Notes on Linear Programming to solve the Product-Mix Auction
Incomplete–contains gaps
Please do not quote without permission

Elizabeth Baldwin and Paul Klemperer

January 14, 2019

These notes describe how the Bank of England's original Product-Mix Auction, and its 2014 revision, can be implemented using linear programming. They also describe the (open-source) implementations of these and related Auctions that are available at `http://pma.nuff.ox.ac.uk/`.[1]

Section 1 introduces preliminaries. Section 2 describes the Bank's original Product-Mix Auction, and corresponds closely to the description in Section 2 of Klemperer (2008, 2010, 2018). Section 3 describes the main enhancement introduced in the Bank's 2014 revision (see Frost et al., 2015, and Klemperer, 2018). (These notes do not exactly correspond to details of the Bank's imple-mentations that are not publicly available.) Section 4 discusses rationing among bidders whose bids are tied. Sections 5 and 6 describe a variety of extensions and modifications of these programs.

The notes are intended for someone who knows a little about linear programming and assume familiarity with linear programming's 'primal' and 'dual' problems (although the discussions of the 'dual', below, can be skipped), but do not assume any knowledge of algorithms for solving linear programmes.

We gratefully acknowledge substantial assistance from Simon Finster; he also wrote Appendix 7.6, below, and developed the companion set of examples. Thanks also to Adam Gundry for his efficient coding of the programs we describe.

[1] Our website also contains variants of Product-Mix Auctions solved by methods other than linear programming (see Klemperer, 2018, especially Appendices IB and II, and Baldwin et al., in preparation).

1 Fixed quantity of each good

1.1 One Good case

Suppose that we only have one good. We first show that a linear programme gives exactly the same answer as the Product-Mix auction.

Suppose there are bids $i = 1, \ldots, M$, each bidding a maximum price v^i and wanting k^i units; x^i is the number each receives. Let R be the quantity available. Consider the linear programme:

$$\begin{aligned} \max_{x^1,\ldots,x^M} \sum_i v^i x^i & \\ \text{such that } x^i &\leq k^i \ \forall i \text{ [bid constraints]} \\ \text{and } \sum_{i=1}^{M} x^i &\leq R \text{ [resource constraint]} \\ \text{and } x^i &\geq 0 \ \forall i. \end{aligned}$$

The values of $x^1, \ldots, x^M$ which solve this optimisation problem are those which pick out the highest bids and give the goods to them, subject to the bid constraints and the resource constraint; this is obviously exactly the solution a one-good uniform-price auction finds.

Note also that it doesn't make any difference whether each bid comes from a distinct bidder, or if some bidders have placed multiple bids.

Maximising Gains from Trade

The linear programme also has a nice economic interpretation: it maximises the gains from trade over all participants: assume each bidder bids its maximal "willingness to pay", v^i, and that its utility is $v^i x^i - P^i$, where P^i is the payment it makes, so the auctioneer receives $\sum_i P^i$. An efficient allocation is one which maximises the joint surplus of all participants, $\sum_i (v^i x^i - P^i) + \sum_i P^i = \sum_i v^i x^i$, subject to the constraints, that is, solves our programme.

The Dual Problem

It is standard that the dual problem:

$$\begin{aligned} \min_{b^1,\ldots,b^M,z} \left[\sum_i k^i b^i + Rz \right] & \\ \text{such that } b^i + z &\geq v^i \ \forall i \\ \text{and } b^i, z &\geq 0 \ \forall i. \end{aligned}$$

yields the same value of the objective. Moreover, in the solution to the dual, z is the "shadow price" on the resource constraint in the original (primal) problem. That is, z is the marginal per-unit value of the resource. Likewise, b^i is the "shadow price" on the ith bid constraint in the original programme, that is, the marginal per-unit value of the ith bid.

It is easy to see that the value from an additional unit of resource is the highest losing bid, while the value lost from having one unit less of the resource is the lowest winning bid. In general, these are the same bid (and this bid is rationed), so z is the value of this bid. If instead the lowest winning and highest losing bids are different, a range of z (and $\{b^i\}$) solve the programme (achieve the same minimal surplus). In this event we will always choose the *lowest* feasible z. So z will always be the value, v^i, of the highest losing bid, i.

Since a bid with value v^i above z is fully allocated, $b^i = v^i - z$ is the marginal per-unit joint surplus created by a winning bid, i (that is, the per-unit joint surplus from a small increase in the size of this bid). And $b^i = 0$ for all bids with value $v^i \leq z$–these bids create no joint surplus.

Competitive Equilibrium

The allocation the original programme (and the Product-Mix auction) selects is not only efficient; it is also supported as a competitive equilibrium by the shadow price on the resource constraint, z. That is, if we were to set the price of the resource at z, all participants (both the auctioneer and the bidders) would choose exactly the allocation that the programme (and the Product-Mix auction) assigns them.

Optimal strategy is honest bidding

It follows from the previous paragraph that if a bidder is 'small' enough that its bid is unlikely to affect the Product-Mix auction's price, then its optimal strategy is to bid its maximum willingness to pay–it then always receives the good when it's worth strictly more to it than it has to pay, and never when it's worth strictly less than it has to pay.

1.2 Many Goods

It's easy to generalise the above to many goods.

Suppose we have goods $j = 1, \ldots, N$, with R_j available of good j.

Assume, w.l.o.g., that all bids are "either/or" bids across *all* the goods. (Bidding 0 for all but one good is equivalent to bidding for only that one good.) So bid i expresses per-unit values $(v_1^i, \ldots, v_N^i)$ for a total of k^i units across all the goods.

We will refer to a bid in which $v_j^i > 0$ for more than one j as a "paired" bid.

A bid's allocation of good j is $x_j^i \geq 0$, with $\sum_{j=1}^N x_j^i \leq k^i$ to ensure that it receives at most k^i units in total.

So the linear programme is now:

$$\max_{\{x_j^i\}} \sum_i \sum_j v_j^i x_j^i$$

$$\begin{aligned} \text{such that } \sum_{j=1}^{N} x_j^i &\leq k^i \ \forall i \text{ [bid constraints]} \\ \text{and } \sum_{i=1}^{M} x_j^i &\leq R_j \ \forall j \text{ [resource constraints]} \\ \text{and } x_j^i &\geq 0 \ \forall i, j. \end{aligned}$$

Exactly as for the one good case, the x_j^i which solve this programme are those that pick out the highest bids on each good – but now subject to that bid not being "either/or" with an even *more* advantageous bid on another good.

To see this, turn again to the dual problem. Let the shadow price for the ith bid constraint be b^i, and the shadow price for the jth resource constraint be z_j. So the dual programme is

$$\min_{b^1,\ldots,b^M,z_1,\ldots,z_N} \left[\sum_i k^i b^i + \sum_j R_j z_j \right]$$

$$\begin{aligned} \text{such that } b^i + z_j &\geq v_j^i \ \forall i, j \\ \text{and } b^i, z_j &\geq 0 \ \forall i, j. \end{aligned}$$

Given the prices z_j, the programme minimises b^i such that $b^i + z_j \geq v_j^i$ for all j (and $b^i \geq 0$), so b^i is the surplus the bid i attains on the good j for which $v_j^i - z_j$ is maximised (or 0 if $v_j^i - z_j < 0$ for all j). Moreover, it follows that the constraint $b^i + z_j \geq v_j^i$ only binds for this good (j), which is the most valuable good to i at the prices z_j, so, by complementary slackness, i's allocation $x_{j'}^i$ of any other good j' is zero. (Any bid for which $v_j^i - z_j < 0$ for all j receives no units of any good, and receives zero surplus.)[2] So each bid is allocated the good which gives it the greatest surplus at the shadow prices on the resource constraints, z_j.

Thus the linear programme–and the Product-Mix auction–finds the efficient allocations ($\{x_j^i\}$), and competitive equilibrium prices ($\{z_j\}$). As above, where there is a range of solutions for the shadow prices, reflecting a range of competitive equilibrium prices, we always select the lowest set of prices thus maximising all bidders' surpluses among this range.[3]

[2] If $v_j^i - z_j = v_{j'}^i - z_{j'}$ for more than one optimal j, j', then the bidder can be allocated any non-negative quantities summing to k^i, of these optimal goods.

[3] That there is a set of prices, $\{z_j\}$, that is lowest for all goods, follows from the fact that each bid is an "either/or" good for all goods (that is, every bid expresses substitute preferences) so it can never be that, for a given allocation, choosing a lower price for one good requires choosing a higher price for another.

2 Variable quantities

A Product-Mix auction give the seller flexibility over the quantity of each good available. In Klemperer (2010), the seller's preferences over the quantities to sell are summarised by a "supply curve". It is not hard to see that it is equivalent to think of the seller as having costs that are a function of the quantities of the different goods sold.[4] Many forms of seller's cost function are possible; we illustrate with a specification consistent with the Bank of England's (see Frost et al, 2015):

- Assume the goods, $j = 1, \ldots, N$, are in order of quality (so $j+1$ is a higher-quality good than j for $j = 1, \ldots, N-1$).
- For goods $j = 2, \ldots, N$.
  - Let the jth "supply curve" relate the price spread of good j above good $j-1$ to the total quantity allocated of *all* goods superior or equal to j. Since prices will be integers (e.g., an integer number of "basis points" for the Bank of England), this "supply curve" is a step function: see Figure 1
  - For good j, the qth step has length s_j^q.
  - The price increment ("margin") the auctioneer requires for good j above good $j-1$, at the qth step, is μ_j^q.
  - Let y_j^q be the amount allocated on the qth step on good j. (So $y_j^q \leq s_j^q$.)
  - So we subtract $\mu_j^q y_j^q$ from the objective function of the previous section, for all q, j.
  - Also, $\sum_q y_j^q$ is the *total* quantity allocated of all goods superior or equal to j (so denotes how far we have gone along the horizontal axis on Figure 1).
  - So $\sum_q y_j^q - \sum_q y_{j+1}^q$ is the total quantity allocated of good j alone. This is equal to the amount allocated to the bidders:

$$\sum_i x_j^i = \sum_q y_j^q - \sum_q y_{j+1}^q \text{ for all } j$$

(in which we think of $y_{N+1}^q = 0$, for all q, here[5]).

[4] A third equivalent perspective (which we develop in Section 6) is to think of the supply of each good as being equal to the maximum possible quantity available of that good under any circumstances, and to think of the seller as 'buying back' units at the appropriate relative prices to bring itself back to its "supply curve". So the seller is just another buyer, but since it knows its own preferences, its bids can be directly coded into the linear programme; its 'bids' are the values it is willing to pay for units, that is, just the costs of units in the programme we specify below.

[5] So we will write $\sum_i x_j^i = \sum_q y_j^q - \sum_q y_{j+1}^q$ for $j \leq N-1$, and $\sum_i x_N^i \leq \sum_q y_N^q$, below.

