组约束下允许并列的 Kemeny 中位排名

Maurizio Romano* Luca Coraggio† Claudio Conversano*
Antonio D'Ambrosio†‡

arXiv:2611.28925v1 [stat.CO] 21 Nov 2026

摘要

中位排名是指使一组评审者给出的排名与其之间的 Kemeny 距离之和最小的排名。在许多应用中，共识需要符合预先规定的结构：例如，指定桶的数量，指定全部或部分桶的大小（如 Q 方法论中的强制分布），或采用 top-k list 的形式。我们指出，所有这些问题本质上都是同一个问题：在桶大小模式可接受的排名中寻找中位排名；而这些问题仅通过组合输入矩阵依赖于数据。该矩阵由 Emond 和 Mason 于 2002年 定义。我们基于这一矩阵构建了 BOLT，这是一种确定性启发式算法，它结合了一组初始解、一次移动一个对象且能够创建或移除并列的局部搜索，以及将固定顺序的对象精确划分为符合可接受模式的桶的最优划分。我们还为中位 top-k list 提供了精确解。我们在真实数据和模拟数据上，将 BOLT 的解与通过二元线性规划 formulation 认证的最优解进行比较；该 formulation 针对 Kemeny 距离和组约束作了调整，并将其与文献中已有的方法进行比较。对 Q排序、top-k lists 以及大型稀疏评分数据集的应用展示了受约束问题在实践中的产生方式，也说明了如何处理评审者未排序的对象会改变解。

关键词： 中位排名；Kemeny 距离；组合输入矩阵；桶排序；启发式方法；二进制线性规划

1 引言

偏好排名广泛收集于众多领域，从社会与行为科学到市场营销、政治学、信息检索和生物信息学（Marden，1996；Alvo and Yu，2014）。当一组 m 评审者对一组 n 对象（或项目、备选方案）进行排名时，一项常见任务是将他们的偏好概括为一个单一排名，称为共识排序或中心排名。在定义共识排序的众多方式中，将其描述为使自身与评审者所给排名之间的距离之和最小的排名，这一定义已成为最主流的定义，这可能要归功于 Kemeny（1959年）以及 Kemeny and Snell（1962年）对中位排名所作的形式化定义。随着时间的推移，所谓的肯梅尼问题已成为排名聚合问题的同义词，而没有顾及

过多关注用于解决该问题的距离，或排名本身的性质。Kemeny 和 Snell（1962）定义的距离建立在一套形式严谨的公理体系之上，能够自然地处理并列，无论并列出现在评审者的排名中，还是中位排名中；在中位排名中，并列表示真正的无差异，或者如 Emond 和 Mason（2002）所述，是一种“一致性的积极表述”。

*卡利亚里大学商业与经济系，意大利。
†那不勒斯费德里科二世大学经济与统计系，意大利。
‡通讯作者：antdambr@unina.it

从历史上看，排名聚合问题主要关注如何在不考虑评审者排名中或中位排名本身存在并列排名（或弱排名）的情况下，求得中心排名，无论问题是否处于 Kemeny 公理框架之内（例如，Cook 和 Saipe，1976年；Young 和 Levenglick，1978；Cook 和 Seiford，1978；Barthélemy 等人，1989；Fagin 等人，2004年；Cook，2006年；Cook 等人，2007年；Ali 和 Meilă，2012）；在这种设置下，该问题已被证明是 NP 困难的（Bartholdi 等人，1989；Dwork 等人，2001）。

本文中，除非另有明确说明，我们统称为“排名”，并以此涵盖所有可能的含并列排名情况。含并列的中位排名精确方法已取得显著进展。Yoo 和 Escobedo（2021）提出了一种二元线性规划公式，其中，对于每一对对象组成的有序对，均使用一个二元变量表示第一个对象是排在第二个对象之前，还是与其并列。通过针对对象三元组设置线性约束来施加传递性。作者还提出了适用于含并列排名的一种社会选择性质，可将某些实例分解为规模更小的实例。Akbari 和 Escobedo（2021）为含并列的问题提供了下界。此外，Akbari 和 Escobedo（2023b）将精确方法和启发式方法扩展到一类距离，其中对并列采用参数化惩罚。Kemeny 距离是这类距离的一个特例。Kawashima 和 Sukegawa（2026）研究了含并列的线性排序问题，其中包括含并列的中位排序问题。他们给出了相关条件，在这些条件下，任意两个对象在每个最优解中都按照给定顺序排列或并列。当解中不允许并列，但评审者给出的排名中可以出现并列时，Azzini 和 Munda（2020）的 MorK 算法、Azzini 等人（2027）的算法、Rico 等人（2023）的精确算法，以及 Phung 和 Hamel（2025）的空间缩减技术，均可在排列空间中求得中位排名。尽管取得了这些进展，但对于包含数百个对象的实例，精确方法在计算上仍然代价高昂，尤其是在评审者意见不一致时。

在同一公理框架下，人们提出了启发式方法。Amodio 等人（2016年）提出了 QUICK 和 FAST，D’Ambrosio 等人（2017年）提出了一种名为 DECoR 的差分进化算法。Romano 等人（2025）提出了一种粒子群优化算法。Badal 和 Das（2018年）针对完整排序提出了子迭代收敛算法 FUR 和 SIgFUR。Badal 和 Singh（2026年）将这些算法扩展到含并列的排名。Akbari 和 Escobedo（2023a）提出了近似孔多塞划分，该方法将大型实例分解为较小的子问题，并为解的质量提供实例特定界。Lorena et al.（2022年）提出了一种偏置随机键遗传算法，并结合局部搜索，其插入和交换移动可在线性时间内评估。该算法的目标函数取自 Yoo et al.（2020），根据一个矩阵计算；该矩阵与 Emond 和 Mason（2002年）的组合输入矩阵一致，其中评判者未排序的对象不产生影响。Andrieu 等人（2023）提出了一个框架，其中包含若干针对含并列的不完整排序的 Kemeny 分数推广形式，这些形式包括针对完整排序的 Kemeny 距离。他们还提出了一种精确算法、划分方法，以及 KwikSort（Ailon 等人，2008）的推广版本和 BioConsert 局部搜索（Cohen-Boulakia 等人，2011）的推广版本。这些启发式方法大多具有随机性，因此多次运行可能产生不同的解。

其他方法具有保证的近似比。KwikSort 算法是一种随机化枢轴算法，后由 van Zuylen 和 Williamson （ 2009 ）去随机化；Brancotte 等人（ 2015 ）将其改编用于含并列的排名。Fagin 等人（2004年）证明，通过位置中位数聚合含并列的排名，再将有序对象最优分组到各个桶中，对于一个包含 Kemeny 距离的距离族而言，可实现常数因子近似。一些为“含并列名次的排序聚合”开发的方法优化的是不同的目标值：Cohen-Boulakia 等人（2011年）和 Brancotte等人（2015）对排序方向相反的对象对，以及仅在两个排名之一中并列的对象对赋予相同代价，而这并非 Kemeny 距离。

另一个相关问题是最优桶序问题（Gionis 等人，2006年；Ukkonen

等人，2009），其中数据用优先比例矩阵概括。其解是与数据在 L_1 范数下距离最近的桶排序。该问题的启发式算法（Aledo 等人，2017年、2018年、2021年）、精确表述（Aledo 等人，2025a）以及扩展到一组桶排序的版本（Aledo 等人，2025b），解决的目标各不相同。例如，考虑 100 位评审者和两个对象 a 和 b：其中 51 位评审者偏好 a 而非 b，49 位评审者偏好 b而非 a，此时最优桶排序将 a 和 b并列，而中位排名则将 a 排在 b 之前（D'Ambrosio 等人，2017年）。

在许多应用中，共识具有预先规定的结构。中位约束分桶排序（MCBO，D'Ambrosio 等人，2019年）是桶数预先规定的中位排序。在 Q方法论（Stephenson，1935；Brown，1980；Watts 和 Stenner，2012）和绩效评估中，对象必须分配到大小预先规定的有序类别中。top-k list 仅要求对前 k 个对象进行排名（Fagin 等人，2003）。最近，Aledo 等人（2025a）将桶数预先规定、桶大小预先规定，以及采用 top-k 结构变体的最优桶序问题表述为一个整数线性规划问题。

本文在 Kemeny 公理框架下探讨有组约束和无组约束的中位排序问题，全文均以组合输入矩阵的性质为基础。自 Emond 和 Mason（2002年）以来，人们便已知晓这一矩阵；精确公式有时也会以其他名称使用它（Yoo 和 Escobedo，2021年）。在本文中，我们将其作为所有过程共同处理的单一对象。目标函数、初始解、局部搜索的移动、固定顺序的最优划分为桶、下界和精确表述，均以该矩阵为基础，因此其代价不取决于评审者的数量。

本研究的贡献有两方面。首先，我们通过解所允许具有的模式，也就是其桶的大小，来描述所有这些问题。中位排名、MCBO、具有全部或部分指定桶大小的中位排名、中位 top-k list，以及排列空间中的中位排名，实际上都是同一个问题，只是可接受模式集合不同。对于中位 top-k list，目标函数分解为选取 k个对象，以及针对所选对象的一个中位排序问题。这引出了一种精确算法。随后，我们提出 BOLT，这是一种启发式方法，使用统一方案求解所有这些问题：初始解；每次移动一个对象的局部搜索；以及将固定顺序划分为具有可接受模式的桶时的最佳划分。虽然这些组成部分单独来看都并非创新，但我们的贡献在于将它们结合应用于组合输入矩阵，将其扩展到部分模式和中位 top-k list，并系统评估该启发式方法与经认证的最优解相比的表现。为此，我们将 Yoo 和 Escobedo（2021年）的二元线性规划 formulation 以及 Aledo 等人（2025a）的约束，调整为适用于 Kemeny 距离和部分模式。我们结合这些调整，以及其他方法在各自基准测试上已发布的结果，评估 BOLT 在真实数据和模拟数据上的解的质量。

本文结构如下：第2节回顾中位排序问题和组合输入矩阵，第3节讨论组合输入矩阵的作用。第4节定义受约束问题。第 5 节介绍 BOLT，第 6 节介绍精确表述。第7节报告实验评估，第 8 节讨论应用。第9节总结。推导过程、伪代码和完整表格见补充材料。

2 Kemeny 公理化方法下的中位排名

设 $\mathcal{Z}^n$ 为 n 个对象的排名全集。其基数为

$$|\mathcal{Z}^n| = \sum_{b=1}^{n} b! \left\{ n \atop b \right\}, \tag{1}$$

其中 $\left\{ n \atop b \right\}$ 是第二类斯特林数，即将包含 n 个对象的集合划分为 b 个非空子集的方法数。这些子集就是桶，而因子 $b!$则计入它们的顺序，因此

$b!\left\{ {n \atop b} \right\}$ 是恰好包含 b 个桶的排名数量（D'Ambrosio 等人，2019年）。数值 $|\mathcal{Z}^n|$ 称为第 n 个有序贝尔数（或 Fubini 数）（Gross，1962年；Good，1975）。完全排名的全集，即前 n 个整数的排列，记为 $\mathcal{S}^n$，其中 $|\mathcal{S}^n| = n!$。排名可以用两种方式表示（Marden，1996）。秩向量（或狭义上的排名）为每个对象指定其位置；顺序向量（或排序）则按偏好从高到低列出对象，并用花括号括起并列的对象。例如，若对象 a, b, c, d 按字母顺序排列，则秩向量 (2, 3,1, 2) 和顺序向量 $\langle c, \{a, d\}, b\rangle$ 描述的是同一个排名，其中a 和 d 并列第二。下文中，我们用"排名"一词指代这两种表示方式，仅在需要时才说明具体采用哪种表示。

设 Y和 X 是对 n个对象的两个排名。按照 Kemeny and Snell（1962年）的定义， X 和 Y之间的 Kemeny 距离为

$$d(X, Y) = \frac{1}{2} \sum_{i,j=1}^{n} |x_{ij} - y_{ij}|\,, \tag{2}$$

其中，x_{ij} 和y_{ij} 分别是 X 和 Y的得分矩阵中的元素：如果第 i个对象排在第 j个对象之前，则为 $x_{ij} = 1$；如果排在其后，则为 $x_{ij} = -1$；如果两个对象并列，或 $i = j$，则为 $x_{ij} = 0$。Kemeny and Snell（1962年）证明，这一距离是满足 Kemeny 公理的唯一度量：排名的任何距离都必须是一个度量；在对对象重新标记时必须保持不变；在测量上必须具有一致性；并且其最小正值必须等于一。

Emond 和 Mason（2002年）定义了秩相关系数 τ_X，并证明它与 Kemeny 距离具有如下明确关系：

$$\tau_X(X, Y) = 1 - \frac{2\,d(X, Y)}{n(n-1)} = \frac{\sum_{i,j=1}^{n} x_{ij}\, y_{ij}}{n(n-1)}. \tag{3}$$

公式 3 右侧需要对得分矩阵进行略微修改：如果第 i个对象的排名高于或等于第 j个对象，则他们将对象对 (i, j) 的得分设为 1，否则设为 -1，其中 $x_{ii} = 0$。因此，对象 i 和 j 并列时得到 $x_{ij} = x_{ji} = 1$。下文中，得分矩阵的元素均按此方式编码。

设 $X^{(1)}, \ldots, X^{(r)}$ 是由 n 个对象的不同排名组成的集合，每个排名均关联一个权重w_h，其中 $\sum_{h=1}^{r} w_h = m$， m 为评审者人数。权重也可以具有其他含义，例如相对频率或重要性权重：由于所有权重同时乘以同一个正数不会改变中位排名，因此我们总可以对权重重新缩放，使其总和为 m。组合输入矩阵 $\mathbf{C}$ （Emond 和 Mason，2002年）的元素为

$$c_{ij} = \sum_{h=1}^{r} w_h\, x_{ij}^{(h)}, \tag{4}$$

也就是说，它是各评委排名对应得分矩阵的加权和。一个排名 Y 在评审者中的平均值 τ_X为

$$\bar{\tau}_X(Y) = \frac{\sum_{i,j=1}^{n} c_{ij}\, y_{ij}}{m\, n(n-1)}, \tag{5}$$

ˆ 中位排名是满足下列条件的排名（或多个排名） Y

$$\hat{Y} = \operatorname*{arg\,min}_{Y \in \mathcal{Z}^n} \sum_{h=1}^{r} w_h\, d(X^{(h)}, Y) = \operatorname*{arg\,max}_{Y \in \mathcal{Z}^n} \bar{\tau}_X(Y). \tag{6}$$

值得注意的是，数据仅通过 $\mathbf{C}$ 进入问题：无论评审者数量多少，问题的维度都由对象数量决定。从现在起

我们将平均值 τ_X 用作本文所讨论所有问题的目标函数。第 3.3 节将讨论部分评审者未排序的对象。

与 $\bar{\tau}_X(Y)$ 对应的距离总和为 $\sum_h w_h d(X^{(h)},\ Y) = m\,n(n-1)\,(1-\bar{\tau}_X(Y))$，因此，平均值 τ_X 相差 δ，对应于距离总和相差 $\delta\, m\, n(n-1)/2$。

3 组合输入矩阵的核心作用

本文中的所有问题都采用相同的目标函数（公式（5）中的平均值 τ_X），其区别仅在于所搜索的 $\mathcal{Z}^n$ 子集不同。在本节中，我们将目标函数改写为一种能够明确体现桶作用的形式。随后，我们推导出本文所有程序所使用的结论：第 5 节中的启发式方法；第 6 节中的下界和精确表述。

3.1 仅计算有序对

设 $Y \in \mathcal{Z}^n$，并令 $i \prec_Y j$ 表示对象 i 属于一个 Y 桶，该桶位于对象 j所在桶之前。回顾一下，按照 Emond 和 Mason（2002年）的编码，当 i 和 j 在 Y中并列时，$y_{ij} = y_{ji} = 1$；当 $i \prec_Y j$时，$y_{ij} = 1$ 和 $y_{ji} = -1$，以及 $y_{ii} = 0$。由于 Y的得分矩阵的对角元素为零，$\sum_{i,\ j} c_{ij}y_{ij}$ 等于对不同对象的无序对 $\{i, j\}$求和所得的结果，其中 $c_{ij}y_{ij} + c_{ji}y_{ji}$在 i 和 j 于 Y中并列时等于 $c_{ij} + c_{ji}$，而在 $i \prec_Y j$ 时等于 $c_{ij} - c_{ji} = (c_{ij} + c_{ji}) - 2c_{ji}$（$j \prec_Y i$ 的情形具有对称性）。因此，对于每个排名$Y \in \mathcal{Z}^n$ 以及每个组合输入矩阵 $\mathbf{C}$,

$$\sum_{i,j=1}^{n} c_{ij}\, y_{ij} = \sum_{i<j} (c_{ij} + c_{ji}) - 2 \sum_{(i,j):\ i \prec_Y j} c_{ji}. \tag{7}$$

公式 7 右侧的第一项不依赖于 Y。因此，在 $\mathcal{Z}^n$ 的任意子集上最大化平均值 τ_X，等价于最小化该子集上的惩罚

$$P(Y) = \sum_{(i,j):\ i \prec_Y j} c_{ji} \tag{8}$$

。只有被放入不同桶中的对象对会对惩罚产生影响，且每一对的贡献恰好等于反对相反决定的共识：c_{ji} 是将 j 排在 i之前或与其并列的评审者人数，减去将 i 排在 j之前的评审者人数。并列对象对不会产生影响。公式 8 中的惩罚与 Emond 和 Mason（2002年）分支定界算法中使用的惩罚密切相关；该惩罚通过逐步考察迄今已处理的对象对中的元素 c_{ij} 和 c_{ji} 来计算。

3.2 组合输入矩阵告诉我们什么

公式 7 有三个贯穿全文的推论。
首先，一旦计算出 $\mathbf{C}$，评估候选解、比较两个解或对解进行局部更改的代价，就不再取决于评审者人数 m，也不取决于不同排名的数量 r。评委排名仅可选择用作启发式方法的初始解。

其次，每个无序对 $\{i, j\}$ 对公式 8 中的惩罚贡献三个值之一：如果 i和 j 并列，则为 0；如果 $i \prec_Y j$，则为 c_{ji}；如果 $j \prec_Y i$，则为 c_{ij}。因此，对于每个 $Y \in \mathcal{Z}^n$,

$$P(Y) \geq \sum_{i<j} \min\{0,\ c_{ij},\ c_{ji}\} = P_{LB}, \tag{9}$$

而且 P_{LB} 是中位排名惩罚的下界，因此也是其平均值 τ_X 的上界。当且仅当可以逐对作出最佳选择，且这些选择共同构成一个排名时，这个界才能达到。Akbari 和 Escobedo（2021年，2023a）也用以 Kemeny 距离表示的同一界来衡量其解的质量；我们在第 7 节中也以相同方式使用该界，并将其作为第 6 节精确算法的起始界。第三，公式 7 表明，对于单独考察的一对对象，只有当 c_{ij}和 c_{ji} 均为正数时，并列才是最佳选择；也就是说，将 i 排在 j 之前或与之并列的评审者人数，超过将 j 排在 i之前的人数，反之亦然。当评审者给出的排名不允许并列时，$c_{ij} = -c_{ji}$，并列的效果永远不会优于这对对象的最佳顺序。在这种情况下，总是存在一个不允许并列的中位排名：在中位排名的每个桶内，都可以按某种顺序排列对象，也可以按相反顺序排列；这两种顺序下，该桶内对象对对公式 7 的贡献符号相反，因此其中一种至少不劣于并列，而其他对象对的情况不变。仍可能存在带有并列的中位排名，但只有在同时存在一个同样好的排列时才会如此。

3.3 评审者未排序的对象

在许多实验中，每位评判者只能对部分对象进行排名。由于数据仅通过$\mathbf{C}$进入问题，因此如何处理未排序的对象仅取决于$\mathbf{C}$ ：无论采用何种约定，目标函数、公式 19 中的子全集以及第 5 节的算法都相同。

令 $X^{(h)}$ 为第 h个排名。 $X^{(h)}$ 中未排序的对象会被放入一个桶中，彼此并列，并置于所有已排序对象之后； $x_{ij}^{(h)}$ 表示 $X^{(h)}$的得分矩阵元素，其定义与第 2 节相同。将未排序的对象放入最后一个桶，只是一种表示 $X^{(h)}$的方式：实际纳入考量的对象对取决于所采用的约定，具体如下文所述。于是，组合输入矩阵可写为

$$c_{ij} = \sum_{h=1}^{r} w_h \, I_{ij}^{(h)} \, x_{ij}^{(h)}, \tag{10}$$

其中，如果对象对 i 和 j 计入 $X^{(h)}$，则 $I_{ij}^{(h)} = I_{ji}^{(h)}$ 等于 1；否则等于 0。对于完整数据，所有对象对都会计入，因此公式 10 与公式 4 一致：需要选择的只是哪些对象对计入其中。可以采用三种约定；具体选择取决于当前数据中缺失判断的含义，并且应与结果一并说明。

第一种约定（Emond 和 Mason，2000年、2002年；Yoo 等人，2020）将评审者未排序的对象视为缺失： $I_{ij}^{(h)} = 1$ 仅当 i 和 j 都在 $X^{(h)}$中排序时。 当未对某个对象排序意味着评审者没有对其进行评估时，这种约定是合适的；例如，用户只对自己看过的电影进行排序。

第二种约定计算所有对象对： $I_{ij}^{(h)} = 1$ 对于每个 i 和 j。 假定评审者偏好每个排序对象，而不是其未排序的任何对象，并且未排序的对象彼此并列。这是将部分排序或前-k列表视为桶排序的通常方式（Fagin 等人，2006年；Ailon，2010），适用于未对某个对象排序本身就是一种判断的情况。在这种约定下，被许多评审者排序的对象排在这些评审者未排序的所有对象之前，因此解也反映了每个对象被排序的频率。

然而，在 top-k 列表框架中，第二种约定所表达的内容超出了评审者所说的内容。受访者从七个可信来源中选出最可信的三个，这告诉我们，这三个来源各自都优于其余四个中的每一个，但并未说明这四个来源的次序：它们之间的并列是编码造成的，而非判断所得。第三种约定只采用评判者明确提供的信息： $I_{ij}^{(h)} = 0$ 仅当 i 和 j 都未在 $X^{(h)}$中排名时。这个想法可追溯至

Emond 和 Mason（2002年，第 8.1 节），他们指出，对象的两个子集可以被排在彼此之前，同时各自子集中的对象之间则保持“未排序”状态，而非并列。这也是 Critchlow（1985）方法中 top-k 列表所包含的信息；在该方法中，部分排序被视为与之相容的完整排序集合：对这些得分矩阵取平均时，评判者未排序的对象对的值为零，而其他对象对的值均为 $x_{ij}^{(h)}$，这恰好就是评判者采用第三种约定时在公式 10 中补充的信息。据我们所知，迄今为止，这种约定尚未在 Kemeny 公理框架中得到发展，也未曾与其他两种约定进行比较。

第二种约定与第三种约定的区别仅在于评审者对哪些对象对不作排名。令 c_{ij} 按照第三种约定计算，并令 u_{ij} 表示既不对 i 排名也不对 j 排名的评审者数量。按照第二种约定，组合输入矩阵的元素为 $c_{ij}+u_{ij}$，其中 $i\neq j$。由于当 Y 按顺序排列 i 和 j 时 $y_{ij}+y_{ji}=0$，而当 Y 将它们列为并列时 $y_{ij}+y_{ji}=2$，因此对于任意排名 Y，我们有

$$\sum_{i\neq j}(c_{ij}+u_{ij})\,y_{ij}=\sum_{i\neq j}c_{ij}\,y_{ij}+2\sum_{i<j,\;y_{ij}=y_{ji}=1}u_{ij}. \tag{11}$$

因此，当解中不允许出现并列时，两种约定具有相同的中位排名。否则，第二种约定会奖励那些将评审者很少进行排名的对象列为并列的解，而与评审者的偏好无关。对于中位前 k 项列表，其最后一个桶始终包含 $n-k$ 个对象，两种约定的区别仅在于哪些对象被放入该桶。

