arXiv:2601.11419v1 [cs.DM] 16 Jan 2026

关于虚拟网络嵌入多面体

Amal Benhamiche[1],Pierre Fouilhoux[2],Lucas Létocart[2],Nancy Perrot[1],和 Alexis Schneider[1,2]

[1] Orange Research，法国沙蒂永 F-92320 {amal.benhamiche, nancy.perrot, alexis.schneider}@orange.com[2] 索邦巴黎北大学、法国国家科学研究中心、巴黎北计算机科学实验室、LIPN，法国维勒塔讷兹 F-93430 {letocart, fouilhoux, schneider}@lipn.univ-paris13.fr

摘要。 我们首次对现代电信网络中出现的虚拟网络嵌入（VNE）问题开展多面体研究。我们针对所谓的流公式提出了新的有效不等式。随后，我们通过专门的流分解算法证明，在将一条虚拟边嵌入到底层路径的情况下，这些不等式刻画了 VNE 多面体。初步实验表明，这些新不等式有望显著加速混合整数规划求解器。

关键词： 虚拟网络嵌入 · 多面体研究 · 数学规划

1 引言

网络虚拟化是一种新兴范式，使电信运营商能够在多个虚拟网络之间共享其基础设施 [4, 6]。它是 5G网络中网络切片的核心，每个虚拟网络都对应一项服务，并根据其需求量身定制，例如速度、安全性或覆盖范围 [2]。网络切片对于部署具有特定特征的全新且前景广阔的服务至关重要，例如远程手术或自动驾驶车辆。

网络虚拟化的核心组合结构称为虚拟网络嵌入（VNE）问题。VNE问题可定义如下。我们考虑简单连通无向图。设虚拟网络为 $\mathcal{G}_r = (V_r, E_r)$，包含 n_r 个节点；底层网络为$\mathcal{G}_s = (V_s, E_s)$，包含 n_s 个节点。将底层网络资源分配给虚拟需求，就是要找到如下映射 （也称为嵌入）：

定义 1. $\mathcal{G}_s$ 上的映射 $m = (m_V, m_E)$ $\mathcal{G}_r$ 是一对满足以下条件的函数：

– m_V： $V_r \to V_s$ 称为节点放置；– m_E： $E_r \to P_s$称为边路由，其中 P_s 是 $\mathcal{G}_s$ 的无环路径集合；对于每个 $e = (u, v) \in E_r$， $m_E(e)$ 是 $\mathcal{G}_s$ 中的一条路径，其端点为 $m_V(u)$ 和 $m_V(v)$。

在本文中，我们考虑一种一对一 节点放置：每个虚拟节点都必须放置在不同的底层节点上。虚拟节点 $u \in V_r$ （虚拟边 $e \in E_r$）具有整数需求 d_u （ d_e）。底层节点 $u \in V_s$（底层边 $e \in E_s$）具有整数容量 c_u （ c_e）。当满足容量约束时，称映射是可行的，也就是说，放置在某个底层节点上的虚拟节点（或通过某条底层边进行路由的虚拟边）的需求总和小于或等于该节点（或边）的容量。

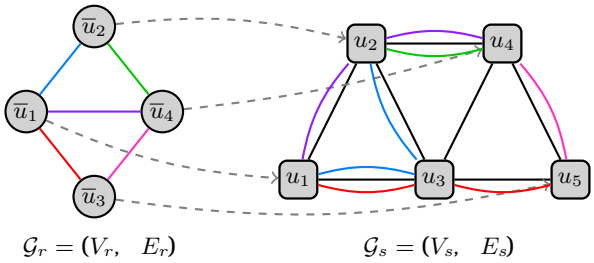


图1. 左侧虚拟网络在右侧底层网络上的映射示例。

图1展示了左侧虚拟图在右侧底层图上的嵌入。虚线箭头表示虚拟节点的放置。给定颜色的虚拟边使用相同颜色的底层路径进行路由。在此示例中， $m_V(u_1) = u_1$, $m_V(u_3) = u_5$ 以及 $m_E(u_1, u_3) = \{(u_1, u_3), (u_3, u_5)\}$。

使用一个底层节点单元（分别为边）会产生使用成本 w_u （分别为 w_e）。映射 m 的成本记为 W_m，对应于放置成本和路由成本之和：
$W_m = \sum_{u \in V_r} d_u w_{m_V(u)} + \sum_{e \in E_r} \sum_{e \in m_E(e)} d_e w_e$。VNE 问题可表述如下：

定义 2。 给定一个虚拟网络 $\mathcal{G}_r = (V_r, E_r)$，其需求为 d；给定一个底层网络 $\mathcal{G}_s = (V_s, E_s)$，其容量为 c，成本为 w；虚拟网络嵌入问题（*VNE*）旨在求出将 $\mathcal{G}_r$ 嵌入到 $\mathcal{G}_s$ 上的最小成本方案。