However, the constraint we use in the LP is actually an inequality: $\sum_i x_j^i \leq \sum_q y_j^q - \sum_q y_{j+1}^q$. This is because the dual problem and shadow prices behave more simply if the primal problem has inequality constraints, and we know that there will always be equality in the optimal solution, because additional ys reduce the objective function, whereas additional xs increase it. (A strict inequality would correspond to throwing away some of the resource.)[6] We call this the *good j consistency constraint*.

- Note that if we specify s_1^q such that $\sum_q s_1^q = R$, we have automatically specified a maximum total quantity constraint, R.
- Specifiying steps s_1^q with costs $\mu_1^q > 0$ for $q > 1$ automatically implements a flexible *total* quantity. A more general way to implement a flexible total quantity is to re-run the process for many possible sets of s_j^q (in particular, implying many possible maximum total quantities, $R = \sum_q s_1^q$). We will discuss below how we can use the set of outcomes generated to form a 'Total Quantity Demand Schedule'. This will have $\sum_{i,j} x_j^i$ on the horizontal axis and an appropriate increasing function of prices on the vertical axis. This schedule can then be intersected with the 'Total Quantity Supply Schedule' to give the overall solution. See section 3, below.[7]

Primal problem

So the linear programme is

$$\max_{\{x_j^i\},\{y_j^q\}} \left[\sum_{i,j} v_j^i x_j^i - \sum_{j,q} \mu_j^q y_j^q \right]$$

such that (with the variables on the right-hand side being the corresponding shadow values)

$$\begin{array}{rcll} \sum_j x_j^i & \leq & k_i \quad \forall i & b_i, \text{ constraint on size of bids} \\ y_j^q & \leq & s_j^q \quad \forall q, \forall j & u_j^q, \text{ constraint on length of steps} \\ \sum_i x_j^i - \left(\sum_q y_j^q - \sum_q y_{j+1}^q \right) & \leq & 0 \quad 1 \leq j \leq N-1 & z_j, \text{ good } j \text{ consistency constraint} \\ \sum_i x_N^i - \sum_q y_N^q & \leq & 0 & z_N, \text{ good } N \text{ consistency constraint} \end{array}$$

And all variables non-negative. We assume that $\mu_j^q \geq 0$[8] and μ_j^q is strictly increasing in q for all j.

[6] For the Bank of England, y_1^1 is an exception because $\mu_1^1 = 0$. So y_1^1 is "free", and may be "too much" in the solution, which does not matter, but we must be careful not to over-interpret the solution for y_1^1.

[7] This is the procedure currently used by the Bank of England. See Frost et al. (2015) and Klemperer (2018, Section 3.3).

[8] It would not be hard to modify what follows to allow some $\mu_j^q < 0$, but we make the assumption in the text for simplicity since it is also consistent with the Bank of England's context.

It is helpful to work with the sums of consistency constraints for $j' \geq j$ (any j):

$$\sum_{j' \geq j,\, i} x^i_{j'} \leq \sum_q y^q_j \quad 1 \leq j \leq N. \tag{1}$$

We will later (Section 2.2) collect a set of 'characteristic conditions', CC1-CC8, that together characterise the solution. The constraints on $\sum_j x^i_j$ and y^q_j, above (corresponding to the dual variables b_i and u^q_j respectively), will be the conditions CC1 and CC2, and (1) will be CC3.

Dual problem

The dual problem can be developed as before, to better interpret the shadow prices:

$$\min_{b^1,\ldots,b^M,\{z_j\}_{j=1,\ldots N},\{u^q_j\}} \left[\sum_i k^i b^i + \sum_{q,j} s^q_j u^q_j \right]$$

such that

$$\begin{array}{rcll} b^i + z_j & \geq & v^i_j & \forall i,j \quad \text{corresp } x^i_j \\ u^q_j - z_j + z_{j-1} & \geq & -\mu^q_j & \forall q, \forall j \geq 2 \quad \text{corresp } y^q_j, j \geq 2 \\ u^q_1 - z_1 & \geq & -\mu^q_1 & \quad \text{corresp } y^q_1 \end{array}$$

and all the variables b^i and u^q_j and z_j are non-negative.

Shadow prices

The shadow price on the ith bid, b^i, is the social surplus (i.e., the sum of the marginal surpluses of the bidder and the auctioneer) from filling an additional unit of bid i.[9] The shadow price, u^q_j, on using the qth step for goods $\geq j$, is the marginal value of relaxing the constraint $y^q_j \leq s^q_j$. It is the marginal value of being able to sell an additional unit of good j instead of a unit of good $j-1$ without it costing the auctioneer any more.[10]

The shadow price on the jth consistency constraint, z_j, is the marginal value of relaxing that constraint, that is, the marginal value of selling an additional

[9] [$b_i = 0$ implies *either* that bid i is not fulfilled at all ($x^i_j = 0$ for all j) *or* we are indifferent about filling the bid at the margin. In the $x^i_j \neq 0$ case, indifference means the "cost" to the auctioneer is the value, v^i_j the bidder placed on this unit of good j. (We can distinguish the cases of indifference about (i) allocating one *additional* unit against this bid (loosening k_i to $k_i + 1$ would make no difference) or (ii) allocating the *final* unit which we actually allocated (tightening k_i to $k_i - 1$ would make no difference). If these values are the same, the shadow price is uniquely defined. If not, it could be either.)]

[10] When the constraint $y^q_j \leq s^q_j$ is relaxed, the good $j-1$ consistency constraint implies that the additional sale of good j must replace a sale of good $j-1$: selling any more goods above j would require a relaxation of constraints for these goods. For example, with 3 goods, a lengthening of a binding step s^q_2 means the total number of units allocated to goods 2 and 3 is increased by 1; but since $\{s^q_3\}$ are unchanged, the total allocated to good 3 is unchanged, so it follows that the total allocated on good 2 is increased by 1, and the total allocated on good 1 is reduced by 1. A lengthening of a binding step s^q_3 means the total allocated to good 3 is increased by 1, but the total allocated to goods 2 and 3 together is unchanged, so the total allocated to good 2 is reduced by 1.

unit of additional good j.[11] So, as we discuss below, z_j is the price of good j in the auction.[12]

2.1 Complementary slackness[13]

CS 1. If $x_j^i \neq 0$, i.e. any of good j is assigned to bid i, then $b^i = v_j^i - z_j$.

CS 2. If $\sum_j x_j^i < k_i$ then $b^i = 0$.

CS 3. if $\sum_i x_j^i < \sum_q y_j^q - \sum_q y_{j+1}^q$ then $z_j = 0$, for $j \neq N$,
and if $\sum_i x_N^i < \sum_q y_N^q$ then $z_N = 0$.

Since b^i is independent of j we have, from CS1, that $v_j^i - z_j$ is independent of j for all goods j, for which $x_j^i \neq 0$. So the surplus, b^i, must be the same on all goods allocated to bid i. So since the dual constraints include $b^i \geq v_j^i - z_j$ for all j, and $b^i \geq 0$, it follows that if $v_j^i - z_j < \max_{j' \neq j}\{v_{j'}^i - z_{j'}, 0\}$, then $v_j^i - z_j < b^i$, and therefore (by the contrapositive of CS1) $x_j^i = 0$. This will form our 'characteristic condition' CC 5.

Using again the fact that the dual constraints include $b^i \geq v_j^i - z_j$ for all j, we have that if $0 < \max_j\{v_j^i - z_j\}$, then $0 < b^i$, and therefore (by the contrapositive of CS2) $\sum_j x_j^i = k^i$. This will form our 'characteristic condition' CCCC 6.

CS 4. If $y_j^q \neq 0$, i.e., if any of goods j and above are allocated on or beyond the qth step of the "supply curve", then

$$\begin{array}{rcll} u_j^q + \mu_j^q & = & z_j - z_{j-1} & \text{for } j \geq 2 \\ u_1^q + \mu_1^q & = & z_1 & \text{for } j = 1 \end{array}$$

CS 5. If $y_j^q < s_j^q$ then $u_j^q = 0$

So if any of goods j and above are allocated at all (so $y_j^1 \neq 0$), then $z_j = z_{j-1} + u_j^q + \mu_j^q \geq z_{j-1}$. So [[(since $\mu_j^q \geq 0$)]] the z_j weakly increase with j (unless we reach a stage such that all goods $\geq j$ are unallocated and irrelevant). It follows (by the contrapositive of CS3) that the consistency constraints, (1), are tight for all but a collection of lowest-ranked goods, for which $z_j = 0$; if $z_j > 0$

[11] We can see this from the fact that the simplest way to relax this constraint is to give an additional unit of good j to a bidder who is just marginal at this price. (An alternative way to relax the constraint is to sell an additional unit of a higher valued good j', which increases bidder surplus by $z_{j'}$ but decreases seller surplus by $u_{j'}^q + \mu_{j'}^q$ which together have the same effect of benefiting "society" by z_j (see CS4, below).)

[12] We discuss below dealing with the issue that z_j may not be uniquely defined. (We can also see that the price of good j is z_j, if $x_j^i \neq 0$,from CS1, below: since b^i is the social surplus, and v_j^i is the valuation, the "cost" to the auctioneer is z_j.)

[13] The beginning of this section isn't needed for understanding the program–but the examples may help the reader.

then (1) holds with equality. This will form our 'characteristic condition' CC 4.[14]

If $\mu_j^q > z_j - z_{j-1}$ with $j \geq 2$, or $\mu_1^q > z_1$, then since $u_j^q \geq 0$, we have (by the contrapositive of CS4) $y_j^q = 0$. This will form our 'characteristic condition' CC 8.

Conversely, if $\mu_j^q < z_j - z_{j-1}$ with $j \geq 2$, or $\mu_1^q < z_1$, then since the dual constraints include $u_j^q + \mu_j^q \geq z_j - z_{j-1}$, we have $u_j^q > 0$, and therefore (by the contrapositive of CS5) $y_j^q \geq s_j^q$. But $y_j^q \leq s_j^q$ is a primal constraint, so in this case $y_j^q = s_j^q$. This will form our 'characteristic condition' CC 7.

Auction Prices

It follows from the previous two results that if q is the final step on which goods j and above are allocated, then

(i) if $0 < y_j^q < s_j^q$, that is, the step is *not* fully allocated, then $z_j - z_{j-1} = \mu_j^q$ for $j \geq 2$, and $z_1 = \mu_1^q$,

(ii) if $y_j^q = s_j^q$ (and $y_j^{q+1} = 0$), so the step *is* fully allocated, then $z_j - z_{j-1} \in [\mu_j^q, \mu_j^{q+1}]$.

Since we want the auction to set prices equal to the "costs" of the final units it supplies, and μ_j^q is the auctioneer's cost of supplying a unit of good j (or higher) instead of good $j-1$, it follows that the difference between the prices the auction sets for goods j and $j-1$ should be $z_j - z_{j-1}$.

So the price of good j is just z_j, where this is uniquely defined. As before, we will select the lowest possible values $\{z_j\}$ where these are non-unique. We discuss this issue further below.

