arXiv:2601.15855v1 [cs.GT] 22 Jan 2026

如何操纵议会：席位分配选举中的策略性竞选*

Robert Bredereck[1] Piotr Faliszewski[2] Michał Furdyna[2]Andrzej Kaczmarczyk[2] Joanna Kaczmarek[4] Martin Lackner[3] Christian Laußmann[4] Jörg Rothe[4,*] Tessa Seeger[4]

[1]克劳斯塔尔工业大学，克劳斯塔尔-采勒费尔德，德国[2]AGH大学，克拉科夫，波兰[3]维也纳工业大学，维也纳，奥地利[4]杜塞尔多夫海因里希·海涅大学，数学与自然科学学院，杜塞尔多夫，德国

*通讯作者：rothe@hhu.de

摘要

在议会选举中，政党竞逐数量有限且通常固定的席位。大多数议会通过席位分配方法组建，这些方法根据各政党的得票数分配席位。常见的席位分配方法包括除数序列法（如德洪特法或圣拉古法）、最大余数法和得票最高者当选制。许多国家设有选举门槛，以防止规模很小的政党进入议会。此外，一些国家的席位分配制度还纳入了多个选区。我们研究，要通过说服有限数量的选民改变选票来改变选举结果（即增加或限制某个指定政党的影响力），在计算上有多困难。我们将这类贿赂式攻击称为策略性竞选活动，并从计算复杂性（包括经典复杂性和参数化复杂性）的角度研究相应问题。我们还基于真实选举数据开展了大量实验，研究最优竞选活动的有效性，尤其是将其与启发式贿赂策略进行比较，并考察选举门槛和选区数量的影响。最后，对于设有选举门槛的席位分配选举，我们提出一种标准首选模式的替代方案——第二次机会模式；在该模式下，

*本文整合并大幅扩展了两篇会议论文，其中一篇发表于第*29*届国际人工智能联合会议（IJCAI'20）的会议论文集 [9]，另一篇发表于第*49*届计算机科学理论与实践最新趋势国际会议（SOFSEM'24）的会议论文集 [32]。我们实验所用的代码和数据可在 https://github.com/bredereck/Strategic-Campaigns-in-Apportionment-Elections 找到。

未达到选举门槛的政党的选民可以再次投票，改投其他政党；我们还针对这一设置得出了计算复杂性结果。

K关键词： 席位分配方法、贿赂攻击、计算复杂性

1 引言

席位分配选举（也称议会选举或政党名单选举） [3, 39]是最常见的议会选举方式之一，例如在奥地利、新西兰、波兰、西班牙、土耳其及许多其他国家均有采用。在这类选举中，选民根据自己的偏好投票；具体而言，取决于选票形式，他们可以只选出自己最支持的政党，也可以对政党进行排序。随后统计得票数，并依据这些数据，采用一种*席位分配方法* 为每个政党分配特定数量的席位。[1] 席位分配完成后，议会议员代表选民提出并讨论法案，然后就法案进行投票。因此，议会尽可能按比例代表选民至关重要。

毫无疑问，最简单的方法是让获得总选票数 $1/x$ 的政党，恰好获得议会总席位数的 $1/x$。然而，由于这个数通常不是整数（而且席位既不能共享，也不能拆分），席位分配方法必须更加精细。人们提出了多种旨在实现代表性的席位分配方法。其中最著名的包括德洪特法（又称杰斐逊法）、圣拉盖法（又称韦伯斯特法）和最大余数法（又称汉密尔顿法）；这些方法已被提出并广泛采用（例如，参见 Balinski and Young 的经典教材 [3]）。我们将在第 2.2 节正式介绍这些方法，并通过示例加以说明。

在许多国家，基本的席位分配程序还会纳入所谓的*法定选举门槛* （简称*阈值*）。阈值规定了政党在获得任何席位之前必须得到的最低选票数（或占总选票的最低比例）。例如，在德国、波兰和苏格兰，政党必须获得至少 5% 的总得票数，才能参与席位分配程序。选举门槛有助于政府迅速组建，并通过尽量减少议会碎片化的影响（即减少议会中的政党数量），促进有效决策（关于机械效应和心理效应如何减少碎片化，参见 Pellicer 和 Wegner [37] 的研究）。毫无疑问，议会中的政党越少，越容易高效地达成妥协。然而，门槛的一个缺点是，支持未能跨过门槛的政党的选民在议会中完全没有代表，因为他们的选票会被直接忽略。例如，在2019年法国欧洲议会选举中，超过19%的选票因5%的门槛而作废，也就是说“五张选票中，只有四张有效，另一张则被视为无效而舍弃” [35,第 30]页。

[1]请注意， 此外，还有一些程序规定哪些特定的政党成员可以进入议会，但我们将忽略这个问题。

此外，有些国家（例如波兰）划分为多个选区，并在各选区分别举行席位分配选举；随后将这些选举的结果汇总。另一些国家（例如奥地利）则只有一个全国性的选区。[2]

在本文中，我们研究所谓的策略性竞选。在这类情形中，外部代理人打算通过在一定预算内贿赂选民、促使其改变选票，来使选举结果朝有利于自己的方向改变。也就是说，外部代理人试图改变有限数量的选票，以确保其支持的政党在议会中至少获得期望席位数（建设性情形），或通过确保其厌恶的政党至多获得期望席位数，来限制该党的影响力（破坏性情形）。策略性竞选在政治选举中日益受到关注，例如，如今处理来自社交网络、搜索引擎等的大量数据的能力，使人们能够预测个人的投票行为，并通过个性化的（政治）广告精准地锁定他们。

传统上，这类问题——即外部代理人以一定成本、在给定预算范围内改变选民偏好——属于贿选 问题 [6, 17,18], ，但它们也有其他更温和、或许更实用的解读。下面列举两种解读（例如，Elkind 和 Faliszewski [14], Faliszewski 等人 [22], Xia [41], Bredereck 等人 [10], 以及 Baumeister 等人 [7]均采用过）：首先，考虑在某次选举前开展的政治竞选。根据民意调查数据，各政党的领导人可能希望知道应该劝说哪些选民投票支持本党，也就是说，他们可能希望找到获得额外席位的最有效方法。其次，在选举结束后，人们可能希望核查各政党距离获得额外席位究竟有多近。如果这个数值足够小，那么请求进行选后审计或许是合理的。

尽管显然策略性竞选在实践中能够奏效，而且这类尝试已经在现实世界中得到应用，但关于选举结果能够被改变的幅度，以及开展合适的竞选活动有多容易，人们所掌握的具体知识仍然相当有限。我们通过研究此类策略性竞选的计算复杂性，即判定是否存在成功竞选以及寻找最优竞选活动的复杂性，来增进对这一问题的理解。

相关工作 本文合并并扩展了两篇会议论文的成果，一篇由 Bredereck 等人发表 [9] ，另一篇由 Laußmann 等人发表 [32]。席位分配方法在数学和政治学领域已得到深入研究（例如，参见 Balinski and Young 的经典著作 [1, 3] 以及 Pukelsheim 的著作 [39]），但在计算机科学和计算社会选择领域受到的关注要少得多——计算社会选择是一个交叉领域，一方面连接《理论计算机科学》和《人工智能》，另一方面连接经济学和社会选择理论；该领域研究选举的计算性质，并且例如应用于多智能体系统。

在计算社会选择领域，关于贿选（又称策略性竞选）的研究由Faliszewski 等人 [18, 19] 发起。贿选与操纵

[2]尽管奥地利有多个选区，但在最终席位分配中，该国被视为一个单一选区。

选举中的策略性选民以及选举控制都有关联。Bartholdi 等人 [4] 针对非加权选举和建设性情形下的单个操纵者，首次提出并研究了操纵问题（其目标是使指定候选人获胜）；Conitzer 等人 [13] 则针对加权选举中的联盟操纵，同时研究了建设性和破坏性情形（在破坏性情形下，策略性选民组成的联盟旨在阻止其不喜欢的候选人在加权选举中获胜）。关于操纵的更多背景，可参阅Conitzer 和 Walsh [12] 以及Baumeister 和 Rothe [6]撰写的书籍章节。Bartholdi 等人 [5] 在建设性情形下首次提出并研究了选举控制，Hemaspaandra等人 [28] 则研究了破坏性情形。在这一情形中，选举主席通过添加、删除或划分选民或候选人等行动改变选举结构，以实现建设性或破坏性的控制目标。Faliszewski 和 Rothe [17] 以及Baumeister 和 Rothe [6] 在其书籍章节中进行了综述，指出贿选和控制已针对广泛的投票规则得到研究。

尽管贿选和控制主要是在单赢家和多赢家投票规则的背景下受到研究，但关于席位分配选举中策略性竞选的文献还不多，尽管可以说，在许多国家，这类选举是最重要的选举。不过，席位分配可以视为基于认可的多赢家投票的一种特殊情形，[11, 31], 尤其是在不设选举门槛时。因此，对于政党数量较少的单选区设置，可以使用 Faliszewski 等人 [22]提出的固定参数可解贿选算法。不过，我们的算法更快，也更直接（关于操纵基于认可的多赢家规则的其他方法，参见例如 Yang [42]、Peters [38], 以及 Lackner 和 Skowron [31]的研究）。

Güney [26] 的研究与我们的工作最为相关：作者研究了一个与我们类似的竞选活动管理问题（具体针对德洪特法），但他没有提供多项式时间算法，而是将其建模为混合整数线性规划。该论文表明，土耳其议会选举（采用多选区制）容易受到策略性竞选活动管理的影响。最后，Ostapenko 等人 [36] 构建了一个与此略有关联的竞选活动管理博弈论模型。

我们的贡献我们针对以下问题（及其变体），在单选区和多选区情形下提出了高效算法：给定一场席位分配选举，其中包含一组政党、一份选票清单、一个阈值、席位数、一个指定政党和期望席位数，需要转移多少选票（即成本）才能使指定政党至少获得（建设性情形）或至多获得（破坏性情形）期望席位数？为了将其表述为判定问题，我们还给定一个预算，所花费的成本不得超过该预算。我们还研究了另一类问题：要使指定政党的席位数多于其他任何政党，即使该政党成为选举的获胜者（建设性情形），或使其不再是最强政党（破坏性情形），所需的最低成本是多少？事实上，有时以这种方式赢得选举对某个政党而言，可能比单纯增加席位更有利，因为其最强对手的席位数可能更多。我们重点研究除数席位分配法（包括著名的德洪特法

和圣拉格法）以及最大余数法。作为一种边界情形，我们还研究得票最多者当选制。

我们发现，对于单选区设置，我们的所有问题都可以在多项式时间内求解。根据席位分配方法的不同，这些算法涵盖了从直接的迭代过程到复杂的动态规划。对于多选区情形，复杂性谱系则更加有趣：在建设性情形下，虽然可以使用单选区算法，使指定政党至少获得给定数量的席位，但对于所有方法而言，使其获得最大数量席位的问题都是 NP困难的。在破坏性情形下，我们也得到了类似结果，唯一例外是：在得票最高者当选制下，确保指定政党不会获得最大数量席位的问题可以在多项式时间内求解。我们还证明了按选区数量进行参数化时，这些结果具有 W[1]-困难性，从而进一步加强了上述结论。

我们通过在真实选举数据上开展一系列实验来补充研究。首先，我们考察使用最优算法相较于采用简单启发式贿赂策略究竟有多大价值。我们发现，在大多数情况下，以最优方式转移选票比采用简单贿赂策略更加有效（且优势显著）：通常，后者所需的选票数量是最优策略的 1.2 到 3 倍，而在某些情况下甚至超过 6.5 倍。在第二个实验中，我们研究阈值对策略性竞选有效性的影响。我们发现，策略性竞选可以利用选举门槛并从中显著受益：当某个政党刚好超过选举门槛时，会在一些阈值处出现“尖峰”，此时只需贿赂 0.25% 的选票，在建设性情形下，便有时可以使指定政党额外获得全部席位的 4.4%，甚至 12.5%。在破坏性情形下，我们在某个政党刚好低于选举门槛的阈值处也观察到了类似结果。第三，我们考察选区数量如何影响改变选举结果的难易程度（以成本计）。结果表明，对于建设性和破坏性情形，选区越少，为获得（相应地，夺走）一定数量的席位所需的选票转移就越多；例如，当选区数量减少到四分之一时，建设性情形下需要改变的选票数量最多超过原来的 3.5 倍，而破坏性情形下则几乎达到原来的 6 倍。

作为第三项贡献，我们提出并研究了对常规选举门槛席位分配程序的一种简单扩展：支持得票未达到阈值的政党之选民，可以将自己的选票重新投给一个得票超过阈值的其余政党。这些选民因此获得了第二次机会。我们给出了复杂性结果，表明这一修改会使相应问题变得难以处理（即 NP困难），无论是在单选区情形还是多选区情形下。

2 预备知识

对于整数 t，我们用 $[t]$ 表示集合 $\{1,\dots,t\}$，用 $[t]_0$ 表示集合 $\{0\}\cup[t]=\{0,1,\dots,t\}$。我们使用艾弗森括号记号，即对于逻辑表达式 P，我们写作 $[P]$：若 P 为真，则其值为 1；若 P 为假，则其值为 0。

2.1 有用的计算问题

为了在后续分析席位分配程序相关问题的计算复杂性时展示各种计算困难性结果，我们将使用三个基本判定问题进行归约。第一个是三次顶点覆盖，它是顶点覆盖的一个简化变体，Garey 等人证明它是 NP完全的 [24]。

三次顶点覆盖	
给定：	一个图 $G = (V,E)$，其中每个顶点恰好有三个邻接顶点，以及一个正整数 k。
问题：	是否存在一个大小至多为k 的G的顶点覆盖，即能否选出至多k个G 中的顶点，使得G的每条边都至少与其中一个顶点相接?

在若干证明中，我们将使用一元装箱的一个变体。它本身是经典装箱问题的变体：所有数字都采用一元编码。值得注意的是，一元装箱不仅是 NP困难问题 [23] ，当以箱子数量为参数时，它还是 W[1]困难问题 [29]。具体而言，我们定义该问题的变体 UNARY-BIN-PACKING*；问题名称中的星号表示，我们对该问题的定义与经典定义略有不同。

一元装箱*	
给定：	箱子容量b，一个由$U = \{u_1,u2,\ldots,u_n\}$构成的集合，其中包含$n \geq 3$个以一元编码表示的偶数正整数采用一元编码，使得对所有 $i \in [n],$ ，$u_i \leq b$ ，并且箱子数量 $k \geq 3$ 满足 $\sum_{i=1}^{n} u_i = kb$。
问题：	是否可以将U 中的所有元素装入这k 个箱子中，且不超过任意一个箱子的容量 bin?

我们的定义与经典定义的不同之处在于：假设U中元素的值均为偶数，数字n和k的下界均为 3，并且U中的所有元素恰好能装满所有箱子。为简化证明及其阐述，在不失一般性的前提下，我们作出这些（标准）假设；不难看出，略加简化的问题变体仍然是 NP困难的，并且是 W[1]-困难的。

最后，我们定义著名的击中集问题，该问题已被 Karp [30]证明为 NP 难问题。

击中集	
给定：	一个集合 $U = \{u1,\ldots,u_p\}$，一个由 U 的非空子集组成的集合族 $S = \{S1,\ldots,S_q\}$，以及一个整数 K，$1 \leq K \leq \min\{p,q\}$。
问题：	是否存在一个击中集 $U' \subseteq U$，$\lvert U'\rvert \leq K$，也就是说，是否存在一个大小至多为 K 的集合 U'，使得 $U' \cap S_i \neq /0$ 对每个 $S_i \in S$ 都成立?

2.2 席位分配程序

选举 E $=(\mathscr{P},\mathscr{V})$ 由 m 个政党 $\mathscr{P}=\{\mathrm{P1},\ldots,\mathrm{P}_m\}$ 以及 n 张选票 $\mathscr{V}$组成，这些选票对 $\mathscr{P}$中的政党进行了排序。对于政党 X $\in\mathscr{P}$，我们用 $\mathscr{P}_{-X}$ 表示集合 $\mathscr{P}\setminus\{\mathrm{X}\}$。$\mathscr{V}$中的每张选票都对政党进行了严格排序，从最偏好到最不偏好；我们记作 A $\succ_v$ B，表示选民 v 偏好政党 A 而非 B（上下文明确时省略下标 v）。有时我们将选民 v 最偏好的政党称为 v 的首选。我们可以从一场选举推导出一个席位分配问题 I $=(\sigma,\tau,\kappa)$，它由支持度分配 $\sigma\colon \mathscr{P}\to\mathbb{N}$、阈值 $\tau\in\mathbb{N}=\{0,1,2,\ldots\}$和席位数 $\kappa\in\mathbb{N}$组成。支持度分配将政党映射为其在选举中获得的支持度数值。令 (p1, ..., p_m) 表示选票分布，即 p_i 表示以 $\mathrm{P}_i\in\mathscr{P}$为首选的选票数量。除非另有说明，我们假设政党 P_i 的支持度为

$$\sigma(P_i)=\begin{cases}p_i & \text{if } p_i\geq\tau\\ 0 & \text{otherwise,}\end{cases}$$

也就是说，当P_i 获得至少 τ 个首选时，P_i 作为首选所获得的选票数；否则为 0。也就是说，获得的首选少于 τ 个的政党，其选票将被忽略，选民也没有机会更改选票。[3] 这确保了未达到阈值的政党不会通过我们的席位分配方法获得席位。我们将这种设置称为首选模式。第 4 节将提出另一种模式（我们称之为第二次机会模式）。

例 1（首选模式下的支持度分配）。考虑这场选举

$$E=(\{P_1,\ldots,P_5\},\{v_1,\ldots,v_{2052}\})$$

根据选票

$$\begin{aligned}
v_1,\ldots,v_{604}&: \quad P_1\succ P_2\succ P_3\succ P_5\succ P_4\\
v_{605},\ldots,v_{819}&: \quad P_2\succ P_1\succ P_5\succ P_4\succ P_3\\
v_{820},\ldots,v_{1174}&: \quad P_3\succ P_2\succ P_1\succ P_4\succ P_5\\
v_{1175},\ldots,v_{1474}&: \quad P_1\succ P_5\succ P_3\succ P_2\succ P_4\\
v_{1475},\ldots,v_{1652}&: \quad P_4\succ P_2\succ P_3\succ P_1\succ P_5\\
v_{1653},\ldots,v_{1800}&: \quad P_2\succ P_4\succ P_3\succ P_1\succ P_5\\
v_{1801},\ldots,v_{2000}&: \quad P_1\succ P_3\succ P_4\succ P_5\succ P_2\\
v_{2001},\ldots,v_{2052}&: \quad P_5\succ P_2\succ P_3\succ P_4\succ P_1
\end{aligned}$$

[3]请注意，也可以采用其他支持度函数，例如使用波达计数法的支持度函数。不过，人们绝不会这样做。波达计数法的问题在于，如果所有人的偏好顺序都是$P_1\succ P_2\succ\cdots\succ P_m$（也就是说，所有人都确信$P_1$ 是最好的政党），该政党仍然只能获得很少的席位。事实上，它的席位数与$m-1$成正比，而其他政党获得的席位总数则与$O(m^2)$成正比。有人可能会说，拿一个政党与所有其他政党相比并不公平，但这里讨论的是完全一致的情况。因此，其他政党竟然能够联合起来对抗这个显而易见且获得一致支持的获胜者，实在相当矛盾。

以及席位分配问题 $I = (\sigma, 100, 6)$，也就是说，阈值为 100，我们要分配6 个席位。总计，P_1 在1104 张选票中排在第一位，P_2 在363张选票中排在第一位，P_3 在355张选票中排在第一位，P_4 在178张选票中排在第一位，而 P_5 的首选票数为52 张。由于 P_5 未达到阈值，其支持度被设为零。其他所有政党均达到阈值，因此其支持度就是各自的首选票数。总体而言，在首选模式下，这五个政党的支持度分配如下：

$$\begin{aligned}\sigma(P_1) &= 1104,\\ \sigma(P_2) &= 363,\\ \sigma(P_3) &= 355,\\ \sigma(P_4) &= 178,\\ \sigma(P_5) &= 0.\end{aligned}$$

我们作出一个非常自然的假设：选票数多于政党数，也多于席位数。请注意，现实中选举门槛通常以百分比形式表示为相对阈值（例如，5% 的阈值）。不过，我们可以轻松地将这种相对阈值转换为定义所要求的绝对阈值。

给定一个席位分配问题，我们可以通过席位分配方法确定席位分配 α：$\mathscr{P} \to \{0, \ldots, \kappa\}$必须满足 $\sum_{A \in \mathscr{P}} \alpha(A) = \kappa$，也就是说，恰好将 κ 个席位分配给政党。请注意，如果没有政党达到阈值，则席位分配没有定义；如果出现平局，席位分配方法可能会产生多个席位分配结果（在现实应用和我们的计算问题中，将假定采用特定的平局裁决规则，因此我们的席位分配规则实际上具有确定性）。

