Fair rent division on a budget revisited

Stéphane Airiau, Hugo Gilbert, Umberto Grandi,
Jérôme Lang, and Anaëlle Wilczynski

Abstract

Rent division consists in simultaneously computing an allocation of rooms to agents and a payment, starting from an individual valuation of each room by each agent. When agents have budget limits, envy-free solutions do not necessarily exist. We propose two solutions to overcome this problem. In the first one, we relax envy-freeness to account for budget disparities. In the second one, we allow fractional allocations, in which agents may change rooms during the duration of the lease.

1 Introduction

A set of n agents agree to pay collectively the rent of a flat that contains n rooms. Rooms are not alike: an agent prefers some rooms to others. We assume preferences are modeled by valuations representing the maximum amount that a given agent is willing to pay for a given room. How should we assign rooms to agents and how should the rent be divided? This is the standard rent division problem. It is known that, provided that the valuations given to the different rooms by a given agent sum up to the rent, there always exists an allocation that is individually rational (no agent should pay more for a room than her maximum payment for that room), envy-free (no agent would prefer the allocation to another agent – room and payment – to their own), and that maximises both utilitarian and egalitarian social welfare [11]. This solution is implemented in the Spliddit platform [12] and is its most popular application.

The standard problem is often not realistic, because agents usually have a budget, that is, a maximum amount of money they can afford to pay. Searching for individually rational envy-free solutions to rent division with individual budgets is nontrivial [18] and may result in a failure to meet both conditions, as shown in the following example.

Example 1. *Two agents whose valuations are shown below. The rent for the flat is 1000.*

	room r_1	*room r_2*	*budget*
agent 1	*800*	*400*	*600*
agent 2	*800*	*400*	*500*

If r_2 is assigned to 1, and r_1 to 2, then individual rationality implies that 1 pays at most 400, therefore 2 has to pay at least 600, which exceeds her budget. Thus r_1 must be assigned to 1, and r_2 to 2. Because of her budget constraint, agent 1 cannot pay more than 600. Because of individual rationality, 2 cannot pay more than 400 for r_2; so, to reach the total rent of 1000*, 1 must pay 600 and 2 must pay 400. Assuming utilities are quasi-linear, 1's utility is* $800 - 600 = 200$ *(her valuation for r_1 minus her payment), and 2's utility is* $400 - 400 = 0$*. However, the utility 2 would enjoy from 1's share is* $800 - 600 = 200 > 0$*, and hence 2 envies 1.*

Example 1 shows that we cannot simultaneously satisfy individual rationality, budget limits, and envy-freeness.[1] Should we conclude that the agents should give up renting the flat and look for another one? We believe not, and propose two solutions:

[1] A similar example can be constructed even if, for each agent, the sum of all valuations is equal to the rent.

1. Allocate r_1 to 1 with payment 600 and r_2 to 2 with payment 400. The allocation is individually rational and respects individual budgets. It is not envy-free in the classical sense, but we may argue that 2's envy towards 1 is not justified: if 2 was allocated r_1 with payment 600, she would not be able to pay. Therefore, this allocation satisfies a weakening of envy-freeness, that we call *budget-friendly* envy-freeness (B-EF).

2. Allocate r_1 to 1 and r_2 to 2 *for the first half of the year* and then swap the rooms for the second half, asking a payment of 500 to each agent. This *fractional* allocation is envy-free (provided preferences do not depend on time), is individually rational, and respect budgets.

We explore these two ways of enlarging the set of fair allocations: budget-friendly envy-freeness and fractional allocations. After discussing related work, we present the basic definitions in Section 3. Section 4 defines B-EF, and shows algorithms to find a B-EF solution if either payments or the allocation is fixed. Section 5 turns to fractional envy-free allocations, which can be computed in polynomial time when they exist, and considers the *temporal implementation* of fractional allocations, with the aim to minimise the number of times agents have to change room. Last, we experimentally show in Section 6 that relaxing EF to B-EF and considering fractional allocations make it possible to enlarge significantly the set of instances for which a fair solution to a rent division problem exists.

2 Related Work

Rent division The rent division problem was first studied in the economics literature and more recently became the most used application on the Spliddit webpage. Initially, Spliddit implemented an algorithm by Abdulkadiroğlu et al. [2], which was updated based on the work of Gal et al. [11], who provided a linear program that finds a solution that both maximises the utility of the worst-off agent and minimises the gap between the best and the worst-off agent. For a unifying treatment of contributions in rent division in economics and computer science we refer to recent work by Velez [20].

Envy-freeness with budgets Our work takes its roots in the contribution of Procaccia et al. [18], who developed a polynomial algorithm to compute the maximin envy-free solution for rent division under budget. In the presence of individual budgets, the algorithm of Gal et al. [11] cannot be used as the existence of payments making any efficient allocation of rooms envy-free is not guaranteed. Our paper tackles the problem left open by Procaccia et al. [18] of what solution to propose when no envy-free solution exists. To the best of our knowledge the work of Velez [22] is the only other paper considering budgets. The solution proposed is to lift the assumption of quasi-linearity of preferences and ask agents to report their marginal disutility for exceeding their budget. Velez [21] investigates the incentive-compatibility of such mechanisms. Our notion of budget-friendly envy-freeness draws inspiration from work on envy-freeness in fair division with budgeted bidders [3, 4, 14, 15], and is related (in spirit) to justified envy-freeness in two-sided matching [1].

Randomised matching and fair division Allowing randomised solutions is a thoroughly studied idea in fair division and matching problems [6, 16]. Indeed, allowing more expressive solutions through, e.g., time sharing mechanisms, makes it possible to increase fairness guarantees and to bypass impossibility results [5, 7]. Randomised solutions for the rent division problem (without budgets) have been already investigated by Dufton and Larson [9], who studied to which extent randomised mechanisms can be strategy-proof and provide envy-freeness guarantees once a deterministic solution is sampled. Technically speaking, a fractional allocation and a randomized allocation are identical objects, but their interpretations differ. Also, our focus is not on strategyproofness but on envy-freeness: we determine if an envy-free fractional solution exists when agents have budgets and we seek for an implementation of such a solution that minimises the number of room swaps.

3 The Model

In this section, we present the model of rent division with individual budgets, and the properties of individual rationality and envy-freeness that are the focus of this paper.

3.1 Basic definitions

We consider a set R of n rooms that need to be allocated to a set A of n agents. Each agent $i \in A$ has a valuation $v_{ij} \in \mathbb{R}^+$ over each room $j \in R$, and L is the total rent that needs to be paid to secure the rooms. Note that differently than previous work we do not assume that $\sum_j v_{ij} = L$. A *rent division problem* is a tuple $\langle n, V, L \rangle$ where $V = (v_{ij})_{i \in A, j \in R}$.

A *solution* to a rent division problem consists of an assignment $\sigma : A \to R$ and a payment vector $p : A \to \mathbb{R}$, such that $\sum_i p_i = L$. Note that payments can possibly be negative. An assignment σ is *efficient* if $\sum_i v_{i\sigma(i)}$ is maximal over all possible allocations. Now we add a *budget* $b_i \in \mathbb{R}^+$ for each agent i. A solution is *affordable* if $p_i \leq b_i$ for all $i \in A$. Without loss of generality we assume that $\sum_i b_i \geq L$ (the agents can afford the total rent). A *rent division problem with individual budgets* is a tuple $\langle n, V, L, b \rangle$, where $b = (b_1, \ldots, b_n)$.

3.2 Envy-freeness

In line with previous work, we assume that agents have quasi-linear utilities, and say that a solution (σ, p) is *envy-free* (EF) if no agent can increase her utility by exchanging her assigned room and payment with another agent: (σ, p) is EF if $v_{i\sigma(i)} - p_i \geq v_{i\sigma(j)} - p_j$ for all agents i, j. While a rent division problem with unlimited budgets always admits an EF solution [19], this is not true in our setting, as shown by the following example.

Example 2. *Consider two rooms r_1 and r_2, and two agents with budget 500 each. Both agents value r_1 at 600 and r_2 at 400. The total rent is 1000, hence each agent has to pay 500, and the agent who gets r_2 envies the agent who gets r_1.*

3.3 Individual rationality

A solution (σ, p) is *individually rational* (IR) if for all agents i we have that $v_{i\sigma(i)} - p_i \geq 0$. Under our assumptions EF does not imply IR, as can be seen in the following example.

Example 3. *Consider the following 2-agent rent division problem where $L = 1000$:*

	room r_1	*room r_2*	*budget*
agent 1	*600*	*100*	*700*
agent 2	*100*	*300*	*300*

Let σ assign r_1 to 1 paying 700 and r_2 to 2 paying 300. (σ, p) is not IR, since the utility of 1 is -100, but it is EF: 2 (resp. 1) would have utility -600 (resp. -200) if she received r_1 and pay 700 (resp. received r_2 and pay 300), which is less than their current utility.

IR solutions to a rent division problem under budget can be found in polynomial time by solving a matching problem.

Proposition 1. *We can determine if there exists an IR and affordable allocation in polynomial time.*

Proof sketch. Consider the bipartite graph $((A, R), E)$. Add arcs $e_{a,r} \in E$ between each vertex $a \in A$ and $r \in R$ with weight $\min\{b_a, v_{a,r}\}$. This weight is the maximal price that agent a can pay for room r in an IR affordable allocation. It is sufficient to test if the matching of maximal weight on $((A, R), E)$ has total payoff greater than L, otherwise no individually rational and affordable allocation exists. □

4 Budget-Friendly Envy-Freeness

When individual payments are bounded by a budget, the notion of envy can be restricted to rooms that are affordable to an agent, obtaining a natural relaxation of envy-freeness.

Definition 1. *A solution (σ, p) is* budget-friendly envy-free *(B-EF) if for all agents i we have that:*

$$v_{i\sigma(i)} - p_i \geq v_{i\sigma(j)} - p_j \text{ for all agents } j \in A \text{ s.t. } p_j \leq b_i.$$

If for agents $i, j \in A$, we have that $v_{i\sigma(i)} - p_i < v_{i\sigma(j)} - p_j$ and $p_j \leq b_i$, we will say that agent i is *B-envious* of agent j.

