathrm{bot}}\big) = \Delta'.$$

然后，我们通过一条长度为 $8kp$的垂直简单网格路径 $\pi_v^{\square}$连接 t_v^{top}和 t_v^{bot}；对于每个 $u \in R$，该路径重复长度为 8 的模式（$\downarrow$，$\downarrow$，$\downarrow$，$\downarrow$，$\leftarrow$，$\downarrow$，$\rightarrow$，$\downarrow$），恰好重复 kp 次。

注意，上述模式的长度为 8，但在网格上仅前进 6 个垂直单位。根据构造，对于每个 $v \in L$，其对应的 w顶点 w_v^{top}和 w_v^{bot}位于同一网格列上，垂直距离为 $\Delta' + 6kp$，这为之字形路径和垂直路径 $\pi_v^{\square}$提供了足够空间。

为了将水平路径嵌入 $G^{\square}$，对于每个 $u \in R$，我们首先在 w_u^{left} 上连接一条长度为 Δ' 的向右水平之字形路径，并将其端点记为 t_u^{left}。类似地，我们在 w_u^{right} 上连接一条长度为 Δ' 的向左水平之字形路径，并将其端点记为 t_u^{right}。因此，

$$d_{G^{\square}}\big(w_u^{\mathrm{left}}, t_u^{\mathrm{left}}\big) = d_{G^{\square}}\big(w_u^{\mathrm{right}}, t_u^{\mathrm{right}}\big) = \Delta'.$$

然后，我们按如下方式用一条长度为 $8kp$ 的水平简单网格路径 $\pi_u^{\square}$ 连接 t_u^{left} 和 t_u^{right}。设 $\{v_1, \ldots, v_{kp}\}$ 为 L 的顶点，并根据它们在网格上对应的 w^{top}-顶点在 W^{top} 中的顺序，从左到右排列。固定 $u \in R$。对于每个 $i \in [kp]$,

- 如果 $uv_i \in E(G)$，我们将 $\pi_u^{\square}$ 延长一条长度为 8 的路径，该路径依次沿以下方向各走一条边：（参见图5a）

  $$(\rightarrow,\downarrow,\rightarrow,\uparrow,\uparrow,\rightarrow,\downarrow,\rightarrow)$$

- 如果 $uv_i \notin E(G)$，我们将 $\pi_u^{\square}$沿一条长度为 8 的路径延伸，该路径依次沿以下各个方向前进一条边：（参见图5b）

  $$(\rightarrow,\downarrow,\rightarrow,\uparrow,\rightarrow,\downarrow,\rightarrow,\uparrow)$$

注意，每种此类模式的总长度均为 8，但可产生 4 的净水平进展。由于$\pi_u^{\square}$ 为 L的每个 kp 顶点各连接一个这样的模式，因此 t_u^{left}与 t_u^{right}之间的总“核心”水平位移为 $4kp$。根据构造， w_u^{left}和 w_u^{right} 位于同一网格行上，水平距离为 $\Delta' + 4kp$，这意味着它们之间的水平距离足以容纳两条之字形路径和核心路径 $\pi_u^{\square}$。上述长度为 8 的模式经过选择，使得对于每个

固定 $u \in R$ 后，生成的网格路径 $\pi_u^{\square}$与列构件路径 $\{\pi_v^{\square}\colon v \in L\}$ 相交的左右顺序，与对应的平面路径 π_u 在 G' 中与 $\{\pi_v\colon v \in L\}$ 相交的顺序相同。



(a) P 长度为 8 的路径在 π_u 处连接，当 $uv_i \in E(G)$ 时。**(b)** 长度为 8 的路径在 π_u 处连接，当 $uv_i \notin E(G$).

图 5 根据 uv_i 是否为 G中的一条边，连接的子路径用于为每个 $v_i \in L$ 构造 π_u 。路径模式分别为 $(\rightarrow,\downarrow,\rightarrow,\uparrow,\uparrow,\rightarrow,\downarrow,\rightarrow)$ 和 $(\rightarrow,\downarrow,\rightarrow,\uparrow,\rightarrow,\downarrow,\rightarrow,\uparrow)$ 。

此外，对于每个 $i \in [kp]$， $\pi_u^{\square}$与 $\pi_{v_i}^{\square}$ 之间的局部交点仅取决于$uv_i \in E(G)$：如果 $uv_i \in E(G)$，则交点子图恰好包含两条共享网格边；而如果 $uv_i \notin E(G)$，则恰好包含一条共享网格边。

这是通过 W^{top}中相邻顶点之间固定为 4 的水平间距（ W_{bot}中也同样如此），以及 W^{left} 中相邻顶点之间为 6 的垂直间距（ W^{right}中也同样如此）来实现的；这些顶点彼此对齐，并为两种不同的长度为 8 的布线路径模式提供了充足空间。图 6b 展示了上文所述的 $G^{\square}$中的局部交点。

$G^{\square}$ 中的终端集合与 G'中的完全相同，具体为

$$S = \{\, l_i^{\text{top}},\, l_i^{\text{bot}},\, r_i^{\text{left}},\, r_i^{\text{right}} \mid i \in [k] \,\},$$

由此完成网格嵌入 $G^{\square}$ 的构造。

为了证明该归约关于 I_{BMCC} 的规模是多项式的，首先我们设定参数 Δ'。在我们的构造中，只需要 $\Delta' \gg c \cdot kp$，其中 $c > 6$。因此 $\Delta' \in O(n)$。

另外注意，所有二叉树的深度为 $O(\log(6kp)) = O(\log(n))$。因此，每棵树中的顶点数为 $O(p) = O(n)$，二叉树的规模关于 n 是多项式的。

所有路径的长度均为 $O(\Delta') + O(kp) \in O(n)$，因此网格嵌入的规模关于 n 是多项式的。

$G^{\square}$ 中最短路径的结构。

接下来，我们证明一些重要性质，用以描述 $G^{\square}$ 中终端对 S 之间的最短路径结构。我们将 $I_{\text{SDP}}^{\square} = (G^{\square},\ S)$ 定义为由嵌入图 $G^{\square}$ 和终端集合 S 诱导的 SDP 实例 。

接下来，我们假设 $\Delta' \gg 10kp$。我们根据终端在构造中的位置，将终端的边定义为“左”“上”“右”和“下”，如图4所示；相邻边指顺时针顺序中相邻的边对（例如，左边和上边相邻，而左边和右边不相邻）。此外，我们将连接这四对边、长度为 $1.5\Delta'$ 的路径称为

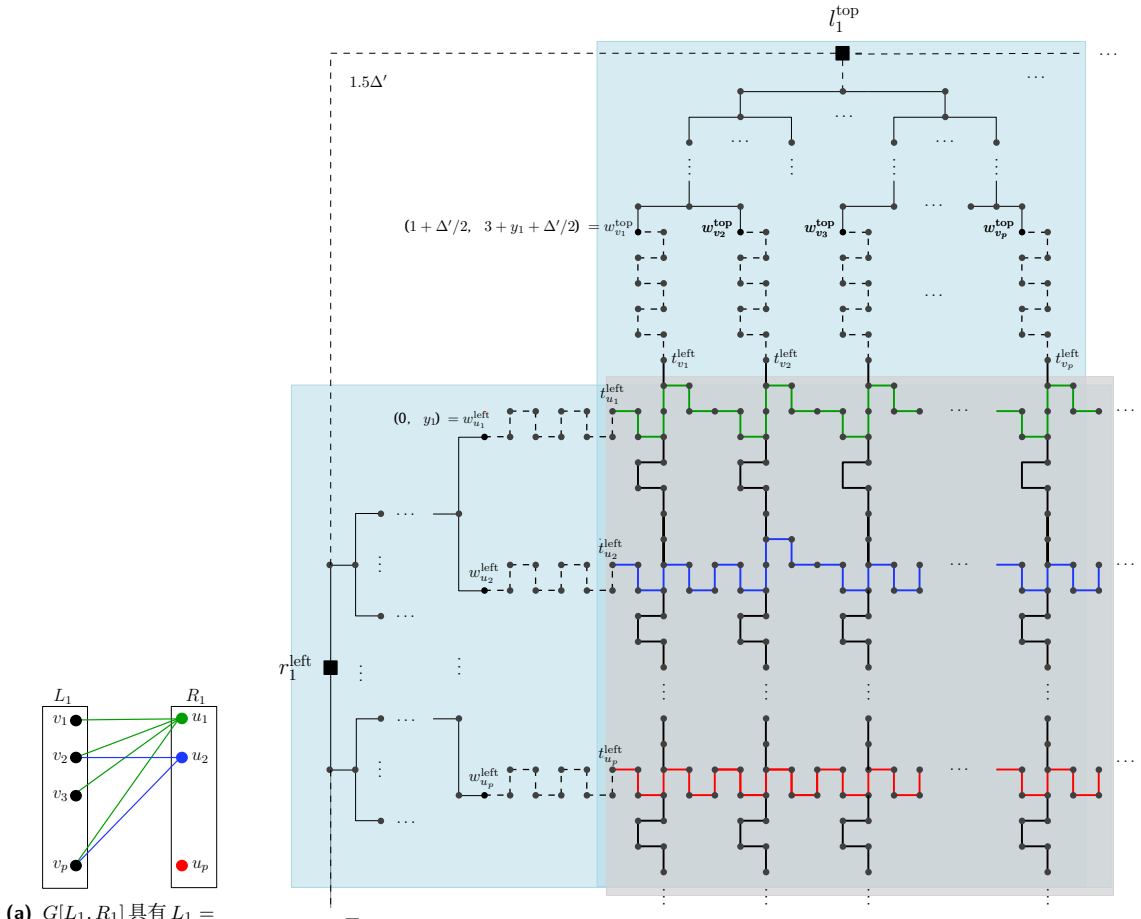
$$r_1^{\text{left}} \to l_1^{\text{top}} \quad l_k^{\text{top}} \to r_1^{\text{right}} \quad r_k^{\text{right}} \to l_k^{\text{bot}} \quad l_1^{\text{bot}} \to r_k^{\text{left}}$$

侧连接器路径。

类似地，我们将 $G^{\square}$ 中对应于 L_i 的每个垂直构件称为 L_i 列构件，并将 $G^{\square}$ 中对应于 R_i 的每个水平构件称为 R_i 行构件。

▶ **观察 18。** 对于同一边上的任意两个终端 s、$t \in S$，每条最短的 s–t 路径都会沿着连接它们的同一行/列上的直线网格路径行进，且该路径位于 $G^{\square}$。

上述观察可直接由构造得出。事实上，对于每一侧，该侧上的所有终端都位于同一网格行或同一网格列上，并且我们明确地将该线上相邻终端之间的直线网格线段纳入 $G^{\square}$ （见图4）。因此，直线 s–t



(a) $G[L_1, R_1]$ 具有 $L_1 = \{v_1, \ldots, v_p\}$ 和 $R_1 = \{u_1, \ldots, u_p\}$。

(b) $G^{\square}$中，R_1 水平构件与 L_1 垂直构件的局部交叉模式。

图6 同一 L_1–R_1 交互的两种视图：**(a)** 底层导出二部图 $G[L_1, R_1]$，以及**(b)** $G^{\square}$中对应的局部交叉结构。注意，所有水平路径都在各自对应的局部交叉区域与所有垂直路径相交。由于 u_1 与L_1的所有顶点相连，因此在 G中，它在 G' 中对应的水平路径与L_1列构件中的所有垂直路径共享长度为 2 的线段。相反， u_p 与 L_1中的任何顶点都不相连，因此它的水平路径只与每条垂直路径共享一条边。顶点 u_2 的交互模式兼有两种情况：它与对应于 v_2 和 v_p 的垂直路径共享长度为 2 的线段，而与其他垂直路径仅共享一条边。 t顶点是连接到 w 顶点的之字形路径的端点。所有水平路径和垂直路径的长度均为 $8kp + 2\Delta'$。

沿该行/列的子路径包含在 $G^{\square}$ 中，其长度为 $d_\Gamma(s,\ t)$。由于 $G^{\square} \subseteq \Gamma$， $G^{\square}$ 中任意一条 s–t路径的长度至少为 $d_\Gamma(s,\ t)$，因此这条直线子路径是$G^{\square}$ 中一条最短的 s–t 路径（参见图4）。

▶ **命题 19.** 对于位于相邻边上的任意两个终端点 $s, t \in S$，每条最短 s–t 路径都会经过它们的侧连接路径。

证明. 考虑一对这样的终端点 $s = r_{kp}^{\text{left}}$和 $t = l_{kp}^{\text{top}}$，其距离最大。根据对称性，共有8 对这样的终端点，且同一证明适用于所有这些终端对。

任意 s–t 路径要么使用对应的侧连接路径，要么进入（并随后离开）至少一个构件区域。

一方面，根据构造，存在一条使用侧连接路径的 s–t 游走，其长度为 $1.5\Delta' + O(kp)$。

另一方面，任何经过构件的路线都必须至少经过两条长度为 Δ' 的之字形附着路径（一条用于进入某个水平/垂直核心，另一条用于离开），因此其长度至少为 $2\Delta'$。由于 $\Delta' \gg 10kp$，我们有 $2\Delta' > 1.5\Delta' + O(kp)$，所以这样的绕行不可能是最短路径。◀

▶ **命题 20**。对于任意两个终端点 $s = l_i^{top}$，$t = l_i^{bot}$ 或 $s = r_i^{left}$，$t = r_i^{right}$，其中 $i \in [k]$，每条最短的 s–t 路径都会完整经过对应的垂直或水平构件内部的一条完整内部路径。

证明。对于每个 $i \in [k]$，在构件内部经过一条完整路径的长度为

$$l_i^{\text{top}} \to (\text{one vertical path}) \to l_i^{\text{bot}}, \quad \text{and} \quad r_i^{\text{left}} \to (\text{one horizontal path}) \to r_i^{\text{right}}$$

两者的长度均为 $2\Delta' + 8kp$。我们对垂直构件中的终端点证明该命题；水平构件的情形是对称的。

将上述路径记为 π，并令 π^* 为 $G^{\square}$ 中一条最短的 $l_i^{\text{top}} \to l_i^{\text{bot}}$路径。

假设 π^* 在不同的垂直路径之间切换，而不是沿着一条完整路径前进。根据构造，每次这样的切换都会在水平方向前进 4 条边，而垂直进展（在最佳情况下）仅为 2 个单位，因此路径长度至少增加 2 条边。这与 π^* 是一条最短路径相矛盾。

对于水平路径，情况更糟，因为它们之间的垂直子路径长度为 6，且没有任何水平进展，所以在不同的水平路径之间切换会使路径长度至少增加 5 条边。

此外，根据构造，l_i^{top} 和 l_i^{bot} 之间的所有垂直路径长度都相同，为 $2\Delta' + 8k$ p.

最后，任何不经过任何构件的路径都至少会经过两条边连接路径，因此长度至少为 $3\Delta'$；由于 Δ' 足够大，这一长度严格大于 $2\Delta' + 8kp$。

因此，该命题成立：在对应于 L_i 的垂直构件（对应于 R_i 的水平构件同理）中，完整 traversing 一条路径是唯一的最短路径，其长度为$2\Delta' + 8kp$。◀

▶ **命题 21.** 对于任意两个终端 $s = l_i^{top}$，$t = l_j^{bot}$ 或 $s = r_i^{left}$，$t = r_j^{right}$，其中 $i, j \in [k]$，每条 s–t 最短路径都会完整地经过两个对应的垂直构件或水平构件之一中的一条内部路径。

证明。特别地，对于所有 $i \neq j$，

$$d_{G^{\square}}(l_i^{\text{top}}, l_j^{\text{bot}}) \leq 2\Delta' + \big(|i-j| \cdot (p+6)\big) + 8kp \leq 2\Delta' + 9kp + 6k$$

$$\text{via} \quad l_i^{\text{top}} \to (\text{traverse } L_i\text{-gadget}) \to l_i^{\text{bot}} \to (\text{traverse on the common row}) \to l_j^{\text{bot}},$$

$$\text{or} \quad l_i^{\text{top}} \to (\text{traverse on the common row}) \to l_j^{\text{top}} \to (\text{traverse } L_j\text{-gadget}) \to l_j^{\text{bot}},$$

同理，

$$d_{G^{\square}}(r_i^{\text{left}}, r_j^{\text{right}}) \leq 2\Delta' + \big(|i-j| \cdot (p+4)\big) + 8kp \leq 2\Delta' + 9kp + 4k$$

$$\text{via } r_i^{\text{left}} \to (\text{traverse } R_i\text{-gadget}) \to r_i^{\text{right}} \to (\text{traverse on the common column}) \to r_j^{\text{right}}$$

$$\text{or } r_i^{\text{left}} \to (\text{traverse on the common column}) \to r_j^{\text{left}} \to (\text{traverse } R_j\text{-gadget}) \to r_j^{\text{right}}.$$

根据对称性，$(l_i^{\text{bot}}, l_j^{\text{top}})$ 和 $(r_i^{\text{right}}, r_j^{\text{left}})$ 这两对也同样成立。

证明与命题 20 类似。◀

结合上述关于 S 中终端点对之间最短路径的所有性质，我们现在陈述 $I^{\square}_{\mathrm{SDP}}$ 的子集保持器的两个重要性质。

▶推论 *22*。 *</style}*令 H 为 $I^{\square}_{\mathrm{SDP}}$ 的距离保持器。那么，根据观察 *18* 和命题 *19*，H 必须包含：

- 连接同侧终端的直线路径上的所有边，即连接.....的两条直线垂直路径

$$r_1^{left} \to r_k^{left} \quad and \quad r_1^{right} \to r_k^{right}$$

以及连接.....的两条直线水平路径

$$l_1^{top} \to l_k^{top} \; and \; l_1^{bot} \to l_k^{bot}$$

- 相邻边之间的所有连接路径。

利用上述性质，我们可以证明，对于 $I^{\square}_{\mathrm{SDP}}$，每个极小子集距离保持器必须在每个构件内恰好包含一条完整路径，并且同一构件内不同路径之间不可能发生切换，即：

▶ **引理 23**。 设 $H \subseteq G^{\square}$ 是 $I^{\square}_{\mathrm{SDP}}$ 的一个极小 子集距离保持器。那么对于每个$i \in [k]$， H 都包含

- 在 L_i-列构件内恰好一条 $l_i^{top} \to l_i^{bot}$内部路径的边，以及在 R_i-行构件内恰好一条
- $r_i^{left} \to r_i^{right}$ 内部路径的边。

利用上文所述的 $G^{\square}$ 中最短路径的性质，可仿照引理 15 证明这一点。

上述观察和命题，正是 G' 中相应最短路径结论的精确类比（将平面参数替换为其嵌入后的对应参数，特别是将 Δ 替换为 Δ'，并将构件内部遍历长度替换为 $2\Delta' + 8kp$）。此外，网格嵌入在定量意义上保留了每个水平/垂直交点处边与非边的区别：对于每个 $u \in R$ 和 $v \in L$，如果 $uv \in E(G)$，则嵌入后的路径 $\pi_u^{\square}$ 和 $\pi_v^{\square}$ 恰好共享两条网格边；如果 $uv \notin E(G)$，则恰好共享一条网格边（因此两种情形恰好相差一条共享边）。

