相互依赖拍卖的复杂性

Patrick Loiseau 法国国家国家信息与自动化研究所，FairPlay团队
patrick.loiseau@inria.fr

Simon Mauras Minrui Xu，法国国家信息与自动化研究所，FairPlay团队 ENSAE，FairPlay团队
simon.mauras@inria.fr minrui.xu@ensae.fr

arXiv:2603.18668v1 [cs.GT] 19 Mar 2026

摘要

我们研究由 Milgrom 和 Weber[1982], 提出的著名相互依赖模型中的拍卖设计。在该模型中，机制设计者分配一件物品，使获得物品的代理人价值最大化，同时通过支付促使其保持真实性。在较少受到研究的采购拍卖中，机制设计者分配一项任务，并使被选中执行该任务的代理人所承担的成本最小化。

过去的大多数理论计算机科学文献都关注如何在受限域和单调估值函数的条件下，为价值设定设计具有常数近似保证的真实机制。

在本研究中，我们探讨确定性和随机化设定下，针对价值与成本优化真实机制近似比的一般计算问题。不同于大多数以往研究，我们去除了对价值函数施加的定义域限制和单调性假设。我们从理论上解释了为何某些以往研究过的特殊情形易于求解：可将其归约为经典组合问题，并据此给出高效算法和刻画。除了正面结果，我们还针对一般情形给出困难性结果，提供查询复杂度下界，并证明一般情形具有 NP-困难性。

1 引言

算法机制设计。 算法机制设计处于计算机科学与经济学的交叉领域，关注开发能够考虑自利型代理人策略行为的算法 [NR99; Nis+07]。该领域的一个核心问题是设计拍卖机制，从 n 代理人中选出最合适的竞标者。值得注意的是，在第二价格拍卖（或维克里拍卖）中，胜者支付第二高的价值；对于具有独立估值 的竞标者，如实报价是最优策略 [Vic61]。然而，在更复杂的情形中，如果竞标者的估值取决于他人掌握的信息，第二价格拍卖便无法保持真实性，这种情形通常称为相互依赖价值 [MW82], 形式化地说，每个代理人 $i \in [n]$ 拥有一个私人信号 $s_i \in [k]$，以及一个公开已知的函数 $v_i : [k]^n \to \mathrm{R}_{>0}$ ，该函数将这些信号汇总为一个价值。

举例来说，考虑这样一种情形： $n = 2$ 代理人 Alice 和 Bob 正在竞购一件物品，比如乡村里的一栋小房子。Alice 对这栋房子的价值高度取决于房屋状况，但相关信息只有 Bob 知道；而 Bob 对房子的价值一般，因为他打算将其彻底重建。令 $k = 2$，并将价值函数定义为 $v_1(s_1,\ s_2) = 1 + 99(s_2 - 1)$ 和 $v_2(s_1,\ s_2) = 10$，也就是说，如果 Bob 的信号较高（$s_2 = 2$），Alice 对房子的价值为 100，否则为 1（$s_2 = 1$）；而无论信号如何，Bob 的价值始终为 10。如果 Bob 如实报告较高的信号（$s_2 = 2$），第二价格拍卖会以 10 的价格将房子分配给 Alice。然而，值得注意的是，Bob 可以通过谎报较低的信号（$s_2 = 1$）赢得拍卖。一种真实的替代方案是抛一枚公平硬币，随机分配房子。尽管这种方法并非最优，但它能保证胜者的期望价值至少达到最优值的一半，近似比[1] 为 2。

相互依赖拍卖。 近期一系列研究探讨了相互依赖拍卖的设计 [RT16；Ede+18；Ede+19；AT21；LSZ22], ，并在受限的估值函数域上实现了常数因子的近似比。特别地，对于次模[2]估值函数，最优近似比至少为 2 [Ede+19]，至多为 3.315 [LSZ22], ；而在$n = 2$ [Ede+18] 或 $k = 2$ [AT21]的特殊情形下，该比率等于 2。有趣的是，这一领域的文献聚焦于物品拍卖，而非反向拍卖（也称为采购拍卖）；后者分配任务，即需要完成的任务或服务，并通过支付来补偿中标承包商的成本。用于保证真实性的大部分基础经济理论，在标准拍卖和反向拍卖中是相同的 [Kri09], ，只是在使用成本函数时有所不同 $c_i : [k]^n \to \mathrm{R}_{>0}$。

回到我们的例子，假设 Alice 和 Bob 现在是承包商，竞标翻修乡村住宅，并假设他们的成本分别为：Alice 将住宅从当前状态翻修时的 $c_1(s_1,\ s_2) = 1 + 99(2 - s_2)$；Bob 将住宅拆除后重建时的 $c_2(s_1,\ s_2) = 10$ 。和之前一样，反向第二价格拍卖容易受到操纵，因为 Bob 可以报告 $s_2 = 1$ 而不是 $s_2 = 2$ ，从而以 100 的付款获得合同。但与价值设定不同，在这里，均匀随机分配并不可取。事实上，如果 $s_2 = 2$，最优成本为 $c_1(s_1,\ 2) = 1$，而随机分配的预期成本为 $(c_1(s_1,\ 2) + c_2(s_1,\ 2))/2 = 5.5$。调整参数时，这会导致近似比任意大。

主要贡献。

- 首先，从我们的简单示例可以看出，从近似的角度来看，任务设定更具挑战性。直观而言，大多数机制 [Ede+19]以常数概率表现良好，这足以在价值设定中保证常数近似，但在成本设定中却不够。本文开启了对相互依赖情形下近似采购拍卖的研究。其次，请注意，我们的简单示例采用单调价值函数和成本函数，因为信号表示有关物品或任务的定量信息；这是上述所有先前研究中采用的标准假设。然而，在许多实际场景中，价值函数或成本函数并非单调，例如代理人具有相反偏好，或者信号无法比较，因为它们反映的是定性信息。本文将 [RT16] 对真实机制的刻画推广至非单调设定。

[1]随机机制的近似比，是最坏情况下（对所有可能的信号取最坏情形）最优价值/成本与机制选定的价值/成本期望之间的比率。[2]关于信号的子模性的定义见 [Ede+19]。

- 先前的研究 [Ede+18; Ede+19; AT21; LSZ22] 主要致力于在价值函数受限的域中设计用于分配物品的贪心常数因子近似机制，而我们研究的是如何计算能够达到最佳（即最小）近似比的机制这一优化问题。具体而言，我们在三种不同设定下研究该问题：物品分配中的随机机制、任务分配中的随机机制，以及确定性机制。我们提出了若干框架，将优化问题转化为经典组合问题，或从经典组合问题归约而来。这些框架有助于更好地理解一般域，也可能改进上述受限域中的贪心机制。

1.1 我们的结果

我们定义优化问题 VAL 和 CST，其目标分别是在价值设定和成本设定下，计算能够达到最小（即最优）近似比的随机机制。此外，我们还考虑确定性机制；对于这类机制，价值和成本方面的近似是相同的，并将相应的优化问题记为 DET。为了度量求解我们的最小化问题的算法的效率，我们根据输入的总大小 $N = nk^n b$ 来度量时间复杂度，其中 b 表示表示一个价值或成本所使用的最大比特数。

定理 1。 当 $n = 2$时，可以在 $O(N \log N)$ 时间内求解 VAL、 CST 和 DET 。

定理2。 当 $k = 2$时，可以在 $N^{1+o(1)}$ 时间内求解 DET 。

定理 3。 可以在 $N^{O(1)}$ 时间内求解 VAL 和 CST 。

接下来，为了证明困难性，我们定义优化问题的间隙变体：给定一个最小化问题 PB 和两个常数 $0 \le \alpha \le \beta$，间隙变体（α，β）-PB 是一个判定问题，其要求区分最优解的度量为 $\le \alpha$ 和 $> \beta$ 的实例（算法在中间实例上可以失败或不终止）。

定理 4。 当 $n = 4$ 时，对于任意 $\beta > 1$，问题 $(1, \beta)$-DET 是 *NP* 困难的。

然而，在机制设计中，我们主要关注的是计算机制在报告信号配置下的结果。因此，对于每个最小化问题 PB，我们引入查询问题 PB_γ：在存在某个度量为 $\le \gamma$ 的解的前提下，回答关于该解的查询（否则算法可以失败或不终止），同时确保各次查询之间保持一致。在我们的语境中，一次查询对应于计算一个信号配置下的结果。我们得出定理 1 至 3 的以下推论。

推论。 对于每个 $\gamma > 1$，并以 N 度量复杂度，我们有：

- 当 $n = 2$ 或 $k = 2$ 时，可以在线性时间内回答 DET_γ 个查询，
- 可以在多项式时间内回答 VAL_γ 和 CST_γ 个查询，
- 假设 $P \neq NP$，则不可能始终在多项式时间内回答 DET_γ 个查询，即使是在存在比率为 1 的确定性分配规则的实例上。

最后，人们或许会想，我们得到的复杂度是否可以改进，因为回答一次查询所需的时间可能少于计算整个分配规则所需的时间。然而，我们通过让输入可经由预言机访问，并统计完成某项计算任务所必需的预言机查询次数，证明了对 n 的指数依赖不可避免。

定理 5. 对于所有固定的 $n \ge 2$ 和 $k \ge 2$，要回答某些 VAL_γ、 CST_γ 和 DET_γ 查询，至少需要向输入预言机（价值或成本）进行 $\Omega(k^n)$ 次查询。

	时间复杂度	查询复杂度
$n = 2$	定理 1： VAL， CST， DET 属于 $O(N \log N)$	定理 5： VAL ，CST, DET 需要 $\Omega(k^n)$ 次查询
$k = 2$	定理2： DET 属于 $N^{1+o(1)}$	
一般情形 case	定理 3： VAL 和 CST 在 $N^{O(1)}$ 定理 4： DET是 NP 困难	

表1：主要结果总结。时间复杂度以输入总规模 N 的函数形式给出，查询复杂度则是值或成本预言机查询的次数。

1.2 相关工作

相互依赖价值模型源自对矿产权拍卖等情境的建模；在这类情境中，竞标者拥有共同价值 [Wil69; Mil79; DM00], ，该模型由 Milgrom 和 Weber [MW82] 加以形式化。经济学家对此模型进行了广泛研究，并利用“单交叉”条件刻画了这样一类估值函数：在这类函数下，将物品分配给价值最高者可以通过真实机制实现[Mas96; Aus+99; Aus+99;JM01]。

具有相互依赖价值的拍卖近期受到理论计算机科学界的关注 [CFK14; RT16], ，研究内容是在单调交叉假设之外，考察如何近似最大化收入（向卖方支付的款项）和社会福利（买方的价值）。在开创性工作中，Roughgarden 和 Talgam-Cohen [RT16] 将 Myerson引理 [Mye81]推广后，用它刻画了代理人具有递增价值函数时可真实实现的分配规则。在引理2.2中，我们将他们的刻画推广到非单调价值函数和成本函数。此前唯一处理非单调价值函数的工作是 [EGZ22], ，但他们只给出了充分条件，因此可能遗漏了某些不满足该条件的真实分配规则。

与我们工作最接近的是近期一系列研究，它们针对属于受限类别的相互依赖价值代理人，设计了常数近似机制，例如“c-单交叉” [Ede+18]或“信号上的子模” [Ede+19；AT21；LSZ22]。特别是，Eden、Feldman、Fiat 和 Goldner[Ede+18] 研究了 DET 的特殊情形：代理人数量为 $n = 2$ 或信号数量为 $k = 2$ ，且价值函数单调。他们提出了 c-单交叉条件，该条件足以保证存在c-近似确定性分配规则。相比之下，定理 1 和定理 2 考虑相同的特殊情形，并构造了达到最佳可能近似比的确定性分配规则，其计算复杂度相当；这些规则不受限于 c-单交叉域，也不要求单调性。此外，在 $n = 2$情形下，我们精确刻画了能够实现 c-近似的价值函数；如命题 4.14 所示，这一刻画严格推广了c-单交叉条件（该条件仍作为特殊情形充分成立）。最后，关于 VAL的既有研究在价值函数单调且为信号上的子模时，给出了随机分配规则，并证明一般情形下最优近似比至多为 3。315 [LSZ22], ；当 $k = 2$ [AT21],时至多为 2；而在 $n = k = 2$ [Ede+19]中，最坏情况下恰为 2。

后续研究考察了简单的拍卖形式，例如时钟拍卖 [Gka+21; Fel+22]，以及更一般的情形，例如在线拍卖 [MMR24; Fel+25]，或具有私人估值函数的情形 [EGZ22; Ede+23; Ede+24]。最后，本文与研究真实机制的计算复杂度和通信复杂度的工作有相似之处 [Dob11; DV12; DD13; DDT14; Dob16; Ass+20; RZ21; BDR23]。

1.3 组织结构

本文其余部分安排如下：第 2 节介绍模型并定义重要符号，第 3 节概述我们技术贡献的主要思路，第 4 节给出正面结果的形式化证明，第 5 节给出负面结果的形式化证明。

2 模型与预备知识

我们考虑这样一种拍卖情形： n 代理人相互竞争，以求被选中，或获得一件物品（价值最大化），或执行一项任务（成本最小化）。记作 $[\ell] := \{1, \dots, \ell\}$，每位代理人 $i \in [n]$ 都有一个（私有）信号 $s_i \in [k]$，它反映了她对自己被选中的情况所掌握的信息。为方便起见，我们将 $\mathbf{s} := (s_i)_{i \in [n]}$ 和 $\mathbf{s}_{-j} := (s_i)_{i \in [n] \setminus \{j\}}$ 记作。

分配规则。 随机化分配规则是一组函数 $\mathbf{x} := (x_i)_{i \in [n]}$，其中 $x_i : [k]^n \to [0,1]$ 给出代理人 i 被选中的概率。我们施加如下约束：对于信号的所有实现，概率之和必须为一：

$$\forall \mathbf{s} \in [k]^n, \qquad \sum_{i \in [n]} x_i(\mathbf{s}) = 1. \tag{1}$$

当所有 x_i 的值都属于 $\{0,1\}$ 时，分配规则 $\mathbf{x}$ 是确定性的。

价值与成本。 每个代理人都有一个（公开已知的）函数，该函数将所有代理人的信息汇总为一个参数：

- 分配物品时，每个代理人 $i \in [n]$ 都有一个价值函数 $v_i : [k]^n \to \mathbf{R}_{>0}$；
- 分配任务时，每个代理人 $i \in [n]$ 都有一个成本函数 $c_i : [k]^n \to \mathbf{R}_{>0}$。

大多数先前的研究 [Ede+18; Ede+19; AT21] 都聚焦于非递减价值函数，即 $\forall i,\ j \in [n],\ \mathbf{s}_{-j} \in [k]^{n-1}$，以及 $\forall s_j \in [k-1],\ v_i(s_j,\ \mathbf{s}_{-j}) \le v_i(s_j+1,\ \mathbf{s}_{-j})$。类似地，我们考虑非递增成本函数，其定义为 $c_i(s_j,\ \mathbf{s}_{-j}) \ge c_i(s_j+1,\ \mathbf{s}_{-j})$。

性能比与信号序。 为统一物品设定和任务设定，我们将假设（不失一般性）给定的价值函数和成本函数满足：

$$\forall i \in [n], \forall \mathbf{s} \in [k]^n, \qquad v_i(\mathbf{s}) \cdot c_i(\mathbf{s}) = 1.$$

稍微滥用一下记号，在任务设定中我们可能使用 $\mathbf{v}$，此时成本应计算为价值的倒数。接下来，为了抽象掉具体设定，我们计算两个量：性能比和信号序。性能比是归一化的价值与成本，将用于计算分配的性能：

$$\forall i \in [n], \forall \mathbf{s} \in [k]^n, \qquad \rho_i(\mathbf{s}) := \frac{v_i(\mathbf{s})}{\max_{j \in [n]} v_j(\mathbf{s})} = \frac{\min_{j \in [n]} c_j(\mathbf{s})}{c_i(\mathbf{s})} \in (0,1]. \tag{$\boldsymbol{\rho}$}$$

在下文中，我们将关注点从价值函数（相应地，成本函数）转向性能比率，并通过以下引理说明，这恰好是对价值设定（相应地，成本设定）的重新表述。

引理 2.1。 给定一组比率 $\boldsymbol{\rho} = (\rho_i(\mathbf{s}))_{i \in [n], \mathbf{s} \in [k]^n}$，其中对于所有信号配置 $\mathbf{s}$ 和竞标者 i，$\rho_i(\mathbf{s}) \in (0,1]$，，并且对于每个信号配置 $\mathbf{s}$，至少存在一个竞标者 i 满足 $\rho_i(\mathbf{s}) = 1$，则存在能够诱导出给定比率的递增价值函数（相应地，递减成本函数）。

证明。 我们将 $\boldsymbol{\rho}$ 的最小价值记为 $r^*(\boldsymbol{\rho})$。容易验证，以下构造满足所有要求：

- 对于价值设定， $v_i(\mathbf{s}) = \rho_i(\mathbf{s}) / (r^*(\boldsymbol{\rho})/2)^\ell$，其中 $\ell = ||\mathbf{s}||_1$；
- 对于成本设定， $c_i(\mathbf{s}) = (r^*(\boldsymbol{\rho})/2)^\ell / \rho_i(\mathbf{s})$，其中 $\ell = ||\mathbf{s}||_1$。

□

接下来，信号排序将使我们能够刻画真实性，而无需假设估值函数是单调的；大多数先前的研究都采用了这一假设 [RT16；Ede+18；Ede+19；AT21]。对于每个代理人 $i \in [n]$ 及信号 $\mathbf{s}_{-i} \in [k]^{n-1}$，我们通过按价值非递减（偏）序或按成本非递增（偏）序排序，在 $s_i \in [k]$ 上构造一个二元关系 $\sigma_i(\mathbf{s}_{-i})$。

$$\forall i \in [n], \forall \mathbf{s}_{-i} \in [k]^{n-1}, \qquad \begin{aligned} \sigma_i(\mathbf{s}_{-i}) &:= \{(s_i, s_i') \in [k]^2 \mid v_i(s_i, \mathbf{s}_{-i}) < v_i(s_i', \mathbf{s}_{-i})\} \\ &= \{(s_i, s_i') \in [k]^2 \mid c_i(s_i, \mathbf{s}_{-i}) > c_i(s_i', \mathbf{s}_{-i})\}. \end{aligned} \tag{$\boldsymbol{\sigma}$}$$

我们观察到，对于所有 i 和 $\mathbf{s}_{-i}$，二元关系 $\sigma_i(\mathbf{s}_{-i})$ 都是严格序：对于所有 s_i 和 s_i'，要么 $(s_i, s_i') \notin \sigma_i(\mathbf{s}_{-i})$，要么 $(s_i', s_i) \notin \sigma_i(\mathbf{s}_{-i})$。然而，这个序未必是全序：可能存在 s_i 和 s_i'，使得 $(s_i, s_i') \notin \sigma_i(\mathbf{s}_{-i})$ 且 $(s_i', s_i) \notin \sigma_i(\mathbf{s}_{-i})$；当 $v_i(s_i, \mathbf{s}_{-i}) = v_i(s_i', \mathbf{s}_{-i})$ 或 $c_i(s_i, \mathbf{s}_{-i}) = c_i(s_i', \mathbf{s}_{-i})$ 时，就会出现这种情况。

2.1 真实性

每个代理人 $i \in [n]$ 都会提交一个报价 $b_i \in [k]$，该报价可能与其私人信号 s_i不同。为了激励代理人如实报告其信号，我们设计了机制，即配备支付函数 $\mathbf{p} := (p_i)^{i \in [n]}$ 的分配规则，其中 $p_i : [k]^n \to \mathbb{R}_{\geq 0}$；根据具体情形（物品或任务），这些支付会向代理人收取或转给代理人。我们假设代理人是理性的，并且会采取行动以最大化其拟线性效用：

- 分配一件物品时，每个代理人 $i \in [n]$ 的效用为 $u_i(\mathbf{b};\mathbf{s}) := x_i(\mathbf{b}) \cdot v_i(\mathbf{s}) - p_i(\mathbf{b})$;
- 分配一项任务时，每个代理人 $i \in [n]$ 的效用为 $u_i(\mathbf{b};\mathbf{s}) := p_i(\mathbf{b}) - x_i(\mathbf{b}) \cdot c_i(\mathbf{s})$.

