arXiv:2601.07279v1 [cs.GT] 12 Jan 2026

联盟策略：议会选举中的贿赂与控制

Hodaya Barr、Eden Hartman、Yonatan Aumann、Sarit Kraus

[a]巴伊兰大学，以色列

摘要

选举的策略性操纵通常是在推动个人候选人的背景下进行研究的。然而，在议会选举中，关注点发生了转移：相比于单个政党的席位数，选民可能更关心整体执政联盟。本文研究这一新问题：以提升政党联盟的集体席位数为目标，操纵议会选举。我们聚焦于比例代表制选举，并考察该问题的两种变体；其中一种的唯一目标是最大化目标联盟所持有的席位总数，另一种则具有双重目标，即既推动该联盟，又提升联盟内部某个偏好政党的相对权力。

我们研究两种策略操纵类型：贿赂，即允许修改选民偏好；以及控制，即允许改变选民和政党的集合。我们考察多种贿赂类型：对其中一些类型给出多项式时间算法，同时证明另一些类型具有NP难性。对于控制问题，我们为通过增加和删除选民进行的控制提供多项式时间算法。相比之下，对于通过增加和删除政党进行的控制，我们表明，该问题要么不可能实现（即该问题对控制免疫），要么在计算上十分困难；特别是，当以可以增加或删除的政党数量为参数时，该问题是W[1]-困难的。

关键词：
投票、控制、贿赂、议会选举

1. 引言

通过贿赂与控制进行的策略性选举操纵已得到广泛研究（例如，参见 [1, 2, 3, 4, 5, 6, 7,8, 9, 10, 11]）。这方面的大量文献聚焦于单一获胜者选举，例如总统选举和国会议员席位选举。然而在实践中，全球许多国家实行采用比例代表制 （或其他形式）的议会制。在此类制度中，多 个政党当选并共同在议会中运作。此外，在许多情况下，甚至可以说在大多数情况下，没有任何单一政党拥有独自执政所需的足够权力，议会的运作依赖于由多个政党组成的联盟 。任何试图操纵多党议会选举的行为，都必须考虑此类政党与联盟动态。事实上，如果某个政党最终未能进入执政联盟，支持该特定政党可能就意义不大。

因此，现有的贿赂与控制模型并不适用于议会选举，因为它们忽视了议会政治丰富的动态。颇为有趣的是，尽管议会制在全球各地十分普遍，关于贿赂与控制的大量文献却基本忽略了这些制度。本文迈出了填补这一空白的第一步。

我们的贡献。 本文提出并研究了在议会选举中通过策略操纵来促成政党联盟，重点关注贿赂 ——即允许修改选民偏好——以及控制 ——即允许改变选民和政党集合。我们的首要贡献是对模型和问题作出形式化定义，并引入该问题的两种变体。第一种变体中，操纵者的唯一目标是促成整个政党联盟。第二种变体则考虑到操纵者可能也希望促成某个特定政党，因此定义了一个双目标问题——既要促成整个联盟，也要促成其中某个特定的偏好政党。我们始终聚焦于比例代表制选举，在这种选举中，每位选民为一个政党投一张选票。

在定义了问题之后，我们研究了各种情形下贿赂和控制的复杂性，并给出了算法解和困难性结果。

对于贿赂，我们证明其复杂性取决于贿赂的成本结构。对于 1 和 $ 成本结构 [5]，该问题可在多项式时间内求解。对于 Swap 和 Shift 成本结构 [2]，若不存在选举门槛，该问题仍可在多项式时间内求解；但引入选举门槛后，该问题将变为 NP困难。表 2 总结了贿赂问题的结果。

对于控制，我们同时考察选民控制（添加或删除选民）和政党控制（添加或删除政党）。我们证明，选民控制可在多项式时间内求解。对于政党控制，我们证明，少数变体免于操纵（即此类操纵无法实现操纵者的目标），而其他所有变体均已证明是困难的。鉴于此，我们还考察了该问题的参数化复杂性，其中参数是要添加或删除的政党数量。我们证明，即使进行参数化分析，这些困难变体仍然是困难的，属于 W[1]-困难或 W[2]-困难。表 3 总结了政党控制问题的结果。

文章结构。 第 2 节介绍模型和定义。第 3 节聚焦于贿赂。第 4 节讨论选民控制。第 5 节研究政党控制。第 6 节总结并提出一些未来工作方向。为提高可读性，对于某些结果，我们仅在论文正文中给出直观的证明概要，并在附录中提供完整证明。

1.1. 相关工作

单一获胜者选举。 在单赢家设定中，策略性操纵（例如贿赂和控制）的复杂性已得到广泛研究。贿赂问题由 Faliszewski 等人 [5], 提出，此后受到广泛关注（例如 [12, 13, 14, 15, 16, 17]）。控制问题最早由 Bartholdi III 等人 [12],研究，也同样成为大量研究的主题（例如 [18, 19, 20, 21, 22]）。关于这两方面的全面综述，参见 Faliszewski 和 Rothe [4] 。

在单赢家设定中，目标是确保某个特定候选人获胜，而这里的目标是推动一组 候选人（政党）获胜。这一区别增加了问题的难度，因为提升联盟总支持度的方式可能更多。要确定是否存在成功的操纵，可能需要考察大量可能性。

多获胜者选举。 Meir 等人 [23], 最早提出了多获胜者选举中的策略操纵与控制，此后这一领域得到了广泛研究（例如， [24, 25, 26, 27, 10, 28]）。在多获胜者模型中，选举结果是一个获胜候选人集合，而非单一赢家。操纵的目标是促成该获胜集合具备某种性质，例如使其与某个目标集合一致 [24], 、使获胜候选人的效用总和超过某个阈值 [23], 、使获胜集合与某个目标集合的交集最大化 [10] （以及其他相关目标），或确保某个非目标候选人集合无人获胜 [28]。

我们的设定有着根本性的不同。在我们的设定中，不存在赢家。相反，所有候选人（达到选举门槛的）都会在议会中获得席位，但席位数量与其得票数成比例；而操纵的目标与这些比例有关——也就是说，联盟和偏好政党获得足够的代表权。因此，这些模型无法直接比较。

加权投票博弈。议会选举可以被视为加权投票博弈的一种特殊情况 [29],，而关于策略性操纵权力指数（如 Shapley-Shubik指数或 Banzhaf指数）的研究日益增多——例如 [30, 31, 32, 33, 34]。尽管两者相关，我们的设定却有着根本性的不同。在传统的加权投票博弈中，关注点在于事后的联盟形成。用我们的术语来说，每个政党拥有固定的选票数，而策略操纵（如假名拆分）旨在增强某个政党在联盟形成过程中的权力（例如，争取获得更多关键部长职位）。然而，在我们的设定中，我们假定联盟已通过已知的政治协议预先固定。目标是增加偏好联盟的总得票数（例如，取得议会多数并组建执政政府）。

议会选举与策略性投票。据我们所知，Slinko 和 White [35] 最早研究了议会选举中的策略行为。不过，他们关注的是策略性投票。与我们的研究不同，这一领域的研究分析选民个人激励，以及选民采取策略行动时所产生的均衡行为。关于议会选举中策略性投票的研究文献十分丰富，例如 [36, 37, 38, 39, 40]。

推动单一政党。与我们的研究最接近的是 Put 和 Faliszewski 的工作 [20], , 他们同样研究了议会选举中各种策略操纵的计算复杂性。不过，他们的分析着眼于推动一个单一政党，而我们的目标是推动整个联盟。从这个意义上说，他们研究的问题可以视为我们问题的一个特例，即联盟仅由一个政党组成。这一区别再次带来了组合上的挑战：推动联盟时，影响总支持度的可能方式会多得多。因此，我们需要采用额外的技术。我们研究了他们论文中讨论的所有操纵方式，以及其他一些操纵方式。关于进一步讨论，我们建议参阅他们的相关工作一节。

2. 模型

一次议会选举是一个三元组$E := (P, V, \succ)$，其中$P := \{p_1, \ldots, p_m\}$是政党的集合，$V := \{v_1, \ldots, v_n\}$是选民的集合，而$\succ := (\succ_{v_1}, \ldots, \succ_{v_n})$是选民对$P$的严格偏好排序 。我们用$\mathrm{Pos}^E(v_i, p)$表示按照$\succ_{v_i}$，政党$p$ 的位置；用$Top^E(v_i)$表示按照$\succ_{v_i}$最偏好的政党。只要上下文含义明确，我们就将 $\succ_i$ 简写为$\succ_{v_i}$。

每位选民贡献一张单票，投给其在E这场选举中最偏好的参选政党。分配给政党p 的选票总数记作$N^E(p)$。政党p 获得的选票占选票总数的比例记作$\mathrm{frac}^E(p) := N^E(p)/n$。

选举门槛。为了避免议会中出现小党，许多选举制度会设置一个阈值 τ，称为选举门槛，使得只有满足 $\mathrm{frac}^p \geq \tau$ 的政党才能在议会中获得代表席位。这些政党称为活跃政党，其集合记为 P_A。投给活跃政党的选票称为有效选票。

最后，政党p 获得的有效选票占比为：

$$\mathrm{frac}_\tau^E(p) := \begin{cases} \dfrac{N^E(p)}{\sum_{p' \in P_A} N^E(p')} & \text{if } p \in P_A \\ 0 & \text{otherwise} \end{cases}$$

函数 $\mathrm{frac}_\tau^E(\cdot)$ 自然地扩展到集合：$\mathrm{frac}_\tau^E(A) = \sum_{p \in A} \mathrm{frac}_\tau^E(p)$。

2.1. 操纵

外部操纵者可以将原始选举 E 修改为另一个选举 $\hat{}E := (\hat{}P, \hat{}V, \hat{}\succ)$，具体可进行哪些修改将在相应章节中详细说明。在贿赂的语境中，这位操纵者称为行贿者；在控制的语境中，则称为主席。

我们假设存在一个政党集合C （这些政党实际上是否参加选举 E 均可），称之为联盟。操纵者的目标是提高联盟所获有效选票的占比，也可能提高其中某个偏好政党所获有效选票的占比。具体而言，操纵者旨在构造一个经过修改的选举实例 $\hat{}E = (\hat{}P, \hat{}V, \hat{}\succ)$，使其满足以下一个或两个目标：

定义 1 （联合目标）。 给定联盟目标比例 $\varphi \in$ （0, 1], 目标是联盟政党获得的有效选票比例至少为 φ:

$$\mathrm{frac}_\tau^{\hat{E}}(C) \geq \varphi \tag{OBJ-J}$$

定义 2 （偏好政党目标）。 对于偏好政党 $p_1 \in C$, 以及目标偏好政党比例 $\rho \in [0, 1]$, 目标是 p_1,获得的选票占联盟选票的比例至少为 ρ:

$$\mathrm{frac}_\tau^{\hat{E}}(p_1) \geq \rho \cdot \mathrm{frac}_\tau^{\hat{E}}(C) \tag{OBJ-F}$$

我们分别考虑两种情形：操纵者旨在满足两个目标的情形，记为**J+F**；以及操纵者旨在仅满足联合目标的情形，记为**J**。

显然，**J** 是**J+F** 的一个特例（通过设置 $\rho = 0$）。因此，前者的任何困难性结果都意味着后者也具有困难性，而后者的任何多项式时间算法同样适用于前者。

比例目标。 在我们的模型中，操纵者的目标完全以分配给联盟和偏好政党的有效选票比例 来表示。现实中的议会制及其选举往往要复杂得多。

例如，在德国，德国联邦议院的299个席位通过单一赢家地区选举产生，另外299个席位则通过全国比例代表制（采用圣拉盖–舍佩斯法）确定。随后，额外的 席位——即所谓的平衡席位 ——会被分配，以纠正全国得票占比与席位数量之间的任何不成比例现象。在韩国，253名议员通过单一赢家选区选举产生，另有47名议员通过全国比例代表制选出；后者又分为17个“传统”比例代表制席位和30个“补偿性”席位。因此，每个国家都有其特定的选举机制。