Example 4. *Consider a two-agent rent division problem with rent $L = 800$. The individual valuations and budgets are given in the following table:*

	room r_1	*room r_2*	*budget*
agent 1	*500*	*200*	*500*
agent 2	*700*	*300*	*300*

Let σ allocate r_1 to agent 1 and r_2 to agent 2, and let $p_1 = 500$ and $p_2 = 300$. In (σ, p), 1 does not envy 2, but 2 envies 1 because she would get utility 700-500 = 200 if she was assigned r_1 with payment 500, therefore (σ, p) is not EF. However, (σ, p) is B-EF: 2 does not envy 1 under Definition 1 because 1's payment (500) exceeds 2's budget (300).

Observe that the allocation σ in Example 4 is not efficient. This contrasts with the classical setting of rent division where EF solutions are necessarily based on efficient allocations. Still, we can show that in a B-EF solution, both the allocation (ignoring payments) and the solution are Pareto-optimal:

Proposition 2. *If (σ, p) is a B-EF solution then the allocation σ is Pareto-optimal, i.e., there is no allocation θ such that for all $i \in A$ we have that $v_{i\theta(i)} \geq v_{i\sigma(i)}$, and such that $v_{i\theta(i)} > v_{i\sigma(i)}$, for some agent $i \in A$.*

Proof. Suppose θ is an allocation that Pareto-dominates σ. Then, there is an improvement cycle, without loss of generality, $i = 1, \ldots, k$, such that $\theta(i) = \sigma(i + 1)$ for all $i < k$, $\theta(k) = \sigma(1)$, and (1) $v_{i\theta(i)} \geq v_{i\sigma(i)}$ for all i and (2) $v_{1\theta(1)} > v_{1\sigma(1)}$. Denote $[i + 1] = i + 1$ if $i < k$ and $[k + 1] = 1$. Consider the envy in (σ, p) between agents in the cycle. Since σ is B-EF, for each $i = 1, \ldots, k$ there are two possible cases.

- Assume $p_{[i+1]} \leq b_i$, then there should be no envy, thus $v_{i\sigma(i)} - p_i \geq v_{i\sigma([i+1])} - p_{[i+1]}$. Using (1) and (2) we obtain $p_{[i+1]} \geq p_i$ for all $i \leq k$, and $p_2 > p_1$.
- Assume $p_{[i+1]} > b_i$. By affordability of (σ, p) we have $p_i \leq b_i$, therefore, $p_{[i+1]} > p_i$.

In both cases we have $p_2 > p_1$, $p_3 \geq p_2$, …, $p_k \geq p_{k-1}$, and $p_1 \geq p_k$, implying $p_1 > p_1$. □

Proposition 3. *If (σ, p) is a B-EF solution then (σ, p) is a Pareto-optimal solution, i.e., there is no solution (θ, q) such that for all $i \in A$ we have $v_{i\theta(i)} - q_i \geq v_{i\sigma(i)} - p_i$, and such that $v_{i\theta(i)} - q_i > v_{i\sigma(i)} - p_i$, for some agent $i \in A$.*

Proof. Let (σ, p) be a B-EF solution. Let us assume towards a contradiction that there exists another solution (θ, q) that Pareto-dominates (σ, p).

We first prove that all rooms are paid the same price in (σ, p) and (θ, q). Let us assume there exists a room j for which the price is strictly larger in (θ, q) than in (σ, p). Let k, l be the two agents such that $\theta(k) = \sigma(l) = j$. Our assumption is that $q_k > p_l$.

We obtain that:

$$\begin{aligned} v_{k\sigma(k)} - p_k &\leq v_{k\theta(k)} - q_k && \text{(Pareto-domination of } (\sigma, p) \text{ by } (\theta, q)) \\ &< v_{k\sigma(l)} - p_l && (q_k > p_l \text{ and } \theta(k) = \sigma(l)). \end{aligned}$$

Since $p_l < q_k \leq b_k$, agent k B-envies l in σ, yielding a contradiction. Hence no room has a larger price in (θ, q) than in (σ, p). As $\sum_{i \in A} q_i = \sum_{i \in A} p_i = L$, this entails that all rooms have exactly the same price in both solutions.

Let $k \in A$ be such that $v_{k\sigma(k)} - p_k < v_{k\theta(k)} - q_k$ (such an agent exists by Pareto dominance). Let $l \in A$ be such that $\sigma(l) = \theta(k)$. We obtain that:

$$\begin{aligned} v_{k\sigma(k)} - p_k &< v_{k\theta(k)} - q_k \\ &= v_{k\sigma(l)} - p_l \text{ (as } \sigma(l) = \theta(k) \text{ and } p_l = q_k) \end{aligned}$$

Since $p_l = q_k \leq b_k$, k B-envies l in (σ, p), yielding a contraction. □

An IR and B-EF solution can be found by solving a mixed-integer linear program (see the Appendix for a detailed formulation). Note however that a B-EF solution does not always exist, as can be seen in the introductory example by Procaccia et al. [18]:

Example 5. *Consider a two-agent rent division problem with $L = 1000$ and $b_1 = b_2 = 500$. Both agents evaluate r_1 at 800 and r_2 at 500. The agent who receives room r_1 has to pay 500, producing (budget-friendly) envy in the other agent who receives r_2 at a price of 500.*

In what follows we give two algorithms that find B-EF solutions in pseudopolynomial (respectively, polynomial) time when the allocation (resp., the payment vector) is fixed.[2]

4.1 Computing B-EF solutions: fixed allocation

Here, we fix an allocation and we check in pseudo-polynomial time whether a B-EF solution exists, and when it does, we output a corresponding price vector. To obtain our pseudo-polynomiality result we restrain in this subsection the input parameters L, v_{ij} and b_i to $\mathbb{Z}^+$ for all $i \in A$ and $j \in R$.

We first define a weakening of budget envy-freeness: given a solution (σ, p), we say that agent i strongly B-envies (SB-envies) j if $p_j < b_i$ and $v_{i\sigma(j)} - p_j > v_{i\sigma(i)} - p_i$; and that (σ, p) is *weakly budget envy-free* (WB-EF) if no agent SB-envies another one. Remark that if i B-envies j but does not strongly B-envies j then $p_j = b_i$ and $v_{i\sigma(j)} - p_j > v_{i\sigma(i)} - p_i$.

Our result uses Algorithm 1 of Kempe et al. [15], which finds minimal payments for a given allocation to make the resulting solution WB-EF, and runs in pseudo-polynomial time. As Kempe et al. [15] do not have the constraint of a rent to be paid, we add a final processing stage guaranteeing that the payments sum up to the rent, and that the solution is B-EF (not only WB-EF).

[2]Complete proofs of Theorems 1 and 2 can be found in the appendix.

Algorithm 1: B-EF payment, allocation fixed

Data: instance $\langle n, V, L, b\rangle$, allocation σ

Start from $p_i = L - \sum_{k\in A\setminus\{i\}} b_k, \forall i \in A$;

Run Strong B-Envy Removal (cf. Algorithm 1 of Kempe et al. [15]);

Run Final Payment Increase;

return (σ, p)

Algorithm 2: Strong B-Envy Removal

Data: instance $\langle n, V, L, b\rangle$, allocation σ, payment p

while *There exists edge* $(i, j) \in G_p$ *with* $p_j < p_i - \lambda_{ij}$ **do**

  $p_j \leftarrow \min\{b_i, p_i - \lambda_{ij}\}$;

  if $p_j = b_i$ **then** delete (i, k) from G_p for all (i, k) such that $p_k \geq b_i$;

  if $p_j > b_j$ *or* $p_j > v_{j\sigma(j)}$ **then return** no solution;

if $\sum_i p_i > L$ **then return** no solution;

return (σ, p)

Algorithm 1 of Kempe et al. [15] starts from a lower bound on initial agents' payments, and iteratively increases payments in order to eliminate SB-envy relations by reasoning on a weighted envy graph G_p: given an allocation σ and a payment vector p, G_p is defined by taking n nodes and add edge (i, j) if $p_j < b_i$, and label existing edges with $\lambda_{ij} = v_{i\sigma(i)} - v_{i\sigma(j)}$. That is, G_p contains an edge from i to j if i can afford the price paid by j. Observe that the labels of the edges do not depend on the payments: they only represent the potential envy generated by the allocation σ.

Theorem 1. *Given a fixed allocation σ, we can determine in pseudo-polynomial time if there exists a payment vector p such that (σ, p) is affordable, IR, and B-EF.*

Proof sketch. Our Algorithm 1 starts from an initial payment vector $p = (L - \sum_{k\in A\setminus\{1\}} b_k, ..., L - \sum_{k\in A\setminus\{n\}} b_k)$ and draws the envy graph G_p. It then uses Algorithm 2 (which corresponds to Algorithm 1 in Kempe et al. [15]) to remove SB-envy, if possible, among the agents. If the algorithm does not output a payment vector, or if the sum of payments exceeds the rent, then no solution exists. If Algorithm 2 returns a solution such that the sum of the payments is lower than the rent, then we increase it uniformly (up to the budget or the valuation of the assigned room) to obtain a B-EF solution using Algorithm 3. If this is not possible because of an incompatibility with B-EF, budget limits, or IR, then we output that there is no solution. □

We explain our proposed algorithm on the following example:

Example 6. *Let $n = 3$, $R = \{r_1, r_2, r_3\}$, $L = 1000$, valuations and budgets as follows:*

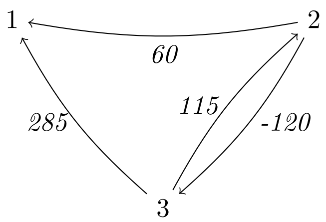
	room r_1	*room r_2*	*room r_3*	*budget*
agent 1	*340*	*300*	*500*	*300*
agent 2	*290*	*350*	*470*	*380*
agent 3	*200*	*370*	*485*	*400*

Consider the allocation $\sigma(i) = r_i$ for $i = 1, 2, 3$. We start from initial payments $p = (220, 300, 320)$, and draw the corresponding envy graph G_p:

Algorithm 3: Final Payment Increase

Data: instance $\langle n, V, L, b\rangle$, allocation σ, payments p

1 $A' \leftarrow \{i \in A : p_i < \min\{b_i, v_{i\sigma(i)}\}\}$;
2 **while** $\sum_{i \in A} p_i < L$ **do**
3   **if** $A' = \emptyset$ **then return** no solution;
4     $\forall i \in A'$, $m_i \leftarrow \min\left\{b_i, v_{i\sigma(i)}, \min_{j \in A \setminus A' \text{ st } p_j \leq b_i} v_{i\sigma(i)} - v_{i\sigma(j)} + p_j\right\}$;
5     $q \leftarrow \max\{0, \min\left\{\frac{L - \sum_{i \in A} p_i}{|A'|}, \min_{i \in A'}\{m_i - p_i\}\right\}\}$;
6   **for** *each* $i \in A'$ **do** $p_i \leftarrow p_i + q$;
7     $A' \leftarrow \{i \in A' : p_i < m_i\}$;
8 **if** (σ, p) *is not B-EF* **then return** no solution;
9 **return** (σ, p)



Then, when we run Algorithm 2, we can select the edge $(2,1)$ *in* G_p *such that* $p_1 < p_2 - \lambda_{21}$*, i.e.,* $\lambda_{21} = 60 < 300 - 220 = 80$*. We treat this edge by updating* p_1 *to* $\min\{300, 300 - 60\} = 240$*. Then, we can select edge* $(2,3)$ *where* $p_3 < p_2 - \lambda_{23}$*, i.e.,* $-120 < -20$*. We treat this edge by updating* p_3 *to* $\min\{380, 300 - (-120)\} = 380$*, hence, removing edge* $(2,3)$ *from* G_p*. We finally obtain payment* $p = (240, 300, 380)$*. This payment generates no SB-envy, and the sum of the payments is lower than the rent.*

Now we increase the payment of the agents to reach the rent using Algorithm 3. We can first uniformly increase the payment of all agents by 20*, reaching payment* $p = (260, 320, 400)$*, implying that agent 3 reaches her budget (and thus she will not be part of subset* A' *anymore). Finally, we can uniformly increase the payment of agents 1 and 2 by* 10*, reaching payment* $p = (270, 330, 400)$ *to exactly reach the rent.*

In Algorithm 2, we have at most n^2 edges to check at each iteration of the while loop, and we will have at most $(\sum_{i \in A} b_i)$ iterations. So the algorithm runs in $(\sum_{i \in A} b_i)n^2$ operations. In Algorithm 3, the while loop runs for at most n iterations, and each iteration of the loop requires n^2 operations due to Line 4. Hence, the algorithm runs in $O(n^3)$ operations. To sum up, we obtain a pseudo-polynomial algorithm, running in time $O((\sum_{i \in A} b_i)n^3)$, which is very reasonable. Obtaining a polynomial algorithm would be even better. We thought of reusing Algorithm 2 of [15], which is claimed to compute a WB-EF solution in polynomial time, but we have doubts about its correctness (and no proof is given in Kempe et al. [15]).

Still, now that we know that *given an initial allocation*, we can compute a payment vector that satisfies B-EF, whenever there exists one, in time $O((\sum_{i \in A} b_i)n^3)$. This implies that given the valuations, the rent and the budgets, we can compute a solution, if any, in time $O((\sum_{i \in A} b_i)n!n^3)$. In everyday rent division problems, n is low (typically, no more than 5), therefore we can compute a solution in a reasonable amount of time.

4.2 Computing B-EF solutions: fixed payments

In practice, agents looking for flat-sharing will often search for apartments with a rent corresponding to their accumulated budget. In such a case, the payments are fixed as each agent i must pay b_i to reach the total rent L. We show that, more generally, given any fixed payment vector p, we can efficiently determine if there exists an assignment σ of agents to rooms such that (σ, p) is affordable, IR, and B-EF.

Theorem 2. *Given a fixed payment vector p, we can determine if there exists an assignment σ of agents to rooms such that (σ, p) is affordable, IR, and B-EF in polynomial time.*

Proof sketch. First, we can easily check whether the payments are compatible with an affordable solution which meets the rent. Our algorithm tries to build an IR and B-EF assignment in a greedy fashion considering agents in decreasing order w.r.t. payments, as follows: We partition the set of agents into k groups $(B_1, \ldots, B_k)$, i.e., $\bigcup_{\ell=1}^{k} B_\ell = A$ and $B_\ell \cap B_{\ell'} = \emptyset$ for every $\ell \neq \ell' \in [k]$, such that for all agents $i, j \in B_\ell$, $p_i = p_j$, and for all agents $i \in B_\ell$ and $j \in B_{\ell'}$ with $\ell < \ell'$, we have that $p_i > p_j$. Then, we consider sets B_ℓ with increasing values of ℓ (hence, with decreasing payments), and try to assign each agent i in B_ℓ to a room in $top(i)$, the set compounded of her most preferred rooms within the remaining ones. This is done by considering a bipartite graph and determining if there exists a perfect matching in this graph. If there is no such an assignment, or if it violates an IR or B-EF constraint, then we conclude that no valid solution exists.

One can prove that this algorithm returns an assignment σ such that (σ, p) is affordable, IR and B-EF iff such an assignment exists. The key idea is that, for an assignment to be IR and B-EF, each agent i must receive a room in $top(i)$. □

The algorithm described in the proof of Theorem 2 is illustrated in the next example.

Example 7. *Consider the following 4-agent rent division problem where $L = 1000$:*

	r_1	r_2	r_3	r_4	*budget = payment*
agent 1	*100*	*450*	*600*	*300*	*400*
agent 2	*400*	*400*	*700*	*200*	*250*
agent 3	*400*	*100*	*500*	*250*	*250*
agent 4	*300*	*100*	*400*	*300*	*100*

The budgets sum to the rent, therefore the payment p_i for each i is fixed to her budget. The agents are partitioned into 3 groups w.r.t. their payments: $B_1 = \{1\}$, $B_2 = \{2, 3\}$, and $B_3 = \{4\}$. We can define for each agent her top subset of rooms, i.e., their most preferred rooms among the remaining ones by considering the agents in the order of their group: $top(1) = \{r_3\}$, $top(2) = \{r_1, r_2\}$, $top(3) = \{r_1\}$, and $top(4) = \{r_4\}$. A perfect matching, that satisfies the IR and B-EF constraints, can be found at each step $\ell \in [3]$ between agents in B_ℓ and rooms in $\bigcup_{i \in B_\ell} top(i)$. This process results in the unique assignment σ such that (σ, p) is IR and B-EF, where $\sigma(1) = r_3$, $\sigma(2) = r_2$, $\sigma(3) = r_1$, and $\sigma(4) = r_4$. Note that if, we change v_{1,r_2} from 450 to 460, such an assignment σ would not exist because agent 1 would necessarily B-envy agent 2.

5 Fractional Solutions

In this section, we propose a second solution to find envy-free allocations under individual budgets. The idea is to allow agents to spend a fraction of their time in different rooms, and we study possible implementations of the resulting fractional allocation that minimise the number of room swaps.

Definition 2. *A* fractional solution *to a rent division problem is an $n \times n$ bi-stochastic matrix X, with x_{ij} be the fraction of time agent i spends in room j, and a price vector $p : A \to \mathbb{R}$.*

The definitions of IR and EF easily extend to fractional solutions. We say that (X, p) is *individually rational* under quasi-linear utilities if for all agents i we have that $\sum_{j \in R} x_{ij} v_{ij} - p_i \geq 0$. Further, we say that a fractional solution (X, p) is *envy-free* under quasi-linear utilities if the following holds for all agents i and i' in A:

$$\sum_{j \in R} x_{ij} v_{ij} - p_i \geq \sum_{j \in R} x_{i'j} v_{ij} - p_{i'}.$$

Observe that the initial Example 2 admits a fractional EF-solution: let agent 1 spend 6 months a year in room r_1 and the remaining part in room r_2 (and symmetrically for agent 2). If both agents pay 500 their utility is 0 and by symmetry no agent envies the other. However, fractional EF-allocations do not always exist, as shown by the following example.

Example 8. *Consider the following rent division problem under budget with $L = 1000$:*

	room r_1	*room r_2*	*budget*
agent 1	*700*	*400*	*700*
agent 2	*800*	*300*	*300*

The only affordable allocation is non-fractional: it assigns room r_2 to agent 2 at a price of 300, with 1 envying 2.

Allowing fractional allocations is a significant weakening that allows to obtain a solution for quite many instances for which there would be otherwise no solution. To illustrate this, we define below a family of instances for which this is indeed the case.

Proposition 4. *For each budget vector $b = (b_1, \ldots, b_n)$ such that $\frac{L}{n} \leq b_i < L$, there exists a rent division problem which does not admit an affordable EF deterministic solution but that admits a fractional one.*

Proof. We first show that if $b_i \geq \frac{L}{n}$ for all $i \in A$, then $\langle n, V, L, b \rangle$ admits an affordable fractional EF allocation. To see this, fix payments $p_i = \frac{L}{n}$ to be equal for all agents. The fractional solution where all agents spend the same fraction of time in each room, i.e., $x_{ij} = 1/n$ for all i and j, is an affordable EF solution.