如果报告真实信号是纳什均衡，则称机制为真实的 （也称为 EPIC，即事后激励相容），也就是说，对每个代理人而言，i 报告 $b_i = s_i$ 都是最佳响应，前提是其他所有代理人都报告$\mathbf{b}_{-i} = \mathbf{s}_{-i}$。更正式地说：

$$\forall \mathbf{s} \in [k]^n, \forall i \in [n], \forall b_i \in [k], \qquad u_i(s_i, \mathbf{s}_{-i}; \mathbf{s}) \geq u_i(b_i, \mathbf{s}_{-i}; \mathbf{s}) \tag{IC}$$

$$\forall \mathbf{s} \in [k]^n, \forall i \in [n], \qquad u_i(s_i, \mathbf{s}_{-i}; \mathbf{s}) \geq 0 \tag{IR}$$

先前的研究 [RT16] 刻画了在价值函数随信号非递减时（分配物品时）可实现期望意义下真实的分配规则。我们进一步细化了这一刻画，使其适用于更一般的情形，包括非单调价值函数或成本函数。

引理2.2（改编自 [RT16]）。 当且仅当分配规则$\mathbf{x}$ 满足以下单调性时，它才能通过真实机制实现：

$$\forall i \in [n], \forall \mathbf{s}_{-i} \in [k]^{n-1}, \forall (s_i, s_i') \in \sigma_i(\mathbf{s}_{-i}), \qquad x_i(s_i, \mathbf{s}_{-i}) \leq x_i(s_i', \mathbf{s}_{-i}). \tag{2}$$

证明。给定这样的分配规则 $\mathbf{x}$，对于每个代理人 $i \in [n]$ 和 $\mathbf{s}_{-i} \in [k]^{n-1}$，我们将偏序$\sigma_i(\mathbf{s}_{-i})$ 扩展为全序 $\tau_i(\mathbf{s}_{-i})$，使其相对于 $x_i(\cdot, \mathbf{s}_{-i})$ 单调；也就是说，对于所有 $(s_i, s_i') \in \tau_i(\mathbf{s}_{-i})$，都有 $x_i(s_i, \mathbf{s}_{-i}) \leq x_i(s_i', \mathbf{s}_{-i})$。根据 $\sigma_i(\mathbf{s}_{-i})$ 的定义可知，对于所有 $(s_i, s_i') \in \tau_i(\mathbf{s}_{-i})$，我们也有

$$\begin{aligned} v_i(s_i, \mathbf{s}_{-i}) &\leq v_i(s_i', \mathbf{s}_{-i}) && \text{（物品），} \\ c_i(s_i, \mathbf{s}_{-i}) &\geq c_i(s_i', \mathbf{s}_{-i}) && \text{(chore).} \end{aligned}$$

为方便起见，给定 $\mathbf{s} \in [k]^n$，我们定义在 $\tau_i(\mathbf{s}_{-i})$ 中排序先于 s_i 的信号集合

$$\forall i \in [n], \forall \mathbf{s} \in [k]^n, \qquad T_i(\mathbf{s}) := \{s_i\} \cup \{t \in [k] \mid (t, s_i) \in \tau_i(\mathbf{s}_{-i})\}$$

接下来，我们定义分配概率的增幅：

$$\forall i \in [n], \forall \mathbf{s} \in [k]^n, \quad \delta x_i(\mathbf{s}) := x_i(\mathbf{s}) - \max(\{0\} \cup \{x_i(t, \mathbf{s}_{-i}) \mid t \in T_i(\mathbf{s}) \setminus \{s_i\}\})$$

具体而言，注意到我们有

$$\forall i \in [n], \forall \mathbf{s} \in [k]^n, \qquad x_i(\mathbf{s}) = \sum_{t \in T_i(\mathbf{s})} \delta x_i(t, \mathbf{s}_{-i})$$

最后，我们定义支付函数：

$$p_i(\mathbf{b}) := \sum_{t \in T_i(\mathbf{b})} \delta x_i(t, \mathbf{b}_{-i}) \cdot v_i(t, \mathbf{b}_{-i}) \qquad \text{(good)},$$

$$p_i(\mathbf{b}) := \sum_{t \in T_i(\mathbf{b})} \delta x_i(t, \mathbf{b}_{-i}) \cdot c_i(t, \mathbf{b}_{-i}) \qquad \text{(chore)}.$$

为验证（IC）性质和（IR）性质成立，我们计算效用 $u_i(b_i, \mathbf{s}_{-i}; \mathbf{s})$。

$$u_i(b_i, \mathbf{s}_{-i}; \mathbf{s}) = \sum_{t \in T_i(b_i, \mathbf{s}_{-i})} \delta x_i(t, \mathbf{s}_{-i}) \cdot (v_i(\mathbf{s}) - v_i(t, \mathbf{s}_{-i})) \qquad \text{(good)},$$

$$u_i(b_i, \mathbf{s}_{-i}; \mathbf{s}) = \sum_{t \in T_i(b_i, \mathbf{s}_{-i})} \delta x_i(t, \mathbf{s}_{-i}) \cdot (c_i(t, \mathbf{s}_{-i}) - c_i(\mathbf{s})) \qquad \text{(chore)}.$$

注意，报价 b_i 只会影响集合 $T_i(b_i, \mathbf{s}_{-i})$，我们在该集合上计算总和，但被加项不依赖于 b_i。此外，每个非零项 $t \in T_i(b_i, \mathbf{s}_{-i})$ 为正，当且仅当 $t \in T_i(s_i, \mathbf{s}_{-i})$，因此 $u_i(b_i, \mathbf{s}_{-i}; \mathbf{s})$ 非负，并在 $b_i = s_i$ 时达到最大值，从而证明（IR）和（IC）。

现在我们证明充分性方向。假设（IC）性质成立。根据定义，对于所有 $s_i,\ s_i' \in [k]$，我们有以下不等式

$$u_i(s_i, \mathbf{s}_{-i}; s_i, \mathbf{s}_{-i}) \geq u_i(s_i', \mathbf{s}_{-i}; s_i, \mathbf{s}_{-i})$$
$$u_i(s_i', \mathbf{s}_{-i}; s_i', \mathbf{s}_{-i}) \geq u_i(s_i, \mathbf{s}_{-i}; s_i', \mathbf{s}_{-i})$$

将这些不等式相加并重新整理各项，可得

$$(x_i(s_i, \mathbf{s}_{-i}) - x_i(s_i', \mathbf{s}_{-i})) \cdot \big(v_i(s_i, \mathbf{s}_{-i}) - v_i(s_i', \mathbf{s}_{-i})\big) \geq 0 \qquad \text{(good)},$$

$$(x_i(s_i, \mathbf{s}_{-i}) - x_i(s_i', \mathbf{s}_{-i})) \cdot \big(c_i(s_i, \mathbf{s}_{-i}) - c_i(s_i', \mathbf{s}_{-i})\big) \leq 0 \qquad \text{(chore)}.$$

这意味着单调性。 □

回顾一下，信号上的（偏）序 $\boldsymbol{\sigma}$ 是根据价值函数或成本函数构造的。在本文余下部分，我们将不讨论支付函数和代理人的效用，而是专注于寻找属于以下多胞体的分配规则。

定义 2.3. 将真实多面体 $\mathcal{T}(\boldsymbol{\sigma})$ 定义为所有单调分配规则组成的集合：

$$\mathcal{T}(\boldsymbol{\sigma}) := \left\{ (x_i(\mathbf{s}))_{i \in [n]}^{\mathbf{s} \in [k]^n} \;\middle|\; \begin{array}{ll} x_i(\mathbf{s}) \geq 0 & \forall i \in [n], \forall \mathbf{s} \in [k]^n \\ \sum_{i \in [n]} x_i(\mathbf{s}) = 1 & \forall \mathbf{s} \in [k]^n \\ x_i(s_i, \mathbf{s}_{-i}) \leq x_i(s_i', \mathbf{s}_{-i}) & \forall i \in [n], \forall \mathbf{s}_{-i} \in [k]^{n-1}, \forall (s_i, s_i') \in \sigma_i(\mathbf{s}_{-i}) \end{array} \right\}.$$

我们进一步定义所有确定性单调分配的集合：

$$\mathcal{T}_D(\boldsymbol{\sigma}) := \{\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma}) \mid x_i(\mathbf{s}) \in \{0, 1\}, \forall i \in [n], \mathbf{s} \in [k]^n\},$$

即，.....的子集　对真实多面体施加整数性约束后得到的可行整数点子集　　.

请注意，我们针对随机机制定义的真实性概念，是对机制的随机性取期望而言的：在均衡时，如实报告是使代理人期望效用最大化的最佳响应。然而，能够获知分配规则内部随机性的代理人可能非常了解情况，他们会知道自己是否会被选中，因此如实报告信号对他们而言未必是最优选择。更强的概念是普遍真实性：随机机制是由真实的确定性机制构成的彩票，即使代理人能够获知分配规则的内部随机性，也无法操纵该机制。形式上，普遍真实机制的集合是$\mathcal{T}_D(\boldsymbol{\sigma})$的凸包；当且仅当 $\mathcal{T}(\boldsymbol{\sigma})$的极点为整数时，这两种真实性概念才一致。

2.2 近似比

我们的目标是将物品或任务分配给最值得获得它的代理人（价值最高或成本最低）。然而，这未必总是可行的，因为最优分配可能不满足引理2.2中的单调性条件，因此无法真实地实现。我们将（真实）机制的效率度量为其表现与最优（非真实）解之间的最坏比率。在分配物品 时，近似比为

$$R_V(\boldsymbol{\rho}, \mathbf{x}) := \max_{\mathbf{s}\in[k]^n} 1/\langle \mathbf{x}(\mathbf{s}), \boldsymbol{\rho}(\mathbf{s})\rangle = \max_{\mathbf{s}\in[k]^n} \frac{\max_{i\in[n]} v_i(\mathbf{s})}{\sum_{i\in[n]} x_i(\mathbf{s})\cdot v_i(\mathbf{s})} \geq 1. \tag{R_V}$$

分配任务 时，近似比为

$$R_C(\boldsymbol{\rho}, \mathbf{x}) := \max_{\mathbf{s}\in[k]^n} \langle \mathbf{x}(\mathbf{s}), 1/\boldsymbol{\rho}(\mathbf{s})\rangle = \max_{\mathbf{s}\in[k]^n} \frac{\sum_{i\in[n]} x_i(\mathbf{s})\cdot c_i(\mathbf{s})}{\min_{i\in[n]} c_i(\mathbf{s})} \geq 1. \tag{R_C}$$

滥用符号时，在上下文能够明确实例的情况下，我们可能写作 $R_V(\mathbf{v},\mathbf{x})$、 $R_C(\mathbf{c},\mathbf{x})$ 和 $R^*_D(\mathbf{v})$，甚至写作 $R_V(\mathbf{x})$ 和 $R_C(\mathbf{x})$。我们通过以下引理发现，这些量密切相关。其主要直觉是，近似最小化比近似最大化更难，这一点可以利用詹森凸性不等式证明。

引理 2.4. 给定一个分配$\mathbf{x}$，有 $R_V(\boldsymbol{\rho},\mathbf{x}) \leq R_C(\boldsymbol{\rho},\mathbf{x})$；若$\mathbf{x}$ 是确定性的，则等号成立，此时我们简记为 $R(\boldsymbol{\rho},\mathbf{x})$。

证明。 根据 $x \mapsto 1/x$ 的凸性，并利用詹森不等式，该不等式成立。 □

给定性能比率 $\boldsymbol{\rho}$，我们定义价值、成本和确定性情形下的最优近似比：

$$R^*_V(\boldsymbol{\rho}, \boldsymbol{\sigma}) := \max_{\mathbf{x}\in\mathcal{T}(\boldsymbol{\sigma})} R_V(\boldsymbol{\rho}, \mathbf{x}) \tag{R^*_V}$$

$$R^*_C(\boldsymbol{\rho}, \boldsymbol{\sigma}) := \max_{\mathbf{x}\in\mathcal{T}(\boldsymbol{\sigma})} R_C(\boldsymbol{\rho}, \mathbf{x}) \tag{R^*_C}$$

$$R^*_D(\boldsymbol{\rho}, \boldsymbol{\sigma}) := \max_{\mathbf{x}\in\mathcal{T}_D(\boldsymbol{\sigma})} R(\boldsymbol{\rho}, \mathbf{x}) \tag{R^*_D}$$

滥用符号时，当上下文中的实例明确时，我们可能将 $R^*_V(\mathbf{v})$ 和 $R^*_C(\mathbf{c})$，甚至将 R^*_V、 R^*_C 和 R^*_D 这样书写。由引理 2.4，我们得到以下推论。

推论 2.5。 对于所有参数 $(\boldsymbol{\rho}、\ \boldsymbol{\sigma})$，都有 $1 \leq R^*_V(\boldsymbol{\rho}、\ \boldsymbol{\sigma}) \leq R^*_C(\boldsymbol{\rho}、\ \boldsymbol{\sigma}) \leq R^*_D(\boldsymbol{\rho}、\ \boldsymbol{\sigma})$。

2.3 问题与复杂度

优化问题。 我们主要研究的问题是优化问题。更正式地说，一个实例可以用两个参数 $(\boldsymbol{\rho},\ \boldsymbol{\sigma})$ 来描述，其中 $\boldsymbol{\sigma}$ 刻画可行解，而 $\boldsymbol{\rho}$ 导出测度函数。我们定义优化问题 Val 、 Cst 和 Det，它们分别要求计算一个分配规则 $\mathbf{x}$，使其达到上一节中定义的 $R^*_V(\boldsymbol{\rho},\ \boldsymbol{\sigma})$、 $R^*_C(\boldsymbol{\rho},\ \boldsymbol{\sigma})$ 和 $R^*_D(\boldsymbol{\rho},\ \boldsymbol{\sigma})$。

查询问题。 我们的优化问题输出分配规则，而这些规则的描述可能非常庞大。在机制设计中，我们主要关注在报告信号配置 $\mathbf{x}$处评估分配规则 $\mathbf{s}$。为了探索机制的计算复杂度，我们引入查询问题$\textsc{Val}_\gamma$, $\textsc{Cst}_\gamma$ 和 $\textsc{Det}_\gamma$：在对某个解 $\mathbf{x}$ 进行一致的跨查询回答时，回答有关该解的评估查询；该解的度量为 $\leq \gamma$，其中 $\gamma \geq 1$。

问题	输出	承诺
VAL	$\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$ 最小化 $R_V(\boldsymbol{\rho}, \mathbf{x})$	–
CST	$\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$ 最小化 $R_C(\boldsymbol{\rho}, \mathbf{x})$	–
DET	$\mathbf{x} \in \mathcal{T}_D(\boldsymbol{\sigma})$ 最小化 $R(\boldsymbol{\rho}, \mathbf{x})$	–
VAL_γ	针对$\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$、输入为 $R_V(\boldsymbol{\rho},\mathbf{x})$ 的预言机 $\le \gamma$	$R_V^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) \le \gamma$
CST_γ	针对$\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$、输入为 $R_C(\boldsymbol{\rho},\mathbf{x})$ 的预言机 $\le \gamma$	$R_C^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) \le \gamma$
DET_γ	针对$\mathbf{x} \in \mathcal{T}_D(\boldsymbol{\sigma})$、输入为 $R(\boldsymbol{\rho},\mathbf{x})$ 的预言机 $\le \gamma$	$R_D^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) \le \gamma$
(α,β)-VAL	$\mathbb{1}[R_V^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) \le \alpha]$	$R_V^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) \le \alpha$ 或 $R_V^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) > \beta$
(α, β)-CST	$\mathbb{1}[R_C^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) \le \alpha]$	$R_C^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) \le \alpha$ 或 $R_C^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) > \beta$
(α, β)-DET	$\mathbb{1}[R_D^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) \le \alpha]$	$R_D^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) \le \alpha$ 或 $R_D^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) > \beta$

表 2：我们考虑的计算问题总结。如果问题带有承诺，那么求解该问题的算法可以在不满足承诺的实例上失败或不终止。

间隙问题。 为了给出 NP-困难性结果，有必要处理判定问题。因此，我们定义间隙变体（α, β）-VAL、（α, β）-CST 和（α, β）-DET，它们询问最优解的度量是否至多为 α 或大于 β，其中 $1 \le \alpha \le \beta$，并承诺我们不处于中间情形（在这种情形下，算法可以失败或不终止）。当一个（α, β）-间隙问题是 NP 困难时，相应的优化问题就难以在因子$\le \beta/\alpha$内近似，因为任何近似算法都能区分 $\le \alpha$ 情形和 $> \beta$ 情形。对于我们的优化问题，近似算法这一说法可能令人困惑，因为目标函数本身就是近似比；不过，我们再次强调，求解计算任务的是算法，而它们的输出则是机制（分配规则和支付函数）。

时间复杂度。 在计算求解我们某个计算问题的算法的时间复杂度时，我们假设输入是描述成本函数或价值函数的比特串。因此，输入的大小为 $N = nk^n b$，其中 b 表示表示一个价值或成本所需的最大比特数。对 b 的依赖是必要的，因为我们的一些算法依赖于求解线性规划，而所有已知算法执行的算术运算次数都取决于 b [Kha79; Kar84; Tar86]。

查询复杂度。 为了给出我们计算问题复杂度的下界，我们考虑这样的情形：算法可以预言机访问价值函数和成本函数，并且可以执行任意多次算术运算。我们将查询复杂度度量为区分具有不同输出的若干实例所需的查询次数。这对应于代数决策树模型中的复杂度 [Ben83]。

3 我们技术的主要思路

本节重点介绍本文的主要技术贡献。为简化表述，我们只讨论价值设定和问题 DET。我们结果的完整证明，包括问题 VAL 和 CST，请见第4节和第5节。

一个实用的技巧是通过对最优近似比 $R_D^*(\mathbf{v})$进行二分搜索，将优化问题转化为其查询变体。

引理3.1. 给定 $\gamma \ge 1$，如果能在 $g(N)$ 时间内回答所有 DET_γ 查询，那么我们就能在 $O(g(N)\log N)$ 时间内求解优化问题 DET。

证明。 为了计算 $R_D^*(\mathbf{v})$及其对应的确定性机制，我们将运行二分搜索：对于所有 $\gamma \ge 1$，我们都可以通过计算所有 DET_γ 查询的答案，并检查由此得到的解$\mathbf{x}$ 是否正确，来判断 $R_D^*(\mathbf{v}) \le \gamma$。由于 $R_D^*(\mathbf{v})$必须等于某个 $i \in [n]$ 和 $\mathbf{s} \in [k]^n$对应的 $1/\rho_i(\mathbf{s})$，可能的取值至多有 nk^n 个，因此我们的二分搜索最多需要遍历 γ的 $O(\log\ N)$个取值。 □

3.1 归约到二分图匹配

首先考虑代理人拥有二元信号的情形，这一情形已在 [Ede+18; AT21] 中针对受限的价值函数定义域进行研究。我们提出一种新方法，将问题 $\textsc{Det}_\gamma$ 转化为在二分图中寻找匹配。有关这一构造的更多细节，请参见第4.1.4节和第4.2.3节。

为了构建图，我们首先需要在每个信号配置$\mathbf{s}$ 处定义可接受代理人的集合 $A_\gamma(\mathbf{s})$（即在给定比率 γ 的情况下可以被选中的代理人）以及受约束智能体的集合 $C(\mathbf{s})$（即其选择受到单调性条件约束的代理人）。

$$\forall \mathbf{s} \in [2]^n, \qquad C(\mathbf{s}) := \{i \in [n] \mid \exists s_i' \in [2], (s_i, s_i') \in \sigma_i(\mathbf{s}_{-i})\},$$
$$A_\gamma(\mathbf{s}) := \{i \in [n] \mid \rho_i(\mathbf{s}) \geq 1/\gamma\}.$$

