arXiv:2601.11755v1 [cs.DS] 16 Jan 2026

带分区和公平性约束的子模覆盖双准则算法

Wenjing Chen Yixin Chen jj9754@tamu.edu
chen777@tamu.edu 计算机科学与工程 计算机科学与工程 德克萨斯农工大学 德克萨斯农工大学

Victoria G. Crawford
vcrawford@tamu.edu 计算机
科学与工程 德克萨斯农工大学

摘要

在许多次模优化应用中，数据集天然地划分为互不相交的子集。这类场景催生了带有基于分区约束的次模优化问题，其中理想的解集应在某种意义上实现均衡、公平，或在各分区间满足资源约束。尽管现有关于次模覆盖的研究在很大程度上忽视了这一结构，我们在此首次对带分区约束的子模覆盖（SCP）问题及其主要变体展开全面研究。我们的主要贡献是针对这些 NP 难优化问题，开发并分析适用于单调和非单调目标函数的可扩展双准则近似算法。值得注意的是，为单调情形提出的算法在显著降低查询复杂度的同时，实现了最优近似保证。最后，在真实世界数据集和合成数据集上的实证评估进一步验证了所提算法的效率和有效性。

1 引言

次模优化算法已成为现代机器学习的基石，推动了众多重要应用的发展。从为大语言模型 [构建高质量的预训练和微调数据集（Ji et al., 2024, Kumari et al., 2024, Agarwal et al., 2024]），到支持多样化的在线推荐系统（ [Hiranandani et al., 2020, Chen 和 Crawford, 2025], ）、机器人学中的多智能体优化（[Zhou 和 Tokekar, 2022, Xu 和 Tzoumas, 2024], ），以及实现计算机视觉中的精确图像归因（ [Chen 等人, 2024a]），次模函数都发挥着重要作用。直观而言，次模函数满足边际收益递减性质，机器学习中许多基础优化问题的目标函数都具有这一性质。形式化地，设 $f\colon\ 2^U \to \mathbb{R}$ 定义在大小为 n的底集 U 的子集上。若对所有 $A \subseteq B \subseteq U$ 和 $x \notin B$，$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$，则函数 f 为次模函数。此外，若对每个 $X \subseteq Y \subseteq U$，$f$满足 $f(Y) \geq f(X)$，则称其为单调函数。

次模覆盖（SC）问题是一类重要的优化问题，具有广泛的应用 [Iyer 和 Bilmes，2013 年，Chen 和 Crawford，2024年，Crawford 等人，2019年，Mirzasoleiman 等人， 2015]。在经典形式中，次模覆盖的目标是找到一个成本最小的子集 $S \subseteq U$，使得 $f(S) \geq \tau$，其中成本函数通常是基数或某种可加成本。现有关于 SC 的研究利用了其与次模最大化之间的关系 [Iyer 和 Bilmes，2013 年，Chen 和 Crawford， 2024], ，后者旨在求解 $\arg\max\{f(S) : c(S) \leq \kappa\}$。例如， [Chen 和 Crawford，2024年，Iyer 和 Bilmes， 2013]提出了转换算法，可将任意用于次模最大化的双准则算法转换为用于 SC 的算法。具体而言，用于 SC 问题的 (α, β)-双准则近似算法返回一个解 X，使得 $|X| \leq \alpha|OPT|$且 $f(X) \geq \beta\tau$。

然而，这些经典表述的一个重大局限在于，它们无法对一些关键应用进行建模：在这些应用中，底集 U被划分为互不相交的组 U_1、.....、U_N，目标是找到一个在每个分区内均满足预算限制的子集，或者找到一个在各分区之间均衡或公平的子集。我们进一步阐述了

带分区约束的次模覆盖设定及其若干应用。在视频摘要中 [Mirzasoleiman 等人， 2018], 集合 U 中的元素是帧，每帧都对应视频中 N 个连续时间区域之一。构造子模函数 f 来衡量帧子集 X 对整个视频 U [的摘要效果 2014]Tschiatschek 等人，。目标是找到一个足够好的帧摘要子集，即 $f(X) \geq \tau$，同时限制解中来自各时间区域的帧所占比例，即 $c(X \cap U_i) \leq p_i v$其中$p_j \in [0,1]$ 且 v 是成本预算。作为第二个例子，考虑影响力最大化 [Tschiatschek 等人，2014], 其中，用户底集 U 可以划分为 N 个分区，以反映语言等人口统计信息。为了选择一个在不同分区间分布均衡的子集，我们施加公平性约束，其中 $p_j|S| \leq |S \cap U_j| \leq q_j|S|$。然后，目标是找到基数最小的公平解，使得 $f(S) \geq \tau$。文献中还有许多其他应用，包括神经网络剪枝 [Chen 等人， 2025],用于学习的高质量数据选择 [Killamsetty 等人， 2021], 以及数据摘要 [El Halabi 等人， 2020]。

尽管该问题十分重要且应用广泛，相关研究仍然有限。Chen 等人 [2025]研究了带有分区约束这一特殊情形的次模覆盖问题，即带公平性约束的次模覆盖。然而，他们提出的离散方法只能达到次优近似比，而连续方法的查询复杂度则高得令人难以承受。相比之下，对于带公平性约束的 SC 问题，我们的方法以显著更低的查询复杂度实现了最优双准则近似比。

在本文中，我们研究了若干不同的、约束定义在全集 U 的分区上的次模覆盖问题，其中包括但不限于带公平性约束的次模覆盖。我们采用通用的转换框架，设计转换算法，将次模最大化算法转化为次模覆盖算法。具体而言，为了构造目标值更接近目标阈值 τ 的解，我们提出了带分区约束的次模最大化双准则算法。值得注意的是，传统贪心算法每次依据边际增益添加一个可行元素，而我们的方法则在每一轮逐步选取元素块；每个元素块都符合不同分区间的成本分布，从而确保元素的选择与预算成本成比例。在次模覆盖问题中，这种块贪心策略尤为有效，因为要达到接近阈值 τ 的目标值，可能需要选择超出标准次模最大化算法可行性限制的集合。具体而言，我们的贡献总结如下。

1. 在第 2.1 节中，我们研究带分区约束的子模覆盖（SCP）问题，即 $\arg\min_{S \subseteq U}\{v : f(S) \geq \tau, |S \cap U_j| \leq p_j v, \ \forall j \in [N]\}$其中 f为非单调。我们首先提出一种通用的转换算法，将带分区约束的次模最大化（SMP）对偶问题的任意随机算法转换为 SCP 算法。通过为 SMP 提出一种双准则算法，我们可以得到一种用于非单调 SCP 的算法，其双准则近似比为（$O(\frac{(1+\alpha)}{\epsilon})$ $1/e - \epsilon$）。

2. 第 2.2 节讨论带背包分区约束的单调子模覆盖（SCKP）问题，即寻找 $\arg\min_{S \subseteq U}\{v : f(S) \geq \tau, c(S \cap U_j) \leq p_j v, \ \forall j \in [N]\}$。我们首先为背包分区约束下的次模最大化（SMKP）对偶优化问题设计一种算法，该算法采用分块贪心结构。通过利用转换过程，我们为 SCKP 实现了（$\frac{(1+\alpha)}{\ln 1/\epsilon \ln 2}$， $1-\epsilon$）双准则近似比。

3. 第 2.3 节讨论了带公平性约束的单调次模覆盖问题（SCF），该问题由 Chen 等人 [2025] 最近提出。所提出的算法实现了近乎最优的近似比（$\mathcal{O}(\ln(1/\epsilon))$ $1-\epsilon$）。这与 Chen 等人 [2025], 的算法的近似比相同，但他们的方法是连续方法，需要 $\mathcal{O}(\frac{n^2(1+\alpha)\log^2(\frac{n}{\varepsilon})\log_{\varepsilon^4}\alpha}{n})$ 次查询，而我们的方法的查询复杂度仅为 $\mathcal{O}(\frac{n}{\log(n)\kappa\ln(1/\epsilon)\epsilon})$。

最后，我们对用于非单调 SCP、单调 SCKP 和单调 SCF 的算法进行了实验评估。结果表明，与基线算法相比，我们提出的非单调 SCP 算法能够取得更高的函数值；而 SCKP 算法则在成本预算方面有所改进。此外，就解集大小和公平性差异而言，我们的 SCF 算法优于 Chen 等人提出的其他算法。 [2025]

1.1 相关工作

在次模最大化中，拟阵约束是一类基本且研究充分的可行性约束，而分区约束则是其中尤为重要、应用广泛的一类特殊情形。 [El Halabi 等人，2020，Chen 等人， 2025]因此，可以采用针对一般拟阵约束下次模最大化问题的算法；这类算法已得到广泛研究。 [Nemhauser 等人，1978，Fisher 等人，1978，Calinescu 等人，2011，Badanidiyuru 和 Vondr´ak，2014，Chekuri 和 Quanrud，2019，Buchbinder 和 Feldman，2024a年]。单调子模最大化在拟阵约束下目前已知的最佳近似比为 $1 - 1/e$ [Calinescu 等人，2011，Buchbinder 和 Feldman， 2018]。对于更一般的拟阵约束下非单调子模函数最大化问题，目前已知的最佳困难性结果为 0。478 [Gharan 和 Vondr´ak，2011，Qi， 2024]。目前近似比最佳的算法是一种连续方法，其近似比为0.401 [Buchbinder 和 Feldman，2024b年]。近似比最佳的组合算法由 Chen 等人提出， [2024b], 该算法在 $O\left(k^5 \log(k) n/\epsilon\right)$ 次查询中可达到 $0.305 - \epsilon$ 近似保证。分区类型的约束广泛存在于次模优化应用中，但尽管如此，专门为这类约束设计的算法仍鲜有研究。一个例外是，公平性约束近年来受到关注 [El Halabi 等人，2020，2023，Chen 等人， 2025]。El Halabi 等人证明，在公平性约束下最大化单调子模函数，可转化为拟阵约束下的单调次模最大化问题。

背包分区约束下的次模最大化（SMKP）问题是 SCKP 的对偶问题，定义为 $\arg\max\{f(S) : \sum_{s\in X\cap U_j} c(s) \leq p_j v, \ \forall j \in [N]\}$。当全集中只有一个分区时，该问题可视为受单个背包约束的次模最大化问题的一般化 [Amanatidis 等人，2020年，Cui 等人， 2025] 。近期研究还探讨了同时受背包约束和分区约束的次模最大化问题 [Cui 等人，2024年，Li 等人， 2025]。具体而言，这些论文研究了满足以下条件的公平性感知次模最大化：（i）对选定子集在所有分区上的总成本施加背包约束；（ii）在每个分区内施加基数约束，以确保公平代表性。相比之下，SMKP 施加的是逐分区背包约束（而非单个全局背包约束），且不设基数要求。

对于目标函数取整数值的经典次模覆盖问题，标准贪心算法——重复选择边际增益最高的元素，直至目标达到阈值 τ——可实现 $O(\log \max_{e\in U} f(e))$ [Wolsey，1982] 的近似比。对于实值次模函数，一种常见的改进做法是在函数值达到 $(1-\epsilon)\tau$ 时停止，由此得到具有 $(\ln(1/\epsilon)\ 1-\varepsilon)$ 双准则近似比的算法 [Krause 等人，2008年，Chen 和 Crawford，2024]。对于公平子模覆盖（FSC） [Chen 等人， 2025] 问题，Chen 等人的离散算法实现了 $(\mathcal{O}(1/\epsilon)\ 1-\epsilon)$ 的双准则近似比，而我们的算法将近似比改进为 $(\mathcal{O}(\ln\ 1/\epsilon)\ 1-\epsilon)$，与 Chen 等人提出的连续方法的近似比相同，但所需查询次数大幅减少。

2 算法与理论分析

现在我们介绍本文的主要结果[1]。首先，我们在第 2.1 节讨论带分区约束、未必单调的次模覆盖一般情形。接着，我们在第 2.2 节研究带分区约束的单调子模覆盖；我们的结果甚至适用于更一般的背包成本情形。最后，我们在第 2.3 节探讨限制更多但具有许多有趣应用的公平子模覆盖。

我们所有结果的核心，是针对子模最大化问题提出的一种新颖算法框架。该框架通过将贪心过程分块执行来实现双准则近似比，其中每个块都是一个可行子集。这种分块贪心策略不同于以往侧重于分区约束拟阵结构的方法。相比之下，我们的方法利用了分区约束与基数约束之间的内在联系，从而改善了查询复杂度和近似比。在全文中，我们将向集合 $S \subseteq U$ 添加元素 $u \in U$ 的边际增益定义为 $\Delta f(S,\ u) = f(S \cup u) - f(S)$。此外， OPT用于表示子模优化实例的最优解，具体含义由上下文决定。

[1]我们在附录的一张表中总结了结果。请参阅表 1。

2.1 带分区约束的非单调次模覆盖

本节中，我们考虑一般的非单调带分区约束的子模覆盖（SCP）问题，其目标是找到一个集合 $S \subseteq U$，使其满足

$$\begin{aligned} \min_{S \subseteq U} \quad & v \\ \text{s.t.} \quad & |S \cap U_j| \leq p_j v, \quad \forall j \in [N], \\ & f(S) \geq \tau. \end{aligned}$$

v 表示分配给各分区集合的预算，我们的目标是在确保 f 足够高的同时最小化该预算。 p_j 表示分配给第 j个分区的预算比例。一般性不失，我们假设 $\sum_{j\in[N]} p_j = 1$。如果全集中只有一个分区，即 $N = 1$，那么 v 的最优值为 $|S|$，此时我们就得到经典的次模覆盖问题。为了进一步阐释问题定义，请考虑第 1 节所述的视频摘要应用示例，其中帧按场景或内容类型分组，且成本一致。这意味着目标是在确保摘要质量足够高的同时，找到总预算最小且在不同分区间分配均衡的解。

在我们的第一个结果中，我们利用次模覆盖与次模最大化之间的关系，提出了一种转换算法 convert-rand，它可以将任何用于带分区拟阵约束的非单调子模最大化（SMP）的随机双准则算法转换为用于非单调 SCP 的双准则算法。具体而言，输入预算为 v的 SMP 定义为 $\max\{f(S) : |S\cap U_i| \leq p_i v, \forall i \in [N]\}$。下面我们将正式定义 SMP 和 SCP 的双准则近似概念。

定义 2.1。 一种针对输入预算为 v 的 SMP 的 (α, β)-近似算法，返回一个满足以下条件的解 X。

$$\begin{aligned} & f(X) \geq \alpha f(OPT), \\ & |X \cap U_j| \leq \beta p_j v, \end{aligned}$$

其中 OPT 是 SMP 的最优解，即 $OPT := \arg\max\{f(S) : |S \cap U_i| \leq p_i v, \ \forall i \in [N]\}$。

定义 2.2. 一个用于 SCP 的 (α, β)-近似算法会返回一个满足以下条件的解 X，其目标值为 v_X

$$\begin{aligned} & v_X \leq \alpha v_{OPT}, \\ & |X \cap U_j| \leq p_j v_X \\ & f(X) \geq \beta\tau. \end{aligned}$$

这里， OPT是 SCP 的最优解，即 $OPT := \arg\min\{v: |S \cap U_j| \leq p_j v, \forall j \in [N], f(S) \geq \tau\}$。 v_{OPT}是 SCP 的最优值。

请注意，Chen 和 Crawford [2024], 提出了一种算法，可在单调子模目标函数的情况下，将随机子模最大化算法转换为次模覆盖算法。事实上，这种情况下的一个关键挑战是：需要通过反复调用子模最大化子程序并应用集中不等式，确保函数值 f 具有高概率保证。为减少预言机查询次数，Chen 和 Crawford [2024]中的算法应用马尔可夫不等式，并在整个转换算法中使用截断目标函数 $f_\tau := \min\{\tau, f\}$。然而，仅当 f为单调函数时， f_τ才具有子模性。相比之下，我们的分析通过避免使用截断目标函数，并在应用集中不等式时采用更精细的分析，扩展到了非单调情形。具体而言，我们分析随机变量 $\beta f(OPT_g) - f(S_i)$ 的偏差，其中 S_i是用于 SMP 的随机子程序算法输出的解集。

现在我们介绍 convert-rand 及其理论保证。convert-rand 的伪代码见补充材料 B 节中的算法 4。convert-rand 通过迭代猜测最优预算 v 的值来运行。对于每个猜测值，convert-rand 会在多次独立试验中运行相应的双重次模最大化算法。convert-rand 的理论保证见定理 2.3。分析推迟至补充材料的 B 节。

算法 1 nonmono-bi

输入： ϵ，分区约束参数，总预算 v
输出： $S \subseteq U$
对于 $i = 1$ **到** $\frac{2}{\epsilon}$ **do**
  对于 $j = 1$ **到** N **执行**
    对于 $l = 1$ **到** $p_j v$ **执行**
      令 $M \subseteq U_j / S$ 为一个大小为 $\frac{2p_j v}{\epsilon}$、使 $\sum_{x \in M} \Delta f(S, x)$ 最大化的集合。
      $u \leftarrow$ 从 M 中均匀随机采样一个元素
      $S \leftarrow S \cup \{u\}$

定理 2.3。对于任何运行时间为 $\mathcal{T}(n)$、且其保证 γ 仅在期望意义下成立的非单调 *SMP* 随机(γ, β)-双准则近似算法，都可以将其转换为非单调 *SCP* 的近似算法；该算法以至少 $1 - \delta$ 的概率成为一个运行时间为 $O(\log_{1+\alpha}(|OPT|) \ln(1/\delta) \mathcal{T}(n) / \ln(\frac{\beta - \gamma + \epsilon}{\beta - \gamma}))$ 的$((1 + \alpha)\beta, \gamma - \epsilon)$-双准则近似算法。

由于目前针对分划拟阵上的非单调子模最大化问题，已知最优的算法结果是 Chen 等人提出的算法 [2024b],，其近似比为 $0.305 - \epsilon$。因此，将定理 2.3 应用于 Chen 等人 [2024b], 的随机化算法，我们以高概率得到一个用于 SCP 的 $(1+\alpha, 0.305-\epsilon)$-双准则近似算法，其查询次数为 $O(n|OPT| \log_{1+\alpha}(|OPT|) \ln(1/\delta)/\ln(1+\epsilon))$。然而，$\tau$ 的 $0.305 - \epsilon$ 倍与可行解相差甚远，因此自然会产生一个问题：是否存在能够提供更好可行性保证的算法？一个相关的重要结果 [Crawford, 2023] 表明，对于非单调子模覆盖问题，可行性因子优于 $1/2$ 是不可能的。由于该问题是 SCP 的特殊情形，这一结果对 SCP 同样成立。尽管如此，我们仍不确定是否存在近似比处于 0.305 到 0.5 之间的可扩展算法。

