arXiv:2302.02567v1 [cs.DS] 6 Feb 2023

一般图中的随机最小顶点覆盖： $3/2$-近似

Mahsa Derakhshan* Naveen Durvasula † Nika Haghtalab †

2023年2月7日

摘要

我们研究随机顶点覆盖问题。在该问题中， $G = (V,\ E)$ 是任意已知图， $\mathcal{G}^\star$ 是 G 的一个未知随机子图，其中每条边 e 都以概率 p 独立实现。 $\mathcal{G}^\star$ 中的边只能通过边查询来验证。该问题的目标是使用少量查询，求出 $\mathcal{G}^\star$ 的最小顶点覆盖。

我们的主要成果是设计了一种算法，该算法仅使用 $O(n/\varepsilon p)$ 次非自适应查询，就能返回一个顶点覆盖，其大小至多为最小顶点覆盖期望大小的 $(3/2+\varepsilon)$ 倍。这改进了 Behnezhad、Blum 和 Derakhshan 在 [SODA'22] 中提出的已知最佳 2-近似算法；他们还证明，要实现任何常数近似，都至少需要 $\Omega(n/p)$ 次查询。

我们的保证也适用于边实现并非完全独立的实例。作为这一上界的补充，我们为边实现呈现轻微相关性的随机图给出了一个紧的 $3/2$近似下界。

*东北大学
†加州大学伯克利分校

1 引言

在随机顶点覆盖问题中，给定一个任意的基图 $G = (V, E)$，它有 n个顶点，但我们不知道 E 中的哪些边实际存在。相反，每条边 $e \in E$ 都会以给定的存在概率 $p_e \in$ 独立地实现（$0, 1]$, ，从而形成一个子图 $\mathcal{G}^\star$。我们的目标是找到 $\mathcal{G}^\star$的一个最小顶点覆盖。虽然 $\mathcal{G}^\star$ 未知，但可以通过查询边 $e \in E$来验证其边集。因此，值得关注的是：能查询一小部分边，并根据查询结果找到 $\mathcal{G}^\star$ 的一个近似最优顶点覆盖的算法。查询的边集应该有多小？这类问题的黄金标准是非自适应地发出若干次查询，其数量与顶点数呈线性关系，并且关于概率倒数 $p = \min_{e \in E} p_e$呈多项式关系。

尽管这些随机设置主要关注信息论问题，但大多数积极成果都聚焦于这样一类问题：其非随机对应问题存在计算高效的算法。相关实例包括最小生成树 [GV04, GV06], 所有顶点对最短路径 [Von07], 最大匹配 [BDH+15, BDH+20b, AKL16, AKL17, BR18, YM18, BDF+19, AB19, BFHR19, BDH20a, BD20], 2-近似最小顶点覆盖，以及二分图最小顶点覆盖 [BBD22]。这种对可高效求解问题的侧重并非偶然。总体而言，长期以来在理解和设计计算高效算法方面发挥关键作用的结构简单性质和启发式方法，已被用于指导算法选择查询，例如 Tutte-Berge 见证集 [AKL16], 短增广路径 [BDH+15]、局部计算 [BDH20a], 以及贪心启发式 [BBD22]。

另一方面，对于超越 2-近似的顶点覆盖问题，以及其他计算困难情形，缺乏结构简单的性质一直是求解这些问题随机变体的障碍。一个自然的问题是：对于缺乏此类结构的问题，我们能否取得任何积极成果？在这项工作中，我们针对最小顶点覆盖问题探讨了这一问题。

问题 1. 尽管缺乏针对该问题的计算高效算法，我们能否为随机最小顶点覆盖问题实现优于 2-近似的结果？

我们的论文对此问题给出了肯定的回答。概括来说，我们提出了一种算法，它以概率 1 返回一个 $\mathcal{G}^\star$ 的顶点覆盖，其期望大小至多为 $3/2 + \varepsilon$ 乘以 $\mathcal{G}^\star$ 的最小顶点覆盖的大小，其中 ε 可以任意小。以下定理在第 5 节中作了正式表述，展示了我们的主要结果。

定理 1.1 (上界)。 对于任意 $\varepsilon \in (0,0.1)$，存在一种算法（即算法 3），使用 $O(n/\varepsilon p)$ 次查询，为随机最小顶点覆盖问题返回一个 $(3/2 + \varepsilon)$-近似解。

该算法使用的查询数量在渐近意义上是最优的，因为 [BBD22] 证明了，要达到任何常数近似比，都必须进行 Ω(n/p) 次查询。有趣的是，即使 1.1 定理 $\mathcal{G}^\star$ 中的边具有相关性（见第7节），该定理的近似保证仍然成立。这使我们能够处理 的实现中的弱相关性 $\mathcal{G}^\star$。也就是说，即使允许以相关的方式实现 $O(n)$ 条边[1], ，我们仍然只需进行 $3/2 + \varepsilon$ 次查询，就能得到一个 ($O(n/\varepsilon p)$-近似解。我们的下一个结果表明，对于这类具有弱相关性的过程，使用 $3/2$ 次查询时，所能期望达到的最佳结果是 -近似。以下定理在第$(n/\varepsilon p$节作了正式陈述，并给出了我们的下界。7。

定理 1.2(在 正式下界）。 存在一种随机过程，用于生成 $\mathcal{G}^\star$ ，其具有 O（n)

[1]关于具有 $O(n)$ 条相关边的随机过程的正式定义，请参见定义 7.1

相关边，使得任何仅使用 $O(n/\varepsilon p)$ 次查询来返回 $\mathcal{G}^\star$ 的顶点覆盖的算法，其近似比至少为 $3/2-\varepsilon$），且该结果以概率 $1-o(1)$ 成立。

定理 1.1和 1.2共同表明，若 $\mathcal{G}^\star$的边之间存在弱相关性，我们的结果就是紧的；而在实际应用中，这种相关性很容易出现。此外，这表明若要进一步突破 $3/2$近似，就必须充分利用所有边之间的独立性。事实上， [AKL16] 和 [AB19]也对随机匹配问题给出了类似的刻画。

随机优化中的轻度相关性与独立性。在若干随机组合优化问题中，已有研究考虑了相关实现 [ADSY12, AP13, AKL16, AB19, BDH+20b, GKMR11]。对于随机匹配问题， [AKL16]最早研究了轻度相关图；除给出算法结果外，他们还构造了一种随机图，其中仅有 $O(n)$ 条相关边，而其近似比无法优于 $2/3$。我们的下界也采用了类似构造，其中仅有 $O(n)$ 条相关边。随后， [AB19] 利用 [BS15] 提出的一种巧妙的匹配稀疏化器，设计出一个 $2/3$近似算法，从而弥合了轻度相关图情形下的差距。对于随机匹配问题，轻度相关图情形下可能达到何种性能的紧致刻画，也为 [BDH20a]在完全独立情形下获得 $(1-\varepsilon)$ 近似比铺平了道路。为了充分利用所有边之间的独立性， [BDH20a] 关键性地采用了在 LOCAL 模型计算中设计的 $(1-\varepsilon)$ 近似匹配算法。随后，这一进展为二分图上的 $(1+\varepsilon)$–近似随机顶点覆盖算法奠定了基础 [BBD22]。然而，在 LOCAL 模型中，最小顶点覆盖尚无优于 2 的近似算法；这或许会成为突破完全独立图上随机顶点覆盖 $3/2$ 障碍的阻碍。

算法设计概述。为了达到 $3/2$-近似，我们首先介绍两种求解随机最小顶点覆盖问题的方法，它们都能实现 2-近似。我们的最终算法是仔细结合这两种方法的见解所得出的结果。

本研究中的所有算法都遵循相同的蓝图：我们考虑一个集合 $P \subseteq V$ 和其诱导子图 $H = G[V \setminus P]$。然后，我们查询 H的所有边，以实现 $\mathcal{H}^\star$，并取其顶点覆盖 M。接着，我们的算法返回 $S = P \cup M$。注意， S是 $\mathcal{G}^\star$的顶点覆盖，因为 $\mathcal{G}^\star$ 中任何未被 P覆盖的边，都会被 M覆盖。我们的算法及其保证仅在于对 P的选择不同。

我们给出两种 2-近似算法，它们在选择 P 时采用不同的原则。算法 1 为 G构造一个随机子图，即 $\mathcal{G}_1$，并使用 P，它是 $\mathcal{G}_1$的一个最小顶点覆盖。另一方面，算法 2 估计任意顶点 $v \in V$属于 $\mathcal{G}^\star$的最小顶点覆盖的概率，记为 c_v，并使用 $P = \{v \mid c_v > 1/2\}$。虽然这两种算法在最坏情况下都能实现 2-近似，但其性能保证会因 c_v的分布而异。具体而言，这两种算法都会过度纳入一些顶点——也就是说，纳入不属于最小顶点覆盖的顶点——但它们过度纳入的顶点类型不同。正如我们的分析所示，第一种算法会显著过度纳入 c_v非常小的顶点，而第二种算法只会过度纳入具有 $c_v > \frac{1}{2}$ 的顶点。

我们的 $3/2$-近似算法（算法3）结合这两种见解来定义集合$P = P_1 \cup P_2$。它首先选择 τ，仔细平衡具有 $c_v > \tau$ 的顶点对最小顶点覆盖期望大小的贡献。对于那些 $c_v \in [1-\tau-\varepsilon, \tau]$的顶点，我们采用算法1 的做法，只有当它们也属于某个 P_1 的最小顶点覆盖时，才将其纳入

虚构随机子图。对于具有 $c_v > \tau$的顶点集合，我们采用算法2的做法，将它们全部纳入 P_2。我们的分析仔细平衡了顶点的过度纳入，以实现 3/2-近似。

