A polynomial algorithm for maxmin and minmax envy-free rent division on a soft budget

Rodrigo A. Velez

Received: 29 August 2020 / Accepted: 15 December 2021 / Published online: 22 January 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract
The current practice of envy-free rent division, led by the fair allocation website Spliddit, is based on quasi-linear preferences. These preferences rule out agents’ well documented financial constraints. To resolve this issue we consider an extension of the quasi-linear domain that admits differences in agents’ marginal disutility of paying rent below and above a given reference, i.e., a soft budget. We construct a polynomial algorithm to calculate a maxmin utility envy-free allocation, and other related solutions, in this domain.

1 Introduction

1.1 Overview of the problem

The envy-free rent division problem is one of the success stories of computational fair division. It addresses the allocation of rooms and payments of rent among roommates who lease an apartment or house. The objective is to find a recommendation in which each roommate finds their assignment is at least as good as that of the other roommates, an envy-free allocation (Foley 1967).

From a computational perspective, which is our main interest, research on this problem first concentrated on quasi-linear environments. An early result showed the existence of a polynomial algorithm to calculate an envy-free allocation that maximizes the minimal payment of rent among agents (Aragones 1995). More recently, thanks to an Associate Editor, a referee, Ariel Procaccia, and seminar audiences at UT Austin, St. Petersburg Workshop on Mechanism Design, and SAET2021 for useful comments. All errors are my own.

See Velez (2018) for a survey on existence, structural, and incentives issues related to this problem.

---

Rodrigo A. Velez
rvelezca@tamu.edu
https://sites.google.com/site/rodrigoavelezswebpage/home

1 Department of Economics, Texas A&M University, College Station, TX 77843, USA
Gal et al. (2017) argued that there are compelling differences among envy-free allocations, presented evidence that these differences are perceived by agents, identified the maxmin utility envy-free allocations as possible candidates to minimize these issues, and constructed a polynomial algorithm to calculate one of these allocations. 

Passing quickly from theory to practice, the fair allocation website Spliddit (Goldman and Procaccia 2014) implemented Gal et al. (2017)’s approach. Along the tens of thousands of instances in which the algorithm has been used, the users of the system have provided a series of requests for its improvement. From this feedback, Procaccia et al. (2018) identified the inability of the system to handle budget constraints as its main shortcoming.

As a response, Procaccia et al. (2018) proposed to elicit agents’ values for rooms and hard budget constraints, and constructed a polynomial algorithm to determine whether an envy-free allocation satisfying the budget constraints exists. Whenever the set of envy-free allocations satisfying budget constraints is non-empty, Procaccia et al. (2018)’s algorithm returns a maxmin utility allocation constrained to this set.

Obviously, it is possible that no envy-free allocation satisfying budget constraints exists. What to do then? If we insist on no-envy, we need to decide who ends up paying above budget and by how much. If we were able to ask agents for their complete preference map and calculate an envy-free allocation for these reports, the problem would be resolved. This is unrealistic, however.

Our approach is to allow agents report preferences in a larger domain of preferences that can encode the difficulty they have to pay above a certain amount. We propose to ask agents for their valuation of their rooms; the amount they have earmarked for housing, i.e., their soft budget constraint; and an index that quantifies the disutility of paying rent above their soft budget. More precisely, the agent’s utility is the value they assign to the room they receive, minus the disutility of paying rent. In our domain, the marginal disutility of paying rent is constant below and above the soft budget, but can be higher for payments above this threshold. We refer to the increase in marginal disutility of paying rent above the agent’s soft budget as the budget violation index. Our results are obtained under the assumption that the number of budget violation indices that are admissible is finite and fixed throughout. We refer to this domain as budget-constrained quasi-linear preferences.

Our main result is the construction of an algorithm, polynomial in the number of agents, to calculate a maxmin envy-free allocation and some other prominent selections from the envy-free set when preferences are in our domain.

Even though quasi-linear mechanisms do not provide agents with reports that are expressive enough to encode their financial difficulties, they have several attractive

---

2 It is even possible that no allocation satisfying budget constraints exists.
3 Whenever no envy-free allocation satisfying budget constraints exist, Procaccia et al. (2018)’s algorithm returns an allocation that minimizes the maximal budget violation among the envy-free allocations for the reported quasi-linear preference. If there is heterogeneity among the agents’ disutility of paying above their budget, these allocations may not be envy-free. See Velez (2019) for an extensive discussion.
4 Soft budgets were introduced by Nicoló and Velez (2017) as a motivation for the study of partnership dissolution problems with non-quasi-linear preferences. In the context of rent division, Budget-constrained quasi-linear preferences were first proposed by Velez (2018).
features. First, elicitation of reports can be done with intuitive questions. Second, computation is feasible in polynomial time. Third, incentives are aligned, to some extent, with the designer’s objective (Velez 2015, 2018).

Our contribution is then to address the computational complexity of the most prominent selections from the envy-free set in our preference domain. We establish that our proposal preserves the practicality of quasi-linear mechanisms. In a companion paper we address the elicitation of reports and the incentives issues created by our domain enlargement (Velez 2019). We provide details in Sect. 3.1.

1.2 Our contribution

We construct a family of polynomial algorithms that compute allocations selected by the maxmin and minmax utility and maxmin and minmax money linear selections of the envy-free set with budget-constrained quasi-linear preferences (Theorem 1). Section 2 presents a precise definition of these selections. For the moment it is relevant to add that they contain the following prominent selections of the envy-free set.

- Maxmin utility envy-free allocation: maximizes the minimal utility across agents within the envy-free set. This selection is advocated by Gal et al. (2017).
- Best envy-free allocation for a given agent: maximizes a given agent’s utility within the envy-free set. This selection is minimally manipulable on several partial orders of manipulability (Andersson et al. 2014).
- Maxmin rent envy-free allocation: maximizes the minimum rent paid by any agent. Calculating such an allocation determines the existence of envy-free allocations in which no agent is compensated to receive a room.
- Minmax rent envy-free allocation: minimizes the maximal rent paid by any agent. This selection implements the maximal uniform rent control that is compatible with no-envy.
- An allocation that minimizes, among all envy-free allocations, the rent paid by the agent who is assigned a given room.
- An allocation that maximizes, among all envy-free allocations, the rent paid by the agent who is assigned a given room.

1.3 Significance for CS and economics

Our contribution is significant in two independent levels.

First, we address an issue of algorithms that has been identified from real-world requests in one of the most successful applications of computational social choice.

Second, at a technical level, we develop new techniques for the design and analysis of algorithms in non-linear domains. We do so by connecting two independent branches of the literature on envy-free rent division. On the one hand, we work on the algorithmic framework developed by Gal et al. (2017) and Arunachaleswaran et al. (2018). On the other hand, we intensively use the topological properties of the envy-free set that has been studied by economists for arbitrary continuous
preferences (Alkan et al. 1991; Velez 2017, 2018). In particular, we manage to leverage a monotonicity property of the solutions covered by our theorem and exploit it from a computational perspective. We provide a precise account of the relationship of our work with previous literature in the body of the paper.

The closest paper to ours is Arunachaleswaran et al. (2018). These authors study the computation of an envy-free allocation when preferences are piece-wise linear, a domain that contains our budget-constrained quasi-linear preferences. They develop a basic algorithm and bound its complexity in two cases. First, when all values of marginal disutility of paying rent belong to a fixed set. Second, when all values of marginal disutility of paying rent are powers of a fixed factor. Based on this second application they construct a Fully Polynomial Time Approximation Scheme.

Our algorithms improve compared to Arunachaleswaran et al. (2018)’s in that we calculate allocations satisfying further criteria of justice within the envy-free set. Our selections, which include the celebrated maxmin utility envy-free allocations, have the further property that if the rent of the house needs to be revised, each agent’s welfare changes in the same direction (Alkan et al. 1991; Velez 2017). For instance, if rent decreases, all agents benefit from this. Our results perfunctorily extend to piece-wise linear preferences and the complexity of our algorithms for the powers model can also be bounded along the lines of Arunachaleswaran et al. (2018). It is not clear what is the normative content of an allocation that maximizes further criteria of justice on a powers model approximation of a particular problem. Because of this, and to a greater extent because we see the fixed indices model as a realistic choice for applications, we restrict our presentation to this environment.

2 Model

A set of \(n\) rooms, \(A := \{a, b, \ldots \}\), is to be allocated among \(n\) agents \(N := \{1, \ldots, n\}\). Each agent is to receive a room and pay an amount of money for it. Agent \(i\)’s generic allotment is \((r_a, a) \in \mathbb{R} \times A\). When \(r_a \geq 0\) we interpret this as the amount of money agent \(i\) pays to receive room \(a\). We allow for negative payments of rent, i.e., \(r_a < 0\).\(^5\)

Each agent has preferences on bundles of rooms and payments of rent represented by a utility function of the following form. There are \((v^a_i)_{a \in A} \in \mathbb{R}^A, b_i \geq 0,\) and \(\rho_i \in \mathbb{R}_+\) such that

\[
u_i(r_a, a) = v^a_i - r_a - \rho_i \max\{0, r_a - b_i\}.
\]  

\(^5\) It is possible to restrict preferences to a cartesian domain that guarantees payments of rent are always nonnegative at each envy-free allocation (Velez 2011). Essentially, one needs to guarantee that the difference in value of the rooms is not too large compared to the house rent. Our results obviously apply to these subdomains. Spliddit requires agents report a positive value for each room that add up to the house rent. This normalization guarantees no agent is compensated to receive a room at any envy-free allocation when there are two agents, but not when there are more than two agents. However, with Spliddit’s normalization, the real-world instances in which negative prices are necessary are rare (Gal et al. 2017).
The space of these preferences is $B$. We refer to them as budget-constrained quasi-linear preferences.

Our preference domain has an intuitive structure. The agent has an underlying quasi-linear preference whose baseline marginal disutility of rent is normalized to one. This baseline preference is modified by two coefficients $b_i$ and $\rho_i$. Coefficient $b_i$ can be interpreted as a housing earmark or budget. The marginal disutility of rent paid above $b_i$ is $(1 + \rho_i)$. Thus, paying one dollar above $b_i$ entails a larger disutility than that of paying one dollar more still under $b_i$. This reflects that either the agent is budget constrained and needs to pay an interest rate on money paid above $b_i$, or that to pay over her housing earmark she needs to reallocate money from other needs for which she has decreasing marginal utility captured by a step function.

**Example 1** (Budget-constrained quasi-linear preference) Consider a three-agent problem where $N = \{1, 2, 3\}$ and $A = \{a, b, c\}$. Agent 1 has budget-constrained quasi-linear utility with values $v_1^a = 800$, $v_1^b = v_1^c = 500$, $b_1 = 600$, and $\rho_1 = 1$. The preference is illustrated in Fig. 1. This figure shows three axes, one for each room. Each point in an axis represents the bundle of room and corresponding rent payment. For instance, $(600, a)$ is the bundle in which an agent pays 600 for room $a$. Preference $u_1$ is illustrated by means of its indifference curves, i.e., lines that join points that are indifferent for the agent. For instance agent 1 is indifferent between $(600, a)$, $(300, b)$ and $(300, c)$.

Indifference curves for $u_1$ can be divided in three different regimes. To the left of the indifference curve passing through $(600, a)$, all curves are horizontal translations of each other. This happens because the rent payments on all these bundles are weakly to the left of the agent’s budget. The utility of these bundles is given by $u_1(r_a, \alpha) = v_1^a - r_a$ where $\alpha \in A$. Thus, for low enough rent $u_1$’s indifference curves coincide with those of a quasi-linear preference. To the right of the indifference curve passing through $(750, a)$, again all curves are horizontal translations of each other. This happens because the rent payments on all these bundles are weakly to the right of the agent’s budget. The utility of these bundles is given by $u_1(r_a, \alpha) = (v_1^a + \rho_1 b_1) - (1 + \rho_1) r_a$ where $\alpha \in A$. Note that this utility function represents the same preference as $\hat{u}_1(r_a, \alpha) = (v_1^a + \rho_1 b_1)/(1 + \rho_1) - r_a$, which is a quasi-linear utility. Thus, for high enough rent, $u_1$’s indifference curves again...
coincide with those of a quasi-linear preference. In between these two regimes, indifference curves proportionally interpolate the indifference curves passing through (600, a) and (750, a).