4 组约束

本节将通过桶的大小描述排序全集 $\mathcal{Z}^n$；这一描述使我们能够将一系列受约束问题定义为单一问题的特殊情形。

4.1 模式和形状

一个包含 b 个桶的排名 $Y\in\mathcal{Z}^n$ 可以写成对象的非空且互不相交子集构成的有序序列，B_1、……、B_b，这些子集的并集是所有对象构成的集合。B_1中的对象并列排在第一位，B_2 中的对象并列排在第二位，依此类推。Y 的秩向量将整数 t 赋给 B_t中的所有对象。

我们将包含桶 B_1、……、B_b的排名 $Y\in\mathcal{Z}^n$ 的模式 定义为其各桶大小按桶的顺序排列而成的向量，

$$\boldsymbol{\nu}(Y)=(|B_1|,|B_2|,\dots,|B_b|)\,, \tag{12}$$

其形状 $\lambda(Y)$则是将 $\boldsymbol{\nu}(Y)$中的元素按非递增顺序排列后得到的向量。

排名的模式是一个由 b 个正整数组成、总和为 n 的序列，即将 n 分成 b 个部分的组合；其形状是将 n分成 b 个部分的整数分拆，也就是说，将 n 写成 b 个正整数之和，而不考虑它们的顺序（Andrews，1976年；Stanley，2012）。通过将 n 个单位排成一行，并决定是否在每两个相邻单位之间的 $n-1$ 个差距处切分，可以得到 n 的一个组合：因此，n有 2^{n-1} 种组合，其中 b个部分的组合有 $\binom{n-1}{b-1}$ 种。

例如，对于 $n=10$ 个对象，排名 $\langle c,\ \{a,e\},\ \{b,d,f,g\},\ \{h,i,j\}\rangle$ 有 $b=4$ 个桶，秩向量为 (2，3，1，3，2，3，3，4，4，4)，模式为 (1，2，4，3)，形状为 (4，3，2，1)。完整排名的模式为 (1，1，…，1)；所有对象并列的排名，其模式为 (n)。我们用 $\lambda\vdash n$ 表示 λ 是 n 的整数分拆，$\ell(\lambda)$ 表示其部分数，$m_j(\lambda)$ 表示等于 j 的部分数。

对于给定的 n 的组合 $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_b)$ 以及给定的 n 的划分 λ，我们定义子全集

$$\mathcal{Z}^{n\backslash\boldsymbol{\nu}} = \{Y \in \mathcal{Z}^n : \boldsymbol{\nu}(Y) = \boldsymbol{\nu}\}, \qquad \mathcal{Z}^{n\backslash\lambda} = \{Y \in \mathcal{Z}^n : \lambda(Y) = \lambda\}. \tag{13}$$

然后，排序全集被组织成一个层级结构：

$$\mathcal{Z}^n = \bigcup_{b=1}^{n} \mathcal{Z}^{n\backslash b}, \qquad \mathcal{Z}^{n\backslash b} = \bigcup_{\lambda\vdash n,\ \ell(\lambda)=b} \mathcal{Z}^{n\backslash\lambda}, \qquad \mathcal{Z}^{n\backslash\lambda} = \bigcup_{\boldsymbol{\nu}:\ \lambda(\boldsymbol{\nu})=\lambda} \mathcal{Z}^{n\backslash\boldsymbol{\nu}}, \tag{14}$$

w这里我们还将 $\lambda(\boldsymbol{\nu})$ 写作模式 $\boldsymbol{\nu}$ 的形状，并且所有并集都是不相交的。

设 $\boldsymbol{\nu} = (\nu_1, \ldots\ldots, \nu_b)$ 是将 n 组合为 b 个部分的一种方式。具有模式 $\boldsymbol{\nu}$ 的排名由以下选择确定：先从 ν_1 个对象中选择第一个桶的对象，再从剩余对象中选择第二个桶的 ν_2 个对象，依此类推，从而

$$|\mathcal{Z}^{n\backslash\boldsymbol{\nu}}| = \binom{n}{\nu_1, \nu_2, \ldots, \nu_b} = \frac{n!}{\nu_1!\,\nu_2! \cdots \nu_b!}. \tag{15}$$

多项式系数不依赖于各部分的顺序：所有形状为 λ 的模式，其数量为 $b!/\prod_{j\geq 1} m_j(\lambda)!$，都包含相同数量的排名，并且

$$|\mathcal{Z}^{n\backslash\lambda}| = |\mathcal{Z}^{n\backslash\boldsymbol{\nu}}| \cdot \frac{b!}{\prod_{j\geq 1} m_j(\lambda)!}, \tag{16}$$

其中，$\boldsymbol{\nu}$ 是任意一个形状为 λ 的模式。对将 n 划分为 b 个部分的所有划分求和可得

$$|\mathcal{Z}^{n\backslash b}| = \sum_{\lambda\vdash n,\ \ell(\lambda)=b} |\mathcal{Z}^{n\backslash\lambda}| = b! \left\{ {n \atop b} \right\}, \tag{17}$$

其中，第二个等式是第2节中回顾的计数结果（D'Ambrosio 等人，2019年），而公式 17 对其进行了细化。补充材料中的表 A.1 展示了 $n = 5$ 和 $n = 6$ 的分解。该分解显示，当解的结构预先确定时，搜索空间会缩小多少：对于 $n = 10$ 个对象，排序全集包含 102，247，563 个排名；具有 $b = 5$ 个桶的排名有 5，103，000 个，分布在仅 7 种形状和 126 种模式中；而单一模式（1，1，2，3，3）就包含 50，400 个排名。

4.2 所有问题的单一定义

式 14 中的层级表明，本文讨论的所有问题都可以通过解所允许的模式来描述。设 $\mathcal{N}$ 是 n 的一个非空组合集合，即可接受模式，并令

$$\mathcal{Z}^{n\backslash\mathcal{N}} = \{Y \in \mathcal{Z}^n : \boldsymbol{\nu}(Y) \in \mathcal{N}\} \tag{18}$$

为模式可接受的排名集合。我们将具有可接受模式的中位排名 $\mathcal{N}$ 定义为满足以下条件的排名（或多个排名）$\hat{}Y$

$$\hat{Y} = \operatorname*{arg\,min}_{Y\in\mathcal{Z}^{n\backslash\mathcal{N}}} \sum_{h=1}^{r} w_h\, d(X^{(h)}, Y) = \operatorname*{arg\,max}_{Y\in\mathcal{Z}^{n\backslash\mathcal{N}}} \bar{\tau}_X(Y). \tag{19}$$

本文中的所有问题都是式 19 的特例，可通过采用不同的集合 $\mathcal{N}$ 得到：

- 如果 $\mathcal{N}$ 包含 n的所有 2^{n-1} 组合，那么 $\mathcal{Z}^{n\backslash\mathcal{N}} = \mathcal{Z}^n$ ，解就是公式 6 的中位排名：没有任何限制；

- 如果 $\mathcal{N}$ 包含由 b 部分组成的组合，那么 $\mathcal{Z}^{n\backslash\mathcal{N}} = \mathcal{Z}^{n\backslash b}$，解就是由 b个桶构成的MCBO，定义见 D'Ambrosio 等人（2019）；
- 如果 $\mathcal{N} = \{\boldsymbol{\nu}\}$ 只包含一个模式，那么 $\mathcal{Z}^{n\backslash\mathcal{N}} = \mathcal{Z}^{n\backslash\boldsymbol{\nu}}$：所有桶的大小都是预先规定的，例如Q方法论和绩效评估中的强制分配，或描述由 k 名成员组成的委员会选举的模式（$k,\ n-k$）；
- 如果 $\mathcal{N} = \{(1,\ \ldots,\ 1)\}$，那么 $\mathcal{Z}^{n\backslash\mathcal{N}} = \mathcal{S}^n$，解就是排列空间中的中位排名；
- 如果 $\mathcal{N} = \{(1,\ \ldots,\ 1,\ n-k)\}$，其中有 k 个元素等于 1，则解是中位前k列表（第4节.3）。

这些情形之间的是部分模式，其中只有部分桶大小是预先规定的。我们将它们写成元素为正整数或符号 $*$ 的模式，该符号表示自由大小的桶：具有 b 个元素的部分模式 $\boldsymbol{\nu} = (\nu_1,\ \ldots,\ \nu_b)$，其中 $\nu_t \in \{1, 2, \ldots, n\} \cup \{*\}$，是集合 $\mathcal{N}$ 的简写，表示将 n 个对象分成 b 个部分的组合，且这些组合在所有预先规定的元素上都与 $\boldsymbol{\nu}$ 一致；对于 $\mathcal{Z}^{n\backslash\mathcal{N}}$，我们仍记作 $\mathcal{Z}^{n\backslash\boldsymbol{\nu}}$。例如，模式 $(1,\ 1,\ *,\ *,\ *)$ 表示需要五个桶，其中前两个桶各包含一个对象；模式 $(2,\ *,\ 2)$ 表示需要三个桶，其中第一个和最后一个桶各包含两个对象。如果 $\boldsymbol{\nu}$ 有 f 个自由元素，且其预先规定元素之和为 s，那么这些自由元素将分配剩余的 $n-s$ 个对象，因此 $\mathcal{N}$ 包含 $\binom{n-s-1}{f-1}$ 种组合。不含符号 $*$ 的模式称为完整模式。

第 5 节和第 6 节中的算法无法处理任意集合 $\mathcal{N}$，而只能处理那些可以逐位置描述的集合：第 t 个桶允许的大小集合 A_t，$t = 1,\ 2,\ \ldots,$ 以及允许的桶数集合 $\mathcal{B}$。上述列出的问题都属于这一类：例如，对于所有 t 和 $\mathcal{B} = \{1, \ldots, n\}$，中位排名都有 $A_t = \{1, \ldots, n\}$；而含有 b 个桶的 MCBO 有 $A_t = \{1, \ldots, n\}$ 和 $\mathcal{B} = \{b\}$。

4.3 中位 top-k 列表

top-k 列表是一种排名，其中前 k 个对象按严格顺序排列，而其余 $n-k$ 个对象不加区分。由于解必须属于 $\mathcal{Z}^n$，且 Kemeny 距离只能通过将两个对象并列来使它们不加区分，因此 top-k 列表是一种模式为 $(1,\ \ldots,\ 1,\ n-k)$ 的桶排序，其中有 k 个元素等于 1。在这方面，解不同于输入排名（第 3.3 节）：评判者未比较的对象对可以从$\mathbf{C}$中删除，而解则必须对每一对对象都作出说明。我们用 $\mathcal{T}^{n,\ k}$ 表示 top-k 列表的子全集，其中 $1 \le k \le n-2$；根据公式 15，$|\mathcal{T}^{n,\ k}| = n!/(n-k)!$。

与 top-k 列表的聚合不同（Fagin 等人，2003，2006年；Ailon，2010年；Mathieu 和 Mauras，2020），在前者中，top-k 列表是问题的输入，而在这里它们是输出，且输入排名可以是任何类型。

对于 top-k 列表，式（7）的形式尤其简单。设 $\gamma_a = \sum_{j \ne a} c_{ja}$ 为$\mathbf{C}$的第 a 列的总和，并设 $Y \in \mathcal{T}^{n,k}$ 是一个 top-k 列表，其中前 k 个对象组成的集合 K 按 $a_1 \prec_Y a_2 \prec_Y \cdots \prec_Y a_k$ 排名。那么

$$\sum_{i,j=1}^{n} c_{ij}\, y_{ij} = \sum_{i<j} (c_{ij} + c_{ji}) - 2\sum_{a \in K} \gamma_a + 2 \sum_{1 \le s < t \le k} c_{a_s a_t}. \tag{20}$$

推导过程见补充材料。

公式 20 表明，中位数 top-k list 解出了两个嵌套问题。第一个问题是选择集合 K，其依据是一个分别取决于每个选定对象的项 $-2\gamma_a$：$\mathbf{C}$ 的第 a 列总和越大，也就是说，评审者将其他对象排在 a 之前或与其并列的次数越多，选择 a 就越不合适。最后一个桶中的对象

只通过这些列总和产生影响。第二个问题是对选定对象进行排序，这只取决于$\mathbf{C}$ 在 K 对象上的限制。

给定 K，其对象的最佳顺序是在 K 的排列空间中，对限制在 K 对应行和列上的组合输入矩阵求得的中位排名：根据式 7 应用于受限矩阵的结果，式 20 的最后一项与其平均值 τ_X 的分子仅相差一个不依赖于顺序的常数。

有两个值得注意的推论。第一，当所有评审者给出的排名都没有并列时，$c_{ja} = -c_{aj}$ 和 $\gamma_a = -\sum_{j \neq a} c_{aj}$，因此选择项是波达计分： a 赢得的两两比较总数减去其输掉的两两比较总数。相反，当评审者给出的排名存在并列时， a 和 j 之间的并列会对选择 a不利，因为这会增加 c_{ja}。第二，由于 $|c_{ij}| \leq m$，式 20 中交互项的贡献受限于 $k(k-1)m$，而选择项会随着 n 增大。因而，对于较大的 n 和较小的 k，集合K 主要由 $\mathbf{C}$ 的列总和决定，只有列总和接近阈值的对象才需要联合考察。

5 BOLT：一种作用于组合输入矩阵的启发式方法

BOLT（桶序局部搜索与最优并列划分）通过结合三种要素来搜索中位排名及其受约束版本，而这三种要素都只作用于组合输入矩阵 $\mathbf{C}$：一组根据$\mathbf{C}$（以及可选的评委排名）计算得出的初始解；一种每次移动一个对象、并可创建或移除并列的局部搜索；以及一种精确算法，它针对对象的固定顺序，找出将该顺序划分为若干桶的最佳划分。这些要素单独来看都并非新方法：插入局部搜索是线性排序问题中的经典方法（Schiavinotto 和 Stützle，2004年；Ceberio 等人，2015年），其含并列的版本已被 Cohen-Boulakia 等人（2011年）和 Lorena et al.（2022年）采用；有序序列的最优分段可追溯至 Fisher（1958年），Fagin 等人（2004年）则将其用于基于脚标距离的含并列排名聚合。BOLT 的做法是将这些要素结合起来，在$\mathbf{C}$ 上针对 Kemeny 距离进行计算，并将划分步骤扩展到第4节中的所有约束。

本节中，我们始终记作 $d_{ij} = c_{ij} - c_{ji}$ 和 $\sigma_{ij} = c_{ij} + c_{ji}$。根据公式 7，将 i 排在 j 之前的对象对对 $\sum_{i,\ j} c_{ij} y_{ij}$贡献 d_{ij} ，并列的对象对贡献 σ_{ij}。

5.1 初始解

BOLT 从以下排名开始，前五个排名通过 $\mathbf{C}$计算得出，每次迭代最多需要 $O(n^2)$ 次操作：

1. 净流（博尔达类型）排名，该排名按照矩阵$\mathbf{D} = (d_{ij})$的行和对对象进行排序，行和相等的对象并列；2. Copeland 排名，该排名按照赢得的成对多数数目减去输掉的成对多数数目对对象进行排序，即按照$\mathbf{D}$符号的行和进行排序；3. 马尔可夫链排序，该排名按照如下链的平稳分布对对象进行排序：该链从一个对象移动到多数偏好的对象，转移概率与$\mathbf{D}$的正元素成比例，阻尼因子为 0.85（通过幂迭代计算）；4. 桶式枢轴排序，通过 Gionis 等人（2006 年）的递归过程获得，其中枢轴采用确定性选择（在待拆分对象中具有最大 $\sum_j |d_{ij}|$ 的对象），对于与枢轴并列的对象采用 Aledo 等人（2017 年）针对 Kemeny 距离调整后的规则：如果 $\sigma_{pj} > |d_{pj}|$，则对象 j 与枢轴 p 并列，即该并列优于该对象对的两种排序；

5. D'Ambrosio 等人（2026 年）所实现的分支定界算法的初始解，该解按照满足 $c_{ij} > 0$ 和 $c_{ji} \neq 0$ 的对象 j 的数量对对象进行排序，数量相等的对象并列：不同于 Copeland 排名，对于并列为最佳选择的对象对，该对象对会同时计入两个对象；

6. 评审者的不同排名（如果可用）。由于每个初始解都会经过局部搜索，因此可以只保留出现频率最高的排名来限制其数量，而$\mathbf{C}$ 始终根据所有排名计算。

此外，对于这些排名中的每一个，将并列按照对象的索引打破后，下面所述的无并列局部搜索会返回一个完全排名，并将其加入初始解。这些完整排序是排列空间中的局部最优解；当中位排名中的并列较少时，它们很有用，例如评审者具有异质性时（第 7.4 节）。

5.2 考虑并列的插入局部搜索

设 Y 为一个包含 b 个桶的排名。插入移动会将对象 z 从其所在的桶中取出，然后将其重新放置：可以单独放置在两个相邻桶之间的某个差距中（或第一个桶之前，或最后一个桶之后），也可以与现有桶中的对象并列。设 $B'_1, \dots, B'_{b'}$ 为从 Y 中移除 z 后得到的排名的各个桶，其中当 z 单独位于其桶中时为 $b' = b - 1$，否则为 $b' = b$，并设

$$U_{zt} = \sum_{j \in B'_t} c_{zj}, \qquad V_{zt} = \sum_{j \in B'_t} c_{jz}, \qquad t = 1, \dots, b',$$

分别是 z 行元素之和以及 $\mathbf{C}$ 中 B'_t 对象对应的 z 列元素之和。根据公式 8，z 对位于其前面的对象，将其所在行的元素计入惩罚；对位于其后面的对象，则将其所在列的元素计入惩罚，而与 z 并列的对象不计入惩罚。若将 z 单独放在前 t 个桶之后的间隙中，则其贡献为 $\sum_{s \le t} U_{zs} + \sum_{s > t} V_{zs}$，其中 $t = 0, \dots, b'$；若与 B'_t 中的对象并列，则其贡献为 $\sum_{s < t} U_{zs} + \sum_{s > t} V_{zs}$，其中 $t = 1, \dots, b'$。借助累积和，可在 $O(n)$ 次运算中评估 z 的全部 $2b'+1$ 个位置，这与 Lorena et al.（2022年，第 IV 节）的局部搜索相同；遍历所有对象的代价为 $O(n^2)$。每次遍历中，按照遍历开始时对象的顺序依次处理对象；如果将某个对象移至其最佳位置能够改进当前解，就将其移至该位置。重复遍历，直到没有对象可以通过移动来改进解。如果对象只放在间隙中，因此不会产生并列，那么这种移动就是线性排序问题中的经典插入操作：我们将此版本称为无并列局部搜索。

5.3 固定次序的最佳划分

设$\mathbf{o} = \langle o_1, \dots, o_n \rangle$ 为一个无并列的顺序向量，其中 o_p 是处于第 p个位置的对象。如果一个排名的桶是$\mathbf{o}$中相邻对象组成的连续对象块，则称该排名与$\mathbf{o}$ 兼容；因此，与$\mathbf{o}$ 兼容且具有 b 个桶的排名，可由每个桶中最后一个对象的位置 $0 = e_0 < e_1 < \cdots < e_b = n$确定。与第4.1节中的组合一样，与$\mathbf{o}$兼容的排名共有 2^{n-1} 个，每种模式对应一个。由于每个排名都至少与一个顺序兼容，因此可以将解的搜索分为两步：搜索一个良好的顺序；以及对于给定顺序，在具有可接受模式的情况下，搜索其最佳桶划分，而无需枚举 2^{n-1} 个划分。令

$$g(s, e) = \sum_{s \le p < q \le e} c_{o_q o_p}, \qquad 1 \le s \le e \le n, \tag{21}$$

表示从第 s个位置到第 e个位置的对象对在不并列时会对惩罚产生的贡献。由于并列的对象对不产生贡献，根据公式 8，与$\mathbf{o}$ 兼容且切分点为 $e_0, \ldots, e_b$ 的排名的惩罚为

$$P(Y) = \sum_{1 \le p < q \le n} c_{o_q o_p} - \sum_{t=1}^{b} g(e_{t-1}+1, e_t), \tag{22}$$

而$\mathbf{o}$ 的最佳划分，是使其各桶的数量 g 之和最大的划分。

如第 4.2 节所述，可用集合 A_t 和 $\mathcal{B}$来描述可接受模式：对于包含 b个元素的完整模式或部分模式 $\boldsymbol{\nu}$，若第 t个桶的大小已指定，则 $A_t = \{\nu_t\}$；若其大小未指定，则$A_t = \{1, \ldots, n\}$；并且 $\mathcal{B} = \{b\}$。令 $D_t(e)$ 表示将$\mathbf{o}$ 的前 e 个对象划分为 t 个桶的所有可接受划分中，各数量 g 之和的最大值。由于各桶的数量之和具有可加性，最后一个桶从位置 s 延伸至位置 e 的最佳划分，可通过将该桶加入前 $s-1$ 个对象划分为 $t-1$ 个桶的最佳划分而得到。因此，当$D_0(0) = 0$ 且 $D_0(e) = -\infty$（ $e > 0$时，对于 $t = 1$、.....、$\max \mathcal{B}$ 以及 $e = 1$、.....、 n，

$$D_t(e) = \max_{s:\ e-s+1 \in A_t} \{D_{t-1}(s-1) + g(s, e)\}, \tag{23}$$

当不存在可行的 s 时，取 $D_t(e) = -\infty$。$\mathbf{o}$ 的最佳划分值为 $\max_{t \in \mathcal{B}} D_t(n)$；通过在每一步存储使其达到最大值的 s，即可恢复该划分；若此值为 $-\infty$，则不存在与$\mathbf{o}$ 兼容且属于 $\mathcal{Z}^n \backslash \mathcal{N}$的排名。对于固定的 e，所有值 $g(s, e)$ 都可以通过$\mathbf{C}$ 中与 o_e 对应的行，在 $O(n)$ 次操作内得到，因为 $g(s, e) = g(s, e-1) + \sum_{p=s}^{e-1} c_{o_e o_p}$。计算出所有值 g 后，只需 $O(n^2)$ 次操作；一个自由桶需要 $O(n^2)$ 次操作，而一个指定桶（对每个 e 仅允许单一 s）需要 $O(n)$ 次操作，因此递归的代价至多为 $O(\max \mathcal{B} \cdot n^2)$。

公式 23 中的递归给出了本文所有问题的划分步骤。对于中位排名，桶的数量无关紧要，可以省略索引 t，递归式变为 $D(e) = \max_s \{D(s-1) + g(s, \ldots, e)\}$，其代价为 $O(n^2)$：这就是 BOLT 的最优并列分区（ OTP ）。使用模式 $(*, \ldots, *)$，可得到与 $\mathbf{o}$ 兼容且恰好有 b 个桶的最佳排名，这是 MCBO 所需的步骤。对于完整模式，只有一个可行的划分，即在模式的累积和处切分$\mathbf{o}$；因此，解仅取决于对象的顺序。使用模式 $(1, \ldots, 1)$ 时，划分就是顺序本身。在所有情况下，该划分仅对给定顺序而言是最优的：与$\mathbf{o}$ 兼容的最佳排名通常并不是该问题的解，因此，寻找良好顺序的任务留给局部搜索。

5.4 保持可接受模式的局部搜索

对于中位排名，每种模式都是可接受的，因此可以直接使用第 5.2 节的插入局部搜索。当 $\mathcal{N}$ 不包含所有模式时，其移动可能会产生不可接受的模式，因此必须加以限制或替换。

桶的数量固定。 如果对象单独位于一个桶中，移除它会删除该桶，因此只能将它单独放回某个差距中。否则，移除它不会改变桶的数量，而且只能将它与另一个桶中的对象并列。惩罚的变化按第 5.2 节中的方法计算。

指定模式。 当桶的大小已指定时，移动单一对象会改变两个桶的大小，因此不存在可执行的插入移动。我们改用级联移动。设 Y有 B_1、.....、B_b个桶，每个桶中的对象按其索引列出；设 $z \in B_u$，并令 $t \neq u$ 为目标桶。将对象 z 移入 B_t；若 $t > u$，则将每个 B_r、 $r = u+1$、.....、t中的第一个对象移入 B_{r-1}；若 $t < u$，则将每个桶中的最后一个对象