在本节余下部分，我们将介绍一种可扩展的算法 nonmono-bi，该算法可以输出任意接近 τ/e 的解集。nonmono-bi 的伪代码见算法 1。nonmono-bi 采用了在分块贪心算法中逐步构建解的思想，并实现了如下的双准则近似比。

定理 2.4。假设 *nonmono-bi* 在预算为 v 的非单调 *SMP* 实例上运行，则 *nonmono-bi* 输出一个解 S，其期望双准则近似比为 $(1/e - \epsilon, \frac{2}{\epsilon})$，并且查询数量至多为 $O(\frac{nv}{\epsilon})$。

分析和证明推迟至补充材料的 B 节。根据这些结果，我们可以得到：将 nonmono-bi 作为子程序用于 convert-rand，可为非单调 SCP 产生一个 $(\frac{2(1+\alpha)}{\epsilon}, 1/e - 2\epsilon)$-双准则近似算法。该算法运行时的查询数量为 $O(\frac{n|OPT|}{\log_{1+\alpha}(|OPT|) \ln(1/\delta) \epsilon \ln(1+\epsilon^2)})$。

2.2 带背包分区约束的单调子模覆盖

现在，我们考虑带背包分区约束的单调子模覆盖问题（SCKP）。SCKP 刻画了这样一种情形：我们希望在不同分区之间平衡成本，而底集中不同元素的成本并不相同。下面举例说明 SCKP。考虑深度学习中的一项神经网络训练任务，我们希望从 N 个不同的预定义组中选取预训练数据点，目标是在预算成本最低的情况下选出一个数据点子集，其中每个数据点的成本可能反映计算、标注或存储开销。同时，我们希望解在数据集的预定义组之间实现均衡的成本分配，并确保子模效用函数（例如，多样特征覆盖）达到指定阈值 τ。

更正式地说，SCKP 的定义如下。定义成本函数 $c: U \to \mathbb{R}_{\geq 0}$，并令 $c(S) = \sum_{x \in S} c(x)$ 对任意子集 $S \subseteq U$ 均成立。那么 SCKP 为：

$$\begin{aligned} &\min v \\ &\text{s.t. } f(S) \geq \tau \\ &\quad c(S \cap U_j) \leq p_j v, \qquad \forall j \in [N]. \end{aligned} \tag{1}$$

算法2 greedy-knapsack-bi

1: **输入：** ϵ，一个SMKP实例
2: **输出：** 解集 $S \subseteq U$
3: **对于** $i = 1$**到** $\frac{\ln \frac{1}{\epsilon}}{\ln 2}$ **do**
4: **对于** $j = 1$**到** N **执行**
5: $A \leftarrow \emptyset, B_j \leftarrow p_j v$。
6: **当真时执行**
7: $s \leftarrow$ 取最大值$_{x \in U_j / S,\ c(x) \le B_j} \frac{\Delta f(S \cup A,\ x)}{c(x)}$
8: $A \leftarrow A \cup \{s\}$
9: **如果** $c(A) \ge B_j$ **则**
10: $S \leftarrow S \cup A$
11: **中断**
12: **返回** S

在 SCKP 的定义中， v 表示总成本的预算。更具体地说， v 是总成本的上限。第二个约束确保每个分区 U_j 内的解集 S 的成本不超过指定比例 p_j 的 v。不失一般性，我们假设 $\sum_{j \in [N]} p_j = 1$。

这种形式自然地出现在各种现实应用中，包括社交网络分析中的影响力最大化（激活不同节点会产生不同成本）、深度学习中的预训练数据选择（不同数据点可能需要不同的计算或内存成本），以及多智能体系统中的任务分配。关于这些启发性示例的详细讨论，请参见附录 C.1。接下来，我们将预训练数据选择作为 SCKP 问题的一个启发性示例进行讨论。

为解决 SCKP，我们首先针对带背包分区约束的子模最大化（SMKP）这一对偶问题提出一种算法，其定义为 $\arg\max\{f(S) : \sum_{s \in X \cap U_j} c(s) \le p_j v,\ \forall j \in [N]\}$。请注意，当成本均匀时，即 $c(s) = c$, $\forall s \in U$。SMKP 是一个带分区拟阵约束的单调子模最大化问题。因此，我们可以应用任何带拟阵约束的子模最大化算法。然而，标准算法的输出无法达到任意接近最优值的目标值。因此，我们提出 greedy-knapsack-bi 算法，该算法分为 $\phi := \ln \frac{1}{\epsilon} \ln 2$ 个区块依次执行。greedy-knapsack-bi 的伪代码见算法2。在第 4 行至第 11 行的每个区块中，算法遍历底集 U 中的每个分区 U_j，并以贪心方式添加边际增益密度最高的元素。下面给出 greedy-knapsack-bi 求解 SMKP 的理论保证。

定理 2.5. 假设对 *SMKP* 的一个实例运行算法*2*中所述的 *greedy-knapsack-bi*，则 *greedy-knapsack-bi* 输出的解集满足

$$f(S) \ge (1 - \epsilon) f(OPT)$$
$$c(S \cap U_j) \le \frac{2 \ln \frac{1}{\epsilon}}{\ln 2} p_j v, \qquad \forall j \in [N],$$

其中 OPT 是 *SMKP* 的最优解。

该定理保证，greedy-knapsack-bi 返回的解集能够取得任意接近最优值的目标值，同时成本约束的违反因子不超过$2 \ln \frac{1}{\epsilon} \ln 2$。在统一成本这一特殊情况下，该定理意味着 greedy-knapsack-bi 可达到 $(1 - \epsilon, O(\ln \frac{1}{\epsilon}))$ 的双准则近似保证。这与单调子模最大化在基数约束下已知的最佳近似比相匹配。
greedy-knapsack-bi 的性能提升源于其分块结构，该结构有效利用了基数约束下的子模最大化问题与分区拟阵约束之间的内在相似性。定理 2.5 的证明与分析见附录C.3。

2.2.1 SCKP 转换定理

为了将任意适用于 SMKP 的双准则算法转换为适用于 SCKP 的双准则算法，我们提出并分析了转换算法，记为 convert。convert 的伪代码见算法 5，其理论

保证在附录的 C.4 节中的定理 C.3 中概述。利用转换定理中的结果，我们可以得到以下推论。

推论 2.6。 通过将 *greedy-knapsack-bi* 作为转换算法 *convert* 的子程序，我们可以获得一个适用于 *SCKP* 的算法，该算法返回一个满足以下条件的解集 S 和 v_S

$$
\begin{aligned}
&v_S \le \frac{2(1+\alpha)\ln(1/\epsilon)}{\ln 2} v_{OPT} \\
&f(S) \ge (1-\epsilon)\tau \\
&c(S \cap U_j) \le p_j v_S
\end{aligned}
$$

其中， OPT 和 v_{OPT} 分别是式 *(1)* 中定义的问题 *SCKP* 的最优解集和最优值。该算法的总运行时间上界为 $\mathcal{O}\left(n^2 \log_{1+\alpha}\left(\frac{c_{\max} n}{c_{\min}}\right)\right)$。其中， $c_{\max}$ 和 $c_{\min}$ 分别是单个元素成本的最大值和最小值。

2.3 带公平性约束的次模覆盖

在本节中，我们考虑 Chen 等人最近提出的带公平性约束的单调子模覆盖问题（SCF） [2025]（在 Chen 等人的工作中也称为公平子模覆盖 [2025]）。SCF 定义如下：给定一个单调子模函数 f、一个阈值 τ，以及每个组中元素比例限额的上下界 p_c 和 q_c，SCF 旨在寻找

$$
\begin{aligned}
& \operatorname{argmin}_{S \in U} |S| \\
s.t. \quad & p_c|S| \le |S \cap U_c| \le q_c|S|, \qquad \forall c \in [N] \\
& f(S) \ge \tau.
\end{aligned}
$$

Chen 等人提出了一种转换方法，该方法将针对带公平性约束的子模最大化（SMF）对偶问题的双准则算法 [El Halabi 等人， 2020] 转换为 SCF 算法。形式上，SMF 的目标是最大化 $f(S)$，同时满足约束 $S \in \mathcal{M}_{fair}$。 $\mathcal{M}_{fair}$ 表示公平拟阵，其定义为 $\mathcal{M}_{fair} = \{S \subseteq U\colon \ |S \cap U_c| \le u_c, \ \forall c \in [N], \ \sum_{c\in[N]} \max\{|S \cap U_c|, l_c\} \le k\}$. 如果我们对所有 $c \in [N]$ 都设置$l_c = 0$，并且设置 $k = \sum_{c\in[N]} u_c$，那么 $\mathcal{M}_{fair}$ 等价于 $\mathcal{M}_{fair} = \{S \subseteq U\colon \ |S \cap U_c| \le u_c, \ \forall c \in [N]\}$. 因此，SMF 可以视为带分区拟阵约束的次模最大化的一种广义形式。SMF 的双准则近似保证定义如下。

定义 2.7。 对于 SMF，一个具有（α， β）-双准则近似比的离散算法返回一个解 X 使得

$$
\begin{aligned}
& f(X) \ge \alpha f(OPT), \\
& |X \cap U_c| \le \beta u_c \qquad \forall c \in [N], \\
& \sum_{c\in[N]} \max\{|X \cap U_c|, \beta l_c\} \le \beta k.
\end{aligned}
$$

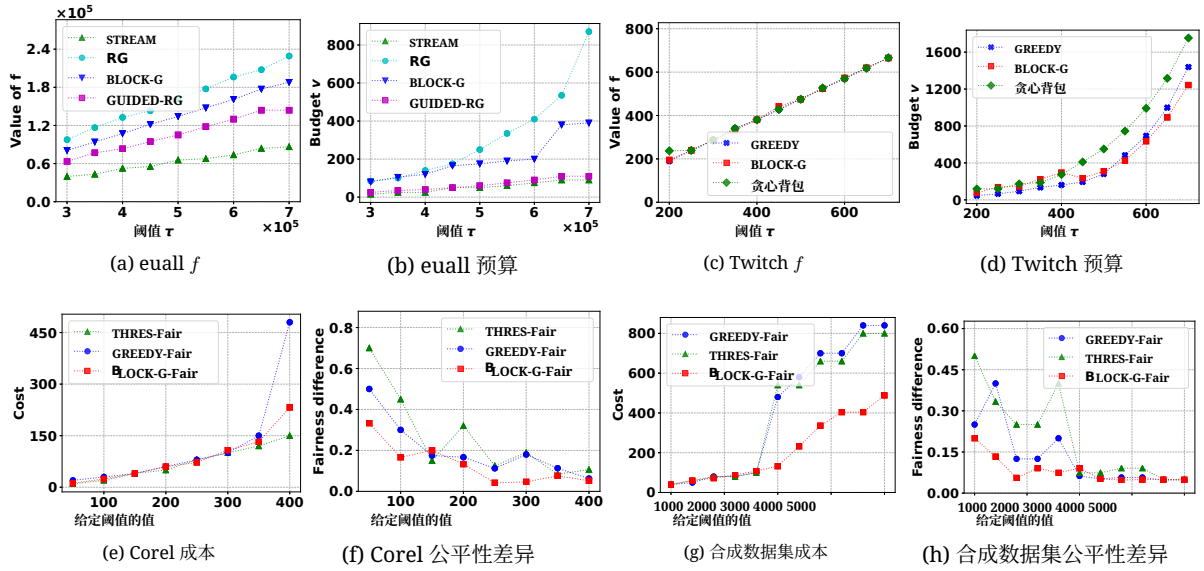
这里 OPT是问题 SMF 的最优解，即 $OPT = \arg\max_{S\in\mathcal{M}_{fair}} f(S)$。

因此，为了利用 Chen 等人的转换方法，我们提出了一种称为 Block-Fair-Bi 的 SMF 算法，该算法采用贪心分块技术。Block-Fair-Bi 改进了 Chen 等人在算法5中的 greedy-fairness-bi [2025]。这为 SCF 带来了（$\frac{1}{\ln \frac{1}{\epsilon}\ln 2} + 1$， $1-\epsilon$）的双准则近似保证，相较于 Chen 等人已知的离散算法最佳结果（$\frac{1}{\varepsilon} + 1$， $1 - \mathcal{O}(\varepsilon)$），有了显著改进。

我们现在介绍 Block-Fair-Bi，其伪代码见算法3。Block-Fair-Bi 采用了类似的方法，在各个区块中运行贪心算法。根据定义，公平性拟阵约束 $\mathcal{M}_{fair}$ 的 β-扩展为 $\mathcal{M}_\beta := \{S \subseteq U\colon |S \cap U_c| \le \beta u_c, \forall c \in [N], \sum_{c\in[N]} \max\{|S \cap U_c|, \beta l_c\} \le \beta\kappa\}$，如 Chen 等人 [2025] 所介绍。因此，对于 SMF问题，近似比为 (α, β) 的双准则近似算法，输出的解集属于 $\mathcal{M}_\beta$。Block-Fair-Bi 的核心直觉是， $\mathcal{M}_\beta$ 中的任意集合

算法3 Block-Fair-Bi

输入： 公平性参数
输出： $S \in U$
$S \leftarrow \emptyset$
对于 $i = 1$，**直到** $\frac{\ln(1/\epsilon)}{\ln 2}$ **do**
  $B \leftarrow \emptyset$
  当 $\exists s$ **满足** $B \cup \{s\} \in \mathcal{M}_{fair}$ **执行**
    $x \leftarrow$ 参数最大值$_{s \in U,\ B \cup \{s\} \in \mathcal{M}_{fair}} \Delta f(S,\ s)$
    $B \leftarrow B \cup \{x\}$
    $S \leftarrow S \cup \{x\}$
返回 S



(a) euall f (b) euall 预算 (c) Twitch f (d) Twitch 预算

(e) Corel 成本 (f) Corel 公平性差异 (g) 合成数据集成本 (h) 合成数据集公平性差异

图 1：在 euall 数据集、Twitch 数据集、Corel5k 数据集和合成数据集上运行算法的实验结果。预算：$\max_{i \in [N]} {}^{c(S \cap U_i)}\ p_i$ 。成本：解的大小。公平性差异：$(\max_c |S \cap U_c| - \min_c |S \cap U_c|)/|S|$。

$\mathcal{M}_\beta$ 中的集合都可以表示为 β 个子集的并集，且每个子集都属于 $\mathcal{M}_{fair}$。这启发我们将解集的容量划分为 β 个区块。这里 $\beta := \ln(1/\epsilon)\ \ln 2$ 。在算法3第 5 行至第 9 行的每个区块中，算法以贪心方式运行：在保持 $B \in \mathcal{M}_{fair}$ 的同时，迭代地加入边际增益最高的元素。主要理论结果陈述于下文的定理 2.8 中。该定理的证明推迟至附录 D。

定理 2.8。 假设对一个 *SMF* 实例运行 *Block-Fair-Bi*，则 *Block-Fair-Bi* 输出一个解 S，该解满足 $(1-\varepsilon,\ \ln \frac{1}{\varepsilon} \ln 2)$-双准则近似保证，且最多需要 $O(n\kappa \ln(1/\varepsilon))$ 次查询。

3 实验

本节对我们提出的算法进行实证评估。具体而言，我们在第 3.1 节中评估 nonmono-bi 算法在图割实例上的表现。接着，我们分别在第 3.2 节和第 3.3 节中，评估 greedy-knapsack-bi 在集合覆盖实例上的表现，以及 Block-Fair-Bi 在最大覆盖和图像摘要任务上的表现。有关这些应用、实验设置和结果的更多细节，请参阅补充材料中的 F 节。

3.1 非单调 SCP 实验

我们在社交网络数据上的若干图割实例中评估算法的性能。正文中使用的数据集是来自 SNAP 大型网络数据集[的 email-EuAll 数据集（n = 265214，420045 条边） 2016]Leskovec 和 Sosič。我们将 convert-rand 算法返回的解与四种子程序进行比较，包括：(i) 我们的 nonmono-bi 算法（“BLOCK-G”）；(ii) Feldman 等人的流式算法[2018] （“STREAM”）；(iii) Chen 等人的随机化算法 [2024b], ，并以 Han 等人提出的双贪心解作为初始解 [2020] （“GUIDED-RG”）；(iv) 用预算输入猜测值 κ [作为基数约束的子模最大化问题的随机贪心算法 2014] Buchbinder 等人（“RG”）。评估指标包括解 S返回的函数值 $f(S)$、查询复杂度、满足分区约束的最小预算值 v，即 $\max_{i\in[N]} \frac{|S\cap U_i|}{p_i}$，以及执行时间。我们还比较了不同阈值 τ下的结果。

关于 f 和最小预算的结果如图 1(a) 和 1(b) 所示。从结果可以看出，RG 和 BLOCK-G 始终比其他方法取得更高的目标值 f。这与我们的理论结果一致，因为这两种算法在函数值上的近似比满足$f(S) \geq (1/e-\epsilon)\tau$和 $f(S) \geq (1/e-2\epsilon)\tau$，而另外两种算法 STREAM 和 GUIDED-RG 在函数值 $1/0.583-\epsilon$、$0.305-2\epsilon$ 上的近似比则较差。不过，就预算而言，可以看到 RG 算法表现不佳，因为它没有考虑分区约束，导致预算分配不均衡。STREAM 和 GUIDED-RG 算法返回的解预算较小，因为这两种算法都实现了双准则近似比 $v_S \leq (1+\alpha)v_{OPT}$。BLOCK- G 的预算则较高，这是因为其双准则保证较弱，为 $v_S \leq {}^{2}_{(1+\alpha)\,\epsilon}\ v_{\text{OPT}}$; 但它确实取得了显著更高的函数值。

3.2 SCKP 上的实验

我们在最大覆盖实例上评估了三种不同的算法：转换算法 convert 及其两种不同的子程序——greedy-knapsack-bi（“BLOCK-G”）和不采用区块结构的标准贪心算法（“GREEDY”），以及不带分区背包约束的次模覆盖贪心算法（“GREEDY-Knapsack”）。有关 GREEDY 和 GREEDY-Knapsack 算法以及实验设置的更多细节，请参见附录 F.1。

