Fairness Notions for Rent Division

Parth Gupta, Rayyan Shahid, Rishi Sarraf

November 2022

1 The Rent Division Problem

We study the rent division problem. The problem defined as follows:

Given n agents, n rooms and a valuation of each agent for each room, we need to allocate exactly one room to each agent and assign a price to each room so that certain fairness notions are guaranteed.

First we need to ensure envy-freeness among agents. For that, we define a utility function u_{ij} of each agent a_i for each room r_j. Then for an allocation of rooms σ at price vector p, the envy-freeness notion says that $u_{i\sigma_i} \geq u_{i\sigma_j}$ for all agents a_i and a_j. We define the utility function $u_{ij} = v_{ij} - p_j$. We assume that the utility function is quasi-linear.
It is observed that envy-freeness is often far from sufficient to ensure fairness. Hence, we explore papers that define further notions on top of envy-freeness. We study maximin, equitable 1 and lexislack 4 notions of fairness. We study how to obtain allocations satisfying these notions in polynomial time. Next we define our own notion of fairness called minimax. We proceed to compare it with other notions and prove some nice properties of maximin. We end this analysis with a user survey to emperically compare the notions of fairness.
Finally we explore more papers that attempt to relax the quasi-linear assumption on utility functions and make it a bit more general (and realistic).

2 Existing Fairness Notions

We will explore various existing fairness notions in the following section.

2.1 Envy Freeness

An assignment σ, coupled with a price vector p is envy free if the following condition holds.

$$u_i(\sigma, p) \geq v_{ij} - p_j \ \forall i, j \in [1, n]$$

Under the quasi linear utility assumption, the existence of envy free solutions is guaranteed. However, sometimes 'fair' envy free solutions look unreasonable. Therefore, a variety of fairness notions which maximize some other objective on top of envy freeness constraint have been explored in literature.

2.2 Equitable

An equitable allocation tries to reduce the disparity in the utilities of the agents. Formally, it minimizes the maximum absolute difference between the utilities of the agents subject to envy-freeness. Mathematically, $(\sigma, p)_{\texttt{equitable}} = \arg\min_{\sigma,p} \max_{i,j} |u_i(\sigma, p) - u_j(\sigma, p)|$

2.3 Maximin

Maximin is another fairness notion which tries reduce disparity among the agents. Formally, it tries to maximize the minimum utility of all the agents. Mathematically, $(\sigma, p)_{\texttt{maximin}} = \arg\max_{\sigma,p} \min_{i\in[1,n]} u_i(\sigma, p)$.

It turns out that maximin is a strictly stronger condition than equitability. This is captured by the following theorem

Theorem 1. *Every maximin envy-free allocation is equitable.1*

2.4 Maxislack 4

We first define slack, which signifies the magnitude of envy-freeness. Formally, slack is defined as the minimum, taken over all agents and all rooms, of the difference of an agent's utility for its own room and its utility for some other room. For allocation σ and price p,

$$\texttt{slack}(\sigma, p) = \min_{i\in[1,n]} \min_{j\neq\sigma(i)} (v_{i\sigma(i)} - p_{\sigma(i)}) - (v_{ij} - p_j)$$

A maxislack solution maximizes the slack. The slack is guaranteed to be non-negative for maxislack solutions (any envy free solution has non-negative slack). Maximizing slack makes the solutions more robust to small changes in the valuations of the agents. If the slack is positive, the envy freeness constraint is maintained even if there are small changes in the valuations of the agents.

2.5 Lexislack

It turns out that maxislack solutions are not unique. Hence, there is a further scope of optimization which is captured by the lexislack solution. Lexislack solution refines the maxislack solution by using a leximin strategy. The lexislack rule selects an allocation (σ, p) that maximizes the smallest of the n^2 values $((v_{i\sigma(i)} - p_{\sigma(i)}) - (v_{ij} - p_j))$ $i, j \in n$, and subject to that maximizes the second-smallest of these values, and so on. In contrast to the maxislack rule, the lexislack rule is essentially single-valued.

