中国科学: 信息科学 2022 年 第 52 卷 第 6 期: 935–946

SCIENTIA SINICA Informationis

《中国科学》杂志社
SCIENCE CHINA PRESS

理论计算机科学专题 · 论文

无权重物品的几乎无忌妒分配

张智杰[1,2†], 应东昊[3†], 张家琳[1,2*], 孙晓明[1,2]

1. 中国科学院计算技术研究所, 北京 100080, 中国
2. 中国科学院大学, 北京 100080, 中国
3. Industrial Engineering and Operations Research Department, University of California, Berkeley, Berkeley 94704, USA

* 通信作者. E-mail: zhangjialin@ict.ac.cn

† 同等贡献

收稿日期: 2021–12–31; 接受日期: 2022–02–24; 网络出版日期: 2022–06–06

国家自然科学基金 (批准号: 61832003, 61872334) 和中国科学院战略性先导科技专项 A 类 (批准号: XDA27000000) 资助项目

摘要 公平分配研究如何把 m 个不可分的物品公平地分配给 n 个玩家. 每个玩家关于物品有一个可加的估值函数. 物品是无权重的, 如果他们的取值范围为 $\{1, 0, -1\}$. 非负估值的物品称为奖品, 非正估值的物品称为苦差. 本文考虑寻找无权重物品的分配, 并满足 "相差任意物品下是无忌妒的" (envy-free up to any item, EFX_0). EFX_0 是本领域内最受关注的公平性度量. 一般可加估值函数下的 EFX_0 分配的存在性仍然是开放的. 本文提出寻找无权重物品的 EFX_0 分配的多项式时间算法. 为了达到这个目的, 本文分别提出了寻找无权重奖品和无权重苦差的 EFX_0 分配的算法. 然后, 通过将二者小心地结合起来, 本文得到了最终的算法. 本文的结果完整刻画了无权重情况下寻找 EFX_0 分配的解决方案.

关键词 公平分配, 无忌妒性, 不可分物品, 奖品, 苦差

1 引言

本文研究如何在 n 个玩家间公平地分配 m 个不可分的物品. 每个玩家关于物品的每个子集各自有一个估值. 本文假设所有玩家的估值函数是可加的. 当在所有玩家的眼里物品的估值是非负时, 它们被称为奖品. 当在所有玩家的眼里物品的估值是非正时, 它们被称为苦差.

奖品的公平分配是近些年来经济学和计算机科学的核心研究课题之一. 已有文献发展了许多概念来衡量分配的公平性. 其中之一是所谓的无忌妒性 (envy-freeness, EF). 在一个分配当中, 玩家 i 忌妒玩家 j, 如果他认为他分配到的部分严格小于玩家 j 分配到的部分. 一个分配是无忌妒的, 如果玩家之间不会互相忌妒. 然而, 无忌妒性对于不可分物品来说是不可能做到的 (只需考虑在两个玩家之间分配一个物品的例子即可明白这点). 因此, 文献中相继提出了无忌妒性的两种弱化版本: 相差某个物

引用格式: 张智杰, 应东昊, 张家琳, 等. 无权重物品的几乎无忌妒分配. 中国科学: 信息科学, 2022, 52: 935–946, doi: 10.1360/SSI-2021-0449
Zhang Z Z, Ying D H, Zhang J L, et al. Almost envy-free allocations of unweighted items (in Chinese). Sci Sin Inform, 2022, 52: 935–946, doi: 10.1360/SSI-2021-0449

© 2022《中国科学》杂志社

品下的无忌妒性 (envy-freeness up to one item, EF1) [1] 和相差任意物品下的无忌妒性 (envy-freeness up to any item, EFX) [2]. 大致地说, 一个分配是 EF1 的, 如果对于任意 i, j, 在拿走分配给玩家 j 的某个奖品后, 玩家 i 不再忌妒玩家 j. 一个分配是 EFX 的, 如果对于任意 i, j, 在拿走分配给玩家 j 的任一非零奖品后, 玩家 i 不再忌妒玩家 j. 后来, 通过移除 EFX 定义中 "非零" 物品的要求, Plaut 和 Roughgarden [3] 引入了 EFX 的一个强化版本, 在文献 [4] 中也被称为 EFX_0.

苦差的公平分配最近也收到了一些关注 [5,6]. 上述所有公平性概念都可以很容易地在这个场景下定义. 一个更一般的场景是分配混合物品. 在这个场景下, 不同的物品对于不同的玩家可以同时取正值、零值和负值.

为了拓展研究者关于物品公平分配的理解, 本文研究无权重混合物品的分配, 这些物品的取值范围是 $\{1, 0, -1\}$, 刻画了人们对于物品最基本的三种态度 ("喜欢"、"无感"、"讨厌"). 现实生活中的许多对象适用于这种刻画. 例如, 在学术会议审稿阶段审稿人选择目标稿件时, 对于特定的论文, 他们态度通常可以分为 "我想评审此论文"、"我可以评审此论文", 以及 "我不想评审此论文".

尽管关于 EF1 分配的存在性有许多正面的结果 [7,8], EFX/EFX_0 分配的存在性自它们被引入以来就是开放的. 对于无权重的情况, EFX 退化为 EF1, 因此是存在的 (见文献 [8,9]). 然而, 即使对于取值为 $\{0, 1\}$ 的无权重奖品, EFX_0 的存在性最近才刚刚被证明 [10]. 而对于取值为 $\{1, 0, -1\}$ 的混合物品的 EFX_0 分配的存在性, 现存文献尚未有系统的讨论. 本文将证明对于无权重的混合物品, EFX_0 分配是存在的.

1.1 本文贡献

本文首先将关于奖品的 EFX/EFX_0 的公平性概念推广到了混合物品的情况. 本文给出了一个反例, 证明了最自然的推广是不合理的. 相反, 本文提出了两种度量作为合适的推广. 本文证明了在任意一种度量下, 公平分配必定存在, 并且能够 (通过不同的算法) 在多项式时间内被找到. 为了达到这一目的, 本文首先提出了一个寻找无权重奖品的 EFX_0 分配的算法和一个寻找无权重苦差 (其取值为 $\{0, -1\}$) 的 EFX_0 分配的算法. 接着, 通过小心地结合这两个算法的设计思想, 本文成功处理了无权重混合物品的情形. 这些结果完整刻画了无权重混合物品 EFX_0 分配的存在性问题.

1.2 相关工作

对于不可分物品, EF1 首次在文献 [1] 中引入, 作为无忌妒性的弱化版. 正如文献 [7] 暗自证明的那样, 当玩家具有单调的估值函数时, EF1 分配总是存在的. Caragiannis 等 [2] 引入了 EFX 的概念. EFX 的略微强化版由 Plaut 和 Roughgarden [3] 引入, 且在文献 [4] 中被称为 EFX_0. 研究者做了大量尝试以证明奖品的 EFX/EFX_0 分配的存在性. Plaut 和 Roughgarden [3] 证明了当玩家的估值函数相同或者仅仅只有两个玩家时, EFX_0 分配是存在的. 他们也指出了, 当玩家对于物品的估值可加且排序相同时, 可以在多项式时间内找到一个 EFX_0 分配. 最近, 研究者证明了当只有三个玩家 [11], 或者奖品是二值时 [10], 奖品的 EFX_0 分配是存在的. 另一条研究线路考虑近似的公平性概念, 称为 α-EFX_0. Plaut 和 Roughgarden [3] 证明了对于次可加估值函数, $\frac{1}{2}$-EFX_0 分配总是存在的. 他们的算法不是多项式时间的, 后来被 Chan 等 [12] 改进成了一个多项式时间算法. 最近, Amanatidis 等 [13] 打破了 $\frac{1}{2}$ 的界限, 并提出了一个多项式时间算法, 对于可加估值函数, 能够得到一个 0.618-EFX_0 分配. 前人工作也考虑了部分分配, 即允许存一些未分配的奖品. Caragiannis 等 [14] 证明了对于可加函数, 存在一个 EFX_0 部分分配, 其纳什社会福利至少是最优解的 $\frac{1}{2}$. Chaudhury 等 [15] 证明了对于一般的估值函数, 存在一个 EFX_0 的部分分配, 使得未分配的奖品少于 n 个. 尽管存在大量关于 EFX_0 分配的结果, 在

可加估值函数下, 它的存在性仍然是开放的.

