收敛至（反）多拟阵中的字典序最优基及其在最密子图和树打包中的应用

ElfaroukHarb ！伊利诺伊大学厄巴纳-香槟分校**Kent Quanrud** ✉普渡大学
ChandraChekuri ！伊利诺伊大学厄巴纳-香槟分校

arXiv:2305.02987v1 [cs.DS] 4 May 2023

摘要

Boob 等人 [1] 提出了一种名为 Greedy++ 的迭代剥离算法，用于最密子图问题（DSG），并猜想该算法会收敛到最优解。Chekuri、Qaunrud 和 Torres [2] 将该算法推广到一般的超模密度问题（DSG 是其中的一个特例），并证明由此得到的 Super-Greedy++ 算法（因此 Greedy++ 也一样）能够收敛。本文重新审视了收敛性证明，并提出了不同的视角。具体而言，我们将其与 Fujishige 用于求解（反）多拟阵中字典序最优基的二次规划 [3], ，以及凸优化中的 Frank-Wolfe 方法的一个带噪版本 [4, 5]联系起来。这不仅带来了更简洁的收敛性证明，还表明 Super-Greedy++ 会收敛到最优稠密分解向量，从而回答了 Harb 等人提出的一个问题 [6]。本文的第二项贡献是通过 Frank-Wolfe 算法来理解 Thorup 关于理想树打包和贪心树打包的工作 [7, 8]；该算法用于求解图拟阵中的字典序最优基。这带来了更简洁明了的证明。尽管这两项结果看似无关，但 Fujishige 的结果与凸优化将它们统一起来。

2012 年美国计算机协会主题分类 图算法

关键词和短语 多拟阵、字典序最优基、最稠密子图、树打包

数字对象标识符 10.4230/LIPIcs...

资助 *Elfarouk Harb*：部分经费由 NSF 资助项目 CCF-2028861 提供*Kent Quanrud*：部分经费由 NSF 资助项目 CCF-2129816 提供*Chandra Chekuri*：部分经费由 NSF 资助项目 CCF-2028861 和 CCF-1910149 提供。

©根据知识共享署名 4.0 国际许可协议（CC-BY 4.0）授权；根据知识共享署名 4.0 国际许可协议（CC-BY 4.0）授权。莱布尼茨国际信息学会议论文集，德国达格施图尔莱布尼茨信息学中心，达格施图尔社

1 引言

本文研究了两个不同组合优化问题的迭代式贪心算法，并表明，结合两种通用工具即可理解这些算法的收敛性：一种来自次模函数理论，另一种来自凸优化。这一统一视角不仅带来了更简洁的证明，也揭示了此前未知的其他性质。

最稠密子图与超模性： 我们从促成本项工作的初始问题——最密子图问题（DSG）——开始。DSG 的输入是一个无向图 $G = (V, E)$，其具有 $m = |E|$ 和 $n = |V|$。目标是返回一个子集 $S \subseteq V$，使 $\frac{|E(S)|}{|S|}$ 最大化，其中 $E(S) = \{uv \in E : u, v \in S\}$ 是两个端点均位于 S 中的边的集合。在全文中，我们令 $\lambda(G) = \frac{|E(G)|}{|V(G)|}$ 表示图 $G(V, E)$ 的密度。为简便起见，我们讨论无权情形；所有结果都可推广到边加权图。Goldberg [9] 以及 Picard 和 Queyranne [10] 表明，DSG 可以通过归约到 s-t 最大流问题来高效求解。

另一种能够证明 DSG 可在多项式时间内求解的联系对本文十分重要。考虑定义在顶点集 V 上的实值集合函数 $f : 2^V \to \mathbb{R}_+$，其中 $f(S) = |E(S)|$。该函数是超模的。函数 f 是超模的，当且仅当 $-f$ 是次模的。实值集合函数 $f : 2^V \to \mathbb{R}$ 是次模的，当且仅当对于所有 $A, B \subseteq B$，均有 $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$。次模和超模集合函数是组合优化中的基本概念——参见 [11, 12]。

回到DSG，最大化 $|E(S)|/|S|$ 等价于寻找最大的 λ，使得对所有 $S \subseteq V$ 都有 $\lambda|S| - |E(S)| \geq 0$。这对应于最小化次模函数 g，其中 $g(S) = \lambda|S| - |E(S)|$。组合优化中的一个经典结果是，在值预言机模型下，可以在多项式时间内找到次模集合函数的最小值。因此，可以通过归约到次模集合函数最小化并结合二分搜索来求解DSG。上述联系还促成了DSG的一种推广——最稠密超模集合问题（DSS）（见 [2]）的定义。输入是一个非负超模函数 $f : 2^V \to \Re_+$，目标是找到使 $\frac{f(S)}{|S|}$ 最大化的 $S \subseteq V$。DSS可通过次模集合函数最小化在多项式时间内求解。DSG、DSS及其变体在实践中有多种应用，并且通常用于图和网络分析，以寻找稠密的簇或社区。关于这一主题的广泛文献，请参阅 [13, 1, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]。DSG还因其与树木度及相关概念的联系而在算法领域受到关注——近期研究见 [28, 29] 。

更快的算法，贪心算法和贪心算法++： 尽管可以通过最大流和次模函数最小化在多项式时间内求解 DSG，但对于许多应用中出现的大型图，相应的算法尚不实用；尽管如今我们已经有了非常快速的最大流和最小费用流理论算法 [30]。因此，人们对更快的（近似）算法产生了浓厚兴趣。20 多年前，Charikar [31]证明，一种简单的“剥离”算法（贪心算法）可为 DSG 提供 $1/2$ 近似。按如下方式计算顶点的一个排序：$v_{i_1}, v_{i_2}, \ldots, v_{i_n}$：$v_{i_1}$ 是 G 中度最小的顶点（若有并列则任意打破），v_{i_2} 是 $G - v_{i_1}$ 中度最小的顶点，依此类推[1]。生成排序后，算法选择 the

[1] 这种剥离顺序与用于构建图的所谓核分解的顺序相同 [32]，而 Greedy 算法本身则由 Asahiro 等人提出 [33]。

在排序的 n-可行后缀中，密度最高的后缀。Charikar 还为 DSG 提出了一种简单而精确的线性规划松弛。Charikar 的研究成果影响深远。Greedy 可在近线性时间内实现，也已被推广到其他变体。线性规划松弛还被用于多种算法，这些算法能给出 $(1-\epsilon)$-近似解 [34, 35],，并由此得到一种基于流的 $(1-\epsilon)$-近似算法 [2]。最近，Boob 等人 [1] 提出了一种名为 Greedy++ 的算法，它结合了 Greedy 与乘法权重更新（MWU）的思想；该算法反复应用一种简单的剥离算法，第一次迭代与 Greedy 相同，但后续迭代取决于在顶点上维护的权重向量——该算法的正式描述见后文。该算法的优点是简单，Boob 等人 [1] 还表明它在实证中表现出色。此外，他们猜想 Greedy++ 经过 $O(1/\epsilon^2)$ 次迭代后会收敛到 $(1-\epsilon)$-近似解。尽管这一强猜想尚未得到验证，Chekuri 等人 [2] 证明了 Greedy++会在 $O(^{\Delta}{}_{1}{}_{\mathrm{og}\,\epsilon^2\lambda}{}^{|V|}_{G}{}_{()})$ 次迭代内收敛，其中 Δ 是 G的最大度。

---

[2] 中的收敛性证明并非易事。该证明的关键在于考察 DSS 和超模性。 [2] 表明，Greedy 和 Greedy++ 可以推广为适用于 DSS 的 SuperGreedy 和 SuperGreedy++ ，并且 SuperGreedy++ 经过 $O(\alpha_f/\epsilon^2)$ 次迭代后，可收敛到一个 $(1-\epsilon)$-近似解，其中 α_f 仅取决于函数 f。

稠密子图分解及相关联系：正如我们所讨论的，DSG 是 DSS 的一个特例，因此 DSG 继承了超模性的一些良好结构性质。其中之一是这样的事实：每个图 $G=(V,E)$ 的顶点集 V 都可以分解为 S_1、S_2、……、S_k，其中 k 满足： S_1 是唯一的极大最稠密子图的顶点集，S_2 是将 S_1“收缩”后得到的唯一极大最稠密子图，依此类推。在 DSS 的情形下，这一事实更容易理解。在这里， S_1 是唯一的极大最稠密集合，这可由超模性推出；如果 A 和 B 是最优稠密集合，那么 $A\cup B$ 也是。接下来，可以考虑一个定义在 $V-S_1$ 上的新超模函数 f_{S_1}： $2^{V-S_1}\to$R，其中对所有 $A\subseteq V-S_1$，均有 $f_{S_1}(A)=f(S_1\cup A)-f(S_1)$。新函数同样具有超模性。于是， S_2 是 f_{S_1}的唯一极大最稠密集合。我们重复这一过程，直到得到空集。该分解还允许我们为每个 $v\in V$ 指定一个密度值 λ_v（这对应于 v 位于极大集合中时该集合的密度）。我们称之为与 f 相关联的密度向量。稠密分解源自主划分理论中的次模函数理论 [36, 37, 38]。在图和 DSG 的背景下，Tatti 和 Gionis 重新发现了这一结果，并将其称为局部稠密分解[39, 40], ，还给出了计算该分解的算法。随后，Danisch 等人 [14] 将著名的 Frank-Wolfe 算法用于约束凸优化，应用于一个由 Charikar 针对 DSG 提出的线性规划松弛推导出的二次规划。最近，Harb 等人 [6] 利用 Charikar 的线性规划，为计算图中的稠密分解提出了更快的算法；他们采用了一种称为 FISTA 的不同方法，通过加速来实现约束凸优化。虽然 DSS 并非主要研究重点， [6] 也建立了一个重要联系：将 Fujishige 关于多拟阵中字典序最优基的结果 [3] 与之联系起来，从而阐明了 Danisch 等人关于 DSG 的工作。下面我们将对此进行介绍。

字典序最优基与稠密分解： 我们简要介绍 Fujishige 的结果 [3] 及其与稠密分解的联系。令 f： $2^V\to$R+ 为一个单调次模集合函数（如果 $A\subset B$，则 $f(A)\le f(B)$），并且它是规范化的（$f(\emptyset)=0$）。按照 Edmonds 的定义，与 f相关联的多拟阵记为 P_f，它是多面体$\{x\in\mathrm{R}^V\mid x\ge 0, x(S)\le f(S)\quad \forall S\subseteq V\}$,其中 $x(S)=\sum_{i\in S}x_{io}$ 与 f相关联的基多面体记为 $B_f=P_f\cap\{x\in\mathrm{R}^V\mid x(V)=f(V)\}$，通过与等式约束相交得到

P_f。每个满足等式约束 $x(V) = f(V)$ 的向量 x 都称为一个基。若 f 是一个单调规范化超模函数，我们考虑逆多拟阵$P_f = \{x \in \mathrm{R}^V \mid x \geq 0, x(S) \geq f(S) \quad \forall S \subseteq V\}$（不等式方向相反）；类似地，$B_f$ 是通过将 P_f 与等式约束 $x(V) = f(V)$ 相交得到的基逆多拟阵。Fujishige 证明，任意多拟阵中都存在唯一的字典序最小基，而且可以通过求解以下二次规划得到：min $\sum_v x_v^2$ s.t $x \in B_f$。对于超模函数，也可以得到类似的结果；二次规划 min $\sum_v x_v^2$ s.t $x \in B_f$（其中 B_f 是与 f 相关联的逆多拟阵）具有唯一解。正如 [6], 中明确指出的，对于 DSS，字典序最优基给出了稠密分解向量。也就是说，如果 x^* 是该二次规划的最优解，那么对于每个 v， $x_v^* = \lambda_v$。特别地，如 [6], 中所述，可以将著名的 Frank-Wolfe 算法应用于该二次规划，并且它会收敛到稠密分解向量。正如我们稍后将看到的，每次迭代都对应于在逆多拟阵中寻找一个最大权基，而通过 Greedy 算法可以轻松找到该基。