最小预算（可通过 $\max_{i\in[N]} {}^{c}{}_{(S\cap U_i)}\ p_i$ 计算）和函数值 f 的结果绘制在图~~1(c)~~和图1(d)中。从结果可以看出，BLOCK-G、GREEDY-Knapsack 和 GREEDY 返回的解的 f 值几乎相同。这是因为不同算法在 f 上的理论保证大致相同。然而，我们的算法 BLOCK-G 返回的解的预算小于另外两种算法，这表明了我们采用分块方式运行贪心算法的有效性。

3.3 SCF 实验

对于 SCF问题，我们采用 Chen 等人的转换框架 [2025] 评估算法，并使用不同的子程序：我们的 Block-Fair-Bi 算法（“BLOCK-G-Fair”）、标准贪心算法（“GREEDY-Fair”）和阈值贪心算法（“THRES-Fair”）。我们针对不同的阈值 τ，从解成本（基数）、公平性差异、目标函数值、查询复杂度和执行时间等方面比较这些算法。这里，我们为 Chen 等人 [2025]的算法 1 中的转换算法设定 $\alpha = 0.2$。公平性约束中的参数设定为 $u_c = 1.1/N$，$l_c = 0.9/N$。（其中 N为组数）。有关应用、实验设置和结果的更多细节，请参阅附录中的 F节。

图1(e)和图1(g)展示了解集的成本（基数），而图1(f)和图1(h)展示了不同 τ 值下的公平性差异。在大多数情况下，BLOCK-G-Fair的成本低于THRES-Fair和GREEDY-Fair，这与我们的理论结果一致。在 corel数据集中 τ 较大的情况下，THRES-Fair的成本低于BLOCK-G-Fair。然而，图1(f)和图1(h)表明，THRES-Fair和GREEDY-Fair的公平性差异显著大于BLOCK-G-Fair，说明BLOCK-G-Fair能够生成更加均衡的解。这是意料之中的，因为Chen 等人提出的公平性约束 [2025] 确保 $\beta\lfloor \frac{p_c|S|}{\beta} \rfloor \leq |S\cap U_c| \leq \beta\lceil \frac{q_c|S|}{\beta} \rceil$，这意味着解集可能会以加性的 ——

因子 β违反公平性约束。值得注意的是，对于THRES-Fair和GREEDY-Fair， $\beta = \mathcal{O}(1/\epsilon)$，而对于BLOCK-G-Fair， $\beta = \ln(\frac{1}{\epsilon})\ln 2$，这意味着我们的方法实现了更强的公平性保证。

4 可复现性声明

本文中的所有理论结果均由完整的证明予以支持，相关证明见正文和附录。文中详细描述了近似保证和证明，以确保这些理论贡献能够得到独立验证。

参考文献

Baijun Ji、Xiangyu Duan、Zhenyu Qiu、Tong Zhang、Junhui Li、Hao Yang 和 Min Zhang。基于子模的上下文示例选择，用于基于大语言模型的机器翻译。载于 *2024* 年计算语言学、语言资源与评测联合国际会议论文集（*LREC-COLING 2024*），第 15398–15409 页，2024年。

Lilly Kumari、Shengjie Wang、Arnav Das、Tianyi Zhou 和 Jeff Bilmes。用于数据高效的上下文学习的端到端子模框架。载于 计算语言学协会会议：*NAACL 2024* 论文集，第 3293–3308 页，2024年。

Ishika Agarwal、Krishnateja Killamsetty、Lucian Popa 和 Marina Danilevsky。Delift：数据高效的语言模型指令微调。载于第十三届学习表征国际会议，2024年。

Gaurush Hiranandani、Harvineet Singh、Prakhar Gupta、Iftikhar Ahamath Burhanuddin、Zheng Wen 和 Branislav Kveton。级联线性子模老虎机：在在线学习排序中考虑位置偏差和多样性。载于人工智能中的不确定性，第722–732页。PMLR，2020年。

Wenjing Chen 和 Victoria G Crawford。具有老虎机反馈的线性子模最大化。载于人工智能与统计国际会议，第4249–4257页。PMLR，2025年。

Lifeng Zhou 和 Pratap Tokekar。面向多机器人协调的风险感知子模优化。*IEEE*机器人学汇刊，38(5):3064–3084，2022年。

Zirui Xu 和 Vasileios Tzoumas。性能感知的自配置多智能体网络：一种用于协同与网络设计的分布式次模方法。载于*2024* 年 *IEEE* 第 *63* 届决策与控制会议（*CDC*），第 5393–5400 页。IEEE，2024 年。

Ruoyu Chen、Hua Zhang、Siyuan Liang、Jingzhi Li 和 Xiaochun Cao。少即是多：通过次模子集选择获得更少的可解释区域。载于第十二届国际学习表征会议，2024a年。

Rishabh K Iyer 和 Jeff A Bilmes。带有次模覆盖和次模背包约束的次模优化。神经信息处理系统进展，26，2013 年。

Wenjing Chen 和 Victoria Crawford。次模覆盖问题的双准则近似算法。神经信息处理系统进展，36，2024年。

Victoria Crawford、Alan Kuhnle 和 My Thai。使用近似预言机的次模成本次模覆盖。载于国际机器学习会议，第 1426–1435 页。PMLR，2019年。

Baharan Mirzasoleiman、Amin Karbasi、Ashwinkumar Badanidiyuru 和 Andreas Krause。分布式次模覆盖：简洁地总结海量数据。神经信息处理系统进展，28，2015。

Baharan Mirzasoleiman、Stefanie Jegelka 和 Andreas Krause。流式非单调次模最大化：即时进行个性化视频摘要。载于*AAAI*人工智能会议论文集，第 32 卷，2018年。

Sebastian Tschiatschek、Rishabh K Iyer、Haochen Wei 和JeffABilmes。用于图像集合摘要的子模函数混合模型学习。神经信息处理系统进展，27，2014年。

Wenjing Chen、Shuo Xing、Samson Zhou 和 Victoria G Crawford。公平子模覆盖。载于第十三届学习表征国际会议，2025年。Krishnateja Killamsetty、Durga Sivasubramanian、Ganesh Ramakrishnan 和 Rishabh Iyer。Glister：基于泛化的数据子集选择，用于高效且稳健的学习。载于*AAAI*人工智能会议论文集，第35卷，第8110–8118页，2021。Marwa El Halabi、Slobodan Mitrovi´c、Ashkan Norouzi-Fard、Jakab Tardos 和 Jakub M Tarnawski。流式次模最大化中的公平性：算法与困难性。神经信息处理系统进展，33：13609–13622，2020年。George L Nemhauser、Laurence A Wolsey 和 Marshall L Fisher。最大化次模集合函数的近似分析——I。数学规划，14(1)：265–294，1978年。Marshall L Fisher、George L Nemhauser 和 Laurence A Wolsey。最大化次模集合函数的近似分析——*II*。施普林格，1978年。Gruia Calinescu、Chandra Chekuri、Martin Pal 和 Jan Vondr´ak。在拟阵约束下最大化单调子模函数。*SIAM*计算杂志，40(6)：1740–1766，2011年。Ashwinkumar Badanidiyuru 和 Jan Vondr´ak。最大化子模函数的快速算法。载于第二十五届 *ACM-SIAM*离散算法年度研讨会论文集，第1497–1514页。SIAM，2014年。Chandra Chekuri 和 Kent Quanrud。通过多线性松弛并行最大化子模函数。载于第三十届 *ACM-SIAM*离散算法年度研讨会论文集，第303–322页。SIAM，2019年。Niv Buchbinder 和 Moran Feldman。针对拟阵约束下子模最大化的确定性算法与更快算法。载于*2024 IEEE*第*65*届年度计算机科学基础研讨会（*FOCS*），第700–712页。IEEE，2024a年。Niv Buchbinder 和 Moran Feldman。子模最大化问题的确定性算法。*ACM*算法汇刊（*TALG*），14(3)：1–20，2018年。Shayan Oveis Gharan 和 Jan Vondr´ak。通过模拟退火进行子模最大化。载于第二十二届 *ACM-SIAM*离散算法年度研讨会论文集，第1098–1116页。SIAM，2011年。Benjamin Qi。关于最大化非单调子模函数与线性函数之和。*Algorithmica*，86(4)：1080–1134，2024年。Niv Buchbinder 和 Moran Feldman。利用 DR次模函数的新界进行约束子模最大化。载于第*56*届 *ACM*计算理论研讨会年会论文集，第1820–1831页，2024b年。Yixin Chen、Ankur Nath、Chunli Peng 和 Alan Kuhnle。离散地超越 $1/e$：用于子模最大化的引导式组合算法。载于第三十八届神经信息处理系统年度会议，2024b年。Marwa El Halabi、Federico Fusco、Ashkan Norouzi-Fard、Jakab Tardos 和 Jakub Tarnawski。在拟阵约束下进行流式子模最大化时的公平性。载于国际机器学习会议，第9150–9171页。PMLR，2023年。Georgios Amanatidis、Federico Fusco、Philip Lazos、Stefano Leonardi 和 Rebecca Reiffenh¨auser。在背包约束下进行快速自适应非单调子模最大化。神经信息处理系统进展，33：16903–16915，2020年。Shuang Cui、Kai Han 和 Jing Tang。数据流中代表性子集选择的线性时间算法。载于*ACM*网络会议论文集*2025*，第4710–4721页，2025年。Shuang Cui、Kai Han、Shaojie Tang、Feng Li 和 Jun Luo。在背包约束下进行流式子模最大化时的公平性。载于第*30*届 *ACM SIGKDD*知识发现与数据挖掘会议论文集，第514–525页，2024年。

Lijun Li、Chenyang Xu、Liuyi Yang 和 Ruilong Zhang。背包约束下的公平子模最大化。载于第*34*届国际人工智能联合会议（*2025*），第3934–3942页。国际人工智能联合会议，2025年。Laurence A. Wolsey。子模集合覆盖问题的贪心算法分析。组合学，2(4): 385–393，1982年。Andreas Krause、H Brendan McMahan、Carlos Guestrin 和 Anupam Gupta。稳健的子模观测选择。机器学习研究期刊，9(12)，2008年。Victoria Crawford。非单调子模覆盖的可扩展双准则算法。载于人工智能与统计国际会议，第9517–9537页。PMLR，2023年。Jure Leskovec 和 Rok Sosič。Snap：通用网络分析与图挖掘库。*ACM*智能系统与技术汇刊（*TIST*），8(1):1–20，2016年。Moran Feldman、Amin Karbasi 和 Ehsan Kazemi。少做多得：采用子采样的流式子模最大化。神经信息处理系统进展，31，2018年。Kai Han、Shuang Cui、Benwei Wu 等。近线性时间内拟阵上子模最大化的确定性近似。神经信息处理系统进展，33:430–441，2020年。Niv Buchbinder、Moran Feldman、Joseph Naor 和 Roy Schwartz。带基数约束的子模最大化。载于第二十五届年度*ACM-SIAM*离散算法研讨会论文集，第1433–1452页。SIAM，2014年。Victoria G Crawford。用于最小成本子模覆盖的高效进化算法。载于第*28*届国际人工智能联合会议论文集，第1227–1233页，2019年。Rishabh Iyer 和 Jeffrey Bilmes。子模点过程及其在机器学习中的应用。载于人工智能与统计，第388–397页。PMLR，2015年。Benedek Rozemberczki 和 Rik Sarkar。Twitch玩家：用于评估邻近性保持和基于结构角色的节点嵌入的数据集。*arXiv*预印本 *arXiv:2101.03091*，2021年。Pinar Duygulu、Kobus Barnard、Joao FG de Freitas 和 David A Forsyth。将物体识别视为机器翻译：为固定图像词汇表学习词典。载于计算机视觉——*ECCV 2002*：第*7*届欧洲计算机视觉会议，哥本哈根，丹麦，*2002*年*5*月*28–31*日，论文集，第*IV*卷 *7*，第97–112页。施普林格，2002年。Jia Deng、Wei Dong、Richard Socher、Li-Jia Li、Kai Li 和 Li Fei-Fei。ImageNet：大型层次化图像数据库。载于*2009*年*IEEE*计算机视觉与模式识别会议，第248–255页。IEEE，2009年。

算法 4 convert-rand

输入： 一个阈值为 τ 的 SCP 实例，一个用于 SMP 的 (γ, β)-双准则近似算法，$\alpha > 0$**输出：** $S \subseteq U$**1:** $S_i \leftarrow \emptyset, \forall i \in \{1, ..., \ln(1/\delta)/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma})\}$**2:** $g \leftarrow (1+\alpha)$ **3: 当** $f(S_i) < (\gamma-\epsilon)\tau \; \forall i$ **时 4: 对** $i \in \{1, ..., \ln(1/\delta)/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma})\}$**执行 5: 用于 SMP 的** $(S_i \leftarrow, \gamma)$**-双准则近似算法，其目标函数为** β**，预算为** f **6:** $g(g \leftarrow)1+\alpha$**7: 返回**g

问题	算法	近似比	查询复杂度
非单调 SCP	convert-rand + nonmono-bi	$\left(O\left(\frac{1+\alpha}{-\epsilon}\right), \frac{1}{e}-\epsilon\right)$	$\mathcal{O}\left(n^2 \log_{1+\alpha} n \ln(1/\delta)/\epsilon/\ln(1+\epsilon)\right)$
单调 SCKP	greedy-knapsack-bi +convert	$\left(\frac{(1+\alpha)\ln(1/\epsilon)\ln 2}{}, \; 1-\epsilon\right)$	$\mathcal{O}\left(n^2 \log_{1+\alpha}\left(\frac{c_{\max} n}{c_{\min}}\right)\right)$
单调次模覆盖函数	Block-Fair-Bi + 公平转换算法 [Chen 等, 2025]	$(\mathcal{O}(\ln(1/\epsilon)) \; 1-\epsilon)$	$\mathcal{O}\left(\frac{n \log \; n\kappa \ln(\quad 1/\epsilon}{\epsilon}\right)$

表 1：本文中次模覆盖问题近似算法的总结。

附录

A 大语言模型的使用

本文仅使用大语言模型润色文稿，以提升表达的清晰度和呈现效果。本文的所有理论结果和技术贡献均非由大语言模型提出。

B 第 2.1 节的附录

本节补充第 2.1 节中遗漏的内容。该节讨论了带分区约束的非单调子模覆盖问题。首先，给出算法 4 中的 convert-rand 转换算法伪代码；我们在第 2.1 节仅对该算法作了非正式描述。接下来，我们给出定理 2.3 和定理 2.4 的证明。

定理 2.3。对于 *SMP* 的任意随机化(γ, β)双准则近似算法，若其运行时间为 $\mathcal{T}(n)$，且其中γ仅在期望意义下成立，则可将其转换为 *SCP* 的$((1+\alpha)\beta, \gamma-\epsilon)$双准则近似算法，其运行时间为 $O(\log_{1+\alpha}(|OPT|)\ln(1/\delta)\mathcal{T}(n))/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma}))$，且 γ以至少 $1-\delta$ 的概率成立。

证明。考虑算法 4 第 5 行中 SMP 算法的运行过程，其中对最优值 g 的猜测落在区域

$$v_{OPT} \leq g \leq (1+\alpha)v_{OPT}.$$

我们将预算为 g 的分划拟阵记为 $\mathcal{M}$，即 $\mathcal{M} := \{S \subseteq U : \; |S \cap U_j| \leq p_j v, \; \forall j \in [N]\}$。SMP 问题定义为寻找 $\arg\max\{f(S): \; S \in \mathcal{M}\}$。我们将预算为 g 的 SMP 问题的最优解记为 OPT_g，即，

$$OPT_g := \arg\max\{f(S) : |S \cap U_j| \leq p_j v, \forall j \in [N]\}.$$

此外，我们将 SCP 的最优解定义为 OPT。由于最优解 OPT 对 SCP 是可行的，我们有

$$|OPT \cap U_j| \leq p_j v_{OPT} \leq p_j v \tag{2}$$

这意味着 $OPT \in \mathcal{M}$，因此 $f(OPT) \leq \max_{S\in\mathcal{M}} f(S) = f(OPT_g)$。由于 $f(OPT) \geq \tau$，我们有$f(OPT_g) \geq \tau$。由此可知，对于每个 $i \in \{1, ..., \ln(1/\delta)/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma})\}$，

$$\begin{aligned} P\Big(f(S_i) \leq (\gamma-\epsilon)\tau\Big) &\leq P\Big(f(S_i) \leq (\gamma-\epsilon)f(OPT_g)\Big) \\ &\leq P\Big(\beta f(OPT_g) - f(S_i) \geq (\beta-\gamma+\epsilon)f(OPT_g)\Big) \end{aligned}$$

根据 SMP 算法的理论保证，对于所有 $i \in \{1, ..., \ln(1/\delta)/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma})\}$，我们有 $\mathbb{E}f(S_i) \geq \gamma f(OPT_g)$，并且 $|S_i \cap U_j| \leq p_j\beta g$对于每个 $j \in [N]$。由此可知

$$P\Big(f(S_i) \leq (\gamma-\epsilon)\tau\Big) \leq P\Big(\beta f(OPT_g) - f(S_i) \geq \frac{\beta-\gamma+\epsilon}{\beta-\gamma}(\beta f(OPT_g) - \mathbb{E}f(S_i))\Big)$$

记 $\mathcal{M}_\beta := \{S \subseteq U\colon |S \cap U_j| \leq p_j\beta g, \forall j \in [N]\}$，即，$\mathcal{M}_\beta$是拟阵 $\mathcal{M}$ 的 β-扩展，其定义见 Chen 等人 [2025]。由于 S_i 满足 $|S_i \cap U_j| \leq p_j\beta g$ 对于任意 $j \in [N]$，我们有 $S_i \in \mathcal{M}_\beta$ 对于每个$i \in \{1, ..., \ln(1/\delta)/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma})\}$。注意，对于任意集合 $A \in \mathcal{M}_\beta$，我们可以将 A 表示为 $\mathcal{M}$ 中 β 个互不相交子集的并。将它们记为 $A_1, A_2, ..A_m$。因此我们有

$$\begin{aligned} f(A) &= f(\cup_{i\in[\beta]} A_i) \\ &\leq \sum_{i\in[\beta]} f(A_i) \leq \beta f(OPT_g), \end{aligned}$$

