Envy-Free Allocations for Budgeted Bidders

David Kempe[1], Ahuva Mu'alem[2], and Mahyar Salek[1]

[1] Department of Computer Science, University of Southern California, CA 90089-0781, USA, {dkempe, salek}@usc.edu
[2] Social and Information Sciences Laboratory, California Institute of Technology, Pasadena, CA 91106, USA, ahumu@yahoo.com

Abstract. We study the problem of identifying prices to support a given allocation of items to bidders in an envy-free way. A bidder will envy another bidder if she would prefer to obtain the other bidder's item at the price paid by that bidder. Envy-free prices for allocations have been studied extensively; here, we focus on the impact of *budgets*: beyond their *willingness* to pay for items, bidders are also constrained by their *ability* to pay, which may be lower than their willingness.
In a recent paper, Aggarwal et al. show that a variant of the Ascending Auction finds a feasible and bidder-optimal assignment and supporting envy-free prices in polynomial time so long as the input satisfies certain non-degeneracy conditions. While this settles the problem of finding a feasible allocation, an auctioneer might sometimes also be interested in a *specific* allocation of items to bidders. We present two polynomial-time algorithms for this problem, one which finds maximal prices supporting the given allocation (if such prices exist), and another which finds minimal prices. We also prove a structural result characterizing when different allocations are supported by the same minimal price vector.

1 Introduction

One of the most central and basic economic problems is the allocation of items to individuals. This is frequently accomplished via auctions, wherein the bidders communicate their values for the items to an auctioneer, who then decides on an allocation of items to bidders and prices to be paid.

An important property of an auction is that it be *envy-free*: no bidder wishes to receive one or more items assigned to other bidders at the price the other bidders are paying. If bidders were envious in this sense, the outcome of the auction might not be stable, or bidders might refuse to participate in the auction in the future. There has been a big surge in interest in envy-free allocations and pricing of items within the computer science community recently [9, 12, 6, 2]. Much of the work focuses on the interplay between combinatorial structure among the item sets bidders are interested in and the revenue that can be extracted, usually with efficient computation.

In reality, bidders are not only constrained by their *willingness* to pay for items, but also by their *ability* to pay [5, 8]. For instance, a bidder looking for a house might have an extremely high valuation for a mansion, but nowhere near the resources to buy it at a price close to her valuation. Then, her envy will only be relevant if another bidder gets to purchase the mansion at a price *which this bidder could afford*.

Introducing budget limitations changes the problem significantly. For instance, there may now be feasible allocations which do not maximize social welfare, and efficient allocations may not be feasible any more. More generally, the structure of feasible allocations and matching prices becomes quite rich. In a recent paper, Aggarwal et al. [1] show that a variant of the Ascending Auction finds, in polynomial time, a feasible assignment and supporting *envy-free budget-friendly truthful* prices so long as the input satisfies certain non-degeneracy conditions. In fact, the allocation they find is *bidder-optimal*, in the sense that the price paid by every bidder is a lower bound on the price the bidder could pay for *any* feasible allocation and corresponding prices.

While this settles the problem of finding a feasible allocation, an auctioneer might sometimes also be interested in a *specific* allocation. For instance, there may be constraints not captured otherwise which prescribe that certain allocations are preferable from the auctioneer's point of view. Thus, an important and natural question is whether, given the bidders' valuations and budgets (as well as the auctioneer's reserve prices), a *given* allocation of items to bidders can be supported with envy-free prices.

In this paper, we give two polynomial-time algorithms for this problem, one which finds maximal envy-free prices supporting the given allocation (if such prices exist), and another which finds minimal prices. In particular, our algorithms show the existence of maximal and minimal price vectors. Both algorithms are based on label-relaxation schemes (of a dynamically constructed graph) in the style of the Bellman-Ford algorithm for shortest paths; in the case of the minimal prices, this algorithm has to be augmented by a further insight to prevent pseudo-polynomial running time. Furthermore, as a first step toward a more complete characterization of feasible allocations and the corresponding supporting envy-free budget-friendly prices, we give a combinatorial condition for minimal price vectors to be the same.