B_r、 $r=t$、.....、 $u-1$移入 B_{r+1}。从 B_u到 B_t的每个桶都失去一个对象并获得一个对象，而其他桶保持不变，因此 Y的模式得以保留。对于每个对象，有 $b-1$ 个目标桶，因此解的邻域包含 n个（$b-1$）级联移动。令 U_{zu} 和 V_{zu} 为第 5.2 节中的和，它们是在 B_u中除 z之外的对象上计算得到的，因此 $z \in B_w$ 对惩罚的贡献为 $\sum_{u<w} U_{zu} + \sum_{u>w} V_{zu}$。利用大小为 $n \times b$的矩阵 U和 V，级联移动所导致的惩罚变化可在 O（b^2）次操作中计算；接受移动后，对于每个被移动的对象，两个矩阵可在 O（n）次操作中更新。

交换移动。第二种保持模式不变的移动是交换两个对象 $x \in B_u$和 $y \in B_t$，与 $u \neq t$，而其他对象保持不动。根据公式 8，惩罚的变化量是由 x和 y引起的变化之和，其中每项仅取决于其在 B_u 和 B_t之间各桶对应的 $\mathbf{C}$ 行和列的和，以及仅取决于 c_{xy} 和 c_{yx}的修正项。利用 U 和 V 在各桶上的累积和，每次交换只需 $O(1)$ 次运算即可评估；遍历所有对象对的代价为 $O(n^2)$。当插入移动（桶数固定）或级联移动（指定模式）无法再改进解时，就应用交换移动；交替使用这两种移动，直到没有移动能够降低惩罚。

部分模式。级联移动和交换移动会保持所有桶的当前大小，而自由桶的大小则通过公式 23 的划分步骤进行调整。当模式中的所有元素都可以自由调整时，即对于 MCBO，保持当前大小的限制过于严格，因此改用桶数固定时的移动。

5.5 完整模式算法

算法 1 概述了针对任意一组 $\mathcal{N}$可接受模式的 BOLT。从每个初始解出发，无并列局部搜索会改进对象的顺序；随后，交替执行带有可接受模式的该顺序的最佳划分（公式 23）以及第 5.4 节的局部搜索，直到惩罚不再降低。各初始解独立处理，BOLT 返回平均值最大的所有解。对于中位排名，这两个步骤分别是 OTP 和考虑并列的插入局部搜索；在首次执行 OTP 之前，也会先执行一次该局部搜索。对于排列空间中的中位排名，划分步骤为恒等排列，因此只使用无并列局部搜索，并从打破并列后的初始解开始。

算法 1BOLT

需要： 组合输入矩阵$\mathbf{C}$、评审者数量 m、可接受模式 $\mathcal{N}$（默认为所有模式）；可选：评委排名

计算初始解（第5.1节）
对每个初始解**执行**
    通过无并列局部搜索改进对象的顺序；令 $\mathbf{o}$ 为顺序向量
    结果（按对象的索引打破并列）
    重复
        $Y \leftarrow$ 具有 $\mathcal{N}$ 中某个模式的$\mathbf{o}$ 的最佳划分（公式 23）
        通过保留 $\mathcal{N}$ 中模式的局部搜索对 Y 进行改进（第 5.4 节）
        $\mathbf{o} \leftarrow$ Y 的顺序向量（桶由前到后，对象按索引排列）
    直到 惩罚不再降低
结束循环

确保： 平均值最大的解$\hat{Y} \in \mathcal{Z}^{n\backslash\mathcal{N}}$ τ_X

插入移动或交换移动的每次遍历需要 $O(n^2)$ 次操作，每次级联移动遍历需要 $O(nb^3)$ 次操作；而 OTP 需要 $O(n^2)$ 次操作，最多带约束的分区步骤需要

$O(\max \mathcal{B} \cdot n^2)$。遍历次数和交替次数没有先验上界，BOLT 的代价通过实验中的计算时间报告。BOLT 是确定性的：没有随机初始解，也没有随机选择，因此重复运行会返回相同的解。通过枚举所有可接受排名，在小规模问题上检验了特殊情况，并与所得最优解进行比较。对于 60 个数据集（对象数为 $n = 4$ 至 7，评审者数为 $m = 3$ 至 12），其中输入排名含并列、部分数据缺失对象且权重为非整数，BOLT 始终返回可接受排名；在所有 60 个无约束问题、所有 60 个排列空间问题，以及所有 220 个含 b 个桶的问题（$b = 2$、……、$n - 1$）中，均返回了最优解；在 60 个完整模式问题中有 59 个返回最优解，在 54 个部分模式问题中有 53 个，在 220 个 top-k 列表问题中有 219 个（$k = 1$、……、$n - 2$）（补充材料的表 A.2）。

5.6 中位数前k 列表的精确算法

对于中位数前k 列表，式 20 使得在许多实际情况下可以精确计算全局最优值。中位数前k 列表最大化

$$F(a_1, \ldots, a_k) = -2\sum_{s=1}^{k} \gamma_{a_s} + 2 \sum_{1 \le s < t \le k} c_{a_s a_t}. \tag{24}$$

在包含对象 a 的前k 列表中，由 a 与其他 $k-1$ 个选定对象之一构成的每一对象对 j，对式 24 的第二项的贡献要么是 $2c_{aj}$，要么是 $2c_{ja}$。令 η_a^+ 为 $\max(c_{aj}, c_{ja})$ 中最大的$k-1$ 个值之和，令 η_a^- 为 $\min(c_{aj}, c_{ja})$ 中最小的 $k-1$ 个值之和，其中 $j \neq a$：涉及 a 的 F 项至多为 $-2\gamma_a + 2\eta_a^+$，至少为 $-2\gamma_a + 2\eta_a^-$。如果$-2\gamma_a + 2\eta_a^+$小于 $-2\gamma_j + 2\eta_j^-$ 中第 k 大的值，则对象 a 不可能属于最优前k 列表：在 $-2\gamma_j + 2\eta_j^-$ 值最大的 k 个对象中，至少有一个（记为 j）不在列表中；在相同位置用 j 替换 a 会严格增大 F。通过此测试的对象即为候选项；仅针对候选项计算 η_a^+ 和 η_a^- 后重复测试，直到候选项集合不再变化。当候选项仍然很多时，第二项测试会用到算法 1 的解值 F_0：由于 F 至多为列表中对象 x 的 $-2\gamma_x + \eta_x^+$ 之和，因此只有当 $-2\gamma_a + \eta_a^+$ 加上其他候选项中 $-2\gamma_x + \eta_x^+$ 最大的$k-1$ 个值之和不小于 F_0 时，才保留对象 a。由于交互项有界（第 4.3 节），当 n 较大且 k 较小时，候选项通常只比k多几个。

给定候选项集合 S，令 $\Phi(\emptyset) = 0$，对于每个至多包含 k 个对象的子集 $R \subseteq S$，

$$\Phi(R) = \max_{a \in R} \left\{ \Phi(R \setminus \{a\}) - 2\gamma_a + 2 \sum_{j \in R \setminus \{a\}} c_{ja} \right\}, \tag{25}$$

其中 a 是由 R 中对象组成的列表的最后一个对象。对于 $|R| = k$，$\Phi(R)$ 是 R 中对象的各种排列下式 24 的最大值，而中位数 top-k 列表由包含 k 个对象且使 Φ 最大化的子集给出；通过回溯可获得共同最优解（在我们的实现中最多为 1000 个）。由于子集数量随 $2^{|S|}$ 增长，当候选项不超过 22 个时应用递归，此时解将得到认证，即它是 $\mathcal{T}^{n,\,k}$ 上式 19 的全局最优解；否则返回算法 1 的解。

6 精确解与认证

启发式方法只能通过参照来评估。本节将介绍我们用于验证 BOLT 解的精确算法；若无法求得最优解，则用它来界定这些解与最优解之间的距离。该算法比 BOLT 慢得多，其计算

时间会随着对象数量和评审者之间的分歧迅速增长：它是评估启发式方法的工具，在问题规模允许时也可用于实践。

6.1 无约束问题

我们采用 Yoo 和 Escobedo（2021）的二元线性规划 formulation。对于每一对不同对象的有序对 $(i,\ j)$，二元变量 y_{ij} 如果 i 排名领先于或与 j并列，则等于 1，否则等于 0，因此并列对应于 $y_{ij} = y_{ji} = 1$。采用这种编码时，解的得分矩阵元素为 $2y_{ij} - 1$，根据公式（5），中位排名求解

$$\max \sum_{i \neq j} c_{ij} \left(2y_{ij} - 1\right) \tag{26}$$

$$\text{s.t. } y_{ij} + y_{ji} \geq 1 \qquad i < j, \tag{27}$$

$$y_{ij} + y_{jk} - y_{ik} \leq 1 \qquad i,\ j,\ k \text{不同}, \tag{28}$$

$$y_{ij} \in \{0, 1\} \qquad i \neq j.$$

约束条件27使每一对对象都具有可比性，而约束条件28施加传递性。当要求解为完全排名时，约束条件27变为等式，$y_{ij} + y_{ji} = 1$，此时该 formulation 就是线性排序问题的经典 formulation。

该 formulation 包含 $n(n-1)$ 个变量和 $O(n^3)$ 个传递性约束，仅有几百个对象时约束数量就已经过多。因此，我们采用延迟方式添加这些约束：首先在不包含传递性约束的情况下求解问题，添加被违反的三元组约束，然后再次求解问题，直到解满足传递性。每一轮都先求解线性松弛；一旦松弛满足传递性，再求解整数问题。由于每一轮的问题只包含部分传递性约束，其解为中位排名平均值得分的上界；当该解满足传递性时，它就是中位排名。第一轮给出的正好是公式（9）的界 P_{LB}，并将 BOLT 的解作为初始解提供给求解器。Aledo 等人（2025a）采用了类似的延迟约束方案，而 Andrieu 等人（2023）使用全部传递性约束，并通过划分来减小问题规模。所有问题均由开源求解器 HiGHS（Huangfu 和 Hall，2018）在时间限制下求解：达到时间限制时，我们报告找到的最佳解和最佳界，即经认证的间隙。

6.2 桶约束

按照 Aledo 等人 (2025a) 的指派公式，通过额外的二元变量施加第4节中的约束。该公式针对最优桶序问题编写，其约束不依赖于目标函数。令 $\delta_{it} = 1$ 表示对象i位于第 t个桶中，其中 $t = 1$、.....、b，并令 $\Delta_{it} = \sum_{s \leq t} \delta_{is}$。于是

$$\sum_{t=1}^{b} \delta_{it} = 1 \qquad i = 1, \ldots, n, \tag{29}$$

$$\sum_{i=1}^{n} \delta_{it} = \nu_t \ \ (\text{prescribed}), \qquad \sum_{i=1}^{n} \delta_{it} \geq 1 \ \ (\text{free}) \qquad t = 1, \ldots, b, \tag{30}$$

$$y_{ij} + y_{ji} \geq \delta_{it} + \delta_{jt} \qquad i < j,\ t = 1, \ldots, b, \tag{31}$$

$$y_{ji} + \Delta_{it} - \Delta_{jt} \leq 1 \qquad i \neq j,\ t = 1, \ldots, b-1. \tag{32}$$

约束 31 将同一桶中的对象并列，约束 32 禁止在 i 位于前 t 个桶之一而 j 不在其中时，将 j 排在 i 前面或与其并列。加入这些约束后，传递性约束便是冗余的；只要变量 δ_{it} 取整数值，变量 y_{ij} 也会取整数值（Aledo 等人，2025a，命题2和命题3）；我们仍会以延迟方式添加这些约束，因为它们能加强

松弛。部分模式仅对规定的大小使用方程 30 中的等式，MCBO 对所有这些大小均使用不等式，而中位数前 k 名列表则是完整模式（1，……，1，$n-k$）。

严格排在对象 i之前、与其并列以及严格排在其后的对象数量，都是变量 y的线性函数：

$$\alpha_i = \sum_{j \neq i} (1 - y_{ij}), \qquad \theta_i = \sum_{j \neq i} (y_{ij} + y_{ji} - 1), \qquad \beta_i = \sum_{j \neq i} (1 - y_{ji}), \tag{33}$$

因为 $y_{ij} = 0$ 表示 j 严格排在 i之前。我们利用这些函数强化公式。给定一个完整模式 $\boldsymbol{\nu}$，令 $N_t = \nu_1 + \cdots + \nu_{t-1}$ 表示排在第 t个桶之前的对象数量。当且仅当某个对象前面有 N_t 个对象，且有 $\nu_t - 1$个对象与其并列时，该对象才位于第 t个桶中，因此可以将约束 31–32 替换为

$$\alpha_i = \sum_{t=1}^{b} N_t \, \delta_{it}, \qquad \theta_i = \sum_{t=1}^{b} (\nu_t - 1) \, \delta_{it}, \qquad i = 1, \ldots, n, \tag{34}$$

并结合方程 29–30。我们称之为位置公式；在我们的实验中，它是处理完整模式时最快的方法。对于部分模式，这些等式仅对指定桶中的对象成立；它们会变为一对不等式，而当对象位于自由桶中时，这些不等式不起约束作用：对于所有指定桶，有 θ_i ；对于位于第一个自由桶之前的指定桶，有 α_i ；对于位于最后一个自由桶之后的指定桶，有 β_i 。最后，两个割对并列情况加以界定：每个桶最多可包含 $\nu_{\max}$ 个对象，即指定桶中的最大规模或自由桶可容纳的最大规模，因此对每个对象都有 $\theta_i \leq \nu_{\max} - 1$ ；并列对象对的总数则介于可接受模式所允许的最小值和最大值之间。

我们将这些公式与 2312 个带并列排名的问题上的可接受排名枚举结果进行了核对，其中部分问题包含缺失值和权重：1892 个问题涉及 $n = 4$ 至 6 个对象以及所有完整模式和部分模式；420 个问题涉及 $n = 8$ 至 12 个对象（其中 240 个采用 MCBO 模拟研究中的完整模式和部分模式，180 个允许自由桶数）。对于指定规模的问题，我们分别在使用和不使用强化不等式的情况下求解，所得的 4444 个解均为最优解。

7 实验评估

7.1 设置

实验旨在回答三个问题：BOLT 的结果与最优解有多接近；与已发布的针对同一问题的方法相比表现如何；以及它的哪些组成部分发挥了作用。只要能够计算，第 6 节中的经认证的最优解即为参照；否则则采用经认证的界。所有计算均在配备 Intel Core i5-6500 处理器（四核，3.2 GHz）和 16 GB RAM 的台式计算机上完成，使用 R 4.6.1、我们对 BOLT 的实现，以及通过 R 软件包 highs 1.14.0.2 调用的 HiGHS 1.14.0 精确算法。启发式算法的计算时间为单进程在其他任务均空闲的计算机上运行所耗费的时间。精确算法每轮最多加入 200,000 个违反约束的三元组，并最多使用四个线程；其时间限制针对每项实验分别给出。BOLT 是确定性的，因此只运行一次；随机化方法运行 100 次，并报告最佳解。解的质量 Y通过最优解 Y^* 与该解 Y 的平均值之差 $\bar{\tau}_X(Y^*) - \bar{\tau}_X(Y) \geq 0$ 来衡量，我们称之为与最优值的差距。若最优解未知，则以经认证的界为参照计算差距，因此它是实际差距的上界。

我们实现的方法均使用 R 编写，并在C上运行，包括：BioConsert 局部搜索（Cohen-Boulakia 等人，2011年），从与 BOLT 相同的初始解开始，从而将移动和 OTP 的贡献与初始解的贡献区分开来；

Brancotte等人（2015年）提出的 NonStrictKwikSort，以及针对完全排名的 Ailon等人（2008年）提出的 KwikSort；Fagin 等人（2004年）提出的基于位置中位数的聚合方法，以及脚标距离下的最优分组。对于 SIgFUR 和 FUR（Badal 和 Das，2018年；Badal 和 Singh，2026年），我们在自己的计算机上运行了作者提供的 MATLAB 代码[1]，该代码返回了我们能够核验的所有已发布数值。对于 ACP 和 MorK，由于其代码不可用，我们将 BOLT 与已发布的结果进行了比较。

7.2 具有经认证最优解的真实数据和模拟数据

第一个基准测试包含83个问题，精确算法在600 秒的时间限制内认证了这些问题的最优解：Emond 和 Mason（2002年）的EMD数据、Badal 和 Singh（2026年）的五个数据集（机器学习算法、包含229个对象的大学及其科研和教学排名、包含240个对象的城市）、13个包含并列的完整排名的PrefLib数据集（Mattei和Walsh，2013），以及64个模拟问题，其中 n 对象数从20到90不等。表1报告了最优解数量和差距。

表1：具有经认证最优解的问题（83个）：各方法返回最优解的问题数、以 $\bar{\tau}_X$ 表示的与最优值的差距（平均值、90%分位数、最大值）以及最长运行时间。随机化方法：100次运行中的最佳结果。

方法	最优解	平均间隙	90%分位数	最大差距	最长时间（秒）
BOLT	72	7.0e-06	3.0e-05	8.3e-05	12.7
BioConsert（相同初始值）	66	2.2e-05	1.2e-04	2.1e-04	193.0
NonStrictKwikSort	17	2.5e-03	6.6e-03	8.2e-03	3.6
KwikSort	4	6.3e-02	5.4e-02	9.0e-01	3.0
Fagin	7	3.6e-02	1.0e-01	1.7e-01	0.6

BOLT 在 83 个问题中的 72 个问题上得到了最优解，在其余问题上的差距至多为 8.3×10^{-5}。与从相同解开始的 BioConsert 相比，BOLT 在 14 个问题上表现更好，在 3 个问题上表现更差：大部分收益来自初始解，但 OTP 和更多种类的移动也有所贡献。BOLT 最长耗时为 12.7 秒，耗时发生在一个 PrefLib 数据集上；该数据集只有 12 个对象，却包含数千种不同排名，且这些排名都用作初始解（第5.1节）。

7.3 已发表的基准测试

近似孔多塞划分 Akbari 和 Escobedo（2023a）报告了 85 个 Pre-fLib 数据集的结果。这些数据集包含含并列的完整排序，且对象数量超过 100 个（最多为 2，819 个）；报告内容为 ACP 与下界 P_{LB} 公式（9）之间的差距。我们核实了在全部 85 个数据集上，我们的下界都与其一致，因此这些差距具有可比性；由于他们将差距报告为两位小数，ACP 的目标值在下界的 ± 0.005%范围内是已知的。在 75 个数据集上，BOLT 优于已发布的最佳 ACP 解；在另外 10 个数据集上，按四舍五入结果，两者相同。BOLT 与 ACP 相对于下界的差距中位数分别为 0.68% 和 0.96%。在对象数量不超过 477 个的 11 个数据集中，BOLT 在其中 9 个达到了下界，因此是最优的。在另外两个数据集 ED-10-50（170 个对象）和 ED-10-49（351 个对象）上，精确算法在四小时的时间限制内，分别于 71 秒和 37 分钟找到最优解：BOLT 在不到一秒内就在第一个数据集上返回了最优解，而在第二个数据集上返回了一个比最优解的距离高 22 个点的解。ACP 的计算时间不具可比性，因为其使用的是不同的计算机和实现；BOLT 的计算时间最多为 142 秒，这一结果对应于一个包含 2，446 个对象的数据集。

SIgFUR 和 FUR 表2将 BOLT 与 Badal 和 Das（2018年）及 Badal 和 Singh（2026年）的方法在我们的计算机上的运行结果进行了比较。

[1]https://github.com/prakashvs613/RankAggSIgFUR-tied-matlab

表2：Badal 和 Singh（2026年）的数据集：解的平均值 τ_X 以及同一台机器上的运行时间。除包含 1,173 个对象的数据集外，最优解均由精确算法认证；对于该数据集，报告的是经认证的界。

数据集（$n \times m$）	最优解	方法	$\bar{\tau}_X$	时间（秒）
ML（12×8）	0.72159	FUR, SIgFUR	0.72159	2.6, 37.1
		BOLT	0.72159	0.02
大学（229×10）	0.72026	SIgFUR（无时间限制）	0.72026	8698
		BOLT	0.72025	0.56
城市（240×4）	0.74793	SIgFUR	0.74789	5151
		BOLT	0.74786	0.26
EMD（15×21）	0.16556	FUR、SIgFUR、BS	0.16556	4.8–466
		BOLT	0.16556	0.02
大学 SOI（1173×19）	≤ 0.57415	最佳 BS	0.55303	1727
		BOLT	0.56299	40

在大学和城市数据集上，最好的 SIgFUR 解略优于 BOLT，距离分别少 2 点和4点；而在科研和教学排名上，BOLT 优于 FUR。对于最大的那个数据集，精确算法因达到经认证的界而停止，BOLT 比已发表的最佳解的距离总和约低 2%。运行时间相差三到四个数量级，这也与使用的编程语言不同有关（MATLAB 与 R 加 C++）。

MorK Azzini 等人（2027年）研究了 3600 个模拟问题中排列空间的中位排名，这些问题包含 n 个对象，数量从 10 到 50 不等。由于无法获取他们的问题，我们按照相同的分布生成了 3600 个问题，并使用带有 $y_{ij} + y_{ji} = 1$的第6.1节公式精确求解：所有问题均得到求解，每个问题耗时最多 26 秒。用于完整排序的 BOLT 在 3017 个问题上返回了最优解。遵循 Azzini 等人（2027年）的做法，我们用最优解与该解得分之间的相对差异来衡量差距：在 n 的各个值上，BOLT 的平均差距为 2.66×10^{-5}，而 MorK-Heuristic 已发布的数值为 9.78×10^{-5}；在 n的 41 个值中，有 39 个值的差距更小。问题的生成方法、衡量方法以及每个 n 的结果均见补充材料。

7.4 带并列的模拟数据

设计 我们使用 Mallows 模型和 Kendall 距离（Mallows，1957），通过 Fligner 和 Verducci（1986）的表示法进行抽样，并使用两个此类模型的混合模型（离散度相同、权重相等，第二个中心随机生成），生成了 1080 个带并列的问题。对抽样排列中每对相邻对象之间的每个差距，以 p_{tie} 的概率将其闭合，从而引入并列。实验设计考察了对象数量 $n \in \{50, 100, 200\}$、评审者数量 $m \in \{10, 50\}$、离散度 $\theta \in \{0.02, 0.05, 0.2\}$（$\theta$ 越小，评审者之间的意见分歧越大）、 $p_{tie} \in \{0.1, 0.3, 0.6\}$ 以及两种模型；每个单元重复实验 10 次。我们对所有 $n \leq 100$ 的问题，以及每个单元中一次重复实验的 $n = 200$ （另加十个 $n = 200$ 问题）运行了精确算法，时间限制为 900 秒，并使用四个线程；该算法在 733 个问题中认证了带并列的最优解，在 760 个问题中认证了排列空间中的最优解。

准确率表3报告了具有经认证的最优解的问题的结果。

BOLT 是最准确的方法，其差距比枢轴方法小一到两个数量级。它并不总能找到最优解：达到最优解的比例会随评审者之间的分歧以及 n 而下降，从 $n = 50$ and$\theta = 0.2$ 时的 99% 降至 $n = 100$ 和 $\theta = 0.02$ 时的 7%；而在 $n = 200$ 和 $\theta \leq 0.05$ 时，没有任何已认证问题能找到最优解。评审者几乎完全随机时，目标函数趋于平坦，许多解的

表3：带并列且具有已认证的最优解的模拟问题：关于 $\bar{\tau}_X$ 的最优解数量及与最优值的差距。上：含并列的解（733个问题）；下：完整模式（760个问题）。对于 $n = 50$、100、200，每个问题的平均运行时间（单进程、计算机空闲）。