Example 1 illustrates a general feature of budget-constrained quasi-linear preferences. For low enough rents, these preferences are quasi-linear. That is, the rankings between bundles that pay rent to the left of the agent’s budget coincide with the rankings of a quasi-linear preference. Surprisingly, for high enough rents, this is also the case. The budget-constrained quasi-linear utility for bundles with rent above the agent’s budget is a linear transformation of a normalized quasi-linear utility. Thus, for high enough rent, the preference is quasi-linear.

Individual payments should add up to \( m \in \mathbb{R} \), the house rent. An economy is described by the tuple \( e := (N, A, u, m) \). An allocation for \( e \) is a pair \( (r, \sigma) \) where \( \sigma : N \to A \) is a bijection and \( r := (r_a)_{a \in A} \) is such that \( \sum_{a \in A} r_a = m \). An allocation is envy-free for \( e \) if no agent prefers the consumption of any other agent at the allocation. The set of these allocations is \( F(e) \).

The weak-envy graph for allocation \( (r, \sigma) \) in economy \( (N, A, u, m) \), \( \Gamma(u, r, \sigma) = (N, E) \), is the directed graph where \( (i, j) \in E \) if and only if agent \( i \) is indifferent between her allotment at \( (r, \sigma) \) and \( j \)'s allotment, i.e., \( u_i(r_{\sigma(i)}, \sigma(i)) = u_i(r_{\sigma(j)}, \sigma(j)) \). If there is a path from \( i \) to \( j \) in \( \Gamma(u, r, \sigma) \) we write \( i \to_{u, r, \sigma} j \).

3 Results

3.1 The main result

For constants \( a > 0 \) and \( b \in \mathbb{R} \), we refer to the function \( x \in \mathbb{R} \mapsto b + ax \) as a positive affine linear transformation. We develop a family of polynomial algorithms to calculate an allocation that maximizes (minimizes) the minimum (maximum) of positive affine linear transformations of individual utility (or payments of rent) among all envy-free allocations when preferences belong to \( \mathcal{B} \).

**Theorem 1** Let \( R \subseteq \mathbb{R}_+ \) with cardinality \( k \in \mathbb{N} \); and \( e := (N, A, u, m) \) such that \( u \in \mathcal{B}^N \) and for each \( i \in N \), \( \rho_i \in R \). For \( S \subseteq N \) and \( C \subseteq A \), let \( (f_i)_{i \in S} \) and \( (g_a)_{a \in C} \) be lists of positive affine linear transformations. Then, for fixed \( k \), there are algorithms, polynomial in \( n \), that compute an element in:

1. \( \arg \max \min_{(r,\sigma) \in F(e)} f_i(u_i(r_{\sigma(i)}), \sigma(i)). \)
2. \( \arg \min \max_{(r,\sigma) \in F(e)} f_i(u_i(r_{\sigma(i)}), \sigma(i)). \)
3. \( \arg \max \min_{a \in C} g_a(r_a). \)
4. \( \arg \min \max_{a \in C} g_a(r_a). \)
Our statement of Theorem 1 makes clear that the complexity of our algorithms is kept in check as we scale the number of agents in the problem, while keeping the number of admissible budget violation indices constant.

In a companion paper, Velez (2019), we prove that if the set of admissible budget violation indices has no upper bound, the complete information incentives of budget-constrained quasi-linear envy-free mechanisms may lead to inefficient outcomes. By contrast, if there is a bound on these reports, complete information non-cooperative outcomes of these mechanisms are exactly the set of envy-free allocations for agents’ true preferences. Thus, this incentives analysis suggests the designer should fix a maximum value for the budget violation index. In practice, this determines a finite set of possible reports.

Additionally, one can think of the budget violation indices as the interest rate at which the agent has access to credit (see Velez 2019). A natural bound for these values is the maximal legal interest rate, when usury laws apply. Thus, even without considering incentives issues, in practice, it is natural that the range of reports offered to each agent be independent of the number of agents in the problem.

For concreteness, we initially limit our discussion to our main application, i.e., the maxmin utility envy-free solution

\[ \mathcal{R}(N, A, u, m) := \arg \max_{(r, \sigma) \in \mathcal{F}(N, A, u, m)} \min_{i \in N} u_i(r_{\sigma(i)}, \sigma(i)). \]

We discuss the proof of the theorem in Sect. 3.4.

Example 2 (Maxmin utility envy-free allocations with budget-constrained quasi-linear preferences) Consider the economy partially described in Example 1. Let \( u_2 \) and \( u_3 \) be identical budget-constrained quasi-linear preferences with values \( v_a^2 = v_b^2 = v_c^2 = 600, b_2 = 600 \), and \( \rho_2 = 1/2 \).

Verifying that an allocation is a maxmin utility envy-free allocation can be done in polynomial time. An allocation is a maxmin utility envy-free allocation, if and only if, it is envy-free and for each agent, say \( i \), there is a path in the associated weak-envy graph for the allocation from \( i \) to one of the agents whose utility is minimal at the allocation (Proposition 5.9, Velez 2018).

A maxmin utility envy-free allocation may require some agents pay above their budget even when there are envy-free allocations satisfying budget constraints. For instance, suppose that \( m = 1800 \). Let \( \sigma \) be the assignment in which agent 1 receives room \( a \), agent 2 room \( b \), and agent 3 room \( c \). The allocation with assignment \( \sigma \) and in which agent 1 pays 680 and agents 2 and 3 pay 560 each is envy-free. Moreover, each agent has the same utility, i.e., 40. Since the envy-free graph at each allocation is reflexive, there is path from each agent to an agent with minimal utility at this allocation. Thus, the allocation is a maxmin utility envy-free allocation. Now, the allocation with assignment \( \sigma \) and in which each agent pays 600 is also envy-free. It is not a maxmin utility envy-free allocation, however. Agent 1’s utility at the allocation is 200. Agents 2 and 3’s utility is zero. Agent 1 prefers her allotment at the allocation to the allotment of the other two agents. Thus, there is no path in the envy-free graph at the allocation from agent 1 to an agent whose utility is minimal.
When preferences are quasi-linear, maxmin utility envy-free allocations, for the standardized representation of utility \( v_a^i - r_a \), are invariant to uniform changes in rent. That is, starting from a maxmin utility envy-free allocation for rent \( m \) one can obtain a maxmin utility envy-free allocation for a different rent \( m' \) by simply adding \((m' - m)/n \) to the rent of each agent.\(^6\) This is not true anymore with budget-constrained quasi-linear preferences even when the translation preserves the same regime of marginal disutility of rent for all agents. For instance, suppose that \( m' = 3000 \). The allocation with assignment \( \sigma \) and in which each agent pays 1000 is envy-free. Moreover, it is a maxmin utility envy-free allocation, for each agent’s utility is \(-600\). If we reduce the rent of each agent by 400, each agent’s payment is still weakly above the budget. Thus, we maintain allotments in the same regime for which the preference coincides with a quasi-linear preference. Thus, the new allocation is again envy-free. However, as shown just before, it is not a maxmin utility envy-free allocation. This seems contradictory, because we shifted a maxmin utility envy-free allocation for a quasi-linear preference and the resulting allocation lost the maxmin property. The key is that the maxmin utility envy-free allocation is with respect to a representation of utility whose marginal disutility of rent is greater than one. Maxmin utility envy-free allocations are translation invariant only for a standardized representation of preferences.

Arunachaleswaran et al. (2018) introduced an algorithm that calculates an envy-free allocation when preferences are represented by piece-wise linear functions, a domain containing \( \mathcal{B} \).\(^7\) When preferences are in \( \mathcal{B} \) with budget violation indices out of a set with cardinality \( k \), their algorithm runs in \( O(n^{k+c}) \) for some \( c > 0 \) (Arunachaleswaran et al. 2018, Sec. 4.1). Their algorithm does not produce an allocation satisfying further criteria.

Thus, the algorithms in Theorem 1 significantly improve over both Gal et al. (2017) and Arunachaleswaran et al. (2018)’s algorithms because they apply to a non-linear domain of preferences and produce allocations in specific selections of the envy-free set. We do pay a price in the generality of our result compared to Gal et al. (2017), for their algorithm applies to arbitrary linear transformations of the vector of utilities. For instance, their algorithm can calculate an allocation that maximizes the summation of utilities for a particular group of agents among all envy-free allocations. We limit our scope to the narrower set of maxmin or minmax individual utility or individual rent selections of the envy-free set. Arguably there is no significant loss of our approach, for we still cover the selections that are actually used in practice. At a technical level, our restriction is necessary to leverage a topological property of these selections, rent monotonicity, that was never taken advantage of.

\(^6\) This is a straightforward consequence of the characterization of maxmin utility envy-free allocations by means of the envy-free graph.

\(^7\) A piece-wise linear utility function has the form \((r_a, a) \mapsto v_{i,s} - λ_{i,s} r_a \) for a collection of consecutive intervals \( \{I_{i,s}\}_{a \in A, s \in S} \) that covers \( \mathbb{R} \). Even though we will state all our results for our domain \( \mathcal{B} \), they all generalize for the piece-wise linear domain when we require for each \( i \), \( |S'| \) is polynomial in \( n \) and marginal disutility of paying rent comes from a given fixed set.
before from a computational perspective. Showing that this can be done constitutes in itself a significant advance in our understanding of non-linear problems.

At a high level, we borrow from Gal et al. (2017) the representation of our problems as linear programs, and from Arunachaleswaran et al. (2018) the strategy of first solving the problem for a rent high enough in which budget constraints are all violated and then proceeding recursively rebating amounts of rent. Our algorithm significantly differs in that we are able to maintain control of the location of our iterations with respect to the solution we want to calculate. Rent monotonicity requires that, as rent increases, one recommends allocations in which the rent of each room is higher and the welfare of each agent is lower (Velez 2017). Because of this, in the \( n \)-dimensional utility and rent spaces one can see each of these selections as a strictly monotone path that is parameterized by the aggregate rent to collect. For instance, the utility of an agent and the rent of each room in each element of \( \mathcal{R}(N, A, u, m) \) is the same for all the members of this set and is a strictly monotone function of \( m \) (see Velez 2018). Our approach is to rebate rent and stay on this path. Due to the piece-wise linearity of our domain, our algorithm may temporarily deviate from this objective. However, when this happens, we are able to increase the rent to collect and return to it without losing significant progress towards our goal. Remarkably, this non-monotone approach does stop in polynomial time.

To provide the reader clarity of the significance of our contribution, we proceed in two steps. First, we describe Arunachaleswaran et al. (2018)’s algorithm. Then, we introduce our algorithms and analysis.

### 3.2 Arunachaleswaran et al. (2018)’s algorithm

The essential step of the algorithms that we will discuss and construct is the following. Starting from an envy-free allocation, decrease the rent of each room and reshuffle rooms so no-envy is preserved. We first state a result guaranteeing this is always possible for a small enough rebate.

We start by introducing notation. The profile of preferences for which we find an envy-free allocation, \( u \in B^N \), and the coefficients associated with these preferences \( (v^i_a)_{a \in A, i \in N} \), \( (b_i)_{i \in N} \), and \( (\rho_i)_{i \in N} \) are fixed in our analysis. Now, suppose that the vector of rents at the initial allocation is \( r := (r_a)_{a \in A} \). Since we intend to rebate a small amount of money, the relevant data from the agents’ preferences are the values and marginal disutility of money to the left of the bundles \( (r_a, a) \) with \( a \in A \). We first identify which agents would pay more than their budget for each \( (r_a, a) \).

**Definition 1** For each \( r \in \mathbb{R}^A \), let \( SB(r) := \{(i, a) \in N \times A : r_a > b_i\} \) and \( B(r) := \{(i, a) \in N \times A : r_a \geq b_i\} \).

---

8 Rent monotonicity is satisfied only by selections that can be written as maxmin or minmax operators of utility or rent (Velez 2017).
For each \((i, a) \in SB(r)\), agent \(i\) pays more than her budget when receiving room \(a\) and paying \(r_a\) for it. Thus, just to the left of this bundle, the agent’s value for room \(a\) is \(v_i + \rho_i b_i\) and her marginal disutility of money is \((1 + \rho_i)\). If \((i, a) \notin SB(r)\), to the left of \((r_a, a)\), agent \(i\)’s value for room \(a\) is \(v_i\) and her marginal disutility of money is 1.