Now, for all $i \in A$ we fix agent i's evaluation of room r_1 at b_i, zero otherwise – that is, for all i, $v_{i1} = b_i$ and $v_{ij} = 0$ for $j \geq 2$. Assume without loss of generality that a deterministic allocation gives room r_1 to agent 1. Given that $b_1 < L$ then $p_i > 0$ for at least one other agent, who has negative utility and envies agent 1. □

5.1 Computing fractional solutions

Fractional solutions that are IR and EF, when they exist, can be found in polynomial time by using the following Linear Program (LP). The LP considers as variables $x_{ij} \in [0, 1]$ for $i \in A$ and $j \in R$ for the fraction of time i spends in room j, and p_i for $i \in A$ as the price of agent i. The set of linear constraints is the following, formalising that each agent has a room allocated all of the time, that the payments sum to the rent, with the last two lines enforcing IR and EF:

$$\begin{array}{ll}
\sum_i x_{ij} = 1 & \forall j \in R \\
\sum_j x_{ij} = 1 & \forall i \in A \\
\sum_i p_i = L & \\
p_i \leq b_i & \forall i \in A \\
(\sum_j x_{ij} v_{ij}) - p_i \geq 0 & \forall i \in A \\
(\sum_j x_{ij} v_{ij}) - p_i \geq (\sum_j x_{i'j} v_{ij}) - p_{i'} & \forall i, i' \in A
\end{array}$$

When this set of linear constraints has a solution, it may in fact have many solutions. As in Procaccia et al. [18] (cf. their Theorem 1), the objective function can be defined so as to maximise a fairness criterion, such as: *maxmin* (with one additional variable y, add constraints $y \leq \sum_j (x_{ij} v_{ij}) - p_i$ for all agents i and, as objective function, maximise y); or *equitability* (with one additional variable y, add constraints $y \geq (\sum_j (x_{ij} v_{ij}) - p_i) - (\sum_j (x_{i'j} v_{ij}) - p_{i'})$ for any i and i' and, as objective function, minimise y).

5.2 Implementing fractional allocations

A fractional solution to a rent division problem can give rise to multiple practical implementations, depending on the sequence of room swaps that agents perform. By Birkhoff's theorem we know that any bi-stochastic matrix X can be decomposed as the convex combination of permutation matrices. In our terminology, this implies that for any fractional solution X there exist $\lambda_1, \dots, \lambda_k \in (0, 1]$, with $\sum_t \lambda_t = 1$, and $\sigma_1, \dots, \sigma_k$ deterministic solutions, such that for all $i \in A$ and $j \in R$ we have that $\sum_{\{t|\sigma_t(i)=j\}} \lambda_t = x_{ij}$. In line with previous work, we call such a representation a Birkhoff-von Neumann (BvN) decomposition of X of size k. The order in which the permutations of a BvN decomposition are considered gives rise to different implementations of a fractional solution X in terms of room swaps:

Definition 3. *An* implementation I *of length* k *of a fractional solution* X *is given by* $(\Lambda, <)$ *where* Λ *is a* k*-BvN decomposition of* X *and* $<$ *is an ordering on* $[k] = 1, \dots, k$*.*

When I is fixed, for simplicity we will assume that $\sigma_1, \dots, \sigma_k$ are given following ordering $<$. To discriminate between possible implementations of a fractional solution X, we define a natural notion that counts the overall number of swaps that an agent has to perform:

Definition 4. *Given an implementation* I *of* X*, the* switch price *of agent* i *is*

$$S_i(I) = |\{t \in \{1, \dots, k-1\} : \sigma_t(i) \neq \sigma_{t+1}(i)\}| .$$

The following example shows that an implementation in which agents never move back to the same room is not guaranteed to exist:

Example 9. *Consider the following fractional allocation* X*:*

	room r_1	*room* r_2	*room* r_3
agent 1	*0.6*	*0.3*	*0.1*
agent 2	*0.2*	*0.5*	*0.3*
agent 3	*0.2*	*0.2*	*0.6*

Agent 1 and 3 have to spend 60% of the time in a room, and agent 2 only 50%. Thus, one of 1 and 3 has to go back to the same room in any implementation of X*.*

5.3 Computing minimal-switching implementations

We now show that finding an implementation of a fractional solution minimising the number of switches is computationally hard. We begin by the following decision problem.

Minsum-Switch-Implementation
INPUT: Fractional solution X, $k \in \mathbb{N}$ QUESTION: is there an implementation I of X such that $\sum_i S_i(I) \leq k$?

Theorem 3. Minsum-Switch-Implementation *is NP-complete.*

The proof (in the appendix) uses a reduction from Partition. Now, we show that even if the deterministic allocations composing a BvN decomposition are fixed, finding an ordering that minimises the switch cost is an intractable problem.

Minsum-Switch-Ordering
INPUT: BvN decomposition Λ of length k, $K \in \mathbb{N}$ QUESTION: is there an ordering $<$ over $[k]$ such that $\sum_i S_i(I) \leq K$ where $I = (\Lambda, <)$?

Theorem 4. Minsum-Switch-Ordering *is NP-complete.*

Proof sketch. Membership to NP is straightforward. To prove hardness we present a reduction from the NP-hard Hamming Salesman Problem (HSP) [10]. An instance of HSP is a string $P = v_1 \dots v_n, L$, where $v_i \in \{0,1\}^m$, for some n and m, and L is an integer in binary representation. The question is to determine if there exists a Hamiltonian cycle over vertices v_i of total cost less than L, where the distance between two nodes is given by the Hamming distance. We first show (in the supplementary material) that finding a Hamiltonian path instead of a Hamiltonian cycle is also NP-Hard. Consider now an instance of HSP $P = v_1 \dots v_n, L$. We create an instance of Minsum-Switch-Ordering where there are $2m$ agents and $2m$ rooms. For each vertex v we create a deterministic allocation σ^v of the rooms as follows: agent i will be assigned to room i (resp. $m+i$) and agent $m+i$ will be assigned to room $m+i$ (resp. i) if the i-th bit of v is 0 (resp. 1), for all i in $[m]$. It is clear that the switch cost between σ^v and $\sigma^{v'}$ is equal to two times the Hamming distance between v and v'. Thus, there is a one-to-one correspondence between Hamiltonian paths on vertices of P and orderings of solutions σ^v. It is therefore sufficient to run Minsum-Switch-Ordering on an implementation composed of σ^v for $v \in P$ and $K = 2L$ to obtain a solution to the initial HSP instance. □

We conjecture that minimising the maximum switch cost is NP-hard as well. Dufossé and Uçar [8] showed that the problem of finding a BvN decomposition with the smallest support (i.e., with the smallest k) is NP-hard, but this does not necessarily correspond to an implementation which minimises the switch cost.

Even if we showed that finding minimal-switching implementations is computationally hard, the number of agents in typical rent division problems is low, thus the size k of a BvN decomposition is also likely to be small, since $k \leq n^2$. Hence, finding minimal-switching implementations can still be performed, e.g., by working on the polytope of deterministic assignments. For instance, Minsum-Switch-Ordering can easily be solved by dynamic programming in $O^*(2^k)$ by using the formula:

$$\Delta(\sigma, S) = \min_{\sigma' \in S \setminus \{\sigma\}} (sc(\sigma, \sigma') + \Delta(\sigma', S \setminus \{\sigma\}))$$

where $sc(\sigma, \sigma') = |\{i \in A | \sigma(i) \neq \sigma'(i)\}|$ is the switch-cost incurred by moving from σ to σ' and $\Delta(\sigma, S)$ is the minimal switch cost incurred by ordering permutations in $S \subseteq \Lambda$ under the constraint that σ is placed in the first position. The base cases are $\Delta(\sigma, \{\sigma\}) = 0$, and the optimal Minsum-Switch-Ordering value is obtained by considering $\min_{\sigma \in \Lambda} \Delta(\sigma, \Lambda)$.

6 Discussion

We proposed two approaches to increase the number of instances where a fair rent division is returned. The first one relaxes the notion of envy-freeness to take budget discrepancies into consideration. The second one allows for fractional allocations that are implemented by having agents swap their rooms (and minimising the number of swaps). We can of course

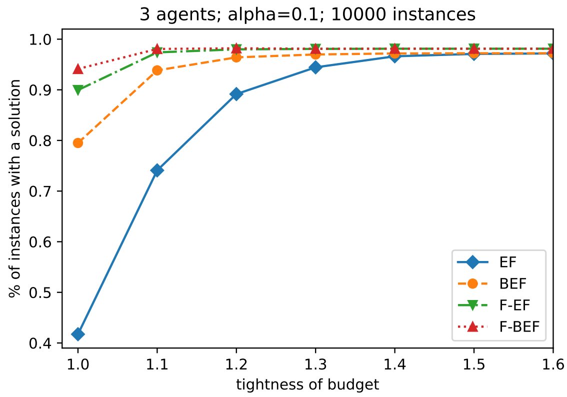


Figure 1: Proportion of rent division instances that admit a solution that is IR and EF, B-EF, fractional (F-EF), or an extension of fractional solutions with B-EF replacing EF (F-BEF), depending on the tightness of the agents' budgets.

combine the two approaches and define *budget envy free fractional solutions*. We leave this mostly for further study (but see below for how we considered it in our experiments).

We evaluated in simulations the number of additional solutions that our proposals can provide in synthetically generated rent division problems. We generated the agents' valuations of rooms starting from a base value M_j for each room, sampled from a uniform distribution in $[25, 50]$. All other parameters are sampled from the following normal distributions:

$$v_{ij} \sim \mathcal{N}(M_j, \alpha M_j),\ L \sim \mathcal{N}(\sum_j M_j, \alpha \sum_j M_j),\ b_i \sim \mathcal{N}(\sum_j \frac{M_j}{n}, \alpha \sum_j \frac{M_j}{n}),$$

where $0 < \alpha < 1$. In this way, we generate rent division problems where agents have a correlated valuations for the rooms, and have a budget that is roughly one n-th of the rent to be paid (i.e., the agents can pay the rent but their budget is tight). We discarded all instances that did not admit an IR solution, and we then increased the individual budgets by multiplying them by a *budget tightness* factor which varies between 1 and 2.

Figure 1 presents our findings for 3 agents setting $\alpha = 0.1$. We observe that B-EF and fractional solutions increase significantly the proportion of instances in which a fair allocation exists. In the extreme case of budget tightness equal to 1, there are twice more instances that can be solved by a B-EF solution than those that allow a classical EF solution. As expected, when the budget is less tight it becomes more and more likely to find a solution (irrespective of the fairness criteria).[3] We find similar results for $n \in \{2, 4, 5\}$ and different values of α. More details on the experimental design and further results can be found in the supplementary material.

For future work, identifying the computational complexity of determining the existence of B-EF solutions seems to be a challenging open problem. A further interesting direction is to estimate the robustness of our solutions under perturbations of the individual budgets, in line with the work of Peters et al. [17] who however focus on the agents' valuations.