众所周知，VNE 问题即使在均匀需求条件下 [3, 13], 也是 NP完全 [5]。针对该问题，人们已广泛研究了启发式算法和元启发式算法，参见 [8]中的综述。相比之下，整数规划等精确方法受到的关注较少。文献 [6, 10], 针对有向情形提出了一种紧凑的流公式 [6, 10], ，但已知该公式存在较弱的线性松弛，这一点在 [11]中有所讨论。本文针对这一自然公式提出了新的有效不等式，并证明在一条虚拟边和一条底层路径的情形下，这些不等式能够刻画 VNE多面体。

2 流公式与 VNE 多面体

VNE流公式已针对有向网络进行了描述 [6, 10]。下面，我们将通过为每条底层边引入双向流变量，对其进行调整以适用于无向情形。

2.1 无向流公式

尽管边是无向的，为了将 VNE 表述为一个流问题，我们仍将它们视为有序集合（即为它们指定任意方向）。因此，由于网络是无向的，虚拟边的路径路由可以包含沿任一方向的底层边（u, v）$\in E_s$（u, v）或（v, u）。为此，我们引入底层网络的双向版本 $\mathcal{G}'_s = (V_s, E'_s)$，其中 $E'_s = \bigcup_{(u,\ v)\in E_s}\{(u,\ v), (v,\ u)\}$。这种双向定向技术在流公式中是标准做法 [1, 12]。此外，对于一个底层节点 $u \in V_s$，令 $\delta^+(u)$（分别为 $\delta^-(u)$）表示节点 u 在 E'_s 中的出边（入边）。

考虑两组二元变量。仅当虚拟节点 $u \in V_r$ 放置在底层节点 $u \in V_s$ 上时，放置变量 x_{uu} 才取值 1。仅当虚拟边 $e \in E_r$ 的路径路由使用底层边 $(u,\ v) = e \in E_s$，从 u 到 v（分别从 v 到 u）时，流变量 $y_{e(u,v)}$（分别为 $y_{e(v,u)}$）才取值 1。注意，可以进行简单的预处理：如果 $d_u > c_u$（分别为 $d_{ee} > c_e$），则将变量 x_{uu}（分别为 y_{ee}）设为 0；这对任意 $u \in V_r$, $u \in V_s$（分别为任意 $e \in E_r$, $e \in E_s$）均成立。因此，以下我们假设，对于任意 $e \in E_s$, $d_e \le c_e$ 均成立；并且对于任意 $u \in V_s$，$d_u \le c_u$、$d_v \le c_u$ 均成立。

流公式（FF）如下。

$$\min \quad \sum_{u\in V_s}\sum_{\overline{u}\in V_r} d_{\overline{u}} w_u x_{\overline{u}u} + \sum_{e\in E'_s}\sum_{\overline{e}\in E_r} d_{\overline{e}} w_e y_{\overline{e}e}$$

$$\text{s.t.} \quad \sum_{u\in V_s} x_{\overline{u}u} = 1 \qquad \forall \overline{u} \in V_r \tag{1a}$$

$$x_{\overline{u}u} - x_{\overline{v}u} = \sum_{e\in\delta^+(u)} y_{\overline{e}e} - \sum_{e\in\delta^-(u)} y_{\overline{e}e} \qquad \forall (\overline{u},\overline{v}) = \overline{e} \in E_r,\ \forall u \in V_s \tag{1b}$$

$$\sum_{\overline{u}\in V_r} x_{\overline{u}u} \le 1 \qquad \forall u \in V_s \tag{1c}$$

$$\sum_{\overline{e}\in E_r} d_{\overline{e}}\left(y_{\overline{e}(u,v)} + y_{\overline{e}(v,u)}\right) \le c_e \qquad \forall (u,v) \in E_s \tag{1d}$$

$$x_{\overline{u}u} \in \{0,1\} \qquad \forall \overline{u} \in V_r, \forall u \in V_s,$$

$$y_{\overline{e}e} \in \{0,1\} \qquad \forall \overline{e} \in E_r, \forall e \in E'_s$$

约束 (1a) 确保虚拟节点放置有效。约束 (1b) 是流守恒约束，保证每条虚拟边都关联到底层图中的一条路由路径，且路径的端节点承载着该虚拟边的端点。约束 (1c) 确保一对一节点放置。约束 (1d) 确保节点和边容量得到满足。最后，目标函数表示嵌入成本，同时考虑节点放置成本和路由成本。

*VNE*多面体 对于每个映射 $m \in \mathcal{M}$，其中 $\mathcal{M}$ 是 $\mathcal{G}_r$ 在 $\mathcal{G}_s$上的可行映射集合，m 的关联向量（变量为 x 和 y）记作 $\chi(m)$。我们关注 VNE多面体 $\mathcal{F}$的描述，即 $\mathcal{G}_r$ 在 $\mathcal{G}_s$上的有效映射的凸包：

$$\mathcal{F}(\mathcal{G}_r, \mathcal{G}_s, d, c) = conv\{\chi(m), m \in \mathcal{M}\} \tag{2}$$

2.2 有效不等式与等式

已有研究 [11] 观察到，流公式的线性松弛效果较差。下文将提出 VNE 问题的首个有效不等式，以强化该公式。

以下流出发不等式表明，当虚拟节点u 被放置在底层节点 $u \in V_s$ 上时，任何虚拟边 $(u, v) = e \in E_r$都必须通过底层网络双向版本中 u的一条出弧进行路由。

