



Equitability and welfare maximization for allocating indivisible items

Ankang Sun¹ · Bo Chen¹ · Xuan Vinh Doan¹

Accepted: 13 October 2022
© The Author(s) 2022

Abstract

We study fair allocations of indivisible goods and chores in conjunction with system efficiency, measured by two social welfare functions, namely utilitarian and egalitarian welfare. To model preference, each agent is associated with a cardinal and additive valuation function. The fairness criteria we are concerned with are equitability up to any item (EQX) and equitability up to one item (EQ1). For the trade-off between fairness and efficiency, we investigate efficiency loss under these fairness constraints and establish the price of fairness. From the computational perspective, we provide a complete picture of the computational complexity of (i) deciding the existence of an EQX/EQ1 and welfare-maximizing allocation; (ii) computing a welfare maximizer among all EQX/EQ1 allocations.

Keywords Fair division · Indivisible items · Price of fairness

1 Introduction

Fairness and efficiency are two fascinating goals in resource allocation problems and have been extensively studied in social science, operations research and computer science [5, 11, 28, 36]. It is known that there is a trade-off between fairness and efficiency, i.e., optimization on one notion may lead to bad performance on the other. Such a trade-off motivates theoretical work on quantifying the efficiency loss under fairness constraints [8, 19, 37] and practical work on balancing fairness and efficiency, such as in GPU cluster [20], organ transplant [10] and transportation [35]. To investigate the relationship between fairness and efficiency, the very first problem is to determine whether there exists a fair allocation that achieves the optimal social welfare. On the one hand, a positive answer dramatically narrows down the search space of the desired allocations, which makes it possible to compute and choose such an allocation in practice efficiently. On the other hand,

✉ Bo Chen
B.Chen@warwick.ac.uk

Ankang Sun
A.Sun.2@warwick.ac.uk

Xuan Vinh Doan
Xuan.Doan@wbs.ac.uk

¹ Warwick Business School, University of Warwick, Coventry, UK

a negative answer implies that some realistic relaxations on one of or both criteria are needed. For example, when international agencies and organizations distribute aid material to refugees, the assignment needs to not only improve the overall happiness of the refugee group, but also treat every member in the group equally, so as to avoid potential internal conflicts caused by unreasonable distributions. For the distributor, a crucial work is to determine whether there is such a distribution that can maximize the solution to the needs of the refugee group while ensuring that everyone is treated equally. If the distributor knows that the desired assignment exists, she can efficiently find such an allocation by searching the set of maximum welfare allocations.

The underlying fairness notion of this work is *equitability* that requires all agents to receive the same value in an allocation. Equitability acts as an interpersonal fairness criterion and affects agents' choice and behaviour when facing one-shot distribution problems [23] and voluntary cooperation games [24]. Herreiner and Puppe [31] conduct free-form bargaining experiments to compare interpersonal (equitability) and intrapersonal (envy-freeness) criteria and indicate that interpersonal criterion plays the dominant role and acts as the cognitive fairness more often than envy-freeness. When assigning indivisible items, the existence of an *equitable* (EQ) allocation is not guaranteed, which motivates us to study two of its realistic relaxations: *equitability up to one item* (EQ1) and *equitability up to any item* (EQX). The idea of relaxing equitability through eliminating some specific items is originated by Gourvès et al. [30] in which it is named as "Near Jealousy-Freeness". Freeman et al. [25] formally define the notions of EQX and EQ1 in goods allocation and show that both notions are satisfiable when agents have additive valuations. Then, the existence of EQX and EQ1 is proved by Freeman et al. [26] in chores allocation.

Social welfare or social efficiency is measured by social welfare functions in welfare economics, microeconomic techniques for evaluating well-being at the aggregate level. Given a set of social states, a welfare function can prioritize every possible pair of social states as more desirable and less desirable or indifferent between these two. In this work, we are interested in two typical ethical principles, *utilitarianism* and *egalitarianism*, important both in theoretical and practical social decision making [36]. Two corresponding cardinal welfare functions, utilitarian welfare and egalitarian welfare (also known as Rawlsian welfare [34]) are studied. While a utilitarian measures the welfare by the sum of individuals' value, egalitarian welfare is the well-being of the worst-off individual (also known as maximin criterion). In some cases, achieving optimal social welfare may lead to very unfair outcomes, and recent work in fair division concern the interaction between fairness and efficiency by quantifying efficiency under fairness constraints [3, 7, 19, 38] and optimizing welfare in the domain of fair allocations [14, 29].

In this work, we consider the problem of allocating indivisible items to several agents and investigate both goods and chores allocations. Although problems under these two settings are similar, results established in one setting do not necessarily hold for the other [15, 16, 26]. According to Bogomolnaia et al. [14, p. 3], "the results for goods and bads are similar but not mirror images of one another." Therefore, our study on fair allocations together with welfare maximization will be on both settings of goods and chores. In particular, we wish to decide whether there exists a nearly equitable allocation that also achieves the maximum utilitarian or egalitarian welfare, and efficiently compute a welfare maximizer among all nearly equitable allocations.

1.1 Related literature

The notion of equitable allocation is originally studied in the cake-cutting problem (a divisible item) and its existence has been proved to be guaranteed by Dubins and Spanier [22]. For indivisible items, Gourvès et al. [30] relax equitability based on the up to one item scheme and prove the existence of this relaxed equitability in the matroid context. The notions of EQX and EQ1 are formally defined by Freeman et al. [25] that study these two fairness criteria together with Pareto efficiency and envy-freeness. They answer the existence and computational complexity of a sequence of related problems. In chores allocation, Freeman et al. [26] also consider EQX and EQ1 together with Pareto efficiency and answer corresponding existence and computation problems. Additional work on equitable allocations imposes connectivity constraints; each item is placed in a vertex of a graph and the bundle received by agents must be connected. Bouveret et al. [18] consider assigning chores in the path, star, and complete graph and establish results on the complexity of the existence of equitable and other fair allocations. None of the above-mentioned work studies these two fairness notions together with another important objective, social welfare.

Study on fairness together with social welfare is considerably intensified recently [8, 9, 19]. To quantify the efficiency loss under fairness requirements, Caragiannis et al. [19] introduce the price of fairness and study the notion of envy-freeness, proportionality, and equitability in divisible and indivisible goods and chores. In the case of indivisible goods, Bei et al. [7] consider fairness notions whose existences are guaranteed and provide characterizations on their price of fairness. They present lower bound $\Omega(\sqrt{n})$ and upper bound $O(n)$ on the price of envy-free up to one item, and this gap is then closed by Barman et al. [6], who show that the price of envy-free up to one item and of $(1/2)$ -approximate maximin share are both $\Theta(\sqrt{n})$. When assigning indivisible chores, Sun et al. [38] provide tight results on several fairness notions that are proposed as relaxation of envy-freeness. The notion of the price of fairness is also applied to more practical topics such as kidney exchange [21] and machine scheduling [1, 12]. One of the papers closest to ours is Aziz et al. [4], which focuses on the notion of (relaxed) envy-freeness and proportionality. The authors study in the setting of goods the computational complexity of computing fair and welfare-maximizing allocations. They also briefly discuss the adaptability of their approach to other notions of fairness. In addition, the relationship between fairness and social welfare has been recently investigated in the online setting [14, 29, 39].

1.2 Main results

The main contribution of this work is to provide a clear picture of the computational complexity of determining the existence of a nearly equitable allocation that also achieves the maximum social welfare and moreover, the complexity of computing a welfare maximizer among all equitable allocations. We also establish the price of fairness under every pairwise fairness and welfare combination.

On the price of fairness, we have the following main results, which are summarized in Table 1.

- In chores allocation, the price of EQX and of EQ1 with respect to utilitarian and egalitarian welfare are both infinite.
- In goods allocation, the price of EQX and of EQ1 with respect to egalitarian welfare are both 1. For utilitarian welfare, if there are two agents, the price of EQX is $3/2$ and the

Table 1 Prices of fairness

	EQX	EQ1	
Utilitarian	$n = 2: \frac{3}{2}$ (T3.4)	$n = 2: \left[\frac{6}{5}, \frac{\sqrt{2}+1}{2}\right]$ (T3.5)	Goods
	$n \geq 3: [n - 1, 3n]$ (T3.6)	$n \geq 3: [n - 1, 3n]$ (T3.6)	
	∞ (T3.3)	∞ (T3.3)	Chores
Egalitarian	1 (T3.1)	1 (T3.1)	Goods
	∞ (T3.2)	∞ (T3.2)	Chores

Interval $[a, b]$ means that the lower bound is equal to a and upper bound is equal to b . Tx.y points to Theorem x.y

price of EQ1 is at least $6/5$ and at most $(\sqrt{2} + 1)/2$. For general n agents, the price of EQX and of EQ1 are both at least $n - 1$ and at most $3n$, asymptotically tight $\Theta(n)$.

After quantifying the welfare loss under fair allocations, we investigate relaxed equitability and welfare maximization from the algorithmic perspective. When concerning egalitarian welfare in goods allocation, results on the price of fairness show that there exist EQX and EQ1 allocations that achieve the optimal egalitarian welfare. We then prove that, on the contrary, when assigning chores, deciding the existence of an EQX (resp., EQ1) allocation that also maximizes the egalitarian welfare is strongly NP-hard for general n and NP-hard for fixed $n \geq 2$ (resp., $n \geq 3$). For optimization problems, we show that computing an EQX (or EQ1) allocation with the maximum egalitarian welfare is strongly NP-hard for general n and NP-hard for fixed $n \geq 2$ in both cases of goods and chores. Moreover, in the case of fixed n , we design pseudo-polynomial time algorithms that output an EQX or EQ1 allocation with the maximum egalitarian welfare.

On the other hand, when focusing on utilitarian welfare, the computational complexity in allocating goods and chores is identical. In particular, for general n , every decision or optimization problem is strongly NP-complete and strongly NP-hard, respectively. For fixed

Table 2 Computational complexity for fixed n

	UW Goods/Chores		EW Goods	EW Chores	
	$n = 2$	$n \geq 3$	$n \geq 2$	$n = 2$	$n \geq 3$
$E(W \times EQ1)$	P (T4.11)	NP-complete (T4.9) pseudo-poly (T5.5)	P (T3.1)	?	NP-hard (T4.14) pseudo-poly (T5.3 & 5.6)
$C(W/EQ1)$	NP-hard (T4.12) pseudo-poly (T5.4)		NP-hard (T4.15) pseudo-poly (T5.1 & 5.4)		
$E(W \times EQX)$	NP-complete (T4.7) pseudo-poly (T5.5)		P (T3.1)	NP-hard (T4.13) pseudo-poly (T5.3 & 5.6)	
$C(W/EQX)$	NP-hard (T4.8) pseudo-poly (T5.4)		NP-hard (T4.15) pseudo-poly (T5.2 & 5.4)		

Note: The problem descriptors in the first column are defined in detailed at the beginning of Sect. 4. Abbreviations “UW” and “EW” refer to utilitarian welfare and egalitarian welfare, respectively. Abbreviation “Tx.y” points to Theorem x.y. The complexity of $E(EW \times EQ1)$ for allocating chores to two agents is open

n , our results are summarized in Table 2. The first column of Table 2 contains the (decision/optimization) problem descriptors. We denote by “E($W \times F$)” the problem of deciding whether there exists an F allocation that also maximizes W among all allocations, and denote by “C(W/F)” the problem of computing an F allocation that maximizes W among all F allocations. The notion of W refers to the welfare function, and the first row of Table 2 introduces the welfare function under consideration.

Most of the proofs of our results are presented in the main body of the paper, whereas the rest (particularly those with similar arguments) is relegated to the Appendix.

2 Preliminaries

A fair division instance $\mathcal{I} = \langle [n], E, \mathcal{V} \rangle$ is composed of a set $[n]$ of agents and a set $E = \{e_1, \dots, e_m\}$ of m indivisible items, where $[n] = \{1, \dots, n\}$ for $n \in \mathbb{N}^+$. Each agent i is associated with a valuation function $v_i \in \mathcal{V}$ and $v_i : 2^E \rightarrow \mathbb{R}$. Given an item $e \in E$, we say that e is a good if for every $i \in [n]$, $v_i(e) \geq 0$ and e is a chore if for every $i \in [n]$, $v_i(e) \leq 0$. We consider the situation where all items are either goods or chores and we call \mathcal{I} a *fair-goods* (resp., *fair-chores*) instance if every item is a good (resp., a chore). Throughout the paper, for every $i \in [n]$, we assume $v_i(\emptyset) = 0$ and function $v_i(\cdot)$ is *additive*, that is, $v_i(S) = \sum_{e \in S} v_i(e)$ for any $S \subseteq E$. For simplicity, instead of $v_i(\{e_j\})$, we use $v_i(e_j)$ to represent the value of item e_j on agent i .

An allocation $\mathbf{A} := (A_1, \dots, A_n)$ is an n -partition of E among agents, i.e., $A_i \cap A_j = \emptyset$ for any $i \neq j$ and $\bigcup_{i \in [n]} A_i = E$. Each subset $S \subseteq E$ also refers to a *bundle* of items. For any bundle S and $k \in \mathbb{N}^+$, we denote by $\Pi_k(S)$ the set of all k -partition of S , and $|S|$ the number of items in S .

2.1 Social welfare functions

Utilitarian welfare is defined as the sum of individuals' values, and egalitarian welfare is equal to the value of the worst-off agents. Formally, we have the following two definitions.

Definition 2.1 Given an allocation \mathbf{A} , the utilitarian welfare (UW) of \mathbf{A} is defined as $\text{UW}(\mathbf{A}) = \sum_{i \in [n]} v_i(A_i)$.

Definition 2.2 Given an allocation \mathbf{A} , the egalitarian welfare (EW) of \mathbf{A} is defined as $\text{EW}(\mathbf{A}) = \min_{i \in [n]} v_i(A_i)$.

Among all allocations, we are interested in the one that has the maximum social welfare and such an allocation is also called a *welfare maximizer*.

Definition 2.3 An allocation \mathbf{A} is said to be a *utilitarian welfare maximizer* (UWM) and a *egalitarian welfare maximizer* (EWM) if it has the maximum utilitarian welfare and maximum egalitarian welfare, respectively.

2.2 Fairness criteria

We study equitable allocations and two of its relaxations. When presenting the definition of approximately equitable allocations, we distinguish between the case of goods and chores.

Definition 2.4 An allocation $\mathbf{A} = (A_1, \dots, A_n)$ is equitable (EQ) if $v_i(A_i) = v_j(A_j)$ for any $i, j \in [n]$.

Definition 2.5 For allocating goods, an allocation \mathbf{A} is equitable up to one item (EQ1) if there exists $e \in A_j$ such that $v_i(A_i) \geq v_j(A_j \setminus \{e\})$ for any $i, j \in [n]$. For allocating chores, an allocation \mathbf{A} is equitable up to one item (EQ1) if there exists $e \in A_i$ such that $v_i(A_i \setminus \{e\}) \geq v_j(A_j)$ for any $i, j \in [n]$.

Definition 2.6 For allocating goods, an allocation \mathbf{A} is equitable up to any item (EQX) if for any $i, j \in [n]$ and any $e \in A_j$ with $v_j(e) \neq 0$, $v_i(A_i) \geq v_j(A_j \setminus \{e\})$ holds. For allocating chores, an allocation \mathbf{A} is equitable up to any item (EQX) if for any $i, j \in [n]$ and any $e \in A_i$ with $v_i(e) \neq 0$, $v_i(A_i \setminus \{e\}) \geq v_j(A_j)$ holds.

2.3 Price of fairness

The *price of fairness* (PoF) quantifies the loss of economic welfare when enforcing allocation fairness. For allocation of goods, PoF is the supremum ratio over all problem instances between the maximum welfare of all allocations and maximum welfare of all fair allocations. In the case of chores, PoF is the supremum ratio over all problem instances between the maximum welfare of all fair allocations and maximum welfare of all allocations. The price of fairness has been applied to quantify the welfare loss under fairness requirements in both settings of goods [3, 7, 19] and chores [19, 32, 33]. Following previous studies [7, 19, 33], we assume agents' valuation functions are normalized¹ to 1 in the case of goods and -1 in the case of chores (only) when we are concerned with the price of fairness. Given an instance \mathcal{I} and a welfare function $W \in \{\text{EW}, \text{UW}\}$, denote by $\text{OPT}_W(\mathcal{I})$ the maximum welfare with respect to W over all allocations of instance \mathcal{I} . Moreover, given a fairness criterion F and an instance \mathcal{I} , denote by $F(\mathcal{I})$ the set of all allocations of instance \mathcal{I} satisfying fairness criterion F .

Definition 2.7 Given a fair-goods instance \mathcal{I} , for any fairness criterion F and welfare function W , the price of F with respect to W is defined as

$$\text{PoF}_W = \sup_{\mathcal{I}} \min_{\mathbf{A} \in F(\mathcal{I})} \frac{\text{OPT}_W(\mathcal{I})}{W(\mathbf{A})}.$$

In the case of chores, swap the positions of the numerator and denominator.

In the above definition, we apply the following convention for the case where the maximum welfare of a fair-chores instance is equal to zero: if some fair allocation can achieve welfare zero, then the price of fairness is 1; otherwise, the price of fairness is infinite. The price of fairness with respect to fairness notion F is also called *price of F* , i.e., price of EQ1 and price of EQX.

¹ Normalization can make utilitarian and egalitarian welfare scale invariance. In our case, if agents have different maximum achievable values, the price of fairness with respect to utilitarian welfare would be infinite.

3 Results on price of fairness

We start with studying the price of fairness for every possible pair of welfare function and fairness criterion, which also answers the question of the existence of a nearly equitable allocation that approximates a welfare maximizer. Intuitively, if the price of fairness with respect to fairness criterion F and welfare function W is 1, then in any instance \mathcal{I} , there exists an F allocation that achieves $\text{OPT}_W(\mathcal{I})$.

3.1 With respect to egalitarian welfare

First we are concerned with egalitarian welfare in both cases of goods and chores, and provide tight results that also reveal differences between goods and chores allocation. Freeman et al. [25] state that leximin² implies EQX in allocating goods when agents have a strictly positive value on every item, while leximin fails to guarantee EQX when some items are valued at 0. Below, we prove that EQX (or EQ1) is compatible with optimal egalitarian welfare, even dropping the requirement of strictly positive values of all items.

Theorem 3.1 *When allocating goods, the price of EQX and of EQ1 with respect to egalitarian welfare are both equal to 1.*

Proof Since EQX is stricter than EQ1, it suffices to show the statement holds for EQX. We explicitly construct such an allocation $\mathbf{A} = (A_1, \dots, A_n)$ as follows.

\mathbf{A} first maximizes the egalitarian welfare among all allocations. If there is a tie, \mathbf{A} minimizes the number of agents who receive the value $\text{EW}(\mathbf{A})$, and subject to that, maximizes the total number of items assigned to all agents who receive the value $\text{EW}(\mathbf{A})$.

By construction, it is straightforward to see that \mathbf{A} is an egalitarian-welfare maximizing allocation. If allocation \mathbf{A} is EQX, then clearly the theorem statement holds.

Next, we focus on the case where \mathbf{A} is not EQX. Without loss of generality, assume $v_1(A_1) \leq \dots \leq v_n(A_n)$. Note that if $v_1(A_1) \geq v_j(A_j \setminus \{e\})$ holds for all j and $e \in A_j$ with $v_j(e) > 0$, then allocation \mathbf{A} is EQX, a contradiction. Thus, agent 1 must violate EQX. Let \mathcal{J} be the set of agents such that agent 1 violates EQX when compare to agent $j \in \mathcal{J}$. For each $j \in \mathcal{J}$, order goods as $A_j = \{e_{j_1}, \dots, e_{j_{|A_j|}}\}$ with $v_j(e_{j_k}) \leq v_j(e_{j_{k+1}})$. We claim that $v_j(e_{j_1}) > 0$ holds; otherwise, reassigning e_{j_1} to agent 1 results in another allocation satisfying one of the following properties: (1) the egalitarian welfare is larger than $\text{EW}(\mathbf{A})$; (2) the number of agents receiving the value $\text{EW}(\mathbf{A})$ is one less than that of \mathbf{A} ; (3) the total number of items assigned to all agents with the value $\text{EW}(\mathbf{A})$ is one more than that of \mathbf{A} . For each $j \in \mathcal{J}$, since agent 1 violates EQX when comparing to agent j , it holds that $v_1(A_1) < v_j(A_j \setminus \{e_{j_1}\})$, and accordingly, there exists an index $l \geq 1$ such that $v_j(A_j \setminus \{e_{j_1} \cup \dots \cup e_{j_l}\}) > v_1(A_1)$ and $v_j(A_j \setminus \{e_{j_1} \cup \dots \cup e_{j_{l+1}}\}) \leq v_1(A_1)$. We then refer bundle $\{e_{j_1}, \dots, e_{j_l}\}$ as S_j , and clearly, for each $j \in \mathcal{J}$, one can construct the corresponding non-empty set S_j . Moreover, we claim that $v_1(S_j) = 0$ for each $j \in \mathcal{J}$; otherwise, reassigning S_j to agent 1 results in another allocation that either has egalitarian welfare larger than

² A leximin solution first selects the allocation that maximizes the value of the least well-off agents, then subject to that, maximizes the value of the second least, and so on.

Table 3 The fair-chores instance for Theorem 3.2

Items	e_1	e_2	\dots	e_m	e_{m+1}
$v_1(\cdot)$	$-\frac{1}{V}$	$-\frac{1}{V}$	\dots	$-\frac{1}{V}$	$-\frac{V-m}{V}$
$v_i(\cdot)$ for $i \geq 2$	$-\frac{2V-1}{2mV}$	$-\frac{2V-1}{2mV}$	\dots	$-\frac{2V-1}{2mV}$	$-\frac{1}{2V}$

$EW(\mathbf{A})$ or has one less agent receiving the value $EW(\mathbf{A})$ than that of \mathbf{A} . We then consider the allocation \mathbf{A}' with $A'_1 = \left(\bigcup_{j \in \mathcal{J}} S_j\right) \cup A_1$, $A'_j = A_j \setminus S_j$ for $j \in \mathcal{J}$ and $A'_j = A_j$ for other j . Due to the construction of \mathcal{J} and $\{S_j\}_{j \in \mathcal{J}}$, allocation \mathbf{A}' achieves the optimal egalitarian welfare, and moreover, the number of agents receiving the value $EW(\mathbf{A})$ in \mathbf{A}' is the same as that of \mathbf{A} . However, in allocation \mathbf{A}' , the total number of items assigned to all agents who receive value $EW(\mathbf{A})$ is $\sum_{j \in \mathcal{J}} |S_j|$ more than that of \mathbf{A} , which contradicts the definition of allocation \mathbf{A} as S_j is non-empty for each $j \in \mathcal{J}$. Therefore, allocation \mathbf{A} must satisfy EQX. \square

We remark that the allocation \mathbf{A} constructed in the proof of Theorem 3.1 is not necessarily leximin. To see this, consider Example 1 in Freeman et al. [25]. There are three agents and a set $E = \{e_1, \dots, e_6\}$ of six goods. The goods e_1, e_2, e_3 are valued at 1 by agent 1 and 0 by agents 2 and 3. The goods e_4, e_5, e_6 are valued at 1 by agents 2 and 3 and at 0 by agent 1. In the constructed allocation, one of e_1, e_2, e_3 is assigned to agent 2 or agent 3, so that the total number of items received by the agents with value 1 is maximized. However, in leximin allocations, all e_1, e_2, e_3 must be assigned to agent 1.