本文暂不考虑这些制度细节（包括具体的席位分配方法），并以选票比例作为议会权力的替代指标。我们承认这是一种简化，但认为，对于本文的分析目的而言，这一指标足够准确。我们还指出，研究表明，在杰斐逊–顿特席位分配法下，席位占比已被证明是得票占比的仿射函数，这一点既有实证依据，也有理论依据 [41, 42]。

命名 规则。问题采用以下格式表示：⟨类型|子类型|目标|阈值⟩，其中 ：

- 类型：**B** 代表贿赂，**C**代表控制。
- 子类型：在相应章节中定义。
- 目标：J 或 J+F。
- 阈值： $\tau = 0$ 或 $\tau \geq 0$。

例如， ⟨B|$|J+F|$\tau \geq 0$⟩ 表示贿赂问题，子类型为 $，目标分别为 (OBJ-J) 和 (OBJ-F)，选举门槛为 $\tau \geq 0$。

¯记号。 我们用 $C_{-1} = C \setminus \{p_1\}$ 表示不包含偏好政党的联盟，用 $C = P \setminus C$表示反对派。此外，我们用 $T := \lceil \tau \cdot |V| \rceil$表示达到选举门槛所需的最少选票数。

示例。 例如，考虑一场有 5 个政党 $P = \{p_1, c_1, c_2, o_1, o_2\}$参与的选举，其中联盟为 $C = \{p_1, c_1, c_2\}$。75 名选民可按其偏好排序分为五种类型，如下所示：

Type	1	2	3	4	5
首选	p_1	c_1	o_1	o_2	o_1
第二选择	o_1	o_1	c_1	c_2	p_1
选民人数	20	10	20	20	5

假设选举门槛为 $\tau = 15\%$，联盟目标比例和目标偏好政党比例均为 $\varphi = \rho = 50\%$。那么，在没有操纵的情况下，联盟中只有 p_1 拥有有效选票，且 $\text{frac}_\tau^E(C) = \text{frac}_\tau^E(p_1) \approx 31\%$，低于所需的 50% 目标——见表1第一行。

假设操纵者可以移除一个（且只能移除一个）政党。表1总结了移除不同政党所产生的影响。注意，p_1 从移除o_1, 中获益最大，因为这会使其份额增至33%。移除该政党也会使联盟份额增至73%。然而，这会使p_1在联盟中的相对份额降至45%，低于要求的50%。另一方面，移除c_1 或c_2, , 可使p_1 在联盟中的份额达到100%，但联盟的总体份额为 $< 50\%$。只有移除o_2 才能达到恰当的平衡：整个联盟的份额为62%，而p_1 占其中的50%，符合要求。

		p_1	c_1	c_2	o_1	o_2	$\cdot\text{frac}_{\tau}^{E}(p_1)$	$\cdot\text{frac}_{\tau}^{E}(C)$	$\frac{\hat{\text{frac}}^{E\tau}(p_1)}{\hat{\text{frac}}^{E\tau}(C)}$	OBJ-J	OBJ-F
未操纵	选票	20	10	0	25	20	31%	31%	100%	×	✓
	有效选票	20	0	0	25	20					
移除o_1	选票	25	30	0	0	20	33%	73%	45%	✓	×
	有效选票	25	30	0	0	20					
移除o_2	选票	20	10	20	25	0	31%	62%	50%	✓	✓
	有效选票	20	0	20	25	0					
移除c_1	选票	20	0	0	35	20	27%	27%	100%	×	✓
	有效选票	20	0	0	35	20					
移除 c_2	选票	20	10	0	25	20	31%	31%	100%	×	✓
	有效选票	20	0	0	25	20					

表1：示例：移除政党的影响。选举门槛为 $\tau = 15\%$，联盟目标比例和目标偏好政党比例均为 $\varphi = \rho = 50\%$。

3. 贿赂

贿赂只允许操纵者改变选民的偏好排序。因此，$\hat{}P = P$，且$\hat{}V = V$。将 $\succ_i$ 修改为$\hat{}\succ_i$ 会产生成本（仅当 $\succ_i \neq \hat{}\succ_i$时成本为正），而操纵者拥有预算$B \geq 0$。

严格来说，我们将贿赂等同于由此产生的修改后偏好顺序序列 $\hat{\succ} = (\hat{}\succ_{v_1}, \ldots, \hat{}\succ_{v_n})$。将选民$v_i$的偏好顺序从 $\succ_i$ 改为 $\hat{}\succ_i$ 的成本为 $Cost_i(\succ_i, \hat{}\succ_i)$。整个贿赂的成本为 $Cost(\succ, \hat{}\succ) = \sum_{i=1}^{n} Cost_i(\succ_i, \hat{}\succ_i)$。根据上下文明确时，我们省略原始顺序 $\succ_i$，写作 $Cost_i(\hat{}\succ_i)$。问题的子类型如下所述。

判定问题。核心判定问题是：

议会选举中的贿赂

输入： (P, V, $\succ$, B,成本, O)，其中：　P：政党，　V：选民，　$\succ$：偏好，　B：预算
成本：成本函数，　O：目标（J 或 J+F）

输出： 是否存在贿赂 $\hat{\succ}$，其成本$(\succ, \hat{\succ}) \leq B$使行贿者的目标 O 在选举中实现 $E' = (P, V, \hat{\succ})$?

3.1. 贿赂类型

文献中讨论了几种不同的贿赂类型，它们的区别既在于允许对选民偏好进行哪些更改，也在于为此付出的成本。本文讨论以下四种贿赂类型：

1-贿赂：
选民偏好排序的所有变更均被允许。所有选民进行任何变更的成本均为固定且相同的；也就是说，对于所有成本$_i^1(\succ_i, \hat{}\succ_i) = 1$ 以及$i, \succ_i, \hat{}\succ_i$，该条件均成立。

$-贿赂：
所有变化均被允许，成本取决于选民。也就是说，对于每个v_i，都存在一个常数值 π_i，使得对于任意 $\succ_i$ 和$\hat{}\succ_i$ [5]，$Cost^{\$}{}_i(\succ_i, \hat{}\succ_i) = \pi_i$。

交换贿赂：

所有变化均被允许。成本取决于偏好排序 [2]中交换顺序的政党对。具体而言，对于选民v_i 和政党p、q，将 q 从p 下方的位置移至p 上方的位置（在v_i的排序中）需要支付成本$sw_i(p,q)$。对于$\hat{\succ}_i$，总成本为以下各项之和：

$$Cost_i^{swap}(\succ_i, \hat{\succ}_i) = \sum_{(p,q)\,:\, p\succ_i q \text{ and } q\hat{\succ}_i p} sw_i(p,q)$$

联盟移位贿赂：

在单一胜者设定中，移位贿赂只允许将偏好的候选人向上移动。成本是关于该候选人移动位置数量的单调函数 [2]。我们将此定义调整为联盟设定。在联盟移位贿赂 中，只有C 联盟中的政党可以被移动，且只能向上移动（也允许交换两个联盟政党）。也就是说，若$p \succ_i q$ 且$q \hat{\succ}_i p$，则$q \in C$。成本为：

$$Cost_i^{shift}(\succ_i, \hat{\succ}_i) := s_i(|\{(p,q) : p \succ_i q \text{ and } q \hat{\succ}_i p\}|)$$

其中s_i是某个单调函数，取决于i。

3.2. 贿赂：结果总结

结果总结于表 2 中。对于 1-贿赂和 $-贿赂，所有变体均为多项式时间可解（定理 3.1）。对于交换贿赂和联盟移位贿赂，其复杂性取决于选举门槛；当 $\tau = 0$时，这些问题为多项式时间可解（定理 3.2）；但当 $\tau > 0$ 时，这些问题为 NP困难（定理 3.4 和定理 3.3）。

贿赂类型		**J目标**	**J+F 目标**
1		P (定理 3.1*)	P (定理 3.1*)
$		P (定理 3.1*)	P （定理 3.1）
Swap	$\tau = 0$ P	（定理 3.2*）	P （定理 3.2）
	$\tau > 0$	NP困难（定理 3.4）	NP困难（定理 3.4*）
联盟移位	$\tau = 0$ P	(定理 3.2*)	P (定理 3.2)
	$\tau > 0$	NP困难 (定理 3.3)	NP困难 (定理 3.3*)

表 2：贿赂：复杂度结果摘要。括号内为相应定理。星号表示该结果直接源自定理。

3.3. 1-贿赂和*$*-贿赂

本节将证明，即使行贿者的目标是 J+F，1-贿赂和 $-贿赂问题也能在多项式时间内求解。注意，这意味着所有其他变体也都能在多项式时间内求解。

定理 3.1。 $\langle B|\$|J\text{+}F|\tau \geq 0\rangle$ 可在多项式时间内求解。

完整证明见附录（参见算法 1）。

证明概要。 由于在 1-贿赂和 $-贿赂中，任何变更都是允许的，且每位选民的成本固定，因此可以分三步处理该问题：(1) 遍历所有可能被贿赂的选民集合；(2) 计算针对该集合的最优贿赂方案；(3) 检查该贿赂方案是否实现了行贿者的目标。

我们证明两点：第一，利用动态规划算法，我们可以高效地考察所有可能被贿赂的相关选民集合；第二，对于每个这样的集合，都能在多项式时间内计算出最优贿赂。如果至少有一种可行贿赂能够实现行贿者的目标，我们就断言答案为“是”；否则，答案为“否”。 □

3.4. 交换与联盟转移贿赂，$\tau = 0$

在没有选举门槛的情况下，即使行贿者的目标是 J+F，交换与联盟转移贿赂也能在多项式时间内求解。

定理 3.2。 $\langle B|$交换$|J+F|\tau=0\rangle$和 $\langle B|$联盟移位$|J+F|\tau=0\rangle$均可在多项式时间内求解。

完整证明见附录。

证明概要。 我们说明，可以通过求解$O(n^2)$个最小费用流（*MCF*）问题实例来判定该问题。该问题（粗略地）定义如下：给定一个流网络，其中每条边都有成本和容量，以及一个需求量d，目标是找到以最低成本将d单位的流从源点s 发送到汇点t的可行方案。MCF可在多项式时间内求解（例如参见 [43]）。

其思路是检查偏好政党p_1, 不包含该政党的联盟C_{-1}以及¯反对派C 之间满足$J + F$ 目标的所有可能选票分配，并检查行贿者能否通过求解如下的最小费用流实例实现这种分配。

该流网络如图 1 所示。流量代表选民人数。第二层（即源点之后的一层）的节点代表投给 p_1, C_{-1} 和C 的选票数。从s到这些节点的边的成本为 0，其容量等于在这种划分下各组所需的选民人数（因此，如果这些边都达到饱和，我们就能得到恰好符合要求的选票划分）。

第三层的节点代表选民真实的偏好排序。从第二层到第三层的每条边的成本为 0，容量为 1（因此，每单位流量都对应一名特定选民投票给一个特定组）。

第四层的节点代表贿赂后选票所体现的偏好，而从第三层到第四层的每条边代表改变这名特定选民的偏好，使其符合要求所需的成本。成本由贿赂成本函数决定，容量为 1。

最后，第四层中的每个节点都通过一条成本为 0、容量为 1 的边连接到汇点 t 。

总需求量为 n，对应所有选民。如果所得流的最小成本不超过行贿者的预算 B，则可以实现所需结果；否则无法实现。 □

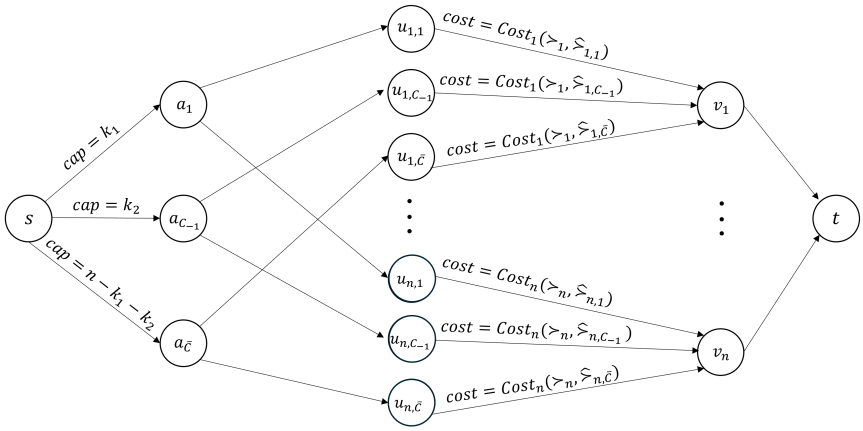

图 1: 纳什均衡 除非另有说明，默认值为成本 = 0 和容量 = 1。容量 表示容量 .