**Definition 2** For each \(r \in \mathbb{R}^A\), let

\[
v_{ia}(r) := \begin{cases} v_i^a + \rho_i b_i & \text{if } (i, a) \in SB(r), \\ v_i^a & \text{otw.} \end{cases}
\]

\[
\lambda_{ia}(r) := \begin{cases} 1 + \rho_i & \text{if } (i, a) \in SB(r), \\ 1 & \text{otw.} \end{cases}
\]

Now, suppose that we start from an envy-free allocation \((r, \sigma)\). Since we intend to reduce the rent of each room and preserve no-envy, each agent will be strictly better off (Lemma 3, Alkan et al. 1991). If our rebate is small enough, each agent needs to receive a room in the best bundles at the starting allocation, for otherwise no-envy would be violated. Thus, we can restrict our attention to the assignments \(\mu\) that make \((r, \mu)\) envy-free. Suppose then that we reshuffle rooms so the new assignment is \(\mu\) and reduce the price of each room. Let \((\epsilon_a)_{a \in A}\) be these price reductions. If we maintain no-envy, it must be the case that for each other assignment \(\gamma\) that makes \((r, \gamma)\) envy-free, for each \(i \in N\), \(\lambda_{ij}(r)\epsilon_{j}(r) \geq \lambda_{ij}(r)\epsilon_{j}(r)\). Thus,

\[
\sum_{i \in N} \log \lambda_{ij}(r) \geq \sum_{i \in N} \log \lambda_{ij}(r).
\]

Thus, to rebate a small amount of rent and preserve no-envy we need to reshuffle rooms so the summation of the logarithms of the marginal disutility of money is maximized among the assignments that preserve no-envy in our starting allocation. It turns out that the converse is also true. If we reshuffle rooms so the summation of the logarithms of the marginal disutility of money is maximized among the assignments that preserve no-envy in our starting allocation, we can always find positive envy-free rebates that preserve no-envy. We state this result next in an equivalent form introduced by Arunachaleswaran et al. (2018).

**Definition 3** For \(u \in B^N\) and \((r, \sigma) \in F(N, A, u, m)\), let \(F(r) := (N, A, E)\) be the bipartite graph were \((i, a) \in E\) if \(u_i(r_{\sigma(i)}, \sigma(i)) = u_i(r_a, a)\); and \(F^w(r)\) the weighted version of \(F(r)\) were for each \((i, a) \in E\), \(w(i, a) := \log \lambda_{ia}(r)\).

**Lemma 1** (Perturbation Lemma; Arunachaleswaran et al. 2018) Let \((r, \sigma) \in F(N, A, u, m)\). Suppose that \(u \in B^N\) and \(\mu\) is a maximum weight
perfect matching in \( F^w(r) \). Then, there is \( \epsilon > 0 \) such that for each \( \delta \in [0, \epsilon] \), there is \((r^\delta, \mu) \in F(N,A,u,m-\delta)\) such that for each \( a \in A \), \( r^\delta_a < r_a \).\(^{10}\)

Arunachaleswaran et al. (2018) leverage Lemma 1 to construct a polynomial algorithm to find an envy-free allocation in a piece-wise linear economy in which the slopes in the different intervals come from a finite set of cardinality \( k \). The essential step in this task is the following. Given \( \eta > 0 \), starting from an envy-free allocation \((r, \sigma) \in F(N,A,u,m)\) with \( u \in \mathcal{B}^N \), rebate \( \eta > 0 \) so no-envy and budget violations are preserved. For arbitrary continuous preferences, Lemma 1 itself does not solve this problem, i.e., stacking the \( \epsilon \)s produced by the lemma, may not allow one to reach a rebate of \( \eta \) (Alkan 1989). Arunachaleswaran et al. (2018)’s breakthrough is to realize that this can be done by concatenating the solution of the following linear program for a maximal weight perfect matching in \( F^w(r) \), \( \mu \):

\[
\begin{align*}
\min_{t \in \mathbb{R}^A} & \sum_{a \in A} t_a \\
\text{s.t.} & : t_a \leq r_a \\
& \quad v_{i\mu(i)}(r) - \lambda_{i\mu(i)}(r)t_{\mu(i)} \geq v_{i\mu(j)}(r) - \lambda_{i\mu(j)}(r)t_{\mu(j)} \quad \forall (i, j) \subseteq N & (2) \\
& \quad t_a \geq b_i \quad \forall a \in A \\
& \quad \sum_{a \in A} t_a \geq \sum_{a \in A} r_a - \eta.
\end{align*}
\]

This problem is feasible because \( r \) is in the option set. Thus, let \((t, \sigma)\) be the allocation associated with its solution. Since \( t \) satisfies the second set of constraints in (2), the allocation is in \( F(N,A,u,m-\epsilon) \) for some \( 0 < \epsilon \leq \eta \). Thus, one of the following is true: (i) the solution is in \( F(N,A,u,m-\eta) \); (ii) the solution hit one of the \( SB \) constraints; or (iii) \( \sigma \) is not a maximal weight perfect matching in \( F^w(s) \) for any \( s \in \mathbb{R}^A \) is bounded above by \((n + 1)^{k-1}\).

Thus, for \( k \) constant, one can construct a polynomial algorithm that stops in \( O(n^{k+1} \zeta) \), where \( \zeta \) is the number of intervals in the piece-wise linear representation of preferences. In our case, with preferences in \( \mathcal{B} \), \( \zeta \) is bounded above by \( 2n^2 \). Thus, the algorithm runs in \( O(n^{k+c}) \).

### 3.3 Directed search within the envy-free set

A solution to (2) leads us to an envy-free allocation, but it does not allow us to control its position within the envy-free set. We would like to be able to direct the algorithm to optimize some further criteria. The following algorithms calculate in polynomial time a maxmin envy-free allocation for an economy \((N, A, u, m)\) with

\(^{10}\) Lemma 1 is an improved version of the Perturbation Lemma of Alkan et al. (1991), which states existence of an envy-free rebate without identifying the assignments that obtain them. Our Lemma 2, which we prove in detail in the Appendix, implies Lemma 1.
preferences in \( B \) for a fixed number of admissible budget violation indices, \( k \). The first, Algorithm 1, initializes our search by looking for an allocation for a rent \( M \) that is high enough so we make sure that all budget constraints will be violated for each agent, for each consumption of the agent or the other agents, at each possible envy-free allocation for \( (N, A, u, M) \). For such a high rent preferences are quasi-linear in the range of the consumption space that contains all envy-free allocations for \( (N, A, u, M) \) (see Example 1). Thus, one can find a maxmin utility envy-free allocation by essentially using Gal et al. (2017)’s algorithm for this quasi-linear preference (lines 3–4).

Algorithm 1 Initializes search of maxmin utility envy-free allocation.

\[
\begin{align*}
\text{Input:} & \quad (N, A, u, m) \quad \text{were} \quad u \in B^N \quad \text{is associated with} \quad (v_{ia})_{i \in N, a \in A}, \quad (b_i)_{i \in N}, \quad \text{and} \quad (\rho_i)_{i \in N} \in \{\rho_1, \ldots, \rho_k\}^N. \\
\text{Output:} & \quad M \geq m \quad \text{and an allocation in} \quad R(N, A, u, M). \\
1: & \quad \text{For each} \quad i \in N \quad \text{and each} \quad a \in A, \quad \text{let} \quad V_{ia} := (v_{ia} + \rho_i b_i)/(1 + \rho_i) \quad \text{and} \quad \tilde{u}_i \text{the function} \\
& \quad (x_a, a) \rightarrow \tilde{u}_i(x_a, a) := (1 + \rho_i)(V_{ia} - x_a). \\
2: & \quad \text{Let} \quad m' := n \left( \max_{i \in N, \{a, b\} \subseteq A} (V_{ib} - V_{ia}) + \max_{j \in N} b_j \right). \\
3: & \quad \text{Let} \quad \sigma \quad \text{be an assignment that maximizes} \quad \sum_{i \in N} V_{i\sigma(i)}. \\
4: & \quad \text{Compute a price vector} \quad r \in \mathbb{R}^A \quad \text{by solving the linear program} \\
& \quad \max_{R, r \in \mathbb{R}^A} \quad R \\
\text{s.t.:} & \quad R \leq \tilde{u}_i(r_{\sigma(i)}, \sigma(i)) \quad \forall i \in N \\
& \quad V_{i\sigma(i)} - r_{\sigma(i)} \geq V_{i\sigma(j)} - r_{\sigma(j)} \quad \forall \{i, j\} \subseteq N \\
& \quad \sum_{a \in A} r_a = \max \{m, m'\} \\
5: & \quad \text{Returns} \quad M := \sum_{a \in A} r_a \quad \text{and} \quad (r, \sigma).
\end{align*}
\]

Theorem 2 Given input \( (N, A, u, m) \) where \( u \in B^N \) with \( (\rho_i)_{i \in N} \in \{\rho_1, \ldots, \rho_k\}^N \), Algorithm 1 stops in polynomial time and returns \( M \) and \( (r, \sigma) \) such that \( M \geq m \) and \( (r, \sigma) \in R(N, A, u, M) \).

**Proof** Lines 1 and 2 are direct definitions. Line 3 is well known to be polynomial in \( n \). For line 4 we should note that it is indeed a linear program, because all \( \tilde{u} \)s are linear in \( r \) (at this point \( \sigma \) is fixed). It is feasible because given a quasi-linear economy and an assignment \( \sigma \) that maximizes the summation of values, there is always an envy-free allocation for that economy with that assignment (Alkan et al. 1991). Because of the second and third sets of constraints in the program, the feasible set is compact. Thus, the program has a solution. Since the number of constraints in the program is polynomial in \( n \), the complexity of solving this problem is known to be polynomial in \( n \). Let \( M \) and \( (r, \sigma) \) be the output of the algorithm. Clearly, \( M \geq m \).

We claim that \( (r, \sigma) \in R(N, A, u, M) \). Since for each \( i \in N, \quad (1 + \rho_i) > 0 \), profile \( \tilde{u} \) is ordinally equivalent to the quasi-linear profile with values \( V \). Thus, the solution of the linear program in line 4 solves \( \max_{(r, \sigma) \in R(N, A, \tilde{u}, M)} \min_{i \in N} \tilde{u}_i(V_{i\sigma(i)}, \sigma(i)) \).

Let \( (t, \mu) \in R(N, A, \tilde{u}, M) \). Since \( \tilde{u} \) is ordinally equivalent to a quasi-linear preference and \( \{(r, \sigma), (t, \mu)\} \subseteq F(N, A, \tilde{u}, M), \quad (t, \sigma) \in F(N, A, \tilde{u}, M) \) (Svensson 2009). Moreover, by Alkan et al. (Lemma 3, 1991), for each \( i \in N, \)
\[
\tilde{u}_i(t_{\mu(i)}, \mu(i)) = \tilde{u}_i(t_{\sigma(i)}, \sigma(i)). \quad \text{Thus, } \min_{t \in \mathcal{E}} \tilde{u}_i(t_{\sigma(i)}, \sigma(i)) = \min_{t \in \mathcal{E}} \tilde{u}_i(t_{\mu(i)}, \mu(i)). \quad \text{Since } \min_{t \in \mathcal{E}} \tilde{u}_i(t_{\sigma(i)}, \sigma(i)) \geq \min_{t \in \mathcal{E}} \tilde{u}_i(t_{\sigma(i)}, \sigma(i)), \quad \text{then } (r, \sigma) \in \mathcal{R}(N, A, \tilde{u}, M).
\]

We claim that \( F(N, A, \tilde{u}, M) = F(N, A, u, M) \) and for each \((t, \mu) \in F(N, A, u, M), \quad u(t_{\mu(i)}, \mu(i)) = \tilde{u}(t_{\mu(i)}, \mu(i)). \) Suppose that \((t, \mu) \in F(N, A, \tilde{u}, M) \cup F(N, A, u, M). \) Consequently, \( \sum_{a \in A} t_a = M. \) Then, there is \( i \in N \) such that \( t_i \geq (\max_{i \in N, [a, b] \subseteq A} (V_{ib} - V_{ia}) + \max_{j \in N} b_j). \)

Then, \( t_i \geq b_i. \) Thus, \( u(t_i, i, \mu(i)) = \tilde{u}(t_i, \mu(i)) = (1 + \rho_i)(V_{ip(i)} - t_i). \) Now, \( V_{ip(i)} - t_i \leq V_{ip(i)} - \max_{i \in N, [a, b] \subseteq A} (V_{ib} - V_{ia}) - \max_{j \in N} b_j \). Thus, for each \( a \in A, \)
\[ V_{ip(i)} - t_i \leq V_{ip(i)} - (V_{ip(i)} - V_{ia}) - \max_{j \in N} b_j = V_{ia} - \max_{j \in N} b_j. \]