如上所述，我们的算法需要知道 c_vs，即顶点属于最小顶点覆盖的概率。然而，我们的算法和分析可以直接扩展到使用 c_v的估计值；当给定最小顶点覆盖问题的预言机时，这些估计值可以高效计算。此外，我们的方法也能轻松应对 c_v的显著误估，例如，它所达到的近似因子是最小顶点覆盖问题任意预言机的 $3/2 + \varepsilon$ 倍。我们将在第6节进一步讨论这一点。

2 符号说明

我们处理一个已知的任意图 $G = (V, E)$，以及 $p_e \in (0, 1]$ 每个$e \in E$的存在概率。我们考虑一个随机子图 $\mathcal{G}^\star$，其中每条边 $e \in E$ 都以概率p_e独立地实现。我们记 $p = \min_{e \in E} p_e$。

令 MVC 表示一个函数：给定任意输入图，输出该图的最小顶点覆盖。我们也将其称为最小顶点覆盖预言机。我们定义 $\mathrm{OPT} = \mathrm{MVC}(\mathcal{G}^\star)$ 为我们问题的最优解。注意，OPT 是一个随机变量，因为 $\mathcal{G}^\star$ 本身是 G的随机实现。我们还令 $\mathrm{OPT} = \mathbb{E}[|\mathrm{OPT}|]$ 表示该最优解的期望大小。此外，对于任意顶点 $v \in V$，我们定义

$$c_v = \Pr[v \in \mathrm{OPT}],$$

这是 v 加入最优解的概率。这意味着 $\sum_{v \in V} c_v = \mathrm{OPT}$. 同样地，对于图 G 中的任意边 $e = (u, v)$，我们令 c_e 表示该边被 OPT覆盖的概率。也就是说，

$$c_{(u,v)} = \Pr[u \in \mathrm{OPT} \text{ or } v \in \mathrm{OPT}].$$

当 c_v 和 c_e事先未知时，我们使用多项式次最小顶点覆盖预言机调用来估算它们，精度可任意高。关于这些估算的更多细节，请参见第 6 节。

3 预热——超越 2-近似

本节首先讨论我们 3/2-近似算法的两个简化变体。这两种算法在最坏情况下的近似比均为 2。不过，它们在不同问题实例上的表现各不相同。如果 OPT 中很大一部分来自 c_v较小的顶点，其中一种算法表现更好；而另一种算法则更适用于 OPT中很大一部分来自 c_v较大的顶点。讨论完这两种算法后，我们将说明，由于它们的特性相反，选用两者中表现更好的算法即可超越 2-近似比。最后，我们将探讨这一观察如何启发我们设计 3/2-近似算法。

本文中设计的所有算法都遵循类似的框架。在所有算法中，我们首先选取一个顶点子集 P，并承诺将其加入最终顶点覆盖。因此，我们只需查询未被这些顶点覆盖的边。我们将这个子图记为 H。形式上，$H = G[V \setminus P]$ 是 G中由 $V \setminus P$诱导的子图。查询 H后，我们找到其实现的边的一个顶点覆盖，记为 M。最后，我们输出 $P \cup M$。我们的算法主要区别在于 P的选择。图1 展示了我们框架的示意图。

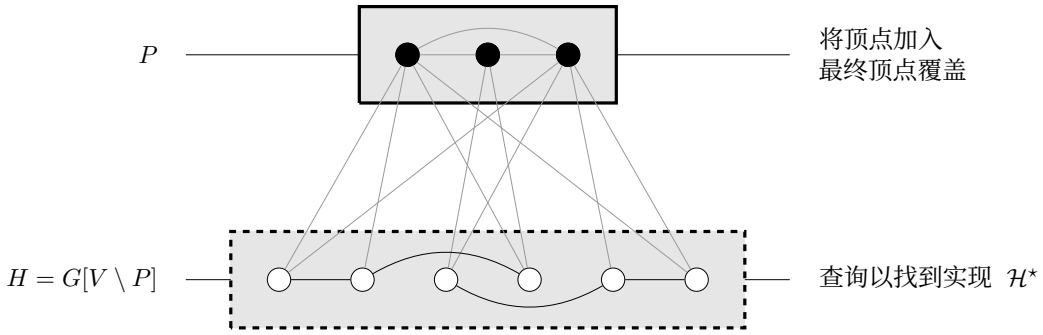

图 1：**先提交，再查询的方法**。将 P 的顶点加入最终顶点覆盖后，我们查询未被 P 覆盖的边所构成的子图，并将其记为 H。这是由 $V \setminus P$ 在 G 中诱导出的子图。我们的算法输出为 $P \cup M$，其中 M 是 $\mathcal{H}^\star$ 的一个顶点覆盖。

我们对算法框架作如下观察。

观察 3.1。令 $P \subseteq V$ 和 $H = G[V \setminus P]$ 为 $V \setminus P$ 上的诱导子图。令 $\mathcal{H}^\star$ 为 H 的边的实现，且 M 为 $\mathcal{H}^\star$ 的一个顶点覆盖。那么，$P \cup M$ 是 $\mathcal{G}^\star$ 的一个顶点覆盖。

选择 M 最简单的方法，是令其为 H 的所有已实现边的最小顶点覆盖（即 $\mathcal{H}^\star$）。然而，有时为了分析，我们要求 M 是满足某项性质的 H 的一个顶点覆盖。具体而言，我们希望 $\Pr[v \in M] = c_v$。为此，我们令 M 为 H 的所有已实现边的最小顶点覆盖，并且是 $G \setminus H$ 的一个幻觉。我们将在下面的算法中对此作更详细的解释。

第一个 2-近似算法。为了构造子集 P，该算法（正式表述为算法 1）会构造一个 G 的随机实现，并令 P 为其最小顶点覆盖。按照前述框架，下一步是找到 M：一个 $\mathcal{H}^\star$的顶点覆盖。为此，算法会构造 G 的另一个实现：独立地以概率 p_e加入 $e \in G \setminus H$中的任意一条边，并加入 $\mathcal{H}^\star$中的边。最后，它找到该实现的一个最小顶点覆盖，记为 M，并输出 $P \cup M$。

算法 1。一个 2-近似随机顶点覆盖算法

令 $\mathcal{G}_1$ 为 G 的一个随机实现，其中独立地以概率 p_e包含任意一条边 $e \in G$。$P \leftarrow$ 最小顶点覆盖($\mathcal{G}_1$)令 H 为 G 中由 $V \setminus P$诱导的子图。查询子图 H，并令 $\mathcal{H}^\star$ 为其实现。令 $\mathcal{G}_2$ 为 G 的一个子图，其中包含 $\mathcal{H}^\star$ 中的所有边，并独立地以概率 p_e包含任意一条边 $e \in G \setminus H$。$M \leftarrow$最小顶点覆盖($\mathcal{G}_2$)返回 $P \cup M$

我们首先证明，该算法只查询 $O(n/p)$ 条边，即 $|H| = O(n/p)$。这是因为，根据定义，H 中的边都不会在 $\mathcal{G}_1$ 中实现，否则，其某个顶点就应当属于顶点覆盖 P（这与假设矛盾）。对于任何包含超过 n/p 条边的子图，其中所有边都不在 $\mathcal{G}_1$ 中实现的概率至多为 $(1-p)^{n/p}$。此外，由于 H 是 G的诱导子图，它至多有 2^n 种可能。应用

并集界可知，以高概率， H 至多有 n/p 条边。即:

$$\Pr[|H| \leq n/p] \geq 1 - 2^n(1-p)^{n/p} \geq 1 - \frac{2^n}{e^n} = 1 - (2/e)^n.$$

我们将在下文陈述这一观察，以供后续参考。

观察 3.2. 设 $\mathcal{G}$ 是 G 的一个随机实现，其中每条边 $e \in G$ 都以概率 p_e 独立地包含在内；设 M 是 $\mathcal{G}$ 的一个最小顶点覆盖。 G 中未被 M 覆盖的边数为 $O(n/p)$。

如前所述，我们不直接令 M 为 $\mathcal{H}^\star$ 的任意一个最小顶点覆盖，这仅仅是出于分析上的考虑。相反，我们使用 $M = \text{MVC}(\mathcal{G}_2)$，因为它满足 $\Pr[v \in M] = c_v$. 。这是因为 $\mathcal{G}_2$ 和 $\mathcal{G}^\star$ 服从相同的分布，即 $\mathcal{G}_2$ 中的每条边 e 都以概率 p_e 独立地包含在内。下文将正式陈述这一观察。

观察 3.3. 设 H 是 G 的一个子图， $\mathcal{H}^\star$ 是其实际实现。此外，我们将 $\mathcal{G}_2$ 定义为 G 的一个子图，其中包含 $\mathcal{H}^\star$ 中的所有边，并且其中的每条边 $e \in G \setminus H$ 都以概率 p_e 独立地包含在内。若 M 是 $\mathcal{G}_2$ 的一个最小顶点覆盖，则它满足 $\Pr[v \in M] = c_v$.