3.5. 交换与联盟移位贿赂： $\tau > 0$

在本节 tion 我们证明，当 $\tau > 0$ 时，所有交换贿赂和联盟移位贿赂的变体都是 NP困难的 rd.

定理 3.3. $\langle B|$联盟移位$|J|\tau\geq 0\rangle$是 *NP*困难的，即使所有 $s_i(x)$*'s* 都是乘法型的。

这意味着 ⟨B|联盟移位|J+F|$\tau \geq 0$⟩ 也是 NP困难的。证明通过从 3-4 精确覆盖问题进行归约得出。

证明。证明通过从 3-4 精确覆盖问题进行归约得出：给定一个集合 Z，以及由 $\mathcal{D}$ 个 Z 的 4 元子集组成的集合族，其中任意 $z \in Z$ 恰好出现在 $\mathcal{D}$ 中的 3 个子集中；判断是否存在 $\mathcal{D}$ 的一个子集族，使得 Z 中的每个元素恰好出现一次。

我们证明，给定一个 3-4精确覆盖问题的实例，可以构造一个 ⟨B|联盟移位|$\tau \geq 0$⟩ 问题的实例，使得当且仅当存在一个覆盖时，才存在一次成功的贿赂。

该实例 $E = (P, V, \succ)$ 的定义如下。政党集合 P 中，对于每个子集，都包含一个政党 p_i for for any subse t $D_i \in \mathcal{D}$；以及 3|Z| 其他政党：

$$P := \{p_1, p_2, \ldots p_{|\mathcal{D}|},\ p_{|\mathcal{D}|+1}, \ldots p_{(|\mathcal{D}|+3|Z|+1)}\}$$

共有 $2|Z|$ 名选民，每个元素 $z_i \in Z$ 对应 2 名选民，其偏好如下 :

- 1 类型 1 的选民：$p_{|\mathcal{D}|+1}$ 排在第一位，接着是对应于包含z_i的 3 个子集的 3 个政党，然后是针对 $|\mathcal{D}| + 2 \leq j \leq |\mathcal{D}| + 3|Z| + 1$ 的政党p_j，最后是所有其他政党。
- 1 类型 2 的选民：政党$p_{|\mathcal{D}|+1}$ 排在第一位，接着是针对 $|\mathcal{D}| + 2 \leq j \leq |\mathcal{D}| + 3|Z| + 1$ 的政党p_j，最后是所有其他政党。

设联盟为 $C := \{p_1, \ldots p_{|\mathcal{D}|}\}$，偏好政党为 p_1。

回顾一下，贿赂类型为联盟移位。设每位选民的价格函数为 $s_i(x) := x$；预算为$B := 3|Z|$。

目标支持度为 $\varphi := \frac{1}{2}$，选举门槛为 $\tau := \frac{4}{2|Z|}$。 —

(⇒) 我们假设存在一个精确覆盖 $\mathcal{D}$，并证明存在一种成功的贿赂。我们证明，贿赂所有类型 1 的选民，让他们投票给与 $\mathcal{D}'$ 中覆盖相应元素 z_i 的子集相对应的政党，是可行且成功的。

首先，这种贿赂是可行的，因为贿赂这些选民中每个人的成本至多为 3（与包含相应元素的集合相对应的政党排名均在前 4 位）；该贿赂涉及 $|Z|$ 名选民，预算为 $3|Z|$。

其次，这种贿赂是成功的，因为贿赂之后：(1) 反对党 $p_{|\mathcal{D}|+1}$ 获得所有类型 2 选民的选票（$|Z|$ 张选票），且由于 $\frac{1}{2} \geq \frac{2}{|Z|}$，这些选票全部有效；(2) 每个与 $\mathcal{D}'$ 中某个子集相对应的联盟政党，恰好从与该子集中元素对应的 4 名类型 1 选民那里获得 4 张选票，因为 $\tau := \frac{4}{2|Z|}$，且共有 $2|Z|$ 名选民，4 张选票足够，并且这些选票全部有效；(3) 所有其他政党都没有获得选票。联盟获得了至少一半的有效选票，这意味着对于 $\varphi = \frac{1}{2}$，联盟目标得以满足。

(⇐) 我们假设存在一次成功的贿赂，并证明存在一个精确覆盖 $\mathcal{D}$。

考虑任意 $p_i \in C$。贿赂之前，它没有获得任何选票。由于他们之前有 $3|Z| + 1$ 个政党，因此不可能贿赂类型 2 的选民，让他们投票给联盟；每次这样的改变都会使成本超过预算 $3|Z|$。注意，这种移位贿赂只允许我们提升联盟政党的排名。因此，$p_{|\mathcal{D}|+1}$ 获得 $|Z|$ 张选票（来自所有类型 2 的选民）。

为了让联盟获得至少 1/2 的有效选票，$c \in C$必须获得 $|Z|$ 张有效选票。由于有 $2|Z|$ 名选民和 $\tau = \frac{4}{2|Z|}$，只有获得至少 4 张选票的政党才能通过选举门槛。这意味着，只有当集合 D_i 中对应各元素的 4 名类型 1 选民都改变其偏好顺序，将p_i 排在首位时，政党p_i 才会进入议会。注意，只有当子集包含z_i 时，才能贿赂一名类型 1 的选民v_i，使其投票给对应的联盟政党（因为其他政党前面有 $3|Z| + 4$ 个来自C的政党）。

为了满足 $\sum_{c \in C}$(分数 $E_\tau(c)) \geq \frac{1}{2}$，必须有来自 P 的 n_4 个政党获得 4 张选票。这些 n_4 个政党对应于构成精确覆盖的 n_4 个子集。□

定理 3.4. ⟨B|交换|$\tau \geq 0$⟩ 是 NP困难。

证明。 给定一个带乘法型s_i的联盟位移贿赂实例，我们将其归约为一个交换贿赂实例。对于选民v_i 以及x、$y \in P$，定义 $sw_i(x, y) = (s_i)'$，当$y \in C$时（其中$(s_i)'$是s_i的斜率）；否则，定义$sw_i(x,y) = B+1$。后一种定义确保只考虑允许的交换，前一种定义则确保成本保持不变。 □

4. 选民控制

选民控制问题允许主席只改变选民集合。此处，政党集合保持不变，即$\hat{P} := P$，而选民集合则修改为$\hat{V}$（如下所述）。偏好为$\hat{\succ} := (\succ_v)_{v\in\hat{V}}$——它们经过调整，仅反映$\hat{V}$中的选民（操纵者不能改变选民偏好）。主席的预算为$B \geq 0$，每次变更都会产生成本。

4.1. 选民控制类型

我们考虑两种选民控制类型。第一种是通过删除选民进行控制（DV），主席可以选择一个选民子集，将其从选举中移除。第二种是通过增加选民进行控制（AV），主席可以引入具有预先设定偏好的额外选民。

通过删除选民进行控制（***DV***）

操纵者可以选择删除选民的一个子集$V^- \subseteq V$。删除选民$v \in V$会产生费用 成本(v)。对于一个选民子集，总费用为逐个删除选民的费用之和。

议会选举中的删除选民控制

输入： (P, V, $\succ$, B,成本, O)，其中：
P：政党，　V：选民，　$\succ$：偏好，　B：预算
成本：成本函数，　O：目标（J 或 J+F）

输出： 是否存在一个集合$V^- \subseteq V$，使得 成本(V^-) $\leq B$，并且满足主席的目标 O，在选举 $E := (P, \hat{V}, \hat{\succ})$，其中 $\hat{V} := V \setminus V^-$ and $\hat{\succ} := (\succ_v)_{v\in\hat{V}}$?

通过增加选民进行控制（***AV***）

给定一组搅局选民W、$W \cap V = \emptyset$，每位选民都对应一个偏好顺序。主席可以选择添加搅局选民的一个子集$W^+ \subseteq W$。添加一位选民$v \in W$需要付出成本$Cost(v)$。对于选民子集，总成本为逐个添加选民的成本之和。

议会选举中的通过增加选民进行控制

输入： (P, V, W, $\succ$, B,$Cost$, O)，其中：
P：政党，　V：选民，　W：搅局选民，
$\succ$：所有选民的偏好（在$V \cup W$中），　B：预算，
成本：成本函数，　O：目标（J 或 J+F）

输出： 是否存在一个集合$W^+ \subseteq W$，使得 $Cost(W^+) \leq B$，且主席的目标 O 在选举 $E := (P, \hat{V}, \hat{\succ})$ 其中 $\hat{V} := V \cup W^+$ 以及 $\hat{\succ} := (\succ_v)_{v\in\hat{V}}$?

4.2 选民控制的多项式时间算法

定理 4.1 表明，通过扩展 Put 和 Faliszewski 的算法 [20]，无论是删除选民还是添加选民这两种选民控制方式，均可在多项式时间内求解。

定理 4.1。 $\langle C|AV|J+F|\tau \geq 0\rangle$和 $\langle C|DV|J+F|\tau \geq 0\rangle$均可在多项式时间内求解。

完整证明见附录。

证明概要。 我们在此给出添加选民问题的证明；在附录中，我们将展示，删除选民问题可以通过一种相对简单的归约，使用相同的技术来解决。

我们考虑所有可能增加的选民数量 $r \in \{0 \ldots\ldots |W|\}$，以及将 r 分成三部分的所有可能方式（a_1,a_{C-1}、a_{-C}）：a_1 表示我们希望增加到偏好政党 p_1 的选民数量；a_{C-1} 表示

我们希望增加到联盟、但不增加到其中的选民数量 $C_{-1} := C \setminus \{p_1\}$；而 $a_{\bar{C}}$ 表示我们希望增加到反对派 $C := P \setminus C$ 的选民数量。

我们证明，可以使用动态规划算法，高效地遍历 r 的所有可能取值以及 r 的所有划分：$(a_1, a_{C_{-1}}, a_{\bar{C}})$。对于每个划分，我们可以直接验证主席的目标是否得到满足。如果满足，我们将检查是否可以通过这种方式添加选民：我们证明，可以在多项式时间内计算出实现这一额外支持所需的最小成本（如果无法实现，则将成本设为 ∞）。关键观察在于，每个集合的相关潜在选民——偏好政党 p_1，、不包含该政党的联盟C_{-1}，以及¯反对派C——彼此不同，因为这些集合本身是不相交的。因此，对于每个群体，我们只考虑其自身的相关选民池，并按成本非递减顺序排序。

