arXiv:2602.03455v1 [cs.MA] 3 Feb 2026

交叉代价下多智能体路由的博弈论与算法分析*

Tesshu Hanaka[1],Nikolaos Melissinos[2],和 Hirotaka Ono[3]

[1]九州大学，日本福冈， hanaka@inf.kyushu-u.ac.jp[2]计算机科学研究所，查理大学数学与物理学院，捷克共和国布拉格， melissinos@iuuk.mff.cuni.cz[3]名古屋大学，日本名古屋， ono@nagoya-u.jp

摘要

在共享网络上协调多个自主智能体的移动，是算法机器人学、智能交通和分布式系统中的一项基础性挑战。主流方法多智能体路径规划（MAPF）依赖集中式控制和同步避碰，通常需要严格同步，并保证全局无冲突执行。本文在混合图上提出了交叉代价下多智能体路由模型，这是一个专为异步设定设计的新型框架。在我们的模型中，冲突不再被视为硬约束，而是为每个智能体分配一条路径，并通过成本函数评估系统，该成本函数用于衡量潜在的对向相遇。“交叉代价”被定义为沿一条边以相反方向行进的智能体数量之积，用于量化去中心化执行中的拥塞和延迟风险。

我们的贡献兼具博弈论和算法层面。我们将这一情境建模为一种具有非标准成本函数的拥塞博弈，证明纯纳什均衡的存在性，并分析趋向这些均衡的动态过程。在温和条件下，可以在多项式时间内找到均衡；而一般情形则是 PLS 完全。从优化角度看，判定是否存在交叉代价为零的解，是 Steiner Orientation 问题的推广，因此该问题为 NP 完全。为应对这一计算难题，我们设计了一系列用于最小化交叉代价的参数化算法，参数包括弧、边和智能体的数量，以及顶点覆盖等图的结构性度量。根据参数不同，这些算法可得到 XP 或固定参数可解的结果，为结构受限的实例提供算法策略。我们的框架为去中心化多智能体路由奠定了新的理论基础，将均衡分析与参数化复杂性相结合，以支持可扩展且能考虑风险的协调。

1 引言

在共享网络上协调多个自主智能体的移动，是算法机器人学、智能交通和分布式系统中的一项基本挑战。这一任务的核心抽象是多智能体路径规划（MAPF）问题，并已广泛应用于现实世界，例如仓库管理、自动驾驶车辆协调和多机器人系统 [23, 31, 32]。在 MAPF 中，目标是计算无碰撞路径，防止智能体同时占据同一顶点，或沿相反方向 traversing 同一条边。经典的 MAPF 表述通常假设采用集中式控制、同步移动和紧密协调来执行这些约束，其主要目标是最小化完工时间或总成本。

*Tesshu Hanaka 部分获得 JSPS KAKENHI 资助，资助项目编号为 JP21K17707、JP22H00513、JP23H04388、JP25K03077；同时获得 JST、CRONOS、日本的资助，项目编号为 JPMJCS24K2。Nikolaos Melissinos 部分获得查理大学项目 UNCE 24/SCI/008 和 PRIMUS 24/SCI/012 的资助，以及 GAˇCR 项目 25-17221S 的资助。Hirotaka Ono 部分获得 JSPS KAKENHI 资助，资助项目编号为 JP20H05967、JP22H00513、JP24K02898、JP25K03077；同时获得 JST、CRONOS、日本的资助，项目编号为 JPMJCS24K2。

相比之下，许多现实世界中的多智能体系统，例如配送机器人车队、道路网络上的自动驾驶车辆，或处于通信受限环境中的智能体，都是在异步时序和分散式决策下运行的。在此类情境中，强制实施严格同步或保证全局无冲突执行往往并不现实。相反，可以为智能体提供路径（即需要遵循的顶点序列），而其执行的具体时序仍未确定。因此，潜在冲突与其被视为硬约束，不如被看作由路线交互引起的、可量化的风险。

用于对此类场景建模的一种强大且经过深入研究的框架，是由 Rosenthal [27] 提出的网络拥塞博弈。此类博弈中，每个智能体都有一对顶点：源点和目的地。随后，每个智能体会自私地在网络上选择一条连接其源点与目的地的路径，以最小化基于边拥塞的成本函数；该函数通常定义为随经过某条边的智能体数量增加而非递减的函数。这种建模方式反映了智能体独立运行、并且在执行过程中可能仅以有限或不可预测的方式相互作用的情形。众所周知，网络拥塞博弈必定存在纯策略纳什均衡 [27]。尽管一般情况下寻找此类均衡是 PLS 完全的 $[1, 15]$, ，但在某些特定情形下可以高效求解，例如所有智能体共享同一个源点—终点对时 [15]。

在本文中，我们提出一种新颖的拥塞博弈模型，称为**交叉代价多智能体路由（CC-MAR）**。与传统网络拥塞博弈不同，代价并非由边上的智能体总数引起，而仅由沿相反方向通行的智能体产生；只有潜在的迎面相遇才会受到惩罚。这种非对称成本结构反映了实际场景，例如狭窄走廊或车道中的单向流动，在这些场景中，同向通行通常不会造成干扰。图 1 展示了我们模型的一个示例。

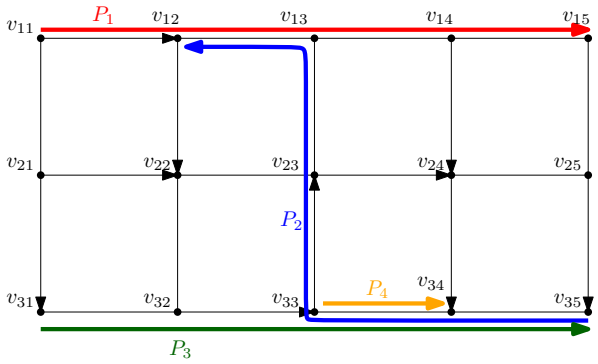

图 1：CC-MAR 示例。在此示例中，有 4 个智能体，其终端对分别为（v_{11}，v_{15}）、（v_{35}，v_{12}）、（v_{31}，v_{35}）和（v_{33}，v_{34}）。每个智能体的策略为 P_i for $i \in \{1, 2, 3, 4\}$。路径 P_1 与路径 P_2 在边 $\{v_{12}, v_{13}\}$上交叉。因此， P_1 与P_2 之间的交叉代价为 1。智能体 1 的个体成本也为 1，因为它没有其他交叉。另一方面，智能体 2 的个体成本为 4。这是因为它在边 $\{v_{12}, v_{13}\}$上与 P_1 交叉，在边$\{v_{33}, v_{34}\}$，$\{v_{34}, v_{35}\}$上与 P_3 交叉，并在边 $\{v_{34}, v_{35}\}$上与 P_4 交叉。策略组合的总成本是所有两两交叉成本之和，即 1（来自 P_1，P_2）$+2$（来自 P_2，P_3）$+1$（来自 P_2，P_4）$=4$。

1.1 我们的贡献

我们从 CC-MAR 博弈论和算法视角研究该模型。表 1 总结了 我们的贡献。

从博弈论的角度，我们分析智能体自私地选择各自路径以最小化其个体交叉代价的策略行为。在此设定下，我们首先证明该博弈始终存在（纯）纳什均衡，并证明最佳响应动态保证在至多 $w_{\max}k^2m$ 步内收敛，其中 $w_{\max}$ 是各边中的最大权重。该结果源于我们的如下证明：任何具有最小代价的策略组合都是纳什均衡，这意味着稳定价格（PoS）为1。与之形成鲜明对比的是，我们给出了一个无政府价格（PoA）无界的实例。接下来，我们考虑寻找纳什均衡的复杂性，表明若权重以输入规模的多项式为界，则可在多项式时间内求解，否则该问题成为PLS完全。

表 1：我们对交叉成本多智能体路径规划（CC-MAR）问题的贡献总结。

视角	**问题设定**	**主要贡献与复杂度**
博弈论分析	均衡的存在性与稳定性	• 始终存在纳什均衡（**NE**）存在。 •稳定性价格（**PoS**）为 1 •无政府状态价格（**PoA**）为无界。
	寻找纳什均衡	• **多项式时间**，对于多项式有界边权。 •**对于一般情况**是 PLS 完全权重。
优化（最小成本解）	一般图	**NP 完全**（广义斯坦纳定向）。
	参数化算法	
	关于终端对数量 k	**XP**(由于 *W[1]*-难解性而紧的斯坦纳定向。)
	关于边数量 $\|E\|$	**FPT**
	关于弧数量 $\|A\| + k$	**FPT**
	关于弧数量 $\|A\| + diam(G)$	**FPT**
	关于顶点覆盖数 vc $+ k$	**固定参数可解**（无权情形）
	关于顶点覆盖数 vc	**固定参数可解**（无权、无重叠终端情形）
	关于 $(V \times V) \setminus \mathcal{T}$的大小	**固定参数可解 (**计算零成本解解**)**

转向集中式优化的视角，我们研究寻找最小成本策略组合的问题，该组合被称为最优解。我们首先观察到，我们的问题推广了Steiner Orientation问题。具体而言，判定是否存在零代价解等价于求解Steiner Orientation问题的一个实例。由于已知斯坦纳定向在非常受限的图类中是NP完全的 [20], ，因此为CC-MAR寻找最优解同样如此。为规避这一困难，我们采用参数化复杂性框架。

我们的分析得出了一套全面的参数化算法。首先，我们给出一个以终端对数量 k 为参数的 XP 算法。接着，我们开发了两种固定参数算法：一种以边数为参数，另一种以弧数加终端对数量为参数。随后，我们考虑限制更严格的实例，尤其关注无权图。在此，我们证明了以顶点覆盖数加终端对数量为组合参数时具有固定参数可处理性。我们还研究了限制更严格的情形，例如不含重叠终端顶点的实例。

至关重要的是，我们的许多算法结果都已证明是紧的，因为它们受到斯坦纳定向已有困难性结果的限制。斯坦纳定向以终端对数量 [5] 为参数时具有 W[1]-困难性，这意味着我们的 XP 算法不太可能改进为固定参数算法。此外，由于斯坦纳定向以顶点覆盖数 [20], 为参数时具有 W[1]-困难性，我们不认为仅以顶点覆盖数为参数能够实现固定参数可处理性。

综上，我们的研究结果不仅提供了深刻的算法见解，也从结构层面揭示了非对称交互成本下的多智能体路由问题，有助于为无法实现严格同步的系统设计更稳健、可扩展的分散式协调机制。

1.2 相关工作

我们的模型处于多智能体路径规划、拥塞博弈和图定向的交叉领域。本节将介绍这些领域的相关工作。

多智能体路径规划（MAPF）多智能体路径规划（MAPF）在理论和实践领域都受到了广泛关注，并有许多现实应用，例如仓库管理、自动驾驶车辆协调和多机器人系统 [23, 25, 26, 28, 31,32]。大量研究致力于通过考虑受限实例和参数化复杂性来克服 MAPF 的计算难题 [11, 14, 16, 18]。此外，多年来还提出了 MAPF 的多种变体，涉及能耗 [10], 通信约束 [17,30], 延迟后重新调度 [22], 等概念 [2,24,29,33]。与经典 MAPF 将同步系统建模为具有严格碰撞约束的系统不同，我们的工作关注具有异步系统和软性成本函数。因此，CC-MAR 为去中心化系统提供了一种更灵活、也可以说更贴近现实的模型，适用于时序不确定的场景。

拥塞博弈 拥塞博弈由 Rosenthal [27] 首次提出。在标准拥塞博弈中，玩家从一组共享资源中进行选择，每个资源的代价取决于使用该资源的玩家数量。Rosenthal 通过势函数论证证明，每个拥塞博弈都存在纯策略纳什均衡。

