

A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation

Haris Aziz¹, Hervé Moulin^{2,3}, and Fedor Sandomirskiy^{3,4}

¹ UNSW Sydney and Data61 Sydney, Australia
haziz@cse.unsw.edu.au

² University of Glasgow, Glasgow, UK
herve.moulin@glasgow.ac.uk

³ Higher School of Economics, St. Petersburg, Russia

⁴ Technion, Haifa, Israel
sandomirski@yandex.ru

Abstract. We consider fair allocation of indivisible items under additive utilities. When the utilities can be negative, the existence and complexity of an allocation that satisfies Pareto optimality and proportionality up to one item (PROP1) is an open problem. We show that there exists a strongly polynomial-time algorithm that always computes an allocation satisfying Pareto optimality and proportionality up to one item even if the utilities are mixed and the agents have asymmetric weights. We point out that the result does not hold if either of Pareto optimality or PROP1 is replaced with slightly stronger concepts.

1 Introduction

We consider fair allocation of indivisible items under additive utilities. For an agent, an item can be a good (yielding positive utility) or a chore (yielding negative utility). Fair allocation of indivisible items has received renewed interest since it was proved that a *Pareto optimal (PO)* and *envy-free up to one item (EF1)* allocation exists for positive utilities (Caragiannis, Kurokawa, Moulin, Procaccia, Shah, and Wang, 2016). However, the existence and complexity of a Pareto optimal (PO) and EF1 allocation is open when utilities may be negative. The complexity of computing such an allocation is also open for the case of positive utilities. In view of these open questions, a natural relaxation of EF1 called *proportionality up to one item (PROP1)* has started to receive deeper interest. PROP1 requires each agent gets utility that is at least her proportionality guarantee if she loses her biggest chore or alternatively obtain the biggest good allocated to some other agent.

Interestingly, even the existence and complexity of PROP1 and Pareto optimal allocation has been an open problem when the utilities are mixed or even negative (see e.g. Freeman and Shah (2019)). We study this central problem. In previous work, the existence of PROP1 and PO allocations has been established

only in the context of goods (positive utilities) and very recently for the case of chores.

For the case of goods, [Conitzer, Freeman, and Shah \(2017\)](#) posed the complexity of computing a PROP1 and PO allocation as an open problem. They had proved that a PROP1 and PO outcomes always exists even for a public decision making setting that is more general than allocation of indivisible goods. [Barman and Krishnamurthy \(2019\)](#) presented a strongly polynomial-time algorithm that always finds a PROP1 and PO allocation for positive utilities. [Brânzei and Sandomirskiy \(2019\)](#) proved that there exists a strongly polynomial-time algorithm for chores that always finds a weighted PROP1 and PO allocation if the number of agents or items is fixed. For mixed utilities, [Aziz, Caragiannis, Igarashi, and Walsh \(2019b\)](#) presented a strongly polynomial-time algorithm to compute a PROP1 and PO allocation when the number of agents is two.

Contribution We show that even for the case of mixed utilities and any number of agents, an fPO (property stronger than PO) and PROP1 allocation always exists. In particular, we design a strongly polynomial-time algorithm that achieves fPO and PROP1 even if the number of agents or items is not fixed, the utilities are mixed and the agents have asymmetric weights. We obtain as corollaries several recent results that have been proved for only goods or only chores or for weaker requirements. Our existence and algorithmic result concerns PO and PROP1 as the central concepts. The result cannot be improved in several directions. If PO is replaced by welfare maximization, the problem becomes NP-hard. If PROP1 is replaced by EF1, then even the existence of an allocation satisfying both EF1 and PO is an open problem. Finally, if PROP1 is replaced by PROPX (a property stronger than PROP1), then even the existence of a PROPX allocation is not guaranteed.

2 Preliminaries

We consider the allocation of m items in set O to n agents in set N . Each agent $i \in N$ has a weight $b_i > 0$ where $\sum_{i \in N} b_i = 1$.

Each item is allocated fully. If each item is allocated to exactly one agent, we call the allocation *integral*. We will generally denote a fractional allocation by $x = (x_1, \dots, x_n)$ where x_i is the allocation of agent i and $x_{i,o}$ is the fraction of item o given to agent i . We will typically denote an integral allocation by π where π_i denotes the allocated set of items of agent i .