如果某个满足主席目标的划分的最小成本不超过预算，我们就断言答案为“是”；否则，我们断言答案为“否”。

□

5. 政党控制

政党控制允许操纵者仅更改政党集合 $\hat{P}$（如下所述）。选民集合保持不变，$\hat{V} := V$，偏好为 $\hat{\succ} := \succ^{\hat{P}}$，其中 $\succ^{\hat{P}}$ 表示限制在 $\hat{P}$ 上的选民偏好（我们假设原始偏好 $\succ$ 定义在所有可能参加选举的政党之上，并且 $\hat{\succ}$ 与 $\hat{P}$上的 $\succ$ 一致）。

与上一节讨论的选民控制设置不同，这里我们关注一个更简单的模型。具体而言，我们假设添加或删除任意政党的统一成本为 1，并对这类操作的次数设置基数上限 k。即使采用这一简化，问题在计算上仍然很难，因此这种简化是合理的。

5.1. 政党控制类型

通过删除政党进行控制

给定一个整数 $1 \le k \le m$，主席可以从P中删除至多k 个政党。

议会选举中的通过删除政党进行控制

输入： $(P, V, \succ, k, O)$，其中：P：由m 个政党组成的集合，V：选民，$\succ$：偏好，k：整数（$1 \le k \le m$），O: 目标（J 或 J+F）

输出： 是否存在一个集合 $P^- \subseteq P$，满足 $|P^-| \le k$，使主席的目标 O在选举 $E' := (\hat{P}, V, \hat{\succ})$ 其中 $\hat{P} := P \setminus P^-$ 以及 $\hat{\succ} := \succ^{\hat{P}}$（即，将 $\succ$ 限制为 $\hat{P}$）？

通过增加政党进行控制

给定 一组搅局政党，S 其中 $S \cap P = \emptyset$ 和 $|S| := l$。请注意，假定偏好排序是在整个选举范围内的轮胎集合 $P \cup S$。给定一个整数 $1 \le k \le l$，主席最多可以添加k 个搅局政党。

议会选举中通过增加政党进行控制

输入： $(P, V, \succ, k, O)$，其中：P：政党，S：由 l 个搅局政党组成的集合，V：选民，$\succ$：偏好，k：整数（$1 \le k \le l$），O：目标（J 或 J+F）

输出： 是否存在一个集合 $S^+ \subseteq S$，满足 $|S^+| \le k$，使主席的目标 O 在选举 $\tilde{E} := (\hat{P}, V, \hat{\succ})$，其中 $\hat{P} := P \cup S^+$ 且 $\hat{\succ} := \succ^{\hat{P}}$（即，$\succ$限制为 $\hat{P}$）？

政党控制变体

我们分别考虑添加/删除联盟政党的情况，以及添加/删除反对党的情况。因此，我们有四种不同的情况：

- 删除联盟政党（DCP）
- 删除反对党（DOP）
- 增加联盟政党（ACP）。
- 增加反对党（AOP）

5.2. 政党控制：结果总结

我们考虑该问题的参数化版本，其中参数为 k ——添加或删除的政党数量[1]。表 3 总结的结果表明，所有相关情形即使从参数意义上看也都是困难的（$W[1]$或 $W[2]$-困难），并且是 NP困难的。其中两种情形 $\langle C|AOP|J|\tau=0\rangle$ 和 $\langle C|DCP|J|\tau=0\rangle$对操纵具有免疫性，也就是说，主席绝不可能通过此类控制推进其目标。

控制类型	**J目标**		**J+F 目标**
DCP	$\tau=0$	免疫（定理5.1）	W[1]-困难（定理5.2）
	$\tau>0$	W[1]-困难（定理 5.3）	W[1]-困难（定理 5.3*）
DOP		W[1]-困难（定理 5.4）	W[1]-困难（定理 5.4*）
ACP		W[2]-困难（定理 5.5）	W[2]-困难（定理 5.5*）
AOP	$\tau=0$	免疫（定理 5.6）	W[2]-困难（定理 5.8）
	$\tau>0$	W[2]-困难（定理 5.7）	W[2]-困难（定理 5.7*）

表 3：政党控制的复杂度结果，按添加或删除的政党数量进行参数化。括号内为相应的定理。星号表示该结果可直接由定理推出。

5.3. 删除联盟政党（*DCP*）

当 $\tau=0$ 时，定理 5.1 证明，删除联盟政党无法推进 J目标，因为删除此类政党只会减少联盟份额。相比之下，若同时考虑偏好政党（$\rho>0$），定理 5.2 表明这些目标可能实现，但这是 W[1]-困难的。

当 $\tau>0$时，定理 5.3 证明，即使 $\rho=0$，该问题仍是 W[1]-困难的。

定理 5.1。 $\langle C|$删除联盟政党（*DCP*）$|J|\tau=0\rangle$是不可能的。

证明。每位选民恰好给其排名第一的政党一分。由于所有被删除的政党都属于联盟，我们区分以下两种情况：

- 对于首选政党不属于联盟的选民，删除一个联盟政党不会改变结果，反对党获得的总得分也保持不变。
- 对于首选政党属于联盟的选民，如果删除的是排在首位的联盟政党，而下一位政党属于反对派，则联盟获得的得分可能会减少。

这个问题只关注联盟整体获得的得分比例，而非其中任何特定偏好政党获得的比例。因此，删除反对党只会损害联盟的整体地位。

□

定理 5.2。 $\langle C|$删除联盟政党$|J+F|\tau=0\rangle$以删除政党数量为参数时是 *W[1]*-困难的。

[1]请注意，该问题关于政党数量是平凡的固定参数可 tractable。

证明。 证明通过从 k-团问题进行归约得出：给定一个无向图 $G=(W,E)$ 和一个参数k，判断是否存在一个大小恰好为 k 的团。

我们证明，给定一个《k-团问题》的实例，可以构造一个参数相同、即k的 ⟨C|删除联盟政党|J+F|$\tau=0$⟩问题实例，使得存在一个k-团，当且仅当可以通过删除k 个联盟政党来实现 J+F 目标。

令$P:=\{p_1,o_1,s_1,\ldots,s_{|W|}\}$，其中每个$s_i$对应于$G$中的一个顶点；$C:=\{p_1,s_1,\ldots,s_{|W|}\}$，以及偏好政党$p_1$；$\varphi:=\frac{1}{2}$，以及$\rho:=\lfloor |E|+\binom{k}{2}\rfloor 2|E|$。

共有 $4|E|$ 名选民，每条边对应 4 名选民（x，y）$\in E$，其偏好定义如下 s:

- 1 名 AA型选民：$s_x \succeq s_y \succ p_1 \succ o_1 \succ \ldots$
- 1 B 类选民：B：$p_1 \succ o_1 \succ$.....
- 2 C 类选民：C：$o_1 \succ$.....

($\Rightarrow$) 假设图中存在一个k-团。我们证明，移除与该团中顶点对应的k 个联盟政党后，可以实现 J+F 目标。

根据定义，团内两端点都位于其中的边有 $\binom{k}{2}$ 条。因此，移除这些政党后：p_1 获得 $|E|+\binom{k}{2}$ 张选票，与 W 对应的政党集合获得 $|E|-\binom{k}{2}$ 个选民；而 o_1 获得 $2|E|$ 张选票。因此，该联盟实现$\lfloor 2|E|$，且p_1 获得该联盟所需的 $\lfloor |E|+\binom{k}{2}\rfloor 2|E|$ 选票。

($\Leftarrow$) 假设通过移除k 个联盟政党可以实现 J+F 目标。我们证明，对应的顶点构成一个k-团。

在原始选举中，o_1 获得了 $2|E|$ 张选票。由于所有首选不是 o_1 的选民都更喜欢 p_1 而非 o_1，，删除C_{-1} 政党不会影响o_1的得票数。这意味着，删除p_1以外的任何联盟政党后，联盟的得票数仍为 $2|E|$。既然我们知道，对于 $\rho=\lfloor |E|+\binom{k}{2}\rfloor 2|E|$，最爱政党目标 (F) 得以满足，那么$p_1$,至少必须获得 $|E|+\binom{k}{2}$ 张选票。在原始选举中，p_1 获得了 $|E|$ 张选票，因此政党删除至少为p_1带来了$\binom{k}{2}$张额外选票。

每一张以这种方式为p_1 带来的新选票，都只能来自一名 A 型选民；该选民的前两个政党分别对应顶点 x 和y。要以这种方式至少获得 $\binom{k}{2}$ 张新选票，被删除的政党必须对应一个包含k 个顶点且能诱导出至少 $\binom{k}{2}$ 条边的集合。只有当这些顶点构成一个大小为k的团时，这才有可能。□

定理 5.3。 ⟨C|删除联盟政党|J|$\tau\geq 0$⟩问题以删除政党数量为参数时是 $W[1]$-困难的。

证明。 证明通过从k-团问题进行归约得到：给定一个无向图$G=(W,E)$和一个参数k，判断其中是否存在大小恰好为k的团。

我们证明，给定一个 k-团问题实例，可以构造一个参数相同（即 k）的 ⟨C|删除联盟政党|J|$\tau\geq 0$⟩问题实例，使得存在 k-团，当且仅当能够通过删除 k 个联盟政党实现 J目标。令 P := $\{p_1,o_1,s_1,\ldots,s_n\}$，其中每个 s_i都对应于 G 中的一个顶点；C $=\{p_1,s_1,\ldots,s_n\}$; $\varphi=\frac{1}{2}$，且选举门槛为 $\tau=\lfloor |E|+\binom{k}{2}\rfloor 3|E|+\binom{k}{2}$。共有 $3|E|+\binom{k}{2}$名选民，其偏好定义如下:

- $|E|$ A 型选民：每位选民对应图中的一条边（x，y）$\in E$。他们的偏好为$s_x \succ s_y \succ p_1 \succ o_1 \succ P$。 ..
- $|E|$ B类选民：$p_1 \succ o_1 \succ$。 ...
- $|E|$ C类选民：$o_1 \succ$。 ...
- $\binom{k}{2}$ D 型选民：$o_1 \succ$。 ...

($\Rightarrow$) 假设图中存在一个k团。我们证明，移除与该团中顶点对应的k个政党即可实现 J 目标。根据定义，有 $\binom{k}{2}$ 条边的两个端点都位于

该团内。因此，移除这些政党后，第一个政党将是p_1,，所以p_1将获得 $|E| + \binom{k}{2}$ 张选票。在其余$|E| - \binom{k}{2}$位选民$v_{1,i}$中，第一个政党不是p_1,，而是联盟的一张非活跃选票。此外，不属于联盟的o_1,会获得 $|E| + \binom{k}{2}$ 张选票，且无法通过删除联盟中的政党来改变。因此，共有 $2(|E| + \binom{k}{2})$ 张有效选票，联盟实现了 $\varphi = \frac{1}{2}$。

(⇐) 假设通过删除k 个政党可以实现J目标。在原始选举中，不属于联盟的o_1, 获得了 $|E| + \binom{k}{2}$ 张选票，而删除联盟中的政党无法改变这一点。政党$s_1, \ldots, s_{|W|}$属于联盟，但它们都没有达到选举门槛，而删除联盟中的政党也无法改变这一点。为满足联盟目标，p_1, 必须至少获得 $|E| + \binom{k}{2}$ 张选票，也就是说，通过控制操作至少获得 $\binom{k}{2}$ 张新选票。如果删除操作使p_1, 获得一张选票，就意味着与相应边的两个端点对应的顶点都在被删除的政党之中。这意味着删除k 个政党，成功地涵盖了 $\binom{k}{2}$ 条边的两个端点所对应的顶点。因此，这些顶点之间的所有边都必须存在于图G中，这表明这些顶点构成一个团。

□

5.4. 删除反对党（*DOP*）

定理5.4证明，即使 $\rho = 0$ 且选举门槛 $\tau =$，该问题仍然是 W[1]-困难的 0.