3 Computing fair solutions

In this section, we describe the polynomial time algorithms for computing the allocations satisfying the different fairness notions described in section 2. [1]

3.1 Walrasian Equilibrium

We digress to define the notion of Walrasian Equilibrium as it will be essential for establishing the correctness of the algorithm for computing fair solutions. Imagine a setting of n agents $a_1, \ldots a_n$

[1] We have implemented all the common fairness schemes for rent division in our GitHub repo

and m rooms $R = \{r_1, \dots, r_m\}$. However, every agent can now be allocated a set of rooms and the valuations of agent a_i is defined by the valuation function $v_i : 2^R \to \mathbb{R}$ which assigns a value to every subset of rooms. A walrasian equilibrium is an allocation $\mathcal{A}$ (where $\mathcal{A}_i$ is the set of rooms assigned to agent a_i) coupled with a price vector p (where $p(r_i)$ is the price for room r_i) such that the following condition is satisfied.

$$\forall i \in [1, n] \; \forall S \subseteq R \; : v_i(\mathcal{A}_i) - p(\mathcal{A}_i) \geq v_i(S) - p(S)$$

Here, p is an additive function such that $p(S) = \sum_{r \in S} p(r)$.
We will now present a few results for this general setting.

Definition 1. *A allocation $\mathcal{A}$ is called **welfare-maximizing** if it maximizes the total welfare*

Theorem 2. ***1st Welfare Theorem.*** *If $(\mathcal{A}, p)$ is a walrasian equilibrium then $\mathcal{A}$ is welfare-maximizing.*

Theorem 3. ***2nd Welfare Theorem.*** *If $(\mathcal{A}, p)$ is a walrasian equilibrium and $\mathcal{A}'$ is welfare-maximizing allocation, then $(\mathcal{A}', p)$ is also a walrasian equilibrium. Moreover, $\forall i \in [1, n] \; : v_i(\mathcal{A}_i) - p_i(\mathcal{A}_i) = v_i(\mathcal{A}'_i) - p_i(\mathcal{A}'_i)$*

If we wish to use the above results for the envy free rent division setting, we would need to show some kind of equivalence between the two. This equivalence can be achieved when the valuation function in the general setting is defined as maximum of the valuations of individual rooms. Formally, consider the setting of n agents $a_1, \dots, a_n$ and n rooms $R = \{r_1, \dots, r_n\}$ where the valuation function $v_i : 2^R \to \mathbb{R}$ of agent a_i is defined as $v_i(S) = \max_{r \in S} v_i(r)$. If $(\mathcal{A}, p)$ is a walrasian equilibrium, then $\mathcal{A}$ must be welfare maximizing. Note that we will always have a welfare maximizing allocation where each agent is assigned a single room. Suppose there exists agents a_i, a_j such that $|\mathcal{A}_i| \geq 2$ and $\mathcal{A}_j = \phi$. Let r be the least valued room for agent a_i in its allocation $\mathcal{A}_i$. We create a new allocation $\mathcal{A}'$ where $\mathcal{A}'_i = \mathcal{A}_i \setminus \{r\}$ and $\mathcal{A}'_j = \{r\}$. The total welfare for $\mathcal{A}'$ exceeds the welfare of $\mathcal{A}$ by $v_j(r)$ (non negative). Therefore, if $\mathcal{A}$ is welfare maximizing, so is $\mathcal{A}'$. Moreover, from Theorem 3, it follows that $(\mathcal{A}', p)$ is a walrasian equilibrium.
As a result, in this setting, there always exists walrasian equilibrium $(\mathcal{A}', p)$ where $\mathcal{A}'$ is a one-to-one assignment of rooms to agents. For such an allocation, the envy freeness constraint is identical to the walrasian equilibrium constraint. A careful analysis reveals that Theorem 2 and Theorem 3 are also valid in the envy free rent division setting. Hence, we have the following two results

Theorem 4. *If (σ, p) is an envy free solution, then σ is welfare-maximizing.*

Theorem 5. *If (σ, p) is an envy free solution and σ' is welfare-maximizing assignment, then (σ', p) is also an envy free solution. Moreover, $\forall i \in [1, n] \; : u_i(\sigma, p) = u_i(\sigma', p)$*