By $u_i(o)$ we denote the agent i 's utility of receiving the whole item o . The utilities $u_i(o)$ may have mixed signs: an item o can be a chore for some i ($u_i(o) < 0$), a good for another agent j ($u_j(o) > 0$), and a neutral item for some agent k ($u_k(o) = 0$). Agents have additive utilities over allocations: $u_i(x_i) = \sum_{o \in O} u_i(o)x_{i,o}$ and similarly $u_i(\pi_i) = \sum_{o \in \pi_i} u_i(o)$ in case of integral allocation.

An allocation y Pareto improves an allocation x if $u_i(y_i) \geq u_i(x_i)$ for all $i \in N$ and for some i the inequality is strict. We will call an integral allocation

Pareto optimal (PO), if no integral allocation improves it. An allocation that cannot be improved by any fractional allocation is called *fPO*. Clearly, an fPO integral allocation is PO as well.

An allocation x is *weighted PROP* if for each agent $i \in N$, $u_i(x_i) \geq u_i(O)b_i$. An integral allocation π is *weighted PROP1* if for each agent $i \in N$,

- $u_i(\pi_i) \geq u_i(O)b_i$; or
- $u_i(\pi_i) + u_i(o) \geq u_i(O)b_i$ for some $o \in O \setminus \pi_i$; or
- $u_i(\pi_i) - u_i(o) \geq u_i(O)b_i$ for some $o \in \pi_i$.

In the literature, PROP1 was studied with respect to goods by [Conitzer et al. \(2017\)](#). It has recently been considered for mixed utilities ([Aziz et al., 2019b](#)).

When $b_i = 1/n$ for all $i \in N$, weighted PROP1 is equivalent to PROP1.

For any fractional or integral allocation x , the corresponding consumption graph is a bipartite graph with vertices $(N \cup O)$ and the edge set $E = \{\{i, o\} \mid x_{i,o} > 0\}$. If an agent i shares an item with an agent j we call j , agent i 's neighbor.

3 Algorithm

We show that a PO and weighted PROP1 always exists and it can be computed in strongly-polynomial time even for mixed utilities. One possible algorithmic approach for achieving such allocations is to start from a PROP1 allocation and find a Pareto optimal Pareto improving allocation. However, finding an integral allocation that is a Pareto improvement over another integral allocation is generally a computationally hard problem (see e.g. [Aziz, Biro, Lang, Lesca, and Monnot. \(2019a\)](#) and [de Keijzer, Bouveret, Klos, and Zhang \(2009\)](#)) even for the case of goods. Also, a Pareto improvement over a PROP1 allocation may not even satisfy PROP1. We provide explicit examples (Examples 2 and 3) for this phenomenon.

In view of the challenges encountered in finding Pareto improvements while maintaining PROP1, we take another route that has been popularized recently (see e.g. [Barman and Krishnamurthy \(2019\)](#) and [Sandomirskiy and Segal-Halevi \(2019\)](#)): deal with fractional allocations that are Pareto improving and convert them to suitable integral allocations.

Our idea is to start with a fractional proportional allocation x^{PROP} and then find a fractional Pareto optimal allocation x that Pareto improves x^{PROP} . We ensure that the consumption graph of x is acyclic. The acyclicity of the consumption graph is critically used to carefully round x into an integral allocation. The algorithm is described as Algorithm 1. The way the rounding is done is illustrated in Example 1. Starting from the consumption tree of the allocation x^{PROP} , the algorithm picks an arbitrary agent i who shares some items with other agents, and rounds all his fractions to his advantage: give to i any good a he shares, and give any bad b to someone with whom he was sharing. In the subtree

starting at an agent j who was sharing a or b with i , break all the other partial shares of j to her advantage. And so on. The acyclicity of the tree guarantees that this algorithm terminates and returns an allocation.

