arXiv:2603.05118v1 [cs.DC] 5 Mar 2026

利用结构知识解决具有共享随机性的匿名网络中的选举问题

J´er´emie Chalopin[1] 和 Emmanuel Godard[1]

{jeremie.chalopin,emmanuel.godard}@lis-lab.fr 法国国家科学研究中心与艾克斯-马赛大学´

摘要。 我们研究匿名网络中的经典选举问题。在这类网络中，解决方案可以依赖随机比特，这些随机比特可以在节点间共享，也可以不共享。对于任意结构知识，我们完整刻画了随机化选举算法存在的条件。我们的分析考察了拉斯维加斯和蒙特卡洛随机化算法，并分别考虑共享与非共享随机性的假设。在我们的设定中，如果随机源的输出比特在特定节点子集中完全相同，就认为这些随机源是共享的。

这些算法和不可能性证明是对确定性情形下 [5]相关结果的扩展。我们的结果完整推广了 [8]中的结果。此外，作为应用，我们考察了多种具体的知识类型：无知识、规模上界、共享同一源的节点数量上界、网络规模，或网络的完整拓扑。对于每一种知识类型，我们都展示了如何应用一般刻画，并说明它们实际上对应于不同的结构知识类别。我们还描述了文献中的随机化选举算法如何融入这一图景。因此，我们全面展现了知识如何影响具有共享随机性的任意匿名图中选举问题的可计算性。

1 引言

领导者选举（或称选举）问题是分布式计算理论的范例问题之一。如果在最终配置中，恰好有一个进程被标记为 Elected ，而所有其他进程都被标记为 Non-Elected，则分布式算法解决了选举问题。选举算法是许多其他分布式算法的构建模块：当选顶点充当协调者、发起者，并且更一般地执行某种特殊角色（参见 [23, 第262]页）。LeLann [13] 首次研究了选举问题，并给出了环中的一种解决方案，其中每个进程都具有唯一名称。该问题的解决方案基于两种经典假设进行研究（详情请参见 [20, 第 3] 章）：每个进程都通过唯一名称（或标识符）进行标识，即具有自己的标识；进程初始时具有相同的状态（匿名网络）。如果进程初始时具有唯一标识符，则总能解决该问题，例如，

选举具有最小标识符的进程。然而，如果我们考虑进程不具有唯一标识符的匿名/同名网络，就不一定能够解决选举问题。Angluin [1] 引入了用于证明选举算法不可能实现的经典证明技术，即使已知网络的完整拓扑结构也是如此。该技术基于图覆盖，这是代数拓扑中的一个已知概念[14]。最后，人们已经获得了若干种图的刻画，用于描述哪些图上存在选举算法 [3, 25, 16, 5]。

众所周知，在某些情况下，随机化算法可以解决选举问题，而 Angluin 式结果却表明确定性解不存在。此类结果的一个早期例子是 [15]。人们常说，随机性有助于在选举中实现“对称性破缺”，但认为随机性总是足以找到解是一种误解。本文提出了新的不可能性结果；当缺乏足够的结构知识时，这些结果对随机化算法仍然成立。对于任意给定的知识，在共享和非共享随机源的一般情形下，我们分别给出了拉斯维加斯算法和蒙特卡罗算法解决选举问题的两个一般性可解性刻画。我们的贡献在于明确了可计算性的确切界限。简而言之，本文阐明了随机性如何帮助（或无法帮助）利用底层网络的结构知识来真正打破对称性并解决选举问题。我们还讨论了我们的不可能性结果与已知随机化选举算法之间的对应关系。

利用知识进行计算。 我们将给定的任意知识编码为一个任意图族 $\mathcal{F}$，即具有给定知识值的图的集合（例如，规模相同的图）。利用这种知识求解，意味着要找到一个在 $\mathcal{F}$ 中任意图上都正确的选举算法。根据一般性刻画，我们在应用章节中考察以下经典研究的结构知识。我们用 $\mathcal{G}$ 表示所有图的集合。我们考察

– 无知识 $\mathcal{F} = \mathcal{G}$；– 对规模 S 的界限，$\mathcal{F} = \{\mathsf{G} \in \mathcal{G}, |\mathsf{G}| \leq S\}$；– 对网络规模的严格 2–近似，$T$ $\mathcal{F} = \{\mathsf{G} \in \mathcal{G}, \frac{1}{2}T \lessdot |\mathsf{G}| \leq T\}$；– 网络规模 S，$\mathcal{F} = \{\mathsf{G} \in \mathcal{G}, |\mathsf{G}| = S\}$；– 网络的拓扑 G，$\mathcal{F} = \{\mathsf{G}\}$。

我们将在第 5.1 节看到，这实际上涵盖了所有可能的可计算性情况。

根据 Angluin 的研究及后续成果可知，所谓的“对称”图，或者更准确地说，如后文所述的“非极小”图，即使已知网络 $\mathbf{G}$ 的确切拓扑，也无法通过确定性算法完成选举（在此情形下为 $\mathcal{F} = \{\mathbf{G}\}$）。本文证明，只要具备足够的结构知识，就可以在任何匿名网络中使用蒙特卡罗随机化算法选举出领导者。从某种意义上说（后文将对此作出形式化说明），对于使用非共享随机源的随机化算法而言，所有网络都是极小的。这与确定性情形不同：在确定性情形下，即使已知整个图的拓扑，某些匿名网络仍无法

通过任何选举算法完成选举。因此，我们全面阐释了随机性如何影响任意匿名图中选举问题的可计算性。一般情况下，如果已知网络规模的上界，则只能使用蒙特卡罗算法。当随机源能够破除对称性时（例如至少存在一个非共享随机源），若已知网络规模，则存在拉斯维加斯选举算法。当网络在没有随机源的情况下本身已能破除对称性（即为覆盖极小）时，对于任何结构知识，确定性算法与随机化算法的计算能力相同。这些结果表明，从分布式可计算性的角度来看，随机性有助于利用知识，在已知拓扑但不存在确定性选举算法时，构造随机化选举算法。最后，研究还表明，如果要求显式终止，那么在缺乏足够网络信息时，即使使用随机化算法，也仍可能无法完成选举。从某种意义上说，随机比特即使是共享的，也确实能够破除对称性，但不足以实现终止检测。要充分利用随机化，还需要补充知识。

证明技巧与贡献。关于如何利用知识在确定性情形下解决选举问题的一般性刻画已在[5]中完成。其主要工具是准覆盖的概念。这个概念刻画了由“所有树和一个三角形”组成的族中出现的现象，Angluin 曾将其概括为：“一个图中存在足够大的区域，看起来像另一个图”（[1] 第 87 页第 13–17 行）。本文考虑节点能够访问（可能共享的）独立同分布随机源的网络。为了表示共享源的随机化算法所涉及的对称性，我们考虑节点以其能够访问的随机源为标签的图。因此，在共享随机性情形下，共享同一随机源的两个节点具有相同的标签。该标记记作 b，具有相同标签的节点集合称为一个 B–类。然而，由于一般而言节点无法（直接）访问自己的标签 b，因此无法直接使用 [5] 中关于 B–标记图的结果。

除了证明 [5] 中的证明方法可以（经过一些技术性处理）扩展到共享源的新设定之外，本研究的主要意义还在于其对标准知识的应用，如第 5.1 节所示：

– 知识仅分为3类（精确规模/拓扑、某些界、无知识），通过定理1和定理2，我们得到了可计算性的精确层级– 已知的经典随机化选举算法很好地融入了这一层级，我们的工作解释了其中的方式和原因。另请参阅第5节中的详细讨论。– 据我们所知，第 5.2 节考虑的知识是首次针对直径无界且仍可求解随机化选举的网络进行研究。鉴于文献中的已知结果，这一点或许令人意外。

相关工作。 领导者选举是分布式计算中的一个长期研究问题。Angluin 引入图覆盖，以刻画匿名和同名分布式系统中的所有对称性。消息传递模型中相关的对称性概念由图的对称覆盖刻画，参见第3.2节。许多研究者扩展了这项工作，并获得了关于哪些图和确定性模型存在选举算法的若干刻画 [3, 25, 16, 5]。

正如近期在 [2]中所示，共享随机源可用于降低算法复杂度。然而，就可计算性而言，所有节点共享同一个源实际上可能会产生负面影响。关于知识相关的时间复杂度和消息复杂度，随机化选举方面的最新进展见 [12]。

采用任意知识的最接近的先前工作，是确定性情形下的 [5] ，以及共享随机性一般设置下的 [8]。相较于 [5]，本文的创新之处在于将确定性准覆盖刻画扩展到随机化设置。在 [8], 中，引入了共享随机源的概念（称为“有偏”）。与 [8], 相比，我们考虑相同的随机性设置，但研究的图拓扑比团更一般。由于当且仅当一个团的 B–类的规模的最大公约数为 1 时，该团才是B–覆盖最小的，因此可以在此推导出 [8], 中给出的刻画。我们强调，我们的结果展示了结构知识对随机化算法的影响。从某种意义上说，在团中，节点可根据度在本地推导出规模，进而推导出网络拓扑。在这一图族中，无知识与完全知识并无区别。

另请注意，我们不考虑带隐式终止的选举（即最终稳定为仅有一个 Elected 节点）。从知识的角度来看，这种情形已在 [18], 中得到解决：即使没有任何知识，也可以通过蒙特卡洛算法实现带隐式终止的选举。

2 模型与主要技术结果

我们的模型是通常的异步消息传递模型（[24, 25, 5]）。网络由一个简单连通图 $G = (V(G), E(G))$ 表示，其中顶点对应进程，边对应直接通信链路。每个进程的初始状态由与相应顶点 $v \in V(G)$ 关联的标签 $\lambda(v)$ 表示；我们将这样的标记图记为 $\mathbf{G} = (G, \lambda)$。当$\lambda(v) \neq \lambda(u)$ 对两个不同的节点 u 和 v成立时，我们可以认为 $\lambda(v)$ 是节点 v的标识 。当对所有节点 u, v， $\lambda(u) = \lambda(v)$ 均成立时，该网络称为匿名的。一般情况下，该网络称为同名的。

我们假设每个进程都能区分与其关联的不同边，即对于每个 $u \in V(G)$, 存在一个双射 δ_u ，将 G 中 u 的邻居与 $[1,\deg_G(u)]$对应起来。我们用 δ 表示函数集合 $\{\delta_u \mid u \in V(G)\}$。每个顶点为其邻居所关联的编号称为端口编号， δ 称为 G的端口编号 。每个进程 v 在每一步都可以访问随机比特源 $b(v)$。我们用 $b(u, t)$表示所抽取的第 t–个比特

由 u 在源 $b(u)$处抽取。在 u 与 v共享同一源的情况下，这些节点总是得到与对方相同的输出，即对所有 t，有 $b(u,t) = b(v,t)$。我们还用 b 表示附着于给定顶点的随机源所对应的标签。我们用$(\mathbf{G}\ \delta, b)$表示带有端口编号 δ和源点 b的标记图$\mathbf{G}$。

网络中的每个进程 v 都代表一个实体，它能够执行计算步骤、从自己的（可能是共享的）源中抽取随机比特、通过某个端口发送消息，并通过某个端口接收对应邻居发送的任意消息。我们考虑异步系统，即每个计算步骤所需的时间都无法预知（但有限）。我们只考虑可靠系统：进程或通信链路均不会发生故障。我们还假设信道是先进先出的，即对于每条信道，消息都按发送顺序送达。消息送达取决于对手选择的调度。