定理5.4。 ⟨*C*|*DOP*|*J*|$\tau = 0$⟩ 以删除政党的数量为参数时是 *W[1]*-困难的。

证明。 证明通过从k-团问题进行归约得出：给定一个无向图$G = (W,D)$和一个参数k，判断其中是否存在大小恰为k的团。

我们证明，给定k-团问题的一个实例，可以构造出具有相同参数k的 ⟨C|DOP|J|$\tau = 0$⟩问题实例，使得存在一个k-团，当且仅当可以通过删除k个反对党来实现J目标。

令 $P = \{p_1, s_1, \ldots, s_{|W|}\}$，其中每个 s_i 对应 G 中的一个顶点。令联盟为 $C = \{p_1\}$。删除政党的数量参数设为 k。令 $\varphi = \binom{k}{2}|E|$（注意，如果 $\varphi > 1$，那么图中不存在大小为 k 的团，因为根本没有 $\binom{k}{2}$ 边）。令 $V = \{v_1, \ldots v_{|E|}\}$ 为选民集合，每位选民对应 G 中的一条边。对应边 (x, y) 的 v_i的偏好顺序定义为 $s_x \succ s_y \succ p_1 \succ P \setminus \{s_x, s_y, p_1\}$。

(⇒) 假设图中存在一个 k 团，则删除对应的k 个政党。这些顶点之间的每条边都存在于图中，因此恰好有 $\binom{k}{2}$ 条这样的边。于是，在所有对应于这些边的选民的偏好顺序中，删除后排在首位的政党将是p_1,，联盟因此获得$\binom{k}{2}|E|$，符合要求。

(⇐) 假设存在一个成功的控制操作。在原始选举中，p_1 没有获得任何选票，且有 $|E|$ 名选民参与。因此，该控制操作必须通过删除k 个政党，确保至少获得 $\binom{k}{2}$ 张选票。如果删除后有选票投给p_1,，则意味着与对应边两端相对应的两个顶点都属于被删除的政党。这说明，删除k 个政党，成功涵盖了 $\binom{k}{2}$ 条边两端所代表的顶点。因此，这些顶点之间的所有边都必须存在于图G中，表明这些顶点构成一个团。

□

5.5. 添加联盟政党（*ACP*）

通过添加联盟政党进行联盟政党控制是 W[2]-困难的，即使 $\tau = 0$ 也是如此。作为最简单的变体，这表明其余变体至少同样困难。

定理5.5。 ⟨*C*|*ACP*|*J*|$\tau = 0$⟩ 参数化于添加政党数量时是 *W[2]*-困难的。

证明。 该证明通过从支配集问题进行归约得到：给定一个无向图 $G = (W,E)$ 以及参数k，需要判定是否存在大小至多为 k 的支配集。

我们证明，给定支配集问题的一个实例，可以构造出一个参数相同（即 k）的 ⟨C|ACP|J|$\tau = 0$⟩ 问题实例，使得当且仅当存在大小至多为 k 的支配集时，通过添加 k 个联盟政党实现 J目标才是可能的。

令搅局政党的集合为$S := \{s_1, \ldots, s_{|W|}\}$，其中每个搅局政党$s_i$ 对应$w_i \in W$中的顶点G。令$P := \{p_1, o_2\}$。令$N[s_i]$为与w_i的邻居对应的政党集合——也就是说，$N[s_i] := \{s_j \in S \mid$

$(w_i, wj) \in D\}$。共有 $2|W|$ 名选民，每个顶点$w_i \in W$对应2名选民，他们的偏好定义如下：

- 1 A 型A选民：其最偏好的政党是 $\{s_i\} \cup N[s_i]$ （它们之间的顺序无关紧要）；
  然后o_2, 接着是p_1, , 最后是其余政党。
- 1 B类B选民：$o_2 \succ p_1 \succ .....$

设联盟为$C := \{p_1, s_1, ..., s_{|W|}\}$，目标比例为 $\varphi := \frac{1}{2}$。

（⇒）假设存在一个大小至多为k的支配集。在这种情况下，添加相应搅局政党的控制操作可确保联盟获得50%的选票。这是因为，对于 A 型选民而言，选民v_i 的每个偏好顺序中的第一个政党都是属于该联盟且属于支配集的搅局政党。对于 B类选民而言，选民v_i 的每个偏好顺序中的第一个政党是o_2, , 它不属于该联盟，且无法通过添加搅局政党来改变。因此，该联盟获得50%的选票。

(⇐) 假设存在一个成功的控制操作，$S' \subseteq S$，它至多增加了k 个搅局政党，并使联盟获得50%的选票。对于 B 类选民，选民v_i 的偏好顺序中的第一个政党是o_2, , 该党不属于联盟，并且无法通过增加政党来改变。因此，所有 A 型选民的偏好顺序中的第一个政党都必须属于联盟。也就是说，对于每位选民，$N[s_i]$ 中的一个必须属于S'，且 $|S'| \leq k$，因此包含相应顶点的集合是给定图的支配集。

□

5.6. 添加反对党（*AOP*）

当 $\tau = 0$ 时，定理5.6证明，增加反对党不会改善 J目标，因为这种增加只会扩大反对方份额。然而，当 $\tau > 0$ 时，或对于 C+F 目标，这些问题是 W[2]-难的（分别见定理 5.7 和定理 5.8）。

定理5.6。 $\langle C|AOP|J|\tau = 0\rangle$是免疫的。

证明。 每位选民恰好给排名第一的政党一分；因此，在其他位置添加政党不会产生任何影响。我们区分两种可能的情况：

- 对于首选不属于联盟的选民，将搅局反对党排在第一位不会影响结果，因为反对派的总得分保持不变。
- 对于首选属于联盟的选民，将搅局反对党排在第一位会减少联盟的得分。

这个问题只涉及联盟整体获得的得分比例，而非其中任何特定偏好政党的得分比例。因此，添加搅局反对党只会损害联盟的整体地位。

□

定理 5.7。 $\langle C|AOP|J|\tau \geq 0\rangle$以新增政党数量为参数是 *W[2]*-困难的。

该证明采用了与 [8]类似的方法。

证明。 该证明通过从支配集问题归约而得：给定一个无向图 $G = (W,E)$和一个参数k，判断是否存在大小至多为k的支配集。

我们证明，给定支配集问题的一个实例，可以构造一个参数相同（k）的 $\langle C|AOP|J|\tau \geq 0\rangle$ 问题实例，使得存在大小至多为k 的支配集，当且仅当通过添加k 个联盟政党来实现J目标是可能的。

令搅局政党集合 $S = \{s_1, \ldots, s_{|W|}\}$ 包含 G 中每个顶点对应的一个搅局政党。令 $P = \{p_1, o_2, o_3\} \cup S$ 为政党集合。令联盟为 $C = \{p_1\}$。令 $\varphi = 50\%$。令选举门槛为 $\tau = 40\%$。令 $N[s_i]$ 为与 w_i's 邻居对应的政党集合——即，$N[s_i] := \{s\, j \in S \mid (w_i, wj) \in D\}$。共有 $5|W|$ 名选民，$W_i \in$ 中的每个顶点对应 5 名选民，他们的偏好定义如下：

- 2 A类A选民：$p_1 \succ o_2 \succ \ldots$

- 2 名 B 型选民：B：$o_2 \succ p_1 \succ \ldots$
- 1 C类C选民：其最偏好的政党是 $\{s_i\} \cup N[s_i]$ 中的政党，并且对它们无差异；其次是o_2，然后是p_1，最后是其余政党。

请注意，在进行任何控制操作之前，联盟获得了40%的选票
(⇒) 假设存在一个大小至多为k的支配集。在这种情况下，添加相应搅局政党的控制操作可确保联盟获得50%的选票。

这是因为，有 $4|W|$ 名选民（即A类和B类选民），他们的偏好政党不是搅局政党之一，而且无法通过增加政党进行控制来改变。控制操作之后，对于所有 $|W|$ 名C类选民而言，他们的偏好政党都是未能达到选举门槛的搅局政党。因此，控制操作之后只有 $4|W|$张选票有效，联盟由此获得50%的选票，符合要求。

(⇐) 假设存在一个成功的控制操作，$S' \subseteq S$，它添加至多k 个搅局政党，并实现成功的控制操作。无法为p_1, 争取到更多选票，因此满足φ 要求的唯一方法是缩小反对派的规模。通过增加政党进行控制，不可能改变所有 $4|W|$ 个 A 型和 B 型选民的偏好政党。

因此，控制操作必定影响 C 类选民的偏好政党，使其在控制操作前投给反对派的选票变为控制操作后的非活跃反对派选票。也就是说，对于每个选民v_i，$\{s_i\} \cup N[s_i]$ 中必有一个属于S'，且 $|S'| \leq k$，因此包含相应顶点的集合是给定图的支配集。

□

定理 5.8。 $\langle C|AOP|J+F|\tau=0\rangle$ 按新增政党数量参数化后是 $W[2]$-难的。

证明。 该证明通过从支配集问题进行归约得出：给定一个无向图 $G = (W,E)$，以及一个参数k，判定是否存在大小至多为k的支配集。

我们证明，给定支配集问题的一个实例，可以构造出一个参数相同（k）的 $\langle C|AOP|J+F|\tau=0\rangle$ 问题实例，使得存在大小至多为k 的支配集，当且仅当通过添加k 个联盟政党可以实现J+F目标。

令搅局政党集合$S = \{s_1, \ldots, s_{|W|}\}$ 中包含对应于G中每个顶点的一个搅局党。令$P = \{p_1, c_2\}$，联盟为$C = \{p_1, c_2\}$，偏好政党为p_1。令添加政党数量这一参数为k。令$\varphi = 50\%$，并令 $\rho = 50\%$。

令$N[s_i]$为与w_i相邻的顶点对应的政党集合——也就是说，$N[s_i] := \{s_j \in S \mid (w_i, w_j) \in D\}$。
Th共有 $4|W|$ 名选民，每个顶点$w_i \in W$对应 4 名选民，其偏好定义如下： 如下 s:

- 1 A 类选民：$p_1 \succ c_2 \succ$ .....
- 1 B 类选民：$c_2 \succ p_2 \succ$ .....
- 2 C 类选民：最喜欢的政党是 $\{s_i\} \cup N[s_i]$ 中的政党，且对它们无差异；接下来是c_2，然后是p_1，再然后是其余政党。

请注意，在进行任何控制操作之前，联盟获得了100%的选票，但p_1 仅获得联盟的0.25%。

(⇒) 假设存在一个大小至多为k的支配集。在这种情况下，添加相应搅局政党的控制操作可确保联盟只能获得50%的选票。这是因为，对于所有 $2|W|$名A类和B类选民而言，他们的偏好政党不是搅局政党，而且无法通过增加政党进行控制来改变。控制操作之后，对于 $2|W|$ 名C类选民而言，他们的偏好政党是一个不属于联盟的搅局政党。因此，联盟获得50%的选票，而p_1 在联盟中的比例变为所要求的0.5%。

(⇐) 假设存在一个成功的控制操作，$S' \subseteq S$，它至多加入了k 个搅局政党，并实现了成功的控制操作。无法为p_1, 争取到更多选票，因此满足 ρ要求的唯一方法是在不减少p_1所得选票数的情况下缩小联盟规模。对于 $2|W|$ 名 A 类和 B 类选民，通过增加政党进行控制无法改变他们偏好的政党。