方法	最优解	平均差距	最大间隙	时间（秒），$n = 50/100/200$
BOLT	433	6.1e-05	9.8e-04	0.08 / 0.23 / 0.70
BioConsert（相同初始值）	275	1.3e-04	1.1e-03	2.7 / 5.7 / 14.7
NonStrictKwikSort（100 次运行）	9	6.8e-03	2.4e-02	0.20 / 0.52 / 1.65
Fagin	0	5.8e-02	2.2e-01	0.01 / 0.01 / 0.02
BOLT（完整排序）	409	8.6e-05	9.8e-04	0.03 / 0.06 / 0.19
KwikSort（100 次运行）	41	5.6e-03	2.2e-02	0.23 / 0.61 / 2.02

平均值 τ_X 在最优解的 10^{-4} 以内，且最优解中的并列很少。若初始解中不包含完整排名，BOLT 在 336 个问题中未能找到最优解，其中 331 个问题的解所含桶数少于最优解（中位数分别为 68 和 93）；两个解在约 2% 的对象对上存在差异，这些差异遍布大多数对象。我们尝试的其他局部移动（整体移动桶、拆分桶）均未能缩小差距，而初始解中包含完整排名（第5.1节）则将最优解数量从 397 提高到 433，代价是计算时间约增加 2.4 倍。

消融 表 4 报告了移除 BOLT 的某个组成部分，或从单一初始解运行 BOLT 时的结果；由于各初始解是独立处理的，每种变体都会从其中一个子集中选取最佳解。在这些数据中，评委排名是最有用的初始解，而仅使用最佳评委的排名就与综合所有评委的排名一样好；当无法获得评委排名时，则需要使用从 $\mathbf{C}$ 计算出的初始解。移除局部搜索或 OTP 会使最优解的数量减少约三分之一，而且任何单一初始解都不足以获得最佳结果。对第 7.2 节的 83 个问题进行同样的消融（补充材料中的表 B.1）也得出了相同结论；在该数据中，完整排名不会增加最优解的数量，但会将最大差距从 1.6×10^{-4} 降至 8.3×10^{-5}。

7.5 受约束问题

对于受约束问题，表 5 报告了在每种规模和约束下，对 BOLT 解进行认证的结果。采用第6.2节的公式，以 BOLT 解为初始解，在十个问题上进行测试：从两个 Mallows 模型的混合物中抽取的 $m = 100$位评审者的排名，其中 $\theta = 0.2$、 $n = 20$ 以及 50 个对象；时间限制为 600 秒，使用四个线程。考察了四种约束：含七个桶的完整模式，其大小按配置 (1，2，3，4，3，2，1) 缩放至 n；具有相同首尾大小、中央五个大小不受限制的部分模式； $b = 5$ 个桶；前5名列表。

在 $n = 20$ 中，全部40个问题都求得了最优解，BOLT在其中38个问题中返回了最优解；另外两个问题具有部分模式，其差距至多为 4.2×10^{-4}。在$n = 50$ 中，前五名列表均在约1秒内求解完成，十个完整模式中有九个在时间限制内求解完成；具有 $b = 5$ 桶的问题，尤其是具有部分模式的问题，难度要大得多，在时间限制内求解完成的仅有7个和3个。在29个具有 $n = 50$的已认证问题中，BOLT在27个问题中返回了最优解。对于未求解的问题，精确表述在四个具有部分模式的问题中改进了BOLT的解，且改进幅度始终小于 2.3×10^{-4}；对于具有 $b = 5$ 桶的问题，精确表述从未改进BOLT的解；在具有部分模式的问题中，经认证的间隙（界与BOLT解之间的差距）达到 4.8×10^{-3}，在具有 $b = 5$ 桶的问题中达到 8.3×10^{-3}。与无约束情形一样，当桶的数量或某些桶的大小可以自由确定时，精确表述较弱，剩余差距主要源于界。BOLT在每个问题上耗时约0.01

表 4：BOLT 在带并列的模拟问题上的消融：在具有经认证的最优解的 733 个问题中，最优解的数量及与最优值的差距（以 τ_X 计）；以及在 1080 个问题中，该变体劣于 BOLT 的问题数量。上半部分：移除一个组件；下半部分：使用单一初始解（及其通过无并列局部搜索得到的完全排名）。

变体	最优解（共 733 个）	mean gap	max gap	劣于 BOLT（共 1080 个）
所有组件	433	6.1e-05	9.8e-04	0
无完整模式排名	397	7.5e-05	1.1e-03	218
仅完整排序	393	7.6e-05	9.8e-04	237
仅初始解 来自C	303	1.1e-04	1.1e-03	512
仅评委的排名 评审者	405	7.8e-05	9.8e-04	177
不使用 OTP	281	1.1e-04	9.8e-04	362
仅使用 OTP（不进行局部搜索）	284	1.2e-04	1.1e-03	538
仅使用净流	153	2.9e-04	2.3e-03	856
仅使用 Copeland	199	2.1e-04	2.0e-03	766
仅马尔可夫链	162	2.7e-04	2.1e-03	848
仅桶枢轴	158	3.4e-04	4.9e-03	861
分支定界起点单独	199	2.2e-04	2.4e-03	769
最佳单独评委	405	7.8e-05	9.8e-04	177
中位数单独评委	135	2.8e-04	2.0e-03	914
仅最差评判	63	6.6e-04	4.6e-03	1001

秒。补充材料报告了一项规模更大的三类约束模拟研究，涉及最多5000个对象，并将结果与 D’Ambrosio 等人（2019）的QUICK和DECoR算法进行了比较。

表 5：BOLT 对受约束问题的认证（每个 n 和约束各十个问题）。已认证：求得最优解的问题；BOLT 最优：在已认证问题中 BOLT 返回最优解的问题；最大间隙：找到的最佳解与 BOLT 之间的最大差异；认证间隙：经认证的界与 BOLT 之间的差异（平均值和最大值）；时间：平均值。以 τ_X 表示的差距。

n	约束	已认证	BOLT 最优	max gap	mean 认证间隙	max 认证间隙	BOLT (s)	BLP (s)
20	完整模式	10	10	0	0	0	0.01	1
20	部分模式	10	8	4.2e-04	6.3e-05	4.2e-04	0.01	8
20	$b = 5$ 桶	10	10	0	0	0	0.01	10
20	前五名列表	10	10	0	0	0	0.01	0
50	完整模式	9	8	1.8e-04	1.9e-05	1.8e-04	0.02	71
50	部分模式	3	2	2.3e-04	1.7e-03	4.8e-03	0.01	545
50	$b = 5$ 桶	7	7	0	2.1e-03	8.3e-03	0.02	442
50	前五名列表	10	10	0	0	0	0.01	1

8 个应用

本节将展示本文中的受约束问题如何源于真实数据，以及对未排序对象采用何种约定会产生什么影响。

8.1 一种强制分布：Q方法论

在 Q 方法论中（Stephenson，1935；Brown，1980；Watts 和 Stenner，2012），每位参与者都会将一组陈述按照规定的每列陈述数量，从最赞同到最不赞同地排列到网格中，因此每个 Q 排序都是一种具有该网格模式的桶排序。我们分析了 Lipset 数据（Lipset，1963；Brown，1980），该数据可在 R 包 qmethod（Zabala，2014年）中获取：9 位参与者将 33 条陈述按照模式为 (2，3，4，5，5，5，4，3，2) 的网格从 $+4$ 排到 -4。由于所有输入排名都具有这一模式，共识 Q 排序也应具有该模式；它就是具有网格模式的中位排名。

表6：Lipset Q排序（$n = 33$，$m = 9$）：网格模式下的中位排名（精确解和 BOLT），以及采用 $b = 9$ 个桶的 MCBO（BOLT）。

解	桶大小	$\bar{\tau}_X$	τ_X 采用精确解
网格模式，精确	2，3，4，5，5，5，4，3，2	0.2471	1.0000
网格模式，BOLT	2, 3, 4, 5, 5, 5, 4, 3, 2	0.2458	0.8750
$b = 9$, BOLT	6, 5, 1, 2, 1, 11, 4, 2, 1	0.2534	0.7803

表6展示了一个值得运行精确算法的案例。采用完整模式时，第6.2节的位置公式在约五分钟内认证了最优解：9位参与者的共识Q排序的平均值 τ_X 等于 0。2471，而 BOLT 的最佳解在 0。25 秒内得出，数值为 0。2458，也就是说，它与评审者的距离还差 6 点。参与者之间存在相当大的分歧，而且与第 7.4 节的模拟一样，即使问题规模较小，BOLT 也可能找不到最优解。在 Q研究中，陈述通常有几十条，网格采用完整模式，因此，实际上可以认证共识Q排序，并以 BOLT 的解作为初始解。不受网格约束时，也就是说，采用 $b = 9$ 自由大小的桶，平均值 τ_X 仅略高一些（0。2534），但桶大小与网格差异很大，其中一个桶包含 11 条陈述，因此无法将该解解读为 Q排序。

标准 Q 程序通过因子数组汇总载荷落在同一因子上的 Q排序：该数组通过计算陈述标准化得分的加权平均值获得，并将结果填入网格。这是一种临时采用的博尔达类型程序，不会优化任何准则。我们使用 qmethod 提取了三个因子（主成分、方差最大旋转、自动标记）；对于每个因子，在反转负载荷 Q排序的顺序后，我们根据被标记在该因子上的 Q排序（三个因子各有三个）的网格模式计算中位排名。在这三种情况下，精确算法均认证 BOLT 返回了最优解。对于每个因子，相较于因子数组，中位排名与被标记的 Q排序具有更高的平均值τ_X（0.6187 对 0.6136，0.5896 对 0.5783，0.5391 对 0.5240）；两种解相似但并不完全相同，它们之间的 τ_X 介于 0.81 至 0.90 之间。

8.2 观测对象对上的平均值 τ_X

在以下各节的大多数数据集中，每位评判者只对部分对象进行排序，我们每次都会说明采用了第 3.3 节中的哪种约定。对于未排序对象，公式（5）中的平均值 τ_X 会将 $n(n-1)$ 个有序对分配给每位评判者，其中也包括未计入的对象对，因此对于稀疏数据，任何解的结果都接近于零。为了报告一个可解读为一致程度的数值，我们还会计算观测到的对象对上的平均值 τ_X。令 n_h 为 $X^{(h)}$ 中已排序对象的数量，令 $T = \sum_h w_h \sum_{i \neq j} I_{ij}^{(h)}$ 为公式（10）中计入的有序对数量：采用第一种约定时为 $T = \sum_h w_h\, n_h(n_h - 1)$，采用第三种约定时为$T = \sum_h w_h \left[n(n-1) - (n - n_h)(n - n_h - 1)\right]$。我们定义

$$\tau_X^{obs}(\mathbf{C}, Y) = \frac{\sum_{i,j=1}^{n} c_{ij}\, y_{ij}}{T}. \tag{35}$$

采用第一种约定时，令 $\tau_X^{(h)}(Y)$ 为 $X^{(h)}$ 与 Y 在 $X^{(h)}$ 中已排序的 n_h 个对象上的限制之间的 τ_X。由于 $\sum_{i\neq j} I_{ij}^{(h)} x_{ij}^{(h)} y_{ij} = n_h(n_h-1)\,\tau_X^{(h)}(Y)$，我们得到

$$\tau_X^{obs}(\mathbf{C}, Y) = \frac{\sum_h w_h\, n_h(n_h-1)\,\tau_X^{(h)}(Y)}{\sum_h w_h\, n_h(n_h-1)},$$

也就是说，τ_X^{obs}是各评审者在其各自排序过的对象上计算所得 τ_X 的加权平均值，权重等于观察到的对象对数量；只有当解在每个观察到的对象对上都与每位评审者的判断一致时，其值才等于 1。τ_X^{obs}并非新的目标函数：T不依赖于 Y，因此无论采用何种约束，平均 τ_X 和τ_X^{obs}都有相同的解；而对 $\tau_X^{(h)}$ 取简单平均（Y）则会得到不同的目标函数。在完整数据和第二种约定下，$T = m\,n\,(n-1)$ 与τ_X^{obs}等于平均 τ_X。

8.3 中位数前k 名列表

在本节中，我们展示中位数前k 名列表的三个应用，其中对象数量从 7 到 613 不等。

欧洲晴雨表：对信息来源的信任在欧洲晴雨表55.2调查（欧盟委员会，2012）中，12，765 位受访者按信任程度排序，从 $n = 7$ 个来源中选出他们最信任的三个来源，以解释重大灾害发生的原因：科学家、记者、消费者组织、环境保护协会、政府代表、医生和公司。不选择某个来源，是对所选来源的一种判断，但并未说明未选出的四个来源之间的排序；因此我们采用了第 3.3 节中的第三种约定。已认证的中位数前三列表为：科学家、环境保护协会和医生（$\tau_X^{obs} = 0.4372$），与这类问题通常采用的汇总结果（首次提及、总提及次数、得分 3、2、1）一致。当解可能将未选来源并列时，约定就会产生影响。采用第三种约定时，含有 $b = 3$个桶的 MCBO 为：（科学家、环境保护协会、医生）、（记者、消费者组织、政府代表）、公司；这使公司成为最不受信任的信息来源。采用第二种约定时，较少被选中的四个来源会被放在同一个最后一个桶中，正如公式11所预测的那样。问题、样本和完整结果见补充材料。

欧洲歌唱大赛 自2016年起，在欧洲歌唱大赛中，每个参赛国都设有一个由五名成员组成的评审团，每位评审员都会为一场比赛中的所有歌曲排名，但不包括其本国的歌曲，因为该歌曲不能被投票。五位评审员的排名会合并为国家评审团的排名，该排名中前十首歌曲分别获得12、10、8、7.....1分；各国得分之和构成该场比赛的评审团排名。我们使用了MIRoVision数据集（Burgoyne等，2023）中各位评审员的个人排名，涵盖2016年至2022年的18场比赛（决赛和半决赛；2020年未举办比赛），其中 n 歌曲数量为16至26首， m评审员人数为75至215人。评审员本国的歌曲属于缺失对象，我们采用了第 3.3 节中的第一种约定：将其放在最后位置，就意味着评审员认为它是最差的歌曲。在2019年决赛以及2022年决赛和第二场半决赛中，部分国家评审团被取消资格，并由汇总结果取代，因此无法获得其排名：在比较时，我们使用了那些排名包含在数据中的评审团所给出的分数。

对于每场比赛，我们计算了所有评审员的中位数前十列表，这是每个评审团为获得得分的十首歌曲提供的自然对应结果，并将其与评审团得分的前十列表进行了比较。表7显示，中位数前十列表在所有这些

场比赛中均已认证，候选项数量为15至19个，且用时最多为0.1秒。这两个列表共有8至10首歌曲相同，并且在18场比赛中的16场由同一获胜者领衔；在2016年半决赛第一场和2019年半决赛第二场，中位数前十列表则由另一首歌曲领衔。根据构造方式，中位数前十列表与评审员的一致性高于评审团得分列表；后者是一个博尔达类型的两阶段程序的结果：τ_X^{obs}在各场比赛中的取值范围为0.19至0.41，并且比评审团得分列表的对应值高出0.0006至0.018。在2019年决赛中，中位数前十列表的获胜者（瑞典）也是数据中评审团得分的获胜者，而官方评审团排名的获胜者则是北马其顿；该排名包含了替代一个被取消评审团的汇总结果。

表7：欧洲歌唱大赛 2016年–2022年：所有已认证评审员的中位数前十列表，与评审团得分前10名列表的比较（精确算法的候选项、两个列表中的共同歌曲、获胜者）

					获胜者	
Show	n	m	候选项	共同项	中位数	Jury
2016年决赛	26	210	17	8	AU	AU
2016年第一场半决赛	18	104	18	9	RU	MT
2016年第二场半决赛	18	105	18	10	AU	AU
2017年决赛	26	210	18	9	PT	PT
2017年第一场半决赛	18	105	18	8	PT	PT
2017年第二场半决赛	18	105	18	9	BG	BG
2018年决赛	26	215	19	8	AT	AT
2018年半决赛1	19	110	18	9	IL	IL
2018年半决赛2	18	105	18	10	SE	SE
2019年决赛	26	200	15	10	SE	SE
2019年半决赛1	17	100	17	9	CZ	CZ
2019年半决赛2	18	105	18	10	SE	MK
2021年决赛	26	195	18	8	CH	CH
2021年第一场半决赛	16	95	16	9	MT	MT
2021年第二场半决赛	17	100	17	9	CH	CH
2022年决赛	25	170	19	9	GB	GB
2022年半决赛1	17	95	17	9	GR	GR
2022年半决赛2	18	75	18	9	SE	SE

AAMAS竞标数据AAMAS 2015竞标数据可在 PrefLib存储库（Mattei和Walsh，2013年）中获取，包含 $m = 201$ 位评审者对 $n = 613$ 篇论文的竞标。每位评审者都将每篇论文归入以下四个类别之一：是、可能、未作答和否。没有缺失对象。每位评审者将大多数论文归入同一类别（未作答），因此两篇论文之间最常见的关系是并列，而且组合输入矩阵的每个元素都是正数：c_{ij} 介于111和201之间，其中 $m = 201$。由于 y_{ij} 只能是1或 -1，平均值 τ_X 的每一项 $c_{ij}y_{ij}$ 都由 $y_{ij} = 1$最大化，而当所有论文并列时，这些项会同时达到最大值。因此，在不施加约束时，该解是唯一的中位排名。此时，只有对解施加约束，才能得到有信息量的共识；中位前k 名列表则挑出了评审者最青睐的论文。表8显示，中位前5名和前10名列表均已认证，且计算耗时不到0.1秒；在 $k = 20$ 和 $k = 50$ 的情况下，候选项分别为23个和55个，解无法获得认证，而BOLT的解分别耗时约0.6秒和6秒计算得出。该表还报告了桶数与类别数相同的MCBO结果：BOLT耗时0.3秒，而QUICK耗时114秒（D'Ambrosio 等人，2019年），并找到了略优的解（平均值 τ_X 为0.862295，而非0.862293）。

表8：AAMAS 2015 竞标数据（$n = 613$ 篇论文，$m = 201$ 位评审者）：采用 $b = 4$ 个桶和中位数 top-k 列表的 MCBO。

问题	桶	$\bar{\tau}_X$	已认证	候选项	时间（秒）
$b = 4$ 桶 (BOLT)	4	0.862293	–	–	0.34
$b = 4$ 桶 (QUICK)	4	0.862295	–	–	114.38
前5名	6	0.857560	yes	5	0.06
前10名	11	0.845570	yes	10	0.03
前20名	21	0.821302	no	23	0.62
前50名	51	0.748848	no	55	5.60

8.4 大型且稀疏的数据集

本节考察两个评判者数量众多的数据集，其中评判者通过评分表达偏好。采用包含 L 个等级的量表进行评分，可为评判者评分的对象生成最多包含 L 个桶的桶排序；对于未评分的对象，我们报告采用第 3.3 节中的第一种和第二种约定所得的结果。

带评分的寿司数据 D'Ambrosio 等人（2019）分析了寿司数据的一个版本（Kamishima，2003）：5000 名受访者各自为 100 种寿司中的 10 种打分，分数为 0 到 4；他们使用 DECoR 计算了具有 $b = 5$ 个桶的 MCBO，耗时 674 秒，平均值 τ_X 等于 0.0032。表9显示，将未评分的寿司视为缺失对象（如 D'Ambrosio 等人（2019）所做的那样）时，BOLT 能在 0.03 秒内找到更好的解（$\bar{\tau}_X = 0.0033517$，即 $\tau_X^{obs} = 0.3687$，而非 0.3522）；而 QUICK 在 0.9 秒内找到的解，其平均值略小 τ_X。将未评分的寿司并列置于最后位置（第二种约定）时，解的各桶大小为 (1，1，1，1，96)：由受访者人数最多的四种寿司各自占一桶，其余寿司归入最后一桶。

表9：带评分的寿司数据（每位受访者从100分中为10种寿司评分），将未评分的寿司视为缺失对象：MCBO，包含 $b = 5$ 个桶。

解	桶大小	$\bar{\tau}_X$	τ_X^{obs}	时间（秒）
已发布（DECoR）	1, 44, 9, 40, 6	0.0032017	0.3522	674.00
BOLT	44, 53, 1, 1, 1	0.0033517	0.3687	0.03
QUICK	42, 1, 55, 1, 1	0.0033516	0.3687	0.91

MovieLens数据 MovieLens 1M数据集（Harper和Konstan，2015年）包含1，000，209条评分，评分范围为1到5星级，涵盖3706部电影，由6040名用户给出，其中每名用户至少评价了20部电影。我们计算了中位数前十列表，以及使用 $b = 5$ 个桶（数量最多与评分等级相同）对评分最多的 N 部电影进行的MCBO；电影数量分别为 $N = 500$、1000和全部3706部，并采用第 3.3 节的第一种和第二种约定。每名用户都会对其评价过的电影给出一个桶排序。数据非常稀疏。在 $N = 500$的情况下，可观察到17.6%的评分，并且在99.5%的情况下，每一对电影都至少由100名用户评分。在 $N = 1000$的情况下，可观察到的评分为12.4%，且74%的对象对至少由100名用户评分。对于全部电影，仅观察到4.5%的评分和0.5%的有序对；18%的电影对象对从未由同一用户同时评分，而同时评价一对电影的用户数中位数为6。组合输入矩阵的计算时间分别为1.5秒、2.2秒和10秒。

表10展示了结果。将未评分的电影视为缺失对象时，对于 N的所有取值，包含10至15个候选项的中位数前十列表均已认证，耗时最多1。5秒。对于所有电影，包含 $b = 5$ 个桶的MCBO耗时79秒。平均值 τ_X 的范围为0。0014至0。0214，而 τ_X^{obs}的范围为0。30至0。42。将未评分的电影在最后

位置并列时，各个解差异很大：包含 $b = 5$ 个桶的解由四部单一电影开头，后面是所有其他电影。平均值 τ_X 很大（从0。72到0。92），因为对于大多数用户而言，大多数电影对象对处于并列状态，这是由于这两部电影都未被评分。

表10：MovieLens 1M 中评分次数最多的 N 部电影：中位数前十列表和包含 $b = 5$ 个桶的 MCBO；未评分的电影视为缺失对象，并在最后位置并列（b：解的桶数）。

		未评分：缺失				未评分：最后位置并列		
N	解	b	$\bar{\tau}_X$	τ_X^{obs}	时间（秒）	b	$\bar{\tau}_X$	时间（秒）
500	前10名	11	0.01877	0.3282	0.11	11	0.7168	0.02
500	$b = 5$	5	0.02141	0.3745	0.38	5	0.7228	0.26
1,000	前10名	11	0.00993	0.3168	0.07	11	0.7903	0.08
1,000	$b = 5$	5	0.01228	0.3919	2.47	5	0.7929	1.48
3,706	前10名	11	0.00142	0.3021	1.45	11	0.9162	1.18
3,706	$b = 5$	5	0.00195	0.4150	78.60	5	0.9167	23.79

表11比较了所有电影的两个中位数前十列表。将未评分的电影视为缺失对象时，列表包含既评分很高、又有许多用户评分的电影（《教父》、《肖申克的救赎》、《辛德勒的名单》、《星球大战》、《非常嫌疑犯》等），使用 $N = 1000$ 时以及涵盖所有电影时，结果也相同。将未评分的电影与最后位置并列时，列表包含评分次数最多的十部电影（其中九部对应 $N = 500$ 和 $N = 1000$），并且与第一个列表有3到5部电影重合。同样，第一种约定总结了每位用户看过的电影之间所表达的偏好，而第二种约定主要奖励被许多用户观看过的电影。作为比较，平均评分最高的十部电影评分次数都很少，而且没有一部出现在所有电影的两个中位数前十列表中：采用第一种约定时，每一对电影的计数次数等于同时给这两部电影评分的用户数，因此，只有少数热情用户评分的电影无法进入中位数前十列表。

表11：MovieLens 1M，全部3,706部电影：将未评分的电影视为缺失对象，并在最后位置并列时的中位数前十名列表

	未评分视为缺失	未评分并列最后
1	《教父》（1972）	《美国丽人》（1999）
2	《肖申克的救赎》（1994）	《星球大战4：新希望》（1977）
3	《辛德勒的名单》（1993）	《星球大战V》（1980）
4	《星球大战4：新希望》（1977）	《拯救大兵瑞恩》（1998）
5	《非常嫌疑犯》（1995）	《沉默的羔羊》（1991）
6	《夺宝奇兵》（1981）	《黑客帝国》（1999）
7	《沉默的羔羊》（1991）	《星球大战VI》（1983）
8	《美国丽人》（1999）	《终结者2：审判日》（1991）
9	《第六感》（1999）	《回到未来》（1985）
10	《星球大战V》（1980）	《侏罗纪公园》（1993）