3.2 A general optimisation equation

Theorem 6. *Let $f_1, \dots, f_t : \mathbb{R}^n \to \mathbb{R}$ be linear functions, where t is polynomial in n. Given a rent division instance V, a solution (σ, p) that maximizes $\min_{q \in [1,t]} f_q(u_1(\sigma, p), \dots, u_n(\sigma, p))$ subject to envy freeness can be computed in polynomial time.*

A linear program based algorithm for the above optimization equation is as follows

1. Compute a welfare maximizing assignment σ by solving the maximum weight bipartite matching problem, where $\forall i \in [1, n]$ $\forall j \in [1, n]$ agent a_i and room r_j is connected by an edge of weight v_{ij}
2. Compute the price vector p by solving the following linear program
   Variables: p, R
   Constraints:
   (a) $R \leq f_q(v_{1\sigma(1)} - p_{\sigma(1)}, \ldots, v_{n\sigma(n)} - p_{\sigma(n)})$ $\forall q \in [1, t]$
   (b) $\sum_{i=1}^{i=n} p_i = 1$
   (c) $v_{i\sigma(i)} - p_{\sigma(i)} \geq v_{ij} - p_j$ $\forall i, j \in [1, n]$

   Objective: $\max R$

Correctness
Suppose the objective is maximized subject to envy freeness for the solution (σ', p'). Now, let us denote by (σ, p) the solution obtained by the algorithm. Since σ is a welfare maximizing assignment, from Theorem 5, (σ, p') is also an envy free solution with $u_i(\sigma, p') = u_i(\sigma', p)$ $\forall i \in [1, n]$. Therefore, the value of objective R is equal is both cases. Since (σ, p') is in the search space of the linear program, therefore, the value of the objective R for (σ, p) is at least as large as the value of objective for (σ', p'). Hence, the algorithm indeed maximizes the objective over all possible envy free solutions.

3.3 Computing Maximin solution

Consider the following reduction to the general optimization equation.
Let $t = n$ and $f_q(u_1(\sigma, p), \ldots, u_n(\sigma, p)) = u_q(\sigma, p)$ $\forall q \in [1, t]$. Then, maximizing $\min_{q\in[1,t]} f_q(u_1(\sigma, p), \ldots, u_n(\sigma, p))$ maximizes the minimum utility. We follow the same algorithm described in subsection 3.2.

3.4 Computing Equitable solution

Consider the following reduction to the general optimization equation.
Let $t = n^2$ and $f_{ij}(u_1(\sigma, p), \ldots, u_n(\sigma, p)) = u_i(\sigma, p) - u_j(\sigma, p)$ $\forall (i, j) \in [n] \times [n]$. Then, maximizing $\min_{q\in[1,t]} f_q(u_1(\sigma, p), \ldots, u_n(\sigma, p))$ minimizes the maximum absolute difference in the utilities of the agents. We follow the same algorithm described in subsection 3.2.

3.5 Computing Maxislack solution

Consider the following reduction to the general optimization equation.
Let $t = n(n-1)$ and $f_{ij}(u_1(\sigma, p), \ldots, u_n(\sigma, p)) = u_i(\sigma, p) - (v_{ij} - p_j)$ $\forall (i, j \neq i)$. Then, maximizing $\min_{q\in[1,t]} f_q(u_1(\sigma, p), \ldots, u_n(\sigma, p))$ maximizes the slack. We follow the same algorithm described in subsection 3.2.

3.6 Computing Lexislack solution

Firstly, we find a welfare maximizing assignment σ. We then follow the leximin strategy for computing the lexislack solution that requires solving $\mathcal{O}(n^4)$ LPs. Denote by Δ_{ij} the value $u_i(\sigma, p) - (v_{ij} - p_j)$. The algorithm is described below.