其中，第一个不等式源于 f 的子模性。由此可得，$\max_{S\in\mathcal{M}_\beta} f(S) \leq \beta f(OPT_g)$。注意到 $S_i \in \mathcal{M}_\beta$，我们有 $\beta f(OPT_g) - f(S_i) \geq 0$。因此，我们可以对随机变量 $\beta f(OPT_g) - f(S_i)$ 应用马尔可夫不等式。因此，对于每个 $i \in \{1, ..., \ln(1/\delta)/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma})\}$，我们可以得到

$$\begin{aligned} P\Big(f(S_i) \leq (\gamma-\epsilon)\tau\Big) &\leq P\Big(\beta f(OPT_g) - f(S_i) \geq \frac{\beta-\gamma+\epsilon}{\beta-\gamma}(\beta f(OPT_g) - \mathbb{E}f(S_i))\Big) \\ &\leq \frac{\beta-\gamma}{\beta-\gamma+\epsilon} \end{aligned}$$

然后 t he 个子集 S_i 均无法达到停止条件的概率可以界定为

$$\begin{aligned} P(f(S_i) \leq (\gamma-\epsilon)\tau, \forall i) &= P(f(S_i) \leq (\gamma-\epsilon)\tau, \forall i) \\ &= \prod_{i=1}^{\ln(1/\delta)/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma})} P(f(S_i) \leq (\gamma-\epsilon)\tau) \\ &\leq (\frac{\beta-\gamma}{1+\epsilon-\gamma})^{\ln(1/\delta)/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma})} = \delta. \end{aligned}$$

这意味着，以至少 $1-\delta$ 的概率，当 g 到达满足 $v_{OPT} \leq g \leq (1+\alpha)v_{OPT}$ 的区域时，convert-rand 会停止，因为**while** 循环的条件不成立。因此，根据子程序算法是一个 (γ, β)-双准则近似算法这一假设，我们有输出解 S 满足$|S \cap U_j| \leq p_j\beta g \leq p_j\beta(1+\alpha)v_{OPT}$。随后，最优解 S的目标值可设为$v_S = \beta(1+\alpha)v_{OPT}$。这也意味着，对最优解基数的猜测次数至多为 $O(\log_{1+\alpha} v_{OPT})$。由于每次猜测时，我们运行 SMP 共 $\ln(1/\delta)/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma})$ 次，因此该算法的运行时间为 $O(\log_{1+\alpha}(v_{OPT})\ln(1/\delta)\mathcal{T}(n)/\ln(\frac{\beta-\gamma+\epsilon}{\beta-\gamma}))$。

□

接下来，我们给出定理 2.4 中结果的证明。

定理 2.4。 假设对预算为 v 的非单调 *SMP* 实例运行 *nonmono-bi*，则 *nonmono-bi* 输出的解 S 的期望双准则近似比为 $(1/e-\epsilon, \frac{2}{\epsilon})$，且查询数量至多为 $O(\frac{nv}{\epsilon})$。

证明。 将算法 1 第 6 行外层 for 循环的第 i 轮中，第 j 个子组添加第 l 个元素后的解集记为 $S_{i,j,l}$；将算法 1 第 i 轮中完成添加第 j 个子组的所有元素后的解集记为 S_{i,j_0}。为简化记法，我们还定义 $\phi := \frac{2}{\epsilon}$。根据贪心选择策略，可得

$$\mathbb{E}[f(S_{i,j,l}) - f(S_{i,j,l-1})] \geq \frac{\sum_{a \in OPT \cap U_j} \Delta f(S_{i,j,l-1}, a)}{p_j v \phi}.$$

根据子模性，可得

$$\mathbb{E}[f(S_{i,j,l}) - f(S_{i,j,l-1})] \geq \frac{\sum_{a \in OPT \cap U_j} \Delta f(S_{i,j}, a)}{p_j v \phi}.$$

对所有 $l \in [p_j v]$求和后，可得

$$\mathbb{E}[f(S_{i,j}) - f(S_{i,j-1}] \geq \frac{\sum_{a \in OPT \cap U_j} \Delta f(S_{i,j}, a)}{\phi}.$$

记为 第 i 轮全部完成后的解记为 S_i。根据子模性，可得 t

$$\begin{aligned}\mathbb{E}[f(S_{i,j}) - f(S_{i,j-1})] &\geq \frac{\mathbb{E}[\Delta f(S_{i,j}, OPT_j)]}{\phi} \\ &\geq \frac{\mathbb{E}[\Delta f(S_i, OPT_j)]}{\phi},\end{aligned}$$

对所有 $j \in [N]$ 求和，可得

$$\begin{aligned}\mathbb{E}[f(S_i) - f(S_{i-1})] &\geq \frac{\sum_{j=1}^{N} \mathbb{E}[\Delta f(S_i, OPT_j)]}{\phi} \\ &\geq \frac{\mathbb{E}[\Delta f(S_i, OPT)]}{\phi}.\end{aligned}$$

因此可得

$$\mathbb{E}[f(S_i) - f(S_{i-1})] \geq \mathbb{E}[\frac{f(S_i \cup OPT) - f(S_i)}{\phi}].$$

注意，根据贪心选择步骤，对于每个组 j，每个元素 $a \in OPT \cap U_j$ 出现在 S_i 中的概率至多为 $1-(1-\frac{1}{p_j v \phi})^{p_j v i}$。由于 $(1-\frac{1}{x})^x$ 在 x 的范围 $[1, +\infty)$ 内递增，因此有 $(1-\frac{1}{p_j v \phi})^{p_j v \phi} \geq (1-\frac{1}{\phi})^{\phi}$。因此，我们得到 $1-(1-\frac{1}{p_j v \phi})^{p_j v i} \leq 1-(1-\frac{1}{\phi})^i$。根据 Buchbinder 等大的引理 2.2 [2014]，我们可以得出

$$\mathbb{E}[f(S_i \cup OPT)] \geq (1 - \frac{1}{\phi})^i f(OPT).$$

通过重新整理上述不等式，我们可以得到

$$\mathbb{E}[f(S_i)] \geq \frac{\phi}{\phi+1}\mathbb{E}[f(S_{i-1})] + \frac{1}{\phi+1}(1-\frac{1}{\phi})^i f(OPT).$$

根据归纳法，我们可以得到输出解集满足

$$\begin{aligned}\mathbb{E}[f(S)] &= \mathbb{E}[f(S_\phi)] \\ &\geq \frac{1}{\phi+1}\sum_{i=1}^{\phi}(\frac{\phi}{\phi+1})^{\phi-i}(1-\frac{1}{\phi})^i f(OPT) \\ &\geq \frac{1}{\phi+1}\sum_{i=1}^{\phi}(\frac{\phi-1}{\phi})^{\phi-i}(1-\frac{1}{\phi})^i f(OPT) \\ &\geq \frac{\phi}{\phi+1}(1-\frac{1}{\phi})^{\phi} f(OPT) \geq \frac{1}{e}(1-\epsilon) f(OPT). \end{aligned} \tag{3}$$

其中 最后一个不等式根据以下事实得出：对于任意–$\phi > 1$，$(1-\frac{1}{\phi})^{\phi-1} \geq e^{-1}$，且 $\phi = \frac{2}{\epsilon}$ –. □

C 附录：第 2.2 节

现在，我们将介绍第 2.2 节中省略的内容。该节研究了带背包分区约束的单调子模覆盖。首先，我们将进一步阐述所提出的优化问题表述，以说明其动机并加以解释。具体而言，C.1 节列举了若干 SCKP 的详细示例，以阐明其动机；随后，C.2 节将对 SCKP 的表述进行详细讨论。接着，在第 2.2 节中，我们将给出主论文第 2.2 节中省略的证明。具体来说，我们将给出定理 2.5 中关于以 Alg-SM 为子程序的 Block-Greedy 算法的理论保证的缺失证明，并介绍定理 C.3 中所述的转换算法 convert，用于将 SMKP 算法转换为 SCKP 算法。

C.1 SCKP 的应用示例

在附录的这一部分，我们给出一系列示例，以阐明我们研究 SCKP 问题的动机。该问题的目标是找到一个解集 S，在保证效用达到一定水平（$f(S) \geq \tau$）且不同分区的成本约束均衡（$c(S \cap U_j) \leq p_j v$）的同时，使总成本最小。该问题的动机示例包括

- **影响力最大化**：在此应用中，我们可能希望以最低成本选择一组节点（例如，分配有限的预算资金），同时确保影响力传播达到一定水平。各分区之间的成本也应保持均衡；这些分区由人口统计属性或地理属性划分。激活不同节点（例如，有影响力的用户或组）可能需要不同的成本（例如，通过定向广告或促销活动），因此不同节点的成本并不一致。

- **预训练数据选择**：在预训练数据选择中，目标是以最低成本选择一部分数据点，其中成本可能反映计算、标注或存储开销。该问题涉及在数据集的预定义组之间均衡成本，同时确保效用函数（例如，多样特征覆盖）达到指定阈值。

- **多智能体任务分配：**目标是找到一组任务，在实现一定整体效用或性能水平，并使不同类型任务（例如配送、检查或清洁）之间的成本保持均衡的同时，最大限度地降低所分配任务的总成本。任务的执行成本因复杂度、持续时间或所需资源而异，因此成本并非均一。

C.2 SCKP 问题定义的澄清

本节将举例说明主论文第 2.2 节定义的 SCKP 问题形式。首先回顾一下，先前工作 [Iyer 和 Bilmes, 2013, Crawford, 2019] 研究的经典最小成本子模覆盖（MCSC）定义为 $\arg\min\{c(S) : f(S) \geq \tau\}$ 其中 $c: 2^U \to \mathbb{R}$ 是一个模函数、正值的成本函数。在我们的设定中，我们还希望确保不同分区之间的预算分配均衡。因此，我们的问题定义之一应为 $\arg\min\{c(S) : f(S) \geq \tau,\ c(S \cap U_j) \leq p_j c(S)\ \forall j \in [N]\}$。

然而，在许多情况下，上述定义的问题可能存在可行性问题。具体而言，对于每个 $j \in [N]$，约束 $c(S \cap U_j) \leq p_j c(S)$ 可能很难满足，甚至可能导致问题不可行。例如，若对每个 $j \in [N]$ 都设定 $p_j = 1/N$，且对每个 $s \in U_{j_1}$ $c(s) = \pi$，并且对每个 $s \in U_{j_2}$，$c(s) = 1$。根据 $(P1)$ 的定义可知，不存在满足每个 $j \in [N]$ 对应的 $c(S \cap U_j) \leq p_j c(S)$ 的子集 $S \subseteq U$。

为了解决这一可行性问题，我们可以放宽对平衡解的约束，使其允许被单个元素的成本轻微违反。定义 $c_j = \max\{c(s) : s \in U_j\}$ 为分区 U_j内的最大单元素成本。由此可知，松弛问题应定义为 $\arg\min\{c(S) : f(S) \geq \tau, \ c(S \cap U_j) \leq p_j c(S) + c_j, \ \forall j \in [N]\}$

为简化记号，我们用 $(P1)$ 表示这个问题，即

$$
\begin{aligned}
(P1): \quad & \min_{S \subseteq U} c(S) \\
& f(S) \geq \tau \\
& c(S \cap U_j) \leq p_j c(S) + c_j, \qquad \forall j \in [N].
\end{aligned} \tag{4}
$$

为了解决这个问题，我们可以通过在约束中引入另一个变量 μ，稍微放宽对成本的约束，即对每个 $j \in [N]$，都有 $c(S \cap U_j) \leq p_j \mu c(S)$。请注意，我们还希望将违反约束的程度降至最低。为此，在优化问题中，我们将目标函数从最小化 $c(S)$替换为 $\mu c(S)$。接下来，用 v 替换项 $\mu c(S)$，即可得到 SCKP 问题的定义。此外，与 $(P1)$ 中定义的优化问题相比，只要阈值 τ 满足 $f(U) \geq \tau$，(1) 中定义的 SCKP 问题就能保持可行性。

$$
\begin{aligned}
(P2): \quad & \min_{S \subseteq U} v(S) \\
& f(S) \geq \tau \\
& c(S \cap U_j) \leq p_j v, \qquad \forall j \in [N],
\end{aligned} \tag{5}
$$

我们将 $(P1)$ 的最优解和最优值分别定义为 OPT_{P1}和 v_{OPT}，并将定义的 SCKP 的最优解记为 OPT。值得注意的是，优化问题 $(P1)$ 中的最优解与我们的 SCKP 问题 $(P2)$ 中的最优解质量相近。特别地， $P1$和 $(P2)$ 的最优值满足以下引理：

引理 C.1. $(P1)$和 $(P2)$ 的最优值满足

1. $v_{OPT} \leq c(OPT_1) + \max_{i \in [N]} \frac{c_i}{p_i}$. *2.* 当$(P2)$ 的最优值满足 $p_j v_{OPT} \leq c(U_j)$时，我们有 $c(OPT_1) \leq \alpha v_{OPT} + \sum_{i \in [N]} c_j$，其中 $\alpha = \sum_{j \in [N]} p_j$。

证明。 我们首先证明引理的第一部分。由于 OPT_1 对于问题 $(P1)$ 是可行的，因此它必须满足 $(P1)$ 的所有约束。特别地，对于每个 $j \in [N]$，我们有：

$$
\begin{aligned}
c(OPT_1 \cap U_j) &\leq p_j c(OPT_1) + c_j \\
&= p_j \left(c(OPT_1) + \frac{c_j}{p_j} \right) \\
&\leq p_j \left(c(OPT_1) + \max_{i \in [N]} \frac{c_i}{p_i} \right)
\end{aligned} \tag{6}
$$

令 $v = c(OPT_1) + \max_{i \in [N]} \frac{c_i}{p_i}$，注意到 $S = OPT_1$ 满足在 $(P2)$ 中定义的 SCKP 问题的约束。因此，$v_{OPT} \leq v$，第一条结论得证。我们通过以下过程构造集合 A，以证明第二条结论。

1. 初始化 $A \leftarrow \text{OPT}_2$。

2. 对于 $j = 1$ 到 N，执行：

(a) 当 $c(A \cap U_j) \leq p_j v_{\text{OPT}}$ 时:

i. $x \leftarrow \arg\min_{x' \in U_j \setminus A} c(x')$ ii. $A \leftarrow A \cup \{x\}$

注意，对于每个 $j \in [N]$，$c(OPT_2 \cap U_j) \leq p_j v_{OPT}$，并且 $c(U_j) \geq p_j v_{OPT}$。因此，上述过程终止时，集合 A 满足

$$p_j v_{OPT} \leq c(A \cap U_j) \leq p_j v_{OPT} + c_j \tag{7}$$

由此可得 $\sum_{j\in[N]} p_j v_{OPT} \leq \sum_{j\in[N]} c(A \cap U_j)$，进而有 $\alpha v_{OPT} \leq c(A)$。因此，对于每个 $j \in [N]$，我们有

$$c(A \cap U_j) \leq p_j v_{OPT} + c_j \leq p_j c(A) + c_j \tag{8}$$

这意味着 A 对问题（$P1$）是可行的，因此，我们可以得出 $c(OPT_1) \leq c(A) \leq \sum_{j\in[N]} p_j v_{OPT} + c_j \leq \alpha v_{OPT} + \sum_{j\in[N]} c_j$。 □

此外，我们想指出，我们的问题定义还有一个好处：它保留了 SCKP 问题与 SMKP 问题之间的对偶关系，该关系定义为 $\arg\max\ f(S): \sum_{s\in X\cap U_j} c(s) \leq p_j v$。具体而言，在 SMKP 中，变量 v 也充当成本约束的预算。这一性质有助于我们应用转换定理，将用于 SMF 的双准则算法转换为用于 SCF 的算法。

C.3 定理 2.5 的证明

在本附录部分，我们给出以 Alg-SM 为子程序的 Block-Greedy 算法理论保证中缺失的证明。定理陈述见定理 2.5。

定理 2.5。 假设对一个 *SMKP* 实例运行算法*2*中描述的 *greedy-knapsack-bi*，则 *greedy-knapsack-bi* 输出的解集满足

$$f(S) \geq (1-\epsilon) f(OPT)$$
$$c(S \cap U_j) \leq \frac{2\ln\frac{1}{\epsilon}}{\ln 2} p_j v, \qquad \forall j \in [N],$$

其中 OPT 是 *SMKP* 的最优解。

将算法2第 4 行至第 11 行的外层 for 循环中，第 i 轮期间向第 j 个子组添加第 l 个元素后得到的解集记为 $S_{i,\,j,\,l}$；将算法2第 i 轮期间添加完第 j 个子组中的所有元素后得到的解集定义为 $S_{i,\,j}$。在证明定理 2.5 之前，我们先证明以下引理中的结果。

引理 C.2。 设 $S_{i,\,j}$ 是算法*2*中 *greedy-knapsack-bi* 算法在第 i 轮期间添加完第 j 个子组中的所有元素后得到的解集，则可得

$$f(S_{i,j}) - f(S_{i,j-1}) \geq \Delta f(S_{i,j}, OPT_j)$$

其中 $OPT_j := OPT \cap U_j$ 是最优解集 OPT 与第 j 个分区 U_j 的交集，并且 t

$$B_j \leq c(S_{i,j}/S_{i,j-1}) \leq 2B_j.$$

这里 $B_j := p_j v$。

证明。 令 $A_{i,\,j,\,l}$ 表示在迭代 i 中将第 l 个元素添加到子组 j 后的集合，并令 $s_{i,\,j,\,l}$ 表示在算法2中子组 j 的第 i 次外层循环期间添加到集合 A 中的第 l 个元素。因此，对于任意元素 $o \in OPT_j$,

$$\frac{\Delta f(S_{i,j-1} \cup A_{i,j,l-1}, s_{i,j,l})}{c(s_{i,j,l})} \geq \frac{\Delta f(S_{i,j-1} \cup A_{i,j,l-1}, o)}{c(o)}.$$

通过重新排列上述不等式，可得

$$c(o)\Delta f(S_{i,j-1}\cup A_{i,j,l-1}, s_{i,j,l}) \geq c(s_{i,j,l})\Delta f(S_{i,j-1}\cup A_{i,j,l-1}, o).$$

对所有 $o\in OPT_j$ 求和，并利用子模性，可得

$$c(OPT_j)\Delta f(S_{i,j-1}\cup A_{i,j,l-1}, s_{i,j,l}) \geq c(s_{i,j,l})\Delta f(S_{i,j-1}\cup A_{i,j,l-1}, OPT_j)$$