苦差的公平分配在文献 [5,6,16,17] 中被研究. 正如文献中所记载的, 这不等价于奖品的分配. 大部分关于奖品 EFX_0 分配的结果对于苦差不再成立. 最近, Aziz 等[8] 把 EF1 的概念推广到了混合物品的情况上, 并且提出了寻找混合物品 EF1 分配的多项式时间算法. Aleksandrov 和 Walsh[9] 提出了一个寻找无权重混合物品 EFX (而不是 EFX_0) 分配的算法.

本文针对无权重物品的算法使用忌妒图这个工具, 这是一项在本领域内使用最广的技术. 简单地说, 本文发现了一种聪明的方式更新忌妒图, 每次把一个新的物品添加进当前的分配当中. 作为对比, 文献 [10] 中的算法基于最大匹配, 并且不依赖于忌妒图. 二者并不能互相推出. 因此本文的算法具备独立的研究兴趣.

最后, 公平分配领域中的另一类研究对象是可分物品. 这类物品可以被任意分割为若干部分, 并以某种公平的方式分配给玩家. 针对可分物品, 也可以定义诸如无忌妒性之类的公平性度量, 并设计算法求解. 可分物品与不可分物品的分配方法十分不同, 研究者们对此进行了大量的研究, 详见文献 [18~24].

2　模型

公平分配问题要求以公平合理的方式把 m 个不可分的物品 $M=\{g_1,g_2,\ldots,g_m\}$ 分配给 n 个玩家 $N=[n]=\{1,2,\ldots,n\}$. 本文使用记号 $\pi=(X_1,X_2,\ldots,X_n)$ 来表示物品的分配, 其中 $\forall i,j\in N$, $X_i\cap X_j=\emptyset$, 且 $X_1\cup X_2\cup\cdots\cup X_n\subseteq M$. 若 $X_1\cup X_2\cup\cdots\cup X_n=M$, π 被称为完全分配; 否则, π 被称为部分分配. 在分配的过程中, 可能出现部分分配, 但最终必须得到一个完全分配. 每个玩家 $i\in N$ 有一个定义在 M 的所有子集上的估值函数 $v_i:2^M\to\mathbb{R}$. 特别地, 对于单个物品 $g\in M$, 本文使用 $v_i(g)$ 表示玩家 i 对物品 g 的估值. 本文仅考虑可加估值函数, 即对于 $X\subseteq M$, $v_i(X)=\sum_{g\in X}v_i(g)$. 此时, 物品 $g\in M$ 可以使用向量 $(v_1(g),v_2(g),\ldots,v_n(g))\in\mathbb{R}^n$ 描述. 如果对于任意 $i\in N$, $g\in M$, 有 $v_i(g)\geqslant 0$, 那么 M 中的物品被称为奖品; 如果对于任意 $i\in N$, $g\in M$, $v_i(g)\leqslant 0$, 那么 M 中的物品被称为苦差; 如果 M 既包含奖品又包含苦差, 或者包含三值的物品, 那么 M 中的物品被称为混合物品. 称玩家 $i\in N$ 拥有二元估值, 如果 M 是奖品, 且 $v_i(g)\in\{0,1\},\forall g\in M$, 或者 M 是苦差, 且 $v_i(g)\in\{0,-1\},\forall g\in M$. 称玩家 $i\in N$ 拥有三元对称估值, 如果 $v_i(g)\in\{-1,0,1\},\forall g\in M$. 如果所有玩家 i 都拥有二元估值或者三元对称估值, 那么问题是无权重的. 本文考虑无权重可加估值函数下的公平分配问题.

2.1　公平性度量

本小节首先介绍若干文献中定义的公平性概念, 包含无忌妒性及相关定义.

定义1 (混合物品的无忌妒性, EF)　称物品的一个分配 $\pi=(X_1,X_2,\ldots,X_n)$ 是无忌妒的 (envy-free, EF), 如果对于任意 $i\neq j$, $v_i(X_i)\geqslant v_i(X_j)$.

无忌妒性要求每位玩家都认为自己分到的物品的价值是最高的. 然而, 在许多情况下这是不可能的. 例如, 考虑两个人分一个物品的例子. 因此, 本文主要关注无忌妒性的弱化版本, 称为相差任意物品下的无忌妒性. 本文将对奖品、苦差和混合物品分别定义这一概念.

定义2 (奖品的 EFX_0 分配)　称奖品的一个分配 $\pi=(X_1,X_2,\ldots,X_n)$ 在相差任意物品下是无忌

妒的 (envy-free up to any item, EFX_0) [1], 如果对于任意 $i \neq j$ 和 $g \in X_j$, $v_i(X_i) \geqslant v_i(X_j \setminus \{g\})$.

定义3 (苦差的 EFX_0 分配)　称苦差的一个分配 $\pi = (X_1, X_2, \ldots, X_n)$ 在相差任意物品下是无忌妒的 (EFX_0), 如果对于任意 $i \neq j$ 和 $g \in X_i$, $v_i(X_i \setminus \{g\}) \geqslant v_i(X_j)$.

接着, 本文考虑混合物品的 EFX_0 分配, 并引入两种不同的定义. 它们分别推广了定义 2 和 3. 为了区分它们, 后者被称为对偶 EFX_0 或者 DEFX_0. 本文将证明, 在三元对称可加估值函数下, 混合物品的 EFX_0 分配和 DEFX_0 分配总是存在的.

定义4 (混合物品的 EFX_0 分配)　称分配 $\pi = (X_1, X_2, \ldots, X_n)$ 在相差任意物品下是无忌妒的 (EFX_0), 如果对于任意 $i \neq j$,

- 对于所有满足 $v_i(g_1) \geqslant 0$ 的物品 $g_1 \in X_j$, $v_i(X_i) \geqslant v_i(X_j \setminus \{g_1\})$; 且
- 对于所有满足 $v_i(g_2) < 0$ 的物品 $g_2 \in X_i$, $v_i(X_i \setminus \{g_2\}) \geqslant v_i(X_j)$.

定义5 (混合物品的 DEFX_0 分配)　称分配 $\pi = (X_1, X_2, \ldots, X_n)$ 在相差任意物品下是对偶无忌妒的 (dual envy-free up to any item, DEFX_0), 如果对于任意 $i \neq j$,

- 对于所有满足 $v_i(g_1) > 0$ 的物品 $g_1 \in X_j$, $v_i(X_i) \geqslant v_i(X_j \setminus \{g_1\})$; 且
- 对于所有满足 $v_i(g_2) \leqslant 0$ 的物品 $g_2 \in X_i$, $v_i(X_i \setminus \{g_2\}) \geqslant v_i(X_j)$.

