Computational Complexity of Necessary Envy-freeness

Haris Aziz[a], Ildikó Schlotter[b,c,*], Toby Walsh[a]

[a] *UNSW Sydney, Australia*
[b] *Centre for Economic and Regional Studies, Hungary*
[c] *Budapest University of Technology and Economics, Hungary*

Abstract

We consider the fundamental problem of fairly allocating indivisible items when agents have strict ordinal preferences over individual items. We focus on the well-studied fairness criterion of necessary envy-freeness. For a constant number of agents, the computational complexity of the deciding whether there exists an allocation that satisfies necessary envy-freeness has been open for several years. We settle this question by showing that the problem is NP-complete even for three agents. Considering that the problem is polynomial-time solvable for the case of two agents, we provide a clear understanding of the complexity of the problem with respect to the number of agents.

Keywords: fair division; envy-freeness
JEL: C62, C63, and C78

1. Introduction

When allocating items among agents, a natural and fundamental concern is fairness [2, 5, 9, 12]. We consider the setting in which agents have strict ordinal preferences over the items. The fairness concept we focus on is necessary envy-freeness [3, 6]. An allocation satisfies necessary envy-freeness if for any two agents i and j with allocations I_i and I_j, there exists an injection f from I_j to I_i such that for each item $x \in I_j$, agent i prefers the item $f(x)$ over x. This requirement has been referred to by different terms in the literature including responsive-set (RS) envy-freeness [3], stochastic-dominance (sd) envy-freeness [3], not possible envy-freeness [7] and itemwise envy-freeness [8].

Bouveret et al. [6] considered the computational complexity of checking whether a complete necessary envy-free allocation exists or not; we will call this problem ExistsNEF (precise definitions follow in Section 2). Bouveret et al. [6] proved that ExistsNEF is NP-complete even if the number of items is twice as much as the number of agents. They also showed that the problem

*Corresponding author
Email addresses: haris.aziz@unsw.edu.au (Haris Aziz), ildi@cs.bme.hu (Ildikó Schlotter), toby.walsh@unsw.edu.au (Toby Walsh)

is polynomial-time solvable when the number of agents is two. Since the work of Bouveret et al. [6] in 2010, the complexity of the problem has been open for constant number of agents [3, 6], even though scenarios where the task is to find a fair allocation of items among a small, fixed number of agents is of high practical interest, and has been the focus of considerable research in the area (see, e.g., [1, 8, 10, 11]).

In this paper, we resolve this open problem by showing that ExistsNEF is NP-complete if the number of agents is a constant at least three. This completes our understanding of the computational complexity of ExistsNEF as a function of the number of agents involved; see Table 1. We remark that our result for the case where the number of agents is exactly three was announced in conference paper [4].

	$n = 2$	fixed $n \geq 3$	unbounded n
ExistsNEF	in P [6]	**NP-complete (Thm. 2)**	NP-complete [6]

Table 1: Complexity of ExistsNEF. Our result is in bold font.

2. Preliminaries

Formally, an instance of our problem is a triple (N, I, L), where N is a set of n agents, I a set of indivisible items, and L is a collection of preference lists L^A for each agent $A \in N$. Each preference list L^A is a strict linear ordering over the set I of items.

An assignment π of items to agents is an *allocation*, and π is *complete* if it assigns each item of I to some agent. A complete allocation can be viewed as a partitioning of he items into n bundles with each bundle corresponding to an agent's allocation.

When reasoning about preferences over bundles of items, an agent may be required to express preferences over an exponential number of bundles. A compact way of expressing preferences over bundles is for agents to express preferences over individual items and then extend them over bundles of items with respect to the *responsive set extension*. In this notion, we say that an agent A *prefers* a set I_1 of items over a set I_2 of items if there exists an injection f from I_2 to I_1 such that for each item $x \in I_2$, agent A prefers the item $f(x)$ over x. An allocation is *necessarily envy-free (NEF)* if each agent prefers its own set of items over any set of items allocated to some other agent. Note that a NEF allocation is envy-free for all additive valuations consistent with the ordinal preferences.

Example 1. Suppose we have four items $1, 2, 3$, and 4, and two agents A and B with the following preferences over the items.

$$A: \quad 1 \succ 2 \succ 3 \succ 4$$
$$B: \quad 2 \succ 1 \succ 4 \succ 3$$

In that case, the unique NEF allocation is one in which A gets 1 and 3, while agent B gets 2 and 4.

The central problem we consider in the paper is the ExistsNEF problem. In this problem, the task is to find a complete NEF allocation if it exists, or return 'no' if no such allocation exists.

ExistsNEF	
Input:	A triple (N, I, L) where N is a set of agents, I a set of items, and L is a collection of preference lists for each agent in N.
Question:	Does a complete NEF allocation exist for (N, I, L)?

The complexity of ExistsNEF for a constant number of agents has been an open problem [6, 3]. We prove its NP-completeness first for the case of three agents in Section 3, and then for the case when the number of agents is a fixed integer at least three in Section 4.

Notation. We let $[h] = \{1, 2 \dots, h\}$ for any positive integer h. For a linear ordering $L = (s_1, \dots, s_m)$ over a set $S = \cup_{i \in [m]} s_i$ of items, we define $L(i : j) = (s_i, s_{i+1}, \dots, s_j)$ for any i and j with $1 \leq i \leq j \leq m$. For $X \subseteq S$, we let $L_{|X}$ be the restriction of L to X, and we write $[L_{|X}]$ for the set of elements in $L_{|X}$.

The definition of necessary envy-freeness can be reformulated using Hall's theorem into the following equivalent form, which we will use throughout the paper.

Proposition 1. *For a given set N of agents, a set I of items, and a preference list L^A for each agent $A \in N$, an allocation $\pi : I \to N$ is NEF if and only if for each pair of agents A and B (where $A \neq B$) and index $i \in [|I|]$ we have:*

$$\left|[L^A(1 : i)] \cap \pi^{-1}(A)\right| \geq \left|[L^A(1 : i)] \cap \pi^{-1}(B)\right|.$$

3. Result for exactly three agents

We start by determining the computational complexity of ExistsNEF in the case when there are exactly three agents.

Before stating and proving our main result, Theorem 1, let us first give some intuition why a complete NEF allocation may be hard to find. By Proposition 1, each agent must be allocated its top-choice item in any complete NEF allocation. Hence, a natural approach would be to consider the requirements of Proposition 1 in an iterative manner, starting with the top-choice items and considering longer and longer prefixes of the preference lists at each step, maintaining throughout a "representative" set of allocations of the items appearing in the current prefixes. If we could keep the size of such a representative set small and, simultaneously, guarantee that at least one allocation in our representative set can be completed into a complete NEF allocation (assuming that such an allocation exists), then such an incremental algorithm would yield a possibility for solving ExistsNEF efficiently.