因此， G 中的一个二部团可使所得保持子图减少 k^2 条边；反过来，实现全部 k^2 的减少量则意味着 G 含有一个二部团。定理 1 的证明至此完成。

3.2 网格上的 FPT 算法

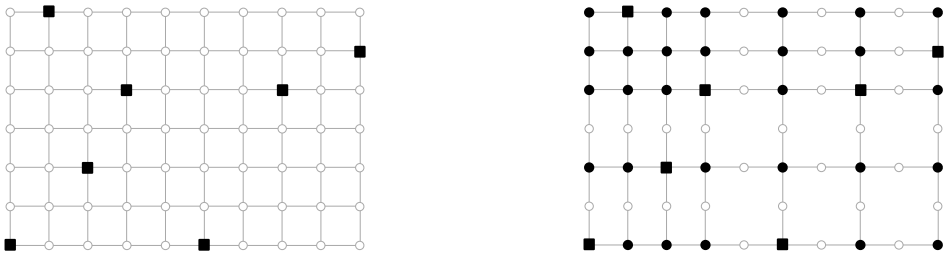
这里我们证明定理 2，现将其重述如下。

▶ **T定理 2**。 PDP 存在一个运行时间为 $4^{|S|^2} \cdot |V(G)|^{O(1)}$的算法，适用于输入图为网格的情形。

证明的主要部分是以下结构引理。对于网格图 $G = (V, E)$ 和终端集合 $S \subseteq V$，将 Hanan 网格定义为 G 的子图，其中仅包含位于 S 的顶点所在行和列上的边和顶点。我们可以假设 G 的最上方一行包含一个终端点，因为否则该行上的任何顶点都不可能位于两个终端之间的任何最短路径上，因此可以安全地移除。对网格的另外三条边也适用同样的论证。我们将位于包含终端的行与列交点上的顶点称为 Hanan 网格的交点；注意，Hanan 网格中任何不属于

交点的顶点的度均为 2，而根据定义，终端都是交点。图示请参见图 7。

Hanan 网格最初的研究动机是：可以假设，网格上给定一组终端点时，其最优直角斯坦纳树包含在这些点的 Hanan 网格内 [27]。我们证明，网格上的最小距离保持器也具有同样的性质。



(a) 网格图 $G = (E,\ V)$，含有 $|S| = 6$ 个终端点。 **(b)** 由 S诱导的 Hanan 网格 G。

图 7 一个实例，包括网格图 G 及其 Hanan 网格。终端点以黑色方块表示。网格图 G 的顶点以灰色边框的圆圈表示，而 Hanan 网格的交点则以实心黑圆表示。注意，Hanan 网格仅由位于至少包含一个终端点的行和列上的边与顶点构成。此外，Hanan 网格中任何非交点的顶点，其度均为 2。

▶ **引理 24**。 设 $G = (V, E)$ 是一个网格图，且 P 是 V 中若干对的集合。存在一个 (G, P) 的最小两两距离保持器，它是 $S = \bigcup P$ 的 *Hanan* 网格的子图。

证明。 设 H 是 $(G,\mathcal{P})$的一个最小距离保持器。我们称一个顶点为 H的分支顶点，当它在 H 中的度至少为 3，或者它是一个终端点时。我们称 P 是 H 中的一条基本路径，当它是 H的一条诱导路径、其端点是分支顶点，且没有内部顶点是分支顶点时。首先，我们证明一个关于基本路径的技术性断言。

▷断言 25。 H 的基本路径内部顶点不相交，并且覆盖了 H 的所有边。

证明。 假设两个不同的基本路径 P_1 和 P_2 共享一个内部顶点 v'。找出 P_1 的一个按包含关系极大的子路径 P'，使其包含 v' 且也是 P_2的子路径。设 u, v 是 P'的两个端点。如果 P' 等于 P_1，那么它要么也等于 P_2，要么 u 或 v 中的一个既是 P_2 的内部顶点，又是分支顶点，这就构成矛盾。

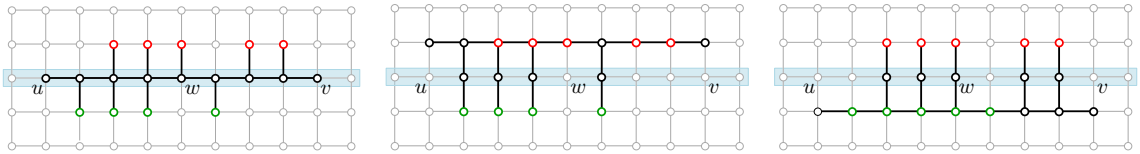
否则， u 和 v 中有一个不是 P_1 的端点，因此不是分支顶点。不失一般性，设 u 是这个顶点，并设 w 是 u 在 P_1 上、但不在 P'上的邻居。根据构造， u 属于 P_2，因此它要么是 P_2的端点，但这与 u 是 P_1的内部顶点矛盾；要么 P_2 在经过 u后继续延伸至顶点 t，且 $w \neq t$，因为 P' 是同时包含在 P_1 和 P_2中的极大子路径。于是， u 在 H中的度至少为 3，因此 u 是分支顶点，这与 P_1 是基本路径矛盾。

对于该论断的第二部分，考虑 H的一条边。如果它的两个端点都是分支顶点，根据定义，它就是一条基本路径。否则，取一个不是分支顶点的端点；它的度为二。用与该顶点相连的另一条边延长当前路径。继续此过程，直到路径的两个端点都成为分支顶点，此时该路径就是一条基本路径。注意，我们不可能得到一个环，因为此时该环至多包含一个终端点；从 H 中移除该环会得到一个更小的距离保持器，这与 H的极小性矛盾。◀

我们认为，只需证明 H 的每个分支顶点都是 Hanan 网格的交点即可。确实，假设这一点成立，并考虑 H 中连接

两个分支顶点 u 和 v 的一条基本路径 v，该路径不包含在 Hanan 网格中。由于 u 和 v 是 Hanan 网格的交点，因此也存在一条沿 Hanan 网格连接 u 和 v 的最短路径 P'。在 H 中用 P' 替换 P，可得到一个边数不超过 H 的新距离保持器，因为 P' 的长度不超过 P 的长度。对每条基本路径重复这一操作，我们便得到一个规模不增且包含在 Hanan 网格中的新距离保持器。根据断言 25， H 的每条边都由某条基本路径覆盖，因此上述结论成立。

因此，还需证明 H 的任意分支顶点都是 Hanan 网格的交点。假设 H 中存在一个不是交点的分支顶点 w。因此， w 所在的行不包含终端，或者 w 所在的列不包含终端；特别地， w 不是终端，但在 H 中的度为 3 或 4。不失一般性，假设 w 所在的行不包含终端。令 u 和 v 为 H 中包含 w 且位于 w 所在行上的按包含关系极大的路径的端点，并令 P_{uv} 为该路径。注意， w 所在的行既不在网格的上边界，也不在下边界，因为这两行都包含终端。令 A 为上一行中与 H 内 P_{uv} 的顶点相邻的顶点；类似地，令 B 为下一行中与 H 内的 P_{uv} 相邻的顶点。图 8a 展示了一个示例。



(a) 顶点 w、路径 P_{uv}，以及集合 A 和 B。

(b) 通过将 P_{uv} 向上移动一行来构造 {v3}。

(c) 通过将 P_{uv} 向下移动一行来构造 {v3}。

图 8 w 是 H 的一个分支顶点，且不是 Hanan 网格的交点； P_{uv} 是包含 w的包含关系意义下极大的水平路径；集合 A 和 B；以及修改后的图 H' 和 H''。w 所在的行以浅蓝色突出显示。 A 和 B 的顶点分别以红色和绿色显示。在(a)中， H 受影响的边以黑色显示；而在(b) 和(c)中， H' 和 H''的边分别以黑色显示。其余所有网格边和顶点均以灰色显示。

现在我们考察 G、 H' 和 H''的两个子图，如图 8b 和图 8c 所示。 H' 通过以下过程由 H 得到：首先，移除 P_{uv}的边，并在其上方一行添加与之平行的边。然后，对于 B 中同一列没有 A 顶点的每个顶点，从该顶点正上方一格处向再上方一格处添加一条边。对于 A中同一列没有 B 顶点的每个顶点，移除从该顶点向下的边。 H'' 的构造与此对称，其中 P_{uv} 的边移至下一行。根据构造， H' 的大小至多为 $|E(H)|+|B|-|A|$，而 H'' 的大小至多为 $|E(H)|+|A|-|B|$。我们证明 H' 和 H'' 也都是距离保持器。

▷ 命题 26. H' 和 H'' 是 $(G, \mathcal{P})$ 的两两距离保持器。

证明。我们针对 H'证明该结论；关于 H'' 的证明类似。设 $(s,t) \in$ P，并假设 H'中不存在 (s,t)-最短路径。由于在 H中曾存在一条最短 (s,t)-路径 P，它必然经过 P_{uv}中的顶点，因为在 H' 中仅移除了与这条路径相邻的边。 w所在的行包含 P_{uv}，但不包含终端，因此 s 或 t 之一在其上方，另一个在其下方。不失一般性，设 s 位于 w 所在行的上方， t 位于其下方。设 s' 是 P 在与 P_{uv} 相交之前的最后一个顶点， t' 是离开 P_{uv} 后在 P 上的第一个顶点。注意， s' 必定在 A 中，而 t' 必定在 B中。现在我们断言， s' 和 t' 在H'中通过最短路径相连，这意味着 s 和 t 也由 H'中的最短路径相连。事实上，考虑顶点 t''： 它与 t' 位于同一列，但在其上方两步，也就是说，位于 s'所在的行。 H' 包含

t' 与 t''之间的两条垂直边：根据 B的定义， t' 上方的边属于其中；于是，要么 t'' 在 A 中，因此从 t'' 向下的边也属于 H 和 H'，要么 t'' 不在 A中，在这种情况下，这条边是在构造 H'时添加的。由于 P_{uv} 在 H'中被上移了一行，t'' 和 s' 由 H'中的水平路径相连。证明至此完成。◀

由于 H' 和 H'' 都是距离保持器，且 H 是一个最小距离保持器，因此必有 $|A| = |B|$，并且 $|E(H)| = |E(H')| = |E(H'')|$。由于这对任意最小距离保持器 H 以及任意非交点的分支顶点 w 都成立，我们可以反复执行以下过程。如果 H 有一个非交点分支顶点 w，就用 H' 替换 H，并重复此操作。（对于在其列中没有终端点的 w，我们将 H' 定义为在该列内将最长路径向左移动后得到的对象。）由于这些操作只会将非交点分支顶点向上、向左移动，而不会向相反方向移动，因此该过程必定有限。由此，最终得到的距离保持器中所有分支顶点都位于交点上，从而完成引理的证明。◀

最后，我们给出定理 2 的证明。

定理 2 的证明。 根据引理 24，存在一个解只使用 Hanan 网格中的边。注意，Hanan 网格由 k^2 个交点组成，它们之间至多由 $2k^2$ 条直线线段连接。显然，对于任意一个 $(G,\mathcal{P})$ 的最小距离保持器，这些直线线段要么全部包含，要么完全不包含。对于每条线段，分别尝试将其纳入解或不纳入。根据上述结论，这些猜测中必有一个与最小距离保持器一致。可供选择的线段组合至多有 2^{2k^2} 种，并且可在 $|V(G)|^{O(1)}$ 时间内验证每个候选解。因此，枚举这些选择并返回其中最小的距离保持器的算法满足定理的陈述。◀

对于定理 2 的运行时间，注意 Krau- thgamer 等人的网格下界构造 [32] 表明，存在一个实例，其中最小距离保持器必须使用 Hanan 网格中的 $\Omega(k^2)$ 个顶点和边。

3.3 PDP 在网格图上是 NP 困难的

这里，我们证明，即使将问题限制在网格图这一特殊类别中，计算最小规模的 PDP 仍然是 NP 困难的。我们的证明通过从直角斯坦纳树形问题进行多项式时间归约来完成；已知该问题是 NP 困难的 [37]。

本节中，我们假设所有点都位于整数网格坐标上。这一限制不会损失一般性。虽然整数性并非 RSA 原始定义的一部分，但 [37]，中给出的从 3SAT 出发的归约所产生的实例，其所有终端点坐标均为整数。因此，在这一假设下，困难性结果仍然成立。

我们回顾该问题的正式定义。

直线型斯坦纳树形图 (RSA)

输入： 一个 $P = \{p_1、p_2、\ldots\ldots、p_n\} \subseteq \mathbb{N}^2$ 个 n 点，称为终端，位于笛卡尔平面的第一象限，其中 $p_i = (x_i, y_i)$ 对每个 i 都成立，且为整数 k。

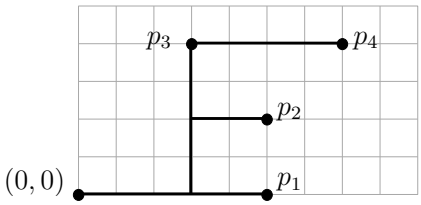
问题： 是否存在一棵以原点为根的有向斯坦纳树 $T = (V, E)$，其根为 $(0, 0)$，并满足 $P \cup \{(0, 0)\} \subseteq V$，$E$ 中的每条边都是从左向右或从下向上的水平或垂直线段，且 T 的边总长度至多为 k?

请注意，底层网格和 RSA 实例都是无向的。不过，由要求 T 是一棵以 (0，0) 为根的最短路径树，方向便隐式地产生了。对于任意终端点 $p=(x,\ y)\in P$，根据定义，从 (0，0) 到 p 的每条最短路径都恰好由 x 个向右的水平步和 y 个向上的垂直步组成。因此，沿着任意这样的路径，每条水平线段都连接一个 x 坐标较小的顶点和一个 x 坐标较大的顶点，而每条垂直线段都连接一个 y 坐标较小的顶点和一个 y 坐标较大的顶点。因此，我们按照根到终端点最短路径所确定的方向，为 T 中的每条边定向：水平边从左向右，垂直边从下向上。

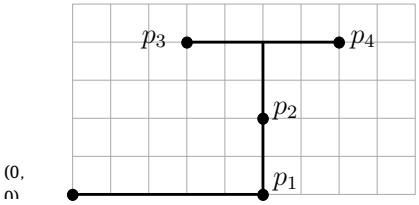
此外，RSA 与直角斯坦纳树的关键区别在于，RSA 还必须是相对于原点的最短路径树，而斯坦纳树只需保持连通性。图 9 展示了从根出发的最短路径所隐含的边方向，以及 RSA 与直角斯坦纳树之间的区别。

▶ **观察 27**。令 T 为一个 *RSA*，其终端集合为 $P=\{p_1,p_2,\ldots,p_n\}$。对于每个终端 $p_i=(x_i,y_i)\in P$，从 $(0,\ 0)$ 到 p_i 在 T 中的有向路径长度恰为 x_i+y_i。

在本节中，我们用不带下标的 $d(p_1,\ p_2)$ 表示底层整数网格图上从 p_1 到 p_2 的最短路径长度。



(a) 总长度为 15 的 RSA。



(b) 总长度为 13 的直角斯坦纳树

图 9 (a) RSA 示意图，以及 (b) 终端 $P=\{(5,0),(5,2),(3,4),(7,4)\}$ 对应的直角斯坦纳树示意图。在 RSA 中，从原点到某个终端 $p_i=(x_i,y_i)$ 的唯一路径上的每条边都朝左到右或朝下到上定向，并且该路径的长度 x_i+y_i 达到可能的最短值。相比之下，直角斯坦纳树虽然保持连通性，却不保证最短路径；从 $(0,0)$ 到 p_3 的路径长度为 $11>d\big((0,0)\ p_3\big)=7$。

在 [37]，中，作者证明了 RSA 是 NP 困难的。利用这一结果，我们证明了 PDP 的以下困难性定理：

▶**定理 28.** PDP 在网格图上是 *NP* 困难的。

证明。 我们通过从 RSA进行多项式时间归约来证明该定理。

设 $I_{\mathrm{RSA}}=(P,\ k)$ 是一个 RSA 实例，其中 $P=\{p_1,\ p_2,\ \ldots\ldots,\ p_n\}$ 且 $p_i=(x_i,\ y_i)$。令 $p_0=(0,\ 0)$ 表示笛卡尔平面上的原点。我们通过设定 $k'=k$ 和……，构造一个 PDP 实例 $I_{\mathrm{PDP}}=(\mathcal{P}',\ k')$

$$\mathcal{P}'=\big\{\{p_0,p_i\}\mid i\in[n]\big\}.$$

至此，构造完成。

我们证明，I_{RSA} 存在总长度至多为 k 的解，当且仅当 I_{PDP} 存在总长度至多为 k' 的解。

为证明正向结论，假设 I_{RSA} 存在一个总长度不超过 k 的解 T。根据定义，T 是一棵以 p_0 为根的最短路径树，因此它通过一条长度为 $p_i\in P$ 的路径将 p_0 连接到每个终端点

，其长度为 $x_i + y_i = d(p_0, p_i)$。因此， T 保持了 $\mathcal{P}'$ 中所有点对之间的距离，因而是 I_{PDP} 的一个可行解，总长度不超过 $k = k'$。

为证明反方向，假设 $I_{\mathrm{PDP}} = (\mathcal{P}', k')$ 存在一个总长度至多为 k' 的成对距离保持子图 H。

根据观察 7，每条边 $e \in E(H)$ 都至少位于一条从 p_0 到某个终端点 p_i 的最短路径上，其中 $i \in [n]$。我们沿着这样一条路径为每条边定向，从 p_0 指向 p_i。这就为 H 的每条边隐式地定义了一个方向。

▷ 断言 29. H 的每条边要么从下到上定向，要么从左到右定向。

证明。 反过来，假设 e 是 H 中一条从上到下定向的边。根据观察 7， e位于某条最短路径上。若 e 位于一条最短路径上， π 该路径从 p_0 到 p_i，则

$$d_H(p_0, p_i) \geq x_i + y_i + 1 = d(p_0, p_i) + 1,$$

这与 H 保持了 p_0 和 p_i 之间最短距离这一事实矛盾。因此， e 不位于任何最短路径上，可以将其移除，而不影响最短距离和连通性。对于存在一条从右到左的边，证明方法类似。◀

▶**推论 30.** H 的每条边都有唯一的隐含方向，该方向由从 p_0 到终端的最短路径决定。

根据断言 29，我们可以假设 H 仅包含从左到右和自下而上的边。因此，从 p_0 到 H 中任意顶点的每条路径都是单调的，也就是说，其中不包含自上而下或从右到左的步。

▷ 断言 31. 如果 H 包含一个环 C，则存在一个成对距离保持子图 H' ，使得$|E(H')| \leq |E(H)|$ ，且环的数量严格更少。

证明。 设 C 是 H中的一个环。根据断言 29， C 的所有边都朝左到右或自下而上的方向，因此 C 构成一个正交多边形。令 $a = (x_a, y_a)$ 和 $b = (x_b, y_b)$ 分别表示 C的左下角和右上角。