设算法2中的迭代总次数为 T。由子模性可得

$$c(OPT_j)\Delta f(S_{i,j-1}\cup A_{i,j,l-1}, s_{i,j,l}) \geq c(s_{i,j,l})\Delta f(S_{i,j-1}\cup A_{i,j,T}, OPT_j).$$

由于

$$\Delta f(S_{i,j-1}\cup A_{i,j,l-1}, s_{i,j,l}) = f(S_{i,j-1}\cup A_{i,j,l}) - f(S_{i,j-1}\cup A_{i,j,l-1}),$$

我们可以对所有 $l\in[T]$ 求和，得到

$$c(OPT_j)\{f(S_{i,j-1}\cup A_{i,j,T}) - f(S)\} \geq c(A_{i,j,T})\Delta f(S_{i,j-1}\cup A_{i,j,T}, OPT_j).$$

由于 $S_{i,\ j-1}\cup A_{i,\ j,\ T} = S_{i,\ j}$，因此

$$c(OPT_j)\{f(S_{i,j}) - f(S)\} \geq c(A_{i,j,T})\Delta f(S_{i,j}, OPT_j).$$

根据算法2的停止条件，有 $c(A_{i,\ j,\ T-1}) \leq B_j$，因此，

$$B_j \leq c(A_{i,j,T}) \leq 2B_j$$

由于 $c(OPT_j) \leq B_j$，因此可得

$$f(S_{i,j}) - f(S) \geq \Delta f(S_{i,j}, OPT_j).$$

根据 $A_{i,\ j,\ T} = S_{i,\ j}/S_{i,\ j-1}$这一事实，我们便可完成证明。 □

借助引理 C.2，我们可以如下证明定理 2.5 的结论。

证明。 记算法2中完成第 i 轮后得到的解集为 S_i，即 $S_i = S_{i,\ N}$。根据引理 C.2 的结果，可得

$$f(S_{i,j}) - f(S_{i,j-1}) \geq \Delta f(S_{i,j}, OPT_j) \geq \Delta f(S_i, OPT_j),$$

whe因为第二个不等式由子模性得出。对所有 $j\in[N]$ 求和，我们将得到 et

$$f(S_i) - f(S_{i-1}) \geq \sum_{j\in[N]} \Delta f(S_i, OPT_j) \geq \Delta f(S_i, OPT)$$

因此， $f(S_i) \geq {}^{f}{}_{(S_{i-1})+f(OPT)2\circ}$ 由归纳法可知，最终输出解集 S 满足

$$f(S) = f(S_\phi) \geq (1-\epsilon)f(OPT).$$

注意到 $S = \cup_{i=1}^{\phi} S_i/S_{i-1}$。根据引理 C.2，我们可以得到 $c((S_i/S_{i-1})\cap U_j) \leq 2p_j v$。因此 $c(S\cap U_j) = \sum_{i=1}^{\phi} c((S_i/S_{i-1})\cap U_j) \leq {}^{2}{}_{\ln\frac{1}{\epsilon}\ln 2}\ p_j v$。 □

算法5 convert

输入： α， ϵ

输出： $S \subseteq U$

$v_g \leftarrow (1+\alpha)c_{\min}$, $S \leftarrow \emptyset$
当 $f(S) < \gamma\tau$ **执行**
  $S \leftarrow$ 使用预算参数 $v = v_g$运行的 SMKP 算法
  $v_g \leftarrow (1+\alpha)v_g$
$v_S = \beta v_g$
返回 S， v_S

C.4 算法5的理论分析

在本附录部分，我们介绍用于将 SMKP 的算法转换为 SCKP 算法的转换算法 convert。伪代码见算法5。convert 的理论保证见定理 C.3。

定理 *C.3*。假设我们有一个用于 *SMKP* 的算法 *Alg-SM*，且给定预算 v，*Alg-SM* 保证在时间 $T(n)$ 内返回一个集合 S，使得 $f(S) \geq \gamma f(OPT_{SM})$ 且 $c(S \cap U_j) \leq \beta p_j v$，其中 OPT_{SM} 是 *SMKP* 的最优解。那么，将 *Alg-SM* 用作子程序的算法 *convert* 会在时间 $\mathcal{O}(\log_{1+\alpha}(\frac{c_{\max} n}{c_{\min}})T(n))$ 内返回一个集合 S 和一个值 v_S，使得 $v_S \leq \beta(1+\alpha)v_{OPT}$，$c(S \cap U_j) \leq p_j v_S$ 且 $f(S) \geq \gamma\tau$.。这里，v_{OPT} 是 *SCKP* 的最优值。$c_{\max}$ 和 $c_{\min}$ 分别是单个元素成本的最大值和最小值。

证明。设 OPT 为 SCKP 实例的最优解。考虑 convert 的迭代过程，其中 v_g 刚刚增至超过 v_{OPT}，即 $v_{OPT} \leq v_g \leq (1+\alpha)v_{OPT}$。然后，我们以预算 $v_{OPT} \leq v_g \leq (1+\alpha)v_{OPT}$运行 Alg-SM。根据对 Alg-SM 的假设，我们有

$$f(S) \geq \gamma f(OPT_{SM}). \tag{9}$$

注意，SCKP 的最优解 OPT满足 $c(OPT \cap U_j) \leq p_j v_{OPT} \leq p_j v_g$。因此，$OPT$对于输入为 v_g的 SMKP 问题是可行的。将 SMKP 的最优解记为 OPT_SM。于是我们有

$$f(OPT_{SM}) \geq f(OPT).$$

由于 OPT 是 SCKP 的最优解，因此

$$f(OPT_{SM}) \geq f(OPT) \geq \tau.$$

结合上述不等式与式（9）的结果，可得 $f(S) \geq \gamma\tau$。因此，算法会在 v_g达到 $(1+\alpha)v_{OPT}$之前停止。每个分区的成本满足

$$c(S \cap U_j) \leq \beta p_j v_g \leq (1+\alpha)\beta p_j v_{OPT}$$

通过设置 $v_S = \beta v_g$即可完成证明。 □

D 第 2.3 节附录

本节给出第 2.3 节中 Block-Fair-Bi 理论结果的缺失证明。定理陈述见定理 2.8。

定理 2.8。 假设针对一个 *SMF* 实例运行 *Block-Fair-Bi*，则 *Block-Fair-Bi* 输出的解 S满足$(1-\varepsilon, \ln \frac{1}{\varepsilon}\ln 2)$-双准则近似保证，且最多使用 $\mathcal{O}(n\kappa\ln(1/\varepsilon))$ 次查询。 f

证明。 将第 i 个块之后的解集记为 S_i，将第 i 个块之后的子集 B 记为 B_i，则可得 $S_i = S_{i-1} \cup \{B_i\}$。我们可以证明以下引理。

引理 D.1。 对于任意 $i \leq \ln 1/\epsilon \ln 2$ ，解集 S_i 满足

$$f(S_i) - f(S_{i-1}) \geq f(OPT) - f(S_i)$$

并且

$$\begin{aligned} &|S_i \cap U_c| \leq u_c i, \\ &\sum_{c \in [N]} \max\{|S_i \cap U_c|, l_c i\} \leq ki. \end{aligned}$$

证明。 记在第 i 个块中，将第 j 个元素添加到解集 S 后得到的解集为 $S_{i,\ j}$。此外，记添加到 B_i 的第 j 个元素为 b_j，并记 $B_{i,\ j} = (b_1, \ldots\ldots, b_{j-1})$。根据拟阵的定义，存在一个从集合 B 到最优解 $OPT = \{o_1, ..., o_\kappa\}$ 的映射，使得 $B_{i,\ j} \cup \{o_j\} \in \mathcal{P}_{fair}$。

$$f(S_{i,j}) - f(S_{i,j-1}) \geq f(S_{i,j-1} \cup \{o_i\}) - f(S_{i,j-1}) \geq \Delta f(S_{i,\kappa}, o_i)$$

对所有 j求和，可得

$$f(S_{i,\kappa}) - f(S_{i,0}) \geq \sum_{i=1}^{\kappa} \Delta f(S_{i,\kappa}, o_i) \geq f(OPT) - f(S_{i,0}).$$

由于 $f(S_{i,\ \kappa}) = f(S_{i+1,\ 0}) = f(S_{i+1})$，

$$f(S_{i+1}) - f(S_i) \geq f(OPT) - f(S_{i+1})$$

接下来，我们证明关于解集 S 大小的结论。当 $i = 1, S_1 = B_1$。根据 $B_1 \in \mathcal{M}_{fair}$ 这一事实，可知引理中的结论成立。假设引理中的结论对 i 成立。那么对于 $i+1$，有 $|S_{i+1} \cap U_c| = |S_i \cup B_i \cap U_c| \leq |S_i \cap U_c| + |B_i \cap U_c| \leq u_c(i+1)$。对于总基数约束，

$$\begin{aligned} \max\{|S_{i+1} \cap U_c|, l_c(i+1)\} &\leq \max\{|S_i \cap U_c| + |B_i \cap U_c|, l_c(i+1)\} \\ &\leq \max\{|S_i \cap U_c|, l_c i\} + \max\{|B_i \cap U_c|, l_c\} \end{aligned}$$

其中，第一个不等式源于 $S_{i+1} = S_i \cup B_i$ 这一事实。第二个不等式则由 $\max\{a+b, c+d\} \leq \max\{a, c\} + \max\{b, d\}$ 的不等式得出。因此，

$$\sum_{c \in [N]} \max\{|S_{i+1} \cap U_c|, l_c(i+1)\} \leq \sum_{c \in [N]} \max\{|S_i \cap U_c|, l_c i\} + \sum_{c \in [N]} \max\{|B_i \cap U_c|, l_c\} \leq \kappa(i+1)$$

□

接下来，利用引理 D.1，我们可以证明定理 2.8 中的结果。令 $\phi = \ln 1/\epsilon \ln 2$ ，则根据该引理，我们有

$$\begin{aligned} &|S_\phi \cap U_c| \leq u_c \phi, \\ &\sum_{c \in [N]} \max\{|S_\phi \cap U_c|, l_c \phi\} \leq k\phi. \end{aligned}$$

由于 $f(S_i) \geq \frac{f(OPT) + f(S_{i-1})}{2}$，由归纳法可得

$$f(S_\phi) \geq (1 - \frac{1}{2^\phi}) f(OPT) = (1 - \epsilon) f(OPT).$$

□

算法 6 `Block-Greedy`

输入： 底集的分区 U_1，U_2，……，U_N、问题定义和参数
输出： $S \subseteq U$
$S \leftarrow \emptyset$
对于 $i = 1$ **至** ϕ，**执行**
    对于 $j = 1$**至** N，**执行**
        贪心子程序（S, i, j）
返回 S

E 分区拟阵约束下的次模最大化 nt

在前面的章节和正文中，我们证明了块贪心算法在解决具有基于分区的约束的次模覆盖问题时颇为有效。有趣的是，这种块贪心方法在设计次模最大化问题的算法时同样很有价值。本节将介绍 Block-Greedy，这是一种专为具有分区拟阵约束的次模最大化问题设计的新型算法框架。

Block-Greedy 通过贪心地将区块（即元素集合）添加到解中来推进算法。与现有方法相比，我们的算法在解的质量和查询复杂度方面都有所改进。

本节安排如下。我们首先在 E.1、E.2 和 E.3 节介绍主要结果。E.1 节介绍了贯穿全文各算法的区块贪心框架。随后，我们讨论两种具体情形：E.2 节讨论带分区拟阵约束的单调子模最大化（单调 SMP），E.3 节讨论带分区拟阵约束的非单调子模最大化（非单调 SMP）。

最后，我们在 E.4 节和 E.5 节补充内容并展开讨论。E.4 节补充了 E.2.1 节中定理 E.2 的相关讨论和证明，而 E.5 节则详述了 E.2.2 节中省略的内容。

E.1 块贪心框架

除了用于公平次模覆盖问题的 Block-Fair-Bi 算法之外，块贪心算法是我们提出的大多数算法的核心框架。Block-Greedy 会反复运行贪心子程序。在每个子程序中，都会从全集 U 的每个分区中选取一个由若干元素组成的“区块”，并将其加入最终解中。参数 ϕ 的取值以及贪心子程序 Greedy-Subroutine 都因问题而异，取决于所求解的次模优化问题。Block-Greedy 的伪代码见算法 6。在下文中，我们将介绍针对不同问题的子程序算法，并分析这些算法。

E.2 单调 SMP

我们首先考虑经典问题：带分区拟阵约束的单调次模最大化（SMP）。给定正整数 $k_1, \cdots k_N$，满足对任意 $j \in [N]$，$k_j \leq |U_j|$，分区拟阵约束定义为 $\mathcal{P} = \{S \subseteq U\colon\ |S \cap U_j| \leq k_j,\ \forall j \in [N]\}$。单调 SMP 要求针对单调次模目标函数 f 求集合 $\arg\max_{S \in \mathcal{P}} f(S)$。在介绍我们的算法之前，我们先通过一个紧确的困难性结果，阐明与标准贪心算法相比，我们所提算法的直觉依据与优势。

E.2.1 紧界示例

标准贪心算法会在保持可行性的同时，迭代选择边际增益最高的元素。众所周知，对于具有一般拟阵约束的单调子模最大化，该算法可达到 $1/2$ 的近似比。尽管分划拟阵是较为简单的特例，我们仍将在下述定理中通过构造一类实例证明这一近似比是紧的：标准贪心算法无法达到优于 $1/2$ 的近似比。

定理 E.1. 对于任意给定的正整数 $k_1, ..., k_N$，存在一个具有大小约束 $k_1, ..., k_N$ 的单调 *SMP* 实例，即：

$$\max_{S \in \mathcal{P}} f(S)$$

其中 $\mathcal{P} := \{S \subseteq U : \ |S \cap U_i| \leq k_i\}$，标准贪心算法所能达到的最佳近似比为 $1/2$。

我们将定理 E.1 的详细证明放在补充材料的 E.4 节中。下面通过构造一个简单示例，对证明作简要说明。假设 $U = [8]$ 被分为两个组 $U_1 = \{1, 2, 3, 4\}$、$U_2 = \{5, 6, 7, 8\}$。设 $t : \ U \to M$ 是一个为全集中的每个元素分配标签的函数：$t(1) = t(5) = t(7) = t(8) =$ "a"，$t(2) = t(6) =$ "b"，$t(3) =$ "c"，以及 $t(4) =$ "d"。定义集合覆盖函数 f，将一个子集映射为其所覆盖的不同标签数量，即 $f(S) = |\cup_{s \in S} t(s)|$。这里，分划拟阵定义为 $k_1 = k_2 = 2$。在此情况下，标准贪心算法可能首先选择元素 1 和 2。随后，所有剩余元素要么变得不可行，要么边际增益为零，从而得到解集 S，其 $f(S) = 2$；而最优解集 $OPT = 3$、4、5、6 可达到 $f(OPT) = 4$。因此，$f(S) = f(OPT)/2$。由此即可得证。

这个例子凸显了标准贪心算法的一个关键局限：该算法在每一步都从所有可行元素中搜索，并贪心地添加边际增益最高的元素，这可能导致某些分区迅速达到其基数上限，使这些子组中剩余的元素在后续选择中变得不可行。这种策略使标准贪心算法无法获得更好的近似比。

事实上，在现有研究提出的大多数连续方法中，[Badanidiyuru 和 Vondr´ak（2014 年）、Calinescu 等人 2011],，实现 $1-1/e$最优近似比的关键思路，是在每一步将某些坐标逐渐增加一小部分。该技术确保底集 U中的所有元素在算法运行的大部分时间内都保持可行。受此启发，Block-Greedy 算法会在每一轮中，仔细平衡不同分区内加入解集的元素数量。具体而言，Block-Greedy 在每一轮从各分区中选取的元素数量 i与该分区的预算容量 k_i 成正比，从而确保这些元素在算法运行的大部分时间内保持可行，有效解决了这一局限。下一节将详细介绍我们为单调 SMP 提出的算法。

E.2.2 单调 SMP 的子程序算法

我们提出子程序算法 Greedy-Subroutine-Mono（算法 7），将其与参数 $\phi = \lfloor\sqrt{\min_{i \in [N]} k_i}\rfloor - 1$ 一起用于 Block-Greedy，并证明它可以用于获得至少与标准贪心算法一样好的近似解，而且通常严格优于标准贪心算法。此外，Block-Greedy 执行的 f 查询较少，具体取决于分区拟阵约束的结构。这里，参数 r_j 定义为：对每个 $j \in [N]$，$r_j := \lfloor k_j/\phi \rfloor$。

算法 7 Greedy-Subroutine-Mono (S, i, j)

输入： S, i, j **对于** $l = 1$ **到** r_j **，**

执行 $S \leftarrow S \cup \arg\max_{x \in U_j} \Delta f(S, x)$

定理 E.2。 假设对一个单调 *SMP* 实例运行 *Block-Greedy*，并使用算法 7 中所述的 *Greedy-Subroutine-Mono* 子程序算法，则 *Block-Greedy* 输出的解集为 S，其近似比为 $1 - 1/e - \frac{1}{\phi+1}$，其中 $\phi = \lfloor\sqrt{\text{最小值}_{i \in [N]} k_i}\rfloor - 1$。

直观而言，当参数 k_i 较大时，Block-Greedy 可实现接近 $1 - 1/e$ 的近似。定理 E.2 中出现涉及 ϕ 的项，是因为 Block-Greedy 的外层循环总共包含 ϕ 轮次。具体而言，若 $\phi \geq 7$，则定理 E.2 中所述的近似保证严格优于 1/2。为进一步确保该界优于 1/2，我们可以通过块贪心算法，不断向其返回解中贪心地添加边际增益最大的元素，直到解集的基数达到分划拟阵的秩；此时，Block-Greedy 的近似比为 $\max\{1/2, 1 - 1/e - \frac{1}{\phi+1}\}$（证明见附录 E.5）。

值得注意的是，Block-Greedy 的近似比与 $1-1/e$ 的最优结果之间的差距至多为 $\mathcal{O}(\frac{1}{\sqrt{k_{\min}}})$，其中 $k_{\min}=\min_{i\in[N]}k_i$ 具体而言，在某些 $k_1=k_2=\cdots=k_N$ 的情况下，该界还可进一步改进为 $\mathcal{O}(\frac{1}{k_1})$，如下列推论所示：