Related Work. Guruswami et al. [11] initiated the study of *envy-free revenue-maximization* for *non-budget-constrained unit-demand* bidders. If all items must be allocated, the maximum price vector can be found in polynomial time [13]. However, if some items can be omitted to increase competition, then this general problem becomes APX-hard; the current best approximation guarantee is $O(\log n)$ [11]. Multi-unit *truthful* auctions for *budget-constrained bidders* with linear valuations were first studied by Borgs et al. [4]. They constructed a truthful randomized mechanism which asymptotically achieves revenue maximization. Dobzinski et al. [8] essentially show that a deterministic truthful Pareto-optimal auction exists if and only if budgets are public information. Additionally, for the case of an infinitely-divisible single good, no anonymous truthful mechanism can produce Pareto-optimal allocations if bidders are budget-constrained [8], whereas if randomization is allowed, such mechanisms do exist [3].

2 Model and Preliminaries

We consider a set M of n distinct indivisible items, and a set N of n bidders. Bidders are *unit-demand*, i.e., each bidder is interested in purchasing at most one item. Bidder i's willingness to pay is captured by a valuation function v. Thus, bidder i has value $v_i(j)$ for item j. Additionally, each bidder has an item-specific budget $b_i(j)$, indicating her *ability* to pay for item j: the maximum amount of money the bidder can afford for this item. A particularly natural special case is when $b_i(j) = b_i$ for all j, i.e., bidder i is constrained by a fixed amount of money. However, our results hold in more generality. If $b_i(j) \leq v_i(j)$ for at least one item j, we call bidder i *budget-constrained*, otherwise, bidder i is *non-budget-constrained*. For convenience, we denote $v_i^{(0)}(j) = \min(v_i(j), b_i(j))$.

Item j will be assigned price p_j; we use $\mathbf{p}$ to denote the vector of all prices. The prices may be constrained by the auctioneer: the auctioneer has *reserve prices* $r_j \geq 0$ for items j, such that an item cannot be sold at a price less than r_j. In other words, a price vector $\mathbf{p}$ is feasible only if $\mathbf{p} \geq \mathbf{r}$. Additionally, when $p_j < b_i(j)$, we say that bidder i *can afford* item j with prices $\mathbf{p}$. (We require strict inequality for technical convenience; among other things, it makes the notion of a minimal price vector well-defined.) When assigned item j at price p_j, bidder i derives a *utility* of $u_i(j) = v_i(j) - p_j$ if $p_j < b_i(j)$ and $-\infty$ otherwise. Therefore, the utility is positive whenever $p_j < v_i^{(0)}(i)$.

In general, an *allocation* a is a partition $A_1, \ldots, A_n$ of the n items among the n bidders, where A_i is the set of items allocated to bidder i. Since we focus on unit-demand bidders, we are particularly interested in allocations that are *assignments*, in that $|A_i| = 1$ for all i, i.e., each bidder gets exactly one item. In that case, we write a_i for the unique item assigned to bidder i.

Definition 1 (Envy-Free Budget-Friendly Allocations, Supporting Prices). *An allocation* $\mathbf{a}$ *is* envy-free budget-friendly *if there exists a price vector* $\mathbf{p} \geq \mathbf{r}$ *such that for every* $i = 1, \ldots, n$*:*

1. $p_{a_i} < b_i(a_i)$ *(bidder i can afford the item allocated to her) and* $p_{a_i} \leq v_i(a_i)$ *(bidder i derives non-negative utility from her item).*
2. $v_i(a_i) - p_{a_i} \geq v_i(j) - p_j$ *for all items j with* $p_j < b_i(j)$*. That is, bidder i would not prefer another item she can afford over her own at the current prices.*

A feasible price vector $\mathbf{p}$ *satisfying these conditions is said to* support *the allocation a.*

The notion of envy-free budget-friendly allocations can be considered a generalization of a Walrasian Equilibrium [7, 10] to budget-constrained bidders.[3] Unlike the case of non-budget-constrained bidders, there need not be any envy-free budget-constrained assignments (e.g., [14]). Furthermore, even when such

[3] We are mainly interested in assignments; therefore, we do not require that any unallocated items have zero price.

assignments do exist, the efficient allocation might not be envy-free budget-friendly.

Formally, the input consists of the matrix of valuations $V = (v_i(j))_{i,j}$, the matrix of budget limits $B = (b_i(j))_{i,j}$, and an allocation $\mathbf{a}$. The goal is to identify a price vector $\mathbf{p}$ supporting $\mathbf{a}$, or to conclude that no such price vector exists.