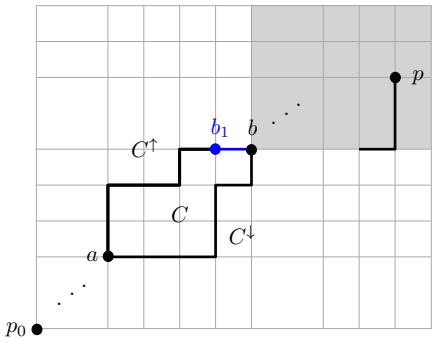
环 C 由两条内部不相交的单调路径 $C^{\uparrow}$ 和 $C^{\downarrow}$ 组成，它们连接 a 与 b。由于两条路径均为单调路径，因此每条路径恰好包含$(x_b - x_a)$条向右的边和$(y_b - y_a)$条向上的边，所以 $|C^{\uparrow}| = |C^{\downarrow}|$。

令 $P_b \subseteq P$ 为从 p_0 出发、在 H 中经过 b的最短路径所到达的终端集合。由于从 p_0 到 H 中任一终端点的每条路径都是单调的，因此对于每个终端点 $p = (x_p, y_p) \in P_b$ ，我们有$x_p \geq x_b$ 且 $y_p \geq y_b$。于是，对于每个这样的终端点 $p \in P_b$，从 p_0 到 p 的任意一条最短路径都可以经过 $C^{\downarrow}$ 而非 $C^{\uparrow}$ ，且不会增加路径长度。图 10a 展示了这一配置。

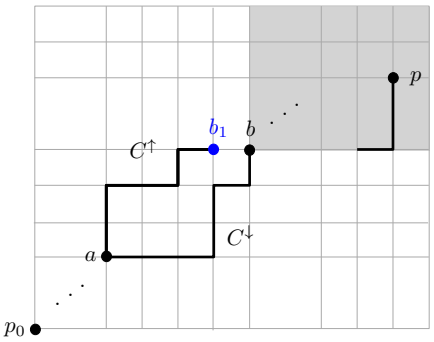
令 $b_1 = (x_b - 1, y_b)$ 为沿着 $C^{\uparrow}$到达 b 的直接前驱。删除边 $b_1 b$后，从 p_0 到 P_b 中所有终端点的所需最短距离均得以保留，同时环 C也被消除。由于没有添加边，因此不会引入新的环。图10b 展示了这一情况。

◀

通过反复应用断言 29 和断言 31，我们得到一个成对距离保持子图 H ，其总长度至多为 k' ，且不含自上而下边、自左向右边和环。因此， H 是一棵仅由自下而上边和自左向右边组成的树，它通过最短路径将 p_0连接到 P 中的每个终端点。因此， H 是一个大小至多为 $k' = k$的有效 RSA。最后，PDP ∈NP，因为只需检查所有必需的终端距离是否得到保持，即可在多项式时间内验证大小至多为 k 的候选解。◀



(a) 一个含有环 C 的 PDP H，该环是一个正交多边形。



(b) 消除环后得到另一个 PDP H' C 通过移除边 b_1b。

图 10 灰色区域标示了 P_b 中所有终端的潜在位置。**(a)** H 中的一个正交环C，其左下角为 a，右上角为 b，两个单调子路径为 $C^{\uparrow}$ 和 $C^{\downarrow}$，且满足$|C^{\uparrow}| = |C^{\downarrow}|$。顶点 b_1 是沿 $C^{\uparrow}$ 的 b 的前驱。**(b)** 移除边 b_1b 后的 H；从 p_0 到 P_b 中终端的最短路径经由 $C^{\downarrow}$ 重新规划。

4 按树宽和终端数量划分的固定参数可解性

本节将证明，以输入图的树宽和终端数量为参数时，PDP 是固定参数可解的。更正式地说，我们将证明定理 3，现重述如下。

▶ **定理 3。** PDP 存在一个运行时间为 $2^{O\left((\mathrm{tw}+|S|)^2\right)} \cdot |V(G)|^{O(1)}$ 的算法，其中tw是 G 的树宽。

直观地说，树宽至多为 τ 的图可以按树状方式分解，在任意时刻只需考虑一个至多包含 $\tau+1$ 个顶点的小集合，即袋。这个袋充当图中已处理部分（“过去”）与尚未探索部分（“未来”）之间的分隔集，因此过去与未来之间的所有连接都必须经过当前的袋。

对于 PDP 实例，这对应于这样一个性质：过去的终端只能通过袋中的顶点与未来的终端交互。

我们的算法利用这一结构，在树分解上执行动态规划，并在每个节点存储一组记录，用以概括当前袋中顶点相关的连通性和距离信息。

我们首先回顾良好树分解的定义。良好树分解是在有界树宽情形下执行动态规划的一种标准结构。

良好树分解。 图 $G = (V, E)$ 的良好树分解 是一个二元组 $(\mathcal{T}, \mathcal{X})$，其中 $\mathcal{T}$ 是以 r 为根的有根树，而 $\mathcal{X}$ 是一个映射，为每个节点 $t \in \mathcal{T}$ 指定一个集合$X_t \subseteq V(G)$，称为节点 t 处的袋。良好树分解对每个 $t \in \mathcal{T}$都满足以下性质：

1. $X_r = \emptyset$ 且对于每个叶节点 ℓ，都有 $X_\ell = \emptyset$。2. 对于每个 $uv \in E$，都存在一个节点 $t \in \mathcal{T}$，使得 $u \in X_t$ 和 $v \in X_t$。3. 对于每个 $v \in V$，节点集合 $t \in \mathcal{T}$，满足 $v \in X_t$，在 $\mathcal{T}$中诱导出一个连通子树。4. 在 $\mathcal{T}$中（根和叶节点除外）有四种节点：

XX:30 寻找最小距离保持子图：参数化研究

a. 引入顶点节点：一个节点 t，它只有一个子节点 t'，并且存在一个顶点 $v \notin X_{t'}$满足 $X_t = X_{t'} \cup \{v\}$。我们称 v为引入的顶点。b. 引入边节点：一个节点 t，标记有一条边 $uv \in E(G)$，满足 $u, v \in X_t$，并且恰有一个子节点 t' 满足 $X_t = X_{t'}$。我们称边 uv 在 t处被引入。c. 遗忘节点：一个 t，它只有一个子节点 t'，并且存在一个顶点 $v \notin X_t$ 满足$X_t = X_{t'} \setminus \{v\}$。遗忘一个顶点时，也会遗忘它的所有关联边。我们称 v为被遗忘的顶点。d. 合并节点：一个节点 t 有且仅有两个子节点 t_1 和 t_2，满足 $X_t = X_{t_1} = X_{t_2}$。

注意，根据性质 3，从叶节点向根遍历时，顶点 $v \in V(G)$ 一旦被遗忘，就不能再次引入。否则，$\mathcal{T}$中包含 v 的袋所对应的节点诱导出的子树将不连通。良好树分解 $(\mathcal{T}, \mathcal{X})$ 的宽度定义为 $\max_{t \in \mathcal{T}} (|X_t| - 1)$，图 G 的树宽定义为 G 的良好树分解的最小宽度，记作 tw(G)。

可以使用 [5], [6] 和 [28] 中描述的固定参数可解算法，获得一个宽度近乎最优且节点数为线性的良好树分解。

现在我们开始正式证明。设 $(G, \mathcal{P})$ 是 PDP 的一个实例，并假设 G 的树宽至多为 τ。令 $S = \bigcup \mathcal{P} \subseteq V(G)$ 表示终端顶点集合。

令 $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ 为 G 的宽度为 τ 的良好树分解。对于节点 $t \in T$，将以 t 为根的 T 子树记作 T_t。我们将 G 中出现在 T_t 的袋中的所有顶点和边所构成的子图记作 G_t。此外，用 $S_t^{\downarrow}$ 表示出现在 T_t 的袋中的所有终端顶点的集合。对于每个节点 $t \in T$，令 X_t 表示其袋，并定义 $U_t := X_t \cup S_t^{\downarrow}$.

树分解的一项关键性质是，每个袋 X_t 都将子图 G_t 与 G 的其余部分分隔开。特别地，任何从 $G_t \setminus X_t$ 出发并离开 G_t 的路径都必须经过 X_t 中的顶点。由于最短路径的每个子路径本身也是最短路径，因此，对于 $S_t^{\downarrow} \setminus X_t$ 中的任意终端顶点，任何连接该顶点与 G_t 之外终端的最短路径都必须与 X_t 相交。因此，在处理节点 t 时，只需记住 U_t 中哪些顶点在限制于 G_t 的部分解中通过最短路径彼此连通。我们将用一张连通性表记录这些信息，以及实现该连通性所需的最少边数。

我们将上述直觉形式化如下。

▶ **定义32**（连通性表）。对于节点 $t \in T$，令 Σ_t 为所有对称函数的集合σ：$U_t \times U_t \to \{0,1\}$，也就是说，对所有 i、$j \in U_t$，均有 $\sigma(i, j) = \sigma(j, i)$。

直观而言，表 $\sigma \in \Sigma_t$ 针对 X_t 中的每个顶点以及迄今遇到的每个终端顶点，都有一行和一列。表项 $\sigma(i, j) = 1$ 表示 i 和 j 在 G_t内部的部分解中由一条最短路径连通，而 $\sigma(i, j) = 0$ 则表示情况并非如此。

令 U和 U'为满足 $U' \subseteq U$的顶点集。对于表 σ: $U \times U \to \{0, 1\}$，我们将其在 U'上的限制记为 $\sigma \restriction_{U'}$，定义为对所有 $i, j \in U'$，$\sigma \restriction_{U'}(i, j) := \sigma(i, j)$。反过来，若 $\sigma \restriction_{U'} = \sigma'$，则称表 σ: $U \times U \to \{0, 1\}$扩展 表 σ': $U' \times U' \to \{0, 1\}$。

对于节点 t 和表 $\sigma \in \Sigma_t$，我们将 $w[t, \sigma]$ 定义为满足以下条件的子图中边数的最小值 $H_t \subseteq G_t$：对于所有 $i, j \in U_t$:

$$\sigma(i,j) = \begin{cases} 1 & \text{if and only if } d_{H_t}(i,j) = d_G(i,j), \\ 0 & \text{otherwise.} \end{cases}$$

如果不存在这样的子图，我们令 $w[t, \sigma] = +\infty$。

任何满足上述条件的子图 H_t 都称为与 (t, σ)兼容，并且实现表 σ。

根据定义，每个表 $\sigma \in \Sigma_t$ 都是对称的。在本节余下部分中，条目 $\sigma(i,j)$默认同时表示 (i,j) 和 (j,i)。因此，对 $\sigma(i,j)$进行的任何操作，都默认以对称方式应用于 $\sigma(j,i)$，无需另行说明。现在我们来描述如何

通过动态规划自 $w[t,\sigma]$ 底向上计算。

- **叶节点：** 根据定义， t 是一个叶节点，因此 $X_t = S_t^{\downarrow} = \emptyset$，所以 $U_t = \emptyset$。由于 U_t 为空，在此节点定义的表是唯一的，即 $\sigma = \emptyset$ 。与 $(t,\emptyset)$ 兼容的唯一子图是空图。因此我们设定 $w[t,\emptyset] = 0$.
- **引入顶点节点：** 设 t 引入顶点 v，其子节点为 t'，使得 $X_t = X_{t'} \cup \{v\}$。回想一下，$U_t = X_t \cup S_t^{\downarrow}$ 且 $U_{t'} \subseteq U_t$。对于每个 $\sigma \in \Sigma_t$，我们设定

$$w[t,\sigma] = \begin{cases} w[t', \sigma \restriction_{U_{t'}}], & \text{if } \sigma(v,u) = 0 \text{ for all } u \in U_t \text{ with } u \neq v, \\ +\infty, & \text{otherwise.} \end{cases}$$

  根据良好树分解的构造，与 v 关联的边不会出现在 G_t中。因此， v不可能通过最短路径连接到任何先前引入的顶点或终端点，所有涉及 v 的对应条目都必须为零。表的其余部分必须与子节点的状态一致。

- **引入边节点：** 令 t 通过子节点 t'引入一条边 uv 。我们有 $X_t = X_{t'}$、 $S_t^{\downarrow} = S_{t'}^{\downarrow}$ 和 $d_G(u, v) = 1$。定义

$$U_{uv} := \big\{\{p,q\} \subseteq U_t \mid d_G(p,q) = d_G(p,u) + 1 + d_G(v,q)\big\},$$

  使用新边 uv可能实现的顶点对集合。若表 $\sigma \in \Sigma_t$ 在引入边节点 t 处是边无效的，则称

$$\sigma(u,v) = 1 \quad \text{and there exists } \{p,q\} \in U_{uv} \text{ with } \sigma(p,q) = 0 \text{ and } \sigma(p,u) = \sigma(q,v) = 1.$$

  令 $\Sigma_t^{\text{edge}-\text{inv}}$ 表示引入边节点 t处所有边无效表的集合。对于每个 $\sigma \in \Sigma_t$，定义

$$\Sigma' := \big\{\sigma' \in \Sigma_{t'} \mid \sigma'(u,v) = 0 \text{ and } \sigma'(i,j) = \sigma(i,j) \text{ for all pairs} \{i,j\} \notin U_{uv}\big\}.$$

  则更新为

$$w[t,\sigma] = \begin{cases} +\infty, & \text{if } \sigma \in \Sigma_t^{\text{edge}-\text{inv}}, \\ w[t',\sigma], & \text{if } \sigma(u,v) = 0, \\ \min\limits_{\sigma' \in \Sigma'} w[t',\sigma'] + 1, & \text{otherwise.} \end{cases}$$

  边 uv 可能会补全 U_{uv}中顶点对之间的最短路径。若 σ 是边无效的，则任何子图 H_t都无法实现它。若 $\sigma(u, v) = 0$，则边 uv 不会用于 H_t，因此 $H_{t'}$ 与 H_t之间没有区别。否则，若 $\sigma(u, v) = 1$，我们会考虑所有在 U_{uv} 之外一致的兼容子节点表，并添加该边以扩展解。

- **遗忘顶点节点：** 令 t 遗忘顶点 v，其子节点为 t'。则 $S_t^{\downarrow} = S_{t'}^{\downarrow}$，且 $X_t = X_{t'} \setminus \{v\}$。我们区分以下情况：

XX:32 寻找最小距离保持子图：参数化研究

- 如果 $v \in S_t^{\downarrow}$，并且存在一个终端点 $u \in S \setminus S_t^{\downarrow}$，使得 $\{u,\ v\} \in \mathcal{P}$，那么 v 必须通过最短路径与 X_t 中的某个顶点保持连通，以便之后能够连接到 u。由于$U_t = U_{t'}$，对于每个 $\sigma \in \Sigma_t$，我们都设置：

$$w[t,\sigma] = \begin{cases} +\infty, & \text{if } \sigma(v,u) = 0 \text{ for all } u \in X_t, \\ w[t',\sigma] & \text{otherwise.} \end{cases}$$

- 如果 $v \in S_t^{\downarrow}$ 但不存在这样的终端点 u，那么 v 没有其他连通性要求，并且对于每个 $\sigma \in \Sigma_t$，我们设

$$w[t,\sigma] = w[t',\sigma].$$

- 如果 $v \notin S_t^{\downarrow}$，那么 U_t 比 $U_{t'}$ 少一行一列（对应于 v）。对于每个$\sigma \in \Sigma_t$，定义

$$w[t,\sigma] = \min_{\substack{\sigma' \in \Sigma_{t'} \\ \sigma'|_{U_t} = \sigma}} w[t',\sigma'].$$

对于那些被“遗忘”但仍需与未来终端点连接的终端点，我们确保它们仍与当前袋保持连通。被遗忘的非终端顶点无需出现在未来的表中，因此会移除其对应的行和列。动态规划在所有兼容的子节点表中取最小值，以保持最小解成本。

- **合并节点：**设 t 为一个合并节点，其子节点为 t_1，t_2。那么 $X_t = X_{t_1} = X_{t_2}$，$S_t^{\downarrow} = S_{t_1}^{\downarrow} \cup S_{t_2}^{\downarrow}$，且 $U_t = U_{t_1} \cup U_{t_2}$。定义 $S^{\cap} = S_{t_1}^{\downarrow} \cap S_{t_2}^{\downarrow}$。
对于 $\sigma \in \Sigma_t$，令 $\sigma_i \in \Sigma_{t_i}$ 表示其在 U_{t_i} 上针对 $i = 1,\ 2$ 的限制，即 $\sigma_i = \sigma \restriction U_{t_i}$。根据定义，这些限制在 $X_t \cup S^{\cap}$ 中的所有索引上都与 σ 一致：

$$\sigma_1(i,j) = \sigma_2(i,j) = \sigma(i,j), \quad \forall i,j \in X_t \cup S^{\cap}.$$

如果存在终端点 $a \in S_{t_1}^{\downarrow} \setminus S_{t_2}^{\downarrow}$ 和 $b \in S_{t_2}^{\downarrow} \setminus S_{t_1}^{\downarrow}$，满足 $\{a,\ b\} \in \mathcal{P}$，且存在一个顶点 $v \in X_t$，则称 σ 在合并节点 t 处是连接无效 的：

$$\sigma(a,b) = 0, \quad \sigma_1(a,v) = \sigma_2(b,v) = 1, \quad d_G(a,b) = d_G(a,v) + d_G(v,b).$$

令 $\Sigma_t^{\text{join-inv}}$ 表示在 t 处合并节点上的所有此类无效表组成的集合。
为了避免对符合 $(t,\ \sigma)$ 的部分解中的边进行重复计数，定义

$$\delta(\sigma) := \left|\{uv \in E(G[X_t]) \mid \sigma(u,v) = \sigma_1(u,v) = \sigma_2(u,v) = 1\}\right|.$$

最后，对于每个 $\sigma \in \Sigma_t$，我们设：

$$w[t,\sigma] = \begin{cases} +\infty, & \text{if } \sigma \in \Sigma_t^{\text{join-inv}}, \\ w[t_1,\sigma_1] + w[t_2,\sigma_2] - \delta(\sigma), & \text{otherwise.} \end{cases}$$

在合并节点处，我们合并来自两棵子树的部分解。如果某个终端对的两个终端分别位于两棵子树中，并且它们可以经由袋 X_t形成最短路径，但表 σ 未将它们标记为已连通，那么表 σ 就是无效的。否则，我们将两个子节点的边数相加，并减去 $\delta(\sigma)$，以避免对完全包含在袋 X_t中的边重复计数。

- **根节点** r**：** 在根节点处，$X_r = \emptyset$ 和 $S_r^{\downarrow} = S$，因为所有终端都已访问。我们要求 $\mathcal{P}$ 中的所有终端对都通过最短路径连通。令

  $$\Sigma^* := \{\sigma \in \Sigma_r \mid \sigma(u,v) = 1 \text{ for all } \{u,v\} \in \mathcal{P}\}.$$