Theorem 3.2 *When allocating chores, the price of EQX and of EQ1 with respect to egalitarian welfare are both infinite.*

Proof Note that EQX is stricter than EQ1, and it suffices to prove the statement for EQ1. Let us consider a fair-chores instance with $n \geq 2$ agents and a set $E = \{e_1, \dots, e_{m+1}\}$ of $m + 1$ chores with $m \geq n$. The valuations are shown in Table 3, where $V > 0$ is arbitrarily large.

Since V is arbitrarily large, we have $-m/V > -1/m + 1/(2mV)$, and so, the unique EWM assigns the first m items to agent 1 and e_{m+1} to any agent i with $i \geq 2$, yielding the maximum egalitarian welfare $-m/V$. But in this allocation, agent 1 violates EQ1 because she still receives less value even eliminating one chore from her bundle. Then, to achieve EQ1, agent 1 cannot receive all the first m items, and thus, the egalitarian welfare of an EQ1 allocation is at most $-1/m + 1/(2mV)$, based on which the price of EQ1 is at least $V/m^2 - 1/(2m^2)$ that approaches to positive infinity as $V \rightarrow +\infty$. \square

According to Theorems 3.1 and 3.2, in goods allocation, both EQX and EQ1 are compatible with EWM, while in chores allocation, achieving EQX or EQ1 sacrifices most of the egalitarian welfare.

3.2 With Utilitarian welfare

On utilitarian welfare, we establish the price of fairness for each of EQX and EQ1. A sharp contrast between goods and chores is also revealed by the results below. Specifically, EQX/EQ1 allocations can provide a bounded welfare guarantee relative to the optimal one in goods allocation, while in the case of chores, the price of fairness is infinite.

Theorem 3.3 *When allocating chores, the price of EQX and of EQ1 with respect to utilitarian welfare are both infinite.*

Proof Note that EQX is stricter than EQ1, and so it suffices to prove that the statement holds for EQ1. Again, we consider the instance constructed in the proof of Theorem 3.2. Since V is arbitrarily large, we have $-1/V > -1/m + 1/(2mV)$ and $-1/(2V) > -1 + m/V$, and as a consequence, in a UWM allocation, the first m items are assigned to agent 1 and e_{m+1} is assigned to agent i with $i \geq 2$, yielding the maximum utilitarian welfare $-(2m+1)/(2V)$. But in such an allocation, agent 1 violates EQ1 because she still receives less value even eliminating one chore from her bundle. Then, to achieve EQ1, agent 1 cannot receive all the first m items, and thus, the utilitarian welfare of an EQ1 allocation is at most $-1/m + 1/(2mV)$, based on which the price of EQ1 is at least $(2V-1)/(2m^2+m) \rightarrow +\infty$ as $V \rightarrow +\infty$. \square

When moving to the case of goods, we distinguish between two cases: $n = 2$ and general $n \geq 3$, and provide (asymptotically) tight results on the price of fairness.

Theorem 3.4 *When allocating goods to two agents, the price of EQX with respect to utilitarian welfare is equal to $3/2$.*

Proof In the proof of Theorem 3.1, we show that, for any instance \mathcal{I} , there exists an EQX allocation achieving the maximum egalitarian welfare. Let \mathbf{A} be such an EQX allocation, and without loss of generality, assume $v_1(A_1) \leq v_2(A_2)$. Thus, the maximum egalitarian welfare is $v_1(A_1)$ and the maximum utilitarian welfare is at most $1 + v_1(A_1)$. Consider another allocation \mathbf{A}' with $A'_1 = A_2$ and $A'_2 = A_1$. Since \mathbf{A} achieves the maximum egalitarian welfare, we must have $v_1(A_1) \geq \min\{v_1(A_2), v_2(A_1)\}$, and accordingly, $v_1(A_1) \geq v_2(A_1)$ implies $v_1(A_1) + v_2(A_2) \geq 1$ due to the normalized valuations; $v_1(A_1) \geq v_1(A_2)$ also implies $v_1(A_1) + v_2(A_2) \geq 1$ due to $v_1(A_1) \leq v_2(A_2)$ and the normalized valuations. Hence, we have $\text{UW}(\mathbf{A}) \geq \max\{2v_1(A_1), 1\}$. Recall that the maximum utilitarian welfare of \mathcal{I} is at most $1 + v_1(A_1)$, then for any $0 \leq v_1(A_1) \leq 1$, we have

$$\text{Price of EQX} \leq \frac{1 + v_1(A_1)}{\max\{2v_1(A_1), 1\}} \leq \frac{3}{2}.$$

For the lower bound, let us consider a fair-goods instance with two agents and a set $E = \{e_1, e_2, e_3\}$ of three goods. The valuations are shown in Table 4, where $\epsilon > 0$ is arbitrarily small. A utilitarian welfare-maximization allocation assigns e_1 to an arbitrary agent and e_2, e_3 to agent 1 and agent 2, respectively, which leads to an optimal utilitarian welfare $3/2 - 2\epsilon$. But in such allocations, the agent who does not receive e_1 violates EQX, and so, in any EQX allocations, one agent only receives e_1 and the other agent receives the rest two goods, yielding utilitarian welfare exactly 1. Therefore, the price of EQX is at least $3/2 - 2\epsilon \rightarrow 3/2$ as $\epsilon \rightarrow 0$. \square

Before we state our result on the price of EQ1 in Theorem 3.5, we first present Algorithm 1, which uses Algorithm 2 as a subroutine and outputs an EQ1 allocation with utilitarian welfare guarantee at least $2/(\sqrt{2} + 1)$ times the maximum one. Intuitively, given a fair-goods instance \mathcal{I} , Algorithm 1 first checks whether a specific partial allocation, in which some items are assigned to the agent having the larger value, can be extended to an EQ1 allocation. If yes, it implements Algorithm 2, a subroutine where in each turn, let the agent

Table 4 The fair-goods instance for Theorem 3.4

Items	e_1	e_2	e_3
$v_1(\cdot)$	$\frac{1}{2}$	$\frac{1}{2} - \epsilon$	ϵ
$v_2(\cdot)$	$\frac{1}{2}$	ϵ	$\frac{1}{2} - \epsilon$

with the smallest current value pick the item of the highest value from the remaining, and makes the partial allocation a complete EQ1 allocation with the maximum utilitarian welfare. If the answer is no, to achieve EQ1, Algorithm 1 carefully reassigns some items from the bundle of the agent with the larger value to the other agent, while avoiding as much welfare loss as possible. For simplicity, in the description of Algorithm 1 and its proof we write $L(k) := \{e_1, \dots, e_k\}$ and $R(k) := \{e_k, \dots, e_m\}$ for any $k \in [m]$.

Algorithm 1

Input: A fair-goods instance $I = \langle [2], E, \mathcal{V} \rangle$.

Output: Allocation **A** of instance I .

- 1: Partition $E = E_0 \cup E_1 \cup E_2$ where $E_1 = \{e \in E \mid v_1(e) > v_2(e)\}$ and $E_2 = \{e \in E \mid v_1(e) < v_2(e)\}$ (assume $v_1(E_1) \leq v_2(E_2)$ and the other case is symmetric).
 - 2: **if** $v_1(E_1 \cup E_0) \geq \min_{e \in E_2} v_2(E_2 \setminus \{e\})$ **then**
 - 3: **A** \leftarrow Greedy($(E_1, E_2), I$);
 - 4: **else**
 - 5: Order goods such that $\frac{v_1(e_1)}{v_2(e_1)} \leq \frac{v_1(e_2)}{v_2(e_2)} \leq \dots \leq \frac{v_1(e_m)}{v_2(e_m)}$, break ties arbitrarily. For good e with $v_1(e) = 0$, put it at the front and good e with $v_2(e) = 0$ at back;
 - 6: Add two virtual items e_0, e_{m+1} with $v_i(e_0) = v_i(e_{m+1}) = 0, \forall i = 1, 2$;
 - 7: Let index s be the one such that $v_2(e_s) > v_1(e_s)$ and $v_2(e_{s+1}) \leq v_1(e_{s+1})$.
 - 8: Find the maximum index $f < s$ such that $v_2(L(f)) \leq v_1(R(f+2))$;
 - 9: $A_1 \leftarrow R(f+2), A_2 \leftarrow L(f+1)$;
 - 10: **end if**
 - 11: **return** **A**
-

Algorithm 2 Greedy (**A'**, I)

Input: An instance $I = \langle [n], E, \mathcal{V} \rangle$ and a partial allocation **A'** of I ;

Output: A complete allocation **A** of instance I ;

- 1: Initialize $E \leftarrow E \setminus \bigcup_{i \in [n]} A'_i$ and $A_i \leftarrow A'_i$ for $i \in [n]$.
 - 2: **while** $E \neq \emptyset$ **do**
 - 3: $i \leftarrow \arg \min_{i \in [n]} |v_i(A_i)|$, break ties arbitrarily;
 - 4: $e \leftarrow \arg \max_{e' \in E} v_i(e')$;
 - 5: Update $A_i \leftarrow A_i \cup \{e\}$ and $E \leftarrow E \setminus \{e\}$;
 - 6: **end while**
 - 7: **return** allocation **A**.
-

Lemma 3.1 *Algorithm1 always terminates and returns an EQ1 allocation.*

Proof If the condition of Step 2 in Algorithm 1 holds, then clearly Algorithm 1 terminates and allocation **A** is returned by Greedy with partial assignment (E_1, E_2) . For any $i \in [2]$, let

$e^{(i)}$ be the last good received by agent i in Greedy, and if $A_i = E_i$, let $e^{(i)} = \emptyset$. The following proof considers two cases. If $A_2 \cap E_0 = \emptyset$, according to Greedy, we have $v_2(A_2) \geq v_1(A_1 \setminus \{e^{(1)}\})$, which implies agent 2 satisfies EQ1. As for agent 1, from the Step 2 of Algorithm 1, we have $v_1(A_1) = v_1(E_1 \cup E_0) \geq \min_{e \in E_2} v_2(E_2 \setminus \{e\})$, and thus agent 1 also meets the condition of EQ1. For the case of $A_2 \cap E_0 \neq \emptyset$, we let $\mathbf{A}^{(i)}$ be the allocation right after agent i receiving item $e^{(i)}$. Then, according to Greedy, when agent 1 just receives $e^{(1)}$, it holds that $v_2(A_2) \geq v_2(A_2^{(1)}) \geq v_1(A_1^{(1)} \setminus \{e^{(1)}\}) = v_1(A_1 \setminus \{e^{(1)}\})$, which implies that agent 2 satisfies EQ1. Similarly, we have $v_1(A_1) \geq v_1(A_1^{(2)}) \geq v_2(A_2^{(2)} \setminus \{e^{(2)}\}) = v_2(A_2 \setminus \{e^{(2)}\})$. Then, agent 1 satisfies EQ1 under allocation \mathbf{A} as well.

When Algorithm 1 goes to Step 5, since valuations are normalized and $v_2(e_0) = 0$, the existence of f is guaranteed. According the order and index s , we have $E_2 = L(s)$ and $E_0 \cup E_1 = R(s + 1)$. Moreover, we can assume without loss of generality $s \geq 2$ because the condition of Step 2 would hold if $s = 1$. For index f , if $f = s - 1$, then $v_2(E_2 \setminus \{e_s\}) = v_2(L(s - 1)) \leq v_1(R(s + 1)) = v_1(E_1 \cup E_0)$ holds, and this relationship satisfies the condition of Step 2, a contradiction. Thus, it must hold that $f \leq s - 2$. We then prove that allocation \mathbf{A} constructed by Step 9 is EQ1. For agent 1, she would not violate EQ1 since $v_1(A_1) = v_1(R(f + 2)) \geq v_2(L(f)) = v_2(A_2 \setminus \{e_{f+1}\})$. As for agent 2, since $f + 1 \leq s - 1$ and $f + 1$ is not chosen by Step 8, then we have $v_2(L(f + 1)) > v_1(R(f + 3))$, equivalent to $v_2(A_2) > v_1(A_1 \setminus \{e_{f+2}\})$. Thus, agent 2 also satisfies EQ1. \square

Theorem 3.5 *When allocating goods to two agents, the price of EQ1 with respect to utilitarian welfare is at least $6/5$, and at most $(\sqrt{2} + 1)/2$.*

Proof We start from the upper bound and consider the allocation \mathbf{A} returned by the Algorithm 1. Based on Lemma 3.1, it suffices to show that the $UW(\mathbf{A})$ is at least $2/(\sqrt{2} + 1)$ times the maximum utilitarian welfare. If allocation \mathbf{A} is returned by Step 3 of Algorithm 1, according to Greedy, we have $E_i \subseteq A_i$ for any $i \in [2]$, which implies $UW(\mathbf{A})$ equals to the optimal utilitarian welfare. We then consider the case where \mathbf{A} is created by Step 9. Denote by $\mathbf{O} = (O_1, O_2)$ an allocation with maximum utilitarian welfare. Clearly, $E_i \subseteq O_i$ for any $i \in [2]$. Then, due to index order in Step 5 and $f \leq s - 2$, it holds that

$$\frac{v_1(A_2)}{v_2(A_2)} \leq \frac{v_1(O_2)}{v_2(O_2)},$$

which then implies the following

$$v_1(A_1) \geq 1 - v_2(A_2) \frac{v_1(O_2)}{v_2(O_2)}.$$

Recall $f \leq s - 2$, then we have $O_1 \subseteq R(f + 3)$, which leads to $v_2(A_2) = v_2(L(f + 1)) > v_1(R(f + 3)) \geq v_1(O_1)$. Then, we have the following:

$$v_1(A_1) + v_2(A_2) \geq 1 + \left(1 - \frac{v_1(O_2)}{v_2(O_2)}\right)v_2(A_2) \geq 1 + \left(1 - \frac{v_1(O_2)}{v_2(O_2)}\right)v_1(O_1),$$

and equivalently,

$$\frac{v_1(O_1) + v_2(O_2)}{v_1(A_1) + v_2(A_2)} \leq \frac{v_1(O_1) + v_2(O_2)}{1 + v_1(O_1)\left(1 - \frac{1-v_1(O_1)}{v_2(O_2)}\right)}$$

We observe that the right hand side is actually a function with two variables $v_1(O_1), v_2(O_2)$, and for simplicity, let $v_1(O_1) = x, v_2(O_2) = y$. As for the domain, since the condition of Step 2 is not satisfied, then $v_2(O_2) \geq v_2(E_2) > v_1(E_1 \cup E_0) \geq v_1(O_1)$, implying $y > x$. Also, $v_2(O_2) > v_1(O_2) = 1 - v_1(O_1)$ due to normalized valuations. Therefore, we have $y > \max\{x, 1 - x\}$. Let $f(x, y)$ corresponds to the right hand side of the above inequality, then we have its derivatives with respect to y ,

$$\frac{\partial f(x, y)}{\partial y} = \frac{(1 + x)y^2 + 2(x^2 - x)y + x^3 - x^2}{\left(1 + x\left(1 - \frac{1-x}{y}\right)\right)^2 y^2}$$

We then let $g(x, y) = (1 + x)y^2 + 2(x^2 - x)y + x^3 - x^2$, and its partial derivative with respect to y is

$$\frac{\partial g(x, y)}{\partial y} = 2(1 + x)y + 2x^2 - 2x$$

The root of equation $\partial g / \partial y = 0$ is $y = \frac{x-x^2}{1+x}$ and inequality $\frac{x-x^2}{1+x} < x$ consistently holds for any $x \in [0, 1)$. Thus, for any $x \in [0, 1)$, function $g(x, y)$ is monotonically increase on y . Accordingly, we have the following:

$$\min_{\max\{x, 1-x\} < y \leq 1} g(x, y) = \begin{cases} 2x^2(2x - 1), & \text{if } x \geq \frac{1}{2} \\ (2x - 1)(x - 1), & \text{if } x < \frac{1}{2} \end{cases}$$

It is not hard to verify that $g(x, y) \geq 0$ consistently holds, which implies that, for any $x \in [0, 1), f(x, y)$ is a monotonically increasing function of y . Then, to find the maxima of $f(x, y)$, we substitute $y = 1$, then by simple calculation, the maximum value of $f(x, 1)$ is equal to $(\sqrt{2} + 1)/2$ which happens when $x = \sqrt{2} - 1$, completing the proof for upper bound.

As for the lower bound, let us consider a fair-goods instance with two agents and a set $E = \{e_1, e_2, e_3\}$ of three items. The valuations are shown in Table 5, where $\epsilon > 0$ is arbitrarily small.

Suppose \mathbf{O} is a UWM, then we have $O_1 = \{e_2, e_3\}$ and $O_2 = \{e_1\}$ yielding $UW(\mathbf{O}) = 3/2 - 2\epsilon$. But agent 2 violates EQ1 in \mathbf{O} due to $v_2(O_2) < v_1(O_1 \setminus \{e\})$ for any $e \in O_1$. Thus, in any EQ1 allocations, agent 1 can not receive both e_2, e_3 , which means the welfare loss of EQ1 allocations is at least $1/4 - \epsilon$. And one can verify that \mathbf{A}' with $A'_1 = \{e_3\}$ and $A'_2 = \{e_1, e_2\}$ is an EQ1 allocation with $UW(\mathbf{A}') = 5/4 - \epsilon$. Therefore, regarding utilitarian welfare, we have

$$\text{Price of EQ1} \geq \frac{\frac{3}{2} - 2\epsilon}{\frac{5}{4} - \epsilon} \rightarrow \frac{6}{5} \text{ as } \epsilon \rightarrow 0,$$

which completes the proof. □

Table 5 The fair-goods instance for Theorem 3.5

Items	e_1	e_2	e_3
$v_1(\cdot)$	0	$\frac{1}{2}$	$\frac{1}{2}$
$v_2(\cdot)$	$\frac{1}{2} - 2\epsilon$	$\frac{1}{4} + \epsilon$	$\frac{1}{4} + \epsilon$

We now consider the case of general $n \geq 3$ and provide asymptotically tight results. Before stating the main result, we first present several lemmas. The following lemma provides a sufficient condition for extending a partial allocation into a complete EQX allocation.

Lemma 3.2 *Given a fair-goods instance \mathcal{I} and a partial allocation \mathbf{A}' of \mathcal{I} , if allocation \mathbf{A}' is EQX and for any i with $A'_i \neq \emptyset$, $\min_{e \in A'_i} v_i(e) \geq \max_{e \in E \setminus \bigcup_{i \in [n]} A'_i} v_i(e)$ holds, then allocation \mathbf{A} returned by Greedy $(\mathbf{A}', \mathcal{I})$ is EQX.*

Proof For the sake of contradiction, we assume that \mathbf{A} is not EQX. Without loss of generality we assume $v_1(A_1) \leq \dots \leq v_n(A_n)$ and agent 1 violates EQX when comparing to agent k , i.e., $\max_{e \in A_k: v_k(e) > 0} v_k(A_k \setminus \{e\}) > v_1(A_1)$. Consider two cases.

Case 1 $A_k \setminus A'_k \neq \emptyset$. Denote by $e^{(k)}$ the last item received by agent k in Greedy and $\mathbf{A}^{(k)}$ the partial allocation right before agent k receiving item $e^{(k)}$, i.e., $A_k = A_k^{(k)} \cup \{e^{(k)}\}$. Then, since $A_1^{(k)} \subseteq A_1$, we have $v_1(A_1) \geq v_1(A_1^{(k)})$. According to the choice of Step 3 in Algorithm 2, it holds that $v_1(A_1^{(k)}) \geq v_k(A_k^{(k)})$, and hence $v_1(A_1) \geq v_k(A_k^{(k)})$. We show that $e^{(k)}$ is the item with the smallest positive value for agent k in bundle A_k . If $A'_k = \emptyset$, since an agent always picks the single item with largest value, item $e^{(k)}$ chosen the last must have a value no larger than any other item in A_k . If $A'_k \neq \emptyset$, the condition $\min_{e \in A'_k} v_k(e) \geq \max_{e \in E \setminus \bigcup_{i \in [n]} A'_i} v_k(e)$ together with the way of picking items can also guarantee $v_k(e^{(k)}) \leq v_k(e)$ for any $e \in A_k$. Accordingly, we have $\max_{e \in A_k} v_k(A_k \setminus \{e\}) = v_k(A_k^{(k)}) \leq v_1(A_1)$, contradicting the assumption that agent 1 violates EQX when comparing to agent k .

Case 2 $A_k \setminus A'_k = \emptyset$. In this case, we have $A_k = A'_k$ as $A'_k \subseteq A_k$. Since $A'_1 \subseteq A_1$ and allocation \mathbf{A}' is EQX, it holds that $v_1(A_1) \geq v_1(A'_1) \geq \max_{e \in A_k: v_k(e) > 0} v_k(A_k \setminus \{e\})$, which again contradicts the assumption that agent 1 violates EQX when comparing to agent k . \square

In the following, we propose algorithm ALG_3 , which efficiently computes an EQX allocation that also has an absolute welfare guarantee. The ALG_3 first assigns one *large item*, with value at least $1/(3n)$, to as many agents as possible, and at the same time, maximizes the welfare of the partial allocation. This is achieved by computing a maximum-weight matching of a bipartite graph. Then, it carefully assigns a bundle to every unmatched agent so that each of them receives value at least $1/(3n)$, while maintaining the partial allocation being EQX. At last, the remaining goods are assigned to agents by running algorithm Greedy. In what follows we formally prove that ALG_3 can efficiently output an EQX allocation with the desired utilitarian welfare guarantee.