3 Polynomial-time Algorithms

For simplicity, we assume that the desired allocation $\mathbf{a}$ is $a_i = i$ for all bidders. We then use p_i to denote the price of the item assigned to bidder i. We can also assume that $v_i^{(0)}(i) \geq r_i$; otherwise, no supporting price vector exists.

Both of our algorithms for the assignment problem are based on the notion of an *envy graph.*

Definition 2 (Envy Graph $G_{\mathbf{p}}$). *Given an arbitrary price vector* $\mathbf{p}$*, the* envy graph $G_{\mathbf{p}}$ *has one node for each bidder, and a directed edge from bidder i to bidder j if and only if $p_j < b_i(j)$, i.e., if and only if bidder i could afford bidder j's assigned item at the current prices. Whenever the edge (i, j) is present, it is labeled $\lambda_{(i,j)} = v_i(i) - v_i(j)$.*

Intuitively, the label captures how much bidder i "prefers" bidder j's item over her own, if both were priced the same. (The more negative $\lambda_{(i,j)}$ is, the more i prefers j's item.) Notice that the edge labels are *independent* of the price vector $\mathbf{p}$, and only the existence or non-existence of edges depends on the prices. The following two simple insights lie at the heart of our algorithms:

Proposition 1. *Let P be any directed path from i to j in $G_{\mathbf{p}}$, and $L = \sum_{e \in P} \lambda_e$ the sum of labels along the path.*

1. *Let $\mathbf{p}$ be any price vector such that for every price vector $\mathbf{p}'$ supporting the allocation $\mathbf{a}$, we have $\mathbf{p} \leq \mathbf{p}'$ (component-wise). Then, $p'_j \geq p_i - L$.*
2. *Let $\mathbf{p}$ be any price vector such that for every price vector $\mathbf{p}'$ supporting the allocation $\mathbf{a}$, we have $\mathbf{p} \geq \mathbf{p}'$ (component-wise). Then, $p'_i \leq p_j + L$.*

Proof. We prove the first statement — the second one is analogous. For any edge $(u, v) \in P$, envy-freeness of $\mathbf{p}'$ implies that $p'_v \geq p'_u - \lambda_{(u,v)}$. Adding the inequalities for all edges $e \in P$, and using that $p'_i \geq p_i$ now proves the claim. ∎

By setting $i = j$ in Proposition 1, we obtain the following simple corollary:

Corollary 1. *If $\mathbf{p}$ is an envy-free price vector, then $G_{\mathbf{p}}$ contains no negative cycles.*

Algorithm 1 Label Relaxation for Minimal Supporting Prices

```
Start with p_i = r_i for all i.
while there is an edge (i, j) ∈ G_p with p_i > p_j + λ_(i,j) do
    Update p_j := min(b_i(j), p_i − λ_(i,j)).
    Remove any edge (u, j) with p_j ≥ b_u(j) from G_p.
if p_i ≥ b_i(i) for any i then
    No supporting prices exist.
else
    p is a supporting price vector.
```

3.1 Finding Minimal Prices

The first part of Proposition 1 suggests a simple pseudo-polynomial algorithm for finding supporting minimal prices for an allocation (or concluding that no supporting prices exist). Algorithm 1 is a label relaxation algorithm in the style of the Bellman-Ford shortest paths algorithm.

The pseudo-polynomial running time results from negative cycles in $G_{\mathbf{p}}$. To speed up the algorithm, we will therefore choose the edge (i, j) in the `while` loop judiciously to break negative cycles fast. Let C be a negative cycle in $G_{\mathbf{p}}$, with nodes $u_1, u_2, \ldots, u_k$. Let P_{ij} denote the unique path from u_i to u_j on C, and $L_{ij} = \sum_{e \in P_{ij}} \lambda_{(u_i, u_j)}$ the total edge weight on P_{ij}. Intuitively, the update step from Algorithm 1 will have to continue until at least one of the edges (u_i, u_{i+1}) is broken, because item $i + 1$ is not affordable to bidder i any more. However, this may take pseudo-polynomial time. Our goal is to "fast-forward" the update steps along the cycle.