因此，控制操作之后， $2|W|$ 名 C 类选民所偏好的政党必定不属于联盟。也就是说，对于每位选民， $\{s_i\} \cup N[s_i]$ 中的某一个必须属于 S'，且 $|S'| \le k$，因此包含相应顶点的集合是给定图的一个支配集。 □

6. 未来工作

我们介绍并研究了议会选举中的贿赂与控制问题，目标是促成整个政党联盟的形成。这项工作为广泛的开放问题奠定了基础。

操纵类型。 拓展议会选举中策略操纵的研究范围尤其引人关注，特别是研究通过政党分裂与合并进行控制的方式；这种做法在现实中屡见不鲜。此外，改变选举门槛可能产生重大影响，而且在实践中也很常见。因此，对选举门槛的操纵是一个自然而有趣的研究方向。另一个方向是破坏性操纵，即操纵者旨在阻止竞争联盟获胜。

联盟动态。 本文考察了操纵者希望促成的单一政党和单一政党联盟。在实践中，政治动态更加丰富复杂。例如，操纵者可能有兴趣促成若干可能联盟中的一个，例如任何中间派／左派／右派联盟。或者，操纵者可能希望确保其偏好政党成为某个 获胜联盟的一部分。同样，在破坏性操纵方面，操纵者可能希望某个特定政党或若干政党不 属于任何获胜联盟。研究这样的情形也很有意思：操纵者不仅偏好某个政党，还会指定联盟政党之间期望的权力平衡。

多个操纵者。 在本文中，我们假设只有一个操纵者，所讨论的问题也仅涉及计算。在实际中，可能存在多个操纵者，各自有不同的目标和目的。在这种情况下，策略考量便会发挥作用。研究此类动态及其均衡，是一个值得探索的未来方向。

7. 致谢

本研究部分由以色列科学基金会的资助项目 2544/24、3007/24 和 2697/22 提供支持。

参考文献

[1] N. Betzler, J. Uhlmann，选举中候选人控制的参数化复杂性及相关有向图问题，《理论计算机科学》410 (2009) 5425–5442. [2] E. Elkind, P. Faliszewski, A. Slinko，交换贿赂，载于：算法博弈论：第二届国际研讨会，SAGT 2009，塞浦路斯帕福斯，2009年10月18–20日。论文集 2，Springer，2009，页 299–310. [3] E. Elkind, P. Faliszewski, S. Gupta, S. Roy，结构化选举中交换与位移贿赂的算法，载于：第19届自主智能体与多智能体系统国际会议论文集，AAMAS '20，自主智能体与多智能体系统国际基金会，里奇兰，南卡罗来纳州，2020，页 366–374. [4] P. Faliszewski, J. Rothe，投票中的控制与贿赂，载于：F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. D. Procaccia（编），《计算社会选择手册》，剑桥大学出版社，2016，页 146–168. URL: https://doi.org/10.1017/CBO9781107446984.008。doi:10.1017/CBO9781107446984.008。[5] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra，选举中贿赂的复杂性，载于：第21届人工智能全国会议论文集——第1卷，AAAI'06，AAAI出版社，2006，页 641–646. [6] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra，选举中的贿赂有多难？《人工智能研究杂志》35 (2009) 485–532.[7] O. Keller, A. Hassidim, N. Hazon，计分规则中贿赂问题的近似，载于：S. A. McIlraith, K. Q. Weinberger（编），第三十二届人工智能 AAAI 会议（AAAI-18）、第30届人工智能创新应用会议（IAAI-18）及 8 18 2 7, 届人 2018, 2018, 1121 1129 https://doi.org/10.1609/aaai.v32i1.11476 10 1609 32 1 11476 工智能教育进展研讨会（EAAI- ）论文集，美国路易斯安那州新奥尔良，2月——AAAI出版社，页 –。URL：。doi：。/AAAI.V I . . [8] H. Liu, H. Feng, D. Zhu, J. Luan，投票系统中控制问题的参数化计算复杂性，《理论计算机科学》410 (2009) 2746–2753. [9] L. Tao, L. Chen, L. Xu, W. Shi, A. Sunny, M. M. Uz Zaman，随机选民选举中的贿赂有多难，载于：第21届自主智能体与多智能体系统国际会议论文集，2022。[10] Y. Yang，基于认可的多赢家投票中操纵与控制的复杂性，载于：第二十八届国际人工智能联合会议论文集，IJCAI-19，国际人工智能联合会议组织，2019，页 637–643. URL: https://doi.org/10.24963/ijcai.2019/90。doi:10.24963/ijcai.2019/90。[11] A. Zhou, J. Guo，迭代选举中位移贿赂的参数化复杂性，载于：第19届自主智能体与多智能体系统国际会议论文集，2020，页 1665–1673. [12] J. J. Bartholdi III, C. A. Tovey, M. A. Trick，控制一场选举有多难？《数学与计算机建模》16 (1992) 27–40. [13] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, I. Schnoor，选举操纵的可近似性，载于：D. Fox, C. P. Gomes（编），第二十三届人工智能 AAAI 会议论文集，AAAI 2008，美国伊利诺伊州芝加哥，2008年7月13–17日，AAAI出版社，2008，页 44–49. URL: http://www.aaai.org/Library/AAAI/2008/aaai08-007.php。[14] P. Faliszewski，非一致贿赂，载于：L. Padgham, D. C. Parkes, J. P. M¨uller, S. Parsons（编），第七届自主智能体与多智能体系统国际联合会议（AAMAS 2008），葡萄牙埃什托里尔，2008年5月12–16日，第3卷，IFAAMAS，2008，页 1569–1572. URL: https://dl.acm.org/citation.cfm?id=1402927。[15] P. Faliszewski, Y. Reisch, J. Rothe, L. Schend，Bucklin 投票与回退投票中的操纵、贿赂及竞选管理的复杂性，《自主智能体与多智能体系统》29 (2015) 1091–1124. [16] C. Maushagen, M. Neveling, J. Rothe, A.-K. Selker，迭代投票规则中位移贿赂的复杂性，《数学与人工智能年刊》90 (2022) 1017–1054. [17] D. Parkes, L. Xia，舒尔策规则与排序对在策略行为复杂性方面的比较，载于：AAAI 人工智能会议论文集，第26卷，2012，页 1429–1435. [18] C. G. Menton, P. Singh，舒尔策投票的控制复杂性，载于：第二十三届国际人工智能联合会议，2013。[19] A. D. Procaccia, J. S. Rosenschein, A. Zohar，多赢家选举：操纵、控制与获胜者确定的复杂性，载于：IJCAI，第7卷，2007，页 1476–1481. [20] T. Put, P. Faliszewski，议会选择规则下选民控制与位移贿赂的复杂性，《计算集体智能汇刊》23 (2016) 29–50. URL: https://doi.org/10.1007/978-3-662-52886-0 3。doi:10.1007/978-3-662-52886-0 3。[21] S. Sina, N. Hazon, A. Hassidim, S. Kraus，通过调整社交网络影响选举，载于：2015年自主智能体与多智能体系统国际会议论文集，Citeseer，2015，页 705–713. [22] C. Maushagen, J. Rothe，最后一种投票规则恰到好处：极小极大选举中通过划分候选人或选民实施控制的复杂性，载于：ECAI 2020，IOS出版社，2020，页 163–170. [23] R. Meir, A. D. Procaccia, J. S. Rosenschein, A. Zohar，多赢家选举中策略行为的复杂性，《人工智能研究杂志》33 (2008) 149–178. [24] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, T. Walsh，多赢家认可投票的计算方面，载于：AAMAS，第15卷，2015，页 107–115. [25] R. Bredereck, P. Faliszewski, R. Niedermeier, N. Talmon，委员会选举中位移贿赂的复杂性，载于：AAAI 人工智能会议论文集，第30卷，2016。[26] R. Bredereck, A. Kaczmarczyk, R. Niedermeier，关于多赢家选举中的联盟操纵：候选名单筛选，《自主智能体与多智能体系统》35 (2021) 38. [27] S. Obraztsova, Y. Zick, E. Elkind，关于基于计分规则的多赢家选举中的操纵，载于：2013年自主智能体与多智能体系统国际会议论文集，2013，页 359–366. [28] Y. Yang，关于基于认可的多赢家投票中破坏性贿赂的复杂性，《ACM 计算理论汇刊》（2025）。URL: https://doi.org/10.1145/3749377。doi:10.1145/3749377，刚刚接收。[29] A. D. Taylor, W. S. Zwicker，《简单博弈：可取关系、交易与伪权重》（2021）。

[30] H. Aziz、Y. Bachrach、E. Elkind、M. Paterson，《加权投票博弈中的虚假身份操纵》，人工智能研究杂志 40 (2011) 57–93。[31] A. Rey、J. Rothe，《加权投票博弈中的虚假身份操纵对于概率多项式时间而言是困难的》，人工智能研究杂志 50 (2014) 573–601。[32] A. Rey、J. Rothe，《加权投票博弈中的结构控制》，BE理论经济学期刊 18 (2018) 20160169。[33] Y. Zick、A. Skopalik、E. Elkind，《Shapley值作为加权投票博弈中配额的函数》，载于：第22届国际人工智能联合会议，IJCAI 2011，2011，第490–495页。[34] M. Zuckerman、P. Faliszewski、Y. Bachrach、E. Elkind，《操纵加权投票博弈中的配额》，《人工智能》180 (2012) 1–19。[35] A. Slinko、S. White，《比例代表制与策略性选民》，《理论政治学杂志》22 (2010) 301–332。[36] S. Bowler、J. A. Karp、T. Donovan，《策略性联盟投票：来自新西兰的证据》，《选举研究》29 (2010) 350–357。[37] G. W. Cox，《议会选举中的利益组合最大化策略投票》，《牛津选举制度手册》(2018) 265。[38] T. Gschwend、L. Stoetzer、S. Zittlau，《是什么促成了租借选票？联盟信号如何促进策略性联盟投票》，《选举研究》44 (2016) 293–306。[39] B. McCuen、R. B. Morton，《实验室中的战术性联盟投票与信息》，《选举研究》29 (2010) 316–328。网址：
https://www.sciencedirect.com/science/article/pii/S0261379410000211。
doi:https://doi.org/10.1016/j.electstud.2010.03.009，专题研讨会：选民与联合政府。[40] M. F. Meffert、T. Gschwend，《策略性联盟投票：来自奥地利的证据》，《选举研究》29 (2010) 339–349。[41] D. Boratyn、W. Słomczy´nski、D. Stolicki，《杰斐逊–丹特分配法下的席位分配与席位偏差》，2024年。网址：https://arxiv.org/abs/1805.08291. arXiv:1805.08291。[42] J. Flis、W. Słomczy´nski、D. Stolicki，《锅与勺：一种估算杰斐逊–丹特分配法下席位分配的公式》，《公共选择》182 (2020) 201–227。doi:10.1007/s11127-019-00680-w。[43] R. K. Ahuja、T. L. Magnanti、J. B. Orlin，《网络流：理论、应用与算法》，美国新泽西州恩格尔伍德克利夫斯，Arrow, KJ：普伦蒂斯·霍尔 (1993)。[44] R. Downey、M. Fellows、R. Downey、M. Fellows，《参数化可约性下的语言结构》，《参数化复杂性》(1999) 389–437。[45] R. G. Downey、M. R. Fellows，《固定参数可处理性与完备性 I：基本结果》，《SIAM计算杂志》24 (1995) 873–921。doi:10.1137/S0097539792228228。

A. 已知 NP 难问题

定义 3 (3-4精确覆盖问题)。给定：一个元素集合 Z。Z 的一个4-子集集合（即，对于任意 $D \in \mathcal{D}: D \subseteq Z$ 且 $|D| = 4$）；使得 Z 中的每个元素 $z \in Z$ 恰好属于3 个子集，且这些子集属于 $\mathcal{D}$。问题：是否存在一个子集族 $\mathcal{D}' \subseteq \mathcal{D}$，使得 Z 中的任意元素恰好出现在 $\mathcal{D}'$的一个子集中？