（理想）图中的树打包与 Tutte–Nash-Williams 定理：到目前为止，我们的讨论聚焦于 DSG。现在我们来介绍图上的另一个问题及相关背景。正如我们所说，我们的目标是对这两个问题提出统一的视角。图论中著名的 Tutte–Nash-Williams 定理（见 [11]）针对多重图 G 中最多可有多少棵边不相交的生成树，给出了一个极小极大结果。给定无向图 $G = (V, E)$, 以及一个顶点划分 P，令 $E(P)$ 表示跨越不同划分部分的边数。我们称一个划分的强度为 $\frac{E(P)}{|P|-1}$。令 T (G) 表示 G 的所有可能生成树。令 $\tau^*(G)$ 表示 G 中边不相交的生成树的最大数量。则 $\tau^*(G) = \min_P \frac{E(P)}{|P|-1}$。此外，如果我们将 $\tau(G)$ 定义为生成树的最大分数打包数，则可以去掉取整符号，并得到 $\tau(G) = \min_P \frac{E(P)}{|P|-1}$。我们注意到，图论中的这一结果是拟阵基打包的一个特例。树打包在许多应用中都很有用。尤其是，Karger [41] 在其著名的、近线性时间的随机化图全局最小割算法中使用了树打包及其他方法。我们在此主要关注 Thorup 在 [7, 8] 中的工作，该工作由动态最小割和 k-割问题所推动。他通过递归地利用 Tutte–Nash-Williams划分分解图，定义了所谓的理想边负载和理想树打包（详情见后文）[7]。他还证明，一个简单的迭代贪心树打包算法会收敛到理想负载 [8]。他利用近似理想树打包，为 k-割问题得到了新的确定性算法，其方法对之后的一系列结果产生了深远影响 [42, 43, 44, 45, 46]。Thorup 从基本原理出发得到了他的树打包结果。我们要问的是：理想树打包与 DSG 之间是否存在联系?

1.1 论文贡献

本文有两项主要贡献。第一，我们为 DSS 中 SuperGreedy++ 的收敛性给出了一个新的证明。我们的证明基于这样一点：SuperGreedy++ 可以被视为将 Frank-Wolfe 算法应用于 Fujishige 所定义的二次规划时的一种 “ 带噪声 ” 或 “ 近似 ” 变体。这个新证明有两方面优势。首先，它表明 SuperGreedy++ 不仅会收敛到一个 $(1-\epsilon)$-近似的最稠密集合，事实上还会收敛到最稠密分解向量。这一现象已在 [6] 中针对 DSG 经实证观察到，但仍是一个有待解决的开放问题。 [2] 中关于 SuperGreedy++ 收敛性的证明基于通过线性规划实现的乘法权重更新方法，并未利用 Fujishige 的结果，而这一结果正是我们在此证明更强性质的关键。

其次，该证明在直接且高层次的意义上连接了两种强大的工具：Fujishige 关于次模函数的结果，以及一种用于约束凸优化的标准方法。

▶**定理 1. </style}设 是非负单调超模集合函数：** $2^V \to \mathbb{R}+$ 的稠密分解向量，其中 $|V| = n$。那么，*SuperGreedy++* 经过 $O(\alpha_f/\epsilon^2)$ </style} 次迭代后收敛至向量，使得，其中 仅取决于 。对于具有 条边和 个顶点的图，Greedy 对于无权多重图经过 (mn^2/ϵ^2) *</style}* 次迭代后收敛。

▶ 注2. 与 [2] 中关于收敛至最大密度的 $(1-\epsilon)$ 相对近似的界相比，新的收敛结果给出的界较弱。然而，它对整个稠密分解向量给出了强有力的加性保证。

我们的第二项贡献基于我们对 DSG 和 DSS 的见解，并将其用于理解理想树打包和贪心树打包。我们将 Thorup 的理想树打包与底层图拟阵（其秩函数是次模的）所对应的稠密分解联系起来。随后我们证明，贪心树打包算法可以视为将 Frank-Wolfe 算法应用于 Fujishige 定义的二次规划，并由此轻松得到收敛保证。

▶ **定理3**。 设 $G = (V, E)$ 是一个图。其理想边负载向量 $\ell^*: E \to \mathbb{R}+$ 由与 G的图拟阵秩函数相关联的多拟阵中的字典序最小基给出。将步长设为 $\frac{1}{k+1}$的 *Frank-Wolfe* 算法应用于计算 G的图拟阵中字典序最小基的二次规划时，该算法与贪心树打包算法一致。对于具有 m 条边的无权图，*Frank-Wolfe* 方法收敛性的通用分析表明，贪心树打包会在 $O(\frac{m}{\log(m/\epsilon)\epsilon^2})$次迭代内收敛到满足 $||\ell - \ell^*|| \leq \epsilon$ 的负载向量 $\ell: E \to \mathbb{R}+$。标准步长算法会在 $O(\frac{m}{\epsilon^2})$ 次迭代内收敛。

▶ 注4。尽管算法相同（贪心树打包），Thorup 的分析即使在带容量情形下也能保证强多项式界 [8]。然而，通过 Frank-Wolfe 的通用分析，我们得到了更强的加性保证；我们的分析依赖于 $1/\epsilon^2$，而 Thorup 的分析则依赖于 $1/\epsilon^3$。我们在第5节中给出更详细的比较。

组织结构： 本文其余部分致力于证明这两个定理。本文依赖次模函数理论中的工具以及对 Frank-Wolfe方法分析的改编。我们首先介绍相关背景，然后在单独的章节中证明这两个结果。由于篇幅限制，大部分证明都放在附录中。未来版本将更详细地讨论其他相关工作。

2 Frank-Wolfe算法背景及其一种变体

设$\mathcal{D} \subseteq \Re^d$ 是一个紧凸集，且 $f: \mathcal{D} \to \Re$ 是一个凸可微函数。考虑问题 $\min_{x\in\mathcal{D}} f(x)$。Frank-Wolfe方法 [4] 是一种一阶方法，它依赖于访问线性最小化预言机 LMO，该预言机对于 f 能够回答 LMO$(w) = \arg\min_{s\in\mathcal{D}}\langle s, \nabla f(w)\rangle$，其中给定任意 $w \in \mathcal{D}$。在若干应用中，这类预言机具有快速运行次存在。给定 f，$\mathcal{D}$ 如上所述，Frank-Wolfe 算法是一种迭代算法，会收敛到 f的最小化器$\mathbf{x}^* \in \mathcal{D}$。参见算法1。算法从对最小化器 $b^{(0)} \in\mathcal{D}$的一个猜测开始。每次迭代中，它通过对当前猜测 $b^{(k)}$调用线性最小化预言机，找到一个移动方向 $d^{(k+1)}$。然后，它会稍微移动

朝该方向移动，并通过凸组合确保新点位于$\mathcal{D}$中。随着 k 增大，算法沿新方向移动的幅度会减小，这表示它对当前猜测是最小化器这一点越来越"有信心"。

算法1 Frank-Wolfe-Original

初始化 $b^{(0)} \in \mathcal{D}$: **对** $k \leftarrow 0$ 到 $T-1$ **执行**: $\gamma \leftarrow \frac{2}{k+2}$ $d^{(k+1)} \leftarrow \arg\min(\langle s, \nabla f(b^{(k)})\rangle)_{s\in\mathcal{D}}$ ▷ 调用 $b^{(k)}$ 上的预言机

$b^{(k+1)} \leftarrow (1-\gamma)b^{(k)} + \gamma d^{(k+1)}$ **返回** $b^{(T)}$

Frank-Wolfe 算法最初的收敛分析来自 [4]。Jaggi [5] 给出了一种优雅且更简洁的分析方法。他的分析通过函数 f 的曲率常数 C_f 来刻画收敛速率。

▶**定义 5.** 设 $D \subseteq \Re^d$ 是一个紧凸集，且 $f : D \to \Re$ 是一个凸可微函数。f 的曲率常数 C_f 定义为

$$C_f = \sup_{x,s\in D,\gamma\in[0,1],y=x+\gamma(s-x)} \frac{2}{\gamma^2}(f(y) - f(x) - \langle y-x, \nabla f(x)\rangle).$$

▶ **定义 6.** 设 $g : D \to \Re$ 是一个可微函数。若对于所有 $x, y \in D$，均有 $\|g(\mathbf{x}) - g(\mathbf{y})\|_2 \le L\|x-y\|_2$，则称 g 是 *Lipschitz* 连续的，其常数为 L。

令 $\text{diam}(\mathcal{D}) = \max_{x,y\in\mathcal{D}}\|x-y\|_2$ 表示 $\mathcal{D}$ 的直径。可以证明 $C_f \le L\cdot\text{diam}(\mathcal{D})^2$，其中 L 是 ∇f的 Lipschitz 常数。

▶ **定理 7** ([5]). 设 $D \subseteq \Re^d$ 是一个紧凸集，且 $f : D \to \Re$ 是一个凸可微函数，其最小化器为 $\mathbf{b}^*$。令 $\mathbf{b}^{(k)}$ 表示 *Frank-Wolfe* 算法第 k次迭代时的估计值。那么 $f(\mathbf{b}^{(k)}) - f(\mathbf{b}^*) \le \frac{2C_f}{k+2}$。

Jaggi 的证明技巧可用于证明 Frank-Wolfe 算法的"含噪/近似"变体的收敛速率。这引出了以下定义。近似线性最小化预言机是一种预言机，对于任意 w ∈ D，返回 ŝ，使得 hŝ(w)i ⩽ hs(w)i + + ϵ，其中 s = LMO(w)。在某些应用中，存在高效的精确线性最小化预言机；而在另一些应用中，只能对其进行 ϵ-近似（使用数值方法或其他方法）。Jaggi 的证明技巧可进一步推广，表明只要近似精度随迭代推进而提高，近似线性最小化预言机就足以保证收敛。假设预言机在第 k 次迭代时提供一个 $\frac{\delta C_f}{k+2}$-近似解，其中 $\delta > 0$ 是某个固定常数。收敛速率只会因此恶化一个 $(1+\delta)$ 倍的因子。从定性角度看，这意味着在早期迭代中，我们可以接受 Frank-Wolfe 方向计算得不够精确，但随着 $k \to \infty$，近似结果应当趋近于 LMO($b^{(k)}$)。

另一个值得关注的问题是 Frank-Wolfe 算法对于学习率 $\gamma_k = \frac{2}{k+2}$ 变化的适应能力。事实上，我们将考察的变体都需要 $\gamma_k = \frac{1}{k+1}$。正如我们将看到的，Jaggi 的证明同样可以调整以处理这种情况，收敛速率仅会出现 $O(\log k)$ 的乘法级恶化。我们陈述以下定理，其证明推迟至附录。