定义 4 和 5 均要求有一个方向必须拿走非零物品. 那么, 为什么不把混合物品的 EFX_0 分配定义为同时满足定义 2 和 3 的条件, 从而合并定义 4 和 5 呢? 实际上, 尽管乍看之下这很合理, 下述例子证明了这样的分配在一般情况下是不存在的. 因此, 这样的定义并不合适.

例1　假定 $N = \{1, 2\}$, $M = \{g_1 = (1, 1), g_2 = (0, 0)\}$. 由对称性, 不妨设 g_1 总是被分配给玩家 1. 如果实际的分配是 $X_1 = \{g_1, g_2\}, X_2 = \emptyset$, 那么 $v_2(X_1 \setminus \{g_2\}) = 1 > 0 = v_2(X_2)$. 如果实际的分配是 $X_1 = \{g_1\}, X_2 = \{g_2\}$, 那么 $v_2(X_2 \setminus \{g_2\}) = 0 < 1 = v_2(X_1)$. 不论何种情况, 定义 2 和 3 都至少有一个被违背了.

2.2　算法中使用的记号

本文的算法需要对物品进行分类处理. 形式上, 定义 $M_+ = \{g \in M : \max_{i \in N} v_i(g) = 1\}$, $M_0 = \{g \in M : \max_{i \in N} v_i(g) = 0\}$, 以及 $M_- = \{g \in M : \max_{i \in N} v_i(g) = -1\}$. 显然, 在三元对称估值下, 三者不相交, 且 $M_+ \cup M_0 \cup M_- = M$. 在二元估值下, M_- 或者 M_+ 是空集. 固定 $g \in M$, 令 $\mathrm{ONE}_g = \{i \in N : v_i(g) = 1\}$, 以及 $\mathrm{ZERO}_g = \{i \in N : v_i(g) = 0\}$.

本文的算法需要在分配物品时指定玩家的顺序. 令 $\mathcal{I} = (1, 2, \ldots, n)$ 表示默认序, 序号更小的玩家享有更高的优先级. 假定 $N_1 = \{i_1, i_2, \ldots, i_k\}$ 和 $N_2 = \{j_1, j_2, \ldots, j_\ell\}$ 是两个不相交的玩家集合, 满足 $i_1 < i_2 < \cdots < i_k$ 和 $j_1 < j_2 < \cdots < j_\ell$. 令 $\mathcal{I}(N_1)$ 表示顺序 $(i_1, i_2, \ldots, i_k)$. 相反地, 令 $\mathcal{I}(N_1)^{\mathrm{rev}}$ 表示 $\mathcal{I}(N_1)$ 的逆序, 即 $(i_k, i_{k-1}, \ldots, i_1)$. 令 $(\mathcal{I}(N_1), \mathcal{I}(N_2))$ 表示 $\mathcal{I}(N_1)$ 和 $\mathcal{I}(N_2)$ 的组合序, 即 $(i_1, i_2, \ldots, i_k, j_1, j_2, \ldots, j_\ell)$. 对于人为定义的顺序, 本文使用记号 $\mathcal{O}$ 表示, 以区别于默认序 $\mathcal{I}$. 本文使用 $\mathcal{O}_i$ 指代 $\mathcal{O}$ 中的第 i 个玩家.

3　二元可加估值函数下的算法

本节提出两个在二元可加估值函数下寻找 EFX_0 分配的算法. 第 3.1 小节介绍第一个算法, 输出无权重奖品的 EFX_0 分配. 第 3.2 小节介绍第二个算法, 输出无权重苦差的 EFX_0 分配.

1) 前人工作通常关注 EFX_0 的一个弱化版本, 定义为对于任意 $i \neq j$ 和所有使得 $v_i(g) > 0$ 的 $g \in X_j$, $v_i(X_i) \geqslant v_i(X_j \setminus \{g\})$. 为了避免混淆, 本文称这一性质为 EFX.

3.1　寻找奖品的 $\mathbf{EFX_0}$ 分配

本小节提出的算法用于寻找无权重奖品的 EFX_0 分配, 如算法 1 所示. 回顾可知, 奖品的 EFX_0 分配要求对于任意 $i \neq j$ 和 $g \in X_j$, $v_i(X_i) \geqslant v_i(X_j \setminus \{g\})$. 算法从空分配开始, 首先运行 $|M_+|$ 轮把 M_+ 中的奖品分配给玩家. 在每一轮中, M_+ 中一个未分配的奖品 g 会被加入到当前的部分分配中, 同时保证更新后的分配仍然是 EFX_0 的. 为了达到这个目的, 算法采取了一个叫做分层忌妒图的工具, 这个工具是文献 [7] 中忌妒图的推广. 令 G_π 表示对应于 (部分) 分配 $\pi = (X_1, \ldots, X_n)$ 的分层忌妒图. G_π 的顶点代表玩家. G_π 中存在有向边 $i \to j$ 当且仅当在 π 中 $v_i(X_i) < v_i(X_j)$, 即玩家 i 忌妒玩家 j. 此外, 顶点被分为若干层, 且顶点 (玩家) i 在第 k 层中当且仅当 $v_i(X_i) = k$. $v_i(X_i)$ 越小, 玩家 i 所在层数越低. 算法的关键在于它总是把一个未分配的奖品 g 分配给在 G_π 中层数最低的玩家 $i_g \in \mathrm{ONE}_g$. 后文将证明这能够确保分配始终是 EFX_0 的. 在 M_+ 中的奖品分配完毕后, 算法将 M_0 中的所有奖品分配给 G_π 中的任意一个源点, 即没有入边的点, 由此完成最终的分配.

算法 1 EFX_0 for goods

```
Input: Agents N = {1, 2, ..., n}, goods M = {g_1, g_2, ..., g_m} = M_+ ∪ M_0, agents' valuations v_i.
Initialize π = (X_1, X_2, ..., X_n) as an empty allocation and let G_π be its layered envy graph;
while M_+ ≠ ∅ do
    Arbitrarily pick g' ∈ M_+;
    Let i_{g'} be the agent in ONE_{g'} that lies in the lowest layer in G_π;
    Initialize i ← i_{g'} and g ← g';
    X_i ← X_i ∪ {g};
    while π is not EFX_0 do
        Arbitrarily pick g* ∈ X_i \ {g} and let X_i ← X_i \ {g*};
        Let i_{g*} be the agent in ONE_{g*} that lies in the lowest layer in G_π;
        i ← i_{g*} and g ← g*;
        X_i ← X_i ∪ {g};
    end while
    M_+ ← M_+ \ {g};
end while
Let agent i_0 ∈ N be a source in G_π and X_{i_0} ← X_{i_0} ∪ M_0;
Output: An EFX_0 allocation π = (X_1, X_2, ..., X_n) of goods M.
```

以下首先证明在 M_+ 中的物品分配完毕后, 所得到的分配是 EFX_0 的. 首先从一些简单的观察开始.

引理1　在算法 1 的外层 **while** 循环过程中, 分配 π 始终满足对于所有 $i \in N$, $v_i(X_i) = |X_i|$. 此外, 如果分配 π 是 EFX_0 的, 那么

(1) 如果玩家 i 忌妒玩家 j, 那么对于所有 $g \in X_j$, $v_i(g) = 1$, 且 $v_i(X_i) = |X_i| = |X_j| - 1 = v_i(X_j) - 1$, 因此, 他们在 G_π 中处于相邻的层中;

(2) 分层忌妒图 G_π 是无圈图.