信号配置分为两类。若信号配置$\mathbf{s}$ 满足 $A_\gamma(\mathbf{s}) \subseteq C(\mathbf{s})$，也就是说，所有可接受代理人都受到约束，则称其为必须匹配；否则，称其为可能匹配。必须匹配配置的集合记为 M_γ。我们按如下方式构造图 $G_\gamma = (V, E)$：

- 每个顶点代表一个信号配置，即 $V = [2]^n$，
- 对于所有 $i \in [n]$、$\mathbf{s} \in [2]^{n-1}$ 和 $s_i' \neq s_i$，若满足以下条件，则添加一条从$\mathbf{s}$ 到 $(s_i', \mathbf{s}_{-i})$ 的边：
  - $(s_i, s_i') \in \sigma_i(\mathbf{s}_{-i})$, and
  - $i \in A_\gamma(\mathbf{s}) \cap A_\gamma(s_i', \mathbf{s}_{-i})$, and
  - $\mathbf{s} \in M_\gamma$,

首先，G_γ 是二分图，因为可以根据$\sum_{i\in[n]} s_i$的奇偶性将顶点$\mathbf{s}$ 划分为两侧，同一侧的两个顶点之间不可能存在边。其次，请注意，一个顶点总是将一个必须匹配的顶点$\mathbf{s}$ 与一个可能匹配的顶点相连（s_i'，$\mathbf{s}_{-i}$），因为 $i \in A_\gamma(s_i', \mathbf{s}_{-i}) \setminus C(s_i', \mathbf{s}_{-i})$。

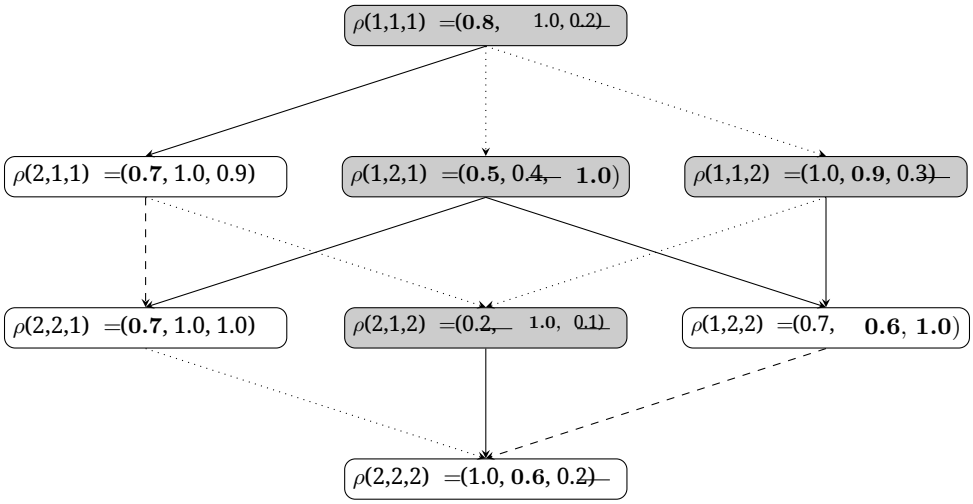


图 1：二分图 G_γ 及 $\gamma = 2$。首先，在每个顶点 $\mathbf{s}$ 处，我们划去不可接受的代理人$i \notin A_\gamma(\mathbf{s})$，其中 $\rho_i(\mathbf{s}) < 1/\gamma$。其次，我们绘制立方体的边，并使用 $\boldsymbol{\sigma}$ （在递增价值函数的特殊情形下）确定边的方向；如果对应的代理人在两个端点之一不可接受，则将边绘制为虚线。第三，我们将满足 $A_\gamma(\mathbf{s}) \subseteq C(\mathbf{s})$ 的必须匹配顶点 $\mathbf{s}$ 涂成灰色；对于这些顶点，所有可接受的代理人都对应于出边。最后，我们将不与必须匹配顶点关联的边画成虚线。最终图由实线边构成。

最后，我们证明，存在一个 γ-近似确定性真实机制，当且仅当 G_γ 中存在一个包含 $|M_\gamma|$ 条边的匹配；该匹配可通过计算一个最大基数匹配找到。

- 假设在 G_γ 中存在一个包含 $|M_\gamma|$ 条边的匹配。如果匹配中的一条边连接$\mathbf{s}$和 $(s_i', \mathbf{s}_{-i})$，我们就在两个配置中都选择竞标者 i。由于每条边都必须连接一个必须匹配顶点，

且匹配中有 $|M_\gamma|$条边，因此其余顶点（如果存在）必须是可能匹配顶点。我们从其余顶点$\mathbf{s}$中任意选择一个属于 $A_\gamma(\mathbf{s}) \setminus C(\mathbf{s})$ 的竞标者。容易验证，这样的分配满足单调性和近似约束。

- 现在假设存在一个 γ-近似的确定性真实机制。如果$\mathbf{s}$ 是必须匹配的，且i 在$\mathbf{s}$处被选中，那么我们就将 $\mathbf{s}$ 与（s_i'，$\mathbf{s}_{-i}$）以 $s_i' \neq s_i$进行匹配。注意，该边确实存在，因为真实性意味着 i 在（s_i'，$\mathbf{s}_{-i}$）处也会被选中。此外，注意每条边恰好连接一个必须匹配的顶点，因此我们得到一个包含 $|M_\gamma|$条边的匹配。

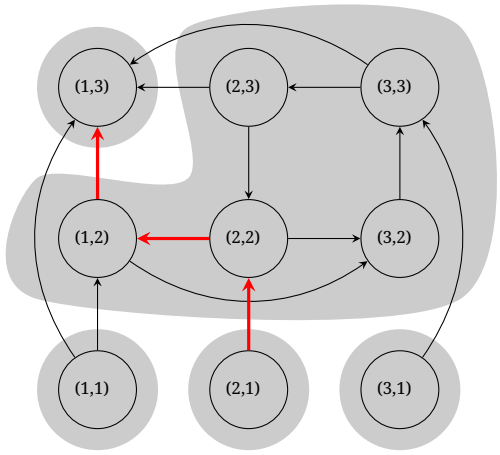
3.2 归约到布尔可满足性

接下来，我们考察只有两个参与者的特殊情形；Ede 在 [+18] 中研究了估值函数受限的域。我们提出一种新方法，将 $\textsc{Det}_\gamma$ 归约为布尔公式的可满足性。关于这一构造的更多细节，请参见第 4.1.3 节和第 4.2.2 节。

当 $n = 2$时，注意到确定性分配规则完全由变量x_1（$\mathbf{s}$）刻画，其中$\mathbf{s} \in [k]^2$，因为 x_2（$\mathbf{s}$） $= 1 - x_1$（$\mathbf{s}$）。由于变量属于 $\{0, 1\}$，我们可以用以下布尔公式编码单调性和比率约束。

$$
\begin{aligned}
&\forall s_2 \in [k], \forall (s_1, s_1') \in \sigma_1(s_2), && x_1(s_1, s_2) \Rightarrow x_1(s_1', s_2), && (\text{i.e., } x_1(s_1, s_2) \leq x_1(s_1', s_2)) \\
&\forall s_1 \in [k], \forall (s_2, s_2') \in \sigma_2(s_1), && x_1(s_1, s_2') \Rightarrow x_1(s_1, s_2), && (\text{i.e., } x_2(s_1, s_2) \leq x_2(s_1, s_2')) \\
&\forall \mathbf{s} \in [k]^2 \text{ s.t. } \rho_1(\mathbf{s}) < 1/\gamma, && \neg x_1(\mathbf{s}), && (\text{i.e., } x_1(s_1, s_2) = 0) \\
&\forall \mathbf{s} \in [k]^2 \text{ s.t. } \rho_2(\mathbf{s}) < 1/\gamma, && x_1(\mathbf{s}). && (\text{i.e., } x_2(s_1, s_2) = 0)
\end{aligned}
$$

该公式可以改写为由大小为 2 的子句构成的合取，对应于 2SAT问题。不过，我们将保留这种表述，因为它对应于使用强连通分量 [APT79] 求解 2SAT 时所用的图。更具体地说，我们在信号配置集合 $[k]^2$ 上构建一个有向图 $G(\boldsymbol{\sigma})$，并将布尔公式中的所有蕴含作为边加入其中，如图 2 所示。



假设 $\gamma = 2$ 并且：

- $\rho_2(2,\ 1) = 0.4 \leq 1/\gamma$，因此 $x_1(2,\ 1) = 1$；
- $\rho_1(1,\ 3) = 0.3 \leq 1/\gamma$，因此 $x_1(1,\ 3) = 0$。

更准确地说，对于所有 {v1}，红色链都存-
对于所有 $\gamma < 2.5 = 1/0.4$ 均成立。

图 2：由包含 $n = 2$ 个代理人的布尔公式得到的图 $G(\boldsymbol{\sigma})$ 示例。对于每个 $i \in [n]$和 $\mathbf{s}_{-i} \in [k]$，单调性约束由（偏）严格序 $\sigma_i(\mathbf{s}_{-i})$ 给出。存在一条以红色表示的阻塞蕴含链。为了高效计算是否存在阻塞蕴含链，可以移除可通过传递性推导出的冗余边，并计算以灰色表示的强连通分量。

可以看出，当且仅当能找到一条阻塞链 $1 = x_1(\mathbf{s}) \Rightarrow \cdots \Rightarrow x_1(\mathbf{s}') = 0$ 时，不存在满足赋值；由于 $\boldsymbol{\rho}$ 的限制，这条链的起点和终点受到约束。使用深度优先搜索可以在多项式时间内完成检查；如果构造图时注意避免添加过多的边，还可以将时间复杂度改进为 $O(k^2)$。

3.3 冲突对

在上一节中，我们讨论了如何在 $n = 2$ 时利用布尔可满足性求解 $\textsc{Det}_\gamma$，而这可以改写为图可达性问题。现在，我们给出最优确定性比率 R_D^* 的一种刻画，无需使用二分搜索归约来求解 $\textsc{Det}_\gamma$，即可直接计算该比率。关于这一构造的更多细节，请参见第4.2.2节。

第一步是观察之前的归约，同时将 γ 从 $+\infty$ 缓慢递减至 1。起初，不存在满足 $\rho_i(\mathbf{s}) \le 1/\gamma$ 的 $\mathbf{s} \in [k]^2$ 和 $i \in [2]$，因此不可能存在蕴含阻塞链。然后，我们开始固定一些变量 $x_1(\mathbf{s})$。直到固定最后一个变量时，便会产生一条阻塞蕴含链，从而阻止满足赋值的存在。利用这一观察，我们可以精确刻画最优确定性比率：

$$R_D^*(\boldsymbol{\rho}, \boldsymbol{\sigma}) = \max_{\mathbf{s} \to \mathbf{s}'} \min \left(\frac{1}{\rho_2(\mathbf{s})}, \frac{1}{\rho_1(\mathbf{s}')} \right),$$

其中，$\mathbf{s} \to \mathbf{s}'$ 表示图 $G(\boldsymbol{\sigma})$ 中的可达性，而我们将 $(\mathbf{s},\mathbf{s}')$ 称为一个冲突对。重要的是，在以 $O(k^2 \log k)$ 的时间对信号排序后，我们可以利用有向无环图（DAG）$G(\boldsymbol{\sigma})$ 的强连通分量进行动态规划，并结合一个满足赋值，在 $O(k^2)$ 时间内计算该值。在第4.2.2节中，我们还给出了比率 R_V^* 和 R_C^* 的刻画，这些比率可在 $O(k^2 \log k)$ 时间内求解问题$\textsc{Val}$、$\textsc{Cst}$ 和 $\textsc{Det}$。

3.4 查询复杂度

在前面的章节中，我们给出了求解特殊情形下 $\textsc{Det}$ 的多项式时间算法（及其直观解释）。然而，在机制设计中，人们可能只关心在特定信号配置$\mathbf{x}(\mathbf{s})$ $\mathbf{s} \in [k]^n$处计算结果。回想一下，查询问题 $\textsc{Det}_\gamma$ 对这种可能性进行了精确形式化。基于前两节的内容，我们现在证明 $n = 2$ 时的特殊情形下的定理 5，表明要在特定信号配置处计算结果，必须读取 $\Omega(k^2)$ 个价值。关于一般构造的更多细节可参见第 5.3 节。

当 $n = 2$ 时，我们已经看到，引理2.2中的单调性条件可以用有向图 $G(\boldsymbol{\sigma})$ 来表述。为简单起见，我们将构造一个价值函数递增的实例，从而得到图 $G(\boldsymbol{\sigma})$，其中 $\mathbf{s} \to \mathbf{s}'$ 当且仅当 $s_1 \ge s_1'$ 且 $s_2 \le s_2'$。我们令 $\varepsilon > 0$，并固定一个满足 $s_1 = s_2 = 1 + \lfloor k/2 \rfloor$ 的信号配置$\mathbf{s}$。然后，我们随机抽取一个信号配置$\mathbf{s}' \in [k]^2$：

- 如果有 $\mathbf{s}' \to \mathbf{s}$，我们设为 $\rho_2(\mathbf{s}') = \varepsilon$，并将所有其他性能比率设为 1，
- 如果有 $\mathbf{s} \to \mathbf{s}'$，我们设为 $\rho_1(\mathbf{s}') = \varepsilon$，并将所有其他性能比率设为 1，
- 否则，我们将所有性能比率设为 1。

注意，始终存在一个非常简单的分配规则，它在所有信号配置下都选择同一个代理人，并达到 1 的比率。遗憾的是，任何能求解 $\textsc{Det}_1$ 的分配规则 $\mathbf{x}$ 都必须在 $\mathbf{s}' \to \mathbf{s}$ 时满足 $x_1(\mathbf{s}) = 1$，并且在 $\mathbf{s} \to \mathbf{s}'$ 时满足 $x_2(\mathbf{s}) = 1$。因此，在最坏情况下，必须查询 $\Omega(k^2)$ 个信号配置下的价值，才能判断当前属于哪种情形。可以使用代数决策树模型将此构造形式化 [Ben83]。顺带一提，即使采用随机探索策略来寻找 $\mathbf{s}'$，也无法降低查询复杂度，因为对手并非自适应的。

3.5 NP-困难性

为了总结本节中关于我们技术的重要思路，我们将简要介绍定理 4 的证明思路。该定理指出，当 $n = 4$ 时，$(1, \beta)$-$\textsc{Det}$ 是 NP 困难的。有关这一构造的更多细节，请参见第5.2节。

为简单起见，我们将构造一个价值函数递增的实例，这样可以简化引理2.2中的单调性条件：

$$\forall i \in [n], \forall \mathbf{s}_{-i} \in [k]^{n-1}, \forall 1 \le s_i \le s_i' \le k, \qquad x_i(s_i, \mathbf{s}_{-i}) \le x_i(s_i', \mathbf{s}_{-i}).$$

我们通过从 1-in-3-SAT 问题归约来证明 NP-困难性。该问题要求为变量指定一个布尔赋值，并满足约束：三个文字（变量或其否定）中恰有一个为真。下文将通过图3至图5阐述归约的主要思路。为方便起见，我们设定一个常数 $\varepsilon \in (0,\ 1/\beta)$。为了建立一些直观认识，我们设定

$$\boldsymbol{\rho}(1,2,1,*) = (1,\varepsilon,1,\varepsilon), \qquad \boldsymbol{\rho}(1,1,2,*) = (1,1,\varepsilon,\varepsilon), \qquad \boldsymbol{\rho}(k,1,1,*) = (\varepsilon,1,1,\varepsilon).$$

首先，由于性能比，如果 $\mathbf{x} \in \mathcal{T}_D(\boldsymbol{\sigma})$ 使得 $R_D(\mathbf{x}\ \boldsymbol{\rho}) = 1$，那么我们有

$$x_1(k,1,1,*) = x_4(k,1,1,*) = x_2(1,2,1,*) = x_4(1,2,1,*) = x_3(1,1,2,*) = x_4(1,1,2,*) = 0.$$

如果我们决定设定 $x_1(1,\ 1,\ 2,\ *) = 1$，那么

$$\begin{aligned}
x_1(1,1,2,*) = 1 \quad &\Rightarrow \quad x_1(k,1,2,*) = 1 && \text{（根据单调性）}\\
&\Rightarrow \quad x_3(k,1,2,*) = 0 && \text{（概率之和为 1）}\\
&\Rightarrow \quad x_3(k,1,1,*) = 0 && \text{（根据单调性）}\\
&\Rightarrow \quad x_2(k,1,1,*) = 1 && \text{（概率之和为 1）}\\
&\Rightarrow \quad \dots\\
&\Rightarrow \quad x_1(1,1,2,*) = 1
\end{aligned}$$

因此，这些条件彼此等价；如果它们成立，我们就称构件为真。反之，如果 $x_1(1,\ 2,\ 1,\ *) = 1$ 成立，我们就称构件为假。这个“变量”构件如图3所示，是我们从 1-in-3-SAT 问题进行归约的一个构建块。

在图 4 中，我们将三个“变量”构件相交于同一个信号配置，要求三个文字中恰有一个为真，从而构成一个“1-in-3 子句”构件。最后，在图 5 中，我们展示如何连接“子句”构件和“变量”构件，要求变量和文字的取值保持一致。

更详细的描述见第5.2节。

4 算法与上界

现在我们给出正面结果的证明：针对 VAL、CST 以及 DET 的特殊情形的多项式时间算法。本节分为两部分。首先，在第4.1节中，我们使用线性和组合优化问题来表述一般情形，其中一些表述易于求解，另一些则是 NP 困难。然后，在第4.2节中，我们探讨 NP 困难问题能够高效求解的特殊情形。

4.1 一般情形的表述

在本节中，我们给出优化问题的一般表述：VAL 和 CST 可以表示为线性规划的解，而 DET 既可以表示为整数规划的解，也可以表示为布尔可满足性问题，或超图中的匹配问题。

4.1.1 线性规划

给定参数（$\boldsymbol{\rho}$，$\boldsymbol{\sigma}$），求解 VAL 和 CST 包括在真实多面体 $\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$ 上最小化 $R_V(\boldsymbol{\rho}, \mathbf{x})$ 和 $R_C(\boldsymbol{\rho}, \mathbf{x})$。根据公式（$R_C$），比率 $R_C(\boldsymbol{\rho}, \mathbf{x})$ 是（关于 $\mathbf{x}$ 的）线性函数的最大值，因此是凸函数。$R_V(\boldsymbol{\rho}, \mathbf{x})$ 的情形略有不同，但可以证明，对于所有 $\mathbf{s}$，$\mathbf{x} \mapsto 1/\langle \mathbf{x}(\mathbf{s})\ \boldsymbol{\rho}(\mathbf{s})\rangle$ 是凸函数，因此根据公式（R_V），$R_V(\boldsymbol{\rho}, \mathbf{x})$ 也是凸函数。由于在凸多面体上最小化凸函数是可处理的，这直观地解释了为什么 VAL 和 CST 可以在多项式时间内计算。重要的是，我们可以进一步细化这一表述，将这两个问题表示为线性规划的解，从而得到定理 3 的证明。

定理 3. 可以在 $N^{O(1)}$ 时间内求解 VAL 和 CST 。

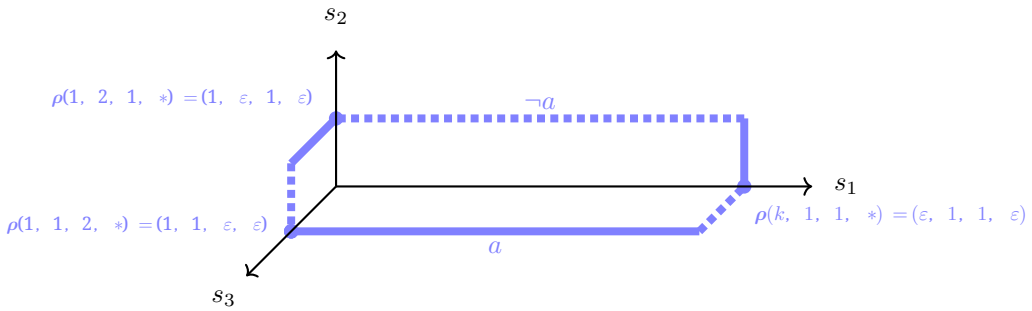


图3：位于 $(s_2,\ s_3)$ 的变量 a 构件 $=(1,\ 1)$。我们有 $n = 4$ 个代理人，但不表示第四维（我们将 $\boldsymbol{\rho}$ 设为在 s_4 上为常数）。在三个顶点处，我们将ε 处的某些性能比设定为特定值，以防止物品分配到该处，其余比率设为 1。