CC-MAR 模型与网络拥塞博弈的关系最为密切，其中图边充当资源。在该设定中，对于拥有终端对 (s, t) 的玩家，其策略是选择一条s-t 路径。玩家的总成本是其所选路径中各边代价之和，其中一条边的代价是使用该边的智能体数量的非递减函数。若所有玩家共享同一终端对 [15], ，则可在多项式时间内计算出纳什均衡，但在一般情形下寻找一个纳什均衡是 PLS 完全的 [1, 15]。

斯坦纳定向 CC-MAR 的零代价版本等价于 Steiner Orientation 问题，即必须对混合图中的无向边进行定向，以连接一组终端对。该问题是 NP 完全的，即使在树宽为 2 [3,20]的图上亦然。从参数化角度看，其复杂性已得到充分理解：当以边数和弧的数量作为参数时，该问题是固定参数可解的，但当以终端对数量或顶点覆盖数作为参数时，它是 W[1]-难的（尽管属于 XP） [6, 9, 20]。这些关于斯坦纳定向的困难性结果直接适用于 CC-MAR。

2 预备知识

记号与问题设定。令 $G = (V, E, A)$ 为一个混合图，其中 $E \subseteq \binom{V}{2}$ 表示 G 中（无向）边的集合，$A \subseteq V \times V$表示 G中有向边的集合。无向边简称为边，有向边简称为弧。如果每条边 $e \in E$ 的权重为 w_e，则称图 G 为加权图。本文始终假设边权为正整数。

顶点序列 $v_1 - \ldots - v_m$ 是 G 中的一条路径，当且仅当 $i, j \in [m]$若 $i \neq j$，则 $v_i \neq v_j$，且对于任意 $i \in [m-1]$，$\{v_i, v_{i+1}\} \in E$或（v_i，v_{i+1}）$\in A$。

如果边 $\{u, v\} \in E$ 上的 P_1 和 P_2 以相反方向通行，即 u 和 v 在两条路径中均相邻出现，且 u 在 P_1 中先于 v，当且仅当 v 在 P_2中先于 u。我们将交叉代价 cost(P_1, P_2) 定义为边 $\{u, v\} \in E$的权重 w_{uv}之和，其中这些边由 P_1 和 P_2以相反方向通行。

对于每条边 $e = \{u, v\} \in E$，我们定义 $\overrightarrow{e} = (u, v)$ 和 $\overleftarrow{e} = (v, u)$。令 $x_{\overrightarrow{e}}$（分别为 $x_{\overleftarrow{e}}$）表示 $\mathcal{P}$中位于 $\overrightarrow{e}$（分别为 $\overleftarrow{e}$）上的智能体数量。不失一般性，我们假设路径 P使用边 $e = \{u, v\}$时，也使用 $\overrightarrow{e} = (u, v)$。我们用 $E(P)$ 表示路径 P所使用的边的集合。

给定一组路径 $\mathcal{P} = \{P_1, \ldots, P_k\}$，路径 P_i 的代价定义为：

$$\mathrm{cost}_{\mathcal{P}}(P_i) = \sum_{e \in E(P_i)} w_e x_{\overleftarrow{e}}$$

$\mathcal{P}$ 的总成本定义为:

$$\begin{aligned}\text{cost}(\mathcal{P}) &= \frac{1}{2}\sum_{i=1}^{k}\sum_{e\in E(P_i)} w_e x_{\overleftarrow{e}} = \sum_{e\in E} w_e x_{\overrightarrow{e}} x_{\overleftarrow{e}} \\ &= \sum_{1\le i<j\le k} \text{cost}(P_i,P_j) = \sum_{\{u,v\}\in E} w_{uv}x_{uv}x_{vu}\end{aligned}$$

我们的模型正式定义如下。

交叉成本多智能体路径规划 (CC-MAR)**输入：** $\Gamma=(G,\ \mathcal{T})$，其中 $G=(V,\ E,\ A)$ 是一个（带权）混合图，$\mathcal{T}$是一个多重集，其中包含 k 对顶点 $(s_i,\ t_i)\in V\times V, i\in[k]$.**策略组合：** k 条路径 $P_i=s_i-\ldots-t_i, i\in k$构成的集合 $\mathcal{P}=\{P_1,\ldots,P_k\}$。**智能体的代价：** 代价$_{\mathcal{P}}(P_i)$ $=\sum_{e\in E(P_i)} w_e x_{\overleftarrow{e}}$**总成本：** 代价$(\mathcal{P})$ $=\sum_{\{u,v\}\in E} w_{uv}x_{uv}x_{vu}$

如果对于每个 $i\in[k]$，G 中都至少存在一条 s_i-t_i 路径，我们就称 $\Gamma=(G,\ \mathcal{T})$ 是可行的。本文假设 Γ是可行的，因为我们可以通过在由以下方式得到的有向图中为 $i\in[k]$ 计算一条 s_i-t_i 路径，从而在多项式时间内检查 Γ 是否可行：将 E 中的每条边 $\{u,v\}$替换为有向弧 $(u,\ v)$ 和 $(v,\ u)$。在处理 CC-MAR的优化版本时，我们将一个策略组合称为可行解，而将总成本最低的可行解称为最优解。

参数化复杂性。正如我们前面提到的，为了克服计算 CC-MAR 最优解时的可处理性难题，我们将转向参数化复杂性。参数化复杂性通过引入一种称为参数的次级度量，扩展了经典复杂性分析。在这一框架下，问题被定义为一对 $(x,\ k)\in\Sigma^*\times\mathbb{N}$，其中 k 是参数。其目标是将所有计算难度都集中到参数上，并提供能够随着输入规模增长而良好扩展的算法。具体而言，如果一个问题能够在 $f(k)|x|^{O(1)}$ 时间内求解（其中 f 是可计算函数），则称其具有固定参数可处理算法（FPT）。类似地，如果一个问题能够在 $|x|^{f(k)}$ 时间内求解，则称其具有逐层多项式（XP）算法。具有 FPT 算法的问题属于复杂性类 FPT，而具有 XP 算法的问题属于 XP 类。在标准计算复杂性假设下，经证明为 W[1]-困难的问题（通过参数化归约）不属于 FPT，而经证明为 ParaNP-困难的问题（通过参数化归约）不属于 XP。我们推荐专著 [8,13,19]，其中专门探讨了参数化复杂性。

3 博弈论结果

本节中，我们研究 CC-MAR 上的纳什均衡。

3.1 纳什均衡的存在性与收敛性

定理1. 纳什均衡总是存在，且纳什动态至多经过 $w_{\max}k^2|E|$ 步即可收敛，其中 $w_{\max}$ 是边的最大权重。

证明. 我们证明 cost($\mathcal{P}$) 是一个有效的势函数。令 $\mathcal{P}$ 为 s_i-t_i 路径 $P_1,\ldots,P_k$ 的集合。对于每条边 $e=\{u,v\}\in E$，我们定义 $\overrightarrow{e}=(u,v)$ 和 $\overleftarrow{e}=(v,u)$。令 $x_{\overrightarrow{e}}$（分别为 $x_{\overleftarrow{e}}$）表示在 $\mathcal{P}$ 中位于 $\overrightarrow{e}$（分别为 $\overleftarrow{e}$）上的智能体数量。那么 $\mathcal{P}$ 的代价记为 cost($\mathcal{P}$) $=\sum_{e\in E} w_e x_{\overrightarrow{e}} x_{\overleftarrow{e}}$。

如果 $\mathcal{P}$ 不是纳什均衡，则存在一个智能体 i 希望将其路径 P_i 更改为一条代价低于 P_i的路径P_i'。令 $\mathcal{P}'=\{P_1,\ldots,P_i',\ldots,P_k\}$。不失一般性，我们假设每当 P_i使用一条边 $u,\ v\in E$时，它都沿方向 $\overrightarrow{e}=(u,\ v)$行进。我们根据 P_i与 P_i'之间的差异对边进行分类:

- $E_{\overrightarrow{i}}$表示仅由 P_i使用的边的集合，即 $E(P_i)\setminus E(P_i')$。不失一般性，假设 P_i使用 $\overrightarrow{e}$。

- $E_{\overleftarrow{i}}$表示仅由 P_i'使用的边的集合，即 $E(P_i') \setminus E(P_i)$。此外，假设 P_i'使用 $\overleftarrow{e}$。
- F_i表示 P_i和 P_i'沿相同方向使用的边集合。此外，假设 P_i和 P_i'都使用 $\overrightarrow{e}$。
- R_i表示 P_i和 P_i'沿不同方向使用的边集合。此外，假设 P_i使用 $\overrightarrow{e}$，而 P_i'使用 $\overleftarrow{e}$。
- $\hat{E}_i$表示 P_i和 P_i'都未使用的边集合，即 $E \setminus (E(P_i) \cup E(P_i'))$。

那么， $\mathcal{P}$ 的总成本可表示如下:

$$\mathrm{cost}(\mathcal{P}) = \sum_{e \in E_{\overrightarrow{i}} \uplus E_{\overleftarrow{i}} \uplus F_i \uplus R_i \uplus \hat{E}_i} w_e x_{\overrightarrow{e}} x_{\overleftarrow{e}}.$$

因此，我们有:

$$\mathrm{cost}(\mathcal{P}') - \mathrm{cost}(\mathcal{P}) \sum_{e \in E_{\overleftarrow{i}}} w_e x_{\overrightarrow{e}} - \sum_{e \in E_{\overrightarrow{i}}} w_e x_{\overleftarrow{e}} + \sum_{e \in R_i} w_e (x_{\overrightarrow{e}} - x_{\overleftarrow{e}} - 1).$$

另一方面，由于

$$\mathrm{cost}_{\mathcal{P}}(P_i) = \sum_{e \in E_{\overrightarrow{i}}} w_e x_{\overleftarrow{e}} x_{\overrightarrow{e}} + \sum_{e \in F_i} w_e x_{\overleftarrow{e}} + \sum_{e \in R_i} w_e x_{\overleftarrow{e}},$$

我们有:

$$\begin{aligned} \mathrm{cost}_{\mathcal{P}'}(P_i') - \mathrm{cost}_{\mathcal{P}}(P_i) &= \sum_{e \in E_{\overleftarrow{i}}} w_e x_{\overrightarrow{e}} - \sum_{e \in E_{\overrightarrow{i}}} w_e x_{\overleftarrow{e}} + \sum_{e \in R_i} w_e (x_{\overrightarrow{e}} - x_{\overleftarrow{e}} - 1) \\ &= \mathrm{cost}(\mathcal{P}') - \mathrm{cost}(\mathcal{P}). \end{aligned}$$

根据假设，$\mathrm{cost}_{\mathcal{P}'}(P_i') - \mathrm{cost}_{\mathcal{P}}(P_i) < 0$，$\mathrm{cost}(\mathcal{P}') - \mathrm{cost}(\mathcal{P}) < 0$成立，这意味着 $\mathrm{cost}(\mathcal{P})$ 是势函数。由于策略组合的最大代价至多为 $w_{\max} k^2 |E|$，纳什动态最多经过 $w_{\max} k^2 |E|$ 步即可收敛。 □

3.2 无政府代价与稳定性价格

定理2. 稳定性价格为 *1*。

证明。 根据定理1的证明，最小代价解是一个纳什均衡。 □

定理3. 即使对于单调边成本，无政府代价也无界。

证明。 设 $G = (\{a, b, c\},\ \{\{a, b\}, \{b, c\}, \{c, a\}\}$ 是一个有三个顶点的完全图。每个顶点上有两个智能体，我们将它们记为 A_1, A_2, B_1, B_2, C_1, C_2。 A_1,A_2, B_1, B_2, C_1, C_2 的终端顶点分别为 b, c, c, a, a, b。