证明　由于在外层 **while** 循环的过程中, 物品 g 总是被分配给 ONE_g 中的玩家, 因此在 π 中, 对于所有 $i \in N$ 和 $g \in X_i$, $v_i(g) = 1$. 由于 v_i 是可加的, 因此 $v_i(X_i) = |X_i|$. 现在假定 π 是 EFX_0 的.

(1) 如果玩家 i 忌妒玩家 j, 那么 $v_i(X_i) < v_i(X_j)$. 如果存在物品 $g \in X_j$ 使得 $v_i(g) = 0$, 那么 $v_i(X_i) < v_i(X_j) = v_i(X_j \setminus \{g\})$. 与 π 是 EFX_0 的矛盾. 因此, 对于所有 $g \in X_j$, $v_i(g) = 1$. 由此可知, $|X_i| = v_i(X_i) < v_i(X_j) = |X_j|$. 另一方面, 由于分配是 EFX_0 的, 因此 $|X_i| = v_i(X_i) \geqslant v_i(X_j \setminus \{g\}) =$

$|X_j|-1$, $v_i(X_i)=|X_i|=|X_j|-1=v_i(X_j)-1$.

(2) 根据 G_π 的定义, 有向边 $i\to j$ 存在当且仅当玩家 i 忌妒玩家 j. 因此, 根据 (1), 只有相邻层的顶点之间才能存在边, 并且由低层指向高层. 因此, G_π 是无圈的.

在算法 1 中, 内层 **while** 循环的每一轮涉及把未分配的奖品 g 分配给 X_i (第 6 行或第 11 行), 且如果当前分配不是 EFX_0 的, 把一个已分配的奖品 $g^*\neq g$ 从 X_i 中拿出来 (第 8 行). 关于这一操作有如下若干观察.

引理2 令 $\pi=(X_1,X_2,\ldots,X_n)$ 为一个 EFX_0 的分配, 令 π' 表示在分配了一个未分配的奖品 g 给 X_i 之后的分配 (第 6 行或第 11 行). 那么, π' 不是 EFX_0 的当且仅当在 G_π 中存在有向边 $j\to i$.

证明 首先证明, 对于任意在 G_π 中层数不低于 i 的玩家 j, 即 $|X_j|\geqslant|X_i|$, j 在 π' 中是 EFX_0 的. 根据引理 1, $v_j(X_j)=|X_j|\geqslant|X_i|\geqslant v_j(X_i)$. 如果 $v_j(X_j)>v_j(X_i)$, 那么在 π' 中 j 显然是 EF 的. 如果 $v_j(X_j)=v_j(X_i)$, 那么 $v_j(g)=1,\forall g'\in X_i$. 因此, 在把 g 分配给 i 后, 如果 $v_j(g)=0$, 那么 j 依然是 EF 的; 如果 $v_j(g)=1$, 那么 j 是 EFX_0 的.

上述论证说明了只需关注在 G_π 中层数比 i 更低的玩家 j, 即 $|X_j|<|X_i|$. 由于算法把 g 分配给了 ONE_g 中层数最低的玩家 i, 因此必然有 $v_j(g)=0$. 由此可得, j 在 π' 中不是 EFX_0 的当且仅当在 π 中 $v_j(X_j)<v_j(X_i)$, 即 $j\to i$.

引理3 令 π 和 π' 如引理 2 中所定义. 如果 π' 不是 EFX_0 的, 令 π'' 表示从 X_i 中取出一个已分配的奖品 $g^*\neq g$ 之后的分配 (第 8 行). 那么, π'' 是 EFX_0 的.

证明 由引理 2, 只有在 π 中忌妒 i 的玩家在 π' 中才不是 EFX_0 的. 令 j 表示这样的玩家, 则 $v_j(g)=0$. 由于 π 是 EFX_0 的, 由引理 1, $v_j(X_j)=v_j(X_i)-1=v_j(X_i\cup\{g\})-1$, 且 $v_j(g^*)=1$. 因此, 把 g^* 从 X_i 中取出后, j 变成了 EF 的. 另一方面, i 在 π 中是 EFX_0 的. 从 π 到 π'' 的过程中, g 被加入到了 X_i 中, 而 g^* 从 X_i 中被取出. 两者对于 i 的价值均为 1. 因此, i 在 π'' 中依然是 EFX_0 的.

引理4 算法 1 的内层 **while** 循环在 $|M_+|$ 轮内结束, 且结束时, 所得到的分配是 EFX_0 的. 这一结论对于外层 **while** 循环也成立.

证明 令 π,π',π'',g,g^*,i 如之前的引理中所定义. 内层 **while** 循环进入下一轮是因为 π' 不是 EFX_0 的. 根据引理 2, 这意味着在 π 中存在玩家 j 忌妒玩家 i. 由引理 1, $v_j(g^*)=1$, 且在 π 中 j 所在的层数低于 i 所在的层数. 此外, 易知所有玩家在 π 和 π'' 中处于相同的层数. 上述观察结合起来能够推出在 π'' 中, 玩家 i_{g^*} 所在的层数严格低于 i 所在的层数. 因此, 在 π'' 中, g^* 会重新分配给一个所在层数严格更低的玩家. 最后, 根据引理 1, 忌妒图是无圈的, 且它最多只有 $|M_+|$ 层, 因此, 内层 **while** 循环在最多 $|M_+|$ 步内会停在图的某个源点上. 对于外层 **while** 循环也是如此.

接着, 本文证明当 M_0 中的奖品分配完毕后, 所得到的分配仍然是 EFX_0 的.

引理5 在 M_0 中的奖品分配给了 G_π 的源点后 (第 15 行), 分配 π 仍然是 EFX_0 的.

证明 由于 i_0 是源点, 因此对于任意 $j\in N$, $v_j(X_j)\geqslant v_j(X_{i_0})$. 根据 M_0 的定义, $v_j(M_0)=0$. 因此, $v_j(X_j)\geqslant v_j(X_{i_0}\cup M_0)$. 因此, 分配仍然是 EFX_0 的.

总结上述引理, 本小节证明了关于算法 1 的如下定理.

定理1 算法 1 在多项式时间内终止, 并返回二元加性估值函数下奖品的 EFX_0 分配.

3.2 寻找苦差的 EFX_0 分配

本小节提出的算法用于寻找无权重苦差的 EFX_0 分配, 如算法 2 所示. 回顾可知, 苦差的 EFX_0 分配要求对于所有 $i\neq j$ 和 $g\in X_i$, $v_i(X_i\setminus\{g\})\geqslant v_i(X_j)$. 算法 2 假定 $|M_-|=k<n$, 由下文的引

理 6, 这不失一般性. 算法首先分配 M_0 中的苦差. 每位玩家依次选择 M_0 的剩余苦差中他认为价值为 0 的部分. 根据 M_0 的定义, M_0 一定能被分配完. 接着, 算法定义顺序 $\mathcal{O}$, 依次把 M_- 中的物品分配给在 $\mathcal{O}$ 中的前 k 个玩家, 并返回最终的分配.