命题 1。对于任意 $e \in E_r$和任意 $u \in V_s$，流出发不等式均有效：

$$x_{\bar{u}u} \leq \sum_{e \in \delta^+(u)} y_{\bar{e}e} \tag{3}$$

以下流连续性不等式表明，当虚拟边 $(u, v) = e \in E_r$ 通过底层边 $(u, v) = \bar{e} \in E'_s$进行路由时，$v$ 要么被放置在 v 上，要么路径路由 e 包含 v的另一条出弧，且该出弧不是 (v, u)。

命题 2. 对于任意 $e = (u, v) \in E_r$，以及任意 $e = (u, v) \in E'_s$，流连续性不等式成立：

$$y_{\bar{e}e} \leq \sum_{e' \in \delta^+(v) \setminus (v,u)} y_{\bar{e}e'} + x_{\bar{v}v} \tag{4}$$

最后，设 L_s 为 $\mathcal{G}_s$ 的叶节点，即只有一个邻居的节点。对于叶节点 $l \in L_s$，设 v_l 为其在 V_s 中唯一的邻居。有趣的是，在这样的叶节点上，将流出发不等式和连续性不等式与流守恒约束结合，可得如下叶等式：

命题 3. 对于任意虚拟边 $(u, v) = e \in E_r$ 以及任意叶节点 $l \in L_s$，叶等式成立：

$$y_{\bar{e}(l,v_l)} = x_{\bar{u}l} \tag{5}$$

2.3 单条虚拟边的情形

在本文余下部分，我们考虑单条虚拟边 $e = (u, v)$ 的嵌入，这是研究 VNE多面体的第一个自然步骤。该情形可以通过 [11] 中描述的改进的最短路径算法在多项式时间内求解。

有趣的是，在这种情况下， x 变量扮演着弧流变量的角色：对于 $u \in V_s$，变量 $x_{u\underline{u}}$ （分别为 $x_{v\underline{v}}$）对应于 e在弧 $(u,\ u)$（分别为 $(u,\ v)$）上的流变量。
于是，节点放置不等式就转化为关于虚拟节点的流不等式约束。

这意味着 $(x,\ y)$ 是扩展网络中的 $u,\ v$-弧流。该扩展网络是底层网络 $\mathcal{G}'_s$ 的双向版本，外加其他弧：对于任意 $u \in V_s$，从 u 到 u 以及从 u 到 v。
我们将该扩展网络记为 $\mathcal{G}^a_s = (V_s \cup V_r,\ E^a_s)$。图 2 展示了一个简单 VNE实例的扩展网络（右）及其对应网络（左）。

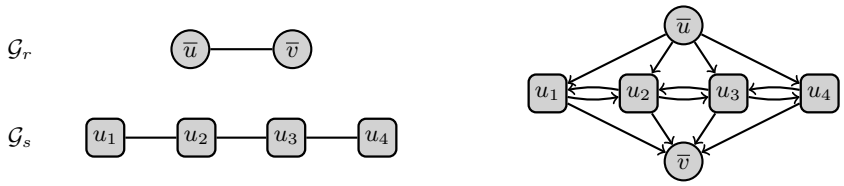


图 2。 扩展网络示例

与经典流问题相比，VNE具有额外的一对一节点放置容量，这破坏了约束矩阵的全幺模结构（见下文示例）。不过，我们仍然可以使用流分解定理 [1], ，该定理表明，任何弧流都可以轻松分解为路径流。由于分数变量 $(\tilde{x},\ \tilde{y})$ 对应于 $u-v$弧流，因此对于具有单条虚拟边的 VNE，可得到以下结果：

命题4。 任何分数解$(\tilde{x}, \tilde{y})$ 都可以分解为$u-v$ 条路径 P和 C 个扩展网络 $\mathcal{G}^a_s$中的回路：

$$(\tilde{x}, \tilde{y}) = \sum_{p \in P} \lambda_p \chi(p) + \sum_{c \in C} \lambda_c \chi(c)$$

使得$\sum_{p \in P} \lambda_p = 1$。

扩展网络$\mathcal{G}^a_s$ 中的有效路径是至少使用 E^a_s 的三条边（或原始底层网络中的一条边）的路径。因此，有效路径是一个整数解（可行映射）。这使我们能够为虚拟网络仅包含一条边时的 VNE 多面体构造刻画工具。

命题 5. 考虑一组有效不等式 $Ax \leq b$。如果对于由这些不等式强化的流公式的任意分数解，都存在一个在扩展网络上的 $u-v$路径与回路分解，使得所有路径均有效且不存在回路，那么带有这些不等式的 *(FF)* 就刻画了 *VNE*多面体，即：

$$\mathcal{F}(\mathcal{G}_r, \mathcal{G}_s, d, c) = \{x, y \in [0,1], s.t.\ 1a - 1d, Ax \leq b\}$$