Algorithm 3 ALG_3

Input: A fair-goods instance $\mathcal{I} = \langle [n], E, \mathcal{V} \rangle$;

Output: An EQX allocation \mathbf{A} with $UW(\mathbf{A}) \geq \frac{1}{3}$;

- 1: Construct weighted bipartite graph $G = ([n] \cup [m], [n] \times [m])$ where agents vertices on one side and goods vertices on the other side. The edge (i, j) from agent i 's vertex to the vertex of good e_j exists only if $v_i(e_j) \geq \frac{1}{3n}$, and the weight of (i, j) is $v_i(e_j)$.
 - 2: Compute the maximum weight matching μ of G and denote by $\mu(i)$ the good matched to agent i . If agent i is unmatched, let $\mu(i) = \emptyset$. Construct the partial allocation \mathbf{B} with $B_i = \mu(i)$ for every $i \in [n]$, and let $N_0 = \{i \in [n] \mid B_i \neq \emptyset\}$ be the set of matched agents and $E_0 = \bigcup_{i \in N_0} B_i$ the set of matched goods.
 - 3: **if** $UW(\mathbf{B}) \geq \frac{1}{3}$ **then**
 - 4: Compute $\mathbf{A} \leftarrow \text{Greedy}(\mathbf{B}, \mathcal{I})$.
 - 5: **else**
 - 6: $N_1 \leftarrow [n] \setminus N_0$ and $E_1 \leftarrow E \setminus E_0$.
 - 7: **while** $N_1 \neq \emptyset$ **do**
 - 8: For each $i \in N_1$, set the subset $S_i \subseteq E_1$ as the one with minimum cardinality such that $v_i(S_i) \geq \frac{1}{3n}$ and $v_i(S_i \setminus \{e\}) < \frac{1}{3n}$ for any $e \in S_i$. If there is a tie, choose the one with the largest value among all candidates. (We will show that S_i always exists.)
 - 9: Let $\Phi = \{i \in N_1 \mid |S_i| \leq |S_j| \text{ for any } j \in N_1\}$. Find $i^* \in \arg \max_{i \in \Phi} v_i(S_i)$ (break ties arbitrarily) and make the assignment $B_{i^*} \leftarrow S_{i^*}$.
 - 10: Update $E_1 \leftarrow E_1 \setminus S_{i^*}$ and $N_1 \leftarrow N_1 \setminus \{i^*\}$.
 - 11: **end while**
 - 12: Compute $\mathbf{A} \leftarrow \text{Greedy}(\mathbf{B}, \mathcal{I})$.
 - 13: **end if**
 - 14: **return** allocation \mathbf{A} .
-

Lemma 3.3 *Algorithm3 returns in polynomial time an EQX allocation \mathbf{A} with utilitarian welfare $UW(\mathbf{A}) \geq 1/3$.*

Proof We first consider the case where allocation \mathbf{A} is returned in Step 4 of ALG_3 . For this case, allocation \mathbf{B} is the partial allocation established based on matching μ and hence $|B_i| \leq 1$ for $i \in [n]$. Observe that allocation \mathbf{A} is returned by Greedy $(\mathbf{B}, \mathcal{I})$, and thus, $B_i \subseteq A_i$ for any $i \in [n]$. Consequently, we have $UW(\mathbf{A}) \geq UW(\mathbf{B}) \geq 1/3$. What remains to be shown is that allocation \mathbf{A} is EQX. Notice that $|B_i| = 1$ for each matched agent $i \in N_0$ and $B_i = \emptyset$ for $i \in [n] \setminus N_0$. Thus, the partial allocation \mathbf{B} is EQX. For each matched agent $i \in N_0$, if $\exists e \in E \setminus E_0$ such that $v_i(e) > v_i(B_i) \geq 1/(3n)$, then by matching i to the corresponding vertex of e and keeping other matched pair in μ , one can find another matching μ' with a weight larger than that of μ , a contradiction. Thus, for every i with $B_i \neq \emptyset$, it holds that $v_i(B_i) \geq \max_{e \in E \setminus E_0} v_i(e)$. According to Lemma 3.2 and the fact that \mathbf{B} is EQX, we conclude that allocation \mathbf{A} is EQX.

Now consider the case where allocation \mathbf{A} is *not* returned in Step 4 of ALG_3 and hence $UW(\mathbf{B}) < 1/3$. Clearly, not all agents are matched in μ . Since μ is a maximum-weight matching, for any $i \in [n] \setminus N_0$ and $e \in E \setminus E_0$, we have $v_i(e) < 1/(3n)$. Moreover, even if $v_i(e') \geq 1/(3n)$ for some $e' \in E_0$ and $e' = \mu(i')$, it must hold that $v_i(e') \leq v_{i'}(e')$. Accordingly, for each $i \in [n] \setminus N_0$, we have $v_i(E_0) \leq UW(\mathbf{B})$ and thus, $v_i(E \setminus E_0) > 1 - UW(\mathbf{B}) > 2/3$. We then prove that every agent $i \in [n] \setminus N_0$ can receive a bundle S_i in the while-loop of ALG_3 , which is equivalent to showing that the value of the remaining items for agent i is always at least $1/(3n)$. Consider an arbitrary point where the

while-loop starts with remaining items E' and agents N' . For each $j \in [n] \setminus (N_0 \cup N')$ and $i \in N'$, it must hold that $v_i(S_j) < 2/(3n)$; otherwise, at the time when agent j receives a bundle, there exists a subset $S^* \subsetneq S_j$ such that $v_i(S^*) > 1/(3n)$ and $v_i(S^* \setminus \{e\}) < 1/(3n)$ for all $e \in S^*$, and thus, instead of agent j , the algorithm assigns a bundle to agent i , a contradiction. Consequently, for each $i \in N'$, we have

$$\begin{aligned} v_i(E') &= v_i(E) - v_i(E_0) - v_i\left(\bigcup_{j \in [n] \setminus (N_0 \cup N')} S_j\right) \\ &> \frac{2}{3} - \frac{2}{3n} \cdot (n - |N_0| - |N'|) > \frac{2}{3n}, \end{aligned}$$

where the last transition is due to $|N'| \geq 1$. Thus, every agent $i \in [n] \setminus N_0$ is able to receive a bundle that satisfies the conditions described in Step 8.

With a slight abuse of notations, we now let \mathbf{B} be the partial allocation when the while-loop ends. Clearly, we have $v_i(B_i) \geq 1/(3n)$ for every $i \in [n]$, which implies $UW(\mathbf{B}) \geq 1/3$. Since allocation \mathbf{A} is computed by Greedy with input \mathbf{B} , it then holds that $UW(\mathbf{A}) \geq UW(\mathbf{B}) \geq 1/3$. In order to prove that allocation \mathbf{A} is EQX, we use Lemma 3.2. We first claim that allocation \mathbf{B} satisfies EQX. For each agent $i \in [n]$, when comparing to agent $j \in N_0$, it does not violate EQX as $|B_j| = 1$, and when comparing to agent $j \in [n] \setminus N_0$, agent i also satisfies EQX because $v_i(B_i) \geq 1/(3n) > \max_{e \in B_j} v_j(B_j \setminus \{e\})$. Thus, allocation \mathbf{B} is EQX. As for the remaining condition, for any $i \in N_0$, it holds that $v_i(B_i) = \min_{e \in B_i} v_i(e) \geq \max_{e \in E \setminus \bigcup_{i \in [n]} B_i} v_i(e)$ because μ computed in Step 2 is the maximum-weight matching. For an agent $i \in [n] \setminus N_0$, if $\exists e' \in E \setminus \bigcup_{i \in [n]} B_i$ such that $v_i(e') > v_i(e)$ for some $e \in B_i$, then at the moment when agent i is chosen in Step 9 and bundle S_i is constructed, bundle $S_i \cup \{e'\} \setminus \{e\}$ is also a candidate and has the same size as $|S_i|$ but with value strictly larger than $v_i(S_i)$. Thus, instead of S_i , bundle $S_i \cup \{e'\} \setminus \{e\}$ would be assigned to agent i , a contradiction. Therefore, based on Lemma 3.2, we can conclude that allocation \mathbf{A} is EQX and has welfare $UW(\mathbf{A}) \geq 1/3$.

As for time complexity of ALG_3 , since the maximum-weight matching can be computed efficiently, ALG_3 clearly finishes in polynomial time. \square

Now, we are at the stage to present and prove the main statement on the price of EQX and of EQ1 with respect to utilitarian welfare.

Theorem 3.6 *When allocating goods to n agents, the price of EQ1 and of EQX with respect to utilitarian welfare are at least $n - 1$ and at most $3n$, asymptotically tight $\Theta(n)$.*

Proof We start from the upper bound. According to Lemma 3.3, there exists an EQX allocation with welfare at least $1/3$. Due to the normalized valuations, the optimal utilitarian welfare is at most n , and thus, the price of EQX is at most $3n$. This upper bound also holds for the price of EQ1 as the notion of EQX is stricter than EQ1.

As for the lower bound, it suffices to prove that the statement holds for EQ1. Let us consider a fair-goods instance \mathcal{I} with n agents and a set $E = \{e_1, \dots, e_{(n-2)p+1}\}$ of $(n-2)p+1$ goods where $p \in \mathbb{N}^+$ is arbitrarily large. For agent $i \in [n-2]$, her valuation function is: $v_i(e_j) = 1/p$ if $(i-1)p+1 \leq j \leq ip$ and $v_i(e_j) = 0$ for other j . Both agents $n-1$ and n value $e_{(n-2)p+1}$ at 1 and other items at 0. In a utilitarian welfare-maximizing

allocation \mathbf{O} , bundle $O_i = \{e_{(i-1)p+1}, \dots, e_{ip}\}$ is assigned to agent i for all $i \in [n-2]$. One of agents $n-1$ and n receives $e_{(n-2)p+1}$, which leaves the other receiving an empty set. The optimal utilitarian welfare equals $\text{UW}(\mathbf{O}) = n-1$. Note that the agent receiving an empty set violates EQ1 in \mathbf{O} when comparing to every agent $i \in [n-2]$. To achieve EQ1, each agent $i \in [n-2]$ can receive at most one of the goods on which she has non-zero value, and accordingly, the welfare of an EQ1 allocation is at most $1 + (n-2)/p$. Therefore, regarding utilitarian welfare, we have

$$\text{Price of EQ1} \geq \frac{n-1}{1 + \frac{n-2}{p}} \rightarrow n-1 \text{ as } p \rightarrow \infty,$$

which completes the proof. \square

According to Caragiannis et al. [19], when assigning indivisible goods, the price of EQ is infinite for general n and two for the case of $n=2$. From our results, the prices of EQX and EQ1 are smaller than that of EQ in both cases of general n and two agents. Moreover, the price of EQX is greater than that of EQ1 when $n=2$. Such observations confirm an intuition in stating that if one relaxes the fairness requirement, then less welfare would be sacrificed.

Equitability aims to reduce the difference between agents' value, while utilitarianism may lead to unbalanced outcomes. This intuition suggests that approximately equitable allocations may have a poor performance in guaranteeing utilitarian welfare, and Theorems 3.3 and 3.6 can be evidence for this intuition. Egalitarianism also aims for "balancing" agents' value, and accordingly, equitability is highly likely to have a considerable guarantee on egalitarian welfare. However, while Theorem 3.1 confirms this guess in goods allocation, Theorem 3.2 states that the price of EQ1 with respect to egalitarian welfare is infinite in the case of chores. We believe the reason behind is that in the case of chores, agents' value are balanced around zero, and the price of fairness as the ratio of two negligible numbers can be enormous.

4 Results on computational complexity

Results on the price of fairness not only quantify the efficiency loss under fairness constraints but also can derive answers to the existence of a welfare maximizer that is also fair. In particular, our results on the price of fairness suggest that relaxed equitability is not always compatible with utilitarian welfare in either goods or chores and not always compatible with egalitarian welfare in chores allocation. These impossibilities motivate two crucial follow-up algorithmic problems, i.e., given an instance, whether one can efficiently determine the existence of fair and welfare-maximizing allocations and whether one can efficiently compute the allocation with maximum welfare among the set of fair allocations. In this section, we settle the computational complexity of these two algorithmic problems.

We introduce some notation for ease of reference to a problem. For a fairness criterion $F \in \{\text{EQX}, \text{EQ1}\}$ and welfare objective $W \in \{\text{UW}, \text{EW}\}$, denote by " $\text{E}(W \times F)$ " the problem of deciding whether there exists an F allocation that also maximizes W , and denote by " $\text{C}(W/F)$ " the problem of computing an F allocation that maximizes W among all F allocations. For example, $\text{E}(\text{UW} \times \text{EQX})$ refers to the problem of deciding the existence of an EQX allocation that also maximizes utilitarian welfare, and $\text{C}(\text{EW}/\text{EQ1})$ denotes the problem of computing an EQ1 allocation that maximizes the egalitarian welfare among all EQ1 allocations.

To establish (strong) NP-hardness, we provide polynomial time reductions from two well-known problems; that is, 3-PARTITION and PARTITION, described as below. According to Garey and Johnson [27], the problem 3-PARTITION is strongly NP-complete, and the problem PARTITION is NP-complete.

3-PARTITION: given a non-empty finite set $B = \{b_i : i \in I = \{1, \dots, 3m\}\}$ of $3m$ positive integers and another positive integer T such that $T/4 < b_i < T/2$ for any $i \in [3m]$ and $\sum_{i \in I} b_i = mT$, can I be partitioned into m disjoint subsets I_1, \dots, I_m such that $\sum_{i \in I_k} b_i = T$ for any $k \in [m]$?

PARTITION: given a non-empty finite set $P = \{p_i : i \in I = \{1, \dots, m\}\}$ of m positive integers such that $\sum_{i \in I} p_i = 2T$, can I be partitioned into two disjoint subsets I_1, I_2 such that $\sum_{i \in I_1} p_i = \sum_{i \in I_2} p_i = T$?

4.1 Non-equivalence between goods and chores

As we already mentioned, the results for goods and chores are not mirror images of one another, which is supported by our results on the price of fairness. Before diving deep into studying the complexity, we argue that even being restricted to the algorithmic problems we are concerned with, the chores problem may not be equivalent to the corresponding goods version, neither do the other direction.

Proposition 4.1 *For any fairness criterion $F \in \{\text{EQX}, \text{EQ1}\}$, there is no mapping $f : [-1, 0] \rightarrow \mathbb{R}_+ \cup \{0\}$ such that a fair-chores instance $\mathcal{I}^c = \langle [n], E, \mathcal{V} \rangle$ admits an F and utilitarian welfare-maximizing allocation if and only if the fair-goods instance $\mathcal{I}^g = \langle [n], E, f(\mathcal{V}) \rangle$ admits an F and utilitarian welfare-maximizing allocation.*

Proof For the sake of contradiction, assume such a mapping f exists. We first consider instance $\mathcal{I}_1^c = \langle [2], E^1, \mathcal{V}^1 \rangle$ with two agents and three items $E^1 = \{e_1^1, e_2^1, e_3^1\}$. The valuation function of agent 1 is: $v_1^1(e_1^1) = v_1^1(e_2^1) = -1/2$ and $v_1^1(e_3^1) = -1$ and of agent 2 is: $v_2^1(e_1^1) = v_2^1(e_2^1) = -1$ and $v_2^1(e_3^1) = 0$. It is not hard to verify that \mathcal{I}_1^c admits a unique UWM, in which agent 1 violates fairness criterion F . Accordingly, no UWM of \mathcal{I}_1^c satisfies F . We then consider the corresponding fair-goods instance $\mathcal{I}_1^g = \langle [2], E^1, f(\mathcal{V}^1) \rangle$, in which the value of agent i on good $e \in E^1$ is $f(v_i^1(e))$. Due to the definition of mapping f , it must hold that instance \mathcal{I}_1^g does not admit F and utilitarian welfare-maximizing allocations, either. Next, we discuss the several cases classified by the possible relationship among $f(-1), f(-1/2)$ and $f(0)$.

If $f(-1) \geq \max\{f(-1/2), f(0)\}$, then assigning e_1^1, e_2^1 to agent 2 and e_3^1 to agent 1 results in an F allocation that is also a UWM, a contradiction.

If $f(-1/2) > f(-1) \geq f(0)$, we consider another fair-chores instance $\mathcal{I}_2^c = \langle [2], E^2, \mathcal{V}^2 \rangle$ with six items $E^2 = \{e_1^2, \dots, e_6^2\}$ and its corresponding fair-goods instance $\mathcal{I}_2^g = \langle [2], E^2, f(\mathcal{V}^2) \rangle$. Under instance \mathcal{I}_2^c , function $v_1^2(\cdot)$ is: $v_1^2(e_j^2) = -1/2$ for $j = 1, 2, 3, 6$ and $v_1^2(e_j^2) = -1$ for $j = 4, 5$, and $v_2^2(\cdot)$ is: $v_2^2(e_j^2) = -1$ for $j \in [4]$ and $v_2^2(e_j^2) = 0$ for $j = 5, 6$. Notice \mathcal{I}_2^c admits an F and utilitarian welfare-maximizing allocation \mathbf{A} with $A_1 = \{e_1^2, e_2^2, e_3^2\}$ and $A_2 = E^2 \setminus A_1$. Due to the construction of f , instance \mathcal{I}_2^g should also admit an F and utilitarian welfare-maximizing allocation. However, in any UWM of instance \mathcal{I}_2^g , since $f(-1/2) > f(-1) \geq f(0)$, agent 2 violates fairness criterion F , a contradiction.

If $f(-1/2) \geq f(0) > f(-1)$, one can verify that \mathcal{I}_1^g does not admit an allocation that is

both F and UWM only if $f(-1/2) > f(0) > f(-1)$ holds. Again, we consider instances \mathcal{I}_2^c and \mathcal{I}_2^g . Under the relationship of $f(-1/2) > f(0) > f(-1)$, in any UWM of instance \mathcal{I}_2^g , agent 2 does not satisfy F , while allocation \mathbf{A} (defined above) is a UWM of instance \mathcal{I}_2^c and satisfies F , a contradiction.

If $f(0) > f(-1/2) \geq f(-1)$, for instance \mathcal{I}_1^g , assigning e_1^1, e_2^1 to agent 1 and e_3^1 to agent 2 leads to a UWM that also satisfies F . This contradicts the fact that \mathcal{I}_1^c does not have an F and utilitarian welfare-maximizing allocation.

If $f(0) > f(-1) > f(-1/2)$, we consider another fair-chores instance $\mathcal{I}_3^c = \langle [2], E^3, \mathcal{V}^3 \rangle$ with five items $E^3 = \{e_1^3, \dots, e_5^3\}$ and its corresponding fair-goods instance $\mathcal{I}_3^g = \langle [2], E^3, f(\mathcal{V}^3) \rangle$. In instance \mathcal{I}_3^c , function $v_1^3(\cdot)$ is: $v_1^3(e_j^3) = 0$ for $j \in [3]$ and $v_1^3(e_j^3) = -1$ for $j = 4, 5$ and function $v_2^3(\cdot)$ is: $v_2^3(e_j^3) = -1/2$ for $j \in [4]$ and $v_2^3(e_5^3) = 0$. It is not hard to verify that any UWM of \mathcal{I}_3^c satisfies fairness criterion F . But in a UWM of \mathcal{I}_3^g , agent 1 has value $3f(0) + f(-1)$ and agent 2 has value $f(0)$. Hence, agent 2 violates F in the unique UWM of \mathcal{I}_3^g , and consequently, instance \mathcal{I}_3^g does not admit a UWM that also satisfies F , a contradiction.

Overall, no possible relationship among $f(-1), f(-1/2)$ and $f(0)$ can make such a mapping f exist, completing the proof. \square

The above assertion indicates that a fair-chores instance can not be transformed to a fair-goods instance by a unified mapping on valuations so that two instances admit the same answer to the question of whether there exists a UWM that also satisfies the notion of F . This statement, together with the price of fairness with respect to egalitarian welfare, motivates us to study the complexity of the existence and optimization problem from both goods and chores perspectives.

4.2 Computational complexity with variable number of agents

We first consider the case of general n agents and study the utilitarian welfare. For the problem of deciding the existence of an EQX and utilitarian welfare-maximizing allocation, we establish the strong NP-hardness in both goods and chores. For some statements, we present the proof regarding chores after the statement, while that for goods is in Appendix A.

Theorem 4.1 *For both goods and chores, the decision problem $E(UW \times EQX)$ is strongly NP-complete.*

Proof of Theorem 4.1 for chores The decision problem is in NP as both utilitarian welfare maximization and EQX can be examined in polynomial time. We then derive a reduction from the problem 3-PARTITION.

Given an arbitrary instance of 3-PARTITION, we construct a fair-chores instance as follows. There are $m + 1$ agents and a set $E = \{e_1, \dots, e_{3m+2}\}$ of $3m + 2$ items. The valuations are shown in Table 6.

It is not hard to verify that an allocation is a UWM if and only if the first $3m$ chores are assigned to the first m agents and the last two chores are assigned to agent $m + 1$.

Suppose we have a “yes” instance of 3-PARTITION, and let I_1, \dots, I_m be a solution. Then, consider the allocation \mathbf{A} with $A_i = \bigcup_{j \in I_i} e_j$ for any $i \in [m]$ and $A_{m+1} = \{e_{3m+1}, e_{3m+2}\}$. It is straightforward to see that allocation \mathbf{A} has the maximum utilitarian welfare. As for the fairness requirement, since $v_i(A_i) = -T$ for any $i \in [m]$ and

Table 6 The fair-chores instance for Theorem 4.1

Items	e_1	e_2	\dots	e_{3m-1}	e_{3m}	e_{3m+1}	e_{3m+2}
$v_i(\cdot)$ for $i \leq m$	$-b_1$	$-b_2$	\dots	$-b_{3m-1}$	$-b_{3m}$	$-(m+1)T$	$-(m+1)T$
$v_{m+1}(\cdot)$	$-T$	$-T$	\dots	$-T$	$-T$	$-T$	$-T$

$v_{m+1}(A_{m+1}) = -2T$, none of the first m agents would violate the condition of EQX. Moreover, for any $e \in A_{m+1}$, we have $v_{m+1}(A_{m+1} \setminus \{e\}) = v_i(A_i)$ for any $i \in [m]$. Thus, allocation \mathbf{A} is a UWM and satisfies EQX.

Now we prove the other direction. Suppose we have a “no” instance of 3-PARTITION. Notice that in any UWM, agent $m + 1$ receives value $-2T$. Because it is a “no” 3-PARTITION instance, assigning the first $3m$ chores to the first m agents always results in an allocation in which at least one agent receives value strictly larger than $-T$. Through the comparison between agent $m + 1$ and the agent receiving the largest value, agent $m + 1$ violates the condition of EQX in any UWM, completing the proof. \square

For the notion of EQ1, the strong NP-hardness on the decision problem is also established for both goods and chores.