算法 2 EFX_0 for chores

Input: Agents $N=\{1,2,\ldots,n\}$, chores $M=\{g_1,g_2,\ldots,g_m\}=M_0\cup M_-$ with $|M_-|=k<n$, agents' valuations v_i;

1: Initialize $\pi=(X_1,X_1,\ldots,X_n)$ as an empty allocation;
2: **for** $i=1,2,\ldots,n$ **do**
3: $\quad X_i \leftarrow \{g\in M_0 : v_i(g)=0\}$;
4: $\quad M_0 \leftarrow M_0\setminus X_i$;
5: **end for**
6: Let $N_1=\{i\in N : X_i\neq\emptyset\}$, $N_2=N\setminus N_1$, and $\mathcal{O}=(\mathcal{I}(N_2),\mathcal{I}(N_1))$;
7: Evenly distribute the k chores in M_- to agents $\mathcal{O}_1,\mathcal{O}_2,\ldots,\mathcal{O}_k$;

Output: An EFX_0 allocation $(X_1,X_2,\cdots,X_n)$ of chores M.

引理6 设 π 是一个部分分配, 且 g_1 和 g_2 是两个未分配的苦差. 如果玩家 i 关于玩家 j 是 $\mathrm{EF}/\mathrm{EFX}_0$ 的, 且 $v_i(g_1)=v_i(g_2)=-1$, 那么在执行 $X_i\leftarrow X_i\cup\{g_1\}, X_j\leftarrow X_j\cup\{g_2\}$ 更新 π 后, 玩家 i 关于玩家 j 仍然是 $\mathrm{EF}/\mathrm{EFX}_0$ 的.

证明 引理的证明是显然的, 因为玩家 i 和 j 分配到的部分同时减少了 1 的价值.

如果 $|M_-|\geqslant n$, 那么可以先从 M_- 里预留 n 个苦差. 接着, 针对剩余部分得到一个 EFX_0 分配. 而后, 把一开始保留的 n 个苦差一人一个分配给每个玩家. 根据引理 6, 最终的分配仍然是 EFX_0 的. 这说明了确实只需要考虑 $|M_-|<n$ 的情况.

下述定理证明了算法 2 的正确性.

定理2 算法 2 在多项式时间内终止, 并返回二元加性估值函数下苦差的 EFX_0 分配.

证明 在算法 2 的 **for** 循环过程中, 由于任意苦差 $g\in M_0$ 都被分配给了 ZERO_g 中的某个玩家, 因此 **for** 循环结束时的分配是 EF 的, 因此是 EFX_0 的. 定义 N_1 为在 **for** 循环中分到了苦差的玩家集合, 定义 N_2 为剩余的玩家集合. 算法的关键在于它是按照顺序 $\mathcal{O}=(\mathcal{I}(N_2),\mathcal{I}(N_1))$ 来分配 M_- 的. 只需证明这样能够保持 EFX_0 的性质即可. 首先, 对于玩家 $i\in N_2$, 因为他只可能在第 7 行分配到一个苦差 $g\in M_-$, 因此 $|X_i|\leqslant 1$, 且当 $|X_i|=1$ 时, $v_i(X_i)=-1$, 因而他显然是 EFX_0 的. 其次, 以下将证明, 对于玩家 $i\in N_2$, 在算法执行第 7 行后, 他仍然是 EF 的. 如果在第 7 行中, i 没有被分配到苦差, 那么 $v_i(X_i)=0$, i 是 EF 的. 如果他被分配到了一个苦差, 那么 $v_i(X_i)=-1$. 对于在顺序 $\mathcal{O}=(\mathcal{I}(N_2),\mathcal{I}(N_1))$ 中优先级比 i 更高的玩家 j, j 也被分配到了 M_- 中的一个苦差, 因此 $v_i(X_j)\leqslant -1$, i 不忌妒 j. 对于在顺序 $\mathcal{O}$ 中优先级比 i 更低的玩家 j, j 一定在 **for** 循环的过程中分配到了苦差, 且在 **for** 循环中, 由于玩家 i 比玩家 j 先分配到苦差. 因此 $v_i(X_j)=-|X_j|\leqslant -1=v_i(X_i)$. 因此, 在任意一种情况下, 玩家 i 都不会忌妒玩家 j.

4 三元对称可加估值函数下的算法

本节处理三元对称可加估值函数的情况, 即 $v_i(g)\in\{1,0,-1\},\forall g\in M,\forall i\in N$. 本节将证明二元可加估值函数下的两个寻找奖品和苦差的 EFX_0 分配的算法可以被推广到这种情况下. 根据引理 6, 不失一般性, 以下假定 $|M_-|<n$.

4.1 寻找混合物品的 $\mathbf{EFX}_0$ 分配

本小节在定义 4 的意义下, 将分配二值奖品的算法 1 推广为下文分配三值混合物品的算法 3. 算法由 3 个阶段组成. 首先, 通过引入 $\tilde{v}_i(g) = \max\{v_i(g), 0\}$ 作为新的估值函数, 算法暂时将 M_+ 中的所有物品转化为奖品, 并通过调用算法 1 来分配 M_+ 中的物品. 接着, 算法依次分配 M_0 中的物品, 确保每个物品 $g \in M_0$ 都被分配给了 ZERO_g 中使得 $v_{i_g}(X_{i_g})$ 最小的玩家 i_g. 而如果有多个这样的玩家, g 会被分配给序号最小的那个. 最后, 算法定义顺序 $\mathcal{O}$, 并将 M_- 中的 k 个物品依次分配给 $\mathcal{O}$ 中的前 k 个玩家.

算法 3 EFX_0 for mixed items

Input: Agents $N = \{1, 2, \ldots, n\}$, items $M = \{g_1, g_2, \ldots, g_m\} = M_+ \cup M_0 \cup M_-$ with $|M_-| = k < n$, agents' valuations v_i;

1: **for all** $i \in N$, $g \in M_+$ **do**
2:     Define $\tilde{v}_i(g) = \max\{v_i(g), 0\}$;
3: **end for**
4: Let $(X_1, X_2, \ldots, X_n) \leftarrow$ Algorithm $1(N, M_+, \tilde{v}_i(\cdot))$;
5: **for all** $g \in M_0$ **do**
6:     Choose $i_g \in \text{ZERO}_g$ with smallest $v_{i_g}(X_{i_g})$, breaking ties by choosing the smallest index;
7:     $X_{i_g} \leftarrow X_{i_g} \cup \{g\}$;
8: **end for**
9: Define order $\mathcal{O}$ using the following rules in order: agent i has higher priority if (1) with larger $v_i(X_i)$, (2) satisfying $v_i(X_i) < |X_i|$, (3) with smaller index i;
10: Evenly distribute the k items in M_- to agents $\mathcal{O}_1, \ldots, \mathcal{O}_k$;

Output: An EFX_0 allocation $(X_1, X_2, \ldots, X_n)$ of mixed items M.

下面的例子深入解释了算法 3 的第 9 行是如何执行的: 假定玩家 1 和 2 满足 $v_1(X_1) = v_2(X_2)$, 玩家 3 和 4 满足 $v_3(X_3) = v_4(X_4) > v_1(X_1)$. 此外, 对于 $i = 1, 2, 3$, $v_i(X_i) = |X_i|$, 且 $v_4(X_4) < |X_4|$. 假定 $|M_-| = k = 3$, 那么在下一行代码中, 根据顺序 $\mathcal{O}$ 的定义, 算法 3 会把 M_- 中的物品依次分配给玩家 4, 3 和 1.