  因此，$(G,\ \mathcal{P})$ 的 PDP 最优规模为

  $$\min_{\sigma \in \Sigma^*} w[r,\sigma].$$

  在根节点处，所有部分解都已合并，因此我们只需选取能够为所有终端对实现最短路径的表，从而得到最终解中的最少边数。

上述动态规划计算出最优距离保持器的规模。对于每个有限值的表项 $(t,\ \sigma)$，记录在递推中取得最小值的见证选择后，便可通过在树分解上执行标准的回溯过程，重构出相应的子图。因此，我们不仅能得到最优值，还能得到一个最优解。

完成对表项和转移规则的描述后，接下来需要证明该动态规划计算出的值确实是 PDP 的最优解。证明基于一个经过精心维护的不变量，该不变量将每个表项 $w[t,\sigma]$ 与 G_t 的子图关联起来，这些子图实现了 σ 所编码的连通性。我们先重述这个不变量，

动态规划不变式。

对于树分解中的每个节点 t 以及满足 $w[t,\sigma] < +\infty$ 的每个表 $\sigma \in \Sigma_t$，都存在一个子图 $H_t \subseteq G_t$，满足 $|E(H_t)| = w[t,\sigma]$，使得对于所有 $i,\ j \in U_t$，

$$\sigma(i,j) = 1 \quad \Longleftrightarrow \quad d_{H_t}(i,j) = d_G(i,j).$$

Mor此外，每个满足此条件的子图 $H_t \subseteq G_t$ 都恰好对应一个表 σ。

我们沿着动态规划所采用的自底向上遍历，对树分解中的节点进行归纳，从而证明这一不变式。归纳证明表明，该不变式在每种节点类型和每条转移规则下都得以保持。这为确立可靠性（即每个有限的表项都对应于图中一个可实现的部分解）和完备性（即每个有效的部分解都会产生一个表项，并在动态规划中以相同代价保留下来）奠定了基础。这些性质共同确保根节点处返回的值是最优的。

▶ **引理 33** (可靠性)。 对于每个节点 $t \in T$ 以及满足 $w[t,\sigma] < +\infty$ 的每个表 $\sigma \in \Sigma_t$，都存在一个子图 $H_t \subseteq G_t$，满足 $|E(H_t)| = w[t,\sigma]$，且对于所有 $i,j \in U_t$:

$$\sigma(i,j) = 1 \quad \Longleftrightarrow \quad d_{H_t}(i,j) = d_G(i,j).$$

证明。 我们通过对良好树分解的节点进行归纳，证明其可靠性。

叶节点。 设 t 是一个叶节点。那么 $U_t = \emptyset$ 和 G_t 不包含任何顶点。因此，Σ_t 仅由一个表组成，即空表 $\sigma = \emptyset$。根据动态规划表的填充方式，有$w[t,\sigma] = 0$。我们定义 $H_t := \emptyset$ 为空图，它实现了 $\sigma = \emptyset$，并且显然$|E(H_t)| = 0 = w[t,\emptyset]$。

引入顶点节点。 令 t 引入顶点 v，其子节点为 t'；也就是说，$X_t = X_{t'} \cup \{v\}$。考虑任意一个满足 $w[t,\sigma] < +\infty$的表 $\sigma \in \Sigma_t$。根据递推，其限制 $\sigma' = \sigma \restriction_{U_{t'}}$满足 $w[t',\sigma'] < +\infty$，因为我们令 $w[t,\sigma] = w[t',\sigma']$.。根据归纳假设，存在

一个子图 $H_{t'} \subseteq G_{t'}$，实现了满足 $|H_{t'}| = w[t', \sigma']$的 σ'。由于在此阶段，v 在 G_t中没有关联边，我们定义

$$H_t := \big(V(H_{t'}) \cup \{v\},\, E(H_{t'})\big) \subseteq G_t.$$

因此 $|E(H_t)| = |E(H_{t'})|$。此外，动态规划仅允许满足 $\sigma(u,\ v) = 0$ 的表 σ，对所有 $u \in U_t$，均成立，这意味着 v 是孤立的。因此，H_t 实现了 σ，因为对于所有 $i,\ j \in U_t$，H_t 中的距离都与 σ 一致，并且

$$|E(H_t)| = |E(H_{t'})| = w[t', \sigma'] = w[t, \sigma].$$

因此不变量成立。

引入边节点。令 t 引入一条边 uv，其子节点为 t'。考虑一个表 $\sigma \in \Sigma_t$，满足 $w[t, \sigma] < +\infty$。根据动态规划递推，$\sigma \notin \Sigma_t^{\mathrm{edge-inv}}$。我们区分两种情况。

情况 1： $\sigma(u,\ v) = 0$。存在一个表 $\sigma' \in \Sigma_{t'}$，使得 $\sigma' = \sigma$ 且 $w[t', \sigma'] < +\infty$，这是递推规则所规定的。根据归纳法，存在一个子图 $H_{t'} \subseteq G_{t'}$，实现了 σ'，且满足 $|E(H_{t'})| = w[t', \sigma']$. 。由于没有使用边 uv，我们定义

$$H_t := \big(V(H_{t'}), E(H_{t'})\big) \subseteq G_t.$$

因此 $|E(H_t)| = |E(H_{t'})|$ 和 H_t 实现了 σ，所以不变量成立，因为

$$|E(H_t)| = |E(H_{t'})| = w[t', \sigma'] = w[t, \sigma].$$

情况 2： $\sigma(u,\ v) = 1$。根据动态规划更新规则，存在 $\sigma' \in \Sigma_{t'}$，使得 σ 由σ'得到，方法是可能将 $\sigma(i,\ j) = 1$赋给成对的$\{i,\ j\} \in U_{uv}$，并在其他位置令 $\sigma(i,\ j) = \sigma'(i,\ j)$。令 σ'为这些选择中使 $w[t', \sigma']$ 最小的表。根据归纳法，存在 $H_{t'} \subseteq G_{t'}$实现 σ'，且 $|E(H_{t'})| = w[t', \sigma']$。

接下来定义

$$H_t := \big(V(H_{t'}), E(H_{t'}) \cup \{uv\}\big) \subseteq G_t,$$

由此得到 $|E(H_t)| = |E(H_{t'})| + 1$。添加 uv 即可实现 U_{uv}中各对顶点的所有最短路径，而 $H_{t'}$ 实现其他所有距离。因此，H_t 实现了 σ，并且

$$|E(H_t)| = |E(H_{t'})| + 1 = w[t', \sigma'] + 1 = w[t, \sigma].$$

注意，引入边的无效性条件确保不会产生或违反其他最短路径关系；因此，H_t 恰好实现了连通性表 σ，从而保持不变量。

遗忘顶点节点。设 t 遗忘顶点 v，其子节点为 t'，并设 $\sigma \in \Sigma_t$ 满足 $w[t, \sigma] < +\infty$。回顾一下，$X_t = X_{t'} \setminus \{v\}$ 且 $S_t^{\downarrow} = S_{t'}^{\downarrow}$。我们分两种情况讨论。

情况 1： $v \in S_t^{\downarrow}$。那么 $U_t = U_{t'}$。根据动态规划递推，任何满足 $w[t, \sigma] < +\infty$的表 $\sigma \in \Sigma_t$ 在 t'处也都是有效表，也就是说，存在 $\sigma' \in \Sigma_{t'}$ 满足 $\sigma' = \sigma$ 且 $w[t', \sigma'] < +\infty$。根据归纳假设，存在一个子图 $H_{t'} \subseteq G_{t'}$ 实现了 σ'，且

$$|E(H_{t'})| = w[t', \sigma'].$$

我们定义 $H_t := H_{t'}$。由于 U_t 中顶点之间的距离保持不变，H_t 实现了 σ，并且

$$|E(H_t)| = |E(H_{t'})| = w[t', \sigma'] = w[t, \sigma].$$

情况 2： $v \notin S_t^{\downarrow}$。于是 $U_t = U_{t'} \setminus \{v\}$。根据动态规划递推，存在一个扩展$\sigma' \in \Sigma_{t'}$，使得

$$\sigma' \restriction_{U_t} = \sigma \quad \text{and} \quad w[t', \sigma'] < +\infty.$$

根据归纳假设，存在一个子图 $H_{t'} \subseteq G_{t'}$，实现了 σ'，且

$$|E(H_{t'})| = w[t', \sigma'].$$

我们再次定义 $H_t := H_{t'}$。由于 $H_{t'}$ 实现了 σ'，它也实现了其限制 σ，并且

$$|E(H_t)| = |E(H_{t'})| = w[t', \sigma'] = w[t, \sigma].$$

因此，在所有情况下，不变量在遗忘节点处都成立。

合并节点。 设 t 是一个合并节点，其子节点为 t_1 和 t_2。设 $\sigma \in \Sigma_t$ 满足 $w[t, \sigma] < +\infty$，并设 $\sigma_i := \sigma \restriction_{U_{t_i}} \in \Sigma_{t_i}$ 对所有 $i = 1, 2$。
根据动态规划递推，$\sigma \notin \Sigma_t^{\text{join-inv}}$ 和 $w[t_i, \sigma_i] < +\infty$ 对于两个 $i = 1, 2$。
根据归纳假设，存在子图 $H_{t_i} \subseteq G_{t_i}$ 实现 σ_i，使得

$$|E(H_{t_i})| = w[t_i, \sigma_i] \quad \text{for } i = 1, 2.$$

我们定义

$$H_t := (V(H_{t_1}) \cup V(H_{t_2}),\ E(H_{t_1}) \cup E(H_{t_2})).$$

由于G_{t_1}，$G_{t_2} \subseteq G_t$，因此可得 $H_t \subseteq G_t$。

要验证 σ的实现，首先注意，完全包含在 U_{t_i} 中的顶点对之间的距离由 H_{t_i} 保持，因此也由 H_t保持。此外，对于顶点对$a \in U_{t_1} \setminus U_{t_2}$ 和$b \in U_{t_2} \setminus U_{t_1}$，合并无效条件保证：只要 a 和 b 能够通过某个顶点 $v \in X_t$连接成一条最短路径，这一情况就会由 σ正确编码。因此，H_t 实现了 σ。

关于边数，注意，只有当一条边的两个端点都位于 X_t中时，该边才可能同时属于 H_{t_1} 和 H_{t_2}。这类边在 $|E(H_{t_1})| + |E(H_{t_2})|$中被重复计数。项 $\delta(\sigma)$恰好减去了这些边，因此

$$\begin{aligned} |E(H_t)| &= |E(H_{t_1})| + |E(H_{t_2})| - |E(H_{t_1}) \cap E(H_{t_2})| \\ &= w[t_1, \sigma_1] + w[t_2, \sigma_2] - \delta(\sigma) = w[t, \sigma]. \end{aligned} \tag{2}$$

因此，该不变量在合并节点处成立。

◀

▶ **引理 34**（完备性）。 设 H^* 是针对$(G, P\ k, \tau)$的 PDP 最优解。对于每个节点 $t \in T$，令 $H_t^* := H^* \cap G_t$. 那么，存在一个表 $\sigma^* \in \Sigma_t$，使得 H_t^*实现 σ^* 并且

$$w[t, \sigma^*] \leq |E(H_t^*)|.$$

证明。 我们对良好树分解中的节点使用归纳法来证明该引理。固定一个节点$t \in T$。定义一个表 σ^*： $U_t \times U_t \to \{0, 1\}$，定义方式如下

$$\sigma^*(i, j) = \sigma^*(j, i) = 1 \quad \text{if and only if} \quad d_{H_t^*}(i, j) = d_G(i, j),$$

对所有 i，$j \in U_t$。根据构造，σ^* 是对称的，因此 $\sigma^* \in \Sigma_t$。此外，H_t^*实现了 σ^*。我们证明 σ^* 满足动态规划递推的所有有效性条件，并且

$$w[t, \sigma^*] \leq |E(H_t^*)|.$$

叶节点。如果 t 是叶节点，那么 $G_t = \emptyset$，因此 $H_t^* = \emptyset$。于是 $U_t = \emptyset$ 和 $\sigma^* = \emptyset$。根据动态规划递推，

$$w[t, \sigma^*] = 0 \leq |E(H_t^*)|.$$

引入顶点节点。令 v 为在节点 t处引入的顶点，令 t' 为其子节点。由于 v 在 G_t中是孤立的，因此它在 H_t^*中也孤立。因此，根据构造，对于所有 $u \in U_t$，我们有 $\sigma^*(u, v) = 0$，并且 σ^* 满足节点 t处的有效性条件。因此 $w[t, \sigma^*] < +\infty$。令 $H_{t'}^* = H^* \cap G_{t'}$。根据归纳假设，存在一个表 $\sigma' \in \Sigma_{t'}$，使得$H_{t'}^*$ 实现了 σ'，且 $w[t', \sigma'] \leq |E(H_{t'}^*)|$。由于 v 是孤立的，H_t^* 与 $H_{t'}^*$ 的区别仅在于添加了一个孤立顶点，因此

$$|E(H_t^*)| = |E(H_{t'}^*)|.$$

此外，$U_{t'}$ 中顶点之间的所有最短路径关系在 H_t^* 和 $H_{t'}^*$中都相同，这意味着

$$\sigma' = \sigma^* \restriction_{U_{t'}}.$$

根据引入顶点节点的递推，我们有

$$w[t, \sigma^*] = w[t', \sigma'] \leq |E(H_{t'}^*)| = |E(H_t^*)|.$$

引入边节点。设 t 是一个子节点为 t' 、且引入边 uv 的节点，并设$H_{t'}^* = H^* \cap G_{t'}$。

情况 1： $uv \notin H_t^*$。那么 $H_t^* = H_{t'}^*$。根据归纳假设，存在一个表 $\sigma' \in \Sigma_{t'}$，使得 $H_{t'}^*$ 实现了 σ'，且 $w[t', \sigma'] \leq |E(H_{t'}^*)|$。由于 $uv \notin G_{t'}$，我们有 $\sigma'(u, v) = 0$。此外，H_t^* 和 $H_{t'}^*$ 中的最短路径关系一致，因此 $\sigma^* = \sigma'$。根据递推，$w[t, \sigma^*] = w[t', \sigma']$，于是

$$w[t, \sigma^*] \leq |E(H_{t'}^*)| = |E(H_t^*)|.$$

情况 2： $uv \in H_t^*$。那么 $H_t^* = H_{t'}^* \cup \{uv\}$且 $|E(H_t^*)| = |E(H_{t'}^*)| + 1$。此外，$\sigma^*(u, v) = 1$。加入边 uv 只会为这样的点对创建新的最短路径：$\{i, j\}$ 其在 G 中的最短路径使用 uv，且已与 u 或 v最短路径连通。根据定义，这些恰好是包含在 U_{uv}中的点对。因此，σ^* 和 σ' 的差异仅可能出现在满足 $\{i, j\} \in U_{uv}$的条目 (i, j) 上。于是，当$\sigma(u, v) = 1$且 $\sigma^* \notin \Sigma_t^{\text{edge}-\text{inv}}$ 时，σ'就是递推中考虑的表之一。根据递推，

$$w[t, \sigma^*] \leq w[t', \sigma'] + 1 \leq |E(H_{t'}^*)| + 1 = |E(H_t^*)|.$$

遗忘顶点节点。设 t 是一个子节点为 t' 的节点，该子节点遗忘顶点 v，并设 $H_{t'}^* = H^* \cap G_{t'}$。由于 $T_t = T_{t'}$，可得 $G_t = G_{t'}$，因此根据定义有 $H_t^* = H_{t'}^*$。根据归纳假设，存在一张表 $\sigma' \in \Sigma_{t'}$，使得 $H_{t'}^*$ 实现 σ'，且 $w[t', \sigma'] \leq |E(H_{t'}^*)|$。

情况 1： $v \in S_t^{\downarrow}$。于是 $U_t = U_{t'}$，并且根据构造，$\sigma^* = \sigma'$。由于 H^* 是一个有效的 PDP 解，对于每个终端点 $u \in S \setminus S_t^{\downarrow}$，满足 $\{u, v\} \in \mathcal{P}$，顶点 v 必须与图的其余部分保持最短路径连通。由于 X_t 将 G_t 与图的其余部分隔开，因此从 v 到此类终端点的任何最短路径都必须经过某个顶点 $x \in X_t$，这意味着$\sigma^*(v, x) = 1$ 对某个 $x \in X_t$成立。因此，σ^* 满足 t处递推的有效性条件，并且

$$w[t, \sigma^*] = w[t', \sigma'] \leq |E(H_{t'}^*)| = |E(H_t^*)|.$$

情况 2： $v \notin S_t^{\downarrow}$。于是 $U_{t'}$ 包含额外的一行和一列，对应于 v。根据构造，$\sigma^* = \sigma' \restriction_{U_t}$。由于递推是在所有此类扩展中取最小值，因此我们得到

$$w[t, \sigma^*] \leq w[t', \sigma'] \leq |E(H_{t'}^*)| = |E(H_t^*)|.$$

合并节点。设 t 是一个合并节点，其子节点为 t_1 和 t_2。设 $H_{t_i}^* = H^* \cap G_{t_i}$ 对于 $i = 1, 2$。根据归纳法，对于每个 $i \in \{1, 2\}$，都存在一个表 $\sigma_i \in \Sigma_{t_i}$，使得 $H_{t_i}^*$ 实现了 σ_i 和 $w[t_i, \sigma_i] \leq |E(H_{t_i}^*)|$。

由于 $U_t = U_{t_1} \cup U_{t_2}$ 和 $H_t^* = H_{t_1}^* \cup H_{t_2}^*$，根据构造可知，对于 $i = 1, 2$，有 $\sigma_i = \sigma^* \restriction_{U_{t_i}}$。此外，

$$|E(H_t^*)| = |E(H_{t_1}^*)| + |E(H_{t_2}^*)| - |E(H_{t_1}^*) \cap E(H_{t_2}^*)|.$$

$H_{t_1}^*$ 和 $H_{t_2}^*$ 中同时出现的边，恰好是满足 $x, y \in X_t$ 的边 $xy \in E(G)$（等价地，满足 $d_G(x, y) = 1$ 的 $d_G(x, y)$），且这些边在表 σ_1 和 σ_2 中均有实现。因此，

$$\delta(\sigma^*) = |E(H_{t_1}^*) \cap E(H_{t_2}^*)|.$$

由于 H^* 是一个 PDP 解，对于任意终端对 $\{a, b\} \in \mathcal{P}$，其中 $a \in S_{t_1}^{\downarrow} \setminus S_{t_2}^{\downarrow}$ 且 $b \in S_{t_2}^{\downarrow} \setminus S_{t_1}^{\downarrow}$，都存在一个顶点 $x \in X_t$，使得 a 和 b 在 H_t^*中均与 x 最短路径连通。因此 $\sigma^* \notin \Sigma_t^{\text{join-inv}}$。

根据递推，

$$\begin{aligned} w[t, \sigma^*] &= w[t_1, \sigma_1] + w[t_2, \sigma_2] - \delta(\sigma^*) \\ &\leq |E(H_{t_1}^*)| + |E(H_{t_2}^*)| - \left|E(H_{t_1}^*) \cap E(H_{t_2}^*)\right| \\ &= |E(H_t^*)|. \end{aligned}$$