如果对于从 $\mathcal{F}$中的任意图开始的每次执行，都会到达一个最终配置，其中恰有一个节点处于状态 ELECTED ，而所有其他节点都处于状态 NON-ELECTED，那么确定性算法就在族 $\mathcal{F}$ 上解决了选举问题。我们假设状态 ELECTED 和 NON-ELECTED是终止状态，即节点一旦进入此状态，之后就不会离开。随机化算法是指可在任意步骤使用随机源 b 的算法。我们认为随机源在时间上均匀，即在任何一次调用中，抽取到 1 或 0 的概率相同且非零；并且这些源是独立同分布的。给定一个随机化算法和一个固定调度，节点 u在时间 s 的局部状态是一个随机变量 $state(u,\ s)$，它可能取决于先前的状态和传入消息。一次执行的概率由全局状态的随机变量决定。因此，由于调度固定，该概率就是共享源上相应随机抽取集合的概率。以下定义是标准定义。拉斯维加斯算法 是指以概率 $0 < p \leq 1$终止，且其最终配置正确的算法（即恰有一个节点处于 ELECTED 状态，所有其他节点都处于 NON-ELECTED 状态）。蒙特卡罗算法 是指总能终止，且其最终配置以概率$0 < p \leq 1$正确的算法。若随机化算法对于任何固定调度都正确，则称其在异步非自适应调度器 下解决了该问题。当对手能够在了解所有局部状态以及已抽取的随机比特后，选择下一步要调度的节点时，我们称该对手为自适应对手。这里，不可能性证明是在非自适应设置下给出的，而算法的正确性则是在自适应设置下证明的。

2.1 主要结果

对称覆盖是一种图同态，要求每个节点处都存在局部双射，形式化定义见第 3.2 节。从现在起，我们假设所有覆盖都是满射；当应用于标记图时，每个标签都保持不变。准覆盖直观上是一种部分图同态

它在图的某个子部分上表现得像覆盖，形式化定义见定义 4.1。这些概念也扩展到图的 B–标记版本（$\mathbf{G}$ b）。为了与底层图上的覆盖明确区分，我们将扩展后的对象称为 B–覆盖和 B–准覆盖。若一个图的任何覆盖实际上都是同构，则称该图为极小图。若图 G 带有B–标记 b，则称其为 B–极小图，当且仅当（G, b）是极小的。当不存在共享源的节点对时，所有节点的标签都不同，且（G, b）是 B–极小的。当至少存在一个非共享随机源（即仅由一个节点使用的随机源）时，情况也如此。

我们考虑任意递归图族 $\mathcal{F}$（编码为对称B–标记有向图，详情见后文），也就是说，存在一个算法可以判定给定的图是否属于 $\mathcal{F}$。这是为了获得以 $\mathcal{F}$ 为参数的实际算法而作出的自然假设，因此，我们给出的选举算法是可能的最一般选举算法。以下定理通过B–标记，概括了共享源和非共享源两种情形。

定理 2.1. 设 $\mathcal{F}$ 是一个由连通对称 B–标记有向图构成的递归族。当且仅当 $\mathcal{F}$ 中的每个标记有向图都是 B–极小图，并且存在一个递归函数 $\tau : \mathcal{F} \to \mathbb{N}$，使得对于 $\mathcal{F}$中的每个标记对称有向图 $(\mathbf{D}\ b)$，在 $\mathcal{F}$中不存在半径大于 $\tau(\mathbf{D})$ 的 $\mathbf{D}$ 的准覆盖（除$\mathbf{D}$ 本身外），才存在一个针对 $\mathcal{F}$ 的拉斯维加斯选举算法。

采用蒙特卡罗算法允许出现一些错误运行，因此也可以考虑非 B–极小图；此外，只需考虑真拟覆盖，即并非实际覆盖的准覆盖。

定理 2.2。设 $\mathcal{F}$ 是一个由连通对称 $B-$ 标记有向图组成的递归族。当且仅当存在一个递归函数 τ： $\mathcal{F} \to \mathbb{N}$，使得对于 $\mathcal{F}$ 中的每个标记对称有向图 $\mathbf{D}$，$\mathcal{F}$ 中不存在半径大于 $\tau(\mathbf{D})$ 的 $\mathbf{D}$ 的真拟覆盖（$\mathbf{D}$ 本身除外）时，才存在针对 $\mathcal{F}$ 的蒙特卡洛选举算法。

3 对称覆盖与标记图的选举问题

3.1 预备知识

下文中，我们将考虑含有多重弧和自环的有向图。有向图 $D = (V(D)\ A(D)\ s_D, t_D)$ 由顶点集 $V(D)$、弧集 $A(D)$ 以及两个映射 s_D 和 t_D 定义；这两个映射为每条弧指定 $V(D)$ 中的两个元素：一个源和一个目标点（通常会省略下标）。如果 a 是一条弧，则称弧 a 从 $s(a)$ 出发并进入 $t(a)$；我们也称 $s(a)$ 和 $t(a)$ 与 a 关联。设 a 是一条弧，若 $s(a) = u$ 且 $t(a) = v$，则 v 是 u 的出邻点，而 u 是 v. 的入邻点。对称有向图 D 是赋有对称性的有向图，也就是说，它具有一个对合 Sym : $A(D) \to A(D)$，使得对于每个 $a \in A(D)$ $s(a)$，$= t(Sym(a))$。

在对称有向图 D 中，顶点 v 的度为 $\deg_D(v) = |\{a \mid s(a) = v\}| = |\{a \mid t(a) = v\}|$，我们用 $N_D(v)$ 表示 v 的邻居集合，该集合等于 v 的出邻居集合，也等于 v.的入邻居集合。

给定两个顶点 u、$v \in V(D)$，有向图 D中从 u到 v的长度为 p的路径 π是弧序列 a_1、a_2、$\ldots a_p$，满足 $s(a_1) = u$、$\forall i \in [1, p-1]$、$t(a_i) = s(a_{i+1})$，且 $t(a_p) = v$。如果对于每个 $i \in [1, p-1]$、$a_{i+1} \neq Sym(a_i)$，π是非停顿的。如果对于所有顶点 u、$v \in V(D)$，在 D中都存在一条从 u 到 v 的路径，则有向图 D是强连通的。在对称有向图 D中，两个顶点 u和 v之间的距离记为 $\mathrm{dist}_D(u, v)$，它是 D中从 u 到v 的最短路径长度。在对称有向图 D 中，我们将 $\mathbf{B_D}(v_0, r)$记为以 $v_0 \in V(D)$为中心、半径为 r 的标记球，其中包含与 v_0 的距离至多为r 的所有顶点，以及源或目标点与 v_0的距离至多为 $r-1$ 的所有弧。

从有向图 D 到有向图 D' 的同态 γ 是一个映射γ: $V(D) \cup A(D) \to V(D') \cup A(D')$，满足对每条弧 $a \in A(D)$，$\gamma(s(a)) = s(\gamma(a))$且 $\gamma(t(a)) = t(\gamma(a))$。若同态 $\gamma: D \to D'$ 是双射，则称其为同构；此时记作 $D \simeq D'$。本文将考虑顶点和弧均带有来自标签集 $L \times B$ 的标签的有向图，其中 L 是递归集（表示假名等任何附加信息），而 B 表示随机源集合。给定标签集L，在 L 上标记的有向图 D 记为(D, λ)，其中 λ: $V(D) \cup A(D) \to L$ 是标记函数。映射 γ: $V(D) \cup A(D) \to V(D') \cup A(D')$是从$(D, \lambda)$到$(D', \lambda')$的同态，当且仅当 γ 是从D 到 D' 的有向图同态且保留标记，即对每个$x \in V(D) \cup A(D)$，均有 $\lambda'(\gamma(x)) = \lambda(x)$。标记有向图将用粗体字母表示，例如D、D′，.....

给定标签集合 $L = \mathbb{N} \times \mathbb{N}$，我们用 $\mathcal{D}_{L \times B}$ 表示所有满足以下条件的对称有向图 $\mathbf{D} = (D, \lambda)$ 的集合：对于每个 $a \in A(D)$，存在 $p, q \in \mathbb{N}$ 使得 $\lambda(a) = (p, q)$ 且 $\lambda(Sym(a)) = (q, p)$；并且对于每个 $v \in V(D)$，$\lambda(v) \in L$ 且$\{p \mid \exists a, \lambda(a) = (p, q)$，以及 $s(a) = v\} = [1, \deg_D(v)]$。换言之，$\mathcal{D}_{L \times B}$ 是这样一类有向图：它们在局部上与某些由简单标记图 G 得到的有向图相似，其中端口编号标签取自 $\mathbb{N}$。

从图到对称有向图的构造，我们沿用 [5] 的表述。设 (G, λ) 是一个带有端口编号 δ. 的标记图。我们将通过以下方式构造的对称标记有向图 $(\mathrm{Dir}(G), \delta)$，记作 $(\mathrm{Dir}(G), (\lambda, \delta))$。$\mathrm{Dir}(G)$ 的顶点就是 G 的顶点，并且它们在 G 和 Dir(G) 中具有相同的标签。G 的每条边 $\{u, v\}$ 在 $(\mathrm{Dir}(G), \delta)$ 中都被替换为两条弧 $a_{(u,v)}, a_{(v,u)} \in A(\mathrm{Dir}(G))$，使得 $s(a_{(u,v)}) = t(a_{(v,u)}) = u$, $t(a_{(u,v)}) = s(a_{(v,u)}) = v$, $\delta(a_{(u,v)}) = (\delta_u(v)\ \delta_v(u))$ 且 $\delta(a_{(v,u)}) = (\delta_v(u)\ \delta_u(v))$。注意，这个有向图不含多重弧或自环。我们研究中使用的对象是 $(\mathrm{Dir}(G), (\lambda, \delta, b))$，而结果均以对称标记有向图为对象，在完全一般的情形下表述。

在一次执行过程中，如果一个节点有待处理操作或待处理的传入消息，就称其为可激活节点。调度器是在每一步反复选择一个非空可激活节点子集所构成的序列。当节点可以在本地访问（如上文所形式化定义的）

数量有限的随机比特时，分布式算法是概率性的。为简单起见，我们在此假设每次调用时都可以访问恰好一个随机比特。我们同时考虑异步设置（每一步由对手选择一个可激活节点子集）和同步设置（每一步都激活所有节点并传递它们的消息）。

3.2 对称覆盖

本节介绍第一个工具：对称覆盖；随后回顾能够在某种知识条件下实现拉斯维加斯选举的标记图的刻画，并介绍选举算法及其主要性质。这里的算法是对 [5], 的改编，更多证明见附录。对称覆盖这一基本概念见 [4]。

若存在从标记有向图 D 到标记有向图 D′ 的同态 φ （记作 φ ），并且对于每条弧 $a' \in A$（D'），以及每个顶点 $v \in \varphi^{-1}$（t（a'））（分别地， $v \in \varphi^{-1}$（s（a'））），都存在唯一一条弧 $a \in A$（D），使得 t（a） $= v$ （分别地， s（a） $= v$）且 φ（a） $= a'$，则称标记有向图 D 是标记有向图 D′ 的覆盖。若对称标记有向图 D 是对称标记有向图 D′ 的覆盖（记作 φ ），并且对于每条弧 $a \in A$（D），都有 φ（Sym（a）） $= Sym$（φ（a）），则称 D 是 D′ 的对称覆盖。这个同态 φ 是从 D 到 D′ 的对称覆盖投影。若不存在与 D 不同构的对称标记有向图 D′ ，使得 D 是 D′ 的对称覆盖，则称对称标记有向图 D 对称覆盖极小，或简称极小。为与 $B-$ 标记有向图区分，当（D b）是对称覆盖极小时，我们称有向图（D b）为对称覆盖极小，或简称 $B-$ 极小。注意，当源点是非共享的，每个标签 $B-$ 只出现一次；因此，至少带有一个非共享源的所有网络都是 $B-$ 极小。采用以下记号：给定源点集合 S（或等价地，标签集合 $B-$）和 $t \in \mathbb{N}$，我们用 X_S 表示对应于从 S 的每个源点进行抽样所得的随机变量。