根据这一观察以及 P 也来自与OPT相同的分布这一事实，我们得到 $\mathbb{E}[|P \cup M|] \leq 2\text{OPT}$，这意味着算法 1 是一个 2-近似算法。不过，我们断言，根据 c_v 的分布方式，该算法可能得到优于 2 的近似比。首先，注意到 $\mathcal{G}_1$ 和 $\mathcal{G}_2$ 是两个独立随机变量。 $\mathcal{G}_1$ 影响 $\mathcal{G}_2$ 构造的唯一方式，是决定其中哪些边来自实际实现，哪些边是虚构出来的。尽管如此， $\mathcal{G}_2$ 包含任意一条边 e的概率为 p_e ，且这一事件与其他边以及 $\mathcal{G}_1$相互独立。这意味着 P =MVC($\mathcal{G}_1$) 和 M =MVC($\mathcal{G}_2$) 也是两个独立随机变量。因此，任意顶点 v 加入 P和 M 的概率都相互独立，均为 c_v ，这意味着

$$\Pr[v \in P \cup M] = 1 - (1 - c_v)^2 = c_v(2 - c_v). \tag{1}$$

由于对于 $c_v \to 0$， $c_v(2-c_v) \approx 2c_v$，该算法的最坏情况是所有顶点的 c_v 都非常小。另一方面，如果对所有顶点都有 $c_v \geq 0.5$ ，那么 $c_v(2 - c_v) \leq 1.5c_v$，从而得到近似比 $3/2$ （这是我们所期望的界）。明确这一点后，接下来我们将讨论另一种性质相反的 2-近似算法。如果最优解中很大一部分来自 c_v 较小的顶点，该算法的表现会更好。

第二个 2-近似算法。 该算法（正式表述为算法 2）并非严格的 2-近似算法。相反，对于任意 $2 + O(\varepsilon)$，它使用 $O(n/\varepsilon p$ 次查询，求得一个 $\varepsilon \in (0,0.1)$。 在该算法中，我们设 $P = \{v \in V: c_v \geq 0.5 - \varepsilon\}$。算法的其余部分遵循与算法 1 类似的标准框架。

算法 2. 一个 $(2+O(\varepsilon))$-近似随机顶点覆盖算法

定义 $P = \{v \in V\colon\ c_v \geq 0.5 - \varepsilon\}$。令 H 为由 $V \setminus P$ 在 G 中诱导的子图。查询子图 H，并令 $\mathcal{H}^\star$ 为其实现。令 $\mathcal{G}$ 为 G 的一个子图，该子图包含 $\mathcal{H}^\star$ 中的所有边，以及每条独立地以概率 p_e。$M \leftarrow$ 包含的边。MVC($\mathcal{G}$) 返回 $P \cup M$

我们首先讨论上述算法为何只查询 $O(n/\varepsilon p)$ 条边。换言之，观察 3.2 表明，G 中未被 OPT 覆盖的边数的期望，或者等价地，$\sum_e (1 - c_e)$ 的期望为 $O(n/p)$。利用马尔可夫不等式可知，满足 $c_e < 1-\varepsilon$ 的边数上界为 $O(n/\varepsilon p)$。由于任意一条边 $e \in H$ 的两个端点都具有 $c_v < 0.5-\varepsilon$，因此所有这些边都具有 $c_e \leq 1 - 2\varepsilon$。所以，这类边至多有 $O(n/\varepsilon p)$ 条。

接下来我们将证明，算法 2是一个 $(2+O(\varepsilon))$-近似算法。显然，对于任意顶点 $v \in P$，我们有 $\Pr[v \in P \cup M] = 1$。此外，任意一个 $v \notin P$以概率 M加入 c_v(根据观察 3.3)，这意味着 $\Pr[v \in P \cup M] = c_v$。因此

$$\mathbb{E}[|P \cup M|] = \sum_{v \in P} \Pr[v \in P \cup M] + \sum_{v \notin P} \Pr[v \in P \cup M] = \mathbb{E}[|P|] + \sum_{v \notin P} c_v. \tag{2}$$

令 α 表示最优解中不在 P 中的部分。即

$$\alpha = \frac{\sum_{v \notin P} c_v}{\mathrm{OPT}}.$$

根据定义，任何顶点 $v \in P$ 都满足 $c_v \geq 0.5 - \varepsilon$，因此我们有

$$\mathbb{E}[|P|] \times (0.5 - \varepsilon) \leq \sum_{v \in P} c_v = \mathrm{OPT} - \sum_{v \notin P} c_v = (1 - \alpha)\mathrm{OPT},$$

从而得到

$$\mathbb{E}[|P|] \leq \frac{(1-\alpha)\mathrm{OPT}}{0.5 - \varepsilon}.$$

将其与（2）结合可得

$$\mathbb{E}[|P \cup M| = \mathbb{E}[|P|] + \sum_{v \notin P} c_v \leq \frac{(1-\alpha)\mathrm{OPT}}{0.5-\varepsilon} + \alpha\mathrm{OPT} = \mathrm{OPT}\frac{1 - \alpha(0.5+\varepsilon)}{0.5-\varepsilon} = \mathrm{OPT}(2 - \alpha + O(\varepsilon)), \tag{3}$$

并意味着 $(2+O(\varepsilon))$-近似比。注意，只有当 $\alpha = 0$ 时，该界才是紧的。如果对于问题的某个实例，大量顶点具有 $c_v < 0.5 - \varepsilon$，从而 α 很大，则该算法可实现更好的近似比。

3.1 击败2-近似

如上所述，算法2在 OPT 中有很大一部分来自具有 $c_v < 0.5 - \varepsilon$ 的顶点时，表现更好；而算法1 则几乎恰好相反。因此，若要超越

2-近似比，一种思路是从这两种算法中选取效果更好的一种来运行。我们将证明，这样做可达到 $5/3+O(\varepsilon)$的近似比。给定参数 $\varepsilon \in (0,\ 0.1)$，回顾一下 α 的定义：

$$\alpha = \frac{\sum_{v:c_v<0.5-\varepsilon} c_v}{\mathrm{OPT}}. \tag{4}$$

此外，令 S_1 和 S_2 分别表示算法1 和算法2 输出的解。作为 $|S_1|$ 的上界，我们有

$$\begin{aligned}\mathbb{E}[|S_1|] = \sum_{v\in V} \Pr[v\in S_1] &\overset{(1)}{=} \sum_{v\in V} c_v(2-c_v) \le \sum_{v:c_v\ge 0.5-\varepsilon}(1.5+\varepsilon)\cdot c_v + \sum_{v:c_v<0.5-\varepsilon} 2\cdot c_v \\ &\overset{(4)}{=} (1.5+\varepsilon)(1-\alpha)\mathrm{OPT} + 2\alpha\mathrm{OPT} = \mathrm{OPT}(1.5+0.5\alpha+O(\varepsilon)).\end{aligned}$$

此外，根据（3）可得

$$\mathbb{E}[|S_2|] = \mathrm{OPT}(2-\alpha+O(\varepsilon))$$

因此，运行这两种算法中表现最佳者所达到的近似比的上界为

$$\max\left[(2-\alpha+O(\varepsilon)), (1.5+0.5\alpha+O(\varepsilon))\right]$$

我们注意到，该项在 $\alpha = 1/3$ 时取得最小值，此时近似比为$5/3+O(\varepsilon)$。当 $2/3$ 中的 OPT 来自具有 $c_v = 0.5$ 的顶点，而其余部分来自具有 $c_v \to 0$ 的顶点时，两种算法都能达到该近似比，因此该分析是紧的。

4 $(3/2+\varepsilon)$-近似算法

受上述算法的启发，本节设计一种 3/2-近似算法。在本节中，我们假设 c_v已预先知晓。我们将在第6节中放宽这一假设，方法是通过调用最小顶点覆盖预言机多项式次，直接将 c_v估计到任意所需的精度。

与算法1 和算法2类似，该算法首先选取一个顶点子集 P，并承诺将其纳入最终解，然后查询未被这些顶点覆盖的边，即 $H = G[V\setminus P]$。算法首先选取一个阈值 τ，该阈值可因实例而异。根据此阈值以及 c_v 这些顶点中的情况，它承诺将集合 $P = P_1 \cup P_2$纳入解中。对于 $c_v \in [1-\tau-\varepsilon, \tau]$的顶点，我们采用算法1 的方式，仅当它们也属于某个虚构随机子图的最小顶点覆盖时，才将其纳入 P_1。对于 $c_v > \tau$的顶点集合，我们采用算法2 的方式，将它们全部纳入 P_2。

考虑该问题给定实例的一个解 S。如果我们断言 S 是一个（$3/2+\varepsilon$）-近似解，则需要证明

$$\sum_{v\in V} \Pr[v\in S] \le (3/2+\varepsilon)\cdot \mathrm{OPT} = (3/2+\varepsilon)\sum_{v\in V} c_v.$$

换句话说， S 应满足

$$\sum_{v\in V} \left((3/2+\varepsilon)\cdot c_v - \Pr[v\in S]\right) \ge 0.$$

受此启发，对于任意顶点 v，我们将该顶点的预算定义为

$$b_v = \max\Big((3/2+\varepsilon)\cdot c_v - \Pr[v\in S],\, 0\Big) \tag{5}$$

其成本为

$$\sigma_v = \max\Big(\Pr[v\in S] - (3/2+\varepsilon)\cdot c_v,\, 0\Big) \tag{6}$$

证明 S 是一个 3/2-近似解，等价于证明我们可以用具有 $b_v > 0$ 的顶点预算来支付具有 $\sigma_v > 0$ 的顶点成本。形式化地，

命题 4.1。设 S 是 $\mathcal{G}^\star$ 的一个顶点覆盖。此外，对于任意顶点 $v \in V$，考虑 b_v 和 σ_v，它们分别由 (5) 和 (6)定义。如果 S 满足

$$\sum_{v\in V} b_v - \sum_{v\in V}\sigma_v \geq 0,$$

然后 $\mathbb{E}[|S|] \leq (3/2+\varepsilon)\mathrm{OPT}$.