Consequently, for each \( a \in A, \)
\[ u(t_i, i, \mu(i)) = \tilde{u}(t_i, \mu(i)) \leq u_{(\max_{j \in N} b_j, a)} = \tilde{u}_{(\max_{j \in N} b_j, a)}. \]

Thus, for each \( a \in A, t_a \geq \max_{j \in N} b_j, \) for otherwise \( \tilde{u}_i \) and \( u_i \) will envy the agent who receives \( a \) at \((t, \mu). \) Consequently, for each pair \( \{i, j\} \subseteq N, \)
\[ u(t_{\mu(i)}, \mu(j)) = \tilde{u}(t_{\mu(i)}, \mu(j)). \]

Thus, \( F(N, A, u, M) \subseteq F(N, A, \tilde{u}, M) \subseteq F(N, A, u, M) \cap F(N, A, \tilde{u}, M) \) and finally \( F(N, A, u, M) = F(N, A, \tilde{u}, M). \)

Since \( F(N, A, \tilde{u}, M) = F(N, A, u, M), \) for each \((t, \mu) \in F(N, A, u, M) \) and each \( i \in N, \)
\[ u(t_{\mu(i)}, \mu(i)) = \tilde{u}(t_{\mu(i)}, \mu(i)) \] and \( (r, \sigma) \in \mathcal{R}(N, A, \tilde{u}, M), \) we have that \( (r, \sigma) \in \mathcal{R}(N, A, u, M). \)

There is a subtle choice in the construction of Algorithm 1. We first identify a rent that is high enough to guarantee all budget constraints are violated in our original economy for each envy-free allocation. Then, we construct a maxmin utility envy-free allocation for this particular rent. One may be tempted to simply consider the normalized quasi-linear economy that coincides with our economy when budgets are violated, construct a maxmin utility envy-free allocation with respect to the budget-constrained quasi-linear representation for an arbitrary rent, and then “slide” it so all budgets are violated. This approach does not work because, as shown in Example 2, these allocations are not invariant to uniform changes in rent.

If Algorithm 1 returns \( M = m, \) we have actually computed an element of \( \mathcal{R}(N, A, u, m). \) Thus, we need to continue our search only when this algorithm returns \( M > m \) and \( (r, \sigma) \in \mathcal{R}(N, A, u, M). \) Algorithm 2 does so. This algorithm shares some of its philosophy with Arunachaleswaran et al. (2018)’s. At a given state in which an allocation in \( \mathcal{R}(N, A, u, m') \) with \( m' > m \) has been calculated, it reshuffles rooms and rebates rent by solving (3), an LP that maximizes the minimum value of the \( u_is \) constrained by no-envy and budget regime changes.

As in Arunachaleswaran et al. (2018)’s algorithm, solving (3) gets us closer to collecting exactly rent \( m \) with an envy-free allocation. The solution to this problem, \( t^*, \) may be such that \( (t^*, \sigma^*) \notin \mathcal{R}(N, A, u, \sum_{a \in A} t^*_a), \) however. The issue is that the SB constraints of (3) may bind at its solution. The following example illustrates it.

\[ \text{The rebate constraints, i.e.}, t^*_a \leq t^*_a - 1, \text{never bind at a solution to (3). They are included for convenience in the analysis.} \]
Example 3 (Correction step in Algorithm 2 may be necessary) Consider the economy introduced in Examples 1 and 2 with \( m = 1800 \). For simplicity let \( M = 3000 \) and the maxmin utility envy-free allocation calculated in Example 2, \((r^0, \sigma^0)\), where \( r^0_a = r^0_b = r^0_c = 1000 \), \( \sigma^0(1) = a \), \( \sigma^0(2) = b \), and \( \sigma^0(3) = c \) be the input of Algorithm 2.\(^{12}\) We can select \( \sigma^1 = \sigma^0 \). Note that \((t^1, \sigma^1)\), where \( t^1_a = t^1_b = t^1_c = 600 \) is a solution to (3), because to increase the utility of an agent at this allocation the agent needs to pay less than her budget. Thus, this allocation maximizes the minimal utility among agents given the constraints of the problem. As shown in Example 2, \((t^1, \sigma^1) \notin \mathcal{R}(N, A, u, 1800)\).

Thus, since our objective is a maxmin utility envy-free allocation, a solution to (3) may lose the maxmin property. When this is so, i.e., line 11 is reached, we need to correct the situation. We do so by grabbing the value of (3), \( R^s \), and increasing rents again constrained by no-envy and maxmin utility \( R^s \), i.e., by solving (4). Note that we keep the same assignment as in (3) just solved. After the formal statement of the theorem, we discuss the subtle reason why this step returns the algorithm to the maxmin path without compromising our progress towards the solution. Let us see now how this step provides the right correction in our running example.

Example 4 (Correction step in Algorithm 2) Consider the instance of Algorithm 2 in Example 3. After the solution to (3), \((t^1, \sigma^1)\), loses the maxmin property due to the binding SB constraints, the algorithm solves (4). Note that \( R^1 = 0 \) because at \((t^1, \sigma^1)\) the utility of agents 2 and 3 is zero and the utility of agent 1 is 200. Problem (4) maximizes the aggregate rent collected by weakly increasing rent in each room constrained by no-envy and guaranteeing a utility no less than zero to each agent. The solution to this problem is \( r^1_b = r^1_c = 600 \) and \( r^1_a = 700 \). The allocation \((r^1, \sigma^1)\) is envy-free for \((N, A, u, 1900)\). Since each agent’s utility at the allocation is zero, we have that \((r^1, \sigma^1) \in \mathcal{R}(N, A, u, 1900)\). Note that some of the SB constraints that were binding at \((t^1, \sigma^1)\) are still binding. In general, a solution to (4), when necessary, returns to the maxmin path while either keeping some of the SB constraints binding or requiring a change of assignment. In our example, in the next iteration of the algorithm, the SB restrictions for rooms b and c are removed for agents 2 and 3 in (3) and thus the solution to this problem produces a maxmin utility envy-free allocation for \( m = 1800 \). The algorithm then stops.

Theorem 3 Given input \((N, A, u, m)\) where \( u \in \mathcal{B}^N \) with \((\rho_i)_{i \in N} \in \{\rho_1, \ldots, \rho_k\}^N\), Algorithm 2 stops in polynomial time and its output belongs to \( \mathcal{R}(N, A, u, m)\).

\(^{12}\) Running Algorithm 1 produces an allocation for \( m' = 2250 \).
Algorithm 2 Calculates a maxmin envy-free allocation.

**Input**: $(N, A, u, m)$, $u \in \mathcal{B}^N$, $b := (b_i)_{i \in N}$, $\rho := (\rho_i)_{i \in N} \in \{\rho_1, \ldots, \rho_k\}^N$, $M > m$, and $(r, \sigma) \in \mathcal{R}(N, A, u, M)$.

**Output**: an allocation in $\mathcal{R}(N, A, u, m)$.

1: Initialize $s \leftarrow 0$
2: Let $(r^s, \sigma^s) := (r, \sigma)$ and $R^s := \min_{i \in N} u_i(r_{\sigma(i)}, \sigma(i))$
3: while $\sum_{a \in A} t^s_a > m$ do
4: \hspace{1em} Update $s \leftarrow s + 1$
5: \hspace{1em} For each $i \in N$ and $a \in A$, let $\hat{u}^s_i(t_a, a) := \nu_{ia}(r^{s-1}) - \lambda_{ia}(r^{s-1})t_a$
6: \hspace{1em} Let $\sigma^s$ be a maximum weight perfect matching in $\mathcal{F}^s(r^{s-1})$
7: \hspace{1em} Let $t^s \in \mathbb{R}^A$ and $R^s$ be solution and value of the following LP
8: \hspace{2em} if $(t^s, \sigma^s) \in \mathcal{R}(N, A, u, \sum_{a \in A} t^s_a)$ then
9: \hspace{3em} $r^s \leftarrow t^s$
10: \hspace{2em} else
11: \hspace{3em} Let $r^s$ be the solution to the following LP
12: \hspace{4em} end if
13: \hspace{2em} end while
14: Return $(r^s, \sigma^s)$.

The first step in the proof of the theorem is to realize that the Perturbation Lemma can be strengthened to guarantee that perturbations can preserve the maxmin utility property.

**Lemma 2** (Maxmin Perturbation Lemma) Let $(r, \sigma) \in \mathcal{R}(N, A, u, m)$ such that $u \in \mathcal{B}^N$ and $\mu$ a maximum weight perfect matching in $\mathcal{F}^r(r)$. Then, there is $\varepsilon > 0$ and a continuous function $\delta \in [0, \varepsilon] \mapsto (r^\delta, \mu) \in \mathcal{R}(N, A, u, m - \delta)$ such that $(r^0, \mu) = (r, \mu)$; and for each pair $\delta < \eta$, and each $i \in N$, $u_i(r^\mu_{\mu(i)}, \mu(i)) > u_i(r^\delta_{\mu(i)}, \mu(i))$, and for each $a \in A$, $r^\delta_a > r^\eta_a$.

We present the proof of our lemmas in the Appendix.

Lemma 2 guarantees that the solution to (3) will decrease the aggregate rent we are collecting. Now, suppose that we are at some iteration of Algorithm 2 for $s > 0$ in which we find that $(r^s, \sigma^s) \notin \mathcal{R}(N, A, u, \sum_{a \in A} t^s_a)$ (see Example 3). Intuitively, the
algorithm deviated from the monotone path (in utility space and rent space) that is determined by the maxmin utility envy-free solution. We argue now that by solving (4) we are able to return to this path without significant loss of progress.

Suppose then that \( R^s \), the value of problem (3), is the minimum utility at \((r^s, \sigma^s)\), an allocation that is not in the maxmin path. Let \( R^s \) be the maximal minimum utility across agents, bounded above by \( R^s \), at a maximal utility envy-free allocation for assignment \( \sigma^s \). If we want to keep our solution in the maxmin path, our best chance with assignment \( \sigma^s \) is to return to an allocation with maxmin value \( R^s \).

It turns out that it has to be the case that \( R^s = R^s \). Let \((r^s, \sigma^s)\) be an allocation associated with \( R^s \) (in the set that defines \( R^s \)). If \( R^s < R^s \), one can show that \((r^s, \sigma^s)\) is obtained by rebating money in each room at \((r^s, \sigma^s)\). Intuitively, this reveals that there was still room to rebate money at \((r^s, \sigma^s)\) without changing the assignment. Indeed, the following key lemma states that if \((r^s, \sigma^s)\) is obtained by rebating money in each room at \((r^s, \sigma^s)\), \( \sigma^s \) must be a maximal weight perfect matching in \( F^w(r^s) \).

**Lemma 3** (Converse perturbation lemma) Let \( u \in B^N \), \( \epsilon > 0 \), \((r, \sigma) \in F(N, A, u, m) \) and \((t, \sigma) \in F(N, A, u, m - \epsilon) \) such that for each \( a \in A \), \( r_a > t_a \). Suppose that \( B^w(r) = B^w(t) \). Then, \( \sigma \) is a maximal weight perfect matching in \( F^w(r^s) \).

Thus, it is impossible that \( R^s < R^s \), for otherwise by Lemma 2, \( R^s \) would not be maximal. Consequently, in order to recover a maxmin allocation with value \( R^s \), we can keep assignment \( \sigma^s \) and increase rent from \( r^s \), i.e., we can solve (4). Again by Lemma 2, either a budget constraint was just released at \((r^s, \sigma^s)\), or \( \sigma^s \) is not a maximal weight perfect matching in \( F^w(r^s) \). Thus, if by the time all budget constraints are released, which must happen in polynomial time, the algorithm has not stopped, it will do so after solving (3) only once more.

**Proof of Theorem 3** Since \((r^0, \sigma^0)\) is the input, \((r^0, \sigma^0) \in R(N, A, u, \sum_{a \in A} r^0_a)\) and \( R^0 = \min_{r \in \mathbb{N}} u_i((r^0, \sigma^0(i))\).