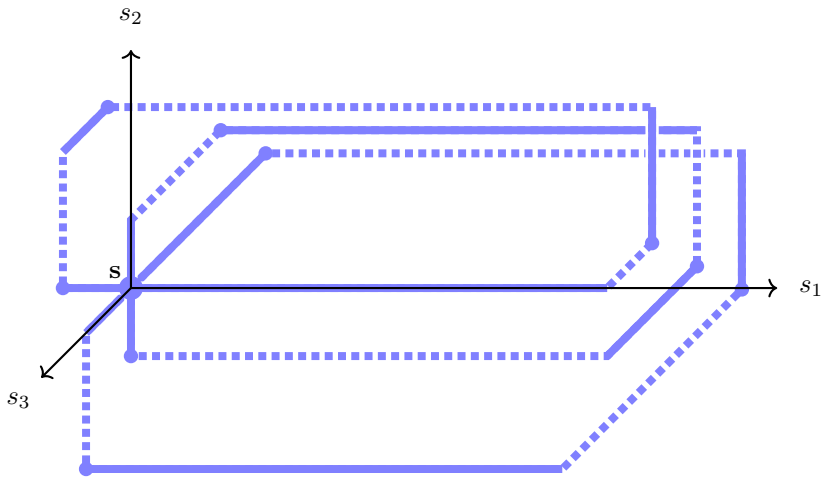


图 4：子句 $\ell_1 \vee \ell_2 \vee \ell_3$的构件。我们为每个文字创建一个环，它们都在一个信号配置 s 处相交。我们设定 $\boldsymbol{\rho}$(s) $=(1,\ 1,\ 1,\ \varepsilon)$，而在 s 处胜者的选择决定了三个文字中的哪一个为真。

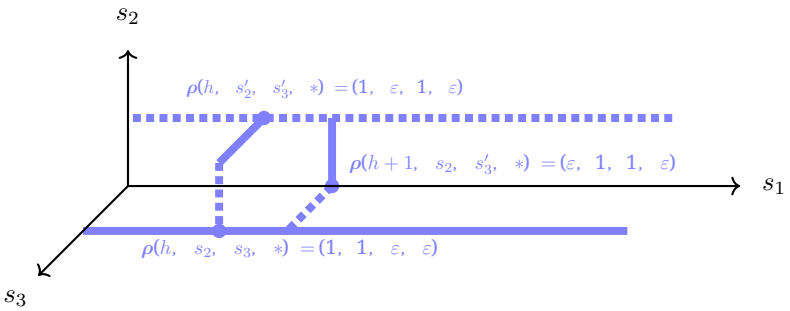


图 5：高度为 h 处异或连接器的构件。它连接两条水平线（$s_2,\ s_3$）和（$s_2',\ s_3'$），使得 $s_2 < s_2'$ and $s_3 > s_3'$，并且没有其他水平线共享同一坐标。

证明。使　用经典构造，我们可以将成本比率和价值比率表示为线性目标函数：

$$\begin{array}{lll} \text{minimize} & \alpha & (R_C^*(\boldsymbol{\rho},\boldsymbol{\sigma})) \\ \text{such that} & \alpha \geq \langle \mathbf{x}(s), 1/\boldsymbol{\rho}(s)\rangle \qquad \forall \mathbf{s} \in [k]^n & \\ & \mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma}), & \end{array}$$

and

$$\begin{array}{lll} \text{maximize} & \beta & (1/R_V^*((\boldsymbol{\rho},\boldsymbol{\sigma}))) \\ \text{such that} & \beta \leq \langle \mathbf{x}(s), \boldsymbol{\rho}(s)\rangle \qquad \forall \mathbf{s} \in [k]^n & \\ & \mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma}). & \end{array}$$

使用椭球法 [Kha79] 或内点法 [Kar84], ，可以在多项式时间内优化这两个线性规划，并得出实现最优近似比的机制。 □

重要的是，求解线性规划的复杂度是"弱"多项式的，因为它随表示系数所用的比特数 b 呈多项式增长。对于线性规划的特殊情形，存在强多项式时间算法 [Tar86], ，但对于约束矩阵含有大系数的线性规划（如我们这里的问题），目前尚无已知的强多项式时间算法。

4.1.2 整数规划

上一节中，我们看到如何利用线性规划计算最优价值比率和成本比率。然而，若用同样的方法求解 DET ，则需要求解一个一般情形下很难处理的整数规划。接下来我们将说明，当真实多面体 $\mathcal{T}(\boldsymbol{\sigma})$ 的极点均为整数极点时，该问题是可处理的。

命题 4.1。 如果 $\mathcal{T}(\boldsymbol{\sigma})$ 具有整数顶点，那么可以在多项式时间内求解 DET 。

证明。 为计算 $R_D^*(\mathbf{v})$ 及相应的确定性机制，我们将运行引理3.1中的二分搜索。为回答 DET_γ 查询，我们求解以下线性规划：

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{\mathbf{s}\in[k]^n}\sum_{i\in[n]} x_i(\mathbf{s}) \cdot \mathbb{1}[\rho_i(\mathbf{s}) \geq 1/\gamma] & (R_D^*(\mathbf{v}) \leq \gamma) \\ \text{such that} & \mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma}). & \end{array}$$

由于目标函数是线性的，且 $\mathcal{T}(\boldsymbol{\sigma})$ 是一个整数多面体，因此最大值在某个整数顶点 $\mathbf{x}$ 处取得。此外，注意到最大值等于 k^n ，当且仅当存在一个确定性分配 $\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$，它在每个信号配置 $\mathbf{s}$ 处都选择一个代理人 $i \in [n]$ ，使得 $\rho_i(\mathbf{s}) \geq 1/\gamma$；也就是说，当且仅当 $R_D^*(\boldsymbol{\rho},\boldsymbol{\sigma}) \leq \gamma$。 □

除了确定性比率的多项式时间可计算性之外，真实多面体的整数性还意味着所有 $\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$ 都是普遍真实的；我们回顾一下，普遍真实是指对真实的确定性机制进行彩票抽取。

命题 4.2。 如果真实多面体 $\mathcal{T}(\boldsymbol{\sigma})$ 的顶点都是整数顶点，那么 $\mathcal{T}(\boldsymbol{\sigma})$ 中的分配规则满足更强的普遍真实性概念。

证明。 任何分配规则 $\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$ 都可以表示为 $\mathcal{T}(\boldsymbol{\sigma})$ 的极点的凸组合。利用真实多面体是整数多面体这一事实，$\mathbf{x}$ 可以通过对确定性分配规则进行彩票抽取来实现，而这些规则都可以如实实现。 □

在第 4.2.1 节中，我们将证明，当 $n = 2$ 或 $k = 2$ 时， $\mathcal{T}(\boldsymbol{\sigma})$ 的顶点总是整数顶点；但当 $n \geq 3$ 和 $k \geq 3$ 时，则可能存在分数顶点。

4.1.3 布尔可满足性

在上一小节中，我们将 $\textsc{Det}_\gamma$ 表述为在 $\mathcal{T}_D(\boldsymbol{\sigma})$上最大化一个线性目标函数；如果相关多面体 $\mathcal{T}(\boldsymbol{\sigma})$ 的顶点是整数顶点，那么该问题便易于求解。由于在确定性情形中，变量$x_i(\mathbf{s})$ 属于 $\{0,1\}$，因此可以将概率、单调性和比率约束编码为布尔公式。因此， $\textsc{Det}_\gamma$ 也可以表述为布尔可满足性问题。

更形式化地说，在 SAT问题中，每个约束都是一个子句 （析取），其中必须有一个文字（变量或其否定）为真。由此得到的公式是所有子句的合取，称为合取范式 （CNF）。如果存在一种变量的布尔赋值，使公式的取值为真，那么该公式就是可满足的。

引理 4.3。 给定参数$(\boldsymbol{\rho},\boldsymbol{\sigma})$ 和 $\gamma \geq 1$，存在机制$\mathbf{x} \in \mathcal{T}_D(\boldsymbol{\sigma})$，其近似比为 $R(\boldsymbol{\rho},\mathbf{x}) \leq \gamma$，这等价于以下公式可满足:

$$\begin{aligned}
\forall \mathbf{s} \in [k]^n, \qquad & x_i(\mathbf{s}) \vee \cdots \vee x_n(\mathbf{s}), \\
\forall \mathbf{s} \in [k]^n, \forall i,j \in [n] \ s.t.\ i \neq j, \qquad & \neg x_i(\mathbf{s}) \vee \neg x_j(\mathbf{s}), \\
\forall i \in [n], \forall \mathbf{s}_{-i} \in [k]^{n-1}, \forall (s_i, s_i') \in \sigma_i(\mathbf{s}_{-i}), \qquad & \neg x_i(s_i, \mathbf{s}_{-i}) \vee x_i(s_i', \mathbf{s}_{-i}), \\
\forall \mathbf{s} \in [k]^n, \forall i \in [n] \ s.t.\ \rho_i(\mathbf{s}) < 1/\gamma, \qquad & \neg x_i(\mathbf{s}).
\end{aligned}$$

证明。 注意，前两个子句对应于公式 (1) 中的概率约束，第三个子句对应于公式 (2) 中的单调性约束，第四个子句则对近似比施加约束。

□

遗憾的是，一般情形下，为 CNF 公式寻找满足赋值的问题是 NP 困难的。不过请注意，当 $n = 2$ 所有子句的大小均不超过 2 时，这对应于 2-SAT问题，而该问题可以在多项式时间内求解。我们将在第4.2.2节中利用这一点。

4.1.4 超图中的完美匹配

最后，本节将 $\textsc{Det}_\gamma$ 重新表述为第三种形式，即在超图中寻找完美匹配的问题。更正式地说，超图由顶点集合 $vertices$ V 和超边集合$hyperedges$ E描述，其中每条超边 $e \in E$ 都是顶点集合 $e \subseteq V$的子集。匹配是由若干条超边 $M \subseteq E$组成的集合，其中每个顶点至多属于一条边。如果每个顶点恰好属于一条超边，则该匹配是完美匹配。

引理4.4。 给定参数 $(\boldsymbol{\rho},\boldsymbol{\sigma})$ 和 $\alpha \geq 1$，存在一个机制 $\mathbf{x} \in \mathcal{T}_D(\boldsymbol{\sigma})$，其近似比率为 $R(\boldsymbol{\rho},\mathbf{x}) \leq \alpha$，等价于下述超图中存在完美匹配:

$$V := [k]^n \qquad and \qquad E := \bigcup_{i \in [n]} \{e \in E_i \mid \forall \mathbf{s} \in e, \rho_i(\mathbf{s}) \geq 1/\alpha\}$$

其中，对所有 $i \in [n]$，有

$$E_i = \left\{ \{(s_i, \mathbf{s}_{-i}) \mid s_i \in S\} \;\middle|\; \begin{array}{l} S \subseteq [k] \ and \ \mathbf{s}_{-i} \in [k]^{n-1} \ such \ that \\ \forall (s_i, s_i') \in \sigma_i(\mathbf{s}_{-i}),\ s_i \in S \Rightarrow s_i' \in S \end{array} \right\}$$

证明。 给定一个完美匹配 $M \subseteq E$，对于每个满足 $e \in E_i$ 的 $e \in M$，我们将对所有 $\mathbf{s} \in e$设置 $x_i(\mathbf{s}) = 1$。当 e 同时属于两个集合 E_i 和 E_j 且满足 $i \neq j$时，必须谨慎处理；在这种情况下，有 $e = \{\mathbf{s}\}$，因此设置 $x_i(\mathbf{s}) = 1$ 或 $x_j(\mathbf{s}) = 1$ 均可。

首先，注意到由于匹配是完美匹配，对于每个 $\mathbf{s} \in [k]^n$，都存在唯一的 $i \in [n]$ 使得 $x_i(\mathbf{s}) = 1$，因此 $\mathbf{x}$ 满足公式（1）。其次，根据 E_i 的构造，得到的分配规则满足公式（2）的单调性条件，因此是真实的。最后，根据 E 的构造，对于所有 $i \in [n]$ 以及满足 $x_i(\mathbf{s}) = 1$ 的 $\mathbf{s} \in [k]^n$，我们有 $\rho_i(\mathbf{s}) \geq 1/\alpha$，因此 $R(\boldsymbol{\rho},\mathbf{x}) \leq \alpha$。 □

这种构造看起来可能过于复杂，因为在超图中计算完美匹配一般是 NP 困难的。不过请注意，该图中的超边大小至多为 k；当 $k = 2$ 时，它对应于一个标准图。我们将在第 4.2.3 节中运用这一观察。

4.2 具有高效算法的特殊情形

在第4.1节中，我们使用线性规划、布尔可满足性和超图匹配，对优化问题进行了几种通用的重新表述，其中大多数在一般情形下都难以求解。本节将探讨特殊情形，并说明如何利用这些重新表述来设计高效算法。

4.2.1 整数性的条件

在第 4.1.2 节的分析中，我们证明了真实多面体的整数性意味着确定性比率可在多项式时间内计算，且具有普遍真实性。现在我们转而讨论整数性成立的特殊情形。

我们首先回顾一下：如果价值（或成本）中存在平局，即存在 $i, \mathbf{s}_{-i}, s_i$ 和 s_i'，使得$v_i(s_i,\mathbf{s}_{-i}) = v_i(s_i', \mathbf{s}_{-i})$（或 $c_i(s_i,\mathbf{s}_{-i}) = c_i(s_i', \mathbf{s}_{-i})$），那么诱导出的严格序就不是全序，因为 (s_i, s_i') 和 (s_i', s_i) 均不属于 $\sigma_i(\mathbf{s}_{-i})$。尽管如此，我们认为，将注意力限制在无平局实例上并不失一般性。由于任何严格偏序都可以扩展为严格全序 [Szp30]，任何严格序集合 $\boldsymbol{\sigma}$ 都可以扩展为 $\tilde{\boldsymbol{\sigma}}$，使其成为全序，我们将此关系记为 $\boldsymbol{\sigma} \subset \tilde{\boldsymbol{\sigma}}$。

引理 4.5。 真实多面体 $\mathcal{T}(\boldsymbol{\sigma})$ 具有整数顶点，前提是对于满足 $\boldsymbol{\sigma} \subset \tilde{\boldsymbol{\sigma}}$ 的严格全序集合 $\tilde{\boldsymbol{\sigma}}$，$\mathcal{T}(\tilde{\boldsymbol{\sigma}})$ 具有整数顶点。

证明。 为便于分析，我们引入记号 $\mathcal{V}(\boldsymbol{\sigma})$，表示真实多面体 $\mathcal{T}(\boldsymbol{\sigma})$ 的顶点集合。我们断言

$$\mathcal{V}(\boldsymbol{\sigma}) \subset \bigcup_{\tilde{\boldsymbol{\sigma}}:\boldsymbol{\sigma}\subset\tilde{\boldsymbol{\sigma}}} \mathcal{V}(\tilde{\boldsymbol{\sigma}}),$$

这直接推出该命题。该结论成立的理由如下。任取 $\mathbf{x} \in \mathcal{V}(\boldsymbol{\sigma})$。那么存在某个 $\tilde{\boldsymbol{\sigma}}$，使得 $\boldsymbol{\sigma} \subset \tilde{\boldsymbol{\sigma}}$，且 $\mathbf{x} \in \mathcal{T}(\tilde{\boldsymbol{\sigma}})$。事实上，我们可以通过添加所有满足以下两个条件的配对 (s_i, s_i')，将每个 $\sigma_i(\mathbf{s}_{-i})$ 扩展为：(s_i, s_i') 和 (s_i', s_i) 均不属于 $\sigma_i(\mathbf{s}_{-i})$，且$x_i(s_i,\mathbf{s}_{-i}) < x_i(s_i',\mathbf{s}_{-i})$。再将得到的 $\boldsymbol{\sigma}$ 任意扩展为严格序的全体集合 $\tilde{\boldsymbol{\sigma}}$，即可得到$\boldsymbol{\sigma} \subset \tilde{\boldsymbol{\sigma}}$。容易验证，$\mathbf{x} \in \mathcal{T}(\tilde{\boldsymbol{\sigma}})$。

注意到，定义 $\mathcal{T}(\tilde{\boldsymbol{\sigma}})$ 的约束可以等价地改写为包含 $\mathcal{T}(\boldsymbol{\sigma})$ 的所有约束；再结合 $\mathbf{x} \in \mathcal{T}(\tilde{\boldsymbol{\sigma}})$ 这一事实，可知 $\mathbf{x} \in \mathcal{V}(\tilde{\boldsymbol{\sigma}})$。这直接由顶点的标准代数刻画 [BT97，命题 2.9]得出。

□

命题 4.6。 如果 $n = 2$ 或 $k = 2$，那么 $\mathcal{T}(\boldsymbol{\sigma})$ 仅有整数顶点。

证明。 根据引理 4.5，不失一般性地假设 $\boldsymbol{\sigma}$ 是全酉模的。在下文中，我们均在这一假设下进行讨论。

我们将证明，$\mathcal{T}(\boldsymbol{\sigma})$ 的约束矩阵是全幺模的（TU）；结合约束常数向量为整数这一事实，可知极点是整数的 [Sch98]。对于所有 $\mathbf{x} \in \mathbb{R}_{\geq 0}^{[n]\times[k]^n}$，有 $\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$ 当且仅当

$$\left(\begin{array}{c} A \\ \hline -A \\ \hline B \end{array}\right) \cdot \mathbf{x} \quad \leq \quad \left(\begin{array}{c} 1 \\ \vdots \\ 1 \\ \hline -1 \\ \vdots \\ -1 \\ \hline 0 \\ \vdots \\ 0 \end{array}\right),$$

其中，A 包含概率约束 $\sum_i x_i(\mathbf{s}) \leq 1$，对所有$\mathbf{s}$均成立；$B$ 包含单调性约束 $x_i(s_i,\ \mathbf{s}_{-i}) - x_i(s_i',\ \mathbf{s}_{-i}) \leq 0$，对所有数对 $(s_i,\ s_i')$ 均成立，其中 s_i 是全严格序 $\boldsymbol{\sigma}_i(\mathbf{s}_{-i})$ 中 s_i' 的前驱。

首先，复制行和/或列，或改变其符号，都不会改变矩阵的全酉模性。因此，只需证明由 A 和 B 叠成的矩阵是全幺模的。根据 [Sch98，定理 19，3]，矩阵是全酉模的，当且仅当每个列集合 C 都能划分为两类 C_1 和 C_2，使得 C_1 中各列之和减去 C_2 中各列之和所得的向量，其各项$\{-1,0,1\}$ 均只有。由于全酉模矩阵的转置也是全酉模的，因此行也满足相同的性质。

- 若 $n=2$，则对于每一组列 C，我们将 C_1 设为类型 $x_1(\mathbf{s})$ 的列，并将 C_2 设为类型 $x_2(\mathbf{s})$ 的列。对于 A 的每一行，至多有两个非零项，且都等于 1，分别位于集合 C_1 和 C_2 中。对于 B 的每一行，至多有两个非零项，分别等于 1 和 -1，且都位于同一个集合 C_1 或 C_2 中。
- 若 $k=2$，则对于每一组行 R，如果 $\sum_j s_j$ 为偶数，就将 A 的行 $\mathbf{s}$ 归入类 R_1；否则归入类 R_2。此外，如果 B 中 $x_i(\mathbf{s})$ 的系数为 -1，就将 B 的行 $(i,\mathbf{s})$ 归入与 A 的行 $\mathbf{s}$ 相同的类；若该系数为 1，则归入相反的类。注意，存在两个不同的对 $(i,\mathbf{s})$ 和 $(i,\mathbf{s}')$，它们对应于 B 的同一行，但可以验证，这种分类不会产生冲突。于是，每一列 $(i,\mathbf{s})$ 至多有 2 个非零项：一个位于 A 中且等于 1，另一个位于 B 中且等于 -1 或 1。根据构造，当且仅当这两个项属于不同的类时，它们的符号相同。