Theorem 4.2 *For both goods and chores, the decision problem $E(UW \times EQ1)$ is strongly NP-complete.*

Proof of Theorem 4.2 for chores The arguments in the proof of Theorem 4.1 for chores can be carried over to the decision problem $E(UW \times EQ1)$. \square

Theorems 4.1 and 4.2 indicate that although the chores and goods versions of the decision problem (with respect to utilitarian welfare) are not equivalent in general, there surprisingly exist some similarities; that is, neither of them has pseudo-polynomial time algorithms. When concerning egalitarian welfare, results are different under these two settings. According to Theorem 3.1, the price of EQX and of EQ1 with respect to egalitarian welfare are both 1 in goods allocation, which implies that the existence of a goods allocation that is an EWM and also satisfies EQX (or EQ1) is guaranteed. Then, the answer to both problems $E(EW \times EQX)$ and $E(EW \times EQ1)$ is “yes” in the case of goods; however, finding the egalitarian welfare-maximizing allocation that satisfies EQX (or EQ1) is computationally hard as shown in our Theorem 4.15 later. On the other hand, when assigning chores, as shown by the results below, deciding the existence of an EQX and egalitarian welfare-maximizing allocation is computationally intractable.

Theorem 4.3 *When allocating chores, the decision problem $E(EW \times EQX)$ is strongly NP-hard.*

Proof Given an arbitrary instance of 3-PARTITION, we construct a fair-chores instance as follows. There are $m + 1$ agents and a set $E = \{e_1, \dots, e_{3m+2}\}$ of $3m + 2$ chores. The valuation functions are shown in Table 7.

Consider an allocation $\hat{\mathbf{A}}$ with $\hat{A}_i = \{e_{3i-2}, e_{3i-1}, e_{3i}\}$ for $i \in [m]$ and $\hat{A}_{m+1} = \{e_{3m+1}, e_{3m+2}\}$. Due to $T/4 < b_i < T/2$, we have $EW(\hat{\mathbf{A}}) > -3T/2$, so that the

Table 7 The fair-chores instance for Theorem 4.3

Items	e_1	e_2	\dots	e_{3m}	e_{3m+1}	e_{3m+2}
$v_i(\cdot)$ for $i \leq m$	$-b_1$	$-b_2$	\dots	$-b_{3m}$	$-\frac{(5m+1)T+1}{2}$	$-\frac{(5m+1)T+1}{2}$
$v_{m+1}(\cdot)$	$-2T$	$-2T$	\dots	$-2T$	$-T$	-1

maximum egalitarian welfare of this instance is larger than $-3T/2$. Thus, in any EWM, agent $m + 1$ must receive exactly the last two chores, resulting in value $-T - 1$ for her.

Suppose we have a “yes” instance of 3-PARTITION, and let I_1, \dots, I_m be a solution. Then, consider allocation \mathbf{A} with $A_i = \bigcup_{j \in I_i} e_j$ for $i \in [m]$ and $A_{m+1} = \{e_{3m+1}, e_{3m+2}\}$. One can compute $\text{EW}(\mathbf{A}) = -T - 1$. Since agent $m + 1$ must receive value $-T - 1$ in an EWM, allocation \mathbf{A} clearly achieves the maximum egalitarian welfare. As for fairness constraint, since for any $i \in [m]$, $v_i(A_i) = -T$ and $v_{m+1}(A_{m+1}) = -T - 1$, none of the first m agents would violate EQX. For agent $m + 1$, it holds that $v_{m+1}(A_{m+1} \setminus \{e\}) \geq v_i(A_i) = -T, \forall i \in [m], \forall e \in A_{m+1}$. Thus, allocation \mathbf{A} is an EWM and also satisfies EQX.

We now prove the other direction. Suppose we have a “no” instance of 3-ARTITION. Recall that in any EWM, agent $m + 1$ must receive value exactly $-T - 1$. Because it is a “no” instance, assigning the first $3m$ items to the first m agents always results in an allocation in which at least one agent receives value larger than $-T$. Through the comparison between agent $m + 1$ and the agent receiving the largest value, one can easily verify that no EWM can also be EQX, completing the proof. \square

For the problem in which the quantity of optimal egalitarian welfare is involved, since computing an EWM is NP-hard [11], which is different from the utilitarian welfare, one may not able to find the maximum egalitarian welfare in polynomial time. As a consequence, we are unclear whether verifying a “yes” instance of $E(\text{EW} \times \text{EQX})$ can be done in polynomial time, based on which we only state hardness in Theorem 4.3. For the same reason, we also only state hardness in the following theorem.

Theorem 4.4 *When allocating chores, the decision problem $E(\text{EW} \times \text{EQ1})$ is strongly NP-hard.*

Proof Given an arbitrary instance of 3-PARTITION, we construct a fair-chores instance as follows. There are $m + 1$ agents and a set $E = \{e_1, \dots, e_{3m+2}\}$ of $3m + 2$ items. The valuation functions are shown in Table 8. We first consider an allocation that assigns three of $\{e_1, \dots, e_{3m}\}$ to each agent $i \in [m]$, and assigns e_{3m+1} and e_{3m+2} to agent $m + 1$. Due to $b_j < T/2$ for any $j \in [3m]$, the value of agent i is larger than $-3T/2$ for each $i \in [m]$. The value of agent $m + 1$ is equal to $-2T$, and hence the egalitarian welfare of this allocation is equal to $-2T$. Due to the value of e_{3m+1} and e_{3m+2} , no allocation can achieve egalitarian welfare larger than $-2T$, which then implies that the maximum egalitarian welfare of the constructed fair-chores instance is equal to $-2T$.

Suppose we have a “yes” instance of 3-PARTITION, and let I_1, \dots, I_n be the solution. Consider an allocation \mathbf{A} with $A_i = \bigcup_{j \in I_i} e_j$ for any $i \in [m]$ and $A_{m+1} = \{e_{3m+1}, e_{3m+2}\}$. It is not hard to see that $v_i(A_i) = -T$ for any $i \in [m]$ and $v_{m+1}(A_{m+1}) = -2T$, then we have $\text{EW}(\mathbf{A}) = -2T$, which implies that the allocation \mathbf{A} is an EWM. As for the fairness

Table 8 The fair-chores instance for Theorem 4.4

Items	e_1	e_2	\dots	e_{3m}	e_{3m+1}	e_{3m+2}
$v_i(\cdot)$ for $i \leq m$	$-b_1$	$-b_2$	\dots	$-b_{3m}$	$-4mT$	$-4mT$
$v_{m+1}(\cdot)$	$-\frac{9m-2}{3m}T$	$-\frac{9m-2}{3m}T$	\dots	$-\frac{9m-2}{3m}T$	$-T$	$-T$

constraint, none of the first m agents would violate EQ1, and for agent $m + 1$, we have $v_{m+1}(A_{m+1} \setminus \{e\}) = v_i(A_i) = -T$ for any $i \in [m]$ and $e \in A_{m+1}$. Thus, the allocation \mathbf{A} is also EQ1.

We now prove the other direction. Suppose we have a “no” instance of 3-PARTITION. Recall that in any EWM, agent $m + 1$ must receive exactly items e_{3m+1} and e_{3m+2} , and must have value exactly $-2T$. Due to “no” instance, there exists an agent receiving a value strictly larger than $-T$ from the assignment of the first $3m$ chores. Through the comparison between agent $m + 1$ and the agent receiving the largest value, one can verify that no EWM can satisfy EQ1, completing the proof. \square

The above results show that, except for concerning egalitarian welfare in goods allocation, all other decision problems are strongly NP-hard in the case of general n agents. These results directly imply the NP-hardness of computing the allocation with maximum utilitarian welfare among all EQX (or EQ1) allocations.

Theorem 4.5 *For both goods and chores, problems $C(UW/EQX)$ and $C(UW/EQ1)$ are strongly NP-hard.*

Proof To prove the strong NP-hardness of $C(UW/EQX)$, we use the strong NP-hardness of $E(UW \times EQX)$ established by Theorem 4.1. For the sake of contradiction, if there exists a (pseudo) polynomial time algorithm ALG for problem $C(UW/EQX)$, then we can compute the maximum UW among all EQX allocations. Notice that an UWM (without EQX requirement) can be found in polynomial time by assigning each item to the agent who has the largest value on it, and so, one can efficiently find the optimal utilitarian welfare. By comparing the optimal utilitarian welfare with the output from ALG , the problem $E(UW \times EQX)$ is then solved in (pseudo) polynomial time, a contradiction. Based on Theorem 4.2, similar arguments can be applied to prove the strong NP-hardness of $C(UW/EQ1)$. \square

Theorem 4.6 *When allocating goods and chores, problems $C(EW/EQ1)$ and $C(EW/EQX)$ are strongly NP-hard.*

Proof It suffices to prove strong NP-completeness of the decision version of $C(EW/EQ1)$ (resp., $C(EW/EQX)$): given an instance $\mathcal{I} = \langle [n], E, \mathcal{V} \rangle$ and a threshold value W , does there exist an EQ1 (resp., EQX) allocation with egalitarian welfare at least W ?

For both goods and chores, we provide a reduction from 3-PARTITION. Given an arbitrary instance of 3-PARTITION, we construct a fair-goods (resp. fair-chores) instance with m agents and a set $E = \{e_1, \dots, e_{3m}\}$ of $3m$ goods (resp. chores). Agents have identical valuation functions and in the fair-goods instance, $v_i(e_j) = b_j$ for any $i \in [m]$ and $j \in [3m]$; in the fair-chores instance, $v_i(e_j) = -b_j$ for any $i \in [m]$ and $j \in [3m]$. The

threshold value is defined as: $W = T$ in the fair-goods instance; $W = -T$ in the fair-chores instance. Then, in the both cases of goods and chores, there exists an EQ1 (or EQX) allocation with egalitarian welfare at least W if and only if the 3-PARTITION instance is a “yes” instance. \square

Results of this section completely answer the computational complexity of the decision and computation problem we are concerned with for general n agents. Most problems are computationally intractable, which means that one can not efficiently determine the existence of a fair and welfare-maximizing allocation or compute the allocation with maximum welfare among all fair allocations. In addition, our results provide more insights into the similarities between goods and chores. According to Theorems 4.1 and 4.2, for both EQX and EQ1, together with utilitarian welfare, problems for goods and chores share the same complexity. When concerning egalitarian welfare, we show that decision problems for chores are strongly NP-hard and one can directly answer “yes” in the case of goods, but this may not indicate goods and chores have different algorithmic features. Instead of linking to algorithmic property, such an “inconsistency” is due to distinct structural properties, i.e., optimal egalitarian welfare is compatible with EQX/EQ1 for goods but not for chores.

4.3 Computational complexity with fixed number of agents

The above complexity results are for general n , but in practice, the number of agents is usually fixed. Theoretically, even a system with two agents, $n = 2$, can already yield valuable and non-trivial results, especially in machine scheduling [1, 2], an application of chores allocation. This observation motivates us to resolve the complexity for the case where n is fixed.

We carry out the analysis for the case of small and fixed n from utilitarian welfare and EQX, and show NP-completeness for both goods and chores.

Theorem 4.7 *For both goods and chores, the decision problem $E(UW \times EQX)$ is NP-complete, even for two agents.*

Proof of Theorem 4.7 for chores To prove NP-hardness, we derive a reduction from the problem PARTITION. Given an arbitrary instance of PARTITION, we construct a fair-chores instance with two agents and a set $E = \{e_1, \dots, e_{m+4}\}$ of $m + 4$ chores. The valuation functions are shown in Table 9, where $0 < \epsilon < 1$. It is not hard to verify that an allocation is a UWM if and only if chores e_{m+1} and e_{m+2} are assigned to agent 2 and e_{m+3}, e_{m+4} are assigned to agent 1.

Suppose we have a “yes” instance of PARTITION, and let I_1, I_2 be a solution. Then, consider an allocation \mathbf{A} in which A_1 contains bundle $\bigcup_{j \in I_1} e_j$ and e_{m+3}, e_{m+4} , and $A_2 = E \setminus A_1$. Clearly, allocation \mathbf{A} is a UWM. Moreover, for agents’ value, we have

Table 9 The fair-chores instance for Theorem 4.7

Items	e_1	e_2	\dots	e_m	e_{m+1}	e_{m+2}	e_{m+3}	e_{m+4}
$v_1(\cdot)$	$-p_1$	$-p_2$	\dots	$-p_m$	$-T$	$-T$	$-\epsilon$	$-\epsilon$
$v_2(\cdot)$	$-p_1$	$-p_2$	\dots	$-p_m$	$-\epsilon$	$-\epsilon$	$-T$	$-T$

$v_1(A_1) = v_2(A_2) = -T - 2\epsilon$. Thus, allocation **A** is also EQX.

We now prove the other direction. Suppose we have a “no” instance of PARTITION. Let $S = \{e_1, \dots, e_m\}$ be the set of first m items, and recall $\Pi_2(S)$ be the set of 2-partition of S . We denote by $\Delta = \min_{B \in \Pi_2(S)} |v_1(B_1) - v_2(B_2)|$, and claim that $\Delta \geq 1$ because $p_i \in \mathbb{N}$ for any $i \in [m]$ and it’s a “no” instance. Notice that any utilitarian welfare-maximizing allocation **O** has the form of $O_1 = B_1 \cup \{e_{m+3}, e_{m+4}\}$ and $O_2 = B_2 \cup \{e_{m+1}, e_{m+2}\}$ where $\{B_1, B_2\} \in \Pi_2(S)$. Without loss of generality we assume $v_2(O_2) > v_1(O_1)$, and accordingly, due to the definition of Δ and ϵ , we have $v_1(O_1 \setminus \{e_{m+3}\}) - v_2(O_2) \leq \epsilon - \Delta < 0$. So agent 1 violates EQX in allocation **O**, and therefore, no UWM can satisfy EQX, completing the proof. \square

The theorem below follows from Theorem 4.7.

Theorem 4.8 *For both goods and chores, the problem $C(UW/EQX)$ is NP-hard, even with two agents.*

We then move to EQ1, a notion weaker than EQX, and show that $E(UW \times EQ1)$ is NP-complete when $n \geq 3$, but in P when only two agents are involved.

Theorem 4.9 *For both goods and chores, the decision problem $E(UW \times EQ1)$ is NP-complete, even for three agents.*

Proof of Theorem 4.9 for chores The problem is in NP as both utilitarian welfare-maximization and EQ1 can be examined in polynomial time. To prove NP-hardness, we derive a reduction from problem PARTITION. Given an arbitrary instance of PARTITION, we construct a fair-chores instance for three agents with a set $E = \{e_1, \dots, e_{m+3}\}$ of $m + 3$ chores. The valuation functions are shown in Table 10. An allocation of the constructed instance is a UWM if and only if the first $m + 2$ chores are assigned to the first two agents and the last chore is assigned to agent 3.

Suppose we have a “yes” instance of PARTITION, and let I_1, I_2 be a solution. Then, consider allocation **A**, in which A_1 contains bundle $\bigcup_{j \in I_1} e_j$ and e_{m+1} ; A_2 contains bundle $\bigcup_{j \in I_2} e_j$ and e_{m+2} ; $A_3 = \{e_{m+3}\}$. Clearly, allocation **A** is a UWM. As for agents’ value, it holds that $v_1(A_1) = v_2(A_2) < v_3(A_3)$, and moreover, $v_i(A_i \setminus \{e_{m+i}\}) = v_3(A_3)$ for any $i \in [2]$. Thus, allocation **A** is a UWM that also satisfies EQ1.

We now prove the other direction. Suppose we have a “no” instance of PARTITION. Since for any $j \in [m]$, $p_j < 2T$, then agent 3 receives value exactly $-T$ in any UWM. Based on the assignment of chores e_{m+1} and e_{m+2} , we discuss two cases: (i) allocations with exactly one $-5T$ chore for each $i \in [2]$; (ii) other assignments. If case (ii) happens, without loss of generality assume agent 1 receives two $-5T$ chores. Then, when comparing agent 1 and agent 3, agent 1 still receives less value no matter which chore is removed from his

Table 10 The fair-chores instance for Theorem 4.9

Items	e_1	e_2	\dots	e_m	e_{m+1}	e_{m+2}	e_{m+3}
$v_i(\cdot)$ for $i = 1, 2$	$-p_1$	$-p_2$	\dots	$-p_m$	$-5T$	$-5T$	$-2mT - 9T$
$v_3(\cdot)$	$-2T$	$-2T$	\dots	$-2T$	$-10T$	$-10T$	$-T$

bundle, violating the condition of EQ1. Consequently, to make an allocation be both EQ1 and utilitarian welfare-maximizing, case (i) must happen. Moreover, since it is a “no” instance of PARTITION, assigning the first m chores to the first two agents always results in allocations in which there exists one agent with value strictly less than $-T$. Then, when comparing to agent 3, the agent receiving the least value violates EQ1. Thus, no UWM can satisfy EQ1, completing the proof. \square

The theorem below follows from Theorem 4.9.

Theorem 4.10 *For both goods and chores, problem $C(UW/EQ1)$ is NP-hard, even for three agents.*

Next, we provide a polynomial-time algorithm that can determine the existence of an EQ1 and utilitarian welfare-maximizing allocation in the case of two agents. In the algorithm, we first guarantee the allocation achieving maximum utilitarian welfare by assigning each item to the agent who values it the most, and then for unassigned items (if exist), in each round, allocate one of them to the agent with a smaller absolute value via Algorithm 2.

Algorithm 4 Computing the UWM that satisfies EQ1 if one exists

Input: An instance $\mathcal{I} = \langle [2], E, \mathcal{V} \rangle$.

- 1: Let $E = E_0 \cup E_1 \cup E_2$ with $E_1 = \{e \in E \mid v_1(e) > v_2(e)\}$ and $E_2 = \{e \in E \mid v_1(e) < v_2(e)\}$.
 - 2: $\mathbf{A} \leftarrow \text{Greedy}((E_1, E_2), \mathcal{I})$;
 - 3: **if** allocation $\mathbf{A} = (A_1, A_2)$ is EQ1 **then**
 - 4: **return** yes and \mathbf{A} .
 - 5: **else**
 - 6: **return** no.
 - 7: **end if**
-

Theorem 4.11 *For both goods and chores allocation with two agents, there exists a polynomial time algorithm that solves problem $E(UW \times EQ1)$*

Proof of Theorem 4.11 for chores The proof uses Algorithm 4. It is not hard to verify that assigning E_i to agent i for all i is a necessary and sufficient condition for the output having maximum utilitarian welfare.

Denote by $\mathbf{A} = (A_1, A_2)$ with $A_1 = \{e_1^{(1)}, \dots, e_{k_1}^{(1)}\}$ and $A_2 = \{e_1^{(2)}, \dots, e_{k_2}^{(2)}\}$. Clearly, we have $k_1 + k_2 = m$. Term $e_j^{(1)}$ refer to the j -th chore assigned by Algorithm 4 to agent 1 and $e_l^{(2)}$ is the l -th chore assigned to agent 2. If the algorithm terminates at Step 4, clearly we find a UWM that satisfies EQ1. We now consider the case where algorithm terminates at Step 6, and show if so, no UWM is EQ1. We first claim that, in this case, either $E_0 \subseteq A_1$ or $E_0 \subseteq A_2$ holds. It suffices to show that if both $A_1 \cap E_0$ and $A_2 \cap E_0$ are not empty set, Algorithm 4 terminates at Step 4. To prove this claim, we discuss two cases. If $|v_1(A_1)| > |v_2(A_2)|$, since $A_1 \cap E_0 \neq \emptyset$, we have $e_{k_1}^{(1)} \in E_0$. Then by algorithm, it must hold that $|v_1(A_1 \setminus \{e_{k_1}^{(1)}\})| \leq |v_2(A_2)|$; otherwise, agent 1 would not receive chore $e_{k_1}^{(1)}$. Accordingly, allocation \mathbf{A} is EQ1, and Algorithm 4 terminates at Step 4. On the other hand, if $|v_1(A_1)| < |v_2(A_2)|$, then since $A_2 \cap E_0 \neq \emptyset$, similarly, $|v_1(A_1)| \geq |v_2(A_2 \setminus \{e_{k_2}^{(2)}\})|$ holds and

A is EQ1. Up to here, the claim is proved.

Next, if $E_0 \subseteq A_1$, then either (i) $|v_1(A_1)| < |v_2(A_2)|$ or (ii) $|v_1(A_1)| > |v_2(A_2)|$ and $|v_1(A_1) \setminus \{e_{k_1}^{(1)}\}| < |v_2(A_2)|$ holds. The latter one indicates that allocation **A** is EQ1, a contradiction. As for the first possible case, recall that terminating at Step 6 implies allocation **A** is not EQ1, and consequently, even when agent 2 only receives E_2 , she still violates the condition of EQ1. Notice that in any UWM, all E_2 must be assigned to agent 2, and so, no UWM satisfies EQ1. If $E_0 \subseteq A_2$, similarly, there are two possible cases: (i) $|v_1(A_1)| > |v_2(A_2)|$; (ii) $|v_1(A_1)| < |v_2(A_2)|$ but $|v_1(A_1)| \geq |v_2(A_2 \setminus \{e_{k_2}^{(2)}\})|$. If case (ii) happens, allocation **A** is EQ1, a contradiction. The only possibility is case (i), which then implies that even when agent 1 only receives bundle E_1 , he still violates the condition of EQ1. Therefore, no UWM is EQ1, completing the proof. \square

We remark that the goods version of Algorithm 4 has been proposed by Aziz et al. [4] to answer E(UW×EQ1) on allocating goods to two agents. For completeness, we have presented the algorithm and proof for both cases of chores and goods.

In the case of two agents, given that E(UW×EQ1) is polynomial-time solvable, the remaining question is whether one can solve C(UW/EQ1) in polynomial time.

Theorem 4.12 *For both goods and chores, the problem C(UW/EQ1) is NP-hard, even for two agents.*

Proof of Theorem 4.12 for chores It suffices to prove NP-completeness of the decision version of C(UW/EQ1): given an instance $\mathcal{I} = \langle [n], E, \mathcal{V} \rangle$ and a threshold value W , does there exist an EQ1 allocation with utilitarian welfare at least W ? We then provide a reduction from the problem of PARTITION.

Given an arbitrary instance of PARTITION, we construct an instance of fair-chores with two agents and a set $E = \{e_1, \dots, e_{m+1}\}$ of $m + 1$ chores. The valuation functions are shown in Table 11, where $0 < \epsilon < 0.1$. The thresholds value is defined as $W = -5T + (4T - 1)\epsilon + \epsilon^2$.

Suppose we have a “yes” instance of PARTITION, and let I_1, I_2 be a solution. Then, consider allocation **A** with $A_1 = \cup_{j \in I_1} e_j$ and $A_2 = E \setminus A_1$. For agents’ value, we have $v_1(A_1) > v_2(A_2)$ and $v_2(A_2 \setminus \{e_{m+1}\}) > v_1(A_1)$, and thus, **A** is an EQ1 allocation with utilitarian welfare

$$UW(\mathbf{A}) = -T - (1 - \epsilon)T - (1 - \epsilon)(3T + \epsilon) = -5T + (4T - 1)\epsilon + \epsilon^2 = W.$$

Therefore, we find an EQ1 allocation **A** with utilitarian welfare W .