[3]There exists however (few) rent division problems that do not admit an IR and EF solution, even with unlimited budgets (recall that we do not assume that individual valuations sum to the rent).

References

[1] Atila Abdulkadiroğlu and Tayfun Sönmez. School choice: A mechanism design approach. *American economic review*, 93(3):729–747, 2003.

[2] Atila Abdulkadiroğlu, Tayfun Sönmez, and M Utku Ünver. Room assignment-rent division: A market approach. *Social Choice and Welfare*, 22(3):515–538, 2004.

[3] Gagan Aggarwal, Shan Muthukrishnan, Dávid Pál, and Martin Pál. General auction mechanism for search advertising. In *Proceedings of the 18th International Conference on World Wide Web (WWW)*, 2009.

[4] Itai Ashlagi, Mark Braverman, Avinatan Hassidim, Ron Lavi, and Moshe Tennenholtz. Position auctions with budgets: Existence and uniqueness. *The B.E. Journal of Theoretical Economics*, 10(1):0000102202193517041648, 2010. doi: 10.2202/1935-1704.1648. URL `https://doi.org/10.2202/1935-1704.1648`.

[5] Haris Aziz. A probabilistic approach to voting, allocation, matching, and coalition formation. In *The Future of Economic Design*, pages 45–50. Springer, 2019.

[6] Anna Bogomolnaia and Hervé Moulin. A new solution to the random assignment problem. *Journal of Economics Theory*, 100(2):295–328, 2001. doi: 10.1006/jeth.2000.2710. URL `https://doi.org/10.1006/jeth.2000.2710`.

[7] Florian Brandl, Felix Brandt, and Hans Georg Seedig. Consistent probabilistic social choice. *Econometrica*, 84(5):1839–1880, 2016.

[8] Fanny Dufossé and Bora Uçar. Notes on Birkhoff–von Neumann decomposition of doubly stochastic matrices. *Linear Algebra and its Applications*, 497:108–115, 2016.

[9] Lachlan Dufton and Kate Larson. Randomised room assignment-rent division. In *Proceedings of the IJCAI Workshop on Social Choice and Artificial Intelligence*, 2011.

[10] Jarmo Ernvall, Jyrki Katajainen, and Martti Penttonen. NP-completeness of the hamming salesman problem. *BIT Numerical Mathematics*, 25(1):289–292, 1985.

[11] Ya'akov Gal, Moshe Mash, Ariel D Procaccia, and Yair Zick. Which is the fairest (rent division) of them all? *Journal of the ACM (JACM)*, 64(6):1–22, 2017.

[12] Jonathan R. Goldman and Ariel D. Procaccia. Spliddit: unleashing fair division algorithms. *SIGecom Exch.*, 13(2):41–46, 2014. doi: 10.1145/2728732.2728738. URL `https://doi.org/10.1145/2728732.2728738`.

[13] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL `https://www.gurobi.com`.

[14] Monika Henzinger and Veronika Loitzenbauer. Truthful unit-demand auctions with budgets revisited. *Theoretical Computer Science*, 573:1–15, 2015.

[15] David Kempe, Ahuva Mu'alem, and Mahyar Salek. Envy-free allocations for budgeted bidders. In *Proceedings of the 15th International Workshop on Internet and Network Economics (WINE)*, pages 537–544, 2009.

[16] Jinpeng Ma. On Randomized Matching Mechanisms. *Economic Theory*, 8(2):377–381, August 1996. URL `https://ideas.repec.org/a/spr/joecth/v8y1996i2p377-81.html`.

[17] Dominik Peters, Ariel D Procaccia, and David Zhu. Robust rent division. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022.

[18] Ariel Procaccia, Rodrigo Velez, and Dingli Yu. Fair rent division on a budget. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 1177–1184, 2018.

[19] Lars-Gunnar Svensson. Large indivisibles: an analysis with respect to price equilibrium and fairness. *Econometrica: Journal of the Econometric Society*, pages 939–954, 1983.

[20] Rodrigo A Velez. Equitable rent division. *ACM Transactions on Economics and Computation (TEAC)*, 6(2):1–25, 2018.

[21] Rodrigo A Velez. Expressive mechanisms for equitable rent division on a budget. *arXiv preprint arXiv:1902.02935*, 2019.

[22] Rodrigo A Velez. A polynomial algorithm for maxmin and minmax envy-free rent division on a soft budget. *Social Choice and Welfare*, 59:1–26, 2022.

Stéphane Airiau, Hugo Gilbert
LAMSADE, PSL
Paris, France
Email: stephane.airiau@dauphine.fr,hugo.gilbert@lamsade.dauphine.fr

Jérôme Lang
LAMSADE, PSL, CNRS
Paris, France
Email: Jerome.Lang@lamsade.dauphine.fr

Umberto Grandi
IRIT, Université Toulouse Capitole
Toulouse, France
Email: umberto.grandi@irit.fr

Anaëlle Wilczynski
MICS, CentraleSupélec, Université Paris-Saclay
Paris, France
Email: anaelle.wilczynski@centralesupelec.fr

Appendix

MIP for computing B-EF solutions

The following mixed linear integer program can be used to find B-EF solutions to a rent division problem with budgets. We use binary variables x_{ij} for $i \in A$ and $j \in R$ to model the assignment, and continuous variables p_i for $i \in A$ to model the payments. We include constraints to build an IR solution:

$$\begin{array}{ll} \sum_i x_{ij} = 1 & \forall j \in R \\ \sum_j x_{ij} = 1 & \forall i \in A \\ \sum_i p_i = L & \\ p_i \leq b_i & \forall i \in A \\ (\sum_j x_{ij} v_{ij}) - p_i \geq 0 & \forall i \in A \end{array}$$

For B-EF we add the following constraints:

$$\begin{array}{ll} \sum_j v_{ij} x_{ij} - p_i + M c_{ii'} \geq \sum_j v_{ij} x_{i'j} - p_{i'} & \forall i, i' \in A \\ c_{ii'} + d_{ii'} = 1 & \forall i, i' \in A \\ p_{i'} - c_{ii'} M \leq b_i & \forall i, i' \in A \\ b_i - d_{ii'} M + \lambda \leq p_{i'} & \forall i, i' \in A \end{array}$$

An agent i can envy another agent i' only when she can afford the payment, i.e. when $p_{i'} \leq b_i$. Our idea is to add in the envy-free statement a value $c_{ii'} M$ (where M is a sufficiently large positive constant) and enforce that $c_{ii'} = 0$ if and only if $p_{i'} \leq b_i$. To do so, we introduce binary variables $d_{ii'}$, a continuous variable $\lambda > 0$, three constraints and we set the objective function as maximising λ. The first constraint ensures that one of $c_{ii'}$ or $d_{ii'}$ has value 1 and the other 0. The remaining two constraints are about affording the payment. If i can afford the payment of i', i.e., $b_i \leq p_{i'}$, then it is not possible to have $c_{ii'} = 1$, for otherwise $d_{ii'}$ would be 0 and no value of $\lambda > 0$ would satisfy the constraint $b_i + \lambda \leq p_{i'}$. Thus, in this case $c_{ii'} = 0$, $d_{ii'} = 1$, λ is bounded by $p_{i'} - b_i + M > 0$, and the envy statement applies from agents i to i' as intended. When i cannot afford the payment of i', i.e., $b_i < p_{i'}$, then $c_{ii'}$ must be 1 to satisfy the constraint $p_{i'} - c_{ii'} M \leq b_i$. Then $d_{ii'} = 0$ and λ is bounded by $p_{i'} - b_i > 0$.

Full proof of Theorem 1

Termination and pseudo-polynomiality of Algorithm 2. Note that at each iteration of the while loop, we increase the payment of agent j in line 2 by setting p_j to $\min(b_i, p_i - \lambda_{ij})$ (payments are never decreased). Additionally, note that the payment values are always integer valued by induction. Indeed, as all input parameters are integer valued, b_i and $p_i - \lambda_{ij}$ must be integer valued if p_i is. Hence, the algorithm stops after at most $n \sum_{i \in A} b_i$ iterations, as additional iterations would lead one agent to violate its budget constraint. Last, note that all operations performed in one iteration of the while loop can be performed in polynomial time.

Correctness of Algorithm 2. Let p^0 be the vector of initial payments. We wish to compute a minimal payment vector p satisfying the following conditions:

(i) $p_i \geq p_i^0$, for every $i \in A$,

(ii) (σ, p) is WB-EF, IR, and affordable,

(iii) $\sum_{i \in A} p_i \leq L$.

We prove the correctness of Algorithm 2 by proving the following lemma.

Lemma 1. *When Algorithm 2 returns a payment vector p^*, then it satisfies (i), (ii), (iii) and is such that $p^*_i \leq p'_i$ for all $i \in A$ and for all payment vectors p' satisfying (i), and (ii). When Algorithm 2 returns no solution, then no payment vector p' satisfies (i), (ii), and (iii).*

Proof. When the algorithm returns a payment vector p^*, then it satisfies (i), (ii) and (iii) by construction (see lines 4 and 5). The rest of the proposition is proven as follows, let p^t be the payment vector at iteration t of the while loop of Algorithm 2. We prove by induction on t that any payment vector p' satisfying (i) and (ii), if any, is such that $p^t_i \leq p'_i$ for all $i \in A$. This is proven in Claim 1. This shows that when Algorithm 2 returns no solution, then no payment vector p' satisfies (i), (ii), and (iii), because this means that we obtain p^t that violates IR, the budget constraints or exceeds the rent. □

Claim 1. *At each iteration t of the while loop of Algorithm 2, if there exists an edge $(i, j) \in G_{p^t}$ with $p^t_j < p^t_i - \lambda_{ij}$, then any payment vector p' satisfying (i) and (ii), is such that $p^{t+1}_j = \min\{b_i, p^t_i - \lambda_{ij}\} \leq p'_j$.*

Proof. We will prove the claim by induction. For the basis case p^0, if there exists an edge $(i, j) \in G_{p^0}$ with $p^0_j < p^0_i - \lambda_{ij}$, then the only solutions to remove SB-envy from i to j are to increase p^0_j or to decrease p^0_i. As any payment vector satisfying (i) has $p'_i \geq p^0_i$, it must have $p'_j \geq \min\{b_i, p^0_i - \lambda_{ij}\}$ in order to satisfy (ii). Now assume that Claim 1 holds up to iteration t. Then, it implies that any payment vector p' satisfying (i) and (ii), is such that $p^t_i \leq p'_i$ for all $i \in A$. Similarly as previously, if there exists an edge $(i, j) \in G_{p^t}$ with $p^t_j < p^t_i - \lambda_{ij}$, then the only solutions to remove SB-envy from i to j are to increase p^t_j or to decrease p^t_i. As p'_i must be greater than or equal to p^t_i, it must be that $p'_j \geq \min\{b_i, p^t_i - \lambda_{ij}\}$ in order to satisfy (ii). □

Termination and polynomiality of Algorithm 3. Concerning Algorithm 3, the while loop 2-7 can only run for n iterations and each iteration runs in $\mathcal{O}(n^2)$. Indeed, we increase the payment of each agent in A' (for whom it is still possible to do it without violating IR or affordability) by the same amount q which corresponds to either the rest of the rent or the maximum that an agent in A' can still pay. At each non-final iteration of this while loop, the payment of at least one agent will be set to the maximum she can pay, and this agent will not be part of A' anymore at the next iteration. Hence, this while loop is executed at most n times.