在这两 种情况下，得到的向量的系数均属于 $\{-1,0,1\}$，这证明该矩阵是全酉模。 □

上述命题与命题 4.1 一起直接推出以下推论。

推论 4.7. 如果 $n=2$ 或 $k=2$，则可以在多项式时间内求解 DET。

我们通过证明在 $n\geq 3$ 和$k\geq 3$时，即使 $\boldsymbol{\sigma}$ 是全体的，且对于任意给定的 i， σ_i（$\mathbf{s}_{-i}$）也会为所有 $\mathbf{s}_{-i}$诱导相同的顺序，从而补充命题 4.6 的结果：此时整数性不成立。

命题 4.8. 如果 $n\geq 3$ 且 $k\geq 3$，那么 $\mathcal{T}(\boldsymbol{\sigma})$ 可能有一些分数顶点。

证明。 我们考虑估值函数严格递增的情形，即对于所有 $i\in[n]$ 和 $\mathbf{s}_{-i}$， $\sigma_i(\mathbf{s}_{-i})$ $=\{(s_i,s_i')$ $\mid 1\leq s_i<s_i'\leq k\}$。在图 6 中，我们绘出了其中两个顶点，它们分别满足 $n=3$ 和 $k=3$，并且有一些分数坐标。我们使用 lrslib 软件生成了这些顶点，可以手动验证它们是该多面体的极点。

要将这些极点扩展到 $\mathcal{T}(\boldsymbol{\sigma})$，并满足 $n\geq 4$，只需对所有 $i\geq 4$ 设置 $x_i(\mathbf{s})=0$。要将它们扩展到 $k\geq 3$，可以设置 $\mathbf{x}(\mathbf{s}):=\mathbf{x}(\mathbf{s}')$，其中 $\mathbf{s}'=(\min(s_i,3))^{i\in[n]}$。 □

最后，当 $n\geq 3$ 和 $k\geq 3$时，注意到对于某些 $\boldsymbol{\sigma}$， $\mathcal{T}(\boldsymbol{\sigma})$ 可能是整数；例如，当所有价值与成本函数都是常数时，此时 $\sigma_i(\mathbf{s}_{-i})=\emptyset$，而 $\mathcal{T}(\boldsymbol{\sigma})$ 则仅由概率约束 $\sum_i x_i(\mathbf{s})=1$ 对所有 $\mathbf{s}\in[k]^n$ 所定义。

4.2.2 两个参与者的精细分析

根据第 4.1.1 节和第 4.2.1 节的分析，我们已经看到，当 $n=2$ 时，可以利用线性规划结合定理 3 计算最优比率 $R_V^*(\mathbf{v})$、 $R_C^*(\mathbf{c})$，并结合推论 4.7 计算 $R_D^*(\mathbf{v})$。然而，这种方法并不能直接给出时间复杂度的精确界限。本节将针对这一特殊情形，通过将问题归约为 2-SAT，对计算 $R_D^*(\mathbf{v})$ 进行更精细的分析，从而得到其时间复杂度的近乎紧确界限。接下来，我们将介绍有向无环图（DAG）和冲突对的概念，以便精确刻画 $R_V^*(\mathbf{v})$、 $R_C^*(\mathbf{c})$ 和 $R_D^*(\mathbf{v})$。借助这一概念，我们设计一种动态规划（DP）方法来计算最优比率，并设计一种贪心算法来求得相应的机制，从而在准线性时间内解决 VAL、 CST和DET。

在双智能体情形中，有一个简单但重要的性质：我们只需通过 x_1 就能分别表述计算 $R_V^*(\mathbf{v})$、 $R_C^*(\mathbf{c})$ 和 $R_D^*(\mathbf{v})$ 的优化问题，从而简化分析。该性质直接源于以下约束：对于所有信号配置 $\mathbf{s}$， $x_2(\mathbf{s})=1-x_1(\mathbf{s})$。因此，关于 v_2 对 x_2 的非递减约束，可以替换为关于 x_1 的非递增约束。我们将在信号配置之间引入有向边，以表示 x_1 的单调性约束。由于这一性质，除非另有说明，我们将 x_1 和 $\mathbf{x}$ 都称为分配。

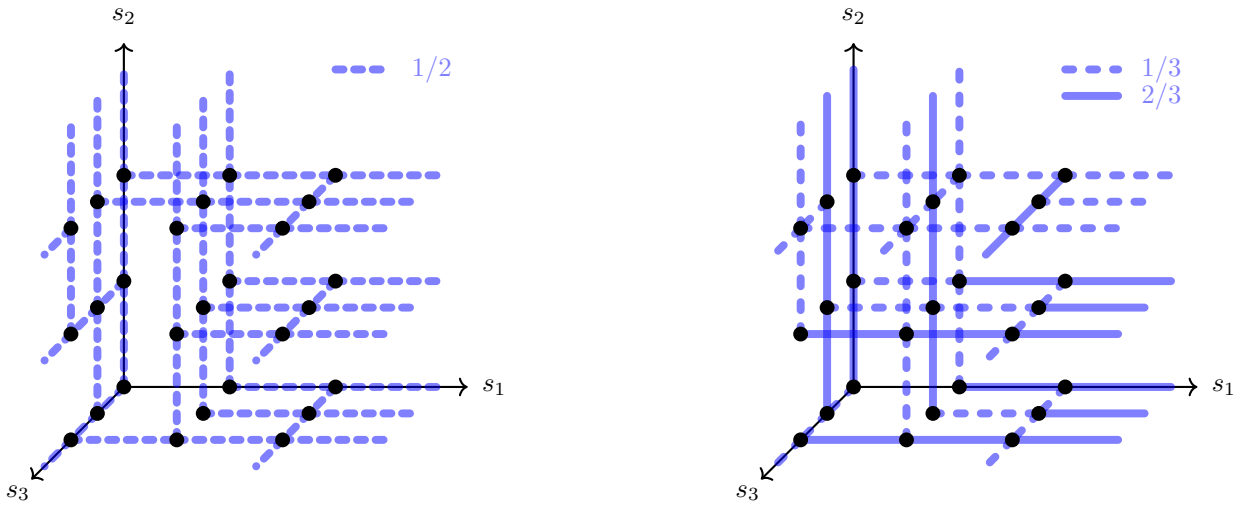


图 6：两个随机分配规则 $\mathbf{x}$ 的图示，其中包含 $n = 3$ 和 $k = 3$。对于所有 $i \in [n]$ 和 $\mathbf{s} \in [k]^n$，变量 $x_i(\mathbf{s})$ 由边 $(s_i, \mathbf{s}_{-i}) — (s_i + 1, \mathbf{s}_{-i})$ 表示，其取值由线型和图例指定。注意，这两个分配都满足引理2.2的单调性条件，并具有单调价值函数：对于所有 $\mathbf{s} \in [k]^n$ 以及所有满足 $s_i < k$ 的 $i \in [n]$，都有 $x_i(s_i, \mathbf{s}_{-i}) \leq x_i(s_i + 1, \mathbf{s}_{-i})$。

有向图（DG）预处理过程。 请注意，尽管 $\boldsymbol{\sigma}$ 可以完整描述给定输入 $\mathbf{v}$ 的单调性约束，但其中可能存在冗余。因此，我们引入以下有向图（DG）预处理过程：对于给定输入$\mathbf{v}$，该过程包含以下两个步骤。

- 对价值进行排序并划分信号。对于给定的 i 和 $\mathbf{s}_{-i}$，我们将价值 $\{v_i(s_i, \mathbf{s}_{-i})\}_{s_i \in [k]}$ 按非递减顺序排序，并将所有具有相同价值的信号配置合并为块

$$B_i^1(\mathbf{s}_{-i}), B_i^2(\mathbf{s}_{-i}), \ldots, B_i^{b_i(\mathbf{s}_{-i})}(\mathbf{s}_{-i}),$$

  其中，$b_i(\mathbf{s}_{-i})$ 是 $\{v_i(s_i, \mathbf{s}_{-i})\}_{s_i \in [k]}$ 中不同价值的数量。

- 添加虚拟顶点并构造有向图。我们将信号配置视为顶点，引入虚拟顶点以减少有向边的数量，并构造有向图。具体而言，我们按如下方式构造 $DG(\mathbf{v})$：

$$\begin{aligned}
V(\mathbf{v}) &:= [k]^2 \cup \left\{ a_i^j(\mathbf{s}_{-i}) \;\middle|\; i \in [2], \mathbf{s}_{-i} \in [k], j \in [b_i(\mathbf{s}_{-i}) - 1] \right\}, \\
E(\mathbf{v}) &:= \left\{ (\mathbf{s}, a_1^j(s_2)) \;\middle|\; \mathbf{s} \in [k]^2 \text{ and } j \in [b_1(s_2) - 1] \text{ such that } \mathbf{s} \in B_1^j(s_2) \right\}, \\
&\quad \cup \left\{ (a_1^j(s_2), \mathbf{s}) \;\middle|\; \mathbf{s} \in [k]^2 \text{ and } j \in [b_1(s_2) - 1] \text{ such that } \mathbf{s} \in B_1^{j+1}(s_2) \right\}, \\
&\quad \cup \left\{ (\mathbf{s}, a_2^j(s_1)) \;\middle|\; \mathbf{s} \in [k]^2 \text{ and } j \in [b_2(s_1) - 1] \text{ such that } \mathbf{s} \in B_2^{j+1}(s_1) \right\}, \\
&\quad \cup \left\{ (a_2^j(s_1), \mathbf{s}) \;\middle|\; \mathbf{s} \in [k]^2 \text{ and } j \in [b_2(s_1) - 1] \text{ such that } \mathbf{s} \in B_2^j(s_1) \right\}.
\end{aligned}$$

我们有如下引理：

引理 4.9. 若 $n = 2$，则可以在 $O(k^2 \log k)$ 时间内对输入 $\mathbf{v}$ 执行 *DG* 程序，且所得有向图 $DG(\mathbf{v})$ 包含 $O(k^2)$ 个顶点和 $O(k^2)$ 条边。

证明。 DG 程序的时间复杂度主要来自排序步骤，使用堆排序或归并排序等众所周知的算法，可以在 $O(k^2 \log k)$ 时间内完成；参见任何标准算法教材（例如 [Cor+09]）。顶点数量可由以下事实得出：对于任意 i 和 $\mathbf{s}_{-i}$，都有$b_i(\mathbf{s}_{-i}) \leq k$。注意，每条边的一端都是一个信号配置，且每个信号配置最多出现在 4 条边中，这意味着边的数量为 $O(k^2)$。 □

我们进一步声称，真实约束可以完全由 $DG(\mathbf{v})$ 表示： $x_1(s_1, s_2) \le x_1(s_1', s_2')$ 当且仅当存在一条从 (s_1, s_2) 到 (s_1', s_2') 的路径。这直接源于这样一个事实：与不添加虚拟顶点的自然有向图相比，不存在虚拟顶点不会改变两个信号配置之间的连通性（此图未展示）。我们用 $(s_1, s_2) \prec (s_1', s_2')$ 表示存在一条从 (s_1, s_2) 到 (s_1', s_2') 的路径，其中 $(s_1, s_2) \neq (s_1', s_2')$。现在，我们准备介绍如何使用这一程序，将引理 4.3 的表述调整到特殊的双智能体情形。

归约到 2-SAT。 要计算 $R_D^*(\mathbf{v})$，我们在第 4.1.2 节中看到，可以运行二分搜索，并检查是否存在能够达到目标近似比的确定性分配规则。由于在确定性情形下，变量 $x_i(\mathbf{s})$ 属于 $\{0, 1\}$，因此可以将概率约束和单调性约束编码为布尔公式，正如引理 4.3 所述。一般而言，检查一个公式是否可满足是 NP 困难的。然而，当 $n = 2$ 所得公式较为简单时，可以在线性时间内求解 [APT79]。

命题 4.10. 若 $n = 2$，则可以在 $O(k^2 \log k)$ 内计算出最优确定性比率 $R_D^*(\mathbf{v})$。

证明。 给定一个输入 $\mathbf{v}$，我们执行 DG 预处理过程，得到一个带有虚拟顶点的有向图$DG(\mathbf{v})$。对于给定参数 $\alpha \ge 1$，我们定义以下约束，变量$\tilde{x}_1 \in \{0,1\}^{|V(\mathbf{v})|}$ 表示所有 $v \in V(\mathbf{v})$ 上的分配：

$$\begin{aligned} \forall (v', v) \in E(\mathbf{v}), \qquad & \neg \tilde{x}_1(v') \vee \tilde{x}_1(v), \\ \forall v \in [k]^2, \text{ s.t. } \rho_1(v) < 1/\alpha, \qquad & \neg \tilde{x}_1(v), \\ \forall v \in [k]^2, \text{ s.t. } \rho_2(v) < 1/\alpha, \qquad & \tilde{x}_1(v). \end{aligned}$$

我们声称，存在近似比至多为 α 的机制，当且仅当上述公式可满足。事实上，若公式有一个可行解 $\tilde{x}_1$，那么将 $\tilde{x}_1$ 限制到 $[k]^2$，即可得到一个可行分配。另一方面，若存在一个可行分配 x_1，容易验证，只需通过设置 $x_1(a_1^j(s_2))$（分别为 $x_1(a_2^j(s_1))$）等于 $\max_{\mathbf{s} \in B_1^j(s_2)} x_1(\mathbf{s})$（分别为 $\max_{\mathbf{s} \in B_2^{j+1}(s_1)} x_1(\mathbf{s})$）来扩充 x_1，即可得到该公式的一个可行解。

请注意，上述公式中的所有子句大小均至多为 2；这属于布尔可满足性的一种特殊情形，称为 2-SAT。对此，可以在线性时间内计算出一个满足赋值，例如将每个子句转换为两个文字之间的蕴含，并计算所得有向图 [APT79] 的强连通分量。我们还注意到有 $O(k^2)$ 个子句，这可直接由引理 4.9 得出。

根据引理 4.9，执行 DG 程序的时间复杂度为 $O(k^2 \log k)$。然后，我们对 $R_D^*(\mathbf{v})$进行二分搜索，该搜索会执行 $O(\log k)$ 次查询，并在 $O(k^2)$ 时间内求解一个 2-SAT 实例。总之，归约到 2-SAT 可得到一个时间复杂度为 $O(k^2 \log k)$ 的算法。 □

现在我们转而介绍另一个算法——算法1，它也能在 $O(k^2 \log k)$ 时间内计算 $R_D^*(\mathbf{v})$。尽管该算法并未降低计算 $R_D^*(\mathbf{v})$ 的时间复杂度，我们仍出于以下几个原因介绍它。首先，它的随机化变体可以用于计算 $R_V^*(\mathbf{v})$ 和 $R_C^*(\mathbf{c})$，而这两者无法通过 2-SAT 求解。其次，借助单调性值或成本，该算法相比命题 4.10 能达到更优的时间复杂度上界。最后，该算法的核心概念——冲突对——为 $R_V^*(\mathbf{v})$、 $R_C^*(\mathbf{c})$ 和 $R_D^*(\mathbf{v})$ 提供了精确的刻画。这一点本身也值得关注。

有向无环图（DAG）预处理程序 为便于分析，我们介绍以下有向无环图（DAG）预处理程序。该程序基于 DG 程序，并以 $\mathbf{v}$ 作为输入。

- 运行 DG 程序。我们得到一个满足引理 4.9 的有向图 $DG(\mathbf{v})$。
- 识别图 $DG(\mathbf{v})$ 中的所有强连通分量（SCC），并将每个分量收缩为一个顶点。令 $m(\mathbf{v})$ 表示分量总数。相应地收缩边：只要两个分量对应的

顶点在 $DG(\mathbf{v})$ 中之间存在边，就连接这两个分量。所得图是 $DG(\mathbf{v})$ 的凝缩图，它是一个有向无环图（DAG）。我们将其记为 $DAG(\mathbf{v})$。稍微滥用一下记号，顶点集（分别是边集）仍记为 $V(\mathbf{v})$（分别是 $E(\mathbf{v})$）。

- 对 $DAG(\mathbf{v})$进行拓扑排序，这会得到其顶点的一个线性顺序，并自然地导出一种层次结构。我们将 $DAG(\mathbf{v})$的顶点重新标记；这些顶点对应于 $DG(\mathbf{v})$的强连通分量，并将它们记为 $C_{\text{sc}1}$，$C_{\text{sc}2}$，……，$C_{\text{sc}m(\mathbf{v})}$

我们在下面的引理中给出DAG过程的性质。

引理 4.11. 如果 $n = 2$，则可以在 $O(k^2 \log k)$ 时间内对输入 $\mathbf{v}$ 执行*DAG*过程，所得有向图 $DAG(\mathbf{v})$ 包含 $O(k^2)$ 个顶点和 $O(k^2)$ 条边。

证明。 我们使用Tarjan线性时间算法 [Tar72],识别所有强连通分量（SCC），然后对所得凝缩图 [Cor+09, 第 22 节。4], 进行拓扑排序；这两步总共花费 $O(N)$ 时间。因此，DAG过程的时间复杂度与DG 程序相同，即为 $O(k^2 \log k)$。顶点数和边数的结果是显然的，因为图 $DAG(\mathbf{v})$是 $DG(\mathbf{v})$的凝缩图。

□

回顾一下，我们用 $(s_1, s_2) \prec (s_1', s_2')$ 表示图 $DG(\mathbf{v})$ 中存在一条从 (s_1, s_2) 到 (s_1', s_2') 的路径，其中 $(s_1, s_2) \neq (s_1', s_2')$。我们注意到，如果 $(s_1, s_2) \prec (s_1', s_2')$，那么 (s_1, s_2) 和 (s_1, s_2) 要么属于同一个分量，要么 (s_1, s_2) 所在的分量在拓扑序中位于 (s_1', s_2') 所在分量之前。注意，在一个分量 C_j^{sc} 内的所有信号剖面都必须具有相同的分配，我们将其记为 $x_1(C_j^{sc})$。此外，我们定义分量之间的记号 $\prec$：$C_j^{sc} \prec C_{j'}^{sc}$，当且仅当在 $DAG(\mathbf{v})$ 中存在一条从 C_j^{sc} 到 $C_{j'}^{sc}$ 的边。

现在，我们可以给出 $R_V^*(\mathbf{v})$、$R_C^*(\mathbf{c})$ 和 $R_D^*(\mathbf{v})$ 的冲突对刻画，以及基于这些刻画的算法。

冲突对。 我们先在以下定理中陈述主要结果。

定理 1。 当 $n = 2$ 时，可以在 $O(N \log N)$ 时间内求解 VAL、CST 和 DET。

我们引入冲突函数来定义冲突对。

$$f(u, v) := \begin{cases} 1 & \text{if } u = v = 1, \\ \frac{uv-1}{u+v-2} & \text{otherwise.} \end{cases}$$

可以验证以下不等式成立：

$$\forall u, v \in (0, 1], \qquad \frac{1}{f(u, v)} \leq f\left(\frac{1}{u}, \frac{1}{v}\right) \leq \min\left(\frac{1}{u}, \frac{1}{v}\right), \tag{3}$$

其中，第一个不等式在变量替换 $(x, y) = (uv, u + v)$ 后可直接得到；第二个不等式则由 f 在每个坐标上均为非递减函数这一事实得出，且 $\lim_{u \to +\infty} f(u, v) = v$。现在，我们可以在下文给出关键概念——冲突对：