We will suppose now that for \( s \geq 1 \), \((r^{s-1}, \sigma^{s-1}) \in R(N, A, u, \sum_{a \in A} r^{s-1}_a),\)

\( R^{s-1} = \min_{r \in \mathbb{N}} u_i((r^{s-1}_a) \in \sigma^{s-1}(i)),\) and \( \sum_{a \in A} r^{s-1}_a > m \) if \( \sum_{a \in A} r^{s-1}_a = m \) the algorithm would have stopped in step \( s - 1 \). We will prove that the processes in each line inside the while loop are well defined and can be individually completed in polynomial time. We also prove that \((r^s, \sigma^s) \in R(N, A, u, \sum_{a \in A} r^s_a),\)

\( R^s = \min_{r \in \mathbb{N}} u_i((r^s_a), \sigma^s(i)),\) and \( r^s \) is a solution to (3).

Since \((r^{s-1}, \sigma^s) \in F(N, A, u, \sum_{a \in A} r^{s-1}_a), \) \( r^{s-1} \) is in the feasible set of (3). Because of the first and last constraints, this set is compact. Thus, it has a solution \( r^*, R^* \). Moreover, since \((r^{s-1}, \sigma^s) \in R(N, A, u, \sum_{a \in A} r^{s-1}_a)\) and \( \sigma^s \) is a maximal weight perfect matching in \( F^w(r^{s-1}) \) and none of the last two sets of constraints in (3) bind, by Lemma 2, \( R^{s-1} < R^* \).

---

13 These two statements hold because \( u \) and \( \bar{u} \) coincide on a left interval at \( r^{s-1} \). That is, there is \( \epsilon > 0 \) such that for each \( r \in \mathbb{R}^s \) such that for each \( a \in A, t_a \in [r^{s-1} - \epsilon, r^{s-1}], u_i(t_a, a) = \bar{u}_i(t_a, a). \)
Now, since (3) is a linear program with a polynomial number of constraints, it can be computed in polynomial time. The solution of (3), \( t^*, R^* \), is such that \((t^*, \sigma^*) \in F(N, A, \tilde{u}^s, \sum_{a \in A} t_a^s) \) and \( R^s = \min_{i \in N} \tilde{u}_i^s(t_{\sigma(i)}^s, \sigma^*(i)) \). By the rebate and SB constraints of (3), for each \((i, a) \in N \times A, u_i(t_{\sigma(i)}^s, a) = \tilde{u}_i^s(t_a^s, a) \). Since \((t^*, \sigma^*) \in F(N, A, \tilde{u}^s, \sum_{a \in A} t_a^s) \) and \( R^s = \min_{i \in N} \tilde{u}_i^s(t_{\sigma(i)}^s, \sigma^*(i)) \), we have that \((t^*, \sigma^*) \in F(N, A, u, \sum_{a \in A} t_a^s) \) and \( R^s = \min_{i \in N} u_i(t_{\sigma(i)}^s, \sigma^*(i)) \).

Line 9 can be completed in polynomial time, i.e., given \( t^* \in \mathbb{R}^A \) and \( \sigma^* : N \rightarrow A \) a bijection, \((t^*, \sigma^*) \in R(N, A, u, \sum_{a \in A} t_a^s) \) is verifiable in polynomial time (Proposition 5.9, Velez (2018) (see also Example 2.2)).

Because of the first and second constraints in (4), the feasible set in this program is compact. Since \((t^*, \sigma^*) \in F(N, A, \tilde{u}^s, \sum_{a \in A} t_a^s) \) and \( R^s = \min_{i \in N} \tilde{u}_i^s(t_{\sigma(i)}^s, \sigma^*(i)) \), \( t^* \) belongs to its feasible set. Thus, (4) has a solution, \( t^* \).

We claim that \((t^*, \sigma^*) \in R(N, A, u, \sum_{a \in A} t_a^s), R^s = \min_{i \in N} u_i(t_{\sigma(i)}^s, \sigma^*(i)) \), and \((t^*, R^s) \) is a solution to (3). This is obviously so when \((t^*, \sigma^*) \in R(N, A, u, \sum_{a \in A} t_a^s) \) and thus \( t^* = t^s \).

Suppose then that \((t^*, \sigma^*) \notin R(N, A, u, \sum_{a \in A} t_a^s) \). Thus, \( t^* \) is a solution to (4). Let \( R^* \) be the maximum of

\[
\left\{ t^s \leq R^*: \exists (t^*, \sigma^*) \in R(N, A, u, \sum_{a \in A} t_a), R = \min_{i \in N} u_i(t_{\sigma(i)}^s, \sigma^*(i)) \right\}.
\]

Since \((r_{a}^{s-1}, \sigma^{s-1}) \in R(N, A, u, \sum_{a \in A} r_a^{s-1}) \) and \( \sigma^s \) is a maximal weight perfect matching in \( F^w(r_{a}^{s-1}) \), by Lemma 2, the set above contains \( R^s \). Since preferences are continuous, the set is also closed (this follows from Velez (Proposition 2.2, 2017) and Alkan et al. (Decomposition Lemma, 1991)). Thus, \( R^* \) is well-defined. By Lemma 2, \( R^* > R^{s-1} \). Let \((r^*, \sigma^*) \in R(N, A, u, \sum_{a \in A} r_a^s) \) be such that \( R^* = \min_{i \in N} u_i(r_{\sigma(i)}^s, \sigma^*(i)) \). Since \( R^* > R^{s-1} \) and \((r_{a}^{s-1}, \sigma^{s-1}) \in R(N, A, u, \sum_{a \in A} r_a^{s-1}) \), by Velez (Theorem 1, 2017) and Alkan et al. (Decomposition Lemma, 1991), for each \( a \in A, r_{a}^{s-1} > r_a^s \).

We claim that for each \( a \in A, r_{a}^{s} \geq t_a^s \). Let \( j \in N \). Since \((r^*, \sigma^*) \in R(N, A, u, \sum_{a \in A} r_a^s) \), by Velez (Proposition 5.9, 2018) there is \( i^* \in N \) such that \( u_{i^*}(r_{\sigma(i^*)}^s, \sigma^*(i^*)) = R^* \) and \( j \rightarrow u_{i^*}, \sigma^*, t^* \). Suppose that \( r_{\sigma(i)}^s < r_a^s \). Since \((t^s, \sigma^s) \in F(N, A, u, \sum_{a \in A} t_a^s) \), by Velez (Lemma 5.7. 2018), \( u_{i^*}(t_{\sigma(i^*)}^s, \sigma^*(i^*)) < u_{i^*}(r_{\sigma(i^*)}^s, \sigma^*(i^*)) = R^* \). This contradicts that \( R^* \leq \min_{i \in N} u_i(t_{\sigma(i)}^s, \sigma^*(i)) \).

Since for each \( a \in A, r_{a}^{s-1} > r_{a}^{s} \geq t_a^s \), we have that for each \((i, a) \in N \times A, u_i(r_{\sigma(i)}^s, a) = \tilde{u}_i^s(t_a^s, a) \). Thus, \( \min_{i \in N} \tilde{u}_i^s(t_{\sigma(i)}^s, \sigma^*(i)) = R^s \) and by Velez (Proposition 5.9, 2018), \((r^*, \sigma^*) \in R(N, A, \tilde{u}^s, \sum_{a \in A} r_a^s) \).

We claim that for each \( a \in A, r_{a}^{s} \leq r_a^s \). Let \( j \in N \). Since \((r^*, \sigma^*) \in R(N, A, \tilde{u}^s, \sum_{a \in A} r_a^s) \), by Velez (Proposition 5.9, 2018) there is \( i^* \in N \) such that \( \tilde{u}_i^s(r_{\sigma(i^*)}^s, \sigma^*(i^*)) = R^s \) and \( j \rightarrow \tilde{u}_i^s, r^*, \sigma^*, t^s \). Since \((r^*, \sigma^*) \in F(N, A, \tilde{u}^s, \sum_{a \in A} t_a^s) \), if

\(^{14}\) Even if utility was not linear, this program is compact because \( R^* \) is the value of (3).
\[
\sigma_{ij}^* > \tau_{ij}^* \quad \text{by Velez (Lemma 5.7, 2018)}, \quad \tilde{u}_{ij}^\tau_r((r_{\sigma(i)}, \sigma^r(i))) > \tilde{u}_{ij}^\tau_r((r_{\sigma(i)}, \sigma^r(i))).
\]

Since, \( R^* = \tilde{u}_{ij}^\tau_r((r_{\sigma(i)}, \sigma^r(i))) \) and \( \tilde{u}_{ij}^\tau_r((r_{\sigma(i)}, \sigma^r(i))) \geq R^*, \quad R^* > R^* \). This contradicts \( R^* \geq R^* \).

Thus, for each \( a \in A, \ t_a^\delta \leq r_a^\delta \leq r_a^\tau < r_a^{\tau - 1} \). Thus, for each \( (i, a) \in N \times A, \ u_t(r_a^\delta, a) = \tilde{u}_{ij}^\tau_r(r_a^\delta, a) \). Thus, \( (r^\delta, \sigma^\delta) \in F(N, A, u, \sum_{a \in A} r_a^\delta) \).

We claim that \( R^* = R^* \). Suppose by contradiction that \( R^* < R^* \). We claim that for each \( a \in A, \ r_a^\delta > r_a^\tau \). Let \( j \in N \). Since \( (r^\delta, \sigma^\delta) \in \mathcal{R}(N, A, u, \sum_{a \in A} r_a^\delta) \), by Velez (Proposition 5.9, 2018) there is \( i^\delta \in N \), such that \( u_t(r_{\sigma(i)}, \sigma(i)) = R^* \) and \( j = u_t(r_{\sigma(i)}, \sigma(i)) \). If \( R^* < R^* \), \( u_t(r_{\sigma(i)}, \sigma(i)) \geq R^* > R^* = u_t(r_{\sigma(i)}, \sigma(i)) \). Since \( (r^\delta, \sigma^\delta) \in F(N, A, u, \sum_{a \in A} r_a^\delta) \), by Velez (Lemma 5.7, 2018), \( r_{\sigma(i)}^\delta > r_{\sigma(i)}^\delta \). Because of the first constraints in (3), this program rebates money from \( (r^{\tau - 1}, \sigma^{\tau - 1}) \). Since the program is also constrained by budget regime changes, i.e., the SB constraints, there is no pair \( (i, a) \in N \times A \) such that \( t_a^\delta < b_i < r_a^{\tau - 1} \). Thus, \( B^d(r^\delta) = B^d(r^\delta) \). Thus, by Lemma 3, \( \sigma^\delta \) is a maximal weight perfect matching in \( \mathcal{F}^d(r^\delta) \). By Lemma 2, \( R^* \) is not the maximal element of (5). This is a contradiction. Thus, \( R^* = R^* \).

Since \( R^* = R^* \), \( (r^\delta, \sigma^\delta) \in \mathcal{R}(N, A, \tilde{u}, \sum_{a \in A} r_a^\delta) \), \( \min_{i \in N} \tilde{u}_{ij}^\tau_r(r_{\sigma(i)}^\delta, \sigma^\delta(i)) = R^* \), and for each \( a \in A, r_a^\delta \geq t_a^\delta \), we have that \( r^\delta \) is in the feasible set of (4). Since \( r^\delta \) is a solution to (4) and for each \( a \in A, r^\delta \leq r_a^\delta \), we have that \( r^\delta = r^\delta \).

Thus, \( (r^\delta, \sigma^\delta) \in \mathcal{R}(N, A, u, \sum_{a \in A} r_a^\delta) \), \( R^* = \min_{i \in N} u_t(r_{\sigma(i)}^\delta, \sigma^\delta(i)) \), and for each \( a \in A, t_a^\delta \leq r_a^\delta < r_a^{\tau - 1} \). Thus, \( (r^\delta, R^*) \) is also a solution to (3).

We claim that the algorithm stops. Moreover, if it returns \( (r^\delta, \sigma^\delta) \), \( s \) is bounded by \( n^{k+2} \). Thus, the algorithm runs in \( O(n^{k+c}) \) for some \( c > 2 \).