为了证明算法 3 的正确性, 下文将分别考察算法的 3 个阶段, 并且证明在每个阶段结束后, 分配都是 EFX_0 的. 在第一阶段, 算法 3 调用算法 1 分配 M_+. 可以证明如下引理.

引理7 假定 $\pi = (X_1, X_2, \ldots, X_n)$ 是算法 3 第 4 行调用算法 1 后返回的分配. 那么, 在估值 v 下, π 是 EFX_0 的.

证明 由定理 1, 在估值 $\tilde{v}$ 下, π 对于奖品是 EFX_0 的. 由于算法 1 总是把物品 g 分配给 ONE_g 中的玩家, 且根据 M_+ 的定义, 任意物品 $g \in M_+$ 对应的 ONE_g 非空, 因此 $\tilde{v}_i(g) = 1, \forall g \in X_i$. 由于算法只把 $\tilde{v}$ 中的 -1 变成了 0, 因此 $v_i(g) = \tilde{v}_i(g) = 1, \forall g \in X_i$, $v_i(X_i) = |X_i|$. 另一方面, 对于 $i \neq j$, $v_i(X_j) \leqslant \tilde{v}_i(X_j)$. 因此, π 在估值 $\tilde{v}$ 下对于奖品是 EFX_0 的能够推出在估值 v 下定义 4 的第一个条件被满足. 对于第二个条件, 由于 $v_i(g) = 1, \forall g \in X_i$, 因此它是自然成立的. 因此, 在估值 v 下, π 对于混合物品是 EFX_0 的.

在第二阶段, 算法 3 分配 M_0 中的物品. 可以证明如下引理.

引理8 设 π 是算法 3 的第 4 行返回的部分分配, 在算法 3 的 **for** 循环结束后, π 会变成 $M_+ \cup M_0$ 的一个 EFX_0 分配.

证明 首先, 定义 4 的第二个条件自然成立, 因为在 **for** 循环中 $g \in M_0$ 被分配给了 ZERO_g 中的玩家, 这意味着不存在 $g_2 \in X_i$ 使得 $v_i(g_2) < 0$. 因此, 下文只需考虑第一个条件是否成立. 以下将证

明, 在任何一次循环中, 如果在把物品 g 分配给玩家 i_g 之前, 部分分配 $\pi=(X_1,X_2,\ldots,X_n)$ 是 EFX_0 的, 那么在那之后, 更新后的分配 $\pi'=(X_1,\ldots,X_{i_g}\cup\{g\},\ldots,X_n)$ 也是 EFX_0 的.

由于 $i_g\in\mathrm{ZERO}_g$, 有 $v_{i_g}(X_{i_g})=v_{i_g}(X_{i_g}\cup\{g\})$, 因此玩家 i_g 在 π' 中满足定义 4 的第一个条件, 是 EFX_0 的. 对于任何玩家 $j\neq i_g$, 由于 $g\in M_0$, 因此 $v_j(g)\leqslant 0$. 这意味着, 对于任何满足 $v_j(g')\geqslant 0$ 的物品 $g'\in X_{i_g}$, 有 $v_j(X_{i_g}\cup\{g\}\setminus\{g'\})\leqslant v_j(X_{i_g}\setminus\{g'\})\leqslant v_j(X_j)$. 对于物品 g, 当 $v_j(g)=0$ 时, 因为算法从 ZERO_g 中选择的 i_g 具有最低的 $v_{i_g}(X_{i_g})$, 因此 $v_j(X_j)\geqslant v_j(X_{i_g})=v_j(X_{i_g}\cup\{g\}\setminus\{g\})$. 因此, 玩家 j 在 π' 中也是 EFX_0 的. 综上所述, 部分分配 π' 是 EFX_0 的.

在第三阶段, 算法 3 分配 M_- 中的物品. 可以证明如下引理.

引理9　设 $\pi=(X_1,X_2,\ldots,X_n)$ 是算法 3 的 **for** 循环结束后的部分分配, $\tilde{\pi}=(\tilde{X}_1,\ldots,\tilde{X}_n)$ 是算法 3 的第 10 行结束后得到的完全分配. 那么, $\tilde{\pi}$ 是 EFX_0 的.

证明　令 N_t 表示顺序 $\mathcal{O}$ 中的前 k 个玩家, 他们在第 10 行中各自分配到了 M_- 中的一个物品. 因此, $\forall i\in N_t, v_i(\tilde{X}_i)=v_i(X_i)-1$, 而 $\forall i\in N\setminus N_t, v_i(\tilde{X}_i)=v_i(X_i)$. 由于对于任何 $g\in M_-$ 和 $i\in N$, $v_i(g)=-1$, 因此在 $N\setminus N_t$ 中的玩家在分配 $\tilde{\pi}$ 中依然是 EFX_0 的. 此外, 由引理 6, N_t 中的玩家互相之间也是 EFX_0 的. 因此, 只需证明, 在 $\tilde{\pi}$ 中, 任何 N_t 中的玩家 i 关于任何 $N\setminus N_t$ 中的玩家 j 也是 EFX_0 的. 令 $X_i=X_{i+}\cup X_{i0}(X_j=X_{j+}\cup X_{j0})$, 其中 $X_{i+}(X_{j+})$ 是在第 4 行中分配给玩家 $i(j)$ 的物品, $X_{i0}(X_{j0})$ 是在 **for** 循环中分配给玩家 $i(j)$ 的物品.

首先证明对任意满足 $v_i(g)\geqslant 0$ 的 $g\in\tilde{X}_j$, $v_i(\tilde{X}_i)\geqslant v_i(\tilde{X}_j\setminus\{g\})$, 即证 $v_i(X_i)-1\geqslant v_i(X_j\setminus\{g\})$. 由于在顺序 $\mathcal{O}$ 中, i 比 j 具有更高的优先级, 因此 $v_i(X_i)\geqslant v_j(X_j)$. 由于 $v_j(X_j)=v_j(X_{j+})=|X_{j+}|\geqslant v_i(X_{j+})\geqslant v_i(X_j)$, 如果 $v_i(g)=1$, 那么 $v_i(X_i)-1\geqslant v_i(X_j\setminus\{g\})$. 因此, 下面假定 $v_i(g)=0$. 如果 $v_i(X_i)>v_j(X_j)$, 由于 $v_j(X_j)\geqslant v_i(X_j)$, 结论成立. 在 $v_i(X_i)=v_j(X_j)$ 的情况下, 如果 $X_{j0}=\emptyset$, $v_i(X_i)=v_j(X_j)>v_j(X_j\setminus\{g\})\geqslant v_i(X_j\setminus\{g\})$, 同样导出所需结论. 第三种情况是 $v_i(X_i)=v_j(X_j)$ 且 $X_{j0}\neq\emptyset$. 根据 $\mathcal{O}$ 的定义, 有 $X_{i0}\neq\emptyset$ 且 $i<j$. 此外, 条件 $v_i(g)=0$ 意味着 $i\in\mathrm{ZERO}_g$. 由于玩家已分配到的部分的价值在 **for** 循环中保持不变, 根据第 6 行, g 应该被分配给玩家 i 而不是玩家 j. 矛盾. 因此, 定义 3 的第一个条件对于玩家 i,j 成立.

其次, 要证明任意满足 $v_i(g)<0$ 的 $g\in\tilde{X}_i$, $v_i(\tilde{X}_i\setminus\{g\})\geqslant v_i(\tilde{X}_j)$. 这等价于证明 $v_i(X_i)\geqslant v_i(X_j)$. 而上述论证已经说明了 $v_i(X_i)\geqslant v_j(X_j)\geqslant v_i(X_j)$.