3-4精确覆盖问题是 NP 困难 [13]。

定义 4 （支配集问题）。给定：一个图 $G = (W, E)$，以及一个参数 k。问题：是否存在一个大小至多为 k 的支配集 $W' \subseteq W$——也就是说，对于任意顶点 $w \in W$，w 要么属于覆盖，$w \in W'$；要么它的某个邻居属于覆盖，$\exists w' \in W'$ s。t $(w, w') \in E$。

支配集问题是 W[2]-困难 [44]。

定义 5 （k-团问题）。给定：一个图 $G = (W, E)$，以及一个参数 k。问题：图中是否存在一个大小为 k 的团。也就是说，一个包含 k 个顶点的集合 W'，使得 $\forall w, w' \in W' (w, w') \in E$。

k-团问题是一个 W[1]-困难问题 [45]。

B. 第 3 节的证明（贿赂）

B.1. 1-贿赂和 *$*-贿赂

定理 3.1. $\langle B|\$|J+F|\tau \geq 0\rangle$ 可在多项式时间内求解。

证明。 为证明该定理，我们将展示算法 1（如下所述）能够解决该问题，并在多项式时间内运行。该算法将贿赂过程分为两步：首先，确定要贿赂的选民集合，并将他们当前的选票从选票池中剔除；之后再确定他们的新选票。因此，我们定义“未确定贿赂”这一概念，指的是一组选民已被贿赂，其˜原始选票因此被移除，但他们的新选票尚未确定的情形。严格来说，未确定贿赂是一个集合 $V = \{v_{i_1}, \ldots, v_{i_\ell}\}$，其中包含（被贿赂的）选民。未确定贿赂 ˜$V$的成本为 $Cost^{\$}(\tilde{V}) = \sum_{i:v_i \in \tilde{V}} \pi_{i_0}$ 请注意，在 $-贿赂中，该成本 ˜s足以随后以任意期望的方式确定 V 中所有成员的选票。

该算法的关键组成部分是对函数 g 进行动态规划计算，使得 g(l, a -C, d, a$_{C_{-1}}$) ˜ ˜表示满足以下四项要求的未确定贿赂 V 的最低成本。(1) | V| = l；(2) 对所有 $v_i \in$ ˜V，都有 $Top^E(v_i) \neq p_1$, ；(3) 从 V 的选票中 \ ˜V（即未被贿赂的选民），有 $a_{\bar{C}}$ 张有效选票投给˜C 的政党；(4) 存在一种将 d 张额外选票分配给 C 的政党的方式$_{-1}$，使得这些选票与 V 的˜选票 \V 一起，C 的政党获得的有效选票总数$_{-1}$为 a$_{C_{-1}}$。如果不存在满足上述条件的未确定贿赂，则 g(l, a -C, d, a$_{C_{-1}}$) = ∞。

令最小成本(p_j, l)表示在E中，贿赂投给p_j的 l 名选民所需的最低成本（若不存在这样的贿赂，则为 ∞）。

现在我们将证明，可以在多项式时间内计算所有 $0 \leq l$、$a_{C_{-1}}$、d、$a - c \leq n$对应的$g(l, a - c, d, a_{C_{-1}})$，以及 $1 \leq j \leq m, 1 \leq l \leq n$对应的最小成本($p_j$, k)。

对于政党p_j，令$V_j = \{v_i : Top^E(v_i) = p_j\}$——即（贿赂前）投给$p_j$的选民集合。对于政党集合$D$，令$V_D = \cup_{p_j \in D} V_j$。要计算最小成本($p_j$, l)，只需先对集合 $\{\pi_i : v_i \in V_j\}$排序。然后，最小成本(p_j, l) 就是该集合中最小的 l 个值之和。

˜ ˜对于 g：在寻找成本最低的未确定贿赂 V 时，我们分别计算 $V_{C_{-1}}$ ——最初投票给 C_{-1} 的被贿赂选民——以及 $V_{\bar{C}}$ ——最初投票给 C 的被贿赂选民。

˜ - ˜计算 V -c。对于政党集合 $D \subseteq C$，令 $f(D, l, a_D)$ 表示未确定贿赂 $W \subseteq V_D$的最低成本，˜ ˜其满足：(i) $|W| = l$；(ii) 根据 $V_D \setminus W$ 的选票，恰有 a_D 张选票有效。

˜我们给出一个动态规划过程，用于计算所有 $D \subseteq C$，以及 $0 \leq l$、$a - c \leq n$ 时的 $f(D, l, a - c)$。首先，考虑大小为 1 的D：

$$f(\{j\}, \ell, a_{\{j\}}) = mincost(p_j, \ell)$$

如果$a_{\{j\}} = |V_j| - l \geq T$或 $|V_j| - l < T$且$a_{\{j\}} = 0$。在所有其他情况下，$f(\{j\}, l, a_{\{j\}}) = \infty$。

Now,

$$f(D \cup \{j\}, \ell, a_{D\cup\{j\}}) = \min_{0\le\ell'\le\ell, 0\le a_D\le a_{D\cup\{j\}}} \{f(D, \ell', a_D) + f(\{j\}, \ell - \ell', a_{D\cup\{j\}} - a_D)\}$$

这样就可以通过逐个加入政党，迭代计算所有 l、$a_{\cdot C}$对应的$f(C, \mathrm{l}, a_{\cdot C})$。

计算 $\tilde{V}_{C_{-1}}$. 对于政党集合 $D \subseteq C_{-1}$，令 $h(D, \mathrm{l}, d, a_D)$ 表示未确定贿赂 $W \subseteq V_D$ 的最低成本使得：(i) $|\tilde{W}| = \mathrm{l}$，(ii) 加上 d 张额外选票后，有可能为 D 获得总计 a_D 张有效选票。

同样，我们使用动态规划计算所有 $0 \le \mathrm{l}$、d、$a_{\cdot C} \le n$ 对应的 $h(C_{-1}, \mathrm{l}, d, a_{C_{-1}})$。

对于大小为 1 的 D，如果以下条件之一成立：

- $|V_j| - \ell + d \ge T$ and $a_{\{j\}} = |V_j| - \ell + d$,
- $|V_j| - \ell + d < T$ and $a_{\{j\}} = 0$

然后我们将 $h(\{j\}, \mathrm{l}, d, a_{\{j\}})$ 设为 最小成本(p_j, l)；否则，将其设为 ∞。现在，

$$h(D \cup \{j\}, \ell, d, a_{D\cup\{j\}}) = \min_{0\le\ell'\le\ell, 0\le d'\le d, 0\le a_D\le a_{D\cup\{j\}}} \{h(D, \ell', d', a_D) + h(\{j\}, \ell - \ell', d - d', a_{D\cup\{j\}} - a_D)\}$$

这样就可以通过逐个添加政党，迭代计算所有 l、d、$a_{C_{-1}}$ 对应的 $h(C_{-1}, \mathrm{l}, d, a_{C_{-1}})$。

Co 计算 g。现在，已计算出所有 l、d、$a_{C_{-1}}$、$a_{\cdot C}$ 对应的 $h(C_{-1}, \mathrm{l}, d, a_{C_{-1}})$ 和 $f(C, \mathrm{l}, a_{\cdot C})$，我们计算 g:

$$g(\ell, a_{\bar{C}}, d, a_{C_{-1}}) = \min_{0\le\ell'\le\ell} \{h(C_{-1}, \ell', d, a_{C_{-1}}) + f(\bar{C}, \ell - \ell', a_{\bar{C}})\} \qquad \square$$

然后，借助对 g 的高效计算，算法 1 遍历 l、$a_{\cdot C}$、d、$a_{C_{-1}}$ 的所有可能组合，并对每种组合执行三个步骤：首先，计算 $g(\mathrm{l}, a_{\cdot C}, d, a_{C_{-1}})$，并检查预算是否充足（第 2 行）。如果充足，则尝试添加 d 张受贿选票来支持C_{-1}，其余选票则支持 p_1 （第 2—6 行）。g 的贿赂提供 l 张“免费”选票，而我们要为 C_{-1}增加 d 张选票。如果 $d > \mathrm{l}$，那么我们需要从 p_1增加 $d - \mathrm{l}$ 张选票。第 3 行检查这是否可行。第 4 行将剩余的 $\mathrm{l} - d$ （可能为负数）加到 p_1的支持度中。接下来，我们检查贿赂是否成功（第 5—7 行）：第 5—6 行根据阈值调整 p_1 的席位。最后，第 7 行检查贿赂是否满足所需的目标支持度和目标比例（φ 和 ρ）。如果满足，则在第 8 行断言预算充足。否则，算法继续下一次迭代。如果任何一次迭代都没有找到有效的贿赂，算法将断言预算不足（第 9 行）。

□

算法 1 用于 J+F-ob 的$-贿赂算法

1: **对所有** $0 \le \mathrm{l}, a_{\cdot C}, d, a_{C_{-1}} \le n$ **执行**
2: **如果** $g(\mathrm{l}, a_{\cdot C}, d, a_{C_{-1}}) \le B$ **则**
3: **如果** $d \le \mathrm{l}$ **且** 最小成本$(p_1, d - \mathrm{l}) \le B - B'$ **则**
4: $a_{p_1} = N^E(p_1) + \mathrm{l} - d$
5: **如果**$(a_{p_1} < T)$ **然后**
6: $a_{p_1} = 0$
7: **if** $\frac{a_{C_{-1}} + a(p_1)}{a_{C_{-1}} + a(p_1) + a_{\cdot C}} \ge \varphi$和 $\frac{a(p_1)}{a_{C_{-1}} + a(p_1)} \ge \rho$**然后**
8: **返回**预算充足。
9: **返回**预算不充足。

B.2. 交换与联盟转移贿赂，$\tau = 0$

定理 3.2。 $\langle B|$交换$|J+F|\tau=0\rangle$ 与 $\langle B|$联盟移位$|J+F|\tau=0\rangle$ 均为多项式时间可解。

证明。 我们证明，可以通过求解$O(n^2)$个最小费用流（MCF）问题实例来判定该问题，最小费用流问题定义如下。

定义 6（最小费用流（*MCF*））。给定：一个有向图 $G = (U, D)$，源点和汇节点 $s, t \in U$，每条边 $e \in D$ 对应非负的容量函数 $cap(e)$ 和成本函数 $cost(e)$，以及需求量 d。目标是找到一个流 $f : D \to \mathbb{R}$，使加权成本 $\sum_{e \in D} f(e) cost(e)$ 最小，同时满足：*(i)* 容量约束：$0 \le f(e) \le cap(e)$，对所有 $e \in D$ 均成立；*(ii)* 流量守恒：$\sum_{u \in V} f(u, v) = \sum_{u \in V} f(v, u)$，对 $v \in V \setminus \{s, t\}$ 均成立；*(iii)* 满足需求量：$\sum_{u \in V} f(s, u) = \sum_{u \in V} f(u, t) = d$。

最小费用流可在多项式时间内求解（例如参见 [43]）。我们将说明，如何给定 $k_1, k_{C_{-1}}$，构造一个最小费用流实例 $M_{k_1, k_{C_{-1}}}$，使其存在一个加权成本为 B' 的解，当且仅当存在一个成本为 B' 的贿赂 $\hat{\succ}$，满足 $N^E(p_1) = k_1$ 且 $N^{\hat{E}}(C_{-1}) = k_{C_{-1}}$。