▶ **定理 8。** *[证明见附录 7。1]* 设 $D \subseteq \Re^d$ 是一个紧凸集，且 $f: D \to \Re$是一个以最小化器 $\mathbf{b}^*$为目标点的凸可微函数。假设在第 k 次迭代中，我们不再通过调用线性最小化预言机($\mathbf{b}^{(k)}$)来计算 $\mathbf{d}^{(k+1)}$，而是调用一个 $\frac{\delta C_f}{k+2}$-近似线性最小化预言机，其中 $\delta > 0$ 为某个固定值。此外，假设我们不使用 $\gamma_k = \frac{2}{k+2}$，而改用 $\gamma_k = \frac{1}{k+1}$ 作为步长。那么$f(\mathbf{b}^{(k)}) - f(\mathbf{b}^*) \leq \frac{2C_f}{k+1}(1+\delta H_{k+1})$，其中 H_n 是第 n 个调和项。 ——————

我们将定理 8 所述的 Frank-Wolfe 算法变体称为带噪声的Frank-Wolfe。

3 次模与超模函数，以及稠密分解

我们已经定义了次模和超模集合函数、多拟阵和逆多拟阵。我们只考虑满足 $f(\emptyset) = 0$ 的函数；结合超模性和非负性，这一条件意味着单调性，即对于 $A \subseteq B$， $f(A) \leq f(B)$。次模性的另一种定义是通过边际值递减给出的。令 $f(v \mid A) = f(A \cup \{v\}) - f(A)$ 表示 v 相对于 A 的边际值。当 $A \subseteq B$ 且 $v \in V \setminus B$ 时，次模性等价于 $f(v \mid A) \geq f(v \mid B)$；对于超模集合函数，不等号方向相反。我们需要以下简单引理。

▶ **引理 9。** *[证明见附录 7。2]* 对于次模函数 $f: 2^V \to \Re$，函数$g(X) = f(V) - f(V \setminus X)$是超模函数。特别地，如果 f 是标准化单调次模函数，那么 g 是标准化单调超模函数。

删除与收缩，以及非负求和： 次模函数和超模函数在若干简单运算下封闭。给定 $f: 2^V \to \mathrm{R}$，将其限制到子集 V' 相当于删除 $V \setminus V'$。给定 $A \subset V$，将 f 收缩到 A 可得到函数 $g: 2^{V \setminus A} \to \mathrm{R}$，其中 $g(X) = g(X \cup A) - g(A)$。给定两个函数 f 和 g，我们可以取它们的非负和 $af + bg$，其中 $a, b \geq 0$。这些运算也保持单调性和标准化。

3.1 次模函数和超模函数的稠密分解

按照引言中的讨论，我们关注超模函数和次模函数的分解。稠密分解可由次模函数的主划分理论推出，而这一理论已得到广泛研究。关于这方面的内容，可参阅 Fujishige 的综述 [38] 以及 Naraynan 的研究 [36, 37]。标准视角是考察函数 f_λ 对标量 λ 的最小化器，其中 $f_\lambda(S) - \lambda|S|$。随着 λ 从 $-\infty$ 变化到 ∞，最小化器只会在有限个断点处发生变化。本文关注的是非负次模函数和超模函数以比值形式表示的密度概念。为此，我们沿用近期研究 [40, 14, 2, 6] 中的记号，并以便于使用的形式陈述引理，同时在附录中给出证明，以求完整。

超模函数稠密分解： 基本观察如下。

▶ **引理 10。** *[证明见附录 7。3]* 令 $f: 2^V \to \Re_+$ 为一个非负超模集合函数。存在唯一的极大 集合 $S \subseteq V$，使 $\frac{f(S)}{|S|}$ 达到最大值。 ——

前面的引理可以很简单地用于推导以下推论（例如， [2] 中就明确提到了这一点）。

▶ **推论 11.** 设 $f: 2^V \to \Re_+$ 是一个非负超模集合函数。存在唯一的划分 $S_1, S_2, \ldots, S_h$，将 V划分为若干部分，并满足以下性质。设 $V_i = V - \cup_{j<i} S_j$，且令 $A_i = \cup_{j<i} S_i$。那么，对于每个从 $i = 1$ 到 h 的 S_i，它都是函数 $f_{D_i}: 2^{V_i} \to \mathbb{R}+$的唯一极大最稠密集合。此外，令 λ_i 为 f_{D_i}的最优密度，则有$\lambda_1 > \lambda_2 \ldots > \lambda_h$。

根据前面的推论，我们可以为每个 $v \in V$ 关联一个值 $\lambda(v)$:$\lambda(v) = \lambda_i$，其中 $v \in S_i$。函数 $f(S) = |E(S)|$ 的稠密分解示例见图 1。

次模函数的稠密分解：现在我们讨论次模函数。我们考虑两种变体。首先介绍一个基本观察。

▶**引理 12。** *[证明见附录 7。4]* 设 $f: 2^V \to \Re_+$ 是一个单调非负次模集合函数，满足 $f(v) > 0$ 对所有 $v \in V$均成立。存在唯一的极小集合 $S \subseteq V$，使得次模函数 f的 $f^{|V|-|S|}_{V-f}()(S)$ 达到最小值。

考虑以下一种针对 f 的分解变体。我们令 $S_0 = V$，并找到使 $f^{|V|-|S|}_{V-f}()(S)$ 最小的唯一 极小 集合 $S \subseteq V$。然后我们“删除” $\hat{} S_1 = V \setminus S_1$，并找到使 $f^{|S_1|-|S|}_{S_1-f}()()$ 最小的极小集合 $S_2 \subseteq S_1$。在第 i 次迭代中，我们找到使$f^{|S_{i-1}|-|S_i|}_{S_{i-1}-f}()()$ 最小的唯一极小集合 $S_i \subset S_{i-1}$。注意到 $S_k \subset S_{k-1} \subset \ldots \subset S_1 \subset V$。我们称 $\hat{} S_i = S_{i-1} \setminus S_i$ 的相对密度为 $\lambda_i = f^{|S_{i-1}|-|S_i|}_{S_{i-1}-f}()()$。对于 $u \in \hat{} S_i$，我们称 u 的 $\hat{}$ 密度为 $\lambda_u = \lambda_i$。因此，f 的稠密分解为 $S_1, \ldots, S_k$，其密度为$\lambda_1, \ldots, \lambda_k$。我们将这种基于删除的分解称为第一种变体。

现在我们描述次模函数的第二种稠密分解。设 $f: 2^V \to \mathrm{R}+$ 是一个单调次模函数。考虑超模函数 $g: 2^V \to \mathrm{R}+$，其中对所有 $X \subseteq V$，$g(X) = f(V) - f(V \setminus X)$。由引理 9 可知，$g$ 是单调超模函数。因此，我们可以应用推论 11，得到 g 的一个稠密分解。令 $T_1, T_2, \ldots, T_{k'}$ 为考虑 g 所得到的唯一分解，并令 $\lambda_1, \ldots, \lambda_{k'}$ 为相应的密度。注意，这第二种分解基于收缩。

不出所料，这两种分解是一致的，我们将在下一个定理中证明这一点。之所以将它们分别考虑，主要是为了在应用中更便于技术处理，因为在某些情况下，其中一种视角会比另一种更自然。

▶ **定理 13。** *[证明见附录 7。5]* 设 $\hat{} S_1, \ldots, \hat{} S_k$ 是次模函数 f 的一个稠密分解（使用删除变体），其密度为 $\lambda_1, \ldots, \lambda_k$。设 $T_1, \ldots, T_{k'}$ 是同一函数的一个稠密分解（使用收缩变体），其密度为 $\hat{} \lambda_1, \ldots, \hat{} \lambda_{k'}$。我们有：*(i)* $k' = k$；*(ii)* $\hat{} S_1, \ldots \hat{} S_k$ 恰好是 $T_1, \ldots, T_k$；以及 *(iii)* 对于 $1 \le i \le k$，$\hat{} \lambda_i = \frac{1}{\lambda_i}$。

3.2 Fujishige 关于字典序最优基的结果

Fujishige [3] 提出了稠密分解的多面体视角，这是我们工作的核心要素。不过，他给出的定理适用于多拟阵，但很容易推广到逆多拟阵。为简化记号，我们仅讨论无权情形—— [3] 讨论加权情形。

对 R^n 中的向量，可以按坐标递增排序，并比较所得的长度为 n的两个有序序列的字典序排列，从而定义全序。下文中，对于 a，$b \in \mathrm{R}^n$，我们用 $a \prec b$ 和 $a \preceq b$ 表示这种顺序。如果向量 x属于集合 D，且对所有 $y \in D$ 都有 $x \preceq y$ $(y \preceq x)$，则称其为字典序最小（最大）。

Fujishige 为多拟阵证明了以下定理。

▶ **定理14** ([3]). 设 $f: 2^V \to \mathbb{R}+$ 是一个单调次模函数（多拟阵），且 B_f 是其基多面体。那么，存在唯一的字典序最大基$b^* \in B_f$，并且对于每个 $v \in V$， $b^*_v = \lambda_v$。此外， b^* 是以下二次规划的最优解：最小化 $\sum_v x_v^2$，满足约束条件 $x \in B_f$。

前述定理可以直接推广到逆多拟阵中，且 [6] 明确指出了这一点。我们将其改述为与前述定理的表述类似。

▶**定理15**。 设 $f: 2^V \to \mathbb{R}+$ 为单调超模函数（逆多拟阵），且 B_f 为其基多面体。那么，存在唯一的字典序最小基 $b^* \in B_f$，并且对每个 $v \in V$， $b^*_v = \lambda_v$。此外， b^* 是以下二次规划的最优解：最小化 $\sum_v x_v^2$ 满足 $x \in B_f$。

3.3 使用 Frank-Wolfe 近似字典序最优基

考虑凸二次规划 min $\sum_{v\in V} x_v^2$ subject to $x \in B_f$，其中 B_f 是 f 的基多面体（可以是次模的，也可以是超模的）。我们可以使用 Frank-Wolfe方法近似求解这个优化问题。二次函数的梯度为 $2x$，因此在每次迭代中，我们需要求解线性最小化预言机 LMO(w) $=$arg min$_{\mathbf{s}\in B_f}\langle \mathbf{s}, 2\mathbf{w}\rangle$ for $\mathbf{w} \in B_f$。这等价于 arg min$_{\mathbf{s}\in B_f}\langle \mathbf{s}, \mathbf{w}\rangle$，也就是说，在基多面体上优化一个线性目标。Edmonds [47] 证明，简单的贪心算法是一个 $O(|V|\log |V|)$ 时间的精确算法（假设可以在 $O(1)$ 时间内访问 f的预言机）。

▶ **定理 16.** *[47]* 固定一个多拟阵 $f: 2^V \to \Re_+$。给定 $\mathbf{w} \in B_f$，将 $V = \{v_1, ..., v_n\}$ 按 $\mathbf{w}_i$ 的降序排列为 $\{s_1, ..., s_n\}$。令 $A_i = \{s_1, ..., s_i\}$，其中 $1 \le i \le n$ 满足 $A_0 = \emptyset$。定义 $\mathbf{s}^*_i = f(A_i) - f(A_{i-1})$。则 $\mathbf{s}^* =$arg min$_{\mathbf{s}\in B_f}\langle \mathbf{s}, \mathbf{w}\rangle$。

