We study the problem of computing optimal prices for a version of the Product-Mix auction with budget constraints. In contrast to the “standard” Product-Mix auction, the objective is to maximize revenue instead of social welfare. We prove correctness of an algorithm proposed by Paul Klemperer and DotEcon which is sufficiently efficient in smaller markets.

1 Introduction

In this paper we consider the arctic version of the Product-Mix auction which was designed by Paul Klemperer for the Icelandic’s government’s strategy for exiting capital controls (Klemperer 2010). The name “arctic” is chosen in contrast to the maybe more familiar version of the Product-Mix auction with quasi-linear utilities that can be analyzed with methods from tropical geometry (Baldwin and Klemperer 2019). In contrast to many other auction formats, it allows bidders to express budget constraints. Additionally, the supply of goods is not assumed to be fixed, but depends on the seller’s cost of providing a certain amount goods.

There are $n$ different goods $i \in \{1, \ldots, n\}$ and a finite number of bidders, each submitting exactly one bid. A bid is a vector $b = (b_1, \ldots, b_n) \in \mathbb{R}_{\geq 0}^n$ together with a budget $\beta(b) > 0$. Here, $b_i$ represents the value of bid $b$ for one unit of good $i$. We identify bidders by their unique bid $b$ and denote the set of all bids by $B$. We assume that for each good $i \in \{1, \ldots, n\}$, there is at least one bid with $b_i > 0$ – otherwise we could just exclude good $i$ from the auction.
A bundle is a vector $x = (x_1, \ldots, x_n) \in \mathbb{R}_{\geq 0}^n$ where $x_i$ denotes the possibly fractional amount of good $i$ contained in $x$. An allocation is an assignment of bundles to bidders, where $x^b$ denotes the bundle assigned to bidder $b \in B$. The auctioneer selects a linear and anonymous price vector $p = (p_1, \ldots, p_n) \in \mathbb{R}_{> 0}^n$ and determines an allocation $\{x^b\}_{b \in B}$. Note that all coordinates of the price vector are strictly positive - a price of 0 for some good would cause an infinite amount of that good to be demanded. We say that an allocation is feasible, if $\sum_{i=1}^n p_i x^b_i \leq \beta(b)$ for all bidders $b$, i.e., the cost of the assigned bundle does not exceed the available budget. A price vector $p$ together with a feasible allocation $\{x^b\}_{b \in B}$ generates the utility

$$\sum_{i=1}^n b_i x^b_i - \sum_{i=1}^n p_i x^b_i = \langle b - p, x^b \rangle$$

for bidder $b$. Given a price vector $p$, the demand set of bidder $b$ is the set of all bundles $x$ maximizing their utility:

$$D^b(p) = \arg \max \{ \langle b - p, x \rangle : x \in \mathbb{R}_{\geq 0}^n \text{ and } \langle p, x \rangle \leq \beta(b) \}.$$ 

We call an allocation $\{x^b\}_{b \in B}$ envy-free, if at given prices $p$ bidders receive bundles in their respective demand sets, i.e., $x^b \in D^b(p) \forall b \in B$.

The auctioneer must provide enough goods such that the selected allocation $\{x^b\}_{b \in B}$ can be realized. The cost of providing the required amount of goods is $\psi(\sum_{b \in B} x^b)$. We assume $\psi : \mathbb{R}_{\geq 0}^n \to \mathbb{R} \cup \{+\infty\}$ to be a lower semi-continuous convex function with $\psi(0) = 0$ which is monotonically increasing: if $x \leq y$ coordinate-wise, then $\psi(x) \leq \psi(y)$. The auctioneer’s revenue for setting prices to $p$, assigning bundles $x^b$ to the bidders and receiving their payment is then given by

$$\sum_{b \in B} \langle p, x^b \rangle - \psi\left(\sum_{b \in B} x^b\right).$$

The budget-constrained Product-Mix auction tries to balance the interests of the seller and the bidders: the auctioneer’s revenue should be maximized among all envy-free allocations:

$$\max_{b \in B} \sum_{b \in B} \langle p, x^b \rangle - \psi\left(\sum_{b \in B} x^b\right)$$

s.t. $x^b \in D^b(p) \forall b \in B$

$$p \in \mathbb{R}_{> 0}^n.$$ 

2 Contribution

Due to the non-linear constraints $x^b \in D^b(p)$, we may not expect existence of an efficient polynomial-time algorithm that finds a global optimal solution to Problem (1.1). Paul Klemperer and DotEcon proposed an algorithm with good running time behavior when the number of different goods is relatively small (they assumed $n = 3$ in their practical
application). We state and analyze this algorithm formally. Our main contribution is a proof of correctness for their algorithm.