◀

在确立了可靠性和完备性之后，我们现在转向主要定理。为清晰起见，我们先在此重述该定理，然后给出证明。

▶ **定理 3。** PDP 存在一种算法，其运行时间为 $2^{O\left((\mathrm{tw}+|S|)^2\right)} \cdot |V(G)|^{O(1)}$，其中 tw是 G的树宽。

证明。 设 $(G, \mathcal{P})$ 是一个树宽为 τ 的 PDP 实例。令 $S := \bigcup \mathcal{P}$表示终端集合。令 $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ 为 G 的一个宽度至多为 τ 的良好树分解。

我们在$\mathcal{T}$上应用上述动态规划算法。对于每个节点 $t \in T$和每张表 $\sigma \in \Sigma_t$，我们计算 $w[t, \sigma]$，即实现连通性表 σ的子图 $H_t \subseteq G_t$所含边数的最小值。

根据引理 33，对于每张满足 $w[t, \sigma] < +\infty$的表 σ，都存在一个实现 σ 且恰好包含 $w[t, \sigma]$ 条边的子图 H_t。根据引理 34，对于一个最优的成对距离保持子图 H^*，在每个节点 t处都存在一张表 $\sigma^* \in \Sigma_t$，使得 $H_t^* = H^* \cap G_t$ 实现 σ^* 并且

$$w[t, \sigma^*] \leq |E(H_t^*)|.$$

在根节点 r处，我们有 $X_r = \emptyset$ 和 $S_r^{\downarrow} = S$，因此每个终端点都在 U_r中有所表示。根据定义，集合 $\Sigma^* \subseteq \Sigma_r$ 包含所有满足以下条件的表 σ：它们通过最短路径连接$\mathcal{P}$ 中的每一对终端点。动态规划递推式保证

$$\min_{\sigma \in \Sigma^*} w[r, \sigma]$$

根据可靠性和完备性，这等于最小两两距离保持器中的边数。因此，动态规划计算出最优解的大小。通过对树分解进行标准回溯，并利用各表中存储的见证选择，可以重构出一个最优子图。

最后，该算法的运行时间为

$$|\Sigma_t| \cdot |V(T)|^{O(1)} \leq 2^{O((\tau+|S|)^2)} \cdot |V(G)|^{O(1)},$$

因为每个表都是一个至多包含 $|U_t| \leq (\tau+1)+|S|$ 个顶点的对称二元矩阵，并且 $|T| = O(|V(G)|)$。在每个节点执行的额外操作——包括检查诸如 U_{uv} 这样的集合的最短路径连通性、更新表项以及检查有效性——其运行时间均被引入边节点的运行时间所支配，即 $O\big((\tau+|S|)^2(n+m)\big)$，其中 $n = |V(G)|$ 且 $m = |E(G)|$。因此，总运行时间为 $2^{O((\tau+|S|)^2)} \cdot |V(G)|^{O(1)}$，如所述。◀

▶ 备注 35. 可以通过直接存储相关终端对的连通性信息，进一步优化动态规划状态。特别是，对于常数树宽，运行时间可降至 $2^{O(|\mathcal{P}|)}$。

由包含关系，我们立即得到 SDP 也有相同的结果。

▶ **推论 36.** SDP 可由一个算法求解，其运行时间为 $2^{O\left((\mathrm{tw}+|S|)^2\right)} \cdot |V(G)|^{O(1)}$，其中 tw是 G的树宽。

5 顶点覆盖

本节关注参数为图的顶点覆盖的相关结果。我们首先证明，顶点覆盖数为 3 的图上的 PDP 是 NP 困难的（定理4），然后介绍以顶点覆盖为参数的 SDP 固定参数可解算法（定理 5）。

5.1 PDP 对顶点覆盖数为 3 的图是 NP 困难的

这里，我们证明定理4，并在下文重述该定理以便查阅。

▶**定理4**。 PDP 即使在顶点覆盖数为3的二分图上也是 *NP* 困难的。

我们的证明通过一系列多项式时间归约展开。首先，我们将 3-多路割（3-MWC）归约到一个辅助问题，记为 3-一致列表着色（3-ALC），随后再将 3-ALC 归约到 PDP。

我们先正式定义归约中用到的两个问题。

3-一致列表着色（3-ALC）

输入： 一个图 $G = (V, E)$，一个允许的颜色集合 $\ell_v \subseteq \mathbb{N}$，其中每个顶点均有 $|\ell_v| \leq 3$ x $v \in V$，以及一个整数 k。

问题： 是否存在一种赋值 $\mathcal{X} = \{X_v \subseteq \ell_v \mid v \in V(G) \quad X_v \neq \emptyset\}$，使得对于每条边 $uv \in E(G)$，$X_u \cap X_v \neq \emptyset$ 并且所赋颜色的总数不超过 k，即 $\sum_{v\in V} |X_v| \leq k$?

我们将满足这些条件的颜色分配称为 3-ALC 实例 $\big(G = (V, E), (\ell_v)_{v\in V}\big)$ 的 $\mathcal{X} = (X_v)_{v\in V(G)}$ 有效颜色分配 。如果不存在有效颜色分配，则

实例不可行。在全文中，我们将“列表”和“集合”交替使用，指代顶点的允许的颜色。

最后，对于图 $G=(V,\ E)$ 及其边子集 $F\subseteq E$，我们用 $G-F$ 表示图 $(V,\ E\setminus F)$。

3-多路割（3-MWC）

输入： 连通图 $G=(V,\ E)$、三个互不相同的终端点 $s_1,\ s_2,\ s_3\in V$，以及一个整数k。

问题： 是否存在一个边集合 $F\subseteq E$，其满足 $|F|\leq k$，使得在图 $G-F$ 中，每个终端点都与另外两个终端点断开连接?

我们的出发点是 Dahlhaus 等人证明的经典结论：3-MWC 是 NP 困难的。

▶**定理 37** ([18])。 即使在无权图上，最小 *3-MWC* 也是 *NP* 困难的。

为了证明本节的主要结果定理4，我们结合两种多项式时间归约。首先，我们给出从 3-MWC 到 3-ALC 的多项式时间归约。然后，我们将 3-ALC 归约到 PDP 的一个受限版本，其中图是二分的，顶点覆盖大小至多为 3，且终端对满足特定结构。结合这两种归约可知，PDP 至少与 3-MWC 一样难，而 3-MWC 是 NP 困难的。

为确保清晰易读，每种归约都作为一个引理（引理 38 和 39）单独呈现在一个不编号的小节中，并包含相应的构造。最后，我们通过结合这些引理的结果，重述并证明主定理。

从3-MWC 到3-ALC的归约

我们首先将 3-MWC 归约为 3-ALC。设 $I_{\text{3-MWC}}=\big(G,\ \{s_1,\ s_2,\ s_3\}\ k\big)$ 是一个 3-MWC 实例。我们构造一个对应的 3-ALC 实例 $I_{\text{3-ALC}}=\big(G',\ (\ell_{v'})_{v'\in V(G')}\ k'\big)$，其中$k'=(n+1)m+k$ 且对所有 $v'\in V(G')$ 均成立，具体如下。

设 $n=|V(G)|$ 和 $m=|E(G)|$ 分别表示 G中的顶点数和边数。对于每个顶点 $v\in V(G)$，我们向 G' 中添加一个大小为 m的团 $K_v=\{x_{v1},\ x_{v,\ 2},\ \ldots\ldots,\ x_{v,\ m}\}$。接下来，对于每条边$e=uv\in E(G)$，我们引入一个新顶点 $w'_e\in V(G')$，并将其与 K_u和 K_v中的每个顶点相连。图 11 展示了这一构造。

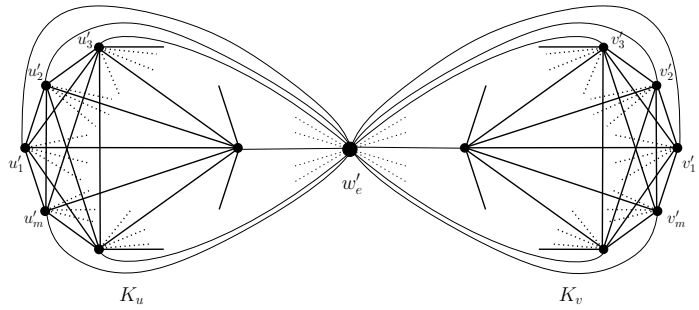
现在，我们定义 G' 中顶点的允许颜色列表。首先，对所有 $v'\in V$（G'），设置 $\ell_{v'}=\emptyset$。对于每个满足 $i\in[3]$的终端点 s_i，将 $\ell_{v'}=\{i\}$ 分配给每个顶点 $v'\in K_{s_i}$。对于所有其余顶点 $v'\in V$（G'）$\setminus\bigcup_{i=1}^{3}V$（$K_{s_i}$），设置 $\ell_{v'}=\{1,\ 2,\ 3\}$。

因此，在 G' 中，每个团 K_{s_i} 都被迫使用颜色 i，而所有其他顶点都可以使用三种颜色中的任意一种。图 12 展示了说明这一构造的示例。

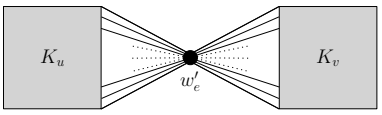
▶**引理 38**。 设 $I_{\text{3-MWC}}=\big(G=(V,E),\ \{s_1,s_2,s_3\}\ k\big)$ 是一个 3-MWC 实例，其中 G 有 n 个顶点和 m 条边。设 $I_{\text{3-ALC}}=\big(G'=(V,E'),\ (\ell_{v'})_{v'\in V(G')}\ k'\big)$ 是按照上述方式构造的 3-ALC 实例。那么，$I_{\text{3-MWC}}$ 存在至多含有 k 条边的 3-MWC，当且仅当 $I_{\text{3-ALC}}$ 存在总大小至多为 $k'=(n+1)m+k$ 的有效颜色分配。

证明。我们首先证明正向。假设 $I_{\text{3-MWC}}=\big(G=(V,E),\{s_1,s_2,s_3\}\ k\big)$存在一个大小至多为 k 的 3-MWC $F\subseteq E(G)$。由 F，我们为实例 $I_{\text{3-ALC}}$构造一个有效的 3-ALC 赋值 X $=(X_{v'})_{v'\in G'}$，其总大小至多为 $k'=(n+1)m+k$。

令 $\mathcal{C}=\{C_1,C_2,\ldots,C_p\}$为 $G-F$ 的连通分量，并令 C_{s_i}表示对于 $i\in[3]$，包含终端点 s_i 的连通分量。



(a) G' 中对应于 G的一条边 $e = (u,\ v)$的构造。



(b) 同一构件的示意图，其中团以正方形表示。

图 11 在 (a) 中，顶点 u 和 v 被替换为团 K_u 和 K_v，每个团都由 m 个顶点组成；此外，引入一个顶点 w'_e，使其与两个团中的所有顶点都相邻。在 (b) 中，以示意图展示了相同的构造，其中团用正方形表示；为清晰起见，后续图中均采用这种表示方式。

对于每个顶点 $v \in V$（G），以及 G' 对应的团中的每个顶点 $u \in K_v$，按如下方式定义分配的颜色集 X_u：

$$X_u = \begin{cases} \{i\}, & v \in C_{s_i} \text{ for some } i \in [3], \\ \{1\}, & v \in C \text{ with } C \in \mathcal{C} \setminus \{C_{s_1}, C_{s_2}, C_{s_3}\} \end{cases}$$

对于每个团 $K_v \subseteq V(G')$，我们定义

$$\gamma(K_v) := \bigcup_{u \in K_v} X_u,$$

我们称其为团 K_v的颜色。根据构造，团 K_v 中的所有顶点颜色都相同，因此 $\gamma(K_v)$始终是一个单元素集合。

用这一记法，对于每条边 $e = uv \in E(G)$，我们将颜色集赋给对应的顶点$w'_e \in V(G')$

$$X_{w'_e} := \gamma(K_u) \cup \gamma(K_v).$$

至此，颜色分配的构造$\mathcal{X}$完成。

现在我们证明$\mathcal{X}$ 是 $I_{3\text{-ALC}}$的有效颜色分配。

首先，我们验证对于所有 $v' \in V(G')$，$X_{v'} \subseteq \ell_{v'}$ 均成立。如果对于某个 $i \in [3]$，$v' \in K_{s_i}$ 成立，那么$X_{v'} = \{i\} = \ell_{v'}$，这是由构造可知的。否则，$v' \notin \bigcup_i K_{s_i}$，因此 $\ell_{v'} = \{1,2,3\}$ 且 $X_{v'} \subseteq \{1,2,3\}$。

接下来，令 $W'_E := \{w'_e \in V(G') \mid e \in E(G)\}$ 为对应于 G的边的顶点集合。考虑 G'的边。对于完全位于团 K_u内的边，所有顶点都被赋予相同的颜色 $\gamma(K_u)$，因此对于任意边 $e' = u'_i u'_j \in K_u$，我们有 $X_{u'_i} \cap X_{u'_j} = \gamma(K_u) \neq \emptyset$.

G' 中剩余的边有一个端点位于 W'_E 中，另一个端点位于某个团中。考虑一个对应于边 $e = uv \in E(G)$的顶点 $w'_e \in W'_E$。根据构造，w'_e 恰好与两个团 K_u 和 K_v中的顶点相邻，即 $N_{G'}(w'_e) = V(K_u) \cup V(K_v)$。根据前文所述的颜色分配，$X_{w'_e} = \gamma(K_u) \cup \gamma(K_v)$。此外，$K_u$ 中的所有顶点（分别是K_v中的所有顶点）都被赋予颜色 $\gamma(K_u)$（分别是 $\gamma(K_v)$）。因此，对每个 $z \in K_u \cup K_v$，可得：

$$X_{w'_e} \cap X_z \neq \emptyset$$

因此，所有邻接约束均得到满足，$\mathcal{X}$ 是 $I_{3\text{-ALC}}$ 的有效颜色分配。

为了界定颜色分配 $\sum_{v' \in V\ (G')}\ |X_{v'}|$ 的总大小，我们按如下方式进行：

$$\sum_{v' \in V(G')} |X_{v'}| = \sum_{K_v \in V(G')} \sum_{v' \in K_v} |X_{v'}| + \sum_{w'_e \in V'_E} |X_{w'_e}|$$

根据构造，对于每个团 $K_v \subseteq G'$，其中所有顶点都被赋予相同的颜色 $\gamma(K_v)$。因此，对于每个 $v' \in K_v$，都有 $|X_{v'}| = 1$。由于每个团有 m 个顶点，且共有 n 个团，所有团中顶点的总贡献为$\sum_{K_v \in V(G')}$ $\sum_{v' \in K_v} |X_{v'}| = n \cdot\ m.$

对于每条边 $e = (u,\ v) \in E(G)$，对应的顶点 $w'_e \in V(G')$ 被赋予其端点所在团的颜色并集：$X_{w'_e} = \gamma(K_u) \cup \gamma(K_v)$，因此 $|X_{w'_e}| \leq 2$. 如果 u 和 v 属于 $G - F$ 中的同一个连通分量，那么 $\gamma(K_u) = \gamma(K_v)$ 且 $|X_{w'_e}| = 1$。否则，$\gamma(K_u) \neq \gamma(K_v)$ 且 $|X_{w'_e}| = 2$。

回顾一下，$|F| \leq k$。根据 3-MWC 的定义，由于 G 是连通的，连接 $G - F$ 中不同连通分量的所有边都必须属于割 F。因此，$G - F$ 中连接不同连通分量的边至多有 k 条。这些边各自最多贡献 2 种颜色，而其余 $m - k$ 条边完全位于某个连通分量内，每条恰好贡献 1 种颜色。因此，边顶点的总贡献满足

$$\sum_{w'_e \in W'_E} |X_{w'_e}| \leq 2|F| + (m - |F|) = m + |F| \leq m + k.$$

结合团和边顶点的贡献，可得

$$\sum_{v' \in V(G')} |X_{v'}| \leq n \cdot m + (m + k) = (n + 1) \cdot m + k = k',$$

由此得到颜色分配总大小的所需上界。

为证明反方向，假设 $I_{3\text{-ALC}}$ 存在一个总大小为 $k' = (n + 1)m + k$ 的有效颜色分配 $\mathcal{X} = (X_{v'})_{v' \in G'}$。由 $\mathcal{X}$，我们为实例$I_{3\text{-MWC}} = \big(G = (V, E), \{s_1, s_2, s_3\}\ k\big)$ 构造一个大小至多为 k 的解 $F \subseteq E(G)$。对于每个满足 $|X_{w'_e}| \geq 2$ 的 $w'_e \in V(G')$，将其对应的边 $e \in E(G)$ 加入割中，即 $F := F \cup \{e\}$。

我们通过反证法证明 F 是一个有效的 3-MWC。为此，假设存在互不相同的 a、$b \in [3]$，使得 s_a 和 s_b 在 $G - F$ 中连通。令 $Q = (s_a$、e_1、v_1、……、v_{q-1}、e_q、$s_b)$ 为 s_a 和 s_b 在 $G - F$ 中的一条连接路径。

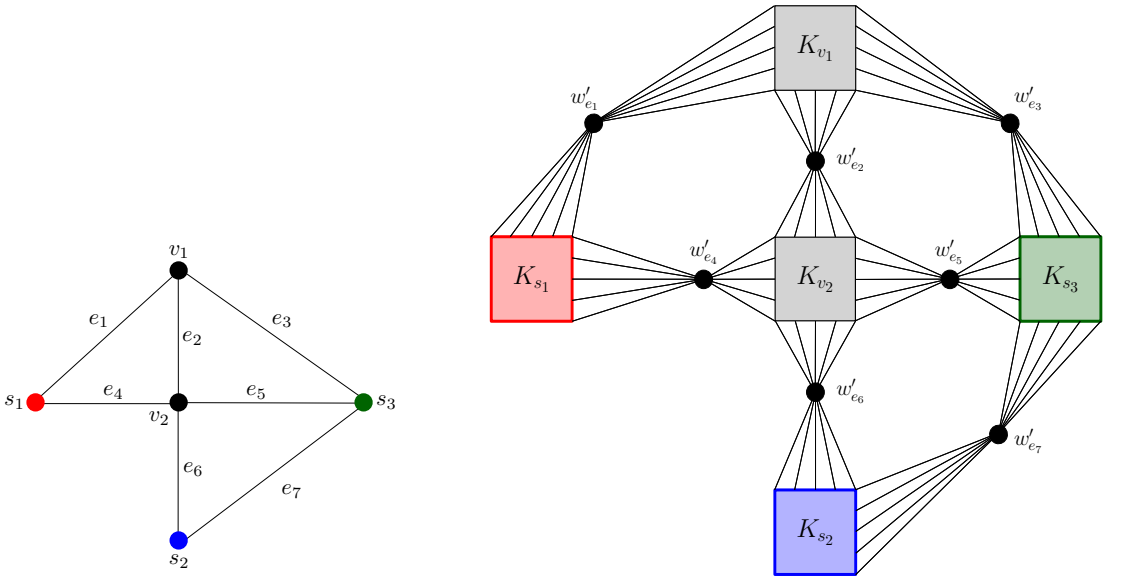
根据 G'的构造，Q 的存在意味着：K_{s_a} 和 K_{v_1} 中的所有顶点都与 w'_{e_1}相邻，K_{v_1} 和 K_{v_2} 中的所有顶点都与 w'_{e_2}相邻，依此类推，直到 $K_{v_{q-1}}$ 和K_{s_b} 中的所有顶点都与 w'_{e_q}相邻。