推论 E.3. 假设对单调 *SMP* 实例运行使用 *Greedy-Subroutine-Mono* 子程序的 *Block-Greedy*，且参数为 $k_1=k_2=...=k_N$。如果我们对每个 j设置 $\phi=k_1$ 和 $r_j=1$，那么 *Block-Greedy* 将输出一个近似比为 $1-1/e-1/k_1$的解集 S。

该推论可通过应用定理 E.8 证明；该定理在附录 E.5 中给出并进行了分析。

与标准贪心算法相比，Block-Greedy 的另一个重要优势是其查询复杂度可能要低得多。之所以能有这一改进，是因为 Block-Greedy 只在一个分区内选择边际增益最大的元素，而不是在整个全集 U中进行选择。具体而言，Block-Greedy 的查询复杂度上界为 $\sum_{i\in[N]}|U_i|k_i\leq n(\sum_{i\in[N]}k_i)$。

接下来，我们给出该定理的证明。首先，我们证明引理 E.4 中的结果。

引理 E.4。我们将 $\{S\subseteq U:\ |S\cap U_j|\leq r_j\phi\}$ 的分划拟阵定义为 $\mathcal{P}'$，并将问题 $\max_{S\in\mathcal{P}'}f(S)$ </style} 的最优解定义为 OPT'。将算法 7 的输入和输出分别记为 S和 S'，则有

$$f(S')-f(S)\geq\frac{\Delta f(S',OPT_j)}{\phi},$$

其中 $OPT_j=OPT'\cap U_j$。

证明。为简化记号，我们将算法 7 的 for 循环执行第 l 步后的解集定义为 $S^{(l)}$。根据第 3 行的贪心选择步骤，对于任意 $o\in OPT_j:=OPT'\cap U_j$，有

$$f(S^{(l)})-f(S^{(l-1)})\geq\Delta f(S^{(l-1)},o).$$

因此,

$$\begin{aligned}f(S^{(l)})-f(S^{(l-1)})&\geq\frac{\sum_{o\in OPT_j}\Delta f(S^{(l-1)},o)}{|OPT_j|}\\&\geq\frac{\sum_{o\in OPT_j}\Delta f(S^{(l-1)},o)}{r_j\phi}\\&\geq\frac{\sum_{o\in OPT_j}\Delta f(S^{(r_j)},o)}{r_j\phi}\\&\geq\frac{\Delta f(S^{(r_j)},OPT_j)}{r_j\phi},\end{aligned}$$

其中，第二个不等式由以下事实得出：$OPT'\in\mathcal{P}'$，因此 $|OPT_j|\leq r_j\phi$。对所有$l\in[r_j]$求和，可得

$$f(S^{(r_j)})-f(S^{(0)})\geq\frac{\Delta f(S^{(r_j)},OPT_j)}{\phi}.$$

注意，$S^{(0)}$ 是算法的输入，$S^{(l)}$ 是算法的输出，因此我们可以证明该结果。 □

利用引理 E.4中的结果，我们可以证明定理 E.2中的结论。

证明。令 $S_{i,j}$ 表示在算法 6 的第 4 行外层 for 循环的第 i 次迭代中，对第 j 个子组执行贪心子程序算法 Greedy-Subroutine-Mono 后得到的解集，并定义 S_i 为完成算法 6 外层 for 循环的第 i 轮后得到的解集，即 $S_i=S_{i,N}$。根据引理 E.4 的结果，我们有wehavethat

$$f(S_{i,j})-f(S_{i,j-1})\geq\frac{\Delta f(S_{i,j},OPT_j)}{\phi}$$

由于 $S_{i,j} \subseteq S_{i,N}$对任意 $j \in [N]$均成立，由子模性可得 $\Delta f(S_i, OPT_j) = \Delta f(S_{i,N}, OPT_j) \leq \Delta f(S_{i,j}, OPT_j)$。然后

$$f(S_{i,j}) - f(S_{i,j-1}) \geq \frac{\Delta f(S_i, OPT_j)}{\phi}$$

对所有 j求和后，可得

$$\begin{aligned} f(S_{i,N}) - f(S_{i,0}) &\geq \frac{\sum_{j\in[N]} \Delta f(S_i, OPT_j)}{\phi} \\ &\geq \frac{\Delta f(S_i, OPT')}{\phi}, \end{aligned}$$

其中，最后一个不等式由子模性以及这一事实 $OPT' = \cup_{j\in[N]} OPT_j$可得。请注意，此处$S_{i,\ N}$等价于 S_i，而 $S_{i,\ 0}$ 等价于 S_{i-1}。于是得到

$$\begin{aligned} f(S_i) - f(S_{i-1}) &\geq \frac{f(OPT' \cup S_i) - f(S_i)}{\phi} \\ &\geq \frac{f(OPT') - f(S_i)}{\phi}. \end{aligned}$$

通过整理不等式并运用归纳法，可得

$$f(S_\phi) - f(\emptyset) \geq (1 - (\frac{\phi}{\phi+1})^\phi) f(OPT').$$

根据 ϕthat $\phi = \lfloor \sqrt{\min_{i\in[N]} k_i} \rfloor - 1$的定义，对于任意 $i \in [N]$，都有 $k_i - \phi\lfloor k_i/\phi \rfloor \leq \lfloor k_i/\phi \rfloor$。由引理 E.7 可得

$$\max_{S\in\mathcal{P}'} f(S) \geq \frac{\phi}{\phi+1} \max_{S\in\mathcal{P}} f(S).$$

因此,

$$\begin{aligned} f(S_\phi) &\geq (1 - (\frac{\phi}{\phi+1})^\phi) f(OPT') \\ &\geq (1 - (\frac{\phi}{\phi+1})^\phi)(\frac{\phi}{\phi+1}) f(OPT) \\ &\geq (1 - e^{-1} - \frac{1}{\phi+1}) f(OPT). \end{aligned}$$

□

E.3 非单调 SMP

在本节中，我们提出了针对分区拟阵上的非单调子模最大化（SMP）问题的算法。所提出的算法遵循算法 6 的框架，其中 $\phi = \lfloor \sqrt{\min_{i\in[N]} k_i} \rfloor - 1$，子程序算法 Greedy-Subroutine-Nonmono 如算法 8 所示。这里参数$r_j := \lfloor k_j/\phi \rfloor$。该算法从边际增益排名前 $r_j\phi$ 的元素集合中均匀选取一个元素，并将其加入解集。Greedy-Subroutine-Nonmono 算法背后的直觉与 Buchbinder 等人提出的随机贪心算法 [2014]类似。然而，在 Greedy-Subroutine-Nonmono 中，为确保一个重要结果 $\mathbb{E}[f(S_i \cup OPT')] \geq (1 - \frac{1}{\phi})^i f(OPT')$，将考虑纳入解集的候选集大小调整为 $r_j\phi$，其中 $\mathcal{P}' = \{S \subseteq U\colon\ |S \cap U_i| \leq r_i\phi,\ \forall i \in [N]\}$ 以及 $OPT' = \arg\max_{S\in\mathcal{P}'} f(S)$。

下面我们给出 Block-Greedy 针对非单调 SMP 问题的主要结果。

算法 8 Greedy-Subroutine-Nonmono (S, i, j)

1: 输入： S，i，j2: **对于** $l = 1$ **至** r_j **执行**3: 令 $M \subseteq U/S$为一个大小为 $r_j\phi$、使 $\sum_{x\in M} \Delta f(S,\ x)$ 最大化的集合。4: 从 M 中 $u \leftarrow$ 均匀采样一个元素5: $S \leftarrow S \cup \arg\max_{x\in U_j} \Delta f(S,\ x)$

定理 E.5。 假设对一个非单调 *SMP* 实例运行 *Block-Greedy*，并使用算法 *7* 中所述的 *Greedy-Subroutine-Nonmono* 子程序，则 *Block-Greedy* 输出一个解 S，其近似比为 $\frac{1}{e} - \frac{3}{e^{(\phi+1)}}$，其中 $\phi = \sqrt{\min_{i\in[N]} k_i} - 1$ （期望意义下）。

请注意，该近似比接近 $1/e$，与基数约束下次模最大化的随机贪心算法的界相匹配，其差异由 $\mathcal{O}(\frac{1}{\sqrt{k_{\min}}})$ 界定，其中 $k_{\min} = \min_{i\in[N]} k_i$。从这个意义上说，所提出的算法为非单调 SMP 实现了一个近似比，弥合了基数约束和分区拟阵约束下次模最大化之间的差距。定理 E.5 的证明如下。

我们定义 $\mathcal{P}' = \{S \subseteq U\colon\ |S \cap U_i| \le r_i\phi,\ \forall i \in [N]\}$ 并将 $OPT' = \arg\max_{S\in\mathcal{P}'} f(S)$ 记为最优解集。首先，我们证明 Greedy-Subroutine-Nonmono 子程序算法的以下引理 .

引理 E.6. 将算法 *8* 的输入和输出分别记为 S 和 S'，则可得

$$\mathbb{E}[f(S') - f(S)] \ge \frac{\mathbb{E}[\Delta f(S', OPT_j)]}{\phi},$$

其中 $OPT_j = OPT' \cap U_j$。

证明。 将添加第 l 个元素后的解集记为 $S^{(l)}$。与单调 SMP 的定理 E.2 的证明类似，我们有

$$\mathbb{E}[f(S^{(l)}) - f(S^{(l-1)})] \ge \frac{\sum_{a\in OPT'\cap U_j} \Delta f(S^{(l-1)}, a)}{r_j\phi}.$$

根据子模性，可得

$$\begin{aligned}\mathbb{E}[f(S^{(l)}) - f(S^{(l-1)})] &\ge \mathbb{E}[\frac{\sum_{a\in OPT'\cap U_j} \Delta f(S^{(l-1)}, a)}{r_j\phi}] \\ &\ge \mathbb{E}[\frac{\sum_{a\in OPT'\cap U_j} \Delta f(S', a)}{r_j\phi}].\end{aligned}$$

对所有 $l \in [r_j]$ 求和，可得

$$\begin{aligned}\mathbb{E}[f(S') - f(S)] &\ge \mathbb{E}[\frac{\sum_{a\in OPT'\cap U_j} \Delta f(S', a)}{\phi}] \\ &\ge \mathbb{E}[\frac{\Delta f(S', OPT_j)}{\phi}].\end{aligned}$$

□

接下来，利用引理 E.6 中的结果，我们证明定理 E.5 中的近似比。

证明。 与定理 E.2 的证明类似，我们将算法 6 第 4 行外层 for 循环第 i 轮中，在第 j 个子组运行子程序算法后得到的解集记为 $S_{i,\ j}$，并将算法 6 完成第 i 轮后的解集定义为 S_i。根据引理 E.6，我们有

$$\begin{aligned}\mathbb{E}[f(S_{i,j}) - f(S_{i,j-1})] &\ge \frac{\mathbb{E}[\Delta f(S_{i,j}, OPT_j)]}{\phi} \\ &\ge \frac{\mathbb{E}[\Delta f(S_i, OPT_j)]}{\phi},\end{aligned}$$

对所有 $j \in [N]$ 求和可得

$$\begin{aligned}\mathbb{E}[f(S_i) - f(S_{i-1})] &\geq \frac{\sum_{j=1}^{N} \mathbb{E}[\Delta f(S_i, OPT_j)]}{\phi} \\ &\geq \frac{\mathbb{E}[\Delta f(S_i, OPT')]}{\phi}.\end{aligned}$$

于是可得

$$\mathbb{E}[f(S_i) - f(S_{i-1})] \geq \mathbb{E}[\frac{f(S_i \cup OPT') - f(S_i)}{\phi}].$$

注意，根据贪心选择步骤，对于每个组 j 以及每个元素 $a \in OPT' \cap U_j$ 出现在 S_i 中的概率至多为 $1-(1-\frac{1}{r_j\phi})^{r_j i}$。由于 $(1-\frac{1}{x})^x$ 在 $[1,+\infty)$ 的范围内随 x 递增，因此有 $(1-\frac{1}{r_j\phi})^{r_j\phi} \geq (1-\frac{1}{\phi})^{\phi}$。因此，我们得到 $1-(1-\frac{1}{r_j\phi})^{r_j i} \leq 1-(1-\frac{1}{\phi})^{i}$。根据 Buchbinder 等人的引理 2.2[2014]，我们可以得出

$$\mathbb{E}[f(S_i \cup OPT')] \geq (1-\frac{1}{\phi})^i f(OPT').$$

通过重新整理上述不等式，可得

$$\mathbb{E}[f(S_i)] \geq \frac{\phi}{\phi+1}\mathbb{E}[f(S_{i-1})] + \frac{1}{\phi+1}(1-\frac{1}{\phi})^i f(OPT').$$

根据归纳法，输出解集满足

$$\begin{aligned}\mathbb{E}[f(S)] &= \mathbb{E}[f(S_\phi)] \\ &\geq \frac{1}{\phi+1}\sum_{i=1}^{\phi}(\frac{\phi}{\phi+1})^{\phi-i}(1-\frac{1}{\phi})^i f(OPT') \\ &\geq \frac{1}{\phi+1}\sum_{i=1}^{\phi}(\frac{\phi-1}{\phi})^{\phi-i}(1-\frac{1}{\phi})^i f(OPT') \\ &\geq \frac{\phi}{\phi+1}(1-\frac{1}{\phi})^{\phi} f(OPT') \geq \frac{1}{e}(1-\frac{2}{\phi+1}) f(OPT').\end{aligned}$$

其中，最后一个不等式是由以下事实得到的：对于任意 $\phi > 1$，$(1-\frac{1}{\phi})^{\phi-1} \geq e^{-1}$。根据定义 $\phi = \lfloor \sqrt{\min_{i\in[N]} k_i} \rfloor - 1$，可得对于任意 $i \in [N]$，$k_i - \phi\lfloor k_i/\phi \rfloor \leq \lfloor k_i/\phi \rfloor$。因此，根据引理 E.7 的结果，可得

$$f(OPT') \geq \frac{\phi}{\phi+1} f(OPT) \tag{10}$$

其中，OPT 是子模最大化问题 $\arg\max_{S\in\mathcal{P}} f(S)$ 的最优解。结合 (3) 和 (10)，即可证明引理中的结论。

□

引理 E.7。假设底集 U 被划分为 N 个互不相交的子组 $U_1, U_2, ..., U_N$，那么对于任意分划拟阵 $\mathcal{P} = \{S \subseteq U: |S \cap U_j| \leq k_j, \forall j \in [N]\}$，定义拟阵 $\mathcal{P}' = \{S \subseteq U: |S \cap U_j| \leq \lfloor \frac{k_j}{c} \rfloor c, \forall j \in [N]\}$，其中 c 为某个正整数。若对于任意 $j \in [N]$，均满足 $\lfloor \frac{k_j}{c} \rfloor \geq k_j - c\lfloor \frac{k_j}{c} \rfloor$，则对于任意子模函数 f，有

$$\max_{S\in\mathcal{P}'} f(S) \geq \frac{c}{c+1} \max_{S\in\mathcal{P}} f(S).$$

证明。为简化记号，我们还定义 $r_j = \lfloor \frac{k_j}{c} \rfloor$，其中 $j \in [N]$，并定义两个拟阵 $\mathcal{P}_0 = \{S \subseteq U: |S \cap U_j| \leq r_j, \forall j \in [N]\}$ 和 $\mathcal{P}_1 = \{S \subseteq U: |S \cap U_j| \leq r_j(c+1)\ \forall j \in [N]\}$。将最优解记为

$\max_{S\in\mathcal{P}_1} f(S)$ 的解为 OPT''。根据定义可知，OPT'' 可划分为 $(c+1)$ 个互不相交子集O_1，.....，O_{c+1}，使得每个 $O_i \in \mathcal{P}_0$。不失一般性，我们假设这些子集是贪心选取的，因此对于任意 $1 \le i \le c$ 和 $l > i$，其索引满足 $\Delta f(\cup_{j\in[i-1]}O_j,\ O_i) \ge \Delta f(\cup_{j\in[i-1]}O_j,\ O_l)$。由此根据子模性可得，对于任意 $l > i$，$\Delta f(\cup_{j\in[i-1]}O_j,\ O_i) \ge \Delta f(\cup_{j\in[i-1]}O_j,\ O_l) \ge \Delta f(\cup_{j\in[l-1]}O_j,\ O_l)$。因此，

$$\begin{aligned} f(OPT'') - f(\cup_{j\in[c]}O_j) &= \Delta f(\cup_{j\in[c]}O_j, O_{c+1}) \\ &\le \frac{\sum_{i\in[c]}\Delta f(\cup_{j\in[i-1]}O_j, O_i)}{c} \\ &= \frac{f(\cup_{j\in[c]}O_j) - f(\emptyset)}{c}. \end{aligned}$$

通过重新整理上述不等式，我们得到 $f(\cup_{j\in[c]}O_j) \ge \frac{c}{c+1} f(OPT'')$。由于 $\cup_{j\in[c]}O_j \in \mathcal{P}'$，因此有 $\max_{S\in\mathcal{P}'} f(S) \ge \frac{c}{c+1} f(OPT'')$。注意，$\lfloor \frac{k_j}{c} \rfloor (c+1) \ge k_j$意味着对于任意子集 $S \in \mathcal{P}$，它也满足 $S \in \mathcal{P}_1$。因此，$\max_{S\in\mathcal{P}_1} f(S) \ge \max_{S\in\mathcal{P}} f(S)$，由此我们可以得出证明。

□

E.4 定理 E.1 的证明

本节中，我们给出定理 E.1 被省略的证明。为证明该定理，我们构造一类实例，以说明标准贪心算法无法达到优于 $1/2$ 的近似比。我们首先介绍集合函数和约束的相关定义，随后正式描述困难实例。

假设底集 U 被划分为 N 个互不相交子集 U_1，U_2，...，U_N，其中每个子集 U_i 包含 $2k_i$ 个元素。定义集合函数 $t: U \to 2^M$，其中 M 是一个有限集，并令 $c: 2^M \to \mathbb{R}_+$ 为一个非负、单调的模函数。我们将子模函数 $f: 2^U \to \mathbb{R}_+$ 定义如下：

$$f(S) = c\left(\bigcup_{s\in S} t(s)\right) = \sum_{x\in\bigcup_{s\in S} t(s)} c(x).$$

将分划拟阵记为 $\mathcal{P} = \{S \subseteq U: |S \cap U_i| \le k_i,\ \forall i \in [N]\}$。不失一般性，假设分划拟阵约束参数满足 $k_1 \le k_2 \le, \ ... \le k_N$。为简化记号，将组 U_i 中的第 j 个元素记为 $s_j^{(i)}$。困难性示例如下定义。假设 ϵ 是一个常数，使得 $\epsilon \in (0,\ 1/2)$，则

1. **中的集合赋值**：在第一个分区 U_1 中，假设对于任意不同的 j_1，j_2，集合 $t(s_{j_1}^{(1)})$ 和 $t(s_{j_2}^{(1)})$ 互不相交，即 $t(s_{j_1}^{(1)}) \cap t(s_{j_2}^{(1)}) = \emptyset$。
2. **中的函数值**：为 U_1 中的元素设置模函数值如下：
   - For $j \le k_1$, $c(t(s_j^{(1)})) = \frac{1}{2} + \epsilon$.
   - For $k_1 < j \le 2k_1$, $c(t(s_j^{(1)})) = \frac{1}{2}$.
3. $i > 1$**的集合赋值和函数值**：对于 i_1，i_2 使得 $1 < i_1,\ i_2 \le N$定义：
   - If $j \le k_1$, then $t(s_j^{(i_1)}) = t(s_j^{(i_2)}) \subseteq t(s_j^{(1)})$.
   - If $j > k_1$, then $t(s_j^{(i_1)}) = t(s_j^{(i_2)}) = t(s_1^{(i_1)})$.
   - Set $c(t(s_j^{(i)})) = \frac{1}{2}$ for any $i > 1$ and $j \in [2k_i]$.