Algorithm 1: Algorithm to find a weighted PROP1 and PO allocation

Input: An instance $I = (N, O, u, b)$
Output: Integral allocation x^*

- 1 Start with a proportional allocation x^{PROP} that gives a share b_i of each item to each agent.
- 2 Find an fPO fractional allocation x that Pareto dominates x^{PROP} and has an acyclic consumption graph G_x . (Computed via the algorithm from Lemma 2.5 in (Sandomirskiy and Segal-Halevi (2019)))
- 3 Round the fractional allocation x into an integral allocation x^* as follows:
- 4 If some j shares an item o for which $u_j(o) = 0$, we give it fully to lowest-index agent i who shares o in x . Update x .
- 5 $Q \leftarrow \emptyset$, an empty FIFO (First-In-First-Out) queue of agents
- 6 **while** *there is an agent i sharing exactly one item o with others* **do**
 - 7 */* By acyclicity of G_x , such i exists if there is at least one shared item */*
 - 8 Add i to Q
 - 9 **while** Q is non-empty **do**
 - 10 Take the first agent j out of Q
 - 11 Add all the neighbors of j to the end of Q
 - 12 **for** each o shared by j **do**
 - 13 **if** $u_j(o) > 0$ **then**
 - 14 | give o fully to j
 - 15 **else if** $u_j(o) < 0$ **then**
 - 16 | give o to a neighbor with whom o is shared
 - 17 Update x
- 18 **return** $x^* = x$

Lemma 1. *Algorithm 1 returns an integral allocation that is weighted PROP1 in time $O(n^2m^2(n+m))$.*

Proof. Allocation x is computed in time $O(n^2m^2(n+m))$ according to the algorithm of Sandomirskiy and Segal-Halevi (2019). After that, the consumption graph G_x is computed in time $O(n \cdot m)$. It has $n+m$ vertices V , and at most $n+m-1$ edges E since, by acyclicity, G_x is a collection of trees. “Root agents” i from the outer while-cycle, one per each subtree of G_x , can be found in $O(|V| + |E|) = O(n+m)$ by the depth-first search. For each such agent i , the internal while-cycle represents breadth-first exploration of the tree $T_i = (V_i, E_i)$ rooted at i with constant-time rounding operations at each step. For each tree, it takes time $O(|V_i| + |E_i|)$ and hence the whole outer while-cycle can be implemented in $O(|V| + |E|) = O(n+m)$. Thus the overall time-complexity is determined by Pareto improvement phase of the algorithm.

We next argue that the returned allocation satisfies weighted PROP1. Since x initially Pareto dominates x^{PROP} , it satisfies weighted PROP. There are only two reasons why an agent j 's utility may decrease. Either a good o partially consumed by j at x is allocated to another agent at x^* or j gets increased share of a bad o . Since all the items within the for-cycle are rounded in favor of j , the only possibility is that o is an item shared by j and its predecessor i in the breadth-first search. Therefore, there is at most one such item o and the resulting allocation x^* satisfies PROP1. \square

Lemma 2. *Algorithm 1 returns an integral allocation that is fPO.*

Proof. Allocation x is fPO in Step 2. Since x is fPO, due to known results by [Varian \(1976\)](#), it maximizes weighted welfare $\sum_i \lambda_i u_i(x_i)$ for some strictly positive weighting $\lambda = (\lambda_i)_{i \in N}$ of the agents (see also Lemma 2.3 in [Sandomirskiy and Segal-Halevi \(2019\)](#) for a particular case of mixed items). After that we modify x to x^* so that G_{x^*} is a subgraph of G_x . Hence each item o is still consumed by agents with highest weighted utility $\lambda_i \cdot u_i(o)$. Therefore x^* maximizes welfare for the same weighting of the agents as x . Thus x^* is fPO as well. \square

Based on the two lemmas we get the following.

Theorem 1. *For mixed utilities, there always exists an integral allocation that satisfies weighted PROP1 and fPO. Furthermore, there exists a strongly polynomial-time algorithm in $n + m$ that returns such an allocation.*

We obtain several recent results as corollaries of Theorem 1.

Corollary 1 ([Aziz et al. \(2019b\)](#)). *For two agents and mixed utilities, a Pareto-optimal and PROP1 allocation exists and can be computed in strongly polynomial time.*

Corollary 2 ([Barman and Krishnamurthy \(2019\)](#)). *For positive utilities, an fPO and PROP1 allocation can be computed in strongly polynomial time.*

Corollary 3 ([Brânzei and Sandomirskiy \(2019\)](#)). *For negative utilities, a Pareto-optimal and weighted PROP1 allocation can be computed in strongly polynomial time if the number of agents or items is fixed.*

Example 1 (Illustration of how our algorithm rounds a fractional allocation into an integral allocation).