9 总结性评论

本文将中位排名及其受约束版本视为同一个问题：在具有可接受模式的排名中，寻找使平均值 τ_X 最大化的排名，并以组合输入矩阵表示。对于每一组可接受模式，均适用相同的启发式方法 BOLT 和相同的精确表述；只有划分步骤和局部移动会随约束而变化。

实验表明，BOLT 的准确性较高，但也存在一些值得说明的局限。在具有经认证的最优解的 83 个真实和模拟问题中，BOLT 返回了最优解的案例数为

72 个，与最优解的最大差距为 8.3×10^{-5}，按平均 τ_X 计算。在 Akbari 和 Escobedo（2023a）的基准测试中，对于最多含 2,819 个对象的问题，BOLT 在 85 个数据集中有 75 个优于已发表的 ACP 解，其余数据集经四舍五入后结果相同。在 Badal 和 Singh（2026）的数据中，对于最多含 240 个对象的数据集，BOLT 的距离与 SIgFUR 相差仅数分或表现更好，耗时不到一秒，而 SIgFUR 则需数分钟或数小时；在最大的数据集中，BOLT 优于已发表的最佳解。然而，当评审者几乎随机时，目标函数趋于平坦，难以找到最优解：在带并列的模拟问题中，BOLT 在 733 个已认证案例中有 433 个找到了最优解；当对象数达到 100 个或更多时，这一比例降至 10% 以下，且 $\theta = 0.02$，不过平均差距仍在 6×10^{-5} 左右。在这些问题中，最优解中的并列很少，而加入初始解的完整排名可防止 BOLT 停在并列过多的局部最优解上；消融实验还表明，局部搜索和最优并列分区都很重要，而且仅靠任何一个初始解都不够。在有约束的情况下，精确表述为所有含 20 个对象的问题都提供了认证；BOLT 在其中 40 个案例中的 38 个返回了最优解，另外两个案例均有部分模式。对于含 50 个对象的问题，精确表述在十分钟内认证了所有前五名列表和十个完整模式中的九个，但对于带有 $b = 5$ 个桶的问题只认证了 10 个中的 7 个，对于带部分模式的问题只认证了 10 个中的 3 个；在 29 个已认证问题中，BOLT 有 27 个返回了最优解，其余问题中精确表述将解改进了不到 2.3×10^{-4}。

这些结果表明，这两种工具的用法很简单。对于任何规模的问题，BOLT 都是首选方法：一旦计算出组合输入矩阵，其计算时间就不受评审者数量的影响，除非将评委排名用作初始解；当评审者很多时，可以限制这类排名的数量。对于规模足够小的问题，精确公式可用于认证解，或为其与最优解之间的距离界定上限：无约束时，对象数量最多为几百个；采用完整模式时，对象数量最多约为五十个，具体取决于评审者之间意见分歧的程度。在这个范围内，实际应用中也值得运行该方法：在第 8.1 节的 Q排序中，对于 33 条陈述和九位意见分歧相当大的参与者，精确公式在几分钟内就认证出了一个 BOLT 未能找到的共识。

仍有若干问题尚待解决。当桶的数量不固定时，精确公式的效果较弱；更强的公式或分解方法（Yoo 和 Escobedo，2021年；Akbari 和 Escobedo，2023a；Kawashima 和 Sukegawa，2026年）或可拓展受约束问题的认证。如何处理评审者未排序的对象，是组合输入矩阵上的一种选择；第 3.3 节中的约定已在第 8 节的应用中产生了不同的解，值得单独比较（Andrieu 等，2023）。最后，当评审者之间存在异质性时，单一共识可能无法很好地概括他们的偏好，而对评审者群体分别求中位数排序，则是本研究的一种自然延伸。

补充材料

补充材料包含 BOLT 各组件的推导、检查和伪代码（附录A），以及复现实验所需的信息、与 MorK 的比较详情、欧洲晴雨表数据的详情，以及实验的完整表格（附录B）。

软件和数据可用性

BOLT 的代码、用于计算中位数前k 列表的精确算法、用于计算中位数约束桶排序的算法，以及复现所有实验的脚本、实验结果和计算环境说明，可向作者索取。

A 方法

A.1 子全集的基数

表 A.1 展示了论文第4.1节中针对$n = 5$ 和 $n = 6$个对象的排序全集分解。对于 $n = 5$，最后一列复现了 D'Ambrosio 等人（2019）表 1 中对应的行。

表 A.1：包含 $n = 5$ 和 $n = 6$ 个对象的排名子全集的基数，按桶数 b、形状 λ 和模式分类。对于每种形状，表中报告模式数量（npat）、每种模式对应的排名数量（$|\mathcal{Z}^{n\backslash\boldsymbol{\nu}}|$），以及具有该形状的排名数量（$|\mathcal{Z}^{n\backslash\lambda}|$）。最后一列报告 $|\mathcal{Z}^{n\backslash b}| = b!\left\{{n \atop b}\right\}$。

n	b	形状 λ	npat	$\lvert\mathcal{Z}^{n\backslash\boldsymbol{\nu}}\rvert$	$\lvert\mathcal{Z}^{n\backslash\lambda}\rvert$	$\lvert\mathcal{Z}^{n\backslash b}\rvert$
5	1	(5)	1	1	1	1
	2	(4, 1)	2	5	10	30
		(3, 2)	2	10	20	
	3	(3, 1, 1)	3	20	60	150
		(2, 2, 1)	3	30	90	
	4	(2,1,1,1)	4	60	240	240
	5	(1, 1, 1, 1, 1)	1	120	120	120
	总计：$\lvert\mathcal{Z}^5\rvert$					541
6	1	(6)	1	1	1	1
	2	(5, 1)	2	6	12	62
		(4, 2)	2	15	30	
		(3,3)	1	20	20	
	3	(4, 1, 1)	3	30	90	540
		(3,2,1)	6	60	360	
		(2,2,2)	1	90	90	
	4	(3,1,1,1)	4	120	480	1,560
		(2,2,1,1)	6	180	1,080	
	5	(2, 1, 1, 1, 1)	5	360	1,800	1,800
	6	(1, 1, 1, 1, 1, 1)	1	720	720	720
	总计：$\lvert\mathcal{Z}^6\rvert$					4,683

A.2 对分区步骤的两项检查

我们将公式 23 的递归与固定阶数下所有 2^{n-1} 划分的枚举结果进行了比对，测试了 200 个随机问题，每个问题包含 $n = 4$ 至 8 个对象，且评审者允许并列。当所有模式均可接受时，递归算法和我们实现中的最优并列分区在所有问题上都得到了最优解；对于随机生成的b，最多使用 b 个桶的递归算法也得到了相同结果。我们还检验了第 3.2 节的第三个推论：对于不允许并列的评审者，在每个 $n = 3$，4，5 下，对 300 个随机问题进行 $\mathcal{Z}^n$ 枚举后发现，其中 900 个问题里有 353 个存在含并列的排名，但从未出现只有含并列的排名的情况，因此总是存在一个不允许并列的中位排名。

A.3 对 BOLT 特殊情况的检查

表 A.2 详细说明了论文第 5.5 节所总结的特殊情况检查。我们生成了 60 个数据集，其中 n 取自 $\{4, \ldots, 7\}$，m 取自 $\{3, \ldots, 12\}$：每位评审者以随机顺序对对象进行排名，并以 0.6 的概率将相邻对象并列；在三分之一的数据集中，每位评审者有一个缺失对象；在二分之一的数据集中，评审者的权重取自 $[0.5, 3]$且为非整数。完整模式从包含 2 到 $n - 1$ 个桶的模式中随机抽取；当模式至少包含三个桶时，部分模式则通过将第二个大小留空而从完整模式获得

。对于每个数据集和每组可接受排名，我们将 BOLT 的解与通过枚举所有可接受排名得到的最优解进行了比较。BOLT 始终返回一个可接受排序，但在三个问题中未找到最优解；这些问题来自两个带有权重和 $n = 6$，7 的数据集：一个前3名列表，以及同一数据集的完整模式和部分模式。

表 A.2：BOLT 与可接受排名的枚举结果对比（60 个数据集，$n = 4$ 至 7）：问题数量、BOLT 的解可接受的问题数量，以及 BOLT 的解为最优解的问题数量。

可接受排名	问题	可接受	最优
无约束（$\mathcal{Z}^n$）	60	60	60
排列（$\mathcal{S}^n$）	60	60	60
b 桶 , $b = 2, \ldots, n-1$	220	220	220
完整模式（随机）	60	60	59
部分模式（一个自由大小）	54	54	53
前k列表，$k = 1, \ldots, n-2$	220	220	219

A.4 前 k 名列表目标函数的推导

设 $Y \in \mathcal{T}^{n,\,k}$ 为一个前 k 名列表，其中前 k 个对象的集合 K 排序为 $a_1 \prec_Y a_2 \prec_Y \cdots \prec_Y a_k$，并设 $\gamma_a = \sum_{j \neq a} c_{ja}$。根据论文中的式 7，若 Y 的惩罚为 $P(Y) = \sum_{a \in K} \gamma_a - \sum_{s<t} c_{a_s a_t}$，则式 20 成立。$Y$ 的有序对是满足 $s < t$ 的对象对 $(a_s,\ a_t)$，它们贡献 $c_{a_t a_s}$；以及对象对 $(a,\ j)$，其中 $a \in K$ 和 j 位于最后一个桶中，它们贡献 c_{ja}。对于每个 $a \in K$，$\sum_{j \notin K} c_{ja} = \gamma_a - \sum_{a' \in K,\ a' \neq a} c_{a'a}$，因此

$$P(Y) = \sum_{s<t} c_{a_t a_s} + \sum_{a \in K} \gamma_a - \sum_{a \neq a' \in K} c_{a'a} = \sum_{a \in K} \gamma_a - \sum_{s<t} c_{a_s a_t},$$

因为对不同对象的有序对求和后，K 为 $\sum_{s<t} (c_{a_s a_t} + c_{a_t a_s})$。

A.5 BOLT 各组件的伪代码

下面的算法给出了论文第 5.2–5.6 节所述各组件的详细内容，与我们代码中的实现一致。在所有算法中，仅当移动或选择能使当前值改善超过 $10^{-10} \max\{1, \sum_{i,j} |c_{ij}|\}$ 的容差时，才会执行该移动或选择；这样可避免舍入误差导致循环。每个桶中的对象按索引列出；每次遍历开始时，排名中的对象按其当时的顺序访问，并以索引打破并列。

对于桶数固定的情况（第 5.4 节），使用相同的算法，但有一处改动：单独位于一个桶中的对象只能放入差距中，其他对象则只能放入现有桶中，从而使桶数保持不变。

对于中位排名，任意大小都可接受，可以省略索引 t，算法便成为最优并列分区（OTP），其 $D(e) = \max_s \{D(s-1) + g(s,\ e)\}$。

桶的数量固定时，Cascade 会替换为桶数量固定的插入局部搜索（算法 S.1），并以同样的方式与 Exchange 交替进行。

算法 S.1 考虑并列的插入局部搜索（第 5.2 节）

要求： 排名 Y、组合输入矩阵 $\mathbf{C}$、标志 并列 （无并列局部搜索时为 false）

重复
  改进 $\leftarrow$false
  对于 每个对象 z，按遍历开始时 Y 的顺序**执行**
    从 Y中移除 z；令 $B'_1, \ldots, B'_{b'}$ 为剩余的桶
    $U_{zt} \leftarrow \sum_{j \in B'_t} c_{zj}$ 和 $V_{zt} \leftarrow \sum_{j \in B'_t} c_{jz}$, $t = 1$，……，b'，以及它们的累积和
    $P_0 \leftarrow$ 在当前位置对惩罚的贡献
    $P^* \leftarrow$ 最小值$_{t=0,\ldots,b'}\{\sum_{s \le t} U_{zs} + \sum_{s>t} V_{zs}\}$ ▷ z 单独位于一个差距中
    如果并列**则**
      $P^{*}_{\text{tie}} \leftarrow$ 最小值$_{t=1,\ldots,b'}\{\sum_{s<t} U_{zs} + \sum_{s>t} V_{zs}\}$ ▷ z 与 B'_t并列
      如果 $P^{*}_{\text{并列}} < P^*$，最佳位置为并列位置，并且 $P^* \leftarrow P^{*}_{\text{并列}}$
    结束如果
    如果 $P^* < P_0$ **则**
      将 z 放在最佳位置；已改进 $\leftarrow$ 为真
    else
      将 z 放回其原位置
    结束 if
  结束 for
直到不再改进

确保： 一个局部最优解 Y

算法 S.2 固定顺序的最优划分（第 5.3 节）

要求： 顺序 $\mathbf{o} = \langle o_1, \ldots, o_n \rangle$，$\mathbf{C}$，允许大小集合 A_t 和允许桶数集合 $\mathcal{B}$

桶
$D_0(0) \leftarrow 0$；$D_t(e) \leftarrow -\infty$ 适用于所有其他 t 和 e
对于 $e = 1, \ldots, n$ **do**
  $\sigma \leftarrow 0$
  对于 $s = e, e-1, \ldots, 1$ **执行** ▷ 操作中的所有值 $g(s, e)$ $O(n)$
    $g(s, e) \leftarrow g(s, e-1) + \sigma$，其中 $g(e, e-1) = 0$
    $\sigma \leftarrow \sigma + c_{o_e o_{s-1}}$
  结束循环
  对于 $t = 1, \ldots, \max \mathcal{B}$ **执行**
    $D_t(e) \leftarrow$ 最大值$\{D_{t-1}(s-1) + g(s, e)\colon \ e - s + 1 \in A_t\}$，且 $S_t(e) \leftarrow$ 使 $s \triangleright -\infty$ 最大的 if no s 是可采纳的
  结束循环
结束循环
$t^* \leftarrow$ 取最大值的参数$_{t \in \mathcal{B}} D_t(n)$
如果 $D_{t^*}(n) = -\infty$ **那么**
  返回 不存在与$\mathbf{o}$ 兼容的排名
结束 if
$e \leftarrow n$
对 $t = t^*, t^*-1, \ldots, 1$ **执行** ▷ 回溯
  第 t 个桶包含位于位置 $S_t(e) \ldots, e$ 的对象；$e \leftarrow S_t(e) - 1$
结束循环

确保： 与 $\mathbf{o}$ 兼容且惩罚最小的可接受排序

算法 S.3 保持模式的局部搜索（第 5.4 节）

要求 B_b，记在桶 $U_{zu} \leftarrow \sum_{j \in B_u} c_{zj}$ 的排名 $V_{zu} \leftarrow \sum_{j \in B_u} c_{jz}$，对每个对象和桶：重复 3：级联；Exchange 4：直到 Exchange 未应用任何移动 保证：具有相同模式的局部最优解 Y 5：过程 Cascade 6：重复 7：按遍历开始时 Y 的顺序，对每个对象 z 执行；令 $z \in B_u$ 8：对每个桶 $t \neq u$ 执行 9：$M \leftarrow \{z:\ u \to t\}$，加上：如果 $t > u$，B_r 的第一个对象，针对 $r = u+1, \ldots, t$；如果 $t < u$，B_r 的最后一个对象，针对 $r = t, \ldots, u-1$ 10：$\Delta_t \leftarrow$ 惩罚的变化：对 M 中的每个对象，计算其贡献相对于不移动对象的变化，依据其累积和 U 和 V；再加上 M ▷ O 对象对的变化（b^2） 11：结束循环 12：如果 $\min_t \Delta_t < 0$ 则 13：应用具有最小 Δ_t 的移动；更新 U 和 V（对每个移动的对象执行 $O(n)$） 14：结束 if 15：结束循环 16：直到本次遍历未应用任何级联移动 17：结束过程 18：过程 Exchange 19：重复 20：按遍历开始时 Y 的顺序，对每个对象 x 执行 21：对每个与 x 位于不同桶中的对象 y 执行 22 $s < t$ $p \in B_s$ $q \in B_t$：令为这两个对象所在的桶，并且：(,) (,), 23 $\Delta_y \leftarrow F_{p\ s\ t} - F_{q\ s\ t} + c_{pq} + c_{qp}$ $F_{z\ s\ t}$ 其中 (,) $= \sum_{s \le w < t} U_{zw} - \sum_{s < w \le t} V_{zw}$ ▷ $O(1)$ 24：结束循环 25：如果 $\min_y \Delta_y < 0$ 则 26：将 x 与具有最小 Δ_y 的对象 y 交换；更新 U、V 及其累积和 27：结束 if 28：结束循环 29：直到本次遍历未应用任何交换 30：结束过程

算法 S.4 精确中位数 top-k 列表（第 5.6 节）

要求： $\mathbf{C}$，k候选项的最大数量 $n_{\max}$（默认值为 22）

$\gamma_a \leftarrow \sum_{j \neq a} c_{ja}$ 对于每个对象 a

$S \leftarrow$候选项（$-\infty$）

如果 $|S| > n_{\max}$，**则**

  $Y_0 \leftarrow$ BOLT 的解，模式为 (1、.....、 1，$n-k$)；$F_0 \leftarrow$ 其公式 24 的值

  $S \leftarrow$候选项(F_0)

  如果 $|S| > n_{\max}$，**则**

    返回 Y_0，尚未认证

  结束如果

结束如果

$\Phi(\emptyset) \leftarrow 0$

对于每个子集 $R \subset S$（含有 $1 \leq |R| \leq k$），按其二进制编码递增的顺序**执行**

  $\Phi(R) \leftarrow$ 最大值$_{a \in R}\{\Phi(R \setminus \{a\}) - 2\gamma_a + 2\sum_{j \in R \setminus \{a\}} c_{ja}\}$

结束循环

$F^* \leftarrow$ 最大值$\{\Phi(R)$： $|R| = k\}$

返回值为 F^* 的前 k 个列表，通过回溯（最多 1000 次）获得，已认证

过程候选项(F_0)

  $S \leftarrow$所有对象

  重复

    对于 每个对象 x **执行**

      $\eta_x^- \leftarrow$ 最小的 $k-1$ 个值之和 min(c_{xj}，c_{jx})，$j \in S \setminus \{x\}$

      $\eta_x^+ \leftarrow$ 最大的 $k-1$ 个值之和 max(c_{xj}，c_{jx})，$j \in S \setminus \{x\}$

      $L_x \leftarrow -2\gamma_x + 2\eta_x^-$；$H_x \leftarrow -2\gamma_x + 2\eta_x^+$；$W_x \leftarrow -2\gamma_x + \eta_x^+$

    结束循环

    对每个 $a \in S$ **执行**　　▷使用上面计算出的值

      如果 H_a 小于第 k大的 L_j，则移除 a，$j \neq a$　　▷ 交换检验

      如果 W_a 加上最大的 $k-1$ 个 W_x后小于 F_0，则移除 a，$x \in S \setminus \{a\}$　　▷ 界限检验

    循环结束

  直到 S 不变

  返回 S

过程结束

B 实验

B.1 可复现性

所有计算均在一台配备 Intel Core i5-6500 处理器（四核，3.2 GHz）和 16 GB 内存的台式电脑上运行，操作系统为 Windows 11，使用 R 4.6.1。精确公式使用 R 软件包 highs 1.14.0.2 调用 HiGHS 1.14.0 求解。SIgFUR 和 FUR 在 MATLAB R2023a 中运行，使用的是作者提供的代码（https: //github.com/prakashvs613/RankAggSIgFUR-tied-matlab）。

每项实验均由一份带注释的 R 脚本执行，脚本在开头固定随机数生成器及其种子，并将结果与 R 会话说明一同保存；长时间运行的实验会为每个问题单独保存一个文件。论文及其补充材料中的摘要和表格均根据已保存的结果生成，无需重复计算。BOLT 具有确定性，因此所有不涉及计算时间的结果都能精确复现。模拟研究中的排列根据 Fligner 和 Verducci（1986）提出的表示方法，从采用 Kendall 距离的 Mallows 模型中抽取；我们将抽样器与包含 $n = 4$ 个和 $n = 5$ 个对象、且 θ 取三个值时所有排列的精确概率进行了核验，每种情况各抽取 50，000 次；卡方检验的 p-值介于 0.33 到 0.97 之间。论文中使用的数据集均可公开获取，但欧洲晴雨表和 MovieLens 数据无法再分发；对于这两者，我们提供了读取原始文件的脚本。

B.2 针对具有经认证的最优解的问题进行消融

表 B.1 重复了论文表 4 中针对第 7.2 节所述 83 个具有经认证最优解的真实问题和模拟问题进行的消融。

表 B.1：BOLT 在 83 个具有经认证最优解的问题上的消融：最优解数量、与最优值的差距（以 $\bar{\tau}_X$ 计）以及变体表现逊于 BOLT 的问题数量。上半部分：移除一个组件；下半部分：单一初始解（及其由无并列局部搜索得到的完全排名）。

变体	最优解（共 83 个）	mean gap	max gap	比 BOLT 差（共 83 个）
所有组件	72	7.0e-06	8.3e-05	0
不含完整排序	72	8.2e-06	1.6e-04	2
仅完整排序	68	1.2e-05	1.6e-04	7
仅来自 C 的初始解	65	2.1e-05	2.1e-04	11
仅评委排名	71	8.6e-06	1.7e-04	3
不使用 OTP	68	2.2e-05	5.3e-04	6
仅 OTP（无局部搜索）	63	8.6e-05	4.5e-03	15
仅净流	50	1.0e-04	1.0e-03	32
仅 Copeland	58	5.1e-05	4.9e-04	24
仅马尔可夫链	51	9.7e-05	1.1e-03	30
仅桶枢轴	52	1.2e-04	1.6e-03	29
仅分支定界起点	53	6.5e-05	6.4e-04	27
最佳单独评委	71	8.6e-06	1.7e-04	3
中位数单独评委	54	6.8e-05	4.6e-04	29
仅最差评判	27	3.3e-04	2.6e-03	56

B.3 约束问题的认证

论文表 5 中的问题按如下方式生成。对于每个 $n \in \{20, 50\}$ 以及十次重复实验中的每一次，$m = 100$ 位评审者的排名（不含并列）均从两个 Mallows 模型的混合中抽取

其中两个模型的参数均为 $\theta = 0.2$，权重相等，中心分别为恒等排列和一个随机排列；随后将四个约束应用于同一组数据。第 6.2 节的精确公式（含其中所述的强化不等式和割）以 BOLT 的解为初始解，使用四个线程，时间限制为 600 秒，并在所有解上检查该约束。在 11 个未能在时间限制内求解的问题中，有 4 个问题均包含部分模式，精确公式改进了 BOLT 的解。各问题的结果见表 B.8。

B.4 与MorK的比较

Azzini 等人（2027）针对每个对象数量 $n = 10,\ldots,50$，生成了包含 $m = 10$名评审者的问题，这些评审者的排名是随机排列，权重也随机生成： $n \leq 40$个问题对应 100 个， $41 \leq n \leq 45$ 对应 67 个， $46 \leq n \leq 50$对应 33 个，总计 3600 个问题。由于他们的问题无法获取，我们以相同方式生成了 3600 个问题，权重从 (0,1) 上的均匀分布中抽取并归一化。所有问题均使用第6.1节的公式和 $y_{ij} + y_{ji} = 1$精确求解，耗时不超过 25 秒；对于 $n \leq 30$，我们还通过 MorK-Exact 的实现确认了最优解。解的差距为 $1 - \phi/\phi^*$，其中 ϕ 是其得分， ϕ^*是最优解的得分；得分 $\phi^{(E)}$ 采用 Azzini 等人（2027，公式11）的定义。他们的表3没有说明使用的是哪种得分，我们假定采用的是同一种。表B.2针对每个n报告了 BOLT 返回最优解的问题数量及其平均差距，并列出了 Azzini 等人（2027）已发布的 MorK-Heuristic 平均差距。需要注意两点：这些问题来自相同的分布，但并非同一批问题；此外，当 MorK-Exact 未能在一小时内求解所有问题时（$n = 43$、45 和 46 至 50），已发布的平均值仅针对已求解的问题（ $n = 50$中 33 个问题里的 17 个）。