3 The Algorithm

The basic idea of the algorithm is as follows. Suppose we fix a price vector \( p \in \mathbb{R}^n_{>0} \). In order to determine a revenue maximizing allocation at this specific price, we must solve

\[
\begin{align*}
    r(p) &= \max_{b \in \mathcal{B}} \sum_{b \in \mathcal{B}} \langle p, x^b \rangle - \psi \left( \sum_{b \in \mathcal{B}} x^b \right) \\
    \text{s.t. } x^b &\in D^b(p) \forall b \in \mathcal{B}.
\end{align*}
\]

Since the demand sets \( D^b(p) \) are convex and compact and the objective function is concave, an optimal solution to this problem exists and can be determined efficiently. Problem (1.1) can now be rewritten as \( \max_{p \in \mathbb{R}^n_{>0}} r(p) \). Our strategy is to determine a finite list of prices \( p_1, \ldots, p_m \) among which a maximizer of \( r(p) \) is guaranteed to be found. By computing \( r(p_j) \) for all these prices, we find a global optimal solution to Problem (1.1).

Algorithm 2 is based on the insight that we can restrict our search for possible maximizing prices to points where bidders are indifferent between multiple goods. For the ease of notation, we introduce a zeroth dummy good, corresponding to the empty bundle. We add an additional zeroth entry to prices \( p \in \mathbb{R}^n_{>0} \) by setting \( p_0 = 1 \) and do the same for all bids \( b \in \mathcal{B} \) by also setting \( b_0 = 1 \). For \( i \in \{1, \ldots, n\} \), we write \( e_i \) for the \( i \)-th standard vector and set \( e_0 = 0 \in \mathbb{R}^n \).

Lemma 3.2. Let \( p \in \mathbb{R}^n_{>0} \) be a price vector and \( b \in \mathcal{B} \) a bid. Then

\[
D^b(p) = \text{conv} \left\{ (\beta(b)/p_i)e_i : i \in \{0, \ldots, n\} \text{ and } b_jp_j \geq b_jp_i \forall j \in \{0, \ldots, n\} \right\}.
\]

The proof directly follows from the definition of the demand set and is omitted. We say that bidder \( b \) demands good \( i \in \{0, \ldots, n\} \) at prices \( p \), if \( (\beta(b)/p_i)e_i \in D^b(p) \). Equivalently, good \( i \) is demanded if and only if \( b_jp_j \geq b_jp_i \) for all \( j \). Given two distinct goods \( i, j \in \{0, \ldots, n\} \), we consider the region in price space where a bidder \( b \) is indifferent between these two goods, i.e., demands both of them.
Definition 3.3. A pair \((b, I)\), where \(b\) is a bid and \(I = \{i, j\} \subset \{0, \ldots, n\}\) with \(|I| = 2\) is an unordered pair of indices, induces the indifference hyperplane

\[ H(b, I) = \{p \in \mathbb{R}_>^n : b_ip_j = b_jp_i \text{ and } b_kp_k \geq b_kp_i \forall k = 0, \ldots, n\}. \]

We say that a collection \((b^1, I^1), \ldots, (b^m, I^m)\) of indifference hyperplanes is linearly independent, if their defining linear equations \(b_ip_j = b_jp_i\) are linearly independent (where we interpret \(p_0\) as the constant 1, not as a variable).

The following proposition states that we can restrict our search for candidate prices to intersections of \(n\) linearly independent indifference hyperplanes (see Figure 1).

Proposition 3.4. There is an optimal price vector \(p^*\) for Problem 1.1 which is the (necessarily unique) intersection point of \(n\) linearly independent indifference hyperplanes: \(\{p^*\} = \bigcap_{i=1}^{n} H(b^i, I^i)\).

A proof can be found in the Appendix. With Proposition 3.4 at hand, we will show that for each intersection point of \(n\) linearly independent indifference hyperplanes, there is an input for the generate candidate Algorithm 1 such that this intersection point is returned.