该定理对超模函数也成立，但需要将顺序从按$\mathbf{w}$ 降序改为升序，并对集合 A_i 取补集。

▶ **定理 17.** *[47]* 固定一个逆多拟阵 $f: 2^V \to \Re_+$。给定 $\mathbf{w} \in B_f$，将 $V = \{v_1, ..., v_n\}$ 按 $\mathbf{w}_i$ 的升序排列为 $\{s_1, ..., s_n\}$。令 $A_i = \{s_i, ..., s_n\}$，其中 $1 \le i \le n$满足 $A_{n+1} = \emptyset$。定义 $\mathbf{s}^*_i = f(A_i) - f(A_{i+1})$。则 $\mathbf{s}^* =$arg min$_{\mathbf{s}\in B_f}\langle \mathbf{s}, \mathbf{w}\rangle$。

两种算法的主导步骤都是排序步骤，因此需要花费 $O(|V|\log |V|)$ 时间。这些简单算法意味着，可以将 Frank-Wolfe 算法用于二次规划，以获得次模（分别为超模）函数的字典序最大（分别为最小）基的近似解。标准 Frank–Wolfe 算法需要 $O(\frac{\text{直径}(B_f)^2}{\epsilon^2})$ 次迭代才能收敛到满足 $\left\|\hat{b} - b^*\right\| \le \epsilon$ 的向量 $\hat{b}$。

4 应用 1：Greedy++ 和 Super-Greedy++ 的收敛

我们首先介绍来自 [1] 的 Greedy++，以及它的推广算法 SuperGreedy++ [2]。Greedy++ 基于对 Greedy 剥离思想的修改，并将其应用于多次迭代。算法初始化每个 $v \in V$ 上的权重/负载，记为 $w(v)$，并将其设为 0。每次迭代中，它通过剥离顶点来生成一个排序：下一个被选中的顶点是 arg min$_v(w(v) + \deg_{G'}(v))$，其中 G' 是当前图（移除先前剥离的顶点后得到的图）。在迭代结束时， $w(v)$ 增加 v 在当前迭代中被剥离时的度。具体描述如下。SuperGreedy

是 Greedy 在超模函数上的自然推广，而 SuperGreedy++ 是 Greedy++ 的推广。算法的形式化描述如下。

算法 2 GREEDY++$(G(V,E)\ T)$ [1]

为所有 $u \in V$ 初始化 $w(u) \leftarrow 0$
$G^* \leftarrow G$
对于 $k \leftarrow 0$ 至 $T-1$ **执行 do**
  $G' \leftarrow G$
  当 $|G'|>1$ **时执行**♀♀♀
    $u \leftarrow \arg\min_{u \in G'} (w(u)+deg_{G'}(u))$
    $w(u) \leftarrow w(u)+deg_{G'}(u)$
    $G' \leftarrow G'-\{u\}$
    若 $\lambda(G')>\lambda(G^*)$ **则**
      $G^* \leftarrow G'$
返回 G^*

算法 3 Super-Greedy++(f,T) [2]

对所有 $u \in V$ 初始化 $w(u) \leftarrow 0$
$S^* \leftarrow V$
当 $k \leftarrow 0$ 到 $T-1$ **时**
  $V' \leftarrow V$
  当 $|V'|>1$ **时**
    $u \leftarrow \arg\min_{u \in V'} \{w(u)+f(V')-f(V'-u)\}$
    $w(u) \leftarrow w(u)+f(V')-f(V'-u)$
    $V' \leftarrow V'-u$
    if $\frac{f(V')}{|V'|} > \frac{f(S^*)}{|S^*|}$ **then**
      $S^* \leftarrow V'$
返回 S^*

本节的目标是证明定理 1，即 SuperGreedy++ 和 Greedy++ 收敛到字典序最大基。

4.1 直觉与主要技术引理

如第 3.3 节所述，若将 Frank-Wolfe 算法应用于求解二次规划 min $\sum_{v \in V} x_v^2$，满足 $x \in B_f$，则每次迭代都对应于寻找 f 的一个最小权基，其中权重由当前向量 x 给出。寻找最小权基等价于根据 x 对 V 排序。然而，SuperGreedy++ 和 Greedy++ 在每次迭代中使用更复杂的剥离算法；剥离过程既依据权重，也依据顶点的度，而且排序并非静态的（剥离过程中度会发生变化）。这使得对这些算法进行形式化分析并非易事。在 [2], 中，作者利用与乘法权重更新方法的联系，通过线性规划松弛进行分析。这里我们依托二次规划和含噪 Frank-Wolfe。源自 [2], 的高层直觉如下：随着算法逐次迭代，顶点上的权重不断累积；回想一下，在 DSG 的情形中，权重的总增量为 $m=|E|$。影响剥离过程的度项在早期迭代中占主导地位，但随着顶点权重增大，它对顶点排序的影响最终会逐渐减弱。因此可以合理地推测，该算法在极限情况下的行为类似于标准 Frank-Wolfe 方法。主要问题是如何将这一直觉精确化。 [2] 依赖于与乘法权重更新方法的联系，而我们则利用与含噪 Frank-Wolfe 的联系。

为此，考虑 Greedy++ 和 SuperGreedy++ 的一次迭代。算法根据当前权重向量和各顶点的度进行剥离。我们将这一剥离算法单独提取并抽象出来，分别称为加权贪心和 Weighted-SuperGreedy，并将权重向量 w 作为参数对它们进行形式化描述。

算法 4 加权贪心(G, w)

输入：$G(V, E)$ 和 $w(u)$ 对于 $u \in V$

$G' \leftarrow G$

初始化 $\hat{d}(u) = 0$ 对所有 $u \in V$。

当 $|G'| > 1$ **时，做**。

$u \leftarrow \arg\min_{u \in G'}(w(u) + deg_{G'}(u))$

$\hat{d}(u) \leftarrow deg_{G'}(u)$

$G' \leftarrow G' - \{u\}$

返回 $\hat{d}$

算法 5 Weighted-SuperGreedy(f, w)

输入：超模 $f: 2^V \to \Re_+$, $w(u)$，其中$u \in V$

$V' \leftarrow V$

初始化 $\hat{d}(u) = 0$ 对所有 $u \in V$。

当 $|V'| > 1$ **时，做**。

$u \leftarrow \arg\min_{u \in G'} (w(u) + f(V') - f(V' - u))$

$\hat{d}(u) \leftarrow f(V') - f(V' - u)$

$V' \leftarrow V' - u$

返回 $\hat{d}$

$\hat{}$ 剥离算法也会计算出一个基 $d \in B_f$。对于图和 DSG，$\hat{d}(u)$ 被设为顶点 u 被剥离时的度。也可以将该基视为对 E 中各边的定向。对每条边 $uv \in G$，定义两个权重 x_{uv}, x_{vu}。若 $x_{uv} + x_{vu} = 1$ 且 $x_{uv}, x_{vu} \geq 0$ 对所有 $\{u, v\} \in E(G)$ 都成立，则称 $\mathbf{x}$ 是有效的。对于 $b \in \Re^{|V|}$，若 $b_u = \sum_{v \in \delta(u)} x_{uv}$ 对所有 $u \in V$ 都成立，则称 x 诱导 b。若存在一个诱导出该向量的有效 x，则称向量 d 是一种定向。

▶ **引理 18** ([6])。对于 $f(S) = |E(S)|$, $b \in B_f$ 当且仅当 b 是一种定向。

回顾一下，对于给定的权重向量 $w: V \to \mathrm{R}+$，Frank-Wolfe 算法会计算相对于 w 的最小权基 b，因为 $\langle w, b\rangle = \min_{y \in B_f}\langle w, y\rangle$。值得花些时间指出，这个基（或根据引理 18 得到的定向）很容易计算：对于每条边，我们都进行整数定向（即 $x_{vu} = 1$, $x_{uv} = 0$）；若 $w(u) \geq w(v)$，则从 v 定向到 u，否则反向定向。通过简单的交换论证即可证明其正确性，许多研究中也隐含地采用了这一论证 [14]。这会得到相对于 w 的一个最优基 d^*_w。我们的目标是比较 Weighted-Greedy（以及 Weighted-SuperGreedy）所生成的剥离顺序与最优基之间的关系。下面两个技术引理将这一关键思想形式化。第一个引理针对 DSG，第二个则适用于 DSS。

▶ **引理19**。*[证明见附录7。6]* 令$\hat{d}$ 为加权贪心(G, w)的输出，令 d^*_w 为相对于 w的最优定向。则 $\langle w, \hat{d}\rangle \leq \langle w, d^*_w\rangle + \sum_u deg_G(u)^2$。特别地，加性误差不依赖于权重向量 w。

▶ **引理 20**。*[证明见附录 7。7]* 对于超模函数 $f: 2^V \to \Re_+$，令 $\hat{d}$ 为 *Weighted-SuperGreedy*(f, w)（算法 *5*）的输出，d^*_w 为定理 *17* 中所述相对于 w 的最优向量。则 $\langle w, \hat{d}\rangle \leq \langle w, d^*_w\rangle + n\sum_{u \in V} f(u|V - u)^2$。特别地，加性误差不取决于权重向量 w。

4.2 Greedy++ 的收敛性证明

引理19为何至关重要？首先，注意到对于任意常数 $K > 0$，$\langle w, d\rangle$ 的最小化器 d^*_w 与$\langle Kw, d\rangle$ 的最小化器完全相同（反之亦然）。

▶ **引理 21.** 设$\hat{}$ d_K 为加权贪心(G, Kw)的输出。则 $\langle w, \hat{d}_K\rangle \leq \langle w, d^*_w\rangle + \frac{\sum_u deg_G(u)^2}{K}$。

证明。根据引理19，$\sum_{u \in V} Kw(u)\hat{d}_K(u) \leq$最小值$_{\text{orientation } d}\left(\sum_{u \in V} Kw(u) d(u)\right) + \sum_u deg_G(u)^2$。两边除以 K 即可得到所述结论。◀

现在我们可以将 Greedy++视为一种带噪声的 Frank-Wolfe 算法。算法6展示了如何解释 Greedy++ 。

算法 6 GREEDY++$(G(V,E))$

$G=(V,E)$ 和 $w(u)$ 对于 $u\in V$
初始化 $b^{(0)}\leftarrow$ 加权贪心$(G,\mathbf{0})$ ▷ $b^{(0)}$ 是一个有效的定向
对于 $k\leftarrow 0$ 到 $T-1$ **执行**
$\gamma\leftarrow\frac{1}{k+1}$
$d^{(k+1)}\leftarrow$ 加权贪心$(G,\ (k+1)b^{(k)})$
$b^{(k+1)}\leftarrow(1-\gamma)b^{(k)}+\gamma d^{(k+1)}$
返回 $b^{(T)}$

该算法与算法 2 中描述的算法完全相同。事实上，通过归纳法可以证明，$kb^{(k)}$ 恰好是 Greedy++ 在第 k 轮结束时的权重。注意，$(k+1)b^{(k+1)}=kb^{(k)}+d^{(k+1)}$ 正是算法 2 中描述的负载（通过归纳法得出）。我们注意到，$\gamma\leftarrow 1/(k+1)$在此处至关重要，以确保我们取的是平均值。附录中的引理25意味着，算法 2 中的每次剥离都是$\frac{\delta C_f}{k+2}$近似线性最小化预言机。根据定理 8，这意味着 Greedy++

（如算法 2 中所述）在 $\tilde{O}(\frac{mn^2}{\epsilon^2})$ 次迭代内收敛到 b^*。因为 $\delta=O(\frac{\sum_u d_G(u)^2}{m})$ 和 $C_f=O(\sum_u d_G(u)^2)$。我们在附录中使用概率方法来界定 C_f。