We claim that at least one of the following three statements should be true for a given \( s \): (i) there is \( (i, a) \in SB(r^{\tau - 1}) \) such that \( r_a^\delta = b_i \); (ii) \( \sigma^\delta \) is not a maximal weight perfect matching in \( \mathcal{F}^d(r^\delta) \); (iii) \( \sum_{a \in A} r_a^\delta = m \). Suppose by contradiction that for each \( (i, a) \in SB(r^{\tau - 1}) \), \( r_a^\delta > b_i \); \( \sigma^\delta \) is a maximal weight perfect matching in \( \mathcal{F}^d(r^\delta) \); and \( \sum_{a \in A} r_a^\delta > m \). By Lemma 2, there is \( \delta > 0 \) and \( t^\delta \in \mathbb{R}^A \) such that \( (r^\delta, \sigma^\delta) \in \mathcal{R}(N, A, u, \sum_{a \in A} r_a^\delta - \delta) \); for each \( a \in A, t_a^\delta < r_a^\delta < r_a^{\tau - 1} \); for each \( i, a \in SB(r^{\tau - 1}) \), \( t_a^\delta > b_i \); \( \sum_{a \in A} r_a^\delta > m \); and for each \( i \in N \), \( u_t(r_{\sigma(i)}^\delta, \sigma^\delta(i)) > u_t(r_{\sigma(i)}^\delta, \sigma^\delta(i)) \). Since for each \( (i, a) \in SB(r^{\tau - 1}) \), \( t_a^\delta > b_i \), we have that \( (r^\delta, \sigma^\delta) \in \mathcal{R}(N, A, \tilde{u}, \sum_{a \in A} r_a^\delta) \) and for each \( i \in N \), \( \tilde{u}_{ij}^\tau_r(r_{\sigma(i)}^\delta, \sigma^\delta(i)) > \tilde{u}_{ij}^\tau_r(r_{\sigma(i)}^\delta, \sigma^\delta(i)) \). Thus, \( \min_{i \in N} \tilde{u}_{ij}^\tau_r(r_{\sigma(i)}^\delta, \sigma^\delta(i)) < \min_{i \in N} \tilde{u}_{ij}^\tau_r(r_{\sigma(i)}^\delta, \sigma^\delta(i)) \). Thus, \( r^\delta \) is not a solution to (3). This is a contradiction.

Thus, for a given \( s \), one of the following cases holds: (i) \( |SB(r^\delta)| < |SB(r^{\tau - 1})| \); (ii) \( |SB(r^\delta)| = |SB(r^{\tau - 1})| \) and the weight of \( \sigma^{\tau + 1} \) in \( \mathcal{F}^d(r^\delta) \) is greater than the weight of \( \sigma^\delta \) in \( \mathcal{F}^d(r^{\tau - 1}) \), which since \( |SB(r^\delta)| = |SB(r^{\tau - 1})| \) is the same as the weight of \( \sigma^\delta \) in \( \mathcal{F}^d(r^{\tau - 1}) \); (iii) the algorithm stops. There are at most \( n^2 \) elements in \( SB(r^\delta) \) and at most \( (n + 1)^{k - 1} \) values for the weight of a perfect matching. Thus, the algorithm either stops or reaches a state in \( SB(r^\delta) = \emptyset \) in \( O(n^{k+c}) \) for some \( c > 0 \). If \( SB(r^\delta) = \emptyset \) it must be the case that \( (r^{\tau - 1} + 1, \sigma^{\tau + 1}) \in \mathcal{R}(N, A, u, m) \). Suppose by contradiction that \( \sum_{a \in A} r_a^{\tau + 1} > m \). Since \( SB(r^\delta) = \emptyset \), all the perfect matchings in \( \mathcal{F}^d(r^{\tau + 1}) \) have the same weight. Thus, \( \sigma^{\tau + 1} \) is a maximal weight perfect matching in \( \mathcal{F}^d(r^{\tau + 1}) \). This contradicts that at least one of (i), (ii), and (iii) in the preceding paragraph

\[\Sigma\]

\[\text{Springer}\]
holds. Thus, if $SB(r^s) = \emptyset$, the algorithm stops in the next iteration of the while loop.

### 3.4 Proof of Theorem 1

Algorithms 1 and 2 can be modified to calculate the allocations for all the selections in Theorem 1. First, consider a non-empty family of positive affine linear transformations $(f_i)_{i \in S}$ for some $S \subseteq N$. One can calculate an allocation in

$$\arg \max_{(r, \sigma) \in F(e)} \min_{i \in N} f_i(u_{\sigma(i)}, \sigma(i))$$

by trivially modifying these algorithms as follows: In the LP in Algorithm 1 replace the maxmin constraints $R \leq \tilde{u}_i(\cdot)$ with $R \leq f_i(\tilde{u}_i(\cdot))$; in Algorithm 2 replace $R \leq f_i(\tilde{u}_i(\cdot))$ in (3), and replace $R^s \leq \tilde{u}_i(\cdot)$ with $R \leq f_i(\tilde{u}_i(\cdot))$ in (4). The analysis of correctness and complexity of the algorithms goes through unmodified. Indeed, the only property of $\mathcal{R}$ that was used in this analysis is its rent monotonicity and that it is defined by means of an LP whose value decreases with an increase in rent.

Similarly, for a non-empty family of positive affine linear transformations $(g_a)_{a \in C}$ for some $C \subseteq A$, one can calculate an element of

$$\arg \min_{(r, \mu) \in F(e)} \max_{a \in C} g_a(r_a)$$

as follows. Replace the LP in Algorithm 1 with

$$\min_{R, r \in \mathbb{R}^A} R$$

s.t.: $R \geq g_a(r_a) \quad \forall a \in A$

$$V_{i \sigma(i)} - r_{\sigma(i)} \geq V_{i \sigma(j)} - r_{\sigma(j)} \quad \forall \{i, j\} \subseteq N$$

$$\sum_{a \in A} r_a = \max\{m, m'\},$$

and replace (3) with

$$\min_{R, r' \in \mathbb{R}^A} R$$

s.t.: $t_a^s \leq r_a^{s-1} \quad \forall a \in A$

$$R \geq g_a(r_a) \quad \forall a \in A$$

$$\tilde{u}_i(t_{\sigma(i)}^s, \sigma^s(i)) \geq \tilde{u}_i(t_{\sigma(j)}^s, \sigma^s(j)) \quad \forall \{i, j\} \subseteq N$$

$$t_{\sigma(i)}^s \geq b_i \quad \forall (i, a) \in SB(r^{s-1})$$

$$\sum_{a \in A} t_{\sigma(i)}^s \geq m$$

and (4) with

$$\max_{r_s \in \mathbb{R}^A} \sum_{a \in A} t_a^s$$

s.t.: $t_a^s \geq r_a^s \quad \forall a \in A$

$$R^s \geq g_a(r_a) \quad \forall a \in A$$

$$\tilde{u}_i(r_{\sigma(i)}^s, \sigma^s(i)) \geq \tilde{u}_i(r_{\sigma(j)}^s, \sigma^s(j)) \quad \forall \{i, j\} \subseteq N$$

$$\sum_{a \in A} t_{\sigma(i)}^s \geq m$$

$$\sum_{a \in A} r_a^s \geq \max\{m, m'\},$$

$$\sum_{a \in A} t_a^s \geq m.$$
The above modifications are rather obvious. It is very useful to emphasize why they do work, however. We will encounter next that the modification of the algorithms for the other two families of solutions in the theorem requires some extra thought.

To fix ideas suppose that \( C = A \) and the \( g_a \)'s are the identity functions. Thus, the modifications above lead to an allocation in which the maximal rent paid by some agent is minimized in the envy-free set, the minmax rent envy-free solution. Clearly, (7) produces a minmax rent envy-free allocation for the quasi-linear preferences that coincide with \( u \) in the range for which all budgets are violated. Thus, it is again a viable seed for Algorithm 2. Now consider (8). This LP is minimizing the individual maximal rent with some constraints for the economies in which rent is at least \( m \). In other words, the problem is trying to rebate money constrained by the limit to rebate up to the point in which aggregate rent is \( m \). This is why its solution inches towards our goal. Similarly, LP (9) is trying to increase the aggregate rent to collect constrained by the maximal rent being \( R^s \). This is why its solution rights a possible overshoot by (8). After realizing these problems are well posed, the replication of our analysis with \( \mathcal{R} \) is perfunctory (if one observes that (6) is rent monotone).

Now consider again a non-trivial family of non-negative affine linear transformations \((f_i)_{i \in N}\). Our objective now is to calculate an element in

\[
\arg \min_{(r,\sigma) \in \mathcal{F}(e)} \max_{i \in N} f_i(u_\sigma(i), \sigma(i)).
\]  

A trivial modification of the LP in Algorithm 1 in which one sets a minimization problem and replaces the maxmin constraints \( R \leq \bar{u}_i(\cdot) \) with the minmax constraints \( R \geq f_i(\bar{u}_i(\cdot)) \) does produce a minmax envy-free allocation for the range in which budget constraints are violated. However, one finds a hurdle if one trivially transforms (3) into a minmax LP. The issue is that this problem is intended to rebate rent. If the objective becomes to minimize the maximal utility, the modified LP does not try to do this, because the maximal utility becomes lower as rent increases.

Thus, in order to find an element of (10) we need to rethink the whole structure of our approach. Instead of initially calculating an allocation for an aggregate rent that is large enough so the budget constraints are violated and then rebate rent, we need to do the opposite: Calculate an allocation for low enough rent so no budget constraint is violated and then increase rent. Modifying and analyzing Algorithm 1 is again perfunctory. Now that we are recursively increasing rent (last constraint in (3) flips to \( \leq \)), a minmax version of (3) and a min version of (4) work in the right direction. There are still two issues we need to address. First, the lemmas that guide our analysis of Algorithm 2 are not useful anymore, for they refer to rebates of rent. Second, we need to revise the choice of the assignment for which these programs must be solved (Line 6 of Algorithm 2). These two issues are related because our choice of this assignment is suggested by Lemma 2.

The key to resolve these issues is to note that if we intend to increase rent instead of rebate it, our choice of assignment should be different. Suppose that the starting vector of prices is \( r \). By continuity, it is true again that if we want to maintain no-envy, we need to restrict to assignments \( \mu \) for which \( (r, \mu) \) is envy-free. Let \( \lambda^+(r) \) be the right slope of the budget-constrained quasi-linear utility for agent \( i \) at \( (r_a, a) \).
Since we are increasing rent, it must be the case that for each other assignment \( \gamma \) that makes \((r, \gamma)\) envy-free, for each \( i \in N \), \( \lambda^+_{i_{\mu(i)}}(r)\varepsilon_{\mu(i)} \leq \lambda^+_{i_{\gamma(i)}}(r)\varepsilon_{\gamma(i)} \). That is, each agent finds her utility loss no greater than that of the other agents. Thus,

\[
\sum_{i \in N} \log \lambda^+_{i_{\mu(i)}}(r) \leq \sum_{i \in N} \log \lambda^+_{i_{\gamma(i)}}(r). \tag{11}
\]

Let \( \mathcal{F}^u(r) \) be the version of \( \mathcal{F}(r^{s-1}) \) weighted by the log of the right slopes \( \lambda^+_{i_{\mu(i)}}(r^{s-1}) \). By (11), if we want to increase rent in each room and maintain no-envy, we need to choose a \textit{minimal} weight perfect matching in \( \mathcal{F}^u(r) \). Thus, we need to modify Algorithm 2 by selecting in Line 6 a minimal weight perfect matching in \( \mathcal{F}^u(r^{s-1}) \).

The analysis of the algorithms can be completed based on the following modifications of Lemmas 2 and 3.

**Lemma 4** (Maxim left perturbation Lemma) Let \((r, \sigma) \in \mathcal{R}(N, A, u, m)\) such that \( u \in B^N \) and \( \mu \) a minimum weight perfect matching in \( \mathcal{F}^u_+(r) \). Then, there is \( \varepsilon > 0 \) and a continuous function \( \delta \in [0, \varepsilon] \mapsto (r^{\delta}, \mu) \in \mathcal{R}(N, A, u, m + \delta) \) such that \((r^0, \mu) = (r, \mu)\); and for each pair \( \delta > \eta \), and each \( i \in N, u_i(r^{\delta}_{\mu(i)}, \mu(i)) > u_i(r^0_{\mu(i)}, \mu(i)) \), and for each \( a \in A, r^0_a > r^\delta_a \).

**Lemma 5** (Converse left perturbation lemma) Let \( u \in B^N \), \( r, \sigma \in F(N, A, u, m) \) and \((t, \sigma) \in F(N, A, u, m - \varepsilon) \) such that for each \( a \in A, r_a > t_a \). Suppose that \( B^u(r) = B^u(t) \). Then, \( \sigma \) is a minimal weight perfect matching in \( \mathcal{F}^u_+(t) \).