对于每个 i，令 $\hat{\succ}_{i,p_1}$ 表示将 v_i 中的 p_1 移至首位（且不移动其他任何政党）的贿赂。同样，令 $\hat{\succ}_{i,C_{-1}}$ 和 $\hat{\succ}_{i,\bar{C}}$ 分别表示将 v_i 中 C_{-1} 的一名成员，以及 $\bar{C}$ 移至首位的最小成本贿赂。请注意，如果无需贿赂，则 $\hat{\succ}_{i,\cdot}$ 可以为 $\succ_i$。此外，为了确定 $\hat{\succ}_{i,C_{-1}}$ 和 $\hat{\succ}_{i,\bar{C}}$，我们只需考虑将单一政党移至首位且不作其他改动的贿赂。因此，对于交换贿赂和联盟移位贿赂，这些贿赂及其费用都可以在多项式时间内计算。

图 M_{k_1, k_2}、容量和成本的定义如下（见图 1）。图节点包括：一个源节点 s 和一个汇点 t，三个节点 a_1，$a_{C_{-1}}$ 和 $a_{\bar{C}}$；对于每位选民 v_i，有四个节点：v_i、$u_{i,1}$、$u_{i,C_{-1}}$ 和 $u_{i,\bar{C}}$。边、容量、成本和需求量如下：

- 边（s，a_1）、（s，$a_{C_{-1}}$）、（s，$a_{\bar{C}}$），成本均为 0，容量分别为容量$(s,a_1) = k_1$、容量$(s,a_{C_{-1}}) = k_2$、容量$(s,a_{\bar{C}}) = n - k_1 - k_2$。
- 对于每个 i，边（a_1，$u_{i,1}$）、（$a_{C_{-1}}$，$u_{i,C_{-1}}$）和（$a_{\bar{C}}$，$u_{i,\bar{C}}$），容量均为 1，成本均为 0。
- 对于每个 i，边 $(u_{i,\alpha}, v_i)$，其中 $\alpha \in \{1, C_{-1}, \bar{C}\}$，其成本$(u_{i,\alpha}, v_i)$ 为 $= Cost_i(\succ_i, \hat{\succ}_{i,\alpha})$，容量为 1。
- 对于每个 i，边 (v_i, t) 的成本为 0，容量为 1。
- 需求量为 $d = n$。

现在我们将证明，$M_{k_1, k_{C_{-1}}}$ 存在加权成本为 B' 的解，当且仅当存在一个成本为 B' 的贿赂 $\hat{\succ}$，使得 $N^E(p_1) = k_1$ 且 $N^E(C_{-1}) = k_{C_{-1}}$。

令 f 为一个求解 $M_{k_1, k_{C_{-1}}}$ 且成本为 B′ 的流。我们构造一个贿赂 $\hat{\succ}$，其费用与引理所述相同。由于所有输入均为整数，我们可以假定 f 是整数流。由于需求量已满足，且 $cap(v_i, t) = 1$，因此对于所有 i，必有 $f(v_i, t) = 1$。于是，对于每个 i，恰有一个 $\alpha_i \in \{1, C_{-1}, \bar{C}\}$ 满足 $f(u_{i,\alpha_i}, v_i) = 1$。因此，定义 $\hat{\succ}_i = \hat{\succ}_{i,\alpha_i}$。根据构造，该贿赂的成本为 $Cost_i(\succ_i, \hat{\succ}_{i,\alpha_i})$，整个贿赂的总成本为 $\sum_{i=1}^{n} Cost_i(\succ_i, \hat{\succ}_{i,\alpha_i})$——这也就是流 f 的加权成本。现在，由于需求量 $d = n$ 已满足，$f(s, a_1) = k_1$ 且 $f(s, C_{-1}) = k_{C_{-1}}$。因此，恰有 k_1（分别为 $k_{C_{-1}}$）个索引 i 满足 $f(a_1, u_{i,1}) = 1$（分别为 $f(a_{C_{-1}}, u_{i,C_{-1}}) = 1$）。所以，$N^E(p_1) = k_1$ 且 $N^E(C_{-1}) = k_{C_{-1}}$。

另一方向的证明类似。

接下来，对于每个 $0 \le k_1, k_{C_{-1}} \le n$、$k_1 + k_{C_{-1}} \le n$，使得 ~~$k_1 + k_2 \ge \varphi$~~ 且 ~~$k_1 k_1 + k_2 \ge \rho$~~，构造最小费用流实例 M_{k_1}、k_2。求解 M_{k_1}、k_2。如果所得解的成本为 $\le B$，那么根据上述断言，还存在一个成本相同的贿赂 $\hat{\succ}$，它在 $\tau = 0$ 时求解相应的贿赂 C+F-op 实例。因此，返回“成功”。如果所有 M_{k_1}、$k_{C_{-1}}$ 都不存在成本为 $\le B$ 的解，则断言“无解”。需要检查的 $k_1, k_{C_{-1}}$ 对数量为 $O(n^2)$。 □

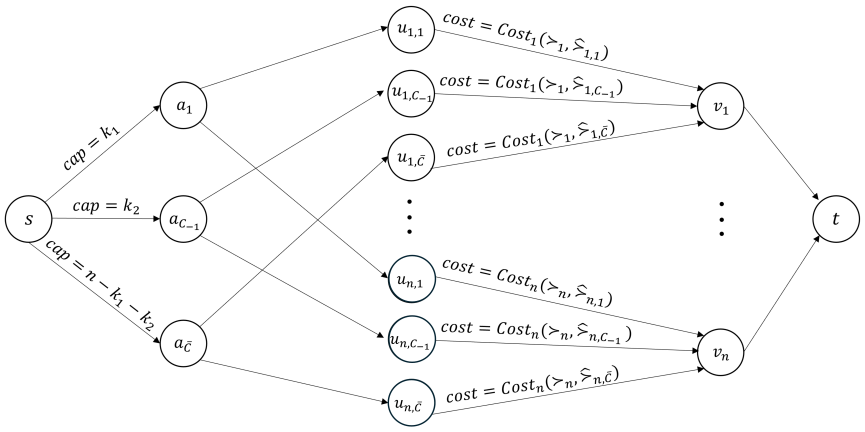

图 1：定理 3.2 证明所用的网络（为方便起见在附录中重现）。除非另有说明，默认值为 成本 = 0和 容量 = 1。容量 表示容量。

C. 第 4 节（选民控制）的证明

定理 4.1。 $\langle C|AV|J+F|\tau \geq 0\rangle$和 $\langle C|DV|J+F|\tau \geq 0\rangle$均为多项式时间可解。

证明。 我们从添加选民的情况开始。算法遍历 $r \in \{0 \ldots |W|\}$ 的所有可能取值，并对每个取值确定是否可以通过恰好添加 r 名选民来实现所需目标。

我们首先描述特定 r 值下的过程。令 $T_r = \lceil \tau \cdot (n + r) \rceil$ 表示有 r 名额外选民时的选举门槛。若某政党的选票数为 $\geq T_r$，则称该政党是 T_r-活跃的 。令 成本(p_i,s) 表示从 W 中添加投票给 p_i 的 s 名选民所需的最小成本。如果不存在这样的 s 名选民，或 $s < 0$，则 成本(p_i,s) $= \infty$。按照 W 中投票给 p_i 的选民的成本进行排序后，便可在多项式时间内计算 成本(p_i,s)。

令 g(r, a_1,$a_{C_{-1}}$, a -C) 表示满足以下条件的集合 $W^+ \subseteq W$ 的最小成本：(1) $|W^+| = r$；(2) 对于 $\hat{E} = (P, \hat{V}, \hat{\succ})$，其中 $\hat{V} = V \cup W^+$，$\hat{\succ} = (\succ_v)_{v \in \hat{V}}$，$p_1$的有效选票数为 a_1，C_{-1}的有效选票数为 $a_{C_{-1}}$，且 $\bar{C}$ 的有效选票数为 a -C。

现在我们将证明，对于任意 $r = 0, \ldots, |W|$ 以及 $a_1, a_{C_{-1}}$, a -C $\in [n + r]$, $\bar{}$，都可以在多项式时间内计算 g(r, $a_1, a_{C_{-1}}$, a -C)。为证明这一点，令 f_r(i, l, l, s) 表示满足以下条件的集合 $W^+ \subseteq W$ 的最小成本：(1) $|W^+| = s$；(2) W^+中的选民投票支持 $\{p_2, \ldots, p_i\}$；(3) 根据 $V \cup \hat{W}$ 的选票，$\{p_2, \ldots, p_i\} \cap C_{-l}$, $\bar{}$ 拥有 l 张 T_r- 有效选票，且 $\{p_2, \ldots, p_i\} \cap C$ 拥有 l 张 T_r- 有效选票。

$\bar{}$接下来，$f(i, l, l, s)$ 可以通过动态规划计算，如下所示。对于 $i = 1$，$f(1, l, s) = 0$当 $l = s = 0\bar{}$时；否则，$f(1, l, l, s) = \infty$。对于 $i = 2$，..... $|C|$:

$$f_r(i, \ell, \bar{\ell}, s) = \min\left\{f_r(i - 1, \ell, \bar{\ell}, s), \min_{T_r - N^E(p_i) \leq s' \leq s} (cost(i, s') + f(i - 1, \ell - (N^E(p_i) + s'), \bar{\ell}, s - s'))\right\}$$

对于$i > |C|$:

$$f_r(i, \ell, \bar{\ell}, s) = \min\left\{f_r(i - 1, \ell, \bar{\ell}, s), \min_{T_r - N^E(p_i) \leq s' \leq s} (cost(i, s') + f(i - 1, \ell, \bar{\ell} - (N^E(p_i) + s'), s - s'))\right\}$$

接下来,

$$g(r, a_1, a_{C_{-1}}, a_{\bar{C}}) = \min_{a_1 - N^E(p_1) \leq s \leq r} \left(cost(a_1 - N^E(p_1), s) + f_r(|P|, a_{C_{-1}}, a_{\bar{C}}, r - s)\right)$$

对于$a_1 \geq T_r$ 或$a_1 = 0$。否则$g(r,a_1,a_{C_{-1}}) = \infty$。

接下来，为了判定该问题，遍历所有取值为 r $\in \{0, \ldots, |W|\}$ 的 r；对于每个这样的 r，再遍历所有满足 $a_1, a_{C_{-1}}$、a -C $\in [n + r]$ 的 a。若存在这样的三元组，使所有目标均成立（即：a_1+$a_{C_{-1}}$$a_1$+$a_{C_{-1}}$+$a_{\bar{C}}$ ~~$\geq q$~~、$a_1$$a_1$+$a_{C_{-1}}$ ~~$\geq p$~~，且 g(r, $a_1, a_{C_{-1}}$, , a -C) $\leq$ B），则返回“是”，否则返回“否”。至此，添加选民算法完成。

对于删除选民的情形，首先注意到，上述算法对负的选民成本同样适用。因此，我们可以采用以下简单变换，将一个删除选民实例归约为添加选民实例。给定一个删除实例 $V, \mathrm{cost}(\cdot), B$，定义 $\tilde{V} = \emptyset$, $\tilde{W} = V$, $\tilde{\mathrm{cost}}(v) = -\mathrm{cost}(v)$（对所有 v），以及 $\tilde{B} = B - \mathrm{cost}(V)$。现在，若添加选民实例的一个解为 $\tilde{W}^+$，则 $V^- = V \setminus \tilde{W}^+$ 是原实例的一个解。为此，只需注意 $\mathrm{cost}(V^-) = \mathrm{cost}(V) + \tilde{\mathrm{cost}}(W^+)$。因此，$\tilde{\mathrm{cost}}(W^+) \leq \tilde{B} = B - \mathrm{cost}(V)$ 意味着 $\mathrm{cost}(V^-) \leq B$，正如所需。

□