扩展至 SuperGreedy++： 对 SuperGreedy++进行的分析基本类似。我们依赖引理 20，而非引理19。由于技术原因，SuperGreedy++ 的收敛分析略逊于 Greedy++。

5 应用2：通过 Frank-Wolfe 解读贪心树打包

设 $G=(V,E)$ 是一个边容量非负的图。本节旨在从不同角度审视 Thorup 对理想边负载及相关树打包的定义，并推导其贪心树打包算法 [7, 8]的另一种收敛分析。在先前的工作中，Chekuri、Quanrud 和 Xu [43] 基于针对 k-割的线性规划松弛，构造了另一种树打包，并以此取代理想树打包。尽管如此，人们仍未能透彻理解 Thorup 的思路。我们弥补了这一空白。

本节中我们仅讨论无权多重图，并在本节末尾讨论带容量情形。设 $G=(V,E)$ 是一个连通多重图，具有 n 个顶点和 m 条边。考虑由 G 诱导的图拟阵 $\mathcal{M}_G(E,\mathcal{F})$；$E$ 是底集，而 $\mathcal{F}$ 由 G 的所有子森林组成。该拟阵的基恰好是 G 的生成树。考虑 $\mathcal{M}_G$ 的秩函数 $r:2^E\to\mathbb{Z}_+$。r 是次模函数；众所周知，对于边子集 $X\subseteq E$，$r(X)=n-\kappa(X)$，其中 $\kappa(X)$ 是由 X 诱导的连通分支数。

5.1 Thorup 的递归算法作为稠密分解

为与前文记号保持一致，我们用 f 表示次模秩函数r。我们首先介绍 Thorup 定义的理想负载。考虑 G的 Tutte–Nash-Williams 划分 P。回顾一下，P 在所有划分中使比值 $\frac{|E_P|}{|P|-1}$达到最小，而该比值为 $\tau(G)$。对于每条边 $e\in E(P)$，赋予其 $\ell^*(e)=\tau^{-1}(G)$。移除 $E(P)$中的边，得到图 G'，此时该图由若干互不连通的分量组成。递归地

计算 G' 每个连通分量中各条边的理想负载（当 G 没有边时，过程终止）。

我们声称，Thorup 的递归分解与 f 的稠密分解（第一种变体）一致。为此，只需考察稠密分解的第一步。我们找出使 $\frac{|E|-|S|}{f_E - f(S)}$ 取最小值的极小集合 $S_1 \subseteq E$。令 $\hat{S}_1 = E \setminus S_1$，并将 $\hat{S}_1$ 中的边赋予密度 $\frac{f(E)-f(S_1)}{|E|-|S_1|}$。然后，我们“删除” $\hat{S}_1$。注意，$\hat{S}_1 = E\setminus S_1$ 正是由 S_1 张成的 $\kappa(S_1)$ 个连通分量所定义的划分 $P(S_1)$ 的跨越边。此外，回忆一下，$\frac{f(E)-f(S_1)}{|E|-|S_1|} = \frac{(\kappa(S_1)-1)|S_1|}{|E|} = \frac{|P_{S_1}|-1}{|E|} = \tau_G(S_1)$。因此，根据 Tutte–Nash-Williams 定理，赋予 $\hat{S}_1$ 中各边的密度恰为 τ_G。下一步是删除 $\hat{S}_1 = E\setminus S_1$；如上所述，这些边是跨越划分 $P(S_1)$ 的边。

我们通过归纳法证明以下引理。

▶**引理22**。 *[证明见附录7。]* 稠密分解算法为边 f 赋予的权重与 ℓ^*一致。

5.2 Greedy 树打包收敛至理想相对负载

Thorup 考虑了以下贪心树打包算法。为每条边定义一个负载 $\ell(e)$，其初始值为 0。该算法按迭代进行。在第 i 次迭代中，算法根据边权 $w(e) = \ell(e)$，在 G 中计算一棵最小生成树 T_i。每条边 $e \in T_i$ 的负载增加 1。Thorup 证明，当 $k \to \infty$ 时，对于每条边 e，量 $\ell(e)/k$ 收敛到 $\ell^*(e)$。他的证明相当复杂。本节中，我们将利用迄今为止建立的工具，给出一个不同的证明来证明这一事实。

▶**引理 23**。 向量 ℓ^* 是生成树多面体的字典序最大基。

证明。 我们证明了，Thorup 的算法只是在图上针对由 G所诱导的图拟阵的秩函数运行稠密分解。该拟阵的基是 G 的生成树，因此 f 的基多面体就是 G的生成树多面体。稠密分解给出了字典序最大基 f，因此ℓ^* 是生成树多面体的字典序最大基。◀

因此，ℓ^* 是二次规划的唯一解，该二次规划旨在最小化 $\sum_e \ell(e)^2$，约束条件为 $\ell \in \mathrm{SPT}(G)$，其中 $\mathrm{SPT}(G)$ 是生成树基多面体。因此，我们可以对该二次规划应用带噪声的 Frank-Wolfe 算法，得到算法 7。

算法 7 Frank-Wolfe-Greedy-TreePack($G(V, E)$)

输入：$G(V, E)$ 初始化 $l^{(0)}(u) = \mathbb{1}\{e \in T\}$ 对任意生成树 T。**对** $k \leftarrow 0$ 至 $T-1$ **执行** $\gamma \leftarrow \frac{1}{k+1}$ $d^{(k+1)} \leftarrow \min_{s\in \mathrm{SPT}(G)} \langle l^{(k)}, s\rangle$。这是关于 $l^{(k)} l^{(k+1)} \leftarrow (1-\gamma)l^{(k)} + \gamma d^{(k+1)}$ **返回** $b^{(T)}$的最小生成树

主要观察是，这个算法**恰好**与 Thorup 的贪心树打包相同！事实上，请注意，$(k+1)\ell^{(k+1)} \leftarrow k\ell^{(k)} + d^{(k+1)} = k\ell^{(k)} + \mathbb{1}\{e \in \text{最小生成树}(G, \ell^{(k)})\}$，其中最小生成树$(G, w)$ 是 G 关于边的最小生成树。

权重 w。由于含噪 Frank-Wolfe 收敛，因此 $\ell^{(k)}$ 收敛到 $\ell^*(e)$，且贪心树打包收敛。

现在我们给出贪心树打包的收敛保证。对于一个有 m 条边的图的生成树多面体，曲率常数为 $C_f \leq 4m$，因为对于 x，$y \in B_f$，$2(x-y)^T(x-y) = \sum_{e\in E}(x_e - y_e)^2 \leq 4m$。将此界代入定理 8，我们得到，经过 $k = O(^m \log(m/\epsilon)\epsilon^2)$ 次迭代后，有 $\left\|\ell^{(k)} - \ell^*\right\| \leq \epsilon$。

假设我们使用 $\gamma = 2/(k+2)$ 运行标准 Frank–Wolfe 算法。此时，收敛保证改进为 $O(\frac{m}{\epsilon^2})$。注意，每次迭代仍对应于在带权图中求最小生成树（MST）。不过，负载向量不再是迄今为止所选树的简单平均。

与 Thorup 的界及分析比较： Thorup [8] 研究了带容量图中的理想树打包；令 $c(e) \geq 1$ (通过缩放)表示边 e 的容量。由 [3], 可知，满足 $x \in SP(G)$ 的二次规划 $\sum_e x_e^2/c(e)$ 的最优解就是理想负载向量 ℓ^*。贪心树打包很容易推广到带容量情形；在每次迭代中，我们根据权重 $w(e) = \ell(e)c(e)$计算最小生成树。Thorup 证明了以下定理。

▶ 定理 *24*（[8]）。设 $G = (V, E)$ 为带容量图。经过$O(^m log(mn/\epsilon)\epsilon^3)$ 次迭代后，贪心树打包输出一个负载向量 ℓ，使得对于每条边 $e \in E$，$(1-\epsilon)\ell^*(e) \leq \ell(e) \leq (1+\epsilon)\ell^*(e)$。

我们注意到，若所有容量均为 1（或相同），则 Thorup 的保证为$|\ell(e) - \ell^*(e)| \leq O(\epsilon)$。对于这种情形，通过 Frank-Wolfe，我们可以得到强得多的保证，即 $||\ell - \ell^*|| \leq \epsilon$，这很容易推出逐边条件；然而，逐边保证并不能推出关于范数的保证。此外，在无权情形下，我们的迭代复杂度对 ϵ 的依赖为 $1/\epsilon^2$，而 Thorup 的为 $1/\epsilon^3$。Thorup 的保证适用于带容量情形，且所需迭代次数为强多项式。我们可以将 Frank-Wolfe 分析调整用于带容量情形，但由此得到的界将依赖于 $C = \sum_e c(e)$（在无权情形下为 $C = m$）；另一方面，Frank-Wolfe 所提供的保证更强。

同一种贪心树打包算法会因采用的分析方法不同而给出不同类型的保证，这或许令人惊讶。我们尚无法对此作出完全令人满意的解释，但指出以下几点。Thorup 的分析是对树打包标准乘法权重更新（MWU）类型分析的一种非平凡改进 [48, 49]。关于乘法权重更新（MWU）的优秀综述，请参见 [50] 。正如 [6], 中所指出的，如果采用 Frank-Wolfe 方法（结合 $\gamma = 1/(k+1)$）以及乘法权重更新框架中常用的 softmax 势函数，那么得到的算法同样会是贪心树打包算法。Fujishige 使用二次目标函数来保证最优解是唯一的极大基，但事实上，任何递增的强凸函数都足以达到这一目的。在对 B_f上的线性函数进行优化时，由于贪心算法对此类问题具有最优性，决定基的唯一因素是根据权重向量对 V 中的元素进行排序；权重本身并不重要。因此，将 Frank-Wolfe 方法应用于不同的凸目标函数，可能得到相同的贪心树／基打包算法。不过，具体采用的目标函数会影响经过若干次迭代后所能获得的保证。softmax 目标函数更适合获得相对误差保证，而二次目标函数更适合获得加性误差保证。由于在容量约束情形下，Thorup 的分析能够提供逐边保证，因此其分析更为复杂。我们希望能有一种统一的分析方法来解释相对保证和加性保证，并将其留作未来研究的一个有趣方向。