De定义 4.12。 给定 $\alpha \geq 1$ 和一对信号剖面 $(s_1, s_2) \prec (s_1', s_2')$，我们称其 *t:*

- (s_1, s_2) 和 (s_1', s_2') 构成一个 α-价值冲突对，当且仅当

$$\frac{1}{f(\rho_2(s_1, s_2), \rho_1(s_1', s_2'))} > \alpha,$$

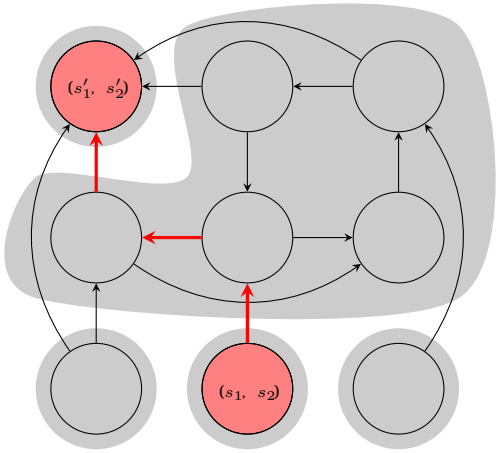
- (s_1, s_2) 和(s_1', s_2') 构成一个成本为 α 的冲突对，当且仅当

$$f\left(\frac{1}{\rho_2(s_1, s_2)}, \frac{1}{\rho_1(s_1', s_2')}\right) > \alpha,$$

- (s_1, s_2) 和(s'_1, s'_2) 构成一个确定性为 α 的冲突对，当且仅当

$$\min\left(\frac{1}{\rho_2(s_1, s_2)}, \frac{1}{\rho_1(s'_1, s'_2)}\right) > \alpha,$$

特别地，根据公式 (3)，在相同的性能比率 $\boldsymbol{\rho}$ 下，如果 $(s_1, s_2) \prec (s'_1, s'_2)$ 是一个 α-价值冲突对，那么它就是一个 α-成本冲突对；如果它是一个 α-成本冲突对，那么它就是一个 α-确定性的冲突对。



给定 $(s_1,\ s_2)$ 和 $(s'_1,\ s'_2)$，使得

- $\rho_2(s_1,\ s_2) = 0.4$,
- $\rho_1(s'_1,\ s'_2) = 0.5$,

我们有

- $1/f(\rho_2(s_1, s_2), \rho_1(s'_1, s'_2)) = 1.375$,
- $f(1/\rho_2(s_1, s_2), 1/\rho_1(s'_1, s'_2)) = 1.6$,
- $\min(1/\rho_2(s_1, s_2), 1/\rho_1(s'_1, s'_2)) = 2$,

它们分别是 $R^*_V(\boldsymbol{\rho})$、 $R^*_C(\boldsymbol{\rho})$ 和 $R^*_D(\boldsymbol{\rho})$ 的下界。

图 7：包含 $n = 2$ 个代理人的信号配置。给定 $(s_1, s_2) \in [k]^2$，所有满足 $(s_1, s_2) \prec (s'_1, s'_2)$ 的 $(s'_1, s'_2) \in [k]^2$，都是通过一条或多条弧可达的信号配置。

图 7 展示了冲突对的概念。直观地说，一个 α 冲突对为相应的 α 近似比提供了下界。我们的主要结果是，这些下界的组合是紧的。当应用于关于信号的单调价值（或成本）这一特殊情形时，我们的刻画推广了 α单交叉 条件，该条件由 [Ede+18]提出，最初仅针对单调价值函数定义。

定义 4.13。 （改编自 *[Ede+18]*）给定 $\alpha \geq 1$，如果对所有 $i \in [n]$、$\mathbf{s} \in [k]^n$ 以及满足 $\tilde{s}_i \geq s_i$ 的 $\tilde{s}_i \in [k]$，都有以下性质，则称价值设定具有 α单交叉性质：

$$\forall j \neq i, \qquad \alpha \cdot (v_i(\tilde{s}_i, \mathbf{s}_{-i}) - v_i(s_i, \mathbf{s}_{-i})) \geq v_j(\tilde{s}_i, \mathbf{s}_{-i}) - v_j(s_i, \mathbf{s}_{-i}),$$

如果对所有 $i \in [n]$、$\mathbf{s} \in [k]^n$ 以及满足 $\tilde{s}_i \geq s_i$ 的 $\tilde{s}_i \in [k]$，都有以下性质，则称成本设定具有 α单交叉性质：

$$\forall j \neq i, \qquad \alpha \cdot (c_i(\tilde{s}_i, \mathbf{s}_{-i}) - c_i(s_i, \mathbf{s}_{-i})) \leq c_j(\tilde{s}_i, \mathbf{s}_{-i}) - c_j(s_i, \mathbf{s}_{-i}).$$

我们证明，不存在冲突对这一条件严格推广了 α单交叉条件。

命题 4.14。 当 $n = 2$ 时，对于给定的 $\alpha \geq 1$，如果价值或成本设定是单调的且具有 α单交叉性质，那么就不存在 α确定性冲突对。当 $\alpha > 1$ 时，反向结论不成立。

证明。为简单起见，我们只给出价值设定的证明。注意，对于单调价值设定，偏序 $\prec$ 的定义变为：$(s_1, s_2) \prec (s'_1, s'_2)$ 当且仅当 $s_1 \geq s'_1$，$s_2 \leq s'_2$ 且 $(s_1, s_2) \neq (s'_1, s'_2)$。反设存在一个 α 确定性冲突对 (s_1, s_2) 和 (s'_1, s'_2)，其中 $(s_1, s_2) \prec (s'_1, s'_2)$，这意味着 $\max(\rho_2(s_1, s_2)\ \rho_1(s'_1, s'_2)) < 1/\alpha$。因此，我们有

$$\alpha \cdot v_2(s_1, s_2) < v_1(s_1, s_2) \quad \text{and} \quad \alpha \cdot v_1(s'_1, s'_2) < v_2(s'_1, s'_2).$$

由 α-单交叉性可得

$$\alpha \cdot (v_1(s'_1, s'_2) - v_1(s_1, s'_2)) \geq v_2(s'_1, s'_2) - v_2(s_1, s'_2)$$
$$\alpha \cdot (v_2(s_1, s_2) - v_2(s_1, s'_2)) \geq v_1(s_1, s_2) - v_1(s_1, s'_2).$$

由上述不等式可得，$\alpha \cdot v_1(s_1, s_2') < v_2(s_1, s_2')$，且 $\alpha \cdot v_2(s_1, s_2') < v_1(s_1, s_2')$，这导致了矛盾。该包含关系是严格的，如下例所示，其中 $n = k = 2$，它不是 α-单交叉的，因为当 $s_1 = 1$ 时，s_2 对 v_1 有很大影响，但不存在任何 α-确定性冲突对。

$v_1(1,1) = 1/\alpha$	$v_1(1,2) = 1$
$v_1(2,\ 1) = 1/\alpha$	$v_1(2,2) = 1$

$v_2(1,1) = 1/\alpha^2$	$v_2(1,2) = 1/\alpha^2$
$v_2(2,\ 1) = 1/\alpha$	$v_2(2,\ 2) = 1$

□

现在我们可以陈述我们的结果了。

引理 4.15。当 $n = 2$ 时，对于所有 $\alpha \geq 1$，在确定性（分别为成本或价值）设置下存在一个 α-近似，当且仅当不存在 α-确定性的（分别为 α-成本或 α-价值）冲突对。

由此可直接得出定理 1 的证明。

定理 1。当 $n = 2$ 时，可以在 $O(N$ 对数 $N)$ 时间内求解 VAL、CST 和 DET。

证明。根据引理 4.15，最优确定性近似比和随机化近似比具有以下显式表达式：

$$
\begin{aligned}
R_v^*(\mathbf{v}) &= \max_{(s_1,s_2) \prec (s_1',s_2')} \frac{1}{f(\rho_2(s_1,s_2), \rho_1(s_1',s_2'))}, \\
R_c^*(\mathbf{c}) &= \max_{(s_1,s_2) \prec (s_1',s_2')} f\left(\frac{1}{\rho_2(s_1,s_2)}, \frac{1}{\rho_1(s_1',s_2')}\right), \\
R_d^*(\mathbf{v}) &= \max_{(s_1,s_2) \prec (s_1',s_2')} \min\left(\frac{1}{\rho_2(s_1,s_2)}, \frac{1}{\rho_1(s_1',s_2')}\right).
\end{aligned}
$$

我们按照拓扑序 C_1^{sc}，C_2^{sc}，……，$C_{m(\mathbf{v})}^{\text{sc}}$逐分量计算比率。之所以可行，是因为如果 $(s_1,\ s_2) \prec (s_1',\ s_2')$，其中 $(s_1,\ s_2) \in C_j^{sc}$且 $(s_1',\ s_2') \in C_{j'}^{sc}$，那么 $j \leq j'$。为了高效地计算近似比，我们注意到函数 f 和 min 在每个坐标上都是非递减的。我们将分量 C_j^{sc}的最小 ρ_i 价值记为

$$
\rho_i(C_j^{sc}) = \min_{(s_1,s_2) \in C_j^{sc}} \rho_i(s_1,s_2), \qquad \text{where } i \in \{1,2\}.
$$

因此，对于每个 $C_{j'}^{sc}$，我们只需计算 $\rho_1(C_{j'}^{sc})$、$\rho_2(C_j^{sc})$，以及所有满足 $C_j^{sc} \prec C_{j'}^{sc}$的 C_j^{sc}对应的最小 $\rho_2(C_j^{sc})$。利用动态规划，可以针对每个 $C_{j'}^{sc}$计算量

$$
DP(C_{j'}^{sc}) = \min_{C_j^{sc} \prec C_{j'}^{sc}} \rho_2(C_j^{sc}) = \min\{\rho_2(C_{j'}^{sc}), \min_{(C_j^{sc}, C_{j'}^{sc}) \in E(\mathbf{v})} DP(C_j^{sc})\}.
$$

我们注意到，DP 过程的整体时间复杂度主要来自两类最小化步骤：在每个分量内选取最小值，以及在不同分量之间选取最小值。前者可以由 $DG(\mathbf{v})$ 中的顶点数量界定，后者可以由 $DAG(\mathbf{v})$ 中的边数量界定。因此，根据引理 4.9 和引理 4.11，我们可以得出，动态规划算法的整体时间复杂度为 $O(N)$。在计算出近似比之后，引理 4.15 的证明是构造性的，并通过对每个 $C_{j'}^{sc}$ 计算所有满足 $(C_j^{sc}, C_{j'}^{sc}) \in E(\mathbf{v})$ 的 C_j^{sc} 中最大的 $x_1(C_j^{sc})$，给出了能够达到这些比率的机制。

□

作为合理性检查，当 $n = 2$ 时，请注意，推论 2.5 比较了 $R_V^*(\mathbf{v})$、$R_C^*(\mathbf{c})$ 和 $R_D^*(\mathbf{v})$ 的比率，它直接由上述表达式以及公式（3）中的不等式推出。定理 1 直接推出以下推论。

推论 4.16。当 $n = 2$ 时，如果价值（或成本）函数是单调的，则可以在 $O(N)$ 内计算 $R_V^*(\mathbf{v})$、$R_C^*(\mathbf{c})$ 和 $R_D^*(\mathbf{v})$。

证明。 我们注意到， $O(N \log N)$ 的时间复杂度仅来自 DG 程序的数值排序步骤，而计算最优比率的所有其他步骤均需 $O(N)$ 时间。在具有单调性的情况下，可以省略数值排序步骤。

□

现在，我们定义一些符号，它们将在算法和引理 4.15 的证明中发挥作用。为避免重复，我们仅针对价值设定给出证明。在本节中，除非另有说明，所有陈述均指价值设定。

证明的核心在于，该问题的结构适合采用贪心方法。在介绍线性规划（LP）表述后，我们将给出更精确的直观解释。注意，当$n = 2$时，由于约束 $x_1(s_1, s_2) + x_2(s_1, s_2) = 1$对所有 $(s_1, s_2) \in [k]^2$均成立，分配 $\mathbf{x}$ 可以完全由 x_1描述。因此，最优随机化近似比可以通过取以下线性规划最优值的倒数来计算：

$$
\begin{aligned}
&\text{maximize} && \beta && \\
&\text{such that} && 0 \le x_1(s_1, s_2) \le 1, && \forall s_1, s_2 \in [k], && \text{(4a)}\\
&&& x_1(s_1, s_2) \cdot \rho_1(s_1, s_2) + (1 - x_1(s_1, s_2)) \cdot \rho_2(s_1, s_2) \ge \beta, && \forall s_1, s_2 \in [k], && \text{(4b)}\\
&&& x_1(s_1, s_2) \le x_1(s_1', s_2'), && \forall (s_1, s_2) \prec (s_1', s_2'). && \text{(4c)}
\end{aligned}
$$

For 在确定性情形下，我们只需向线性规划添加以下整数性约束：

$$x_1(s_1, s_2) \in \{0, 1\}, \quad \forall s_1, s_2 \in [k], \tag{5}$$

从而得到一个整数线性规划（ILP）。

注意，前两类约束，即约束 (4a) 和 (4b)，仅涉及单个信号配置，可称为逐配置约束。相比之下，第三类约束，即约束 4c，关联两个信号配置，可称为跨配置约束。还要注意，在跨配置约束中，既有不同层之间的约束，也有同一强连通分量内的约束。我们进一步将前一类跨配置约束称为跨分量约束，后一类称为逐分量约束。

该证明的直观依据是，上述 LP/ILP (4) 可以按照 $DAG(\mathbf{v})$ 的拓扑序得到紧致求解。为使阐述简洁，我们使用 LP 版本说明这一方法。考虑以下过程：我们为所有 $C_1^{sc} \in L_1$ 赋值，使其满足每个分量中信号配置所对应的逐配置约束和逐分量约束；然后，在已确定的 C_1^{sc} 的价值基础上，我们为 C_2^{sc} 选择满足逐配置约束、逐分量约束以及 C_1^{sc} 与 C_2^{sc} 之间跨分量约束的价值；依此类推。注意，如果该过程能够一直持续到为 $C_{m(\mathbf{v})}^{sc}$ 赋值，那么它就为 LP (4) 提供了一个可行解。

算法1。 Zero as Possible（ZaP）：给定输入$\mathbf{v}$，不存在 α-确定性的冲突对时 .

1：对 $j' = 1$ **到** $m(\mathbf{v})$ **执行循环 2：如果存在** $\exists (s_1, s_2) \in C_{j'}^{sc}$，**使得** $\rho_2(s_1, s_2) < \frac{1}{\alpha}$，**或 3：**$\exists C_j^{sc}$，**其中** $x_1(C_j^{sc}) = 1$ **且** $(C_j^{sc}, C_{j'}^{sc}) \in E(\mathbf{v})$，**则 4：** $x_1(C_{j'}^{sc}) \leftarrow 1$ **5：否则 6：** $x_1(C_{j'}^{sc}) \leftarrow 0$ **7：如果存在** $\exists (s_1, s_2) \in C_{j'}^{sc}$，**使得** $x_1(s_1, s_2)\rho_1(s_1, s_2) + (1 - x_1(s_1, s_2))\rho_2(s_1, s_2) < \frac{1}{\alpha}$，**则 8：返回 ERROR 9：返回 x**

–

我们的算法由参数 $\alpha \ge 1$ 参数化，使得不存在 α 冲突对。可以通过显式表达式预先计算出不存在此类冲突对的最小 α。

引理 *4.15* 的证明。 我们先考虑确定性的 情形。为证明若 方向，我们假设不存在 α-确定性冲突对；也就是说，对于所有 $(s_1, s_2) \prec (s_1', s_2')$，我们都有

$$\max(\rho_2(s_1, s_2), \rho_1(s_1', s_2')) \ge 1/\alpha.$$

算法　嗯 2。 尽可能小（SaP$_v$）：在不存在 α-价值冲突对的情况下，给定输入$\mathbf{v}$　　.

1: **for** $j' = 1$ to $m(\mathbf{v})$ **do**
2: $\quad I_v(C^{sc}_{j'}) \leftarrow \left\{ x \in [0,1] \;\middle|\; \begin{array}{l} x \cdot \rho_1(s_1,s_2) + (1-x)\cdot\rho_2(s_1,s_2) \geq 1/\alpha,\ \forall (s_1,s_2) \in C^{sc}_{j'}, \\ x \geq x_1(C^{sc}_j),\ \forall C^{sc}_j \text{ s.t. } (C^{sc}_j, C^{sc}_{j'}) \in E(\mathbf{v}). \end{array} \right\}$
3: $\quad$ **if** $I_v(C^{sc}_{j'}) = \emptyset$ **then**
4: $\quad\quad$ **return** ERROR
5: $\quad$ **else**
6: $\quad\quad x_1(C^{sc}_{j'}) \leftarrow \min I_v(C^{sc}_{j'})$
7: **返回 x**

算法3. 尽可能小 (SaP$_c$)，当不存在 α-成本冲突对时，给定输入 $\mathbf{c}$。

1: **for** $j' = 1$ to $m(\mathbf{c})$ **do**
2: $\quad I_c(C^{sc}_{j'}) \leftarrow \left\{ x \in [0,1] \;\middle|\; \begin{array}{l} x/\rho_1(s_1,s_2) + (1-x)/\rho_2(s_1,s_2) \geq \alpha,\ \forall (s_1,s_2) \in C^{sc}_{j'}, \\ x \geq x_1(C^{sc}_j),\ \forall C^{sc}_j \text{ s.t. } (C^{sc}_j, C^{sc}_{j'}) \in E(\mathbf{v}). \end{array} \right\}$
3: $\quad$ **if** $I_c(C^{sc}_{j'}) = \emptyset$ **then**
4: $\quad\quad$ **return** ERROR
5: $\quad$ **else**
6: $\quad\quad x_1(C^{sc}_{j'}) \leftarrow \min I_c(C^{sc}_{j'})$
7: **返回 x**

我们运行算法1；如果成功，它会构建一个 α-近似确定性机制，并且我们证明它永远不会进入第8行：

- 如果 $x_1(C^{sc}_{j'}) = 0$，那么我们无法触发第8行，因为对于所有 $(s_1, s_2) \in C^{sc}_{j'}$，都有 $\rho_2(s_1, s_2) \geq \frac{1}{\alpha}$；
- 如果 $x_1(C^{sc}_{j'}) = 1$ 且 $\exists (s_1, s_2) \in C^{sc}_{j'}$ 满足 $\rho_2(s_1, s_2) < \frac{1}{\alpha}$，那么我们无法触发第 8 行，因为对于所有 $(s'_1, s'_2) \in C^{sc}_{j'}$，都有 $\rho_1(s'_1, s'_2) = 1$；
- 如果 $x_1(C^{sc}_{j'}) = 1$ 且 $\rho_2(s_1, s_2) \geq \frac{1}{\alpha}$ 对于所有 $(s_1, s_2) \in C^{sc}_{j'}$ 都成立，那么存在 C^{sc}_j，使得 $x_1(C^{sc}_j) = 1$ 且 $(C^{sc}_j, C^{sc}_{j'}) \in E(\mathbf{v})$。在此情形下，必然存在一个配置 $(s''_1, s''_2) \in C^{sc}_{j''}$，使得 $C^{sc}_{j''} \prec C^{sc}_{j'}$ 且 $\rho_2(s''_1, s''_2) < \frac{1}{\alpha}$。否则，所有满足 $(C^{sc}_j, C^{sc}_{j'}) \in E(\mathbf{v})$ 的 C^{sc}_j 都会有 $x_1(C^{sc}_j) = 0$。回顾一下，在不存在 α-确定性的冲突对时，我们有 $\max(\rho_2(s_1, s_2)\ \rho_1(s''_1, s''_2)) \geq \frac{1}{\alpha}$，这意味着对于所有 $(s'_1, s'_2) \in C^{sc}_{j'}$，都有 $\rho_2(s_1, s_2) \geq \frac{1}{\alpha}$。因此，在此情形下也不会触发第 8 行。

我们来证明只有在 这一方向。反证法，假设存在一对 α-确定性冲突对 (s_1, s_2) 和 (s'_1, s'_2)，其中 $(s_1, s_2) \prec (s'_1, s'_2)$ 且

$$\max(\rho_2(s_1, s_2), \rho_1(s'_1, s'_2)) < 1/\alpha.$$

为了达到 α-近似，我们必须令 $x_1(s_1, s_2) = 1$ 且 $x_1(s'_1, s'_2) = 0$。然而，单调性约束要求 $x_1(s_1, s_2) \leq x_1(s'_1, s'_2)$，由此产生矛盾。