Theorem 1 shows, however, that ExistsNEF is unlikely to be polynomial-time solvable even for three agents. The intuition behind our NP-hardness proof builds on the flaw in the approach described in the above paragraph. First, there may be several partial allocations that respect the requirements of Proposition 1 for some index i, and in fact, it is not hard to see that the number of such allocations can grow exponentially in i. Second, as our reduction shows, selecting a relatively small subset of partial solutions among those that satisfy the requirements of Proposition 1 for some index i, so that we can safely disregard the remaining ones when considering larger indices, is not possible. It turns out that were such an approach viable, it could be used for determining the value of certain variables in a given Boolean formula (or to narrow down the set of possible truth assignments on them) without even knowing the formula itself. So the hardness of the problem lies in deciding how to allocate those items that appear early in the agents' preference lists in a way that we will not regret our choices later on, when we allocate the less-desired items.

Theorem 1. ExistsNEF*, the problem of deciding whether a complete NEF allocation exists, is* NP*-complete for instances with three agents.*

Proof. Containment in NP is trivial due to Proposition 1. We will show the NP-hardness of our problem by a reduction from the NP-complete Not-All-Equal 3SAT problem [13]. The input for Not-All-Equal 3SAT is a Boolean formula $\varphi = c_1 \wedge \cdots \wedge c_m$ in conjunctive normal form with variables $x_1, \ldots, x_n$, where each clause contains three literals. The task is to find a truth assignment for φ such that each clause contains at least one true literal and at least one false literal; such an assignment is *valid*.

Not-All-Equal 3SAT	
Input:	A Boolean formula $\varphi = c_1 \wedge \cdots \wedge c_m$ in conjunctive normal form with variables $x_1, \ldots, x_n$, where each clause contains three literals.
Question:	Does there exist a truth assignment for φ such that each clause contains at least one true literal and at least one false literal?

We construct an instance (N, I, L) of ExistsNEF with $N = \{A, B, C\}$ such that (N, I, L) admits a complete NEF allocation if and only if φ has a valid assignment.

Construction. Let μ_i denote the number of occurrences of variable x_i in φ as a positive or negative literal; note $\sum_{i=1}^{n} \mu_i = 3m$. W.l.o.g. we may assume that each μ_i is an even number; this can be achieved by adding the clause $(x_i \vee x_i \vee \overline{x_i})$ for each variable x_i with an odd number of occurrences.

The set I of items is defined as follows; note that $|I| = 66m + 3$.

$$I = \Big(\{a^3_{1,0}, b^3_{1,0}, c^3_{1,0}\} \cup \bigcup_{k \in [m]} \{s_k, t^1_k, t^2_k\} \\ \cup \bigcup_{i \in [n], j \in [\mu_i]} \Big(\{\alpha_{i,j}, \beta_{i,j}, \gamma_{i,j}\} \cup \big\{[ab]^h_{i,j}, [bc]^h_{i,j}, [ca]^h_{i,j} : h \in \{0,1,2\}\big\} \\ \cup \big\{a^h_{i,j}, b^h_{i,j}, c^h_{i,j} : h \in \{1,2,3\}\big\} \Big)$$

We will define the preferences of agents through several types of "building blocks". A *block* is a triple of lists where each list is a linearly ordered subset of I. Given two blocks $L = (L_1, L_2, L_3)$ and $L' = (L'_1, L'_2, L'_3)$ such that L_i and L'_i are disjoint for any $i \in [3]$, we define their *concatenation* as $L + L' = (L_1 + L'_1, L_2 + L'_2, L_3 + L'_3)$, where $L_i + L'_i$ denotes the (standard) concatenation of lists.

Preference lists: a high-level view. We begin with a single *initial block* I_0. Then, for each variable x_i, $i \in [n]$, we define the following blocks. For each occurrence of x_i in φ, we construct a *literal block*: for some $j \in [\mu_i]$, we denote the literal block corresponding to the j-th occurrence of variable x_i by $X_{i,j}$. Then we construct $\mu_i/2$ *equivalence blocks* $E_{i,2j}$ where $j \in [\mu_i/2]$. We denote the concatenation $X_{i,1} + \cdots + X_{i,\mu_i} + E_{i,2} + \cdots + E_{i,\mu_i}$ by Y_i.

Each literal block will represent the choice of a truth assignment for the given occurrence of a variable, as there will be two possible ways to allocate the items appearing in a given literal block to the agents. The equivalence blocks will ensure that these choices are consistent for a given variable x_i. Thus, the blocks in Y_i together represent the choice of a truth assignment for the variable x_i. Next, for each clause c_k of φ, we define a *validity block* V_k; this block will make sure that any complete NEF allocation corresponds to a truth assignment that is valid for the clauses c_k. Finally, we define a *closing block* Z whose sole function is to ensure that each preference list contains all items in I. The full preference lists of the agents are obtained by the concatenation $I_0 + Y_1 + \cdots + Y_n + V_1 + \cdots + V_m + Z$.

Details of the blocks. We give the definitions of the building blocks below. For better readability, we give each block as subsequences of the preference lists of the agents in $N = \{A, B, C\}$. Moreover, we define a *triad* as a group of three items contained in $L^X[3k+2 : 3k+4]$ for some $k \in \mathbb{Z}$ and $X \in N$. In the arguments below, it will be crucial to view the list contained in some block (other than the short blocks I_0 and Z) as sequences of triads.

Initial block I_0:

A: $a^3_{1,0}$
B: $b^3_{1,0}$
C: $c^3_{1,0}$

Literal block $X_{i,j}$:

A: $b^3_{i,j-1}, c^3_{i,j-1}, a^1_{i,j}$, $b^1_{i,j}, [ca]^1_{i,j}, [ca]^2_{i,j}$, $c^1_{i,j}, \beta_{i,j}, a^2_{i,j}$,
$c^2_{i,j}, [ab]^1_{i,j}, [ab]^2_{i,j}$, $b^2_{i,j}, \gamma_{i,j}, a^3_{i,j}$, $[bc]^2_{i,j}, [ab]^0_{i,j}, [ca]^0_{i,j}$

B: $a^3_{i,j-1}, [bc]^1_{i,j}, [bc]^2_{i,j}$, $c^3_{i,j-1}, \alpha_{i,j}, b^1_{i,j}$, $a^1_{i,j}, c^1_{i,j}, b^2_{i,j}$,
$c^2_{i,j}, [ab]^1_{i,j}, [ab]^2_{i,j}$, $a^2_{i,j}, \gamma_{i,j}, b^3_{i,j}$, $[ca]^2_{i,j}, [ab]^0_{i,j}, [bc]^0_{i,j}$

C: $a^3_{i,j-1}, [bc]^1_{i,j}, [bc]^2_{i,j}$, $b^3_{i,j-1}, \alpha_{i,j}, c^1_{i,j}$, $b^1_{i,j}, [ca]^1_{i,j}, [ca]^2_{i,j}$,
$a^1_{i,j}, \beta_{i,j}, c^2_{i,j}$, $a^2_{i,j}, b^2_{i,j}, c^3_{i,j}$, $[ab]^2_{i,j}, [ca]^0_{i,j}, [bc]^0_{i,j}$

To "attach" the blocks of some variable x_i to the blocks of the previous variable x_{i-1}, we let $a^3_{i,0} = a^3_{i-1,\mu_{i-1}}$, $b^3_{i,0} = b^3_{i-1,\mu_{i-1}}$, and $c^3_{i,0} = c^3_{i-1,\mu_{i-1}}$ whenever $i \geq 2$; we only have duplicate names for these items to ease the formalization. For similar reasons, we let $[ab]^0_{i,\mu_i+1} = [ab]^0_{i,1}$, $[ab]^1_{i,\mu_i+1} = [ab]^1_{i,1}$, and $\gamma_{i,\mu_i+1} = \gamma_{i,1}$ in the definition of $E_{i,2j}$ below (so indices are taken modulo μ_i for these items).