设 $\mathcal{P}_1$ 是如下策略组合： $P_{A_1} = a - b$, $P_{A_2} = a - c$, $P_{B_1} = b - c$, $P_{B_2} = b - a$,$P_{C_1} = c - a$和 $P_{C_2} = c - b$。在 $\mathcal{P}_1$下，每个智能体的代价为1。此外，很容易看出，每个智能体都无法通过更改当前路径来降低自己的代价。因此， $\mathcal{P}_1$ 是纳什均衡，且 $\mathcal{P}_1$ 的代价为6。

接下来，考虑如下策略组合 $\mathcal{P}_2$： $P_{A_1} = a - b$、 $P_{A_2} = a - b - c$、 $P_{B_1} = b - c$、$P_{B_2} = b - c - a$、$P_{C_1} = c - a$、 $P_{C_2} = c - a - b$。那么， $\mathcal{P}_2$的代价为 0。因此，无政府代价是无界的。 □

3.3 计算纳什均衡

定理 4。 给定一个策略组合 $\mathcal{P}$，可以在$O(k(|E| \log w_{\max} + |V| \log |V|))$时间内确定 $\mathcal{P}$ 是否为纳什均衡。此外，如果 $\mathcal{P}$不是纳什均衡，则可以在相同的运行时间内找到一条改进路径。

证明。 设 $G=(V,\ E,\ A)$为一个输入图。给定策略组合 $\mathcal{P}$，我们检查每个智能体 i是否有动机改变当前路径 P_i。为此，我们定义有向图$G'=(V,\ A')$：它由 G 通过将 E 中的（无向）边替换为双向弧得到。对于每条弧$(u,\ v)\in A'$，我们将权重 w'_{uv} 设定如下：

$$w'_{uv}=\begin{cases} w_{uv}(x_{vu}-1) & (\text{if } vu\in P_i \text{ and } \{u,v\}\in E)\\ w_{uv}x_{vu} & (\text{if } vu\notin P_i \text{ and } \{u,v\}\in E)\\ 0 & (\text{otherwise}) \end{cases}$$

令 P'_i 为 G' 中从 s_i 到 t_i 的一条最短路径。可以看出， G' 中路径 P'_i 的权重，等于智能体 i 将策略改为 P'_i 时路径 P'_i 的代价。因此，如果 G' 中 P'_i 的权重小于路径 P_i 在 $\mathcal{P}$ 下的代价 $\mathsf{cost}_{\mathcal{P}}(P_i)$，智能体 i 就有动机改变策略，而 P'_i 是智能体 i 的一条改进路径。另一方面，如果不存在这样的路径，那么在其他智能体的路径固定时，任何其他从 s_i 到 t_i 的路径的代价都至少与 P_i 一样高，因此智能体 i 没有动机改变自己的路径。如果任何智能体都没有改进路径，那么 $\mathcal{P}$ 就是一个纳什均衡。由于可以在 $O(|E|\log w_{\max})$ 时间内设置边权，并且可以在 $O(|E|+|V|\log |V|)$ 时间内计算从 s_i 到 t_i 的最短路径，因此可以在 $O(k(|E|\log w_{\max}+|V|\log |V|))$ 时间内验证 $\mathcal{P}$ 是否为纳什均衡。 □

由定理1和定理 4，可得以下推论。

推论1。可以在时间$O(w_{\max}k^2|E|(|E|\log w_{\max}+|V|\log |V|))$内找到一个纳什均衡。

在本节末尾，我们将通过展示从二次阈值博弈（QTG）到该问题的 PLS 归约，证明寻找纳什均衡是 PLS 完全的；二次阈值博弈是 PLS 完全的 [1]。

定理 5。 在 *CC-MAR* 中寻找纳什均衡是 *PLS* 完全的。

由定理 4 可知，在加权 CC-MAR 中寻找纳什均衡属于 PLS。为证明其 PLS 完全性，我们引入二次阈值博弈（QTG），它是拥塞博弈的一种特殊情形。

令 $B=\{1,\dots,n\}$ 为 n 智能体的集合。令 $R=R^{\text{in}}\cup R^{\text{out}}$ 为一个集合，称为资源集，其中 $R^{\text{in}}\cap R^{\text{out}}=\emptyset$，且 $R^{\text{in}}=\{r_{ij}\mid i,j\in B\}$ 和 $R^{\text{out}}=\{r_i\mid i\in B\}$。也就是说，资源 r_i 仅由智能体 i使用。如果 i 使用 r_i，则其代价为 T_i，称为阈值。元素 $r_{ij}\in R^{\text{in}}$ 是资源，适用于$i,j\in B$。因此， r_{ij} 由 i 或 j 使用，也可能二者都使用。 r_{ij} 的代价由一个非递减函数c_{ij}定义： $\{0,1,2\}\to\mathbb{Z}_{\geq 0}$取决于使用 r_{ij}的智能体 $x_{r_{ij}}\in\{0,1,2\}$的数量。这里， $\mathbb{Z}_{\geq 0}$表示非负整数集。

每个智能体 i有两种策略 $S_i^{\text{out}}=\{r_i\}$，其中 $r_i\in R^{\text{out}}$且 $S_i^{\text{in}}=\{r_{ij}\mid r_{ij}\in R^{\text{in}},j\in B,i\neq j\}$。根据上述定义，当 i 选择策略 S_i^{out}时， i 的代价为 T_i；当 i 选择策略 S_i^{in}时， i的代价为$\sum_{j\in B,\ j\neq i}c_{ij}(x_{r_{ij}})$。因此，二次阈值博弈表示为 $\Delta=(B,\ \{T_i\mid i\in B\},\ \{c_{ij}\mid i,j\in B,i\neq j\})$

在这里，每个智能体都能看到其他智能体的策略，并且倾向于选择代价较低的策略。例如，如果选择 S_i^{in}的代价低于当前选择 S_i^{out}的代价，那么智能体 i 就有动机改变自己的策略。如果任何智能体都没有改变策略的动机，那么这样的策略组合就是纳什均衡。

[1] 表明二次阈值博弈上的纳什均衡是 PLS 完全的。

定理 6 ([1])**.** 在以下假设下，在具有非递减成本函数的二次阈值博弈上寻找纳什均衡是 *PLS* 完全的。

- 对于每一对 i 和 j，成本函数 c_{ij} 满足 $c_{ij}(0)=c_{ij}(1)=0$ 以及 $c_{ij}(2)\in\mathbb{Z}_{\geq 0}$，并且
- 对于每个 i，阈值 T_i 为正。

因此，只需证明任意二次阈值博弈 $\Delta=(B,\{T_i\mid i\in B\},\{c_{ij}\mid i,j\in B\})$ 都可以 PLS 归约到某个 CC-MAR 实例 $\Gamma=(G,\mathcal{T})$。本文给出的归约修改了 [1] 中从 QTG 到网络拥塞博弈的归约的一部分。

定理 *5* 的证明。给定一个具有 $n=|B|$ 个智能体的 QTG $\Delta=(B,\ \{T_i\mid i\in B\},\ \{c_{ij}\mid i,j\in B\})$，我们构造一个如图 2 所示的网格状图。该图由一个网格部分组成，其顶点

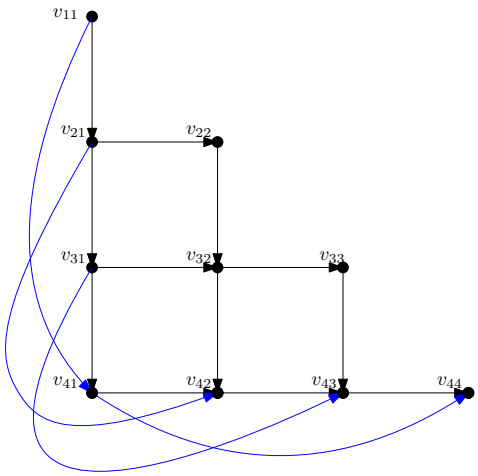

图 2：当 $n = 4$ 时的网格状图。蓝色弧为阈值弧。

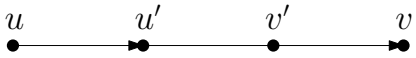

图3：将弧 (u, v) 替换为由 (u, u') $\{u', v'\}$,(v', v) 组成的混合 u-v 路径。

排列在一个 $n \times n$ 网格的左下三角区域中，其中包括对角顶点；垂直弧的方向向下，水平弧的方向从左向右。我们将位于第 i 行、第 j 列的顶点记为 v_{ij} 。此外，对于每个 $i \in [n]$，图中都有从 v_{i1} 到 v_{ni} 的弧。我们称弧 (v_{i1}, v_{ni}) 为智能体 i 的 阈值弧。

然后，我们定义 n 个智能体 $p_1, \ldots, p_n$，称为主代理。对于 $i \in [n]$，它们的终端对给定为 $(s_i, t_i) = (v_{i1}, v_{ni})$。直观上，每个智能体 i 有两种策略： i 的阈值弧 (s_i, t_i)，以及从 s_i 到 t_i的行列路径，该路径先沿第 i行的水平弧行进，直到第 i列，然后沿第 i列的垂直弧向下到达 t_i 。选择这两条路径对应于在 QTC 中选择 S_i^{out}和 S_i^{in}。

为迫使每个智能体选择其阈值弧或其行列路径，我们进一步修改该实例。首先，我们将每条水平弧和阈值弧 (u, v) 替换为由 (u, u') $\{u', v'\}, (v', v)$ 组成的混合 u-v 路径，如图3所示。设 D 为一个大整数。若弧 (u, v) 是第 i行中的水平弧，我们将边 $\{u', v'\}$ 的权重定义为 $w_{u'v'} = D \cdot i$。若弧 (u, v) 是阈值弧，我们将边 $\{u', v'\}$ 的权重定义为 $w_{u'v'} = D \cdot i \cdot (i-1) + T_i$。此外，对于每条边 $\{u', v'\}$，我们引入一个智能体 $a_{v'u'}$ ，其终端对为 (v', u')。注意，根据该构造，智能体 $a_{v'u'}$ 总是选择路径 $\langle v', u' \rangle$ 作为其策略。

最后，为了模拟资源 r_{ij}的代价，对于 $1 \le i < j \le n$，我们将顶点 v_{ij} 拆分为两个顶点 $v_{ij}^{(1)}$ 和 $v_{ij}^{(2)}$ ，并添加权重为 c_{ij}的边 $\{v_{ij}^{(1)}, v_{ij}^{(2)}\}$ 。指向 v_{ij} 的入弧与 $v_{ij}^{(1)}$ 相关联，而从 v_{ij} 出发的出弧与 $v_{ij}^{(2)}$ 相关联。然后我们引入智能体 a_{ij} ，其终端对为$(v_{ij}^{(2)}, v_{ij}^{(1)})$。与 $a_{v'u'}$类似， a_{ij}的策略仅为路径 $\langle v_{ij}^{(2)}, v_{ij}^{(1)} \rangle$。我们将这一替换部分称为 r_{ij}的资源构件（见图4）。至于 $2 \le i \le n$所对应的智能体 p_i的终端对，我们将其设为$(s_i, t_i) = (v_{i1}^{(1)}, v_{ni}^{(1)})$。智能体 p_1的终端对设为$(s_1, t_1) = (v_{11}, v_{n1}^{(1)})$。由于 D是一个足够大的整数，每个智能体 p_i 仅有两种策略： i 的阈值弧(s_i, t_i)，以及从 s_i 到 t_i的行列路径；该路径先沿第 i 行的水平弧行进至第 i 列，再沿第 i列的垂直弧向下到达 t_i 。

i的阈值路径代价显然为 $D \cdot i \cdot (i-1) + T_i$。 i 的行列路径代价是其经过的 i 水平无向边和资源构件的代价之和，等于 $D \cdot i \cdot (i-1) + \sum_{j \in B,\ j \ne i} c_{ij}(x_{r_{ij}})$，其中 $x_{r_{ij}} \in \{0, 1, 2\}$ 表示资源构件 r_{ij}的使用次数。事实上，如果智能体 i 选择行列路径，它会在水平无向边上与其他智能体交叉，并在资源构件中与若干同样选择行列路径的主代理交叉。注意，成本函数 c_{ij} 满足 $c_{ij}(0) = c_{ij}(1) = 0$ 且 $c_{ij}(2) \in \mathbb{Z}_{\ge 0}$。因此，对于每个智能体 i，其在 QTG Δ 中的策略 S_i^{out}可以与加权 CC-MAR Γ中的阈值弧 i 一一对应，而其策略 S_i^{in}对应于行列