The next lemma provides an insight into the structure of linearly independent indifference hyperplanes.

Lemma 3.5. Let \((b^1, I^1), \ldots, (b^n, I^n)\) be \(n\) linearly independent indifference hyperplanes intersecting in a price \(p \in \mathbb{R}_>^n\). Let \(\mathcal{I} \subset \{0, \ldots, n\}\) with \(0 \in \mathcal{I}\) and \(|\mathcal{I}| < n + 1\). Then there exists some \(l\) with \(|I^l \cap \mathcal{I}| = 1\).
Algorithm 1: generate_candidate

Input: Bids $(b^1, \ldots, b^n) \in B^a$, permutation $\sigma \in S_n$.
Output: Price vector $p$, or INFEASIBLE

1. Set $C^1_1, \ldots, C^1_n = 1$;
2. for $k = 1, \ldots, n$ do
   3. Set $p_{\sigma(k)} = C^k_k b^k_{\sigma(k)}$;
   4. if there exists $m < k$ with $b^m_{\sigma(m)} p_{\sigma(k)} < b^m_{\sigma(m)} p_{\sigma(m)}$ then
      5. return INFEASIBLE;
   6. for $l = k + 1, \ldots, n$ do
      7. if $p_{\sigma(k)} < C^k_l b^l_{\sigma(k)}$ then
         8. Set $C^l_{k+1} = p_{\sigma(k)}/b^l_{\sigma(k)}$;
      9. else
         10. Set $C^l_{k+1} = C^l_k$;
   11. return $p$;

Proof. Assume this is not the case. Then for any indifference hyperplane $(b^1, I^1)$, either both indices of the variables of the defining linear equation are in $I$, or none of them. So by linear independence, there must be equations with both variables in $I^c$ and those determine the values $p_i$ for $i \in I^c$ uniquely. These equations are all of the form $b^l_j p_j = b^l_j p_i$ for $i, j \neq 0$ and some $l$. One and thus the unique solution to these equations is $p_i = 0$ for $i \in I^c$, contradicting the positivity of $p$.

The variables $C^l_k$ in Algorithm 1 have a simple intuitive meaning, as described in the next lemma. The proof only consists of inspecting steps 6-10 of the algorithm, so we omit it.

Lemma 3.6. For $1 \leq k \leq l \leq n$ we have that $C^l_k = \min_{m=0,\ldots,k-1} p_{\sigma(m)} / b^l_{\sigma(m)}$, where we set $\sigma(0) = 0$.

We are now in a position to prove our main result: if a price vector $p$ is an intersection of $n$ linearly independent indifference hyperplanes, then there is an input to the generate_candidate Algorithm 1 such that $p$ is returned.

Proposition 3.7. For each price vector $p \in \mathbb{R}_{>0}^n$ that is the intersection point of $n$ linearly independent indifference hyperplanes $(b^1, I^1), \ldots, (b^n, I^n)$, there is a reordering of the bids $(b^1, \ldots, b^n)$ and a permutation $\sigma \in S_n$ such that Algorithm 1 returns $p$.

Proof. By Lemma 3.5, we may assume without loss of generality that the indifference hyperplanes are ordered such that the following holds: $|\{0\} \cap I^1| = 1$ and for all $k > 1$, $|I^k \cap \bigcup_{i=1}^{k-1} I^i| = 1$. Set $\{\sigma(1)\} = I^1 \setminus \{0\}$ and $\{\sigma(k)\} = I^k \setminus \bigcup_{i=1}^{k-1} I^i$.

Let be $\bar{p}$ the price returned by Algorithm 1. We show by induction that in the $k$-th iteration the algorithm sets $\bar{p}_{\sigma(k)} = p_{\sigma(k)}$.

For $k = 1$, the algorithm sets $\bar{p}_{\sigma(1)} = b^1_{\sigma(1)}$. The linear equation defined by $(b^1, I^1)$ is $b^1_0 \bar{p}_{\sigma(1)} = b^1_{\sigma(1)} p_0$. Since $p_0 = b^1_0 = 1$, it follows that also $p_{\sigma(1)} = b^1_{\sigma(1)}$. 