We now prove the other direction. Suppose we have a “no” instance of PARTITION. Denote by **B** = (B_1, B_2) an EQ1 allocation and it must be in the form of $\{S_1 \cup \{e_{m+1}\}, S_2\}$ where $\{S_1, S_2\}$ is a partition of set $\{e_1, \dots, e_m\}$. We highlight that S_1 is not necessarily

Table 11 The fair-chores instance for Theorem 4.12

Items	e_1	e_2	...	e_m	e_{m+1}
$v_1(\cdot)$	$-p_1$	$-p_2$...	$-p_m$	$-3T - \epsilon$
$v_2(\cdot)$	$-(1 - \epsilon)p_1$	$-(1 - \epsilon)p_2$...	$-(1 - \epsilon)p_n$	$-(1 - \epsilon)(3T + \epsilon)$

assigned to agent 1. Let $-\sum_{e \in S_1} v_1(e) = \Delta \geq 0$, and accordingly, $-\sum_{e \in S_2} v_1(e) = 2T - \Delta \geq 0$. To meet the condition of EQ1, chore e_{m+1} has to be eliminated when comparing; otherwise, the agent who receives bundle $S_1 \cup \{e_{m+1}\}$ would violate the condition of EQ1, even when $S_1 = \emptyset$. As a consequence, the condition of EQ1 is equivalent to $\Delta \leq T$. Moreover, since it's a "no" instance of PARTITION, we have $\Delta < T$. Given the form of $\{S_1 \cup \{e_{m+1}\}, S_2\}$, there are two possible allocations: assigning bundle $S_1 \cup \{e_{m+1}\}$ to agent 1; assigning bundle $S_1 \cup \{e_{m+1}\}$ to agent 2.

If $B_1 = S_1 \cup \{e_{m+1}\}$ and $B_2 = S_2$, then such an assignment results in welfare

$$\begin{aligned} \text{UW}(\mathbf{B}) &= -\Delta - 3T - \epsilon - (1 - \epsilon)(2T - \Delta) \\ &= -5T + (2T - \Delta - 1)\epsilon < W. \end{aligned}$$

If $B_1 = S_2$ and $B_2 = S_1 \cup \{e_{m+1}\}$, then allocation \mathbf{B} has welfare

$$\begin{aligned} \text{UW}(\mathbf{B}) &= -2T + \Delta - (1 - \epsilon)\Delta - (1 - \epsilon)(3T + \epsilon) \\ &= -5T + (3T + \Delta - 1)\epsilon + \epsilon^2 < W, \end{aligned}$$

where the last transition is due to $\Delta < T$. Therefore, in both cases, no EQ1 allocations can have utilitarian welfare at least W , completing the proof. \square

We have completely settled the computational complexity of algorithmic decision and optimization problems when utilitarian welfare is concerned. Most of results are computationally intractable. In addition, the above algorithmic problems (regarding utilitarian welfare) for goods and chores have the same computational complexity, which somewhat reveals similarities between these two settings.

Next, we investigate egalitarian welfare-maximizing allocations and provide the computational complexity in the case of chores.

Theorem 4.13 *When allocating chores, the problem $E(EW \times EQX)$ is NP-hard, even for two agents.*

Proof To prove NP-hardness, we derive a reduction from the problem PARTITION. Given an arbitrary instance of ARTITION, we construct a fair-chores instance with two agents and a set $E = \{e_1, \dots, e_{m+2}\}$ of $m + 2$ items. The valuation functions are shown in Table 12, where $0 < \epsilon < 0.1$. Since $\sum_{i \in [m]} p_i = 2T$, it is not hard to verify that in an EWM, chores e_{m+1} and e_{m+2} must be assigned to agent 1 and agent 2, respectively, and moreover, the maximum egalitarian welfare is at most $-T - \epsilon$.

Suppose we have a "yes" instance of PARTITION, and let I_1, I_2 be a solution. Then, consider an allocation \mathbf{A} , in which A_1 contains bundle $\bigcup_{j \in I_1} e_j$ and chore e_{m+1} , and $A_2 = E \setminus A_1$. For agents' value, we have $v_1(A_1) = v_2(A_2) = -T - \epsilon$, and so allocation \mathbf{A} is

Table 12 The fair-chores instance for Theorem 4.13

Items	e_1	e_2	\dots	e_m	e_{m+1}	e_{m+2}
$v_1(\cdot)$	$-p_1$	$-p_2$	\dots	$-p_m$	$-\epsilon$	$-2T$
$v_2(\cdot)$	$-p_1$	$-p_2$	\dots	$-p_m$	$-2T$	$-\epsilon$

an EWM that satisfies EQX.

We now prove the other direction. Suppose we have a “no” instance of PARTITION. Let $S = \{e_1, \dots, e_m\}$ be the set of the first m chores, and $\Pi_2(S)$ be the set of 2-partition of S . We then denote by $\Delta = \min_{B \in \Pi_2(S)} |v_1(B_1) - v_2(B_2)|$, and claim that $\Delta \geq 1$ because $p_i \in \mathbb{N}, \forall i \in [m]$ and it’s a “no” instance. Notice that every egalitarian welfare-maximizing allocation \mathbf{O} is in the form of $O_1 = B_1 \cup \{e_{m+1}\}$ and $O_2 = B_2 \cup \{e_{m+2}\}$ where $\{B_1, B_2\} \in \Pi_2(S)$. Without loss of generality we assume $v_1(O_1) < v_2(O_2)$, and based on the definition of Δ and ϵ , we have $v_1(O_1 \setminus \{e_{m+1}\}) - v_2(O_2) \leq \epsilon - \Delta < 0$, based on which, agent 1 violates EQX in allocation \mathbf{O} . Therefore, no EWM is EQX, completing the proof. \square

Theorem 4.14 *When allocating chores, the problem $E(EW \times EQ1)$ is NP-hard, even for three agents.*

Proof To prove NP-hardness, we derive a reduction from PARTITION. Given an arbitrary instance of PARTITION, we construct a fair-chores instance with three agents and a set $E = \{e_1, \dots, e_{m+2}\}$ of $m + 2$ items. The valuation functions are shown in Table 13. Consider an allocation of arbitrarily assigning the first m chores to the first two agents, and assigning chores e_{m+1} and e_{m+2} to agent 3. Since $\sum_{j=1}^m p_j = 2T$, the egalitarian welfare of that allocation is $-2T$. Due to the value of e_{m+1} and e_{m+2} , no allocation can achieve egalitarian welfare larger than $-2T$, which then implies that the maximum egalitarian welfare of the constructed fair-chores instance is equal to $-2T$.

Suppose we have a “yes” instance of PARTITION, and let I_1, I_2 be a solution. Consider an allocation \mathbf{A} with $A_1 = \bigcup_{j \in I_1} e_j, A_2 = \bigcup_{j \in I_2} e_j$ and $A_3 = \{e_{m+1}, e_{m+2}\}$. It is not hard to see that $v_1(A_1) = v_2(A_2) = -T$ and $v_3(A_3) = -2T$, and accordingly, the allocation \mathbf{A} is an EWM. Since $v_3(A_3 \setminus \{e\}) = v_i(A_i)$ holds for any $e \in A_3$ and $i = 1, 2$, the allocation \mathbf{A} is also EQ1.

We now prove the other direction. Suppose we have a “no” instance of PARTITION. Recall that in any EWM, agent 3 must receive exactly items e_{m+1} and e_{m+2} and have value exactly $-2T$. Due to “no” instance, one of the first two agents receives a value strictly larger than $-T$ from the assignment of the first m chores. Through the comparison between agent 3 and the agent receiving the largest value, one can verify that no EWM can satisfy EQ1, completing the proof. \square

Theorem 4.15 *When allocating goods and chores, problems $C(EW/EQ1)$ and $C(EW/EQX)$ are NP-hard, even for two agents.*

Proof It suffices to show NP-completeness of the decision version of problems $C(EW/EQ1)$ and $C(EW/EQX)$ (see the proof of Theorem 4.6). For both goods and chores, we provide a reduction from . GIVEN AN ARBITRARY INSTANCE OF PARTITION, WE CONSTRUCT A

Table 13 The fair-chores instance for Theorem 4.14

Items	e_1	e_2	\dots	e_m	e_{m+1}	e_{m+2}
$v_i(\cdot)$ for $i = 1, 2$	$-p_1$	$-p_2$	\dots	$-p_m$	$-\frac{3m}{2}T$	$-\frac{3m}{2}T$
$v_3(\cdot)$	$-3T$	$-3T$	\dots	$-3T$	$-T$	$-T$

FAIR-GOODS (RESP., FAIR-CHORES) INSTANCE WITH TWO AGENTS AND A SET $E = \{e_1, \dots, e_m\}$ OF M GOODS (RESP., CHORES). AGENTS HAVE IDENTICAL VALUATION FUNCTIONS AND IN THE FAIR-GOODS INSTANCE, $v_i(e_j) = p_j$ FOR $i \in [2]$ AND $j \in [m]$; IN THE FAIR-CHORES INSTANCE, $v_i(e_j) = -p_j$ FOR $i \in [2]$ AND $j \in [m]$. THE THRESHOLD VALUE IS DEFINED AS $W = T$ IN THE FAIR GOODS INSTANCE; $W = -T$ IN THE FAIR-CHORES INSTANCE. THEN, IN BOTH CASES OF GOODS AND CHORES, THERE EXISTS AN EQ1 (OR EQX) ALLOCATION WITH EGALITARIAN WELFARE AT LEAST W IF AND ONLY IF THE PARTITION INSTANCE IS A “YES” INSTANCE. \square

Concerning all complexity results in Section 4, we remark that, except for Theorem 4.12, the established (strong) NP-hardness still holds even when agents’ valuations are normalized to a constant. The results of this section leaves the following two interesting open questions.

Open question 1: For fixed n , what is the time complexity of computing an EQ1 allocation maximizing UW among all EQ1 allocation when agents’ valuations are normalized?

Open question 2: For chores and two agents, what is the time complexity of deciding whether there exists an EWM that is also EQ1?

5 Pseudo-polynomial-time algorithms for fixed number of agents

From the results established in the previous sections, for general n , all decision and computation problems are strongly NP-hard. However, for fixed n it is still unknown whether problems are pseudo-polynomial time solvable. In this section, we design pseudo-polynomial time algorithms that can output the approximately equitable allocation with the maximum welfare for both goods and chores. Similar to the algorithms in Aziz et al. [4], our pseudo-polynomial time algorithms mainly rely on dynamic programming with the sub-problem of assigning the first k items. Once the assignment of k -th item has been settled, we augment k by one and analyse the assignment of $(k + 1)$ -th item upon the allocation of the first k items. Throughout this section, we assume agents’ values are integers and $V = \max_{i \in [n]} \sum_{j \in [m]} |v_i(e_j)|$.

When considering utilitarian welfare, Aziz et al. [4] already provide an algorithm to compute an EQ1/EQX allocation with the maximum utilitarian welfare in the case of goods. For each $i \in [n]$, their algorithm uses t_i and s_i to represent the lower bound of agent i ’s value and a specific item in agent i ’s bundle, respectively. In particular, it applies dynamic programming to compute $G_k(t_1, \dots, t_n, s_1, \dots, s_n)$ for all k , t_i and s_i , and sets it as True if there exists an allocation of e_1, \dots, e_k such that for all i , the value of agent i is at least t_i and item s_i is in agent i ’s bundle; otherwise, False. The desired EQ1/EQX allocation can be found by visiting all $G_m(t_1, \dots, t_n, s_1, \dots, s_n)$. Since for each i , parameter $t_i \in \{0, 1, \dots, V\}$ has $V + 1$ possible values and parameter $s_i \in E$ has m possible values, the worst-case running time of their algorithm is at least $O(m^n V^n)$.

In what follows, we borrow the idea from Aziz et al. [4] and design pseudo-polynomial time algorithms for optimization/decision problems regarding egalitarian welfare. Our algorithms rely on dynamic programmings subroutines, Algorithms 5 and 6, which compute $B(k, \mathbf{t}, \mathbf{s})$ for all possible k , \mathbf{t} and \mathbf{s} . For every $B(k, \mathbf{t}, \mathbf{s})$, the parameter k refers to the number of items; vector $\mathbf{t} = (t_1, \dots, t_n)$ represents agents’ values; vector $\mathbf{s} = (s_1, \dots, s_n)$ represents specific items in bundles of individual agents. To make the case of $k = 0$ well-defined, we assume that every agent is endowed with a dummy item e_0 of zero value (in total there are n

dummy items) and moreover, the dummy item of an agent cannot be reassigned to others. We remark that for all i , s_i takes value from $\{e_0, e_1, \dots, e_m\}$. Note that Algorithms 5 and 6 work for both fair-goods and fair-chores instances. After computing $B(k, \mathbf{t}, \mathbf{s}) \in \{\text{True}, \text{False}\}$ for every k , \mathbf{t} and \mathbf{s} , the desired EQ1 or EQX allocations can be found by visiting all $B(m, \mathbf{t}, \mathbf{s})$ and backtracking the specific one.

Algorithm 5 Dynamic programming for C(EW/EQ1)

Input: An instance $\mathcal{I} = \langle [n], E, \mathcal{V} \rangle$.

Output: Tuple $B(k, \mathbf{t}, \mathbf{s})$ for all k , \mathbf{t} and \mathbf{s} .

```

1: Initialize  $B(k, \mathbf{t}, \mathbf{s}) = \text{False}$  for all  $k$ ,  $\mathbf{t}$  and  $\mathbf{s}$ .
2: Let  $B(0, \mathbf{t}, \mathbf{s}) = \text{True}$  if  $s_i = e_0$  and  $t_i = 0$  for all  $i$ .
3: for  $k = 1, \dots, m$  do
4:   for all  $t_1, \dots, t_n$  and  $s_1, \dots, s_n$  do
5:     for  $i = 1, \dots, n$  do
6:       if  $s_i = e_k$  then
7:         Set  $B(k, t_1, \dots, t_i, \dots, t_n, s_1, \dots, s_i, \dots, s_n) = \text{True}$  if  $B(k-1, t_1, \dots, t_i - v_i(e_k), \dots, t_n, s_1, \dots, s'_i, \dots, s_n) = \text{True}$  for some  $s'_i \in \bigcup_{r=0}^{k-1} e_r$ .
8:       end if
9:       if  $s_i \neq e_k$  then
10:        Set  $B(k, t_1, \dots, t_i, \dots, t_n, s_1, \dots, s_i, \dots, s_n) = \text{True}$  if  $B(k-1, t_1, \dots, t_i - v_i(e_k), \dots, t_n, s_1, \dots, s_i, \dots, s_n) = \text{True}$ 
11:      end if
12:    end for
13:  end for
14: end for

```

The following lemma holds in both cases of goods and chores.

Lemma 5.1 *Given a $B(k, \mathbf{t}, \mathbf{s})$ returned by Algorithm 5, $B(k, \mathbf{t}, \mathbf{s}) = \text{True}$ if and only if there exists an allocation of e_1, \dots, e_k such that for all i , the value of agent i is equal to t_i and $s_i \in \bigcup_{r=0}^{k-1} e_r$ is in agent i 's bundle.*

Consequently, we have the following theorem.

Theorem 5.1 *Given an instance $\mathcal{I} = \langle [n], E, \mathcal{V} \rangle$, one can compute an EQ1 allocation with the maximum egalitarian welfare in time $O(m^{n+2}V^n)$.*

Proof Note that Algorithm 5 can return $B(m, \mathbf{t}, \mathbf{s})$ for all \mathbf{t} and \mathbf{s} . By visiting the entire $B(m, \mathbf{t}, \mathbf{s})$, we can find the set Γ , of which the construction depends on the underlying items. The full construction of Γ is presented below, and the second condition for both goods and chores is to verify the underlying fairness notion.

$$\Gamma = \begin{cases} \{(\mathbf{t}, \mathbf{s}) | B(m, \mathbf{t}, \mathbf{s}) = \text{True} \text{ and } t_i + v_j(s_j) \geq t_j \text{ for all } i, j\}, & \text{for goods;} \\ \{(\mathbf{t}, \mathbf{s}) | B(m, \mathbf{t}, \mathbf{s}) = \text{True} \text{ and } t_i - v_i(s_i) \geq t_j \text{ for all } i, j\}, & \text{for chores.} \end{cases}$$

Given an arbitrary EQ1 allocation $\mathbf{A}' = (A'_1, \dots, A'_n)$, construct $(\mathbf{t}', \mathbf{s}')$ as follows: for all i , $t'_i = v_i(A'_i)$ and $s'_i \in A'_i$ is the item with the largest absolute value for agent i . Then, we have $(\mathbf{t}', \mathbf{s}') \in \Gamma$ due to the property of EQ1 and the construction of \mathbf{t}' and \mathbf{s}' . Accordingly, by visiting all element of Γ , we are able to find the element $(\mathbf{t}^*, \mathbf{s}^*) \in \Gamma$, of which \mathbf{t}^* represents the agents' values in the EQ1 allocation maximizing egalitarian welfare over all EQ1

allocations. In particular, one can pursue the $(\mathbf{t}^*, \mathbf{s}^*)$ with $\min_{i \in [n]} t_i^* \geq \min_{i \in [n]} t_i$ for all $(\mathbf{t}, \mathbf{s}) \in \Gamma$. The specific EQ1 allocation can be found by backtracking $B(m, \mathbf{t}^*, \mathbf{s}^*)$ in the following way: assigning e_m to agent i_m if the value of $B(m, \mathbf{t}^*, \mathbf{s}^*)$ is set to True by $B(m - 1, \mathbf{t}^{m-1}, \mathbf{s}^{m-1}) = \text{True}$ and at that time the for-loop in Step 5 is $i = i_m$; then assigning e_{m-1} to agent i_{m-1} if the value of $B(m - 1, \mathbf{t}^{m-1}, \mathbf{s}^{m-1})$ is set to True by $B(m - 2, \mathbf{t}^{m-2}, \mathbf{s}^{m-2}) = \text{True}$ and at that time the for-loop in Step 5 is $i = i_{m-1}$; repeat this process until all items are assigned. If in some step, the choice of $B(h, \mathbf{t}^h, \mathbf{s}^h)$ is not unique, then arbitrarily pick one.

As for the time complexity, the running time of Algorithm 5 is $O(m^{n+2}V^n)$, and visiting the entire $B(m, \mathbf{t}, \mathbf{s})$ and backtracking takes time $O(m^n V^n)$. Therefore, the running time of the algorithm is $O(m^{n+2}V^n)$. □

Moving on to our consideration of EQX, the dynamic programming for problem C(EW/EQX) is shown as Algorithm 6. Different from Algorithm 5, we now use s_i to represent the item with least non-zero absolute value in agent i 's bundle if agent i receives a non-zero value. We establish that the EQX allocation with the maximum egalitarian welfare can be found by visiting the entire $B(m, \mathbf{t}, \mathbf{s})$ and backtracking specific ones.

Algorithm 6 Dynamic programming for C(EW/EQX)

Input: An instance $\mathcal{I} = \langle [n], E, \mathcal{V} \rangle$.

Output: Tuple $B(k, \mathbf{t}, \mathbf{s})$ for all k, \mathbf{t} and \mathbf{s} .

- 1: Initialize $B(k, \mathbf{t}, \mathbf{s}) = \text{False}$ for all k, \mathbf{t} and \mathbf{s} .
 - 2: Let $B(0, \mathbf{t}, \mathbf{s}) = \text{True}$ if $s_i = e_0$ and $t_i = 0$ for all i .
 - 3: **for** $k = 1, \dots, m$ **do**
 - 4: **for** all t_1, \dots, t_n and s_1, \dots, s_n **do**
 - 5: **for** $i = 1, \dots, n$ **do**
 - 6: **if** $s_i = e_k$ **and** $t_i = v_i(e_k) \neq 0$ **then**
 - 7: Set $B(k, t_1, \dots, t_i, \dots, t_n, s_1, \dots, s_i, \dots, s_n) = \text{True}$ if $B(k-1, t_1, \dots, t_{i-1}, 0, t_{i+1}, \dots, t_n, s_1, \dots, s_{i-1}, e_0, s_{i+1}, \dots, s_n) = \text{True}$.
 - 8: **end if**
 - 9: **if** $s_i = e_k$ **and** $t_i \neq v_i(e_k) \neq 0$ **then**
 - 10: Set $B(k, t_1, \dots, t_i, \dots, t_n, s_1, \dots, s_i, \dots, s_n) = \text{True}$ if $B(k-1, t_1, \dots, t_i - v_i(e_k), \dots, t_n, s_1, \dots, s'_i, \dots, s_n) = \text{True}$ for some $s'_i \in \bigcup_{r \in [k-1]} e_r$ with $|v_i(s_i)| \leq |v_i(s'_i)|$.
 - 11: **end if**
 - 12: **if** $e_0 = s_i \neq e_k$ **and** $t_i = v_i(e_k) = 0$ **then**
 - 13: Set $B(k, t_1, \dots, t_i, \dots, t_n, s_1, \dots, s_i, \dots, s_n) = \text{True}$ if $B(k-1, t_1, \dots, t_{i-1}, 0, t_{i+1}, \dots, t_n, s_1, \dots, s_{i-1}, e_0, s_{i+1}, \dots, s_n) = \text{True}$.
 - 14: **end if**
 - 15: **if** $e_0 \neq s_i \neq e_k$ **and** $t_i \neq v_i(e_k)$ **then**
 - 16: Set $B(k, t_1, \dots, t_i, \dots, t_n, s_1, \dots, s_i, \dots, s_n) = \text{True}$ if $B(k-1, t_1, \dots, t_i - v_i(e_k), \dots, t_n, s_1, \dots, s_i, \dots, s_n) = \text{True}$ and either $0 < |v_i(s_i)| \leq |v_i(e_k)|$ or $v_i(e_k) = 0$.
 - 17: **end if**
 - 18: **end for**
 - 19: **end for**
 - 20: **end for**
-

The following lemma holds in both cases of goods and chores.

Lemma 5.2 *Given a $B(k, \mathbf{t}, \mathbf{s})$ returned by Algorithm 6, $B(k, \mathbf{t}, \mathbf{s}) = \text{True}$ if and only if there exists an allocation such that for all i :*

- (i) the value of agent i is equal to t_i ;
- (ii) if $s_i = e_0$, then $t_i = 0$; if $s_i \neq e_0$, item $s_i \in \bigcup_{r \in [k]} e_r$ is in agent i 's bundle and moreover, either s_i is the unique item of non-zero value in agent i 's bundle or $0 < |v_i(s_i)| \leq |v_i(e)|$ for every item e of non-zero value in agent i 's bundle.