总结上述引理, 本节证明了关于算法 3 的如下定理.

定理3　算法 3 在多项式时间内终止, 并返回三元对称可加估值函数下物品的 EFX_0 分配.

注1　算法 3 的第 5 行至最后一行实际上是算法 2 的推广. 如果物品估值限定在 $\{0,-1\}$ 上, 那么 $M_+=\emptyset$. 在 **for** 循环中, 所有玩家都保持 $v_i(X_i)=0$, 因此它退化为了算法 2 的 **for** 循环. 在 **for** 循环后, $v_i(X_i)=0$ 对所有玩家都成立, 因此第 9 行的第一个排序规则是多余的. 集合 $\{i\in N:v_i(X_i)<|X_i|\}$ 等价于 $N_1=\{i\in N:X_i\neq\emptyset\}$. 因此, 算法 3 中的顺序 $\mathcal{O}$ 会退化为算法 2 中的顺序 $\mathcal{O}$.

4.2　寻找混合物品的 $\mathbf{DEFX}_0$ 分配

本小节在定义 5 的意义下, 将分配二值苦差的算法 2 推广为下文分配三值混合物品的算法 4. 算法首先调用算法 2 得到一个 $M_0\cup M_-$ 的 DEFX_0 分配. 它同时记录了算法 2 第 6 行构造的集合 $N_1=\{i\in N:X_i\neq\emptyset\}$ 和 $N_2=N\setminus N_1$, 并基于 N_1 和 N_2 定义了一个新顺序 $\mathcal{O}=(\mathcal{I}(N_1)^{\mathrm{rev}},\mathcal{I}(N_2))$. 最后, 算法依次分配 M_+ 中的物品 g, 并将 g 分配给使得 $v_{i_g}(X_{i_g})$ 最小的玩家 $i_g\in\mathrm{ONE}_g$, 如果存在多个这样的玩家, 则令 i_g 为在 $\mathcal{O}$ 中优先级最高的那个. 下述定理证明了算法 4 的正确性.

算法 4 DEFX$_0$ for mixed items

Input: Agents $N=\{1,2,\ldots,n\}$, items $M=\{g_1,g_2,\ldots,g_m\}=M_+\cup M_0\cup M_-$ with $|M_-|=k<n$, agents' valuations v_i;
1: $(X_1,X_2,\ldots,X_n)\leftarrow$ Algorithm 2 $(N,M_0\cup M_-,v_i)$, and let N_1,N_2 be defined as in line 6 of Algorithm 2;
2: Let $\mathcal{O}=(\mathcal{I}(N_1)^{\mathrm{rev}},\mathcal{I}(N_2))$;
3: **for all** $g\in M_+$ **do**
4: Let i_g be the agent in ONE$_g$ that has the smallest $v_{i_g}(X_{i_g})$. If there are multiple such agents, let i_g be the agent with the highest priority in $\mathcal{O}$;
5: $X_{i_g}\leftarrow X_{i_g}\cup\{g\}$;
6: **end for**
Output: An DEFX$_0$ allocation $(X_1,X_2,\ldots,X_n)$ of mixed items M.

定理4 算法 4 在多项式时间内终止, 并返回三元对称可加估值函数下物品的 DEFX$_0$ 分配.

证明 由定理 2, 算法 2 返回 $M_0\cup M_-$ 的一个 DEFX$_0$ 分配. 下面只需证明算法 4 在 **for** 循环中保持 DEFX$_0$ 的性质. 设 $\pi=(X_1,\ldots,X_n)$ 表示某轮 **for** 循环前的分配, 且 π 是 DEFX$_0$ 的. 在这轮循环中, 物品 $g\in M_+$ 被分配给了玩家 $i=i_g$. 令 $\pi'=(X_1,\ldots,X_i\cup\{g\},\ldots,X_n)$ 表示此轮循环结束后的分配. 只需证明 π' 是 DEFX$_0$ 的.

首先证明, 在 π' 中, 对于任意 $j\neq i$, 玩家 i 关于玩家 j 是 EFX$_0$ 的. 由于在 π 中, 玩家 i 是 DEFX$_0$ 的, 且 $v_i(g)=1$, 因此对于任何满足 $v_i(g')>0$ 的 $g'\in X_j$, $v_i(X_i\cup\{g\})>v_i(X_i)\geqslant v_i(X_j\setminus\{g'\})$. 因此在 π' 中, 定义 5 的第一个条件对于玩家 i 成立. 此外, 对于任何满足 $v_i(g')\leqslant 0$ 的 $g'\in X_i$, 有 $v_i(X_i\cup\{g\}\setminus\{g'\})>v_i(X_i\setminus\{g'\})\geqslant v_i(X_j)$. 因此, 定义 5 的第二个条件对于玩家 i 也成立.

接着, 需要证明在 π' 中, 对于任何 $j\neq i$, 玩家 j 关于玩家 i 是 EFX$_0$ 的.

若 $v_j(X_j)<v_j(X_i)$. 由于 $v_j(X_i)\leqslant v_i(X_i)$, 有 $v_j(X_j)<v_i(X_i)$. 因此 $v_j(g)\leqslant 0$, 否则在第 4 行, g 会被分配给 j 而不是 i. 由于玩家 j 在 π 中是 DEFX$_0$ 的, 因此对于任何满足 $v_j(g')=1$ 的 $g'\in X_i$, $v_j(X_j)\geqslant v_j(X_i\setminus\{g'\})$. 因此在 π' 中, 定义 5 的第一个条件对于玩家 j 成立. 此外, 对于任何满足 $v_j(g')\leqslant 0$ 的 $g'\in X_j$, $v_j(X_j\setminus\{g'\})\geqslant v_j(X_i)\geqslant v_j(X_i\cup\{g\})$. 因此在 π' 中, 定义 5 的第二个条件对于玩家 j 成立. 综上所述, 在 π' 中, j 是 DEFX$_0$ 的.

若 $v_j(X_j)\geqslant v_j(X_i)$. 如果 $v_j(g)\leqslant 0$, 那么 $v_j(X_j)\geqslant v_j(X_i\cup\{g\})$. 因此, 玩家 j 是 EF 的, 因此是 DEFX$_0$ 的. 如果 $v_j(g)=1$ 且 $v_j(X_j)>v_j(X_i)$, 那么 $v_j(X_j)\geqslant v_j(X_i\cup\{g\})$. 因此, 玩家 j 在 π' 中是 EF 的, 因此是 DEFX$_0$ 的. 假定 $v_j(g)=1$ 且 $v_j(X_j)=v_j(X_i)$. 那么对于任何满足 $v_j(g')=1$ 的 $g'\in X_i\cup\{g\}$, $v_j(X_j)=v_j(X_i\cup\{g\}\setminus\{g'\})$; 且对于任何满足 $v_j(g')=-1$ 的 $g'\in X_j$, $v_j(X_j\setminus\{g'\})=v_j(X_i\cup\{g\})$. 因此, 如果可以证明对于任何 $g'\in X_j$, $v_j(g')\neq 0$, 那么在 π' 中, 玩家 j 是 DEFX$_0$ 的.