证明。 在这种情况下，任意分数解（$\tilde{x}$, $\tilde{y}$）都可以表示为可行整数解的凸组合，这意味着 $\{x, y \in [0,1], s.t.\ 1a - 1d, Ax \geq b\}$ 的所有极点都是整数的。□

图3展示了一个简单路径上的边实例中，分数点（左）的流分解示例（右）。分数点 $(\tilde{x}, \tilde{y})$ 如下：$\tilde{x}_{uu_1} = \tilde{x}_{uu_4} = \tilde{x}_{vu_1} = \tilde{x}_{vu_4} = 0.5$, $\tilde{x}_{uu_2} = \tilde{x}_{uu_3} = \tilde{x}_{vu_2} = \tilde{x}_{vu_3} = 0$, $\tilde{y}_{e(u_1,u_2)} = \tilde{y}_{e(u_2,u_1)} = \tilde{y}_{e(u_3,u_4)} = \tilde{y}_{e(u_4,u_3)} = 0.5$, $\tilde{y}_{e(u_2,u_3)} = \tilde{y}_{e(u_3,u_2)} = 0$。

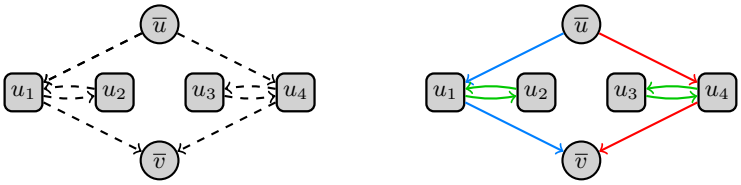


图3. 流分解示例

唯一的路径与环分解如下。它包含两条路径：$p_1 = (u, u_1, v)$（图中蓝色）和 $p_2 = (u, u_4, v)$（图中红色），其中 $\lambda_{p_1} = \lambda_{p_2} = 0.5$；以及两个2节点环（图中绿色）。

该分数点满足流出发不等式。由于它包含无效路径，这个例子表明，对于底层路径上的虚拟边，加入流出发不等式的 (FF) 并非整数。请注意，在此处加入流连续性不等式可以消除所有此类分数解。在下一节中，我们将证明这一点确实始终成立。

3 底层路径的多面体刻画

现在我们证明，当底层网络是一条路径 $\mathcal{P}_s$ 时，加入流出发不等式和流连续性不等式的流公式刻画了 VNE 多面体。

考虑一个分数解 $(\tilde{x}, \tilde{y})$。为了证明该结论，我们必须构造 $(\tilde{x}, \tilde{y})$ 的流分解，且其中只包含有效路径。我们发现，常用的流分解算法（见 [1] 中的描述）无法做到这一点。因此，我们提出了自己的流分解算法，称为 ComputeFlowDecomposition。给定该分数解，算法 1 返回一个路径-环分解 (P, C, λ)。

该算法通过逐步构建有效路径集合来实现这一点：首先在第 3 行的 for循环中构建正向路径（从节点 $u_i \in V_s$ 出发并在 u_j 结束，且满足 $i < j$）；然后在第 16 行的 for循环中构建反向路径。该算法依赖于残余分数解$(\tilde{x}', \tilde{y}')$，在过程的任意步骤中，它都对应于 $(\tilde{x}, \tilde{y}) - \sum_{p \in P} \lambda_p \chi(p)$。

一个重要性质是，残余分数解始终满足流出和连续性不等式，从而保证

算法 1 ComputeFlowDecomposition

要求： 一个分数解 $(\tilde{x},\ \tilde{y})$

确保： 一个流分解 $(P,\ C,\ \lambda)$，其中 P 的所有路径均有效且 $C=\emptyset$

1：初始化空数组 $P,\ C,\ \lambda$

2：初始化数组 $(\tilde{x}',\ \tilde{y}') \leftarrow (\tilde{x},\ \tilde{y})$

3: **对于** $u_i \in (u_1,..,u_{n_s-1})$ **执行** ▷ 前向映射

4: $j \leftarrow i+1$

5: **当** $\tilde{y}'_{e(u_i,u_{i+1})} > 0$ **do**

6: $\delta \leftarrow$ 最小值$(\tilde{y}'_{e(u_i,u_{i+1})},\ \tilde{x}'_{\overline{v}u_j} -$ 最大值$(\tilde{y}'_{e(u_{j+1},u_j)} - \tilde{y}'_{e(u_j,\ u_{j-1})},\ 0))$

7: **如果** $\delta > 0$ **则**

8: $p \leftarrow (\overline{u}, u_i, \ldots, u_j, \overline{v})$

9: $P \leftarrow P \cup p$

10: $\lambda_p \leftarrow \delta$

11: $(\tilde{x}', \tilde{y}') \leftarrow (\tilde{x}', \tilde{y}') - \delta \cdot \chi(p)$