Equivalence block $E_{i,2j}$:

A: –
B: $[ca]^1_{i,2j-1}, [ca]^0_{i,2j}, \beta_{i,2j-1}$, $[ca]^0_{i,2j-1}, [ca]^1_{i,2j}, \beta_{i,2j}$
C: $[ab]^1_{i,2j}, [ab]^0_{i,2j+1}, \gamma_{i,2j}$, $[ab]^0_{i,2j}, [ab]^1_{i,2j+1}, \gamma_{i,2j+1}$

For defining the validity block V_k for some $k \in [m]$, let us assume that clause c_k contains the j_u-th, j_v-th, and j_z-th occurrence of the variables x_u, x_v, x_z, respectively, in the formula φ. If x_u appears in c_k as a positive literal, then we define the item ℓ_u as $\ell_u = [bc]^1_{u,j_u}$, otherwise we set $\ell_u = [bc]^0_{u,j_u}$. We define ℓ_v and ℓ_z analogously, and we denote the items corresponding to the negated form of these literals by $\overline{\ell_u}$, $\overline{\ell_v}$, and $\overline{\ell_z}$ (thus, if $\ell_u = [bc]^1_{u,j_u}$, then $\overline{\ell_u} = [bc]^0_{u,j_u}$, and vice versa). Now we are ready to describe the validity block V_k.

Validity block V_k:

A: $\alpha_{u,j_u}, s_k, \ell_u$, ℓ_v, ℓ_z, t^1_k, $\alpha_{v,j_v}, \alpha_{z,j_z}, \overline{\ell_u}$, $\overline{\ell_v}, \overline{\ell_z}, t^2_k$
B: s_k, t^1_k, t^2_k
C: s_k, t^1_k, t^2_k

Closing block Z:

A: $b^3_{n,\mu_n}, c^3_{n,\mu_n}$
B: $a^3_{n,\mu_n}, c^3_{n,\mu_n}$
C: $a^3_{n,\mu_n}, b^3_{n,\mu_n}$

Well-formed instance. It is clear that the construction takes polynomial time. It is, however, not so obvious to see that the concatenation of the constructed blocks yields a well-formed instance: one has to check that each preference list contains each item exactly once.

Items of the form $a^h_{i,j}$, $b^h_{i,j}$ and $c^h_{i,j}$ with $h \in [3]$ appear in the literal block $X_{i,j}$, with two exceptions for each agent: in agent A's preference list, items $b^3_{i,j}$ and $c^3_{i,j}$ only appear in the literal block following $X_{i,j}$ (or, for $i = n$

and $j = \mu_n$, in the closing block Z); the same happens in the preference list of agents B and C regarding the items of $\{a^3_{i,j}, c^3_{i,j}\}$ and $\{a^3_{i,j}, b^3_{i,j}\}$, respectively.

Items of $\{[bc]^0_{i,j}, [bc]^1_{i,j}, \alpha_{i,j}\}$ for some $i \in [n]$, $j \in [\mu_i]$ appear in the preferences of agents B and C within the literal block $X_{i,j}$, and they appear in L^A within the validity block V_k corresponding to the clause c_k containing the j-th occurrence of variable x_i in φ. Items of $\{[ca]^0_{i,j}, [ca]^1_{i,j}, \beta_{i,j}\}$ appear within the literal block $X_{i,j}$ for agents A and C, and in the equivalent block $E_{i,j'}$ for agent B, where $j' = 2\lceil j/2 \rceil$. Similarly, items of $\{[ab]^0_{i,j}, [ab]^1_{i,j}, \gamma_{i,j}\}$ appear within $X_{i,j}$ for agents A and B, and in $E_{i,j'}$ for agent C, where $j' = 2\lfloor j/2 \rfloor$ if $j > 1$, and $j' = \mu_i$ if $j = 1$. For any agent, all remaining items of the form $[ab]^h_{i,j}$, $[bc]^h_{i,j}$, or $[ca]^h_{i,j}$ for some $h \in \{0, 1, 2\}$ can be found within $X_{i,j}$. This leaves us with the items of $\{s_k, t^1_k, t^2_k\}$, appearing in the validity block V_k for each $k \in [m]$, and the items of the initial block I_0, also appearing in $X_{1,1}$.

We can thus conclude that each constructed preference list is indeed a strict linear order over I.

To verify the correctness of our reduction, we need the following crucial lemma that proves certain key properties of the constructed instance.

Lemma 1. *Suppose π is a complete NEF allocation for (N, I, L).*

(i) For all indices $i \in [n]$, $j \in [\mu_i]$ and $h \in [3]$ (and also for the case $i = 1$, $j = 0$ and $h = 3$) we have

$$\begin{aligned} \pi(a^h_{i,j}) &= A, & \pi(\alpha_{i,j}) &= A, \\ \pi(b^h_{i,j}) &= B, & \pi(\beta_{i,j}) &= B, \\ \pi(c^h_{i,j}) &= C, & \pi(\gamma_{i,j}) &= C. \end{aligned}$$

(ii) In any literal block $X_{i,j}$, one of the followings hold:
(C1) $X_{i,j}$ is of type 1, *meaning*

$$\begin{aligned} \pi([bc]^1_{i,j}) &= C, & \pi([bc]^2_{i,j}) &= B, & \pi([bc]^0_{i,j}) &= B, \\ \pi([ca]^1_{i,j}) &= A, & \pi([ca]^2_{i,j}) &= C, & \pi([ca]^0_{i,j}) &= C, \\ \pi([ab]^1_{i,j}) &= B, & \pi([ab]^2_{i,j}) &= A, & \pi([ab]^0_{i,j}) &= A, \end{aligned}$$

(C2) $X_{i,j}$ is of type 2, *meaning*

$$\begin{aligned} \pi([bc]^1_{i,j}) &= B, & \pi([bc]^2_{i,j}) &= C, & \pi([bc]^0_{i,j}) &= C, \\ \pi([ca]^1_{i,j}) &= C, & \pi([ca]^2_{i,j}) &= A, & \pi([ca]^0_{i,j}) &= A, \\ \pi([ab]^1_{i,j}) &= A, & \pi([ab]^2_{i,j}) &= B, & \pi([ab]^0_{i,j}) &= B. \end{aligned}$$

(iii) Let S be the list $L^X[1 : 3k+1]$ for some $k \in \mathbb{N}$ and agent X, where either $X = A$ and $k \in [18m]$, or $X \in \{B, C\}$ and $k \in [22m]$. Then $[S]$ contains exactly $k+1$ items allocated to X by π, and exactly k items allocated to each of the other two agents.