表B.2：与 MorK 的比较：对于每个 n，问题数量、BOLT 返回最优解的问题数量，以及 BOLT 和 MorK-Heuristic（已发布值）的平均差距（$\times 10^{-5}$）。

n	问题	BOLT 最优值	BOLT	MorK-H	n	问题	BOLT 最优值	BOLT	MorK-H
10	100	100	0.00	0.00	31	100	85	1.60	8.70
11	100	100	0.00	1.32	32	100	84	2.38	13.27
12	100	99	0.56	2.18	33	100	81	2.74	10.40
13	100	100	0.00	3.40	34	100	82	2.60	12.58
14	100	99	1.07	3.02	35	100	77	3.65	11.36
15	100	98	1.22	4.66	36	100	73	2.76	13.14
16	100	99	0.38	4.79	37	100	70	3.77	10.10
17	100	100	0.00	10.49	38	100	76	2.86	9.72
18	100	97	1.11	11.15	39	100	65	4.43	16.16
19	100	98	0.69	5.42	40	100	61	5.76	17.09
20	100	98	0.57	6.98	41	67	48	3.78	13.80
21	100	99	0.22	11.28	42	67	35	5.17	10.25
22	100	96	1.02	4.32	43	67	34	5.57	14.24
23	100	96	1.35	6.38	44	67	37	5.24	12.00
24	100	96	1.02	13.34	45	67	26	7.69	13.57
25	100	90	2.26	11.64	46	33	17	5.99	9.58
26	100	90	2.17	11.52	47	33	8	7.78	12.85
27	100	96	0.49	8.63	48	33	18	4.07	11.11
28	100	92	0.82	10.10	49	33	14	6.68	15.83
29	100	88	1.25	16.61	50	33	16	6.58	5.77
30	100	79	1.87	12.32					

n 的 41 个值的平均值：BOLT 为 2.66，MorK-Heuristic 为 9.78；在 3600 个问题中，BOLT 在 3017 个问题上达到最优。

B.5 欧洲晴雨表：问题、样本和完整模式结果

欧洲晴雨表55.2调查（欧盟委员会，2012）于2001年5月和6月在欧盟15个成员国开展，共有16,130 名受访者。问题 Q21 是：“假设您所在地区发生重大灾害。您最信任谁来解释灾害发生的原因？”，随后还问：“其次最信任谁？”以及“第三最信任谁？”。受访者须从 $n = 7$ 种来源中选择：科学家（科学）、记者（期刊）、消费者组织（会议）、环境保护协会（环境）、政府代表（政府）、医生（文档）和公司（商业）；“其他”“以上皆非”和“不知道”是自发回答。我们使用了其中选择七种来源中的三种的 $m = 12{,}765$ 名受访者，因此每人都提供了一个前3名列表。数据由 GESIS 提供（研究 ZA3509，doi:10.4232/1.10943），不得再分发；读取原始文件的脚本可向作者获取。

表 B.3 按第三种约定，报告了经 3 个候选项认证的中位数前三列表、常见汇总结果中的前三名列表，以及另外两种约束中位排名，并附有其τ_X^{obs}。表 B.4 比较了按第二种和第三种约定得到的解：每个解都用两种约定的目标函数进行评估，第二种约定的平均 τ_X 和第三种约定的 τ_X^{obs}。中位数前三列表和模式为 (3,4) 的中位排名（即不考虑顺序的最受信任的三个来源）不受约定影响，而具有 $b = 3$ 个桶的 MCBO 则会受到影响：在第二种约定下，未被选中来源之间的并列计为一致，从而使解倾向于采用较大的最后一个桶（公式11）；在第三种约定下，较不受信任的来源仅通过被选中来源与未被选中来源之间的比较来区分，且公司被单独列为最不受信任的来源。对于这些数据，我们还验证了：采用第三种约定时，每位受访者的得分矩阵是所有与前三名列表相容的完整排序的得分矩阵的平均值，并且两种约定的组合输入矩阵相差u_{ij}。

表 B.3：欧洲晴雨表 55.2（$m = 12,765$ 个关于 $n = 7$ 来源的前三名列表），第三种约定：中位数前三列表、通常摘要的前三名列表以及其他约束中位排名，并附上它们在观测到的对象对上的平均值 τ_X。

解	排名	τ_X^{obs}
中位数前三列表	科学 > 环境 >文档 > (期刊、会议、政府、商业)	0.4372
首次提及的前3名列表	科学 > 环境 >文档 > (期刊、会议、政府、商业)	0.4372
总提及次数的前3名列表	科学 >环境 > 文档 >(期刊、会议、政府、商业)	0.4372
3-2-1 分数的前3名列表	科学 >环境 > 文档 >(期刊、会议、政府、商业)	0.4372
中位数前三集合，模式 (3,4)	(科学, 环境, 文档) > (期刊, 会议, 政府, 商业)	0.4048
MCBO，$b = 3$	(科学, 环境, 文档) > (期刊, 会议, 政府) >商业	0.4492

表 B.4：欧洲晴雨表 55.2：中位数前三列表、模式为 (3,4) 的中位排名（前3名集合），以及使用第二种和第三种约定的 MCBO（括号内为得出该解的约定），其中桶数为 $b = 3$；每项均使用两种约定的目标函数进行评估。

解	排名	$\bar{\tau}_X$ (第二)	τ_X^{obs} (第三)
前3名列表（两者）	科学 >环境 > 文档 >(期刊、会议、政府、商业)	0.4745	0.4372
前3名集合（两者）	(科学、环境、文档) >(期刊、会议、政府、商业)	0.4598	0.4048
$b = 3$ (第二)	科学 > (环境、文档) >(期刊、会议、政府、商业)	0.4742	0.4320
$b = 3$ (第三)	(科学、环境、文档) > (期刊、会议、政府) > 商业	0.3973	0.4492

B.6 受约束问题的模拟研究

目标与设计。 评估启发式方法需考察两个方面：它找到最优解的频率，以及所需的计算成本。由于只有在可接受排名的子全集可以枚举时才能知道最优解，而只有在无法枚举时计算时间才有意义，因此这两个方面无法在相同条件下评估。根据公式15，包含 $n = 12$ 个对象的模式 (1, 3,4, 3, 1) 对应 554,400 种桶排序，枚举这些排序约需 7 秒；而包含 $n = 20$ 个对象的模式 (2,4, 8, 4,2) 则对应约 2.5×10^{10} 种桶排序。因此，每项实验都包括一部分对象较少的情形，在这种情形下将 BOLT 与通过枚举得到的最优解进行比较；在适用时，还会包括一部分对象较多的情形。每种约束各设一项实验。对于中位数前k 名列表，主要问题是第 5.6 节的精确算法能在多大程度上认证该解；对于具有指定模式的中位排名，没有其他可用算法，因此将 BOLT 与最优解进行比较；对于具有 b 个自由大小桶的 MCBO，也将 BOLT 与 D'Ambrosio 等人（2019）的 QUICK 和 DECoR 算法进行比较。

数据生成。 我们从两种标准排名模型中抽样得到排列或其前若干个位置，随后创建桶和并列。第一种是基于 Kendall 距离的 Mallows 模型（Mallows，1957），其中心为 (1，2，.....， n)，其中 $\theta = 0.05$ 为 0。2 和 1（ θ 越大，评审者之间的一致程度越高），抽样采用 Fligner 和 Verducci（1986）提出的表示方法。第二种是 Plackett–Luce 模型（Luce，1959年；Plackett，1975），其中对象的价值分别为 exp(u_1)，.....，exp(u_n)， $u_1 > \cdots > u_n$ 从均值为零、标准差为 s 的正态分布中抽取；每位评审者依次选择对象，选择概率与尚未选取对象的价值成正比； s 越大，评审者之间的一致程度越高。当所有评审者都来自单一 Mallows 模型时，对于大多数对象对，明确的多数意见与中心一致；在我们的初步试验中，几乎所有数据集上的所有算法都找到了最优解。因此，我们还从两个具有相同 θ 且权重相等的 Mallows 模型的混合中抽样生成数据：第一个模型的中心为 (1，2，.....， n)，第二个模型的中心则是一个随机排列。平均而言，这两个中心对半数对象对的排序方式相反，这些对象对的 c_{ij} 接近零，惩罚函数因而趋于平坦，并出现许多局部最优解：正如论文第 7.4 节所述，这种混合是对算法的压力测试。由于这种混合没有单一中心，只有在单一 Mallows 模型下才研究真实中心的恢复情况。

中位数前k 列表。 </style}每位评审者都会提供一个前20名列表，即从两个模型之一中抽取的排列的前20个位置，其余对象并列位于最后位置（第 3.3 节中的第二种约定）。我们考察了、500、1000 和 5000 个对象，、0.2、1 和 $s = 0.5$、1、2；每种设置各有 10 个数据集。我们计算了参数为 $k = 5$、10 和 20 的中位数前k列表，共计 720 个问题。关键指标是经认证的解所占比例：当候选项（即唯一可能属于最优前k 列表的对象）不超过 22 个时，精确算法无需枚举 $n!/(n-k)!$个前k 列表（约为 10^{37}，其中 $n = 5000$ 和 $k = 10$），即可认证该解；否则，它会返回 BOLT 的解。

图 1(a) 显示，精确算法在 94.2% 的问题中认证了解：对于 $k = 5$ 和 $k = 10$，始终如此；对于 $k = 20$ ，则在 82.5% 的问题中如此（Mallows 模型为 96.7%，Plackett–Luce 模型为 68.3%）。未能认证的问题主要出现在 $n = 100$ 以及 Plackett–Luce 模型一致性最低的情形 $s = 0.5$中：此时多个对象的列总和与第 20 个对象的列总和接近，因而无法排除。当解获得认证时，BOLT 除两个问题外，在所有问题中都找到了最优解，平均 τ_X的相对差异至多为 3.7×10^{-8}，因此在解无法获得认证时，它是可靠的备用方案。精确算法的速度也很快（图 1(b)）：在已认证的问题中，最多有 1000 个对象时，计算时间的中位数不到 0.1 秒；有 5000 个对象时为 2.2 秒，而 BOLT 在 5000 个对象时耗时 22.5 秒。至于

预期结果，恢复真实 top-k 列表的效果取决于评审者之间的一致性：在$\theta = 1$ 情形下，中位数 top-k 列表始终是真实列表；在 $\theta = 0.2$ 情形下，76.7% 的问题中包含真实的 k 个对象，其中 65.8% 的问题中对象顺序也正确；在 $\theta = 0.05$ 情形下以及使用 Plackett–Luce 模型时，列表包含正确对象的频率较低，顺序正确的情况则很少。在 $s = 0.5$ 情形下，共最优 top-k 列表可能非常多，因为从不出现在列表中的对象在组合输入矩阵中的列完全相同。

指定模式。每位评审者给出的桶排序，是将从 Mallows 模型或混合物中抽取的一个排列按指定大小切分成桶后得到的，如同 Q排序；以相同方式切分的真实中心具有相同的模式。在实验的第一部分，通过枚举子全集 $\mathcal{Z}^{n\backslash\nu}$： $n = 8$ （模式为 (1, 2, 2, 2, 1)，共 5040 种桶排序）、 $n = 10$（模式为 (1, 2, 4, 2, 1)，共 37,800 种）、 $n = 10$（部分模式为 (2, $*$, $*$, 2)，共 78,120 种；数据按大小 (2, 3, 3, 2) 切分）以及 $n = 12$（模式为 (1, 3, 4, 3, 1)，共 554,400 种）来求得最优解；每种设置下均使用 50 个包含 $m = 100$ 位评审者的数据集。图2显示，BOLT 在 1200 个数据集中有 97.4% 找到了最优解；使用 Mallows 模型时这一比例为 98.7%，使用混合物时为 96.2%； $\theta = 0.05$ 且使用混合物时，该比例最低，也就是说，此时惩罚是平坦的。在未得到最优解的 31 个数据集中，与最优平均值 τ_X 相比，相对差异平均为 1.1%，最高为 3.5%；BOLT 每个问题耗时约 0.005 秒。在单一 Mallows模型下，对于所有包含 $\theta = 1$ 的数据集，真实中心都是中位排名；对于包含 $\theta = 0.2$ 的数据集，这一比例为 37.5%；而对于包含 $\theta = 0.05$ 的数据集，这一比例仅为 1%：当离散度较大时，即便模式正确，中位排名也不是生成数据的模型中心。

在第二部分中，我们考虑了 $n = 50$、200 个和 1000 个对象，将它们按网格排列。该网格有七列，各列大小与 (1，2，3，4，3，2，1) 成比例；评审者来自 Mallows 模型，评审者数量为 $m = 500$，每种设置包含 10 个数据集。这里无法计算最优解，因此我们将中位排名与网格模式（其真实中心）以及具有 $b = 7$ 个自由大小桶的 MCBO 进行了比较。采用该模式的解的平均值从未低于真实中心的平均值，而且两种解非常接近（它们之间的 τ_X 为 0.91 到 1）。模式的代价，即具有 $b = 7$ 个桶的 MCBO 的平均值与采用该模式的解的平均值之差，很小（至多为 0.006，且 $n = 50$ 和 $\theta = 0.05$），并且在评审者意见一致时会消失。对于 1000 个对象，BOLT 耗时不到一秒。

中位数约束桶排序。 在最后一项实验中，评审者给出的排列取样自 Mallows 模型或混合物，且解必须恰好包含 b 个自由大小的桶。对于最优解，分支定界算法在对象数达到 $n = 10$ 时就已经太慢，因此我们枚举了 $\mathcal{Z}^{n\backslash b}$ 种排列，其中（n，b） = 包括（8，4）、（9，5）和（10，4），也就是 40，824、834，120 和 818，520 种桶排序；每种设置下均使用包含 $m = 100$ 位评审者的 50 个数据集。在第二部分中，我们考虑了 $n = 20$、50、100、200 和 500 个对象， $b = 5$ 个桶和 $m = 100$ 位评审者；每种设置使用 10 个数据集，并将每种算法与三种算法中任一种找到的最佳解进行比较。

图 3(a) 显示，在对象较少时，三种算法在大多数数据集中都找到了最优解：BOLT 在其中 98.2% 的数据集中找到最优解，QUICK 为 96.2%，DECoR 为 99.2%；它们与最优平均值的最大相对差异分别为 τ_X 1.7%、2.8% 和 1.4%。最难的设置是单一 Mallows 模型，参数为 $\theta = 0.05$、$n = 10$ 和 $b = 4$，此时三者的占比分别为 92%、84% 和 96%。BOLT 和 QUICK 每个问题耗时约 0.01 秒，而 DECoR 约需 6 秒：对象较少时，DECoR 的准确度略高于 BOLT，但代价要大得多。随着对象数量增加，情况有所不同（图 3(b)）。在单一 Mallows 模型下，三种算法的表现相近（BOLT 在 99.3% 的数据集中找到最佳解，QUICK 为 96%，DECoR 为 97.3%）；但在混合模型下，占比分别为 88.7%、78.7% 和 34.7%。DECoR 的表现会随着 n迅速恶化；在混合模型且 $n \geq 200$ 时，它从未

找到最佳解；而对于每个 n，BOLT 在 83% 到 93% 的数据集中找到了最佳解。对于 500 个对象，BOLT 的平均耗时为 0.5 秒，QUICK 为 163 秒，DECoR 为 154 秒。

B.7 完整结果

表B.5–B.8逐个问题报告结果；对于模拟研究，则逐个单元格报告结果。这些结果用于计算论文第7节中的汇总数据。表B.5中的64个模拟问题各有 $m = 20$ 位评审者，其排列是根据 Kendall 距离从 Mallows 模型中抽取的，参数为 $\theta = 0.05$ 和 0.2；并列则通过以 0.3 的概率合并每对相邻对象之间的间隔来引入；每个$n = 20$，30、 40、 50对应五个问题，而每个 $n = 60$，70、 80、 90对应三个问题。

表B.5：第 7.2 节中具有经认证的最优解的83个问题：每种方法的最优解平均值 τ_X 以及与最优值的差距（PrefLib数据集按其代码列出；随机化方法：100次运行中的最佳结果；NSKS：NonStrictKwik- Sort）。

问题	n	m	$\bar{\tau}_X$最优。	BOLT BioConsert		NSKS KwikSort		Fagin
00028-00000001	5	18,723	0.23869	0	0	0	8.4e-02	0
机器学习算法	12	8	0.72159	0	0	0	3.8e-03	0
00001-00000001	12	43,942	0.39751	0	0	0	1.6e-01	2.3e-03
EMD	15	112	0.16556	0	0	0	2.4e-03	1.2e-01
00033-00000002	17	65	0.60928	0	0	0	4.5e-01	2.3e-02
00006-00000019	28	9	0.86949	0	0	0	0	1.2e-02
00006-00000013	29	9	0.85769	0	0	5.5e-04	5.5e-04	1.3e-02
00006-00000001	30	9	0.88455	0	0	0	2.6e-04	7.4e-03
00006-00000039	30	9	0.86820	0	0	0	0	1.7e-02
00006-00000041	30	9	0.84751	0	0	0	0	1.4e-02
00038-00000003	102	32	0.90390	0	0	0	8.4e-01	0
00038-00000005	103	31	0.90482	0	0	0	8.5e-01	0
00038-00000006	133	38	0.92595	0	0	0	8.8e-01	0
00038-00000008	147	51	0.92028	0	0	0	8.7e-01	0
00038-00000007	155	51	0.93632	0	0	0	8.9e-01	0
大学	229	10	0.72026	7.7e-06	6.9e-05	4.2e-03	3.6e-03	5.9e-03
大学 ，研究	229	4.5	0.73866	6.8e-05	1.3e-04	7.7e-03	8.6e-03	7.4e-03
大学教学	229	3	0.65611	0	0	3.9e-03	6.5e-02	1.3e-02
城市	240	4	0.74793	7.0e-05	2.1e-04	3.5e-03	4.3e-03	1.1e-02
模拟， $\theta = 0.05$， 重复 1	20	20	0.19921	0	0	1.6e-03	4.7e-03	1.0e-01
模拟， $\theta = 0.05$， 重复 2	20	20	0.17816	0	0	3.2e-03	5.3e-03	1.4e-01
模拟， $\theta = 0.05$， 重复 3	20	20	0.19947	0	0	5.8e-03	3.7e-03	1.0e-01
模拟， $\theta = 0.05$， 重复 4	20	20	0.19632	0	0	2.6e-03	2.1e-03	1.5e-01
模拟， $\theta = 0.05$， 重复 5	20	20	0.21632	0	0	5.3e-03	3.7e-03	1.3e-01
模拟， $\theta = 0.20$， 重复 1	20	20	0.45895	0	0	0	5.3e-04	3.6e-02
模拟， $\theta = 0.20$， 重复 2	20	20	0.46474	0	0	5.3e-04	1.1e-03	1.7e-02
模拟， $\theta = 0.20$， 重复 3	20	20	0.40158	0	0	3.2e-03	2.1e-03	2.5e-02
模拟， $\theta = 0.20$， 重复 4	20	20	0.46684	0	0	0	5.3e-04	2.5e-02
模拟， $\theta = 0.20$， 重复 5	20	20	0.49079	0	0	0	1.1e-03	3.3e-02
模拟， $\theta = 0.05$， 重复 1	30	20	0.20115	0	0	8.0e-03	6.7e-03	1.6e-01
模拟， $\theta = 0.05$， 重复 2	30	20	0.26483	0	0	6.2e-03	1.4e-03	1.0e-01
模拟， $\theta = 0.05$， 重复 3	30	20	0.27414	0	0	3.4e-03	5.1e-03	3.8e-02
模拟， $\theta = 0.05$， 重复 4	30	20	0.25851	0	0	3.2e-03	3.2e-03	1.4e-01
模拟， $\theta = 0.05$， 重复 5	30	20	0.26402	0	0	2.1e-03	1.8e-03	1.7e-01
模拟， $\theta = 0.20$， 重复 1	30	20	0.55471	0	0	1.4e-03	9.2e-04	2.5e-02
模拟， $\theta = 0.20$， 重复 2	30	20	0.53529	0	0	6.9e-04	6.9e-04	3.1e-02
模拟， $\theta = 0.20$, 重复 3	30	20	0.54862	0	0	4.6e-04	1.6e-03	1.9e-02
模拟， $\theta = 0.20$， 重复 4	30	20	0.58448	0	0	6.9e-04	0	1.6e-02
模拟， $\theta = 0.20$， 重复 5	30	20	0.55161	0	0	9.2e-04	1.6e-03	2.6e-02
模拟， $\theta = 0.05$， 重复 1	40	20	0.25532	0	1.3e-04	3.6e-03	4.4e-03	1.4e-01
模拟， $\theta = 0.05$， 重复 2	40	20	0.26526	0	0	6.7e-03	4.9e-03	7.6e-02
模拟， $\theta = 0.05$， 重复 3	40	20	0.27949	0	0	3.7e-03	3.2e-03	9.2e-02
模拟， $\theta = 0.05$， 重复 4	40	20	0.31276	0	0	5.8e-03	3.6e-03	4.8e-02
模拟， $\theta = 0.05$， 重复 5	40	20	0.26577	0	1.3e-04	5.0e-03	4.0e-03	8.4e-02
模拟， $\theta = 0.20$， 重复 1	40	20	0.61904	0	0	6.4e-04	5.1e-04	1.1e-02
模拟， $\theta = 0.20$， 重复 2	40	20	0.64712	0	0	9.0e-04	6.4e-04	1.3e-02
模拟， $\theta = 0.20$， 重复 3	40	20	0.65692	0	0	2.6e-04	7.7e-04	9.1e-03
模拟， $\theta = 0.20$， 重复 4	40	20	0.62295	0	0	5.1e-04	6.4e-04	1.7e-02
模拟， $\theta = 0.20$， 重复 5	40	20	0.64295	0	0	3.8e-04	3.8e-04	1.1e-02
模拟， $\theta = 0.05$， 重复 1	50	20	0.31792	0	1.6e-04	5.6e-03	2.9e-03	3.8e-02