1. Initialize a set `non-fixed` of Δs as the set of all $n(n-1)$ Δ_{ij}. Also initilize a map `fixed` of Δs as an empty map.
2. Solve the following linear program to maximize the minimum of the `non-fixed` Δs
Variables: p, L
Constraints:
   (a) $L \leq \Delta_{ij}\ \forall \Delta_{ij} \in$ `non-fixed`
   (b) $\Delta_{ij} = \texttt{fixed}[\Delta_{ij}]\ \forall \Delta_{ij} \in keys(\texttt{fixed})$
   (c) $\sum_{i=1}^{i=n} p_i = 1$
   (d) $v_{i\sigma(i)} - p_{\sigma(i)} \geq v_{ij} - p_j\ \forall i, j \in [1, n]$

   Objective: $\max L$
3. There must exist a non-fixed Δ such that for achieving the lexislack solution, it must be assigned the value L. For finding such Δ, iterate over the `non-fixed` Δs and for each $\Delta_{i_1j_1}$, check if the following LP admits a solution.
Variables: p
Constraints:
   (a) $\Delta_{i_1j_1} > L$
   (b) $\Delta_{ij} \geq L\ \forall \Delta_{ij} \in \texttt{non-fixed} \setminus \{\Delta_{i_1j_1}\}$
   (c) $\Delta_{ij} = \texttt{fixed}[\Delta_{ij}]\ \forall \Delta_{ij} \in keys(\texttt{fixed})$
   (d) $\sum_{i=1}^{i=n} p_i = 1$
   (e) $v_{i\sigma(i)} - p_{\sigma(i)} \geq v_{ij} - p_j\ \forall i, j \in [1, n]$

   Objective: `NONE`
Add the first $\Delta_{i_1j_1}$, for which the above LP doesn't admit a solution, to the `fixed` map with $\texttt{fixed}[\Delta_{i_1j_1}] = L$ and also remove it from the `non-fixed` set.
4. Continue steps 2 and 3 for $n(n-1)$ iterations until the `non-fixed` set becomes empty.
5. Solve the LP in step 2 without the maximization objective to find the lexislack solution (σ, p)

4 Our Contributions

4.1 Minimax

Minimax allocation is a notion of fairness that we have defined. It says that given n agents, n rooms and the valuation v of each agent for each room, we need to find an envy free allocation that minimizes the utility of the agent that is best off (the agent with the highest utility). Since the allocation is envy-free, the sum of the utilities is fixed. Hence, minimizing the maximum utility aims to minimizes the disparity among the agents in some sense.

4.1.1 Computing Minimax allocation

First we find a welfare maximizing assignment σ. We fix the room allocation to be σ, which is necessary to ensure envy-freeness. Next, we need to assign a price to each room.
We use the optimization equation presented in the paper "Which is the Fairest Rent Division of Them All?". Given t(a polynomial in n) linear functions $f_1, f_2, \ldots, f_t$ in the agent's utilities, it is possible to find the values of utilities that maximizes $\min(f_1, f_2, \ldots f_t)$.
We set $t = n$ and $f_i(u_{1\sigma_1}, u_{2\sigma_2}, \ldots u_{n\sigma_n}) = -u_{i\sigma_i}$. Hence,

$$\max_p \min_t (f_1, f_2, \ldots, f_n) = -\min_p \max_t (u_{1\sigma_1}, u_{2\sigma_2}, \ldots u_{n\sigma_n})$$

So, the price vector p returned on solving the optimization formula minimzes the maximum utility. Hence (σ, p) is our required minimax allocation. Since solving the optimization formula takes polynomial time, it is possible to find a minimax allocation in polynomial time.

4.1.2 Minimax is Equitable

Theorem 7. *A minimax envy-free allocation with price vector p is equitable.*

Proof. Consider a graph $G = (V, E)$ where $V = (1, 2, \ldots, n)$ where i represents agent a_i. There is an edge from i to j if agent a_i weakly envies a_j i.e. $u_{i\sigma_i} = u_{i\sigma_j}$. We call an agent a_i poor if he/she has the minimum utility in the minimax allocation and rich if he/she has the maximum utility.
We claim that there is a path from each rich agent to each non-rich agent in G. Suppose for the sake of contradiction, this is not the case. There is a set of vertices $T(\neq \phi)$ s.t there is not path to them from a poor agent. Consider a price vector q as follows:

$$q_i = \begin{cases} p_i - \epsilon & i \in T \\ p_i + \frac{\epsilon |T|}{n - |T|} & i \notin T \end{cases}$$