参考文献

1 Digvijay Boob、Yu Gao、Richard Peng、Saurabh Sawlani、Charalampos Tsourakakis、Di Wang 和 Junxing Wang。Flowless：无需流计算即可提取最稠密子图，第 573–583 页。计算机协会，纽约，纽约州，美国，2020年。 2 Chandra Chekuri、Kent Quanrud 和 Manuel R. Torres。最稠密子图：超模性、迭代剥离与流。载于《2022年 ACM-SIAM 离散算法年会（SODA）论文集》，第 1531–1555 页，2022年。 3 Satoru Fujishige。关于权重向量的多拟阵字典序最优基。《运筹学数学》，5(2):186–196，1980年。 4 Marguerite Frank 和 Philip Wolfe。一种二次规划算法。《海军研究后勤季刊》，3(1-2):95–110，1956年。 5 Martin Jaggi。重新审视 Frank-Wolfe：无投影稀疏凸优化。载于 Sanjoy Dasgupta 和 David McAllester（编），《第30届国际机器学习会议论文集》，《机器学习研究论文集》第 1 卷，第 427–435 页，美国佐治亚州亚特兰大，2013年6月17–19日。PMLR。 6 Elfarouk Harb、Chandra Chekuri 和 Kent Quanrud。用于最稠密子图和分解的更快速、可扩展算法。 7 Mikkel Thorup。全动态最小割。《组合数学》，27(1):91–127，2007年。初步版本发表于 2001 年 ACM 计算理论研讨会论文集。 8 Mikkel Thorup。通过确定性贪心树打包求解最小 k 路割。载于《第四十届 ACM 计算理论研讨会论文集》，第 159–166 页，2008年。 9 A. V. Goldberg。寻找最大密度子图。技术报告 UCB/CSD-84-171，加州大学伯克利分校 EECS 系，1984年。 10 Jean-Claude Picard 和 Maurice Queyranne。一种求解若干非线性 0-1 规划问题的网络流方法及其在图论中的应用。《网络》，12(2):141–159，1982年。 11 Alexander Schrijver 等。组合优化：多面体与效率，第 24 卷。施普林格，2003年。 12 Satoru Fujishige。次模函数与优化。爱思唯尔，2005年。 13 Chenhao Ma、Yixiang Fang、Reynold Cheng、Laks V.S. Lakshmanan、Wenjie Zhang 和 Xuemin Lin。用于大型有向图最稠密子图发现的高效算法。载于《2020年 ACM SIGMOD 数据管理国际会议论文集》，SIGMOD '20，第 1051–1066 页，美国纽约，纽约州，2020年。计算机协会。 14 Maximilien Danisch、T.-H. Hubert Chan 和 Mauro Sozio。通过凸规划实现大规模密度友好图分解。载于《第26届万维网国际会议论文集》，WWW '17，第 233–242 页，日内瓦共和国和州，瑞士，2017年。国际万维网会议指导委员会。 15 Charalampos Tsourakakis。k-团最稠密子图问题。载于《第24届万维网国际会议论文集》，WWW '15，第 1122–1132 页，日内瓦共和国和州，瑞士，2015年。国际万维网会议指导委员会。 16 Bintao Sun、Maximilien Danisch、T-H. Hubert Chan 和 Mauro Sozio。Kclist++：一种在大型图中寻找 k-团最稠密子图的简单算法。VLDB基金会会刊，13(10):1628–1640，2020年6月。 17 Reid Andersen 和 Kumar Chellapilla。具有规模界限的稠密子图。载于 Konstantin Avrachenkov、Debora Donato 和 Nelly Litvak（编），《网络图算法与模型》，第 25–37 页，柏林，海德堡，2009年。施普林格·柏林·海德堡。 18 Charalampos Tsourakakis、Francesco Bonchi、Aristides Gionis、Francesco Gullo 和 Maria Tsiarli。比最稠密子图还要稠密：提取具有质量保证的最优准团。载于《第19届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集》，KDD '13，第 104–112 页，美国纽约，纽约州，2013年。计算机协会。

19 Alessandro Epasto、Silvio Lattanzi 和 Mauro Sozio。演化图中的高效最密子图计算。载于第24届万维网国际会议论文集，WWW '15，第300–310页，日内瓦共和国和州，瑞士，2015年。国际万维网会议指导委员会。20 Andrew McGregor、David Tench、Sofya Vorotnikova 和 Hoa T. Vu。动态图流中的最密子图。载于 Giuseppe F. Italiano、Giovanni Pighizzini 和 Donald T. Sannella（编），2015年计算机科学数学基础，第472–482页，柏林、海德堡，2015年。施普林格·柏林·海德堡。21 Polina Rozenshtein、Nikolaj Tatti 和 Aristides Gionis。发现交互网络中的动态社区。载于 Toon Calders、Floriana Esposito、Eyke Hüllermeier 和 Rosa Meo（编），数据库中的机器学习与知识发现，第678–693页，柏林、海德堡，2014年。施普林格·柏林·海德堡。22 Oana Denisa Balalau、Francesco Bonchi、T-H. Hubert Chan、Francesco Gullo 和 Mauro Sozio。寻找总密度最大且重叠受限的子图。载于第八届ACM网络搜索与数据挖掘国际会议论文集，WSDM '15，第379–388页，纽约，纽约州，美国，2015年。计算机协会。23 Yuko Kuroki、Atsushi Miyauchi、Junya Honda 和 Masashi Sugiyama。通过模糊图反馈在线发现稠密子图。载于国际机器学习会议，2020年。24 Albert Angel、Nikos Sarkas、Nick Koudas 和 Divesh Srivastava。流式边权更新下的稠密子图维护与实时故事识别。VLDB基金会会刊，5(6)：574–585，2012年2月。25 Kijung Shin、Tina Eliassi-Rad 和 Christos Faloutsos。Corescope：使用k核分析进行图挖掘——模式、异常与算法。载于2016年第16届IEEE国际数据挖掘会议（国际数据挖掘会议），第469–478页，2016年。26 Xiangfeng Li、Shenghua Liu、Zifeng Li、Xiaotian Han、Chuan Shi、Bryan Hooi、He Huang 和 Xueqi Cheng。Flowscope：基于图识别洗钱行为。载于美国人工智能协会年会，2020年。27 Tommaso Lanciano、Atsushi Miyauchi、Adriano Fazzone 和 Francesco Bonchi。最密子图问题及其变体综述，2023年。28 Saurabh Sawlani 和 Junxing Wang。近最优的全动态最密子图。载于 Konstantin Makarychev、Yury Makarychev、Madhur Tulsiani、Gautam Kamath 和 Julia Chuzhoy（编），第52届ACM SIGACT计算理论年会论文集，2020年计算理论研讨会，芝加哥，美国伊利诺伊州，2020年6月22–26日，第181–193页。美国计算机协会，2020年。29 Aleksander B. G. Christiansen、Jacob Holm、Ivor van der Hoog、Eva Rotenberg 和 Chris Schwiegelshohn。自适应出向定向及其应用，2023年。30 Li Chen、Rasmus Kyng、Yang P. Liu、Richard Peng、Maximilian Probst Gutenberg 和 Sushant Sachdeva。近线性时间内的最大流与最小费用流，2022年。31 Moses Charikar。寻找图中稠密分量的贪心近似算法。载于 Klaus Jansen 和 Samir Khuller（编），组合优化近似算法，第84–95页，柏林、海德堡，2000年。施普林格·柏林·海德堡。32 Fragkiskos D Malliaros、Christos Giatsidis、Apostolos N Papadopoulos 和 Michalis Vazirgiannis。网络的核分解：理论、算法与应用。VLDB期刊，29(1)：61–92，2020年。33 Yuichi Asahiro、Kazuo Iwama、Hisao Tamaki 和 Takeshi Tokuyama。贪心地寻找稠密子图。算法杂志，34(2)：203–221，2000年。34 Bahman Bahmani、Ashish Goel 和 Kamesh Munagala。面向MapReduce的高效原始-对偶图算法。载于网络图算法与模型国际研讨会，第59–78页。施普林格，2014年。35 Digvijay Boob、Saurabh Sawlani 和 Di Wang。用于混合装箱与覆盖线性规划的更快宽度相关算法。神经信息处理系统进展第32卷（2019年神经信息处理系统会议），2019年。36 H Narayanan。次模函数的主划分格。线性代数及其应用，144：179–216，1991年。

37 Hariharan Narayanan。次模函数与电网络，第54卷。爱思唯尔，1997年。38 Satoru Fujishige。重新审视主分割理论。组合优化研究趋势：波恩2008，第127–162页，2009年。39 Nikolaj Tatti 和 Aristides Gionis。密度友好图分解。载于第24届万维网国际会议论文集，第1089–1099页，2015年。40 Nikolaj Tatti。密度友好图分解。ACM《数据知识发现汇刊》（TKDD），13(5)：1–29，2019年。41 David R Karger。近线性时间内的最小割。《ACM会刊》（JACM），47(1)：46–76，2000年。42 Takuro Fukunaga。从超树打包计算超图的最小多路割。载于整数规划与组合优化会议，第15–28页。施普林格，2010年。43 Chandra Chekuri、Kent Quanrud 和 Chao Xu。最小k割的线性规划松弛与树打包。SIAM离散数学杂志，34(2)：1334–1353，2020年。44 Daniel Lokshtanov、Saket Saurabh 和 Vaishali Surianarayanan。最小k割的参数化近似方案。SIAM计算杂志，（0）：FOCS20–205，2022年。45 Jason Li。简单图的更快最小k割。载于2019年IEEE第60届计算机科学基础年会（计算机科学基础研讨会），第1056–1077页。IEEE，2019年。46 Anupam Gupta、David G Harris、Euiwoong Lee 和 Jason Li。k割问题的最优界。美国计算机协会《ACM会刊》（JACM），69(1)：1–18，2021年。47 Jack Edmonds。次模函数、拟阵与某些多面体。载于 R. Guy、H. Hanani、N. Sauer 和 J. Schönheim（编），组合结构及其应用（组合结构及其应用国际会议论文集，加拿大阿尔伯塔省卡尔加里，1969年；R. Guy、H. Hanani、N. Sauer、J. Schönheim 编），纽约，1970年。戈登与布里奇。48 Serge A. Plotkin、David B. Shmoys 和 Éva Tardos。分数打包与覆盖问题的快速近似算法。《运筹学数学》，20(2)：257–301，1995年。49 N. Young。不求解线性规划的随机舍入。载于ACM-SIAM离散算法研讨会，1995年。50 Sanjeev Arora、Elad Hazan 和 Satyen Kale。乘法权重更新方法：一种元算法及其应用。《计算理论》，8(1)：121–164，2012年。

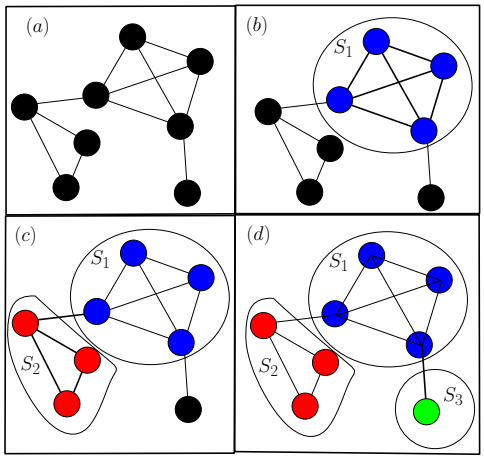


图 1 最稠密子图分解示例。最稠密子图 S_1 显示在（b）中，其中蓝色顶点的密度为 $6/4 = 1.5$。我们将 S_1（蓝色顶点）“收缩”，并找到密度为（$3+1$）$/3 = 4/3$的最稠密子图 S_2。最后，我们收缩 S_2，并找到密度为（$0+1$）$/1 = 1$的 S_3。

6 辅助引理

▶ **引理25**。对于 $k \geq 0$

$$\sum_{u \in V} b^{(k)}(u) d^{(k+1)}(u) \leq \left(\min_{orientation\ \vec{d}} \sum_{u \in V} b^{(k)}(u) \vec{d}(u) \right) + \frac{\delta C_f}{k+2}$$

证明。 使用引理 21，并以 $(k+1)b^{(k)}$ 作为权重，我们有

$$\sum_{u \in V} b^{(k)}(u) d^{(k+1)}(u) \leq \left(\min_{\text{orientation } \vec{d}} \sum_{u \in V} b^{(k)}(u) \vec{d}(u) \right) + \frac{\sum_{u \in V} deg_G(u)^2}{k+1}$$