下面的引理说明了在处理匿名网络时，对称覆盖的重要性。这是 Angluin 针对简单图的覆盖所给出的提升引理的对应版本 [1] ，证明见 [3,6]。在这里，我们将其推广到随机源：可以通过覆盖，将下层图中的执行步骤转移（即提升）到相应节点；这样一来，随机化执行在上层图 $\mathbf{D}$ 上仍会以相同的概率产生 $\mathbf{D}'$ 的对称覆盖。

引理 3.1（概率提升引理）。设 $\mathbf{D}$ 和 $\mathbf{D}'$ 是两个$B-$标记的对称有向图，$\mathcal{D}_{L\times B}$. 。如果 $\mathbf{D}$ 是 $\mathbf{D}'$ 经由 φ 的一个对称覆盖，那么算法 $\mathcal{A}$ 在 $\mathbf{D}'$ 上的任意一次以概率 $p > 0$ 发生的执行，都可以提升为在$\mathbf{D}$上的一次以概率 p发生的执行，使得执行结束时，对于任意 $v \in V(D)$， v 与 $\varphi(v)$ 处于相同状态。

证明。我们针对在$\mathbf{D}'$上执行的一步来证明这一点：该步骤以概率$p > 0$ 在节点集合 V' 上执行。对于每个满足 $\varphi(v) = v'$ 且具有 $v' \in V'$的 v，我们都可以借助局部双射执行相同的步骤。由于

$B-$对称覆盖，所有这些 v 都与 v'共享相同的随机源，因此$\mathbf{D}$ 的新全局状态出现的概率为 $Pr(X_{S'} = x')$，其中 S' 是与 V' 对应的源点集合，而 x' 是在这一步执行中得到的比特。这恰好就是概率 p。

3.3 $B-$标记图中的选举与对称覆盖

首先，我们给出一个刻画：在异步消息传递系统中可以解决选举的网络。该表述意味着，目前我们假设已知图的拓扑结构。覆盖的层数q 是任意节点的原像数量（由于是覆盖，当覆盖为满射时，所有节点都具有相同数量的原像）。

定理 3.2. 给定一个带有端口编号的 $B-$标记图 $\mathbf{G} = (G, \lambda, b)\,\delta$，当且仅当$(Dir(G), (\lambda, b, \delta))$ 是对称覆盖极小时，才存在针对$(\mathbf{G}\ \delta)$ 的拉斯维加斯选举算法。

证明（必要性部分）。该定理的必要性直接由引理 3.1 得出。假设我们有一个针对$\mathbf{G}$的拉斯维加斯选举算法，考虑$\mathbf{G}' \neq \mathbf{G}$ ，使得$\mathbf{G}$ 是$\mathbf{G}'$的一个覆盖。尽管该选举算法是为$\mathbf{G}$设计的，但我们可以考虑该算法在$\mathbf{G}'$上的同步执行。每一步都可以提升为$\mathbf{G}$上的同步执行，因此它将以某个正概率终止。现在考虑带有最终标记的图：它与一个对称覆盖一致，并且由于$\mathbf{G}' \neq \mathbf{G}$，层数大于 2，因此$\mathbf{G}$上将有多于一个 $elected$ 节点。矛盾。

3.4 已知规模的拉斯维加斯选举算法

充分性部分需要用到以下命名算法（后文也会使用）。命名算法的目标是得到一个所有节点都具有唯一身份标识的最终配置。同样，这是许多其他分布式算法正确运行的基本前提，因为这些算法只有在所有节点都能被明确识别的假设下才能正确工作。枚举问题是命名问题的一种变体。分布式枚举算法的目标是为每个网络顶点分配一个唯一整数，从而在顶点集合 $V(G)$ 与 $\{1, 2, \ldots, |V(G)|\}$ 之间建立双射。

算法 1 描述了一种已知规模的随机化枚举算法；由此，只需将编号为 $|V(G)|$ 的顶点视为已当选，即可得到一种选举算法（顶点知道 $|V(G)|$）。该算法是 [5] 的扩展，其灵感源自 [6], 中提出的方法；后者则是对 Mazurkiewicz 在 [16] 中给出的枚举算法的改编。

非正式描述。 我们首先给出算法$\mathcal{M}$的一般描述：它在具有端口编号δ 和随机源 b的连通带标签简单图$\mathbf{G}$上执行。

在算法执行期间，每个顶点 v 都会尝试获取自己的唯一标识，该标识是介于 1 和 $|V(G)|$ 之间的一个数字。在每一步

中，算法都会调用随机源 b，以便逐步生成一个二进制序列 $b(v)$。最终，该序列会与其他不共享同一随机源的节点所生成的序列不同。顶点 v 选定数字 $n(v)$ 后，会将该数字连同初始标签、端口编号 $\delta_v(u)$ 和序列 b 一起发送给每个邻居 u。当顶点 u 收到来自某个邻居 v的消息时，会将数字 $n(v)$ 连同端口编号 $\delta_u(v)$ 和$\delta_v(u)$ 一并存储。根据从邻居处收集到的所有信息，每个顶点都可以构建自己的局部视图（即与相应端口编号相关联的邻居数字集合）。随后，顶点会广播自己的数字、标签、 b 序列以及邮箱（其中包含一组局部视图）。如果顶点 u 发现另一个顶点 v 具有相同的暂定数字，就应决定是否更改自己的数字。为此，它会将自己的局部视图与 v 的局部视图进行比较。如果 u 的标签与随机比特序列 $b(u)$，或 u 的局部视图严格较弱，那么 u 就会选取另一个数字——即新的临时标识——并再次广播该数字以及自己的局部视图和新的 b 序列。计算结束前，节点会等待，直到在邮箱中看到数字 $|V(G)|$。此时，如果有向图 $(Dir(G), \bar{(}\lambda, \delta))$ 是对称覆盖极小的，那么每个顶点都将拥有一个来自 $\{1, 2, \ldots, |V(G)|\}$ 的唯一数字：该算法就是一个枚举算法。

标签。我们考虑一个网络（$\mathbf{G}$ δ），其中$\mathbf{G}$ $= (G,$ $\lambda)$是一个简单标记图，且 δ 是$\mathbf{G}$的端口编号。函数 $\lambda : V(G) \to L$ 是初始标记。注意， B–标签（即随机源）对节点来说是未知的。

我们假设存在一个全序 $<_L$，定义在 $L.$ 上。我们将该序 $<_L$ 扩展到$L \cup \{\perp\}$（假设 $\perp \notin L$）如下：对所有 $\ell \in L$， $\perp < \ell$。

如果 b 在二进制字符串的字母序中位于 b' 之前，我们就称$b <_B b'$。

在执行过程中，每个 v的标签是一个元组（$\lambda(v)$ b， $n(v)$ $\bar{N}(v)$ $M(v)$），其中：

– $\lambda(v) \in L$ 是 v.的初始标签。– $n(v) \in \mathbb{N}$ 是算法计算出的当前 v 数量；初始时$n(v) = 0$。– $b(v)$是一个 $\{0, 1\}$的有限序列。– $N(v) \in \mathcal{P}_{\text{fin}}($ $\mathbb{N} \times L \times \{0, 1\}^* \times \mathbb{N}^2)$，其中 $\mathcal{P}_{\text{fin}}(–)$表示有限子集的集合，是 v的局部视图。在执行结束时，如果$(m, \ell, b, p, q) \in N(v)$，那么 v 有一个邻居 u，其编号为 m，标签为 ℓ，随机序列以b 为前缀，并且从 u 到 v 的弧标记为(p, q)。初始时$N(v)$ $= \{(0, \perp, \varepsilon, 0, q) \mid q \in [1, \deg_G(v)]\}$。– $M(v)$是一个集合，即 v的邮箱；初始时 $M(v) = \emptyset$。 $M(v)$中的元素具有以下形式：(m, ℓ, b, N)，其中 $m \in \mathbb{N}$、$\ell \in L$、 b属于 $\{0, 1\}^*$，且 N是一个局部视图。它包含 v 在算法执行过程中收到的所有信息。如果$(m, \ell, b, N) \bar{\in} M(v)$，则表示在执行过程中的某个先前步骤，存在一个顶点 u，使得 $n(u) = m$， $\lambda(u) = \ell$， $b(u) = b$，并且 $N(u)$ $= N$。 – –

消息。节点交换形式为 $<(n,$ $\ell,$ $b,$ $M)$ $p >$的消息。如果顶点 u 向其一个邻居 v发送消息 $<(n,$ $\ell,$ $b,$ $\bar{M})$ $p >$，那么

消息包含以下信息： n 是当前数量 $n(u)$，对应于 u；ℓ 是 u 的标签 $\lambda(u)$； b 是随机比特序列； M 是 u 的邮箱；以及 $p = \delta_u(v)$。

局部视图上的全序。该算法的有趣性质依赖于局部视图上的全序。为考虑随机比特序列，需要对其进行调整，使之区别于 [5] 。其思路是在 λ 标签后附加随机比特序列；比较这些扩展后的标签时，采用字母序。这样，来自同一节点的先前序列总是排在当前序列之前。

给定两个不同的集合 N_1， $N_2 \in \mathcal{P}_{\text{fin}}(\mathbb{N} \times L \times \{0,1\}^* \times \mathbb{N}^2)$，若对字典序而言，对称差 $N_1 \triangle N_2 = (N_1 \setminus N_2) \cup (N_2 \setminus N_1)$ 中的最大元素属于 N_2，则定义 $N_1 \prec N_2$。也可以说，$(\ell,\ b,\ N) \prec (\ell',\ b',\ N')$，当且仅当 $\ell <_L \ell'$，或者 $\ell = \ell'$ 且 $b <_B b'$；或者 $\ell = \ell'$ 且 $b = b'$ 且 $N \prec N'$。我们用 $\preceq$ 表示 $\prec$的自反闭包。比较邮箱（判断相等或集合包含关系）时，不考虑比特序列，并继续使用相同的 $=$ 或 $\leq$ 运算符，以简化记法。

考虑图 (G, b, δ) 上某个顶点 v 在算法 $\mathcal{M}$ 执行期间的邮箱 $M = M(v)$。如果不存在 $(n, \ell', b', N') \in M$ 使得 $(\ell, b, N) \preceq (\ell', b', N')$，则称元素 $(n, \ell, b, N) \in M$ 在 M 中是极大的。我们用 $S(M)$ 表示 M 的极大元素集合。根据命题 3.3，在算法 $\mathcal{M}$ 的每一步之后，$(n(v)\ \lambda(v)\ b, N(v))$ 都是 $M(v)$ 中的极大元素。如果集合 $S(M)$ 非空，并且对于所有 $(n_1, \ell_1, b_1, N_1) \in S(M)$、所有 $(n_2, \ell_2, b_2, p, q) \in N_1$，以及 $p \neq 0$，$n_2 \neq 0$ 和 $\ell_2 \neq \perp$ 对于 $(n_2, \ell_2', b', N_2') \in S(M)$（根据极大性仅有一个），均有 $\ell_2 = \ell_2'$，b' 是 b_2 的前缀，且 $(n_1, \ell_1, b_1, q', p') \in N_2'$，则称集合 $S(M)$ 是相干的。

动作**I** 只有在唤醒时未收到任何消息的情况下，节点才能执行。它选择数字 1，更新其邮箱并通知其邻居。

动作**R**描述了顶点 v_0 通过端口 q_1从邻居接收消息 $< (n_1, \ell_1, b_1, M_1)\ p_1 >$时必须遵循的指令。首先，它记忆该消息，并将 M_1 添加到其邮箱中以更新邮箱。然后，当其数字等于 0，或存在 $(n(v_0)\ \ell', b', N') \in M(v_0)$ 使得 $(\lambda(v_0)\ b(v_0)\ N(v_0)) \prec (\ell', b', N')$ 时，它会修改自己的数字。新的数字是下一个可用的数字。随后，它通过移除与端口 q_1 对应的元素（如果存在），并将$(n_1, \ell_1, b_1, p_1, q_1)$添加到 $N(v_0)$中，来更新其局部视图。然后，它将新状态添加到邮箱中。最后，如果执行所有这些指令后其邮箱发生了修改，它就向所有邻居发送其数字和邮箱。如果 v_0 的邮箱在执行动作**R**后未被修改，则意味着 v_0 关于其邻居的信息（即其数字）是正确的， M_1 的所有元素已经属于 $M(v_0)$，并且对于每个$(n(v_0)\ \ell, b(v_0)\ N) \in M(v_0)$，都有 $(\ell, b, N) \preceq (\lambda(v_0)\ b(v_0)\ N(v_0))$。当 $n(v_0) = |V(G)|$且 v_0 已当选时，或当 $|V(G)|$ 出现在 $M(v_0)$中且 v_0 未当选时，该算法停止。