(iv) Let S be the list $L^A[1 : 54m+6k+1]$ for some $k \in [2m]$. Then $[S]$ contains exactly $18m+2k+1$ items allocated to A by π, and exactly $18m+2k$ items allocated to each of the agents B and C.

(v) *For any $i \in [n]$, all literal blocks in Y_i are of the same type; we call this the type of Y_i.*

Proof. We prove statements (i)–(iv) of the lemma in an inductive manner, block by block. Within a block, however, we will move from triad to triad. Let us consider such prefixes S_A, S_B, and S_C of the preference lists L^A, L^B, and L^C, respectively, for which $\mathcal{B} = (S_A, S_B, S_C)$ is the concatenation of the first few blocks in our constructed instance, and let B_{next} be the next block. We prove the lemma by induction, so we assume that the statements of (i) hold for all items appearing in $\mathcal{B}$, statement (ii) holds for all literal block contained in $\mathcal{B}$, and that (iii) and (iv) hold for all lists S contained in $\mathcal{B}$.[1] We refer to these claims as the *induction statements*, to distinguish them from the statements of the lemma.

First observe that the induction statements indeed hold if $\mathcal{B} = I_0$. To see this, observe that in a complete NEF allocation each agent must get its most preferred item.

We are going to prove that the induction statements also hold for $\mathcal{B} + B_{\text{next}}$. We distinguish between the following cases, depending on B_{next}.

Case for a literal block: $B_{\text{next}} = X_{i,j}$ for some i and j.

By the induction, we know $\pi(a^3_{i,j-1}) = A$, $\pi(b^3_{i,j-1}) = B$ and $\pi(c^3_{i,j-1}) = C$, since these items already appear in the previous block. Induction statement (iii) for S_A, S_B, and S_C imply by Proposition 1 that each of the agents has to obtain at least one item from his or her three most preferred items in $X_{i,j}$ to ensure necessary envy-freeness. Therefore, the first triad for A shows that π must allocate $a^1_{i,j}$ to A. With the same reasoning, the first triads for agents B and C show that one of $[bc]^1_{i,j}$ and $[bc]^2_{i,j}$ must be allocated to B, and the other to C.

Looking at the second triads for B and C in $X_{i,j}$, we get that $\alpha_{i,j}$ can only be allocated to A, so as not to create too many items in the preference list of B allocated to C, or vice versa. This yields also $\pi(b^1_{i,j}) = B$ and $\pi(c^1_{i,j}) = C$. Now, considering agents A and C and their second and third triads in $X_{i,j}$, respectively, we get that one of $[ca]^1_{i,j}$ and $[ca]^2_{i,j}$ must be allocated to A, and the other to C. Considering the third triad for agent B, $\pi(b^2_{i,j}) = B$ follows.

Next, looking at the third triad for A and the fourth triad for C, we can observe that $\beta_{i,j}$ must be allocated to B to ensure necessary envy-freeness, and $\pi(a^2_{i,j}) = A$ and $\pi(c^2_{i,j}) = C$ follow as well. By the fourth triads for A and B, one of $[ab]^1_{i,j}$ and $[ab]^2_{i,j}$ must be allocated to A, and the other to B. Considering the fifth triads, arguing as above we get $\pi(a^3_{i,j}) = A$, $\pi(b^3_{i,j}) = B$ and $\pi(c^3_{i,j}) = \pi(\gamma_{i,j}) = C$. This shows that the induction statement (i) holds for $\mathcal{B} + B_{\text{next}}$.

Now, consider the last triads of $X_{i,j}$. Clearly, each agent has to be allocated

[1] More precisely, we assume that (iii) and (iv) hold for all lists S that are of the form specified by the corresponding statement, and which, additionally, are contained in one of S_A, S_B, or S_C. Note that the statement of (iv) is empty if $|S_A| \leq 54m + 1$, that is, if $\mathcal{B}$ does not contain any validity blocks.

at least one item from his or her triad, and there are exactly three items ($[bc]^0_{i,j}$, $[ca]^0_{i,j}$, and $[ab]^0_{i,j}$) that they can get. Supposing that π allocates both $[bc]^2_{i,j}$ and $[ca]^2_{i,j}$ to C, one can see that neither $[bc]^0_{i,j}$, nor $[ca]^0_{i,j}$ can be allocated to C, as that would create too many items allocated by π to C in the list of either A or B. Analogously, we obtain that neither $\pi([bc]^2_{i,j}) = \pi([ab]^2_{i,j}) = B$, nor $\pi([ca]^2_{i,j}) = \pi([ab]^2_{i,j}) = A$ is possible. Hence, we must have that either $\pi([bc]^2_{i,j}) = C$, $\pi([ca]^2_{i,j}) = A$ and $\pi([ab]^2_{i,j}) = B$, or $\pi([bc]^2_{i,j}) = B$, $\pi([ca]^2_{i,j}) = C$ and $\pi([ab]^2_{i,j}) = A$. In the former case, we quickly get that A cannot have $[ab]^0_{i,j}$ (as then B would have two items in his last triad of $X_{i,j}$ allocated to A), yielding $\pi([ab]^0_{i,j}) = B$. Similarly, we get $\pi([bc]^0_{i,j}) = C$ and $\pi([ca]^0_{i,j}) = A$ as well. In the latter case, the analogous arguments prove $\pi([bc]^0_{i,j}) = B$, $\pi([ca]^0_{i,j}) = C$ and $\pi([ab]^0_{i,j}) = A$. Recalling our observations in the previous paragraph on $[ab]^1_{i,j}$, $[bc]^1_{i,j}$ and $[ca]^1_{i,j}$, we get that $X_{i,j}$ is either of type 0 or of type 1. Hence, the induction statement for (ii) holds as well.

It remains to observe that π allocates exactly one item to each of the agents from every triad, showing that (iii) holds for $\mathcal{B} + B_{\text{next}}$. Since (iv) holds vacuously, we can conclude that all the induction statements hold for $\mathcal{B} + B_{\text{next}}$.

Case for an equivalence block: $B_{\text{next}} = E_{i,2j}$ for some i and j.