现在我们讨论价值 情形。首先，考虑如果 这一方向。假设不存在 α-价值冲突对，也就是说，对于所有 $(s_1, s_2) \prec (s'_1, s'_2)$，我们都有

$$f(\rho_2(s_1, s_2), \rho_1(s'_1, s'_2)) \geq \frac{1}{\alpha}.$$

我们运行算法 2；如果成功，它将构造一个价值近似比至多为 α 的随机机制。我们证明，该算法永远不会执行到第 4 行。该过程能够成功的关键在于：在每个分量中，由逐配置约束和跨分量约束定义的区间都有非空交集。请注意，只要最终得到的区间非空，逐分量约束就会自动满足，因为我们为该分量中的所有信号配置分配相同的价值，而这正是逐分量约束所要求的。

为此，我们引入以下量：

$$lp(s_1, s_2) := \begin{cases} \frac{1/\alpha - \rho_1(s_1,s_2)}{1-\rho_1(s_1,s_2)}, & \text{if } \rho_1(s_1, s_2) \neq 1, \\ -\infty, & \text{if } \rho_1(s_1, s_2) = 1, \end{cases} \tag{6}$$

and

$$rp(s_1, s_2) := \begin{cases} \frac{1-1/\alpha}{1-\rho_2(s_1,s_2)}, & \text{if } \rho_2(s_1, s_2) \neq 1, \\ +\infty, & \text{if } \rho_2(s_1, s_2) = 1, \end{cases} \tag{7}$$

which 表示由约束 (4b) 定义的区间端点。我们进一步定义：

$$lp(C^{sc}_{j'}) = \max_{(s'_1,s'_2)\in C^{sc}_{j'}} lp(s'_1, s'_2) \quad \text{and} \quad rp(C^{sc}_{j'}) = \min_{(s'_1,s'_2)\in C^{sc}_{j'}} rp(s'_1, s'_2).$$

根据定义可直接得到，对于所有 (s'_1,s'_2) 和 $C^{sc}_{j'}$，$lp(s_1, s_2) \leq 1$，$rp(s_1, s_2) \geq 0$, $lp(C^{sc}_{j'}) \leq 1$ 以及 $rp(C^{sc}_{j'}) \geq 0$ 均成立。更具体地说，我们有

$$I_v(C^{sc}_{j'}) = [0, 1] \cap [lp(C^{sc}_{j'}), rp(C^{sc}_{j'})] \cap \bigcap_{C^{sc}_j \prec C_{j'}} [x_1(C^{sc}_j), +\infty), \tag{8}$$

最后一部分在 $\ell \geq 2$ 时出现。接下来，我们声称，对于所有满足 $(s_1, s_2) \prec (s'_1,s'_2)$ 的 (s_1, s_2) 和 (s'_1, s'_2)，不等式

$$lp(s_1, s_2) \leq rp(s'_1, s'_2), \tag{9}$$

可由 lp 和 rp 的定义直接推出，并且 $f(\rho_2(s_1, s_2)\ \rho_1(s'_1, s'_2)) \geq 1/\alpha$。该不等式进一步意味着

$$lp(C^{sc}_j) \leq rp(C^{sc}_{j'}), \tag{10}$$

其中 $C^{sc}_j \prec C^{sc}_{j'}$。容易验证，出于相同原因，不等式 $lp(s'_1,s'_2) \leq rp(s'_1,s'_2)$ 和 $lp(C^{sc}_{j'}) \leq rp(C^{sc}_{j'})$ 均成立。最后，我们对 $j' \in [m(\mathbf{v})]$ 使用归纳法，证明该算法在前 j' 步中不会失败，并且

$$x_1(C^{sc}_{j'}) = \max\left(\{0\} \cup \{lp(C^{sc}_{j'})\} \cup \{lp(C^{sc}_j)\}_{C^{sc}_j \prec C^{sc}_{j'}}\right) \leq \min\left(1, rp(C^{sc}_{j'})\right). \tag{11}$$

归纳假设对 $j' = 1$ 直接成立，因为我们有 $x_1(C^{sc}_1) = \min I_v(C^{sc}_1)$，其中

$$I_v(C^{sc}_1) = [0, 1] \cap [lp(C^{sc}_1), rp(C^{sc}_1)].$$

接下来，假设归纳假设在 $j' < m(\mathbf{v})$ 处成立，那么对于 $C^{sc}_{j'+1}$，我们利用等式 (8) 和 $\prec$ 的传递性来证明

$$I_v(C^{sc}_{j'+1}) = [0, 1] \cap [lp(C^{sc}_{j'+1}), rp(C^{sc}_{j'+1})] \cap \bigcap_{C^{sc}_j \prec C^{sc}_{j'+1}} [lp(C^{sc}_j), +\infty).$$

然后，利用等式 (10) 和上述推导出的其他不等式，我们得到 $I_v(C^{sc}_{j'}) \neq \emptyset$，这证明算法在步骤 $j' + 1$ 不会失败，且 $x_1(C^{sc}_{j'+1})$ 满足归纳假设。至此归纳完成，我们证明了算法永远不会失败，并给出了 $x_1(C^{sc}_j)$ 的高效可计算定义。

现在我们转向证明仅当方向。通过反证法，假设存在一个 α-价值冲突对，这表明存在 $(s_1,\ s_2)$ 和 $(s'_1,\ s'_2)$，其中 $(s_1,\ s_2) \prec (s'_1,\ s'_2)$，并且

$$f(\rho_2(s_1, s_2), \rho_1(s'_1, s'_2)) < \frac{1}{\alpha}.$$

由于 $\alpha \geq 1$，我们有 $f(\rho_2(s_1,\ s_2)\ \rho_1(s'_1,\ s'_2)) < 1$，这意味着 $\rho_2(s_1,\ s_2) \neq 1$ 且 $\rho_1(s'_1,\ s'_2) \neq 1$。要实现$\alpha$-近似随机机制，需要满足

$$x_1(s'_1, s'_2) \leq rp(s'_1, s'_2) = \frac{1-1/\alpha}{1-\rho_1(s'_1, s'_2)} \quad \text{and} \quad x_1(s_1, s_2) \geq lp(s_1, s_2) = \frac{1/\alpha - \rho_2(s_1, s_2)}{1-\rho_2(s_1, s_2)}.$$

由单调性约束可知

$$\frac{1/\alpha - \rho_2(s_1, s_2)}{1 - \rho_2(s_1, s_2)} \le x_1(s_1, s_2) \le x_1(s_1', s_2') \le \frac{1 - 1/\alpha}{1 - \rho_1(s_1', s_2')},$$

这与假设（s_1， s_2）和（s_1'， s_2'）是一个 α-价值冲突对相矛盾。 □

4.2.3 二元信号的 RefinedAnalysis

回顾一下，我们可以对最优确定性比率进行二分搜索，将 DET化简为查询变体DET_γ。基于第4.1.4节中的见解，我们现在转向另一个特殊情形 $k = 2$；对于该情形，将判定问题表述为超图中的完美匹配后，可以高效求解。

定理2. 当 $k = 2$时，可以在 $N^{1+o(1)}$ 时间内求解 DET 。

证明。 为计算 R_D^*(**v**)及其对应的确定性机制，我们将运行引理3.1中的二分搜索。为了回答 $\text{DET}\gamma$ 查询，我们转而关注引理4.4。回顾一下，当且仅当以下超图存在完美匹配时，才存在一个解：

$$V := [k]^n \qquad \text{and} \qquad E := \bigcup_{i \in [n]} \{e \in E_i \mid \forall \mathbf{s} \in e, \rho_i(\mathbf{s}) \ge 1/\gamma\}$$

其中，对于所有 $i \in [n]$ ，我们有

$$E_i = \left\{ \{(s_i, \mathbf{s}_{-i}) \mid s_i \in S\} \;\middle|\; \begin{array}{l} S \subseteq [k] \text{ and } \mathbf{s}_{-i} \in [k]^{n-1} \text{ such that} \\ \forall (s_i, s_i') \in \sigma_i(\mathbf{s}_{-i}),\ s_i \in S \Rightarrow s_i' \in S \end{array} \right\}.$$

首先，我们注意到每条超边 $e \in E$ 的大小至多为 2，并将 E 划分为集合 E' （由大小为 1 的超边组成）和集合 E'' （由大小为 2 的超边组成）。由于 E 至多包含 $n \cdot 3 \cdot 2^{n-1}$ 个元素，因此可以在 $O(n2^n)$ 时间内构造 E' 和 E'' 。直观地说，(V, E'') 是一个标准的无向图，而$E' \subseteq V$ 可以看作一组允许保持未匹配状态的节点。更形式化地说，解是 (V, E'') 中的一个匹配，其中每个顶点 $\mathbf{s} \in V$ 要么被一条边 $e \in E''$覆盖，要么在 $\mathbf{s} \in E'$时可以保持未匹配状态。

现在请注意，每条边 $e \in E''$ 连接两个仅在第 i 个坐标上不同的信号剖面。若根据 $\sum_i s_i$的奇偶性对顶点 $\mathbf{s} \in [2]^n$ 进行划分，则可得 (V, E'') 是一个二分图。

有趣的是，我们可以证明，对于每条边 $\{(1, \mathbf{s}_{-i}), (2, \mathbf{s}_{-i})\} \in E''$，要么有 $(1, \mathbf{s}_{-i}) \in E'$ 要么有 $(2, \mathbf{s}_{-i}) \in E'$。事实上，若 $(1, \mathbf{s}_{-i}) \notin E'$，那么 $(1, 2) \in \sigma_i(\mathbf{s}_{-i})$；而若 $(2, \mathbf{s}_{-i}) \notin E'$，那么 $(2, 1) \in \sigma_i(\mathbf{s}_{-i})$，这将与 $\sigma_i(\mathbf{s}_{-i})$ 是严格序这一事实矛盾。

为了进一步简化结构，我们通过移除连接 E' 中两个可能各自独立的顶点的边，定义 $E''' = E'' \setminus \{\{\mathbf{s}, \mathbf{s}'\} \mid \mathbf{s}, \mathbf{s}' \in E'\}$；这不会改变解的存在性。在图 $G_\gamma = (V, E''')$ 中，每条边恰好覆盖 $V \setminus E'$ 中的一条边。

因此，当且仅当 G 存在大小为 $|V| - |E'|$的匹配时，才存在解。我们将在二分图 G中计算最大基数匹配。该图有 2^n 个顶点和 $N = O(n2^n)$条边，可以使用 [Che+25]的准线性算法在 $N^{1+o(1)}$时间内完成，或者使用更为标准的 Hopcroft–Karp–Karzanov 算法 [HK73]在 $O(N^{3/2})$时间内完成。 □

5 困难性与下界

在本节中，我们证明定理 4 和定理 5，分别给出我们所研究优化问题的时间复杂度和查询复杂度下界。在两个证明中，我们都构造具有递增价值函数的实例，因此，我们的困难性结果适用于先前工作所考虑的设定[RT16; Ede+18; Ede+19; AT21; LSZ22]。

5.1 根据性能比率构造估值函数

我们已在引理 2.1 中看到，可以找到能够诱导给定性能比率的递增价值函数（或递减成本函数）。因此，为了在本节中构造适用于单调价值（或成本）设定的困难实例，我们转而关注性能比率。在本小节中，我们进一步证明，基于满足一个简单条件的性能比率，总能构造出具有信号上的次模性（SOS）的单调价值函数（或成本函数）。因此，较弱版本的困难性结果也可以推广到 SOS 设定。

我们回顾 [Ede+19] 中信号上的次模性的定义：如果对于所有 i、$j \in [n]$、$s_j \in [k]$ 和 $\Delta \in [k]$（其中 $s_j + \Delta \le k$），以及任意满足 $\mathbf{s}_{-j}, \mathbf{s}'_{-j} \in [k]^{n-1}$ 且 $\mathbf{s}_{-j} \le \mathbf{s}'_{-j}$ 逐分量成立的情况，价值函数都满足信号上的次模性（*SOS*），则称其具有信号上的次模性：

$$v_i(s_j + \Delta, s_{-j}) - v_i(s_j, s_{-j}) \;\ge\; v_i(s_j + \Delta, s'_{-j}) - v_i(s_j, s'_{-j}).$$

类似地，我们将成本函数的信号上的次模性（SOS）定义为：

$$c_i(s_j, s_{-j}) - c_i(s_j + \Delta, s_{-j}) \;\ge\; c_i(s_j, s'_{-j}) - c_i(s_j + \Delta, s'_{-j}).$$

给定性能比率 $\boldsymbol{\rho}$，定义 $r^*(\boldsymbol{\rho}) = \min\{\rho_i(\mathbf{s}) \mid i \in [n],\ \mathbf{s} \in [k]^n\}$。

引理 5.1。给定性能比率 $\boldsymbol{\rho}$，其中 $r^*(\boldsymbol{\rho}) \ge 1 - \frac{1}{(nk)^2+1}$，可以构造具有 *SOS* 性质且诱导出 $\boldsymbol{\rho}$ 的递增价值函数（分别为递减成本函数）。

证明。注意，如果价值函数 $\{v_i\}_{i\in[n]}$ 是递增且满足 SOS，那么通过对价值函数取倒数得到的成本函数是递减的、满足 SOS，并且与价值函数产生相同的性能比率。因此，只需在价值设定下证明该引理。我们定义 $a_\ell = \ell(2nk - \ell)$ 和 $v_i(\mathbf{s}) = \rho_i(\mathbf{s})(a_\ell + 1)$，其中 $\ell = ||\mathbf{s}||_1$。容易验证，对于所有 i 和 $\mathbf{s}$，由 $v_i(\mathbf{s})$ 诱导的性能比率为 $\rho_i(\mathbf{s})$。由于 $1 - \frac{1}{(nk)^2+1} \le \rho_i(\mathbf{s}) \le 1$，我们有 $a_\ell \le v_i(\mathbf{s}) \le a_\ell + 1$。我们进一步注意到，$a_{\ell'} \ge a_\ell + 1$ if $\ell' > \ell$。因此，当 $||\mathbf{s}||_1 < ||\mathbf{s}'||_1$ 时，$v_i(\mathbf{s}) \ge v_i(\mathbf{s}')$，这意味着价值函数的单调性。为证明价值函数满足 SOS，我们注意到，当 $\ell < \ell'$ 时，$a_{\ell+\Delta} - a_\ell - 1 \le v_i(s_j + \Delta, \mathbf{s}_{-j}) - v_i(s_j, \mathbf{s}_{-j}) \le a_{\ell+\Delta} - a_\ell + 1$ and $a_{\ell+\Delta} - a_\ell \ge a_{\ell'+\Delta} - a_{\ell'} + 2$，因为 $a_{\ell+\Delta} - a_\ell = \Delta(2nk - 2\ell - t)$ 和 $\Delta \ge 1$，这表明价值函数满足 SOS。

□

5.2 NP-困难性

我们在命题 4.10 中看到，求解 DET 可以归约为判定子句大小至多为 n的公式的可满足性；当 $n = 2$ 时，这很容易，而当 $n \ge 3$时，则是 NP 困难。本节将给出反向归约，证明一般而言，计算最优确定性比率 R^*_D，甚至对其进行近似，都是 NP 困难的。为使归约更简单，我们从 1-in-3-SAT 问题出发。这是 3-SAT 的一种结构化变体，同样是 NP 困难的 [Sch78]。重要的是，我们需要有 $n = 4$ 个代理人，因为按照当前的构造，无法将任意公式嵌入仅有三个代理人的实例中。至于在 $n = 3$ 时计算 R^*_D 的复杂度，则是一个引人入胜的开放问题，我们将其留待今后研究。

定义 5.2 (1-in-3-SAT)。在 *1-in-3-SAT* 问题中，给定一个合取范式（*CNF*）布尔公式 ϕ，其中每个子句恰好由三个文字组成（即变量或其否定）。目标是判断是否存在一个变量的真值赋值，使得每个子句中恰好一个 文字为真，另外两个为假。形式化地，令

$$\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m,$$

其中每个子句 C_i 的形式为 $(\ell_{i,1}, \ell_{i,2}, \ell_{i,3})$，且每个 $\ell_{i,j}$ 都是一个文字（即变量 x 或其否定 $\neg x$）。如果存在一种真值赋值，使得对于每个子句 C_i，文字 $\ell_{i,1}, \ell_{i,2}, \ell_{i,3}$ 中恰有一个取真值，则称公式 ϕ 为*1-in-3*可满足。

定理 4。当 $n = 4$ 时，问题 $(1, \beta)$-DET 对任意 $\beta > 1$ 都是 *NP* 困难的。

证明。 在间隙问题 (1， β)-DET 中，我们需要区分确定性比率为$R_D^* = 1$ 的实例和 $R_D^* > \beta$。为方便起见，我们固定一个常数 $\varepsilon \in (0,\ 1/\beta)$。

给定 $\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$，我们将定义一个 (1， β)-DET 实例，其中有 $n = 4$ 个代理人和$k = O(m)$个信号，其规模关于 m为多项式级别。如果 ϕ 是1-in-3可满足的，那么确定性比率将等于1；否则将等于 $1/\varepsilon > \beta$。

为简单起见，我们构造一个价值函数单调的实例 $\mathbf{v}$，也就是说，对于所有 $i \in [n]$ 和 $\mathbf{s}_{-i}$，都有 $\sigma_i(\mathbf{s}_{-i}) = \{(s_i,\ s_i') \mid 1 \le s_i < s_i' \le k\}$。根据引理 2.1，只需构造性能比 $\boldsymbol{\rho}$。在我们的构造中，需要将每个性能比 $\rho_i(\mathbf{s})$ 设为 1 或 ε。

为建立一些直观认识，考虑图3中变量 a对应的构件。首先，由于性能比的缘故，如果 $R_D(\mathbf{x}\ \boldsymbol{\rho}) = 1$，那么我们有

$$x_1(k,1,1,*) = x_4(k,1,1,*) = x_2(1,2,1,*) = x_4(1,2,1,*) = x_3(1,1,2,*) = x_4(1,1,2,*) = 0.$$

如果我们决定将 $x_1(1,\ 1,\ 2,\ *) = 1$ 设为，那么

$$\begin{aligned}
x_1(1,1,2,*) = 1 \quad &\Rightarrow \quad x_1(k,\ 1,\ 2,\ *) = 1 && \text{（根据单调性）}\\
&\Rightarrow \quad x_3(k,1,2,*) = 0 && \text{（概率之和为 1）}\\
&\Rightarrow \quad x_3(k,1,1,*) = 0 && \text{（根据单调性）}\\
&\Rightarrow \quad x_2(k,1,1,*) = 1 && \text{（概率之和为 1）}\\
&\Rightarrow \quad \cdots\\
&\Rightarrow \quad x_1(1,1,2,*) = 1
\end{aligned}$$

因此，这些条件都等价；若它们成立，我们就称 a 为真。反过来，若$x_1(1,\ 2,\ 1,\ *) = 1$ 成立，我们就称 a 为假。注意，我们引入第四个代理人来设置 $x_4(1,\ 1,\ 1,\ *) = 1$，因为单调性使得代理人 1、2 和 3 无法在 (1, 1, 1, $*$) 处被选中。我们为每个变量引入一个构件，并将它们放置在不同的 s_2 和 s_3 中，以消除这些构件之间任何不必要的相互作用。坐标为 $(s_2,\ s_3 + 1)$ 的水平线称为 a-线，坐标为 $(s_2 + 1,\ s_3)$ 的水平线称为 $\neg a$-线。当 a 为真（分别为假）时，我们称 a-线为激活（分别为未激活），而 $\neg a$-线为未激活（分别为激活）。

接下来，我们使用图 4 为每个子句构建一个构件 C_i。每个子句由三个变量构件组成，每个文字对应一个 $\ell_{i,\ j}$，它们交于一个信号配置 $\mathbf{s}$，我们将其设为 $\boldsymbol{\rho}(\mathbf{s}) = (1,\ 1,\ 1,\ \varepsilon)$。因此，我们必须选出一个胜者 $j \in \{1,2,3\}$，它对应于被设为真的文字（恰好有一个文字为真）。