Correctness of Algorithm 3. Note that by construction, if Algorithm 3 returns a payment vector p, then (σ, p) is B-EF, IR, and affordable and the values in p sum up to the rent. We shall now show that if the algorithm does not return a solution, then none exists.

Let p^* denote the payment vector obtained from Algorithm 2. From the correctness of Algorithm 2, we observe that the desired payment vector p should satisfy:

(i′) $p_i \geq p^*_i$ for every $i \in A$,

(ii′) (σ, p) is WB-EF, IR, and affordable.

We now show the following lemma.

Lemma 2. *Assume Algorithm 3 returns no solution and let $\hat{p}$ be the payment vector and $\hat{A}'$ the set A' when the algorithm terminates. Then $\hat{p}$ satisfies (i′) and (ii′) and is such that $p'_i \leq \hat{p}_i$ for all payment vectors p' satisfying (i′) and (ii′) and $i \in A \setminus \hat{A}'$.*

Proof. The payment vector $\hat{p}$ satisfies (i′) and (ii′) by construction. Indeed, we only increase payments (Line 6) and we do so by making sure that we do not violate any budget or individual rationality constraint nor do we create any new envy relation.

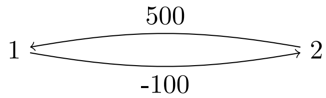
Let p' be a payment vector satisfying (i′) and (ii′). Let A'_t denote the set A' at step t. Recall that by construction, $A'_{t+1} \subseteq A'_t$. For $i \in A \setminus \hat{A}'$, let t_i be the first step of the algorithm for which agent i belongs to $A \setminus A'_{t_i}$. Let us consider agents in $A \setminus \hat{A}'$ by sorting them by increasing t_i values. By induction, we prove that for each agent i, $p'_i \leq \hat{p}_i$. It is true for 1, as she necessarily has $\hat{p}_1 = v_{1\sigma(1)}$, or $\hat{p}_1 = b_1$. Let us assume this property true for agents $1, \ldots, i-1$, then the property is true for i as when i is removed from A' we either have $\hat{p}_i = v_{i\sigma(i)}$, or $\hat{p}_i = b_i$, or $v_{i\sigma(i)} - \hat{p}_i = v_{i\sigma(j)} - \hat{p}_j$, for a value $j \in \{1, \ldots, i-1\}$ such that $p_j \leq b_i$. In this last case, we use the induction assumption to conclude. $\square$

We now conclude by investigating the two conditions for which the algorithm may not return a solution. If the algorithm terminates in Line 3, then by Lemma 2, it follows that if $\hat{p}$ cannot reach the rent, then no payment vector satisfying (i′) and (ii′) can. If the algorithm terminates in Line 8, then it means that there exist two agents i and j such that $v_{i\sigma(i)} - \hat{p}_i < v_{i\sigma(j)} - \hat{p}_j$ with $\hat{p}_j = b_i$. This may only happen if 1) $\sum_i p^*_i = L$ or if 2) $j \in A \setminus \hat{A}'$ and j has never increased her payment in Line 6. In the first case, the argument follows from the fact that increasing the payment of j, which is the only way to remove envy while respecting (i′), would exceed the rent. In the second case, necessarily $v_{i\sigma(i)} - p^*_i < v_{i\sigma(j)} - p^*_j$ and $p^*_j = \hat{p}_j = b_i$. Then increasing the payment of j, which is the only way to remove envy while respecting (i′), is not possible due to Lemma 2.

Discussion about the polynomial algorithm in [15] We explain here our doubts about Algorithm 2 in [15], which is claimed to be a polynomial-time algorithm for finding minimal payments that make the solution affordable, IR, and WB-EF, given an initial allocation. We start with the following example with two agents and two rooms, and the following valuations:

	room r_1	room r_2	budget
agent 1	400	500	450
agent 2	0	500	500

Let σ allocate r_1 to agent 1 and r_2 to agent 2. Starting with payments $p = (0, 0)$, we obtain the following envy graph G_p.



Algorithm 2 in [15] runs a while loop which is executed while there exists a negative cycle in G_p. At the end of the while loop the algorithm returns an affordable, IR, and B-EF price vector or claims that none exist. Note that in our example there is no negative cycle in G_p. Hence the algorithm would just return $p = (0, 0)$ which is not B-EF as agent 1 B-envies agent 2.

Note that in a positive cycle containing one negative arc, such as in the previous example, we can treat the negative arc by raising payments along the cycle (in a similar way as does Algorithm 1 in Kempe et al. [15]) without having to pass twice by the same arc. This may not be true with more than one negative arc in such a positive cycle.

We believe that it may be possible to give some conditions, like those established in Algorithm 2 of Kempe et al. [15], to identify the agent from which to start the treatment of negative arcs in a positive cycle, in order to avoid treating several times the same arc in the cycle. This is left for future work.

Algorithm 4:

Data: Rent division with budgets instance $\langle n, V, L, b \rangle$, and a payment vector p, definition of B_ℓ and π_ℓ as described in the proof.

1 $M \leftarrow R$; $\sigma \leftarrow \emptyset$;
2 **for** $\ell = 1$ *to* $\ell = k$ **do**
3   **for** *each agent* $i \in B_\ell$ **do**
4         $v(i) \leftarrow \max_{j \in M} v_{ij}$;
5         $top(i) \leftarrow \{j \in M : v_{ij} = v(i)\}$;
6   **if** $v(i) < \pi_\ell$ **then return** false;
7     $top(B_\ell) \leftarrow \bigcup_{i \in B_\ell} top(i)$;
8   **if** $|B_\ell| \neq |top(B_\ell)|$ **then return** false;
9   **for** *every agent* $i \in \bigcup_{\ell'=1}^{\ell-1} B_{\ell'}$ **do**
10   **if** *there exists a room* $j \in top(B_\ell)$ *such that* $v(i) - p_i < v_{ij} - \pi_\ell$ **then return** false;
11   **for** *every agent* $i \in B_\ell$ **do**
12   **if** *there exists a room* $j \in top(B_{\ell'})$ *such that* $\ell' < \ell$, $\pi_{\ell'} \leq b_i$ *and* $v(i) - p_i < v_{ij} - \pi_{\ell'}$ **then return** false;
13   Let $G = (B_\ell \cup top(B_\ell), E)$ be a bipartite graph where $(i, j) \in E$ iff $j \in top(i)$;
14   **if** *there is no perfect matching in* G **then**
15   **return** false
16   **else** $\theta \leftarrow$ a perfect matching in G;
17     $\sigma \leftarrow \sigma \cup \theta$; $M \leftarrow M \setminus top(B_\ell)$;
18 **return** (σ, p)

Full proof of Theorem 2

First, note that one can easily check if $p_i \leq b_i$ for all $i \in A$ and that $\sum_{i \in A} p_i = L$ to ensure that the payments are compatible with an affordable solution which meets the total rent. If this condition is not met, we can return that finding an affordable solution given payments p is not feasible.

To prove Theorem 2, we consider Algorithm 4, which tries to build an IR and B-EF assignment in a greedy fashion considering agents in decreasing order w.r.t. payments.

Description. In a preliminary step, we partition the set of agents into k groups $(B_1, \ldots, B_k)$, i.e., $\bigcup_{\ell=1}^{k} B_\ell = A$ and $B_\ell \cap B_{\ell'} = \emptyset$ for every $\ell \neq \ell' \in [k]$, such that for all agents $i, j \in B_\ell$, $p_i = p_j$ holds, and for all agents $i \in B_\ell$ and $j \in B_{\ell'}$ with $\ell < \ell'$, we have that $p_i > p_j$. We denote by π_ℓ the payment that is common to all agents in group B_ℓ, i.e., $\pi_\ell = p_i$ for all agents $i \in B_\ell$, and by $B(i)$ the group of agent i, i.e., $i \in B(i)$.

Then, we consider sets B_ℓ with increasing values of ℓ (hence, with decreasing values of payments), and try to assign each agent i in B_ℓ to a room in $top(i)$, the set compounded of her most preferred rooms within the remaining ones. This is done by considering a bipartite graph and determining if there exists a perfect matching as detailed in lines 13-17. If such an assignment is not possible (line 14), or if it violates an IR (line 6) or B-EF constraint (lines 10 or 12), then the algorithm returns that no affordable, IR, and B-EF solution exists.

Termination and polynomiality. Termination and polynomiality of Algorithm 4 are straightforward as the algorithm consists in an outer loop of k ($\leq n$) iterations, with inner loops of at most n^2 iterations (e.g., lines 9 and 10) and all operations realized in these loops can be performed in polynomial time. Notably, note that determining if there exists a perfect matching in a bipartite graph can be performed in polynomial time.