动作 **C** 用于扩展随机比特序列，以帮助打破不共享同一随机源的节点之间的对称性。只有当$M(v)$相干时，才能应用此动作，从而避免阻碍**R** 规则的执行。

算法 1： 算法 $\mathcal{M}$，其中 $n = |V(G)|$。

I：$\{n(v_0) = 0$，且没有消息到达 $v_0\}$

开始

  $n(v_0) := 1$；

  $\bar{b}(v_0) := \bar{b}(v_0)rbit()$;

  $M(v_0) := \{(n(v_0)\ \lambda(v_0)\ b(\bar{v}_0)\ \emptyset)\}$；

  对于 $i := 1$ **到** $\deg(v_0)$，**执行**

    发送 $<(n(v_0)\ \lambda(v_0)\ b(\bar{v}_0)\ M(v_0))\ i>$ 经由 i；

R：$\{$一条消息 $<(n_1,\ \ell_1, \overline{b_1},\ M_1)\ p_1>$ 已通过端口 $q_1\}$到达 v_0 **开始**

  $M_{old} := M(v_0)$;

  $M(v_0) := M(v_0) \cup M_1$；

  如果 $n(v_0) = 0$ *or*

  $\exists(n(v_0)\ \ell',\ \overline{b'},\ N') \in M(v_0)$ *such that* $(\lambda(v_0)\ b(\bar{v}_0)\ N(v_0)) \prec (\ell',\ \overline{b'},\ N')$

  then

    $n(v_0) := 1 +$ 最大值$\{n' \mid \exists(n',\ \ell',\ \overline{b'},\ N') \in M(v_0)\}$；

  $N(v_0) := N(v_0) \setminus \{(n',\ \ell',\ b', \overline{p'},\ q_1) \mid \exists(n',\ \ell',\ b', \overline{p'},\ q_1) \in N(v_0)\} \cup \{(n_1,\ \ell_1,\ \overline{b_1},\ p_1,\ q_1)\}$；

  $M(v_0) := M(v_0) \cup \{(n(v_0)\ \lambda(v_0)\ b(\bar{v}_0)\ N(v_0))\}$；

  如果 $M(v_0) \neq M_{old}$ **那么**

    对于 $i := 1$ **到** $\deg(v_0)$ **执行**

      发送 $<(n(v_0)\ \lambda(v_0)\ b(\bar{v}_0)\ M(v_0))\ i>$ 通过端口 i；

C：$\{M(v_0)$是相干的，且不存在$(n,\ \ell,\ b,\ \bar{N}) \in M(v_0)\}$

开始

  $b(v_0) := \bar{b}(v_0)rbit()$;

  对于 $i := 1$ **到** $\deg(v_0)$ **执行**

    发送 $<(n(v_0)\ \lambda(v_0)\ b(\bar{v}_0)\ M(v_0))\ i>$ 通过端口 i；

3.5 算法$\mathcal{M}$ 的一些性质

我们考虑在$(\mathbf{G}\ b, \delta)$上 $\mathcal{M}$ 的一次执行 ρ。对于每个顶点 $v \in V(G)$，我们用$(\lambda(v)\ n_i(v)\ b_i(v)\ N_i(v)\ M_i(v))$表示 ρ 在 v. 上执行第 i次计算步骤后，v的状态。如果顶点 v 从第 i 步到第 $i+1$步, 执行了一个动作，则称其在第 $i+1$. 步处于活跃状态。下面的命题总结了在$(\mathbf{G}\ b, \delta)$上执行 ρ期间满足的性质。

命题 *3.3*。考虑一个顶点 v 和一个步骤 i。那么，$n_i(v) \leq n_{i+1}(v)$,$b_i(v) \leq_B b_{i+1}(v)$, $N_i(v) \preceq N_{i+1}(v)$, *and* $M_i(v) \subseteq M_{i+1}(v)$。对于每个 $(m, \ell, b, N) \in M_i(v)$ 以及每个 $m' \in [1, m]$, $\exists(m', \ell', b', N') \in M_i(v)\ \exists v' \in V(G)$，使得 $n_i(v') = m'$。

证明。我们假设某个顶点 $v \in V(G)$ 在步骤 $i+1$ 执行了某个内部事件。对于任意顶点 $w \in V(G) \setminus \{v\}$，该性质显然成立，而且很容易看出 $M_i(v) \subseteq M_{i+1}(v)$。

如果 $n_i(v) \neq n_{i+1}(v)$，那么 $n_{i+1}(v) = 1 + \max\{n' \mid (n', \ell', b, \mathcal{N}') \in M_i(v)\}$，并且要么 $n_i(v) = 0 < n_{i+1}(v)$，要么 $(n_i(v)\ \lambda(v)\ N_i(v)) \in M_i(v)$，因此$n_i(v) < n_{i+1}(v)$。

随机比特序列通过添加其后缀而扩展，因此 $b_i(v) \leq_B b_{i+1}(v)$。

如果 $N_i(v) \neq N_{i+1}(v)$，那么 v已通过端口 q收到消息 $<(n', \cdots, M')\ p>$，并且 $N_{i+1}(v) = N_i(v) \setminus \{(n'_{old}, p, q)\} \cup \{(n', p, q)\}$，其中某个（先前的）数值为 n'_{old}。令 v' 为 v 的邻居，且满足 $\delta_v(v') = q$；我们知道 $\delta_{v'}(v) = p$。

如果 $(n'_{old}, p, q) \notin N_i(v)$，那么 $\max N_{i+1}(v) \triangle N_i(v) = (n', p, q) \in N_{i+1}(v)$，进而 $N_i(v) \prec N_{+1}(v)$。

如果 $(n'_{old}, p, q) \in N_i(v)$，那么 $n'_{old} \neq n'$。令 $j < i+1$为 v'发送消息 $<(n', \cdots, M')\ p>$时的计算步骤。我们知道 $n'_{old} \leq n' = n_j(v')$，因此 $\max N_{i+1}(v) \triangle N_i(v) = (n', p, q) \in N_{i+1}(v)$，并且 $N_i(v) \prec N_{+1}(v)$。注意，根据 $\prec$的定义，前一个不等式的推导独立于 b 标签。

对于命题的第二部分：首先注意到，$(m, \ell, \beta, \mathcal{N})$ 在某一步 $\bigcup_{v \in V(G)} M_i(v)$ 被加入，当且仅当存在一个顶点 $v' \in V(G)$，使得 $n_i(v') = m$, $\lambda(v') = \ell$, $\bar{b}(v') = \beta$ 和 $N_i(v') = \mathcal{N}$.

给定一个顶点 $v \in V(G)$、一个步骤 i 以及一个元素 $(m, \ell, \beta, \mathcal{N}) \in M_i(v)$，令 $m' \leq M$和 $U = \{(u, j) \in V(G) \times \mathbb{N} \mid j \leq i, n_j(u) = m'\}$和 $U' = \{(u, j) \in U \mid \forall (u', j') \in U、(\lambda(u')\ N_{j'}(u')) \prec (\lambda(u)\ N_j(u))$ 或 $(\lambda(u')\ N_{j'}(u')) = (\lambda(u)\ N_j(u))$, 并且 $j' \leq j\}$。由于 $(m, \ell, \beta, \mathcal{N}) \in M_i(v)$，$U$和 U'均非空，因此很容易看出，存在 i_0，使得对每个 $(u, j) \in U'$，$j = i_0$。

如果 $i_0 < i$，令 $(u, i_0) \in U'$；我们知道 $n_{i_0+1}(u) \neq n_{i_0}(u)$，但这是不可能的，因为根据 $(\lambda(u) b(u)\ N_{i_0}(u))$ 的极大性，u不可能修改其数值。因此，$i_0 = i$，并且存在 $v' \in V(G)$，使得 $n_i(v') = m'$。证明到此结束。

由 [6]，可知，一旦 $n(v)$ $N(v)$ 和 $M(v)$ 对所有 v 都达到最终值，那么对于任意 v，$S(M(v))$ 都是相干的。添加 b 时也是如此，因此，如果 $S(M(v))$ 不相干，我们知道 $M(v)$ 将被修改。

如果集合 $S(M)$ 是相干的，则可以如下构造一个带标签的对称有向图 $\mathbf{D}_M = (D_M, \lambda_M)$。顶点集 $V(D_M)$ 是集合 $\{n \mid \exists (n, \ell, N) \in S(M)\}$。对于任意 $(n, \ell, N) \in S(M)$ 和任意 $(n', \ell', p, q) \in N$，存在一条弧 $a_{n,n',p,q} \in A(D_M)$，使得 $t(a) = n, s(a) = n'$, $\lambda_M(a) = (p, q)$。由于 $S(M)$ 是相干的，我们可以通过 $Sym(a_{n,n',p,q}) = a_{n',n,q,p}$ 来定义 Sym。

可以证明，算法 $\mathcal{M}$ 以概率 1 终止。考虑两个源点不同的节点，由于源点彼此独立，随机比特有限序列最终会以概率 1 出现差异。因此，根据规则 $\mathbf{R}$，它们将获得不同的编号。

当 $\mathcal{M}$终止时，最终标记验证以下性质：有向图$(Dir(G),(\lambda, \delta))$是$\mathbf{D}_M$的一个对称覆盖（见[6]中的命题 4.1）。因此，若$(Dir(G), (\lambda, b, \delta))$是对称覆盖极小的，则$\mathbf{D}_M$同构于$(Dir(G), (\lambda, \delta))$，从而编号集合恰好为 $[1, |V(G)|]$:

每个顶点都有唯一的编号。此外，该算法的终止检测是可行的。事实上，一旦某个顶点获得身份编号 $|V(G)|$（每个顶点都知道该编号），根据命题 3.3，它便知道所有顶点都具有互不相同且不再改变的身份编号，从而可以断定计算已经结束。在这种情况下，也可以解决选举问题，因为该顶点可以取得标签已当选 并广播某个顶点已当选的信息。最后，我们得到上文所述的定理 3.2。

4 准覆盖与标记图族上的选举问题

4.1 准覆盖

在上一节中，我们假设已知网络的精确规模；本节则考虑一般的结构知识。本节介绍我们使用的第二种工具：准覆盖。这一工具为标记图族上的随机化选举提供了必要条件，适用于非共享随机源和共享随机源两种情况。

准覆盖最初是为研究终止检测问题而引入的 [17]。准覆盖背后的思想是：在较大图的一个受限区域内，模拟给定图上的局部计算，从而可以使用重放技术，通过反证法证明不可能性结果。随着模拟步数增加，我们能够进行模拟的受限区域会逐渐缩小，因此，当已知某个界时，便无法使用重放技术。在 [9], 中，准覆盖的定义经过了略微修改，以便更简洁地通过准提升引理表达这一性质。下一个定义是将这一工具改编到标记有向图上的版本，图 1 展示了其示意图。

定义 *4.1.* 给定两个对称标记有向图 D_0,D_1, 、一个整数 r、一个顶点 $v_1 \in V(D_1)$, 以及一个从 $B^{\mathbf{D}_1}(v_1, r)$ 到 D_0, 的同态 γ，如果存在一个对称标记有向图 D_2 ，它通过同态 φ 对 D_0 构成对称覆盖，并且存在 $v_2 \in V(D_2)$ 和一个从 $B^{\mathbf{D}_1}(v_1, r)$ 到 $B^{\mathbf{D}_2}(v_2, r)$ 的同构 ψ ，使得对于任意 $x \in V(B_{D_1}(v_1, r)) \cup A(B_{D_1}(v_1, r))$，$\gamma(x) = \varphi(\psi(x))$, 则有向图 D_1 是有向图 D_0 的一个以 v_1 为中心、半径为 r 、经由 γ 的准覆盖。

我们将层数 q 定义为由 γ 给出的原像集合的最小基数： $q = \text{最小值}_{v \in V(D_0)} |\{w \in \gamma^{-1}(v) | B_{\mathbf{D}_1}(w, 1) \subset B_{\mathbf{D}_1}(v_1, r)\}|$. 若一个准覆盖是真的，则称其为真拟覆盖；若 $B_{\mathbf{D}_1}(v_1, r-1)$ 不等于 D_1. 。任何非真准覆盖都是覆盖。

引理 4.2。 令$\mathbf{D}_1$ 为$\mathbf{D}_0$ 的真拟覆盖，其中心为 v_1 、半径为r ，通过 γ。那么，对于任意 $q \in \mathbb{N}$，若 $r \geq q|V(D_0)|$ ，则 γ 至少有 q 层。

证明。记相应覆盖为$\mathbf{D}_2$ ，通过 φ 给出。由于该准覆盖是适当的，我们有 $|B_{\mathbf{D}_1}(v_1, r)| \geq r \geq q|V(\mathbf{D}_0)|$, 因此 $|V(\mathbf{D}_2)| \geq q|V(\mathbf{D}_0)|$.