Consequently, we obtain the following theorem.

Theorem 5.2 *Given an instance $\mathcal{I} = \langle [n], E, \mathcal{V} \rangle$, one can compute an EQX allocation with the maximum egalitarian welfare in time $O(m^{n+2}V^n)$.*

Proof Similar proof to that of Theorem 5.1 applies. The proof is deferred to Appendix A.9. \square

Theorems 5.1 and 5.2 indicate that both problems of C(EW/EQ1) and C(EW/EQX) can be solved in pseudo-polynomial time. Accordingly, for fixed n , problems C(EW/EQ1) and C(EW/EQX) are weakly NP-hard. Note that for goods, optimal egalitarian welfare is compatible with EQX/EQ1. To address the decision problems of E(EW \times EQ1) and E(EW \times EQX) for chores, we remark that in Algorithms 5 and 6, tuples $B(m, \mathbf{t}, \mathbf{s})$ actually record all possible values that agents can receive in nearly equitable allocations. Moreover, by eliminating the conditions for parameters $\{s_i\}_{i=1}^n$, it is possible to compute the maximum egalitarian welfare in time $O(mV^n)$ via a dynamic program similar to Algorithm 5. Therefore, in the case of chores, the problems of E(EW \times EQ1) and E(EW \times EQX) are also weakly NP-hard.

Theorem 5.3 *In the allocations of chores, problems E(EW \times EQ1) and E(EW \times EQX) can be answered in time $O(m^{n+2}V^n)$.*

In some resource allocation scenarios, the number of items m to be allocated can be larger than V , and allocating items to agents with dichotomous preferences³ is a typical example of $m \gg V$. We then further explore the possibility of algorithms whose running time is smaller than $O(m^{n+2}V^n)$ in the case of $m \gg V$. The main strategy is to decrease the running time related to input size m , which may inevitably increase the running time caused by V . Below we design pseudo-polynomial time algorithms that can compute the EQ1/EQX allocation with the maximum utilitarian/egalitarian welfare in time $O(mV^{2n+1})$. The algorithms rely on dynamic programming subroutines, Algorithm 7, in which for each $i \in [n]$, we use t_i to represent the value of agent i and p_i to represent the absolute value of a special item in agent i 's bundle. The special item is in particular used to examine whether the underlying fairness notion is satisfied or not. To take an example, when considering EQ1, p_i would represent the absolute value of the item with the largest absolute value in agent i 's bundle. The dynamic programming subroutines therefore admit two n -dimensional vectors $\mathbf{t} = (t_1, \dots, t_n)$ with $t_i \in \{-V, \dots, 0, \dots, V\}$ and $\mathbf{p} = (p_1, \dots, p_n)$ with $p_i \in \{0, \dots, V\}$. For any \mathbf{t} and \mathbf{p} , the dynamic programming examines that for each $k \in [m]$ whether the assignments of the first k items can satisfy the constraints regarding \mathbf{t} and \mathbf{p} , and returns a tuple $B(k, \mathbf{t}, \mathbf{p})$ that takes value from $\{\text{True}, \text{False}\}$. Informally, given fixed k , \mathbf{t} and \mathbf{p} , if there exists an assignment of e_1, \dots, e_k such that for each $i \in [n]$, agent i receives value t_i and the value of the special item in agent i 's bundle is equal to p_i , then the dynamic programming sets $B(k, \mathbf{t}, \mathbf{p}) = \text{True}$; otherwise, $B(k, \mathbf{t}, \mathbf{p}) = \text{False}$.

Algorithm 7 Dynamic programming subroutines

Input: An instance $\mathcal{I} = \langle [n], E, \mathcal{V} \rangle$ and a fairness notion $F \in \{\text{EQ1}, \text{EQX}\}$.

Output: Tuple $B(k, \mathbf{t}, \mathbf{p})$ for all k, \mathbf{t} and \mathbf{p} .

```

1: Initialize  $B(k, \mathbf{t}, \mathbf{p}) = \text{False}$  for all  $k, \mathbf{t}$  and  $\mathbf{p}$ .
2: Let  $B(0, \mathbf{t}, \mathbf{p}) = \text{True}$  if  $t_i = 0$  and  $p_i = 0$  for all  $i$ .
3: for  $k = 1, \dots, m$  do
4:   for all  $t_1, \dots, t_n$  and  $p_1, \dots, p_n$  do
5:     for  $i = 1, \dots, n$  do
6:       if  $F = \text{EQ1}$  then
7:         Set  $B(k, t_1, \dots, t_i, \dots, t_n, p_1, \dots, p_i, \dots, p_n) = \text{True}$  if  $B(k - 1, t_1, \dots, t_i - v_i(e_k), \dots, t_n, p_1, \dots, p'_i, \dots, p_n) = \text{True}$  holds for some  $p'_i$  with  $\max\{|v_i(e_k)|, p'_i\} = p_i$ .
8:       end if
9:       if  $F = \text{EQX}$  then
10:        Set  $B(k, t_1, \dots, t_i, \dots, t_n, p_1, \dots, p_i, \dots, p_n) = \text{True}$  if  $B(k - 1, t_1, \dots, t_i - v_i(e_k), \dots, t_n, p_1, \dots, p'_i, \dots, p_n) = \text{True}$  holds for some  $p'_i$  satisfying the following condition,
                
$$\begin{cases} \min\{|v_i(e_k)|, p'_i\} = p_i, & \text{if } v_i(e_k) \neq 0 \text{ and } p'_i \neq 0; \\ \max\{|v_i(e_k)|, p'_i\} = p_i, & \text{otherwise.} \end{cases}$$

11:        end if
12:      end for
13:    end for
14:  end for

```

Lemma 5.3 Given a $B(k, \mathbf{t}, \mathbf{p})$ returned by Algorithm 7, $B(k, \mathbf{t}, \mathbf{p}) = \text{True}$ if and only if there exists an allocation $\mathbf{A} = (A_1, \dots, A_n)$ of e_1, \dots, e_k such that for all i :

- (i) when considering EQ1, if $v_i(A_i) = 0$ then $p_i = 0$, and otherwise $p_i = \max_{e \in A_i} |v_i(e)|$; when considering EQX, if $v_i(A_i) = 0$ then $p_i = 0$, and otherwise $p_i = \min_{e \in A_i: v_i(e) \neq 0} |v_i(e)|$;
- (ii) $v_i(A_i) = t_i$.

Proof We first prove the “only if” part by mathematical induction. In the case of $k = 0$, Step 2 of Algorithm 7 sets $B(0, \mathbf{t}, \mathbf{p}) = \text{True}$ only if $t_i = p_i = 0$ for all i . Note that when $k = 0$, each agent receives nothing and has value 0. Thus, the statement holds for the case of $k = 0$. We now assume that the statement holds for $k = 0, \dots, h$, and show that it also holds for the case of $k = h + 1$. Fix \mathbf{t} and \mathbf{p} with $B(h + 1, \mathbf{t}, \mathbf{p}) = \text{True}$ and suppose that Algorithm 7 makes $B(h + 1, \mathbf{t}, \mathbf{p}) = \text{True}$ when the for-loop in Step 5 is $i = i^*$. By Steps 7 and 10, it holds that $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, p_1, \dots, p'_{i^*}, \dots, p_n) = \text{True}$ for some p'_{i^*} . As the statement holds for $k = h$, there exists an allocation \mathbf{R} of e_1, \dots, e_h satisfying properties (i) and (ii) regarding $(t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n)$ and $(p_1, \dots, p'_{i^*}, \dots, p_n)$. We now consider the allocation \mathbf{R}^* of e_1, \dots, e_{h+1} with $R_j^* = R_j$ for $j \neq i^*$ and $R_{i^*}^* = R_{i^*} \cup \{e_{h+1}\}$. It is not hard to see that $v_j(R_j^*) = t_j$ for any $j \in [n]$.

³ If agent i has dichotomous preference on the set of goods E , then $v_i(e) \in \{0, 1\}$. For a detailed discussion of dichotomous preferences, we refer the reader to Bogomolnaia and Moulin [13] and Bogomolnaia et al. [17].

As for property (i), since each agent $j \neq i^*$ receives an identical bundle in allocations \mathbf{R} and \mathbf{R}^* and the parameter p_j is consistent in the two tuples under consideration, it suffices to prove that property (i) regarding agent i^* is satisfied by the allocation \mathbf{R}^* . When considering EQ1, we split the proof into two cases:

Case 1 $p'_{i^*} = 0$. As the allocation \mathbf{R} satisfies property (i) regarding $(p_1, \dots, p'_{i^*}, \dots, p_n)$, the equality $v_{i^*}(R_{i^*}) = 0$ holds. Based on Step 7, we have $p_{i^*} = \max\{|v_{i^*}(e_{h+1})|, p'_{i^*}\} = |v_{i^*}(e_{h+1})|$. If $v_{i^*}(e_{h+1}) = 0$, then it holds that $v_{i^*}(R_{i^*}^*) = 0$ and $p_{i^*} = 0$, which implies that the allocation \mathbf{R}^* satisfies property (i). If $|v_{i^*}(e_{h+1})| > 0$, we have $p_{i^*} = \max_{e \in R_{i^*}^*} |v_{i^*}(e)|$, and thus property (i) is also satisfied by \mathbf{R}^* .

Case 2 $p'_{i^*} > 0$. In this case, the following holds,

$$p_{i^*} = \max\{|v_{i^*}(e_{h+1})|, p'_{i^*}\} = \max\left\{|v_{i^*}(e_{h+1})|, \max_{e \in R_{i^*}^*} |v_{i^*}(e)|\right\} = \max_{e \in R_{i^*}^*} |v_{i^*}(e)|,$$

where the first equality is due to Step 7 and the second is owing to the construction of \mathbf{R} . Thus, property (i) regarding \mathbf{p} is also satisfied by \mathbf{R}^* .

Therefore, \mathbf{R}^* is an allocation of e_1, \dots, e_{h+1} that satisfies properties (i) and (ii) regarding \mathbf{t} and \mathbf{p} .

When considering EQX, we also split the proof into two cases.

Case 1: $v_{i^*}(e_{h+1}) = 0$. According to Step 10, we have $p'_{i^*} = p_{i^*}$ since $p'_{i^*} \geq 0$ always holds. If $p'_{i^*} = 0$, then we have $v_{i^*}(R_{i^*}) = 0$, which implies $v_{i^*}(R_{i^*}^*) = 0$ due to $R_{i^*}^* = R_{i^*} \cup \{e_{h+1}\}$ and $v_{i^*}(e_{h+1}) = 0$. Note that $p_{i^*} = p'_{i^*} = 0$, and thus property (i) is satisfied by \mathbf{R}^* . If $p'_{i^*} \neq 0$, then we have the following:

$$p_{i^*} = p'_{i^*} = \min_{e \in R_{i^*}^* : v_{i^*}(e) \neq 0} |v_{i^*}(e)| = \min_{e \in R_{i^*}^* : v_{i^*}(e) \neq 0} |v_{i^*}(e)|,$$

where the second equality is due to the construction of \mathbf{R} and the third is due to $v_{i^*}(e_{h+1}) = 0$. Thus, the property (i) is also satisfied.

Case 2: $v_{i^*}(e_{h+1}) \neq 0$. If $p'_{i^*} = 0$, then according to Step 10, it holds that $p_{i^*} = |v_{i^*}(e_{h+1})|$. Since $p'_{i^*} = 0$ and $R_{i^*}^* = R_{i^*} \cup \{e_{h+1}\}$, item e_{h+1} is the unique non-zero value item in $R_{i^*}^*$ for agent i . Thus, we have $p_{i^*} = |v_{i^*}(e_{h+1})| = \min_{e \in R_{i^*}^* : v_{i^*}(e) \neq 0} |v_{i^*}(e)|$ and property (i) is satisfied. If $p'_{i^*} \neq 0$, then the following holds

$$p_{i^*} = \min\{|v_{i^*}(e_{h+1})|, p'_{i^*}\} = \min\left\{|v_{i^*}(e_{h+1})|, \min_{e \in R_{i^*}^* : v_{i^*}(e) \neq 0} |v_{i^*}(e)|\right\} = \min_{e \in R_{i^*}^* : v_{i^*}(e) \neq 0} |v_{i^*}(e)|,$$

where the first equality is due to Step 10; the second is due to the construction of \mathbf{R} ; and the third is due to $v_{i^*}(e_{h+1}) \neq 0$ and $R_{i^*}^* = R_{i^*} \cup \{e_{h+1}\}$. As a consequence, property (i) is also satisfied. Therefore, when considering EQX, \mathbf{R}^* is an allocation of e_1, \dots, e_{h+1} that satisfies properties (i) and (ii) regarding \mathbf{t} and \mathbf{p} .

Overall, by mathematical induction, the “only if” part is proved.

Now let us prove the “if” part, again with mathematical induction. In the case of $k = 0$, Step 2 of Algorithm 7 sets $B(0, \mathbf{t}, \mathbf{p}) = \text{True}$ if $t_i = p_i = 0$ for all i . Accordingly, if $B(0, \mathbf{t}', \mathbf{p}') = \text{False}$, then either $t'_q \neq 0$ or $p'_q \neq 0$ holds for some q . Note that the value of an agent can only be zero in the case of $k = 0$. Thus, the statement holds for the case of $k = 0$. We now assume the statement holds for $k = 0, \dots, h$ and show that it also holds for the case of $k = h + 1$. Fix \mathbf{t} and \mathbf{p} with $B(h + 1, \mathbf{t}, \mathbf{p}) = \text{False}$. For a contradiction, assume that \mathbf{S} is an allocation of e_1, \dots, e_{h+1} satisfying the properties (i) and (ii) regarding \mathbf{t} and \mathbf{p} . Without loss of generality, we assume $e_{h+1} \in S_{i^*}$. Construct an allocation \mathbf{S}' of e_1, \dots, e_h with $S'_j = S_j$

for $j \neq i^*$ and $S'_i = S_i \setminus \{e_{h+1}\}$. In the following, we show that S' is an allocation that makes $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, p_1, \dots, \tilde{p}_{i^*}, \dots, p_n) = \text{True}$ for some \tilde{p}_{i^*} (will be specified later on) satisfying the condition in Steps 7 and 10, which then results in $B(h + 1, \mathbf{t}, \mathbf{p}) = \text{True}$, a desired contradiction. Note that $v_j(S'_j) = v_j(S_j) = t_j$ for $j \neq i^*$ and $v_{i^*}(S'_i) = t_{i^*} - v_{i^*}(e_{h+1})$, then the property (ii) regarding $(t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n)$ is satisfied by S' .

As for property (i), since each agent $j \neq i^*$ receives an identical bundle in allocations S and S' and the parameter p_j is consistent in the two tuples under consideration, it suffices to prove that property (i) regarding agent i^* is satisfied by the allocation S' .

When considering EQ1, if $v_{i^*}(S'_i) = 0$ then let $\tilde{p}_{i^*} = 0$, and otherwise $\tilde{p}_{i^*} = \max_{e \in S'_i} |v_{i^*}(e)|$. It is not hard to verify that the allocation S' makes $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, p_1, \dots, \tilde{p}_{i^*}, \dots, t_n) = \text{True}$. If $\tilde{p}_{i^*} = 0$, we clearly have $\max\{|v_{i^*}(e_{h+1})|, \tilde{p}_{i^*}\} = p_{i^*}$. If $\tilde{p}_{i^*} > 0$, the following holds,

$$\max\{|v_{i^*}(e_{h+1})|, \tilde{p}_{i^*}\} = \max\left\{|v_{i^*}(e_{h+1})|, \max_{e \in S'_i} |v_{i^*}(e)|\right\} = \max_{e \in S'_i} |v_{i^*}(e)| = p_{i^*},$$

where the second equality is due to $S'_i = S'_i \cup \{e_{h+1}\}$. Thus, Step 7 always sets $B(h + 1, \mathbf{t}, \mathbf{p}) = \text{True}$ when the for-loop in Step 5 is $i = i^*$, a contradiction.

When considering EQX, we define \tilde{p}_{i^*} as follows,

$$\tilde{p}_{i^*} = \begin{cases} 0, & \text{if } v_{i^*}(S'_i) = 0; \\ \min_{e \in S'_i: v_{i^*}(e) \neq 0} |v_{i^*}(e)|, & \text{otherwise.} \end{cases}$$

Then one can verify that the allocation S' makes $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, p_1, \dots, \tilde{p}_{i^*}, \dots, p_n) = \text{True}$ because the statement holds for $k = h$. In the following, we split the proof into two cases and for each case prove that the condition in Step 10 is satisfied.

Case 1: $v_{i^*}(e_{h+1}) = 0$. Since $S'_i = S'_i \cup \{e_{h+1}\}$ and $v_{i^*}(e_{h+1}) = 0$, it holds that $p_{i^*} = \max\{|v_{i^*}(e_{h+1})|, \tilde{p}_{i^*}\}$ no matter whether $\tilde{p}_{i^*} = 0$ or not.

Case 2: $v_{i^*}(e_{h+1}) \neq 0$. If $\tilde{p}_{i^*} = 0$, then e_{h+1} is the unique item in S'_i having non-zero value for agent i^* . Accordingly, we have $p_{i^*} = |v_{i^*}(e_{h+1})| = \max\{|v_{i^*}(e_{h+1})|, \tilde{p}_{i^*}\}$, and thus the condition of Step 10 is satisfied. If $\tilde{p}_{i^*} \neq 0$, then the following holds,

$$p_{i^*} = \min_{e \in S'_i: v_{i^*}(e) \neq 0} |v_{i^*}(e)| = \min\left\{|v_{i^*}(e_{h+1})|, \min_{e \in S'_i: v_{i^*}(e) \neq 0} |v_{i^*}(e)|\right\} = \min\{|v_{i^*}(e_{h+1})|, \tilde{p}_{i^*}\},$$

where the second equality is due to $S'_i = S'_i \cup \{e_{h+1}\}$; the third equality comes from the definition of \tilde{p}_{i^*} . Thus, in both cases, Step 10 sets $B(h + 1, \mathbf{t}, \mathbf{p}) = \text{True}$ when the for-loop in Step 5 is $i = i^*$, a contradiction,

Therefore, with mathematical induction, we have also proved the ‘‘if’’ part of the lemma. □

Theorem 5.4 *Given an instance $\mathcal{I} = \langle [n], E, \mathcal{V} \rangle$, one can compute an $F \in \{\text{EQ1}, \text{EQX}\}$ allocation with the maximum $W \in \{\text{EW}, \text{UW}\}$ welfare in time $O(mV^{2n+1})$.*

Proof Note that Algorithm 7 can return $B(m, \mathbf{t}, \mathbf{p})$ for all \mathbf{t} and \mathbf{p} . By visiting the entire $B(m, \mathbf{t}, \mathbf{p})$, we can find the set Γ , of which the construction depends on the underlying items.

The full construction of Γ is presented below, and the second condition for both goods and chores is to examine whether the underlying fairness notion is satisfied or not.

$$\Gamma = \begin{cases} \{(\mathbf{t}, \mathbf{p}) \mid B(m, \mathbf{t}, \mathbf{p}) = \text{True and } t_i + p_j \geq t_j \text{ for all } i, j\}, & \text{for goods;} \\ \{(\mathbf{t}, \mathbf{p}) \mid B(m, \mathbf{t}, \mathbf{p}) = \text{True and } t_i + p_i \geq t_j \text{ for all } i, j\}, & \text{for chores.} \end{cases}$$

For an arbitrary F allocation $\mathbf{A}' = (A'_1, \dots, A'_n)$, consider two vectors \mathbf{t}' and \mathbf{p}' with for all i , $t'_i = v_i(A'_i)$, and for p'_i , we distinguish between EQ1 and EQX: for EQ1, if $v_i(A'_i) = 0$ then let $p'_i = 0$, and otherwise let $p'_i = \max_{e \in A'_i} |v_i(e)|$; for EQX, if $v_i(A'_i) = 0$, then let $p'_i = 0$ and otherwise let $p'_i = \min_{e \in A'_i: v_i(e) \neq 0} |v_i(e)|$.

We first claim that $(\mathbf{t}', \mathbf{p}') \in \Gamma$. Based on Lemma 5.3, the allocation \mathbf{A}' makes $B(m, \mathbf{t}', \mathbf{p}') = \text{True}$. Moreover, due to the construction of \mathbf{t}' and \mathbf{p}' and the fact that \mathbf{A}' is an F allocation, the second condition is also satisfied. As a consequence, it holds that $(\mathbf{t}', \mathbf{p}') \in \Gamma$. Then, by visiting all elements of Γ , we are able to find the element $(\mathbf{t}^*, \mathbf{p}^*) \in \Gamma$, of which \mathbf{t}^* represents the agents' values in the F allocation maximizing W over all F allocations. In particular, when $W = UW$ (resp., $W = EW$), one can pursue the $(\mathbf{t}^*, \mathbf{p}^*)$ with $\sum_{i \in [n]} t_i^* \geq \sum_{i \in [n]} t_i$ (resp., $\min_{i \in [n]} t_i^* \geq \min_{i \in [n]} t_i$) for all $(\mathbf{t}, \mathbf{p}) \in \Gamma$. The specific F allocation that maximizes W over all F allocations can be constructed by backtracking $B(m, \mathbf{t}^*, \mathbf{p}^*)$ in the following way: assigning e_m to agent i_m if the value of $B(m, \mathbf{t}^*, \mathbf{p}^*)$ is set to True by $B(m - 1, \mathbf{t}^{m-1}, \mathbf{p}^{m-1}) = \text{True}$ and at that time the for-loop in Step 5 is $i = i_m$; then assigning e_{m-1} to agent i_{m-1} if the value of $B(m - 1, \mathbf{t}^{m-1}, \mathbf{p}^{m-1})$ is set to True by $B(m - 2, \mathbf{t}^{m-2}, \mathbf{p}^{m-2}) = \text{True}$ and at that time the for-loop in Step 5 is $i = i_{m-1}$; repeat this process until all items are assigned. If in some step, the choice of $B(h, \mathbf{t}^h, \mathbf{p}^h)$ is not unique, then arbitrarily pick one.

As for the time complexity, the running time of Algorithm 7 is $O(mV^{2n+1})$, and visiting all $B(m, \mathbf{t}, \mathbf{p})$ and backtracking the specific one takes time $O(mV^{2n})$. Therefore, the running time of the algorithm is $O(mV^{2n+1})$. □

For the decision problems, note that the maximum utilitarian welfare can be computed in linear time, and accordingly, problems $E(UW \times EQ1)$ and $E(UW \times EQX)$ can be answered in time $O(mV^{2n+1})$.