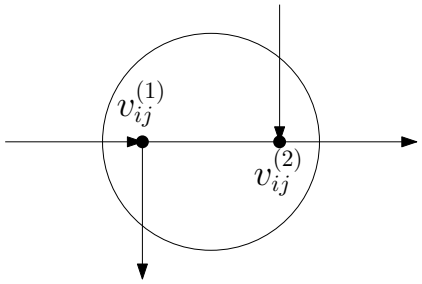

图4：将顶点 v_{ij}替换后得到的 r_{ij} 资源构件。

i的路径。此外，我们很容易看出，QTG Δ 中的策略组合是纳什均衡，当且仅当加权 CC-MAR Γ 中的策略组合 $\mathcal{P}$ 是纳什均衡。因此，我们的构造是从 QTG 到加权 CC-MAR 的 PLS 归约。

□

4 通过参数化求解最小成本解

下文中，我们考虑寻找最小成本策略组合（即最优解）的问题。在介绍主要结果之前，我们先给出贯穿本节及下一节所使用的关键观察。

环 我们始终可以假设 G 没有环。如果并非如此，只要存在混合环，我们就反复将一个混合环 C 收缩为单个顶点。利用经典的深度优先搜索算法 [7]，可以在多项式时间内找到这样的环。最终，我们得到结果图 $G' = (V',\ E',\ A')$，其中不含混合环。该过程是安全的，因为任何从 $u \in C$ 移动到 $v \in C$ 的智能体都可以在 C 内无代价移动。下文中，我们假设 G 是无环的，因此 $G[A]$是有向无环图，而 $G[E]$ 是森林。

悬挂顶点 我们始终可以假设给定图中没有悬挂顶点，即度为1的顶点。具体而言，在删除悬挂顶点之前，我们说明如何处理它们。

注意，任何不含终端的悬挂顶点都是多余的，因为它不会出现在任何路径中（若使用它，结果将是游走）。因此，可以将其删除。

现在，考虑作为终端出现的悬挂顶点 v。令 u为G中 v 的唯一邻居。若 $\{u, v\} \notin E$，则 v 与一条弧关联。由于弧不会影响解的代价，且存在可行解（根据假设），我们可以将 $\mathcal{T}$中所有出现的 v 替换为 u，并删除v。

接下来，考虑 $\{u, v\} \in E$的情况。若 v 仅作为源点或仅作为汇终端出现，则边 $\{u, v\}$ 要么仅从 v 通向 u，要么仅从 u 通向 v。因此，它不会影响代价，我们可以放心地将 $\mathcal{T}$中所有出现的 v 替换为 u，并删除 v。

现在，假设 v 在 $\mathcal{T}$中既作为源点又作为汇终端出现。令 ℓ_s 和 ℓ_t 分别表示 v 作为源点和汇终端出现的次数。由于我们假设不存在满足条件的点对 $(v,\ v) \in \mathcal{T}$，任何可行解都包含 ℓ_s条路径，它们使用 $\{u, v\}$从 v通往 u；还包含 ℓ_t条路径，它们使用 $\{u, v\}$从 u通往 v。此外，没有其他路径使用边 $\{u, v\}$。因此，代价为 $\ell_s \cdot \ell_t$。所以，我们可以将 $\mathcal{T}$中所有出现的 v 替换为 u，删除 v，并相应调整代价（即，在判定版本中将代价设为 $\ell - \ell_s \cdot \ell_t$，或者在计算出新实例的解后将代价增加 $\ell_s \cdot \ell_t$）。

由此，我们得出了一个关键观察。

观察1（输入简化）。 不失一般性，可以假设输入图是无环的，且不包含度为1 的顶点。

相同的终端对 请回想定义中，我们允许同一对在 $\mathcal{T}$ 中多次出现。这一点很重要，因为相同的终端对多次出现会影响

代价，并且根据每对出现的次数，可能会导致完全不同的最优解。

在本部分中，我们证明，无论某个终端对出现多少次，都存在一个最优解，使所有相同的终端对都使用完全相同的路径来连接其终端。为证明这一点，考虑一个实例 $(G, \mathcal{T})$ 及一个最优解 $\mathcal{P}$，满足：

- 存在两个终端对 $(s_i, t_i) = (s_j, t_j)$，其中 $0 < i < j \leq k$，并且
- 最优解中的 s_i-t_i 路径和 s_j-t_j 路径为 $P_i, P_j \in \mathcal{P}$，且 $P_i \neq P_j$。

我们将说明，可以选取路径 P_i 或 P_j，并将两条路径都替换为所选的那条路径 所选择的那个。

令 $P = P_i$。注意，$\mathsf{cost}(\mathcal{P} \setminus P_i) = \mathsf{cost}(\mathcal{P} \setminus P_j)$。事实上，若 $\mathsf{cost}(\mathcal{P} \setminus P_i) < \mathsf{cost}(\mathcal{P} \setminus P_j)$ 或 $\mathsf{cost}(\mathcal{P} \setminus P_i) > \mathsf{cost}(\mathcal{P} \setminus P_j)$，则可行解 $(\mathcal{P} \setminus \{P_j\}) \cup \{P_i\}$ 或 $(\mathcal{P} \setminus \{P_i\}) \cup \{P_j\}$ 的代价会小于 $\mathcal{P}$，这与最优性矛盾。

最后请注意，$(\mathcal{P} \setminus \{P_j\}) \cup \{P_i\}$ 的代价必须与 $\mathcal{P}$相同，因此，存在一种解，使得$(s_i,\ t_i)$和$(s_j,\ t_j)$由相同的路径连接。

通过重复相同的过程数次，我们可以轻松地将其推广到任意数量的相同终端对。

观察 2 (解结构)。 对于任意 CC-MAR 实例$(G, \mathcal{T})$，都存在一个最小代价解，使相同的终端对由相同的路径连接。

按顶点数参数化 我们提出一种固定参数可解（FPT）算法，该算法按顶点数 $|V|$参数化。此算法将作为本文后续介绍的若干方法的子程序。

命题 1。 对于 CC-MAR 实例$(G, \mathcal{T})$，可以在$|V|^{O(|V|^3)}k^{O(1)}$时间内计算出一个最小代价解。

证明。 $\mathcal{T}$是一个多重集。令 $\mathcal{T}'$ 为从 $\mathcal{T}$中移除重复项后得到的终端对集合。令$(s'_i,\ t'_i)$为 $\mathcal{T}'$中终端对的一个枚举，并令 k_i 为$(s'_i,\ t'_i)$在 $\mathcal{T}$中出现的次数。注意，对于所有 $i \in |\mathcal{T}'|$，都有 $k_i \geq 1$。

回想一下，观察 2 表明，$(G, \mathcal{T})$ 存在一个最优解，其中对于所有$i \in |\mathcal{T}'|$，(s'_i, t'_i) 的所有 k_i 个副本都使用完全相同的路径。

注意，如果给定每个 $i \in |\mathcal{T}'|$ 一条路径 P_i，使得 P_i 是 G 中的一条 s'_i-t'_i 路径，那么我们可以在多项式时间内计算仅使用这些路径连接终端对的可行解的代价。事实上，这样一个可行解的代价为：

$$\sum_{0<i<j\leq|\mathcal{T}'|} k_i \cdot k_j \cdot \mathrm{cost}(P_i, P_j)$$

其中，$\mathsf{cost}(P_i, P_j)$ 可以在多项式时间内计算，且 $|\mathcal{T}'| \leq |V(G)|^2$。因此，我们只需计算每个终端对在 $\mathcal{T}'$ 中使用哪些路径。

固定一对 $(s', t') \in \mathcal{T}'$。注意，两者之间至多存在 $2^{|V_{(G)}|} \cdot |V(G)|!$ 条不同路径。实际上，我们可以通过猜测哪些顶点属于 P_i 以及它们出现的顺序，枚举出所有这些路径。这也给出了路径数量的上界。

因此，为了计算 $(G, \mathcal{T})$ 的最优解，只需：

- 对每一对 $(s'_i, t'_i) \in \mathcal{T}'$，猜测其副本将使用的路径，
- 计算猜测路径的代价，并且，
- 返回代价最小的猜测。

由于在 $\mathcal{T}'$ 中至多有 $|V(G)|^2$ 对，因此猜测次数上界为 $|V(G)|^{O(|V(G)|^3)}$。由于其余计算可在多项式时间内完成，该算法的运行时间为$|V(G)|^{O(|V(G)|^3)}(|V(G)| + k)^{O(1)}$。

□

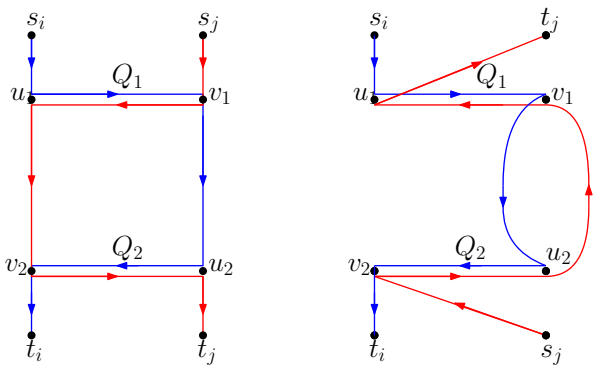

图 5：声明 1 的情形分析。

4.1 按智能体数量参数化

定理 7。 一个 CC-MAR 实例 $(G, \mathcal{T})$ 的最小代价解可在时间 $|V|^{O(k^2)}k^{O(1)}$ 内计算得到。

证明。 策略如下：(1) 我们证明一个结构性质：在最优解中，任意一对智能体至多在一条无向路径上相交；(2) 我们猜测所有 $O(k^2)$ 对智能体的这些交叉交互；(3) 对于每种猜测，我们将寻找剩余非交叉路径的问题规约为 Steiner Orientation。

声明 1。 在最小代价解 $\mathcal{P}^*$ 中，任意一对智能体在 $G[E]$ 内至多在一条无向路径上相交。

断言的证明。 反设智能体 i 和 j 在 $G[E]$ 中至少沿两条互不相交的无向路径 Q_1 和 Q_2 相交。设它们在 $\mathcal{P}^*$ 中各自的路径为 P_i 和 P_j。假设沿着 P_i，Q_1 上的交叉发生在 Q_2 上的交叉之前。令 (u_1, v_1) 和 (u_2, v_2) 分别为 Q_1 和 Q_2 的端点。不失一般性，假设 P_i 沿 Q_1 从 u_1 到 v_1，并沿 Q_2 从 u_2 到 v_2。因此，P_i 具有结构 $s_i \rightsquigarrow u_1 \rightsquigarrow v_1 \rightsquigarrow u_2 \rightsquigarrow v_2 \rightsquigarrow t_i$。相应地，$P_j$ 必须沿 Q_1 从 v_1 到 u_1，并沿 Q_2 从 v_2 到 u_2。