根据颜色分配约束，每个顶点 $v' \in K_{s_a}$ 都被分配唯一颜色 a，即$X_{v'} = \{a\}$，且每个顶点 $u' \in K_{s_b}$ 都被分配唯一颜色 b，即 $X_{u'} = \{b\}$。此外，由于边 e_1、……、e_q均不属于 F，因此对于所有 $i \in [q]$，都有 $|X_{w'_{e_i}}| = 1$。

因此，要使颜色分配 $\mathcal{X}$ 有效，路径上必须存在某个团 K_{v_t}，其顶点被分配到一个同时包含颜色 a 和 b的集合中。这意味着 K_{v_t}对分配 $\mathcal{X}$的基数贡献了 $2m$，而 G' 的其余顶点各自至少为 $\mathcal{X}$的基数贡献 1 种颜色。因此，由于有 $n - 1$ 个团，每个团包含 m个顶点和 $|W'_E| = m$，颜色分配的总规模满足

$$\sum_{v' \in G'} |X_{v'}| \geq 2m + (n - 1) \cdot m + m = (n + 2) \cdot m > (n + 1) \cdot m + k$$

这与假设 $\sum_{v' \in V\ (G')}\ |X_{v'}| \leq k' = (n + 1) \cdot\ m + k$ 相矛盾。因此，这样的路径不可能存在，而 F 确实是 $I_{3\text{-MWC}} = \big(G = (V,\ E),\ \{s_1,\ s_2,\ s_3\}\ k\big)$ 的一个 3-MWC。



（a） 3 $I_{\text{3-MWC}}$ =-MWC 的一个实例（，）{，，}。（b）由（a）中的实例构造出的对应 3-ALC 实例 $I_{\text{3-ALC}}$ =(G' =(V E')，($\ell_{v'}$)$_{v' \in V(G')}$ k')

图 12 G 中的每个顶点都被替换为一个大小为 $m = 7$ 的团，如正方形所示。对于 G 中的每条边$e = (u, v)$，我们引入一个顶点 w'_e，使其与团 K_u 和 K_v中的所有顶点都相邻。红色、蓝色和绿色分别表示三种允许的颜色 1、2 和 3。 K_{s_1}、 K_{s_2} 和 K_{s_3} 中顶点的允许颜色列表分别仅限于红色、蓝色和绿色，而其余顶点可以使用这三种颜色中的任意一种。

剩下的问题是界定 F 的大小。这归结为界定满足 $|X_{w'_e}| \geq 2$ 的顶点 $w'_e \in V(G')$ 的数量。图 G' 包含 $(n+1) \cdot m$ 个顶点，且每个顶点至少被分配一种颜色。由于 $\mathcal{X}$的总大小至多为 $(n+1) \cdot m + k$，颜色集大小大于 1 的顶点至多有 k 个。根据构造，每个这样的顶点都对应于 F 中的一条边，因此我们得出 $|F| \leq k$。◀

从3-ALC 到PDP的归约

作为归约的第二步，我们引入 PDP 的一个受限变体，并将 3-ALC 归约到该变体。

顶点覆盖-3二分图PDP（VC3-BiPDP）

输入： 一个由二分图 $G =$ （V， E）组成的 PDP 实例，其二分划分为 $V = L \cup R$，其中 $|L| = 3$，以及一个 t 终端对集合和一个整数 k，其中：

$$\mathcal{P} = \Big\{ p_i = \{u_i, v_i\} \mid i \in [t],\ u_i, v_i \in R,\ d_G(u_i, v_i) = 2 \Big\}.$$

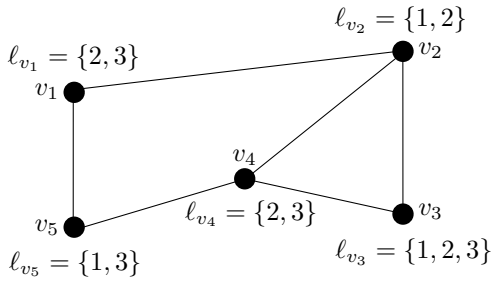
问题： 是否存在一个针对$\mathcal{P}$ 的 PDP，其边数至多为 k？

注意，在 VC3-BiPDP 中，输入图存在一个大小至多为 3 的顶点覆盖，即集合 L，因为 G是二分的。针对每个 3-ALC 实例，从 3-ALC 到 VC3-BiPDP 的归约构造了一个二分图

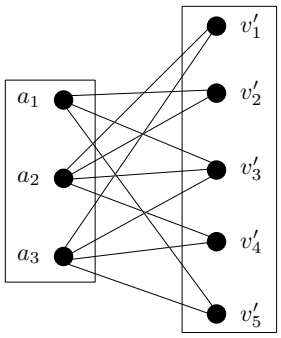
顶点覆盖数为 3 且终端对之间距离为 2 的图，使得一个最小 PDP 恰好对应于原始 3-ALC 实例中的一个最小有效颜色分配。

令 $I_{3\text{-ALC}} = \left(G, (\ell_v)_{v \in V(G)}\ k\right)$ 为一个 3-ALC 实例，使得 $\ell_v \subseteq \{1,2,3\}$ 并且 $\ell_v \neq \emptyset$对每个顶点 $v \in V(G)$ 都成立。我们按如下方式构造一个 VC3-BiPDP 实例 $I_{\text{VC-3-BIP}} = (G', \mathcal{P}', k)$。

图 G' 是二分图，其划分为 L' 和 R'。我们向 L'中添加三个顶点 a_1, a_2, a_3，分别对应三种颜色。对于每个顶点 $v \in V(G)$，我们添加一个对应顶点v' 到 R'，并且当且仅当 $i \in \ell_v$时连接 v' 和 $a_i \in L'$。直观而言，v' 在 L' 中的邻域编码了 v 允许的颜色集合。因此，R' 中的每个顶点恰好与 L' 中代表其对应顶点 G允许颜色的那些顶点相邻。最后，对于每条边$uv \in E(G)$，我们将终端对 (u', v') 添加到 $\mathcal{P}'$中。因此，$\mathcal{P}'$中的终端对数量等于 $|E(G)|$。构造示意见图13。



(a) 一个 3-ALC 实例，其中每个顶点都被分配了一个至多包含三种允许颜色的列表。



(b) 由 (a) 中的 3-ALC 实例构造出的相应 VC3-BiPDP 实例

图13 从 3-ALC 实例 $\left(G,\ (\ell_v)_{v \in V(G)}\ k\right)$ 到 VC3-BiPDP 实例 $\left(G' = (L', R', E'),\ \mathrm{P}', k\right)$的归约示意图。$G$ 中的每个顶点都映射到 G' 的右侧分区中的一个顶点，而左侧分区中的三个顶点 $\{a_1, a_2, a_3\}$ 表示可用颜色。终端对 $\mathrm{P}' = \left\{(v'_1, v'_2),\ (v'_1, v'_5),\ (v'_2, v'_3),\ (v'_2, v'_4),\ (v'_3, v'_4),\ (v'_4, v'_5)\right\}$ 对应于 G 的边。

▶ **引理 39.** 设 $I_{3\text{-ALC}} = \left(G, (\ell_v)_{v \in V(G)}\ k\right)$ 是一个 3-ALC 实例，其中 $\ell_v \subseteq \{1,2,3\}$，且对所有 $v \in V(G) \ell_v \neq \emptyset$ 均成立。设 $I_{VC\text{-}3\text{-}BIP} = (G' = (L', R', E'), P', k)$ 是如上构造的 VC3-BIPDP实例。那么，$I_{3\text{-ALC}}$ 存在一个总大小至多为 k 的有效颜色分配，当且仅当 $I_{VC\text{-}3\text{-}BIP}$ 存在一个至多包含 k 条边的 PDP 。

证明。 为说明正向方向，假设 $I_{3\text{-ALC}}$ 存在一个有效颜色分配 $\mathcal{X} = (X_v)_{v \in V(G)}$ 满足 $\sum_{v \in V} |X_v| = k$。

我们如下构造一个 PDP $H \subseteq G'$，用于处理 $\mathcal{P}'$ 中的终端对，且包含 k 条边。对于每个顶点 $v \in V(G)$ 和每种颜色 $i \in X_v$，我们将边 $v'a_i$ 加入 H。注意，由于 $i \in \ell_v$，根据构造，边 $v'a_i$ 存在于 $E(G')$ 中。

现在证明 H 是一个 PDP。考虑任意终端对 $(u', v') \in \mathrm{P}'$。根据 P' 的定义，G' 中的每个终端对都对应 G中的一条边，因此顶点 u 和 v 在G中相邻。由于 X 是有效颜色分配，因此 $(X_u \cap X_v) \neq \emptyset$。令 $i \in (X_u \cap X_v)$。根据 G' 的构造，u' 和 v' 都与 a_i 相邻，因此 $d_{G'}(u', v') = 2$。由此，只需证明在 H中，u' 和 v' 之间存在一条长度为 2 的路径。由于 $i \in (X_u \cap X_v)$，边 $u'a_i$和 $v'a_i$ 包含在 H中。因此，H 包含一条连接 u' 和 v'的长度为 2 的路径，从而保留了终端对 (u', v')之间至少一条最短路径。

还需界定 H 的大小。对于每个对应于 $v \in V$（G）的顶点 $v' \in V$（H），我们恰好将与其关联的 $|X_v|$ 条边加入 H。因此，$|E$（H）$| = \sum_{v \in V\ (G)} |X_v| = k.$ 由于 G'（且

因此 H）是二分的，并且所有终端对都位于右侧划分中， H 的每条边在此求和中恰好计算一次。因此， H 是针对 $\mathcal{P}'$ 、具有 k 条边的 PDP。

为证明反向结论，假设 $I_{\text{VC-3-BIP}}$ 有一个包含 k 条边的 PDP H 。我们构造一个有效颜色分配 $\mathcal{X} = (X_v)_{v\in V(G)}$，使得 $I_{3\text{-ALC}}$ 满足 $\sum_{v\in V(G)} |X_v| = k$。

我们先处理一个退化情形。如果存在一个终端对 $(u', v') \in \mathcal{P}'$ ，使得 $N_{G'}(u') \cap N_{G'}(v') = \emptyset$，那么 $I_{3\text{-ALC}}$ 就是一个平凡的否实例。事实上，终端对 (u', v') 对应一条边 $uv \in E(G)$，而条件 $N_{G'}(u') \cap N_{G'}(v') = \emptyset$ 意味着 $(\ell_u \cap \ell_v) = \emptyset$。因此， G 不存在有效颜色分配。

因此，我们可以假设，对于每个终端对 $(u', v') \in \mathcal{P}'$，都有 $d_H(u', v') = d_{G'}(u', v') = 2$。由于 G' 是二分的，每条这样的路径都具有形式 $u' - a_i - v'$，其中 $i \in [3]$。

对于每个终端对 $(u', v') \in \mathcal{P}'$，仅当颜色 $i \in \big(N_H(u') \cap N_H(v')\big)$ 尚未出现在两者中时，才将其添加到 X_u 和 X_v。也就是说，设 $X_w := X_w \cup \{i\}$ 为 $w \in \{u, v\}$，若 $i \notin X_w$.

现在我们证明 $\mathcal{X}$ 是有效颜色分配。只需证明，对于每条边 $(u, v) \in E(G)$，都有 $X_u \subseteq \ell_u$、$X_v \subseteq \ell_v$，且 $(X_u \cap X_v) \neq \emptyset$。

考虑任意一条边 $uv \in E(G)$。对应的终端对 (u', v') 满足 $d_H(u', v') = 2$，因此至少存在一个 $i \in [3]$ ，使得 $u'a_i$ 和 $v'a_i$ 都属于 $E(H)$。根据 G' 的构造，且由于 $E(H) \subseteq E(G')$，可推出 $i \in (\ell_u \cap \ell_v)$。由于 i 同时被添加到 X_u和 X_v，我们得到 $i \in (X_u \cap X_v) \neq \emptyset$。此外，只有当颜色属于相应的允许列表时，才会将其添加到 X_u 和 X_v 。因此， $X_u \subseteq \ell_u$ 且 $X_v \subseteq \ell_v$。由此，$\mathcal{X}$ 是有效颜色分配。

还需要界定颜色分配的总大小。对于每种颜色 $i \in [3]$，令 $\alpha(i) = \big|\{v \in V(G) \mid i \in X_v\}\big|$ 表示 G 中其分配的颜色集包含颜色 i的顶点数。通过重复计数可得 $\sum_{v\in V(G)} |X_v| = \sum_{i\in[3]} \alpha(i)$。

注意，根据 $\mathcal{X}$ 的构造，只有当存在一个终端对 (u', v') 满足 $u' \in R'$，使得 $a_i \in (N_H(v') \cap N_H(u'))$ 时，才会将颜色 i 添加到顶点 v 的集合 X_v 中。此外，每种颜色在任意集合 X_v 中最多添加一次。注意，当且仅当 $v'a_i \in E(H)$ 时， $a_i \in N_H(v')$ 成立。

因此，某个集合 X_v 中颜色 i 的每次出现，都唯一对应于 H 中与 a_i 相 incident 的一条边，即 (a_i, v')。反过来，每条与 a_i 相 incident 的边恰好对应这样的一次出现。因此，所分配的集合中包含颜色 i 的顶点数量，恰好等于 a_i 在 H 中的度，也就是说，

$$\sum_{v\in V(G)} |X_v| = \sum_{i\in[3]} \alpha(i) = \sum_{i\in[3]} |N_H(a_i)| = |E(H)| = k.$$

因此， $I_{3\text{-ALC}}$ 存在一种总大小至多为 k 的有效颜色分配，从而完成证明 .

注意，不必将 $N_{G'}(u') \cap N_{G'}(v')$ 中的所有颜色都加入；也可以选择单个索引 $i \in N_H(u') \cap N_H(v')$，只加入对应的颜色。分析保持不变，因为界限仅取决于 H 中顶点 a_i 的度。

最后，由于我们已经证明 VC3-BiPDP 是 NP 困难的，因此立即可知，一般的 PDP 问题也是 NP 困难的。◄

我们重述本节的主要定理，并利用前面几个小节中建立的归约给出一个简明证明。

►**定理4**。 PDP 即使在顶点覆盖为3的二分图上也是 *NP* 困难。

证明。 设 $I_{3\text{-MWC}}$ 是一个 3-MWC 实例。根据引理 38， $I_{3\text{-MWC}}$ 可以在多项式时间内转化为一个 $I_{3\text{-ALC}}$ 的 3-ALC 实例，使得 $I_{3\text{-MWC}}$ 是一个是实例，当且仅当 $I_{3\text{-ALC}}$ 存在一个规模有界的有效颜色分配。

接下来，根据引理 39， $I_{3\text{-ALC}}$ 可以在多项式时间内转化为一个受限的$I_{\text{VC3-BiPDP}}$ 实例，该实例是定义在顶点覆盖数为 3 的二分图上的 PDP 实例，并且 $I_{3\text{-ALC}}$ 是一个是实例，当且仅当 $I_{\text{VC3-BiPDP}}$ 存在一个规模有界的成对距离保持子图。

由于 $I_{\text{VC3-BiPDP}}$ 是 PDP 的一个受限版本，这一系列归约表明，PDP 至少与 3-MWC 一样难。已知 3-MWC 是 NP 困难的 [18], ，因此即使在顶点覆盖数为 3 的二分图上，最小 PDP 也是 NP 困难的。

◀

5.2 SDP 可由顶点覆盖实现固定参数可解

本节将证明，当参数取输入图的顶点覆盖大小时，子集距离保持子图（SDP）是固定参数可处理（FPT）的。具体而言，我们将证明定理 5，现将其重述如下，以便参考。

▶ **定理 5.** SDP 存在一个 $2^{O(2^{\text{vc}})} \cdot |V(G)|^{O(1)}$-时间算法，其中 vc 是 G的最小顶点覆盖的大小。

众所周知，可以在 $O(1.2738^k + kn)$ 时间内计算出大小至多为 k 的最小顶点覆盖 C [13]。

设 (G, S, k) 是 SDP 的一个实例，其中 G 存在大小至多为 k 的顶点覆盖。固定一个顶点覆盖 $C \subseteq V(G)$，使得 $|C| \leq k$，并令 $I := V(G) \setminus C$。注意， I 是独立集；因此，对于每个非孤立顶点 $v \in I$， v 的所有邻居都必须属于 C，也就是说， $N_G(v) \subseteq C$。

我们根据等价关系将 I 划分为等价类 $\{I_1, \ldots, I_q\}$

$$u \sim v \iff N_G(u) = N_G(v).$$

注意，这类等价类的数量至多为 $q \leq 2^k$。对于每个 $i \in [q]$，令 $N_G(I_i)$ 表示 G 中 I_i 内各顶点的公共邻域，并令 $S_i := S \cap I_i$ 表示 I_i 中包含的终端集合。

下面的引理表明，存在一个最优解，使得同一等价类中的所有终端点都以对称的方式连接。

▶**引理 40**。 对于大小至多为 h 的每个 SDP H，都存在一个大小至多为 h 的 SDP H'，使得对于每个等价类 I_i，所有终端 $t \in S_i$ 都满足 $N_{H'}(t) = X_i$，其中存在某个 $X_i \subseteq N_G(I_i)$。

证明。 令 H 为一个大小至多为 h 的 SDP。如果对于每个等价类 I_i，其中的所有终端 S_i 在 H 中已经具有相同的邻域，那么结论成立。否则，假设存在t_1、$t_2 \in S_i$，使得 $N_H(t_1) \neq N_H(t_2)$。不失一般性，假设 $|N_H(t_1)| \leq |N_H(t_2)|$。

我们通过设置 $N_{H'}(t_2) = N_H(t_1)$ 和 $N_{H'}(v) = N_H(v)$ 来构造 H'，对所有 $v \neq t_2$ 均如此。也就是说，我们删除 H 中与 t_2 相 incident 的所有边，并在 t_2 与 $N_H(t_1)$ 中的每个顶点之间添加边。由于 $N_H(t_1) \subseteq N_G(t_1) = N_G(t_2)$，因此可得 $H' \subseteq G$。

我们声称 H' 是一个有效的 SDP。固定任意一个终端点 $t_j \in S$。

若 $t_j \in N_G(I_i)$，则 $d_G(t_1, t_j) = d_G(t_2, t_j) = 1$。由于 H 是一个 SDP，边 t_1t_j 和 t_2t_j 均属于 H，因此 $t_j \in N_H(t_1) \cap N_H(t_2)$。根据构造，这些边在 H' 中得到保留，这意味着 $d_{H'}(t_1, t_j) = d_{H'}(t_2, t_j) = 1$，且 H' 是一个有效的 SDP。