Example 1. If $\mu_1^1 = 5$, $s_1^1 = 2$; $\mu_1^2 = 10$, $s_1^2 = 2$; $\mu_1^3 = 15$, $s_1^3 > 0$; and all bids are for single units of good 1 then:

if the only bids are $12, 4, 4$, the price is $z_1 = 5$ ($y_1^1 = 1$);

if the only bids are $12, \mathbf{11}, 4$, then $z_1 \in [5, 10]$, and the price is 5 ($y_1^1 = 2$);[15]

if the only bids are $12, 11, \mathbf{8}$, then $z_1 \in [8, 10]$, and the price is 8 ($y_1^1 = 2$);

if the only bids are $12, 11, \mathbf{11}$, the price is $z_1 = 10$ ($y_1^1 = 2,\ y_1^2 = 1$).

Example 2. If, also, $\mu_2^1 = 0$, $s_2^1 = 1$; $\mu_2^2 = 50$, $s_2^2 > 0$; and there are additional bids for single units of good 2 then:

if the additional bids on good 2 are $30, 20$, then $z_2 \in [20, 30]$, and the price (on good 2) is 20 ($y_2^1 = 1$).

In this case, if also

the good-1 bids are $12, 4, 4$, then $z_1 \in [5, 10]$, and the price is 5 ($y_1^1 = 2$);

the good-1 bids are $12, \mathbf{11}, 4$, the price is $z_1 = 10$ ($y_1^1 = 2,\ y_1^2 = 1$);

the good-1 bids are $12, 11, \mathbf{8}$, the price is $z_1 = 10$ ($y_1^1 = 2,\ y_1^2 = 1$);

[14] In particular, if $\mu_1^1 > 0$ then every consistency constraint binds–CC 4 holds unconditionally. More generally, if any $\mu_j^1 > 0$ then CC 4 holds unconditionally for all $j' \geq j$.

[15] We assume here "the price" is the lowest feasible z, as discussed above. (Note that when we generalise to TQSS issues, the price might actually be a different value of z from the feasible interval.)

the good-1 bids are $12, 11, \mathbf{11}$, then $z_1 \in [10, 11]$, and the price is 10 ($y_1^1 = 2,\ y_1^2 = 2$).

Example 3. If the bid of 20 on good 2 was actually paired with the highest bid (of 12) on good 1, this would make no difference to the prices on good 1, but would yield prices of good 2 of $13, 18, 18, 18$, respectively (since the bid would have obtained surplus of $7, 2, 2, 2$, respectively, on its pair) and, correspondingly, $z_2 \in [13, 30]$ in the first case, and $z_2 \in [18, 30]$ in the remaining three cases. (The set of feasible z_1 would be unaffected, but the feasible sets of z_1 and z_2 would now be related–a higher z_1 requires a higher z_2 to prevent the paired bidder preferring good 2.)

Example 4. If, instead, the bid of 20 on good 2 had been paired with the second-highest bid $(4, 11, 11, 11)$ on good 1, this would make no difference to the prices on good 1, but would yield prices of good 2 of $20, 19, 19, 19$, respectively (since the bid would have obtained surplus of $0, 1, 1, 1$, respectively, on its pair).

Example 5. If instead $\mu_2^1 = 12$, but the bid of 20 on good 2 was not paired, this would make no difference to the prices on good 1, but would yield prices of good 2 of $20, 22, 22, 22$, respectively (that is, the latter three prices are the price of good 1 plus μ_2^1) and, correspondingly, $z_2 \in [20, 30]$ in the first case, and $z_2 \in [22, 30]$ in the remaining three cases.

If $\mu_2^1 = 12$, but the bid of 20 on good 2 was paired with the highest bid (of 12) on good 1, this would make no difference to the prices on good 1, but would yield prices of good 2 of $17, 22, 22, 22$, respectively (that is, since the paired bidder makes surplus on good-1 in the first case, all the good-2 prices are the price of good 1 plus μ_2^1) and, correspondingly, $z_2 \in [17, 30]$ in the first case, and $z_2 \in [22, 30]$ in the remaining three cases. (But if, instead, the bid of 20 on good 2 had been paired with the second-highest bid $(4, 11, 11, 11)$ on good 1, the prices of good 2 would again be $20, 22, 22, 22$, respectively, since the paired bidder would make no surplus on good-1 in the first case.)

2.2 Characterisation of the solution[16]

We have thus derived "characteristic conditions":

CC 1. $\sum_j x_j^i \leq k_i\ \forall i$.

CC 2. $y_j^q \leq s_j^q\ \forall q \forall j$.

CC 3. $\sum_{i,j' \geq j} x_{j'}^i \leq \sum_q y_j^q$. $(\forall j)$

CC 4. if $z_j > 0$ then $\sum_{i,j' \geq j} x_{j'}^i = \sum_q y_j^q$. $(\forall j)$

CC 5. If $v_j^i - z_j < \max_{j' \neq j}\{v_{j'}^i - z_{j'}, 0\}$ then $x_j^i = 0$ $(\forall i \forall j)$.

[16] This section isn't needed for understanding our program, but it shows the program solves the economic problem we want to solve.

CC 6. If $\max_j\{v^i_j - z_j\} > 0$ then $\sum_j x^i_j = k^i$ $(\forall i \forall j)$.

CC 7. If $\mu^q_j < z_j - z_{j-1}$ then $y^q_j = s^q_j$ $(\forall j$; let $z_0 \equiv 0)$.

CC 8. If $\mu^q_j > z_j - z_{j-1}$ then $y^q_j = 0$ $(\forall j$; let $z_0 \equiv 0)$.

We now show that these conditions, together with non-negativity of all variables, are necessary and sufficient both for a solution to the linear programme and a competitive equilibrium in which the $\{z_j\}$ are the prices.

Proposition 2.1. *The following are equivalent:*

1. $x^i_j, y^q_j \geq 0$ *are a solution to the primal problem, with shadow prices* $z_j \geq 0$ *on the good* j *consistency constraints.*
2. $x^i_j, y^q_j, z_j \geq 0$ *satisfy the "characteristic conditions" above.*
3. *If the bids and RSSs represent agents' actual preferences, then prices* z_j *support a competitive equilibrium, with allocation* x^i_j *to bid* i *and* $\sum_i x^i_j$ *units of good* j *sold. Let* $y^q_j = \min\left\{s^q_j, \max\left\{0, \left(\sum_{i,\, j' \geq j} x^i_{j'} - \sum_{q' < q} y^{q'}_j\right)\right\}\right\}$.

Proof. 1⇒2 follows from the previous discussion.

2⇒3. First consider a bid i: if CC 5 holds on every good, every good is too expensive for the bid, so nothing is allocated to bid i. If any goods would give the bid strictly positive surplus, then the total quantity bid for is received (CC 6), and nothing goes to goods that are not surplus-maximising (CC 5 again) so only surplus-maximising goods are received. Finally no more than the total quantity bid for is ever received (CC 1).

Now consider the seller: if $y^q_j > 0$ then CC 8$\Longrightarrow z_j \geq z_{j-1} + \mu^q_j \Longrightarrow z_j > \mu^q_j$. So the seller only sells any units of any good j for which $z_j = 0$ on a step for which $\mu^q_j = 0$. On the other hand, for goods j such that $z_j > 0$, CC 4$\Longrightarrow \sum_{i,j' \geq j} x^i_{j'} = \sum_q y^q_j$, that is, the total quantity of goods $\geq j$ sold is $\sum_q y^q_j$. And CC 2 and CC 7–CC 8 imply that the coordinates $(\sum_q y^q_j, z_j - z_{j-1})$ all lie on the curve shown in Figure 1, which curve defines the seller's preferences.

So every agent receives an allocation that maximises its surplus at the prices z_j.

3⇒1 It is a standard that a competitive equilibrium is Pareto efficient, and so maximises the objective function. So x^i_j form part of a solution to the linear programme.

For goods j with $z_j > 0$, the y^q_j are part of a solution to the linear programme, by CC 4 (indeed these are the unique y^q_j that are part of a solution). If $z_j = 0$, then the seller is indifferent about selling extra units of this good, so it must be that $y^q_j = 0$ for $q > 1$ and that $\mu^1_j = 0$, in which case any $y^1_j \leq s^1_j$ is part of a solution to the linear programme.

If the z_j are equilibrium prices, then being able to allocate an additional unit of good j at no cost (that is, relaxing the good j consistency constraint) increases total surplus by at most z_j, since no agent strictly wants to buy an additional

unit at this price. Conversely, removing such a unit (that is, tightening the good j consistency constraint) decreases total surplus by at least z_j, since no agent strictly wants to sell at this price. So the z_j are shadow prices on the good j consistency constraints in a solution to the linear programme. □

Note that we could also show 1⇒3 directly, since it is standard an allocation that maximises the objective is Pareto efficient, and so (by the second fundamental theory of welfare economics) can be achieved as a competitive equilibrium. So it then suffices to show that the shadow prices z_j support this allocation. But our 1⇒2⇒3⇒1 proof structure also shows the equivalence of 1 and 3 with the "characteristic conditions" CC 1–CC 8, which is useful later.

2.3 Non-commensurable quantities to avoid ambiguity in prices

The solution of the linear program does not necessarily imply a unique set of prices. As we discussed above, and illustrated in the examples, we want to choose the lowest prices consistent with the solution. (The structure of the problem – the fact that all bids, and supply functions, express substitutes preferences– means "the lowest prices" is well defined—see Note 3.) In fact, the set of prices will typically not be unique if the solution involves the quantity of a good exactly filling all the steps on which it is allocated, without the need to ration any of the bids to which that good is allocated. Since the program will simply choose the first solution it reaches, it is best to slightly "tweak" the problem so that the set of prices we want is the unique set consistent with the solution, and therefore the set that the program will report.

We can do this by slightly altering the length of the steps, so that they can never exactly "fit" any set of bids. Since all bids, and all steps of the supply functions are integer quantities, this is straightforward–"tweaking" the supply curves by adding $(N+1-j)\eta$ to s_j^1, where $0 < \eta < \frac{1}{2N}$, guarantees that the z_j are all uniquely determined. And where this "tweaking" resolves ambiguities, it always selects the *lowest* z_j. Moreover, slightly increasing the length of the initial step on each good in this way will only slightly change the optimal allocations of goods to bids, which can be removed by simply rounding to the nearest integer.[17]

We can show this by using the fact that the incommensurability of the steps both between different goods and with the bids means that, for any good, either a bid is rationed, or a step is not fully used on either this or the next "higher" good.

If a bid is rationed, either part of it is unfilled, in which case the price of the good must equal the price bid (so that the bidder is indifferent about being

[17] Note that the quantity changes need *not* all be increases. For example, if there is a paired bid in which both parts are exactly accepted, and must be accepted in definite proportions, this might shift the allocation from one to the other–e.g., if there is just one bid for 10 at prices (3,4), $\mu_1^1 = 0$, $s_1^1 = 5$; $\mu_1^2 = 2$, $s_1^2 = 10$; $\mu_2^1 = 0, s_2^1 = 15$, the untweaked solution is to allocate quantities (5,5) at prices (1,2), but a tweaked solution is to allocate $(5+\eta, 5-\eta)$ at the same prices.

rationed), or (if the bid is a "paired" bid) part of it may be filled on another good or goods, in which case bidder-indifference fixes the price-difference(s) between the goods.