最后，我们使用图 5 中的异或连接构件，将每个文字 ℓ与对应的变量连接起来。当两条线通过异或连接构件连接时，它会添加一条约束，要求两条线中恰好有一条处于激活状态。重要的是，每个文字在其构件中都有两条水平线，但其中一条线没有唯一的坐标，因此无法连接。因此，我们将具有唯一坐标的水平线连接到 a 线或 $\neg a$ 线，具体取决于文字的符号以及水平线的符号。

可以验证，如果存在一种真值赋值，使得每个子句中恰有一个文字为真，那么就可以构造一个确定性机制$\mathbf{x}$ 满足 $R_D(\mathbf{x}\ \boldsymbol{\rho}) = 1$，从而证明 $R_D^*(\boldsymbol{\rho}) = 1$。更确切地说，我们对每个构件都使用该赋值，按照单调性进行传播，并对任何其他信号配置设定 $x_4(\mathbf{s}) = 1$。反之，如果 $R_D^*(\boldsymbol{\rho}) = 1$，那么我们就可以为 ϕ 构造一个满足赋值。因此，区分 $R_D^*(\boldsymbol{\rho}) = 1$ 和 $R_D^*(\boldsymbol{\rho}) > \beta$ 是 NP 困难的。 □

接下来，作为一个推论，我们要证明机制设计问题同样是困难的。直观地说，回答 DET_γ 查询与 SAT 的搜索版本相关；众所周知，利用 Cook 归约，搜索版本与其判定版本在多项式时间内等价。

推论 5.3。 假设 $P \neq NP$，则对于每个 $\gamma \ge 1$，都不可能总能在多项式时间内回答 DET_γ 查询。即使在更强的承诺下——存在一个比率为1的确定性分配规则——该 DET_γ 问题仍然是困难的。

证明。 我们通过逆否命题进行证明，并假设存在一种算法，能够在存在比率为 1 的确定性分配规则的实例上，在 $f(N) = N^{O(1)}$时间内回答 DET_γ 查询。

给定一个 $(1,\sigma)$ 的实例 (ρ)，我们在其上运行该算法，遍历所有 k^n信号配置；每次若算法尚未停止，就在运行 $f(N)$ 步后将其停止。我们检查所得分配规则 $\mathbf{x}$ 是否真实且比率为 $R(\rho,\mathbf{x}) \leq \beta$。若是，则返回 true；若出现任何其他失败情况，则返回 false。根据构造，刚才描述的过程可以在多项式时间内判定 $(1,\beta)$-DET，这意味着 $P = NP$。 □

我们注意到，当 ε 足够大时，上述困难性结果可以推广到 SOS 设定。形式上，我们有以下推论：

推论 5.4. 当 $n = 4$ 时，问题 $(1,1)$-DET 是 *NP* 困难的，即使价值（或成本）函数是单调 *SOS* 函数也是如此。

证明. 取 $\varepsilon \in (1 - \frac{1}{(nk)^2+1}, 1)$，并应用定理 4 的相同证明。根据引理 5.1，始终存在能够导出给定性能比率的单调 SOS 函数。 □

推论 5.5. 假设 *P* $\neq$*NP*，则无法总是在多项式时间内回答 DET_1 查询，即使在存在性能比为 1 的确定性分配规则，且（价值或成本）函数单调并具有 *SOS* 性质的实例上也是如此。

证明. 结合推论 5.3 的证明与推论 5.4 的结果即可。 □

5.3 查询复杂度

定理 1 至 3 给出了算法，可相对于输入 $N = nk^k b$ 的总规模，在多项式时间内求解 VAL、 CST 以及 DET 的特殊情形。这些算法可用于求解相应的机制设计问题 VAL_γ、 CST_γ 和 DET_γ，其中输出的分配规则可以在不同的信号剖面上进行评估，但需满足跨查询一致的约束。然而，人们可能会问：当我们只被要求计算 k^n 个信号剖面之一上的结果时，是否可以消除时间复杂度中对 n 的指数级依赖。

我们利用决策树模型中的查询复杂度来回答这一问题。更正式地说，我们假设我们的算法可以通过一个预言机访问输入，该预言机回答价值或成本查询。我们构造了一组实例，其中存在比率为 1 的确定性分配规则，但任何算法若不查询大部分输入，就无法在某个特定信号配置上正确计算出一个有效结果。

定理 5。对于所有固定的 $n \geq 2$ 和 $k \geq 2$，要回答某些 VAL_γ、 CST_γ 和 DET_γ 查询，至少需要对输入预言机（价值或成本）进行 $\Omega(k^n)$ 次查询。

证明。 我们将构造一组实例，其中输入价值函数均为单调递增。与定理 4 的证明类似，我们可以先指定性能比 $\boldsymbol{\rho}$，然后构造价值函数 $\mathbf{v}$，使得对于所有 $i \in [n]$ 和 $\mathbf{s}_{-i}$， $\sigma_i(\mathbf{s}_{-i}) = \{(s_i, s_i') \mid 1 \leq s_i < s_i' \leq k\}$。

我们从 DET_1 的证明开始，其主要思路如下。首先，我们固定一个特定的信号配置 $\bar{\mathbf{s}}$，并定义初始性能比 $\bar{\boldsymbol{\rho}}$。然后，我们构造两组实例 P_1 和 P_2，使其满足以下性质：

- 每个 $\boldsymbol{\rho} \in P_1 \cup P_2$ 与 $\bar{\boldsymbol{\rho}}$ 恰好在一个性能比 $\rho_i(\mathbf{s})$ 上不同，
- 对于所有 $\boldsymbol{\rho} \in P_1 \cup P_2$，都存在 $\mathbf{x} \in \mathcal{T}_D(\boldsymbol{\sigma})$，其比率为 $R(\boldsymbol{\rho},\mathbf{x}) = 1$，
- 若 $\boldsymbol{\rho} \in P_1$ 则对于所有满足 $R(\boldsymbol{\rho},\mathbf{x})$ 的 $\mathbf{x} \in \mathcal{T}_D(\boldsymbol{\sigma})$， $= 1$ 我们有 $x_1(\bar{\mathbf{s}}) = 1$，
- 若 $\boldsymbol{\rho} \in P_2$ 则对于所有满足 $R(\boldsymbol{\rho},\mathbf{x})$ 的 $\mathbf{x} \in \mathcal{T}_D(\boldsymbol{\sigma})$， $= 1$ 我们有 $x_2(\bar{\mathbf{s}}) = 1$。

那么，我们要么需要查询所有在实例取自 P_1 时与 $\bar{\boldsymbol{\rho}}$ 不同的性能比率，要么查询所有在实例取自 P_2 时与 $\bar{\boldsymbol{\rho}}$ 不同的性能比率，否则就无法计算结果$\mathbf{x}(\bar{\mathbf{s}})$。

接下来，我们的目标是构造 $\bar{\boldsymbol{\rho}}$，使得对于每个 $\mathbf{s} \in [k]^n$，至多有两个代理人 i 满足$\bar{\rho}_i(\mathbf{s}) = 1$，而其他所有代理人都满足 $\bar{\rho}_i(\mathbf{s}) = \varepsilon$。这样，我们的 DET 实例就可以理解为一个 2-SAT 公式，其变量为 $x_i(\mathbf{s})$；其中每个子句都是两个文字之间的蕴含。令 L_1 和 L_2 分别为

由 $x_1(\bar{\mathbf{s}})$ 和 $x_2(\bar{\mathbf{s}})$ 蕴含的文字集合（包括传递蕴含），并使用 $\boldsymbol{\sigma}$的单调性约束。我们将确保 L_1 和 L_2 互不相交，且两者的大小都为 $\Omega(k^n)$。这样，我们将 P_1 和 P_2 定义为这样的实例集合：它们与 $\boldsymbol{\rho}$的差异仅在于与 L_1 和 L_2中的文字相关联的变量，从而迫使相应的文字为假。

现在我们可以继续构造 $\bar{\mathbf{s}}$

$$\forall i \in [n], \qquad \bar{s}_i = \begin{cases} 1 + \lfloor k/2 \rfloor & \text{if } i \in \{1, 2\}, \\ 1 & \text{if } i \geq 3 \text{ and } i \text{ is odd}, \\ k & \text{if } i \geq 3 \text{ and } i \text{ is even}. \end{cases}$$

为了构造 $\boldsymbol{\rho}$，我们首先处理满足以下条件的 $\mathbf{s}$：对于所有 $i \geq 3$，$s_i = \bar{s}_i$：

- 如果 $s_1 \geq \bar{s}_1$，那么我们令 $\rho_2(\mathbf{s}) = 1$,
- 如果 $s_2 \geq \bar{s}_2$，那么我们令 $\rho_1(\mathbf{s}) = 1$,
- 如果 $n \geq 3$，且 $s_1 < \bar{s}_1$ 或 $s_2 < \bar{s}_2$ 中任一条件成立，那么我们令 $\rho_3(\mathbf{s}) = 1$,
- 对于其他每个 i，我们令 $\rho_i(\mathbf{s}) = \varepsilon$。

注意，当 $n = 2$ 时，代理人 1 和 2 的构造与第3.4节中的构造类似。然后，对于其余的 $\mathbf{s} \in [k]^n$，我们将 $J(\mathbf{s})$ 定义为满足 $s_j \neq \bar{s}_j$ 的最大 j，并将 $\rho_i(\mathbf{s})$ 设为 $= 1$，对所有满足 $J(\mathbf{s}) \leq i \leq J(\mathbf{s}) + 1$ 的 i 均如此。通过这一构造，可以对 $i \geq 3$ 进行归纳法证明:

- 对所有奇数 $i \geq 3$，以及所有满足 $J(\mathbf{s}) = i$ 的 $\mathbf{s} \in [k]^n$，

  – 如果 $s_1 \geq \bar{s}_1$ 且 $s_2 < \bar{s}_2$，那么 $x_i(\mathbf{s}) \in L_1$–
  如果 $s_1 < \bar{s}_1$ 且 $s_2 \geq \bar{s}_2$，那么 $x_i(\mathbf{s}) \in L_2$

- 对所有偶数 $i \geq 4$，以及所有满足 $J(\mathbf{s}) = i$ 的 $\mathbf{s} \in [k]^n$，

  – 如果 $s_1 \geq \bar{s}_1$ 且 $s_2 < \bar{s}_2$，那么 $\neg x_i(\mathbf{s}) \in L_1$–
  如果 $s_1 < \bar{s}_1$ 且 $s_2 \geq \bar{s}_2$，那么 $\neg x_i(\mathbf{s}) \in L_2$

因此，两组的大小都是 $\Omega(k^n)$。此外，可以证明，集合 $L_1 \subseteq \{\mathbf{s} \in [k]^n \mid s_1 \geq \bar{s}_1$ 和 $s_2 < \bar{s}_2\}$以及 $L_2 \subseteq \{\mathbf{s} \in [k]^n \mid s_1 \geq \bar{s}_1$ 和 $s_2 < \bar{s}_2\}$ 分别包含在信号配置集合的不同象限中，因此彼此不相交。至此，$\boldsymbol{\rho}$的构造以及 $\textsc{Det}_1$的证明就完成了。为完成定理的证明，注意到对于完全相同的实例集合:

- 如果 $\boldsymbol{\rho} \in P_1$，那么对于所有满足 $R_V(\boldsymbol{\rho}, \mathbf{x}) = 1$ 的 $\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$，都有 $x_1(\bar{\mathbf{s}}) = 1$,
- 若 $\boldsymbol{\rho} \in P_2$ 则对所有满足 $R_V(\boldsymbol{\rho}, \mathbf{x}) = 1$ 的 $\mathbf{x} \in \mathcal{T}(\boldsymbol{\sigma})$，我们有 $x_2(\bar{\mathbf{s}}) = 1$。

成本设定中的论证完全相同。这证明，要回答某些 $\textsc{Val}_1$、$\textsc{Cst}_1$ 查询，至少需要 $\Omega(k^n)$ 次查询。

□

根据引理 5.1，我们将定理 5 推广到 SOS 设定如下:

推论 5.6。对于所有固定的 $n \geq 2$ 和 $k \geq 2$，即使函数是单调 *SOS*（价值或成本）函数，要回答某些 $\textsc{Val}_1$、$\textsc{Cst}_1$ 和 $\textsc{Det}_1$ 查询，也至少需要向输入预言机（价值或成本）发出 $\Omega(k^n)$ 次查询。

证明。取 $\varepsilon \in (1 - \frac{1}{(nk)^2+1}, 1)$，并应用定理 5 的同一证明。根据引理 5.1，始终存在能够导出给定性能比率的单调 SOS 函数。 □

参考文献

[APT79] Bengt Aspvall、Michael F. Plass 和 Robert Endre Tarjan。“用于检验某些带量词布尔公式真值的线性时间算法”。载于：信息处理快报 8.3 (1979)，第 121–123 页。[Ass+20] Sepehr Assadi 等。“区分真实与非真实组合拍卖的通信复杂度”。载于：STOC。ACM，2020，第 1073–1085 页。[AT21] Ameer Amer 和 Inbal Talgam-Cohen。“具有相互依赖性和 SOS 的拍卖：改进的近似”。载于：SAGT。第 12885 卷。《计算机科学讲义》。Springer，2021，第 34–48 页。[Aus+99] Lawrence M Ausubel 等。“广义维克里拍卖”。载于：《计量经济学》(1999)。[BDR23] Moshe Babaioff、Shahar Dobzinski 和 Shiri Ron。“单交叉环境下机制设计的计算复杂度”。载于：ACM 经济与计算会议。ACM，2023，第 183 页。[Ben83] Michael Ben-Or。“代数计算树的下界”。载于：第十五届 ACM 计算理论年度研讨会论文集。1983，第 80–86 页。[BT97] Dimitris Bertsimas 和 John N. Tsitsiklis。《线性优化导论》。马萨诸塞州贝尔蒙特：Athena Scientific，1997。[CFK14] Shuchi Chawla、Hu Fu 和 Anna R. Karlin。“相互依赖价值环境中的近似收入最大化”。载于：ACM 经济与计算会议。ACM，2014，第 277–294 页。[Che+25] Li Chen 等。“近线性时间内的最大流和最小费用流”。载于：《ACM 杂志》72.3 (2025)，19:1–19:103。[Cor+09] Thomas H. Cormen 等。《算法导论》。第 3 版。麻省理工学院出版社，2009。[DD13] Shahar Dobzinski 和 Shaddin Dughmi。“随机化在算法机制设计中的力量”。载于：《SIAM 计算杂志》42.6 (2013)，第 2287–2304 页。[DDT14] Constantinos Daskalakis、Alan Deckelbaum 和 Christos Tzamos。“最优机制设计的复杂度”。载于：离散算法研讨会。工业与应用数学学会，2014，第 1302–1318 页。[DM00] Partha Dasgupta 和 Eric Maskin。“高效拍卖”。载于：《经济学季刊》115.2 (2000)，第 341–388 页。[Dob11] Shahar Dobzinski。“真实次模估值组合拍卖的不可能性结果”。载于：STOC。ACM，2011，第 139–148 页。[Dob16] Shahar Dobzinski。“计算效率要求简单征税”。载于：计算机科学基础研讨会。IEEE 计算机学会，2016，第 209–218 页。[DV12] Shahar Dobzinski 和 Jan Vondr´ak。“组合拍卖中真实性的计算复杂度”。载于：ACM 经济与计算会议。ACM，2012，第 405–422 页。[Ede+18] Alon Eden 等。“没有单交叉条件的相互依赖价值”。载于：ACM 经济与计算会议。ACM，2018，第 369 页。[Ede+19] Alon Eden 等。“具有相互依赖估值的组合拍卖：SOS 助力解决问题”。载于：ACM 经济与计算会议。ACM，2019，第 19–20 页。[Ede+23] Alon Eden 等。“私人相互依赖估值的常数近似”。载于：计算机科学基础研讨会。IEEE，2023，第 148–163 页。[Ede+24] Alon Eden 等。“私人相互依赖估值：单物品拍卖与拟阵的新下界”。载于：ACM 经济与计算会议。ACM，2024，第 448–464 页。[EGZ22] Alon Eden、Kira Goldner 和 Shuran Zheng。“私人相互依赖估值”。载于：离散算法研讨会。工业与应用数学学会，2022，第 2920–2939 页。[Fel+22] Michal Feldman 等。“贝叶斯随机时钟拍卖”。载于：ACM 经济与计算会议。ACM，2022，第 820–845 页。[Fel+25] Michal Feldman 等。“具有相互依赖价值的在线组合分配”。载于：ACM 经济与计算会议。ACM，2025，第 189–205 页。

[Gka+21] Vasilis Gkatzelis 等。“面向具有相互依赖价值的竞标者的无先验时钟拍卖”。载于：SAGT。第 12885 卷。《计算机科学讲义》。Springer，2021 年，第 64–78 页。[HK73] John E. Hopcroft 和 Richard M. Karp。“二分图最大匹配的 n5/2 算法”。载于：《SIAM 计算杂志》2.4（1973 年），第 225–231 页。[JM01] Philippe Jehiel 和 Benny Moldovanu。“相互依赖估值下的有效设计”。载于：《计量经济学》69.5（2001 年），第 1237–1259 页。[Kar84] Narendra Karmarkar。“一种用于线性规划的新型多项式时间算法”。载于：第十六届 ACM 计算理论年度研讨会论文集。1984 年，第 302–311 页。[Kha79] Leonid G Khachiyan。“线性规划中的多项式算法（英文译文）”。载于：《苏联数学文献》第 20 卷。1979 年，第 191–194 页。[Kri09] Vijay Krishna。《拍卖理论》。Academic Press，2009 年。[LSZ22] Pinyan Lu、Enze Sun 和 Chenghan Zhou。“相互依赖 SOS 估值的更优近似”。载于：WINE。第 13778 卷。《计算机科学讲义》。Springer，2022 年，第 219–234 页。[Mas96] Eric S Maskin。“拍卖与私有化”。载于：《私有化：世界经济的批判性视角》（1996 年），第 433–453 页。[Mil79] Paul R Milgrom。“具有差异信息的竞争性投标的收敛定理”。载于：《计量经济学：计量经济学会杂志》（1979 年），第 679–688 页。[MMR24] Simon Mauras、Divyarthi Mohan 和 Rebecca Reiffenh¨auser。“具有相互依赖价值的最优停止”。载于：EC。ACM，2024 年，第 246–265 页。[MW82] Paul R Milgrom 和 Robert J Weber。“拍卖与竞争性投标理论”。载于：《计量经济学：计量经济学会杂志》（1982 年），第 1089–1122 页。[Mye81] Roger B Myerson。“最优拍卖设计”。载于：《运筹研究数学》6.1（1981 年），第 58–73 页。[Nis+07] Noam Nisan 等，编。《算法博弈论》。剑桥大学出版社，2007 年。[NR99] Noam Nisan 和 Amir Ronen。“算法机制设计”。载于：STOC。ACM，1999 年，第 129–140 页。[RT16] Tim Roughgarden 和 Inbal Talgam-Cohen。“相互依赖价值下的最优与稳健机制设计”。载于：《ACM 经济与计算汇刊》4.3（2016 年），18:1–18:34。[RZ21] Aviad Rubinstein 和 Junyao Zhao。“随机拍卖的随机通信复杂度”。载于：STOC。ACM，2021 年，第 882–895 页。[Sch78] ThomasJ. Schaefer。“可满足性问题的复杂度”。载于：STOC。ACM，1978 年，第 216–226 页。[Sch98] Alexander Schrijver。《线性与整数规划理论》。John Wiley & Sons，1998 年。[Szp30] Edward Szpilrajn。“关于偏序的扩展”。载于：《数学基础》16（1930 年），第 386–389 页。[Tar72] Robert Endre Tarjan。“深度优先搜索与线性图算法”。载于：《SIAM计算杂志》1.2（1972 年），第 146–160 页。[Tar86] ´Eva Tardos。“求解组合线性规划的强多项式算法”。载于：《运筹学》34.2（1986 年），第 250–256 页。[Vic61] William Vickrey。“反投机、拍卖与竞争性密封投标”。载于：《金融学杂志》16.1（1961 年），第 8–37 页。[Wil69] Robert B Wilson。“致编辑的通信——具有不同信息的竞争性投标”。载于：《管理科学》15.7（1969 年），第 446–452 页。