First, statements (i), (ii), and (iv) automatically remain true for $\mathcal{B} + B_{\text{next}}$ by the induction statements. Since the induction statements for claims (ii) and (iii) hold for S_B, and both $[ca]^1_{i,2j-1}$ and $[ca]^0_{i,2j}$ appear in the first triad of L^B within $E_{i,2j}$, we obtain that either $\pi([ca]^1_{i,2j-1}) = A$ and $\pi([ca]^0_{i,2j}) = C$, or vice versa. Hence, $X_{i,2j-1}$ and $X_{i,2j}$ must be of the same type, implying also that each agent obtains exactly one item from both triads of L^B within the block (here we used $\pi(\beta_{i,2j}) = \pi(\beta_{i,2j-1}) = B$ by induction statement (i)). Similarly, the triads of L^C within $E_{i,2j}$ show that $X_{i,2j}$ and $X_{i,2j+1}$ have the same type[2], and that π allocates an item from each triad to each agent. This proves that induction statement (iii) holds for $\mathcal{B} + B_{\text{next}}$.

Case for a validity block: $B_{\text{next}} = V_k$ for some k.

Induction statements (i) and (ii) again remain true for $\mathcal{B} + B_{\text{next}}$ automatically. By induction statement (iii), the triads for B and C imply that π allocates exactly one item from $\{s_k, t^1_k, t^2_k\}$ to each of the agents; this proves induction statement (iii) for $\mathcal{B} + B_{\text{next}}$. By induction statement (ii), we also know that each of the items ℓ_u, ℓ_v, and ℓ_z is allocated to one of B or C by π. This means that π can allocate at most two items to A from $\{\alpha_{u,j_u}, s_k, \ell_u, \ell_v, \ell_z, t^1_k\}$. Thus, by induction statement (iv), π must allocate exactly two items to each of the agents from the first two triads for A. Similarly, each agent gets two items from the last two triads for A. This proves that induction statement (iv) remains true for $\mathcal{B} + B_{\text{next}}$.

Case for the closing block: $B_{\text{next}} = Z$.

Notice that all induction statements remain true for $\mathcal{B} + B_{\text{next}}$ trivially.

[2]Recall that indices within the equivalence block $E_{i,2j}$ are taken modulo μ_i, so for $j = \mu_i/2$ we obtain that $X_{i,\mu(i)}$ and $X_{i,1}$ have the same type.

Using the induction statements for the whole instance, claims (i)–(iv) follow immediately. Finally, our arguments for the case of an equivalence block also prove that all literal blocks contained in some Y_i are of the same type, so claim (v) holds as well. $\square$

Correctness of our reduction. We now prove that there exists a valid allocation for the input formula φ if and only if the constructed instance (N, I, L) admits a complete NEF allocation.

Direction "$\Rightarrow$": Let us first suppose that $\pi : I \to N$ is a complete NEF allocation. Using Lemma 1 we construct a valid truth assignment for φ based on the allocation π. Namely, we set x_i to true if and only if the literal blocks in Y_i are of type 1; by claim (v) of Lemma 1 π is well-defined.

Consider the validity block V_k for some $k \in [m]$, involving the j_u-th, j_v-th, and j_z-th occurrence of the variables x_u, x_v, and x_z, respectively. Observe that there are exactly $54m + 12(k-1) + 1$ items preceding block V_k in the preference list L^A of agent A. By claim (iv) of Lemma 1, we know that among these items exactly $18m + 4(k-1) + 1$ are allocated to A by π, and exactly $18m + 4(k-1)$ are allocated to each of the other two agents.

Thus, using again claim (iv), we get that π allocates exactly two items from $\{\alpha_{u,j_u}, s_k, \ell_u, \ell_v, \ell_z, t_k^1\}$ to each agent. By claim (i) of Lemma 1, we know $\pi(\alpha_{u,j_u}) = A$, and from claim (ii) we get that each of ℓ_u, ℓ_v, and ℓ_z is allocated to one of the agents B or C. Thus, either s_k or t_k^1 is allocated to A. Therefore we obtain that π allocates either 1 or 2 among the items ℓ_u, ℓ_v, and ℓ_z to C.

Using now the definition of these items, and that condition (C1) holds for the variables set to true, we get that the number of true literals in the clause c_k equals the number of items in $\{\ell_u, \ell_v, \ell_z\}$ allocated to C by π. Since this value must be either 1 or 2 (as argued above), we get that c_k contains at least 1 but at most 2 true literals. Hence, our truth assignment is indeed valid for φ.

Direction "$\Leftarrow$": For the converse direction, suppose that we are given a valid truth assignment σ for φ. We construct an allocation π as follows. First, we allocate all items appearing in claim (i) of Lemma 1 as required there. Next, for each variable x_i, we let Y_i have type 1 exactly if σ sets x_i to true, and we let Y_i have type 2 otherwise (yielding the allocations as given in claim (ii) of Lemma 1). We also set $\pi(s_k) = A$ for each clause c_k. Finally, we set $\pi(t_k^1) = B$ and $\pi(t_k^2) = C$ if there are 2 true literals in the clause c_k according to σ, and we set $\pi(t_k^1) = C$ and $\pi(t_k^2) = B$ otherwise.

It is clear that π is complete. To verify that it is NEF, we use the characterization given in Proposition 1: it suffices to check that π assigns exactly one item to each agent from each triad of any preference list, except for the triads of L^A contained in a validity block. There, π always assigns two items to agent A first, followed by four items distributed among B and C evenly, thus fulfilling the requirements of Proposition 1. $\square$

4. Result for at least three agents

In this section we generalize Theorem 1 to the case where the number of agents is a constant integer at least three.

Theorem 2. *For any fixed integer $q \geq 3$,* ExistsNEF *(deciding whether a complete NEF allocation exists) for instances with q agents is* NP*-complete.*

Proof. ExistsNEF is clearly in NP, so we need to show its NP-hardness. To this end, we are going to modify the reduction given in the proof of Theorem 1, so we will re-use most of the notation. The reduction is from the same variant of Not-All-Equal 3SAT as in the proof of Theorem 1, meaning that we again assume that μ_i, the number of occurrences of variable x_i, is an even integer for each $i \in [n]$.

Dummy agents and items. We construct an instance $(\widetilde{N}, \widetilde{I}, \widetilde{L})$ of ExistsNEF that contains agents A, B, C and $q-3$ additional dummy agents $D^1, \dots, D^{q-3}$. We will keep the set I of items used in the proof of Theorem 1, and we define our current set of items as

$$\widetilde{I} = I \cup \{d^r_\tau \mid r \in [q-3], 0 \leq \tau \leq |I|/3 - 1\}.$$

The dummy item d^r_τ will appear at the $(\tau q+1)$-st position in the preference list of agent D^r; we will make sure that any NEF allocation assigns d^r_τ to D^r. For brevity, we let $\langle d_\tau \rangle$ denote the sequence $d^1_\tau, \dots, d^{q-3}_\tau$, and we let $\langle d_\tau \rangle^{-r}$ denote the sequence obtained from $\langle d_\tau \rangle$ by removing the item d^r_τ.

Preferences. We define preferences using the preference lists L^A, L^B, and L^C defined in the proof of Theorem 1 for agents A, B, and C. Instead of considering triads (i.e., sequences of three items in the preference lists) we now decompose each preference list within a block (except for the initial and closing blocks) into sequences of q items which we will call *q-ads*.