本文主要考察除数序列法（重点关注德洪特法和圣拉盖法）、最大余数法以及得票最高者当选制。

除数序列法 除数序列方法由一个（有限）序列 $\mathrm{d} = (\mathrm{d}_1, \mathrm{d}2, \ldots, \mathrm{d}_\kappa) \in \mathbb{R}^\kappa$定义，其中对所有满足 $\mathrm{i} < \mathrm{j}$ 的 $\mathrm{i}、\mathrm{j} \in \{1, \ldots, \kappa\}$，$\mathrm{d}_i < \mathrm{d}\,\mathrm{j}$，且 $\mathrm{d}_1 \geq 1$。我们要求每个值 d_i 都能在关于 i 的多项式时间内计算得出，并且关于 i 是多项式有界的（事实上，标准的除数序列方法使用非常简单的除数序列）。对于每个政党 $\mathrm{P} \in \mathscr{P}$，我们计算其分数列表 $\left[\frac{\sigma(P)}{1}\mathrm{d}, {}^{\sigma}(\mathrm{P})\,\mathrm{d}2, \ldots, {}^{\sigma}(\mathrm{P})\,\mathrm{d}_\kappa\right]$。接着，我们遍历所有政党的分数列表，找出 κ 最高的值。如遇并列，则存在多个正确的席位分配方案。每个政党在 κ 最高的值中，每有一个属于其列表的值，就获得一个席位。

德洪特法 （又称杰斐逊法）的除数序列为 1、2、3、.....；圣拉古法 （又称韦伯斯特法）的除数序列为 1、3、5、.....。此外还有其他除数序列方法，例如亨廷顿–希尔法、亚当斯法和迪恩法 [3]，但本文不再进一步讨论。

例 2 (德洪特法)。 考虑例 *1* 中的选举和席位分配问题。然后，将各政党的支持度分配除以除数序列 1、2、3、.....，

由此得到的德洪特分数列表为

$$
\begin{array}{lll}
P_1 & : & [\mathbf{1104},\ \mathbf{552},\ \ \mathbf{368},\ \ \mathbf{276},\ \ 220.8,\ 184\ \ ], \\
P_2 & : & [\ \mathbf{363},\ 181.5,\ 121,\ \ \ 90.8,\ \ 72.6,\ \ 60.5], \\
P_3 & : & [\ \mathbf{355},\ 177.5,\ 118.3,\ \ 88.8,\ \ 71,\ \ \ 59.2], \\
P_4 & : & [\ 178,\ \ 89,\ \ \ 59.3,\ \ 44.5,\ \ 35.6,\ \ 29.7], \\
P_5 & : & [\ \ \ 0,\ \ \ 0,\ \ \ \ 0,\ \ \ 0,\ \ \ 0,\ \ \ 0\ \ ].
\end{array}
$$

所有政党的分数序列中最高的 $\kappa = 6$ 个数值以粗体标出。因此，政党 P_1 获得四个席位，政党 P_2 和 P_3 各获得一个席位，而政党 P_4 和 P_5 均未获得席位。

请注意，采用德洪特法或圣拉古法等席位分配方法时，即使某个政党的选票足以超过阈值，它也可能在议会中一个席位都得不到。也就是说，我们只能得出这样的结论：如果一个政党的选票不足以达到阈值，它就不会获得任何席位。然而，反过来的推论并不成立。前面的例子也说明了这一点：政党P_4的支持度大于零，即超过了阈值，但采用德洪特法时却没有获得任何席位。

最大余数法（LRM） 最大余数法（又称汉密尔顿法）直接基于公平份额 $\mathrm{fs}(\mathrm{P}_i) = \kappa \cdot \sigma(\mathrm{P}_i)\ \mathrm{n}$，其中 $\mathrm{n} = \sum_{P_i \in \mathscr{P}} \sigma(\mathrm{P}_i)$（也就是说，在首选模式下，它就是超过阈值的政党所得选票数）。直观而言，公平份额是该政党在完全比例代表制议会中本应获得的席位数。遗憾的是，这个数通常不是整数，因此现在必须对各政党的公平份额进行取整。因此，每个政党 P_i 至少分配到 $\mathrm{lqu}(\sigma(\mathrm{P}_i), \mathrm{n}\ \kappa) = \lfloor \kappa \cdot \sigma(\mathrm{P}_i)\ \mathrm{n} \rfloor$ 个席位，即 $\mathrm{fs}(\mathrm{P}_i)$ 的整数部分（我们称之为下限配额席位）。剩余席位分配给余数值 $\mathrm{rem}(\sigma(\mathrm{P}_i), \mathrm{n}\ \kappa) = \kappa \cdot \sigma(\mathrm{P}_i)\ \mathrm{n} - \lfloor \kappa \cdot \sigma(\mathrm{P}_i)\ \mathrm{n} \rfloor$ 最大的政党（如果还有 r 个席位待分配，则余数值最高的 r 个政党各获得一个席位；我们称之为余数席位）。[4]

例 *3*（*LRM*）。考虑例 *1* 中的选举和席位分配问题。回顾一下，支持度分配为 $\sigma(P_1) = 1104, \sigma(P_2) = 363, \sigma(P_3) = 355, \sigma(P_4) = 178, and \sigma(P_5) = 0$。此外，回顾一下 $n = 2000$。首先，我们计算各政党的公平份额（四舍五入到小数点后两位）：*P1* 有 $6 \cdot \frac{1104}{2000} = 3.31$ 个席位，P_2, 有 $6 \cdot \frac{363}{2000} = 1.09$ 个席位，*P3* 有 $6 \cdot \frac{355}{2000} = 1.07$ 个席位，P_4, 有 $6 \cdot \frac{178}{2000} = 0.53$ 个席位，P_5 有 $6 \cdot \frac{0}{2000} = 0$ 个席位。根据下配额，P_2 获得三个席位，而 P_3 和 P_1 各获得一个席位。还剩一个席位；由于政党 P_4 的余数最大，为 *0.53*，因此该席位分配给 P_4。

[4]虽然可能性不大，但确实存在这种情况：足够多的政党的余数值恰好为零，因此低于阈值的政党可能因平局决胜而获得一个余数席位。因此，我们将所有低于阈值的政党排除在余数席位的分配之外。

得票最高者当选制（简单多数制）最后，在给定的选区中，得票最高者当选制会将所有席位都分配给最强政党，也就是支持度最高的政党。简单多数制不是一种席位分配方法（因为它不符合比例原则）。不过，我们将其视为一种极端情况；考虑到它在美国总统选举中的应用——每个州各自举行选举，获胜者赢得全部选举人票（用我们的话说就是席位）——以及英国通过选举议会议员进入下议院的情况，因此仍将其纳入考量。请注意，由于我们假设至少有一个政党达到阈值，因此无论是否存在阈值 $\tau > 0$，简单多数制的结果都没有差别。

示例4 (简单多数制**)**。 再次考虑示例*1*中的选举和席位分配问题。由于 P_1 的支持度最高，即为最强政党，因此所有 $\kappa = 6$ 个席位都分配给 P_1。

2.3 策略性竞选与席位分配贿选问题

现在我们定义策略性竞选，并将其建模为贿赂情景。下文假设席位分配方法具有确定性，即采用平局裁决规则并返回单一的席位分配结果。具体而言，我们考虑字典序平局裁决（除非另有说明），即平局时优先选择索引最小的政党（这简化了部分算法的表述；我们的结果也可以轻松适用于其他平局决胜顺序）。

设 $\mathscr{R}$ 为一种席位分配方法。在我们的（建设性）贿选问题中，给定一场选举 E、一个阈值 τ、[5] 一个席位数 κ、一个预算 K，以及指定政党 P^*的期望席位数 ℓ。通过最多贿赂 K 名选民，使其改变选票以支持我们（即我们可以随意修改其选票），我们希望确保政党 P^* 至少获得 ℓ个席位。

$\mathscr{R}$-席位分配-贿选（$\mathscr{R}$-AB）	
给定：	一次选举 $E = (\mathscr{P}, \mathscr{V})$、一个阈值 τ、席位总数 κ、一个指定的政党 P 类$^* \in \mathscr{P}$、期望席位数 ℓ、 $1 \leq \ell \leq \kappa$，以及预算 K， $0 \leq K \leq \lvert\mathscr{V}\rvert$。
问题：	是否存在成功竞选，也就是说，是否可以让 P 类* 获得至少 ℓ席位使用席位分配方法 $\mathscr{R}$，通过在 $\mathscr{V}$ 中更改至多K 张选票?

该问题的破坏性变体定义如下：

$\mathscr{R}$-破坏性-席位分配-贿选（$\mathscr{R}$-DAB）	
给定：	一次选举 $E = (\mathscr{P}, \mathscr{V})$、阈值 τ、席位总数 κ、一个指定政党 $P^* \in \mathscr{P}$、期望席位数 $\ell, 0 \leq \ell \leq \kappa$ 和预算 K， $0 \leq K \leq \lvert\mathscr{V}\rvert$。
问题：	是否存在成功的破坏性竞选？也就是说，能否通过改变最多 $*$ 张选票，使 P^* 获得至多 ℓ 个席位，所用的席位分配方法为 $\mathscr{R}$，且选票属于 $\mathscr{V}$?

[5]如果某次席位分配选举没有阈值， τ可以直接设为零。

请注意，在破坏性情形下，我们要求 $0 \leq \ell \leq \kappa$，因此指定政党甚至可能完全不被允许获得任何席位。此外，由于 $\ell \leq \kappa \leq |\mathscr{V}|$ 和 $K \leq |\mathscr{V}|$，对于复杂性分析而言，κ、ℓ以及K 采用二进制编码还是一元编码并不重要。然而，我们假设选民人数采用一元编码，因为我们主要关注议会选举，在这类选举中，每张选票都对应一名个人投出。因此，选票数的规模是合理的（通常不超过约 10^9）。此外，每张选票都以独立选票的形式实体存储。

可以对刚刚定义的问题稍作修改，要求在指定的贿赂预算范围内确定一种选票分布，使指定政党获得的席位多于其他每个政党，即赢得选举。请注意，与此前定义的问题不同，这些问题的输入因此不包含期望席位数：

$\mathscr{R}$-席位分配-获胜者-贿选（$\mathscr{R}$-AWB）	
已知：	一场选举 $E = (\mathscr{P}, \mathscr{V})$、一个阈值 τ、席位总数 κ、一个指定政党 $P^* \in \mathscr{P}$，以及预算 K，$0 \leq K \leq \lvert\mathscr{V}\rvert$。
问题：	是否存在成功的获胜竞选，即是否可以确保 $\alpha(P \text{ 类}^*) > \alpha(P \text{ 类}_i)$，使得对于所有 P 类$_i \in \mathscr{P}_{-P^*}$，通过席位分配方法 $\mathscr{R}$，最多更改 $\mathscr{V}$ 中的 K 张选票？

该问题的破坏性变体以类似方式定义，即询问是否可以确保 P 类* 不 赢得选举：

$\mathscr{R}$-破坏性-席位分配-获胜者-贿选（$\mathscr{R}$-DAWB）	
给定：	一次选举 $E = (\mathscr{P}, \mathscr{V})$、一个阈值 τ、席位总数 κ，以及一位特定的政党$P^* \in \mathscr{P}$，以及预算K，$0 \leq K \leq \lvert\mathscr{V}\rvert$。
问题：	是否存在一种成功竞选，能够阻止P^*获胜，也就是说，是否可以确保至少存在一个政党P_i，满足 $\alpha(P^*) < \alpha(P_i)$，并通过席位分配方法 $\mathscr{R}$，最多更改K 张选票？

一些国家（例如波兰）被划分为多个选区，并在每个选区分别举行席位分配选举。随后将这些选举的结果汇总。我们将多个选区的问题变体建模如下。

$\mathscr{R}$-多选区席位分配贿选（$\mathscr{R}$-多臂老虎机）	
已知：	选区数量 q、选举列表 $[E_1, E2, \ldots, E_q]$，其中 $E_i = (\mathscr{P}_i, \mathscr{V}_i)$，以及相应的阈值 τ_i 和席位数 κ_i, $1 \leq i \leq q$，一个指定政党 P 类$^* \in \mathscr{P} = \bigcup_{q\, i=1} \mathscr{P}_i$，期望席位数 ℓ, $1 \leq \ell \leq \kappa = \sum^{q}_{i=1} \kappa_i$，以及预算 K，$0 \leq K \leq \sum^{q}_{i=1} \lvert\mathscr{V}_i\rvert$。
问题：	是否存在成功竞选，即是否可以让P 类* 获得至少 ℓ个席位使用席位分配方法 $\mathscr{R}$，在每个选区中分别通过总共改变至多大K张选票，使 $\mathscr{V}_1, \ldots, \mathscr{V}_q$，其中每个政党的席位数是所有选区席位数之和？

同样，破坏性变体的定义也与此类似。

$\mathscr{R}$-破坏性-多选区-席位分配-贿选（$\mathscr{R}$-DMAB）	
给定：	选区数量q，选举列表 $[E_1,E2,\ldots,E_q]$，其中$E_i=(\mathscr{P}_i,\mathscr{V}_i)$，以及 相应的阈值 τ_i 和席位数 κ_i, $1\le i\le q$，指定政党 $P^*\in\mathscr{P}=\bigcup_{q\,i=1}\mathscr{P}_i$，期望席位数 ℓ, $0\le\ell\le\kappa=\sum^q{}_{i=1}\kappa_i$，以及预算$K$, $0\le K\le\sum^q{}_{i=1}\lvert\mathscr{V}_i\rvert$.
问题：	是否存在成功的破坏性竞选活动，即是否可以通过改变 每个选区中最多 ℓ 个席位的席位分配方法 $\mathscr{R}$，使 P 获得 总计至多 K 张选票，在 $\mathscr{V}_1,\ldots,\mathscr{V}_q$ 中，其中每个政党的席位数是以下各项之和 所有选区中的席位数?

与之前一样，我们还考虑相应的获胜者问题：要求P^* 获得的席位多于其他每个政党，以及其破坏性变体。同样需要注意的是，在获胜者变体中，这些问题的输入不包含期望席位数。

$\mathscr{R}$-多选区席位分配获胜者贿选（$\mathscr{R}$-MAWB）	
给定：	选区数量q、选举列表 $[E_1,E2,\ldots,E_q]$，其中 $E_i=(\mathscr{P}_i,\mathscr{V}_i)$，以及 相应的阈值 τ_i 和席位数 κ_i, $1\le i\le q$，一个指定政党 P 类$^*\in\mathscr{P}=\bigcup_{q\,i=1}\mathscr{P}_i$，以及预算$K$，$0\le K\le\sum^q{}_{i=1}\lvert\mathscr{V}_i\rvert$。
问题：	是否存在一场成功的获胜竞选，即是否可能确保 $\alpha(P$ 类$^*)>\alpha(P$ 类$_i)$ 对每个选区单独使用席位分配方法 $\mathscr{R}$，对所有 P 类$_i\in\mathscr{P}_{-P^*}$ 总共最多更改 K 张选票，在 $\mathscr{V}_1,\ldots,\mathscr{V}_q$ 中，其中每个政党的席位数为 所有选区的席位数之和?

$\mathscr{R}$-破坏性-多选区-席位分配-获胜者-贿选（$\mathscr{R}$-DMAWB）	
给定：	选区数量 q、选举列表 $[E_1,E2,\ldots,E_q]$，其中 $E_i=(\mathscr{P}_i,\mathscr{V}_i)$，以及 相应的阈值 τ_i 和席位数 κ_i, $1\le i\le q$，一个指定政党 $P^*\in\mathscr{P}=\bigcup_{q\,i=1}\mathscr{P}_i$，以及预算 K，$0\le K\le\sum^q{}_{i=1}\lvert\mathscr{V}_i\rvert$。
问题：	是否存在一种成功竞选，可以阻止P^*获胜？也就是说，是否有可能 确保至少存在一个政党P_i，其 $\alpha(P^*)<\alpha(P_i)$，并在每个选区单独使用席位分配方法 $\mathscr{R}$，同时通过改变选票，总共至多改变K 张选票 $\mathscr{V}_1,\ldots,\mathscr{V}_q$，其中每个政党的席位数是所有选 区席位 数之和?

3 经典首选模式

现在，我们介绍经典首选模式下的结果，首先从单选区情形开始。

3.1 单选区情形

我们首先讨论单选区情形，即考虑$\mathscr{R}$-AB 和 $\mathscr{R}$-AWB 问题。对于这两个问题，我们为所有分配

方法都找到了多项式时间算法。作为热身，我们先来考虑简单多数制；对于这两个问题，一个简单的贪心算法就足够了。

命题 1. 简单多数制-AB 和简单多数制-AWB 均属于P 类。

证明。 在简单多数制下，得票数最高的政党获得全部 κ 席位，因此同一个算法适用于我们的两个问题。考虑给定席位分配问题的支持度分配，并令K 为贿赂预算。我们的算法执行K 轮迭代，每轮按如下步骤进行：

1. 我们找出当前得票数最高的政党，记为P_i（若出现并列，则优先选择索引最小的政党；若该党是指定政党P^*，则终止并接受），然后

2. 我们将一张选票从政党P_i 转给P^*。

如果最后政党P^* 的得票数最高，则接受；否则拒绝。正确性和多项式运行时间很容易验证。 □

类似的方法也适用于破坏性变体：我们不再将选票从最强政党转给P^*，而是在每轮迭代中将选票从P^* 转给当前最强的政党。

命题 2. 简单多数制-DAB 和简单多数制-DAWB 都属于 P 类。

证明。 同样，由于在简单多数制下，得票数最高的政党获得全部 κ 席位，因此同一算法适用于这两个问题。设 σ 为给定席位分配问题中的支持度分配，K 为贿赂预算。我们的算法执行 K 轮迭代，每轮按如下步骤进行：

1. 我们找出当前得票数最高的政党 $P_i \in \mathscr{P}_{-P^*}$（若有并列，则优先选择索引最小的政党；如果 $\sigma(P_i) > \sigma(P^*)$，则终止并接受），并且

2. 我们将一张选票从政党P^* 转给 P_i。

如果最终政党P^* 并非得票数最高的政党，则接受；否则，拒绝。正确性和多项式运行时间同样显而易见。

□

遗憾的是，贪心算法似乎不适用于其他席位分配方法，因此我们对这些方法采用动态规划。对于除数序列法和 LRM，我们提出了几乎完全相同的算法。[6] 我们算法的一般结构是：首先，在贿赂预算允许的范围内，尽可能多地向政党 P^*增加选票，然后

[6]注意，Bredereck 等人 [9] 此前提出了一种不处理阈值的替代性多项式时间算法。

我们计算必须从哪些政党中取得这些选票，以确保P^*获得所需数量的席位。

$\mathscr{R}$-AB 的多项式时间可计算性证明（其中 $\mathscr{R}$ 是一种除数序列方法）依赖于以下引理。

引理1。 以下两个结论对所有除数序列法均成立。

(i) 通过贿选至多 K 张选票，政党 P^* 最多可获得的额外席位数，总能通过恰好将 K 名选民从 $\mathscr{P}_{-P^*}$ 中的政党转移到 P^*来实现。*(ii)* 通过贿选至多 K 张选票，我们从政党 P^* 最多能移除的席位数，总能通过恰好将 K 名选民从 P^* 转移到 $\mathscr{P}_{-P^*}$ 中的政党来实现。

证明。 注意，当一个政党的支持度增加时，该政党的分数列表值会增加；而当其支持度降低时，这些值则会减少。

我们先证明第一个结论。假设我们找到了一种说服选民改变选票的办法（下文称为贿赂操作），使得P^* 获得X 个额外席位。

情况 1：除P^* 之外，还有其他政党获得了额外选票。设$P_i \neq P^*$ 是这样的政党。现在考虑将所有原本转给P_i 的选票改为转给P^* 。显然，P^*的分数列表值会增加，而P_i 的分数列表值会减少。因此，在这次贿赂操作所产生的修改后的选举中，P^* 获得的席位至少与原始选举中一样多。

情况 2：选票仅从 $\mathscr{P}_{-P^*}$ 中的政党转给P^*。将（部分）选票转给政党P_i ，而非P^*，会使P^* 的分数列表值减少，而P_i 的分数列表值增加。因此，在这次贿赂操作所产生的修改后的选举中，P^* 不可能比原始选举获得更多席位。

最后请注意，根据除数序列法的单调性，向P^* 转移更多选票绝不会使P^* 失去任何席位。因此，我们可以将全部预算K 用于将选票转给P^*。结合上述两种情况可知，只需将K 张选票从 $\mathscr{P}_{-P^*}$ 转给P^* ，就总能实现可能获得的最多额外席位（尽管也可能存在其他同样好的方案）。

要证明第二个断言，只需交换P^* 与其他政党的角色。 □

引理1对于我们多项式时间算法的正确性至关重要，因为它意味着，在建设性情形下，我们应当用尽全部预算K ，将选票从其他政党转移到P^* ；在破坏性情形下，则应当用尽全部预算，将选票从P^* 转移到其他政党。也就是说，我们无需考虑在 $\mathscr{P}_{-P^*}$内部转移选票。

现在我们介绍我们的算法（算法 1），并对其进行直观描述。随后，我们将证明该算法确实是正确的，并且能在多项式时间内运行。为提高可读性，在不失一般性的情况下，在本第 3.1 节的其余部分中，我们假设P_1 是我们的指定政党，即$P^* = P_1$ （只要存在指定政党）。