根据这一构造，标准贪心算法的执行过程如下：在前 k_1 步中，算法会从 U_1中选取元素 $s_1^{(1)}$、$s_2^{(1)}$..... $s_{k_1}^{(1)}$，每一步的边际增益为 $\frac{1}{2} + \epsilon$。因此，经过这几步后，总值为$\frac{k_1}{2} + k_1\epsilon$。前 k_1 步之后，算法只能从满足 $i > 1$的分区 U_i 中选取元素。由于当前集合赋值下 f 的结构，每一步的边际增益均为 0。因此，标准贪心算法返回的次模目标函数值为 $\frac{k_1}{2} + k_1\epsilon$。

接下来，我们考虑所构造示例的最优解。注意，对于任意满足 $i > 1$的分区 U_i， $f(U_i) = k_1/2$。此外，根据构造，对于任意 i_1、$i_2 > 1$，都有 $\bigcup_{s\in U_1} t(s) = \bigcup_{s\in U_2} t(s)$。因此， $f(\bigcup_{i>1} U_i) = k_1/2$。由此可知，对于任意子集 $S \subseteq U$，

$$f(S \cap \bigcup_{i>1} U_i) \leq k_1/2. \tag{11}$$

接下来，我们声称最优解应满足 $f(OPT) \leq k_1$。我们通过考虑以下情况来证明这一主张。对于任意 $S \in \mathcal{P}$，则

1. 若 $f(S \cap \bigcup_{i>1} U_i) = k_1/2$，这意味着 $\bigcup_{s\in S\cap\bigcup_{i>1} U_i} t(s) = \bigcup_{j\leq k_1} t(s_j^{(2)})$，那么对于每个 $j \leq k_1$，将第一分区中的第 j 个元素加入集合 $S \cap \bigcup_{i>1} U_i$ 所产生的边际增益满足 $\Delta f(S \cap \bigcup_{i>1} U_i, s_j^{(1)}) = c(t(s_j^{(1)})) - c(t(s_j^{(2)})) = \epsilon$；并且对于每个 $j > k_1$， $\Delta f(S \cap \bigcup_{i>1} U_i, s_j^{(1)}) = f(s_j^{(1)}) = 1/2$。由此可得 $\Delta f(S \cap \bigcup_{i>1} U_i, S \cap U_1) \leq \sum_{s\in S\cap U_1} \Delta f(S \cap \bigcup_{i>1} U_i, \Delta f\ S \cap \bigcup_{i>1} U_i\ S \cap U_1\ + f\ S \cap \bigcup_{i>1} U_i\ \leq k_1)$ $\leq k_1/2$。因此， $f(S)$ (,) ( )。
2. 若 $f(S \cap \bigcup_{i>1} U_i) < k_1/2$，则可知，至少存在一个元素 $s_j^{(2)}$，对于 $j \leq k_1$，使得集合 $t(s_j^{(2)}) \notin \bigcup_{s\in S\cap\bigcup_{i>1} U_i} t(s)$。记 $E = \{j \leq k_1 : t(s_j^{(2)}) \notin S \cap \bigcup_{i>1} U_i\}$。由此可得 $f(S \cap \bigcup_{i>1} U_i) = \frac{k_1-|E|}{2}$。对于每个 $j \leq k_1$，若 $j \in E$，则 $\Delta f(S \cap \bigcup_{i>1} U_i, s_j^{(1)}) = c(t(s_j^{(1)})) = 1/2 + \epsilon$；若 $j \notin E$，则 $\Delta f(S \cap \bigcup_{i>1} U_i, s_j^{(1)}) = c(\overline{t(s_j^{(1)})}) - c(t(s_j^{(2)})) = \epsilon$。对于每个 $j > k_1$，有 $\Delta f(S \cap \bigcup_{i>1} U_i, s_j^{(1)}) = f(s_j^{(1)}) = 1/2$。类似地，有 $\Delta f(S \cap \bigcup_{i>1} U_i, S \cap U_1) \leq \sum_{s_j^{(1)}\in S\cap U_1} \Delta f(S \cap \bigcup_{i>1} U_i, s_j^{(1)}) \leq |E|(1/2 + \epsilon) + (k_1 - |E|)/2 = |E|\epsilon + k_1/2$。因此，我们可以得出 $f(S) = \Delta f(S \cap \bigcup_{i>1} U_i, S \cap U_1) + f(S \cap \bigcup_{i>1} U_i) \leq k_1 - |E|(1/2 - \epsilon) \leq k_1$。

由此可得， $f(\text{OPT}) \leq k_1$。注意，集合 $O = \{s_{k_1+1}^{(1)}, ...s_{2k_1}^{(1)}, s_1^{(2)}, ..., s_{k_1}^{(2)}\}$ 的目标值为： $f(O) = k_1$。因此， $f(\text{OPT}) = k_1$。由此，标准贪心算法的近似比应为 $\frac{k_1/2+k_1\epsilon}{k_1} = 1/2 + \epsilon$。当 ϵ 趋近于 0 时，近似比趋近于 $1/2$。

E.5 关于定理 E.2 的讨论

在本附录部分，我们阐释定理 E.2 的结果。首先，我们讨论所提出的算法 Block-Greedy 的近似比与先前连续方法所达到的最优近似比 $1 - 1/e$之间的差异。具体而言，该差异为 $\mathcal{O}(\frac{1}{\sqrt{k_{\min}}})$，其中 $k_{\min} = \min_{i\in[N]} k_i$。事实上，我们注意到，这一差异源于其按 $\mathcal{O}(\frac{1}{\phi})$ 的阶缩放。在第 E.2 节中以 Greedy-Subroutine-Mono 为子程序的算法 Block-Greedy 中， ϕ 被设为 $\phi = \lfloor\sqrt{k_{\min}}\rfloor - 1$，旨在界定 k_i 与 $\lfloor\frac{k_i}{\phi}\rfloor\phi$ 之间的差异，从而确保单调 $\max_{S\in\mathcal{P}'} f(S)$ 的最优值能够近似 $\max_{S\in\mathcal{P}} f(S)$ 的最优值，其中 $\mathcal{P}' := \{S \subseteq U : |S\cap U_i| \leq \lfloor\frac{k_i}{\phi}\rfloor\phi,\ \forall i \in [N]\}$以及 $\mathcal{P} := \{S \subseteq U : |S \cap U_i| \leq k_i,\ \forall i \in [N]\}$。

这引出了以下结果：在某些情况下，如果我们能设计参数 ϕ，使得对于任意 $i \in [N]$都有 $\lfloor\frac{k_i}{\phi}\rfloor = \frac{k_i}{\phi}$，那么分划拟阵 $\mathcal{P} = \mathcal{P}'$，并且我们无需界定 $\max_{S\in\mathcal{P}'} f(S)$ 与 $\max_{S\in\mathcal{P}} f(S)$ 之间的差异。因此，我们可以进一步细化 Block-Greedy 的近似比与 $1 - 1/e$ 的最优结果之间的差异。结果如下所述。

定理 E.8。 假设 $gcd(k_1, k_2, \ldots, k_N) = c$，并且针对分划拟阵 $\mathcal{P} := \{S \subseteq U : |S \cap U_i| \leq k_i, \forall i \in [N]\}$ 上的单调 *SMP* 实例，运行以 *Greedy-Subroutine-Mono* 为子程序的 *Block-Greedy*，其中每个 $j \in [N]$ 都满足 $\phi = c$ 和 $r_j = k_j/c$，那么 *Block-Greedy* 输出的解集 S 满足近似比 $1 - 1/e - 1/c$。

证明。 按照定理 E.2 的类似证明，我们可以得到输出解集 S 满足

$$f(S) - f(\emptyset) \geq (1 - (\frac{\phi}{\phi+1})^{\phi}) f(OPT'),$$

其中 $OPT' = \arg\max_{S\in\mathcal{P}'} f(S)$ 且 $\mathcal{P}' := \{S \subseteq U\colon |S \cap U_i| \le r_i\phi,\ \forall i \in [N]\}$。根据此情形下对 r_i 和 ϕ 的赋值，我们可以得到 $r_i\phi = k_i$。由此可得 $\mathcal{P}' = \mathcal{P}$，且 OPT' 也是我们问题的最优解。因此，

$$\begin{aligned} f(S) &\ge (1 - (\frac{\phi}{\phi+1})^{\phi}) f(OPT) \\ &\ge (1 - 1/e - 1/\phi) f(OPT) = (1 - 1/e - 1/c) f(OPT). \end{aligned}$$

□

特别地，若 $c = \mathcal{O}(k_{\min})$，例如 $k_1 = k_2 =, \ ... = k_N = k$ 的情形，则近似比为 $1 - 1/e - 1/k$。因此，近似比与最优近似比之间的差距缩小至$\mathcal{O}(\frac{1}{k_{min}})$。该结果见推论 E.3

接下来，我们证明，通过向解集中添加更多元素，可以改进块贪心算法的近似比。首先，我们注意到，与标准贪心算法相比，所提出的 Block-Greedy 算法有两个缺点。首先，$1 - 1/e - \frac{1}{\lfloor\sqrt{\min_{i\in[N]} k_i}\rfloor}$ 的近似比为

只有当分区 U_i 中的容量 k_i 满足对每个 $i \in [N]$ 都有 $k_i \ge 64$ 时，其近似比才优于标准贪心算法的近似比，即 $1/2$。

其次，输出解满足：$|S \cap U_i| \le r_i\phi$ 对于每个 $i \in [N]$。注意到，$r_i\phi \le k_i$。如果 $r_i\phi < k_i$，我们可以向解集 S 中添加更多元素，直到其达到分划拟阵的满秩。由于目标函数是单调的，我们可以看出，添加更多元素不会导致边际增益下降。在接下来的部分中，我们声称，如果将标准贪心过程（算法 9）应用于以 Greedy-Subroutine-Mono 为子程序的 Block-Greedy 的输出，则所得解集可达到 $\max\{1/2, 1 - 1/e - \frac{1}{\phi+1}\}$ 的近似比。

算法 9 贪心

1: 输入： 通过运行以 Greedy-Subroutine-Mono 为子程序的 Block-Greedy 获得的输出解集 S 以及 $\phi = \lfloor\sqrt{\min_{i\in[N]} k_i}\rfloor - 1$ 和 $r_j := \lfloor k_j/\phi \rfloor$2: **输出：** $A \in U$3: $A \leftarrow S$4: **当** $\exists x$ 使得 $A \cup \{x\} \in \mathcal{P}$ **时执行**
5: $A \leftarrow A \cup \arg\max_{x\in U,\ A\cup\{x\}\in\mathcal{P}} \Delta f(A,\ x)$ **返回** A

定理 E.9. 假设我们运行算法 *9* 中的标准贪心算法，并将块贪心算法的输出解集作为输入，那么输出解集的近似比为$\max\{1/2, 1 - 1/e - \frac{1}{\phi+1}\}$ 其中 $\phi = \lfloor\sqrt{\min_{i\in[N]} k_i}\rfloor - 1$。

证明。首先注意到 $S \subseteq A$，根据定理 E.2 的结果，我们可以看到 $f(S) \ge (1 - 1/e - \frac{1}{\phi+1}) f(OPT)$。由于 f 是单调的，$f(A) \ge f(S) \ge (1 - 1/e - \frac{1}{\phi+1}) f(OPT)$。因此，要证明定理 E.9 中的结论，只需证明 $f(A) \ge f(OPT)/2$。这里我们使用与定理 E.2 的证明中相同的记号，也就是说，将 $\{S \subseteq U$ 的分划拟阵定义为：$|S \cap U_j| \le r_j\phi\}$ 为 $\mathcal{P}'$，并将问题 $\max_{S\in\mathcal{P}'} f(S)$ 的最优解定义为 OPT'。将算法 6 第 4 行外层 for 循环完成第 i 轮后的解集记为 S_i。沿用定理 E.2 证明中的类似思路，我们可以看到，对于任意 $O \in \mathcal{P}'$，都有

$$\begin{aligned} f(S_i) - f(S_{i-1}) &\ge \frac{\Delta f(S_i, O)}{\phi} \\ &\ge \frac{\Delta f(S, O)}{\phi}, \end{aligned}$$

其中最后一个不等式 性由子模性以及事实 $S_i \subseteq S_\phi = S$ 得出。对所有 i 求和后，我们得到 t

$$f(S) - f(\emptyset) \ge \Delta f(S, O). \tag{12}$$

定义分划拟阵 $\mathcal{P}'' := \{S \subseteq U\colon |S \cap U_i| \leq k_i - r_i\phi,\ \forall i \in [N]\}$。定义算法 9 中第 i 轮之前的解集 A 为 A_i，并将第 i 轮加入的元素记为 a_i。由于 $\mathcal{P}''$ 是拟阵，对于任意 $O' \in \mathcal{P}''$，存在 $O' = \{o'_1, o'_2, ..., o'_t\}$ 的一个排序，使得对每个 $i \in [t]$，都有 $A_i/S \cup \{o'_i\} \in \mathcal{P}''$。因此，对每个 $i \in [t]$，都有 $A_i \cup \{o'_i\} \in \mathcal{P}$。根据算法 9 中的贪心选择规则，我们有

$$\Delta f(A_i, a_i) \geq \Delta f(A_i, o'_i) \geq \Delta f(A, o'_i),$$

其中，第二个不等式由以下事实得出： A是算法 9 的输出，并且 $A_i \subseteq A$。对所有 i求和，可得

$$f(A) - f(S) \geq \sum_i \Delta f(A, o'_i) \geq \Delta f(A, O'). \tag{13}$$

将（12）和（13）相加，可得

$$\begin{aligned} f(A) - f(\emptyset) &\geq \Delta f(A, O') + \Delta f(S, O) \\ &\geq \Delta f(A, O') + \Delta f(A, O) \\ &\geq \Delta f(A, O' \cup O). \end{aligned}$$

由于上述不等式对任意 $O \in \mathcal{P}'$ 和 $O' \in \mathcal{P}''$均成立，因此，

$$f(A) - f(\emptyset) \geq \max_{O \in \mathcal{P}', O' \in \mathcal{P}''} \Delta f(A, O' \cup O).$$

注意， $\mathcal{P}$ 中的任意集合都可以分解为 $\mathcal{P}'$ 中的一个集合与 $\mathcal{P}''$中的一个集合的并。因此，$\max_{O \in \mathcal{P}',\ O' \in \mathcal{P}''} \Delta f(A,\ O' \cup O) \geq \Delta f(A,\ OPT)$。因此， $f(A) \geq f(OPT)/2$。 □

F 附录：第 3 节

在本节中，我们给出第 3 节中省略的其他实验设置和实验结果。具体而言，我们在 F.1 节中介绍实验设置的更多细节，并在 F.2 节中给出其他实验结果。

F.1 实验设置

本节将补充介绍用于评估我们算法的应用，包括集合覆盖、最大覆盖和图割。下文将定义每种应用，并详细描述相应的设置。

在集合覆盖应用中，函数 f 定义为子集中各元素所覆盖的标签数量。该问题定义如下。

定义 F.1。 (集合覆盖) 假设共有 n 个元素，记为 U。令 T为一个标签集合。 U 中的每个元素都通过函数 $t\colon U \to 2^T$与 T 中的一组元素相关联。函数 f 定义为

$$f(S) = |\cup_{s \in S} t(s)|, \qquad \forall S \in U.$$

接下来，我们引入最大覆盖的定义，它是在图上定义的单调子模函数。

定义 F.2. (最大割) 设 $G = (V,\ E)$ 是一个图， $w\colon E \to \mathbb{R}_{\geq 0}$ 是一个函数，为图中的每条边赋予权重。函数 $f\colon 2^V \to \mathbb{R}_{\geq 0}$ 将顶点子集 $X \subseteq V$ 映射为 X 与 $V \backslash X$ 之间边的总权重。更具体地说，

$$f(X) = \sum_{x \in X \text{ or } y \in X} w(x, y).$$

我们还在图像摘要实例上评估了实验。对于这项任务，我们使用 ImageNet 数据集的一个子集（ImageNet 50）。

定义 F.3。 (图像摘要) 令 $N \subseteq \mathbb{R}^d$ 表示底集，其中每个项目 $x \in N$（例如，一张图像）都由一个特征向量表示。目标是最大化 行列式点过程 *(DPP)* 函数 [Iyer 和 Bilmes， 2015], 该函数是一个定义如下的单调子模函数:

$$f(S) = \log\det(I + K_S),$$

其中， I 是单位矩阵， $K \in \mathbb{R}^{|N|\times|N|}$ 是半正定核矩阵，而 K_S 表示由子集 $S \subseteq N$索引的 K 的主子矩阵。