令 $X_{i+}=\{g\in X_i:v_i(g)=1\}$, $X_{i0}=\{g\in X_i:v_i(g)=0\}$ 且 $X_{i-}=\{g\in X_i:v_i(g)=-1\}$. 用反证法, 假设存在物品 $g^*\in X_j$ 使得 $v_j(g^*)=0$, 那么 $j\in N_1$. 因此, $i\in N_1$ 且在顺序 $\mathcal{I}(N_1)^{\mathrm{rev}}$ 中, i 比 j 享有更高的优先级, 即 $i>j$, 因为否则的话, g 会被分配给 j 而不是 i. 结合算法 2 中 **for** 循环的分配规则, 这些事实能够推出 $X_{i0}\neq\emptyset$ 且 $v_j(X_{i0})=-|X_{i0}|<0$. 另一方面, 由于 $v_j(g)=1$, 有 $v_j(X_j)\geqslant v_i(X_i)$, 否则在算法 4 的第 4 行, g 会被分配给 j 而不是 i. 因此, $v_j(X_j)=v_j(X_i)=v_i(X_i)$. 注意到 $v_i(X_i)=|X_{i+}|-|X_{i-}|$, 且 $v_j(X_i)\leqslant|X_{i+}|-|X_{i0}|-|X_{i-}|$, 因此 $v_j(X_{i+})\leqslant|X_{i+}|$ 且 $v_j(X_{i-})=-|X_{i-}|$. 因此, $|X_{i0}|=0$ 必然成立. 矛盾.

5 结论

本文提出了若干计算无权重混合物品的 EFX_0 分配的多项式时间算法, 即当玩家具有三元对称可加估值函数时. 虽然主要的开放问题是在可加估值下确定 (奖品) 的 EFX_0 分配的存在性, 仍然存在一些与本文研究的问题更相关的开放问题. 例如, 可以考虑三元不对称估值的情况, 即物品具有估值 $\{-\alpha_1, 0, \alpha_2\}$, 其中 α_1, α_2 是两个不相同的定值. 也可以考虑一般二值苦差的分配问题. 由于本文的方法和文献 [10] 中的方法并不能直接解决这些问题, 因此需要发展全新的技术方法求解这些问题.

参考文献

1 Budish E. The combinatorial assignment problem: approximate competitive equilibrium from equal incomes. J Political Economy, 2011, 119: 1061–1103

2 Caragiannis I, Kurokawa D, Moulin H, et al. The unreasonable fairness of maximum Nash welfare. In: Proceedings of the ACM Conference on Economics and Computation, Maastricht, 2016. 305–322

3 Plaut B, Roughgarden T. Almost envy-freeness with general valuations. In: Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, 2018. 2584–2603

4 Kyropoulou M, Suksompong W, Voudouris A A. Almost envy-freeness in group resource allocation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, 2019. 400–406

5 Aziz H, Rauchecker G, Schryen G, et al. Algorithms for max-min share fair allocation of indivisible chores. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, 2017. 335–341

6 Barman S, Krishnamurthy S K. Approximation algorithms for maximin fair division. In: Proceedings of the ACM Conference on Economics and Computation, Cambridge, 2017. 647–664

7 Lipton R J, Markakis E, Mossel E, et al. On approximately fair allocations of indivisible goods. In: Proceedings of the 5th ACM Conference on Electronic Commerce, New York, 2004. 125–131

8 Aziz H, Caragiannis I, Igarashi A, et al. Fair allocation of indivisible goods and chores. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, 2019. 53–59

9 Aleksandrov M, Walsh T. Greedy algorithms for fair division of mixed manna. 2019. ArXiv:1911.11005

10 Amanatidis G, Birmpas G, Filos-Ratsikas A, et al. Maximum Nash welfare and other stories about EFX. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence, Yokohama, 2020. 24–30

11 Chaudhury B R, Garg J, Mehlhorn K. EFX exists for three agents. In: Proceedings of the 21st ACM Conference on Economics and Computation, Virtual Event, 2020. 1–19

12 Chan H, Chen J, Li B, et al. Maximin-aware allocations of indivisible goods. In: Proceedings of the 18th International Conference on Autonomous Agents and Multi Agent Systems, Montreal, 2019. 1871–1873

13 Amanatidis G, Markakis E, Ntokos A. Multiple birds with one stone: beating 1/2 for EFX and GMMS via envy cycle elimination. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, 2020. 1790–1797

14 Caragiannis I, Gravin N, Huang X. Envy-freeness up to any item with high Nash welfare: the virtue of donating items. In: Proceedings of the ACM Conference on Economics and Computation, Phoenix, 2019. 527–545

15 Chaudhury B R, Kavitha T, Mehlhorn K, et al. A little charity guarantees almost envy-freeness. In: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, Salt Lake City, 2020. 2658–2672

16 Lan T, Kao D, Chiang M, et al. An axiomatic theory of fairness in network resource allocation. In: Proceedings of the 29th IEEE International Conference on Computer Communications, San Diego, 2010. 1343–1351

17 Heydrich S, van Stee R. Dividing connected chores fairly. Theor Comput Sci, 2015, 593: 51–61

18 Aziz H, Mackenzie S. A discrete and bounded envy-free cake cutting protocol for four agents. In: Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, Cambridge, 2016. 454–464

19 Aziz H, Mackenzie S. A discrete and bounded envy-free cake cutting protocol for any number of agents. In: Proceedings of IEEE 57th Annual Symposium on Foundations of Computer Science, New Brunswick, 2016. 416–427

20 Procaccia A D. Thou shalt covet thy neighbor's cake. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, 2009. 239–244

21 Edmonds J, Pruhs K. Cake cutting really is not a piece of cake. In: Proceedings of the 17th Annual ACM-SIAM

Symposium on Discrete Algorithms, Miami, 2006. 271–278
22 Even S, Paz A. A note on cake cutting. Discrete Appl Math, 1984, 7: 285–296
23 Bei X, Sun X, Wu H, et al. Cake cutting on graphs: a discrete and bounded proportional protocol. In: Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, Salt Lake City, 2020. 2114–2123
24 Tucker-Foltz J. Thou shalt covet the average of thy neighbors' cakes. 2021. ArXiv:2106.11178

Almost envy-free allocations of unweighted items

Zhijie ZHANG[1,2†], Donghao YING[3†], Jialin ZHANG[1,2*] & Xiaoming SUN[1,2]

1. *Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080*, China*;
2. *University of Chinese Academy of Sciences, Beijing* 100080*, China*;
3. *Industrial Engineering and Operations Research Department, University of California, Berkeley, Berkeley* 94704*, USA*
* Corresponding author. E-mail: zhangjialin@ict.ac.cn
† Equal contribution

Abstract Fair allocation studies how to fairly allocate m indivisible items among n agents. Each agent has an additive valuation function over items, which are unweighted if their values are in $\{1, 0, -1\}$. Those items with non-negative values are called goods, and those with non-positive values are called chores. We consider the problem of finding an allocation of unweighted items which is envy-free up to any item (EFX_0). EFX_0 is one of the most concerned fairness notions in the literature. It remains open whether EFX_0 allocations always exist even under additive valuation functions. In this paper, we present polynomial-time algorithms for finding an EFX_0 allocation of unweighted items. To achieve this, we present algorithms for finding EFX_0 allocations of unweighted goods and unweighted chores separately, which are then carefully merged to deliver the final algorithms. As a result, we provide a complete solution for finding EFX_0 allocations in the unweighted scenario.

Keywords fair allocation, envy-freeness, indivisible items, goods, chores