---

5
Algorithm 2: filtered_prices

Input: Set of bids $B$
Output: Set of candidate price vectors $P$
1 Set $P = \emptyset$
2 for $(b_1, \ldots, b_n) \in B^n$ do
3     Set $S = S_n$, the set of all permutations;
4     for $\sigma \in S$ do
5         Execute generate_candidate with input $(b^1, \ldots, b^n)$ and $\sigma$;
6         if INFEASIBLE then
7             Let $k$ be the iteration in which INFEASIBLE was returned;
8             Remove all $\sigma'$ from $S$ where $\sigma'(l) = \sigma(l) \forall l \leq k$;
9         else
10            Set $P = P \cup \{p\}$;
11 return $P$;

Now let $k > 1$. The defining equation of $(b^k, I^k)$ is $b_{\sigma(k)}^k p_{\sigma(l)} = b_{\sigma(l)}^k p_{\sigma(k)}$ for some $l < k$, so $p_{\sigma(k)} = p_{\sigma(l)} b_{\sigma(k)}^k / b_{\sigma(l)}^k$. By Lemma 3.6,

$$C_k = \min_{m=0, \ldots, k-1} \frac{\tilde{p}_m}{b_{\sigma(m)}^k} = \min_{m=0, \ldots, k-1} \frac{p_{\sigma(m)}}{b_{\sigma(m)}^k}.$$

Since $p$ is contained in the indifference hyperplane generated by $(b^k, I^k)$, we have that $b_{\sigma(l)}^k / p_{\sigma(l)}$ is maximal among all quotients $b_{\sigma(m)}^k / p_{\sigma(m)}$. Consequently, $C_k = p_{\sigma(l)} / b_{\sigma(l)}^k$ which implies that the algorithm also sets $\tilde{p}_m = p_{\sigma(l)} b_{\sigma(k)}^k / b_{\sigma(l)}^k$. Note that the algorithm does not return INFEASIBLE: Since $p \in H(I^m, b^m)$ for all $m < k$ and $\sigma(m) \in I^m$, we have that $b_{\sigma(m)}^m p_{\sigma(k)} \geq b_{\sigma(k)}^m p_{\sigma(m)}$. This finishes the proof.

We can now prove our main result.

Proof of Theorem 3.1. By Proposition 3.4, there is an optimal price vector at the intersection of $n$ linearly independent indifference hyperplanes, and by Proposition 3.7, there is some input $(b^1, \ldots, b^n) \in B^n$ and $\sigma \in S_n$ for Algorithm 1 returning that price vector. Note that this input permutation $\sigma \in S_n$ has not been removed from the set $S$ before: otherwise, there would exist a permutation $\sigma'$ with $\sigma'(l) = \sigma(l) \forall l = 1, \ldots, k$ for some $k$ which caused Algorithm 1 to return INFEASIBLE in iteration $k$. This however would cause Algorithm 1 to also return INFEASIBLE for the permutation $\sigma$.

Remark. The number of all combinations of $n$-tuples of bids and all permutations is equal to $|B|^n n!$. While this grows very fast in the number of goods $n$, it is polynomial in the number of bids when the number of goods is fixed. We also may expect that due to the feasibility check in steps 4-5 of the generate_candidate algorithm, the practical runtime is reduced significantly: whenever INFEASIBLE is returned in step $k$, we remove $(n - k)!$ permutations from $S$. So in practice we may expect the algorithm to iterate over much less than $|B|^n n!$ combinations of inputs, and consequently to return the list of candidate prices reasonably fast when the number of different goods is not too large.
4 Conclusion

We have studied an algorithm proposed by Paul Klemperer and DotEcon for finding revenue maximizing prizes in the arctic Product-Mix auction and formally proved correctness of their approach. The algorithm is sufficiently fast when the number of different goods is small. Further research questions include a formal proof of the hardness of the problem, as well as the design of efficient approximation algorithms.

Acknowledgements

I would like to thank Edwin Lock from the University of Oxford for his very valuable comments and suggestions.

References

Baldwin, E., Klemperer, P., 2019. Understanding preferences: demand types, and the existence of equilibrium with indivisibilities. Econometrica 87, 867–932.

Klemperer, P., 2010. The product-mix auction: A new auction design for differentiated goods. Journal of the European Economic Association 8, 526–536.
A  Proof of Proposition 3.4

We prove Proposition 3.4 with the help of two lemmas. Let us define the set $G^b(p)$ of all goods demanded by bidder $b$ at prices $p$:

$$G^b(p) = \{i \in \{0, \ldots, n\} : (\beta(b)/p_i)e_i \in D^b(p)\} = \{i \in \{0, \ldots, n\} : b_i/p_i \text{ maximal}\}.$$ 

The first lemma states that we only need to consider prices at which every good is demanded by some bidder.