If a step is not fully used, then "seller-indifference" fixes the price-difference(s) between the goods (that is, the marginal-cost difference, and hence the price-difference, between two adjacent goods is well-defined). So either the good's price is determined directly, or it is "tied" to another good's price, which can itself either be determined directly, or "tied" to yet another good's price, and so on; a final price is pinned down by either buyer- or seller-indifference, since the incommensurabilities mean that the total allocation cannot *both* comprise only complete bids *and* comprise only complete steps.

(Absent paired bids, the goods whose prices are "tied together" are all consecutive, but a paired bid can pin a good's price to any other good's price, so that the chains of prices formed need not be of consecutive goods.)

All the above is fairly straightforward [see Appendix 7.1 and 7.2 for details] provided there are sales on all goods.

Small dummy bids to avoid ambiguity in prices of *unsold* goods

If there are some unsold goods, their prices are not necessarily pinned down by the above–but we may still care about the prices we report on the unsold goods, especially if we use a TQSS (see Section 3 below) that might depend on these prices. Moreover, in this exceptional case we don't necessarily want to report the lowest competitive equilibrium price–we probably do not want to report prices lower than the lowest feasible sale prices, that is, prices that correspond to "reserve prices". (For example, with a single bid of 5 for 1 unit of a good whose supply curve has a single step at price 8, the competitive equilibrium is any $p \in [5, 8]$. We probably want to report $p = 8$.) To do this we include a small dummy bid of size $\eta/2$, at a high enough price that it is guaranteed to win, on every good. (So the first step of the jth supply curve has $(n+1-j)\eta/2$ in volume used by these bids, halving the "tweaks" we introduced above, without changing their effect; as with the "tweaks" we introduced above, these dummy bids, and other tweaks are removed after solving for the prices, and before the rationing procedure we discuss below.)

Alternative Pricing Rules for cases of *unsold* goods

The following example discusses alternative pricing rules that we might wish to implement when some goods are unsold.

Example 6. A seller has one unit for sale. The seller has reserve prices 5 and 7 for goods 1 and 2, respectively (i.e., the first step on the supply curve for good 2 is of height 2].There is a bid of 8 on good 1, and bids of 20 and 14 on good 2 (all for a single unit).

Clearly a single unit of good 2 is sold at price 14. If we simply program supply curves length 1 & 1, the price on good 1 will be 8. This is arguably the "correct" price to report. If the seller had any ability to sell additional good 1 (and the cost of good 2 jumps to 13 or more beyond one unit), the price would be 8.

However, if the problem is better understood as one in which the seller values good 2 at 2 more than good 1, but just wishes to sell only a single unit on this occasion, then the "correct" price to report for good 1 is probably 12. We can achieve this outcome by extending the length of the supply curves on good 2 to, e.g. length 2.

2.4 Non-commensurable prices to avoid ambiguity in quantities

Just as the solution of the linear programme does not necessarily imply a unique set of prices, nor need the optimal quantity vector be unique, since the locally flat supply curves mean the objective function may be indifferent between allocating or holding back units of goods, and between allocating on any one of several goods that are being rationed. As before, since the program will simply report the first optimum it comes to, we "tweak" the bids so that the outcome is unique and corresponds to our preferred outcome. (Again, the only possible distortion to the optimal set of prices can be removed by rounding the prices to the nearest integer.)

We assume the auctioneer has lexicographic preferences among solutions maximising the objective function, first to allocate as many units in total as possible and then, subject to this, to allocate units to goods in a fixed order of priority. (This order might be higher-quality goods first, but could be completely different from the "quality" order on goods.)

So we proceed by "tweaking" the bids by adding $\epsilon^{(a_j+1)}$ to all bids on the jth good, in which the $a_j \in \{1, ..., N\}$are such that j is the a_jth preference good (with no "ties" between goods), and $\epsilon \leq 1/2$ (assuming all bids are integers,

e.g., whole numbers of basis points).[18] That is, the objective function becomes

$$\max_{\mathbf{x}^1,\ldots,\mathbf{x}^M,\{y_j^q\}} \left[\sum_{i,j} (v_j^i + \epsilon^{(a_j+1)}) x_j^i - \sum_{j,q} \mu_j^q y_j^q \right]$$

with the constraints being unchanged.

The solution $\mathbf{x}^1, \ldots, \mathbf{x}^M, \{y_j^q\}$ to this programme, with the shadow prices rounded to the nearest integer, gives us the prices and the values of $\sum_i x_j^i$ for all j, that we want. It is still necessary, of course, to specify the rationing among marginally accepted bids (i.e., to specify how the total quantities of each good, $\sum_i x_j^i$, are to be allocated across the winning bids).

3 The Total Quantity Supply Schedule ("TQSS")

The previous arguments have shown that "tweaks" guarantee an unique price and quantity solution (up to fairly allocating between rationed bidders)

[18] The reason for choosing $\epsilon^{(a_j+1)}$ is that a good's price may end up being tweaked by the sum of the tweaks on other goods. (e.g., with 1 unit of Y and 2 units of X to be sold, assume A bids for 2 at (2,2) and B bids for 2 at (1,0). Equilibrium is at (1,1) with A being rationed between Y and X, and B being rationed between X and 0. If we tweak A's bid to (2.1,2), equilibrium shifts to (1,0.9).) [Note also, that tweaking bids on good X up sends good Y's price down–so to remove the tweaks, we must round to the nearest integer (up or down).]

Using tweaks of this form means that solving our linear program for large numbers of goods may require increased arithmetic precision, as this approach requires at least one binary digit of precision per good (plus one) to accurately represent tweaked prices–it is possible to solve the LP in higher-precision or arbitrary-precision arithmetic using the techniques described in, e.g., "Exact solutions to linear programming problems" (Applegate et al., 2007). Alternatively we can "tweak" a single good at a time using a method analogous to that used in Baldwin et al (fortcoming); this avoids the need for increased arithmetic precision, but is considerably more cumbersome.

In section 6's extension to "asymmetric paired bids", a more careful choice of ϵ is needed to guarantee uniqueness of the prices and the values of $\sum_i x_j^i$ for all j.

On the other hand, absent "paired bids", it suffices to "tweak" the bids by adding $a_j\epsilon$ to all bids on the jth good, in which the $a_j \in \{1, ..., N\}$ are such that j is the $(N + 1 - a_j)$th preference good (so the good we most prefer to allocate to has $a_j = N$, and the good we allocate to last has $a_j = 1$), and $\epsilon < \frac{1}{2N}$ (assuming all bids are integers), and we than always round *down* to the nearest integer–see the Appendix.

[With "paired bids", we can construct counterexamples to uniqueness of $\{y_j^q\}$ and $\sum_i x_j^i$ for all j, if we simply "tweak" the bids by adding $a_j\epsilon$ to all bids on the jth good, with $a_j \in \{1, ..., N\}$ are such that j is the $(N+1-a_j)$th preference good. For example, let $a_j = j$; let there be bids of $(a, 0, c, 0)$ and $(0, b, 0, d)$, each for a single unit; let the seller's supply functions both (i) exhibit indifference between selling a unit of good 1 at price A and selling a unit of good 2 at price B, and also (ii) exhibit indifference between selling a unit of good 3 at price C and selling a unit of good 4 at price D. If $a \geq A$, $b \geq B$, $c \geq C$, $d \geq D$ and $(a - A) + (d - D) = (b - B) + (c - C)$, there is a solution in which the quantities sold are (1,0,0,1), and also a solution in which the quantities sold are (0,1,1,0), both with and without the tweaks. (E.g., $s_1^1 = s_2^1 = 2; s_3^1 = s_4^1 = 1$; $\mu_1^1 = \mu_2^1 = \mu_3^1 = \mu_4^1 = 1$; bids: 1 at (2,0,103,0) and 1 at (0,3,0,104). So goods cost seller1,2,3,4 respectively, and the seller can sell at most 1 of goods 3 and 4. So the seller will either sell goods 2+3 or goods 1+4, and with tweaks of 1/10, 2/10, 3/10, 4/10, both sale options yield a total tweak of 5/10.)]

when the "auction size", $\sum_q s_1^q = R$, is fixed. It follows that the competitive equilibrium is also unique when R is fixed.

In fact, the "auction size", R, will also be determined by the bidding. Moreover, as we vary R, we will scale all the s_1^q in proportion to R, and scale all the s_j^q, $j > 1$, in proportion to $\lambda\widetilde{R} + (1-\lambda)R$, for some minimum size $\widetilde{R}$, and $\lambda \in [0,1]$. (We thus encompass the cases (i) for $j > 1$ leave all the s_j^q unaffected, *and* (ii) scale all the s_j^q in proportion to R.) Scaling in this way guarantees that all goods' prices, $\{z_j\}$, are weakly decreasing in R–this is a special case of the following result:

Proposition: Multiplying all the s_j^q by β_j, where $\beta_j \geq \beta_{j+1}$ for all $j < N$, and $\beta_N \geq 1$, weakly reduces all the prices, $\{z_j\}$.

Proof: Let the total quantity of good j allocated to the bidders at a solution to our program for some $R, \{s_j^q\}$, be $X_j = \sum_i x_j^i$, and the prices be $\{z_j\}$. Let $Y_j = \sum_q y_j^q$, so in the solution $Y_j = X_j + X_{j+1} + ...+ X_N$. (We noted above that $j = 1$ is an exception to the rule that compatibility constraints bind in the solution–but we can think of the Y_1 that satisfies this equality as being the "correct" Y_1, so this is w.l.o.g.). So $X_j = Y_j - Y_{j+1}$ for $j < N$, and $X_N = Y_N$. Now consider multiplying all the s_j^q by β_j, where $\beta_j \geq \beta_{j+1}$ for all $j < N$, and $\beta_N \geq 1$. At the prices, $\{z_j\}$, of the initial solution, the supply conditions would be satisfied by quantities $\left\{\widetilde{X}_j\right\}$ that satisfy $\beta_j Y_j = \widetilde{X}_j + \widetilde{X}_{j+1} + ...+ \widetilde{X}_N$, so $\widetilde{X}_j = \beta_j Y_j - \beta_{j+1} Y_{j+1}$ for $j < N$, and $\widetilde{X}_N = \beta_N Y_N$. That is, they would be satisfied by $\left\{\widetilde{X}_j\right\}$ such that $\widetilde{X}_j - X_j = (\beta_j - 1)Y_j - (\beta_{j+1} - 1)Y_{j+1} \geq 0$ for $j < N$, and $\widetilde{X}_N - X_N = (\beta_N - 1)Y_N$. So, since $\beta_j \geq \beta_{j+1}$ and $Y_j \geq Y_{j+1}$, excess supply, $\widetilde{X}_j - X_j$, is weakly positive for every good j. So since both the new supply curves and bids represent ordinary substitute preferences, it follows from Theorem 23 of Milgrom and Strulovici (2009) that there exists an equilibrium at which all prices are weakly lower than $\{z_j\}$. □ [19]

Since all goods' prices, $\{z_j\}$, are weakly decreasing in R, a unique equilibrium R can be determined by the program using a Total Quantity Supply Schedule ("TQSS"), $\tau(z_1, ..., z_N)$, if τ is a continuous strictly increasing function of all the $\{z_j\}$. That is, we can find the unique R which yields $\{z_j\}$ such that $\tau(z_1, ..., z_N) = R$ by simply using a "bisection method", running the linear

[19] It is straightforward that the argument extends to any weakly increasing supply curves (since the argument doesn't rely in any way on the step form of supply curves we have imposed).