12: **结束如果**

13: $j \leftarrow j+1$

14: **结束循环**

15: **结束循环**

16: **对于** $u_i \in (u_{n_s}、\ldots\ldots、u_2)$ **执行** ▷ 反向映射

17: $j \leftarrow i-1$

18: **当** $\tilde{x}'$ **时**$_{\overline{u}u_i} > 0$ **执行**

19: $\delta \leftarrow \tilde{x}'_{\overline{v}u_j}$

20: **如果** $\delta > 0$ **则**

21: $p \leftarrow (\overline{u}, u_i, \ldots, u_j, \overline{v})$

22: $P \leftarrow P \cup p$

23: $\lambda_p \leftarrow \delta$

24: $(\tilde{x}',\ \tilde{y}') \leftarrow (\tilde{x}',\ \tilde{y}') - \delta \cdot \chi(p)$

25: **结束 if**

26: $j \leftarrow j-1$

27: **结束 while**

28: **结束 for**

29: **返回** $(P,\ C,\ \lambda)$

算法始终能够继续推进。在正向路径阶段，该性质通过以下方式得以维持。在节点 $u_l \in V_s$处，必须保留足够的残余变量$\tilde{x}'_{vu_l}$，以便稍后构造反向映射。具体而言，所需数量为 $\tilde{y}'_{e(u_{l+1},u_l)} - \tilde{y}'_{e(u_l,u_{l-1})}$（若大于 0）：第6行会强制满足这一要求。

现在给出一个例子。图 4 展示了算法 1 对于一条 4 节点路径和一个分数解 $(\tilde{x},\ \tilde{y})$ 所执行的各个步骤，该分数解如下：$\tilde{x}_{uu_1}=0.4,\quad \tilde{x}_{uu_2}=0.3,$ $\tilde{x}_{uu_3}=0.3,\ \tilde{x}_{uu_4}=0.,\ \tilde{x}_{vu_1}=0.,\quad \tilde{x}_{vu_2}=0.3,\quad \tilde{x}_{vu_3}=0.4,\quad \tilde{x}_{vu_4}=0.3,$ $\tilde{y}_{e(u_1,u_2)}=0.4,\quad \tilde{y}_{e(u_2,u_1)}=0.,\ \tilde{y}_{e(u_2,u_3)}=0.7,\ \tilde{y}_{e(u_3,u_2)}=0.3,\tilde{y}_{e(u_3,u_4)}=0.3,$ $\tilde{y}_{e(u_4,u_3)}=0.$。

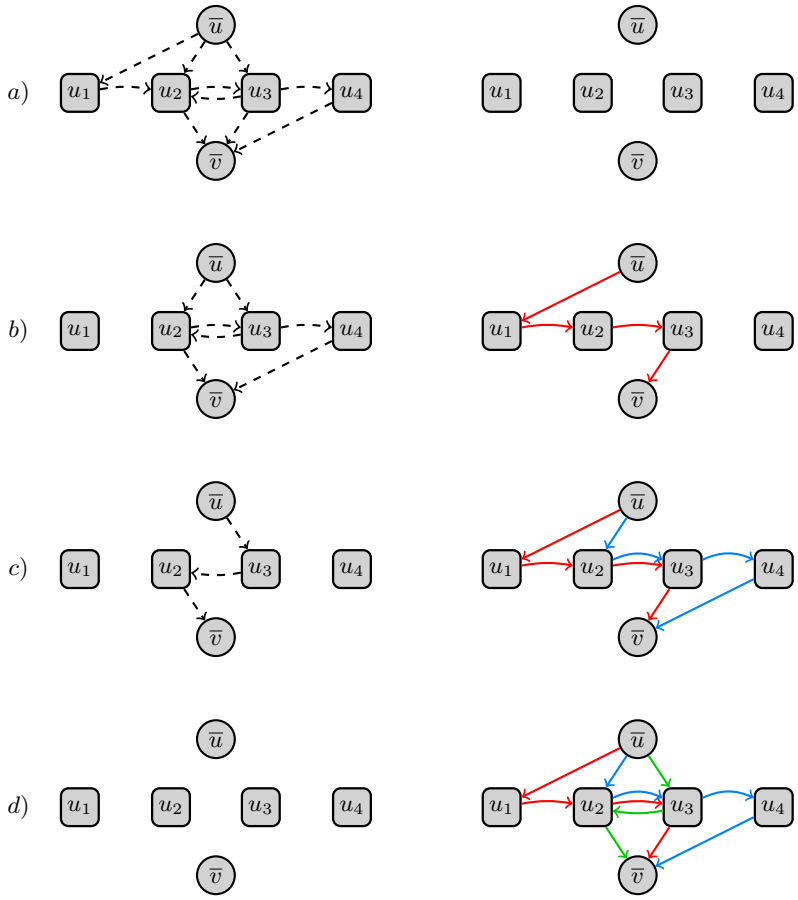


图4. 算法在4节点底层路径上的步骤。左侧为剩余分数变量；右侧为算法构造的有效路径 P，a）算法执行前，b）在 u_1 上完成第一阶段后，c）在 u_2 上完成第一阶段后，d）完成第二阶段后（算法完成）。