**Lemma A.1.** Let $\bar{p} \in \mathbb{R}_{>0}^n$ be arbitrary. Then there exists $p \in \mathbb{R}^n$ with $
abla b \in B \quad \forall b \in B$.

**Proof.** Let $\{\bar{x}^b\}_{b \in B}$ be an optimal allocation for $r(\bar{p})$. Then each $\bar{x}^b$ can be written as a convex combination $\bar{x}^b = \sum_{i \in G^b(\bar{p})} \lambda_i(\beta(b)/\bar{p}_i)e_i$. Since $G^b(\bar{p}) \supseteq G^b(\bar{p})$, the allocation $\{x^b\}_{b \in B}$ defined by $x^b = \sum_{i \in G^b(p)} \lambda_i(\beta(b)/p_i)e_i$ is feasible for $r(p)$, and $(\bar{p}, x^b) = (p, x^b)$ for all $b$. Since $x^b \leq \bar{x}^b$ for every bid $b$, we have by monotonicity that $\psi(\sum_{b \in B} x^b) \leq \psi(\sum_{b \in B} \bar{x}^b)$. It follows that $r(p) \geq r(\bar{p})$.

The second lemma says that if we increase prices such that the set of demanded goods get larger, the revenue does no decrease.

**Lemma A.2.** Let $\bar{p}, p \in \mathbb{R}_{>0}^n$ such that $p \geq \bar{p}$ coefficient-wise and $G^b(p) \supseteq G^b(\bar{p})$ for all $b \in B$. Then $r(p) \geq r(\bar{p})$.

**Proof of Proposition 3.4.** Let $\bar{p}$ be an arbitrary price vector. We demonstrate that there is a price vector $p^*$ which is the unique intersection point of $n$ linearly independent indifference hyperplanes with $r(p^*) \geq r(\bar{p})$. This proves the statement. By Lemma A.1 we may assume that $\bigcup_{b \in B} G^b(\bar{p}) \supseteq \{1, \ldots, n\}$. Define the set $P$ of not too small prices at which bidders demand a superset of goods they demand at $\bar{p}$:

$$P = \{ p : G^b(p) \supseteq G^b(\bar{p}) \forall b \in B \quad \text{and} \quad p_i \geq \bar{p}_i/2 \forall i \}.$$ 

$P$ is convex as it can be written as

$$P = \{ b \geq b_j \forall j \in G^b(\bar{p}) \forall j \in \{0, \ldots, n\} \quad \text{and} \quad p_i \geq \bar{p}_i/2 \forall i \}.$$
Since each good is demanded by some bid \( b \) at \( \bar{p} \), we have that \( p_i = b_0p_i \leq b_i \bar{p}_0 = b_i \) for some \( b \) and all \( p \in P \), so \( P \) is compact. Furthermore, it can be checked that for \( p^1, p^2 \in P \), their coefficient-wise maximum also lies in \( P \). It follows that \( p^* \in \mathbb{R}^n_{>0} \) defined by \( p^*_i = \sup_{p \in P} p_i \geq \bar{p}_i \) is contained in \( P \). Since \( p^* \) is the unique solution to the linear program \( \max_{p \in P} \sum_{i=1}^n p_i \) for \( n \) linearly independent of the defining inequalities of \( P \) are equalities at \( p^* \). As \( p^* \geq \bar{p} \), none of the inequalities \( p^*_i \geq \bar{p}_i / 2 \) can be tight, so there are \( n \) linearly independent equalities of the form \( b_j p^*_i = b_i p^*_j \) for some \( i \neq j \). By definition of \( P \), for each of the equalities \( b_j p^*_i = b_i p^*_j \), we have that \( b_j p^*_k \geq b_k p^*_i \) for all \( k \), so \( p^* \in H(b, \{i, j\}) \). It follows that \( p^* \) is the unique intersection point of the \( n \) linearly independent indifference hyperplanes corresponding to these equations. By definition of \( P \) we have that \( G^b(p^*) \supseteq G^b(\bar{p}) \), so \( r(p^*) \geq r(\bar{p}) \) by Lemma A.2. \qed