Th如图5所示，P_j 连接这些线段有两种基本方式。

情况1： P_j 的形式为 $s_j \rightsquigarrow v_1 \rightsquigarrow u_1 \rightsquigarrow v_2 \rightsquigarrow u_2 \rightsquigarrow t_j$。令 P_i' 为 P_i 中从v_1 到 u_2的有向子路径，令 P_j' 为 P_j 中从 u_1 到 v_2的有向子路径。我们可以通过交换这些有向子路径来为智能体重新规划路径，从而生成新路径 $P_i^* = s_i \rightsquigarrow u_1 \overset{P_j'}{\rightsquigarrow} v_2 \rightsquigarrow t_i$和 $P_j^* = s_j \rightsquigarrow v_1 \overset{P_i'}{\rightsquigarrow} u_2 \rightsquigarrow t_j$。在这个新解中，智能体 i 和 j 不再于 Q_1 或 Q_2上发生交叉。由于重新规划路径仅发生在有向子路径上（此处不会产生代价），因此不会引入新的代价，但 Q_1 和 Q_2 上的两项原有交叉代价会被消除。由此得到的解具有严格更低的总成本，这与 $\mathcal{P}^*$的最优性相矛盾。

情况2： P_j 的形式为 $s_j \rightsquigarrow v_2 \rightsquigarrow u_2 \rightsquigarrow v_1 \rightsquigarrow u_1 \rightsquigarrow t_j$。在这种情况下，路径段构成一个混合环 $v_1 \rightsquigarrow u_2 \rightsquigarrow v_1$，这与图中不存在混合环这一标准假设相矛盾。

因此，任意智能体对在 $G[E]$中至多沿一条无向路径发生交叉。 ◇

基于声明 1，我们的算法可以猜测所有交叉交互。对于 $\binom{k}{2} = O(k^2)$个智能体对中的每一对，我们猜测（a）它们是否交叉，以及（b）若交叉，则在 $G[E]$ 中的哪条路径上交叉。由于$G[E]$ 是森林，一条路径由其端点唯一确定。每个智能体对需要猜测的可能路径数以 $O(|V|^2)$为上界。因此，猜测的总数为 $|V|^{O(k^2)}$。

对于所有交叉交互的一种固定猜测，我们必须确定每个智能体的完整路径。这不仅涉及指定的无向交叉路径，还涉及每个智能体遍历这些路径的顺序。该顺序也可以通过猜测确定，每个智能体至多增加一个 $(k-1)!$ 的因子，这一因子可并入 $|V|^{O(k^2)}$ 项。

给定一个固定的猜测（包括每个智能体的交叉顺序），问题就转化为寻找一组互不交叉的有向路径。对于每个智能体 i，我们需要将其终端 s_i 连接到其第一条交叉路径的入口，将一条交叉路径的出口连接到下一条的入口，

并将最后一条交叉路径的出口连接到 t_i。这是 Steiner Orientation 的一个实例。该子问题中的终端对集合 $\mathcal{T}'$ 的大小为 $O(k^2)$。

如果该 Steiner Orientation 实例返回一个解，我们就为每个智能体构造完整路径 P_i。该 Steiner Orientation 子问题可在 $|V|^{O(|\mathcal{T}'|)} = |V|^{O(k^2)}$ 时间内求解 [9]。总运行时间为猜测次数乘以验证所需时间：$|V|^{O(k^2)} \cdot |V|^{O(k^2)} = |V|^{O(k^2)}$。最后，我们返回代价最低的有效猜测。 □

4.2 按边数参数化

定理 8. 可以在时间$2^{2^{O(|E|)}}(k|V|)^{O(1)}$内计算出 CC-MAR 实例$(G, \mathcal{T})$ 的一个最小代价解。

证明。我们将一个实例（$G, \mathcal{T}$）归约为一个等价实例（$G', \mathcal{T}'$），其规模由 $|E|$的某个函数所界定，并且该归约可通过 $|E|$在固定参数可解时间内完成。

我们构造约化图 $G' = (V', E', A')$。令 $V_E \subseteq V$为 E 中无向边端点的集合，令 $r = |V_E|$ 为与无向边关联的顶点数。注意，$r = |V_E| \leq 2|E|$。接着，我们准备两个顶点集 $S' = \{s'_U \mid U \subseteq V_E\}$ 和 $T = \{t'_U \mid U \subseteq V_E\}$，它们分别对应于 V_E的子集。G'的顶点集定义为 $V' = V_E \cup S' \cup T'$。G' 的边集 E' 与 E相同，即 $E' = E$。我们还定义弧集 A'_E，使得（u, v）$\in A'_E$，其中 $u, v \in V_E$ 当且仅当在移除 G中的所有边后得到的有向子图 $G - E$中，存在一条从 u 到 v 的有向路径。最后，我们分别将 S' 与 V_E 相连，将 T' 与 V_E相连。对于每个顶点 $s_U \in S'$，我们向 $U \subseteq V_E$中的顶点添加弧。类似地，我们从$U \subseteq V_E$中的顶点向 $t_U \in T'$添加弧。我们将从 S'到 V_E以及从 V_E到 T'的弧集分别记为 $A'_{S'}$和 $A'_{T'}$，并定义 $A' = A'_E \cup A'_{S'} \cup A'_{T'}$。所构造的图 G' 至多有 $r + 2 \cdot 2^r$ 个顶点、r^2 条边和 $2^{r+1}r + r^2$ 条弧。由于可在 $O(|G|)$内完成可达性检查，因此可以在时间 $2^{O(r)}|G|$内构造图 G'。

新的终端对 $\mathcal{T}'$ 定义在 G'上。对于每个原始智能体 i，新的起始终端为$s_U \in S'$，其中 U是 V_E 中满足 $s_i \in V$能够在 $G - E$中到达的顶点集合。类似地，新的终止终端为 $t_U \in T'$，其中 U是 V_E 中能够在 $G - E$中到达 $t_i \in V$的顶点集合。由于定义新的终端对所需时间为 $2^{O(r)}k|V|^{O(1)}$，构造新实例的总运行时间为 $2^{O(r)}k|V|^{O(1)}$。

新实例（G'，$\mathcal{T}'$）与原实例等价。G 中的一条路径对应于 G' 中的一条路径，只需将其有向路段收缩为单条弧即可，反之亦然。由于交叉代价仅产生于无向边 E上，而这些边是两个图共有的，因此任何解的代价都保持不变。

由于 G' 中的顶点数至多为 $2^{O(r)}$，根据命题 1，可以在 $2^{2^{O(r)}}k^{O(1)}$时间内找到一个最小代价解。因此，总运行时间为 $2^{2^{O(|E|)}}(k|V|)^{O(1)}$。 □

4.3 按弧数加上 k 参数化

定理 9。 对于一个 CC-MAR 实例$(G, \mathcal{T})$，可以在时间$|A|^{O(k|A|)}|V|^{O(1)}$内计算出一个最小代价解。

证明。考虑 G中的任意一条路径 P。如果 $\{a_i \mid i \in [m]\}$ 是 P 中出现的弧的集合，并且对任意 $0 < i < j \leq m$，在沿着P前进时，a_i 都先于 a_j出现，那么我们称 P具有路径形式 $a_1, \ldots, a_m$。

注意，至多有 $2^{|A|}|A|!$ 种可能的路径形式。此外，给定一个终端对（s_i, t_i）和一种路径形式 $a_1, \ldots, a_m$，我们可以在多项式时间内检查是否存在具有给定形式的 $s_i - t_i$路径。具体而言，只需验证以下三个条件是否成立：

1. 要么 $a_1 = (s_i, v)$，对某个 $v \in V$成立；要么 $a_1 = (u, v)$，并且存在一条 $s_i - u$路径 $G - A$。
2. Either $a_m = (u, t_i)$ 对于某个 $u \in V$，或者 $a_m = (u, v)$，并且存在一条 $v - t_i$路径 $G - A$.
3. 对于所有 $i \in [m-1]$，要么 $a_i = (u, v)$ 且对于某个 u，$a_{i+1} = (v, w)$，$w \in V$要么 $a_i = (u, v)$，$a_{i+1} = (v', w)$，并且存在一条 $v - v'$路径 $G - A$。

我们通过猜测最优解中每条 $s_i - t_i$ 路径的路径形式来继续。这给了我们 $2^{k|A|}(|A|!)^k$ 种可能的猜测。由于很容易检查猜测是否有效，我们可以假设只保留有效的猜测。

我们接下来定义一个称为预描述 CC-MAR 的子问题。

输入： CC-MAR的一个实例（G，$\mathcal{T}$）。此外，对于每个（s_i，t_i）$\in \mathcal{T}$路径形式 a_1^i，……，$a_{m_i}^i$ 以及一个整数 ℓ。

问题： 是否存在 k 条路径 P_i，使得对所有 $i \in k$，P_i 都是一条 s_i-t_i 路径，其路径形式为 a_1^i，……，$a_{m_i}^i$ 且 cost($\{P_1, \ldots, P_k\}$) 至多为 ℓ。

断言 2. *Pre-Described* CC-MAR可在多项式时间内求解。

断言的证明。 我们只需找到一种代价最小的方式，仅使用无向边来补全这些路径形式。为此，我们构造如下实例：

对于每一对 (s_i，t_i)，令 a_1^i，……，$a_{m_i}^i$ 为我们要寻找的 $s_i - t_i$ 条路径的路径形式。我们创建一组终端对，按如下方式替换 (s_i，t_i)：

- 若 $a_1^i = (u, v)$ 且 $u \neq s_i$，则创建终端对 (s_i，u)
- 若 $a_{m_i}^i = (u, v)$ 且 $v \neq t_i$，则创建终端对 (v，t_i)
- 如果 $a_j^i = (u, v)$、$a_{j+1}^i = (v', w)$，并且 $v \neq v'$，我们就创建这对 (v，v')。

我们对 $\mathcal{T}$ 中的所有终端对重复这一过程。令 $\mathcal{T}'$ 为新的终端集合。注意，为了创建符合给定路径形式的 s_i-t_i 条路径，我们需要用无向路径连接 $\mathcal{T}'$ 中的所有终端对。

我们可以通过求解实例 ($G - A$，$\mathcal{T}'$)，计算 $\mathcal{T}'$ 中所有顶点对之间的最小代价路径。由于 $G - A$ 是森林，因此 $G - A$ 中任意一对顶点之间都有唯一的路径，故可以在多项式时间内完成计算。

利用我们猜测的路径形式，以及 ($G - A$，$\mathcal{T}'$) 的解，我们就得到一个符合这些路径形式且交叉代价尽可能小的解。$\diamond$

注意，我们的每种猜测都会定义一个预先描述的 CC-MAR 问题实例。由于我们可以在多项式时间内计算每个此类实例的解，因此可以通过求解所有实例并保留代价最小的解，来计算 CC-MAR 的最优解。

我们算法中唯一非多项式的部分是对路径形式的猜测。因此，该算法在 $2^{k|A|}(|A|!)^k|V|^{O(1)} = |A|^{O(k|A|)}|V|^{O(1)}$ 内终止。 □

4.4 按弧数加直径参数化

定理10。 对于一个 CC-MAR 实例 $(G, \mathcal{T})$，当参数为 $|A| + diam(G)$ 时，可以在固定参数可解时间内计算出最小代价解，其中 $diam(G)$ 是 G 的直径。

证明。 我们首先证明，有界 $A + diam(G)$ 的混合图的顶点数有上界。回顾观察1可知，我们可以假设给定的图是无环的，且没有度为1的顶点。

注意，$G - A$ 是一个至多有 $2|A|$ 个叶节点的森林。此外，$\Delta(G - A) \leq 2|A|$，因为任意树的最大度至多等于其叶节点数。因此，G 的最大度至多为 $3|A|$。最后，任何直径为 $diam(G)$ 且最大度为 Δ 的图都至多有 $1 + diam(G)\Delta^{diam(G)}$ 个顶点 [12]。