每一步中，非零的剩余分数变量显示在图的左侧，算法构造的有效路径显示在右侧。最后构造出三条路径： $p_1 = (u, u_1, u_2, u_3, v)$ （图中为红色），$p_2 = (u, u_2, u_3, u_4, v)$ （蓝色）和 $p_3 = (u, u_3, u_2, v)$ （绿色），此时所有剩余分数变量的值均为零。注意，由于 $\tilde{y}_{e(u_3,u_2)} = 0.3$ 和 $\tilde{y}_{e(u_2,u_1)} = 0.$，因此不会构造路径 (u, u_1, u_2, v)。这些路径连同 $\lambda_{p_1} = 0.4$、$\lambda_{p_2} = 0.3$ 和 $\lambda_{p_3} = 0.3$，构成了该分数点的路径分解，其中所有路径均有效且不含回路。

接下来，我们证明该算法有效。首先请注意，残余变量 $\tilde{x}'$ 始终为正（直接）。并且，在算法的任何时刻

,

$$\sum_{u\in V_s}\tilde{x}'_{\bar{u}u}=\sum_{u\in V_s}\tilde{x}'_{\bar{v}u}=1-\sum_{p\in P}\lambda_p \tag{6}$$

现在我们考察构造出的第一条路径，并观察一些有用的性质。随后，我们考虑算法的后续步骤，对于这些步骤，上述观察可以轻松推广。这使我们能够展示该算法的有效性和终止性。

构造出的第一条路径： 首先，我们考虑算法在第一个节点 u_1 上的迭代。我们假设 $\tilde{x}_{uu_1}>0$ （在另一种情况下，我们的论证可以轻松调整），并考虑算法构造的第一条路径，记为 $p=(u,\ u_1,\ .....,\ u_j,\ v)$，其中 $u_j\in\{u_2,\dots,u_{n_s-1}\}$。流连续性不等式确保了这样一条路径的存在。我们针对这次第一次迭代给出一系列有用的断言。

断言 *1*。

$$\tilde{y}_{\bar{e}(u_1,u_2)}\le\tilde{y}_{\bar{e}(u_l,u_{l+1})},\ \text{for } l\in\{2,\dots,j-1\} \tag{7}$$

证明。 我们证明 $\tilde{y}_{e(u_1,u_2)}\le\tilde{y}_{e(u_2,u_3)}$ （在没有构造出以 u_2 结束的路径的情况下）。由于没有在该节点上构造路径，根据算法的第6行，存在两种情况：

– $\tilde{x}_{\bar{v}u_2}=0$：结果由（u_1，u_2）上的流连续性不等式得出。– $\tilde{x}_{vu_2}\le\tilde{y}_{e(u_3u_2)}-\tilde{y}_{e(u_2u_1)}$：将其代入节点 u_2 上的流守恒等式，可得 $\tilde{y}_{e(u_2,u_3)}\ge\tilde{y}_{e(u_1,u_2)}+\tilde{x}_{uu_2}\ge\tilde{y}_{e(u_1,u_2)}$。

对于以下底层节点，证明是类似的。 □

断言 *2*。

$$\tilde{y}_{\bar{e}(u_1,u_2)}-\lambda_p\le\tilde{y}_{\bar{e}(u_j,u_{j+1})} \tag{8}$$

证明。 根据第6行，有三种可能的情况：

– $\lambda_p=\tilde{y}_{e(u_1u_2)}$：直接得出。– $\lambda_p=\tilde{x}_{vu_j}$：结果由（$u_{j-1}$，$u_j$）上的流连续性不等式以及公式 (7) 得出。– $\lambda_p=\tilde{x}_{vu_j}-(\tilde{y}_{e(u_{j+1}u_j)}-\tilde{y}_{e(u_ju_{j-1})})$：将其代入节点 u_j 上的流守恒等式，可得 $\tilde{y}_{e(u_j,u_{j+1})}=\tilde{y}_{e(u_{j-1},u_j)}-\lambda_p+\tilde{x}'_{uu_j}$。使用公式 (7) 即可得到该结果。□

前面两个断言意味着，代入 $\lambda_p\chi(p)$ 后，残余变量 $\tilde{y}'$ 仍大于 0。接下来，我们证明附加的不等式对残余变量也成立。

断言 *3.* 对于 $(\tilde{x},\tilde{y})-\lambda_p\chi(p)$，流连续性不等式仍然成立。

证明. 我们逐一证明这些不等式：

– 在 (u_l,u_{l+1}) 上，对于 $u_l\in\{u_1,\dots,u_{j-1}\}$，$\lambda_p$ 同时从不等式的左侧（LHS）和右侧（RHS）中移除。

– 在 (u_{j-1}, u_j) 上： λ_p 仅从左侧（LHS）中移除。–
在 (u_{j+1}, u_j) 上：根据第6行，

$$\begin{aligned} \lambda_p &\leq \tilde{x}_{\bar{v}u_j} - (\tilde{y}_{\bar{e}(u_{j+1},u_j)} - \tilde{y}_{\bar{e}(u_j,u_{j-1})}) \\ \Leftrightarrow\ \tilde{y}_{\bar{e}(u_{j+1},u_j)} &\leq \tilde{x}_{\bar{v}u_j} - \lambda_p + \tilde{y}_{\bar{e}(u_j,u_{j-1})} \end{aligned}$$