First, to construct the new preference list $\widetilde{L}^X$ for some agent $X \in \{A, B, C\}$, for each $\tau \in [22m]$, we insert $\langle d_{\tau-1} \rangle$ at the beginning of the τ-th triad in L^X, that is, the triad $L^X[3\tau - 1 : 3\tau + 1]$. This way, the τ-th triad of L^X becomes the τ-th q-ad of $\widetilde{L}^X$. Second, for each $r \in [q-3]$ we construct the preference list $\widetilde{L}^{D^r}$ of dummy agent D^r based on L^C as follows: for each $\tau \in [22m]$, we insert $\langle d_{\tau-1} \rangle^{-r}$ at the beginning and d^r_τ at the end of the τ-th triad of L^C, thus obtaining the τ-th q-ad for agent D^r. In addition, we insert the item d^r_0 as the most preferred item for D^r. To construct the closing block for some agent $X \in \{A, B, C\}$, we append $\langle d_{22m} \rangle^{-r}$ to the end of the preference list. Similarly, for each dummy agent D^r we append $\langle d_{22m} \rangle^{-r}$ and also the item $c^3_{1,0}$ to the end of the preference list.

The preferences thus defined are shown below. For each block B_{orig} of the instance constructed in the proof of Theorem 1 we let $\widetilde{B}_{\text{orig}}$ denote the block we obtain by modifying B_{orig} as described above. Each block comprises q lists, one for each agent, so index r in the definitions below takes on each value in $[q-3]$.

Modified initial block $\widetilde{I}_0$:

A: $a^3_{1,0}$
B: $b^3_{1,0}$
C: $c^3_{1,0}$
D^r: d^r_0

For each of the following blocks, we set integers τ and ρ such that the original block starts with the τ-th triad in agent A's preference list, and starts with the ρ-th triad in agent B's preference list. For a literal block $\tau = \rho$ holds, since a literal block comprises lists of the same length, but τ and ρ differ in case of an equivalence or a validity block (except for the first equivalence block).

Modified literal block $\widetilde{X}_{i,j}$:

A: $\langle d_{\tau-1}\rangle, b^3_{i,j-1}, c^3_{i,j-1}, a^1_{i,j},$ $\langle d_{\tau}\rangle, b^1_{i,j}, [ca]^1_{i,j}, [ca]^2_{i,j},$
$\langle d_{\tau+1}\rangle, c^1_{i,j}, \beta_{i,j}, a^2_{i,j},$ $\langle d_{\tau+2}\rangle, c^2_{i,j}, [ab]^1_{i,j}, [ab]^2_{i,j},$
$\langle d_{\tau+3}\rangle, b^2_{i,j}, \gamma_{i,j}, a^3_{i,j},$ $\langle d_{\tau+4}\rangle, [bc]^2_{i,j}, [ab]^0_{i,j}, [ca]^0_{i,j}$

B: $\langle d_{\tau-1}\rangle, a^3_{i,j-1}, [bc]^1_{i,j}, [bc]^2_{i,j},$ $\langle d_{\tau}\rangle, c^3_{i,j-1}, \alpha_{i,j}, b^1_{i,j},$
$\langle d_{\tau+1}\rangle, a^1_{i,j}, c^1_{i,j}, b^2_{i,j},$ $\langle d_{\tau+2}\rangle, c^2_{i,j}, [ab]^1_{i,j}, [ab]^2_{i,j},$
$\langle d_{\tau+3}\rangle, a^2_{i,j}, \gamma_{i,j}, b^3_{i,j},$ $\langle d_{\tau+4}\rangle, [ca]^2_{i,j}, [ab]^0_{i,j}, [bc]^0_{i,j}$

C: $\langle d_{\tau-1}\rangle, a^3_{i,j-1}, [bc]^1_{i,j}, [bc]^2_{i,j},$ $\langle d_{\tau}\rangle, b^3_{i,j-1}, \alpha_{i,j}, c^1_{i,j},$
$\langle d_{\tau+1}\rangle, b^1_{i,j}, [ca]^1_{i,j}, [ca]^2_{i,j},$ $\langle d_{\tau+2}\rangle, a^1_{i,j}, \beta_{i,j}, c^2_{i,j},$
$\langle d_{\tau+3}\rangle, a^2_{i,j}, b^2_{i,j}, c^3_{i,j},$ $\langle d_{\tau+4}\rangle, [ab]^2_{i,j}, [ca]^0_{i,j}, [bc]^0_{i,j}$

D^r: $\langle d_{\tau-1}\rangle^{-r}, a^3_{i,j-1}, [bc]^1_{i,j}, [bc]^2_{i,j}, d^r_{\tau},$ $\langle d_{\tau}\rangle^{-r}, b^3_{i,j-1}, \alpha_{i,j}, c^1_{i,j}, d^r_{\tau+1},$
$\langle d_{\tau+1}\rangle^{-r}, b^1_{i,j}, [ca]^1_{i,j}, [ca]^2_{i,j}, d^r_{\tau+2},$ $\langle d_{\tau+2}\rangle^{-r}, a^1_{i,j}, \beta_{i,j}, c^2_{i,j}, d^r_{\tau+3},$
$\langle d_{\tau+3}\rangle^{-r}, a^2_{i,j}, b^2_{i,j}, c^3_{i,j}, d^r_{\tau+4},$ $\langle d_{\tau+4}\rangle^{-r}, [ab]^2_{i,j}, [ca]^0_{i,j}, [bc]^0_{i,j}, d^r_{\tau+5}$

Modified equivalence block $\widetilde{E}_{i,2j}$:

A: –
B: $\langle d_{\rho-1}\rangle, [ca]^1_{i,2j-1}, [ca]^0_{i,2j}, \beta_{i,2j-1},$ $\langle d_{\rho}\rangle, [ca]^0_{i,2j-1}, [ca]^1_{i,2j}, \beta_{i,2j}$
C: $\langle d_{\rho-1}\rangle, [ab]^1_{i,2j}, [ab]^0_{i,2j+1}, \gamma_{i,2j},$ $\langle d_{\rho}\rangle, [ab]^0_{i,2j}, [ab]^1_{i,2j+1}, \gamma_{i,2j+1}$
D^r: $\langle d_{\rho-1}\rangle^{-r}, [ab]^1_{i,2j}, [ab]^0_{i,2j+1}, \gamma_{i,2j}, d^r_{\rho},$ $\langle d_{\rho}\rangle^{-r}, [ab]^0_{i,2j}, [ab]^1_{i,2j+1}, \gamma_{i,2j+1}, d^r_{\rho+1}$

Modified validity block $\widetilde{V}_k$:

A: $\langle d_{\tau-1}\rangle, \alpha_{u,j_u}, s_k, \ell_u,$ $\langle d_{\tau}\rangle, \ell_v, \ell_z, t^1_k,$
$\langle d_{\tau+1}\rangle, \alpha_{v,j_v}, \alpha_{z,j_z}, \overline{\ell_u},$ $\langle d_{\tau+2}\rangle, \overline{\ell_v}, \overline{\ell_z}, t^2_k$
B: $\langle d_{\rho-1}\rangle, s_k, t^1_k, t^2_k$
C: $\langle d_{\rho-1}\rangle, s_k, t^1_k, t^2_k$
D^r: $\langle d_{\rho-1}\rangle^{-r}, s_k, t^1_k, t^2_k, d^r_{\rho}$

Modified closing block $\widetilde{Z}$:

A: $b^3_{n,\mu_n}, c^3_{n,\mu_n}, \langle d_{22m} \rangle$
B: $a^3_{n,\mu_n}, c^3_{n,\mu_n}, \langle d_{22m} \rangle$
C: $a^3_{n,\mu_n}, b^3_{n,\mu_n}, \langle d_{22m} \rangle$
D^r: $a^3_{n,\mu_n}, b^3_{n,\mu_n}, \langle d_{22m} \rangle^{-r}, c^3_{1,0}$

It is not hard to verify that the above modified preferences are well-formed, i.e., each preference list is a linear ordering of the set of items.

Proving the correctness of our construction can be done along the same lines as in the proof of Theorem 1, by showing the following key lemma, an analog of Lemma 1 that also deals with dummies.

Lemma 2. *Let π be a complete NEF allocation for the instance of* EXISTSNEF *created above. Then the statements (i), (ii), and (v) of Lemma 1 hold, and additionally:*

(i') For each $r \in [q-3]$ and for each $0 \le \tau \le |I|/3 - 1$ we have $\pi(d^r_\tau) = D^r$.

(iii') Let S contain the top $qk+1$ items for some $k \in \mathbb{N}$ in the preference list of agent X, where either $X = A$ and $k \le 18m$, or $X \neq A$ and $k \le 22m$. Then S contains exactly $k+1$ items allocated to X by π, and exactly k items allocated to each of the other agents.

(iv') Let S contain the top $18qm + 2qk + 1$ items for some $k \in [2m]$ in the preference list of agent A. Then S contains exactly $18m + 2k + 1$ items allocated to A by π, and exactly $18m + 2k$ items allocated to each of the other agents.

Proof. The proof is a direct analog of the proof of Lemma 1. Again we use induction to prove each claim, except for claim (v) of Lemma 1 whose proof remains unchanged. We prove the induction statements, modified according to the claims of the current lemma in the straightforward way, block by block using essentially the same arguments as in the proof of Lemma 1. Nevertheless, we clearly have to take into account the presence of dummy agents and dummy items, and show that our arguments can be applied in the modified instance as well.

So let $\mathcal{B}$ denote the concatenation of the first few blocks of our modified instance $(\widetilde{N}, \widetilde{I}, \widetilde{L})$, and let B_{next} be the next block. Assuming that the induction statements hold for $\mathcal{B}$, we now prove that they prove for $\mathcal{B} + B_{\text{next}}$.

Case for the initial block: $B_{\text{next}} = \widetilde{I}_0$.

Note that since d^r_0 is the top choice of D^r, it is clear that π must assign it to D^r. Therefore all induction statements hold for the base case.

Case for a literal block: $B_{\text{next}} = \widetilde{X}_{i,j}$ for some i and j.

Suppose that the preference list of some agent A, B, or C within $\widetilde{X}_{i,j}$ starts with the series $\langle d_{\tau-1} \rangle$ for some $\tau \in [22m]$. Since each item of $\langle d_{\tau-1} \rangle$ has already appeared in the previous block within the dummy agents' preference lists, that is, in the last block of $\mathcal{B}$, by induction statement (i') we know that all of these items are allocated to dummy agents; namely $\pi(d^r_{\tau-1}) = D^r$ for each r. Thus, we can argue about the remaining three items within the first q-ads of L^A, L^B, and L^C exactly as we did for the corresponding triads in the proof of Lemma 1 to obtain that $\pi(a^1_{i,j}) = A$ and $\{\pi([bc]^1_{i,j}), \pi([bc]^2_{i,j})\} = \{B, C\}$.

Considering now the first q-ad of the preference list of the dummy agent D^r within $\widetilde{X}_{i,j}$, this means that π assigns exactly one item from this q-ad to each agent other than D^r. Hence, by induction statement (iii') we know that π must assign the single remaining item, namely d^r_τ, to D^r.

We can proceed this way, q-ad by q-ad, using the arguments of Lemma 1 combined with the above reasoning about dummy items to show that each induction statement remains true for $\mathcal{B} + \widetilde{X}_{i,j}$. For details, see Table 2 that gives a short description of the chain of our reasoning for proving induction statements (i), (i'), (ii), and consequently (iii') as well. Statement (iv') remains true vacuously.

step	q-ads considered in $X_{i,j}$	consequence
0.	ind. statements (i), (i')	$\pi(a^3_{i,j-1}) = A$, $\pi(b^3_{i,j-1}) = B$, $\pi(c^3_{i,j-1}) = C$, $\pi(d^r_\tau) = D^r$
1.	1st in $\widetilde{L}^A$	$\pi(a^1_{i,j}) = A$
2.	1st in $\widetilde{L}^B$ and $\widetilde{L}^C$	$\{\pi([bc]^1_{i,j}), \pi([bc]^2_{i,j})\} = \{B, C\}$
3.	1st in $\widetilde{L}^{D^r}$	$\pi(d^r_\tau) = D^r$
4.	2nd in $\widetilde{L}^B$ and $\widetilde{L}^C$	$\pi(\alpha_{i,j}) = A$, $\pi(b^1_{i,j}) = B$, $\pi(c^1_{i,j}) = C$
5.	2nd in $\widetilde{L}^{D^r}$	$\pi(d^r_{\tau+1}) = D^r$
6.	2nd in $\widetilde{L}^A$, 3rd in $\widetilde{L}^C$	$\{\pi([ca]^1_{i,j}), \pi([ca]^2_{i,j})\} = \{A, C\}$
7.	3rd in $\widetilde{L}^B$	$\pi(b^2_{i,j}) = B$
8.	3rd in $\widetilde{L}^{D^r}$	$\pi(d^r_{\tau+2}) = D^r$
9.	3rd in $\widetilde{L}^A$, 4th in $\widetilde{L}^C$	$\pi(\beta_{i,j}) = B$, $\pi(a^2_{i,j}) = A$, $\pi(c^2_{i,j}) = C$
10.	4th in $\widetilde{L}^A$ and $\widetilde{L}^B$	$\{\pi([ab]^1_{i,j}), \pi([ab]^2_{i,j})\} = \{A, B\}$
11.	4th in $\widetilde{L}^{D^r}$	$\pi(d^r_{\tau+3}) = D^r$
12.	5th in $\widetilde{L}^C$	$\pi(c^3_{i,j}) = C$
13.	5th in $\widetilde{L}^A$ and $\widetilde{L}^B$	$\pi(\gamma_{i,j}) = B$, $\pi(a^3_{i,j}) = A$, $\pi(b^3_{i,j}) = B$
14.	5th in $\widetilde{L}^{D^r}$	$\pi(d^r_{\tau+4}) = D^r$
15.	6th in $\widetilde{L}^A$, $\widetilde{L}^B$, and $\widetilde{L}^C$	$\{\pi([ab]^2_{i,j}), \pi([bc]^2_{i,j}), \pi([ca]^2_{i,j})\} = \{A, B, C\}$, $X_{i,j}$ is of type 1 or type 2

Table 2: Chain of reasoning for proving induction statements (i), (i') and (ii) for the modified literal block $\widetilde{X}_{i,j}$. We also use induction statement (iii') repeatedly to argue that each agent has to obtain at least one item from each q-ad of its preference list within $\widetilde{X}_{i,j}$.