Theorem 5.5 *When allocating goods and chores, problems $E(UW \times EQ1)$ and $E(UW \times EQX)$ can be answered in time $O(mV^{2n+1})$.*

When considering egalitarian welfare, recall that for goods, decision problems $E(EW \times EQ1)$ and $E(EW \times EQX)$ trivially have answer “yes”. For the allocation of chores, the maximum egalitarian welfare can be computed in time $O(mV^n)$ via a dynamic program similar to Algorithm 7 (eliminating \mathbf{p} and the corresponding conditions). Consequently, for chores, the problems of $E(EW \times EQ1)$ and $E(EW \times EQX)$ can also be answered in time $O(mV^{2n+1})$.

Theorem 5.6 *When allocating chores, problems $E(EW \times EQ1)$ and $E(EW \times EQX)$ can be answered in time $O(mV^{2n+1})$.*

We remark that, when considering egalitarian welfare, we have presented two pseudo-polynomial algorithms, one using Algorithms 5 and 6 as subroutines, and the other using Algorithm 7 as subroutines. The two types of algorithms are mutually non-dominating in terms of running time. Specific resource allocation problems determine which one is more efficient.

6 Concluding remarks

In this paper, we have conducted an analysis on both indivisible goods and indivisible chores and studied two notions of relaxed equitability, namely EQX and EQ1, together with efficiency measured by two welfare allocations, utilitarian and egalitarian welfare. On every pairwise fairness and welfare combination, we have provided (almost) tight results on the price of fairness. In the case of chores, to achieve relaxed equitability, almost all efficiency would be sacrificed, while the prices of fairness in goods allocation are all bounded. Particularly, with two agents, fairness can be achieved at the cost of at most half of welfare in goods allocations; however, in the case of chores, fair allocations cannot have a bounded guarantee on welfare. Our results on the price of fairness somewhat reflect the differences between goods and chores.

From the results on the price of fairness, relaxed equitability is not always compatible with optimal social welfare, which motivates us to investigate whether one can efficiently determine the existence of a fair and welfare-maximizing allocation and compute the one with maximum welfare among fair allocations. We have depicted a complete picture of the computational complexity of all decision and optimization problems. In particular, when utilitarian welfare is concerned, all decision and optimization problems are strongly NP-complete or strongly NP-hard for general n agents. For the case of fixed n , except for E (UW×EQ1) with two agents, other problems are still intractable in polynomial time. On the positive side, we are able to propose a pseudo-polynomial time algorithm that output the fair allocation with the maximum utilitarian welfare. For problem E(UW×EQ1) with two agents, a polynomial time algorithm exists. When focusing on egalitarian welfare, EQX and EQ1 are compatible with optimal egalitarian welfare in goods allocations. On the contrary, in the case of chores, deciding the existence of EQX (resp., EQ1) and egalitarian welfare-maximizing allocation is strongly NP-hard for general n and weakly NP-complete for fixed $n \geq 2$ (resp., $n \geq 3$). Both goods and chores versions for optimization problems are strongly NP-hard for general n and weakly NP-hard for fixed n . Our results indicate that although goods and chores yield different results in the problem where the numerical value of input matters, for example, price of fairness, they may have similar features in terms of computational complexity.

Looking forward, areas that allure immediate exploration include: (a) bound tightness on the price of EQX in the case of general n and the price of EQ1 with respect to utilitarian welfare; (b) whether Theorem 4.12 still holds when agents' valuations are normalized to a constant; (c) for the allocations of chores with two agents, what is the time complexity of deciding the existence of an EQ1 allocation that also achieves the maximum egalitarian welfare? Besides, given unboundedness of the PoF in our consideration of fair and efficient allocation of chores, it is desirable to improve our current lens of the PoF to see a refined picture about the efficiency loss of a fair allocation of chores. One possible way could be to add a positive parameter intrinsic to the problem instance to both the numerator and denominator of our current definition of the PoF ratio.

Appendix

A.1 Proof of Theorem 4.1 for goods

The problem is in NP as both utilitarian welfare maximization and EQX can be tested in polynomial time. We then derive a reduction from 3-PARTITION.

Given an arbitrary instance of 3-PARTITION, we construct a fair-goods instance as follows. There are $m + 1$ agents and a set $E = \{e_1, \dots, e_{3m+2}\}$ of $3m + 2$ goods. The valuation functions are shown in Table 14. It is not hard to verify that an allocation is a UWM if and only if the first $3m$ goods are assigned to the first m agents and the last two goods are allocated to agent $m + 1$.

Suppose we have a “yes” instance of 3-PARTITION, and let I_1, \dots, I_m be a solution. Then, consider allocation \mathbf{A} with $A_i = \bigcup_{j \in I_i} e_j$ for any $i \in [m]$ and $A_{m+1} = \{e_{3m+1}, e_{3m+2}\}$. Clearly, allocation \mathbf{A} is a UWM. For agents’ value, we have $v_i(A_i) = T$ for every $i \in [m]$ and $v_{m+1}(A_{m+1}) = 2T$. Notice that $v_i(A_i) = v_{m+1}(A_{m+1} \setminus \{e\})$ holds for any $i \in [m]$ and $e \in A_{m+1}$, and consequently, \mathbf{A} is a UWM that satisfies EQX.

Now we prove the other direction. Suppose we have a “no” instance of 3-PARTITION. Recall that in any UWM, agent $m + 1$ only receives the last two goods and has value $2T$. Because it is a “no” instance, assigning the first $3m$ goods to the first m agents always results in an allocation in which at least one agent receives value strictly smaller than T . In any UWM, when comparing to agent $m + 1$, the agent receiving the least value violates the condition of EQX, completing the proof.

A.2 Proof of Theorem 4.2 for goods

This decision problem is in NP as both utilitarian welfare maximization and EQ1 can be examined in polynomial time. The polynomial time reduction in the proof of Theorem 4.1 for goods can be carried over to problem E(UW×EQ1).

A.3 Proof of Theorem 4.7 for goods

To prove NP-hardness, we derive a reduction from the problem PARTITION. Given an arbitrary instance of PARTITION, we construct a fair-goods instance with two agents and a set $E = \{e_1, \dots, e_{m+2}\}$ of $m + 2$ goods. The valuation functions are shown in Table 15, where $0 < \epsilon < 1$. Given these valuation functions, an allocation is a UWM if and only if e_{m+1} and e_{m+2} are assigned to agent 2 and agent 1, respectively.

Suppose we have a “yes” instance of PARTITION, and let I_1, I_2 be a solution. Then, consider allocation \mathbf{A} in which A_1 contains bundle $\bigcup_{j \in I_1} e_j$ and good e_{m+2} , and $A_2 = E \setminus A_1$. Clearly, allocation \mathbf{A} is a UWM. Then, since $v_1(A_1) = v_2(A_2) = T + \epsilon$, allocation \mathbf{A} is also EQX.

We now prove the other direction. Suppose we have a “no” instance of PARTITION. Then, let $S = \{e_1, \dots, e_m\}$ be the set of first m goods and $\Pi_2(S)$ be the set of 2-partition of S . We denote by $\Delta = \arg \min_{B \in \Pi_2(S)} |v_1(B_1) - v_2(B_2)|$, and we claim that $\Delta \geq 1$ because $p_i \in \mathbb{N}, \forall i \in [m]$ and it’s a “no” PARTITION instance. Let \mathbf{O} be an arbitrary UWM, and without loss of generality assume $v_1(O_1) > v_2(O_2)$. According to the construction of Δ , it must hold that $0 < \Delta - \epsilon \leq v_1(O_1 \setminus \{e_{m+2}\}) - v_2(O_2)$ implying that agent 2 violates the condition of EQX. Therefore, no UWM can also be EQX, completing the proof.

A.4 Proof of Theorem 4.9 for goods

The problem is in NP as both utilitarian welfare maximization and EQ1 can be tested in polynomial time. To prove NP-hardness, we derive a reduction from the problem of PARTITION. Given an arbitrary instance of PARTITION, we construct a fair-goods instance for three agents with a set $E = \{e_1, \dots, e_{m+2}\}$ of $m + 2$ goods. The valuation

Table 14 The fair-goods instance for Theorem 4.1

Items	e_1	e_2	\dots	e_m	e_{m+1}	e_{m+2}
$v_i(\cdot)$ for $i \leq m$	b_1	b_2	\dots	b_{3m}	0	0
v_{m+1}	$\frac{(m-2)b_1}{m}$	$\frac{(m-2)b_2}{m}$	\dots	$\frac{(m-2)b_{3m}}{m}$	T	T

functions are shown in Table 16. An allocation is a UWM if and only if the first m goods are assigned to the first two agents and the last two goods are assigned to agent 3.

Suppose we have a “yes” instance of PARTITION, and let I_1, I_2 be a solution. Then, consider allocation \mathbf{A} with $A_1 = \bigcup_{j \in I_1} e_j$, $A_2 = \bigcup_{j \in I_2} e_j$ and $A_3 = \{e_{m+1}, e_{m+2}\}$. It is straightforward to verify that allocation \mathbf{A} is a UWM. As for agents’ value, we have $v_1(A_1) = v_2(A_2) = T$ and $v_3(A_3) = 2T$, and moreover, $v_3(A_3 \setminus \{e\}) = T$ holds for any $e \in A_3$. Thus, allocation \mathbf{A} is also EQ1.

We now prove the other direction. Suppose we have a “no” instance of PARTITION. Recall that in any UWM, agent 3 receives value $2T$. Since it is a “no” PARTITION instance, assigning the first m goods to the first two agents always results in an allocation in which one agent receives value strictly smaller than T . In an UWM, when comparing to agent 3, the agent receiving the least value violates the condition of EQ1. Therefore, no UWM is EQ1, completing the proof.

A.5 Proof of Theorem 4.11 for goods

The proof uses Algorithm 4. It is not hard to verify that assigning E_i to agent i for all i is the necessary and sufficient condition for guaranteeing the output being an UWM.

Denote by $\mathbf{A} = (A_1, A_2)$ the with $A_1 = \{e_1^{(1)}, \dots, e_{k_1}^{(1)}\}$ and $A_2 = \{e_1^{(2)}, \dots, e_{k_2}^{(2)}\}$.

Clearly, we have $k_1 + k_2 = m$. Term $e_j^{(i)}$ refers to the j -th goods assigned by algorithm to agent i . If Algorithm 4 terminates at Step 4, clearly we find a UWM that satisfies EQ1. We now consider the case where Algorithm 4 terminates at Step 6, and claim in this case, either $E_0 \subseteq A_1$ or $E_0 \subseteq A_2$. It suffices to show that if both $A_1 \cap E_0$ and $A_2 \cap E_0$ are not empty set, then Algorithm 4 terminates at Step 4. To prove this claim, we discuss two cases. If $v_1(A_1) > v_2(A_2)$, since $A_1 \cap E_0 \neq \emptyset$, we have $e_{k_1}^{(1)} \in E_0$. Then by algorithm, it must hold that $v_1(A_1 \setminus \{e_{k_1}^{(1)}\}) \leq v_2(A_2)$; otherwise, agent 1 would not receive good $e_{k_1}^{(1)}$. Accordingly, allocation \mathbf{A} is EQ1, and Algorithm 4 terminate at Step 4. On the other hand, if $v_1(A_1) < v_2(A_2)$, then since $A_2 \cap E_0 \neq \emptyset$, similarly, $v_1(A_1) \geq v_2(A_2 \setminus \{e_{k_2}^{(2)}\})$ holds and allocation \mathbf{A} is EQ1. The claim is proved.

Table 15 The fair-goods instance for Theorem 4.7

Items	e_1	e_2	\dots	e_m	e_{m+1}	e_{m+2}
$v_1(\cdot)$	p_1	p_2	\dots	p_m	0	ϵ
$v_2(\cdot)$	p_1	p_2	\dots	p_m	ϵ	0

Next, if $E_0 \subseteq A_1$, then either (i) $v_1(A_1) < v_2(A_2)$ or (ii) $v_1(A_1) > v_2(A_2)$ and $v_1(A_1 \setminus \{e_{k_1}^{(1)}\}) \leq v_2(A_2)$. The latter one indicates that allocation \mathbf{A} is EQ1, a contradiction. As for the first possible case, recall that terminating at Step 6 means that allocation \mathbf{A} is not EQ1, and as a consequence, even when agent 1 receives all E_0 , he still violates the condition of EQ1. Notice that in any UWM, agent 1 receives value at most $v_1(E_0 \cup E_1) = v_1(A_1)$, and so, no UWM is EQ1. If $E_0 \subseteq A_2$, similarly, there are two possible cases: (i) $v_1(A_1) > v_2(A_2)$; (ii) $v_1(A_1) < v_2(A_2)$ and $v_1(A_1) \geq v_2(A_2 \setminus \{e_{k_2}^{(2)}\})$. If case (ii) happens, then allocation \mathbf{A} is EQ1, a contradiction. The only possibility is case (i), which then implies that even when agent 2 receives both E_0 and E_2 , she still violates that condition of EQ1. Thus, no UWM is EQ1, completing the proof.

A.6 Proof of Theorem 4.12 for goods

It suffices to prove that the decision version of C(UW/EQ1) is NP-complete. We prove it by deriving a reduction from the problem of PARTITION.

Given an arbitrary instance of PARTITION, we construct an instance of the decision version of C(UW/EQ1) with a set $E = \{e_1, \dots, e_{m+3}\}$ goods. The valuation functions are shown in Table 17, where $0 < \epsilon < 1/T$. The threshold value is defined as $W = 5T$.

We denote by $S = \{e_1, \dots, e_m\}$. Suppose we have a “yes” instance of PARTITION, and let I_1, I_2 be a solution. Then, consider an allocation \mathbf{A} , in which A_1 contains bundles $\bigcup_{j \in I_1} e_j$ and good e_{m+1} and $A_2 = E \setminus A_1$. It holds that $v_1(A_1) = 2T = v_2(A_2 \setminus \{e_{m+3}\})$, which implies that allocation \mathbf{A} is EQ1. Also, we have $UW(\mathbf{A}) = 5T$ implying that \mathbf{A} is an EQ1 allocation with utilitarian welfare $5W$.

We now prove the other direction. Suppose we have a “no” instance of PARTITION. For an arbitrary EQ1 allocation \mathbf{B} , its utilitarian welfare is no less than $5T$ only if e_{m+1} is assigned to agent 1 and e_{m+2}, e_{m+3} are assigned to agent 2. Moreover, it is not hard to verify that in order to gain utilitarian welfare $5T$, agent 2 has to receive value at least $(1 + \epsilon)T$ from bundle S . Then we discuss two cases according to the largest p_i . Let $i^* \in \arg \max_{i \in [m]} p_i$.

If $p_{i^*} > T$, then due to $p_{i^*} \in \mathbb{N}^+$ and it is an “no” instance, we have $p_{i^*} - T \geq 1$. Then, good e_{i^*} must be assigned to agent 2 so that the utilitarian welfare is no less than $5T$. Then, in such an allocation \mathbf{B} , agent 1 has value $v_1(B_1) \leq 3T - p_{i^*} \leq 2T - 1 < 2T - \epsilon T \leq \min_{e \in B_2} v_2(B_2 \setminus \{e\})$. Thus, \mathbf{B} is not EQ1, which implies that the utilitarian welfare of any EQ1 allocation is smaller than $5T$.

If $p_{i^*} < T$ ($p_{i^*} \neq T$ because of “no” instance), suppose S_1, S_2 be a 2-PARTITION of S such that allocation \mathbf{B} is composed of $B_1 = S_1 \cup \{e_{m+1}\}$ and $B_2 = S_2 \cup \{e_{m+2}, e_{m+3}\}$ and has utilitarian welfare no less than $5T$. Recall that to meet the requirement on welfare, $v_2(S_2) \geq (1 + \epsilon)T$ must hold, and moreover, owing to “no” PARTITION instance, it holds that $v_2(S_2) > (1 + \epsilon)T$. Accordingly, we have $v_1(S_1) < T$, derived by $\sum_{i \in [m]} p_i = 2T$. As a consequence, $v_1(B_1) < 2T \leq \min_{e \in B_2} v_2(B_2 \setminus \{e\})$ where the last transition is because, in this

Table 16 The fair-goods instance for Theorem 4.9

Items	e_1	e_2	\dots	e_m	e_{m+1}	e_{m+2}
$v_i(\cdot)$ for $i = 1, 2$	p_1	p_2	\dots	p_m	0	0
$v_3(\cdot)$	0	0	\dots	0	T	T

case, e_{m+3} is the most valuable item in bundle B_2 for agent 2 given $p_{i^*} < T$. This inequality contradicts the fact that allocation \mathbf{B} is EQ1, completing the proof.

A.7 Proof of Lemma 5.1

We first prove the “only if” part by mathematical induction. In the case of $k = 0$, Step 2 of Algorithm 5 sets $B(0, \mathbf{t}, \mathbf{s}) = \text{True}$ if $s_i = e_0$ and $t_i = 0$ for all i . Note that when $k = 0$, each agent i receives only e_0 with value equal to 0. Thus, the statement holds for the case of $k = 0$. We now assume that the statement holds for $k = 0, \dots, h$ and show that it also holds for the case of $k = h + 1$. Fix \mathbf{t} and \mathbf{s} with $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$ and suppose that Algorithm 5 makes $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$ when the for-loop in Step 5 being $i = i^*$.

If $s_{i^*} = e_{h+1}$, then Step 7 sets $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$ and thus $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, s_1, \dots, s'_{i^*}, \dots, s_n) = \text{True}$ for some $s'_{i^*} \in \bigcup_{r=0}^h e_r$. As the statement holds for $k = h$, there exists an allocation \mathbf{P} of e_1, \dots, e_h such that (i) $v_j(P_j) = t_j$ for $j \neq i^*$ and $v_{i^*}(P_{i^*}) = t_{i^*} - v_{i^*}(e_{h+1})$; (ii) $s_j \in \bigcup_{r=0}^h e_r$ and $s_j \in P_j$ for all $j \neq i^*$; $s'_{i^*} \in \bigcup_{r=0}^h e_r$ and $s'_{i^*} \in P_{i^*}$. Then consider allocation \mathbf{P}^* with $P_j^* = P_j$ for $j \neq i^*$ and $P_{i^*}^* = P_{i^*} \cup \{e_{h+1}\}$. One can verify that \mathbf{P}^* is an allocation of e_1, \dots, e_{h+1} with $v_i(P_i^*) = t_i$ and $s_i \in P_i^*$ for all i .

If $s_{i^*} \neq e_{h+1}$, then Step 10 sets $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$ and thus $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, s_1, \dots, s_{i^*}, \dots, s_n) = \text{True}$. As the statement holds for $k = h$, there exists an allocation \mathbf{Q} of e_1, \dots, e_h such that (i) $v_j(Q_j) = t_j$ for $j \neq i^*$ and $v_{i^*}(Q_{i^*}) = t_{i^*} - v_{i^*}(e_{h+1})$; (ii) $s_j \in \bigcup_{r=0}^h e_r$ and $s_j \in Q_j$ for all j . Then consider allocation \mathbf{Q}^* with $Q_j^* = Q_j$ for $j \neq i^*$ and $Q_{i^*}^* = Q_{i^*} \cup \{e_{h+1}\}$. One can verify that \mathbf{Q}^* is an allocation of e_1, \dots, e_{h+1} with $v_i(Q_i^*) = t_i$ and $s_i \in Q_i^*$ for all i . Up to here, the statement also holds when $k = h + 1$.

Overall, by mathematical induction, the “only if” part is proved.

Now let us prove the “if” part, again with mathematical induction. In the case of $k = 0$, Step 2 of Algorithm 5 sets $B(0, \mathbf{t}, \mathbf{s}) = \text{True}$ if $s_i = e_0$ and $t_i = 0$ for all i . Accordingly, if $B(0, \mathbf{t}', \mathbf{s}') = \text{False}$, then either $t'_q \neq 0$ or $s'_q \neq e_0$ holds for some q . Note that in the case of $k = 0$, no allocation can make agent q receive non-dummy item s'_q or a non-zero value. Thus, the statement holds for the case of $k = 0$. We now assume that the statement holds for $k = 0, \dots, h$ and show that it also holds for the case of $k = h + 1$. Fix \mathbf{t} and \mathbf{s} with $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{False}$. For a contradiction, assume that \mathbf{P} is an allocation of e_1, \dots, e_{h+1} with $v_j(P_j) = t_j$ and $s_j \in P_j$ for all j . Without loss of generality, we assume $e_{h+1} \in P_{i^*}$. Construct an allocation \mathbf{P}' of e_1, \dots, e_h with $P'_j = P_j$ for $j \neq i^*$ and $P'_{i^*} = P_{i^*} \setminus \{e_{h+1}\}$.

If $s_{i^*} = e_{h+1}$, due to the construction of \mathbf{P} and \mathbf{P}' and the fact that the statement holds for $k = h$, one can verify that allocation \mathbf{P}' makes $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, s_1, \dots, s'_{i^*}, \dots, s_n) = \text{True}$, where $s'_{i^*} \in \bigcup_{r=0}^h e_r$. Accordingly, when the for-loop in Step 5 is $i = i^*$, Step 7 will set $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$, a contradiction.

If $s_{i^*} \neq e_{h+1}$, since $s_{i^*} \in P_{i^*}$ and $s_{i^*} \neq e_{h+1}$, we have $s_{i^*} \in P'_{i^*}$. Due to the construction of \mathbf{P} and \mathbf{P}' and the fact that the statement holds for $k = h$, one can verify that allocation \mathbf{P}'

Table 17 The fair-goods instance for Theorem 4.12

Items	e_1	e_2	\dots	e_m	e_{m+1}	e_{m+2}	e_{m+3}
$v_1(\cdot)$	p_1	p_2	\dots	p_m	T	0	0
$v_2(\cdot)$	$(1 + \epsilon)p_1$	$(1 + \epsilon)p_2$	\dots	$(1 + \epsilon)p_m$	0	$(1 - \epsilon)T$	T

makes $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, s_1, \dots, s_{i^*}, \dots, s_n) = \text{True}$. As a result, when the for-loop in Step 5 is $i = i^*$, Step 10 will set $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$, another contradiction. Therefore, the statement also holds when $k = h + 1$.

Therefore, with mathematical induction, we have also proved the “if” part of the lemma.

A.8 Proof of Lemma 5.2

We first prove the “only if” part with mathematical induction. In the case of $k = 0$, Step 2 of Algorithm 6 sets $B(0, \mathbf{t}, \mathbf{s}) = \text{True}$ if $s_i = e_0$ and $t_i = 0$ for all i . Note that when $k = 0$, each agent i receives only e_0 with value equal to 0. Thus, the statement holds for the case of $k = 0$. We now assume that the statement holds for $k = 0, \dots, h$ and show that it also holds for the case of $k = h + 1$. Fix \mathbf{t} and \mathbf{s} with $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$ and suppose that Algorithm 6 makes $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$ when for-loop in Step 5 is $i = i^*$.