We omit the proofs of Lemmas 4 and 5, which can be completed along the lines of those of Lemmas 2 and 3.

Finally, the modification to compute an element of a maxmin rent envy-free allocation (third family in the theorem) is then perfunctory.

## 4 Concluding remarks

We constructed an algorithm to calculate a maxmin utility envy-free allocation, and other related selections from the envy-free set, when preferences are in the budget-constrained quasi-linear domain. Our algorithm is polynomial in the number of agents when the number of possible values for the parameters that determine the marginal disutility of rent is kept constant.

Our work shows that it is possible to allow agents report more accurately their financial difficulties, use the maxmin utility and related criteria of justice to select an envy-free allocation, and maintain the complexity of computation of quasi-linear mechanisms. At a technical level, the novelty of our work consists on finding a

---

15 We also need to change the \( \tilde{\nu}^\prime \) to be defined by the right values and slopes at \( r^{s-1} \).

16 We state the lemmas for the maxmin utility envy-free allocations. The lemmas hold for money-monotone selections of the envy-free set.
way to leverage a topological property of the maxmin utility envy-free allocations, money monotonicity, from a computational perspective.

Besides analyzing their computational complexity, it is also relevant to evaluate the incentive properties of mechanisms. In a companion paper we advance this analysis for envy-free budget-constrained quasi-linear mechanisms (Velez 2019). Interestingly, it is necessary that the mechanism designer bound the marginal disutility of rent that agents can report to guarantee that the envy-free budget-constrained quasi-linear mechanisms have only complete information non-cooperative equilibria that are envy-free with respect to true preferences.

Thus, all together our work shows that whenever the mechanism designer limits the possible values of the marginal disutility of rent to a finite set, the maxmin utility envy-free budget-constrained quasi-linear mechanisms (and all mechanisms in Theorem 1) retain both the complexity and complete information incentive properties of quasi-linear mechanisms.

It is interesting, but beyond the scope of this paper, to implement our proposal in laboratory experiments and field applications. It is an open question to evaluate how budget-constrained quasi-linear envy-free mechanisms perform compared with quasi-linear mechanisms in these environments.

Appendix

We remind the reader that, in our analysis, the profile \( u \) and its associated coefficients \((v_i^a)_{i \in N, a \in A}\), \((b_i)_{i \in N}\), and \((\rho_i)_{i \in N}\) are fixed.

Starting from an allocation \((r, \sigma)\) we will show that there is a way to rebate rent in each room while preserving no-envy and additional properties. As discussed in Sect. 3.2 (see (1)), we need to constrain to assignments \( \mu \) that make \((r, \mu)\) envy-free and maximize the summation of the log of the left slopes of the budget-constrained quasi-linear representation. Our analysis is simpler if for each agent \( i \) we modify her initial preferences by decreasing the marginal disutility of money for the rooms in allotments that are worse than her allotment at \((r, \sigma)\). This allows us to consider all possible assignments when we rebate rent, and transform the problem into a standard envy-free allocation problem.

Definition 4 Let \( r \in \mathbb{R}^A \) for which there is \((r, \sigma) \in F(N, A, u, m) \) and \( \Lambda > 0 \). For each \( i \in N \) and \( a \in A \), let \( \lambda_{ia}(r, \Lambda) := \lambda_{ia}(r) \) if \( u_i(r_{\sigma(i)}, \sigma(i)) = u_i(r, a) \), and \( \lambda_{ia}(r, \Lambda) := \Lambda \) otherwise.\(^{17}\)

Lemma 6 Let \( u \in B^N \) and \((r, \sigma) \in F(N, A, u, m) \). There is \( \Lambda > 0 \), such that for each \( 0 < \Lambda' < \Lambda \), if \( \mu \) is a solution to

\[^{17}\] Note that \( \lambda_{ia}(r, \Lambda) \) is well defined, i.e., it is invariant for the choice of \( \sigma \) as long as \((r, \sigma) \in F(N, A, u, m) \).
\[
\max_{\gamma : N \to A, \gamma \text{ a bijection}} \sum_{i \in N} \log(\lambda_{i\gamma(i)}(r, A')), \n\]
then for each \(i \in N\), \(u_i(r_{\sigma(i)}, \mu(i)) = u_i(r_{\sigma(i)}, \sigma(i))\).

**Proof** Let \(\Lambda > 0\) be such that for each \(a \in A\) such that \(u_i(r_{\sigma(i)}, \sigma(i)) > u_i(r_{\sigma(i)}, \sigma(i))\), \(\log \Lambda + (n - 1) \max_{i \in N, a \in A} \lambda_{i\alpha} < \sum_{i \in N} \lambda_{i\sigma(i)}\).

The following definition generalizes the notion of economy and envy-free allocation for arbitrary utility functions, including functions for which more money is preferable.

**Definition 5** For a list of functions \((\hat{u}_i)_{i \in N}\) where \(\hat{u}_i : \mathbb{R} \times A \to \mathbb{R}\), and \(m \in \mathbb{R}\), an allocation for \((N, A, \hat{u}, m)\) is a pair \((y, \gamma)\) where \(y \in \mathbb{R}^A\), \(\sum_{a \in A} y_a = m\), and \(\gamma : N \to A\) is a bijection. \(F(N, A, \hat{u}, m)\) is the set of envy-free allocations for \((N, A, \hat{u}, m)\), i.e., \((y, \gamma)\) for which for each pair \(i, j \subseteq N\), \(\hat{u}_i(y_{\gamma(i)}, \gamma(i)) \geq \hat{u}_i(y_{\gamma(j)}, r(j))\).

**Proof of Lemma 2** Consider a rebate and reshuffle at \((r, \sigma)\), i.e., a vector \(x := (x_a)_{a \in A} \in \mathbb{R}_{++}^A\) and an assignment \(\mu : N \to A\) that induce allocation \((r - x, \mu)\). Since \(r\) will not change in the course of this proof, for simplicity in the notation, we have that \(\lambda_{i\alpha} := \lambda_{i\alpha}(r)\) and \(\hat{\lambda}_{i\alpha} := \hat{\lambda}_{i\alpha}(r, A)\) for some \(\Lambda\) satisfying the property in Lemma 6.

For each \(i\) let \(\hat{u}_i\) be the function \((y_a, a) \in \mathbb{R} \times A \mapsto \hat{u}_i(y_a, a) := \log \hat{\lambda}_{ia} + y_a\). Let \(\varepsilon > 0\) be such that \((i)\) for each \(i \in N\) and each \(a \in A\) such that \(u_i(r_{\sigma(i)}, \sigma(i)) > u_i(r_{\sigma(i)}, \sigma(i))\), we have that \(u_i(r_{\sigma(i)}, \sigma(i)) > u_i(r_a - 2\varepsilon, a)\); and (ii) for each \((i, a) \in SB(r), r_a - \varepsilon > b_i\). Thus, \(\hat{u}_i\) exits because preferences \(u\) are continuous.

Fix \(\delta \in (0, \varepsilon]\).

**Step 1:** Let \((y^\delta, \gamma) \in F(N, A, \hat{u}, \sum_{a \in A} y^\delta_a)\) be such that \(\sum_{a \in A} \exp(y^\delta_a) = \delta\). Let \(r^\delta := (r_a - \exp(y^\delta_a))_{a \in A}\). We claim that \((r^\delta, \gamma) \in F(N, A, u, m - \delta)\).

Since \(\sum_{a \in A} \exp(y^\delta_a) = \delta\) and \(\sum_{a \in A} r_a = m\), then \(\sum_{a \in A} r_a^\delta = m - \delta\). Since \(\sum_{a \in A} \exp(y^\delta_a) = \delta\), for each \(a \in A\), \(\exp(y^\delta_a) < \delta < \varepsilon\). Since \(\hat{u}\) is quasi-linear and \(\gamma\) admits an envy-free allocation for an economy with preferences \(\hat{u}\), \(\gamma\) maximizes the summation of values for \(\hat{u}\) (Svensson 1983). By Lemma 6, \(\gamma\) is a perfect matching in \(F(r)\). Thus, \(u_i(r_{\gamma(i)}, \gamma(i)) = u_i(r_{\sigma(i)}, \sigma(i))\). Thus, for each \(a \in A\) such that \(u_i(r_{\sigma(i)}, \sigma(i)) > u_i(r_{\sigma(i)}, \sigma(i))\), we have that \(u_i(r_{\sigma(i)}, \sigma(i)) > u_i(r_a, a)\). Thus, \(u_i(r_{\gamma(i)} - \exp(y^\delta_{\gamma(i)}), \gamma(i)) > u_i(r_a - \delta, a)\). Let \(a \in A\) be such that \(u_i(r_{\sigma(i)}, \sigma(i)) > u_i(r_a, a)\). Since both \((r_{\gamma(i)}, \gamma(i))\) and \((r_a, a)\) maximize \(u_i\) among the bundles in \((r, \sigma)\), we have that \(\hat{\lambda}_{i\gamma(i)} = \hat{\lambda}_{i\gamma(i)}\) and \(\hat{\lambda}_{ia} = \hat{\lambda}_{ia}\). Since \(u_i(r_{\gamma(i)} - \exp(y^\delta_{\gamma(i)}), \gamma(i)) \geq u_i(r_a - \gamma^\delta_a, a)\) if and only if \(\hat{\lambda}_{i\gamma(i)} \exp(y^\delta_{\gamma(i)}) \geq \hat{\lambda}_{ia} \exp(y^\delta_{\gamma(i)})\). This happens if and only if \(\log \hat{\lambda}_{i\gamma(i)} + y^\delta_{\gamma(i)} \geq \log \hat{\lambda}_{ia} + y^\delta_{\gamma(i)}\).