Correctness. Note that any solution (σ, p) returned by the algorithm is necessary affordable and meets the total rent L due to our preprocessing step. We now show the correctness of Algorithm 4 through two lemmas.

Lemma 3. *If Algorithm 4 returns a solution (σ, p), it is necessarily IR, and B-EF.*

Proof. Note that any solution (σ, p) returned by the algorithm is necessary IR due to the condition in line 6 and the fact that each agent i is matched to a room in $top(i)$.

Suppose that the allocation σ returned by the algorithm is not B-EF. Then, there exist two agents i and j such that $p_j \leq b_i$ and $v_{i\sigma(i)} - p_i < v_{i\sigma(j)} - p_j$.

- If $p_i = p_j$ (hence $B(i) = B(j)$), then agents i and j were assigned their rooms at the same step of the algorithm. Moreover, since $p_i = p_j$, we must have $v_{i\sigma(i)} < v_{i\sigma(j)}$, which means that room $\sigma(j)$ that i prefers to $\sigma(i)$ was still available at this step but i was assigned room $\sigma(i)$, which contradicts the fact that i was assigned a room in $top(i)$.
- If $p_i > p_j$, we have $B_\ell := B(i)$ and $B_{\ell'} := B(j)$ with $\ell < \ell'$. Therefore, $\sigma(i)$ is assigned to agent i before the assignment of $\sigma(j)$ to agent j. However, this would imply that at iteration ℓ' of the outer loop, the condition given in line 10 is fulfilled and thus the algorithm must have returned "false", a contradiction (note that in line 10, we must have $b_i \geq p_i > \pi_\ell$).
- If $p_i < p_j$, we have $B_\ell := B(i)$ and $B_{\ell'} := B(j)$ with $\ell > \ell'$. Therefore, $\sigma(i)$ is assigned to agent i after the assignment of $\sigma(j)$ to agent j. However, this would imply that at iteration ℓ of the outer loop, the condition given in line 12 is fulfilled and thus the algorithm must have returned "false", a contradiction.

Hence, if the algorithm returns a solution (σ, p), it is necessarily IR, and B-EF. □

Lemma 4. *If there exists an affordable IR and B-EF solution, then Algorithm 4 returns a solution (σ, p).*

Proof. Suppose that there exists a B-EF allocation σ. We will first prove by induction that, in allocation σ, each agent $i \in B_\ell$ must be assigned a room in $top(i)$, as defined in line 5 of the algorithm, where $M = R \setminus \bigcup_{\ell'=1}^{\ell-1} top(B_{\ell'})$.

Each agent i in B_1 must receive a room $\sigma(i)$ that she values the most. Otherwise there exists another agent j with $p_j \leq p_i \leq b_i$ assigned to a room r such that $v_{ir} > v_{i\sigma(i)}$, and therefore agent i would be B-envious towards j as $v_{i\sigma(i)} - p_i < v_{ir} - p_j$, a contradiction.

Assume now that the claim holds for every B_ℓ where $\ell \in [m]$ for some $m \in [k]$, and analyze the case of B_{m+1}. First of all, a room r in $\bigcup_{\ell=1}^{m} top(B_\ell)$ cannot be assigned to an agent i in B_{m+1}. Otherwise, there exists an agent $j \in B_\ell$ with $\ell \leq m$ such that $r \in top(j)$, and by induction assumption $\sigma(j) \in top(j)$. Therefore, we would have $v_{jr} = v_{j\sigma(j)}$ and thus $v_{j\sigma(j)} - p_j < v_{jr} - p_i$ with $p_i < p_j \leq b_j$, meaning that j is B-envious towards i, a contradiction. It follows that agent i must be assigned a room in $M := R \setminus \bigcup_{\ell=1}^{m} top(B_\ell)$.

If agent i is not assigned to one of her most preferred room in M, then it means that there exists a room $r \in M$ such that $v_{ir} > v_{i\sigma(i)}$, that has been assigned to an agent $j \in \bigcup_{\ell=m+1}^{k} B_\ell$ (by induction assumption, agents in $\bigcup_{\ell=1}^{m} B_\ell$ cannot receive it). It follows from $b_j \leq b_i$ that $v_{ir} - b_j > v_{i\sigma(i)} - b_i$, meaning that i is B-envious towards j, a contradiction. This concludes the proof of the claim stating that $\sigma(i) \in top(i)$ for every $i \in A$.

	R^1_1	R^1_2	$\dots$	R^1_n	R^2_1	R^2_2	$\dots$	R^2_n	R^3_1	R^3_2	$\dots$	R^3_n
A^1_1	v_1	v_n	$\dots$	v_2	S	0	$\dots$	0	S	0	$\dots$	0
A^1_2	v_2	v_1	$\dots$	v_n	0	S	$\dots$	0	0	S	$\dots$	0
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
A^1_n	v_n	v_{n-1}	$\dots$	v_1	0	0	$\dots$	S	0	0	$\dots$	S
A^2_1	$2S$	0	$\dots$	0	$\frac{1}{2}S$	0	$\dots$	0	$\frac{1}{2}S$	0	$\dots$	0
A^2_2	0	$2S$	$\dots$	0	0	$\frac{1}{2}S$	$\dots$	0	0	$\frac{1}{2}S$	$\dots$	0
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
A^2_n	0	0	$\dots$	$2S$	0	0	$\dots$	$\frac{1}{2}S$	0	0	$\dots$	$\frac{1}{2}S$
A^3_1	0	0	$\dots$	0	$\frac{3}{2}S$	0	$\dots$	0	$\frac{3}{2}S$	0	$\dots$	0
A^3_2	0	0	$\dots$	0	0	$\frac{3}{2}S$	$\dots$	0	0	$\frac{3}{2}S$	$\dots$	0
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
A^3_n	0	0	$\dots$	0	0	0	$\dots$	$\frac{3}{2}S$	0	0	$\dots$	$\frac{3}{2}S$

Table 1: The matrix used in the reduction of Theorem 3. It can be made bistochastic by dividing every cell by $3S$.

By definition, we have that $\bigcup_{\ell=1}^{k} top(B_\ell) = R$ and $top(B_\ell) \cap top(B_{\ell'}) = \emptyset$ for every $\ell \neq \ell' \in [k]$. Therefore, since $\sigma(i) \in top(i)$ for every $i \in A$ and σ is a valid B-EF allocation, $|B_\ell| = |top(B_\ell)|$ must be true. Hence, the algorithm cannot stop at line 8.

It follows from the fact that $\sigma(i) \in top(i)$ for every $i \in A$, that the utility of each agent $i \in A$ is fixed, no matter the choice of the room in $top(i)$. As payments and utilities obtained by each agent are fixed, one can check if these values induce any B-envy relation either from an agent in B_ℓ towards an agent in $B_{\ell'}$ with $\ell < \ell'$ (line 10) or vice-versa (line 12), or if they violate the IR constraints (line 6). Hence, since there exists a valid IR and B-EF solution (σ, p), the algorithm cannot stop at any of lines 6, 10, and 12.

Finally, since there exists a valid B-EF allocation σ, it means that it was possible to assign to each member i of a group B_ℓ a room in $top(i)$. It follows that the algorithm cannot stop at line 14. □

Note that Algorithm 4 can easily be adapted to find an affordable, IR, and EF (instead of B-EF) solution when p is fixed. Indeed, it is only sufficient to remove the condition $p_{\ell'} \leq b_i$ in line 12.

Full proof of Theorem 3

We show the full proof of Theorem 3: Minsum-Switch-Implementation is NP-complete.

Proof. Membership to NP is straightforward. For hardness we reduce from the NP-complete problem of Partition. Given an instance $U = \{v_1, \dots, v_n\}$ of Partition with $S = \sum_{i=1}^{n} v_i$, create a $3n \times 3n$ bi-stochastic matrix X as depicted in Table 1. We first observe that X admits an implementation in which no agent returns to the same room iff there is an implementation with $\sum_i S_i(I) = (n+2)n + 3n + 2n = n^2 + 7n$ (this can be seen by counting the number of non-zero cells in each row of X). Further, observe that for each value $v \in U$ and agent in A^i_1, there is a room $R^*(i, v) \in \{R^1_1, R^1_2, \dots, R^1_n\}$ with value v in the matrix. We call $R^*(i, v)$ the v-corresponding room for agent A^i_1. We now show that there is a solution to Partition iff X admits an implementation with $\sum_i S_i(I) \leq n^2 + 7n$, which is equivalent to the existence of an implementation where no agent returns in any room.

If there is a subset $V \subset U$ such that $\sum_{v \in V} v = 0.5S$, we provide the following implementation. For each element $v \in V$ we give to agents in $\{A^1_1, \dots, A^1_n\}$ their room corresponding to v, giving rise to $|V|$ switches for a total time of $0.5S$. Meanwhile agents $\{A_1^2, \dots, A_n^2\}$ receive rooms $\{R_1^2, \dots, R_n^2\}$ and agents $\{A_1^3, \dots, A_n^3\}$ receive rooms $\{R_1^3, \dots, R_n^3\}$. When agents $\{A_1^2, \dots, A_n^2\}$ are done with $\{R_1^2, \dots, R_n^2\}$, they exchange with agents in $\{A_1^1, \dots, A_n^1\}$. Similarly, when agents $\{A_1^3, \dots, A_n^3\}$ are done with $\{R_1^3, \dots, R_n^3\}$, they exchange with agents in $\{A_1^1, \dots, A_n^1\}$. Lastly, when agents $\{A_1^2, \dots, A_n^2\}$ are done with $\{R_1^1, \dots, R_n^1\}$, they exchange with agents in $\{A_1^1, \dots, A_n^1\}$ such that they receive rooms corresponding to values not in V. Such implementation allows the agents never to return to a room they visited and is thus a solution to Minsum-Switch-Implementation with $K = n^2 + 7n$.