Lemma 1. *There exists a node u_i such that $p_{u_i} > p_{u_j} + L_{ij}$ for all j.*

Proof. Suppose for contradiction that for each i, there exists a $j(i)$ such that $p_{u_i} \leq p_{u_{j(i)}} + L_{ij(i)}$. Consider the graph on nodes u_i with an edge from u_i to $u_{j(i)}$. Because each node has an outgoing edge, this graph must contain some cycle $C' = \{u_{i_1}, \ldots, u_{i_\ell}, u_{i_{\ell+1}} = u_{i_1}\}$ such that $p_{u_{i_r}} \leq p_{u_{i_{r+1}}} + L_{i_r i_{r+1}}$ for all $1 \leq r \leq \ell$. Because each node appears once on the right and left side, after adding up these inequalities and canceling out, we obtain that $\sum_{r=1}^{\ell} L_{i_r i_{r+1}} \geq 0$. But the sum is exactly the weight of going around C one or more times (following C'), and thus negative, a contradiction.[4] ■

If we update the node prices in the order $u_2, u_3, \ldots, u_k$, it is easy to see by induction that (1) each node will need to be updated upon its turn, and (2) u_i will be updated to $p_{u_1} - L_{1i}$. Extending this observation to updates continuing around C, we can see the following:

Proposition 2. *If the algorithm has updated the prices going around C, and has updated node u_i c times, then its new price is $p'_{u_i} = p_{u_1} - cL - L_{1i} > p_{u_i}$.*

[4] An alternative proof reduces this statement to the well-known "Racetrack" puzzle. We thank Peter Winkler for this observation.

Thus, we can determine the outcome of the update process as follows: For each i, let $c_i = \lfloor \frac{b_{u_{i-1}}(u_i)-(p_{u_1}-L_{1i})}{L} \rfloor$ be the number of iterations around the cycle after which bidder u_{i-1} cannot afford item u_i any more (where $u_0 = u_k$). Then, let $j = \mathrm{argmin}_i\, c_i$, with ties broken for the smallest i. According to Proposition 2 and the definition of j, if we update each u_i (for $i \leq j$) c_i times, and each u_i for $i > j$ $c_i - 1$ times, then $p'_{u_j} > b_{u_{j-1}}(u_j)$, and $p'_{u_i} \leq b_{u_{i-1}}(u_i)$ for all $i \neq j$. In particular, this means that the updates are consistent with an execution of the relaxation algorithm.

Thus, Algorithm 2 is a polynomial-time version of Algorithm 1.

Algorithm 2 Polynomial-Time Minimal Supporting Prices

Start with $p_i = r_i$ for all i.
while $G_{\mathbf{p}}$ contains a negative cycle C **do**
    Let $u_1 \in C$ be a node satisfying Lemma 1, and $C = \{u_1, \ldots, u_k\}$.
    Compute $L_{1i} = \sum_{j=1}^{i-1} \lambda_{(u_j,u_{j+1})}$ for all i.
    Compute $c_i = \lfloor \frac{b_{u_{i-1}}(u_i)-(p_{u_1}-L_{1i})}{L} \rfloor$ for all i.
    Let $j = \mathrm{argmin}_i\, c_i$, ties broken for smallest i.
    Update $p'_{u_i} = p_{u_1} - L_{1i} - c_j L$ for $i \leq j$, and $p'_{u_i} = p_{u_1} - L_{1i} - (c_j - 1)L$ for $i > j$.
    Update $\mathbf{p} = \mathbf{p}'$, and update $G_{\mathbf{p}}$.
if $p_i \geq b_i(i)$ for any i **then**
    No supporting prices exist.
else
    $\mathbf{p}$ is a supporting price vector.

The running time in each iteration is dominated by finding a negative cycle, which can be accomplished in time $O(mn)$ by a simple extension of the Bellman-Ford algorithm. All other operations take time $O(n)$. Since each iteration of the `while` loop removes at least one edge, the total running time is at most $O(m^2 n)$.