If $s_{i^*} = e_{h+1}$ and $t_{i^*} = v_{i^*}(e_{h+1}) \neq 0$, then Step 7 sets $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$ and $B(h, t_1, \dots, t_{i^*-1}, 0, t_{i^*+1}, \dots, t_n, s_1, \dots, s_{i^*-1}, e_0, s_{i^*+1}, \dots, s_n) = \text{True}$. As the statement holds for $k = h$, there exists an allocation \mathbf{P} of e_1, \dots, e_h such that $v_j(P_j) = t_j$ for $j \neq i^*$ and $v_{i^*}(P_{i^*}) = 0$, and moreover, satisfies property (ii) described in the statement. We now consider \mathbf{P}' with $P'_j = P_j$ for $j \neq i^*$ and $P'_{i^*} = P_{i^*} \cup \{e_{h+1}\}$. Since $v_{i^*}(P'_{i^*}) = v_{i^*}(P_{i^*}) + v_{i^*}(e_{h+1}) = t_{i^*}$ and $e_{h+1} \in P'_{i^*}$ is the unique item of non-zero value in P'_{i^*} , \mathbf{P}' is an allocation of e_1, \dots, e_{h+1} that satisfies the properties (i) and (ii) described in the statement regarding \mathbf{t} and \mathbf{s} .

If $s_{i^*} = e_{h+1}$ and $t_{i^*} \neq v_{i^*}(e_{h+1}) \neq 0$, then Step 10 sets $B(k, \mathbf{t}, \mathbf{s}) = \text{True}$ and $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, s_1, \dots, s'_{i^*}, \dots, s_n) = \text{True}$, where $s'_{i^*} \in \bigcup_{r \in [h]} e_r$ satisfies $|v_{i^*}(s'_{i^*})| \geq |v_{i^*}(e_{h+1})|$. As the statement holds for $k = h$, there exists an allocation \mathbf{Q} of e_1, \dots, e_h such that $v_j(Q_j) = t_j$ for $j \neq i^*$ and $v_{i^*}(Q_{i^*}) = t_{i^*} - v_{i^*}(e_{h+1})$, and moreover, satisfies property (ii) described in the statement regarding $s_1, \dots, s'_{i^*}, \dots, s_n$. We now consider \mathbf{Q}' with $Q'_j = Q_j$ for $j \neq i^*$ and $Q'_{i^*} = Q_{i^*} \cup \{e_{h+1}\}$. Since $v_{i^*}(Q'_{i^*}) = v_{i^*}(Q_{i^*}) + v_{i^*}(e_{h+1}) = t_{i^*}$ and $Q'_j = Q_j$ for $j \neq i^*$, then we have $v_j(Q'_j) = t_j$ for all j . As for property (ii), note that $|v_{i^*}(s'_{i^*})| \geq |v_{i^*}(e_{h+1})| \neq 0$ and s'_{i^*} is the item with the least non-zero absolute value for agent i^* in Q_{i^*} , thus e_{h+1} is the item with the least non-zero absolute value in Q'_{i^*} . Then, property (ii) regarding \mathbf{s} described in the statement is also satisfied by \mathbf{Q}' .

If $e_0 = s_{i^*} \neq e_{h+1}$ and $t_{i^*} = v_{i^*}(e_{h+1}) = 0$, then Step 13 sets $B(k, \mathbf{t}, \mathbf{s}) = \text{True}$ and also sets $B(h, t_1, \dots, t_{i^*-1}, 0, t_{i^*+1}, \dots, t_n, s_1, \dots, s_{i^*-1}, e_0, s_{i^*+1}, \dots, s_n) = \text{True}$. As the statement holds for $k = h$, there exists an allocation \mathbf{R} of e_1, \dots, e_h such that $v_j(R_j) = t_j$ for $j \neq i^*$ and $v_{i^*}(R_{i^*}) = 0$, and moreover, satisfies property (ii) described in the statement. We now consider \mathbf{R}' with $R'_j = R_j$ for $j \neq i^*$ and $R'_{i^*} = R_{i^*} \cup \{e_{h+1}\}$. Note that $v_{i^*}(R'_{i^*}) = 0$, then one can verify that \mathbf{R}' is an allocation of e_1, \dots, e_{h+1} that satisfies the properties (i) and (ii) described in the statement regarding \mathbf{t} and \mathbf{s} .

If $e_0 \neq s_{i^*} \neq e_{h+1}$ and $t_{i^*} \neq v_{i^*}(e_{h+1})$, then Step 16 sets $B(k, \mathbf{t}, \mathbf{s}) = \text{True}$ and $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, s_1, \dots, s_{i^*}, \dots, s_n) = \text{True}$ and either $0 < |v_{i^*}(s_{i^*})| \leq |v_{i^*}(e_{h+1})|$ or $v_{i^*}(e_{h+1}) = 0$. As the statement holds for $k = h$, there exists an allocation \mathbf{S} of e_1, \dots, e_h such that $v_j(S_j) = t_j$ for $j \neq i^*$ and $v_{i^*}(S_{i^*}) = t_{i^*} - v_{i^*}(e_{h+1})$, and moreover, satisfies property (ii) described in the statement regarding \mathbf{s} . We now consider \mathbf{S}' with $S'_j = S_j$ for $j \neq i^*$ and $S'_{i^*} = S_{i^*} \cup \{e_{h+1}\}$. Since $v_{i^*}(S'_{i^*}) = v_{i^*}(S_{i^*}) + v_{i^*}(e_{h+1}) = t_{i^*}$ and $S'_j = S_j$ for $j \neq i^*$, then we have $v_j(S'_j) = t_j$ for all j . As for property (ii), note that either $|v_{i^*}(s_{i^*})| \leq |v_{i^*}(e_{h+1})|$ or $v_{i^*}(e_{h+1}) = 0$ and s_{i^*} is the item with the least non-zero absolute

value for agent i^* in S_{i^*} , then one can verify that property (ii) regarding \mathbf{s} described in the statement is also satisfied by \mathbf{S}' . Up to here, the statement also holds when $k = h + 1$.

Overall, with mathematical induction, we have proved the “only if” part of the lemma.

Now let us prove the “if” part, again with mathematical induction. In the case of $k = 0$, Step 2 of Algorithm 6 sets $B(0, \mathbf{t}, \mathbf{s}) = \text{True}$ if $s_i = e_0$ and $t_i = 0$ for all i . Accordingly, if $B(0, \mathbf{t}', \mathbf{s}') = \text{False}$, then either $t'_q \neq 0$ or $s'_q \neq e_0$ holds for some q . Note that in the case of $k = 0$, no allocation can make agent q receive non-dummy item s'_q or non-zero value. Thus, the statement holds for the case of $k = 0$. We now assume that the statement holds for $k = 0, \dots, h$ and show that the statement also holds for the case of $k = h + 1$. Fix \mathbf{t} and \mathbf{s} with $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{False}$, and for a contradiction, assume \mathbf{P} is an allocation of e_1, \dots, e_{h+1} satisfying properties (i) and (ii) described in the statement regarding \mathbf{t} and \mathbf{s} . Without loss of generality, we assume $e_{h+1} \in P_{i^*}$. Construct allocation \mathbf{P}' with $P'_j = P_j$ for $j \neq i^*$ and $P'_{i^*} = P_{i^*} \setminus \{e_{h+1}\}$. We then split the proof into four cases based on the possibilities of s_{i^*} and t_{i^*} .

Case 1: $s_{i^*} = e_{h+1}$ and $t_{i^*} = v_{i^*}(e_{h+1})$. As $s_{i^*} \neq e_0$, we must have $t_{i^*} = v_{i^*}(P_{i^*}) \neq 0$, and accordingly, $v_{i^*}(e_{h+1}) \neq 0$ holds. Note that $v_{i^*}(P'_{i^*}) = t_{i^*} - v_{i^*}(e_{h+1}) = 0$. As the statement holds for the case of $k = h$, it is not hard to verify that allocation \mathbf{P}' makes $B(h, t_1, \dots, t_{i^*-1}, 0, t_{i^*+1}, \dots, t_n, s_1, \dots, s_{i^*-1}, e_0, s_{i^*+1}, \dots, s_n) = \text{True}$. Thus, when the for-loop in Step 5 of Algorithm 6 is $i = i^*$, Step 7 sets $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$, a contradiction.

Case 2: $s_{i^*} = e_{h+1}$ and $t_{i^*} \neq v_{i^*}(e_{h+1})$. For this case, we must have $v_{i^*}(e_{h+1}) \neq 0$; otherwise, item s_{i^*} has zero value for agent i^* , contradicting the property (ii) satisfied by \mathbf{P} . Note that $v_{i^*}(P'_{i^*}) = t_{i^*} - v_{i^*}(e_{h+1}) \neq 0$, and accordingly, bundle P'_{i^*} contains items of non-zero value for agent i^* . Denote by e' the item with the least non-zero absolute value in P'_{i^*} . As \mathbf{P} satisfies property (ii) regarding \mathbf{s} and $e' \in P'_{i^*} \subseteq P_{i^*}$, we have $|v_{i^*}(s_{i^*})| \leq |v_{i^*}(e')|$. As for agent i^* 's value, we have $v_{i^*}(P'_{i^*}) = t_{i^*} - v_{i^*}(e_{h+1})$. Consequently, one can verify that allocation \mathbf{P}' makes $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, s_1, \dots, e', \dots, s_n) = \text{True}$. Thus, when the for-loop in Step 5 of Algorithm 6 is $i = i^*$, Step 10 sets $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$, a contradiction.

Case 3: $e_0 = s_{i^*} \neq e_{h+1}$. Note that $s_{i^*} = e_0$ implies $t_{i^*} = v_{i^*}(P_{i^*}) = 0$, and consequently, $v_{i^*}(e_{h+1}) = 0$ holds as $e_{h+1} \in P_{i^*}$. Also, it is evident that $v_{i^*}(P'_{i^*}) = 0$. As the statement holds for the case of $k = h$, it is not hard to verify that \mathbf{P}' makes $B(h, t_1, \dots, t_{i^*-1}, 0, t_{i^*+1}, \dots, t_n, s_1, \dots, s_{i^*-1}, e_0, s_{i^*+1}, \dots, s_n) = \text{True}$. Thus, when the for-loop in Step 5 of Algorithm 6 is $i = i^*$, Step 13 sets $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$, a contradiction.

Case 4: $e_0 \neq s_{i^*} \neq e_{h+1}$. Note that both items s_{i^*} and e_{h+1} are in P_{i^*} , then $|t_{i^*}| = |v_{i^*}(P_{i^*})| \geq |v_{i^*}(s_{i^*}) + v_{i^*}(e_{h+1})|$. Since agent i^* has a non-zero value on item s_{i^*} , it must hold that $|t_{i^*}| \neq |v_{i^*}(e_{h+1})|$. Since s_{i^*} is the item with the least non-zero absolute value for agent i^* in P_{i^*} , then s_{i^*} is also the item with the least non-zero absolute value in P'_{i^*} , and moreover, either $|v_{i^*}(s_{i^*})| \leq |v_{i^*}(e_{h+1})|$ or $v_{i^*}(e_{h+1}) = 0$ holds. For agent i^* 's value, we have $v_{i^*}(P'_{i^*}) = t_{i^*} - v_{i^*}(e_{h+1})$. Consequently, one can verify that allocation \mathbf{P}' makes $B(h, t_1, \dots, t_{i^*} - v_{i^*}(e_{h+1}), \dots, t_n, s_1, \dots, s_{i^*}, \dots, s_n) = \text{True}$ with either $0 < |v_{i^*}(s_{i^*})| \leq |v_{i^*}(e_{h+1})|$ or $v_{i^*}(e_{h+1}) = 0$. Thus, when the for-loop in Step 5 of Algorithm 6 is $i = i^*$, Step 16 sets $B(h + 1, \mathbf{t}, \mathbf{s}) = \text{True}$, another contradiction. Up to here, the statement also holds when $k = h + 1$.

Overall, with mathematical induction, we have also proved the “if” part of the lemma.

A.9 Proof of Theorem 5.2

Note that Algorithm 6 can return $B(m, \mathbf{t}, \mathbf{s})$ for all \mathbf{t} and \mathbf{s} . By visiting the entire $B(m, \mathbf{t}, \mathbf{s})$, we can find the set Γ , of which the construction depends on the underlying items. The full construction of Γ is presented as follows.

$$\Gamma = \begin{cases} \{(\mathbf{t}, \mathbf{s}) | B(m, \mathbf{t}, \mathbf{s}) = \text{True and } t_i + v_j(s_j) \geq t_j \text{ for all } i, j\}, & \text{for goods;} \\ \{(\mathbf{t}, \mathbf{s}) | B(m, \mathbf{t}, \mathbf{s}) = \text{True and } t_i - v_i(s_i) \geq t_j \text{ for all } i, j\}, & \text{for chores.} \end{cases}$$

Given an arbitrary EQX allocation $\mathbf{A}' = (A'_1, \dots, A'_n)$, construct $(\mathbf{t}', \mathbf{s}')$ as follows: for all i , $t'_i = v_i(A'_i)$ and if $v_i(A'_i) = 0$, then $s'_i = e_0$ and otherwise, $s'_i \in A'_i$ is the item with the least non-zero absolute value for agent i . Then, we have $(\mathbf{t}', \mathbf{s}') \in \Gamma$ due to the property of EQX and the construction of \mathbf{t}' and \mathbf{s}' . Accordingly, by visiting all element of Γ , we are able to find the element $(\mathbf{t}^*, \mathbf{s}^*) \in \Gamma$, of which \mathbf{t}^* represents the agents' value in the EQX allocation maximizing egalitarian welfare over all EQX allocations. In particular, one can pursue the $(\mathbf{t}^*, \mathbf{s}^*)$ with $\min_{i \in [n]} t_i^* \geq \min_{i \in [n]} t_i$ for all $(\mathbf{t}, \mathbf{s}) \in \Gamma$. The specific EQX allocation can be found by backtracking $B(m, \mathbf{t}^*, \mathbf{s}^*)$ in the following way: assigning e_m to agent i_m if the value of $B(m, \mathbf{t}^*, \mathbf{s}^*)$ is set to True by $B(m-1, \mathbf{t}^{m-1}, \mathbf{s}^{m-1}) = \text{True}$ and at that time the for-loop in Step 5 is $i = i_m$; then assigning e_{m-1} to agent i_{m-1} if the value of $B(m-1, \mathbf{t}^{m-1}, \mathbf{s}^{m-1})$ is set to True by $B(m-2, \mathbf{t}^{m-2}, \mathbf{s}^{m-2}) = \text{True}$ and at that time the for-loop in Step 5 is $i = i_{m-1}$; repeat this process until all items are assigned. If in some step, the choice of $B(h, \mathbf{t}^h, \mathbf{s}^h)$ is not unique, then arbitrarily pick one.

As for the time complexity, the running time of Algorithm 5 is $O(m^{n+2}V^n)$, and visiting the entire $B(m, \mathbf{t}, \mathbf{s})$ and backtracking takes time $O(m^n V^n)$. Therefore, the running time of the algorithm is $O(m^{n+2}V^n)$.

Acknowledgements We are very grateful to the anonymous referees, whose comments and suggestions have significantly improved the paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Agnetis, A., Chen, B., Nicosia, G., & Pacifici, A. (2019). Price of fairness in two-agent single-machine scheduling problems. *European Journal of Operational Research*, 276(1), 79–87.
2. Agnetis, A., Mirchandani, P. B., Pacciarelli, D., & Pacifici, A. (2004). Scheduling problems with two competing agents. *Operations Research*, 52(2), 229–242.
3. Aumann, Y., & Dombb, Y. (2015). The efficiency of fair division with connected pieces. *ACM Transactions on Economics and Computation*, 3(4), 1–16.
4. Aziz, H., Huang, X., Mattei, N., & Segal-Halevi, E. (2021). Computing welfare-maximizing fair allocations of indivisible goods, [arXiv:2012.03979v2](https://arxiv.org/abs/2012.03979v2) [cs].
5. Balseiro, S. R., Gurkan, H., & Sun, P. (2019). Multiagent mechanism design without money. *Operations Research*, 67(5), 1417–1436.

6. Barman, S., Bhaskar, U., & Shah, N. (2020). Optimal bounds on the price of fairness for indivisible goods. In X. Chen, N. Gravin, M. Hoefer, & R. Mehta (Eds.), *Web and Internet Economics* (pp. 356–369). Cham: Lecture Notes in Computer Science, Springer International Publishing.
7. Bei, X., Lu, X., Manurangsi, P., & Suksompong, W. (2021). The price of fairness for indivisible goods. *Theory of Computing Systems*, 65(7), 1069–1093.
8. Bertsimas, D., Farias, V. F., & Trichakis, N. (2011). The price of fairness. *Operations Research*, 59(1), 17–31.
9. Bertsimas, D., Farias, V. F., & Trichakis, N. (2012). On the efficiency-fairness trade-off. *Management Science*, 58(12), 2234–2250.
10. Bertsimas, D., Papalexopoulos, T., Trichakis, N., Wang, Y., Hirose, R., & Vagefi, P. A. (2020). Balancing efficiency and fairness in liver transplant access: tradeoff curves for the assessment of organ distribution policies. *Transplantation*, 104(5), 981–987.
11. BezákováIvona., & DaniVarsha (2005). Allocating indivisible goods, *ACM SIGecom Exchanges* .
12. Bilò, V., Fanelli, A., Flammini, M., Monaco, G., & Moscardelli, L. (2016). The price of envy-freeness in machine scheduling. *Theoretical Computer Science*, 613, 65–78.
13. Bogomolnaia, A., & Moulin, H. (2004). Random matching under dichotomous preferences. *Econometrica*, 72(1), 257–279.
14. Bogomolnaia, A., Moulin, H., & Sandmirskiy, F. (2022). On the fair division of a random object. *Management Science*, 68(2), 1174–1194.
15. Bogomolnaia, A., Moulin, H., Sandmirskiy, F., & Yanovskaia, E. (2019). Dividing bads under additive utilities. *Social Choice and Welfare*, 52(3), 395–417.
16. Bogomolnaia, A., Moulin, H., Sandmirskiy, F., & Yanovskaya, E. (2017). Competitive division of a mixed manna. *Econometrica*, 85(6), 1847–1871.
17. Bogomolnaia, A., Moulin, H., & Stong, R. (2005). Collective choice under dichotomous preferences. *Journal of Economic Theory*, 122(2), 165–184.
18. Bouveret, S., Cechlárová, K., & Lesca, J. (2019). Chore division on a graph. *Autonomous Agents and Multi-Agent Systems*, 33(5), 540–563.
19. Caragiannis, I., Kaklamanis, C., Kanellopoulos, P., & Kyropoulou, M. (2012). The efficiency of fair division. *Theory of Computing Systems*, 50(4), 589–610.
20. Chaudhary, S., Ramjee, R., Sivathanu, M., Kwatra, N., & Viswanatha, S. (2020). Balancing efficiency and fairness in heterogeneous GPU clusters for deep learning. In: *Proceedings of the Fifteenth European Conference on Computer Systems*, ACM, Heraklion Greece, pp. 1–16.
21. Dickerson, J. P., Procaccia, A. D., & Sandholm, T. (2014). Price of fairness in kidney exchange. In: *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, AAMAS '14, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 1013–1020.
22. Dubins, L. E., & Spanier, E. H. (1961). How to cut a cake fairly. *The American Mathematical Monthly*, 68(1), 1–17.
23. Engelmann, D., & Strobel, M. (2004). Inequality aversion, efficiency, and maximin preferences in simple distribution experiments. *The American Economic Review*, 94(4), 857–869.
24. Fehr, E., & Schmidt, K. M. (1999). A theory of fairness, competition, and cooperation. *The Quarterly Journal of Economics*, 114(3), 817–868.
25. Freeman, R., Sikdar, S., Vaish, R., & Xia, L. (2019). Equitable allocations of indivisible goods, In: *Proceedings of the twenty-eighth international joint conference on artificial intelligence, International Joint Conferences on Artificial Intelligence Organization*, Macao, China, pp. 280–286.
26. Freeman, R., Sikdar, S., Vaish, R., & Xia, L. (2020). Equitable allocations of indivisible chores, In: *Proceedings of the 19th international conference on autonomous agents and multiagent systems*, AAMAS '20, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, pp. 384–392.
27. Garey, M. R., & Johnson, D. S. (1990). *Computers and intractability: A guide to the theory of NP-completeness*. USA: W. H. Freeman & Co.
28. Ghodsi, M., Hajiaghayi, M. T., Seddighin, M., Seddighin, S., & Yami, H. (2021). Fair allocation of indivisible goods: Improvement. *Mathematics of Operations Research*, 46(3), 1038–1053.
29. Gkatzelis, V., Psomas, A., & Tan, X. (2021). Fair and efficient online allocations with normalized valuations. *Proceedings of the AAAI conference on artificial intelligence*, 35(6), 5440–5447.
30. Gourvès, L., Monnot, J., & Tlilane, L. (2014). Near fairness in matroids, In: *21st European conference on artificial intelligence (ECAI 2014)*, Vol. 263 of *Frontiers in artificial intelligence and applications*, Prague, Czech Republic, pp. 393–398.
31. Herreiner, D. K., & Puppe, C. D. (2009). Envy freeness in experimental fair division problems. *Theory and Decision*, 67(1), 65–100.

32. Heydrich, S., & van Stee, R. (2015). Dividing connected chores fairly. *Theoretical Computer Science*, 593, 51–61.
33. Höhne, F., & van Stee, R. (2021). Allocating contiguous blocks of indivisible chores fairly. *Information and Computation*, 281, 104739.
34. Jiang, Y., Wu, X., Chen, B., & Hu, Q. (2021). Rawlsian fairness in push and pull supply chains. *European Journal of Operational Research*, 291(1), 194–205.
35. Lesmana, N. S., Zhang, X., & Bei, X. (2019). Balancing Efficiency and Fairness in On-Demand Ridesourcing, *Advances in Neural Information Processing Systems*, Vol. 32.
36. Myerson, R. B. (1981). Utilitarianism, egalitarianism, and the timing effect in social choice problems. *Econometrica*, 49(4), 883.
37. Okun, A. M. (2015). *Equality and efficiency: The big tradeoff*. Washington, D.C.: Brookings Institution Press.
38. Sun, A., Chen, B., & Vinh Doan, X. (2021). Connections between fairness criteria and efficiency for allocating indivisible chores, In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multiagent Systems*, Richland, SC, pp. 1281–1289.
39. Zeng, D., & Psomas, A. (2020). Fairness-efficiency tradeoffs in dynamic fair division, In: *Proceedings of the 21st ACM conference on economics and computation, EC '20*, Association for Computing Machinery, New York, NY, USA, pp. 911–912.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.