Consider a fractional allocation represented in Table 1. The allocation has a consumption graph represented in Figure 1. Since graph is acyclic, it can be viewed as a tree rooted with agent 1. The fractional allocation is rounded to an integral allocation as follows. Agent 1 who is at the root of the tree is given a and b whereas c which is a chore of agent 1 is given to agent 5 who is a ‘neighbor’ of agent 1. After that agent 1’s neighbor 2 gets d , 3 gets e , 4 gets f and g , and 5 gets h . The new integral allocation is represented in Table 2.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
1	+	+	-	+	+	+	+	+
2	+	-	-	+	+	+	+	+
3	+	+	-	-	+	-	-	-
4	-	+	-	-	-	+	+	-
5	-	-	-	+	+	-	+	+
6	-	+	-	+	+	+	+	+

Table 1. Table for Example 1 indicating the signs of the utilities of agents as well as an allocation. A square indicates that the agent is allocated a part of the item. The allocation has an acyclic consumption graph represented in Figure 1.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
1	+	+	-	+	+	+	+	+
2	+	-	-	+	+	+	+	+
3	+	+	-	-	+	-	-	-
4	-	+	-	-	-	+	+	-
5	-	-	-	+	+	-	+	+
6	-	+	-	+	+	+	+	+

Table 2. Table for Example 1 indicating the signs of the utilities of agents as well as an allocation. A square indicates that the agent is allocated a part of the item. The allocation is integral.

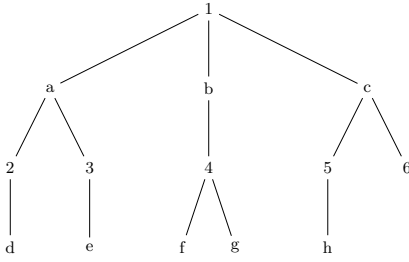


Fig. 1. Acyclic consumption graph corresponding to the allocation in Table 1.

4 Discussion

Recently, approaches based on maximin share fairness (a property weaker than proportionality) have been considered for computing fair allocation of indivisible goods to *asymmetric* agents (Farhadi, Hajiaghayi, Ghodsi, Lahaie, Pennock, Seddighin, Seddighin, and Yami, 2019, Babaioff, Nisan, and Talgam-Cohen, 2017, Aziz, Chan, and Li, 2019c). The results in these papers are either for the case of goods or for chores whereas we consider mixed utilities. Our approach uses weighted PROP1 which is a relaxation of the more traditional proportionality guarantee.

Algorithm 2: Algorithm to find a Pareto improvement

Input: An instance $I = (N, O, u, y)$
Output: fPO allocation x that Pareto improves allocation y and for which G_x is acyclic

- 1 $G_x \leftarrow$ complete bipartite graph
- 2 $T \leftarrow \emptyset$
- 3 **while** consumption graph G_x has some cycle C **do**
- 4 Solve LP_T with optimum value opt_T :
- 5 $\max \sum_{i \in N} (\sum_{o \in O} u_i(o) \cdot x_{i,o})$ s.t.

$\sum_{i \in N} \sum_{o \in O} u_i(o) \cdot x_{i,o} \geq \sum_{i \in N} \sum_{o \in O} u_i(o) \cdot y_{i,o}$	for all $i \in N$
$\sum_{i \in N} x_{i,o} = 1$	for all $o \in O$
$x_{i,o} = 0$	for all $(i, o) \in T$
$x_{i,o} \geq 0$	for all $i \in N$ and $o \in O$
- 7 **if** there exists some $(i, o) \in C$ such that $\text{opt}_T = \text{opt}_{T \cup \{(i, o)\}}$ **then**
 | $T \leftarrow T \cup \{(i, o)\}$
- 8 **return** $x = x^*$