The argument also extends to the more general programs of Sections 5. For program (2) (which is more general than program (1)), using the obvious notation, $Y_{j,l} = X_{j,l} + X_{j+1,l} + ...+ X_{N_l,l}$ for all $l > 1$, and $Y_{1,1} = X_{1,1} + \sum_{j,l>1} X_{j,l} = X_{1,1} + \sum_{1,l>1} Y_{1,l}$ (more precisely, $Y_{1,1}$ can be thought of as equal to this), in the initial solution. So, after rescaling all the $s_{j,l}^q$ by $\beta_{j,l}$, we have weakly excess supply of good j, l at the initial prices (i.e., $\widetilde{X}_{j,l} - X_{j,l} \geq 0$), if $(\beta_{j,l} - 1)Y_{j,l} - (\beta_{j+1,l} - 1)Y_{j+1,l} \geq 0$ for all $l > 1$, $j < N_l$, and $(\beta_{N_l,l} - 1)Y_{N_l,l} \geq 0$ for all $l > 1$, and $(\beta_{1,1} - 1)Y_{1,1} - \sum_{l>1}(\beta_{1,l} - 1)Y_{1,l} \geq 0$. So sufficient conditions for our monotonicity result are that $\beta_{j,l} \geq \beta_{j+1,l}$ for all $l > 1$, $j < N_l$, and $\beta_{N_l,l} \geq 1$ for all $l > 1$, and $\beta_{1,1} \geq \max_l\{\beta_{1,l}\}$ (since $Y_{1,1} \geq \sum_{l>1} Y_{1,l}$).

It also extends to permitting asymmetric paired bids–see Section 6 (since the argument works for ordinary as well as strong substitutes).

programme for sufficient fixed R to achieve any desired accuracy.[20] We will program a flexible choice of τ, subject to either $\tau(z_1, ..., z_N) = \widehat{\tau}(z_1 + ... + z_N)$, or $\tau(z_1, ..., z_N) = \widehat{\tau}(z_j)$ for some single j.

It may help to think of graphs of τ and R (on the y-axis), and ψ (on the x-axis), where $\psi(z_1, ..., z_N) = (z_1 + ... + z_N)/N$ if $\tau(z_1, ..., z_N) = \widehat{\tau}(z_1 + ... + z_N)$, and $\psi(z_1, ..., z_N) = z_j$ if $\tau(z_1, ..., z_N) = \widehat{\tau}(z_j)$. Solving the program obtains the $\{z_j\}$, and hence obtains ψ as a downward-sloping function of R, and the TQSS specifies τ as an upward-sloping function of ψ.

We will draw these graphs (approximating $\psi(R)$ if drawing the entire graph of $\psi(R)$ accurately is computationally intensive).

TQSS tweaking

Although all supply curve steps will be specified in integers, scaling them as above may mean they are no longer integers. So tweaking the step lengths could recreate integers. To avoid this (small) risk, we round up steps up to the nearest integer after the scaling but before the "$\eta-$tweaking" we described above. (More precisely if, e.g., step lengths on a supply curve are now 1.2, 3.6, 3.6, 2.4, yielding partial sums are 1.2, 4.8, 8.4, 10.8, we round these up to 2, 5, 9, 11, and thus round the step lengths to 2, 3, 4, 2.)

Alternative TQSS

Section 5 describes an alternative form of TQSS.

4 Rationing

We assume we have done the earlier tweaks, so that we have now determined a unique price and quantity for each good.

So if and when we run the linear programme again to sort out the rationing, we can use a simplified version which fixes the the total quantity of each good to be allocated at the quantity we have determined. That is, we now replace our program with the program in Section 1.2, with the amount of the jth good to be allocated being fixed at R_j (and dispensing with the supply functions

[20] An "intersection" is guaranteed by choosing a TQSS covering a sufficiently wide range of $\{z_i\}$ for relevant R. (There may be an interval of R yielding the same set of $\{z_i\}$).

If τ is discontinuous (but increasing), there may be an interval of R at which R equals the TQSS.

One might prefer to use an alternative form of TQSS, and/or to permit a different dependence of the s_j^q on R. If it remains guaranteed that the $\{z_j\}$, are decreasing in R,and the TQSS is increasing in the $\{z_j\}$, we can use "bisection" to find a (unique) equilibrium. If the monotonicity conditions are not satisfied, but we are not too fussed about achieving exact equality between R and the TQSS, we might run the linear programme for a finite number of fixed $R \in [\underline{R}, \overline{R}]$ and choose either the largest of these Rs which is no more than the value of the TQSS that using that R yields, or the smallest of these Rs which is no less than its corresponding value of the TQSS. For example, the monotonicity conditions are *not* satisfied by a TQSS that is a function of prices *weighted* by the quantities allocated at them. (E.g., with just two goods, increasing R might result only in selling more of the higher-priced good to a single bid that is indifferent, and so rationed, between that good and not being allocated, while neither price changes, thus increasing the quantity-weighted average price.)

and the $\{y_j^q\}$ etc.). This not only makes for a more efficient program, but also, importantly, means we can use larger "tweaks" for rationing purposes than we would otherwise be able to use without interfering with our solution.[21]

We need to determine how to divide the quantity allocated to a good between the different bids:

For example, assume the prices on goods 1 and 2 are 11 and 30, respectively, and we can only fill 2 of $m > 2$ bids of 11 (for single units of good 1), and 1 of $n > 1$ bids of 30 (for single units of good 2).[22]

We will call a bid "multiply-marginal" if it is a "paired" bid in which more than one good is being rationed (where "reject" is not counted as a "good"). (So, in the above example, a bid of 11 on good 1 and 30 on good 2 is "multiply-marginal"; a bid of 11 on good 1 and 20 on good 2 is not.) *Note there may be multiply-marginal bids, where the marginal parts are (all) above the allocation prices.* For example, a bid of 21 on good 1 and 40 on good 2 is "multiply-marginal". So *we have to search through all the multiply-marginal bids*–we can't just look at bids which actually have values of 30 and 11.

In our basic problem, a bid can't be marginal and above the allocation prices unless the bid is "multiply-marginal".[23]

We will focus on the case where there are at least some "multiply-marginal" bids. (So we can't simply use the multiply-marginal bid in both places. E.g., if $m = 3$ and $n = 2$, we would be trying to allocate 2/3 of an 11 and 1/2 of a 30 which is too much.)

4.1 Objectives

1. Lexicographic preferences for allocating more rather than less.

We argue, of course, that if bidders bid "honestly" then they should be indifferent about whether or not they are allocated any marginal bid, and that they should bid "honestly" if the number of bidders is large. But with any finite number of bidders, they should rationally shave their bids below their valuations, so will strictly prefer to be allocated rather than not allocated. (Roughly, the more any bidder bids for relative to total expected demand the more it should shave its bid.) Moreover, if we do a discriminatory-pricing version of the auction,

[21] Running the simpler programme won't ever lead to an essentially different set of prices, because our rationing will yield a unique "highest-value allocation" of the goods to the bidders, and it can't be that different sets of prices would lead to different highest-value allocations (because the efficient allocation across bidders can't depend on prices which are, of course, only a transfer). That is, it may be that prices are non-unique, again, but the prices we originally found will be consistent with any solution of the simpler programme. And since we want the minimal possible prices, we will simply go back and use the prices we originally found, after having sorted out the rationing.

[22] This example is similar to the last (4th) case of example (2) above. It had $\mu_1^1 = 5$, $s_1^1 = 2$; $\mu_1^2 = 10$, $s_1^2 = 2$; $\mu_1^3 = 15$; $\mu_2^1 = 0$, $s_2^1 = 1$; $\mu_2^2 = 50$; bids for single units of good 1 were $12, 11, 11$, and for single units of good 2 were $30, 20$. Modify this so that there are m bids of 11, and n bids of 30.

[23] A single (unpaired) bid above the allocation price *can* be marginal if, as in extensions of the model we discuss in section 6 below, we permit additional constraints, such as a constraint on the total quantity a bidder can buy.

bidders will have strong preferences that their marginal bids be allocated rather than unallocated.

2. Not arbitrary–although we could argue arbitrariness is random, it would lead to some awkwardness in the real world if it really depends upon the fine details of the programme and the way bids are input.

3. Fairness between bids. We interpret this to mean that bids should, ceteris paribus, be rationed in proportion to their size. (That is, if one bid that is twice the quantity of another, should receive twice the allocation. Otherwise participants may have incentive to subdivide large bids into smaller ones.) Note we are not trying to allocate bids to bidders, and then do fairness between bidders.

4. Favor multiply-marginal bids.[24] (We want to encourage paired bids, since they create more competition between bidders on different goods (they link the otherwise-separate auctions) and make collusion harder. But we can't favour all paired bids, because bidders could then put in rubbish as one part of the bid, in the knowledge that that part will never be chosen, and they get a benefit on the part that is serious.)

Possibly favour triply-marginal (in which 3 components are marginal) over doubly-marginal etc.

Possibly favour higher bidders among multiply-marginal [see later example].

5. Not divide bids into little bits (especially multiply-marginal bids) [this isn't necessarily consistent with "fairness", of course]

6. Not take too much computing time (so as not to limit number of bids and goods etc.). Especially if we are using a TQSS this may not be such an issue, since we only have to workout the rationing between bids at the very final stage after we have fixed the final price and quantity vectors

4.2 Strategy

There are a variety of alternatives approaches, of which the following seems most promising (alternatives A,B, and C are in Appendix 7.5):

D. We identify all marginal bids or parts of bids. We then tweak them to "smear them out" as follows. Choose $\delta < 1/N$. Convert each marginal bid, or

[24] If we do this, it will probably mean (and so we will need to explain to bidders)

(i) that a paired bid both parts of which are marginal has a higher chance of being filled, since it may be the only one allocated a good, and others may see that they got none of the good even though they bid the price at which the good was allocated,

BUT (ii) if a paired bidder has lexicographic preferences between the different sides of its pair, it may feel it got the wrong side of its bid, and was hurt by making a paired bid. The reason is that the auctioneer will fill bids in the order the auctioneer wants, even though both sides of the pair might have been allocated in full in the end. So the paired bidder may be the only one allocated his less-preferred good.

part of a bid, for k_i units at value v_j^i, to a demand that is linear and is for k_i units at value v_j^i, decreasing to 0 units at value $v_j^i + j\delta$ (so the different goods $j = 1, ..., N$ are tweaked by different amounts). Now approximate these linear demands as finely as desired by a set of "little steps", and then run the linear programme. This will be "efficient" (up to the size of "little step"), and it is completely fair among bidders.[25]

We might save computing timing by doing this in several phases. We start with "medium-size steps". Having identified an approximate solution, we turn the relevant medium-size steps into "little steps" and run the program again....