直观地说，我们首先将 K 设为 $n - p_1$ 和 K 中的较小值，因为这是我们能够从其他政党转移给 P_l的选票的最大数量。如果 P_1 凭借这个新的 K 不可能达到阈值，那么我们已经可以回答 NO，因为 P_1 根本不会获得任何席位。算法的关键部分是计算字典 γ。正如算法 1 中所述， γ 被定义为一个数组列表 $\gamma[P]$: 对于每个政党 $P \in \mathscr{P}$，其对应数组 $\gamma[P]$ 中的条目是（可实施的）二元组 (x, 成本)，其中成本是必须从政党 P 中移除的选票的最小数量，使得在 P_l获得第 ℓ个席位之前，P 仅获得 x 个席位，并假设最终 P_1 恰好获得 K 张额外选票。稍后我们将在定理 1 的证明中说明如何高效地计算 γ。现在，我们定义政党 $\mathscr{P}_{-P_1}$上的一个顺序 o。该顺序可以是任意的；我们只是利用它将每个政党对应到表中的一行，而我们现在开始填充这张表。对于每个 i， $1 \leq i \leq |\mathscr{P}_{-P_1}|$，以及每个 s， $0 \leq s \leq \kappa - \ell$，单元格 tab[i][s]包含从政党 o(1).....o(i) 转移走的选票的最小数量，使得 o(1).....o(i) 总共获得 s 个席位，并且在 P_1 被分配第 ℓ个席位之前（再次假设最终 P_1 恰好获得 K 张额外选票）。这张表也可以通过动态规划高效计算，具体说明将在后文给出。最后，我们检查表格最后一行中是否存在一个不超过 K 的值。如果存在，我们就回答 YES，因为确实存在不超过 K 的贿赂，并能确保其他政党将第ℓ个席位留给 P_l。

注意，通过回溯表 tab，我们可以找出需要从各政党转移给P_1 的具体选票数，从而实现成功竞选。这个数量在许多情况下并不等于K 。如果如此，我们只需从任意政党（P_l除外）移除剩余选票。

定理 1. 对于每种除数序列方法 $\mathscr{R}$, $\mathscr{R}$-AB ， 其均属于P 类。

证明。 算法 1 可在建设性情形下判定是否存在成功竞选。该算法适用于每一种除数序列方法。我们首先证明该算法确实能在多项式时间内运行。算法的大部分步骤都很容易看出这一点：本质上，我们要填充一个有 $|\mathscr{P}|$ 行、至多 κ 列的表。由于 $\kappa \leq |\mathscr{V}|$，表的规模确实关于输入规模为多项式。然而， γ 的计算方式尚不明确。计算 γ 时，会使用二分搜索来寻找函数 ϕ 的跳跃点；该函数定义为：在P_1 最终恰好获得K 张额外选票的假设下，一个拥有y 张选票的政党在P_1 获得 ℓ 个席位之前能够获得的席位数。令P_1 的最终得票数为q （即计入K 张额外选票后）。那么，对于序列为$d = (d_1, d2, \ldots, d_\kappa)$ 的除数序列方法，我们有

$$\phi(y) = \begin{cases} 0 & \text{if } y < \tau \\ 0 & \text{if } y \leq q/d_\ell \\ \max\{z \in \{1, \ldots, \kappa\} \mid y/d_z > q/d_\ell\} & \text{otherwise.} \end{cases}$$

跳跃点可通过二分搜索在 $\mathscr{O}(\kappa \cdot \log(K))$ 时间内找到。

现在我们证明该算法的正确性。首先，将K 设为$n - \sigma(P_l)$和K中的较小值是必要的，以确保我们不会将超过允许数量的选票从 $\mathscr{P}_{-P_1}$ 中的政党转移到P_1 ； 具体而言，若将K 设得更大，就会导致

假阳性结果。在本证明余下部分中，我们假设所有K张选票都从$\mathscr{P}_{-P_1}$转移到P_1,，也就是说，最终P_1会额外获得K张选票。根据引理1，这是最优的。第一个 if 语句会在P_1无法达到阈值时返回 NO。这个回答是正确的，因为只要P_1低于阈值，就永远无法获得任何席位；也就是说，在这种情况下，贿赂不会成功。

在算法的中间部分，我们填充一个表。回想一下，对于每个 i，$1 \leq i \leq |\mathscr{P}_{-P_1}|$，以及每个 s，$0 \leq s \leq \kappa - \ell$，单元格 tab$[i][s]$ 记录了在P_1获得其第 ℓ 个席位之前，为使政党 $o(1)\ldots,o(i)$ 总共获得 s 个席位而需要从这些政党中移除的最少选票数。这些数值通过从前一行到后一行动态计算得出。这之所以可行，是因为在P_1获得其第 ℓ 个席位之前，政党 $o(1)\ldots,o(i)$ 总共获得的席位数，恰好等于这些政党各自在P_1获得其第 ℓ 个席位之前的席位数之和。此外，由于可以通过直接比较某个政党的除数列表与P_1的除数列表来计算这个数值（也就是说，每个政党的函数都独立于其他政党的支持度），所需的贿赂预算也恰好等于各项贿赂之和。因此，表中的数值确实计算正确。

最后，如果最后一行中存在一个不大于 K 的值，我们就可以根据以下论证正确地回答“是”。假设单元格 tab$[|\mathscr{P}_{-P_1}|][s]$ 中有一个不大于K的值。那么，存在一些贿赂操作，其成本不超过 K，并能确保在P_1获得第 ℓ 个席位之前，其他政党至多获得 s 个席位。由于总共有 κ 个席位可供分配，而且在P_1获得第 ℓ 个席位之前，其他政党获得了 $s \leq \kappa - \ell$ 个席位，因此P_1确实会获得第 ℓ 个席位。然而，如果最后一行的所有单元格中的值都大于 K，给定的预算就不足以确保在P_1获得第 ℓ 个席位之前，其他政党至多获得 $\kappa - \ell$ 个席位。因此在这种情况下，其他政党至少会获得 $\kappa - \ell + 1$ 个席位，留给P_1,的至多只有$\ell - 1$个席位，所以我们正确地回答“否”。 □

我们可以轻松地将算法 1 调整用于破坏性情形。现在，我们从政党 P_l 中移除 min{K $\sigma(P_l)$} 选票，并将这些选票分配给其他政党。当然，如果 P_l 跌破阈值，我们就立即回答“是”。对于破坏性情形，ϕ 和 γ 的定义需要略作调整。这里，我们将 ϕ(y) 定义为：拥有 y 张选票的政党在 P_l 获得第 $(\ell + 1)$ 个席位（这是我们试图阻止的席位）之前获得的席位数。而 γ[P] 中的 (x, cost) 定义为：为了使政党 P 至少在 P_l 获得第 $(\ell + 1)$ 个席位之前获得 x 个席位，我们需要向政党 P 添加的最少选票数。同样，我们通过动态规划填充该表，但这一次，每当一行全部填完后，我们就检查是否能通过贿选，使所有已填行对应的政党在 P_l 获得第 $(\ell + 1)$ 个席位之前至少获得 $\kappa - \ell$ 个席位。也就是说，我们检查当前行的最后一个单元格是否填入了不超过 K 的值，或者是否能在当前行表格的最后一个单元格之后再填入一个这样的值。如果可以，我们就回答“是”，因为通过这一贿赂操作，剩余席位不足以让 P_l 获得第 $(\ell + 1)$ 个席位。如果始终无法做到这一点，我们就回答“否”，因为我们至多只能让 $\mathscr{P}_{-P_1}$ 中的政党占据 $\kappa - \ell - 1$ 个席位，而这仍会给 P_l留下第 $(\ell + 1)$ 个席位。这就证明了以下结果。

定理 2. 对于每种除数序列方法，$\mathscr{R}$, $\mathscr{R}$-DAB 属于P 类。

注意，由于我们可以在多项式时间内判定是否存在一种成功的竞选活动：通过贿赂至多K 名选民，保证P_1 至少获得 ℓ 个席位，因此我们也可以在多项式时间内，利用简单的二分搜索，找出在预算为K 的情况下，能够保证P_1 获得的最大席位数。类似地，对于破坏性变体，也可以高效地确定我们能从P_1手中夺走的最大席位数。我们在第3.3节通过实验展示了这一点，测试了这些竞选活动在现实世界选举中的有效性。

现在，我们证明此前所有结果的 LRM 对应版本，首先是引理1的 LRM 对应版本。

引理2。 以下两个命题对 *LRM* 成立。[7]

(i) 增加支持指定政党 P_1 的选票，绝不会导致 P_1 失去席位。此外，减少 P_1 的选票，绝不会使 P_1 获得席位。*(ii)* 增加对 P_1 以外其他政党的选票，导致 P_1 席位数增加的幅度，不会超过将相同数量的支持者直接增加给 P_1时的幅度。反过来，减少其他政党的选票，导致 P_1席位数减少的幅度，不会超过直接从 P_1移除相同数量的支持者时的幅度。

证明。 第一个断言很容易看出，它表明 LRM 满足某种“单调性”。对于第二个断言，请注意，增加或减少选票会改变n ，从而改变所有政党的公平份额。实际上，给P_1 以外的其他 政党增加一张选票，有时（虽然很少见）会因n值变化导致余数改变，而使P_1 比之前获得更多席位。[8] 不过，请注意，如果把相同的选票加给P_1 而不是其他政党，也会得到相同的n 。在这种情况下，P_1的公平份额甚至更高，因此P_1获得的席位不会少于之前。反向情形也可用类似的论证，因为从P_1 移除选票，对P_1席位数的减少，至少与从其他政党移除选票造成的减少一样多。 □

接下来，我们给出定理 1 的 LRM 类比结果。

定理 3。*LRM*-AB 属于P 类。

证明。 我们证明算法 2（显然可在多项式时间内运行）能够正确判定 LRM-AB。首先注意，根据引理2，我们应尽可能多地将选票转移到P_1。如果P_1 仍未达到阈值，我们可以回答“否”，因为P_1 不可能获得任何席位。

否则，while 循环现在会先检查是否可以在不让其他政党跌破阈值的情况下，使P_1 获得目标席位；然后检查在恰好让一个政党跌破阈值的情况下是否可行，依此类推。请注意，某个政党

[7]注意，将其他政党的选票转移到P_1 ，以及将P_1 的选票转移到其他政党时，这两个陈述同样成立。
[8]政党P_1的余数可能比一个大得多的政党的余数少减少一些。

P_i在从P_i中移除 $\sigma(P_i) - \tau + 1$ 张选票后，会跌破阈值。这种循序渐进的方法很重要，因为每当我们让一个政党跌破阈值时，n 都会发生变化，仍在竞逐的所有政党的公平份额也会随之变化。在 while 循环的每次迭代中，我们会区分三种情况：

情况 1：如果P_1 通过下配额获得了 ℓ 个席位，我们就可以回答“是”，因为将选票从哪些政党转移到P_1并不重要。

情况 2：如果P_1 按下配额获得的席位少于 $\ell - 1$ 个，我们就无法在当前迭代中判定该实例，因此会移除下一个政党（即将其支持度降至阈值以下），并继续下一轮迭代。这是正确的，因为P_1 最多只能通过策略性地将其他政党的选票转移给P_1，再获得一个额外的（余数）席位。

情况 3：最后，在第三种情况下，P_1 按下配额恰好获得 $\ell - 1$ 个席位。这是最棘手的部分，因为现在我们必须检查是否能够策略性地转移选票，使P_1 拥有最大的余数之一，从而获得一个余数席位。

> 这采用了与我们处理除数序列方法时类似的动态规划方法。不过，这次 γ的定义略有不同。也就是说，在 γ[P]中，(x, 成本)表示必须从政党 P 移除的最少选票数，使得在 P_1 获得余数席位之前，P 仅获得 x 个席位（假设最终 P_1 恰好多出 K 张选票），且 P 仍高于阈值。再次通过对函数ϕ的跳跃点进行二分搜索来计算γ的值。这里，ϕ(y)定义为拥有 y 张选票的政党在 P_1获得余数席位之前所获得的席位数；也就是说，当余数高于 P_1, 的余数时，取下配额，否则取下配额加一。我们通过动态规划填充一个包含 $\kappa - \ell$ 列的表。如果在表的最后一行找到一个不超过预算的值，就可以回答“是”，因为该值表明，我们可以降低其他政党的支持度，使它们在 P_1 获得余数席位之前最多获得 $\kappa - \ell$ 个席位。由于 P_1 获得的席位数为 $\ell - 1$，因此至少还剩下 $\kappa - \ell -(\ell - 1)$ 个 $= 1$ 余数席位，而该席位自然归 P_1所有。如果表的最后一行中没有任何值小于预算，我们就继续移除最小的政党，因为在当前 n 值下（结合支持度分配），不可能为 P_1 提供一个余数席位。
>
> 请注意，在每次后续迭代中，都将支持度最低的政党压到阈值以下，最终我们便能涵盖所有可能的被移除政党数量（即被压到阈值以下的政党数量）。

因此，如果通过 LRM 进行的分配贿赂能够成功，那么在 while 循环的某次迭代中就会找到成功的竞选活动。如果 while 循环的所有迭代都没有返回 YES（也就是说，我们无法让更多政党跌破阈值），就可以确定通过 LRM 进行的分配贿赂无法成功，因此答案为 NO。

□

算法 2 也很容易改造为判定该问题的破坏性变体。改造方式与使用除数序列法将算法 1 改造为适用于破坏性情形的方式相同。在破坏性变体中，我们依次增加政党，也就是通过增加支持者使其超过阈值（而不是像算法 2 中那样，通过将政党推至阈值以下来移除政党）。

定理 4. *LRM*-DAB 属于P 类。

请注意，这两种算法也可以轻松改造为实际计算出一场竞选活动（如果存在的话）。

类似的算法也适用于确保P_1 获得的席位严格多于任何其他政党（但不要求席位数为特定值）的情形，即 AWB。不过，在这种情形下，算法需要更加谨慎。问题在于，尽管我们用于 AB 的算法会找出向政党P_1 分配 ℓ个席位的方法（前提是这可行），但作为副作用，它们也可能增加分配给其他政党的席位数。还有可能出现这种情况：要解决 AWB，只需以某种方式转移选票，使得席位最多的政党失去部分席位，而这些席位转而分配给P_1以外的政党。最后，我们还需要更加注意平局决胜。

Theo推论 5. 对于 $\mathscr{R}$ 为除数序列方法或 *LRM* 的情形，$\mathscr{R}$-AWB 属于 P.

证明。 设 σ为输入的支持度分配， κ为待分配的席位数，K 为贿赂预算。我们先描述除数方法且具有除数序列$d=(d_1,d2,\ldots\ldots,d_\kappa)$时的算法，然后说明针对 LRM 需要做哪些修改。

我们的算法首先猜测，对于该问题的一个是实例，P_1 最终会获得的席位数 ℓ 。在这种情况下，所有其他政党最终获得的席位都应更少。接下来，算法猜测某个政党 P_s，满足 $\sigma(\mathrm{P}_s)>0$；猜测 P_s 最终会获得的席位数 ℓ_s（$\ell_s<\ell$）；还猜测我们从 P_s 转移到 P_1 的选票数 b'_s（在整个证明中，我们假设 $\mathrm{s}\neq 1$；$\mathrm{s}=1$ 的情况可以类似处理）。其思路是，值 $\mathrm{x}=(\sigma(P_s)-\mathrm{b}'_s)/d_{\ell_s}$ 将是最终 k-除法序列中能够获得席位的最低值（平局决胜除外）。最后，我们猜测政党 P_t（$\mathrm{t}\geq\mathrm{s}$），使得对于政党 $\mathrm{P1},\ldots,\mathrm{P}_t$，k-除法序列中的值 x 足以使其获得席位；而对于政党 $\mathrm{P}_{t+1},\ldots,\mathrm{P}_m$，只有大于 x 的值才足以使其获得席位（换言之，$\mathrm{P}_t$ 是平局决胜有利的最后一个政党）。我们的目标是判断，能否从其他政党转移 K 张选票给 P_1,，从而实现上述所有设想。

为此，我们需要引入一些额外的记号。对于整数 $i\in[m]$ 以及支持度 $\sigma(P\text{ 类})$，令 $\kappa_i(\sigma(P\text{ 类}))$ 为一个整数 λ，使得 $\sigma(P\text{ 类})/d_{\lambda+1}<x$，且满足

1. 若 $i\leq t$，则 $\sigma(P\text{ 类})/d_\lambda\geq x$;
2. 若 $i>t$，则 $\sigma(P\text{ 类})/d_\lambda>x$。

直观地说，$\kappa_i(\sigma(P\text{ 类}))$ 表示政党 P 类$_i$ 在支持度为 $\sigma(P\text{ 类})$ 时可获得的席位数，前提是能够保证获得一个席位的除数序列中的最低值为 x（平局决胜除外）。

考虑一些 $i \in [m], j \leq \kappa$，以及 $b \in [K]_0$。这些值的含义是：我们考虑前 i 个政党，即 P 类1、.....、P 类$_i$，它们将被分配 j 个席位；而 b 表示我们需要从政党 P 类2、.....、P 类$_i$ 转移到 P 类$_1$的选票数。我们将布尔函数 $f(i,j,b)$ 定义为：当且仅当存在一个由非负整数构成的序列 $(b2、.....、b_i)$（直观上表示从政党 P 类2、.....、P 类$_i$ 转移到P 类$_1$的选票数），使得：

1. $\sum_{r=2}^{i} b_r = b$（即，我们恰好将b张选票从$P2,\ldots,P_i$转移到P_1）。

2. 如果$i \geq s$，那么$b_s = b'_s$且 $\kappa_s(\sigma(P_s) - b'_s) = \ell_s$（即，我们恰好将$b'_s$张选票从$P_s$转移到$P_1$,，并且$P_s$ 恰好获得 ℓ_s 个席位），

3. $\kappa(p_1 + K) \geq \ell$并且对于每个$r \in [i] \setminus \{1\}$， $\kappa_1(\sigma(P_i) - b_i) < \ell$（即，$P_1$ 至少获得 ℓ 个席位，且其他每个政党获得的席位都更少）。

4. $\kappa_1(p_1 + K) + \sum_{r=2}^{i} \kappa_r(\sigma(p_r) - b_r) = j$（即，分配给政党$j$的席位总数为$P1,\ldots,P_i$）。

我们的主算法当且仅当$f(m,k,K)$为$true$时接受，这确实意味着P_1 至少获得 ℓ 个席位，而所有其他政党获得的席位都更少。接下来需要说明如何在多项式时间内计算$f(i,j,b)$的值。

尽管函数 f 看起来可能有些复杂，但使用动态规划计算它相当容易。首先，对于所有$j < 0$和所有$b < 0$，我们令$f(i,j,b) = false$。其次，我们注意到，对于$i = 1$，只需直接按照定义计算$f(i,j,b)$。第三，对于$i \geq 2$，我们递归地表示$f(i,j,b)$。如果$i \geq 2$但$i \neq s$，那么$f(i,j,b)$等于：

$$\bigvee_{\substack{b_i \in [K]_0, \\ t_{i-1} \in T(t)}} f(i-1, j - \kappa_i(\sigma(p_i) - b_i), b - b_i) \wedge (\kappa_i(\sigma(p_i) - b_i) < \ell).$$

此外，$f(s,j,b)$等于：

$$f(i-1, j - \ell_s, b - b'_s) \wedge (\kappa_s(\sigma(p_s) - b'_s) = \ell_s).$$

利用这些递归关系和标准的动态规划技术，我们得到一个用于计算函数f的多项式时间算法。至此，除数序列法这一情形的证明完成。

对于LRM方法，我们的算法与除数序列法相同，不同之处如下：

1. 我们不再猜测除数序列中足以获得席位的最低值 x，而是猜测足以获得余数席位的最低余数值x（与除数序列方法的情况一样，我们通过猜测一个政党以及从该政党扣除的选票数来进行，然后计算其余数）。

2. 我们重新定义 $\kappa_i(\sigma(P))$，使其要么为 $\mathrm{lqu}(\sigma(P),\ n\ \kappa) + [\mathrm{rem}(\sigma(P),\ n\ \kappa) \geq y]$（适用于猜测的平局决胜结果对其有利的政党 P_i），要么为 $\mathrm{lqu}(\sigma(P),\ n\ \kappa) + [\mathrm{rem}(\sigma(P),\ n\ \kappa) > y]$（适用于平局决胜结果对其不利的政党）。

证明至此完成。 □

最后，我们指出，定理1和定理3中的算法可以调整，以适用于如下情形：对于每个政党P_i，$i \in \{2,\ldots,m\}$，都有一个以多项式为界的非递减函数 $\mathrm{cost}_i(x)$（作为输入的一部分提供），用于指定将x 张选票从P_i 转移到P_1, 的成本；我们要判断是否存在总成本至多为K的贿选。这样，我们便可以例如对说服更多选民投票支持P_1这一过程日益增加的难度进行建模。为了将这类函数纳入考量，我们会在算法中修改f 函数，使其不再返回t *rue/false*值，而是返回实现特定效果的最低成本（其中 cost ∞表示无法实现）。

获胜者贿选中的破坏性情形稍微简单一些，因为我们只需找到一个政党$P_i \in \mathscr{P}_{-P_1}$，使其获得的席位数超过我们特别关注的（也就是说，在这种情况下，我们厌恶的）政党P_1。然而，虽然简单的贪心算法足以应对除数序列法，但 LRM 需要更加谨慎。因此，我们分别给出两个证明。