Proposition 1 implies by induction that in each iteration, the vector $\mathbf{p}$ of the algorithm satisfies $\mathbf{p} \leq \mathbf{p}'$ (component-wise) for any price vector $\mathbf{p}'$ supporting $\mathbf{a}$. Thus, whenever Algorithm 1 outputs a price vector $\mathbf{p}$, we have that $\mathbf{p} \leq \mathbf{p}'$ for any price vector $\mathbf{p}'$ supporting $\mathbf{a}$. Because Algorithm 2 outputs the same final vector as Algorithm 1, we have proved:

Corollary 2. *If* $\mathbf{a}$ *is an envy-free budget-friendly allocation for* $V, \mathbf{b}$*, then Algorithm 1 outputs the (unique) minimal price vector* $\mathbf{p}^-$ *satisfying* $\mathbf{p}^- \leq \mathbf{p}'$ *(component-wise) for all price vectors* $\mathbf{p}'$ *supporting* $\mathbf{a}$*. In particular, there* exists *a unique minimal price vector supporting* $\mathbf{a}$.

Maximal Prices It is possible to find maximal prices supporting $\mathbf{a}$. In this case, the procedure starts with prices $p_i = v_i^{(0)}(i)$ and iteratively makes price-adjustment similar to Algorithm 1, except prices are *decreased* in response to envy. If there remains a negative cycle once the algorithm terminates, we deduce that no supporting prices exist. The algorithm can be shown to run in polynomial time even without fast-forwarding. Due to space constraints, the algorithm will be discussed in detail in the full version of this paper.

4 Affordability Graphs and Minimal Price Vectors

The structure of feasible allocations and corresponding supporting prices is much richer in the presence of budgets than for traditional envy-free auctions. If all bidders are non-budget-constrained, an allocation is feasible if and only if it is efficient (i.e., $\sum_i v_i(a_i) \geq \sum_i v_i(a_{\pi(i)})$ for any permutation π). A price vector supports either all allocations, or none of them [10]. However, once we introduce budgets, the situation changes significantly. The efficient allocation may not be feasible with budgets, while inefficient allocations are. Furthermore, there can be allocations $\mathbf{a}, \mathbf{a}'$ with corresponding supporting prices $\mathbf{p}, \mathbf{p}'$ such that $\mathbf{p}$ does not support $\mathbf{a}'$, and vice versa. As a first step toward a complete characterization, we give a combinatorial condition for *minimal* price vectors to be the same. The condition is based on the concept of an affordability graph.

Definition 3 (Affordability Graph $H_\mathbf{p}$). *The* affordability graph $H_\mathbf{p}$ *is a bipartite graph on bidders and items, containing an edge (i, j) if and only if bidder i can afford item j at the prices $\mathbf{p}$, i.e., $p_j < b_i(j)$.*

If $\mathbf{p}$ is a *minimal* price vector, $H_\mathbf{p}$ captures all of the essential information about $\mathbf{p}$, in the following sense (a generalization of Lemma 6 in [10]):

Lemma 2. *Let $\mathbf{a}, \mathbf{a}'$ be two envy-free budget-friendly assignments, and $\mathbf{p}, \mathbf{p}'$ the corresponding* minimal *supporting prices. Then $\mathbf{p} = \mathbf{p}'$ if and only if $H_\mathbf{p} = H_{\mathbf{p}'}$. Furthermore, if $H_\mathbf{p} = H_{\mathbf{p}'}$, then the social welfare of all bidders is the same under $(\mathbf{a}, \mathbf{p})$ and $(\mathbf{a}', \mathbf{p}')$, i.e., $\sum_i v_i(a_i) = \sum_i v_i(a'_i)$.*

Proof. One direction is obvious: if $\mathbf{p} = \mathbf{p}'$, then the edge (i, j) is in $H_\mathbf{p}$ if and only if it is in $H_{\mathbf{p}'}$. Hence, $H_\mathbf{p} = H_{\mathbf{p}'}$. For the converse direction, assume that $H_\mathbf{p} = H_{\mathbf{p}'}$. Because $\mathbf{a}$ is envy-free and supported by $\mathbf{p}$, each bidder prefers her own assigned item to all items she can afford, i.e.,

$$v_i(a_i) - p_{a_i} \geq v_i(j) - p_j \tag{1}$$

for every item j with $p_j < b_i(j)$. Because $\mathbf{a}'$ is an allocation, we can write $j = a'_k$ for a (unique) k in the right-hand side above, obtaining:

$$v_i(a_i) - p_{a_i} \geq v_i(a'_k) - p_{a'_k} \tag{2}$$

for each k with $p_{a'_k} < b_i(a'_k)$. Because bidder i can afford item a'_i with the price vector $\mathbf{p}'$, and the affordability graphs are the same, i can also afford a'_i with prices $\mathbf{p}$. Thus, we can apply Inequality (2) with $k = i$, to obtain that $v_i(a_i) - p_{a_i} \geq v_i(a'_i) - p_{a'_i}$. Summing this inequality over all bidders i, and noticing that both $\mathbf{a}$ and $\mathbf{a}'$ are permutations, gives us that

$$\textstyle\sum_i (v_i(a_i) - p_{a_i}) \geq \sum_i (v_i(a'_i) - p_{a'_i})$$

Adding $\sum_i p_{a_i}$ on both sides shows that $\sum_i v_i(a_i) \geq \sum_i v_i(a'_i)$. A completely symmetric argument shows the opposite inequality, so we have proved that $\sum_i v_i(a_i) = \sum_i v_i(a'_i)$.

Subtracting $\sum_i p_{a_i} = \sum_i p_{a'_i}$ on both sides implies that $\sum_i(v_i(a_i) - p_{a_i}) = \sum_i(v_i(a'_i) - p_{a'_i})$. If there were an i with $v_i(a_i) - p_{a_i} > v_i(a'_i) - p_{a'_i}$, then there would have to be some k with $v_k(a_k) - p_{a_k} < v_k(a'_k) - p_{a'_k}$, which would contradict the fact that $\mathbf{p}$ supports $\mathbf{a}$. Thus, $v_i(a_i) - p_{a_i} = v_i(a'_i) - p_{a'_i}$ for all bidders i. Combining this with Inequality (1) we get that $v_i(a'_i) - p_{a'_i} \geq v_i(j) - p_j$ for every item j with $p_j < b_i(j)$. Thus, $\mathbf{p}$ supports the assignment a', and by the minimality of $\mathbf{p}'$, we get that $\mathbf{p}' \leq \mathbf{p}$ component-wise. A symmetric argument shows that $\mathbf{p} \leq \mathbf{p}'$, and thus completes the proof. ■

Acknowledgments We would like to thank Itai Ashlagi, Liad Blumrosen, Ron Lavi, John Ledyard, Debasis Mishra and Noam Nisan for helpful discussions and comments.

References

1. Gagan Aggarwal, S. Muthukrishnan, David Pál, and Martin Pál. General auction mechanism for search advertising. In *WWW*, 2009.
2. Lior Amar, Ahuva Mu'alem, and Jochen Stoesser. On the importance of migration for fairness in online grid markets. In *GRID*, 2008.
3. Sayan Bhattacharya, Vincent Conitzer, Kamesh Munagala, and Lirong Xia. Incentive compatible budget elicitation in multi-unit auctions, working paper, 2009.
4. Jennifer T. Chayes, Christian Borgs, Nicole Immorlica, Mohammad Mahdian, and Amin Saberi. Multi-unit auctions with budget-constrained bidders. In *EC*, 2005.
5. Yeon-Koo Che and Ian Gale. Standard auctions with financially constrained bidders. *Review of Economic Studies*, 65(1):1–21, January 1998.
6. Ning Chen, Arpita Ghosh, and Sergei Vassilvitskii. Optimal envy-free pricing with metric substitutability. In *EC*, 2008.
7. Gabrielle Demange, David Gale, and Marilda Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.
8. Shahar Dobzinski, Ron Lavi, and Noam Nisan. Multi-unit auctions with budget limits. In *FOCS*, 2008.
9. Andrew V. Goldberg and Jason D. Hartline. Envy-free auctions for digital goods. In *EC*, 2003.
10. Faruk Gul and Ennio Stacchetti. Walrasian equilibrium with gross substitutes. *Journal of Economic Theory*, 87(1):95–124, 1999.
11. Venkatesan Guruswami, Jason D. Hartline, Anna R. Karlin, David Kempe, Claire Kenyon, and Frank McSherry. On profit-maximizing envy-free pricing. In *SODA*, pages 1164–1173, 2005.
12. Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *EC*, 2004.
13. Lloyd S. Shapley and Martin Shubik. The assignment game I: The core. *Journal of Game Theory*, 1(1):111–130, 1972.
14. Gerard van der Laan and Zaifu Yang. An ascending multi-item auction with financially constrained bidders. Tinbergen Institute Discussion Papers 08-017/1, Tinbergen Institute, February 2008.