We should go to some appropriate minimum size, then check that all identical bids are treated identically, by rounding final outcomes down if necessary.[26]

Variants of D

(i) We just do D for multiply-marginal bids. We then go through the goods in turn, rationing all the other marginal bids equally (which is easy since they are all "singly-marginal"). This *both* favours multiply-marginal bids *and* should save computing timing. Indeed it is likely to be fast since the number of multiply-marginal bids is likely to be tiny.

---

[25] The reason this achieves an (essentially) unique solution is that we are breaking up marginal bids into lots of (arbitrarily) tiny bids which cannot be collinear with any "facet" (line of prices along which the original bid is indifferent). So only a finite number of (arbitrarily) tiny bids can end up marginal.

(These tweaks cannot, of course, bias which goods are allocated because that is predetermined by the ϵ tweaks. The fact higher-indexed goods are given higher values is compensated by the fact that we will [in effect] round prices back down [by going back to the original prices].)

These tweaks are "fair" in, e.g., examples such as (in 2 dimensions): (1) if there is a marginal paired bid for 10 on each of the horizontal, vertical, and diagonal indifference lines through the equilibrium prices [e.g. the bids are (0,30), (21,40), and (11,0), with equilibrium prices (11,30)], and we must sell 10 of each of the 2 goods [sell 20 in all, so reject 10] this takes the "top half "of each of the 3 "linearised bids" to get the maximum value; (2) if there are two (multiply-)marginal paireds bid for 10 on the diagonal indifference line through the equilibrium prices [e.g. the bids are (21,40), and (31,50), with equilibrium prices (11,30)], and we must sell 10 of each of the 2 goods [sell 20 in all, so reject 0] this takes the "top half "of each of the 2 "linearised bids" to get the maximum value.

These tweaks do, however, mean higher-indexed goods are distributed "more equally" among the multiply-marginal bids, in asymmetric problems. (The reason is that it is most "profitable" to allocate to the favorably-tweaked parts of the bids on higher-indexed goods, even if that requires allocating to some less favorably-tweaked parts of bids on lower-indexed goods.) If, for example, we need to allocate 2 apples and 1 banana to bids 1 and 2, and these bids must receive 1 unit and 2 units, respectively, then if 1 receives x and $1 - x$ units of A and B, respectively, 2 must receive $2 - x$ and x units of A and B, respectively. If we tweak A and B by $4w\delta$ and $4v\delta$, respectively, the "additional profit" is $[x(4-x)+(2-x)(2+x)]w\delta+[(1-x)(2+2x)+x(4-2x)]v\delta = [2+2x-x^2]2w\delta+[1+2x-2x^2]2v\delta$. So (since the LP chooses x to maximise the "additional profit"), if $w/v \to \infty$, $x \to 1$ (i.e., apples equally shared, and all bananas allocated to bid 2), if $w/v \to 0$, $x \to 1/2$ (i.e., bananas equally shared, and apples disproportionately allocated to bid 2), and if $w \approx v$, $x \approx 2/3$ (i.e., apples and bananas shared in $\approx$ the same ratio, 1:2, between the two bids).

[26] We would not necessarily deem bids to be identical if the bidders who made them (i) were under additional constraints, such as constraints on the total quantity a bidder can buy, *and also* (ii) had made different sets of bids. Nor do we assume bids are identical in any sense, if one is a multiple of another (though in an ideal world it might be natural to argue, e.g., that a bid that is exactly twice another bid should have exactly twice the allocation).

(Again, we check that all identical bids are treated identically, by rounding down if necessary.[27])

(ii) We could do (i) but give greater preference to triply-marginal, etc., by tweaking their demand for any good j by an amount between $j\delta$ and $2j\delta$ ($2\delta < 1/N$).

(iii) We could do (i) but also allow paired bidders to specify their own lexicographic preferences about where they are used. E.g., each bid can label the goods in any order $j' = 1, ..., N$, and have good j' tweaked by an amount between $j'\delta$ and $(j'+1)\delta$ in the case that they turn out to be multiply-marginal ($\delta < 1/N$).

At the time of writing, the web application is using variant D(i), and "no rationing" (simply picking the first solution the program comes to) as an alternative option, but the command line program also allows D. (The command line program also allows the option to randomly re-order bids and bidders on input, thus ensuring a kind of equity in the "no rationing" case.) It is not hard to edit the program to do other forms of rationing.

None of these rationing methods (or probably any other) seems ideal in every possible circumstance. For detailed discussion of the behaviour of different forms of rationing (especially the main variant, D(i), used by our programs), including in the extensions discussed in section 6, please write to the authors.

5 Extension to "Horizontal" as well as Vertical Goods

We have assumed goods $j = 1, ..., N$ are ordered "vertically", with higher-numbered goods always worth (weakly) more, and priced (weakly) higher than lower-numbered goods.

We can also consider goods that are more symmetric, or "horizontal".

Both alternatives are useful.

(1) below does both "vertical" and "horizontal" at the same time: we have a set $l = 1, ..., L$ of "columns" of goods. Each column is a vertical list of $j = 1, ..., N_l$ goods. But the columns are "horizontal" in relation to each other.

(2) below generalises further, by adding one further "base" good that is below all of the "columns". This might cover a particular case of interest–the special case in which each column is of length one, and with a "base good" is below all those columns.

We could generalise further still the kinds of relationships between goods we permit.

Tweaking in the general case

[27] See note 26.

We have not thought through in detail the best tweaking strategy. We doubt there are any issues in principle (the proofs may be more involved) but this may limit the number of goods or bids we can permit without computational (or run-time) issues. So we do price-tweaks as usual–a different tweak for every good. Quantity-tweaks are the natural generalisation of before. We tweak the first step of every supply curve by (η+ sum of tweaks on all goods immediately above the current good). Thus with n goods in all, the sum of all the additional quantities that the tweaks make available to goods at any price vector is (as before) $n\eta$, and we set $n\eta = 1/4$ (if all bids, step sizes, etc., are in integers).

Variables

$x^i_{j,l}$ allocation of bid i to good j of group l (bid size k_i, values $v^i_{j,l}$)

$y^q_{j,l}$ allocation to qth step of goods j *and above* of group $l = 1, ..., L$ (step length $s^q_{j,l}$, height $\mu^q_{j,l}$)

We assume that $\mu^q_{j,l} \geq 0$, and $\mu^q_{j,l}$ is strictly increasing in q for all j, l.[28]

Program (1)

$$\max_{\{x^i_{j,l}\},\{y^q_{j,l}\}} \left[\sum_{i,j,l} v^i_{j,l} x^i_{j,l} - \sum_{j,q,l} \mu^q_{j,l} y^q_{j,l} \right]$$

such that (with the variables on the right-hand side being the corresponding shadow values)

$\sum_{j,l} x^i_{j,l}$	$\leq$	k_i	$\forall i$	b_i, constraint on size of bids
$y^q_{j,l}$	$\leq$	$s^q_{j,l}$	$\forall q, \forall j, \forall l$	$u^q_{j,l}$, constraint on length of steps
$\sum_i x^i_{j,l} - \left(\sum_q y^q_{j,l} - \sum_q y^q_{j+1,l} \right)$	$\leq$	0	$1 \leq j \leq N_l - 1, \forall l$	$z_{j,l}$, good j, l consistency constraint
$\sum_i x^i_{N_l,l} - \sum_q y^q_{N_l,l}$	$\leq$	0	$\forall l$	$z_{N_l,l}$, good N_l, l consistency constraint

And all variables non-negative.

The problem specified in the earlier sections of the paper is the case $L = 1$.

The (basic version) of a "horizontal" problem is the case with $N_l = 1, \forall l$.[29] (If it is also without paired bids, then there is only one non-zero $v^i_{j,1}$ for each j.)

If we work with a TQSS, we will let $R = \sum_{q,l} s^q_{1,l}$.

Program (1*) [Alternative version of TQSS]

[28] In both programs (1) and (2), it would not be hard to allow some $\mu^q_{j,l} < 0$, and perhaps not always increasing in q. We may explore these things.

[29] Cases with $N_l > 1$ corresponds to some variations of a "horizontal" problem that involves "tying" reserve prices. We may develop programs (1) and (2) further to correspond to other cases of tying.

An alternative way to program a variable total quantity, is to not change any step sizes, but instead insert the additional constraint $\sum_{q,l} y^q_{1,l} \leq R$.[30] A simple approach is to relax the constraint as we increase R, and look for consistency with the TQSS as usual. (That is, as before, we would run the linear programme for a sufficient number of of different values of R to find the R which yields $\{z_j\}$ such that $\tau(z_1, ..., z_N) = R$ to the desired accuracy; $\sum_{q,l} s^q_{1,l}$ will now be the maximum value of R that we use.) We will use a more elegant approach of adding an additional constraint into the LP.

Note that if we use this alternative form of TQSS, we need an additional tweak on (each value of) R which should dominate the effects of the other tweaks, so we simply add $2n\eta$ to R. Since the "dummy bids" are automatically sold and thus come out of the "allocation" to R, we are in effect tweaking the allowable total sales by just $3n\eta/2 = 3/8 < 1/2$ so rounding the quantity solutions to the nearest integer as usual will eliminate the distortion. Otherwise, the fact that R is left untweaked can vitiate the effect of the other tweaks. (For example, with a single bid of 8 for 1 unit of a good whose supply curve has a single step at price 5 up to $s = 1$ unit, the basic tweaked problem has $s < 1 + \eta$, with the intention of forcing $p = 5$, but if the TQSS says $q \leq 1$, the competitive equilibrium is any $p \in [5, 8]$. "Tweaking" the TQSS to (in effect) $q = 1.375$ restores $p = 5$ as the unique competitive equilibrium price.)(When we implement the TQSS by adding an additional constraint into the LP, the constraint will be tweaked by adding $2n\eta$ to the first step.)