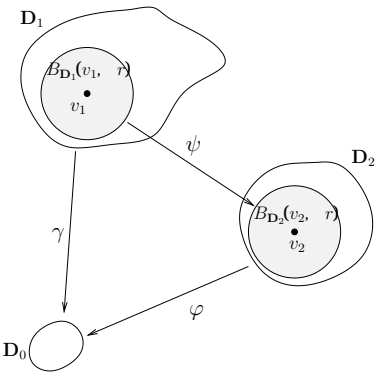

图 1 $B_{\mathbf{D}_1}$ v_1 r *"the existence of large*。准覆盖图。球(,)捕获 *enough area of one graph"* 1 *"that looks locally like another graph"* 0 **(D) (D)**。

我们可以推断，$\mathbf{D}_2$ 至少有 q 层。现在，考虑 $\mathbf{D}_0$ 的一棵生成树 T，其根为 $\gamma(v_1)$。注意，通过 φ 对 T 的提升以 $v_2 = \psi(v_1)$为根。根据 Reidemeister [19], 定理，存在 $q-1$ 棵互不相交的提升生成树 T_2, ..., T_q, 它们位于 $\mathbf{D}_2$ 上，使得由 $T_1 \cup \cdots \cup T_q$ 诱导的子图是连通的。由于 T 的直径至多为 $|V(\mathbf{D}_0)| - 1$，我们有 $T_1 \cup \cdots \cup T_q \subset B_{\mathbf{D}_2}(v_2, q|V(\mathbf{D}_0)|)$。这意味着 $\mathbf{D}_0$ 的每个顶点在 $B_{\mathbf{D}_2}(v_2, r)$ 中至少有 q 个原像，因此在 $B_{\mathbf{D}_1}(v_1, r)$ 中也至少有这么多个原像。

我们精确描述同步执行经过 k 轮次后半径的缩小。

引理 *4.3*（准提升引理）。设 D_1 是一个对称标记有向图，它是以 v_1 为中心、半径为 r 、经由 γ 对 D_0 的准覆盖。设 $k < r$ 是一个非负整数。对于任意算法 $\mathcal{A}$、任意源标记 b_0 以及满足以下条件的 b_1 : $(D1, b_1)$ 是一个对称标记有向图，它是以 v_1 为中心、半径为 r 、经由 γ 对 $(D0, b_0)$ 的准覆盖。令 D'_0 为在 $(D0, b_0)$ 上对 $\mathcal{A}$ 执行随机同步执行 k 轮后得到的有向图，其概率为 $p > 0$。那么，存在一个在 $(D1, b_1)$ 上对 $\mathcal{A}$ 执行随机同步执行后以概率 p 得到的 D'_1 ，它是以 v_1 为中心、半径为 $r - k$. 的 D'_0 的准覆盖

证明。考虑算法 $\mathcal{A}$ 和有向图 $\mathbf{D}_1 = (D_1, \lambda_1, b_1)$，它通过 γ 以 v_1 为中心、半径为 r 准覆盖 $\mathbf{D}_0 = (D_0, \lambda_0, b_0)$。存在一个对称标记有向图 $\mathbf{D}_2 = (D_2, \lambda_2, b_2)$，它通过同态 φ 对称覆盖 $\mathbf{D}_0$，且存在一个顶点 $v_2 \in V(D_2)$，使得 $(B_{D_1}(v, r)\ \lambda_1, b_1)$ 通过同构 ψ 与 $(B_{D_2}(v, r)\ \lambda_2, b_2)$ 同构，并且对于任意 $v \in B(v_1, r)$,$\gamma(v) = \varphi(\psi(v))$。

设 $\mathbf{D}'_0 = (D_0, \lambda'_0)$（分别为 $\mathbf{D}'_1 = (D_1, \lambda'_1), \mathbf{D}'_2 = (D_2, \lambda'_2)$）是带标签的有向图，其中每个 v, $\lambda'_0(v)$（分别为 $\lambda'_1(v)$、 $\lambda'_2(v)$）都是 v in D_0 （分别为D_1, D_2）在 $\mathbf{D}_0$ 上执行一个计算步骤后得到的状态，概率为 $p > 0$。要证明该引理，只需说明我们可以在 $\mathbf{D}_1$,$\mathbf{D}_2$ 上执行类似的步骤

概率同样为 p，其中 D'_1 是 D'_0 的一个准覆盖，其中心为 v_1 、半径为 $r-1$，通过 γ 得到。由引理 3.1 可知，我们可以以概率 p 得到 D'_2，它是 D'_0 的一个覆盖。此外，对于每个 $v \in V(B_{\mathbf{D}_1}(v_1, r-1))$,$\lambda'_1(v) = \lambda'_2(\varphi(v))$，因为 $(B_{\mathbf{D}_1}(v, 1)\ \lambda_1, b_1)$ 与 $(B_{\mathbf{D}_2}(\varphi(v), 1)\ \lambda_2, b_2)$ 同构。因此，$(B_{\mathbf{D}_1}(v, r-1)\ \lambda'_1, b_1)$ 通过 ψ 与 $(B_{\mathbf{D}_2}(v, r-1)\ \lambda'_2, b_2)$ 同构，所以 D'_1 是 D'_0 的一个准覆盖，其中心为 v_1 、半径为 $r-1$，通过 γ 得到。

这种配置的概率为 $Pr(X_{S_1} = x^1)$，其中 S_1 是 $B_{\mathbf{D}_1}(v_1, r-1)$ 中的源点集合，x^1 是在这些源点上抽取的比特。由复合$\phi \circ \psi$可知，$\mathbf{D}_0$ 中的源点集合同样为 S_1。因此该概率等于 p。

以下是提升引理关于拟覆盖的对应结论。

推论 4.4（随机化准提升推论）。设$\mathbf{D}_1$ 是经由 γ、$\mathbf{D}_0$ 的一个中心为 v_1 、半径为 r 的准覆盖。对任意算法 $\mathcal{A}$，在以概率 $p > 0$ 于 $\mathbf{D}_1$ 上对算法 $\mathcal{A}$ 进行r 轮次随机同步执行之后， v_1 与 $\gamma(v_1)$ 处于相同状态，而后者是在以概率 p 于$\mathbf{D}_0$ 上对 $\mathcal{A}$ 进行 r 轮次同步执行之后的状态。

4.2 标记图族中的拉斯维加斯与蒙特卡洛选举

利用随机化准提升引理，我们给出拉斯维加斯与蒙特卡洛选举算法的两个必要条件。

命题 4.5（拉斯维加斯必要条件）。设 $\mathcal{F}$ 是由连通的、对称覆盖极小的 $b-$ 标记有向图组成的递归族，并且该族存在拉斯维加斯选举算法。那么存在一个可计算函数 $\tau:\ \mathcal{I} \to \mathbb{N}$，使得对于 $\mathcal{F}$ 中任意标记有向图 $\mathbf{D}$，在 $\mathcal{F}$ 中都不存在半径大于 $\tau(\mathbf{D})$ 且不同于 $\mathbf{D}$ 的 $\mathbf{D}$ 的准覆盖。

证明。设 $\mathcal{A}$ 表示 $\mathcal{F}$ 上的一个选举算法。考虑一个标记有向图$\mathbf{D} \in \mathcal{F}$。由于这是一个拉斯维加斯算法，因此存在一次在 $\mathbf{D} \in \mathcal{F}$ 上正确的 $\mathcal{A}$ 终止同步执行。记 T 为这次在 $\mathbf{D}$ 上有限的、随机的成功同步执行的轮数。相应地，我们有一个序列 $\mathcal{C} = (\mathbf{C}_0 = \mathbf{D},\ \mathbf{C}_1,\ \ldots\ldots,\ \mathbf{C}_T)$，其中 $\mathbf{C}_i$ 是第 i 轮之后得到的标记图。在 $\mathcal{A}$ 的任何步骤中，都无法对 $\mathbf{C}_T$ 执行（在第 t 轮结束时，没有任何顶点发送消息）。这次执行关联着某个概率 $p > 0$。定义 $\tau(\mathbf{D}) = 2|V(\mathbf{D})| + T$。于是 τ 具有所需性质。

反证，设$\mathbf{D}' \in \mathcal{F}$是$\mathbf{D}$ 的一个半径为 $\tau(\mathbf{D})$，且不同于$\mathbf{D}$。的准覆盖。通过迭代引理 4.3，我们以某个非零概率得到$\mathbf{D}''$，使得$\mathbf{D}''$ 是$\mathbf{C}_T$的一个半径为 $\tau(D) - T = 2|V(D)|$的准覆盖。由于标记有向图$\mathbf{D}$ 是对称覆盖极小的，且不同于$\mathbf{D}'$，这个最终的准覆盖$\mathbf{D}''$ of$\mathbf{C}_n$ 是适当的。因此，根据构造，标签已当选 在$\mathbf{C}_T$.中恰好出现一次；由引理 4.2可知，标签已当选 在$\mathbf{D}''$中至少出现两次。这就产生了矛盾。

现在我们考虑蒙特卡洛算法。由于它们在某些执行中可能失败，前述技术并不适用。半径条件仅对适当的准覆盖是必要的。

命题 4.6（蒙特卡洛必要条件）。设 $\mathcal{F}$ 是一个由连通 b–标记有向图组成的递归族，且该族存在一个蒙特卡洛选举算法。那么存在一个可计算函数 $\tau\colon\ \mathcal{I}\to\mathbb{N}$，使得对于 $\mathcal{F}$中的任意标记有向图$\mathbf{D}$ ，在 $\mathcal{F}$中，$\mathbf{D}$ 都不存在半径大于 $\tau(\mathbf{D})$ 的真拟覆盖。

证明。设 $\mathcal{A}$ 表示 $\mathcal{F}$ 上的一个蒙特卡洛选举算法，其正确性为$\varepsilon>0$。我们考察算法 $\mathcal{A}$ 在$\mathbf{D}$上的同步调度。由于 $\mathcal{A}$ 是一个蒙特卡洛选举算法，存在一组随机比特序列 b_u、$u\in V(\mathbf{D})$，以及一个时刻 $T\in\mathbb{N}$，算法在该时刻正确结束。我们将这组序列 $b_u(1)\ \cdots,\ b_u(T)$、 $u\in V(\mathbf{D})$的概率记为$p>0$。