Notice that the sum of prices is still 1. Also, since each rich agent has a path to itself, no rich agent belongs to T. Consequently, the utility of each rich agent decreases. Hence, the maximum utility is less for price vector q compared to p. We choose ϵ small enough so that if any agent x didn't weakly envy y before, it doesn't envy y now. We need to check envy from an agent i to j if it weakly envied j before. If $i \in T$ and $j \notin T$, then the utility of i increases and utility of j decreases. Hence, i doesn't envy j. If $i \in T$ and $j \in T$, then the utility of i and j change by same amount. If $i \notin T$ and $j \notin T$, then also the utilities change by same amount. Finally we are left with the case $i \notin T$ and $j \in T$. This case is not possible because there would then be a path from a rich agent to j, which contradicts the definition of T. So, there must be a path to each agent from a rich agent.
If all rich agents are poor, then disparity is 0, hence the allocation is equitable and there is nothing to prove. So, we assume that there is atleast one rich agent that is not poor.
Let $U(p)$ denote the maximum utility under price vector p. Further, let $D(p)$ denote the disparity under price vector p.
Now, let's assume the the allocation under p is not equitable. Assume that there is another price vector q which makes the allocation equitable. Also, let $U(q) - U(p) = \epsilon$. Note that $\epsilon \geq 0$ since p gives a minimax allocation. So,

- For a rich agent i under p, we have $p_i - q_i \leq \epsilon$
- For any poor agent i under p, we have $p_i - q_i > \epsilon$. This follows from the earlier case and the fact that $D(p) > D(q)$.

Consider any path from a rich agent x to a poor agent y. We have $p_x - q_x < p_y - q_y$. So, there must be an edge (i, j) in the path s.t. $p_i - q_i < p_j - q_j$. Moreover, since there is an edge from i to j, we have $v_{ii} - p_i = v_{ij} - p_j$. So,

$$v_{ii} - q_i = v_{ii} - p_i + (p_i - q_i) < v_{ij} - p_j + (p_j - q_j) = v_{ij} - q_j$$

contradicting envy freeness under the allocation for price vector q. So, we have a contradiction. Hence, p must give an equitable allocation. □

4.1.3 Comparing Minimax and Maximin

Observation 1: For a welfare maximizing assignment σ, if it is possible to assign p such that $u_{i\sigma_i} = u_{j\sigma_j}$ for all agents i, j (subject to envy-freeness), then this solution is both maximin and minimax.

Proof: Since the allocation is envy-free, $\sum_{i=1}^{n} u_{i\sigma_i}$ is a constant. Thus, if we try to increase one utility, some other utility must decrease. Thus, the current allocation maximizes the minimum and hence, is a minimax allocation. By similar reasoning, this is also a maximin allocation. Hence, we are done.

Observation 2: For $n = 2$, the minimax and maximin envy-free allocations are the same (and both are equitable with zero disparity).

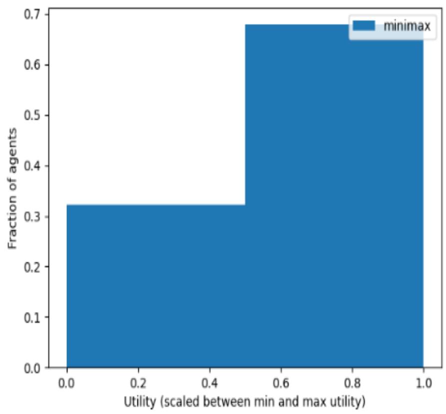
Proof: Since we need a welfare maximizing allocation (for envy-freeness), we assign room r_i to the agent which values it higher. WLOG, assume $\sigma_1 = r_1$ and $\sigma_2 = r_2$. Then consider prices $p = (p_1, p_2)$ s.t. $v_{11} - p_1 = v_{22} - p_2$ and $p_1 + p_2 = 1$. This price vector ensures that the utilities of the agents are equal. We claim that this allocation is also envy-free. To see this, observe that $u_{12} = v_{12} - p_2 \leq v_{22} - p_2 = v_{11} - p_1 = u_{11}$. Hence, $u_{12} \leq u_{11}$. Hence, a_1 doesn't envy a_2. By similar reasoning, a_2 doesn't envy a_1. Now we have an envy-free allocation which equalizes the utility among both agents. By Observation 1, it is both maximin and minimax. Hence we have proved our claim.