Case for an equivalence block: $B_{\text{next}} = \widetilde{E}_{i,2j}$.

Suppose that the preference list of some agent A, B, or C within the block $\widetilde{E}_{i,2j}$ starts with the series $\langle d_\rho \rangle$ for some $\rho \in [22m]$. Induction statements (i), (i') and (ii) for $\mathcal{B}$ imply that the single item that π can assign to D^r from its first q-ad within the block is d^r_ρ, so by induction statement (iii') we know $\pi(d^r_\rho) = D^r$. Repeating this argument again we obtain $\pi(d^r_{\rho+1}) = D^r$ as well, proving (i') for $\mathcal{B} + B_{\text{next}}$. Statements (i), (ii), and (iv') remain true trivially.

Considering the first q-ad of $\widetilde{L}^B$ within the block, induction statement (iii') implies that the literal blocks $X_{i,2j}$ and $X_{i,2j-1}$ must be of the same type, as otherwise π would assign too many items to either A or to C from the prefix of $\widetilde{L}^B$ ending with this q-ad. Similarly, considering the first q-ad of $\widetilde{L}^C$ within $\widetilde{E}_{i,2j}$ yields that $X_{i,2j}$ and $X_{i,2j+1}$ must be of the same type (when taking indices modulo μ_i). Observe that this proves not only the induction statement (iii') for $\mathcal{B} + B_{\text{next}}$, but also claim (v).

Case for a validity block: $B_{\text{next}} = \widetilde{V}_k$ for some k.

Suppose that the preference lists of agents A and B within the block start with the series $\langle d_{\tau-1} \rangle$ and $\langle d_{\rho-1} \rangle$. Note that if $k = m$, then $\tau + 2 = \rho - 1$, otherwise (i.e., if $k \in [m-1]$) we know $\tau + 2 < \rho - 1$. Therefore, all four items in $\{d^r_{\tau-1}, d^r_\tau, d^r_{\tau+1}, d^r_{\rho-1}\}$ for some $r \in [q-3]$ have already appeared in $\mathcal{B}$, and hence by induction statement (i') π assigns them to D^r. As in the proof of Theorem 1, by observing the q-ads of $\widetilde{L}^B$ and $\widetilde{L}^C$ within $\widetilde{V}_k$ we obtain $\{\pi(s_k), \pi(t^1_k), \pi(t^2_k)\} = \{A, B, C\}$; this shows statement (iii) for $\mathcal{B} + B_{\text{next}}$. Taking into account the q-ad of $\widetilde{L}^{D^r}$ this implies $\pi(d^r_\rho) = D^r$ as well, proving (i').

Observe that since π can assign at most one item from $\{s_k, t^1_k, t^2_k\}$ to agent A, we also obtain that π must assign exactly two items from the first two q-ads of $\widetilde{L}^A$ within the block to each agent. Repeating this argument again for the third and fourth q-ads, statement (iv') for $\mathcal{B} + B_{\text{next}}$ follows. All remaining induction statements remain true vacuously.

Case for the closing block: $B_{\text{next}} = \widetilde{Z}$.

Again, in this case, all induction statements remain true for $\mathcal{B} + B_{\text{next}}$ trivially. □

It is straightforward to verify that using Lemma 2 the same arguments we applied in the proof for Theorem 1 also imply Theorem 2; we leave the details to the reader. □

5. Conclusion

In this paper, we examined a fundamental fair division problem under ordinal preferences. We resolved an outstanding open problem and proved that checking whether a necessary envy-free allocation exists is NP-complete when the number of agents is at least 3. It will be interesting to identify conditions under which the problem is polynomial-time solvable. For example, does it help if the preferences are single-peaked or if there are only two types of preferences?

Acknowledgments

Ildikó Schlotter is supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2021-2) and the Hungarian Scientific Research Fund through OTKA grants no. K128611 and K124171.

References

[1] Aziz, H., 2015. A generalization of the al method for fair allocation of indivisible objects. Economic Theory Bulletin.

[2] Aziz, H., 2020. Developments in multi-agent fair allocation. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI 2020). pp. 13563–13568.

[3] Aziz, H., Gaspers, S., Mackenzie, S., Walsh, T., 2015. Fair assignment of indivisible objects under ordinal preferences. Artificial Intelligence 227, 71–92.

[4] Aziz, H., Schlotter, I., Walsh, T., 2016. Control of fair division. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016). pp. 67–73.

[5] Bouveret, S., Chevaleyre, Y., Maudet, N., 2016. Fair allocation of indivisible goods. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A. D. (Eds.), Handbook of Computational Social Choice. Cambridge University Press, pp. 284–310.

[6] Bouveret, S., Endriss, U., Lang, J., 2010. Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods. In: Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010). Vol. 215 of Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 387–392.

[7] Brams, S. J., Edelman, P. H., Fishburn, P. C., 2001. Paradoxes of fair division. Journal of Philosophy 98 (6), 300–314.

[8] Brams, S. J., Kilgour, D. M., Klamler, C., 2014. Two-person fair division of indivisible items: An efficient, envy-free algorithm. Notices of the AMS 61 (2), 130–141.

[9] Brams, S. J., Taylor, A. D., 1996. Fair division: from cake-cutting to dispute resolution. Cambridge University Press.

[10] Bredereck, R., Figiel, A., Kaczmarczyk, A., Knop, D., Niedermeier, R., 2021. High-multiplicity fair allocation made more practical. In: Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '21). ACM, pp. 260–268.

[11] Bredereck, R., Kaczmarczyk, A., Knop, D., Niedermeier, R., 2019. High-multiplicity fair allocation: Lenstra empowered by n-fold integer programming. In: Proceedings of the 2019 ACM Conference on Economics and Computation (EC 2019). ACM, pp. 505–523.

[12] Moulin, H., 2003. Fair division and collective welfare. MIT Press.

[13] Schaefer, T. J., 1978. The complexity of satisfiability problems. In: Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC '78). Association for Computing Machinery, New York, NY, USA, pp. 216–226.