命题3。 对于每种除数序列方法 $\mathscr{R}$，$\mathscr{R}$-DAWB 均属于P 类。

证明。 首先，我们检查是否存在一个政党P_i，使得 $\alpha(P_i) > \alpha(P_1)$。如果存在这种情况，我们可以立即回答“是”。否则，不失一般性地，假设对于每个P_i和P_j，$i < j$，都有 $\sigma(P_i) \geq \sigma(P_j)$，并且 P_1 是席位数最多的政党之一。现在，我们只需将全部 K 张选票从P_1 转移到P_2, ，也就是说，转移给支持度不超过P_1 的政党。[9] 如果这意味着 $\alpha(P_2) > \alpha(P_1)$，我们回答“是”；否则回答“否”。

很容易看出，该算法在多项式时间内运行。根据引理1的第二部分，我们知道，在破坏性情形下，应当用全部预算K 将选票从P_1 转移给政党$P_i \in \mathscr{P}_{-P_1}$。剩下需要证明的是：将$K$ 张选票从P_1 转给另一个支持度低于P_2 的$P_i \in \mathscr{P}_{-\{P_1,P_2\}}$，或将$K$ 张选票分配给多个政党$P_i \in \mathscr{P}_{-P_1}$，永远不会更优。这一点由引理1的第一部分可得：该部分指出，给定预算K时，政党P_i 能够获得的额外席位数最大值，总可以通过将K 张选票从 $\mathscr{P}_{-P_i}$（此处，如上所述，是从政党P_1）转移给P_i来实现。在获胜者问题的破坏性情形下，我们只需要一个政党击败P_1（即拥有更多席位），因此，与P_1 之间距离最小的政党 $\sigma(P_1) - \sigma(P_i)_1$（此处为政党$P_2$）始终是击败$P_1$所需选票数最少的政党。

□

命题4. *LRM*-DAWB 属于 P 类。

证明。 我们先分析应将选票转移到何处，再检查应从何处取走选票，以此证明该命题。不失一般性，假设

[9]如果$K > \sigma(P\text{ 类}_1)$，我们将$K - \sigma(P\text{ 类}_1)$张选票从政党$\mathscr{P}_{-\{P_1 P\text{ 类}_2\}}$转移到$P\text{ 类}_2$。

对于每个 P 类$_i$和 P 类$_j$，$i < j$</style}，有 (P 类$_i$) $\geq \sigma(P$ 类$_j)$，并且 P 类$_1$是席位数最多的政党之一，也是我们的指定政党。

首先，我们证明，将所有可能的选票转移给P_2，能最大限度地提高使P_1不成为获胜者，即不拥有最多席位的机会。设将K'张选票转给P_2，并将K''张选票转给某个P_t，其中$t > 2$满足$p_t \geq \tau$，且$K' + K'' \leq K$，同时不改变分母$n = \sum^{|\mathscr{P}|}{}_{i=1} \sigma(P_i)$。假设贿选之后

$$\frac{\sigma(P_2)+K'}{n} < \frac{\sigma(P_t)+K''}{n}.$$

但是，

$$\frac{\sigma(P_t)+K''}{n} \leq \frac{\sigma(P_2)+K''}{n} \leq \frac{\sigma(P_2)+K'+K''}{n},$$

因此，如果我们将所有选票直接转给P_2,，P_2获得的席位数不会少于P_t。如果我们还贿赂那些原本投给低于阈值的政党的选民，使其改为投给高于阈值的政党 τ τ，分母会增加某个值L，情况将与前一种情况类似。也就是说，我们可以将选票直接转给P_2：对于任何一个P_t，如果在获得额外K''张选票后，其新的支持度高于P_2，则有

$$\frac{\sigma(P_t)+K''}{n+L} \leq \frac{\sigma(P_2)+K''}{n+L} \leq \frac{\sigma(P_2)+K'+K''}{n+L}.$$

如果我们贿赂一些选民，让他们投给某些低于阈值的政党，并使这些政党在贿选后获得正的支持度，那么对于$K''' < K'$和$L' > L$，有

$$\frac{\sigma(P_2)+K'-\sigma(P_1)+K''}{n+L} - \frac{\sigma(P_2)+K'''-\sigma(P_1)+K''}{n+L'} > 0.$$

因此，这种情况下也不比直接将选票转给 P 类$_2$ 更好。最后一种情况是，我们贿赂投票给超过阈值的政党的选民，使得至少有一个政党的新支持度为零——我们称这些政党为 P 类$_r$。因此，分母减少了某个值 D。如果我们至少不将一张选票转给 P 类$_2$,，那么我们有，如果

$$\frac{\sigma(P_s)+K'}{n-D} \geq \frac{\sigma(P_2)+K''}{n-D}$$

对于任意 P 类$_s \notin \{P$ 类$_1$,P 类$_2\}$，则成立：

$$\frac{\sigma(P_s)+K'}{n-D} \leq \frac{\sigma(P_2)+K'}{n-D} \leq \frac{\sigma(P_2)+K'+K''}{n-D}.$$

因此，我们也可以改为将选票直接转给 P 类$_2$。现在来考虑上述贿赂的混合情况：对于一些K', K''，其中有 $0 \leq K', K'' < K$，一些 $L, D \geq 0$，以及一些 P 类$_t \notin \{P$ 类$_1$,P 类$_2\}$，其中有 $p_t \geq \tau$，

$$\frac{\sigma(P_t)+K''}{n+L-D} \leq \frac{\sigma(P_2)+K''}{n+L-D} \leq \frac{\sigma(P_2)+K'+K''}{n+L-D},$$

也就是说，我们同样可以改为将选票直接转给 P 类$_2$。最后请注意，对于 $K' > K''$，其中 $K' - K''$ 张选票从正支持度为正的政党转移到了阈值以下的某些政党，使这些政党仍低于阈值；并且对于 n'，即贿赂操作后、P 类$_2$ 获得额外K'张选票时的分母，有

$$\frac{\sigma(P_2)+K'}{n'} \geq \frac{\sigma(P_2)+K''}{n'-(K'-K'')}$$

当且仅当

$$n'\sigma(P_2)+n'K'-K'(K'-K'')-n'\sigma(P_2)-n'K'' \geq 0,$$

这又等价于

$$(n'-K')(K'-K'') \geq 0.$$

也就是说，初始不等式在$n' \geq K'$时成立，而这始终为真（另请注意，对于P_1, 的任何支持度，当分母较大时，其 fs(P_1) 会更小）。

现在让我们分析，是否应该贿赂所有支持 P_1 的 K 名选民，如果 $\sigma(P_1) \geq K$。如果我们贿赂他们所有人，分母将不会改变（除非这会使 P_1 的支持度变为零；但此时很容易判断 P_2 获得的席位是否多于指定政党）。假设从现在起，在我们考虑的情形中，从 P_1 转移少于 K 张选票的情况都属于成功贿赂。显然，当 P_2获得的选票多于 P_1, 时，如果我们从 P_1 转移 $K''' < K$ 张选票，那么 P_2 的新选票数与 P_1 的支持度之间的差距，小于前者与 $\sigma(P_1) - K$ 之间的差距，且 n 不变。因此，在后一种情况下，P_2 获得比 P_1更多席位的机会更大。如果我们还贿赂一些投给低于阈值的政党和／或高于阈值的政党的 $K' < K$ 名选民，同时不将这些政党的支持度降至零，而不是贿赂投给 P_1, 的选民，那么分母将增加；对于 $K'' < K$ 和 $K' + K'' \leq K$，有：

$$\frac{\sigma(P_2)+K-\sigma(P_1)+K}{n} - \frac{\sigma(P_2)+K-\sigma(P_1)+K''}{n+K'} > 0.$$

也就是说，这一策略并不比从P_1处获取更多选票更好。我们还可以将选票转移给P_2，这些选票来自支持度至少达到阈值的政党，并使这些政党的新支持度为零，即分母减少。由于P_2 的支持度不能小于P_1，否则就无法获得比P_1, 更多的席位（对于$K'' < K$ 以及某个$P_t \notin \{P_1, P_2\}$，或这类政党的某个组），因此有

$$\frac{\sigma(P_2)+K}{n-\sigma(P_t)+K-K''} - \frac{\sigma(P_1)-K''}{n-\sigma(P_t)+K-K''} \geq 0.$$

也就是说，如果 P_2 应当成为获胜者，那么 $\sigma(P_2) + K \geq \sigma(P_1) - K''$。此外，当 K'' 尽可能大，且/或 P_t 的支持度尽可能高时，这一距离也最大。对于

每个支持度需要降至 0 的联盟 P_t，只需从中取出 $p_t - \tau + 1$ 张选票即可：对于任意 $L < L'$，其中二者都足以将 P_t的支持度降至零，都有

$$\frac{\sigma(P_2) + K - \sigma(P_1) + K - L}{n - \sigma(P_t) + L} > \frac{\sigma(P_2) + K - \sigma(P_1) + K - L'}{n - \sigma(P_t) + L'}.$$

因此，只需检查支持度最低且达到阈值的m 个政党的贿选成本；我们将m 从 1 开始逐步增加，直到达到预算。

当 $\sigma(P_1) < K$ 时，我们可以将P_1的支持度降至零，也就是说，它将无法获得任何席位，只要仍有政党至少获得一个席位。因此，可以在多项式时间内判断我们能否贿选至多K 名选民，使P_2 比P_1, 获得更多席位；也就是说，LRM-DAWB 属于 P 类。

□

3.2 多选区情形

事实证明，我们可以使用单选区算法来解决 $\mathscr{R}$-多臂老虎机问题。简而言之，我们可以分别计算在每个选区中让特殊政党P^* 获得每种可能席位数所需的成本，然后求解一个背包类问题，以判断是否能通过转移至多B 张选票，让P^* 获得 ℓ 个席位（由于我们假设B 以一元编码给出，因此可以在多项式时间内完成）。

命题5。 当 $\mathscr{R}$ 为简单多数制、除数序列方法或 LRM 时， $\mathscr{R}$-多臂老虎机 属于 P 类。

证明。 考虑一个具有 δ 个选区的 $\mathscr{R}$-多臂老虎机实例，其中至多移动 K名选民，以使指定政党 P^*获得 ℓ个席位。此外，对于每个$j \in [\delta]$，令 κ_j表示在第j个选区内分配的席位数。为便于说明且不失一般性，我们假设选区采用任意但固定的编号，从 1 到 δ。

一般而言，我们的算法分为以下两个步骤：

1. 对于每个$j \in [\delta]$和每个$s \in [\kappa_j]_0$，我们计算在j个选区中需要移动的最少选票数，以确保P^* 在该选区中获得（至少）s个席位。我们将该数值记为 cost(j,s)。[10]

2. 我们检查是否存在一种贿选，使$s_1, s2$、.....、 s_δ在相应的选区中为P^*带来 δ 个额外席位，使得

   (a) $\sum_{j=1}^{\delta} s_j \geq \ell$（$P^*$ 在所有选区中总共至少获得 ℓ 个席位），且

   (b) $\sum_{j=1}^{\delta} \text{cost}(j, s_j) \leq K$（移动的选票至多为 K 张）。

如果存在这样的贿选，我们就接受；否则就拒绝。

[10]根据多选区情境下简单多数制的定义，在每个选区 δ_j 中，所有席位 κ_j均分配给选区 δ_j 中最强的政党。因此，简单多数制下，s 只能取 0 和 κ_j。请注意，对于所有研究的席位分配方法，如果多个s对应的 cost(j,s) 相同，我们只考虑较大的s 值。

显然，根据我们的定理 1或定理3，第一步可以在多项式时间内完成。因此，我们接下来要为第二步展示一个多项式时间动态规划。该程序的思路与背包问题的程序类似，但其中的物品属于互不相交的组，而且我们必须从每组中恰好选取一个物品。

我们先定义一个布尔函数 $F(j,g,c)$：如果在前$j \in [\delta]_0$ 个选区中，存在一项成本不超过$c \in [K]_0$ 的贿选，使P^* 至少获得$g \in [\ell]_0$个额外席位，则输出*true*；否则输出 *false* 。于是，当$F(\delta,\ell,K)$为*true*时，就存在所需的贿选。

出于技术原因（并略微滥用了符号），我们假定：如果$g > 0$ （P^* 在一项涉及零个选区的贿选中无法获得任何额外席位），或者$c < 0$，则$F(0,g,c)$为*false* 。反过来，对于$g \leq 0$，所讨论的F 值为*true* （自然，假定$c \geq 0$）。

根据上述解释和假设，我们给出以下递归公式来计算F的值：

$$F(j,g,c) = \bigvee_{g' \in [\kappa_j]_0} F(j-1, g-g', c-\text{cost}(j,g')).$$

本质上，上述公式表明，要检查能否在前j 个选区中，以最多K 单位的预算为P^*获得g个额外席位，我们按以下步骤进行。假设已知前$j-1$ 个选区对应的F 值，我们尝试在第j个选区中获得每一种可能数量g' 的额外席位，其成本为 cost(j,g)。对于每种尝试，我们验证能否以不超过$K -$ cost(j,g)的成本，在前$j-1$ 个选区中获得$g - g'$ 个额外席位。如果至少有一种尝试验证结果为肯定，我们就将正在计算的F值设为*true* ；否则设为*false*。

我们的方法具有正确性，这显然是因为它穷尽地考虑了所有可能导致所需贿选的情形。我们通过标准动态规划方法计算 $O(K\delta\ell)$ 个可能的 F 值，每个值的计算耗时 $O(K)$。因此，根据我们的约定，选民数量采用一元编码，可知我们的动态规划算法在多项式时间内运行。由此，我们的整个两步算法也具有同样的性质。

□

同样的方法也可用于求解破坏性情形。在此，我们将 cost(j,s) 定义为：为了确保P^*在第 j 个选区中获得至多 s 个席位，我们需要在该选区转移的最少选票数；然后，我们检查是否存在一个序列 $s1,\ldots,s_\delta$，满足$\sum_{j=1}^{\delta} s_j \leq \ell$且 $\sum_{j=1}^{\delta} \text{cost}(j,s_j) \leq K$。

命题 6。 对于 $\mathscr{R}$ 为简单多数制、除数序列方法或 *LRM* 的情形， $\mathscr{R}$-DMAB 属于 P 类。

现在我们转向多个选区的获胜者问题。遗憾的是，对于所有考察的席位分配方法，即除数序列法、LRM，甚至简单多数制， $\mathscr{R}$-MAWB 都是 NP困难的。直观地说，这是因为该问题允许我们灵活地要求给定集合中的所有政党各自失去一个席位，并划分选区，使得在每个选区中，只有这些政党中的某些子集可能失去席位。

定理 6. 设 $\mathscr{R}$ 为除数序列为 $d_1,d_2,\ldots$ 的除数序列方法，或为 LRM。$\mathscr{R}$-MAWB 是 NP-完全的，即使每个选区的席位数以及每个选区转移的选票数都至多为三，且每个选区的选票总数是一个只取决于 d_1 和 d_2 的常数。

证明。 由于除数序列法和 LRM 均可在多项式时间内计算，因此该问题属于 NP 显而易见。

我们通过从三次顶点覆盖问题进行归约来证明其 NP 困难性。下面给出洪德法席位分配方法这一情形下的归约。

构造。 设 (G,k) 为三次顶点覆盖问题的输入实例，其中 $G=(V,E)$ 是一个图，且 k 是一个整数（不失一般性，假设 $k \geq 3$）。此外，令 $V=\{v1,\ldots,v_r\}$，并令 $E=\{e1,\ldots,e_t\}$。我们按如下方式构造一个洪德法 -MAWB 实例。首先，将贿赂预算设为 $K=3k$。接下来，按照平局决胜顺序创建 $1+t+10r+1$ 个政党；也就是说，首先定义的政党也会由平局裁决规则首先选中，依此类推：

1. 令 P 类* $=P$ 类$_1$ 为我们的偏好政党。

2. 对每条边 e_i，我们构造一个边党 P 类$(e_i)=P$ 类$_{1+i}$

3. 对每个顶点 v_j，我们构造 10 个虚拟政党，因此对每个 $\ell \in [10]$，令 P 类$(v_j\ell)$ $=P$ 类$_{1+t+10(j-1)+\ell}$ 为顶点 v_j 对应的第 ℓ 个虚拟政党。

4. 我们还构造一个阻挡政党 P 类(b) $=P$ 类$_{1+t+10r+1}$。

接着，我们构造 $r+t(k-2)+(k-1)$ 个选区，每个选区的阈值均为零：

1. 对每个顶点 v_j，我们构造一个顶点选区 $d(v_j)$，其中有三个席位可分配；P 类$_1$ 获得 7 张选票，与 v_j 关联的每个虚拟政党获得 10 张选票，并且对于每条与 v_j 关联的边 e_i，政党 P 类(e_i) 获得 10 张选票。2. 我们构造 $t(k-2)$ 个虚拟选区，每个选区只有一个席位可分配。对于每个边党，恰好有 $k-2$ 个虚拟选区，其中该政党获得 $2K+1$ 张选票，所有其他政党均获得零张选票（这样，该边党便能获得席位，且成本不超过 K 的贿选无法改变这一点）。3. 我们构造 $k-1$ 个阻挡选区，每个选区只有一个席位可分配。在每个此类选区中，阻挡政党获得 $2K+1$ 张选票，所有其他政党均获得零张选票。

我们的构造描述至此完成。

初始席位分配。 现在我们来说明贿选发生前的初始席位分配。阻挡政党从阻挡选区恰好获得 $k-1$ 个席位（它在所有其他选区的选票数均为零，因此无法在其他任何选区获得席位）。同样，每个边政党从虚拟选区获得 $k-2$ 个席位。对于顶点选区，我们回顾一下，平局决胜规则优先选择政党 P_1，其次是边政党，再其次是虚拟政党和阻挡政党。现在考虑某个顶点选区 $d(v_j)$，令 e_a、e_b 和 e_c 为与该顶点相连的三条边。

与v_i相连的三条边。共有三个席位需要分配，可以验证，由于平局决胜规则，政党$P(e_a)$、$P(e_b)$和$P(e_c)$各获得一个席位。总的来说，政党P_1和虚拟政党没有获得席位，阻挡政党获得$k-1$个席位，而每个边政党获得k个席位。尤其是，P_1在未贿选举中没有获得席位多数。主要思路。我们构造背后的直觉是，预算$K=3k$足以让P_1在k个顶点选区中各获得一个席位，从而取得k个席位。我们确保当且仅当这k个选区对应于G中的一个顶点覆盖时，这一席位数才足以超过任何其他政党。

正确性。为了证明我们的归约正确，我们将证明：图G存在大小为k的顶点覆盖，当且仅当存在一种有效贿选，移动至多K张选票，并确保政党P_1获得的席位多于任何其他政党。

对于"仅当"方向，假设G存在大小为k的顶点覆盖。具体而言，设S为由k个顶点组成的集合，它构成G的一个顶点覆盖。我们按如下方式进行贿选：对于每个对应顶点$v \in S$的顶点选区，我们从与连接v的边对应的三个边政党各转移一张选票给P_1。在每个这样的选区中，P_1现在有10张选票，每个边政党至多有9张选票，而虚拟政党仍各有10张选票。因此，依据平局决胜规则，每个这样的选区中有一个席位分配给P_1，另外两个席位分配给虚拟政党（分别分配给不同的虚拟政党）。由此可直接得出，P_1获得k个席位。此外，由于每条边都至少被S中的一个顶点覆盖，每个边政党至少失去一个席位，最终获得$k-1$个席位。阻挡政党也获得$k-1$个席位，因此P_1的席位数多于任何其他政党。

对于"若"方向，假设存在一种贿选，至多转移K张选票，并确保政党p获得的席位多于任何其他政党。要胜过阻挡政党，P_1必须至少获得k个席位，这意味着平均而言，每个这样的席位至多需要转移$K/k=3$张选票。现在我们将论证，P_1每多获得一个额外席位，就必须至少（因而也恰好）转移三张选票。

根据构造，P_1只能在顶点选区中获得额外席位。反设P_1在某个顶点选区中，通过转移$y \leq 3x-1$张选票获得了$x \geq 1$个席位。由于可用席位只有三个，因此$x \leq 3$且$y \leq 8$。此外，在转移$y \leq 8$张选票后，贿选结束时该选区中得票数为10的政党必须多于三个。[11] 因此，我们注意到必须转移至少$y \geq 3$张选票，使得P_1至少能与得票数为10的政党打成平手。这意味着$x > 1$，即P_1在该选区中至少获得两个席位。然而，根据德洪特法，这要求P_1至少获得20张选票。这就产生了矛盾，因为我们必须转移$y \geq 13$张选票，而我们假设$y \leq 8$。

总而言之，我们知道，要获得k个席位，必须在恰好k个顶点选区中，每个选区恰好转移三张选票。反设相应的顶点选区不构成顶点覆盖，也就是说，存在一条边e_i，它与这些顶点均不相连。这意味着政党$P(e_i)$仍能获得k个席位（其中$k-2$个席位来自虚拟选区，另有2个席位来自顶点选区），因此，P_1获得的席位不会多于任何其他政党。