Our strongly polynomial-time algorithm relies in Step 2 on an algorithm proposed by Sandomirskiy and Segal-Halevi (2019). One may wonder whether there is a conceptually simpler self-contained algorithm in Step 2 achieving an fPO

allocation that is an fPO improvement over x^{PROP} and has an acyclic consumption graph. Algorithm 2 satisfies these requirements. It maximizes the sum of utilities subject to proportionality so the resultant allocation x^* is fPO and proportional. Acyclicity of G_{x^*} is ensured by running the while loop. If an interim fPO allocation x has a cycle C in G_x , there always exists an allocation x' such that all agents get the same utilities and the graph $G_{x'}$ is a subgraph of G_x but does not contain some edge $\{i, o\}$ from C (existence of x' follows from a “cyclic trade” argument as in the proof of Lemma 2.5 from Sandomirskiy and Segal-Halevi (2019)). Therefore, we can put the edge $\{i, o\}$ to T without affecting the optimal value of the LP. Thus the consumption graph of the final allocation x^* contains no cycle C .

Since Algorithm 2 uses linear programming, it only gives a guarantee of weakly polynomial-time. Note that instead of Algorithm 2, one can simply solve $LP_{T=\emptyset}$ via the simplex algorithm which returns a basic feasible solution, i.e, the extreme point of the set of solutions. It is easy to see that for such an extreme solution x^* , the graph G_{x^*} is acyclic. Indeed, if x^* contains a cycle C , the allocation can be represented as the convex combination of allocations obtained by “forward trade” and “backward trade” along C . The simplex algorithm works very well in practice but can in theory take exponential time in the worst case.

Extending the result We now point out that our result is un-improvable or challenging to improve in several respects. If PROP1 is replaced by the stronger property of EF1, then it is an open problem whether an EF1 and PO allocation exists or not (Aziz et al., 2019b). The problem remains open even for the case of chores.

If PROP1 is strengthened to a concept called proportionality up to the extreme item (PROPX), then the existence of an allocation satisfying the property is not guaranteed for the case of goods (Moulin, 2019). We provide a self-contained and simpler example (Example 4).

Finally, one may wonder whether our main result can be strengthened by considering maximum welfare rather than Pareto optimality. However, computing an allocation that is utilitarian-maximal within the set of PROP1 allocations is NP-hard (Aziz, Huang, Mattei, and Segal-Halevi, 2019d).

Bibliography

- H. Aziz, P. Biro, J. Lang, J. Lesca, and J. Monnot. Efficient reallocation under additive and ordinal preferences. *Theoretical Computer Science*, 2019a.
- H. Aziz, I. Caragiannis, A. Igarashi, and T. Walsh. Fair allocation of combinations of indivisible goods and chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2019b.
- H. Aziz, H. Chan, and B. Li. Weighted maxmin fair share allocation of indivisible chores. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2019c.
- H. Aziz, X. Huang, N. Mattei, and E. Segal-Halevi. Fairness-welfare interpolation in indivisible item allocation. 2019d.

- M. Babaioff, N. Nisan, and I. Talgam-Cohen. Competitive equilibria with indivisible goods and generic budgets. *CoRR*, abs/1703.08150, 2017. URL <http://arxiv.org/abs/1703.08150>.
- S. Barman and S. Kumar Krishnamurthy. On the proximity of markets with integral equilibria. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1748–1755, 2019.
- S. Brânzei and F. Sandomirskiy. Algorithms for competitive division of chores. *CoRR*, abs/1907.01766, 2019.
- I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. The unreasonable fairness of maximum nash welfare. In *Proceedings of the 17th ACM Conference on Electronic Commerce (ACM-EC)*, 2016.
- V. Conitzer, R. Freeman, and N. Shah. Fair public decision making. In *Proceedings of the 18th ACM Conference on Electronic Commerce (ACM-EC)*, pages 629–646. ACM Press, 2017.
- B. de Keijzer, S. Bouveret, T. Klos, and Y. Zhang. On the complexity of efficiency and envy-freeness in fair division of indivisible goods with additive preferences. In *Proceedings of the 1st ADT Conference*, 2009.
- A. Farhadi, M. Hajiaghayi, M. Ghodsi, S. Lahaie, D. M. Pennock, M. Seddighin, S. Seddighin, and H. Yami. Fair allocation of indivisible goods to asymmetric agents. *Journal of Artificial Intelligence Research (JAIR)*, 2019.
- R. Freeman and N. Shah. Recent Advances in Fair Resource Allocation (ACM EC 2019 Tutorial). 2019.
- H. Moulin. Fair division in the age of internet. *Annual Review of Economics*, 11:407–441, 2019.
- F. Sandomirskiy and E. Segal-Halevi. Fair division with minimal sharing. *CoRR*, abs/1908.01669, 2019. URL <http://arxiv.org/abs/1908.01669>.
- H. R. Varian. Two problems in the theory of fairness. *Journal of Public Economics*, 5(3):249 – 260, 1976.