对于一般的 SCP（其中 f 可以是非单调的），我们考虑的应用场景是 f 为图割函数的情形；图割函数是一个次模但不一定单调的函数。

定义 F.4。 (图割) 设 $G = (V, E)$是一个图，且 $w\colon E \to \mathbb{R}_{\geq 0}$ 是一个为图中的每条边赋予权重的函数。函数 $f\colon 2^V \to \mathbb{R}_{\geq 0}$ 将顶点子集 $X \subseteq V$映射为连接 X 与 $V\backslash X$的边的总权重。更具体地说，

$$f(X) = \sum_{x\in X, y\in V\backslash X} w(x, y).$$

接下来，我们按照所考虑问题的顺序，进一步介绍实验设置。对于非单调 SCP 的实验，所用数据集为 email-EuAll 数据集，该数据集根据其合成标签划分为 5 个不同的子组。各组的比例均匀，即本实验中使用的参数 p_j 满足 $p_1 = p_2 = \cdots = p_5 = 1/5$。为加快实验速度，转换算法的子程序在 10 个线程上并行运行。实验中各参数取值的更多细节如下。参数 $\alpha = 0.2$、 $\epsilon = 0.05$ 和 $\delta = 0.1$。

对于单调 SCKP 的实验，我们在两个实例上进行实验，分别是最大覆盖和集合覆盖。对于最大覆盖实例，我们使用 Twitch Gamers 数据集的一个子集 [Rozemberczki 和 Sarkar， 2021],选取了 2,000 名使用六种主要语言的用户，这些语言包括英语、德语、法语、西班牙语、俄语和中文。对于集合覆盖实例，我们使用两个数据集。第一个是核心数据集，即 Duygulu 等人提出的 Corel5k 图像集 [2002]($n = 4500$)。我们为数据集中的每个元素分配一个标签，该标签从 $\{0, 1, 2, 3, 4\}$中均匀随机选取。我们使用的另一个数据集是合成数据集。该合成数据集由 5 个分区生成，每个分区包含 $40 * i + 200$ 个元素 i ，每个 $i \in [4]$。合成数据集的结构与附录 E.4 节中的紧致性示例类似。在第一个分区中，每个元素都映射到一个互不相交的标签集。对于元素分区 i，其中 $i > 1$，100 个元素所映射的标签集相同，其余元素则映射到互不相交的 25 个标签集。合成数据集和 Twitch 数据集中每个元素的成本均在 $[0.001, 10]$范围内随机生成。比较不同 τ取值的实验所用的其他参数包括： $\alpha = 0.2$、 $\epsilon = 0.05$。比较不同 α 取值的实验所用参数包括：阈值 $\tau = 700$、 $\epsilon = 0.05$。本实验中使用的参数 p_j 满足 $p_1 = p_2 = \cdots = p_5 = 1/5$。接下来，我们介绍实验中使用的两种算法。GREEDY 算法在算法 4 中使用转换定理，并以贪心算法作为子程序。具体而言，子程序贪心算法将元素 $s = \arg\max_{x:S\cup x\in\mathcal{P}} \Delta f(S,x)/c(x)$ 添加到解集 S中，其中 $\mathcal{P} = \{S \subseteq U\colon c(S \cap U_j) \leq p_j v, \forall j \in [N]\}$。这里，GREEDY 的子程序算法不保证任何近似比。从这个意义上说，该算法可以视为一种启发式算法。GREEDY-Knapsack 算法通过迭代添加边际增益密度最高的元素来运行，即 $s = \arg\max_{x\in U} \Delta f(S,x)/c(x)$，直到 $f(S) \geq (1 - \epsilon)\tau$。

对于 SCF 实验，我们采用与 SCKP 实验相同的合成数据集和 corel数据集。除这些数据集外，我们还考察图像摘要任务，其目标是从所有类别中选出一个多样且具有代表性的图像子集。此处使用的数据集是 ImageNet [Deng 等人， 2009], 包含 50 个类别和 26,112 张图像（ImageNet 50）。每张图像均由使用 ResNet-18 提取的特征向量表示。此外，在实验中，我们将 K 设为高斯核矩阵，使得 $K_{ij} = e^{-||x_i - x_j||^2/\sigma^2}$。

为了确保所用算法之间的公平比较，我们通过为 THRES-Fair 设置 $\varepsilon = 0.05$ 、为 GREEDY-Fair 和 BLOCK-G-Fair 设置 $\varepsilon = 0.1$ ，使函数值上的近似比 f保持一致，同时保持其他参数不变。

F.2 更多实验结果

不同算法在不同问题上的更多实验结果见图 2、3 和 4。SCKP 算法在 corel数据集和合成数据集上的更多实验结果

见图 2 和 3。结果表明，我们的算法 BLOCK-G 的成本略低于 GREEDY，并且在这方面显著优于贪心背包。此外，BLOCK-G 所需的函数查询大幅减少，运行时间也远快于 GREEDY，凸显了其实用效率。不同算法的函数值 f 相近，这是因为实验中的所有算法（包括基线算法）都设计为在函数值达到 $(1-\epsilon)\tau$后终止。对于 α 取不同值的实验，我们可以看到，增大 α 会显著减少评估次数，从而大幅缩短运行时间。这也与定理 C.3 中证明的转换算法（算法5）的查询复杂度结果相符。

关于非单调SCP问题的查询复杂度和运行时间的进一步结果如图 4(i) 和 4(j) 所示。从结果可以看出，BLOCK-G算法的运行速度快于STREAM算法和GUIDED-RG算法，这证明了我们算法的高效性。

关于SCF问题实验中不同算法查询复杂度的比较结果如图 4(c) 和 4(g) 所示。从结果可以看出，BLOCK-G-Fair算法的查询复杂度优于GREEDY-Fair算法，但逊于THRES-Fair。这是因为这三种算法在 Chen 等人提出的算法 1 中，用于将SMF算法转换为SCF的转换算法所采用的子程序算法不同。具体而言，THRES-Fair采用了阈值贪心算法，其时间复杂度为 $\mathcal{O}(\frac{n}{\epsilon}\log\frac{n}{\epsilon})$；而BLOCK-G-Fair和GREEDY-Fair所采用的子程序算法的运行时间均为 $\mathcal{O}(nk_g\beta)$，其中 k_g 是对 $|OPT|$ 的猜测，参数 β 表示近似比。因此，BLOCK-G-Fair的查询复杂度低于GREEDY-Fair，这是因为BLOCK-G-Fair的参数 β 为$\ln\frac{1}{\epsilon}\ln 2$，小于GREEDY-Fair所采用的 $\mathcal{O}(\frac{1}{\epsilon})$。

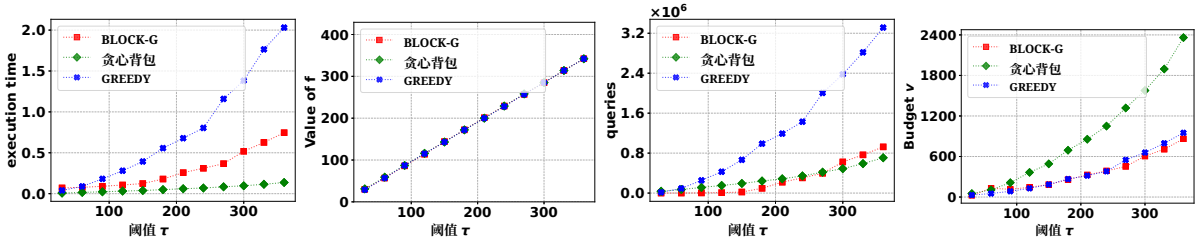
图 4(f) 和 4(b) 展示了 SCF 实验中 τ 不同赋值对应的函数值结果。从图中可以看出，不同算法返回解的函数值几乎相同，并且随阈值 τ 呈线性变化。这与不同算法的理论保证相符，该保证要求所有算法均满足 $f(S) \geq 0.9\tau$。最后，图 4(d) 和 4(h) 还展示了运行不同算法的执行时间结果。

ImageNet 50数据集上的补充实验结果如图 5 所示。从这些结果可以看出，Block-Greedy 始终能显著提升公平性表现，并且在许多情况下，返回解的成本低于或与基线方法相当。这表明该算法在实践中行之有效，尤其适用于公平性敏感型应用。

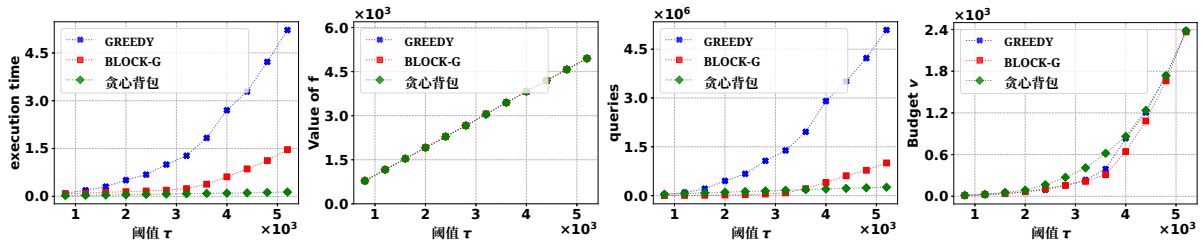
最后，我们还绘制了这些算法在 corel数据集上使用 $\tau = 300$ 时所生成解中不同标签的分布，如图 6(a)、6(b) 和 6(c) 所示。从图中可以看出，GREEDY-Fair 和 THRES-Fair 返回解中超过 30% 的元素标签为 1，这表明输出分布缺乏公平性。而我们的算法 BLOCK-G-Fair 生成的解在不同标签间呈现出显著更公平的分布，展现了所提算法的有效性。

G 更广泛的影响

本文旨在推动机器学习领域的发展。我们的工作可能对社会产生诸多影响，但我们认为其中没有任何一项需要在此特别强调。

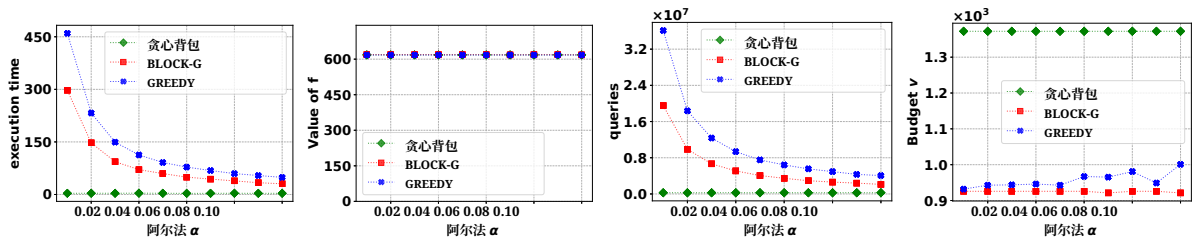


(a) Corel 时间 (SCKP) (b) corel f (SCKP) (c) Corel 查询（SCKP） (d) Corel 预算（SCKP）



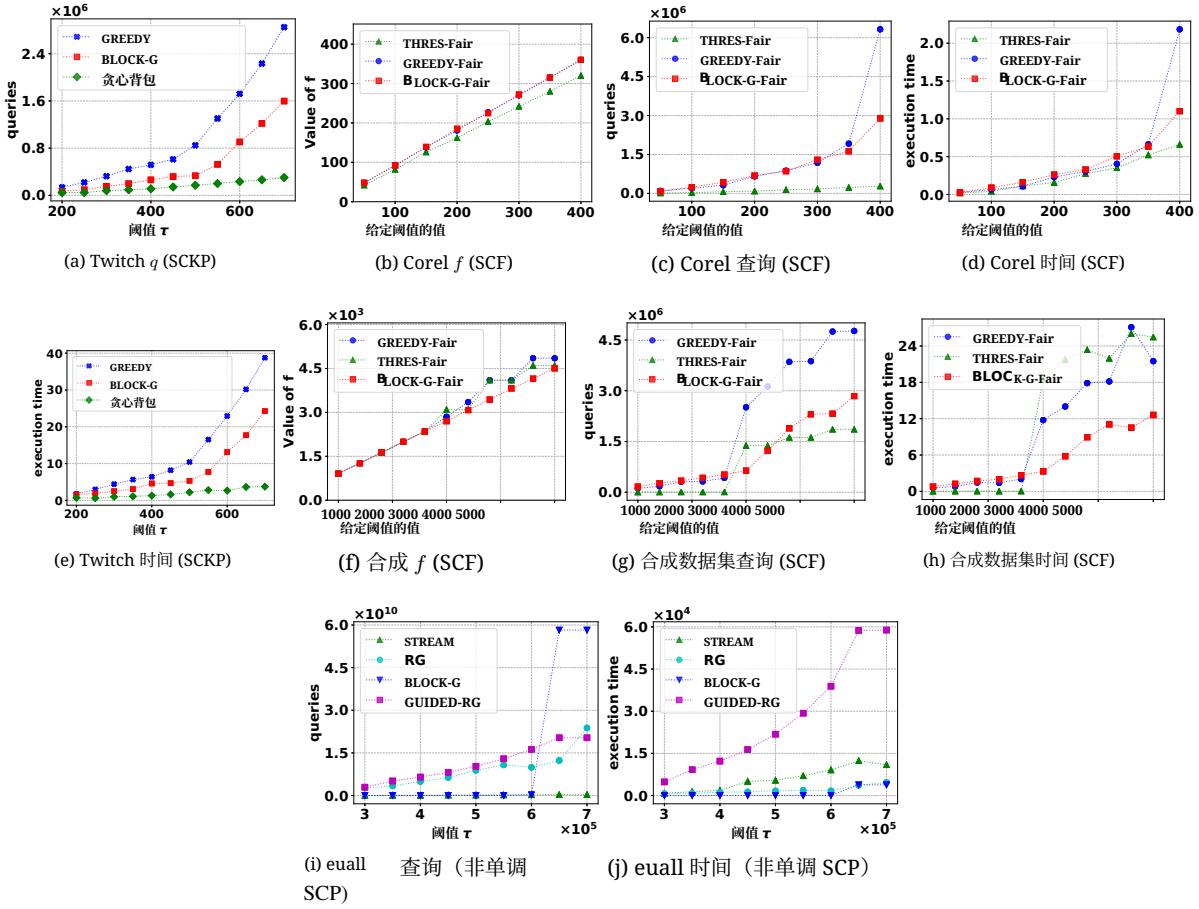
(e) 合成数据集时间（SCKP） (f) 合成数据集 f （SCKP） (g) 合成数据集查询（SCKP） (h) 合成数据集预算（SCKP）

图 2：在 Corel5k 数据集和合成数据集上运行算法的实验结果。样本：查询数量。预算：$\max_{i\in[N]} {}^{c}(S\cap U_i)$ p_i 。



(a) Twitch 时间（SCKP） (b) Twitch f （SCKP） (c) Twitch 查询（SCKP） (d) Twitch 预算（SCKP）

图 3：不同 α 值下 Twitch 数据集上的 SCKP 问题实验结果。样本：查询数量。预算：$\max_{i\in[N]} {}^{c}(S\cap U_i)$ p_i 。

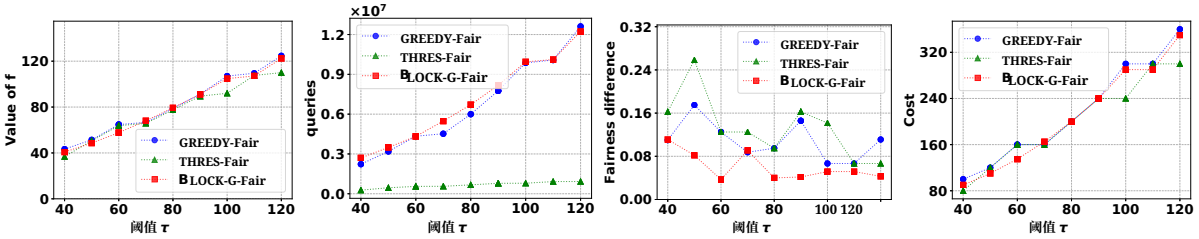


(a) Twitch q (SCKP) (b) Corel f (SCF) (c) Corel 查询 (SCF) (d) Corel 时间 (SCF)

(e) Twitch 时间 (SCKP) (f) 合成 f (SCF) (g) 合成数据集查询 (SCF) (h) 合成数据集时间 (SCF)

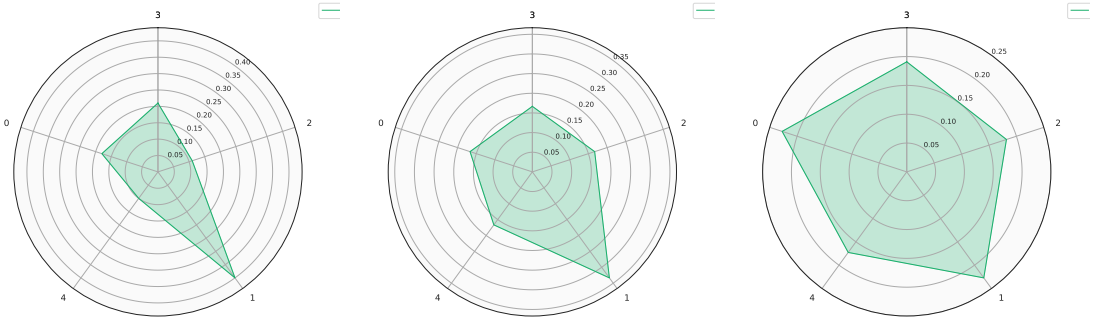
(i) euall 查询（非单调 SCP) (j) euall 时间（非单调 SCP）

图 4：在 Corel5k 数据集和合成数据集上运行算法的实验结果。样本：查询数量。成本：返回解的大小。预算：$\max_{i\in[N]} {}^{c(S\cap U_i)}\ p_i$ 。公平性差异：$(\max_c |S\cap U_c| - \min_c |S\cap U_c|)/|S|$。



(a) ImageNet _50 f (SCF) (b) ImageNet 50查询数量(SCF) (c) ImageNet 50公平性差异-异 (SCF) (d) ImageNet 50成本 (SCF)

图 5：在 SCF 问题上运行算法于 ImageNet 50数据集的实验结果。样本：查询数量。成本：返回解的大小。公平性差异：$(\max_c |S\cap U_c| - \min_c |S\cap U_c|)/|S|$。



(a) THRES-Fair (b) GREEDY-Fair (c) BLOCK-G-Fair

图6：带有$\tau = 300$的 corel数据集上 SCF 实例实验的标签分布雷达图。