最后，由于 G 中顶点的数量受 $|A| + diam(G)$ 的某个函数所界定，利用命题 1 的结果即可得到一个以 $|A| + diam(G)$ 为参数、在固定参数可解时间内运行的算法。证明至此完成。 □

5 受限设置下的最小代价解

5.1 无权图中按顶点覆盖数参数化

本节讨论无权图。我们首先证明，当以顶点覆盖数加上 k 为参数时，该问题变得易处理。随后，我们证明，如果进一步假设终端对互不重叠，则仅以顶点覆盖数作为参数就足够了。

定理11。 CC-MAR 在无权图上的问题，当以顶点覆盖数vc加智能体数量 k为参数时，是固定参数可解的。

证明。 </style}核心思路是构造一个小型等价实例，其规模关于参数 vc+k呈指数增长。我们首先计算一个大小至多为 2vc 的顶点覆盖 S（可在多项式时间内实现 [4]）。随后，我们将 S 扩充至包含所有终端顶点，使其新大小不超过 $|S| \leq$2vc $+ 2k$。

根据顶点覆盖的定义， $I = V \setminus S$ 是一个独立集。这意味着，任何从 S 进入 I 中某个顶点的路径，都必须立即返回 S中的某个顶点。

我们可以根据 I 中顶点在 S中的邻域，将它们归类为不同的“类型”。具体而言，如果两个顶点 v_1、$v_2 \in I$ 通过无向边、入弧和出弧连接到 S 中完全相同的顶点子集，那么它们属于同一类型。对于任意顶点 $v \in I$，其邻域由 S的三个子集定义。因此，不同类型的总数最多为$2^{|S|} \cdot 2^{|S|} \cdot 2^{|S|} = 8^{|S|}$。

现在，考虑 I 中一组类型完全相同的顶点。由于只有 k 个智能体，在任何给定的解中，最多只能使用其中 k 个功能相同的顶点。该类型的任何其他顶点都是冗余的。因此，我们可以放心地将每种类型的顶点数量减少到仅 k个。

该归约规则构造出一个规模较小的实例，其顶点数至多为 (2vc $+ 2k$) $+ 8^{|S|} \cdot k$。根据命题1，当以 vc $+ k$ 为参数时，该问题是固定参数可解的。 □

定理12。 设$(G, \mathcal{T})$ 是 CC-MAR 的一个实例，其中 G 是无权图，并且 V的任何顶点(G) 都不会出现在多个终端对中。当以 G的顶点覆盖数vc 为参数时，可以在 *FPT*时间内计算出$(G, \mathcal{T})$ 的一个最小代价解。

证明。 回顾一下，根据观察1，我们可以假设 G 是无环的。设 G 是一个顶点覆盖数为 vc 的图。设 $S \subseteq V$ 是 G 的一个大小为 vc 的顶点覆盖。它可以在时间$2^{\text{vc}} n^{O(1)}$ [21]内计算得出。此外，设 $I = V \setminus S$ 是相应的独立集。

现在我们定义终端对的类型。具体如下：

- 任意一对 $(s, t) \in \mathcal{T}$只要满足 $s, t \cap S \neq \emptyset$就具有唯一的类型。
- 如果满足以下条件，则两个终端对 (s, t),$(s', t') \in \mathcal{T}$具有相同的类型，其中 $s, t, s', t' \in I$：
  - $N^+(s) = N^+(s')$, $N^-(s) = N^-(s')$, and $N(s) = N(s')$, and
  - $N^+(t) = N^+(t')$, $N^-(t) = N^-(t')$, and $N(t) = N(t')$.

注意，这样至多有 vc 种不同类型，另外至多有 16^{vc} 种类型。事实上，额外的类型至多有 16^{vc} 种，因为 I 中顶点的不同邻域数量为 4^{vc}。

利用顶点对的类型，我们如下定义顶点的类型：

- 对于每种终端对类型，我们定义两种顶点类型：一种对应于该类型的源顶点，另一种对应于汇顶点。这样一来，我们至多有 $16^{\text{vc}+1} + 2$ 种 vc 类型。
- 如果两个非终端顶点 $u, v \in I$满足 $N^+(u) = N^+(v)$、 $N^-(u) = N^-(v)$，且$N(u) = N(v)$，则它们具有相同的类型。这样一来，我们至多还有 4^{vc}种额外类型。
- 最后，我们为 S 中的每个非终端顶点指定一个唯一类型，因此至多增加 vc 种类型。

注意，每种顶点类型要么包含邻域相同的非终端顶点，要么包含某个终端对类型中全为源点或全为汇点的终端顶点。此外，S 中的每个顶点都有自己的唯一类型。

最后，如果一种顶点类型包含多个顶点，那么该类型中的所有顶点都只与一条边关联。事实上，否则我们就能构造出一个环，这与 G 无环的假设矛盾。

我们利用先前定义的顶点类型，为 G 中的路径定义类型。为此，我们首先注意到，任意路径的长度至多为 2vc + 1，且任意两个相邻顶点都不属于 I。这是因为 I 中的顶点构成一个独立集。

这使得路径类型至多有 $(2^{\text{vc}}\text{vc}!)(16^{\text{vc}+1} + 4^{\text{vc}} + 1)^{\text{vc}+1}$ 种，其中第一项考虑路径中出现的 S 顶点及其排列顺序，第二项则考虑 I 中的一个顶点是否出现在 S的顶点之间；如果出现，该顶点属于什么类型。

定义了终端、顶点和路径的类型之后，我们现在可以证明，存在一个满足若干严格限制条件的最优解。

在陈述命题之前，我们先定义路径的中间顶点 。给定一条 u–v 路径 P，我们将 P的中间顶点定义为集合 $V(P) \setminus u,\ v$。

命题 3。 存在一个最优解 $\mathcal{P}$，使得对于 I 中的任意一种顶点，路径 $\mathcal{P}$中至多有两个这种类型的顶点作为中间顶点出现在 $(G, \mathcal{T})$ 中。

断言的证明。 我们从一个最优解 $\mathcal{P}$开始。设 ν 是 I 中的一种顶点类型，且至少有三个类型为 ν 的顶点在 $\mathcal{P}$ 中作为中间顶点出现。我们将说明如何修改路径，使得作为中间顶点出现的类型为 ν 的顶点仅有两个，同时不改变任何其他类型的顶点。

注意，每个类型为 ν 的顶点至多关联一条边。事实上，否则我们就能构造出一个环，这与 G 无环的假设矛盾。

我们分两种情况讨论：类型为 ν 的每个顶点要么没有关联任何边，要么恰好关联一条边。

情况1（每个顶点都不与任何边相接）：固定一个给定类型的顶点 u。考虑类型为 ν 的顶点作为中间顶点出现的任意情况；设 v 为这样的顶点。将v 替换为 u是安全的。事实上，由于 u 和 v 类型相同，它们的邻域也相同。此外，代价不会改变，因为我们只替换路径中的弧。我们可以重复此操作，直到只有 v 作为类型为 ν的中间顶点出现。

情况2（每个顶点都与一条边相接）：令 w 为所有类型为ν的顶点的公共邻居，使得对于所有类型为 ν的顶点 v，都有 $v, w \in E$。

我们定义两个顶点集 V_ν^+和 V_ν^-。我们将 V_ν^+设为由满足以下条件的类型为 ν 的顶点构成的集合： ：

- v 在路径 $P \in \mathcal{P}$中作为中间顶点出现，并且
- v 在 P中恰好紧接在 w 之前出现。

我们将 V_ν^- 定义为满足以下条件的 ν 类型顶点集合:

- v 在路径 $P \in \mathcal{P}$ 中作为中间顶点出现，并且
- v 在 P 中恰好出现在 w 之后。

我们首先修改路径，使得 V_ν^+ 至多包含一个顶点。如果 $|V_\nu^+| \leq 1$，则无需进行任何操作。因此，假设 $|V_\nu^+| \geq 2$，并任意选取一个顶点 $v \in V_\nu^+$。对于 $\mathcal{P}$ 中的每条路径，我们按以下方式进行修改:

- 对于每个 $P \in \mathcal{P}$,
- 找出任意一个 ν 类型的顶点，该顶点在 P 中出现在 w 之前,
- 将这个顶点替换为 v。

令 $\mathcal{P}'$ 为经过这些修改后的路径集合。注意，由于相同类型的顶点具有相同的邻域， $\mathcal{P}'$ 是（G, $\mathcal{T}$）的一个可行解。现在我们证明它也是一个最优解。

我们证明，从一个最优解出发，仅替换一个顶点会得到另一个最优解。令 u 为我们在路径 $P \in \mathcal{P}$中替换为 v 的顶点。此外，令 P' 为一条路径，其中 v在 $\mathcal{P}$ 中出现在 w 之前（这样的路径存在，因为 $v \in V_\nu^+$）。 $\mathcal{P}$ 中出现的边集合唯一的变化是，我们将 u，w 替换为 v，w。

令 x_1 为边 w，u（方向为（w，u））在 $\mathcal{P}$的路径中出现的次数，并令 x_2 为边 w，v（方向为（w，v））出现的次数。注意， $x_1 = x_2$。事实上，否则，在 P中将 u替换为 v，或在 P'中将 v 替换为 u，都会得到一个代价更低的可行解。

因此，在 P 中用 v 替换 u 会得到一个新的最优解。这意味着前述过程会生成一系列最优解，直至 $V_\nu^+ = v$。

利用对称论证，我们可以将 V_ν^-归约为一个单元素集合。

这表明，作为中间顶点出现的类型为 ν 的顶点仅有集合 V_ν^+ 和 V_ν^- 中的顶点。

由于前述过程只会影响类型相同的顶点，因此我们可以对所有作为中间顶点出现次数超过两次的类型重复该过程。由此，结论成立。◇

此后，我们假设存在一个满足命题 3 所述性质的最优解。由于每个终端对恰好关联两种顶点类型，对于每种终端类型，在所考虑的最优解中，至多有 4 对该类型的终端对的顶点作为中间顶点。现在，我们将证明可以预先确定哪些终端对可以作为路径中的中间顶点。在此之前，我们先定义希望允许作为中间顶点的顶点集。

对于每种终端类型 τ，令 $\mathcal{T}_\tau \subseteq \mathcal{T}$为所有类型为 τ的终端对的集合。我们任意选取 $\min 4, |\mathcal{T}_\tau|$ 对类型为 τ的终端；令 $\mathcal{T}_\tau^*$为这些终端的集合。

命题 4. 存在一个最优解 $\mathcal{P}$，使得任何在 $\mathcal{P}$ 中的某条路径上作为中间顶点出现的终端顶点 v 都属于集合 $\mathcal{T}_\tau^*$。

断言的证明。 注意，我们需要证明对于满足 $|\mathcal{T}_\tau| > 4$ 的类型 τ，该断言成立。考虑一种类型 τ 和一对 $(s,\ t) \in \mathcal{T}_\tau \setminus \mathcal{T}_\tau^*$，使得 s 或 t 会作为中间顶点出现在 $\mathcal{P}$ 的某条路径中。我们将展示如何修改 $\mathcal{P}$，以便将所有作为中间顶点出现的 s 和 t 替换为 s' 和 t' 的出现，其中 $(s',\ t') \in \mathcal{T}_\tau^*$。此外，这个过程不会影响任何其他顶点。

注意，由于在 $\mathcal{P}$ 中，至多有 4 对类型为 τ 的顶点被用作中间顶点，且 $(s,\ t) \in \mathcal{T}_\tau \setminus \mathcal{T}_\tau^*$，因此至少存在一对 $(s',\ t') \in \mathcal{T}_\tau^*$，使得 s' 和 t' 都不会作为中间顶点出现在 $\mathcal{P}$ 的任何路径中。