证明。由于对于任意顶点 $v \in V$，我们有

$$\begin{aligned} b_v - \sigma_v &= \max\Big((1.5+\varepsilon)\cdot c_v - \Pr[v\in S],\, 0\Big) - \max\Big(\Pr[v\in S] - (1.5+\varepsilon)\cdot c_v,\, 0\Big) \\ &= (1.5+\varepsilon)\cdot c_v - \Pr[v\in S], \end{aligned}$$

我们得到

$$\begin{aligned} \sum_{v\in V}(b_v-\sigma_v) &= \sum_{v\in V}\Big((1.5+\varepsilon)\cdot c_v - \Pr[v\in S]\Big) = \sum_{v\in V}(1.5+\varepsilon)\cdot c_v - \sum_{v\in V}\Pr[v\in S] \\ &= (1.5+\varepsilon)\mathrm{OPT} - \mathbb{E}[|S|]. \end{aligned}$$

Th因此，断言陈述中的不等式 $\sum_{v\in V}(b_v-\sigma_v) \geq 0$ 也意味着 es

$$(1.5+\varepsilon)\mathrm{OPT} - \mathbb{E}[|S|] \geq 0,$$

an因此，我们有 $\mathbb{E}[|S|] \leq (1.5+\varepsilon)\mathrm{OPT}$，从而完成了该断言的证明。 □

我们观察到，如果 S 由算法1找到，那么具有 $c_v < (0.5-\varepsilon)$的顶点具有正成本。另一方面，如果 S 由算法2找到，那么所有这些顶点都有正预算。基于这一观察，我们希望找到一个阈值 τ，使得：

- 如果在 $\{v\in V\colon\ c_v \in [1-\tau-\varepsilon, \tau]\}$上运行算法1，那么我们可以用具有 $c_v > (0.5-\varepsilon)$的顶点的预算，支付该集合中具有 $c_v < (0.5-\varepsilon)$的顶点的成本。
- 如果对其余顶点运行算法2，我们可以用具有$c_v < 1-\tau-\varepsilon$ 的顶点的预算，支付具有 $c_v > \tau$的顶点的成本。

我们断言，将 τ 设为满足条件的 $[0.5, 1]$ 中最小的数

$$\sum_{v:c_v>\tau} c_v \leq \sum_{v:c_v<1-\tau-\varepsilon} c_v$$

即可得到这些性质。然而，显然，我们不能简单地在这两个顶点子集上分别运行两个算法，因为它们之间可能存在大量的边。因此，我们需要证明，遵循这一思路并不会迫使我们查询大量边。我们在下文正式给出我们的 3/2-近似算法，即算法 3。稍后，我们将在第5节证明，对于任意 $\varepsilon \in (0,0。1)$，该算法仅使用 $3/2+\varepsilon$ 次查询即可输出一个 ($O(n/\varepsilon p)$-近似顶点覆盖。

算法 3. 我们的 3/2-近似算法。1 令 $\mathcal{G}_1$ 为 G 的一个随机实现，其中每条边 $e \in G$ 都以概率 p_e 独立地包含在内。2 $C \leftarrow$MVC($\mathcal{G}_1$)。3 令 τ 为 $[0.5,1]$ 中满足 $\sum_v$ 的最小数: $: c_v>\tau\, c_v \leq \sum_{v:\ c_v<1-\tau-\varepsilon} c_v$。4 $P \leftarrow \{v \in V:\ c_v > \tau\} \cup \{v \in V:\ c_v \in [1-\tau-\varepsilon, \tau]$ 和 $v \in C\}$。5 令 H 为 G 中由 $V \setminus P$ 诱导的子图。6 查询 H 中的边，并令 $\mathcal{H}^\star$ 为其实现7 **返回** $P \cup$MVC($\mathcal{H}^\star$)。

首先，由于该算法遵循我们的标准框架，根据观察 3.1，它输出 $\mathcal{G}^\star$ 的一个顶点覆盖。注意，在该算法中，集合 $\{v \in V:\ c_v \in [1-\tau-\varepsilon, \tau]\}$ 中的任意顶点当且仅当它属于顶点覆盖 C 时，才会加入集合P。这与算法 1 构造 P 的方式类似。因此，对于该集合中的任意顶点 v，在算法 3 中，v 加入 P 的概率与算法 1 中相同。另一方面，对于其余顶点，即 $\{v \in V:\ c_v < 1-\tau-\varepsilon$ 或 $\tau < c_v]$，P 会纳入任何具有 $\tau < c_v$ 的顶点。这些是该集合中唯一具有 $c_v \geq 0.5-\varepsilon$ 的顶点。因此，这与算法 2 构造集合 P 的方式类似。

5 分析

在下面的引理中，我们证明算法查询的边数为 $O(n/\varepsilon p)$。

引理 5.1。 来自算法 H 3 满足 $\mathbb{E}[|H|] = O(\frac{n}{\varepsilon p})$。 —

证明。 考虑一条边 $e = (u,\ v)$。我们首先证明，如果 $e \in H$，那么要么 $c_e \leq 1-\varepsilon$，要么它在 C 中未被覆盖。不妨设 $c_v \geq c_u$。由于 $e \in H$，我们知道 $v \notin P$ 和 $u \notin P$，这意味着 $c_v \leq \tau$，否则 v 会连接 P。我们将通过考察 c_u 的所有可能取值来证明这一断言。

- $c_u \geq 1-\tau-\varepsilon$: 由于我们知道 $c_v \geq c_u$ 和 $c_v \leq \tau$，在这种情况下，c_u 和 c_v 都属于 $[1-\tau-\varepsilon, \tau]$。因此，$e$ 连接 H 当且仅当它的两个端点都不在 C 中，这意味着 e 在 C 中未被覆盖。

- $c_u < 1-\tau-\varepsilon$: 由于我们知道 $c_v \leq \tau$，因此可得 $c_v + c_u \leq 1-\varepsilon$。此外，由于对于任意边 $c_v + c_u \geq c_e$ 都成立，我们得到 $c_e \leq 1-\varepsilon$。

因此，我们有

$$|H| \leq |\{e : c_e \leq 1-\varepsilon\}| + |\{e : e \text{ not covered by } C\}|.$$

观察 3.2指出，随机实现 G 中未被最小顶点覆盖覆盖的边数为 $O(n/p)$。这直接意味着

$$\mathbb{E}[|\{e : e \text{ not covered by } C\}|] = O(n/p).$$

由于 OPT 本身就是 G的一个随机实现的最小顶点覆盖，因此观察 3.2 还意味着

$$\mathbb{E}|\{e : e \text{ not covered by OPT}\}|] = \sum_{e}(1 - c_e) = O(n/p).$$

利用马尔可夫不等式，可得

$$\mathbb{E}[|\{e : c_e \leq 1 - \varepsilon\}|] = \mathbb{E}[|\{e : 1 - c_e > \varepsilon\}|] \leq \left(\sum_{e \in G}(1 - c_e)\right)/\varepsilon = O(n/\varepsilon p).$$

将这些不等式结合起来，我们得出

$$\mathbb{E}[|H|] \leq \mathbb{E}[|\{e : c_e \leq 1 - \varepsilon\}| + |\{e : e \text{ not covered by } C\}|] = O(n/\varepsilon p),$$

由此完成了对该断言的证明。 □

如前所述，为了进行分析，我们需要让 $\mathcal{H}^\star$ 的顶点覆盖以 c_v 的概率包含任意顶点 v 。为此，我们需要对算法稍作修改。下面解释修改后的算法。

算法 4. 一种仅用于分析的算法。

1 考虑子图 H 和顶点子集 P，它们来自算法3 2 查询子群 H，并令 $\mathcal{H}^\star$ 为其实现。 3 令 $H \leftarrow G \setminus H$。 4 令 $\mathcal{H}$ 为 H 的一个随机实现，其中每条边 e 均以概率 p_e独立出现。 5 $M \leftarrow \text{MVC}(\mathcal{H}^\star \cup \mathcal{H})$。 6 **返回** $M \cup P$。

在本文余下部分，我们将证明算法4 的近似比满足预期，而不采用算法3。不过，要做到这一点，我们首先需要以下观察。

观察 5.2。 令 S_1 和 S_2 分别为算法3和算法4的输出。我们有 $\mathbb{E}[|S_1|] \leq \mathbb{E}[|S_2|]$。

证明。 注意， S_1 和 S_2 都包含 P 以及 $\mathcal{H}^\star$的一个顶点覆盖。由于算法$\mathcal{H}^\star$ 中的3顶点覆盖是最小的，因此该算法的输出不大于算法4的输出。

□

现在我们可以证明关于算法4输出的解的规模的主要引理。引理陈述如下。

引理 5.3. 设 S 为算法 4的输出。我们有

$$E[|S|] \leq (3/2 + \varepsilon)\text{OPT}.$$

证明。 根据断言 4.1，要证明此引理，只需证明

$$\sum_{v\in V} b_v - \sum_{v\in V} \sigma_v \geq 0$$

其中 b_v 和 σ_v分别表示顶点 v 的预算和成本，其定义分别见式（5）和式（6）。为证明这一点，我们将图 V 的顶点划分为三个互不相交的子集 V_1、 V_2和 V_3，并分别对它们证明该等式。也就是说，对于任意 V_i，我们证明

$$\sum_{v\in V_i} b_v - \sum_{v\in V_i} \sigma_v \geq 0. \tag{7}$$

我们将这些子集定义如下（如图 2所示）：

- $V_1 = \{v:\ 0.5-\varepsilon \leq c_v \leq 0.5\}$.
- $V_2 = \{v:\ c_v > \tau\} \cup \{v:\ c_v < 1-\tau-\varepsilon\}$.
- $V_3 = \{v:\ 0.5 < c_v \leq \tau\} \cup \{v:\ 1-\tau-\varepsilon \leq c_v < 0.5-\varepsilon\}$.