Conversely, assume there is an implementation of X where agents never come back into a room they occupied. In this case, we show that agents in $\{A_1^2, \dots, A_n^2\}$ necessarily start with a room corresponding to value $0.5S$. Otherwise, after a period of $2S$ no agent would be free to exchange with her. To see this, consider that agents in $\{A_1^3, \dots, A_n^3\}$ should still should occupy their current room for some time (they have swapped at time $0.5S + S$). The same is true for agents in $\{A_1^1, \dots, A_n^1\}$. Indeed, suppose that agent A_j^1 can leave her room at time $2S$. This is only possible if she first occupied room R_j^2 and then directly switched to R_j^3 (or vice-versa). This is not possible as if A_j^1 is in R_j^2 then the occupant of R_j^3 is necessarily A_j^3, which will not be free to exchange with her at time S. Hence, agents in $\{A_1^2, \dots, A_n^2\}$ necessarily start with a room corresponding to value $0.5S$, which in turn implies that after $0.5S$ all agents need to swap their rooms. In particular, any agent in $\{A_1^2, \dots, A_n^2\}$ necessarily exchange with an agent in $\{A_1^1, \dots, A_n^1\}$. This implies that the rooms occupied by A_j^1 provide a set $V \subset U$ such that $\sum_{v \in V} v = 0.5S$, i.e., a solution to the initial Partition problem. □

Full proof of Theorem 4

We provide here the full proof of Theorem 4: Minsum-Switch-Ordering is NP-complete.

Proof. Membership to NP is straightforward. To prove hardness we present a reduction from the NP-hard Hamming Salesman Problem (HSP) [10]. An instance of HSP is a string $P = v_1 \dots v_n, L$, where $v_i \in \{0,1\}^m$, for some n and m, and L is an integer in binary representation. The question is to determine if there exists a Hamiltonian cycle over vertices v_i of total cost less than L, where the distance between two nodes is given by the Hamming distance. Note that L can be assumed less than nm.

We first show that finding a Hamiltonian path instead of a Hamiltonian cycle is also NP-Hard. Indeed, one can guess two consecutive nodes v and v' in a Hamiltonian cycle providing a solution to the former problem (one out of the $n-1$ guesses (v_i, v_n), i in $[n-1]$ must be correct). Then, there is a Hamiltonian cycle of total cost less than L iff for one guess there exists a Hamiltonian path starting in v and ending in v' of total cost less than $L - d(v, v')$. The last argument is that one can modify the instance to enforce that any two nodes v and v' are the endpoints by increasing the distances of v and v' to all other nodes by increasing the sizes of the bit strings. Formally, given v and v' and a bound T, we create two nodes $\tilde{v}$ and $\tilde{v}'$ by adding to v, T zeros and then T ones and to v', T ones and then T zeros. To all other nodes, we add $2T$ zeros. Clearly, there exists a Hamiltonian path with v and v' as endpoints and total cost lower than T iff in the modified instance, there exists a Hamiltonian path with total cost lower than $3T$.

Consider now an instance of HSP $P = v_1 \dots v_n, L$. We create an instance of Minsum-Switch-Ordering where there are $2m$ agents and $2m$ rooms. For each vertex v we create a deterministic allocation σ^v of the rooms as follows: agent i will be assigned to room i (resp. $m+i$) and agent $m+i$ will be assigned to room $m+i$ (resp. i) if the i-th bit of v is 0 (resp. 1), for all i in $[m]$. It is clear that the switch cost between σ^v and $\sigma^{v'}$ is equal to two times the Hamming distance between v and v'. Thus, there is a one-to-one correspondence between Hamiltonian paths on vertices of P and orderings of solutions σ^v. It is therefore sufficient to run MINSUM-SWITCH-ORDERING on an implementation composed of σ^v for $v \in P$ and $K = 2L$ to obtain a solution to the initial HSP instance. $\square$

Experimental design and further results

We detail in this section our experimental design to evaluate the number of additional solutions that our proposal bring with respect to classical rent division under budget. For a fixed number of agents, we generated instances of rent division under budget $\langle n, V, L, b\rangle$ using the following methodology:

Room base values: we draw base values for rooms uniformly at random between 25 and 50, then multiplied by 10. Call M_j the base value of room $j \in R$.

The rent, valuations, and budget are sampled from normal distributions where the means are based on the base values of rooms and the standard deviations are percentages $0 < \alpha < 1$ of the mean (we used $\alpha = 0.1$ for most of our experiments, see Figure 3 for a comparison with $\alpha = 0.2$).

Rent: we draw the rent as a normal distribution centered at $\sum_j M_j$ with standard deviation $\alpha \sum_j M_j$.

Individual valuations of rooms: for each $i \in A$ and $j \in R$ we draw v_{ij} from a normal distribution centered at M_j with standard deviation αM_j.

Individual budgets: for each $i \in A$ we draw the budget b_i from a normal distribution centered at $\frac{\sum_j M_j}{n}$ with standard deviation $\alpha \frac{\sum_j M_j}{n}$.

This way of generating profiles allows us to start with situations where agents have a correlated evaluation of rooms, and have a budget that is roughly one n-th of the rent to be paid. If a generated instance does not allow for an IR solution we disregard it (we assume that roommates would not consider to rent a flat together if they notice no IR solution exist). Then, we check the existence of solutions that are EF, B-EF, or their fractional counterparts. Then, we investigate whether more solutions exist when the budget increases. For fixed valuations and rent, we increased the tightness of the budget by multiplying all individual budgets by $x \in [1, 2]$. When $x = 1$ there are still a number of instances where the agents cannot afford the apartment, while at $x = 2$ we are close to rent division with unlimited budgets, that always admits an EF solution (note that however it does not always admit an EF **and** IR solution, as we do not assume that individual valuations sum to the rent, cf. Section 3.3).

Checking the existence of solutions was performed using an MIP under Gurobi [13]. We investigated the case of $n \in \{2, 3, 4, 5\}$ and we generated 1000 instances for each value of n. We provide the results for $\alpha = 0.1$ for different values of n in Figure 2, showing that the number of agents does not seem to have an impact on our results. For $x = 1$, the sum of the budget of each agent equals the rent in expectation, so the budgets are tight. We observe that using B-EF or fractional allocations increase significantly the likelihood of having a solution compared to EF. In particular, a significant number of instances that do not admit an EF solution do have a B-EF solution (for instance, when $x = 1$ in most plots there are twice more B-EF instances than EF ones). This is a particularly positive message, since B-EF solutions are deterministic, and thus easier to accept by the agents than fractional ones. When x increases, i.e. when the budgets are less tight, the difference becomes less significant.

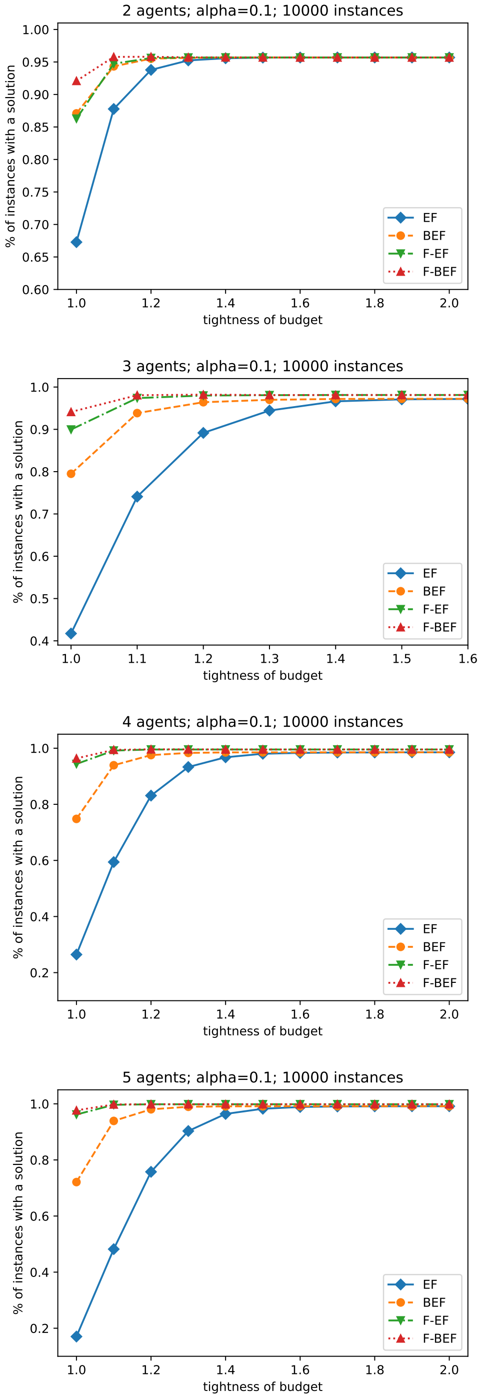
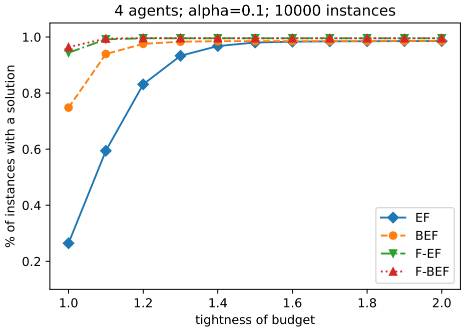


Figure 2: Frequency of existence of solutions for different values of $n = 2, 3, 4, 5$. All experiments are run with $\alpha = 0.1$ for 1000 instances.

$\alpha = 0.1$



$\alpha = 0.2$

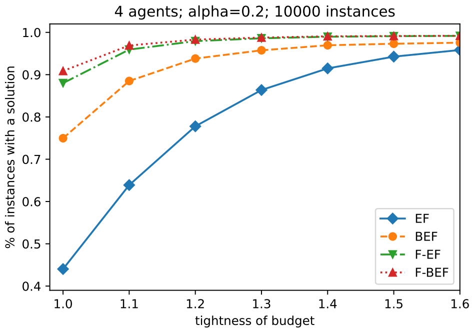


Figure 3: Frequency of existence of solutions for $n = 4$ and different values of α. All experiments are run for 1000 instances.

In Figure 3, we fixed $n = 4$ and we present results for two values of α: the values of agents for the same room are closer when α is small. Having more variations in the valuations decreases the likelihood of existence of EF solutions. Surprisingly, it also decreases the likelihood of existence of fractional solutions. Increasing α further than 0.2 flattens the four curves.