[11] 不失一般性，我们可以假设所有选票都转而支持P类，这样其他政党都不会得到超过10张选票10选票。

下面简要说明如何将证明调整为适用于其他席位分配方法。首先，我们注意到，虚拟选区和阻挡选区无需更改——它们仍能确保相应政党获得所需数量的席位。顶点选区可能需要调整，但对于我们的每种席位分配方法，只需找到一个常数 x，使得 P_1 获得 $x-3$ 张选票，而其他政党（选票数非零）各获得 x 张选票。具体来说，对于圣拉古法，我们当前使用的常数，即$x = 10$，已经足够。也可以验证，对于 LRM，$x = 10$ 也足够（具体而言，每个顶点选区有 137 名选民，因此在贿选前，这些选区只分配余数席位；若要获得两个席位，某个政党必须获得一个配额席位，而这至少需要 46 张选票，足以保证我们的证明仍然成立）。 □

破坏性情形同样是 NP 困难的。这可以通过扩展之前的归约来证明。

定理 7。 设 $\mathscr{R}$ 为除数序列为 d_1,d_2、.....的除数序列方法，或为 LRM。$\mathscr{R}$-DMAWB 是NP-完全的，即使每个选区的席位数以及每个选区转移的选票数最多为 3，且每个选区的选票总数是一个取决于 d_1 和 d_2的常数。

证明。 属于 NP 是显然的，因为除数序列法和 LRM 都可以在多项式时间内计算。

我们通过从三次顶点覆盖问题进行归约来证明 NP 困难性。与之前一样，我们针对德洪特法给出一个归约。关于如何将其调整为其他方法的具体细节，可参见定理 6 的证明。

设 (G, k) 是三次顶点覆盖问题的输入实例，其中 $G = (V, E)$，$V = \{v1, ..., v_r\}$，$E = \{e1, ..., e_t\}$，是一个图，k 是一个整数（不失一般性地，我们假设 $k \geq 3$）。我们按照定理 6 证明中的方式构造 D'Hondt-DMAWB 的一个实例，并进行如下调整。在政党的集合中加入一个辅助政党 P_1, ，其中 $P^* \neq P_1$ 是我们的指定政党。在平局决胜顺序中，将 P_1 置于 P^* 和 P_2之间。然后，我们像之前证明中的归约那样构造 $r + t(k-2) + (k-1)$ 个选区，唯一的区别是，在顶点选区中，辅助政党 P_1 获得 7 张选票。指定政党在这些选区中不获得任何选票。接下来，我们再加入两个选区，每个选区的阈值也为零。注意，由于这些选区中的每个都只包含一个政党，所有选民只能为该政党投票。因此，这些选区中有多少选民并不重要，因为无论如何席位都只能分配给唯一存在的政党。这两个额外选区定义如下：

1. 我们只在有 $3k$ 个席位需要分配的地方设立一个选区P^*。因此，P^* 获得所有席位，且这些席位不能重新分配给其他政党。

2. 只有在有 $2k+1$ 个席位可供分配的情况下，我们才会建立一个选区，并且只有P_1 类政党参与。因此，P_1 类政党获得全部席位，这些席位无法重新分配给任何其他政党。

至此，我们的构造所需的调整就完成了。

现在我们来描述初始席位分配。在前$r+t(k-2)+(k-1)$个选区中，席位按照定理 6 的证明中贿选之前的方式分配：阻挡政党从阻挡选区中恰好获得$k-1$个席位。同样，每个边党，即形式为 $\{v_i,v_j\}$ 的边党，从虚拟选区中获得$k-2$个席位，并从选区$d(v_i)$和$d(v_j)$中获得 2 个席位，即总共获得k 个席位。虚拟政党没有获得席位。P_1 只在最后一个选区获得席位，即获得 $2k+1$ 个席位；而P^* 在P^* 也是唯一被选中的政党的那个选区中获得 $3k$ 个席位。特别地，在未贿选举中，P^* 拥有最多席位。

现在我们想通过贿选，使P^* 的席位少于某个其他政党。首先注意，除了顶点选区之外，其他所有选区都无法将席位转给其他政党。因此，每个边党最多只能获得$k+4$个席位（边党恰好出现在两个顶点选区中，因此最多可在其中获得 6 个席位），每个虚拟政党最多只能获得 3 个席位，而阻挡政党最多只能获得$k-1$个席位。这意味着，只有P_1 可能比P^*获得更多席位。由于我们的指定政党P^*不能少于 $3k$个席位，P_1 必须至少再获得k 个席位，而这些席位只能来自顶点选区。因此，该问题等价于定理 6 的证明中提出的问题，由此可得 $\mathscr{R}$-DMAWB 是 NP困难问题。

□

考察 $\mathscr{R}$-MAWB 的参数化复杂性也很有意思。我们证明，对于以选区数量为参数的参数化，该问题是 W[1]-困难的。

定理 8. 设 $\mathscr{R}$ 是除数序列方法或 *LRM*。关于选区数量， $\mathscr{R}$-MAWB 是 $W[1]$-困难的。

证明。 我们给出从一元装箱*出发的参数化多项式时间归约。为此，固定一个一元装箱*实例，其中包含 $U=\{u_1,u2,\ldots,u_n\}$、箱子容量 b 以及箱子数量 k。为简洁起见，令 $\delta=\max\{n,b\}$。

对于任意（但固定的）除数序列方法 $\mathscr{R}$，其序列为 $d_1,d2,\ldots$，我们将上述实例归约为一个预算为 $K=kb$ 的新 $\mathscr{R}$-MAWB 实例，具体如下。我们创建一个偏好政党 P^*、 δ 虚拟政党，并为全集中的每个元素 u 创建一个元素政党 $P(u)$。我们构造的核心部分是一个箱子构件。箱子构件是一个有 n 个席位待分配的选区。在该选区中，偏好候选人获得d_2K-b张选票；每个虚拟政党获得 d_2K 张选票；对于每个元素 u，其政党 $P(u)$ 恰好获得 d_2K+u 张选票。我们将箱子构件复制 k 次，以此创建整个选举，并对政党进行标记，使平局决胜时偏好政党优先于虚拟政党，而虚拟政党又优先于元素政党。显然，所述实例可以在多项式时间内构造出来。

以下断言总结了箱子构件的四个关键性质。为便于阅读，我们将在证明的末尾证明断言 1。

断言 1。 考虑一个 UNARY-BIN-PACKING* 问题的实例，其中包含集合 $U=\{u_1,u2、\ldots\ldots、u_n\}$，箱子容量 b 以及数目 k。则箱子构件的选区满足：

1. 政党 P^* 在转移不超过$2b$张选票后，无法获得两个席位；

2. 至少需要向 P^* 转移 b 张选票，才能使其获得一个席位；*3.* 对于每个元素 $u_i \in U$，至少需要 $u_i/2$ 次选票转移，才能使 $P(u_i)$ 失去一个席位；*4.* 已知 P^* 获得了一个席位，存在一个元素 $u \in U$，至少需要转移 u 张选票，才能使 $P(u)$失去一个席位。

构造出的选举结果中，每个元素对应的政党恰好获得k 个席位，而优先政党一个席位也得不到。由于有k 个箱子构件，且P^* 从单个构件中至多获得一个席位，因此P^* 至多能获得k 个席位。因此，每个元素对应的政党都必须至少失去一个席位，才能让P^* 获得最多的代表。

如果存在一种装箱方案，那么对于每个箱子，我们选择一个（互不相同的）选区，并将b 张选票转给P^*，使该箱子中每个元素对应的政党都获得d_2K 张选票。这是可行的，因为每个箱子都恰好装满了大小为b 的元素，而且根据构造，每张转移的选票都对应一个单位的大小。最终，P^* 和相应元素对应的政党都获得d_2K 张选票。因此，席位将分配给其他元素对应的政党、P^*以及一些虚设候选人。对每个箱子重复这一过程后，P^* 将获得k 个席位，一些（彼此不同的）虚拟政党各获得一个席位，而每个元素对应的政党都获得$k-1$ 个席位。这显然构成一个“是”实例。

现在我们证明，事实上，在所构造的 $\mathscr{R}$-MAWB 的每个“是”实例中，政党 P^*必须恰好获得 k 个席位。如果 P^* 获得 k 个 -1 席位，那么每个元素政党都必须至少失去两个席位。根据断言 1 的第 3 点，以及我们的预算为 $K = kb$ 这一事实，我们知道，这种情况下，每个元素 $u \in U$ 所对应的政党至多需要通过 $u/2$ 次选票转移来失去一个席位。否则，选票变更的数量将超过 K。然而，P^*必须至少赢得一个席位，并且根据断言 1 的第 4 点，至少存在一个元素政党 $P(u)$，$u \in U$，通过 u 次选票转移失去一个席位。注意，事实上，对于 P^* 最终可能获得的任何其他数量 $x \in [k-2]$，这一论证同样成立。因此可知，P^* 必须获得 k 个席位。最终，根据断言 1 的第 1 点和第 2 点，以及 $K = kb$ 的取值可知，在 k 个选区中的每一个选区，恰好花费 b 个单位的预算，并且 P^*（拥有 d_2K 名选民）在所有这些选区中各获得一个席位。因此，每个元素政党至多拥有 $k-1$ 个席位。根据构造，如果 P^* 在某个选区通过贿选获得 b 张选票后拥有 d_2K 张选票，则每个元素政党 $P(u)$，$u \in U$，都必须恰好失去 u 张选票，才能在该选区失去一个席位。因此，将每个选区视为一个箱子，并将对应元素政党拥有 b 张选票的元素视为放入相应箱子的元素，我们就得到了原实例 UNARY-BIN-PACKING 的一个解。

显然，该归约在多项式时间内运行，并且参数“选区数量”仅是 一元装箱问题参数k 的函数。为完成该归约的证明，接下来只需证明箱子构件具有命题1所述的性质。

断言 *1* 的证明。我们将分别证明该断言的各个要点。在所有论证中，我们的目标都是最大化政党 P 类*获得的席位数。对于我们研究的规则，始终将选票转给政党 P 类*是最优做法，因此在所有断言的证明中，我们都采用这一假设。

开始之前，请回想一下，在一个箱式构造区中，有n个席位可供分配，而在贿选之前，这些席位中没有任何一个分配给政党P 类* 。此外请注意，除偏好政党外，至少还有 $2n$ 个政党，各自至少拥有 d_2K 张选票。

命题 *1* 的证明。为证明这一主张，我们说明，即使P^* 再获得 $2b$ 张选票，P^*也无法获得两个席位，无论哪些政党的选票被拿走。为导出矛盾，假设P^* 拥有$d_2K - b + 2$ b 张选票并获得两个席位。注意，$d_2K - 2b = b(d_2k - 2)$ 是任一政党q 在转移 $2b$ 张选票后可能拥有的最少选票数（特别是，任何虚拟政党都符合这一情况）。因此，偏好政党若要获得第二个席位，一个必要条件是：

$$\frac{b(d_2k+1)}{d_2} \geq \frac{b(d_2k-2)}{d_1}.$$

换言之，假设全部 $2b$ 张选票都从q 转给p，那么P^* 必须在q 获得第一个席位之前，先获得足够的选票以赢得自己的第二个席位（由于平局决胜规则偏向P^* 而非q，此处不等式可以取等号）。否则，选票数严格高于q 的政党数量足以分配$n-1$ 个席位，每个政党一个，从而p至多只能获得一个席位。我们将上述表达式变换如下：

$$\begin{aligned}
\frac{d_2k+1}{d_2} &\geq \frac{d_2k-2}{d_1} \\
d_1(d_2k+1) &\geq d_2(d_2k-2) \\
d_1d_2k+d_1 &\geq (d_2)^2k-2d_2 \\
kd_2(d_1-d_2) &\geq -2d_2-d_1.
\end{aligned}$$

由于右侧为负数，且k 和d_2 必须为正数，因此$d_1 - d_2 \geq 0$，这与除数序列d_1,$d2,\ldots$递增这一事实矛盾。最终，我们得出结论：要使政党P^* 在一个构造区中至少获得两个席位，需要超过 $2b$ 个预算单位。

命题 *2* 和 *4* 的证明。 假设我们执行$b - x$ 次选票转移，使P^* 拥有$p^* = d_2K - x$ 张选票。

我们首先证明，当 $\mathrm{x} > 0$时，政党 P^* 无法获得任何席位。假设 $\mathrm{x} > 0$，P^* 获得席位的一个必要条件（但非充分条件）是，至少有 $\mathrm{n} + \mathrm{b} - (\mathrm{n} - 1) = \mathrm{b} + 1$ 个政党的得票数小于或等于 $\mathrm{p}^* < \mathrm{d_2K}$。事实上，由于 P^* 在平局决胜顺序中排第一，得票数高于 P^*的政党至多有 $\mathrm{n} - 1$ 个。注意，除 P^* 之外，所有政党起初至少有 $\mathrm{d_2K} > \mathrm{p}^*$ 张选票。因此，我们至少需要转移 $\mathrm{b} + 1$ 张选票来减少它们的得票数。由此可得矛盾，因为当 $\mathrm{x} > 0$时，我们只有 $\mathrm{b} - \mathrm{x} < \mathrm{b}$ 次转移可用。

现在我们转向$x = 0$的情形，并证明P^* 可能获得一个席位。这一次，P^* 的最终选票数为$p^* = d_2K$。重要的是，P^* 在平局决胜顺序中位列第一，且虚拟政党最初恰好有d_2K 张选票。因此，我们不必从虚拟政党那里窃取选民，因为在P^*获得席位之前，它们都不会获得席位。由于恰好有n个元素政党，其选民数都多于d_2K,

要让P^* 获得一个席位，只需从一个元素政党那里窃取选票。根据假设，对每个$i \in [n]$ 都有$b \geq u_i$，我们可以任意选择这样一个政党$P(u_i)$，并窃取其u_i 张选票。剩余的$b - u_i$ 张选票则从任意选定的政党那里窃取。这样，我们就能保证P^* 以成本b通过贿选获得一个席位。此外，$P(u_i)$恰好是那个因不少于u_i 次选票转移而失去席位的政党，这证明了该命题的第 4 点。

命题 3 的证明。我们考察选区的初始状态。不失一般性，固定某个 $u \in U$ 对应的政党 P(u)。回顾一下，P(u) 初始时获得一个席位。由于平局决胜规则以及席位数 n，只有在某个虚设政党或偏好政党通过一次贿赂操作获得至少同样多的选票时，P(u) 才可能失去一个席位。由于 P(u) 初始拥有 $d_2K + u$ 张选票，而每个虚设政党拥有 d_2u 张选票，恰好经过 $u/2$ 次选票转移后，P(u) 就会与某个虚设政党打成平局。事实上，此时两个政党都会拥有 $d_2K + u/2$ 张选票（注意，根据我们对一元装箱实例的假设，u 是偶数）。显然，要让 P(u) 与偏好政党 P^* 打成平局，需要更多次选票转移，因为 P^* 初始时的选票数比每个虚设政党都少 b 张。重要的是，如果存在另一个元素 $u' \in U$，满足 $u' < u/2$，那么 P(u) 无论如何都会获得一个席位。也就是说，在这种情况下，除数法会先考虑 P(u)，再考虑 P(u′)，因为前者的选票更多。实际上，P(u) 会保住其席位。这种情况进一步说明，$u/2$次选票转移只是 P(u) 失去席位所需转移次数的最佳情形下界。我们只针对选区初始状态（即任何贿赂操作发生之前）陈述了这一论证。然而，由于要让任何元素政党失去席位，恰好需要一个失去席位的政党获得一个席位，因此我们可以将同样的论证应用于任何可实现的情形。既然我们已经证明了上文的命题 4，这就完成了该结论的证明。

□

值得注意的是，只需对上述归约稍作调整，就能得到关于最大余数法的所述结果。令 N 表示选票数，S 表示箱子构件中待分配的席位数。回想一下，LRM 在每个选区中独立分两个阶段进行。第一阶段，它为构件中的每个政党 P 恰好分配 $\mathrm{lqu}(\sigma(P), N, S) = \lfloor \sigma(P) \cdot S/N \rfloor$ 个下限配额席位，其中 p 表示 P 获得的选票数。第二阶段，如果仍有 $r > 0$ 个未分配席位，则获得最高 $\mathrm{rem}(\sigma(P), N, S) = \sigma(P) \cdot S/N - \mathrm{lqu}(\sigma(P), N, S)$ 的 r 个政党 P，将各自获得 $\mathrm{rem}(\sigma(P), N, S)$ 个余数席位。

首先，我们将证明第一阶段不会分配任何席位，即 $\lfloor \sigma P \text{ 类} \cdot S/N \rfloor = 0$。由于 $d_2 > 1$，我们将通过取 $d_2 = 1$ 来调整上述归约，并重新定义 δ 为 $\max\{n^2, b\}$。现在，计算更新后的箱子构件的值 N 的下界，

$$\begin{aligned} N &= K - b + K\delta + \sum_{u_i \in U} (K + u_i) \\ &= K - b + K\delta + nK + K \\ &= K(2 + \delta + n) - b \\ &\geq K(n + n^2), \end{aligned}$$

令 $S = n$。于是，我们得到

$$S/N \leq n/(K(n+n^2)).$$

该归约的构造确保，贿选后箱子构件中的政党所能获得的选票最大数量 $\hat{\sigma}(P\text{ 类})$，等于某个元素政党的选票数加上预算 K （这是超预算贿选的结果）。因此，取 b 作为单个元素大小的一个显然上界，可得 $\hat{\sigma}(P\text{ 类}) \leq K + b + K \leq 3K$。由此可知，构件中的每个政党都无法获得任何下限配额席位，因为

$$\begin{aligned} \mathrm{lqu}(\hat{\sigma}(P), N, S) = \lfloor \sigma(P) \cdot S/N \rfloor &\leq \left\lfloor \frac{3Kn}{K(n+n^2)} \right\rfloor \\ &= \left\lfloor \frac{3}{n+1} \right\rfloor = 0 \end{aligned}$$

对于 K 和 $n \geq 3$ 的任意正值，情况均如此。因此，没有政党能获得超过两个席位；这尤其证明了断言 1 的要点 1。没有下限配额席位获分配的另一个结果是，在每个选区中，LRM 总是将席位分配给支持度最高的n个政党（因为 S/N 是一个独立于政党支持度的常数），并按照平局决胜顺序打破平局。虽然我们没有明确说明，但在断言 1 的证明中（要点 1 除外），我们始终考虑了没有政党获得两个席位的情形。在这些情况下，LRM 与除数序列法的表现完全相同，因此断言 1 仍然成立。出于同样的原因，关于该归约正确性的整个论证也适用于 LRM 情形。

□

同样，对于破坏性情形，困难性结果依然成立；我们通过调整前述证明得出这一结论。

定理 9. 设 $\mathscr{R}$ 是除数序列方法或 *LRM*。关于选区数量，$\mathscr{R}$-DMAWB 是 $W[1]$-困难的。

证明。 我们扩展定理 8 证明中的归约，从一元装箱问题* 到 $\mathscr{R}$-DMAWB 得到一个参数化多项式时间归约。沿用相同的记号，固定一个一元装箱问题*实例，其中包含 $U = \{u_1, u2, \ldots, u_n\}$，箱子容量为$b$，箱子数量为$k$ 。

固定一个除数序列方法 $\mathscr{R}$，其序列为 d_1,$d2$，.....。我们构造一个预算为 $K = kb$ 的 $\mathscr{R}$-MAWB 新实例，具体如下。首先，复制定理 8 的证明中的构造。除了复制构造中的偏好政党P^*、 δ 虚拟政党和n 个元素政党之外，我们还引入厌恶政党 P_d。此外，我们在选举中加入两个辅助选区。其中一个只有厌恶政党P_d ，并获得 $2nk$ 个席位；另一个只有偏好政党P^*，并获得 $(2n-1)k$ {v43> 个席位。我们设定平局决胜顺序，使其依次优先偏好政党、虚拟政党、元素政党和厌恶政党。虚拟政党组和元素政党组内的顺序任意，但固定不变。显然，所述实例可以在多项式时间内构造出来。

容易验证（另见定理 8 的证明），所构造的选举结果是：每个元素政党恰好获得 k 个席位，虚拟政党一个席位也没有，偏好政党获得 $(2n-1)k+1$ 个席位（箱子构件不提供席位），厌恶政党获得 $2nk$ 个席位。我们的目标是让某个政党获得比P_{d}更多的席位，而P_{d}原本获得的席位最多。

首先请注意，分配给辅助选区的席位不能重新分配给任何其他政党。与定理 8 的证明中的构造相比，这些是唯一新引入的选区。因此，从现在起，我们将关注其余选区，即箱子构件。根据断言 1 中所述的箱子构件的性质，除P^* 和P_{d} 之外，没有任何政党能拥有超过nk 个席位，因此只有P^* 可能比P_{d}获得更多席位。由于有k 个箱子构件，且P^* 从单个构件中最多只能获得一个席位，P^* 最多可以拥有$(2n-1)k+1+k=2nk+1$ 个席位。重要的是，政党P_{d} 恰好拥有$2nk$ 个席位（而且在其对应的辅助选区中不能失去任何席位）。因此，P^* 必须恰好获得k 个额外席位，每个箱子构件各一个，才能比P^* 拥有更多席位。关于P^* 获胜的这一条件，与定理 8 的证明中的条件等价，因此根据该证明可知，$\mathscr{R}$-DMAWB 关于选区数量是 W[1]-困难的。□