我们将在引理 5中分别对子集 V_1、 V_2 和 V_3 证明式（5.5），引理 5.6，以及引理 5.7。

由于这三个子集两两不相交，且满足 $V = V_1 \cup V_2 \cup V_3$，因此我们得到

$$\sum_{v\in V} b_v - \sum_{v\in V} \sigma_v = \sum_i \left(\sum_{v\in V_i} b_v - \sum_{v\in V_i} \sigma_v \right) \geq 0 \tag{8}$$

由此完成该引理的证明。 □

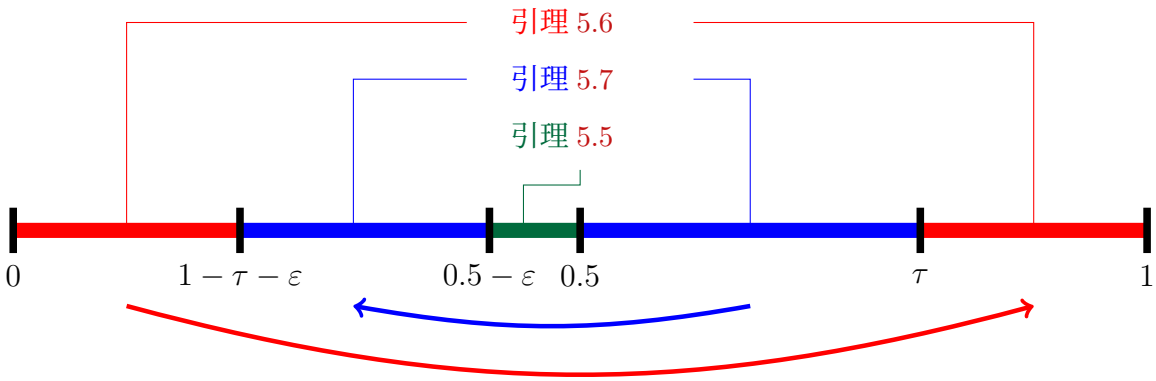

图 2： **针对不同的 c_v 管理顶点成本**。 对于集合 V_1 （绿色区域），我们在引理 5.5 中证明该集合中的顶点没有成本。对于集合 V_2 （红色区域），在引理 5.6 中，我们使用 $\{v$ 中顶点的预算 $c_v < 1-\tau-\varepsilon\}$ 来支付 $\{v$ 中顶点的成本 $c_v > \tau\}$。最后，对于集合 V_3 （蓝色区域），在引理 5.7 中，我们使用$\{v$ 中顶点的预算 $0.5 < c_v \leq \tau\}$ 来支付 $\{v$ 中顶点的成本 $1-\tau-\varepsilon \leq c_v < 0.5-\varepsilon\}$。

在正式陈述上述三个引理之前，我们需要以下断言来证明它们。

断言 5.4. 考虑 τ 在算法3中定义，令 S 为算法 4的输出。对于任意满足 v 且有 $c_v \in [1-\tau-\varepsilon, \tau]$的顶点，我们有 $\Pr[v \in S] = c_v(2-c_v)$。

证明。 考虑 M 和 P 来自算法 4。回想一下，该算法输出 $M \cup P$。也就是说$S = M \cup P$。集合 P 本身在算法 3 中定义为

$$P = \{v \in V : c_v > \tau\} \cup \{v \in V : c_v \in [1-\tau-\varepsilon, \tau] \text{ and } v \in C\},$$

其中 C 是 G的一个随机实现的最小顶点覆盖。这意味着，对于任意满足$c_v \in [1-\tau-\varepsilon, \tau]$的 v ，我们有

$$\Pr[v \in S] = \Pr[v \in C \cup M].$$

由于 M 并且 C 分别是 G的两个独立实现的最小顶点覆盖，我们得到 t

$$\mathbf{Pr}[v \in M] = \mathbf{Pr}[v \in C \cup M] = \mathbf{Pr}[v \in C] + \mathbf{Pr}[v \in M] - \mathbf{Pr}[v \in M] \times \mathbf{Pr}[v \in C] = 2c_v - c_v^2.$$

这就完成了该断言的证明。 □

引理 5.5。 考虑 τ 由算法 3定义，并定义 $V_1 = \{v : \ 0.5-\varepsilon \le c_v \le 0.5\}$。我们有

$$\sum_{v \in V_1} b_v - \sum_{v \in V_1} \sigma_v \ge 0,$$

其中 b_v 和 σ_v分别是顶点 v 相对于算法 (5) 和 (6) 中定义的预算和成本，这里的解是 S 由算法 4输出的。

证明。 根据断言 5.4，对于任意顶点 $v \in V_1$，有 $\mathbf{Pr}[v \in S] = c_v(2-c_v)$。 结合 V_1 中的任意顶点都满足 $c_v > 0.5-\varepsilon$ 这一事实，可得

$$\Pr[v \in S] \le c_v(2-0.5+\varepsilon) = c_v(1.5+\varepsilon).$$

这意味着，对于任意顶点 $v \in V_1$，都有 $b_v = (1.5+\varepsilon)c_v - \mathbf{Pr}[v \in S] \ge 0$ 和 $\sigma_v = 0$， 因此

$$\sum_{v \in V_1} b_v - \sum_{v \in V_1} \sigma_v \ge 0,$$

从而完成了该断言的证明。 □

引理 5.6。 考虑 τ 在算法 3中定义，并令 $V_2 = \{v : \ c_v > \tau\} \cup \{v : \ c_v < 1-\tau-\varepsilon\}$.该集合满足

$$\sum_{v \in V_2} b_v - \sum_{v \in V_2} \sigma_v \ge 0,$$

其中， b_v 和 σ_v，顶点 v 的预算和成本是根据解（5）和（6）定义的；该解由算法4输出。

证明。 我们先来定义

$$A = \{v : c_v > \tau\} \quad \text{and} \quad B = \{v : c_v < 1-\tau-\varepsilon\},$$

其中 $V_2 = A \cup B$。考虑集合 P和 M，它们是算法 4 的输出。也就是说$S = M \cup P$。注意，根据 P的定义，我们有 $A \subset P$，因此对于任意$[v \in S] = 1$ ，$\mathbf{Pr} v \in A$。因此，

$$b_v - \sigma_v \overset{(6),(5)}{=} \max\Big((1.5+\varepsilon)c_v - \Pr[v \in S], 0\Big) - \max\Big(\Pr[v \in S] - (1.5+\varepsilon)c_v, 0\Big) = (1.5+\varepsilon)c_v - 1.$$

此外，我们有 $B \cap P = \emptyset$，并且根据观察 3.3，对于任意 $u \in B$，我们有 $\mathbf{Pr}[u \in M] = c_u$.。这意味着 $\mathbf{Pr}[u \in S] = c_u$，因此顶点 u 的预算为

$$b_u = \max\Big((1.5+\varepsilon)c_u - \Pr[v \in P \cup M],\, 0\Big) = (1.5+\varepsilon)c_u - c_u = c_u(0.5+\varepsilon),$$

以及 $\sigma_u = 0$。将这些结果结合起来可得

$$\begin{aligned}\sum_{v\in V_2}(b_v-\sigma_v) &= \sum_{v\in A}(b_v-\sigma_v) + \sum_{u\in B}(b_u-\sigma_u) = \sum_{v\in A}\big((1.5+\varepsilon)c_v - 1\big) + \sum_{u\in B}(0.5+\varepsilon)c_u \\ &= (1.5+\varepsilon)\sum_{v\in A}c_v - |A| + (0.5+\varepsilon)\sum_{u\in B}c_u.\end{aligned}$$

注意，根据 τ 的定义，我们有 $\sum_{v\in A} c_v \le \sum_{u\in B} c_u$。因此，可以写作

$$\begin{aligned}\sum_{v\in V_2}(b_v-\sigma_v) &= (1.5+\varepsilon)\sum_{v\in A}c_v - |A| + (0.5+\varepsilon)\sum_{u\in B}c_u \\ &\ge (2+2\varepsilon)\sum_{v\in A}c_v - |A|.\end{aligned}$$

由于对于任意顶点 $v \in A$，都有 $c_v > \tau \ge 0.5$，因此可得

$$\sum_{v\in V_2}(b_v-\sigma_v) \ge (2+2\varepsilon)\sum_{v\in A}c_v - |A| \ge (2+2\varepsilon)\sum_{v\in A}0.5 - |A| \ge \varepsilon|A| \ge 0.$$

证明至此完成。 □

引理 5.7. 考虑 τ 在算法 3中定义，并令

$$V_3 = \{v : 0.5 < c_v \le \tau\} \cup \{v : 1-\tau-\varepsilon \le c_v < 0.5-\varepsilon\},$$

该集合满足

$$\sum_{v\in V_3} b_v - \sum_{v\in V_3}\sigma_v \ge 0,$$

其中 b_v 和 σ_v分别表示顶点 v 的预算和成本，其定义见 (5) 和 (6)，这里的解为 S 算法4输出的结果。

由于证明详细且篇幅较长，我们将第 5.1 节专门用于证明这个引理。下面，我们重述主要定理，并正式证明该算法所需的近似比和查询次数。随后，在第 6节中，我们将说明如何仅通过调用最小顶点覆盖预言机（n对数 n/ε^2）次就能获得相同的界。

定理 1.1 （重述）。对于任意 $\varepsilon \in (0,0.1)$，算法 3 可找到一个顶点覆盖 $\mathcal{G}^\star$，其期望大小至多为 $(1.5+\varepsilon)\mathrm{OPT}$，方法是查询 $O(n/\varepsilon p)$ 条总边数。

证明。根据引理5.1，我们知道算法 3 只需要 $O(n/\varepsilon p)$ 次查询。令 S 为算法 3输出的解。根据观察 3.1， S 是 $\mathcal{G}^\star$ 的一个顶点覆盖；根据观察 5.2，其期望大小不超过算法 4的输出。在引理5.3 中，我们证明算法 4 的输出不超过 $(1.5+\varepsilon)\mathrm{OPT}$.。综合以上结果可知， S 是 $\mathcal{G}^\star$ 的一个顶点覆盖，其期望大小至多为 $(1.5+\varepsilon)\mathrm{OPT}$。 □

5.1 引理 5.7 的证明

定义子集

$$A = \{v : 0.5 < c_v \le \tau\} \quad \text{and} \quad B = \{v : 1 - \tau - \varepsilon \le c_v < 0.5 - \varepsilon\},$$