由于 $(s,\ t)$ 和 $(s',\ t')$ 属于相同的终端类型，将所有 s 替换为 s'，并将 t 替换为 t'，即可得到一个交叉代价完全相同的可行解。为说明这一点，注意到任何额外代价都可能来自与 s、t、s' 和 t' 相接的边。然而，由于我们替换了所有出现，因此与某条边 $\{v, w\}$ 相关的任何代价（对于 $v \in \{s, t\}$ 和一个 $w \in V$）原本就已因某个 $v' \in \{s', t'\}$ 对应的边 $\{v', w\}$ 而出现在 $\mathcal{P}$ 中。类似地，与某条边 $\{v', w\}$ 相关的任何代价（对于 $v' \in \{s', t'\}$ 和一个 $w \in V$）原本就已因某个 $v \in \{s, t\}$ 对应的边$\{v, w\}$ 而出现在 $\mathcal{P}$ 中。

最后，我们重复这一过程，直到属于类型为 τ 的终端的中间顶点仅为 $\mathcal{T}_\tau^*$中的顶点。此外，这不会影响任何其他终端类型。因此，我们可以对所有终端类型重复这一过程。◇

现在，我们证明一个类似的断言，讨论来自 I 且不是终端顶点的顶点出现的情况。对于每一种不包含终端顶点的顶点类型 ν，令 V_ν 为所有类型为 ν的顶点组成的集合。对于每种这样的类型，我们任意选取 $\min 2, |V_\nu|$ 个类型为 ν的顶点；令 V_ν^*为所选顶点组成的集合。

断言 5。 存在一个最优解 $\mathcal{P}$，使得任何出现在 $\mathcal{P}$ 中的路径上的非终端顶点 v 都属于集合 V_ν^*。

断言的证明。 证明沿用命题 4 证明中的相同论证。

我们需要证明对于满足 $|V_\nu| > 2$的类型 ν，该断言成立。考虑类型为 ν 的类型，以及一个出现在 $\mathcal{P}$中的路径上的顶点$v \in V_\nu \setminus V_\nu^*$。由于我们考虑的是非终端顶点，因此我们知道这些顶点只会作为 $\mathcal{P}$中的中间顶点出现。我们将说明如何修改 $\mathcal{P}$，以便将 v 的所有出现替换为 v'的出现，其中 $v' \in V_\nu^*$。此外，这一过程不会影响任何其他顶点。

根据命题 3，由于在 $\mathcal{P}$ 中，至多有 2 个类型为 ν 的顶点被用作中间顶点，且 $v \in V_\nu \setminus V_\nu^*$，因此至少存在一个顶点 $v' \in V_\nu^*$，它不会作为 $\mathcal{P}$ 中任何路径的中间顶点出现。

由于 v 和 v' 属于同一类型的顶点，将 v 的所有出现替换为 v' 后，我们得到一个交叉代价完全相同的可行解。为说明这一点，请注意，任何额外代价都源于与 v 和 v' 相 incident 的边。然而，由于我们替换了所有出现，对于某个 $w \in V$，与边 $v,\ w$ 相关的任何代价，已因边 $v',\ w$ 而存在于 $\mathcal{P}$ 中。类似地，对于某个 $w \in V$，与边 $v',\ w$ 相关的任何代价，已因边 $v,\ w$ 而存在于 $\mathcal{P}$ 中。因此，情形相同，新的解也同样是最优的。

我们重复此过程，直到类型为 ν 的中间顶点只有 V_ν^* 中的那些。此外，这不会影响任何其他类型。因此，我们可以对所有类型重复此过程。◇

断言4和5使我们无需猜测最优解中每种类型的哪些顶点会被用作中间顶点，因为我们可以（任意地）选定每种类型中将用作中间顶点的顶点。

在继续介绍算法之前，最后一步是证明：对于每种类型，没有作为中间顶点出现的终端对可以使用相同类型的路径。

断言6. 存在一个最优解 $\mathcal{P}$ ，使得对于每种终端类型 τ 和终端对(s,t)、$(s',t') \in \mathcal{T}_\tau \setminus \mathcal{T}_\tau^*$， s-t 路径与 s'-t' 路径在 $\mathcal{P}$ 中具有相同的路径类型。

断言的证明. 考虑一个满足断言5和断言4的最优解。考虑一种终端类型 τ和终端对 $(s,t),(s',t') \in \mathcal{T}_\tau \setminus \mathcal{T}_\tau^*$，其中 s-t 路径 P与 s'-t' 路径 P'在 $\mathcal{P}$中具有不同的路径类型。我们将证明，只需修改 P、 P'中的一个，就能使它们具有相同的类型。设 Q 和 Q' 分别是通过交换顶点 s 与 s'、以及 t 与 t'，从 P 和 P'得到的路径。注意， Q 是一条与 P类型相同的 s'-t' 路径，而 Q' 是一条与 P'类型相同的 s-t 路径。

我们声称，$(\mathcal{P} \setminus P) \cup Q'$ 和 $(\mathcal{P} \setminus P') \cup Q$ 都是最优解。为此，设 $e_1, \ldots, e_p$是出现在 P 中的边，且这些边也以相反方向出现在 $\mathcal{P}$的其他路径中。另外，设 $x_i, i \in [p]$表示边 e_i 在 $\mathcal{P}$ 的路径中以与其出现在 P 时相反的方向出现的次数。同理，设 $e'_1, \ldots, e'_q$是出现在 P'中的边，且这些边也以相反方向出现在 $\mathcal{P}$的其他路径中；设 $x'_j, j \in [q]$表示边 e'_j 在 $\mathcal{P}$ 的路径中以与其出现在 P'时相反的方向出现的次数。

注意，边 $e_i, i \in [p]$ 或 $e'_j, j \in [q]$ 均不与 s, s', t 或 t' 相接。这是因为这些顶点都不会作为任何路径的中间顶点，而且由于不存在重叠终端，它们只会出现在各自的路径中。

根据 Q 和 Q'的构造，以下两个不等式成立：

$$\mathrm{cost}(\mathcal{P}) \leq \mathrm{cost}((\mathcal{P} \setminus \{P\}) \cup \{Q'\}) \leq \mathrm{cost}(\mathcal{P}) - \sum_{i \in [p]} x_i + \sum_{j \in [q]} x'_j,$$

$$\mathrm{cost}(\mathcal{P}) \leq \mathrm{cost}((\mathcal{P} \setminus \{P'\}) \cup \{Q\}) \leq \mathrm{cost}(\mathcal{P}) - \sum_{j \in q} x'_j + \sum_{i \in [p]} x_i.$$

因此，我们可以得出结论： $\sum_{j \in [q]} x'_j = \sum_{i \in [p]} x_i$ 和全部三个解都是最优的。通过选取其中一对 $(s,\ t) \in \mathcal{T}_\tau \setminus \mathcal{T}_\tau^*$，我们可以重复上述过程，直到所有其他终端对都使用与 $(s,\ t)$ 相同路径类型的路径。这不会影响其他终端类型的终端所使用的路径类型，因此我们可以针对每种终端类型分别应用此过程。由此，断言成立。$\diamond$

现在，我们可以给出算法了。首先，对于所有类型的非终端顶点和终端对，我们分别固定集合 V_ν^*和 $\mathcal{T}_\tau^*$。考虑一种终端类型 τ。根据断言6，我们知道存在一个解，使得至多有 5 种不同类型的路径连接类型为 τ的终端对： $\mathcal{T}^*\tau$ 中的每一对各使用一种路径类型，其余终端对则使用一种路径类型。注意，我们可以在 FPT时间内猜出连接所有终端对的路径类型，因为路径类型的数量受 vc 的函数所界定。

猜出路径后，我们需要确定哪些是中间顶点。任意路径最多包含来自 I 的 $\mathrm{vc}+1$ 个顶点，而且我们知道所有此类顶点的类型。因此，对于每种猜测，最多有 $5\mathrm{vc}+5$ 个未知的中间顶点。对于非终端中间顶点，最多有 2 种可能的选项。因此，我们可以对它们进行猜测，最多需要 $2^{5\mathrm{vc}+5}$ 次猜测。对于终端顶点，在检查其类型后，我们需要确定哪种终端类型包含这种类型的顶点。由于每种类型最多有 4 个终端对，且顶点的类型取决于该顶点是源点还是汇点，因此最多有 4 种可能的猜测。因此，最多需要 $4^{5\mathrm{vc}+5}$ 次猜测。完成这些猜测后，我们就确定了连接任意一对类型为 τ 的终端的所有路径。

由于我们有 16^{vc} 许多不同类型的终端，因此可以为每种不同类型猜测这些路径。猜测之后，我们创建了固定参数可解数量的可行解。此外，由于我们进行了所有可能的猜测，因此至少创建了一个最优解。由于我们可以在多项式时间内计算一个可行解的代价，只需计算所创建的所有可行解的代价并保留最小值即可。 $\square$

5.2 数量庞大的智能体

本节将证明，如果几乎所有顶点对都作为终端对出现，那么我们就能判定是否存在代价为 0 的解。

定理13。 我们可以在以 $|\mathcal{N}|$ 为参数的固定参数可解时间内，判定一个 CC-MAR 实例 $(G, \mathcal{T})$ 是否存在一个代价为 0 的解，其中 $\mathcal{N} = (V \times V) \setminus \mathcal{T}$。

证明。 考虑一条边 $e = \{u, v\} \in E$。我们断言：任意解的代价要么至少为 1，要么 (u, v)、(v, u) 中至少有一个属于 $\mathcal{N}$。

假设 (u, v) 和 (v, u) 均不属于 $\mathcal{N}$。那么 (u, v)，$(v, u) \in \mathcal{T}$。由于我们可以假设 G 是无环的，因此不存在排除 $\{u, v\}$ 的 u-v 路径，也不存在排除 $\{u, v\}$ 的 v-u 路径。因此，两条路径都必须使用 $\{u, v\}$，不存在代价为 0 的解。注意，这一条件可以在多项式时间内验证。

下文中，我们假设对于任意 $e = \{u, v\} \in E$，(u, v) 和 (v, u) 中至少有一个属于 $\mathcal{N}$，因此 $|\mathcal{N}| \geq |E|$。根据定理 8，我们可以在以 $|E| \leq |\mathcal{N}|$ 为参数的 FPT时间内计算出最优解。 □

6 结论

我们提出了交叉成本多智能体路径规划（CC-MAR）模型，为异步多智能体路径规划提供了一个合理的框架：其中，迎面交互通过软成本函数来体现，而非作为硬约束。这一设定反映了现实场景：智能体在没有全局同步的情况下运行，且相向穿越边会导致拥塞。我们的分析结合了博弈论和算法视角。我们证明，纳什均衡总是存在，且可通过收敛动态达到；在温和假设下，这些均衡可在多项式时间内计算，而在一般情形下，问题为 PLS 完全。在优化方面，我们首先观察到该问题是 Steiner Orientation 问题的推广，因此是 NP 困难。为解决这一问题，我们针对多种结构参数开发了具有 FPT 和 XP 保证的参数化算法，包括弧数、边数、终端对数以及顶点覆盖数。

这一框架为未来研究开辟了多个方向。特别是，尽管我们针对涉及弧数和边数的参数给出了固定参数算法，但仅就 $|A|$ 而言的参数化复杂性仍未解决。进一步探索适用于实际实例的高效启发式算法或近似方案，仍是一项尚待解决且具有实际意义的挑战。

参考文献

[1] Heiner Ackermann、Heiko R¨oglin 和 Berthold V¨ocking. 组合结构对拥塞博弈的影响. 《*ACM* 杂志》, 55(6):25:1–25:22, 2008年.[2] Zain Alabedeen Ali 和 Konstantin S. Yakovlev. 改进的匿名多智能体寻路算法. 载于第三十八届 *AAAI* 人工智能大会、*AAAI 2024*、第三十六届人工智能创新应用会议、*IAAI 2024*、第十四届人工智能教育进展研讨会、*EAAI 2014*，*2024*年*2*月*20*日至*27*日，加拿大温哥华，第 17291–17298 页。AAAI出版社，2024年。[3] Esther M. Arkin 和 Refael Hassin. 关于混合图定向的一则注记. 《离散与应用数学》, 116(3):271–278, 2002.[4] Reuven Bar-Yehuda 和 Shimon Even. 加权顶点覆盖问题的线性时间近似算法. 《算法》, 2(2):198–203, 1981.[5] Rajesh Chitnis、Andreas Emil Feldmann 和 Ondrej Such´y. 平面斯坦纳定向的紧下界. 《算法》, 81(8):3200–3216, 2019.[6] Rajesh Chitnis、Andreas Emil Feldmann 和 Ondrej Such´y. 平面斯坦纳定向的紧下界 . 《算法》 , 81(8):3200–3216, 2019.