（续）

问题	n	m	$\bar{\tau}_X$最优。	BOLT BioConsert		NSKS KwikSort		Fagin
模拟，$\theta = 0.05$，重复 2	50	20	0.30163	0	0	6.4e-03	5.6e-03	3.5e-02
模拟，$\theta = 0.05$，重复 3	50	20	0.31314	0	0	8.2e-03	4.8e-03	4.5e-02
模拟，$\theta = 0.05$，重复 4	50	20	0.29571	8.2e-05	8.2e-05	7.6e-03	5.6e-03	8.8e-02
模拟，$\theta = 0.05$，重复 5	50	20	0.30722	0	0	6.9e-03	4.2e-03	4.3e-02
模拟，$\theta = 0.20$，重复 1	50	20	0.70237	0	0	9.0e-04	2.4e-04	9.1e-03
模拟，$\theta = 0.20$，重复 2	50	20	0.69073	0	0	1.2e-03	7.3e-04	1.1e-02
模拟，$\theta = 0.20$，重复 3	50	20	0.69233	0	0	9.8e-04	5.7e-04	1.2e-02
模拟，$\theta = 0.20$，重复 4	50	20	0.72249	0	0	5.7e-04	4.9e-04	1.1e-02
模拟，$\theta = 0.20$，重复 5	50	20	0.69673	0	0	2.4e-04	4.1e-04	1.0e-02
模拟，$\theta = 0.05$，重复 1	60	20	0.35181	0	5.6e-05	7.5e-03	5.4e-03	3.4e-02
模拟，$\theta = 0.05$，重复 2	60	20	0.36014	0	0	6.4e-03	4.6e-03	3.9e-02
模拟，$\theta = 0.05$，重复 3	60	20	0.35056	5.6e-05	1.7e-04	6.7e-03	4.7e-03	4.0e-02
模拟，$\theta = 0.20$，重复 1	60	20	0.74712	0	0	9.0e-04	8.5e-04	8.6e-03
模拟，$\theta = 0.20$，重复 2	60	20	0.74347	0	0	1.1e-03	7.3e-04	6.2e-03
模拟，$\theta = 0.20$，重复 3	60	20	0.75749	0	0	7.9e-04	4.0e-04	8.2e-03
模拟，$\theta = 0.05$，重复 1	70	20	0.40118	4.1e-05	8.3e-05	4.3e-03	4.8e-03	3.0e-02
模拟，$\theta = 0.05$，重复 2	70	20	0.38665	8.3e-05	0	5.3e-03	4.2e-03	3.0e-02
模拟，$\theta = 0.05$，重复 3	70	20	0.36072	0	8.3e-05	7.0e-03	6.4e-03	3.6e-02
模拟，$\theta = 0.20$，重复 1	70	20	0.77739	0	0	4.6e-04	4.6e-04	4.9e-03
模拟，$\theta = 0.20$，重复 2	70	20	0.77660	0	0	5.0e-04	7.0e-04	5.3e-03
模拟，$\theta = 0.20$，重复 3	70	20	0.76867	0	0	7.0e-04	1.7e-04	4.9e-03
模拟，$\theta = 0.05$，重复 1	80	20	0.44502	6.3e-05	3.2e-05	5.2e-03	3.8e-03	1.8e-02
模拟，$\theta = 0.05$，重复 2	80	20	0.43285	3.2e-05	9.5e-05	4.8e-03	3.7e-03	3.1e-02
模拟，$\theta = 0.05$，重复 3	80	20	0.41983	0	1.3e-04	4.8e-03	4.9e-03	3.2e-02
模拟，$\theta = 0.20$，重复 1	80	20	0.79437	0	0	1.2e-03	5.4e-04	4.1e-03
模拟，$\theta = 0.20$，重复 2	80	20	0.79965	0	0	5.1e-04	4.7e-04	5.3e-03
模拟，$\theta = 0.20$，重复 3	80	20	0.79389	0	0	4.7e-04	6.0e-04	3.7e-03
模拟，$\theta = 0.05$，重复 1	90	20	0.47099	2.5e-05	1.2e-04	4.8e-03	2.9e-03	2.0e-02
模拟，$\theta = 0.05$，重复 2	90	20	0.45019	0	1.2e-04	4.8e-03	5.3e-03	2.1e-02
模拟，$\theta = 0.05$，重复 3	90	20	0.45838	5.0e-05	2.5e-05	5.2e-03	4.2e-03	2.2e-02
模拟，$\theta = 0.20$，重复 1	90	20	0.81094	0	0	8.5e-04	5.7e-04	4.7e-03
模拟，$\theta = 0.20$，重复 2	90	20	0.82037	0	0	4.5e-04	3.2e-04	3.5e-03
模拟，$\theta = 0.20$，重复 3	90	20	0.82235	0	0	7.0e-04	4.7e-04	3.9e-03

表 B.6：Akbari 和 Escobedo（2023a）的 85 个 PrefLib数据集：下界 P_{LB}（以距离总和表示）、BOLT 与已发布的最佳 ACP 解相对于下界的差距（占界的百分比，已发布数值保留两位小数），以及 BOLT 的运行时间。

数据集	n	m	下界	BOLT(%)	ACP(%)	时间(s)
ED-14-02	100	5000	4,725,000	0.00	0.00	8.58
ED-14-03	100	5000	4,651,850	0.00	0.00	8.62
MD-03-03	102	32	15,840	0.00	0.00	0.14
MD-03-05	103	31	15,500	0.00	0.00	0.20
MD-03-06	133	38	24,700	0.00	0.00	0.26
MD-03-08	147	51	43,629	0.00	0.00	0.31
MD-03-07	155	51	38,760	0.00	0.00	0.31
ED-10-50	170	4	24,378	0.79	1.44	0.19
ED-10-49	351	4	100,411	0.47	0.65	0.72
ED-18-01	379	36655	30,454,824	0.00	0.00	13.07
ED-18-03	477	32086	23,258,021	0.00	0.00	18.47
ED-11-12	1210	4	1,590,332	1.10	2.77	19.26
ED-11-31	1223	4	1,517,556	1.21	1.69	16.91
ED-11-09	1272	4	1,561,690	1.44	3.39	18.08
ED-11-23	1342	4	1,843,920	1.27	1.62	20.11
ED-11-21	1347	4	1,834,227	1.27	2.66	27.31
ED-11-37	1351	4	1,786,116	1.27	3.04	29.45
ED-11-25	1356	4	1,850,744	1.31	2.27	20.33
ED-11-13	1363	4	1,964,622	1.02	1.60	24.02
ED-11-29	1368	4	1,750,605	0.85	3.31	25.50
ED-11-14	1375	4	1,863,664	1.09	3.20	29.00
ED-11-30	1386	4	1,828,275	0.86	2.38	21.42
ED-11-06	1449	4	2,095,369	1.07	2.71	32.80
ED-11-04	1467	4	2,226,074	1.18	2.59	35.19
ED-11-07	1474	4	2,305,641	1.25	1.81	33.83
ED-11-34	1509	4	2,372,642	0.93	1.50	29.83
ED-11-22	1514	4	2,244,794	0.84	1.83	29.78

（续）

数据集	n	m	下界	BOLT(%)	ACP(%)	时间(s)
ED-11-11	1545	4	2,272,137	1.05	2.01	39.01
ED-11-15	1563	4	2,420,476	0.88	2.01	33.98
ED-11-08	1572	4	2,704,150	1.07	1.26	32.88
ED-11-28	1616	4	2,716,107	0.60	0.78	25.64
ED-11-40	1623	4	2,482,556	0.26	0.32	31.72
ED-11-36	1634	4	2,732,229	0.92	1.51	32.45
ED-11-33	1646	4	2,752,549	0.95	1.85	45.97
ED-11-05	1673	4	2,590,774	0.49	0.63	38.49
ED-11-18	1681	4	2,932,501	1.04	1.94	39.17
ED-11-16	1708	4	2,911,360	0.72	1.17	44.44
ED-11-32	1751	4	3,267,824	0.91	1.68	45.44
ED-11-38	1754	4	3,308,787	0.94	1.97	44.91
ED-11-39	1788	4	2,940,244	0.29	0.35	34.58
ED-11-68	1826	4	3,495,142	0.52	0.62	30.36
ED-11-49	1845	4	3,911,132	0.69	0.90	67.14
ED-11-20	1870	4	3,449,797	0.71	1.47	48.05
ED-11-26	1931	4	3,710,296	0.75	1.33	47.18
ED-11-35	1936	4	3,805,316	0.80	1.53	46.17
ED-11-74	1976	4	4,176,464	0.59	1.13	45.29
ED-11-60	1977	4	4,147,277	0.71	0.87	45.18
ED-11-58	2011	4	4,366,852	0.75	1.09	59.47
ED-11-62	2014	4	4,230,329	0.52	0.86	47.67
ED-11-17	2015	4	4,056,848	0.75	1.26	64.61
ED-11-66	2024	4	4,283,401	0.81	1.98	59.91
ED-11-24	2049	4	4,290,818	0.71	1.41	69.87
ED-11-67	2066	4	4,473,066	0.68	1.30	75.83
ED-11-27	2092	4	4,647,453	0.70	1.11	51.46
ED-11-10	2096	4	4,355,605	0.40	0.62	49.42
ED-11-19	2104	4	4,369,932	0.66	1.24	53.30
ED-11-50	2111	4	4,871,442	0.54	0.61	62.65
ED-11-51	2112	4	4,755,457	0.74	1.51	66.12
ED-11-65	2119	4	4,741,402	0.24	0.27	46.81
ED-11-41	2123	4	5,017,714	0.91	1.04	93.83
ED-11-71	2127	4	4,901,052	0.37	0.51	61.90
ED-11-46	2133	4	4,835,028	0.94	1.36	76.72
ED-11-43	2153	4	5,008,377	0.80	1.27	77.39
ED-11-48	2194	4	5,060,729	0.68	1.12	60.53
ED-11-52	2242	4	5,282,143	0.35	0.62	55.25
ED-11-73	2258	4	5,409,851	0.20	0.29	59.15
ED-11-45	2265	4	5,705,419	0.07	0.07	65.15
ED-11-70	2276	4	5,624,119	0.28	0.49	65.88
ED-11-59	2281	4	5,590,364	0.41	0.68	74.54
ED-11-77	2317	4	4,946,315	0.07	0.08	78.53
ED-11-53	2321	4	5,601,106	0.15	0.28	58.63
ED-11-69	2338	4	5,955,107	0.37	0.52	79.25
ED-11-55	2353	4	6,116,004	0.56	0.59	76.41
ED-11-75	2391	4	6,379,179	0.61	0.76	93.75
ED-11-44	2434	4	6,435,079	0.49	0.55	74.02
ED-11-64	2446	4	6,416,412	0.45	0.59	141.55
ED-11-72	2447	4	6,408,613	0.58	0.82	93.95
ED-11-63	2510	4	6,550,637	0.45	0.65	81.47
ED-11-54	2512	4	6,485,654	0.41	0.66	87.93
ED-11-57	2559	4	6,822,253	0.44	0.76	125.44
ED-11-76	2581	4	6,197,768	0.09	0.10	79.36
ED-11-42	2598	4	6,411,396	0.13	0.15	86.55
ED-11-56	2632	4	7,193,229	0.33	0.55	85.25
ED-11-61	2726	4	7,751,722	0.69	0.96	116.95
ED-11-47	2819	4	8,049,813	0.18	0.23	93.53

表 B.7：带并列的模拟问题（第 7.4 节），每个单元格包含十个问题：具有经认证的最优解的问题数量、其中 BOLT 返回最优解的问题数量、在已认证问题中与最优值的平均值和最大间隙，以及 BOLT 在该单元格所有问题上的平均运行时间（模型：Mallows 模型或两个 Mallows 模型的混合）。

n	m	θ	p_{tie}	模型	已认证 BOLT 最优值	平均间隙	最大间隙	时间（秒）
50	10	0.02	0.1	混合物	10	3	2.6e-04	4.9e-04	0.04

（续）

n	m	θ	p_{tie}	模型	已认证 BOLT 最优值		平均间隙	最大间隙	时间（秒）
50	10	0.02	0.1	单一	10	2	2.9e-04	8.2e-04	0.04
50	10	0.02	0.3	混合物	10	3	2.1e-04	4.9e-04	0.05
50	10	0.02	0.3	单一	10	7	6.5e-05	3.3e-04	0.05
50	10	0.02	0.6	混合物	10	4	1.1e-04	3.3e-04	0.05
50	10	0.02	0.6	单一	10	8	4.9e-05	3.3e-04	0.06
50	10	0.05	0.1	混合物	10	6	1.5e-04	9.8e-04	0.04
50	10	0.05	0.1	单一	10	6	1.3e-04	4.9e-04	0.04
50	10	0.05	0.3	混合物	10	6	6.5e-05	1.6e-04	0.05
50	10	0.05	0.3	单一	10	6	6.5e-05	1.6e-04	0.05
50	10	0.05	0.6	混合物	10	7	8.2e-05	4.9e-04	0.04
50	10	0.05	0.6	单一	10	7	6.5e-05	3.3e-04	0.05
50	10	0.20	0.1	混合物	10	9	3.3e-05	3.3e-04	0.04
50	10	0.20	0.1	单一	10	10	0	0	0.05
50	10	0.20	0.3	混合物	10	10	0	0	0.05
50	10	0.20	0.3	单一	10	10	0	0	0.05
50	10	0.20	0.6	混合物	10	10	0	0	0.05
50	10	0.20	0.6	单一	10	10	0	0	0.04
50	50	0.02	0.1	混合物	10	8	6.5e-06	3.3e-05	0.11
50	50	0.02	0.1	单一	10	8	1.3e-05	9.8e-05	0.11
50	50	0.02	0.3	混合物	10	6	1.3e-05	3.3e-05	0.11
50	50	0.02	0.3	单一	10	7	9.8e-06	3.3e-05	0.12
50	50	0.02	0.6	混合物	10	9	3.3e-06	3.3e-05	0.15
50	50	0.02	0.6	单一	10	10	0	0	0.13
50	50	0.05	0.1	混合物	10	10	0	0	0.11
50	50	0.05	0.1	单一	10	10	0	0	0.11
50	50	0.05	0.3	混合物	10	9	3.3e-06	3.3e-05	0.12
50	50	0.05	0.3	单一	10	10	0	0	0.11
50	50	0.05	0.6	混合物	10	9	3.3e-06	3.3e-05	0.13
50	50	0.05	0.6	单一	10	8	6.5e-06	3.3e-05	0.14
50	50	0.20	0.1	混合物	10	10	0	0	0.10
50	50	0.20	0.1	单一	10	10	0	0	0.12
50	50	0.20	0.3	混合物	10	10	0	0	0.12
50	50	0.20	0.3	单一	10	10	0	0	0.11
50	50	0.20	0.6	混合物	10	10	0	0	0.12
50	50	0.20	0.6	单一	10	10	0	0	0.11
100	10	0.02	0.1	混合物	8	0	4.5e-04	5.3e-04	0.13
100	10	0.02	0.1	单一	10	0	3.5e-04	4.8e-04	0.11
100	10	0.02	0.3	混合物	7	0	3.7e-04	6.5e-04	0.12
100	10	0.02	0.3	单一	9	0	2.9e-04	4.8e-04	0.16
100	10	0.02	0.6	混合物	4	0	1.8e-04	2.0e-04	0.15
100	10	0.02	0.6	单一	9	0	1.9e-04	4.0e-04	0.16
100	10	0.05	0.1	混合物	10	0	2.8e-04	6.1e-04	0.11
100	10	0.05	0.1	单一	10	1	1.5e-04	3.2e-04	0.10
100	10	0.05	0.3	混合物	10	0	1.6e-04	3.2e-04	0.12
100	10	0.05	0.3	单一	10	1	1.1e-04	2.4e-04	0.11
100	10	0.05	0.6	混合物	10	0	1.1e-04	2.4e-04	0.12
100	10	0.05	0.6	单一	10	3	6.5e-05	2.0e-04	0.13
100	10	0.20	0.1	混合物	10	4	5.7e-05	2.0e-04	0.11
100	10	0.20	0.1	单一	10	9	4.0e-06	4.0e-05	0.10
100	10	0.20	0.3	混合物	10	6	2.0e-05	8.1e-05	0.14
100	10	0.20	0.3	单一	10	10	0	0	0.09
100	10	0.20	0.6	混合物	10	2	4.8e-05	1.2e-04	0.13
100	10	0.20	0.6	单一	10	10	0	0	0.10
100	50	0.02	0.1	混合物	10	2	3.6e-05	8.1e-05	0.39
100	50	0.02	0.1	单一	10	1	1.5e-05	3.2e-05	0.30
100	50	0.02	0.3	混合物	10	0	4.1e-05	8.1e-05	0.32
100	50	0.02	0.3	单一	10	1	2.3e-05	4.0e-05	0.40
100	50	0.02	0.6	混合物	9	2	2.1e-05	4.0e-05	0.38
100	50	0.02	0.6	单一	10	1	1.5e-05	3.2e-05	0.39
100	50	0.05	0.1	混合物	10	1	1.6e-05	4.0e-05	0.32
100	50	0.05	0.1	单一	10	8	2.4e-06	1.6e-05	0.29
100	50	0.05	0.3	混合物	10	3	1.3e-05	3.2e-05	0.31
100	50	0.05	0.3	单一	10	10	0	0	0.31
100	50	0.05	0.6	混合物	10	6	4.0e-06	1.6e-05	0.44
100	50	0.05	0.6	单一	10	6	9.7e-06	4.8e-05	0.40
100	50	0.20	0.1	混合物	10	8	1.6e-06	8.1e-06	0.31
100	50	0.20	0.1	单一	10	10	0	0	0.28
100	50	0.20	0.3	混合物	10	8	1.6e-06	8.1e-06	0.35
100	50	0.20	0.3	单一	10	10	0	0	0.31
100	50	0.20	0.6	混合物	10	7	2.4e-06	8.1e-06	0.42

（续）

n	m	θ	p_{tie}	模型	已认证 BOLT 最优值		平均间隙	最大间隙	时间（秒）
100	50	0.20	0.6	单一	10	8	1.6e-06	8.1e-06	0.31
200	10	0.02	0.1	混合物	0	0	–	–	0.32
200	10	0.02	0.1	单一	1	0	2.8e-04	2.8e-04	0.40
200	10	0.02	0.3	混合物	0	0	–	–	0.38
200	10	0.02	0.3	单一	0	0	–	–	0.40
200	10	0.02	0.6	混合物	0	0	–	–	0.40
200	10	0.02	0.6	单一	0	0	–	–	0.52
200	10	0.05	0.1	混合物	1	0	2.3e-04	2.3e-04	0.32
200	10	0.05	0.1	单一	1	0	1.3e-04	1.3e-04	0.37
200	10	0.05	0.3	混合物	1	0	2.0e-04	2.0e-04	0.33
200	10	0.05	0.3	单一	1	0	1.5e-04	1.5e-04	0.42
200	10	0.05	0.6	混合物	0	0	–	–	0.37
200	10	0.05	0.6	单一	1	0	1.2e-04	1.2e-04	0.51
200	10	0.20	0.1	混合物	1	0	4.0e-05	4.0e-05	0.30
200	10	0.20	0.1	单一	1	1	0	0	0.39
200	10	0.20	0.3	混合物	1	0	4.0e-05	4.0e-05	0.31
200	10	0.20	0.3	单一	1	1	0	0	0.39
200	10	0.20	0.6	混合物	1	0	3.0e-05	3.0e-05	0.39
200	10	0.20	0.6	单一	1	1	0	0	0.42
200	50	0.02	0.1	混合物	0	0	–	–	0.89
200	50	0.02	0.1	单一	1	0	3.8e-05	3.8e-05	1.06
200	50	0.02	0.3	混合物	0	0	–	–	0.95
200	50	0.02	0.3	单一	1	0	2.8e-05	2.8e-05	1.15
200	50	0.02	0.6	混合物	0	0	–	–	1.09
200	50	0.02	0.6	单一	1	0	1.2e-05	1.2e-05	1.37
200	50	0.05	0.1	混合物	1	0	1.0e-05	1.0e-05	0.88
200	50	0.05	0.1	单一	1	0	8.0e-06	8.0e-06	0.96
200	50	0.05	0.3	混合物	1	0	1.4e-05	1.4e-05	0.94
200	50	0.05	0.3	单一	1	0	4.0e-06	4.0e-06	1.07
200	50	0.05	0.6	混合物	1	0	1.6e-05	1.6e-05	1.10
200	50	0.05	0.6	单一	1	0	4.0e-06	4.0e-06	1.33
200	50	0.20	0.1	混合物	1	0	3.0e-05	3.0e-05	0.89
200	50	0.20	0.1	单一	1	1	0	0	0.86
200	50	0.20	0.3	混合物	1	1	0	0	0.94
200	50	0.20	0.3	单一	1	1	0	0	0.91
200	50	0.20	0.6	混合物	1	0	1.4e-05	1.4e-05	1.04
200	50	0.20	0.6	单一	1	1	0	0	0.92

表 B.8：受约束问题的 BOLT 认证（第 7.5 节），逐个问题列出：精确表述在 600 秒内的状态、BOLT 是否返回了最优解（仅限已认证问题）、精确表述找到的最佳解与 BOLT 之间的差距、经认证的界与 BOLT 之间的差距（均以 τ_X 计），以及耗时。

n	约束	Rep.	精确	BOLT 最优值	Gap	认证间隙	BOLT（秒）	精确值（秒）
20	完整模式	1	最优	yes	0	0	0.02	2.9
20	完整模式	2	最优	yes	0	0	0.00	0.5
20	完整模式	3	最优	yes	0	0	0.03	0.8
20	完整模式	4	最优	yes	0	0	0.00	0.7
20	完整模式	5	最优	yes	0	0	0.01	0.5
20	完整模式	6	最优	yes	0	0	0.02	1.6
20	完整模式	7	最优	yes	0	0	0.00	0.2
20	完整模式	8	最优	yes	0	0	0.00	1.0
20	完整模式	9	最优	yes	0	0	0.00	0.3
20	完整模式	10	最优	yes	0	0	0.00	0.3
20	部分模式	1	最优	no	4.2e-04	4.2e-04	0.01	15.2
20	部分模式	2	最优	yes	0	0	0.02	5.4
20	部分模式	3	最优	yes	0	0	0.01	4.2
20	部分模式	4	最优	yes	0	0	0.00	9.5
20	部分模式	5	最优	yes	0	0	0.02	6.0
20	部分模式	6	最优	yes	0	0	0.00	3.0
20	部分模式	7	最优	yes	0	0	0.00	7.2
20	部分模式	8	最优	yes	0	0	0.00	12.8
20	部分模式	9	最优	yes	0	0	0.00	5.5
20	部分模式	10	最优	no	2.1e-04	2.1e-04	0.00	14.6
20	$b = 5$ 桶	1	最优	yes	0	0	0.00	9.9
20	$b = 5$ 桶	2	最优	yes	0	0	0.00	7.7

（续）

n	约束	Rep.	精确	BOLT 最优值	Gap	认证间隙	BOLT（秒）	精确值（秒）
20	$b = 5$ 桶	3	最优	yes	0	0	0.00	7.6
20	$b = 5$ 桶	4	最优	yes	0	0	0.02	5.1
20	$b = 5$ 桶	5	最优	yes	0	0	0.00	14.6
20	$b = 5$ 桶	6	最优	yes	0	0	0.00	13.0
20	$b = 5$ 桶	7	最优	yes	0	0	0.01	12.0
20	$b = 5$ 桶	8	最优	yes	0	0	0.02	7.5
20	$b = 5$ 桶	9	最优	yes	0	0	0.02	6.8
20	$b = 5$ 桶	10	最优	yes	0	0	0.00	11.0
20	前五名列表	1	最优	yes	0	0	0.00	0.1
20	前五名列表	2	最优	yes	0	0	0.00	0.2
20	前五名列表	3	最优	yes	0	0	0.00	0.6
20	前五名列表	4	最优	yes	0	0	0.00	0.5
20	前五名列表	5	最优	yes	0	0	0.01	0.2
20	前五名列表	6	最优	yes	0	0	0.00	0.1
20	前五名列表	7	最优	yes	0	0	0.02	0.1
20	前五名列表	8	最优	yes	0	0	0.00	0.1
20	前五名列表	9	最优	yes	0	0	0.00	0.1
20	前五名列表	10	最优	yes	0	0	0.02	0.1
50	完整模式	1	时间限制	–	0	1.5e-05	0.02	600.0
50	完整模式	2	最优	yes	0	0	0.02	5.9
50	完整模式	3	最优	yes	0	0	0.02	23.8
50	完整模式	4	最优	yes	0	0	0.02	2.8
50	完整模式	5	最优	no	1.8e-04	1.8e-04	0.02	10.6
50	完整模式	6	最优	yes	0	0	0.02	15.8
50	完整模式	7	最优	yes	0	0	0.02	23.9
50	完整模式	8	最优	yes	0	0	0.02	3.6
50	完整模式	9	最优	yes	0	0	0.00	12.7
50	完整模式	10	最优	yes	0	0	0.01	6.1
50	部分模式	1	最优	yes	0	0	0.02	378.3
50	部分模式	2	时间限制	–	0	1.8e-03	0.02	600.1
50	部分模式	3	最优	yes	0	0	0.00	564.1
50	部分模式	4	时间限制	–	8.2e-05	1.3e-03	0.01	600.3
50	部分模式	5	时间限制	–	3.3e-05	1.0e-03	0.01	600.1
50	部分模式	6	时间限制	–	0	4.8e-03	0.01	600.1
50	部分模式	7	时间限制	–	0	2.5e-03	0.01	600.2
50	部分模式	8	时间限制	–	3.3e-05	3.7e-03	0.02	600.1
50	部分模式	9	最优	no	2.3e-04	2.3e-04	0.00	303.2
50	部分模式	10	时间限制	–	1.6e-04	1.3e-03	0.01	600.1
50	$b = 5$ 桶	1	最优	yes	0	0	0.00	448.3
50	$b = 5$ 桶	2	最优	yes	0	0	0.01	186.6
50	$b = 5$ 桶	3	最优	yes	0	0	0.01	326.9
50	$b = 5$ 桶	4	最优	yes	0	0	0.02	473.5
50	$b = 5$ 桶	5	最优	yes	0	0	0.01	526.7
50	$b = 5$ 桶	6	时间限制	–	0	8.3e-03	0.02	600.2
50	$b = 5$ 桶	7	时间限制	–	0	6.5e-03	0.02	600.1
50	$b = 5$ 桶	8	最优	yes	0	0	0.02	311.9
50	$b = 5$ 桶	9	最优	yes	0	0	0.02	349.4
50	$b = 5$ 桶	10	时间限制	–	0	5.8e-03	0.02	600.0
50	前五名列表	1	最优	yes	0	0	0.02	0.7
50	前五名列表	2	最优	yes	0	0	0.00	0.9
50	前五名列表	3	最优	yes	0	0	0.00	1.1
50	前五名列表	4	最优	yes	0	0	0.00	0.8
50	前五名列表	5	最优	yes	0	0	0.02	0.8
50	前五名列表	6	最优	yes	0	0	0.02	0.8
50	前五名列表	7	最优	yes	0	0	0.00	0.7
50	前五名列表	8	最优	yes	0	0	0.01	1.1
50	前五名列表	9	最优	yes	0	0	0.01	0.7
50	前五名列表	10	最优	yes	0	0	0.01	1.2