其中 $V_3 = A \cup B$。根据断言 5.4，对于任意顶点 $v \in V_3$，我们有 $\Pr[v \in S] = c_v(2 - c_v)$，因此 s

$$\Pr[v \in S] - (1.5 + \varepsilon)c_v = c_v(2 - c_v) - (1.5 + \varepsilon)c_v = (0.5 - \varepsilon)c_v - c_v^2.$$

注意，对于 $v \in A$ 中的顶点，这一项非正，因此该顶点的成本为零，预算为

$$b_v = c_v^2 - (0.5 - \varepsilon) \cdot c_v.$$

另一方面，由于 $(0.5 - \varepsilon)c_u - c_u^2 > 0$ 对任意顶点 $u \in B$ 都成立，因此该顶点的预算为零，成本为

$$\sigma_u = (0.5 - \varepsilon) \cdot c_u - c_u^2$$

现在，我们将证明可以用 A 中顶点的预算来支付 B 中顶点的成本。证明这一点很大程度上依赖于阈值 τ 的选取方式。

要完成该引理的证明，我们还需要以下两个断言。

断言 5.8. 对于任意一对顶点 $u \in B$ 和 $v \in A$，若 $c_u \ge 1 - c_v - \varepsilon$，则

$$b_v - \sigma_u \ge (\sigma_u / c_u)(c_v - c_u).$$

证明. 我们首先如下证明 $\sigma_u / c_u \le b_v / c_v$。

$$\begin{aligned}
\sigma_u / c_u &= \frac{(0.5 - \varepsilon) \cdot c_u - c_u^2}{c_u} = 0.5 - \varepsilon - c_u \\
&\le 0.5 - \varepsilon - (1 - c_v - \varepsilon) = c_v - 0.5 = \frac{c_v^2 - (0.5 - \varepsilon) \cdot c_v}{c_v} - \varepsilon \\
&\le b_v / c_v - \varepsilon \\
&< b_v / c_v. \qquad (9)
\end{aligned}$$

因此，我们可以写成

$$b_v - \sigma_u = c_v(b_v / c_v) - c_u(\sigma_u / c_u) \overset{(9)}{\ge} (\sigma_u / c_u)(c_v - c_u),$$

从而完成该断言的证明。 □

断言 5.9. 我们按顶点在 B 中的 c_u 递增顺序对其排序，并用 u_i 表示第 i 个顶点。我们断言，对于任意 $i \in |B|$，都可以用 $A_i = \{v \in A : \ c_v \ge 1 - c_{u_i} - \varepsilon\}$中顶点的预算支付 $B_i = \{u \in B : \ c_u \le c_{u_i}\}$ 中顶点的成本。也就是说

$$\sum_{v \in A_i} b_v - \sum_{u \in B_i} \sigma_u \ge 0. \qquad (10)$$

证明。在 ord 为证明这一断言，我们通过归纳法证明以下更强的不等式 .

$$\sum_{v \in A_i} b_v - \sum_{u \in B_i} \sigma_u \ge (\sigma_{u_i} / c_{u_i}) \left(\sum_{v \in A_i} c_v - \sum_{u \in B_i} c_u \right). \qquad (11)$$

这样便证明了这一断言，因为根据 τ 的定义，对于任意 $i \in |B|$，我们有

$$\sum_{v \in A_i} c_v - \sum_{u \in B_i} c_u \geq 0. \tag{12}$$

作为 $i = 1$ 的基础情形，我们需要证明等式 10 对 $A_1 = \{v \in A : \ c_v \geq 1 - c_{u_1} - \varepsilon\}$以及 $B_1 = \{u_1\}$成立。也就是说

$$\sum_{v \in A_1} b_v - \sigma_{u_1} \geq (\sigma_{u_1}/c_{u_1}) \left(\sum_{v \in A_1} c_v - c_{u_1} \right). \tag{13}$$

由于对于任意 $v \in A_1$，都有 $c_v \geq 1 - c_{u_1} - \varepsilon$，因此由断言 5.8 可得该不等式，从而证明了基础情形。现在，在归纳步骤中，我们将证明等式（10）对 $i = j$ 成立，假设它对 $i = j - 1$成立。我们可以写成

$$\begin{aligned}
\sum_{v \in A_j} b_v - \sum_{u \in B_j} \sigma_u &= \sum_{v \in A_{j-1}} b_v + \sum_{v \in A_j \setminus A_{j-1}} b_v - \sum_{u \in B_{j-1}} \sigma_u - \sigma_{u_j} \\
&\geq (\sigma_{u_{j-1}}/c_{u_{j-1}}) \left(\sum_{v \in A_{j-1}} c_v - \sum_{u \in B_{j-1}} c_u \right) + \left(\sum_{v \in B_j \setminus B_{j-1}} b_v - \sigma_{u_j} \right) \\
&\geq (\sigma_{u_j}/c_{u_j}) \left(\sum_{v \in A_{j-1}} c_v - \sum_{u \in B_{j-1}} c_u \right) + \left(\sum_{v \in B_j \setminus B_{j-1}} b_v - \sigma_{u_j} \right),
\end{aligned}$$

其中第二个不等式由归纳假设得出，最后一个不等式则由以下事实得出

$$\sigma_{u_{j-1}}/c_{u_{j-1}} \geq \sigma_{u_j}/c_{u_j}.$$

要证明归纳步骤，只需证明以下结论成立。

$$\sum_{v \in B_j \setminus B_{j-1}} b_v - \sigma_{u_j} \geq (\sigma_{u_j}/c_{u_j}) \left(\sum_{v \in B_j \setminus B_{j-1}} c_v - c_{u_j} \right).$$

注意，这是上文等式（13）的一个一般化版本。根据定义，对于任意$u \in B_j \setminus B_{j-1}$，都有 $c_v \geq 1 - c_{u_j} - \varepsilon$。因此，由断言 5.8可得

$$\sum_{v \in B_j \setminus B_{j-1}} b_v - \sigma_{u_j} \geq \sum_{v \in B_j \setminus B_{j-1}} (\sigma_{u_j}/c_{u_j})(c_v - c_{u_j}) \geq (\sigma_{u_j}/c_{u_j}) \left(\sum_{v \in B_j \setminus B_{j-1}} c_v - c_{u_j} \right).$$

这就完成了归纳步骤以及该引理的证明。 □

请注意，证明断言 5.9也就完成了引理 5.7的证明。令 $x = |B|$。断言 5.9 对于 $i = x$ 成立意味着

$$\sum_{v \in A} b_v - \sum_{u \in B} \sigma_u \geq 0.$$

由于 A 中的顶点成本为零，这也意味着

$$\sum_{v \in V_3} (b_v - \sigma_u) \geq 0,$$

完成引理5.7的证明。 □

6 使用近似 c_vs

本节将讨论如何仅通过多项式次调用最小顶点覆盖预言机来实现算法 3。我们还将在备注 6.3 中讨论，这个预言机不必是精确的。注意，在算法 3中，我们需要知道所有顶点的 c_v （即某个顶点属于最小顶点覆盖的概率）。如果不限制调用预言机的次数，我们就能精确计算这些值。不过，我们断言，在算法 3 中使用这些参数的估计值，仍然可以得到相同的界。下面的算法会先为任意 c_v 求出一个估计值，然后将其输入算法 3。

算法 5。 预言机高效的 3/2-近似算法

1 抽取 $t = \frac{n^2}{8\varepsilon^2}\ln(2n/\delta)$ 个 G 的实现，并将其记为 $\mathcal{G}_1, \ldots, \mathcal{G}_t$. 2 对任意 $i \in [t]$，令 $C_i =$ 最小顶点覆盖$(\mathcal{G}_1)$。3 对任意顶点 $v \in V$，令 $\bar{c}_v$ 为包含 v 的 C_i所占的比例。4 使用参数 $\bar{c}_v$s 和 $\varepsilon' = \varepsilon/2$ 运行算法 3（而非 $c_v s$ 和 ε）。

我们首先证明，算法 5 以高概率返回 $\bar{c}_v$s，其与相应的 $\varepsilon/2n$ s 的差距在 c_v以内。

断言 6.1。 在算法 $\delta = 1/n$ 中设置 5，我们得到 $|c_v - \bar{c}_v| \le \varepsilon/2n$，对所有 $v \in V$，其成立概率至少为 $1 - 1/n$。

证明。 该证明可由霍夫丁界的简单应用得出。设 $\alpha = \varepsilon/2n$。令$X_i^v = 1(v \in C_i)$为一个示性变量，用于表示 $v \in C_i$是否成立。注意，对于任意 i 和 $v \in V$， X_i^v是一个伯努利随机变量，且其 $\mathbb{E}[X_i^v] = c_v$。此外， $\bar{c}_v = \frac{1}{t}\sum_{i=1}^{t} X_i^v$。利用霍夫丁界和并集界，我们有

$$\Pr[\exists v \in V \text{ s.t. } |\bar{c}_v - c_v| \ge \alpha] \le n \cdot \Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t} X_i^v - c_v\right| \ge \alpha\right] \le 2n\exp\left(-2t\alpha^2\right) \le \delta = 1/n,$$

其中，最后一个不等式由 $t = \frac{1}{2\alpha^2}\ln\binom{2n}{\delta}$的选取可得。 —

□

接下来，我们证明算法5 返回一个顶点覆盖 $\mathcal{G}^\star$，其期望大小以高概率至多为 $(3/2+\varepsilon)$ $\mathrm{OPT} + O(\varepsilon)$，且仅查询 $O(n/p\varepsilon)$ 条边。

定理 6.2。 对于任意 $\varepsilon \in (0,0.1)$，算法 5 通过查询 $\mathcal{G}^\star$，找到一个期望大小至多为 $(3/2+\varepsilon)$ $\mathrm{OPT} + O(\varepsilon)$ 的顶点覆盖，总共查询 $O(n/\varepsilon p)$ 条边。此外，该算法的预言机查询效率很高。

证明概要。我们首先给出查询数量（ H 中边的数量）的一个上界，用于算法 3。令 $\alpha = \varepsilon/2n$。参照引理 5.1的证明，很容易验证，若 $e = (u,v) \in H$，则要么 $\bar{c}_v + \bar{c}_u \le 1 - \varepsilon'$，要么它未被覆盖在 C。根据断言 6.1，以高概率有 $|c_v - \bar{c}_v| \le \alpha$。因此，以高概率，要么 e 满足 $c_v + c_u \le 1 - \varepsilon' + 2\alpha \le 1 - \varepsilon$，要么它未被覆盖在 C。类似于引理 5.1的证明，我们得到 $|H| = O(n/\varepsilon p)$。

接下来，我们来界定算法 5 的近似比。设 τ 为算法 3在参数 $\bar{c}_v$s 和 $\varepsilon' = \varepsilon/2$ （而非 $c_v s$ 和 ε）下运行时的参数。也就是说， τ 是最小的

数，使得 $[0.5, 1]$ 满足 $\sum_{v:\ \bar{c}_v > \tau} \bar{c}_v \leq \sum_{v:\ \bar{c}_v < 1-\tau-\varepsilon'} \bar{c}_v$。我们将根据 τ 的值，把顶点划分为互不相交的子集。

- $A = \{v \in V:\ \bar{c}_v > \tau\}$.
- $B = \{v \in V:\ \bar{c}_v \in [1 - \tau - \varepsilon, \tau]\}$.
- $C = \{v \in V:\ \bar{c}_v < 1 - \tau - \varepsilon\}$.