我们在引理26中证明， $C_f \geq 2m$ 使用概率方法，因此对于 $\delta = \Theta(\sum_u deg_G(u)^2)$，有

$$\frac{\sum_{u \in V} deg_G(u)^2}{k+1} \leq \frac{\delta C_f}{k+2}$$

◀

▶ **引理26**。 设 $f(b) = \sum_u b_u^2$ 为某个定向负载向量的平方和。那么$2m \leq C_f \leq 2\sum_u deg_G(u)^2$。

证明。 令 $\mathcal{D}$ 为有效定向的集合。根据 C_f 的定义并化简可得

$$C_f = \sup_{x,s \in \mathcal{D}, \gamma \in [0,1]} \frac{2}{\gamma^2}(f(x+\gamma(s-x)) - f(x) - \langle \gamma(s-x), 2x \rangle) = \sup_{x,s \in \mathcal{D}} 2(s-x)^T(s-x)$$

设 x， $s \in \mathcal{D}$。显然 x_u， $s_u \leq deg_G(u)$，因为它们是定向。因此 $(s_u - x_u)^2 \leq deg_G(u)^2$。对 u 求和即可得到上界。

对于下界，我们使用概率方法。任意选择一条边 (i,j)，以概率 $1/2$ 朝向 i ，以概率 $1/2$ 朝向 j。这会产生一个负载 b^1。重复此操作。

独立地 以产生负载 b^2。请注意，对于任意顶点 u， b_u^1， $b_u^2 \sim Bin(deg_G(u),\ 1\,2)$ 且彼此独立。因此， $b_u^1 - b_u^2$ 的方差为 $(deg_G(u) + deg_G(u)) \times 1/2 \times 1/2 = \frac{1}{2} deg_G(u)$。因此，我们有：

$$\frac{1}{2} deg_G(u) = \mathbb{E}\left[(b_u^1 - b_u^2)^2\right] - \mathbb{E}\left[(b_u^1 - b_u^2)\right]^2 = \mathbb{E}\left[(b_u^1 - b_u^2)^2\right]$$

因此我们有

$$\mathbb{E}\left[2(b^1 - b^2)^T (b^1 - b^2)\right] = 2 \sum_{u \in V} \mathbb{E}\left[(b_u^1 - b_u^2)^2\right] = 2m$$

因此，必定存在一种取值，使得 $2(b^1 - b^2)^T(b^1 - b^2)$ 至少达到期望值 $2m$，从而上确界至少为 $2m$。

◀

7 证明

7.1 定理 8 的证明

证明。 我们将指出，这个证明是对 Jaggi 证明的略微改编 [5], ，并将其收录于此以求完整。

设$\hat{\mathbf{d}}^{(k+1)}$ 为第 k 次迭代中由近似线性最小化预言机返回的方向。对于任意 $\gamma \in [0, 1]$, 根据曲率常数 C_f 的定义，我们有

$$f(\mathbf{b}^{(k+1)}) = f(\mathbf{b}^{(k)} + \gamma \hat{\mathbf{d}}^{(k+1)}) \leq f(\mathbf{b}^{(k)}) + \gamma \langle \hat{\mathbf{d}}^{(k+1)} - \mathbf{b}^{(k)}, \nabla f(\mathbf{b}^{(k)}) \rangle + \frac{\gamma^2}{2} C_f \tag{1}$$

注意，我们使用的$\hat{\mathbf{d}}^{(k+1)}$ 是一个 $\frac{\delta C_f}{k+2}$ -近似线性最小化预言机，因此$\langle \hat{\mathbf{d}}^{(k+1)}, \nabla f(\mathbf{b}^{(k)}) \rangle \leq \langle \mathbf{d}^{(k+1)}, \nabla f(\mathbf{b}^{(k)}) \rangle + \frac{\delta C_f}{k+2}$。结合 (1) 式并重新整理，可得

$$f(\mathbf{b}^{(k+1)}) \leq f(\mathbf{b}^{(k)}) - \gamma g(\mathbf{b}^{(k)}) + \frac{\gamma^2}{2} C_f (1 + \delta) \tag{2}$$

其中 $g(\mathbf{b}^{(k)}) = \mathrm{h}\mathbf{d}^{(k+1)} - \mathbf{b}^{(k)}, \nabla f(\mathbf{b}^{(k)})\mathrm{i}$。接下来，记 $h(\mathbf{b}^{(k)}) = f(\mathbf{b}^{(k)}) - f(\mathbf{b}^*)$ 为原始误差。 f 的凸性意味着，线性化 $f(\mathbf{b}) + \mathrm{h}\mathbf{s} - \mathbf{b}\ \nabla f(\mathbf{b})\mathrm{i}$ 始终位于 f 的图像下方。这意味着 $g(\mathbf{b}^{(k)}) \geq h(\mathbf{b}^{(k)})$。结合 (2) 式，我们得到

$$f(\mathbf{b}^{(k+1)}) \leq f(\mathbf{b}^{(k)}) - \gamma h(\mathbf{b}^{(k)}) + \frac{\gamma^2}{2} C_f (1 + \delta) \tag{3}$$

将 (3) 两边减去 $f(\mathbf{b}^*)$，得到

$$h(\mathbf{b}^{(k+1)}) \leq (1 - \gamma) h(\mathbf{b}^{(k)}) + \frac{\gamma^2}{2} C_f (1 + \delta) \tag{4}$$

令 $C = \frac{1}{2} C_f(1 + \delta)$ 且 $\epsilon_k = h(\mathbf{b}^{(k)})$。由 (4) 得到递推关系

$$\epsilon_{k+1} \leq (1 - \gamma)\epsilon_k + \gamma^2 C \tag{5}$$

我们断言 $\epsilon_k \leq \frac{4CH_{k+1}}{k+1}$，这就意味着该定理。对于 $k = 0$, (5) 与 $\gamma = \frac{1}{0+1}$ 一起推出 $\epsilon_{0+1} \leq C \leq 4C$。对于 $k \geq 1$，我们希望 (5) 的右端满足

$$(1 - \frac{1}{k+1}) \frac{4CH_{k+1}}{k+1} + \frac{C}{(k+1)^2} \leq \frac{4CH_{k+2}}{k+2}$$

或者，重新排列后，这等同于 $4(k+2)H_{k+1} + k(3k+4) \geq 0$, 对于 $k \geq 1$ 成立。◀

7.2 引理 9 的证明

证明。 我们将证明 $h(X) = f(V \setminus X)$ 对于次模 f是次模的。这意味着 $-f(V \setminus X)$ 是超模的，并且由于 $f(V)$ 是模的，因此 g 是超模的。

令 $A, B \subseteq V$。为说明 h 是次模的，我们有以下不等式：

$$h(A) + h(B) = f(V \setminus A) + f(V \setminus B) \geq f(V \setminus A \cup V \setminus B) + f(V \setminus A \cap V \setminus B)$$

No注意， $V \setminus A \cup V \setminus B = V \setminus (A \cap B)$。此外， $V \setminus A \cap V \setminus B = V \setminus (A \cup B)$。因此 e

$$h(A) + h(B) \geq f(V \setminus (A \cup B)) + f(V \setminus (A \cap B)) = h(A \cup B) + h(A \cap B)$$

◀

7.3 引理 10 的证明

证明。 令 $S_1, S_2 \subseteq V$为达到最大值 $\lambda = {}^{f}{}_{(S)\,|S|}$ 的极大集合。由超模性可得

$$\frac{f(S_1 \cup S_2)}{|S_1 \cup S_2|} = \frac{f(S_1 \cup S_2)}{|S_1| + |S_2| - |S_1 \cap S_2|} \geq \frac{f(S_1) + f(S_2) - f(S_1 \cap S_2)}{|S_1| + |S_2| - |S_1 \cap S_2|}$$

No注意到 $f(S_1 \cap S_2) \leq \lambda |S_1 \cap S_2|$，这是由 λ 的最优性所致，这意味着后续链式关系。

$$\frac{f(S_1 \cup S_2)}{|S_1 \cup S_2|} \geq \frac{f(S_1) + f(S_2) - \lambda|S_1 \cap S_2|}{|S_1| + |S_2| - |S_1 \cap S_2|} = \frac{\lambda |S_1| + \lambda |S_2| - \lambda |S_1 \cap S_2|}{|S_1| + |S_2| - |S_1 \cap S_2|} = \lambda$$

由 $\lambda, f(S_1 \cup S_2) = \lambda |S_1 \cup S_2|$的最优性可知。由 S_1， $S_2, S_1 = S_1 \cup S_2 = S_2$ 的极大性可知。◀

7.4 引理 12 的证明

证明。 设 S_1、 $S_2 \subseteq V$是使比值 $(|V| - |S|)/(f(V) - f(S))$达到最小值的极小集合，其值为 λ。根据 $g(S) = f(V) - f(S)$的超模性，我们有

$$\frac{|V| - |S_1 \cap S_2|}{f(V) - f(S_1 \cap S_2)} = \frac{|V| - |S_1 \cap S_2|}{g(S_1 \cap S_2)} \leq \frac{|V| - |S_1 \cap S_2|}{g(S_1) + g(S_2) - g(S_1 \cup S_2)}$$

注意到 $|V| - |S_1 \cap S_2| = |V| - |S_1| + |V| - |S_2| - |V| + |S_1 \cup S_2|$。此外，根据 λ 的最优性，$\lambda g(S_1 \cup S_2) \leq |V| - |S_1 \cup S_2|$。于是我们得到如下不等式链。

$$\frac{|V| - |S_1 \cap S_2|}{f(V) - f(S_1 \cap S_2)} \leq \frac{\lambda g(S_1) + \lambda g(S_2) - \lambda g(S_1 \cup S_2)}{g(S_1) + g(S_2) - g(S_1 \cup S_2)} = \lambda$$

根据 λ 的最优性，$(|V| - |S_1 \cap S_2|)/(f(V) - f(S_1 \cap S_2)) = \lambda$。根据 S_1 的极小性， S_2, $S_1 = S_1 \cap S_2 = S_2$。◀

7.5 定理 13 的证明

Pr <escape>ˆ ˆ证明。 在整个证明过程中参见图 2。注意， S_1，……， S_k 与稠密度相关</escape> ies

$$\frac{|V| - |S_1|}{f(V) - f(S_1)} < \frac{|S_1| - |S_2|}{f(S_1) - f(S_2)} < \ldots < \frac{|S_{k-1}| - |S_k|}{f(S_{k-1}) - f(S_k)}$$

类似地， T_1，……， $T_{k'}$ 与密度相关

$$\frac{f(V) - f(V \setminus T_1)}{|T_1|} > \frac{f(V \setminus T_1) - f(V \setminus T_1 \setminus T_2)}{|T_2|} > \ldots > \frac{f(T_{k'}) - f(\phi)}{|T_{k'}|}$$