– 在其他边上，该不等式不受影响。 □

断言*4.* 对于 $(\tilde{x},\ \tilde{y}) - \lambda_p \chi(p)$，流出发不等式仍然成立。

证明. 我们对每个不等式分别进行说明：

– 在 u_1 上：从不等式的左侧和右侧都移除了 λ_p 。–在 $u_l \in u_2, \ldots, u_{j-1}$ 上：由于没有构造出以 u_l 结束的路径，因此根据算法的第6行，以下情况之一成立：

- $\tilde{x}_{\bar{v}u_l} = 0$： u_l 上的流守恒等式变为

$$\tilde{x}_{\bar{u}u_l} + \tilde{y}_{\bar{e}(u_{l-1}u_l)} + \tilde{y}_{\bar{e}(u_{l+1}u_l)} = \tilde{y}_{\bar{e}(u_l u_{l+1})} + \tilde{y}_{\bar{e}(u_l u_{l-1})}$$

  根据公式 (7)，$\lambda_p \leq \tilde{y}_{e(u_\Gamma,u_2)} \leq \tilde{y}_{e(u_{l-1},u_l)}$。我们得到：

$$\tilde{x}_{\bar{u}u_l} \leq \tilde{x}_{\bar{u}u_l} + \tilde{y}_{\bar{e}(u_{l+1}u_l)} \leq \tilde{y}_{\bar{e}(u_l u_{l+1})} + \tilde{y}_{\bar{e}(u_l u_{l-1})} - \lambda_p$$

- $\tilde{x}_{\bar{v}u_l} \leq \tilde{y}_{\bar{e}(u_{l+1}u_l)} - \tilde{y}_{\bar{e}(u_l,u_{l-1})}$：将其代入关于 u_l 的流量守恒约束中，

$$\tilde{x}_{\bar{u}u_l} + \tilde{y}_{(u_{l-1}u_l)} \leq \tilde{y}_{(u_l u_{l+1})}$$

  再次使用 $\lambda_p \leq \tilde{y}_{e(u_{l-1},u_l)}$，即可得到结果。

– 对于其他节点，该不等式不受影响。 □

后续对 u_1 的迭代： 此后，当 $\tilde{y}'_{e(u_1,u_2)} > 0$ 时，算法会继续构造以 (u, u_1) 开始的路径。之前的所有结论都可轻松推广到这些新路径。因此，由于流连续性不等式对剩余分数变量仍然成立，这样的路径总是可以构造出来的。剩余的$\tilde{y}'$ 变量将始终保持为正。最终，对节点 u_1 的迭代会完成，最迟在节点u_{n_s}处完成，因为 $\tilde{y}_{e(u_1,u_2)} \leq \tilde{y}_{e(u_{n_s-1},u_{n_s})} = \tilde{x}_{vu_{n_s}}$（根据公式 7）。此后，$\tilde{y}'_{e(u_1,u_2)} = \tilde{x}'_{uu_1} = 0$。

正向路径阶段的后续节点： 如果存在，考虑下一个节点 u_i（属于 $u_2, \ldots, u_{n_s}$，使得 $\tilde{y}'_{e(u_i,u_{i+1})} > 0$。由于此时流连续性不等式对剩余分数变量仍然成立，因此，之前所有的论断和观察都适用于在该节点 u_i 上进行的迭代。于是，算法构造以 (u, u_i) 开始的正向路径，直到 $\tilde{y}'_{e(u_2,u_3)} = 0$。注意，此时$\tilde{x}'_{uu_i}$ 可能非空。算法继续处理后续节点，直至正向路径阶段的迭代结束，此时，对于所有 $u_l \in \{u_1, \ldots, u_{n_s-1}\}$，$\tilde{y}'_{e(u_l,u_{l+1})} = 0$。

反向路径阶段： 随后，算法遍历 $u_{n_s}, \ldots, u_2$并构造反向路径。考虑这些节点中第一个满足以下条件的节点 u_i

：在该时刻， $\tilde{x}'_{uu_i} > 0$。由于此时残余分数解满足流量流出不等式和连续性不等式（引理 3-4），因此也可以继续构造路径，直到变量 $\tilde{x}'_{uu_i}$ 和 $\tilde{y}'_{e(u_i,u_{i-1})}$ 的值为零。

算法结束时，所有残余变量的值都将为零。随后构造的有效路径集合 P 是解 $(\tilde{x}, \tilde{y})$ 的流分解，其中所有路径均有效且不存在回路，且 $\sum_{p\in P} \lambda_p = 1$ （根据方程（6））。利用命题 5，可证明其整数性：

定理 1。 对于底层路径 $\mathcal{P}_s$ 上的虚拟边 e ，包含流离开不等式和连续性不等式的流公式刻画了 *VNE* 多面体：

$$\mathcal{F}(\bar{e}, \mathcal{P}_s, d, c) = \{x, y \in [0,1],\ s.t.\ (1a) - (1d),\ (3),\ (4)\}$$