令 S 为该算法输出的顶点覆盖。根据算法 3，集合 A 中的顶点以概率 **1** 加入 S，而集合 C 中的顶点以概率 c_v（即 v 加入 OPT的概率）加入该顶点覆盖。此外，按照断言 5.4 的证明，我们可以验证 B 中的顶点以概率 S 加入 $2c_v - c_v^2$. 。也就是说

$$\mathbb{E}[|S|] = \sum_{v \in A} 1 + \sum_{v \in B} (c_v - c_v^2) + \sum_{v \in C} c_v.$$

为了证明我们所需的近似比，我们将证明

$$\mathbb{E}[|S|] \leq (3/2 + \varepsilon)\mathrm{OPT} + O(\varepsilon).$$

回顾方程 8，我们在证明引理 5.3时会用到它：

$$\sum_{v \in V} b_v - \sum_{v \in V} \sigma_v \geq 0.$$

按照我们的证明步骤，可以验证，如果算法 3 使用参数 $\bar{c}_v$ 而不是c_vs运行，则该等式意味着

$$\sum_{v \in A} 1 + \sum_{v \in B} (2\bar{c}_v - \bar{c}_v^2) + \sum_{v \in C} \bar{c}_v - (3/2 + \varepsilon)\mathrm{OPT} \leq 0. \tag{14}$$

由于断言 6.1指出，对于所有 $v \in V$， $|c_v - \bar{c}_v| \leq \varepsilon/2n$ 以高概率成立，因此下式也以高概率成立：

$$\sum_{v \in A} 1 + \sum_{v \in B} (2\bar{c}_v - \bar{c}_v^2) + \sum_{v \in C} \bar{c}_v - \mathbb{E}[|S|] = \sum_{v \in B} (2\bar{c}_v - \bar{c}_v^2 - 2c_v + c_v^2) + \sum_{v \in C} (c_v - \bar{c}_v) \geq -2n(\varepsilon/2n) \geq -\varepsilon. \tag{15}$$

Co将其与 (14) 结合可得 $\mathbb{E}[|S|] - \varepsilon - (3/2 + \varepsilon)\mathrm{OPT} \leq 0$ 进而得到

$$\mathbb{E}[|S|] \leq (3/2 + \varepsilon)\mathrm{OPT} + O(\varepsilon).$$

至此，证明概要完成。

备注 6.3。 我们指出，上述讨论中使用的最小顶点覆盖预言机不必是精确的。我们的分析只需相对于 c_vs 的总和得到一个近似因子，而这个因子取决于预言机的近似能力。因此，我们的结果可以理解为：在能够访问一个 α-近似预言机的情况下，得到一个 $3/2\alpha + \varepsilon$近似因子。这一观察的一个好处是，它可以直接利用高效的启发式方法，例如经过高度优化的整数规划工具。这些工具已知在实践中表现良好，能得到非常接近最优的结果，尽管理论上并不存在质量相当且可证明高效的近似算法。

7 轻度相关性下的紧致性

在前几节中，我们展示了一种算法，对于边独立实现的随机图，可提供一个（3/2+ε）近似。事实上，上一节给出的分析对于只有少量相关边的图仍然成立。本节将证明，对于这类图，（$3/2+\varepsilon$）近似因子是紧的。也就是说，我们构造一个只有少量相关边的随机图，并证明对于该图，任何非自适应算法以高概率都必须具有（$3/2-\varepsilon$）近似因子。我们的论证基于第 6 节 [AKL16]中的论证。

定义 7.1 (轻度相关图)。 我们称随机图 $G=(V,E)$ 为轻度相关的，若其边集 E 可以划分为集合 E_1 和 E_2 ，且满足以下条件：

- E_1 中的边彼此独立地实现：对于任意 $S_1 \subseteq E_1$

$$\Pr \bigcap_{e \in S_1} \{e \in G_r\} = \prod_{e \in S_1} \Pr[e \in G_r]$$

- 中的边 E_1 与 E_2中的边相互独立地实现：对于任意 $S_1 \subseteq E_1$ 和 $S_2 \subseteq E_2$ ，

$$\Pr\left[\bigcap_{e \in S_1} \{e \in G_r\} \;\middle|\; \bigcap_{e \in S_2} \{e \in G_r\}\right] = \prod_{e \in S_1} \Pr[e \in G_r]$$

- E_2 很小： $|E_2| = O(n)$。

值得注意的是，在我们对轻度相关图的定义中， E_2 中边的实现可能取决于 E_1中的边：我们不对以下形式的概率作任何假设

$$\Pr\left[\bigcap_{e \in S_2} \{e \in G_r\} \;\middle|\; \bigcap_{e \in S_1} \{e \in G_r\}\right],$$

其中 $S_1 \subseteq E_1$ 和 $S_2 \subseteq E_2$。

注释 7.2。 给定任意轻度相关随机图 G，以及参数 $\varepsilon \in (0,\ 0。1)$，算法 *3* 输出一个顶点覆盖 $\mathcal{G}^\star$，其期望大小至多为 $(3/2+\varepsilon)$，且只需 $O(n/\varepsilon p)$ 次查询，无需知道边的划分 E_1 和 E_2。

证明。为此，我们首先证明算法 4 至多查询 $O(n/\varepsilon p)$ 条边。仔细查看引理 5.1 中的论证后可以发现，我们唯一用到独立性假设的地方是应用观察 3.2。因此，要证明引理 5.1 在轻度相关图的情况下仍然成立，只需证明观察 3.2 在这种情况下也成立。具体而言，我们证明：若 $\mathcal{G}$ 是轻度相关随机图 G的一个随机实现，则 G 中未被 M 覆盖的边数至多为 $O(n/p)$。与之前一样，令 $H = G[V \setminus M]$ 为由顶点覆盖 M的补集诱导的子图。同样，H 中的边都不能在 $\mathcal{G}$ 中实现，否则 H 中的某个顶点必定属于顶点覆盖 M。令 E_1 和 E_2为根据定义 7.1 从图的边集 E 中得到的子集，我们定义 $H_1 = H \cap G[E_1]$和 $H_2 = H \cap G[E_2]$。注意，由于 $|E_2| = O(n)$，我们发现

$$\Pr[|H| \geq O(n/p)] = P[|H_1| + |H_2| \geq O(n/p)] \leq \Pr[|H_1| \geq O(n/p)] \leq (2/e)^n$$

在最后一个不等式中，我们对随机图 3.2 应用观察，该图的边是独立实现的 $G[E_1]$。因此， $|H|$ 在这种情况下仍然很小，可以直接应用引理5.1 ，从而得知我们只需采样 $O(n/\varepsilon p)$ 条边。

接下来，我们证明近似保证在这种情况下依然成立。第 5 节中其余的分析只需对算法 4稍作修改即可得出：当¯ ¯我们从 $\mathcal{H}$ 中采样实现时， H 在算法第 4 行，我们应以第 2 行得到的实现 $\mathcal{H}^\star$ 为条件进行采样。需要指出的是，我们的算法（算法 3）实际上并不需要这种能力也能运行；我们仅为分析之用才使用算法 4 。作出此修改后，算法 M 4 给出的顶点覆盖仍是从与真实最小顶点覆盖相同的分布中独立抽取的，而这正是后续分析所需的全部条件。因此，算法 3 实现了所需的近似比 1。$5+\varepsilon$ 且只需采样 $O(n/\varepsilon p)$ 条边。

□

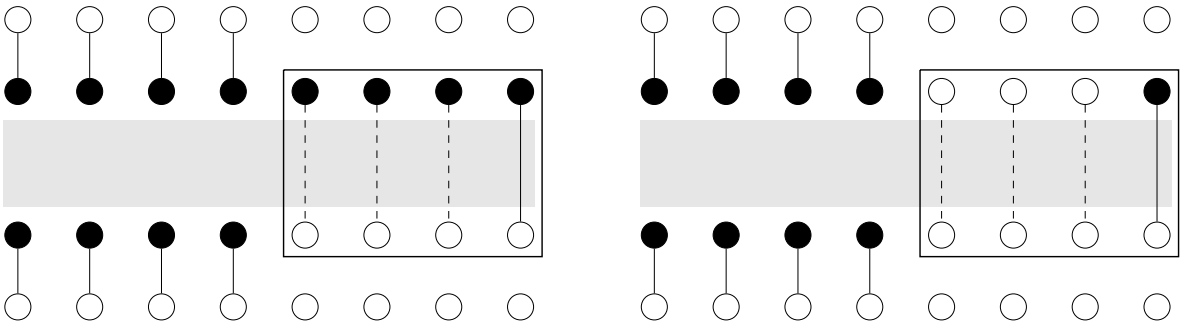

图 3： **的示意图**。中间的 $2n$ 个顶点（边由灰色矩形表示）构成 Ruzsa-Szemer´édi 图，其边彼此独立地实现。然后，我们从导出匹配中均匀随机地选取 M^* （方框标出），并实现所有不在 M^* 中的顶点对应的外部边。以高概率，任何仅以非自适应方式查询 $O(n)$ 条边的算法都必须覆盖 M^* 中几乎所有的边（如左图所示，覆盖中的顶点以黑色标出）；但只需覆盖 M^* 中实际实现的少数几条边（如右图所示）。