The "Normalised TQSS"

In the "horizontal" context, it is natural to specify this kind of TQSS in terms of the "normalised price", that is, the auctioneer's "marginal surplus". More precisely, we define the "normalised price" to be the value to the auctioneer of the ability to sell an extra unit (of any good), respecting all constraints apart from the TQSS (so it is the shadow price of a TQSS that restricts the auctioneer to the current total quantity). The normalised price typically equals the price-cost margin on every good, that is, the price of any particular good less the value of the supply curve at the quantity of that good being supplied, but will not always do so since the program chooses the lowest possible prices on goods when multiple prices would give the same allocation. (If there were no lumpiness in bids, so price-cost margins were continuous functions of quantities sold, then the price-cost margin on every good would equal the normalised price.) As usual the TQSS has to start at a very low price (we should think of it being zero for an arbitrarily small quantity) and go up to a high price. Appendix 7.6 gives the

[30] This form of TQSS is equivalent to using a TQSS in the usual way, after reindexing all the goods $j, l \to (j + 1), l$; then defining additional "goods" $1, l$ that all correspond to "no sale", each of which has only a single step of length $s^q_{1,l} = \sum_q s^q_{2,l}$ and height $\mu^1_{1,l} = 0$, and inserting singleton "bids" of sizes equal to the initial lengths of the steps $s^q_{1,l}$ at price 0 on each of the additional "goods" $1, l$, respectively (with lexicographic preferences to prefer any sale of a good j, l for $j > 1$ to the sale of any "good" $1, l$); and using "absolute" supply curves (i.e., for $j > 1$ leave all the $s^q_{j,l}$ unaffected, equivalently "$\lambda = 1$", as we increase R according to the TQSS).

details of how to implement a "normalised TQSS".

Implementing Alternative Pricing Rules for cases of *unsold* goods
Reconsider example 6 as a horizontal problem:

As before, a seller has one unit for sale, and reserve prices 5 and 7 for goods 1 and 2, respectively. So the first steps on the two supply curves are now of height 5 and 7. As before, there is a bid of 8 on good 1, and bids of 20 and 14 on good 2 (all for a single unit).

Clearly a single unit of good 2 is sold at price 14. If we simply program supply curves length 1 & 1, and use the alternative TQSS to impose a maximum sale of 1 (at any price) the price on good 1 will be 8. This is arguably the "correct" price to report. If the seller had any ability to sell additional good 1, the price would be 8. (And it corresponds to the "edge" case in which the cost of good 2 jumps to 13 or more beyond one unit.) However, if the problem is better understood as one in which the seller values good 2 at 2 more than good 1, but just wishes to sell only a single unit on this occasion, then the "correct" price to report for good 1 is probably 12, and, as before, we can achieve this outcome by extending the length of the supply curves on good 2 to, e.g. length 2.

Program (2) We add a special good (1,1), and all other goods have their 2nd suffix increased by 1.

$$\max_{\{x^i_{j,l}\},\{y^q_{j,l}\}} \left[\sum_{i,j,l} v^i_{j,l} x^i_{j,l} - \sum_{j,q,l} \mu^q_{j,l} y^q_{j,l} \right]$$

such that (with the variables on the right-hand side being the corresponding shadow values)

$$\begin{array}{rcll} \sum_{j,l} x^i_{j,l} & \leq & k_i & \forall i \\ y^q_{j,l} & \leq & s^q_{j,l} & \forall q, \forall j, \forall l \\ \sum_i x^i_{j,l} - \left(\sum_q y^q_{j,l} - \sum_q y^q_{j+1,l} \right) & \leq & 0 & 1 \leq j \leq N_l - 1, \forall l > 1 \\ \sum_i x^i_{N_l,l} - \sum_q y^q_{N_l,l} & \leq & 0 & \forall l > 1 \\ \sum_i x^i_{1,1} - \left(\sum_q y^q_{1,1} - \sum_{q,l>1} y^q_{1,l} \right) & \leq & 0 & \end{array}$$

b_i, constraint on size of bids
$u^q_{j,l}$, constraint on length of steps
$z_{j,l}$, good $j, l > 1$ consistency constra
$z_{N_l,l}$, good $N_l, l > 1$ consistency cons
$z_{1,1}$, good $1,1$ (= good $N_1, 1$) consistency constrai

And all variables non-negative.

The only differences from (1) are that the 5th constraint is new and substitutes for the $l = 1$ case of the 3rd and 4th set of constraints (which now therefore apply only for $l > 1$).

[Program (1) is a special case of program (2): good $j, l > 1$ of (2) corresponds to good $j, l-1$ of (1). Then setting all $v^i_{1,1} = 0$, $\mu^1_{1,1} = 0$, and $s^1_{1,1} = arbitrarily\ large$, will mean (2) gives the same solution as (1). (In the solution to (2), an allocation of all or part of a bid i to $x^i_{1,1}$ is equivalent to an allocation to "no sale"; that is, the allocation of i to $x^i_{1,1}$ + "no sale" in (2) corresponds to the allocation of i to "no sale" in (1).)]

The problem specified in the earlier sections of the paper is the case in which $L = 2$, in which what was good 1 before becomes good $1, 1$ here, and in which all goods $j > 1$ before become goods $j - 1, 2$ here.

The (basic version) of a "horizontal" problem is the case with $N_l = 1, \forall l$, and all $v^i_{1,1} = 0$, $\mu^1_{1,1} = 0$, and $s^1_{1,1} = \textit{arbitrarily large}$. (If it is also without paired bids, then there is only one non-zero $v^i_{j,1}$ for each j.)

If we work with a TQSS, we will let $R = \sum_{q,1} s^q_{1,1}$.

6 Other Extensions/Variants

Not all of these extensions are in the current program, but future development should be easy.

Note that (unlike in the preceding) some of the below require a concept of a "bidder" that is separate from the concept of a "bid". *(Of course, our program anyway keep tracks of who makes which bids at the input stage, for reporting purposes at the output stage; we could also use this information at the "rationing" stage of our program.)*

Procurement Auctions

[*At the time of writing, not implemented in our program.*]

Simply the opposite problem of an auctioneer buying goods from bidders who offer prices at which to sell.

Discriminatory Pricing

[*Easy to read off from the program's results, although not explicitly implemented in the program.*]

Without changing the solution mechanism, we let the prices be those that were actually bid.[31]

Non-competitive bids

[*At the time of writing, not implemented in our program.*]

For the discriminatory pricing case, we would like to also allow non-competitive bids.

A non-competitive bid for a single good specifies a good and a quantity of that good, but no price; a non-competitive paired bid specifies the goods and a quantity, and also the differences in willingnesses to pay for the different goods. For example, a non-competitive paired bid for one of two goods can be represented as (NC,NC+x), if the bidder is willing to pay up to x more for the second good than for the first good. A non-competitive bid is always filled, but

[31] Just like we ordinarily do discriminatory pricing, we implicitly assume that the highest bids are the highest value ones, so take the bids (including the selections we make from paired bids) that give the seller the highest surplus. (We don't try to guess whether doing something else might be more efficient.)

the price for it is set at the (weighted) average price paid by all other winning bids on the good which it is allocated.

A maximum size would be imposed on all non-competitive bids, and a limit would be placed on a number of non-competitive bids that any individual bidder can make. (We could also insist that "bidders" who make any non-competitive bids can make only non-competitive bids, and might also restrict non-competitive bids to singleton (*un*paired) bids.)

Introducing non-competitive paired bids (for the discriminatory pricing case) is a little complex. The issue is that the linear program would choose to allocate non-competitive paired bids of the form "(NC,NC+x)" according to the goods' marginal values (i.e., what uniform prices would be). However, it would then be natural to re-allocate the non-competitive paired bids to whichever goods were best for them at the goods' average prices (i.e., according to the prices that non-competitive winners would actually be required to pay). This re-allocation of non-competitive winners might take us off the auctioneer's supply curves and/or TQSS. So, with discriminatory pricing, we would probably only permit non-competitive paired bids if the auctioneer was willing to be sufficiently flexible about the relationship between prices and quantities (which would likely require tight limits on the size of the bids).

For our standard uniform pricing case, however, we could easily permit non-competitive paired bids of the form "(NC,NC+x)", etc., that will always be filled, and will be filled on good 2 if the price of good 2 exceeds that of good 1 by less than x. (Of course, a bidder can also do this by simply substituting a sufficiently large value for "NC", and we will, of course, solve the program by treating these bids in this way. However, it may aid bidders' understanding to offer this option for all bids in the uniform-pricing case.)

Lowest Winning Bid Pricing

[*Implemented in the command line program, but not, at the time of writing, in the web application*]

Our basic program does "true second pricing" (so, for example, if there is just one bid, and that bid wins, it pays the minimum price that it could have bid and won, not the actual price bid). By contrast, most real-world uniform-price auctions are priced at the lowest winner's bid rather than the highest loser's. So for problems in which paired bidding is *not* permitted, it is natural to consider (and we will code) the variant in which the price paid by any winning bid on a good is the lowest price bid on that good by any winner of that good.

Our current implementation should be used with caution if there are any paired bids, because in this case the linear program might allocate a bidder its less-preferred side of the pair at the prices the auction chooses. (For example, a bid of (10,9) would be allocated good 1 if the highest losing prices were (5,5), but it would prefer to be allocated good 2 if the lowest winning bids turned out to be (10,6).[32])

[32]This could arise, for example, if there are three bids, each for one unit, at (10,9), (0,6), and (5,5), and a single unit of each of good 1 and good 2 is to be allocated. The point, of

Generalised paired bids

[*Implemented*]

Our program permits "generalised paired bids". Such a bid is specified by $\{k_i, v^i_{j,l}, \kappa^i_{j,l}\}$ and the constraints $x^i_{j,l} \leq \kappa^i_{j,l}, \forall j, l$, apply to it, in addition to the standard constraint, $\sum_{j,l} x^i_{j,l} \leq k_i$.

(That is, an ordinary "paired bid" has $\kappa^i_{j,l} = k_i \ \forall j, l$ (so the new constraints, $x^i_{j,l} \leq \kappa^i_{j,l}$, are irrelevant), and the bid selects the single best option out of all the goods (or no purchase) for the bidder. A "generalised paired bid" with $\kappa^i_{j,l} = (k_i/n) \ \forall j, l$ selects the n best options out of all the goods j, l (and no purchase, which option can be chosen up to n times), for allocating (k_i/n) units, for the bid.)

Asymmetric (and Generalised-asymmetric) Paired Bids

[*Implemented*]

Our program allows bidders to express more general trade-offs than 1-1 between goods. An asymmetric paired bid i can express indifference between $(1/a^i_{j',l})$ units of good j' and $(1/a^i_{j'',l})$ units of good j'', etc., that is, the bid specifies a constraint $\sum_{j,l} a^i_{j,l} x^i_{j,l} \leq k_i$.[33,34]

A generalised asymmetric paired bid i specifies constraints of the forms $\sum_{j,l} a^i_{j,l} x^i_{j,l} \leq k_i$ and $a^i_{j,l} x^i_{j,l} \leq \kappa^i_{j,l}, \forall j, l$.[35]

Additional Sellers/Swappers

[*At the time of writing, not implemented in our program.*]

We could permit bidders to input goods to sell, as well as bids to buy. That is, a bidder might say that he would like to sell one unit of good 1 if the price exceeds 98, and two units of good 1 if the price exceeds 100. We would add those into the supply we have for good 1. This does not preclude a bidder from making a bid on good 1. (In particular, the bidder might make a "paired" bid

---

course, is that (10,6) are not competitive-equilibrium prices.