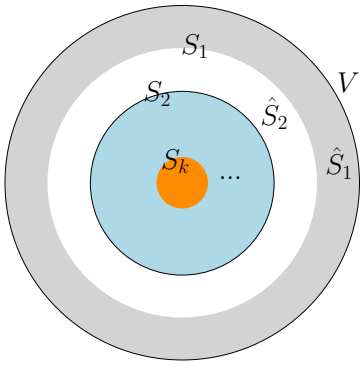


图 2 次模函数 f 的基于收缩的分解。图中展示了 $S_1,...,S_k$ 和 $\hat{} S_1,...,\hat{}S_k$ 的维恩图

$\hat{}\hat{}$ 我们通过归纳法证明该断言。假设 T_1，……，T_{i-1} 与 S_1，……，S_{i-1} 相同，并且基例是平凡的。

首先注意到 $T_i \subseteq S_{i-1}$，因为 $T_1 \cup ... \cup T_{i-1} = (V \setminus S_1) \cup (S_1 \setminus S_2) \cup ... (S_{i-2} \setminus S_{i-1}) = V \setminus S_{i-1}$ 以及 $\{T_j\}$ 的互不相交性。现在注意到，根据 S_i 的最优性，

$$\frac{|S_{i-1}| - |S_i|}{f(S_{i-1}) - f(S_i)} \leq \frac{|S_{i-1}| - |S_{i-1} \setminus T_i|}{f(S_{i-1}) - f(S_{i-1} \setminus T_i)} = \frac{|T_i|}{f(S_{i-1}) - f(S_{i-1} \setminus T_i)} = \frac{|T_i|}{f(V \setminus \bigcup_{j<i} T_j) - f(V \setminus \bigcup_{j \leq i} T_j)}$$

反过来，根据 T_i 的最优性

$$\frac{f(V \setminus \bigcup_{j<i} T_j) - f(V \setminus \bigcup_{j \leq i} T_j)}{|T_i|} \geq \frac{f(V \setminus \bigcup_{j<i} T_j) - f(V \setminus \bigcup_{j<i} T_j \setminus (S_{i-1} \setminus S_i))}{|S_{i-1} \setminus S_i|} = \frac{f(S_{i-1}) - f(S_i)}{|S_{i-1}| - |S_i|}$$

因此，$\lambda_i = \hat{}\lambda_i$。这也迫使 $S_i = S_{i-1} \setminus T_i$ 或 $T_i = S_{i-1} \setminus S_i = \hat{}S_i$。 ◀

7.6 引理19的证明

证明。考虑边的最优定向 d^*_w。将一条边反向（从(u, v) 到 (v, u)）如何影响最小化预言机的代价？ u 的出度减少 1，而 v 的出度增加 1，因此目标函数 **增加** 了 $w_v - w_u$。

假设加权贪心++ 按照以下顺序剥离顶点 $u_1, ..., u_n$。我们采用归纳法进行证明。考虑所有“错误”定向的边 $W(u_1) = \{(u_1, u_i) : i > 1, w(u_1) > w(u_i)\}$。这些边使目标函数的值相较于最优解的值增加了$\sum_{v \in W(u)} (w(u_1) - w(v))$。但请回顾，加权贪心++ 选择 u_1，因为对于所有 $v \in W(u)$，均有$w(u_1) + deg_{G'}(u_1) \leq w(v) + deg_{G'}(v)$。这意味着

$$w(u_1) - w(v) \leq \deg_{G'}(v) - deg_{G'}(u_1) \leq deg_G(v)$$

对 u_2，...，u_n应用归纳法，方向“错误”的边相对于正确方向 d^*_w 造成的加性误差总计至多为 $\sum_u deg_G(u)^2$，因为每个顶点v 从其所有邻居处贡献至多 $deg_G(v)$。◀

7.7 引理 20 的证明

证明。假设 WEIGHTED-SUPERGREEDY++ 按以下顺序剥离 V：s_{i_1}，……，s_{i_n}，而 Edmonds 算法按以下顺序剥离它们 s_1，……，s_n，其中 $w_1 \leq ... \leq w_n$。令 $A_j = \{s_{i_{j+1}}$，……，$s_{i_n}\}$。

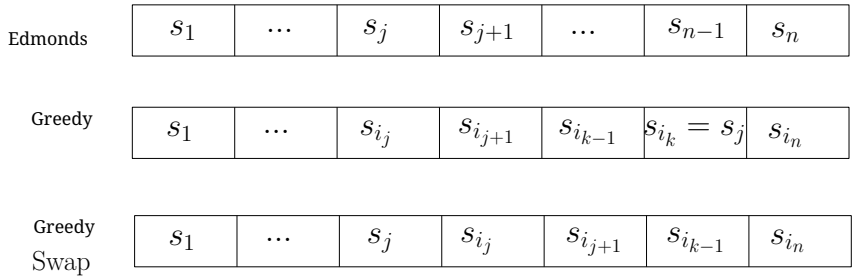


图3 引理 引理 20 的证明思路

令 j 为第一个出现分歧的索引，其中 $s_j \neq s_{i_j}$。Weighted-SuperGreedy++选择 s_{i_j} 而不是 s_{i_k} （$k \geq j$），因为

$$w(s_{i_j}) + f(s_{i_j}|A_j - s_{i_j}) \leq w(s_{i_k}) + f(s_{i_k}|A_j - s_{i_k})$$

这由超模性可知

$$w(s_{i_j}) - w(s_{i_k}) \leq f(s_{i_k} \mid V - s_{i_k}) \tag{6}$$

假设 $s_{i_k} = s_j$，那么我们会在 Weighted-SuperGreedy++ 顺序中将 s_{i_j}、$s_{i_{j+1}}$、..... $s_{i_{k-1}}$ 移到 $s_{i_k} = s_j$ 之后。（见图3）。在 Weighted-SuperGreedy++ 顺序中交换相邻元素 s_{i_a}、$s_{i_{a+1}}$ 会使内积代价变化

$$w(s_{i_{a+1}})f(s_{i_{a+1}}|A_a+s_{i_a}-s_{i_{a+1}})+w(s_{i_a})f(s_{i_a}|A_{a+1})-w(s_{i_a})f(s_{i_a}|A_a)-w(s_{i_{a+1}})f(s_{i_{a+1}}|A_{a+1})$$

$$= w(s_{i_a})(f(s_{i_a}|A_{a+1})-f(s_{i_a}|A_a))-w(s_{i_{a+1}})(f(s_{i_{a+1}}|A_{a+1})-f(s_{i_{a+1}}|A_a+s_{i_a}-s_{i_{a+1}})) \tag{7}$$

But

$$f(s_{i_a}|A_{a+1}) - f(s_{i_a}|A_a) = f(A_{a+1} + s_{i_a}) - f(A_{a+1}) - f(A_a + s_{i_a}) + f(A_a)$$

And

$$f(s_{i_{a+1}}|A_{a+1})-f(s_{i_{a+1}}|A_a+s_{i_a}-s_{i_{a+1}}) = f(A_{a+1}+s_{i_{a+1}})-f(A_{a+1})-f(A_a+s_{i_a})+f(A_a+s_{i_a}-s_{i_{a+1}})$$

$$= f(A_a) - f(A_{a+1}) - f(A_a + s_{i_a}) + f(A_{a+1} + s_{i_a})$$

因此，7 中 $w(s_{i_a})$ $w(s_{i_{a+1}})$ 的两个系数相等。于是由 (6) 可得

$$(7) = (w_{i_a} - w_{i_{a+1}})(f(s_{i_{a+1}}|A_{a+1}) - f(s_{i_{a+1}}|A_{a+1} + s_{i_a})) \geq -f(s_{i_{a+1}}|V - s_{i_{a+1}})^2$$

因此，将所有 s_{i_j}、.....、 $s_{i_{k-1}}$移至 $s_{i_k} = s_j$至多会使成本降低 $\leq (k-j)f(s_{i_k}|V)^2 \leq nf(s_{i_k}|V-s_{i_k})^2$。对顶点的所有重新排序求和后可知，Edmonds 的排序内积与 Weighted-SuperGreedy++排序内积至多相差 $n\sum_{u\in V} f(u|V-u)^2$。对于函数 $f(S) = |E(S)|$，引理19给出的界比引理引理 20 给出的界好 n 倍。我们将改进此界留待未来工作。

◀

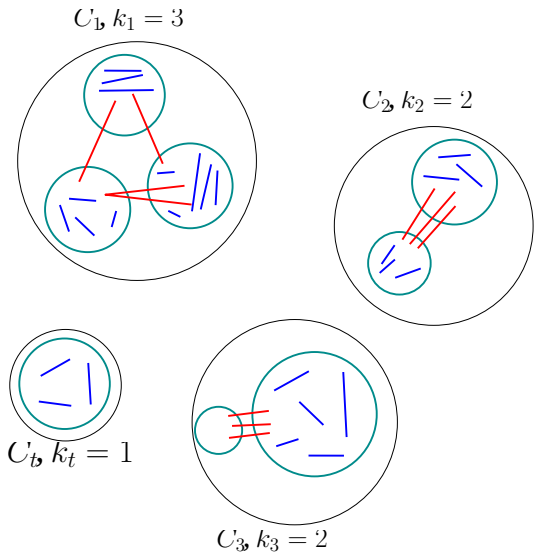


图4 蓝色边为 S_i。红色边为 $\hat{} S_i$。红色边和蓝色边合起来为 S_{i-1}。连通分量 C_q 中的红色边为 E_q。C_q 内的青色圆圈表示删除 $\hat{} S_i$ 后 C_q 内的连通分量。

7.8 引理22的证明

证明。 在第 i 次迭代中，注意到 $\hat{} S_i = S_{i-1} \backslash S_i$ 正是图 $G' = G(V, S_{i-1})$ 中从第 $i-1$ 次迭代后保留下来的、跨越划分 $P(S_i)$ 的边。此外，~~$f_{(S_{i-1})-f(S_i)}|S_{i-1}|-|S_i| = \frac{}{\kappa(S_i)-\kappa(S_{i-1})}$ $E(P(S_i))$。~~我们证明，这就是与 Thorup 的理想相对权重一致的正确值，即应将 S_i 中的边设为该值。

我们对第 i 次迭代进行归纳。

整个证明过程中请参见图4。固定 $G' = G(V, S_{i-1})$。设 G' 的连通分支为 C_1、……、C_t，其中连通分量 C_q 在删除 $\hat{} S_i = S_{i-1} \setminus S_i$ 后有 k_q 个连通分支，并为 $\hat{} S_i$ 中的跨割边贡献边 E_q。设 $k = \sum_i k_i$。

I若对于某个连通分量 C_q，有 ~~$\frac{E_q}{k_q - 1} < \frac{|S_{i-1}|-|S_i|}{k-1}$，~~则对于 $S'_i = S_{i-1} \setminus E_q$，有

$$\frac{|S_{i-1}| - |S'_i|}{\kappa(S_{i-1} - \kappa(S'_i))} = \frac{|E_q|}{k_q - 1} < \frac{|S_{i-1}| - |S_i|}{k - 1}$$

这与 S_i 的最优性矛盾。

因此，对于所有满足 $k_q > 1$ 的连通分量 C_q，~~都有 $\frac{E_q}{k_q - 1} \geq \frac{|S_{i-1}|-|S_i|}{k-1}$~~。但此时对于 $S'_i = S_i \cup E_q$，有

$$\frac{|S_{i-1}| - |S'_i|}{\kappa(S'_i) - \kappa(S_{i-1})} = \frac{|S_{i-1}| - |S_i| - |E_q|}{\kappa(S_i) - \kappa(S_{i-1}) - k_q + 1} \leq \frac{|S_{i-1}| - |S_i|}{\kappa(S_i) - \kappa(S_{i-1})}$$

由于 S_i 的最优性，等式必须成立。

因此，必然有：若 $k_q > 1$，则 $(k_1 - 1)/|E_q| = (k-1)/|\hat{} S_i|$。所以，该算法按照 Thorup 的算法为 S_i 中的边设置正确的密度。◀