在建设性情形下，按选区数量参数化时，NP完全性和$W[1]$-困难性在简单多数制下同样成立；对此，我们再次通过从一元装箱*归约来证明。

定理 10. 简单多数制-MAWB 是NP完全的，并且关于选区数量是 $W[1]$-困难的。

证明。 显然，该问题属于 NP，因为简单多数制可在多项式时间内计算。为证明所述困难性结果，我们给出从一元装箱*出发的参数化多项式时间归约。因此，我们固定一个一元装箱* 实例，其中包含一个集合 $U=\{u_1,u2,\ldots,u_n\}$、箱子容量b和k个箱子。

在我们构造的 FPTP-MAWB 新实例中，预算 K 为 =kb。我们为每个箱 $\mathrm{i}\in[\mathrm{k}]$分别创建一个偏好政党 P*、一个厌恶政党 $\mathrm{P_d}$和一个箱政党 P_i。为便于说明，接下来描述归约时，我们采用一个“足够大”的常数 X = 4K。对于每个元素 $\mathrm{u}\in\mathrm{U}$，我们构造一个元素选区 D(u)，其中恰有 u 个席位；偏好政党没有选票，厌恶政党获得 (X +u) 张选票，每个箱政党获得 (X −u) 张选票。此外，我们还创建一组虚拟选区，用以确保特定政党获得一定数量的席位。前两个选区分别有 (X − 1) 个和 (X) 个待分配席位，且各有恰好 X 名选民。在第一个选区，所有选民都支持 $\mathrm{P_d}$；在第二个选区，相应的选民都支持 P*。类似地，我们再准备 k 个虚拟选区，每个 $\mathrm{i}\in[\mathrm{k}]$,对应一个；每个选区有 (X −b − 1) 个席位和恰好 X 名选民，所有选民都投票给相应的箱政党 P_i。在新实例中，我们的目标是通过实施成本至多为 K 的获胜竞选，使政党 P* 获得最多席位。我们假设发生内部平票时，P* 优先于所有其他政党，且箱政党优先于厌恶政党。由于原实例中的元素采用一元编码，新实例可以自然地在多项式时间内构造出来。

在证明归约的正确性之前，我们先强调这些构造的重要特征。首先请注意，在各自虚拟选区中获胜的政党具有非常大的胜利幅度$X > 2K$。因此，无论实施何种成本不超过预算K的竞选活动，偏好政党和厌恶政党分别至少获得X席和$(X-1)$席，而元素政党则各自至少获得（来自虚拟选区的）$(X-b-1)$席。其次，X也是P^*在任何成本至多为K的竞选活动后所能获得席位总数的上界，也就是说，该政党不可能再获得任何额外席位。这是因为在每个元素选区中，实施任意竞选活动后，政党P^*最多获得K张选票，而其他政党至少有$(X-b-K) \geq 2K$张选票。最后，从现在起，我们假设P^*始终恰好获得X个席位，因此，在成功竞选过程中，向P^*转移选票或从其转出选票都不会是最优选择。

设$S_1, S2, \ldots\ldots, S_k$是将 U 中的元素装入 k 个箱子的一种装箱方式，并证明了原实例存在解。我们将该装箱方式转换为相应 FPTP-MAWB 实例的解，具体如下。对于每个元素 $u \in U$，以及装箱 $\mathcal{S}$中包含 u 的相应箱子 S_j，我们将 u 张选票从 P_d 转移给选区 D(u) 中与该箱子对应的政党 P_j。由于 U 中元素之和为 K，这种贿选不会超过预算 K。此外，竞选活动结束后，在每个元素选区 D(u) 中，对于 $u \in U$，与包含 u 的箱子 S_j 对应的政党 P_j以及厌恶政党恰好各有 X 张选票。因此，选区 D(u) 的 u 个席位将分配给 P_j。最终，由于每个箱子 S_i（$i \in [k],$）都包含总和为 b 的元素，我们得到 P_d 失去 $kb = K$ 个席位，最终总计获得（$X-1$）个席位。相反，每个箱子政党都会获得 b 个额外席位，从而总计获得 $X-1$ 个席位。由于偏好政党拥有最多的席位，即 X 个席位，我们得到了相应构造实例 FPTP-MAWB 的一个有效解。

现在我们证明反方向：如果存在一次成功竞选，使偏好政党获得的席位多于任何其他政党，那么这次竞选就可以转化为我们所要寻找的、对 U 中元素的装箱。首先回顾一下，贿选前偏好政党获得 X 个席位，厌恶政党获得 $(X+kb-1)$ 个席位，而每个箱政党获得 $(X-b-1)$ 个席位。此外，正如我们先前所证明的，在任何未超出预算的竞选中，偏好政党都必须恰好获得 X 个席位。因此，一次成功竞选会将分配给厌恶政党的席位数减少至至多 $(X-1)$ 个。由于我们已经证明厌恶政党至少必须获得这么多席位，因此厌恶政党最终恰好获得 $(X-1)$ 个席位。也就是说，成功竞选使其失去 kb 个席位。现在我们来考察元素选区：在任何竞选之前，这些选区总共会给厌恶政党 kb 个席位。考虑某个选区 D(u)，其中 $u \in U$。要从该选区的 P_d 手中夺取 u 个席位，至少需要从 P_d转移 u 张选票；此外，这些选票中至少有 u 张必须转移给一个任意选定的箱政党 P_i，$i \in [k]$。如此一来，P_d 剩下至多 X 张选票，而被选中的箱政党则至少获得 $(X-u+u) = X$ 张选票。因此，根据平局决胜规则，P_i 获得该选区的全部 u 个席位，而政党 P_d 失去其全部 u 个席位。将这一论证推广到所有元素选区，我们得出，每次成功竞选都会恰好使用 $kb = K$ 次选票转移，在元素选区中总共从 P_d 手中夺取 kb 个席位。此外，由于我们使用 kb 个预算单位恰好从 P_d手中夺取了 kb 个席位，我们知道，对于每个元素 $u \in U$，成功竞选在相应的

元素选区 $D(u)$ 中。因此，竞选之后，选区 $D(u)$ 的席位会分配给某个政党 P_i，$i \in [k]$。由于竞选前每个箱政党恰好拥有 $(X - b - 1)$ 个席位，并且偏好政党恰好拥有 X 个席位，我们知道，一次成功竞选至多会给每个箱政党带来 b 个额外席位。事实上，每个箱政党都必须恰好获得 b 个额外席位，因为箱政党恰有 k 个，而一次成功竞选总共从 P_{d} 手中夺取了恰好 bk 个席位（且没有夺取其他席位）。由此完成证明：将每个元素 $u \in U$ 分配给代表相应箱政党的箱子；在一次成功竞选行动中，该政党会在元素所在的选区 $D(u)$ 获得席位。 □

对于上述问题，也就是说，对于 $\mathscr{R}$-MAWB（其中 $\mathscr{R}$ 是除数序列方法、LRM 或简单多数制），以及对于 $\mathscr{R}$-DMAWB（其中 $\mathscr{R}$ 是除数序列方法或 LRM），我们既未能找到参数化困难性归约，也未能找到以“政党数量”为参数的固定参数可 tract 算法。一种自然的做法是，例如构造一个整数线性规划，并用固定参数可 tract 算法求解；遗憾的是，使用 Lenstra 算法 [33] 时，变量数量过多，而在包括 n 重 IP 在内的许多其他方法中，约束中又会出现两个较大的系数；参见 Gavenciak 等人 [25]的综述。

与所有其他多选区获胜者问题不同，FPTP-DMAWB 是唯一可在多项式时间内求解的问题。

定理 11. *FPTP*-DMAWB 可在多项式时间内求解。

证明。令 P^* 为在一场有 q 个选区的选举中获胜的指定政党。我们要寻找一场针对 P^* 、预算不超过 K 的成功破坏性竞选。首先，我们猜测一个政党 $\mathrm{P} \in \mathscr{P}_{-P^*}$，并计算分配给政党 P^* 与 P 的席位数之差 d。将该差值加一，并将所得数记为 Δ，即 $\Delta = \mathrm{d} + 1$。显然，Δ 是正整数，否则 P 已经比 P^*获得更多席位。接下来，我们应用命题 1 中的算法，并针对每个选区 $\mathrm{i} \in [\mathrm{q}],$，计算使 P 赢得选区 i 中席位所需的最少选票变更数 b_i；如果 P 已经在 i 中获胜，则令 b_i 为无穷大。利用计算所得的数值，我们构造一个背包问题实例，其中物品与各选区相对应。具体而言，对于某个物品 $\mathrm{i} \in [\mathrm{q}],$，其重量为 b_i，价值则根据 P^* 最初在选区 i 中落败还是获胜，分别设为 κ_i 或 $2\kappa_i$；这对应于在该选区席位转移前后，分配给 P^* 和 P 的席位数之差的变化。然后，我们求解价值为 Δ 且重量上限为 K 的背包问题实例，并记录输出。对政党 P 的所有可能猜测均执行上述过程后，如果至少有一个猜测产生了肯定的背包问题实例，我们就输出“是”。

由于猜测的数量是多项式级的，且所构造的背包实例的权重相对于原实例的输入规模是多项式有界的，因此我们的算法能在多项式时间内运行。该算法的正确性可直接从我们为所构造的背包实例计算物品权重和大小的方式中得出。

□

表1：我们实验所选全部数据集的选区数量、席位、阈值和总票数概览。

	奥地利 2019	以色列 2022	荷兰 2023	波兰 2023	葡萄牙 2024	阿根廷 2021
选区数量	1	1	1	41	22	24
席位数量	183	120	150	460	230	127
阈值	4%	3.25%	$^1/_{150}$	5%	0%	3%
（有效）选票数量	4,777,246	4,764,742	10,432,526	21,596,674	6,194,290	23,602,493

3.3 实验

正如我们在上一节所展示的，对于单个或多个选区中的席位分配贿选，无论是建设性情形还是破坏性情形，计算成功的乃至最优竞选活动在计算上都是可行的。这表明，竞选经理执行此类攻击相对简单（至少从计算角度来看是如此）。然而，策略性竞选在现实场景中的有效性 仍有待研究。因此，在本节中，我们通过实验研究此类竞选活动的有效性，重点考察以下三个问题：

1. 与简单的贿赂启发式策略相比，我们的最优算法有多有效？衡量标准是：要使指定政党的席位数增加或减少一个，所需的额外预算是多少？

2. 选举门槛如何影响战略竞选活动的有效性？衡量标准是：在最优竞选活动中，指定政党可以获得或失去多少席位？

3. 选区数量如何影响改变选举结果的难度？衡量标准是所需的最低预算。

我们使用三个来自单选区选举的数据集，以及三个来自多选区选举的数据集。[12] 对于单选区选举，我们采用2019年奥地利议会选举、2022年以色列议会选举和2023年荷兰议会选举的数据。对于多选区选举，我们采用2023年波兰议会选举、2021阿根廷议会选举和2024年葡萄牙议会选举的数据。选择这些国家，是因为它们都采用德洪特法。表1概述了这些国家各自的选区数量、议会席位数和阈值。[13]

[12]实验所用的代码和数据可在 https://github.com/bredereck/Strategic-Campaigns-in-Apportionment-Elections. 找到。[13]请注意，阿根廷众议院 每两年改选一半席位，也就是说，总共有257个席位。

3.3.1 最优竞选活动的有效性

在初步实验中，我们评估最优贿赂策略相较于较简单启发式方法在建设性和破坏性情形下的有效性。具体而言，我们考察启发式方法为实现与最优算法相同的目标所需的额外预算：即将某个政党的席位数增加（建设性情形）或减少（破坏性情形）一个席位。我们的精确算法有时会产生相当复杂的策略性竞选活动（主要体现在选民转移方面）。简单的启发式方法所产生的竞选活动复杂度最低，因此易于实施。

我们先来看建设性情形，其中我们考虑针对政党 P 类$_i$的以下贿赂策略：

1. 最优贿赂：我们以最优方式将其他政党的选票转移给 P 类$_i$（与第 3.1 节和第 3.2 节中的算法相同，但以 P 类$_i$ 替代 P 类$_1$）。2. 均衡贿赂：我们按照其他政党的得票数比例，将选票转移给 P 类$_i$（这是对均匀随机贿赂选票的一种去随机化类比）。3. 最弱竞争对手：我们将最弱政党的选票转移给 P 类$_i$（如果该竞争政党的选票不足，我们就从第二弱的竞争对手那里转移选票，依此类推）。[14] 4. 最强竞争对手：我们将最强政党的选票转移给 P 类$_i$（如果该竞争政党的选票不足，我们就从第二强的竞争对手那里转移选票，依此类推）。[15]

请注意，最优贿赂对信息的要求非常高（需要精确票数），而其他形式所需的信息则少得多。因此，也可以将后三种策略视为基于现实信息假设的贿赂策略。

在每个数据集（即每个国家）中，我们假设所有政党和所有选区采用相同的阈值（各国实际阈值的概览见表1）。政党若要在某个选区赢得席位，必须在该选区达到阈值。对于每个国家，我们运用上述贿赂策略1至4，计算指定政党要赢得一个额外席位所需改变的选票数。请注意，在某些情况下，额外席位的数量会大于一个，因为某个政党已经达到阈值。我们按如下方式计算不同策略的效率：针对每种算法，我们计算实现该目标所需的预算，即指定政党采用相应贿赂策略以赢得一个额外席位所需改变的选票数；再将其除以最优算法对应的数值，即指定政党赢得一个额外席位所需改变的最少选票数。

建设性情形的结果见表2a（针对单选区选举）和表2b（针对多选区选举）。直观来说，我们想知道，与最优策略相比，采用简单的启发式贿选策略时，还需要改变多少选票。

[14]我们所说的最弱政党，是指原本获得最少票数的政党。[15]我们所说的最强政党，是指原本获得最多票数的政党。

例如，表2a和表2b中的具体数字“2”表示，为了让指定政党再获得一个席位，启发式算法需要改变的选票数是最优算法的两倍。请注意，在某些情况下，选票数翻倍甚至意味着简单策略需要多贿赂数千张选票，尽管这只关系到改变一个席位。最后，在表格的“平均”列中，我们将所有政党的平均结果视为指定政党的数据；[16] 而在下一列中，我们选择最强政党作为指定政党，在最右列中则选择最弱政党。

表2：为获得一个额外席位而采取的不同贿赂策略的有效性。

（a）单选区国家。

		平均	最强政党	最弱政党
Austria	最优贿赂	1	1	1
	均衡贿赂	1.01406	2.07465	1
	最弱竞争对手	1.0256	3.3224	1
	最强竞争对手	1.01547	1	1
Israel	最优贿赂	1	1	1
	均衡贿赂	1.01520	3.84078	1
	最弱竞争对手	1.01984	5.02418	1
	最强竞争对手	1.01659	1	1
Netherlands	最优贿赂	1	1	1
	均衡贿赂	1.17935	4.99234	1
	最弱竞争对手	1.22679	6.552335	1
	最强竞争对手	1.22383	6.46223	1

（b）多选区国家。

		平均	最强政党	最弱政党
Poland	最优贿赂	1	1	1
	均衡贿赂	1.08521	1.65625	1.07160
	最弱竞争对手	1.17646	3.18750	1.16582
	最强竞争对手	1.10409	1	1.07399
Portugal	最优贿赂	1	1	1
	均衡贿赂	1.14033	1.447489	1.12533
	最弱竞争对手	1.34459	2.32420	1.31880
	最强竞争对手	1.24686	1	1.24862
Argentina	最优贿赂	1	1	1
	均衡贿赂	1.26643	1.07874	1.46548
	最弱竞争对手	1.46549	1.50394	2
	最强竞争对手	1.21806	1	2

可以看出，在大多数情况下，最优贿赂比简单贿选策略有效得多，即只需较小的预算就能多获得一个席位。通常，当最强政党被选为指定政党时，有效性差异最大。例如，在2023年荷兰议会选举中，对于自由党 而言，我们的算法所采用的最优贿赂策略所需选票数，仅为其他简单贿选策略的 1/4.99 至 1/6.55。不过，也有一些情况，例如在单选区国家中选择最弱政党作为指定政党时，所有算法都必须转移相同的（最少）选票数，才能使指定政党赢得一个席位。这是因为各政党获得的选票数低于阈值，只要使其达到阈值，就足以赢得一个席位（有时甚至更直接），而选票来自哪些政党并无太大区别。在简单贿选策略中，按比例从其他政党获取选票（即采用均衡贿赂）似乎是最有希望的方法。

对于破坏性情形，我们考虑与建设性情形类似的贿赂策略：

[16]具体而言，对于每个国家，我们先计算特定算法所获得的所有预算的算术平均数，再将其除以最优算法结果的算术平均数，以此得出所有政党结果的平均值。

1. 最优贿赂：我们以最优方式将选票从 P 类$_i$ 转移给其他政党（如第 3.1 节和第 3.2 节中的算法，但用P 类$_i$ 代替P 类$_l$）。2. 均衡贿赂：我们按照其他政党的得票数比例，将选票从 P 类$_i$ 转移给其他政党（这是对均匀随机贿赂选票的去随机化类比）。3. 最弱竞争对手：我们将选票从 P 类$_i$ 转移给最弱政党。4. 最强竞争对手：我们将选票从 P 类$_i$ 转移给最强政党。

我们以与建设性情形类似的方式进行了实验，不同之处在于，现在我们的目标是计算必须改变多少选票，才能使指定政党失去 一个席位。请注意，在破坏性情形中，我们不考虑没有获得任何席位的政党，也就是说，不会将这些政党选为指定政党。结果汇总于表 3a（单一选区的选举）和表 3b（多个选区的选举）中。

破坏性情形的结果与建设性情形类似。同样，在几乎所有情况下，最优贿赂都比简单贿选策略更有效（有时差距显著）；有效性差距最大时，指定政党是最强政党。例如，在以色列2022年选举中，利库德集团是最强政党，我们的算法所需的贿赂成本远低于其他策略：在此情况下，简单贿赂策略所需的选票最多可达我们的最优算法的 5.59 倍。同样，也存在预算不受所用算法影响的情况（但比建设性情形少），例如，某政党只有一个席位，只需失去足够多的选票，使其跌破阈值。在荷兰选举中就可以观察到这种情况：拥有至少一个席位的最弱政党——JA21——被选为指定政党时，所有算法都必须转移相同（最少数量）的选票，才能使指定政党失去一个席位。最后，在选择简单贿选策略时，视具体情况而定，采用均衡贿赂或从最强竞争对手处转移选票似乎最有希望。不过需要指出的是，在若干情况下，各种简单策略的数值彼此相近。

3.3.2 阈值的影响

在第二个实验中，我们希望了解选举门槛对策略性竞选有效性的影响。此处，我们将有效性定义为：在最优建设性竞选活动中，我们能够为指定政党 P^* 赢得的席位数；或者在最优破坏性竞选活动中，我们能够从指定政党 P^* 手中夺走的席位数。

我们对三个单选区数据集开展实验，具体如下。在这三场选举中，我们将预算 K 设为总得票数的 0.25%。这占选民总数的比例相对较小，我们认为竞选经理有可能影响这么多选民。为了展示阈值对竞选活动有效性的影响，我们在实验中逐步提高阈值。作为指定政党P^*

Tabl表 3：从一个政党手中夺取一个席位的不同贿赂策略的有效性 y.