4 计算结果

现在，我们开展初步实验，以研究我们提出的新不等式的有效性。我们使用 IBM ILOG CPLEX 22.1 求解器，运行该求解器的机器配备了 Intel® Xeon® E5-2650v3CPU，主频为 2.30GHz。

虚拟网络为 100 个（连通的）Erdős–Rényi 网络 [7], ，每个网络有 14 个节点和 22 条边，每个节点和边的需求均为单位需求。底层网络为 Intellifiber、Uninett 和 TW（70–75 个节点），它们是来自 Internet Topology Zoo [9] 的真实网络拓扑。边的容量随机生成；对于节点容量，我们考虑 25%、50% 和 100% 的节点容量为 1（其余为 0）的情形，以测试在容量分布较稀疏的情况下不等式的效率。成本随机生成。

容量	约束	Time	节点	v(LP)
0.25	FF	46.5	11564.6	29.6
	FF + FD	22.4	7440.5	72.1
	FF + FD +FC	24.6	4740.8	86.4
0.5	FF	107.0	25638.5	20.4
	FF + FD	38.4	12683.6	51.8
	FF + FD +FC	65.1	9124.3	52.7
1.0	FF	211.1	38124.9	14.7
	FF + FD	106.6	16377.0	40.9
	FF + FD +FC	224.4	12340.2	40.9

表1。实验结果

结果见表1。每行给出了在给定比例的容量受限节点（第一列）情况下，针对一组不等式的 CPLEX 平均性能。其余各列报告求解时间（秒）、分支定界节点数以及线性松弛的值。

流量离开（FD）不等式非常有效，可带来 2 到 3 倍的加速效果，并显著改善松弛。请注意，使用来自 [9],的小型真实虚拟网络（如 Abilene 和 Polska）时，我们获得了更显著的效果（最高可达

20 倍）。流量连续性（FC）不等式也能缩小 B&B 树规模并改善线性松弛（在容量稀疏的情形下），但总体而言会拖慢 MIP 求解器。这是因为，尽管这些不等式的数量呈多项式增长，但数量仍然很多（超过 4000 条）。仅添加其中一小部分不等式（例如采用分支割平面法），或许就能使其在实践中发挥作用。

5 结论

本文首次研究了虚拟网络嵌入问题的多面体。我们提出了新的有效不等式，并证明在虚拟边对应一条底层路径的情形下，这些不等式刻画了该多面体。实验表明，这些不等式有助于混合整数规划求解器。我们认为，通过调整证明方法，这一理论结果可以推广到底层网络树。随后，还应研究与比单条边更大的虚拟网络子图（例如星形图）相关的新有效不等式，以进一步提升混合整数规划求解器的性能。

参考文献

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: 网络流。马萨诸塞州剑桥：Alfred P. Sloan 管理学院，马萨诸塞州（1988） 2. Alliance, N.: 网络切片概念说明。下一代移动网络联盟 5G P **1**(1), 1–11（2016） 3. Amaldi, E., Coniglio, S., Koster, A.M., Tieves, M.: 关于虚拟网络嵌入问题的计算复杂性。《离散数学电子札记》**52**, 213–220（2016） 4. Anderson, T., Peterson, L., Shenker, S., Turner, J.: 通过虚拟化克服互联网僵局。《计算机》**38**(4), 34–41（2005） 5. Benhamiche, A., Fouilhoux, P., Létocart, L., Perrot, N., Schneider, A.: 均匀需求下虚拟网络嵌入的复杂性。arXiv 预印本 arXiv:2501.10154（2025） 6. Chowdhury, M., Rahman, M.R., Boutaba, R.: Vineyard：协调节点和链路映射的虚拟网络嵌入算法。《网络》**20**(1), 206–219（2011） 7. ERDdS, P., R&wi, A.: 关于随机图 I。《德布勒森数学出版物》**6**(290-297), 18（1959） 8. Fischer, A., Botero, J.F., Beck, M.T., De Meer, H., Hesselbach, X.: 虚拟网络嵌入：综述。《IEEE 通信调查与教程》**15**(4), 1888– 1906（2013） 9. Knight, S., Nguyen, H.X., Falkner, N., Bowden, R., Roughan, M.: 互联网拓扑动物园。《IEEE 通信领域精选期刊》**29**(9), 1765– 1775（2011） 10. Melo, M., Sargento, S., Killat, U., Timm-Giel, A., Carapinha, J.: 最优虚拟网络嵌入：节点-链路公式。《IEEE 网络与服务管理汇刊》**10**(4), 356–368（2013） 11. Moura, L.F., Gaspary, L.P., Buriol, L.S.: 单路径虚拟网络嵌入问题的分支定价算法。《网络》**71**(3), 188–208（2018） 12. Raack, C., Koster, A.M., Orlowski, S., Wessäly, R.: 容量约束网络设计多面体的基于割的不等式。《网络》**57**(2), 141–156（2011）

13. Rost, M., Schmid, S.：关于虚拟网络嵌入的困难性和不可近似性。《IEEE/ACM 网络汇刊》**28**(2), 791–803 (2020)