A Examples

Example 2 (Pareto improvement over a PROP1 allocation may not even satisfy PROP1 when there are goods.). Consider the following instance with 3 agents and 31 items. Items in sets B and C are divided respectively into 10 and 20 smaller items each

	A	$B = \{b_1, \dots, b_{10}\}$	$C = \{c_1, \dots, c_{20}\}$
1	0.3	0.2	0.5
2	0.34	0.16	0.5
3	0.16	0.5	0.34

Initially the allocation x is as follows (indicated via the squares). Agent 2 and 3 get utility 0.34 which exceeds the proportionality value. Agent 1 gets total utility 0.2 but the utility increases to 0.5 if agent 1 additionally gets item A . Therefore, the allocation is PROP1.

	A	$B = \{b_1, \dots, b_{10}\}$	$C = \{c_1, \dots, c_{20}\}$
1	0.3	0.2	0.5
2	0.34	0.16	0.5
3	0.16	0.5	0.34

Suppose we obtain the following Pareto improving allocation y .

	A	$B = \{b_1, \dots, b_{10}\}$	$C = \{c_1, \dots, c_{20}\}$
1	0.3	0.2	0.5
2	0.34	0.16	0.5
3	0.16	0.5	0.34

Agent 2 and 3 get utility 0.5 which exceeds the proportionality value. Agent 1 gets total utility 0.3 but even if agent 1 is given any other item, the total utility does not exceed $1/3$. Therefore, allocation y is not PROP1. \diamond

Example 3 (Pareto improvement over a PROP1 allocation may not even satisfy PROP1 when there are chores.). Consider the following instance with 3 agents and 12 items. Items in sets A are divided respectively into 10 smaller items.

	$A = \{a_1, \dots, a_{10}\}$	B	C
1	-0.4	-0.5	-0.1
2	-0.3	-0.6	-0.1
3	-0.6	-0.1	-0.3

Initially the allocation x is as follows (indicated via the squares). The allocation is PROP1.

	$A = \{a_1, \dots, a_{10}\}$	B	C
1	-0.4	-0.5	-0.1
2	-0.3	-0.6	-0.1
3	-0.6	-0.1	-0.3

Suppose we obtain the following Pareto improving allocation y .

	$A = \{a_1, \dots, a_{10}\}$	B	C
1	-0.4	-0.5	-0.1
2	-0.3	-0.6	-0.1
3	-0.6	-0.1	-0.3

Allocation y is not PROP1 because even if agent 1 gets rid of one of the small A items, her utility is $-0.36 < -1/3$. \diamond

An integral allocation π satisfies *proportionality up to extreme item (PROPX)* if for each agent $i \in N$,

- $\forall o \in \pi_i$ s.t. $u_i(o) < 0$: $u_i(\pi_i \setminus \{o\}) \geq u_i(O)/n$; and
- $\forall o \notin \pi_i$ s.t. $u_i(o) > 0$: $u_i(\pi_i \cup \{o\}) \geq u_i(O)/n$.

Example 4 (a PROPX allocation may not exist for the case of goods).

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
1	3	3	3	3	1
2	3	3	3	3	1
3	3	3	3	3	1

In any most balanced allocation one agent gets two big items, one agent gets one big item (utility 3) and the small item *e*, and one agent gets only one big item.

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
1	3	3	3	3	1
2	3	3	3	3	1
3	3	3	3	3	1

The last agent does not achieve the proportionality value of $13/3 > 4$ even if she gets the small item. Therefore PROPX is not satisfied.