参考文献

Ailon, N., 2010年。部分排名、p-评分和前 m 名列表的聚合。Algorithmica 57, 284–300。

Ailon, N., Charikar, M., Newman, A., 2008. 不一致信息聚合：排名与聚类。《美国计算机协会会刊》55, 23:1–23:27。

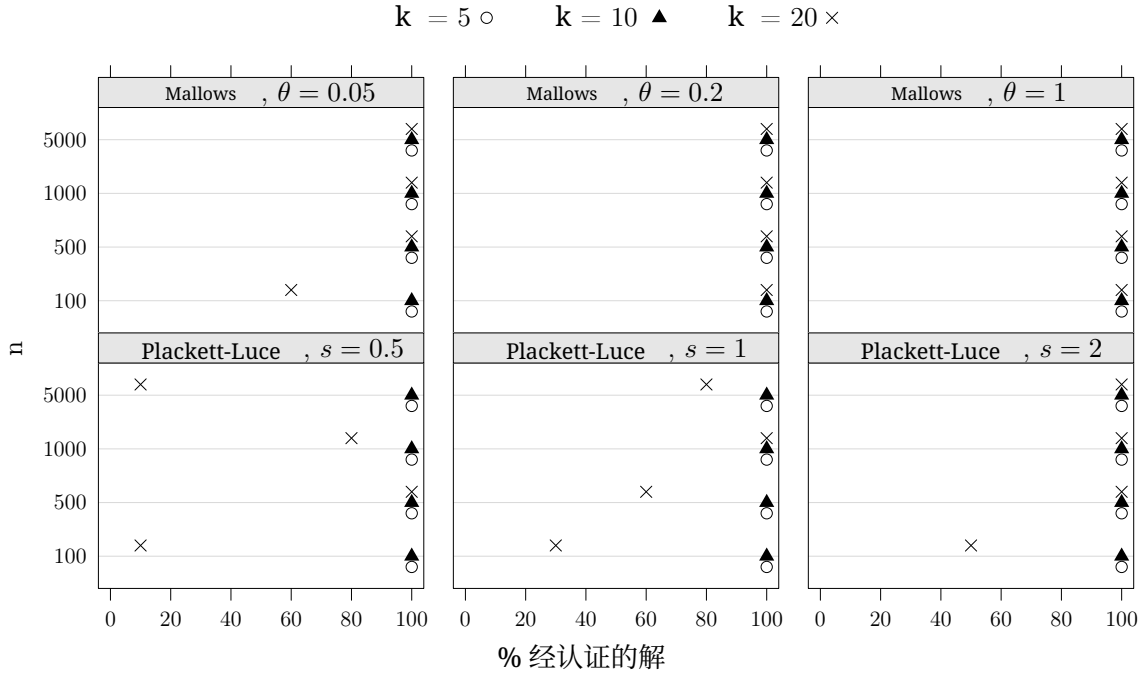
Akbari, S., Escobedo, A.R., 2021. 非严格排名下 Kemeny 排名聚合的下界，载于：2021 IEEE 计算智能研讨会系列（SSCI），IEEE。第 1–8 页。Akbari, S., Escobedo, A.R., 2023a. 近似孔多塞划分：求解大规模排名聚合问题。《计算机与运筹学》153, 106164。Akbari, S., Escobedo, A.R., 2023b. 超越 Kemeny 排名聚合：一种用于带并列的稳健排名聚合的可参数化惩罚框架。《Omega》119, 102893。Aledo, J.A., Domínguez, C., Jaime-Alcántara, J.d.D., Landete, M., 2025a. 一种基于优化的方法，用于生成弱序输出的排名聚合。arXiv 预印本 arXiv:2511.19345（第 2 版，2026年7月）。Aledo, J.A., Gámez, J.A., Rosete, A., 2017. 桶序问题求解中的乌托邦。《决策支持系统》97, 69–80。Aledo, J.A., Gámez, J.A., Rosete, A., 2018. 运用进化策略处理排名聚合问题：以最优桶序问题为例。《欧洲运筹学杂志》270, 982–998。Aledo, J.A., Gámez, J.A., Rosete, A., 2021. 一种高度可扩展的弱排名聚合算法。《信息科学》570, 144–171。Aledo, J.A., Gámez, J.A., Rosete, A., 2025b. 部分排名聚合的共识集：以最优桶序集合问题为例。《信息融合》124, 103353。doi:10.1016/j.inffus.2025.103353。Ali, A., Meilă, M., 2012. Kemeny 排名实验：何时采用何种方法？《数学社会科学》64, 28–40。Alvo, M., Yu, P.L.H., 2014. 《排名数据的统计方法》。施普林格，纽约。Amodio, S., D’Ambrosio, A., Siciliano, R., 2016. 在 Kemeny 公理化方法下处理弱排名和部分排名时识别中位排名的精确算法。《欧洲运筹学杂志》249, 667–676。Andrews, G.E., 1976. 《分拆理论》。第 2 卷，《数学及其应用百科全书》。艾迪生-韦斯利，马萨诸塞州雷丁。Andrieu, P., Cohen-Boulakia, S., Couceiro, M., Denise, A., Pierrot, A., 2023. 一种统一的排名聚合框架，可恰当地、高效地聚合任意类型的排名。《国际近似推理杂志》162, 109035。doi:10.1016/j.ijar.2023.109035。Azzini, I., Morandi, N., Munda, G., 2027. 改进 MorK 算法以识别 Kemeny 中位排名。《欧洲运筹学杂志》336, 605–616。Azzini, I., Munda, G., 2020. 一种识别 Kemeny 中位排名的新方法。《欧洲运筹学杂志》281, 388–401。Badal, P.S., Das, A., 2018. 运用子迭代收敛的 Kemeny 排名问题高效算法。《计算机与运筹学》98, 198–210。Badal, P.S., Singh, R., 2026. 处理 Kemeny 排名聚合中并列问题的高效启发式算法。《运筹学视角》17, 100413。doi:10.1016/j.orp.2026.100413。

Barthélemy, J.P., Guénoche, A., Hudry, O., 1989. 中位线性序：启发式方法与分支定界算法。《欧洲运筹学杂志》42, 313–325. doi:10.1016/ 0377-2217(89)90442-6. Bartholdi, J., Tovey, C.A., Trick, M.A., 1989. 有些投票方案很难判定谁赢得了选举。《社会选择与福利》6, 157–165. Brancotte, B., Yang, B., Blin, G., Cohen-Boulakia, S., Denise, A., Hamel, S., 2015. 含并列名次的排序聚合：实验与分析。《VLDB基金会会议录》8, 1202–1213. Brown, S.R., 1980. 《政治主观性：Q方法论在政治学中的应用》。耶鲁大学出版社，纽黑文。Burgoyne, J.A., Spijkervet, J., Baker, D.J., 2023. 衡量欧洲歌唱大赛：一个面向现实世界音乐信息检索的动态数据集，载于：第24届国际音乐信息检索学会会议论文集，米兰。第817–823页。Ceberio, J., Mendiburu, A., Lozano, J.A., 2015. 重新审视线性排序问题。《欧洲运筹学杂志》241, 686–696. Cohen-Boulakia, S., Denise, A., Hamel, S., 2011. 利用中位数为生物数据生成共识排序，载于：科学与统计数据库管理（SSDBM 2011），施普林格，柏林，海德堡。第73–90页。doi:10.1007/978-3-642-22351-8_5. Cook, W.D., 2006. 序数偏好排序中的基于距离与临时共识模型。《欧洲运筹学杂志》172, 369–385. Cook, W.D., Golany, B., Penn, M., Raviv, T., 2007. 根据评审者的部分序数排名创建提案的共识排序。《计算机与运筹学》34, 954–965. Cook, W.D., Saipe, A.L., 1976. 优先级规划的委员会方法：中位排序法。《运筹学研究中心学报》18, 337–351. Cook, W.D., Seiford, L.M., 1978. 优先级排序与共识形成。《管理科学》24, 1721–1732. Critchlow, D.E., 1985. 《部分排序数据分析的度量方法》。第34卷《统计学讲义》。施普林格，纽约。D’Ambrosio, A., Amodio, S., Mazzeo, G., Albano, A., Plaia, A., 2026年。ConsRank：根据 Kemeny 公理化方法计算中位排名。R软件包版本3.0。网址：https://CRAN.R-project.org/package=ConsRank。D’Ambrosio, A., Iorio, C., Staiano, M., Siciliano, R., 2019年。中位数约束桶序排序聚合。《计算统计学》34, 787–802. doi:10.1007/s00180-018-0858-z. D’Ambrosio, A., Mazzeo, G., Iorio, C., Siciliano, R., 2017年。一种基于 Kemeny 公理化方法寻找中位排名的差分进化算法。《计算机与运筹学》82, 126–138. Dwork, C., Kumar, R., Naor, M., Sivakumar, D., 2001. 面向万维网的排序聚合方法，载于：第10届万维网国际会议论文集，美国计算机协会，纽约。第613–622页。Emond, E.J., Mason, D.W., 2000. 一种新的高层决策支持技术。技术报告。国防部，运筹研究部。渥太华，加拿大。

Emond, E.J., Mason, D.W., 2002年。一种新的秩相关系数及其在共识排序问题中的应用。《多准则决策分析杂志》11, 17–28. 欧盟委员会，2012。欧洲晴雨表 55.2（2001年5月至6月）。GESIS 数据档案馆，科隆。ZA3509 数据文件版本1.0.1，doi:10.4232/1.10943. Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., Vee, E., 2004. 比较并聚合含并列的排名，载于：第23届 ACM SIGMOD-SIGACT-SIGART 数据库系统原理研讨会论文集，第47–58页。doi:10.1145/1055558.1055568. Fagin, R., Kumar, R., Mahdian, M., Sivakumar, D., Vee, E., 2006. 比较部分排名。《SIAM 离散数学杂志》20, 628–648. Fagin, R., Kumar, R., Sivakumar, D., 2003. 比较前 k 列表。《SIAM 离散数学杂志》17, 134–160. Fisher, W.D., 1958年。关于实现最大同质性的分组。《美国统计学会杂志》53, 789–798. Fligner, M.A., Verducci, J.S., 1986. 基于距离的排序模型。英国皇家统计学会期刊：B辑48, 359–369. Gionis, A., Mannila, H., Puolamäki, K., Ukkonen, A., 2006年。根据数据发现桶排序的算法，载于：第12届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集，美国计算机协会，纽约。第561–566页。Good, I.J., 1975. 允许并列时 n候选项的排列数量。《斐波那契季刊》13, 11–18. Gross, O.A., 1962年。优先排列。《美国数学月刊》69, 4–8. Harper, F.M., Konstan, J. A., 2015年。MovieLens 数据集：历史与背景。美国计算机协会《交互式智能系统汇刊》5, 19:1–19:19. Huangfu, Q., Hall, J.A.J., 2018年。对偶修正单纯形法的并行化。《数学规划计算》10, 119–142. Kamishima, T., 2003. Nantonac 协同过滤：基于排序响应的推荐，载于：第九届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集，第583–588页。Kawashima, D., Sukegawa, N., 2026年。含并列的线性排序问题的孔多塞型性质。arXiv:2609.19593. Kemeny, J.G., 1959年。无数字的数学。《代达罗斯》88, 577–591. Kemeny, J.G., Snell, J.L., 1962年。偏好排名：一种公理化方法，载于：《社会科学中的数学模型》。Blaisdell，纽约，第9–23页。Lipset, S.M., 1963. 民主的价值模式：一项比较分析案例研究。《美国社会学评论》28, 515–531. Lorena, L.H.N., Chaves, A.A., Mauri, G.R., Lorena, L.A.N., 2022年。一种用于含并列和不完整排序的排序聚合的自适应偏置随机键遗传算法，载于：2022 IEEE 进化计算大会（CEC）。doi:10.1109/CEC55065.2022.9870203. Luce, R.D., 1959年。《个体选择行为：理论分析》。Wiley，纽约。

Mallows, C.L., 1957. 非零排名模型 I. Biometrika 44, 114–130. Marden, J.I., 1996.《排名数据分析与建模》。Chapman & Hall，伦敦。Mathieu, C., Mauras, S., 2020. 如何聚合头部列表：通过得分和平均排名实现的近似算法，载于：第31届ACM-SIAM离散算法年度研讨会论文集，SIAM。Mattei, N., Walsh, T., 2013. PrefLib：偏好库，载于：算法决策理论。ADT 2013，施普林格，柏林，海德堡。第259–270页。Phung, X.K., Hamel, S., 2025. 针对Kemeny聚合问题，利用优化的多数规则实现高效的空间缩减技术。arXiv:2506.15097。Plackett, R.L., 1975. 排列分析。英国皇家统计学会期刊。C辑（应用统计）24, 193–202。Rico, N., Vela, C.R., Díaz, I., 2023. 通过利用成为获胜者的必要条件，缩短寻找Kemeny排名所需的时间。《欧洲运筹学杂志》305, 1323–1336。doi:10.1016/j.ejor.2022.07.031。Romano, M., Conversano, C., Siciliano, R., D’Ambrosio, A., 2025. 用于偏好排名的粒子群优化。《数据分析与分类进展》doi:10.1007/s11634-025-00626-9。Schiavinotto, T., Stützle, T., 2004. 线性排序问题：实例、搜索空间分析与算法。《数学建模与算法杂志》3, 367–402。Stanley, R.P., 2012.《计数组合学》第1卷。第2版，剑桥大学出版社，剑桥。Stephenson, W., 1935. 因子分析技术。《自然》136, 297。Ukkonen, A., Puolamäki, K., Gionis, A., Mannila, H., 2009. 一种计算桶排序的随机近似算法。《信息处理快报》109, 356–359。Watts, S., Stenner, P., 2012.《开展Q方法论研究：理论、方法与诠释》。Sage，伦敦。Yoo, Y., Escobedo, A.R., 2021. 一种新的二元规划公式及其用于Kemeny排名聚合的社会选择性质。《决策分析》18, 296–320。doi:10.1287/deca.2021.0433。Yoo, Y., Escobedo, A.R., Skolfield, J.K., 2020. 一种用于比较和聚合非严格及不完整排名的新相关系数。《欧洲运筹学杂志》285, 1025–1041。doi:10.1016/j.ejor.2020.02.027。Young, H.P., Levenglick, A., 1978. 孔多塞选举原则的一致扩展。《SIAM应用数学杂志》35, 285–300。doi:10.1137/0135023。Zabala, A., 2014. qmethod：一个利用Q方法论探索人类观点的软件包。《R期刊》6, 163–173。van Zuylen, A., Williamson, D.P., 2009. 用于受约束排名和聚类问题的确定性枢轴算法。《运筹学数学》34, 594–620。

(a) 经认证的解



(b) 精确算法的计算时间（经认证的解）

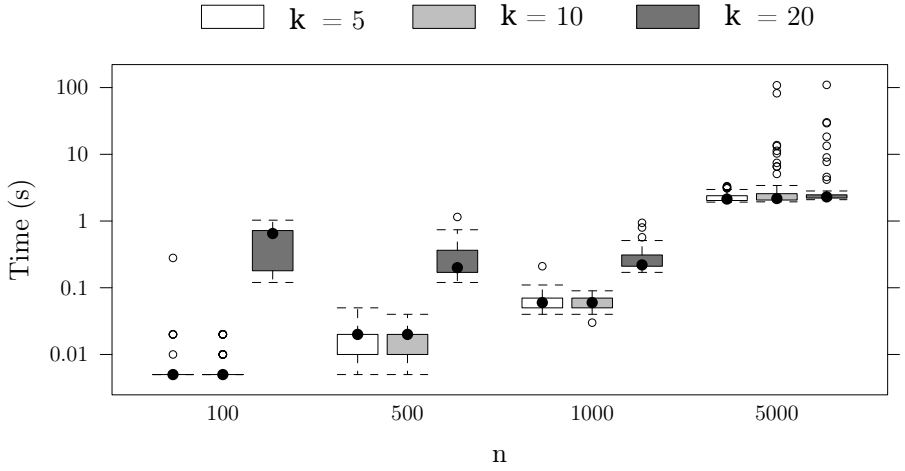


图 1：模拟，中位数 top-k 列表（$m = 500$ 名评审者提供 top-20 列表，每种设置下有 10 个数据集）。(a) 第 5.6 节精确算法认证的解所占百分比。(b) 精确算法对已认证解的计算时间，涵盖所有模型；低于计时器分辨率（0.01 秒）的时间以 0.005 秒表示。

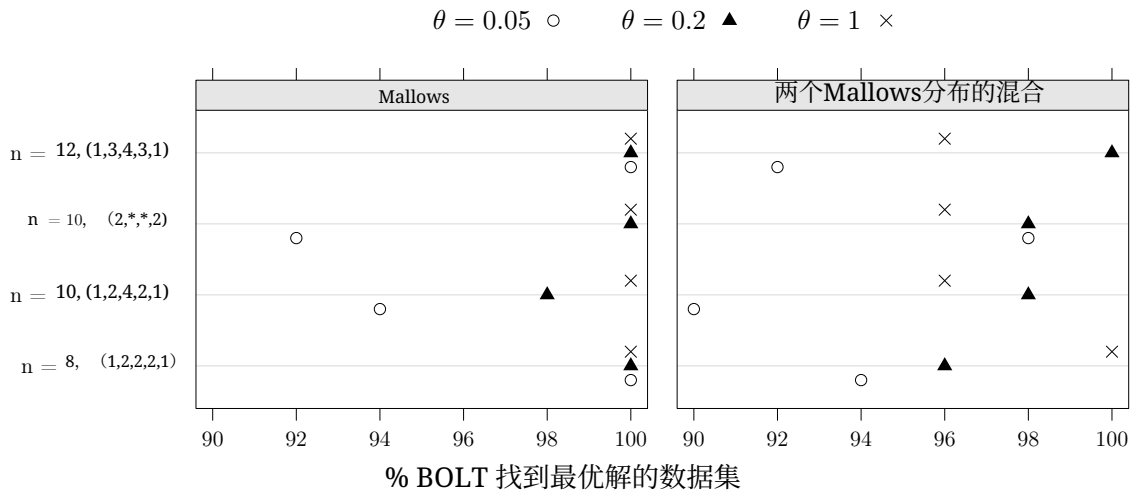
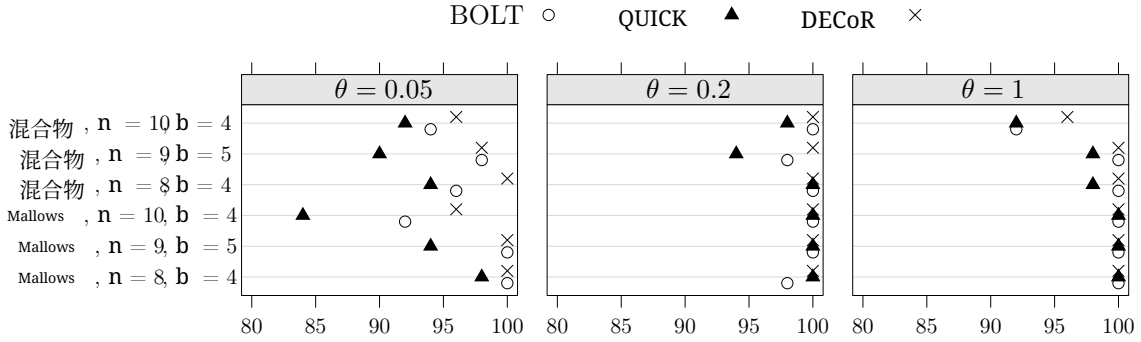


图2：模拟，指定模式（$m = 100$ 评审者，每种设置下50个数据集）：通过枚举子全集$\mathcal{Z}^{n\backslash\nu}$计算得出的BOLT找到最优解的数据集百分比。

(a) 小规模 n：通过枚举求得精确最优解



(b) 较大的 n：三种算法中的最佳解

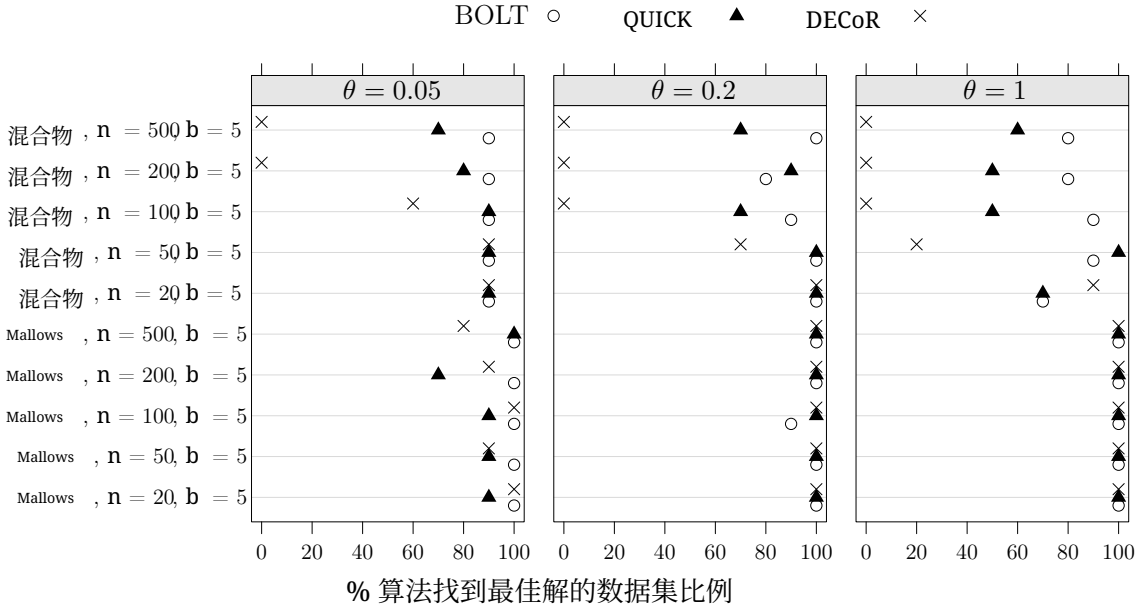


图 3：模拟，MCBO 中包含 b 个自由大小的桶（$m = 100$ 位评审者）。(a) 通过枚举计算各算法找到最优解的数据集所占百分比（每种设置下有 $\mathcal{Z}^{n\backslash b}$ 个数据集，共 50 个）。(b) 各算法在三个算法中找到最佳解的数据集所占百分比（每种设置下有 10 个数据集）。