Now, \(\log \hat{\lambda}_{i\gamma(i)} + y^\delta_{\gamma(i)} \geq \log \hat{\lambda}_{ia} + y^\delta_{\gamma(i)}\) holds because \((y^\delta, \gamma) \in F(N, A, \hat{u}, \sum_{a \in A} y^\delta_a)\). Thus, \(u_i(r_{\gamma(i)} - \exp(y^\delta_{\gamma(i)}), \gamma(i)) \geq u_i(r_a - \exp(y^\delta_{\gamma(i)}), a)\). Thus, for each pair \(i, j \subseteq N\), \(u_i(r^\delta_{\gamma(i)}, \gamma(i)) \geq u_i(r^\delta_{\gamma(j)}, \gamma(j))\). Thus, \((r^\delta, \gamma) \in F(N, A, u, m - \delta)\).
Step 2: Let \((r^\delta, \gamma)\) be a solution to
\[
\max_{(r^\delta, \gamma) \in F(N^\delta, \mu, u, m - \delta)} \min_{i \in N} u_i(r^\delta, \gamma(i)).
\] (12)
We claim that for each maximal weight perfect matching in \(F^\delta(r), \mu, (r^\delta, \mu)\) is a solution to (12). Since \((r, \sigma) \in \mathcal{R}(N, A, u, m)\), for each \(a \in A\), \(r^\delta_a < r_a\) (Alkan et al. 1991; Velez 2017). Thus, by Lemma 6, \(\Lambda\) in our definition of \(\hat{\delta}\) can be taken arbitrarily close to zero. Thus, we can suppose without loss of generality that for each \(a \in A\) such that \(u_i(r_{\sigma(i)}, \sigma(i)) > u_i(r_a, a)\) and each \(b\) such that \(u_i(r_{\sigma(i)}, \sigma(i)) = u_i(r_b, b)\),
\[
\log \hat{\lambda}_{ia} + \log(r_b - r^\delta_b) > \log \hat{\lambda}_{ia} + \log(r_a - r^\delta_a).
\] (13)
For each \(a \in A\), let \(y^\delta_a := \log(r_a - r^\delta_a)\). We claim that \((y^\delta, \gamma) \in F(N, A, \hat{u}, \sum_{a \in A} y_a)\). Let \(i \in N\) and \(a \in A\) such that \(u_i(r_{\sigma(i)}, \sigma(i)) = u_i(r_a, a)\). Thus,
\[
u_i(r^\delta, \gamma(i)) \geq u_i(r^\delta, \gamma(i)) > u_i(r_a - \delta, a) > u_i(r_a, a),
\] (14)
where the first inequality holds because \((r^\delta, \gamma) \in F(N, A, u, m - \delta)\); the second and last hold because for each \(b \in A\), \(r^\delta_b < r_b\), the third holds by the definition of \(\epsilon\); and the fourth holds because \(\delta < \epsilon\). By (14), \(\gamma(i) \neq a\). Since \((r^\delta, \gamma) \in F(N, A, u, m)\), \(u_i(r_{\sigma(i)}(\gamma(i)) \geq u_i(r_{\gamma(i)}(\gamma(i)))\). Thus, \(u_i(r_{\sigma(i)}(\gamma(i)) = u_i(r_{\gamma(i)}(\gamma(i)))\). Thus, for each \(i \in N\), \(\hat{\lambda}_{i\gamma(i)} = \lambda_{i\gamma(i)}\). Thus, by (13) for each \(a \in A\) such that \(u_i(r_{\sigma(i)}(\gamma(i)) = u_i(r_a, a)\),
\[
u_i(y^\delta, \gamma(i)) = \log \lambda_{i\gamma(i)} + \gamma_{\gamma(i)} > \log \hat{\lambda}_{ia} + \gamma_{\gamma(i)} = \hat{\nu}_i(y^\delta_a, a).
\]
Let \(a \in A\) be such that \(u_i(r_{\sigma(i)}(\gamma(i)) = u_i(r_a, a)\). Then, \(\hat{\lambda}_{ia} = \lambda_{ia}\). Recall that \(u_i(r_{\sigma(i)}(\gamma(i)) = u_i(r_{\gamma(i)}(\gamma(i)))\). Since \((r^\delta, \gamma) \in F(N, A, u, m - \delta)\), \(u_i(r^\delta, \gamma(i)) \geq u_i(r_a, a)\). Thus,
\[
u_i(r^\delta, \gamma(i)) - u_i(r_{\gamma(i)}(\gamma(i)) \geq u_i(r_a, a) - u_i(r_a, a)\]
Thus, \(\hat{\lambda}_{i\gamma(i)}(r^\delta) = \lambda_{i\gamma(i)}(r_a - r^\delta_a)\). That is,
\[
u_i(y^\delta, \gamma(i)) = \log \lambda_{i\gamma(i)} + \gamma_{\gamma(i)} \geq \log \lambda_{ia} + \gamma_{\gamma(i)} = \hat{\nu}_i(y^\delta_a, a)\.
\]
Since \((r, \sigma) \in F(N, A, u, m)\), for each \(a \in A\), \(u_i(r_{\sigma(i)}(\gamma(i)) \geq u_i(r_a, a)\). Thus, for each pair \(\{i, j\} \subseteq N\), \(\hat{\nu}_i(y^\delta, \gamma(i)) \geq \hat{\nu}_i(y^\delta, \gamma(j))\). Thus, \((y^\delta, \gamma) \in F(N, A, \hat{u}, \sum_{a \in A} y_a)\). By our choice of \(\Lambda\) and since \(\mu\) is a maximal weight perfect matching in \(F^\delta(r)\), \(\mu\) maximizes the summation of values for quasi-linear preference \(\hat{u}\). Thus, there is an allocation in \(F(N, A, \hat{u}, \sum_{a \in A} y_a)\) with assignment \(\hat{\mu}\) (Alkan et al. 1991). Thus, \((y^\delta, \mu) \in F(N, A, \hat{u}, \sum_{a \in A} y_a)\) (Svensson 2009). Since \(\sum_{a \in A} \exp(y_a) = \delta\), by Step 1, \((r^\delta, \mu) \in F(N, A, u, m - \delta)\). By Alkan et al. (Lemma 3, 1991), for each \(i \in N\), \(u_i(r_{\mu(i)}(\gamma(i)) = u_i(r^\delta, \gamma(i))\). Thus, \(\min_{i \in N} u_i(r_{\mu(i)}(\gamma(i)) = \min_{i \in N} u_i(r^\delta, \gamma(i))\). Thus, \((r^\delta, \mu)\) is a solution to (12).

Step 3: Concludes. Let \(\mu\) be a maximal weight perfect matching in \(F^\delta(r)\). Consider the function \(\delta \in \{0, \epsilon\} \rightarrow (r^\delta, \mu)\) where \((r^0, \mu) = (r, \mu)\) and for each \(\delta > 0\), \((r^\delta, \mu)\) is a solution to (12), which exists by Step 2. Since \(\mu\) is a perfect matching in \(F(r)\), \((r, \mu) \in F(N, A, u, m)\). By Alkan et al. (Lemma 3, 1991), for each \(i \in N\), \(u_i(r_{\mu(i)}(\gamma(i)) = u_i(r_{\sigma(i)}(\gamma(i))\). Since \((r, \sigma) \in \mathcal{R}(N, A, u, m)\), we have that
(r, μ) ∈ ℛ(N, A, u, m). Thus, for each δ ∈ [0, ε], (r^δ, μ) ∈ ℛ(N, A, u, m − δ). Thus, the function δ ↦ (r^δ, μ) is continuous; for each pair 0 < δ < η < ε, and each i ∈ N, u_i(r^η_{μ(i)}, μ(i)) > u_i(r^δ_{μ(i)}, μ(i)); and for each a ∈ A, r^δ_a > r^η_a (Alkan et al. 1991; Velez 2017).

The following lemma allows us to easily prove Lemma 3.

Lemma 7 Let u ∈ ℜ(N), ε > 0, (r, σ) ∈ F(N, A, u, m), and (t, σ) ∈ F(N, A, u, m − ε) such that for each a ∈ A, r_a > t_a. Suppose that B^u(r) = B^u(t). Then, there is Λ > 0 satisfying the property of Lemma 6, such that

$$(\log(r - t), \sigma) ∈ F\left(N, A, \hat{\lambda}_i, \sum_{a ∈ A} \log(r_a - t_a)\right),$$

where for each i ∈ N, \(\hat{\lambda}_i\) is the function \((y_a, a) ∈ ℝ × A ↦ \hat{\lambda}_i(y_a, a) := \log \lambda_{ia}(r, \Lambda) + y_a\).

Proof of Lemma 7 Consider a rebate and reshuffle at \((r, σ)\), i.e., a vector \(x := (x_a)_{a ∈ A} ∈ ℝ^A_{++}\) and an assignment \(μ : N → A\) that induce allocation \((r - x, μ)\). Since u and r will not change in the course of this proof, for simplicity in the notation, for each i ∈ N, let \(\lambda_{ia} := λ_{ia}(r)\) and \(\hat{\lambda}_{ia} := λ_{ia}(r, \Lambda)\) for some Λ satisfying the property in Lemma 6. For each i ∈ N, let \(\hat{\lambda}_i\) be the function \((y_a, a) ∈ ℝ × A ↦ \hat{\lambda}_i(y_a, a) := \log \hat{\lambda}_{ia} + y_a\). By Lemma 6, Λ can be chosen arbitrarily close to zero. Thus, we can suppose without loss of generality that for each a ∈ A such that \(u_i(r_{σ(i)}, σ(i)) > u_i(r_a, a)\) and each b such that \(u_i(r_{σ(i)}, σ(i)) = u_i(r_b, b)\),

$$\log \lambda_{ib} + \log(r_b - t_b) > \log \lambda_{ia} + \log(r_a - t_a).$$

For each a ∈ A, let \(y_a := \log(r_a - t_a)\). We claim that \((y, y) ∈ F(N, A, \hat{\lambda}_i, \sum_{a ∈ A} y_a)\). Since \((r, σ) ∈ F(N, A, u, m)\), for each i ∈ N, \(\hat{\lambda}_{ia}(i) = \hat{\lambda}_{ia}(i)\). Thus, by (15) for each a ∈ A such that \(u_i(r_{σ(i)}, σ(i)) > u_i(r_a, a)\),

$$\hat{\lambda}_i(y_{σ(i)}, σ(i)) = \log \hat{\lambda}_{ia}(i) + y_{σ(i)} > \log \lambda_{ia} + \log(r_a - t_a) = \hat{\lambda}_i(y_a, a).$$

Let \(a ∈ A\) be such that \(u_i(r_{σ(i)}, σ(i)) = u_i(r_a, a)\). Then, \(\hat{\lambda}_{ia} = \hat{\lambda}_{ia}\). Since \((t, σ) ∈ F(N, A, u, m - δ)\), \(u_i(t_{σ(i)}, σ(i)) ≥ u_i(r_a, a)\). Thus, \(\hat{\lambda}_{ia}(r_{σ(i)} - t_{σ(i)}) ≥ \lambda_{ia}(r_a - t_a)\). That is,

$$\hat{\lambda}_i(y_{σ(i)}, σ(i)) = \log \lambda_{ib} + \log(\delta_a ≥ \log \lambda_{ia} + \log(r_a - t_a) = \hat{\lambda}_i(y_a, a).$$

Since \((r, σ) ∈ F(N, A, u, m)\), for each a ∈ A, \(u_i(r_{σ(i)}, σ(i)) ≥ u_i(r_a, a)\). Thus, for each pair \(i, j ∈ N\), \(\hat{\lambda}_i(y_{σ(j)}, σ(i)) ≥ \hat{\lambda}_i(y_{σ(j)}, σ(j))\). Thus, \((y, σ) ∈ F(N, A, \hat{\lambda}_i, \sum_{a ∈ A} y_a)\). □

Proof of Lemma 3 By Lemma 7 there is Λ > 0 satisfying the property of Lemma 6, such that \((\log(r - t), σ) ∈ F(N, A, \hat{\lambda}_i, \sum_{a ∈ A} \log(r_a - t_a))\), where for each i ∈ N, \(\hat{\lambda}_i\) is the function \((y_a, a) ∈ ℝ × A ↦ \hat{\lambda}_i(y_a, a) := \log \lambda_{ia}(r, \Lambda) + y_a\). Since \(\hat{\lambda}_i\) is quasi-linear,
\( \sigma \) maximizes the summation of the values for \( \hat{u} \) (Svensson 1983). By Lemma 6, \( \sigma \) is a maximal weight perfect matching in \( \mathcal{F}^w(r) \). \( \Box \)

References

Alkan A (1989) Existence and computation of matching equilibria. Eur J Polit Econ 5:285–296
Alkan A, Demange G, Gale D (1991) Fair allocation of indivisible goods and criteria of justice. Econometrica 59(4), 1023–1039. http://www.jstor.org/stable/2938172
Andersson T, Ehlers L, Svensson L-G (2014) Budget-balance, fairness and minimal manipulability. Theor Econ 9(3):753–777. http://econtheory.org/ojs/index.php/te/article/view/20140753/0
Aragonès E (1995) A derivation of the money Rawlsian solution. Soc Choice Welf 12:267–276. https://doi.org/10.1007/BF00179981
Arunachaleswaran ER, Barman S, Rathi N (2018) Fully polynomial-time approximation schemes for fair rent division. Proceedings SODA 2019, arXiv:1807.04163
Foley D (1967) Resource allocation and the public sector. Yale Econ Essay 7:45–98
Gal Y, Mash M, Procaccia AD, Zick Y (2017) Which is the fairest (rent division) of them all? J ACM 64(6)
Goldman J, Procaccia AD (2014) Spliddit: unleashing fair division algorithms. SIGecom Exchanges 13(2):41–46
Nicolò A, Velez RA (2017) Divide and compromise. Math Soc Sc 90:100–110. http://www.sciencedirect.com/science/article/pii/S0165489617300690
Procaccia A, Velez RA, Yu D (2018) Fair rent division on a budget. In: Proceedings of the 2018, 32nd AAAI conference on artificial intelligence. AAAI. https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16936/15807
Svensson L-G (1983) Large indivisibles: an analysis with respect to price equilibrium and fairness. Econometrica 51:939–954. http://www.jstor.org/stable/1912044
Svensson L-G (2009) Coalitional strategy-proofness and fairness. Econ Theor 40(2):227–245. https://doi.org/10.1007/s00199-008-0366-4
Velez RA (2011) Are incentives against economic justice? J Econ Theory 146:326–345. https://doi.org/10.1016/j.jet.2010.10.005
Velez RA (2015) Sincere and sophisticated players in an equal-income market. J Econ Theory 157:1114–1129. https://doi.org/10.1016/j.jet.2015.03.006
Velez RA (2017) Sharing an increase of the rent fairly. Soc Choice Welf 48(1):59–80. https://doi.org/10.1007/s00355-016-1018-4
Velez RA (2018) Equitable rent division. ACM Trans Econ Comput (TEAC) 6(2):Article 9. https://doi.org/10.1145/3274528
Velez RA (2019) Expressive mechanisms for equitable rent division on a budget. arXiv:1902.02935

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