[7] Thomas H. Cormen、Charles E. Leiserson、Ronald L. Rivest 和 Clifford Stein. 算法导论. 麻省理工学院出版社，第 2 版，2001 年。[8]Marek Cygan 、 Fedor V. Fomin 、 ˙Lukasz Kowalik、Daniel Lokshtanov、D´aniel Marx、Marcin Pilipczuk、Micha˙l Pilipczuk 和 Saket Saurabh. 参数化算法. 施普林格，2015 年。[9] Marek Cygan、Guy Kortsarz 和 Zeev Nutov. 斯坦纳森林定向问题. SIAM 离散数学杂志，27(3):1503–1513，2013 年。[10] Argyrios Deligkas、Eduard Eiben、Robert Ganian、Iyad Kanj 和 M. S. Ramanujan. 用于协同运动规划的参数化算法：能量最小化. 载于 Karl Bringmann、Martin Grohe、Gabriele Puppis 和 Ola Svensson（编），第 51 届国际自动机、语言与编程研讨会，ICALP 2024，2024年7月8日至12日，爱沙尼亚塔林，第 297 卷 LIPIcs，第 53:1–53:18 页。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2024 年。[11] Argyrios Deligkas、Eduard Eiben、Robert Ganian、Iyad Kanj 和 M. S. Ramanujan. 树上的多智能体寻路参数化算法. 载于 Sanmay Das、Ann Now´e 和 Yevgeniy Vorobeychik（编），《第 24 届国际自主智能体与多智能体系统会议论文集》，AAMAS 2025，美国密歇根州底特律，2025年5月19日至23日，第 584–592 页。国际自主智能体与多智能体系统基金会 / 美国计算机协会，2025 年。[12] Reinhard Diestel. 图论，第 4 版，数学研究生教材第 173 卷. 施普林格，2012 年。[13] Rodney G Downey 和 Michael Ralph Fellows. 参数化复杂性. Springer Science & Business Media，2012 年。[14] Eduard Eiben、Robert Ganian、Iyad Kanj 和 M. S. Ramanujan. 使用小图测试方法进行含滑动机器人的协同运动规划. 载于 Oswin Aichholzer 和 Haitao Wang（编），第 41 届国际计算几何研讨会，SoCG 2025，2025年6月23日至27日，日本金泽，第 332 卷 LIPIcs，第 44:1–44:15 页。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2025 年。[15] Alex Fabrikant、Christos H. Papadimitriou 和 Kunal Talwar. 纯纳什均衡的复杂性. 载于 L´aszl´o Babai（编），《第 36 届 ACM 计算理论年会论文集》，美国伊利诺伊州芝加哥，2004年6月13日至16日，第 604–612 页。美国计算机协会，2004 年。[16] Foivos Fioravantes、Dusan Knop、Jan Maty´as Kristan、Nikolaos Melissinos 和 Michal Opler. 多智能体寻路的精确算法与下界：树状拓扑的优势. 载于 Michael J. Wooldridge、Jennifer G. Dy 和 Sriraam Natarajan（编），第三十八届 AAAI 人工智能大会，AAAI 2024，第三十六届人工智能创新应用会议，IAAI 2024，第十四届人工智能教育进展研讨会，EAAI 2014，2024年2月20日至27日，加拿大温哥华，第 17380–17388 页。AAAI出版社，2024 年。[17] Foivos Fioravantes、Dusan Knop、Jan Maty´as Kristan、Nikolaos Melissinos 和 Michal Opler. 面向树状结构且具有通信约束的多智能体寻路精确算法. 载于 Toby Walsh 、 Julie Shah 和 Zico Kolter （编），AAAI-25，由人工智能促进协会赞助，2025 年 2月25日至3月4日，美国宾夕法尼亚州费城，第 23177–23185 页。AAAI出版社，2025 年。[18] Foivos Fioravantes、Dusan Knop、Jan Maty´as Kristan、Nikolaos Melissinos、Michal Opler 和 Tung Anh Vu. 在高度中心化网络中求解多智能体寻路问题. 载于 Toby Walsh、Julie Shah 和 Zico Kolter（编），AAAI-25，由人工智能促进协会赞助，2025年2月25日至3月4日，美国宾夕法尼亚州费城，第 23186–23193 页。AAAI出版社，2025 年。[19] F.V. Fomin、D. Lokshtanov、S. Saurabh 和 M. Zehavi. 核化：参数化预处理理论. 剑桥大学出版社，2019 年。[20] Tesshu Hanaka、Michael Lampis、Nikolaos Melissinos、Edouard Nemery、Hirotaka Ono 和 Mano- lis Vasilakis. 斯坦纳定向的结构参数，2025 年。

[21] David G. Harris 和 N. S. Narayanaswamy。针对解大小参数化的顶点覆盖问题的更快算法。载于 Olaf Beyersdorff、Mamadou Moustapha Kant´e、Orna Kupferman 和 Daniel Lokshtanov（编），第*41*届国际计算机科学理论方面研讨会，*STACS 2024*，*2024*年*3*月*12*日至*14*日，法国克莱蒙费朗，LIPIcs 第289卷，第40:1–40:18页。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2024年。[22] Justin Kottinger、Tzvika Geft、Shaull Almagor、Oren Salzman 和 Morteza Lahijanian。在多智能体路径寻找中引入延迟。载于 Ariel Felner 和 Jiaoyang Li（编），第十七届国际组合搜索研讨会，*SOCS 2024*，*2024*年*6*月*6*日至*8*日，加拿大艾伯塔省卡纳纳斯基斯，第37–45页。AAAI出版社，2024年。[23] Jiaoyang Li、Andrew Tinka、Scott Kiesel、Joseph W. Durham、T. K. Satish Kumar 和 Sven Koenig。大型仓库中的终身多智能体路径规划。载于第三十五届*AAAI*人工智能会议，*AAAI 2021*，第三十三届人工智能创新应用大会，*IAAI 2021*，第十一届人工智能教育进展研讨会，*EAAI 2021*，线上活动，*2021*年*2*月*2*日至*9*日，第11272–11281页。AAAI出版社，2021年。[24] Hang Ma、Craig A. Tovey、Guni Sharon、T. K. Satish Kumar 和 Sven Koenig。带有载荷转运的多智能体路径寻找与包裹交换机器人路径规划问题。载于 Dale Schuurmans 和 Michael P. Wellman（编），第三十届*AAAI*人工智能大会论文集，*2016*年*2*月*12*日至*17*日，美国亚利桑那州菲尼克斯，第3166–3173页。AAAI出版社，2016年。[25] Keisuke Okumura。Lacam：用于快速多智能体路径寻找的基于搜索的算法。载于 Brian Williams、Yiling Chen 和 Jennifer Neville（编），第三十七届*AAAI*人工智能会议，*AAAI 2023*，第三十五届人工智能创新应用会议，*IAAI 2023*，第十三届人工智能教育进展研讨会，*EAAI 2023*，美国华盛顿特区，*2023*年*2*月*7*日至*14*日，第11655–11662页。AAAI出版社，2023年。

[26] Keisuke Okumura 和 S´ébastien Tixeuil。容错离线多智能体路径规划。载于 Brian Williams、Yiling Chen 和 Jennifer Neville 编，第三十七届 *AAAI* 人工智能会议（*AAAI 2023*）、第三十五届人工智能创新应用会议（*IAAI 2023*）、第十三届人工智能教育进展研讨会（*EAAI 2023*），美国华盛顿特区，*2023*年*2*月*7*日至*14*日，第 11647–11654 页。AAAI出版社，2023年。

[27] Robert W. Rosenthal。具有纯策略纳什均衡的一类博弈。《国际博弈论杂志》，2(1):65–67，1973年。

[28] Roni Stern、Nathan R. Sturtevant、Ariel Felner、Sven Koenig、Hang Ma、Thayne T. Walker、Jiaoyang Li、Dor Atzmon、Liron Cohen、T. K. Satish Kumar、Roman Bart´ak 和 Eli Boyarski。多智能体路径寻找：定义、变体和基准测试。载于 Pavel Surynek 和 William Yeoh 编，第十二届国际组合搜索研讨会论文集（*SOCS 2019*），加利福尼亚州纳帕，*2019*年*7*月*16*日至*17*日，第 151–158 页。AAAI出版社，2019年。

[29] Roni Stern、Nathan R. Sturtevant、Ariel Felner、Sven Koenig、Hang Ma、Thayne T. Walker、Jiaoyang Li、Dor Atzmon、Liron Cohen、T. K. Satish Kumar、Roman Bart´ak 和 Eli Boyarski。《多智能体路径寻找：定义、变体与基准测试》。载于 Pavel Surynek 和 William Yeoh 编，《第十二届国际组合搜索研讨会论文集，*SOCS 2019*，加利福尼亚州纳帕，*2019*年*7*月*16*日至*17*日》，第151–158页。AAAI出版社，2019年。

[30] Davide Tateo、Jacopo Banfi、Alessandro Riva、Francesco Amigoni 和 Andrea Bonarini。《多智能体连通路径规划：PSPACE 完全性及其应对方法》。载于 Sheila A. McIlraith 和 Kilian Q. Weinberger 编，第三十二届 *AAAI* 人工智能会议论文集（*AAAI-18*）、第三十届人工智能创新应用会议（*IAAI-18*）和第八届 *AAAI* 人工智能教育进展研讨会（*EAAI-18*），美国路易斯安那州新奥尔良，*2018*年*2*月*2*日至*7*日，第4735–4742页。AAAI出版社，2018年。

[31] Peter R.Wurman、Raffaello D'Andrea 和 Mick Mountz。在仓库中协调数百辆协作型自动驾驶车辆。《*AI Mag.*》，29(1)：9–20，2008年。

[32] Jingtian Yan 和 Jiaoyang Li。通过静止状态搜索实现差速驱动机器人的多智能体运动规划。载于 Toby Walsh、Julie Shah 和 Zico Kolter（编），*AAAI-25*，由人工智能促进协会主办，*2025*年*2*月*25*日至*3*月*4*日，美国宾夕法尼亚州费城，第23360–23368页。AAAI出版社，2025年。

[33] Shuai Zhou、Shizhe Zhao和Zhongqiang Ren。面向具有异步动作的多智能体路径规划的松散同步的基于规则规划。载于Toby Walsh、Julie Shah和Zico Kolter主编，*AAAI-25*，由人工智能促进协会主办，*2025*年*2*月*25*日至*3*月*4*日，美国宾夕法尼亚州费城，第14763–14770页。AAAI出版社，2025。