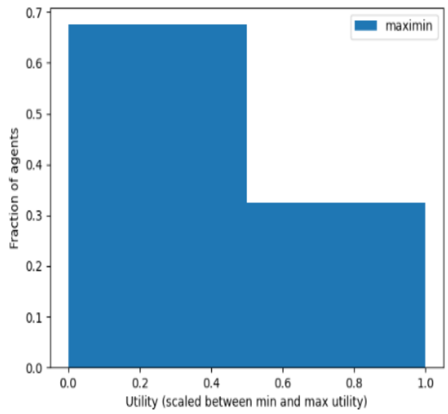
Observation 3: Let m_1 and M_1 be the minimum and maximum utilities for a minimax allocation and m_2 and M_2 be the minimum and maximum utilities for a maximin allocation. Then,

1. $M_1 - m_1 = M_2 - m_2$
2. $m_1 \leq m_2 \leq M_1 \leq M_2$

Proof: Part 1 follows from the fact that both maximin and minimax allocations are equitable.
For Part 2, notice that $M_1 \leq M_2$ by definition of maximin. Similarly, $m_1 \leq m_2$ due to definition of maximin. For $m_2 \leq M_1$ notice that the sum of utilities is fixed due to envy-freeness, so if $m_2 > M_1$ then all utilities in minimax case would be less than all utilities in maximin case, so the sum cannot be the same. Hence, $m_2 \leq M_1$.
Further note that for a minimax allocation the utilities would be clustered more towards maximum side. On the other hand, for maximin case, the utilities would clustered more towards minimum side. This again follows from Observation 3 and the fact that the sum of utilities is same in either case.

To verify with this, we created a random rent division instance with $n = 10$, found the minimax and maximin allocation and divided the agents into 2 clusters in either case. If the minimum utility is m and maximum is M, then one cluster contains all the agents with utility between $[m, \frac{m+M}{2})$ and the other cluster contains all agents with utility in $[\frac{m+M}{2}, M]$. We plotted a bar graph to display the count of agents in each cluster. Below is the graph obtained for both allocations:





We can verify that the results indeed agree with the observations.

4.2 Maximin and Minimax are not robust

Theorem 8. *Given a minimax/maximin allocation* (σ, p) *if the utilties of all agents are not equal, then slack=0.*

Proof. We provide the proof for a minimax allocation. The proof for maximin allocation is similar. Suppose for the sake of a contradiction, we have a minimax allocation (σ, p) with unequal utilities where slack is not 0. Then consider the set T of agents with maximum utility. We decrease the price of room σ_i by ϵ if $i \in T$. Otherwise, we increase the price of room σ_i by $\frac{\epsilon|T|}{n-|T|}$. This decreases the maximum utility in the allocation, keeping the sum of prices fixed. Also, if slack is greater than 0, then no agent weakly envies any other agent. Consequently, it is possible to choose a small enough ϵ s.t. no agent envies any other agent now. Thus, we have an allocation with smaller value of maximum utility contradicting the fact that (σ, p) was minimax. Hence, we are done. $\square$

So, we have proved that slack is 0 when utilities are not equal. We don't expect utilities to be equal often in real scenarios. So, small changes in agent's valuations can easily make a previously minimax/maximin allocation to not be envy-free now. Hence, these are not robust and are subject to envy among agents given small changes in valuations.

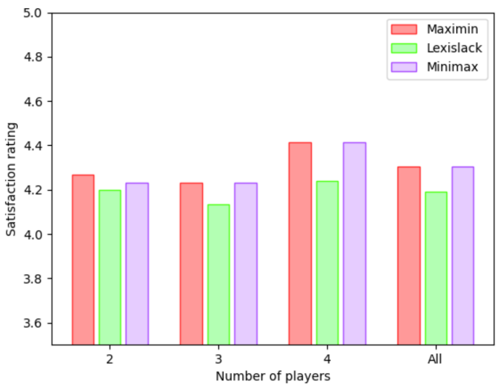
4.3 User Study

We simulated the rent division problem by artificially constructing 3 instances of apartments and asked users their valuations. We grouped their responses and computed the Maximin, Lexislack and Minimax allocations. The users were then asked to rate the allocations. The survey was conducted over 31 users, each participating in 3 instances. The following questions were posed in the survey:

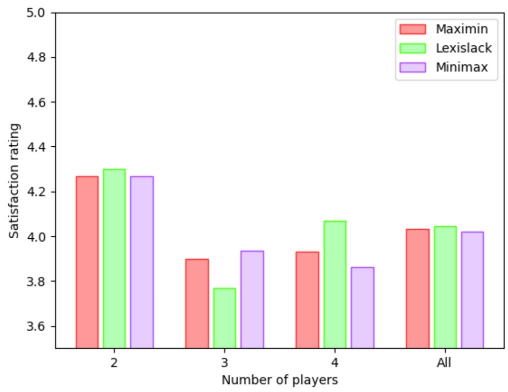
1. **Individual**: This question relates to your own allocation. In other words, we would like you to pay attention only to your own benefit. How happy are you with getting the allocated room for the asked price?

2. **Others**: This question relates to the allocation for everyone else. How fair do you rate the allocation for other agents?

Here are the results of the survey:



Individual



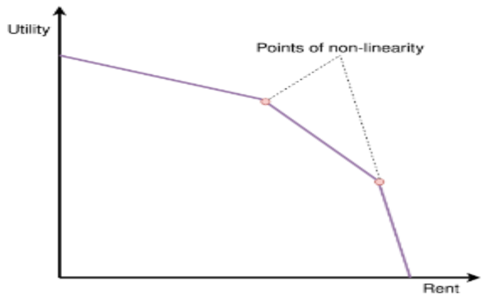
Others

We observe that for individual satisfaction, lexislack allocation performs slightly worse than the other 2 allocations. For the others satisfaction, all 3 perform more or less same.

Remark: Our survey was designed to replicate the survey conducted by the authors of 1 to gain a comparative insight about above notions of fairness. The 'others' question of this survey appeared ambiguous to the users, and led to various interpretations of the questions asked. In particular, it wasn't very clear if they were supposed to get into the other users' shoes and judge fairness or look through the lens of their own valuations. To get a more meaningful insight, the survey can be redesigned by asking an unambiguous question. Nevertheless, 'Individual' fairness shows a significant preference of the users towards maximin and minimax over lexislack.

5 Relaxing Quasi Linear Utilities

It is possible to relax the quasi-linear utility assumption and have continuous, piecewise non-linear utilities. The utility function may have many points where the slope changes. Such a utility function may often be seen in the real world scenario. Each agent may have a "soft-budget" and it will be willing to pay that amount. However, if the price of its room exceeds his soft-budget, he/she may have to take a loan decreasing the utility more sharply. Here is how a non-linear utility function looks like:



The following results have been proved about such utility functions:

1. For a single point of non-linearity, there exists polynomial time algorithms to find envy-free allocations.

2. For more than one point of non-linearity, there exists polynomial time approximation algorithms to find an allocation.

6 Conclusion

We observed that there are many ways to ensure fairness (on top of envy-freeness). We explored maximin, equitable and lexislack. We then define our own notion - minimax. It is clearly stronger than equitable. It is in some sense better than maximin since the utilities are clustered more towards the maximum side. The user study also reveals that it gives more satisfying results than lexislack allocation. However, our comparision between minimax and maximin allocations is a bit informal and intuitive. Further work is required to define formal notions to compare the two and see if one outperforms the other in the real world setting.

7 References

1. Kobi Gal, Ariel D. Procaccia, Moshe Mash, and Yair Zick. 2018. Which is the fairest (rent division) of them all? Commun. ACM 61, 2 (February 2018), 93–100. `https://doi.org/10.1145/3166068`

2. `https://www.mimuw.edu.pl/~ps219737/courses/scw/slides-morning/Rent_Division_and_Spliddit.pdf`

3. Fully Polynomial-Time Approximation Schemes for Fair Rent Division `https://arxiv.org/pdf/1807.04163.pdf`

4. Robust Rent Division `https://dominik-peters.de/publications/robust-rent-division.pdf`

5. A polynomial algorithm for maxmin and minmax envy-free rent division on a soft budget `https://arxiv.org/pdf/2002.02966.pdf`