定理 1.2 （重述）。存在一个弱相关随机图 G ，对于该图，任何非自适应算法的近似比都至少为 $1.5-\varepsilon$ ，其概率为 $1-o(1)$。

证明。我们如下定义 G 。首先，令一个 (r,t)-Ruzsa-Szemer´edi 图为一个具有$2n$ 个顶点的二分图，其边集可划分为 t 个大小为 r 的导出匹配。对于 $r=\frac{n}{2}-\varepsilon_1$ 和 $t=n^{\Omega(1/\log\log n)}$ [GKK12]，这样的图是存在的。我们通过从这样一个图开始，并为 Rusza-Szemer`edi 图中的$2n$ 个顶点各添加一个额外顶点和一条外部边，来定义 $G$ 的基图。然后，对于边集中的所有边 $e$ ，我们分别以 $p_e=\varepsilon_2$ 实现 Rusza-Szemer`edi 图中的所有边。接下来，我们从这些导出匹配中随机选取一个，M_1、M_2、...、M_t，并将其称为 M^*。对于 Rusza-Szemer`edi 图中不参与 M^* 的 $O(n)$ 个顶点，我们实现其对应的外部边。显然，根据定义 7.1，这个随机图是弱相关的，其中 E_1 表示 Ruzsa-Szemer´edi 图中的边，而 E_2 表示外部边。

要证明每个非自适应算法的近似比都至少为 $1.5-\varepsilon$，注意到该算法最多只能查询 $O(n)=o(r\cdot t)$ 条边，因此根据一个简单的计数论证，算法查询的边集必须包含对所有

至多 $o(1)$ 比例的匹配 M_1，……，M_t。因此，以 $1-o(1)$ 的概率，该算法必须覆盖 $r-o(r)$ 条边于 M^* 中，并且还必须覆盖与不在 M^* 中的顶点对应的每条外部边。由于 M^* 中的边与外部边互不重合，因此，以 $1-o(1)$ 的概率，任何非自适应算法返回的顶点覆盖大小必定等于

$$\underbrace{2(n-r)}_{\#\text{ exterior edges that must be covered}} + \underbrace{r-o(r)}_{\#\text{ edges in } M^* \text{ that must be covered}} = 1.5n+\varepsilon_1-o(n)$$

然而，根据切尔诺夫界可知，M^*中实现超过 $2n\varepsilon_2$ 条边的概率上界为 $o(1)$。因此，以概率 $1-o(1)$，G 的最小顶点覆盖大小为

$$\underbrace{2(n-r)}_{\#\text{ exterior edges that must be covered}} + \underbrace{2n\varepsilon_2}_{\#\text{ edges in } M^* \text{ that must be covered}} = n(1+2\varepsilon_2)+\varepsilon_1$$

由此，根据并集界可知，以概率 $1-o(1)$，任何此类算法的近似比都必须为

$$\frac{1.5n+\varepsilon_1-o(n)}{n(1+2\varepsilon_2)+\varepsilon_1} \to \frac{1.5}{1+2\varepsilon_2} \geq 1.5-\varepsilon$$

当 n足够大时，并适当选择 ε_2（请注意，此参数仅取决于 ε，而不取决于 n）。

□

参考文献

[AB19] Sepehr Assadi 和 Aaron Bernstein。《匹配问题稀疏化统一理论探讨》。载于第二届算法简约性研讨会（*SOSA@SODA 2019*），*2019*年*1*月*8*日至*9*日，美国加利福尼亚州圣迭戈，第 11:1–11:20 页，2019 年。[ADSY12] Shipra Agrawal、Yichuan Ding、Amin Saberi 和 Yinyu Ye。随机优化中的相关性代价。《运筹学》，60(1):150–162，2012 年。[AKL16] Sepehr Assadi、Sanjeev Khanna 和 Yang Li。少量查询下的随机匹配问题。载于*2016*年 *ACM*经济学与计算会议论文集（*EC '16*），*2016*年*7*月*24*日至*28*日，荷兰马斯特里赫特，第 43–60 页，2016 年。[AKL17] Sepehr Assadi、Sanjeev Khanna 和 Yang Li。随机匹配问题：通过非自适应算法突破二分之一。载于*2017*年 *ACM*经济学与计算会议论文集（*EC '17*），*2017*年*6*月*26*日至*30*日，美国马萨诸塞州剑桥，第 99–116 页，2017 年。[AP13] Shabbir Ahmed 和 Dimitri J Papageorgiou。带相关性的概率集合覆盖。《运筹学》，61(2):438–452，2013 年。[BBD22] Soheil Behnezhad、Avrim Blum 和 Mahsa Derakhshan。少量查询下的随机顶点覆盖。载于*2022*年 *ACM-SIAM*离散算法年会论文集（*SODA*），第 1808–1846 页。SIAM，2022 年。

[BD20] Soheil Behnezhad 和 Mahsa Derakhshan。随机权重匹配：（$1-\varepsilon$）近似。载于计算机科学基础（*FOCS 20*），即将发表，2020 年。[BDF+19] Soheil Behnezhad、Mahsa Derakhshan、Alireza Farhadi、MohammadTaghi Hajiaghayi 和 Nima Reyhani。均匀稀疏图上的随机匹配。载于算法博弈论——第十二届国际研讨会（*SAGT 2019*），*2019*年*9*月*30*日至*10*月*3*日，希腊雅典，论文集，第 357–373 页，2019 年。[BDH+15] Avrim Blum、John P. Dickerson、Nika Haghtalab、Ariel D. Procaccia、Tuomas Sand- holm 和 Ankit Sharma。无知几近福祉：少量查询下近乎最优的随机匹配。载于第十六届 *ACM*经济学与计算会议论文集（*EC '15*），*2015*年*6*月*15*日至*19*日，美国俄勒冈州波特兰，第 325–342 页，2015 年。[BDH20a] Soheil Behnezhad、Mahsa Derakhshan 和 MohammadTaghi Hajiaghayi。少量查询下的随机匹配：（$1-\varepsilon$）近似。载于第*52*届 *ACM SIGACT*计算理论研讨会论文集（*STOC 2020*），*2020*年*6*月*22*日至*26*日，美国伊利诺伊州芝加哥，第 1111–1124 页，2020 年。[BDH+20b] Avrim Blum、John P. Dickerson、Nika Haghtalab、Ariel D. Procaccia、Tuomas Sand- holm 和 Ankit Sharma。无知几近福祉：少量查询下近乎最优的随机匹配。《运筹学》，68(1):16–34，2020 年。[BFHR19] Soheil Behnezhad、Alireza Farhadi、MohammadTaghi Hajiaghayi 和 Nima Reyhani。少量查询下的随机匹配：新算法与工具。载于第三十届 *ACM-SIAM*离散算法研讨会论文集（*SODA 2019*），*2019*年*1*月*6*日至*9*日，美国加利福尼亚州圣迭戈，第 2855–2874 页，2019 年。[BR18] Soheil Behnezhad 和 Nima Reyhani。少量查询下近乎最优的随机加权匹配。载于*2018*年 *ACM*经济学与计算会议论文集，*2018*年*6*月*18*日至*22*日，美国纽约州伊萨卡，第 235–249 页，2018 年。[BS15] Aaron Bernstein 和 Cliff Stein。二分图中的全动态匹配。载于自动机、语言与编程——第*42*届国际学术讨论会（*ICALP 2015*），*2015*年*7*月*6*日至*10*日，日本京都，论文集，第一部分，第 167–179 页，2015 年。[GKK12] Ashish Goel、Michael Kapralov 和 Sanjeev Khanna。最大二分图匹配的通信复杂度与流式复杂度。载于第二十三届 *ACM-SIAM*离散算法研讨会论文集（*SODA 2012*），*2012*年*1*月*17*日至*19*日，日本京都，第 468–485 页，2012 年。[GKMR11]Anupam Gupta、Ravishankar Krishnaswamy、Marco Molinaro 和 Ramamoorthi Ravi。相关背包问题与非鞅赌博机的近似算法。载于*2011*年 *IEEE* 第*52*届计算机科学基础年会，第 827–836 页。IEEE，2011 年。[GV04] Michel X. Goemans 和 Jan Vondr´ak。随机子图的最小生成树覆盖。载于第十五届 *ACM-SIAM*离散算法研讨会论文集（*SODA 2004*），*2004*年*1*月*11*日至*14*日，美国路易斯安那州新奥尔良，第 934–941 页，2004 年。

[GV06] Michel X. Goemans 和 Jan Vondr´ak。随机子图的最小生成树覆盖。随机结构算法，29(3):257–276，2006年。[Von07] Jan Vondr´ak。随机子图的最短路径度量近似。随机结构算法，30(1-2):95–104，2007年。[YM18] Yutaro Yamaguchi 和 Takanori Maehara。少量查询下的随机装箱整数规划。载于第二十九届 *ACM-SIAM* 离散算法年度研讨会论文集，*SODA 2018*，美国路易斯安那州新奥尔良，*2018*年*1*月*7*日至*10*日，第293–310页，2018年。