(a) 单选区国家。

		平均	最强政党	最弱政党拥有席位的
Austria	最优贿赂	1	1	1
	均衡贿赂	1.26931	1.69816	1.08033
	最弱竞争对手	1.66797	2.69417	1.17211
	最强竞争对手	1.22228	1.43293	1.08417
Israel	最优贿赂	1	1	1
	均衡贿赂	1.93924	4.30920	1.56716
	最弱竞争对手	2.16676	5.59157	1.625
	最强竞争对手	2.02504	5.44377	1.625
Netherlands	最优贿赂	1	1	1
	均衡贿赂	1.1388	1.63824	1
	最弱竞争对手	1.18093	2.13672	1
	最强竞争对手	1.17158	2.13672	1

(b) 多选区国家。

		平均	最强政党	最弱政党带席位
Poland	最优贿赂	1	1	1
	均衡贿赂	1.17703	2.02857	1.12121
	最弱竞争对手	1.40546	2.99286	1.21212
	最强竞争对手	1.17703	2.99286	1
Portugal	最优贿赂	1	1	1
	均衡贿赂	1.21057	1.25109	1.0703
	最弱竞争对手	1.40685	1.5	1.09050
	最强竞争对手	1.38736	1	1.09050
Argentina	最优贿赂	1	1	1
	均衡贿赂	1.1331	1.46196	1.08754
	最弱竞争对手	1.27494	2	1.18748
	最强竞争对手	1.19092	2	1.12473

我们总是选择在选举中获得选民最高支持度的政党，因为它达到所有测试的阈值，因此始终在议会中占有席位。最后，我们在实验中使用德洪特法作为除数序列法的代表，因为它是席位分配选举中应用最广泛的方法之一。

图1展示了在三个现实世界选举中开展的建设性（上排）和破坏性（下排）竞选活动的有效性。人们可能会预期某种比例性，例如，0.25%的选民控制着大约0.25%的席位。在这里，对于三个国家（奥地利、以色列和荷兰）而言，议会中0.25%的席位对应于零席位至一席位之间的数量，而对于阈值的许多取值，我们确实观察到了这一点。然而，在某些峰值处，仅凭0.25%的选票，就可以使指定政党P^*（即最强政党）在建设性情形下有时额外获得全部席位中的3.3%（荷兰）、4.4%（奥地利），甚至12.5%（以色列）。这远远超过了我们有限预算所能期待的效果。请注意，这些峰值主要出现在某个政党恰好位于阈值之上的位置。例如，在奥地利，我们在1.87%和8.10%的阈值处看到峰值，而这正是JETZT – Pilz名单 和*NEOS* – 新奥地利略高于阈值的数值。这表明，在这些阈值下，竞选活动的重点是将某个政党压至阈值以下，从而释放其席位。

对于破坏性情形，我们可以看到与建设性竞选活动中类似的峰值。不过，这一次峰值主要出现在某个政党恰好低于阈值的情形下。也就是说，竞选活动的重点是将某个政党推过阈值，从P^*手中夺取席位。例如，在奥地利，略高于0%、1.87%和8.10%的阈值处都出现了峰值；这与有七个政党的全部（有效）选票占比低于1%相吻合（奥地利共产党、*WANDL*、*BIER*、*GILT*、奥地利未来联盟、*SLP*和*CPÖ*），而*JETZT – Pilz*名单 和*NEOS* – 新奥地利 的阈值则分别为1.87%和8.10%。[17]

[17]请注意，在建设性情形中，0% 阈值附近没有峰值，因为选票占比低于 1% 的政党无法为指定政党腾出很多席位，而它们总共只分得一个席位（具体而言，奥地利共产党 有一个席位），其他所有六个支持度低于 1% 的政党则没有席位。

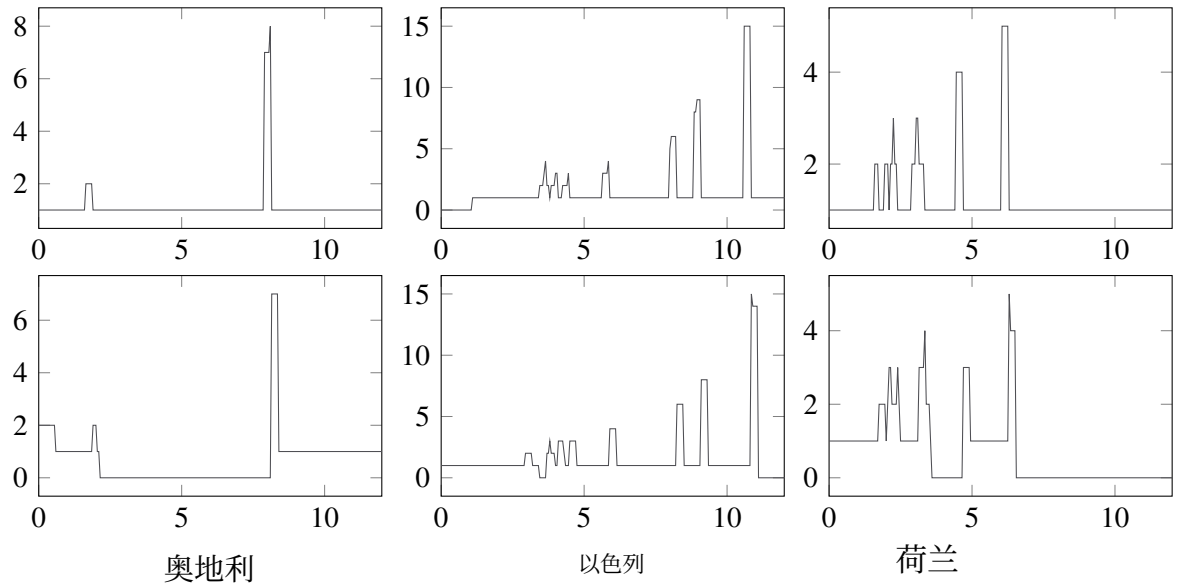


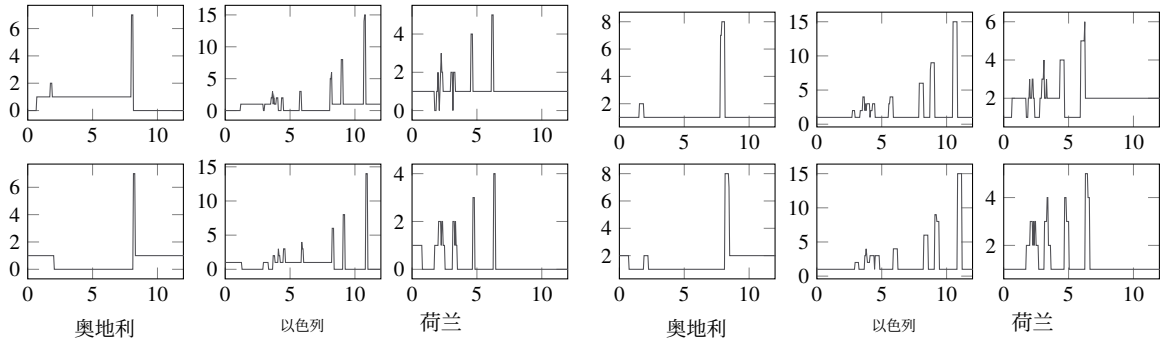
图1：x轴显示了以选民人数n的百分比表示的各种阈值。y轴显示了在给定预算$K = 0.0025 \cdot n$的情况下，最强政党通过德洪特法-AB（上排）和德洪特法-DAB（下排）所能实现的额外席位数（或相应地，阻止获得的席位数）的最大值。

请注意，我们还针对贿赂预算K 的其他取值进行了实验，并观察到以下现象：对于较小的预算，在某个政党恰好高于阈值（在破坏性情形下，则是恰好低于阈值）的位置，我们会看到更窄（有时也更低）的峰值；而对于较大的预算，峰值会变宽（有时也会变高）。图2a示例性地展示了0.15%的结果，图2b展示了0.35%的结果。最后，图3a和图3b还说明了当我们选择第二强或第三强政党作为指定政党P^*时所产生的影响。结果同样相似。

3.3.3 选区数量的影响

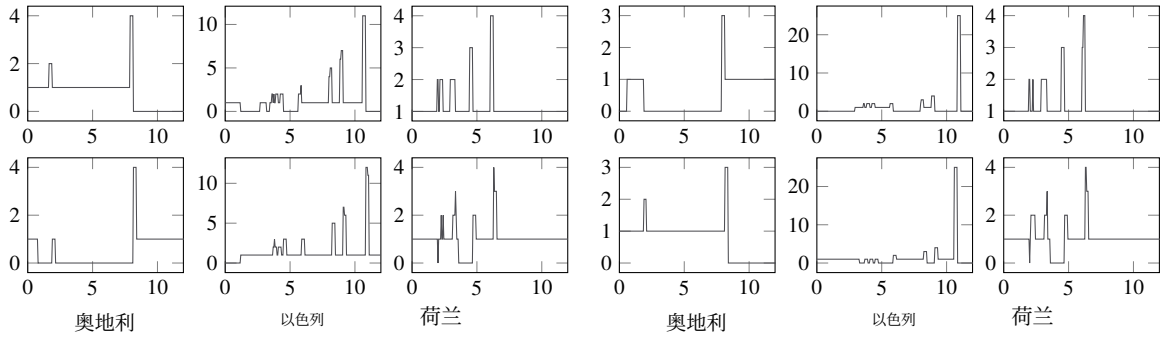
在第三个实验中，我们研究选区数量对改变选举结果所需预算的影响。这里，我们考察三个多选区选举，即波兰、葡萄牙和阿根廷数据集（各选区的数量、席位数、阈值和选票数概览请再次参见表1）。对于每个数据集，我们为一个政党计算最优贿赂[18]，使其在建设性情形下获得席位多数（$50\% + 1$）（请注意，在所有选举中，原本都没有政党赢得席位多数），并在破坏性情形下失去其原有一半的席位。为了研究选区数量的影响，我们从该国的原始划分开始，然后通过合并选区来减少选区数量。我们按顺序进行，每次随机选取两个选区合并，始终合并

[18]对于波兰选举和葡萄牙选举，我们选择席位数最多的政党；对于阿根廷选举，我们选择第二强政党，因为最强政党已经相当接近多数。



（a）预算为 $K = 0.0015 \cdot n$ 的实验。 (b) 预算为 $K = 0.0035 \cdot n$ 的实验。

图 2：x 轴显示了以选民人数 n 的百分比表示的各种阈值。y 轴显示了在第一行中使用 D’Hondt-AB、第二行中使用 D’Hondt-DAB 时，最强政党最多可获得的额外席位数（可阻止的席位数）。



（a）与第二强政党进行实验。 （b）与第三强党进行实验。

图3：x轴显示了以选民人数n百分比表示的各种阈值。y轴显示了在给定预算$K = 0.0025 \cdot n$的情况下，D’Hondt-AB（上排）和 D’Hondt-DAB（下排）可实现的额外席位（阻止的席位）最大数量。

两个随机选取的选区，直到剩下指定数量的选区。对于波兰选举，我们合并10、20和30个选区；对于葡萄牙和阿根廷选举（它们的选区数量相近，约为波兰选举选区数量的一半），我们合并5、10和15个选区。给定选区数量下的结果按如下方式计算：

1. 我们取三次试验中被贿赂选民数量的平均值，每次试验采用不同的选区划分（按上述方法计算）；2. 我们计算每增加（或减少）一个席位所需选票转移数量的平均值；3. 将该数值除以原始选举中每增加（或减少）一个席位所需的选票转移数量。

结果汇总于表4。在所有情况下，选区数量越少，政党平均而言赢得（或失去）一个席位所需的最低预算就越高；也就是说，一场选举中的选区越多，从成本角度看，成功贿选就越容易。例如，在波兰选举中（见表4a），如果选区数量减少约一半（至21个选区），赢得一个席位所需的预算将是原始选区数量下的两倍多（2，371倍）。在破坏性情形下，则必须改变多于四倍的选票（4，213倍），才能使政党失去一个席位。请注意，即便合并选区的影响较小，也就是说增幅较缓（以剩余选区的百分比来看），例如葡萄牙选举（见表4b），最低预算之间的差距仍然至少相当于数千名需要贿赂的选民。

表4：在不同选区数量的选举中，为使当选政党多获得一个席位或失去一个席位而进行最优贿赂所需的平均最低预算；结果以比率 $\mathrm{MpS}/_{\mathrm{MpS\text{-}o}}$ 表示，其中 MpS 是当选政党获得（或失去）该席位所需最低预算的平均值，而 MpS-o 是原始选举（即未合并选区）的 MpS 值。

（a）波兰

选区数量	41	31	21	11
建设性	1	1.622	2.371	3.607
破坏性	1	1.968	4.213	5.865

（b）葡萄牙

选区数量	22	17	12	7
建设性	1	1.168	1.487	1.843
破坏性	1	1.293	1.621	2.260

(c) 阿根廷

选区数量	24	19	14	9
建设性	1	1.276	1.961	2.789
破坏性	1	1.607	2.805	3.337

4 第二次机会模式

从前面的章节中我们知道，如果最优竞选活动很容易计算，问题就相当严重，因为这会让竞选经理轻而易举地对选举结果施加巨大影响。因此，如果能对通常的席位分配设置进行修改，使最优竞选活动的计算变得棘手，将会大有裨益。正如引言中所述，设有阈值的席位分配选举还有另一个普遍问题：支持未达到阈值政党的选民会被完全忽略，导致议会的代表性降低。

现在，我们介绍席位分配选举中的第二次机会模式投票，它有助于解决上述两个问题。与首选模式不同，在第二次机会模式下，支持未达到阈值政党的选民可以再次使用自己的选票。也就是说，我们首先确定至少获得 τ 个首选的政党 $\mathscr{P}_\tau$，即超过阈值的政党。现在，每位选民都被视为支持其最偏好的 $\mathscr{P}_\tau$中的政党；换言之，从选票中移除不属于 $\mathscr{P}_\tau$ 的政党后，投给被移除政党的选票将转给 $\mathscr{P}_\tau$中最受偏好的政党。第二次机会投票过程有些类似于两轮制相对多数制或单记可转移投票制等迭代投票规则。不过，它与这些规则也有所不同，因为后者是投票规则，而非用于计算支持度分配的规则。

例5 （第二次机会模式下的支持度分配）。 回想一下选举 n

$$E = (\{P_1, \ldots, P_5\}, \{v_1, \ldots, v_{2052}\})$$

例 *1* 中的选票

$$\begin{aligned}
v_1, \ldots, v_{604}: &\quad P_1 \succ P_2 \succ P_3 \succ P_5 \succ P_4 \\
v_{605}, \ldots, v_{819}: &\quad P_2 \succ P_1 \succ P_5 \succ P_4 \succ P_3 \\
v_{820}, \ldots, v_{1174}: &\quad P_3 \succ P_2 \succ P_1 \succ P_4 \succ P_5 \\
v_{1175}, \ldots, v_{1474}: &\quad P_1 \succ P_5 \succ P_3 \succ P_2 \succ P_4 \\
v_{1475}, \ldots, v_{1652}: &\quad P_4 \succ P_2 \succ P_3 \succ P_1 \succ P_5 \\
v_{1653}, \ldots, v_{1800}: &\quad P_2 \succ P_4 \succ P_3 \succ P_1 \succ P_5 \\
v_{1801}, \ldots, v_{2000}: &\quad P_1 \succ P_3 \succ P_4 \succ P_5 \succ P_2 \\
v_{2001}, \ldots, v_{2052}: &\quad P_5 \succ P_2 \succ P_3 \succ P_4 \succ P_1
\end{aligned}$$

以及席位分配问题 I =（σ，100，6）。正如我们从例 *1* 中所知，$P_1, P_2, P_3,$ 和 P_4 均超过阈值，而政党 P_5 没有，因此 $\mathscr{P}_\tau = \mathscr{P}_{100} = \{P_1, P_2, P_3, P_4\}$。在首选模式下，相应的支持度分配如下：

$$\begin{aligned}
\sigma(P_1) &= 1104, \\
\sigma(P_2) &= 363, \\
\sigma(P_3) &= 355, \\
\sigma(P_4) &= 178, \\
\sigma(P_5) &= 0.
\end{aligned}$$

然而，在第二次机会模式下，我们不会忽略所有将 P_5 列为首选的选票（此处为选票 v_{2001} 至 v_{2052}），而是从这些选票中移除 P_5，然后检查移除 P_5后，这些选票中最受偏好的阈值以上政党是 P_1, P_2, P_3, 还是 P_4。在此例中，在所有选票 v_{2001}、.....、v_{2052}, 中，最受偏好的都是 P_2，因此我们将这52 张选票计入 P 类的支持度

，并得出最终的支持度分配

$$\sigma(P_1) = 1104,$$
$$\sigma(P_2) = 415,$$
$$\sigma(P_3) = 355,$$
$$\sigma(P_4) = 178,$$
$$\sigma(P_5) = 0.$$

请注意，澳大利亚已经在使用类似的投票制度，例如用于众议院 和参议院。[19] 在这些选举中，选民会按照偏好从高到低排列候选人或政党，并根据给定的排序，将投给被淘汰选项的选票转移，直至计完所有选票。

4.1 单选区情形

在第 3.1 节中，我们证明了在经典席位分配情境下，对于简单多数制、每一种除数序列方法以及 LRM， $\mathscr{R}$-AB 和 $\mathscr{R}$-DAB 在单选区情形下都可以高效求解。与此相反，我们将证明，在第二次机会模式下，这些问题是 NP困难的。为此，我们从 NP完全问题击中集 [30], 进行归约，该问题已在第2.1节中定义。

以下我们不再只关注特定的席位分配方法，而是将结果推广到一整类席位分配方法。

定义 1 （多数一致性）。如果在 $\mathscr{P}$ 中，支持度低于 A 的政党所获席位均不多于 A，则称一种席位分配方法为多数一致的，其中 $A \in \mathscr{P}$ 是支持度最高的政党。

毫无疑问，这是每一种合理的席位分配方法都应满足的标准。请注意，简单多数制、所有除数序列法以及 LRM 显然都是多数一致的。

我们现在将证明，席位分配投票的第二次机会模式使得计算最优策略性竞选在计算上难以处理，因此可以阻止人们尝试开展此类竞选活动。

定理12。 对于每一种多数一致分配方法， $\mathscr{R}$ $\mathscr{R}$-AB 和 $\mathscr{R}$-DAB在第二次机会模式下都是NP-困难的。如果 $\mathscr{R}$ 可在多项式时间内计算，则它们是NP-完全的。

证明。 只要 $\mathscr{R}$ 可在多项式时间内计算，这两个问题显然都属于 NP。我们通过从 HITTING-SET 归约，证明 $\mathscr{R}$-AB 是 NP 困难的。设 $(U, S, K) = (\{u1, ..., u_p\}, \{S1, ..., S_q\}, K)$ 是 HITTING-SET 的一个实例，其中 $q \geq 4$。在

[19] https://www.aec.gov.au/learn/preferential-voting.htm

多项式时间内，我们构造一个 $\mathscr{R}$-AB 实例，其中政党为 $\mathscr{P} = \{P^*, P\} \cup U$，阈值为 $q+1$，$\ell = 1$ 目标席位、 $\kappa = 1$ 可用席位，以及选票

$$\begin{aligned}
\mathscr{V} = (4q+2 \text{ votes} \quad & P^* \succ \cdots, \\
4q+K+2 \text{ votes} \quad & P \succ \cdots, && (1)\\
\text{for each } j \in [q], 2 \text{ votes} \quad & S_j \succ P \succ \cdots, && (2)\\
\text{for each } i \in [p], q-\gamma_i \text{ votes} \quad & u_i \succ P^* \succ \cdots, && (3)\\
\text{for each } i \in [p], q-\gamma_i \text{ votes} \quad & u_i \succ P \succ \cdots), && (4)
\end{aligned}$$

其中 $S_j \succ P$ 表示，S_j 中的每个元素都优先于 P，但我们不关心 S_j 中各元素的确切顺序。此外， $2\gamma_i$ 表示在组（2）的选票中，u_i 处于第一位的选票数。因此可以保证

- 每个 $u_i \in U$ 恰好获得 $2\gamma_i + (q-\gamma_i) + (q-\gamma_i) = 2q < \tau$ 个首选（低于阈值，因此无法获得席位），而
- P^* 有 $4q+2 \geq \tau$ 个首选，并获得 $(p+4)q + 2 - \sum_{p\,i=1} \gamma_i$ 名选民的支持度，而
- P 有 $4q+K+2 \geq \tau$ 个首选，获得 $(p+6)q + K + 2 - \sum_{p\,i=1} \gamma_i$ 名选民的支持度。

请注意，组（2）和组（4）的选民使用第二次机会投票给 P，而组（3）的选民则使用第二次机会投票给 P^*。因此，P 目前恰好比 P^* 多获得 $2q+K$ 张选票，从而赢得该席位。

现在我们证明，当且仅当存在一个大小至多为 K 的击中集时，我们才能通过贿赂至多 K 名选民，使指定政党 P^* 赢得该席位。

（⇐）假设存在一个大小为 K 的击中集 $\mathrm{U}' \subseteq \mathrm{U}$（若 $|\mathrm{U}'| < \mathrm{K}$，则可从 U 中任意添加元素，将其补足至恰好 K 个元素）。对于每个 $\mathrm{u}_i \in \mathrm{U}'$，我们贿赂组（1）中的一名选民，让其将 u_i 排在第一位。这样一来，这些 u_i 现在各自获得了达到阈值所需的 $2\mathrm{q} + 1 = \tau$ 个首选，即它们参与后续的席位分配过程。每个 u_i 最多可获得 $4\mathrm{q} + 1$ 的支持度。由于任何贿赂都不会影响 P^* 的支持度，因此在任何多数一致的席位分配方法 $\mathscr{R}$ 下，都没有 u_i 能击败 P^* 赢得该席位。组（3）和组（4）不会改变 P^* 与 P 之间的支持度差，因此可以忽略。然而，由于 U' 中的所有元素现在都达到阈值，且 U' 是一个击中集，组（2）中的全部 2q 名选民现在都会投票给 U' 中的某个政党，而不再投给 P，从而使 P^* 与 P 之间的差距缩小 2q。此外，我们贿赂了组（1）中的 K 名选民，使他们不再投票给 P，这又使 P^* 与 P 之间的差距缩小 K 票。因此，P^* 和 P 现在获得了相同的支持度，各有 $4\mathrm{q} + 2$ 名选民支持。由于我们假设平局决胜规则倾向于 P^*，政党 P^* 赢得目标席位。

(⇒) 假设最小击中集的大小为 $K' > K$。也就是说，仅凭 K 个 U 中的元素，我们最多可以击中 $q-1$ 个来自 S 的集合。因此，通过贿赂组（1）中的 K 名选民，使其投票给某个 $u_i \in U$，而不是投给 P，我们最多只能阻止组（2）中多达 $2(q-1)$ 名选民使用第二次机会投票给 P。因此，我们至多只能将 P^* 与 P 之间的支持度差减少 $2(q-1)+K$，这不足以让 P^* 赢得席位。现在考虑不将全部预算 K 用于这一策略，即不把它全部用于贿赂组（1）的选民，而只使用