假设存在$\mathbf{K}_1$，它是$\mathbf{D}$ 的一个半径为 $T+2V(\mathbf{D})$的真拟覆盖。通过反复应用引理 4.3，我们可以提升该执行。我们将从$\mathbf{D}$到$\mathbf{K}_1$ 、在 T 轮次上拟提升得到的比特序列记为 b_1 。该序列具有某个概率 $p_1>0$。现在假设$\mathbf{K}$是$\mathbf{D}$ 的一个半径为$R=N(T+2V(\mathbf{D})$的真拟覆盖，其中 N满足$(1-p_1)^N<\frac{1}{2}\varepsilon$。这个真拟覆盖可以分解为 N 个不相交区域，它们构成一个半径为 $T+2V(\mathbf{D})$的准覆盖。对于给定的准覆盖，从$\mathbf{D}$ 和 b 提升到$\mathbf{K}_1$ 和 b_1 的概率至少为 p_1。发生这种提升时，已当选标签会在$\mathbf{K}_1$上出现两次。因此，在 N个不相交区域之一中出现错误输出的概率至少为$1-\frac{1}{2}\varepsilon$，这与 $\mathcal{A}$的正确性水平矛盾。因此，D不存在半径为 R 的真拟覆盖。令 $\tau(\mathbf{D})$为 R，即可完成证明。

证明定理 2.1 和 2.2 所需的部分内容放在附录中，因为它们是对 [5] 中算法的扩展。

5 应用

5.1 结构知识的代表性案例

我们总结这两种刻画所带来的结果，并讨论它们与文献中相关工作的关系。

我们考虑非共享随机性、共享随机性，以及有界共享随机性（见第 5.2 节）。共有三种情况：一般情况、 B–极小图和极小图。注意，$(\mathbf{G}\ b)$ 是 B–极小的，并不意味着 $\mathbf{G}$ 是极小的，但反过来成立。 B–极小图可以是至少有一个非共享随机源的匿名图、覆盖极小图，或共享源与覆盖结构所产生的对称性不一致的非覆盖极小图。与 [8],相比，我们指出， B–极小性将 [8]中针对团给出的条件推广了。例如，考虑环；如果 B 类的规模的最大公约数恰好为 1，那么该环就是 B–极小的。

	B–极小图		
知识	**None**	**共享界/界 2-近似/规模/拓扑**	
确定性	✗	✗	✗ （[1] 针对环）
拉斯维加斯	✗	✗ ([11] 用于环)	✔ ([11] 用于环)
蒙特卡洛	✗ ([11] 用于环)	✔ ([21] 用于环)	✔ ([7] 具有极高概率)

图 2. 总结了针对 B–极小图及不同知识条件下的选举可计算性结果，以及此前已知的结果。

当一个图不是 B–极小图时，即使已知其拓扑（包括随机源），它也不具备拉斯维加斯选举算法。当一个图是 B–极小图时，只要我们掌握足够的知识，它就具备拉斯维加斯选举算法。特别地，定理 2.1 的一个推论是，已知规模时，B–极小图具备拉斯维加斯选举算法。知道规模的一个上界确实可以提供一个（统一的）τ 函数，用于限制适当的拟覆盖，但无法限制拟覆盖，因为任意半径的覆盖都是拟覆盖。因此，若要实现拉斯维加斯选举，必须掌握足够的知识，以排除在同一知识条件下的覆盖。严格 2-近似知识就符合这种情况，因为任何严格覆盖的规模至少是原来的两倍。因此，虽然知道拓扑是最强的知识，但实际上并非必需，也不必确切知道规模。

一般情况下，蒙特卡洛选举算法又如何呢？根据定理 2.2，当一无所知时，不可能存在选举算法（因为这会使任意大半径的准覆盖成为可能）。不过，知道规模的一个界确实能限制真拟覆盖；其他界也是如此，例如第 5.2 节中的共享界，它能实现一个非一致的 τ 函数，因此是一个更一般的例子。最后，在考虑极小图时，以下几种刻画似乎是等价的：

定理 5.1. 设 $\mathcal{F}$ 是一族覆盖极小图，那么在 $\mathcal{F}$ 上可以用确定性算法解决选举，当且仅当在 $\mathcal{F}$ 上可以用拉斯维加斯算法解决选举，当且仅当在 $\mathcal{F}$上可以用蒙特卡洛算法解决选举。

我们还讨论了以往仅针对匿名环的研究。这表明，我们的一般性结果扩展了所有已知情形。这些结果汇总于图 2。

– [1] Angluin 的开创性工作首次证明了匿名算法中选举的不可能性。该工作基于局部模型，而局部模型比异步消息传递模型更强。– [10, 11]: Itai 和 Rodeh 证明了在已知规模上界的匿名环中，拉斯维加斯算法无法实现选举。在其教科书 [24, Th.9.12]，中，Tel 也为环给出了这一证明。Itai 和 Rodeh 在已知图规模时提出了一种概率选举算法。此外，他们还给出了一些精确的复杂度界。

– [21]:Schieber 和 Snir 提出了一种在已知规模上界时使用的蒙特卡罗算法。– [7]: Codenotti*et al* 提出了一种选举算法，该算法使用 1 个随机比特时以极高概率正确运行，并且要求已知网络规模。该结果可以轻松扩展到仅知规模上界的情况，但需要使用更多随机比特。

当拓扑已知时，概率算法是比确定性算法更好的解决方案。由于不可能永远提升执行，知道确切的规模或拓扑意味着可以等待Mazurkiewicz算法终止。目前尚无枚举这一点可以很容易地从知识中加以检验。

5.2 随机源的有界共享

我们现在考虑图族 $\mathcal{B}_K$ ，其中对每个图而言，共享源都是已知的（即标签与源点之间存在双射），并且与另一节点共享某一源的节点数目以某个数 $K \in \mathbb{N}$为上界。我们有一个蒙特卡洛选举算法，这实际上是引理 4.2的一个推论。由于在 $\mathcal{B}_K$中存在 B–个覆盖，对于这一知识不存在Las Vegas算法。我们强调，节点需要知道其随机源所属的类。否则，由于 $\mathcal{B}_K$中的网络规模无界，就会存在无界的真拟覆盖，因此不可能在 $\mathcal{B}_K$的无标签版本中解决选举问题。

命题 5.2. 令 $K \geq 1$，则存在一个针对$\mathcal{B}_K$的蒙特卡洛选举算法。

证明。 由引理 4.2 可知，对于$\mathbf{D} \in \mathcal{B}_K$，不存在半径为 $(K+1)|V(D)|$ 的真拟覆盖。因此，$\mathcal{B}_K$ 满足定理 2.2 的条件，其中 $\tau(\mathbf{D}) = (K+1)|V(D)|$。因此，对于任意 K，都可以在 $\mathcal{B}_K$ 中进行蒙特卡洛选举。

参考文献

1. D. Angluin. 处理器网络中的局部性质与全局性质。载于 *Proceedings of the 12th Symposium on Theory of Computing* 82 93, 1980 ，页码 – 。2. Alkida Balliu 、 Sebastian Brandt 、 Fabian Kuhn 、 Krzysztof Nowicki 、 Dennis Olivetti、Eva Rotenberg 和 Jukka Suomela。局部建议与局部解压缩。 *CoRR*, abs/2405.04519，2024。3. P. Boldi、B. Codenotti、P. Gemmell、S. Shammah、J. Simon 和 S. Vigna。匿名网络中的对称性破缺：刻画。载于 *Proc. 4th Israeli Symposium on Theory of Computing and Systems* 16 26 1996 ，页码 – 。IEEE 出版社，。4. P. Boldi 和 S. Vigna。图的纤维化。 *Discrete Math.*，243:21–66, 2002。5. J. Chalopin、E. Godard 和 Y. M´etivier。在消息传递系统中，具有任意知识的部分匿名网络上的选举。 *Distributed Comput.*，25 4 297 311，2012 ( )： –。6. J. Chalopin 和 Y. M´etivier。一种基于 Mazurkiewicz 算法的高效消息传递选举算法。 *Fundam. Inform.*，80(1-3):221–246，2007。[1] 极高概率

7. B. Codenotti、P. Gemmell、P. Pudlak 和 J. Simon。分布式计算所需的随机性量。载于 *OPODIS*，页码 237–248。Hermes，1997。8. P. Fraigniaud、R. Gelles 和 Z. Lotker。随机化对称性破缺分布式计算的拓扑。 *J. Appl. Comput. Topol.*，8(4):909–940，2024。9. E. Godard 和 Y. M´etivier。选举可行的图族的刻画（*ext. abstract*）。载于 M. Nielsen 和 U. Engberg 编，*Proc.of Foundations of Software Science and Computation Structures, FOSSACS'02*，计算机科学讲义第 2303 卷，页码 159–171。施普林格出版社，2002。10. Alon Itai 和 Michael Rodeh。分布式网络中的对称性破缺。载于 *IEEE 54th Annual Symposium on Foundations of Computer Science*，第 0 卷，页码 150–158。IEEE计算机学会，1981。11. Alon Itai 和 Michael Rodeh。分布式网络中的对称性破缺。 *Information and Computation* 88 1 60 87, 1990，( )：–。

12. Dariusz R. Kowalski 和 Miguel A. Mosteiro。匿名网络中领导者选举的时间与通信复杂度。载于 *ICDCS*，页码 449–460。IEEE，2021。13. G. LeLann。分布式系统：迈向一种形式化方法。载于 B. Gilchrist 编， *Information processing'77*，页码 155–160。North-Holland，1977。14. W. S. Massey。 *A basic course in algebraic topology*。施普林格出版社，1991。数学研究生教材。15. Yossi Matias 和 Yehuda Afek。匿名网络的简单高效选举算法。载于 *WDAG*，*Lecture Notes in Computer Science* 第 392 卷，页码 183–194。施普林格，1989。16. A. Mazurkiewicz。分布式枚举。 *Inf. Proc. Letters*，61:233–239，1997。17. Y. M´etivier、A. Muscholl 和 P.-A. Wacrenier。关于图中局部计算终止的局部检测。载于 D. Krizanc 和 P. Widmayer 编，*SIROCCO 97*，Proceedings in Informatics，页码 188–200。Carleton Scientific，1997。18. Yves M´etivier、John Michael Robson 和 Akka Zemmari。对完全分布式拆分与命名概率过程及其应用的分析。 *Theor.*，：–。19. *Comput. Sci.* 584 115 130, 2015
K. Reidemeister。*Einführung in die Kombinatorische Topologie*。Vieweg，布伦瑞克，1932。20. N. Santoro。 *Design and analysis of distributed algorithm*。Wiley，2007。21. B. Schieber 和 M. Snir。无名网络中的名称调用。 *Information and Computation* 113 1 80 101, 1994，( )：–。22. B. Szymanski、Y. Shy 和 N. Prywes。同步分布式终止。 *IEEE Transactions on software engineering* 11 10 1136 1140, 1985，SE- ( )：–。

23. A. Tanenbaum 和 M. van Steen。 *Distributed Systems - Principles and Paradigms*。Prentice Hall，2002。24. G. Tel。 *Introduction to distributed algorithms*。剑桥大学出版社，2000。25. M. Yamashita 和 T. Kameda。匿名网络上的计算：第一部分——可解情形的刻画。 *IEEE Transactions on parallel and distributed systems* 7 1 69 89, 1996，( )：–。

A 一类标记图的选举算法

在本节中，我们给出用于获得与前述必要条件命题 4.5 和命题 4.6 相对应的充分条件的算法，并最终给出引言中定理 2.1 和定理 2.2 所述的、允许选举算法的图族的完整刻画。

该算法结合了第 3.3 节中针对对称最小标记有向图提出的随机化选举算法 $\mathcal{M}$，以及对 Szymanski、Shy 和 Prywes 的算法（简称 SSP 算法）的推广 [22]。SSP 算法最初用于检测具有局部终止的算法的全局终止，前提是节点最初知道图直径的一个界。