若 $t_j \in S \setminus N_G(I_i)$，由于 H 是一个 SDP 且 I 是一个独立集，存在顶点 $u \in N_H(t_2)$ 和 $v \in N_H(t_1)$，使得

$$d_H(t_j, t_2) = d_H(t_j, u) + 1 = d_G(t_j, t_2) \qquad \text{and} \qquad d_H(t_j, t_1) = d_H(t_j, v) + 1 = d_G(t_j, t_1).$$

反设 $d_H(t_j, u) < d_H(t_j, v)$。那么，用 u 连接 t_1，而不是 v，会使 t_1 和 t_j 之间在 H 中的路径严格变短，这与 H 保持

最短路径距离这一事实矛盾。因此，$d_H(t_j, u) = d_H(t_j, v)$，并且用边 t_2v 替换边 t_2u 可保持 t_2 和 t_j 之间的距离。

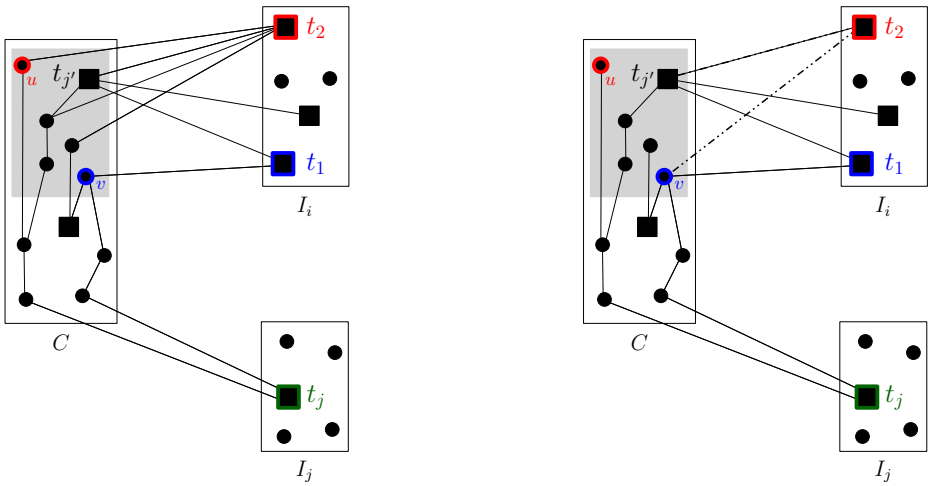
由于 $t_1 \sim t_2$，我们有 $N_G(t_1) = N_G(t_2)$，因此 $v \in N_G(t_2)$。所以，边 t_2v 存在于 G 中，并且可以用于构造。

如果 u 或 v 是终端点，那么它在 H中与 t_1 相邻。根据构造，在 H'中也会为 t_2 引入相同的邻接关系，因此该论证仍然成立，无需修改。

最后，由于 $|N_H(t_1)| \leq |N_H(t_2)|$，我们得到

$$|E(H')| = |E(H)| - |N_H(t_2)| + |N_H(t_1)| \leq |E(H)|.$$

因此，H' 是一个大小至多为 h的 SDP。通过反复应用这一变换，我们得到一个图，其中属于同一等价类的所有终端都具有相同的邻域。变换从 H 到 H'的示意图见图14。◀



(a) SDP H，其中终端 $t_1 \sim t_2$，但邻域不同，即 $N_H(t_1) \neq N_H(t_2)$。**(b)** 通过删除 $\delta_H(t_2)$中的所有边，并在 t_2 与 $N_H(t_1)$ 中的所有顶点之间添加边（虚线），得到的 SDP H'。

图14 引理 40 的证明示意图。为清晰起见，图中未显示所有顶点和边。在每幅图中，顶点覆盖 C 在左侧以矩形表示，等价类则位于右侧。终端以方形表示，非终端点以圆形表示。阴影区域表示公共邻域 $N_G(I_i)$。边分别属于 (a) 中的 SDP H 和 (b) 中的 SDP H'。$S \setminus \{t_1, t_2\}$ 中所有可能的相关终端位置均已示出。顶点 v 和 u 分别位于 t_j 与 t_1 之间以及 t_j 与 t_2之间的最短路径上。**(a)** $N_G(I_i)$ 中的终端与 H 中的 t_1 和 t_2 均相邻，且在 H'中仍然如此。**(b)** 替换与 t_2 关联的边可保持终端之间的所有最短路径距离。

因此，根据引理 40，我们不失一般性地假设存在一个最优 SDP，使得同一等价类中的所有终端都具有相同的连接。

下一个引理表明，对于每个等价类，都可以假设一个最优稀疏距离保持器至多包含一个非终端顶点。

▶**引理 41**。对于每个大小至多为 h 的 SDP H，存在一个大小至多为 h 的 SDP H'，使得对于每个 $i \in [q]$，均有 $\left|V(H') \cap (I_i \setminus S_i)\right| \leq 1$.

证明。令 H 为一个大小至多为 h 的 SDP h。假设存在一个索引 $i \in [q]$ 使得$|V(H) \cap (I_i \setminus S_i)| \geq 2$。令 $u, v \in V(H) \cap (I_i \setminus S_i)$ 为该集合中的两个不同非终端点。

现在我们按如下方式构造 H'。在 H 中删除顶点 u 及其所有关联边。对于每个顶点 $x \in N_H$（u），添加边 vx。H 中的所有其他顶点和边保持不变。更正式地，我们定义 $H' =$（V（H'） E（H'））， 其中

$$V(H') := V(H) \setminus \{u\} \quad \text{and} \quad E(H') := \big(E(H) \setminus \delta_H(u)\big) \cup \{vx \mid x \in N_H(u)\}.$$

我们声称 H' 是一个有效的 SDP。根据观察 8，不失一般性，我们可以假设 $V(H) \cap (I_i \setminus S_i)$ 中的每个顶点都位于至少一条终端对之间的最短路径上。令 $T_u := \{(t_i,\ t_j) \mid t_i,\ t_j \in S,\ \ i \neq j,$ 以及 u 属于某条 $t_i - t_j$ 最短路径 $\}$ 的终端对集合，其中 u 位于 H 中的对应最短路径上。类似地定义 T_v。

任取一个终端对 $(t_1^u,\ t_2^u) \in T_u$，并考虑它们在 H中的一条最短路径。由于 u 是一个非终端点，且 $u \in I_i$，这条路径经由两个不同的邻居$u_1,\ u_2 \in N_H(u) \subseteq C$进入并离开 u，因此

$$d_H(t_1^u, t_2^u) = d_H(t_1^u, u_1) + d_H(u_1, u) + d_H(u, u_2) + d_H(u_2, t_2^u) = d_H(t_1^u, u) + 2 + d_H(u_2, t_2^u). \quad (3)$$

由于 $u,\ v \in I_i$，因此有 $N_G(u) = N_G(v)$。特别地，$u_1,\ u_2 \in N_G(v)$ 和 $d_G(u_1,\ v) = d_G(v,\ u_2) = 1$。因此，添加边 vu_1 和 vu_2，即可在 H' 中得到一条长度相同的 t_1^u–t_2^u路径：

$$\begin{aligned} d_{H'}(t_1^u, t_2^u) &= d_{H'}(t_1^u, u_1) + d_{H'}(u_1, v) + d_{H'}(v, u_2) + d_{H'}(u_2, t_2^u) \\ &= d_H(t_1^u, u) + 2 + d_H(u_2, t_2^u) \overset{(3)}{=} d_H(t_1^u, t_2^u). \end{aligned} \quad (4)$$

因此，t_1^u 与 t_2^u 之间的最短路径距离在 H'中得以保持。

最后，注意在构造 H'时，并未移除与 v 相 incident 的任何边。因此，所有在 H 中的最短路径经过 v 的终端对，在 H'中仍有一条长度相同的最短路径。所有其他距离均保持不变。

我们得出结论：H' 是一个大小至多为以下数值的 SDP：

$$|E(H')| = |E(H)| - |\delta_H(u)| + |\delta_H(u)| = |E(H)| \leq h.$$

通过反复执行此操作直至无法继续，我们得到一个 SDP，其中每个等价类 I_i至多包含一个非终端点。图 15 展示了一个示例。

◀

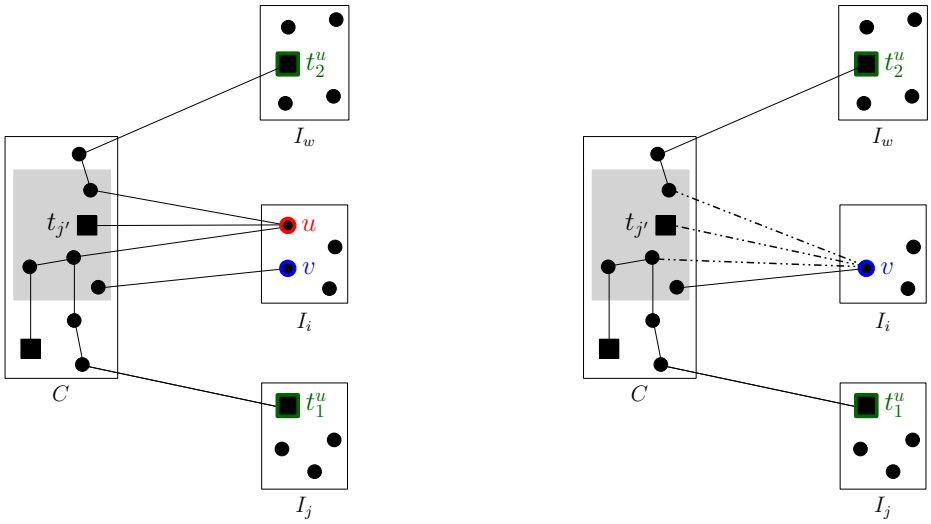
下一个引理表明，如果一个等价类包含一个终端点，那么存在一个最优 SDP，其仅使用该等价类中的终端点。

▶ **引理 42.** 对于任意大小至多为 h的SDP H，如果存在一个指标 $i \in [q]$，使得 $S_i \neq \emptyset$，且 $V(H) \cap I_i \not\subseteq S_i$，那么存在一个大小至多为 h 的SDP H'，满足 $V(H') \cap I_i \subseteq S_i$。

证明。 令 H 为一个大小至多为 h的 $S_i \neq \emptyset$，其中 $i \in [q]$ 且 $V(H) \cap I_i \not\subseteq S_i$。根据引理 41，我们可以假设 $|V(H) \cap I_i \not\subseteq S_i| = 1$。令 $v \in V(H) \cap I_i$ 为等价类 I_i 中属于 H的非终端顶点。令 $t_i \in S_i$ 为 I_i中的任意一个终端点。

我们通过移除 v 及 $\delta_H(v)$ 中的所有边，并将 t_i 与 $N_H(v)$ 中的所有顶点相连，来构造 H'。

证明 H' 中所有终端距离均得以保留的论证，与引理 41 的证明所用论证相同。唯一的区别是，v 被同一等价类中的一个终端点 $t_i \in S_i$ 替换，因此它在 G 中具有相同的邻域。◀



(a) 当至少两个来自同一等价类的非终端点 $u \sim v$ 被纳入解中时的 SDP H。**(b)** 移除 u 及其所有相邻边，并将 v 与 N_H（u）中的所有顶点相连（虚线边）后得到的 SDP H'。

图 15 引理 41 的证明示意图。(a)解中使用了来自同一等价类 I_i 的两个非终端顶点 u 和 v。(b) 该构造移除 u，并将 v 与 N_H（u）中的所有顶点相连。所有终端点之间的最短路径距离均得以保持。

注意，引理 41 表明，每个等价类在最优解中至多需要一个非终端顶点。引理 42 进一步强化了这一结论：如果某个类包含一个终端点，那么该类中就不需要任何非终端顶点。

我们根据这些引理所确立的结构性质，提出两条归约规则。对于每个满足 $i \in [q]$ 的等价类 I_i，我们定义如下规则：

(i) 非终端删除：如果 $I_i \cap S \neq \emptyset$，则丢弃 I_i 中所有非终端顶点。**(ii) 代表元选择**：如果 $I_i \cap S = \emptyset$，则在 I_i 中保留一个任意的代表非终端顶点，并移除该类中的所有其他顶点。

有了所有性质和定义后，我们证明本节的主要定理。为方便起见，现重述如下：

▶ **定理 5.** SDP 可由一个运行时间为 $2^{O(2^{\mathrm{vc}})} \cdot |V(G)|^{O(1)}$ 的算法求解，其中 vc 是 G 的最小顶点覆盖的大小。

证明。 设 (G, S, k) 是一个 SDP 实例，且 G 有一个大小至多为 k 的顶点覆盖。我们描述一种分支算法，它构造所有符合引理 40、41 和 42 中所确立结构性质的候选解，并选出大小最小的解。

首先，使用标准的顶点覆盖 FPT 算法，我们计算一个顶点覆盖 $C \subseteq V(G)$，其中 $|C| \leq k$。令 $I := V(G) \setminus C$。然后，我们根据 I 中的邻域将其划分为等价类 $\{I_1, \ldots, I_q\}$，其中 $q \in \mathbb{N}$；当且仅当 $N_G(u) = N_G(v)$ 时，$u \sim v$。

接下来，我们穷尽式地应用归约规则**非终端删除** 和**代表元选择**。经过这些归约后，对于每个等价类 I_i，$I_i \subseteq S$ 或 $V(H) \cap I_i$ 中恰好包含一个非终端顶点。

然后，我们枚举由顶点覆盖诱导的所有可能子图。令 $G[C]$ 表示由 C诱导的 G 的子图，并令$\mathcal{H}_C$ 表示 $G[C]$的所有子图构成的族。对于每个 $H_C \in \mathcal{H}_C$，我们构造将 H_C 扩展到 I中顶点的候选解。

对于每个等价类 I_i，I_i 中的所有顶点只能与 $N_G(I_i) \subseteq C$ 中的顶点相连。我们对 $N_G(I_i)$ 的所有可能子集进行分支，包括空集 $\emptyset$。形式化地，定义

$$\mathcal{H}_i := \{X \mid X \subseteq N_G(I_i)\}.$$

对于每个 $X \in \mathcal{H}_i$，该分支对应于在最终解中选择 X 作为等价类 I_i 在 C 中的邻域。也就是说，解图中出现的 I_i 的每个顶点都恰好与 X 中的顶点相连。

分支过程的一片叶节点对应于一个图

$$H = H_C \cup \bigcup_{i=1}^{q} E_i,$$

其中，对于每个等价类 I_i 以及沿分支选取的每个子集 $X_i \in \mathcal{H}_i$，我们定义

$$E_i := \{vx \mid v \in I_i,\ x \in X_i\}.$$

令 r_i 表示穷尽应用归约规则后，I_i 中出现在 H 内的顶点数量。由引理 41 和 42 可知，若 $S_i \neq \emptyset$，则有 $r_i = |S_i|$；否则 $r_i \in \{0,1\}$。在 H 中仅保留与这些 r_i 个顶点关联的边。因此，I_i 对 H 的边集的贡献为 $|E_i| = r_i \cdot |X_i|$.。于是，H 的总大小为

$$|E(H)| = |E(H_C)| + \sum_{i=1}^{q} r_i \cdot |X_i|.$$

因此，尽管每个等价类只进行一次分支决策，所得图 H 仍包含该决策为 I_i 的每个顶点所确定的全部边，这使我们能够正确评估 SDP 性质以及 H 的大小。在所有此类图中，我们选择一个大小最小且保持终端之间所有两两最短路径距离的图。

等价地，对于每个固定的子图 $H_C \subseteq G[C]$，分支过程可视为深度为 q 的有根搜索树，其中深度为 i 的节点遍历 $\mathcal{H}_i$ 中的所有子集，这些子集对应于等价类 I_i 与顶点覆盖之间可能的连接选择。

我们论证该算法构造了一个最优解。

完备性。令 $H^\star$ 为 G的一个最优稀疏距离保持器。根据引理 40，我们可以假设属于同一等价类的所有终端点在 $H^\star$ 中具有相同的邻域。根据引理 41，每个等价类在 $H^\star$ 中至多包含一个非终端顶点。最后，根据引理 42，如果 $S_i \neq \emptyset$，那么 $H^\star$ 不包含来自 I_i的非终端顶点。

因此，$H^\star$ 完全由以下内容描述：(i) 其在顶点覆盖 C 上的限制；(ii) 对于每个等价类 I_i，集合 $N_G(I_i)$ 的子集，类 I_i 中的顶点与该子集中的顶点相连。因此，$H^\star$对应于算法考察的某个分支。

最优性。对于算法构造的每个候选图 H，我们都会明确验证它是否保持终端之间的所有两两最短路径距离。在所有有效候选图中，算法输出规模最小的一个。由于算法考虑了所有满足上述结构性质的最优稀疏距离保持器，因此返回的解是最优的。

接下来，我们界定运行时间。H_C至多有 $2^{|C|}$ 种选择。对于每个等价类 I_i，我们至多分支为 $2^{|N_G(I_i)|} \leq 2^{|C|}$ 种可能。由于 $q \leq 2^{|C|}$，候选项的总数

至多为 $2^{|C|\cdot 2^{|C|}}$. 。对于每个候选图，我们可以通过在 O 中计算终端之间的全点对距离，在 $n+m$（$|S|^2$）时间内验证 SDP 性质。因此，总运行时间为 O（$2^{|C|(2^{|C|}+1)}$）· O（$n+m$）$|S|^2$ $=2^{O(2^{|C|})}\cdot O$（k^2（$n+m$））。至此，定理 5 得证。◀

6 W[1]-硬度（按解的大小）

本节中，我们证明按解的大小参数化的 SDP 是 W[1]-困难的。我们通过从按解的大小参数化的多色团问题进行参数化归约来证明这一点。

我们先回顾这个问题。

多色团（MCC）

输入： 一个图 $G=(V,E)$、一个整数 k，以及将 V 划分为 k 个独立集的一个划分（V_1, V_2, .. . , V_k）。

问题： 是否存在顶点的一个子集 $X\subseteq V$，使得对于每个$i\in[k]$ $G[X]$ ， $|X\cap V_i|=1$ 都是一个团?