$K'' < K$。注意，仅将 U 中的 K'' 个政党提升到阈值之上，我们最多只能击中 S 中的 $2(q - 1 - (K - K''))$ 个集合。因此，使用这一策略，P^* 与 P 之间的支持度差至多减少 $2(q - 1 - (K - K'')) + K''$。然而，现在我们还剩下 $K - K''$的预算，可以贿赂选民（例如组（2）中的选民），使其将 P^* 作为首选，而无需再将任何其他政党 u_i 提升到阈值之上。很容易看出，使用这一策略，P^* 与 P 之间的支持度差至多减少 $2(K - K'')$，因为在最佳情况下，每次贿赂操作只能使 P^* 增加一名支持者，同时使 P 减少一名支持者。因此，采用这种混合策略，我们无法将 P^* 与 P 之间的支持度差减少超过

$$2(q-1-(K-K''))+K''+2(K-K'') = 2(q-1)+K''.$$

对于每个 $K'' \le K$，都有 $2(q-1)+K'' < 2q+K$。因此，如果不存在大小至多为 K 的击中集，我们就无法让指定政党 P^* 战胜 P。

破坏性变体的证明通过交换P^* 和 P的角色并设置$\ell = 0$来实现。

□

在 $\kappa = 1$ 的情形下（即我们只有一个席位可分配），询问指定政党P^* 是否能够获得至少 $\ell = 1$ 个席位（在破坏性情形下分别为 $\ell = 0$），等价于询问P^* 是否能够成为最强政党（分别地，是否不能成为最强政党）。因此，同样的归约可以用于获胜者问题。

定理13。 对于每种多数一致的席位分配方法 $\mathscr{R}$， $\mathscr{R}$-AWB 和 $\mathscr{R}$-DAWB 在第二次机会模式下都是NP-困难的。如果 $\mathscr{R}$ 是多项式时间可计算的，则它们是NP-完全的。

4.2 多选区情形

由于在第二次机会模式下，对于所有多数一致席位分配方法，所有单选区问题都已是 NP 困难的（见定理 12 和 13），因此，只需将单选区问题简单归约为相应的多选区问题，即可证明多选区情形下的 NP 困难性。具体而言，对于建设性情形（即从 $\mathscr{R}$-AB 归约到 $\mathscr{R}$-MAB）和破坏性情形（即从 $\mathscr{R}$-DAB 归约到 $\mathscr{R}$-DMAB）的归约，我们使用与单选区问题相同的输入，并将选区数量设为 $q = 1$。对于获胜者问题，在建设性情形下，我们从 $\mathscr{R}$-AWB 归约到 $\mathscr{R}$-MAWB；在破坏性情形下，我们从 $\mathscr{R}$-DAWB 归约到 $\mathscr{R}$-DMAWB，采用与上述相同的简单构造，只是这里没有给定期望席位数。

命题 7。 对于每一种多数一致的席位分配方法$\mathscr{R}$、$\mathscr{R}$-多臂老虎机、$\mathscr{R}$-DMAB、$\mathscr{R}$-MAWB以及 $\mathscr{R}$-DMAWB，在第二次机会模式下都是 NP-困难的。如果 $\mathscr{R}$ 可在多项式时间内计算，则它们是 NP-完全的。

表 5：我们关于席位分配选举中贿选复杂性的结果概览。最右列中的 NP 困难性结果，是针对可在多项式时间内计算的多数一致的席位分配方法得出的 NP 完全性结果。所有 W[1]-困难性结果均以选区数量为参数。

		首选模式			第二次机会模式
		FPTP	除数序列	LRM	多数一致的
AB	构造	P 类（命题1）	P 类（定理1）	P 类（定理3）	NP困难（定理12）
	析取	P 类（命题2）	P 类（定理2）	P 类（定理4）	NP困难（定理12）
AWB	构造	P 类（命题1）	P 类（定理5）	P 类（定理5）	NP困难（定理13）
	析取	P 类（命题2）	P 类（命题3）	P 类（命题 4）	NP困难（定理13）
MAB	构造	P 类（命题 5）	P 类（命题 5）	P 类（命题 5）	NP困难（命题 7）
	析取	P 类（命题 6）	P 类（命题 6）	P 类（命题 6）	NP困难（命题 7）
MAWB	构造	NP-完全（定理 10）	NP-完全（定理 6）	NP-完全（定理 6）	NP困难（命题 7）
		W[1]-困难（定理 10）	W[1]-困难（定理 8）	W[1]-困难（定理 8）	
		P 类（定理 11）			
	析取		NP-完全（定理 7）	NP-完全（定理 7）	NP困难（命题 7）
			W[1]-困难（定理 9）	W[1]-困难（定理 9）	

5 结论与未来工作

我们研究了单选区和多选区情形下选举的策略性竞选管理（建模为贿赂问题），并引入了二次机会投票模式：在该模式下，投给低于阈值的政党的选民可以获得第二次使用其选票的机会。表 5 概述了我们关于建设性贿赂问题的理论复杂性结果。我们还开展了广泛的实验分析，研究最优竞选活动的有效性，特别是将其与使用启发式贿赂策略进行比较，并考察阈值和选区数量的影响。我们发现，在大多数情况下，以最优方式转移选票比简单的贿赂策略更为（显著地）有效；我们的最优竞选活动能够利用阈值并从中显著获益；此外，选区越少，为获得（相应地，夺走）一定数量的席位，所需的选票转移就越多。

我们的算法需要关于得票数的精确信息，即需要获取在选举之前难以获得的信息。我们认为，未来研究基于不完美信息（例如民意调查数据）来研究席位分配选举中的贿选和策略性竞选活动，是一项有趣的任务。在实验

第3.3.1节的设置中对简单贿赂方案进行比较，可以视为朝这一方向迈出的第一步。

此外，我们关注的是流行的席位分配方案，但还有其他此类方法，例如 Balinski 和 Young 的配额法 [1, 2]，或弗雷格席位分配法。后者记载于弗雷格一篇久已失传、后来于1998年在德国耶拿重新发现的著作中（参见 Harrenstein等人 [27]的研究）。

此外，我们计划研究其他类型的策略性竞选（例如克隆政党；其理念类似于单获胜者选举和多获胜者选举中的克隆 [15, 34, 40]）。我们已经研究过选举控制问题（参见 Faliszewski 和 Rothe [17]以及 Baumeister 和 Rothe [6]撰写的书籍章节），特别是通过增加或删除政党或选民来进行的建设性控制和破坏性控制。事实证明，首选模式和第二次机会模式都能抵抗全部四种政党控制问题。关于证明，只需调整 Bartholdi 等人 [5] 和 Hemaspaandra等人 [28] 的证明即可；他们证明了多数票制能够抵抗相应的选举控制问题。对于首选模式下的选民控制，可以调整算法 1，以证明对于除数序列方法，选民控制的四种情形均属于 P 类。然而，这只有在阈值固定时才成立，即阈值不是以选民数量n 的百分比给出。至于第二次机会模式，我们已经证明，如果阈值固定，则所有多数一致席位分配方法都能抵抗选民控制。

未来研究的另一个方向，是探究这些问题在单峰偏好 [16, 20, 21]等受限领域（或近似受限领域）中的复杂度。此外，利用整数线性规划或近似算法研究第二次机会模式下策略性竞选的有效性，也是未来研究的一个有趣方向。

为了让我们的策略性竞选更加贴近现实，我们建议研究更复杂的成本函数，例如距离贿选 [8]，在这种情形下，贿赂选民的成本取决于我们改变其选票的幅度。我们推测，在距离贿选的假设下，问题会更加困难，这一推测基于如下观察：引理1不再成立。也就是说，有些情况下，在 $\mathscr{P}_{-P^*}$ 内部移动选票比将其移至 P^*更有效。举例来说，假设有三个政党、两个可分配席位，支持度分配为 $\sigma(P^*) = 7$、 $\sigma(P_A) = 4$和 $\sigma(P_B) = 2$，阈值为 $\tau < 2$。根据德洪特法，P^* 和 P_A 各获得一个席位。假设预算为 $K = 1$，但将一张选票从 P_A 改投给 P^* 的成本为 2，而将选票从 P_B 改投给 P^* 的成本更高。因此，我们无法将任何一张选票改投给 P^*，也就是说，采用这种策略无法为 P^* 争取到任何席位。然而，如果将一张选票从 P_A 改投给 P_B 的成本为 1，那么将一张选票从 P_A 改投给 P_B，就能为 P^*争取到一个席位。

在策略性竞选的背景下，NP困难性是我们所期望的；但在胜利幅度或鲁棒性等背景下，对其的解读可以反过来，这也可以作为未来研究方向。最后，我们建议研究采用第二次机会模式时，选民对议会的满意度会在多大程度上提高。

致谢

我们感谢 Niclas Boehmer 和 Martin Bullinger 提出的真知灼见与想法；我们曾在2019年9月于达格施图尔城堡举行的研讨会 19381“面向应用的计算社会选择”期间与他们讨论这些内容。我们也感谢匿名的 IJCAI’20 和 SOFSEM’24 审稿人提出的宝贵意见。本研究部分由奥地利科学基金（FWF）通过项目编号 P31890 资助，由德国研究联合会（DFG）通过项目 BR 5207/1 和 NI 369/15（项目编号 284041127）以及 RO 1202/21-1 和 RO 1202/21-2（项目编号 438204498）资助，由波兰科学与高等教育部拨付给 AGH 的经费资助；此外，Piotr Faliszewski 获得的弗里德里希·威廉·贝塞尔研究奖（由亚历山大·冯·洪堡基金会颁发）也为本研究提供了资助。

参考文献

[1] M. Balinski 和 P. Young。《席位分配配额法》。《美国数学月刊》，82(7):701–730，1975。[2] M. Balinski 和 P. Young。《席位分配方案与配额法》。《美国数学月刊》，84(6):450–455，1977。[3] M. Balinski 和 P. Young。《公平代表：实现一人一票的理想》。耶鲁大学出版社，纽黑文，美国康涅狄格州，1982。[4] J. Bartholdi III、C. Tovey 和 M. Trick。操纵选举的计算难度。《社会选择与福利》，6(3):227–241，1989。[5] J. Bartholdi III、C. Tovey 和 M. Trick。控制选举有多难？《数学与计算机建模》，16(8/9):27–40，1992。[6] D. Baumeister 和 J. Rothe。通过投票进行偏好聚合。载于 J. Rothe 编，《经济学与计算：算法博弈论、计算社会选择与公平分配导论》，《经济学课堂伴侣》丛书，第4章，第233–367页。Springer，第2版，2024。[7] D. Baumeister、P. Faliszewski、J. Lang 和 J. Rothe。针对懒惰选民的竞选活动：截断选票。载于第*11*届自主智能体与多智能体系统国际会议论文集，第577–584页。IFAAMAS，2012年6月。[8] D. Baumeister、T. Hogrebe 和 L. Rey。广义距离贿选。载于第*33*届*AAAI*人工智能大会论文集，第1764–1771页。AAAI出版社，2019年1月/2月。[9] R. Bredereck、P. Faliszewski、M. Furdyna、A. Kaczmarczyk 和 M. Lackner。席位分配选举中的策略性竞选管理。载于第*29*届国际人工智能联合会议论文集，第103–109页。ijcai.org，2020年1月。

[10] R. Bredereck、P. Faliszewski、A. Kaczmarczyk、R. Niedermeier、P. Skowron 和 N. Talmon。多赢家投票规则的鲁棒性。《人工智能》，290：103403，2021。[11] M. Brill、J. Laslier 和 P. Skowron。作为席位分配方法的多赢家认可规则。《理论政治学杂志》，30：358–382，2018。[12] V. Conitzer 和 T. Walsh。投票中的操纵障碍。载于 F. Brandt、V. Conitzer、U. Endriss、J. Lang 和 A. Procaccia 编，《计算社会选择手册》，第6章，第127–145页。剑桥大学出版社，2016。[13] V. Conitzer、T. Sandholm 和 J. Lang。候选人较少的选举在什么情况下难以操纵？《美国计算机协会会刊》，54(3)：第14号文章，2007。[14] E. Elkind 和 P. Faliszewski。竞选活动管理的近似算法。载于第*6*届互联网与网络经济国际研讨会论文集，第473–482页，2010。[15] E. Elkind、P. Faliszewski 和 A. Slinko。选举中的克隆：寻找可能的获胜者。《人工智能研究杂志》，42：529–573，2011。[16] G. Erdélyi、M. Lackner 和 A. Pfandler。近似单峰选民群体的计算方面。载于第*27*届*AAAI*人工智能大会论文集，第283–289页。AAAI出版社，2013年7月。[17] P. Faliszewski 和 J. Rothe。投票中的控制与贿选。载于 F. Brandt、V. Conitzer、U. Endriss、J. Lang 和 A. Procaccia 编，《计算社会选择手册》，第7章，第146–168页。剑桥大学出版社，2016。[18] P. Faliszewski、E. Hemaspaandra 和 L. Hemaspaandra。选举中的贿选有多难？《人工智能研究杂志》，35：485–532，2009。[19] P. Faliszewski、E. Hemaspaandra、L. Hemaspaandra 和 J. Rothe。卢尔投票与科普兰投票在计算上能够抵御贿选和建设性控制。《人工智能研究杂志》，35：275–341，2009。[20] P. Faliszewski、E. Hemaspaandra、L. Hemaspaandra 和 J. Rothe。那面从未存在过的盾牌：拥有单峰偏好的群体更容易受到操纵和控制。《信息与计算》，209(2)：89–107，2011。[21] P. Faliszewski、E. Hemaspaandra 和 L. Hemaspaandra。近似单峰选民群体中操纵性攻击的复杂性。《人工智能》，207：69–99，2014。[22] P. Faliszewski、P. Skowron 和 N. Talmon。以贿选衡量候选人成功：基于认可的多赢家规则的复杂性结果。载于第*16*届自主智能体与多智能体系统国际会议论文集，第6–14页。IFAAMAS，2017年5月。

[23] M. Garey 和 D. Johnson。《计算机与不可约性：NP完全性理论指南》。W. H. Freeman and Company，1979年。[24] M. Garey、D. Johnson 和 L. Stockmeyer。某些简化的 NP完全图问题。《理论计算机科学》，1:237–267，1976年。[25] T. Gavenciak、D. Knop 和 M. Koutecký。参数化复杂性中的整数规划：三个小专题。载于《第13届参数化与精确计算国际研讨会论文集》，LIPIcs第115卷，21:1–21:16页。达格施图尔城堡—莱布尼茨信息学中心，2018年。[26] E. Güney。基于德洪特法的竞选活动优化混合整数线性规划。载于《2017年运筹学研究进展：德国运筹学学会年度国际会议精选论文》，73–79页。Springer，2017年9月。[27] P. Harrenstein、M.-L. Lackner 和 M. Lackner。对Gottlob Frege提出的选举制度的数学分析。《认识》，87:2609–2644，2022年。[28] E. Hemaspaandra、L. Hemaspaandra 和 J. Rothe。除了他之外的任何人：排除候选方案的复杂性。《人工智能》，171(5–6):255–285，2007年。[29] K. Jansen、S. Kratsch、D. Marx 和 I. Schlotter。固定箱子数量的装箱问题再探。《计算机与系统科学杂志》，79(1):39–49，2013年。[30] R. Karp。组合问题之间的可约性。载于 R. Miller 和 J. Thatcher（编）：《计算机计算复杂性》，85–103页。Plenum Press，1972年。[31] M. Lackner 和 P. Skowron。基于批准的多获胜者规则与策略性投票。载于《第27届国际人工智能联合会议论文集》，340–436页。ijcai.org，2018年7月。[32] C. Laußmann、J. Rothe 和 T. Seeger。带阈值的席位分配：在首选模式下策略性竞选很容易，但在第二次机会模式下却很困难。载于《第49届计算机科学理论与实践最新趋势国际会议论文集》，355–368页。Springer-Verlag《计算机科学讲义》第14519卷，2024年2月。[33] H. Lenstra Jr.。固定变量数量的整数规划。《运筹学数学》，8(4):538–548，1983年。[34] M. Neveling 和 J. Rothe。多获胜者选举中克隆候选人的复杂性。载于《第19届自主智能体与多智能体系统国际会议论文集》，922–930页。IFAAMAS，2020年5月。[35] K.-F. Oelbermann 和 F. Pukelsheim。2019年5月的欧洲议会选举：选举制度与结果，2020年。欧洲议会研究服务处研究报告。

[36] V. Ostapenko、O. Ostapenko、E. Belyaeva 和 Y. Stupnitskaya。选民之间的政党间竞争或公司之间的市场竞争的数学模型。《控制论与系统分析》，48(6):814–822，2012年。[37] M. Pellicer 和 E. Wegner。法定阈值的机械与心理效应。《选举研究》，33:258–266，2014年。[38] D. Peters。多获胜者选举中的比例性与策略无关性。载于《第17届自主智能体与多智能体系统国际会议论文集》，1549–1557页。IFAAMAS，2018年5月。[39] F. Pukelsheim。《比例代表制：席位分配方法及其应用》。Springer，第2版，2017年。[40] N. Tideman。克隆独立性作为投票规则的判据。《社会选择与福利》，4(3):185–206，1987年。[41] L. Xia。计算各种投票规则下的胜利幅度。载于《第13届ACM电子商务会议论文集》，982–999页。ACM出版社，2012年6月。[42] Y. Yang。操纵和控制基于批准的多获胜者投票的复杂性。载于《第28届国际人工智能联合会议论文集》，637–643页。ijcai.org，2019年8月。

算法 1：用于 $\mathscr{R}$-AB，其中 $\mathscr{R}$ 是除数序列方法 输入： $\mathscr{P}$, $\mathscr{V}$, τ, κ, P_1,K, ℓ1: K $\leftarrow$ min{n $-p_1$,K}2: if p_1 +K $< \tau$then 3: return 否 4: end if 5: $\sigma(P_l) \leftarrow p_1 + K$ 6: 计算 γ{对于每个政党 P $\in \mathscr{P}$，数组 γ[P] 中的条目均为 (x, cost) 对，其中 cost 是确保 P 在 P_l 获得ℓ 席位之前恰好获得 x 个席位所需的贿赂预算} 7: 初始化表 tab，包含 $\kappa - \ell$ 列和 m 行，其中 tab[0][0] $\leftarrow 0$，其余条目为 ∞8: 令 o : $\{1,\ldots,|\mathscr{P}_{-P_1}|\} \to \mathscr{P}_{-P_1}$为一个排序 9: for i $\leftarrow 1$ to $|\mathscr{P}_{-P_1}|$ do 10: for s $\leftarrow 0$ to $\kappa - \ell$ do 11: for (x, cost) $\in \gamma$[o(i)] do 12: if s $-$x ≥ 0then 13: tmp $\leftarrow$tab[i -1][s $-$ x] +cost 14: if tmp $<$tab[i][s] then 15: tab[i][s] =tmp 16: end if 17: end if 18: end for 19: end for 20: end for 21: for s $\leftarrow 0$ to $\kappa - \ell$ do 22: if tab[$|\mathscr{P}_{-P_1}|$][s] $\leq$ K then 23: return 是 24: end if 25: end for 26: return 否

算法 2：用于 LRM-AB 输入： $\mathscr{P}$, $\mathscr{V}$, τ, κ, P_1,K, ℓ1: K $\leftarrow$min$\{\mathrm{n} - \mathrm{p}_1$,K$\}$2: if $\mathrm{p}_1 +$ K $< \tau$then 3: return 否 4: end if 5: $\sigma(\mathrm{P}_l) \leftarrow \mathrm{p}_1 + \mathrm{K}$ 6: while True do 7: n $\leftarrow$ $(\sum_{P \in \mathscr{P}} \sigma(\mathrm{P})) - \mathrm{K}$ 8: if $\kappa \cdot {}^{\sigma}(\mathrm{P}_l)$ n $\geq \ell$ then 9: return 是 10: end if 11: if $\kappa \cdot {}^{\sigma}(\mathrm{P}_l)$ n $== \ell - 1$ then 12: 计算 γ13: 初始化表 tab，包含 $\kappa - \ell$ 列和 m 行，其中 tab $[0][0] \leftarrow 0$，其余条目为 ∞14: 令 o : $\{1, \ldots, |\mathscr{P}_{-P_1}|\} \to \mathscr{P}_{-P_1}$为一个排序 15: for i $\leftarrow 1$ to $|\mathscr{P}_{-P_1}|$ do 16: for s $\leftarrow 0$ to $\kappa - \ell$ do 17: for (x, cost) $\in \gamma$[o(i)] do 18: if s $- \mathrm{x} \geq 0$ then 19: tmp $\leftarrow$tab[i $- 1$][s $- \mathrm{x}$] + cost 20: if tmp $<$tab[i][s] then 21: tab[i][s] = tmp 22: end if 23: end if 24: end for 25: end for 26: end for 27: for s $\leftarrow \overline{0}$ to $\kappa - \ell$ do 28: if tab$[|\mathscr{P}_{-P_1}|]$[s] $\leq$ K then 29: return 是 30: end if 31: end for 32: end if 33: 令 P_i 为支持度最小的正值政党 34: if $\sigma(\mathrm{P}_i) - \tau + 1 \leq$ K then 35: {将 P_i 推至阈值以下} 36: $\sigma(\mathrm{P}_i) \leftarrow 0$37: K $\leftarrow$K $- (\sigma(\mathrm{P}_i) - \tau + 1)$ 38: else 39: return 否 40: end if 41: end while