实际上，算法 1 在任意网络（Dir(G) δ）上总会终止；在执行过程中，每个节点 v 都能在某个计算步骤i 重构出一个对称标记有向图 $\mathbf{D}_i(v)$，使得 (Dir(G) δ) 是$\mathbf{D}_i(v)$ 的准覆盖。然而，该算法无法 v 计算此准覆盖的半径。我们使用 SSP 算法的推广版本，使每个节点都能计算当前这些准覆盖半径的下界。

A.1 终止检测机制

顶点将利用命题 4.6 给出的必要条件中的界来检测其状态是否为最终状态：它会检查计算在大于准覆盖半径界的半径范围内是否稳定，因此，我们为每个顶点的标签添加两项：

– $c(v) \in \mathbb{Z}$ 是一个计数器，初始值为 $c(v) = -1$。在某种意义上，$c(v)$ 表示所有顶点与 v 具有相同邮箱的距离范围（随机比特序列允许有后缀差异）。– $A(v) \in \mathcal{P}_{\text{fin}}(\mathbb{N} \times \mathbb{N})$ 编码了 v 所掌握的关于每个邻居 u 的 $c(u)$ 值的信息。初始时，$A(v) = \{(q, -1) \mid q \in [1, \deg_G(v)]\}$。

因此，在执行过程中，每个v 的标签都是一个元组（$\lambda(v)$ $bn(v)$ $N(v)$ $M(v)$ $c(v)$ $A(v)$）。

顶点 u通过端口 p 向顶点 v 发送的消息形式如下：$<(n, \ell, b, M, a)$ $p>$,– 其中 n 是当前的 $n(u)$数量，u是标签$\lambda(u)$，b 是当前的随机比特序列，M 是 u的邮箱，a 是计数器 $c(u)$和 $p = \delta_u(v)$的值。

A.2 算法$\mathcal{M}_\tau$

我们考虑一个具有命题 4.5 或命题 4.6 所述性质的图族。因此，我们考虑这样的函数 τ，并在算法 2 中描述算法 $\mathcal{M}_\tau$。回顾一下，节点 v 仅当尚未收到任何消息时，才能在唤醒时应用第一条规则 **I**：它取数值 1，更新自己的邮箱，并通知其邻居。第二条规则 **R** 描述了节点 v 从邻居收到消息 m 时必须遵循的指令。它根据 m更新自己的邮箱 $M(v)$和局部视图 $N(v)$。然后，如果它发现存在另一个具有相同数值但局部视图更强的顶点，就取一个大于所有已知数值的新数值。如果其邮箱没有变化，它就更新 $A(v)$，并可能通过查看邻居的计数器来增加 $c(v)$。最后，如果 $M(v)$ 或 $c(v)$ 已被修改，它就通知其邻居。

利用邮箱中存储的信息，每个节点 v 都能够重构一个标记有向图 $\mathbf{D}$，使得 (Dir(G) δ) 在距离 $c(v)$ 范围内的局部视图与 $\mathbf{D}$ 相同。

算法的性质 $\mathcal{M}_\tau$. 我们考虑一个带有端口编号 δ 的图$\mathbf{G}$，以及算法 $\mathcal{M}_\tau$ 在（$\mathbf{G}$ δ）上的一次执行 ρ。对于每个顶点 $v \in V(G)$，我们将经过 i步计算后 ρ的状态记为（$\lambda_i(v)$ $b_i(v)$ $n_i(v)$ $N_i(v)$ $M_i(v)$ $c_i(v)$ $A_i(v)$）。

命题 3.3 的一个有趣推论是：存在一步 i_0，使得在这一步之后，对于任意 v，$\lambda(v)$ $n(v)$ $N(v)$和 $M(v)$的值都不再改变。

命题 A.1。考虑一个顶点 v 和一步 i。

若 $M_i(v) = M_{i+1}(v)$，且 v 在第 $i+1$步处于活跃状态，则 $c_i(v) \leq c_{i+1}(v) \leq c_i(v) + 1$，并且 $c_{i+1}(v) \geq min\{a \mid \exists (q, a) \in A_{i+1}(v)\}$，若 $\exists (q, a) \in A_{i+1}(v)$。

若 $c_{i+1}(v) \geq 1$，则对于每个 $w \in N_G(v)$，都存在一个步骤 $j \leq i$ 使得$c_j(w) \geq c_{i+1}(v) - 1$ 且 $M_j(w) = M_{i+1}(v)$。

尽管证明使用了与 [5], 命题 3.3 的证明相同的状态递增论证，但它并不是对 [5] 的直接推广。随机比特序列仅以渐进的方式揭示，因此必须定义一种新的标记顺序，使其与 Mazurkiewicz 算法的动态交互不会使基础算法偏离正轨。关键在于认识到（并仔细验证），使用二进制序列的字母顺序时，节点过去的标签总是弱于其当前标记。命题 A.1 和命题 A.5 的证明是对 [5] 中证明的相当直接的推广，因为广义 SSP 算法本身并非随机化算法，所以在考察 $\lambda(v)$ 以检测变化并重置计数器 $c(v)$ 时，只需在 $\lambda(v)$ $b(v)$ 上进行记录。

–

证明。我们通过对 i. 使用归纳法来证明该命题。考虑在步骤 $i+1$.修改其状态的顶点 v

初始时，如果 v 已应用规则$\mathbf{I}$，则容易看出这些断言成立。

如果 v 已应用规则$\mathbf{C}$，则该断言由归纳法成立。

现假设 v 在经由端口 q_1接收消息 $m_1 =< (n_1, l_1, b_1, M_1, a_1)\ p_1 >$之后应用了规则$\mathbf{R}$。根据该算法，我们有：$M_i(v) \subseteq M_{i+1}(v)$。

如果 $M_i(v) = M_{i+1}(v)$且 $c_i(v) \neq c_{i+1}(v)$，则 $c_{i+1}(v) = c_i(v) + 1$。

设 $min_A = min\{a | \exists (q, a) \in A_{i+1}(v)\}$.

若 $min_A \neq a_1$，则由归纳法 $c_{i+1} \geq min_A$. 若 $M_{i+1}(v) \neq M_i(v)$则$c_{i+1}(v) \geq -1 \geq min_A$. 假设 $M_{i+1}(v) = M_i(v)$. 若$(q_1, a_1) \in A_i(v)$则$A_{i+1}(v) = A_i(v)$并由归纳法 $c_{i+1}(v) \geq min_A$. 若 $M_1 \neq M_{i+1}(v)$则$A_{i+1}(v) = A_i(v)$且 $c_{i+1}(v) \geq min_A$.

现假设 $M_1 = M_{i+1}(v)$. 若 $a_1 = 0$ 则 v 会增大 $c(v)$，如果它等于 -1，因为与 QC相关的不等式平凡地成立。因此 $c_{i+1}(v) \geq 0$ 从而 $c_{i+1}(v) \geq min_A$.

否则，我们可以假设 $min_A = a_1 > 0$，$M_{i+1}(v) = M_i(v) = M_1$ 以及 $(q_1, a_1) \notin A_i(v)$。令 $m_2 =< (n_2, \ell_2, \beta_2, M_2, a_2)\ p_1 >$为前一条消息

通过端口 q_1.接收。由于通信信道采用先进先出， m_2 先于 m_1发送。由于 $a_1 > 0$, $M_2 = M_1$ ，因此根据归纳法， $a_2 = a_1 - 1$. 。令$j \leq i$ 为 v 收到 m_2. 的步骤。由于 $M_2 \subseteq M_j(v) \subseteq M_i(v) = M_1 = M_2$, $M_j(v) = M_2$. 。因此，$(q, a_2) \in A_i(v)$。由于 $a_1 = min_A$， $a_2 = a_1 - 1 = min\{a | \exists (q, a) \in A_i(v)\}$。如果 $c_i(v) \geq a_1$ ，那么 $c_{i+1}(v) \geq c_i(v) \geq min_A$. 。否则，根据归纳法， $c_i(v) = a_2 = a_1 - 1$; 并且，由于 $c_i(v) < a_1$ 且 $c_i(v) \leq min_A$， v增加了 $c_i(v)$。因此， $c_{i+1}(v) = 1 + c_i(v) = a_1 \geq min_A$.

假设 $c_{i+1}(v) \geq 1$ ，并考虑一个顶点 $w \in N_G(v)$。根据以上内容，存在 $(\delta_v(w), a) \in A_{i+1}(v)$，满足 $a \geq c_{i+1}(v) - 1 \geq 0$ ，且 v 在步骤 $i' \leq i + 1$. 从 w 收到一条消息 $m = <(n, \ell, \beta, M, a), \delta_w(v)>$。令 $j < i + 1$为 w 发送该消息的步骤。由于 $a \geq 0$, $M_{i'}(v) = M_{i+1}(v) = M_j(v)$ 且 $c_j(w) = a \geq c_{i+1}(v) - 1$.

A.3 准覆盖计算

我们首先证明一个命题，它使我们能够给出准覆盖的另一种定义，并证明节点可以计算半径不断增大的准覆盖。证明此命题需要用到视图的定义，现给出如下。

定义 A.2. 考虑一个对称有向图 $\mathbf{D} = (D, \lambda, b) \in \mathcal{D}_{L \times B}$ 和一个顶点 $v \in V(D)$。v 在$\mathbf{D}$ 中的视图 是一棵无限根树，记作$\mathbf{T_D}(v) = (T_D(v), \lambda', b')$ ，定义如下：

– $V(T_D(v))$ 是 D 中非停顿路径 $\pi = a_1, \ldots, a_p$ 的集合，其中 $s(a_1) = v$。对于每条路径 $\pi = a_1, \ldots, a_p$, $\lambda'(\pi) = \lambda(t(a_p))$，且 $b'(\pi) = b(t(a_p))$。– 对每个 π, $\pi' \in V(T_D(v))$，存在两条弧 $c_{\pi,\pi'}, c_{\pi',\pi} \in A(T_D(v))$，使得 $Sym(c_{\pi,\pi'}) = c_{\pi',\pi}$ 当且仅当 $\pi' = \pi, a$。在这种情况下， $\lambda'(c_{\pi,\pi'}) = \lambda(a)$，且 $\lambda'(c_{\pi',\pi}) = \lambda(Sym(a))$。$b'$ 和 b 也类似。– $T_D(v)$ 的根是与空路径对应的顶点，其标签为 $(\lambda(v), b(v))$。

注释 *A.3.* 对于所有顶点， u 和 v： $T_D(u)$ 与 $T_D(v)$ 同构。我们用 T_D 表示这个在同构意义下定义的图。它是 D. 的泛覆盖。给出准覆盖的示例很有用。

考虑有向图 $\mathbf{D} \in \mathcal{D}_{L \times B}$ 中顶点 v 的视图 $\mathbf{T_D}(v)$，以及满足 $s(a) = v$ 的弧 a。我们将 $\mathbf{T_{D-}}_a(v)$ 定义为由 $\mathbf{T_D}(v)$ 去掉以对应于路径 a 的顶点为根的子树后得到的无限树。

命题 A.4. 给定两个对称标记有向图 $\mathbf{D}_0$,$\mathbf{D}_1$, 、一个整数 r、一个顶点 $v_1 \in V(D_1)$ ，以及一个从 $\mathbf{B^{D_1}}(v_1, r)$ 到 $\mathbf{D}_0$, 的同态 γ ，则 $\mathbf{D}_1$ 是$\mathbf{D}_0$ 的准覆盖 ，其中心为 v_1 ，半径为 r ，且经由 γ ，当且仅当以下条件成立：

(i) 对于每条弧 $a \in A(B_{D_1}(v_1, r))$， $\gamma(Sym(a)) = Sym(\gamma(a))$; *(ii)* 对于任意 $v \in \mathbf{B^{D_1}}(v_1, r)$， γ 在 v 的入弧（分别为出弧）与 $\gamma(v)$的入弧（分别为出弧）之间诱导一个单射，