已知，以解的大小为参数的 MCC 问题是 W[1]-困难的 [17]。我们给出从以解的大小为参数的 MCC 到 SDP 的一个参数化归约，并得到定理 6 的结论，现重述如下。

▶**定理 6。** SDP 以解的大小为参数是 *W*[1]-困难的。

证明。 设 $I_{\mathrm{MCC}}=\big(G=(V,\ E),\ (V_1,\ .....,\ V_k)\ k\big)$ 是一个 MCC 实例，其中 k 是参数。我们构造一个 SDP 实例，记为 $I_{\mathrm{SDP}}=\big(G'=(V'E')\ S,\ k'\big)$，其中 $k'=k+\binom{k}{2}$。

首先，我们按如下方式构造图 $G'=(V'E')$。对于每个 $i\in[k]$，创建一份顶点集V_i的副本，记作 V'_i。对于每个顶点 $v\in V_i$，用 $v'\in V'_i$ 表示其副本。对于每条边 $uv\in E(G)$，其中 $u\in V_i$ 和 $v\in V_j$，添加边 $u'v'$ 至 E'。对于每个 $i\in[k]$，添加一个终端顶点 t_i 至 V'_i，并将 t_i 与 V'_i 中的所有顶点相连。令 $S=\{t_i\mid i\in[k]\}$ 为终端集合。$(G'=(V'E')\ S,\ k')$ 是构造得到的 SDP 实例。关于该归约的示意图，请参见图 16。

我们证明， I_{MCC} 是一个是实例，当且仅当 I_{SDP} 是一个是实例。等价地， I_{MCC}有一个大小至多为 k 的解，当且仅当 I_{SDP} 有一个大小至多为 $k'=k+\binom{k}{2}$的解。

为说明正向结论，假设 G 包含一个具有 k 个顶点的多彩团 C，其中 $V(C)$ 和 $E(C)$ 分别表示 C的顶点集合和边集合。根据定义，对于每个 $i\in[k]$，我们有 $|V(C)\cap V_i|=1$。令 c_i 表示 $V(C)\cap V_i$中的唯一顶点。于是，对于每个i， $j\in[k]$ 满足 $i\neq j$, C 的对象，都包含边 c_ic_j；该边是 C 中连接 V_i 和 V_j的唯一边。

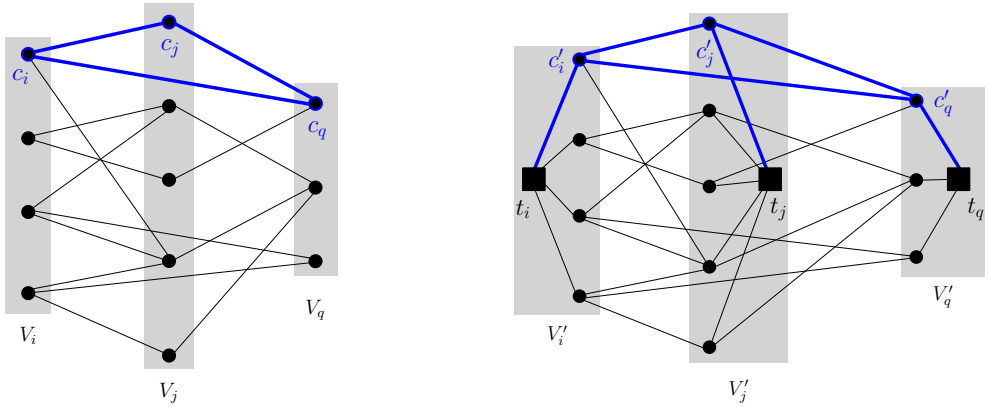
我们如下构造 $(G',\ S)$ 的子集距离保持器 H 。

对于每个 $c_i\in V(C)$，令 $c'_i\in V'_i$为其在 G'中的对应顶点。令 $V(H)=S\cup\{c'_1,\ c'_2,\ .....,\ c'_k\}$。对于每条边 $c_ic_j\in E(C)$，将对应的边 $c'_ic'_j$ 添加到 $E(H)$ 中。此外，对于每个 $i\in[k]$，将边 c'_it_i 添加到 H。等价地，

$$E(H)=\{c'_it_i\mid i\in[k]\}\cup\{c'_ic'_j\mid i,j\in[k]\text{ with }i\neq j\}.$$

至此， H 的构造完成。

为证明 H 是有效的子集距离保持器，首先注意，对于任意 $i,j\in[k]$ 且满足$i\neq j$， G' 中 t_i 与 t_j 之间的最短路径为 $\pi_{G'}(t_i,t_j)=\langle t_i,c'_i,c'_j,t_j\rangle$，因此 $d_{G'}(t_i,t_j)=3$。由于这些边都包含在 H中，因此 $d_H(t_i,t_j)=3=d_{G'}(t_i,t_j)$，所以 H 是有效的子集距离保持器。



(a) 一个 MCC 实例。 **(b)** 对应的 SDP 实例。

图 16 从 MCC 实例 $\left(G=(V,E),(V_1,\ldots,V_k)\ k\right)$ 到 SDP 实例 $\left(G'=(V'\,E')\ S,k'\right)$ 的归约示意图。 G 中的顶点集 V_i、V_j 和 V_q 以灰色突出显示，且各自构成一个独立集。对于 G 中的每个顶点集 V_i，我们在 G' 中创建一个对应的副本 V'_i，并引入一个终端点 t_i。每个终端点 t_i 都通过边与 V'_i 中的所有顶点相连。正方形表示添加到 G' 的终端点 $S=\{t_i,t_j,t_q\}$，圆形表示非终端点。在 (a) 中，蓝色的边和顶点表示 MCC 实例的解子图；在 (b) 中，蓝色的边和顶点表示 SDP 实例的解子图。

为界定 H的大小，注意到 H 对每个 $i\in[k]$ 都包含一条边 c'_it_i，并对每个满足 $i\neq j$的 i, $j\in[k]$ 都包含一条边 $c'_ic'_j$，因此 $|E(H)|=k+\binom{k}{2}=k'$。

为证明反方向，假设 G' 有一个包含 $k+\binom{k}{2}$条边的子集距离保持器 H。首先注意，对于每个 $i\neq j$， G'中任何连接 t_i 和 t_j 的路径，都必须经过至少一条与 t_i关联的边、一条与 t_j关联的边，以及一条连接 V'_i和 V'_j的边，因此 $d_H(t_i,t_j)\geq 3$。另一方面，如果 $d_H(t_i,t_j)\geq 4$，那么 H中就没有连接顶点集 V'_i和 V'_j的边。根据 G'的构造，这意味着在 G中， V_i 和 V_j 之间没有边，因此 G 是 MCC 的否实例。由于在 G' 中，每个终端点 t_i 都与其对应集合 V'_i中的每个顶点相邻，因此，如果 V'_i和 V'_j之间存在边，就会在 G'中产生一条连接 t_i和t_j 、长度为 3 的路径；而且，由于 H 是子集距离保持器，在 H中也会存在这样的路径。因此，我们可以将讨论范围限定为：对于所有满足 $i\neq j$的 i、$j\in[k]$， $d_H(t_i,t_j)=3$。

为了实现距离 3， H必须在每一对 V_i和 V_j之间至少包含一条边，因此至少有 $\binom{k}{2}$ 条边。由于 $|E(H)|=k+\binom{k}{2}$，每个终端点 t_i 都应恰好与 H中的一条边关联。令 c'_i 为 t_i 在 V_i中的唯一邻居，并令 c_i 为V_i中对应的顶点。定义 $C=\{c_i:i\in[k]\}$。由于 H 保持距离，对于每个 $i\neq j$， H中都存在一条边 $c'_ic'_j$，这意味着 G中存在一条边 c_ic_j。因此， C 诱导出的团恰好在每个 V_i中包含一个顶点，所以 G 包含一个有 k 个顶点的多彩团。◀

由于 PDP 是 SDP 的推广，我们得到如下结果。

▶**推论 43.** PDP 参数化于解的大小时是 $W[1]$-困难的。

7 结论

在本研究中，我们根据终端数量和输入图的结构性质，探讨了寻找最小距离保持器的参数化复杂性。除了补充关于距离保持器上下界的现有文献外，我们的结果还揭示了与已知不相交路径问题之间有趣的对应关系。

首先，在指定终端对之间寻找 k 条最短路径，使所得交点最大化——即以 $|\mathcal{P}|$ 为参数的 PDP 问题——与 k-不相交最短路径问题相呼应；该问题同样是 W[1]-困难的，并且在终端对数量作为参数时属于 XP。后一个问题的算法复杂得多， [33], 而且曾长期是一个开放问题。另一方面，我们关于 PDP 乃至 SDP 的困难性结果在非常受限的条件下仍然成立，展现了网格上距离保持器的非平凡结构，并与已知的保距子式瓶颈相吻合 [32]。

其次，在顶点覆盖数为 3 的图上，PDP 的 NP 困难性可以与同一图类上已知的边不相交路径问题的 NP 困难性相比较 [24], 在这两种情况下，终端对的任意结构都使得即使是非常受限的图结构也无济于事。然而，PDP 的归约更具挑战性，因为所需的路径必须是最短路径，却不必彼此不相交。

上述讨论也自然引出了我们留待未来工作的开放问题。虽然我们得出结论：以终端数量为参数时，PDP 和 SDP 在网格上都是 FPT，且 PDP 在网格上是 NP 困难的，但我们尚未发现 SDP 有类似的结果。因此，关于网格上最短路径的排列方式，SDP 在这种情况下的复杂度仍是一个开放问题。

另一个开放问题源于 PDP 与 SDP 在具有小型分隔集的图上的差异。虽然我们关于 PDP 的否定结果排除了以树宽等最常见的其他分解参数为参数的 FPT 或 XP 算法，但对于 SDP，我们只能提供一个以顶点覆盖为参数的 FPT 算法。对于支配顶点覆盖的参数（包括树宽，以及介于树宽与顶点覆盖之间的参数，例如反馈顶点集或树深度），SDP 的复杂度仍未解决。借鉴不相交路径问题中的对应关系，我们注意到该领域也尚无此类结果——SDP 对应于边不相交路径问题，并附加一项限制：终端对集合包含终端集合中的所有点对。据我们所知，尚未有人从图的分解结构角度研究这一问题。

最后，尽管我们的工作聚焦于寻找精确的最小距离保持器，但探索 PDP 和 SDP 的固定参数近似也会很有意思。对于距离保持器，多项式时间近似会遭遇严峻的不可近似性障碍； [14, 1], 将参数的指数时间复杂度和／或受限图结构与近似算法结合起来，或许会有所帮助。尤其是在目标是同时近似子图的规模和最短路径的长度时，这对应于一般伸展子图的固定参数近似。

参考文献

1 Mojtaba Abdolmaleki、Yafeng Yin 和 Neda Masoud。最小权重成对距离保持子图。*arXiv*预印本 *arXiv:2007.07554*，2020年。2 Reyan Ahmed、Greg Bodwin、Faryad Darabi Sahneh、Keaton Hamm、Mohammad Javad Latifi Jebelli、Stephen Kobourov 和 Richard Spence。图伸展子图：教程综述。《计算机科学评论》，37:100253，2020年。网址：https://www.sciencedirect.com/science/article/pii/S1574013719302539, doi:10.1016/j.cosrev.2020.100253。3 Baruch Awerbuch、Bonnie Berger、Lenore Cowen 和 David Peleg。稀疏邻域覆盖的近线性时间构造。《*SIAM*计算杂志》，28(1):263–277，1998年。arXiv:https://doi.org/10.1137/S0097539794271898， doi:10.1137/S0097539794271898。4 Matthias Bentert、Fedor V. Fomin、Petr A. Golovach、Tuukka Korhonen、William Lochet、Fahad Panolan、M. S. Ramanujan、Saket Saurabh 和 Kirill Simonov。短圈打包。《*ACM*算法汇刊》，22(1):8:1–8:35，2026年。 doi:10.1145/3765285。5 Hans L. Bodlaender。寻找小树宽树分解的线性时间算法。《*SIAM*计算杂志》，25:1305–1317，1996。 doi:10.1137/S0097539793251219。

6 Hans L Bodlaender、Pål Grønås Drange、Markus S Dregi、Fedor V Fomin、Daniel Lokshtanov 和 Michał Pilipczuk。一个用于树宽的 $c^k n$ 5-近似算法。《SIAM计算杂志》，45:317–378，2016。7 Greg Bodwin。线性规模距离保持器的新结果。《SIAM计算杂志》，50(2):662–673，2021年。8 Greg Bodwin 和 Virginia Vassilevska Williams。更优的距离保持器和加性伸展子图。载于《第二十七届ACM-SIAM离散算法年度研讨会论文集》，第855–872页。SIAM，2016。9 Béla Bollobás、Don Coppersmith 和 Michael Elkin。稀疏距离保持器和加性伸展子图。载于《第十四届ACM-SIAM离散算法年度研讨会论文集》，SODA '03，第414–423页，美国，2003年。工业与应用数学学会。10 Ulrik Brandes 和 Dagmar Handke。最小平面伸展子图问题的NP完全性结果。载于Rolf H. Möhring编，《计算机科学中的图论概念》，第23届国际研讨会 WG '97，德国柏林，1997年6月18日至20日，论文集，《计算机科学讲义》第1335卷，第85–99页。施普林格，1997年。网址：https://doi.org/10.1007/BFb0024490, doi:10.1007/BFB0024490。11 Leizhen Cai。最小伸展子图问题的NP完全性。《离散应用数学》，48(2):187–194，1994年。网址：https://www.sciencedirect.com/science/article/pii/0166218X94900736, doi:10.1016/0166-218X(94)90073-6。12 Chandra Chekuri 和 Julia Chuzhoy。网格小定理的多项式界。《ACM期刊》，63(5)，2016年12月。doi:10.1145/2820609。13 Jianer Chen、Iyad A. Kanj 和 Ge Xia。顶点覆盖问题的改进参数化上界。载于《2006年计算机科学数学基础》，第238–249页。施普林格柏林海德堡，2006年。14 Eden Chlamtáč、Michael Dinitz、Guy Kortsarz 和 Bundit Laekhanukit。近似伸展子图和有向斯坦纳森林：上下界。《ACM算法汇刊》，16(3)，2020年6月。doi:10.1145/3381451。15 Edith Cohen。无向最短路径的多对数时间、近线性工作量近似方案。《ACM期刊》，47(1):132–166，2000年1月。doi:10.1145/331605.331610。16 Don Coppersmith 和 Michael Elkin。稀疏源点和成对距离保持子图。《SIAM离散数学杂志》，20(2):463–501，2006年。doi:10.1137/050630696。17 Marek Cygan、Fedor V. Fomin、Lukasz Kowalik、Daniel Lokshtanov、Dániel Marx、Marcin Pilipczuk、Michal Pilipczuk 和 Saket Saurabh。《参数化算法》。施普林格，2015年。doi:10.1007/978-3-319-21275-3。18 E. Dahlhaus、D. S. Johnson、C. H. Papadimitriou、P. D. Seymour 和 M. Yannakakis。多终端割的复杂性。《SIAM计算杂志》，23(4):864–894，1994年。arXiv:https://doi.org/10.1137/S0097539792225297, doi:10.1137/S0097539792225297。19 David P. Dobkin、Steven J. Friedman 和 Kenneth J. Supowit。德劳内图几乎与完全图一样好。《离散与计算几何》，5(4):399–407，1990年8月。doi:10.1007/BF02187801。20 Michael Elkin。计算近似最短路径。《ACM算法汇刊》，1(2):283–323，2005年10月。doi:10.1145/1103963.1103968。21 Michael Elkin、Arnold Filtser 和 Ofer Neiman。终端嵌入。《理论计算机科学》，697:1–36，2017年。网址：https://doi.org/10.1016/j.tcs.2017.06.021, doi:10.1016/J.TCS.2017.06.021。22 Michael Elkin 和 Seth Pettie。一般图的线性规模对数伸展路径报告距离预言机。《ACM算法汇刊》，12(4)，2016年8月。doi:10.1145/2888397。23 Michael Elkin 和 Idan Shabat。具有对数伸展和 o(n log log n)规模的路径报告距离预言机。载于《第64届IEEE计算机科学基础年度研讨会》，FOCS 2023，美国加利福尼亚州圣克鲁斯，2023年11月6日至9日，第2278–2311页。IEEE，2023年。doi:10.1109/FOCS57990.2023.00141。

24 Krzysztof Fleszar、Matthias Mnich 和 Joachim Spoerhase。基于树相似性的最大不相交路径新算法。《数学规划》，171(1):433–461，2018年9月。 doi:10.1007/s10107-017-1199-3。25 Fedor V. Fomin、Petr A. Golovach、William Lochet、Pranabendu Misra、Saket Saurabh 和 Roohani Sharma。有向跨距子图问题的参数化复杂性。《算法学》，84(8):2292–2308，2022年。URL: https://doi.org/10.1007/s00453-021-00911-x, doi:10.1007/S00453-021-00911-X。26 Kshitij Gajjar 和 Jaikumar Radhakrishnan。区间图的距离保持子图。载于第25届欧洲算法研讨会（ESA 2017），《莱布尼茨国际信息学会议录》（LIPIcs）第87卷，第39:1–39:13页，德国达格施图尔，2017年。Schloss Dagstuhl – Leibniz-Zentrum für Informatik。URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ESA.2017.39, doi:10.4230/LIPIcs.ESA.2017.39。27 M. Hanan。关于具有直角距离的斯坦纳问题。《SIAM应用数学杂志》，14(2):255–265，1966年。 arXiv:https://doi.org/10.1137/0114025, doi:10.1137/0114025。28 Ton Kloks。《树宽、计算与近似》，《计算机科学讲义》第842卷。施普林格，1994年。 doi:10.1007/BFB0045375。29 Yusuke Kobayashi。最小跨距子图问题的NP-困难性与固定参数可解性。《理论计算机科学》，746:88–97，2018年。URL: https://www.sciencedirect.com/science/article/pii/S030439751830447X, doi:10.1016/j.tcs.2018.06.031。30 Shimon Kogan 和 Merav Parter。对跳数抱有希望：新型伸展子图、保持子图以及跳数集的下界。载于第63届IEEE计算机科学基础年会（FOCS 2022），美国科罗拉多州丹佛，2022年10月31日至11月3日，第766–777页。IEEE，2022年。 doi:10.1109/FOCS54457.2022.00078。31 Shimon Kogan 和 Merav Parter。对缺失的伸展子图抱有希望：新型距离保持器与轻量跳数集。载于2025年ACM-SIAM离散算法研讨会年会论文集（SODA），第4352–4374页。SIAM，2025年。32 Robert Krauthgamer、Huy L. Nguy˜ên 和 Tamar Zondiner。使用图 minor 保持终端距离。SIAM离散数学杂志，28(1):127–141，2014年。 doi:10.1137/120888843。33 William Lochet。k-条不相交最短路径问题的多项式时间算法。载于Dániel Marx（编），2021年ACM-SIAM离散算法研讨会论文集（SODA 2021），线上会议，2021年1月10日至13日，第169–178页。SIAM，2021年。 doi:10.1137/1.9781611976465.12。34 L. Paul Chew。有些平面图几乎与完全图一样好。《计算机与系统科学杂志》，39(2):205–219，1989年。URL: https://www.sciencedirect.com/science/article/pii/0022000089900445, doi:10.1016/0022-0000(89)90044-5。35 David Peleg 和 Alejandro A. Schäffer。图伸展子图。《图论杂志》，13(1):99–116，1989年。URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.3190130114, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/jgt.3190130114, doi:10.1002/jgt.3190130114。36 Seth Pettie。低畸变跨距子图。《ACM算法汇刊》，6(1)，2010年12月。 doi:10.1145/1644015.1644022。37 Weiping Shi 和 Chen Su。直线型斯坦纳树形图问题是NP完全的。《SIAM计算杂志》，35(3):729–740，2005年。 doi:10.1137/S0097539704371353。38 G. Venkatesan、U. Rotics、M.S. Madanlal、J.A. Makowsky 和 C.Pandu Rangan。最小跨距子图问题的限制情形。《信息与计算》，136(2):143–164，1997年。URL: https://www.sciencedirect.com/science/article/pii/S0890540197926419, doi:10.1006/inco.1997.2641。