(iii) 对于任意 $v \in \mathbf{B}^{\mathbf{D}_1}(v_1, r-1)$， γ 在 v 的入弧（分别为出弧）与 $\gamma(v)$的入弧（分别为出弧）之间诱导一个满射。

证明。 如果$\mathbf{D}_1$ 是$\mathbf{D}_0$ 的一个准覆盖，其中心为 v_1 、半径为 r ，且通过 γ，那么很容易看出， γ 满足这些性质。

反过来，我们如下构造 $\mathrm{D}_2 = (D_2, \lambda_2, b_2)$ 的一个 $\mathrm{D}_0 = (D_0, \lambda_0, b_0)$ 的无限覆盖。首先，取 $\mathrm{B}^{\mathbf{D}_1}(v_1, r)$ 的一个副本 B_1，并记 δ为从 $\mathrm{B}^{\mathbf{D}_1}(v_1, r)$ 到 B_1 的同构。然后，考虑一个满足 $\mathrm{dist}_{B_1}(v_1, v) = r$ 的顶点 v ，以及一条满足 $v \in \gamma^{-1}(t(a_0))$ 的弧 $a_0 \in A(D_0)$。如果不存在满足 $t(a) = v$ 且 $\gamma(a) = a_0$ 的弧 $a \in A(B_1)$，那么我们将 $\mathrm{T}^{\mathbf{D}_0 - a_0}(s(a_0))$ 的一个副本添加到 D_2 中，并记 $v_2(a_0)$ 为这棵树的根。我们添加两条弧a_2, a_2' ，使得 $t(a_2) = s(a_2') = v$、 $s(a_2) = t(a_2') = v_2(a_0)$、 $\lambda_2(a_1) = \lambda_0(a_0)$、$\lambda_2(a_2) = \lambda_0(Sym(a_0))$ 以及 $Sym(a_2) = a_2'$。对于 b' 和 b，同样如此。

有向图 $\mathbf{D}_2$ 是在针对所有满足 $\mathrm{dist}_{B_1}(v, v_1) = r$ 的 $v \in V(B_1)$ 完成这些构造后得到的有向图。容易看出，$\mathbf{D}_2$通过某个同态 φ 是 $\mathbf{D}_0$ 的一个对称覆盖，并且对于任意$v \in V(B_{\mathbf{D}_1}(v, r))$， $\gamma(v) = \varphi(\delta(v))$。

下面的命题表明，$\mathbf{D}^{M(v)}$ 与 $(\mathrm{Dir}(\mathbf{G}), \delta)$ 有某种精确的相似性。

命题 A.5。 如果 $S(M(v))$ 是相干的，那么$(\mathrm{Dir}(\mathbf{G}), \delta)$ 是$\mathbf{D}^{M(v)}$ 的一个半径为 $c(v)$ 、中心为 v 的准覆盖。

算法 $\mathcal{M}_\tau$ 是第 3.3 节中介绍的算法 $\mathcal{M}$ 与 Szymanski、Shy 和 Prywes 算法的一种推广的组合。试探性枚举算法以严格大于零的概率在任意网络 $(\mathrm{Dir}(\mathbf{G}), \delta)$. 上终止。在执行过程中，每个节点 v 都会在步骤 i 重构一个对称标记有向图 $\mathbf{D}_i(v)$，使得 $(\mathrm{Dir}(\mathbf{G}), \delta)$ 是$\mathbf{D}_i(v)$ 的一个准覆盖. 枚举算法终止后，对于每个顶点 v, ，在顶点 v 的步骤 t_v 之后，都存在一个步骤 t_v' ，使得 $c(v) \geq \tau(\mathbf{D}^{M(v)})$. 这些知识以及关于重构图$\mathbf{D}^{M(v)}$,的知识意味着，每个节点都知道枚举算法已经终止，并且知道自己的编号是否是图中出现的最大编号：每个节点都能判断自己是否已当选。

证明。 考虑一个步骤 i 和一个节点 v ，使得 $S(M_i(v))$ 是相干的。若 $c_i(v) = 0$，则证毕。现在假设 $c_i(v) \geq 1$。根据命题 A.1，对于每个满足 $\mathrm{dist}^{Dir(G)}(v, w) \leq c_i(v)$ 的$w \in V(Dir(G))$，都存在一个步骤 $j_w \leq i$，使得 $c_{j_w}(w) \geq c_i(v) - \mathrm{dist}^{Dir(G)}(v, w)$，且 $M_{j_w}(w) = M_i(v)$。

因此，对于每个 $w \in V(B_{Dir(\mathbf{G})}(v, c_i(v)))$， j_w 有定义，且 $c_{j_w}(w) \geq 0$；对于每个 $w \in V(B_{Dir(\mathbf{G})}(v, c_i(v) - 1))$， $c_{j_w}(w) \geq 1$。对于每个 $w \in V(B_{Dir(\mathbf{G})}(v, c_i(v)))$，$(n_{j_w}(w), \lambda(v), b(v), N_{j_w}(w)) \in S(M_{j_w}(w)) = S(M_i(v))$，因此 $\deg_G(w) = \deg^{D_{M_i(v)}}(n_{j_w}(w))$。因此，我们可以定义 $\gamma(w) = n_{j_w}(w) \in V(D_{M_i}(v))$，并且有 $\lambda_{M_i}(v)(\gamma(w)) = \lambda(w)$。

对于每条弧 $a \in A(B_{Dir(\mathbf{G})}(v, c_i(v)))$，令 $w = t(a)$ $w' = s(a)$（相应地， $w = s(a)$ $w' = t(a)$），并不失一般性地假设 $\mathrm{dist}_G(v, w) \leq c_i(v) -$

1. 令 $m = <(n, \ell, b, \bar{M}, a)\ p >$为 w在步骤 j_w之前通过端口 $\delta_w(w')$接收到的最后一条消息。由于 $c_{j_w}(w) \geq 1$, $M = M_{j_w}(w) = M_{j_{w'}}(w')$, $n = n_{j_{w'}}(w')$, $p = \delta_{w'}(w)$ 以及 $a \geq 0$。因此，我们可以定义 $\gamma(a) = c_{n,\ n',\ p,\ q}$（相应地，$\gamma(a) = c_{n',\ n,\ q,\ p}$），其中 $n = n_{j_w}(w)$, $n' = n_{j_{w'}}(w')$, $p = \delta_{w'}(w)$ 以及 $q = \delta_w(w')$。容易看出 $\lambda_{M_i(v)}(\gamma(a)) = \lambda(a) = (p, q)$（相应地，$\lambda_{M_i(v)}(\gamma(a)) = \lambda(a) = (q, p)$），并且 $Sym(\gamma(a)) = \gamma(Sym(a))$。从而，$\gamma$ 是从 $\mathbf{B}^{Dir(\mathbf{G})}(v, c_i(v))$ 到 $\mathbf{D}_{M_i}(v)$ 的同态。

对于每个 $w \in V(B_{Dir(\mathbf{G})}(v, c_i(v)))$，对于所有满足 $s(a) = s(a') = w$（分别为 $t(a) = t(a') = w$）的弧 a, a', $\lambda(a) \neq \lambda(a')$ 或 $b(a) \neq b(a')$，因此，$\gamma(a) \neq \gamma(a')$。对于每个 $w \in V(B_{Dir(\mathbf{G})}(v, c_i(v) - 1))$，由于 $\deg_G(w) = |N_{j_w}(w)| = \deg^{D_{M_i(v)}}(n_{j_w}(w))$, γ 在 $Dir(\mathbf{G})$ 中的 w 的入弧（分别为出弧）与 $\mathbf{D}_{M_i}(v)$ 中 w 的入弧（分别为出弧）之间诱导出一个双射。根据命题 A.4, $Dir(\mathbf{G})$ 是 $\mathbf{D}_{M_i}(v)$ 的一个准覆盖，其中心为 v，半径为 $c_i(v)$，通过 γ 实现。

注 *A.6*。正如证明中所示，$\gamma(w)$ 的值并不取决于实际的 j_w。准覆盖 γ 是从 n 得到的，即 $\gamma(w)$ 在 $w \in B_{Dir(G)}(v,\ c(v))$ 处的值，等于 $n(w)$ 在 $M(w) = M(v)$ 所对应的时间步的值。

算法 2: 算法 $\mathcal{M}_\tau$. $QC(v)$ is a 捷径 for $(S(M(v))$ 是相干的) 且 $(c(v) \neq c_{old}(v))$ 且 $(\mathbf{D}_{M(v)} \in \mathcal{F})$ 且 $(c(v) \leq \tau(\mathbf{D}^{M(v)}))$。

I: $\{n(v_0) = 0$ 且没有消息到达 $v_0\}$

开始

- $n(v_0) := 1$;
- $c(v_0) := -1$;
- $\bar{b}(v_0) := \bar{b}(v_0)rbit()$;
- $M(v_0) := \{(n(v_0)\ \lambda(v_0)\ b(\bar{v}_0)\ \emptyset)\}$;
- **对于** $i: = 1$ **到** $\deg(v_0)$ **执行**
  - **发送** $<(n(v_0)\ \lambda(v_0)\ b(\bar{v}_0)\ M(v_0))\ i >$ 通过 i ;

R: $\{$一条消息 $M_1 =< (n_1,\ \ell_1,\ \overline{b_1},\ M_1,\ a_1)\ p_1 >$ 已到达 v_0 通过端口 $q_1\}$

开始

- $M_{old} : = M(v_0)$;
- $c_{old} : = c(v_0)$;
- $M(v_0): = M(v_0) \cup M_1$;
- **if** $n(v_0) = 0$ *or* $\exists(n(v_0)\ \ell',\ \overline{b'},\ N') \in M(v_0)$ *such that* $(\lambda(v_0)\ \bar{b}(v_0)\ N(v_0)) \prec (\ell',\ \overline{b'},\ N')$ **然后**
  - $n(v_0): = 1 +$ 最大值$\{n' \mid \exists(n',\ \ell',\ N') \in M(v_0)\}$;
  - $N(v_0): = N(v_0) \setminus \{(n',\ \ell',\ b',\ \overline{p'},\ q_1) \mid \exists(n',\ \ell',\ b', \overline{p'},\ q_1) \in N(v_0)\} \cup \{(n_1,\ \ell_1,\ \overline{b_1},\ p_1,\ q_1)\}$;
  - $M(v_0) := M(v_0) \cup \{(n(v_0)\ \lambda(v_0)\ b(\bar{v}_0)\ N(v_0))\}$;
- **如果** $M(v_0) \neq M_{old}$ **则**
  - $c(v_0): = -1$;
  - $A(v_0) : = \{(q',\ -1) \mid 1 \leq q' \leq deg(v_0)\}$;
- **如果** $M(v_0) = M_1$ **则**
  - $A(v_0) : = A(v_0) \setminus \{(q_1, a') \mid \exists(q_1, a') \in A(v_0)\} \cup \{(q_1, a_1)\}$;
- **如果** $(\forall(q', a') \in A(v_0)\ c(v_0) \leq a'\ and\ \mathbf{QC}(v_0))$ **则** $c(v_0) := c(v_0) + 1$;
- ;
- **如果** $M(v_0) \neq M_{old}\ or\ c(v_0) \neq c_{old}$ **则**
  - **对于** $i := 1$ **到** $\deg(v_0)$ **执行**
    - **发送** $<(n(v_0)\ \lambda(v_0)\ b(\bar{v}_0)\ M(v_0)\ c(v_0))\ i >$ 通过端口 i;

C: $\{c(v) < \tau(\mathbf{D}^{M(v)}).\}$

开始

- $b(v_0) := \bar{b}(v_0)rbit()$;
- **对于** $i := 1$ **到** $\deg(v_0)$ **执行**
  - **发送** $<(n(v_0)\ \lambda(v_0)\ b(\bar{v}_0)\ M(v_0))\ i >$ 通过端口 i;