The highest-feasible competitive-equilibrium prices in this example are (7,6). We could interpret "lowest winning prices" as meaning these prices (namely the *maximal* shadow prices in the solution of the linear programme). This is another possible pricing rule, but it seems to have neither the advantage of easy comprehensibility, nor all the advantages of "true second pricing". [In this example, with this form of "lowest winning prices", either of the two winners could gain by bidding "untruthfully".]

[33] Note that if bid i places no value on good $\widetilde{j}$, we set $a^i_{\widetilde{j},l}$=1, and $v^i_{\widetilde{j},l} = 0$.

[34] For example, if bid i is indifferent among: buying 2 apples at price 12; buying 3 bananas at price 23, and buying nothing at all, the bid's utility is $Max\{2(12 - z_A), 3(23 - z_B), 0\}$, and so $a^i_{A,l} = 1/2, a^i_{B,l} = 1/3, k_i = 1$.

[35] Rationing is harder when there are asymmetric (or generalised-asymmetric) paired bids.

First, identifying marginal bids requires care. For example, in the example of Note 34, bid i is (multiply-)marginal at price (12-2θ, 23-3θ) for all $\theta \geq 0$.

Second, to ensure uniqueness we need to avoid the "direction" of any tweak being the same as that of any tradeoff, which adds complexity since the directions of tradeoffs can be arbitrary (in the example of Note 34 the tradeoff is 2:3), by contrast with "ordinary" paired bids for which tradeoffs are always 1:1. For simplicity, at the time of writing, our implementation just tweaks (multiply-)marginal bids in the usual way, and accepts the theoretical possibility of non-uniqueness.

on a unit of goods 1 and 2, so that if the bid on 2 is successful, the bidder will then have swapped a unit of 1 for a unit of 2; if he bid on 2 is not successful, he will have bought back the unit of good 1 that he started with, unless the price of 1 was very high.)

If multiple sellers offer supply curves, they can be added horizontally *if* they order the goods in the same way. Whether or not the sellers order the goods in the same way, we can also handle a multiple-seller case by "unpacking" all the supply curves as we describe below, and then finding equilibrium when all the sellers' supplies, and all the sellers' and bidders' demands, are expressed simultaneously.[36,37] With multiple sellers, the auctioneer (probably one of the sellers) needs to specify a single TQSS (as a continuously (weakly) increasing function of all the prices $\{z_i\}$,as usual).[38,39]

Observation related to the above: We have specified our problem as that of a seller who uses supply curves to determine the quantity of each good that it will sell. This problem is equivalent to that of a seller who will sell a pre-determined quantity of each good, but who also makes bids to buy some of those goods back.[40]

For simplicity, we consider the basic program of Section 2.[41] Let $\sum_q s_j^q = R$, for all j. (This is w.l.o.g., since we can make the final step on each supply curve arbitrarily large.) So the maximum quantity that can be sold is R units *in total* of the N goods. Then our problem is equivalent to there being R units of *each* of the N goods available for sale, and the auctioneer buying back between $(N-1)R$ and NR units by making a bid corresponding to each step of each supply curve. The bid corresponding to the qth step of the supply curve on good 1 is for s_1^q units at value μ_1^q; the bid corresponding to the qth step of the supply curve on good $j > 1$ is a "paired bid" for s_j^q units at value (0,...,0,$T, T + \mu_j^q$,0,...,0) with T arbitrarily large.

[36]The unpacked approach can represent more general seller preferences than supply functions can represent. (The former allows any strong-substitutes preferences, in full (N) dimensions, that can be represented with positive dotbids. The latter can represent only preferences that can be expressed as a series of 2-dimensional slices. The more general programs of Section 5 expands the set of slices, but not to permit the full range of seller preferences that positive dotbids can represent. On the other hand, unpacking "generalised paired bids" and some other kinds of constraints across bids would require using negative dotbids.)

[37]It might superficially seem that the two approaches are *not* equivalent, even when both are feasible, and that the "unpacked" approach allows additional trade between sellers: e.g., if two sellers each wish to sell a total of R units of goods A,B, and C, might the equilibrium involve one seller buying back $2R$ of good A, thus ending up with more of good A than it started with? (while, e.g., the other seller buys back $2R$ of good B, and bidders buy $2R$ of good C). The answer is "no" because $\sum_q s_j^q = R$ implicitly assumes that the seller places no value on more than R units of j, so would never want to buy back more than R units of j in this transaction, even if they were available.

[38]Unless there's a single dominant seller, a TQSS doesn't seem an especially natural thing.

[39]The "unpacking" method automatically handles any needed rationing among sellers. Rationing among sellers would, in principle, be needed in the supply curves approach.

[40]This is, of course, just the generalisation of the standard point that setting a reserve price when selling a single object is equivalent to promising to definitely sell the object, but entering a bid of one's own at the reserve price.

[41]It is easy to generalise to the programs of Section 5.

To see the equivalence, observe that large T guarantees that all the paired bids will be filled, so the auctioneer will buy back at least $(N-1)R$ units. Moreover, the values of the "paired bids" mean that if, for simplicity, $p_j - p_{j-1} \neq \mu_j^q \ \forall j, q$, the auctioneer buys back $\sum_{\{q|\mu_j^q > p_j - p_{j-1}\}} s_j^q + \sum_{\{q|\mu_{j+1}^q < p_{j+1} - p_j\}} s_{j+1}^q$ $= (R - \sum_{\{q|\mu_j^q < p_j - p_{j-1}\}} s_j^q) + \sum_{\{q|\mu_{j+1}^q < p_{j+1} - p_j\}} s_{j+1}^q$ units of good j, for $1 < j < N$, that is, on net sells $\sum_{\{q|\mu_j^q < p_j - p_{j-1}\}} s_j^q - \sum_{\{q|\mu_{j+1}^q < p_{j+1} - p_j\}} s_{j+1}^q$ units of good j, for $1 < j < N$, exactly as the supply curve formulation requires (and it is also easy to check the cases in which $j = 1$, $j = N$, and/or $p_j - p_{j-1} = \mu_j^q$ for some j, q).

Constraints on how much individual bidders can win

[*Implemented in the command line program, but not, at the time of writing, in the web application.*]

We can limit the amount any individual bidder can win as a fraction of the auction "size" $\sum_q s_1^q = R$.[42] *Our current implementation should be used with caution if there are any asymmetric paired bids,*[43] *and/or we are using a TQSS.*[44]

Additional Constraints

[*The ability to specify a total quantity constraint across all of a bidder's bids is implemented in the command line program but not, at the time of writing, in the web application.*]

We permit bidders to input a total quantity constraint across a group of

[42] Limiting the amount any individual bidder can win as a fraction of the total actually allocated would be harder: we cannot simply implement this as a constraint in the LP, because the program might then allocate some units at prices below the costs of the relevant steps and/or prices above bids' values, because the objective function loses less from that than it gains from the relaxed constraint on allocating goods to a different bidder who is total-quantity constrained. (Another issue is that we would always allow a bidder to win some minimum–otherwise we would have a problem if there were only 1 or 2 bidders. Also a very tight a constraint might facilitate collusion.)

[43] With asymmetric paired bids, constraints can generate complements. (Consider a singleton bid for 1A at price 3 and an asymmetric paired bid for (1B or 2C) at prices 3 and 3(per unit). Assume an overall constraint of 2 units. If prices are 0,0,1, the bidder wants A+B. If the price of A rises to 3, the bidder wants 2C.)

[44] At least in theory, there may be no solution that satisfies the TQSS. The problem is that whereas ordinarily if we increase the size of the auction, then prices must go down, now increasing the size of the auction relaxes the constraint on the amounts bids can win, and could conceivably push at least one auction price up, so (see section 3) there may be no intersection between the graphs of R and τ. However, we can simply say we don't mind what the prices are at the maximum total auction size, i.e., force the maximum total auction size to be the solution if there is no other solution. (Alternatively, a fixed limit on how much people are allowed to bid for could guarantee a solution to the TQSS.)

Also, at least in theory, we can construct cases in which there is more than one intersection. In this case we can take the first one we come to, which would either be the one that allocates the least, or the most, depending on where we start. In fact we use binary search to save computer time (this always yields a solution if the TQSS starts at zero and covers a large enough range).

bids.[45,46] We could easily permit additional linear constraints that don't generate new constraints on other bidders, or new incentives for allocations to other bidders.[47]

Budget-Constrained Bidders

[*Implemented*]

Our implementation for budget-constrained bidders does not use linear programming, so is beyond the scope of the current paper.

Maximising Profitability rather than Efficiency

[*Implemented, but restricted to horizontal auctions without a TQSS. We currently permit standard paired bids, but not asymmetric or generalised paired bids. (Improvements are currently under development.)*]

This is trivial for discriminatory pricing–our program solves the problem! (i.e., conditional on using discriminatory pricing, our program maximises profits).

For our standard uniform pricing this is harder, and does not use linear programming, so discussion of the details is beyond the scope of this paper.

7 Appendix

7.1 Proof that the η "tweaks" determine the $z_1, ..., z_N$ uniquely when all the goods are sold

Assume we already know, without ambiguity, the prices in set $J \subsetneq \{1, \ldots, N\}$ of goods. We will show that we can also find, without ambiguity, the price of some good $j' \notin J$. We can then apply this argument, first when J is empty, and then repeatedly, to show that the price of every good can be found without ambiguity.

Assume that the equality in CC 4 holds for every j. (This is w.l.o.g. because although this equality need not hold when $z_j = 0$, these values of y_j^q are always an optimal solution.) It follows that the good j consistency constraint holds with equality for every j, so we may take the sum over all these constraints for every $j' \notin J$, yielding:

$$\sum_{i,\, j' \notin J} x_{j'}^i = \sum_{j' \notin J, j' \neq N} \left(\sum_q y_{j'}^q - \sum_q y_{j'+1}^q \right) + \sum_{j' \notin J, j'=N} \sum_q y_N^q. \qquad (2)$$

[45] A total quantity constraint across a group of ordinary paired bids is just an easy generalisation of a generalised paired bid. (The (single) generalised paired bid $\{k_i,\, v_{j,l}^i,\, \kappa_{j,l}^i\}$ can be written as a set of bids (one for each value of j, l) of sizes $\kappa_{j,l}^i$ and values $(0,....0, v_{j,l}^i, 0, ...0)$ which is constrained by $\sum_{j,l} x_{j,l}^i \leq k_i$.)

[46] If the constraints create complements, our program maximises welfare but there may not be a unique lowest competitive equilibrium price.

See also note 44 above, for the issues arising if prices are not be monotonic in the "size" of the auction.

[47] Note 42 discusses the difficulties created by allowing bidders to enter constraints that generate new constraints on, or new incentives for allocations to, other bidders.

If every time we allocate on a step on any good $j' \notin J$, we allocate the step in full, then every term $\left(\sum_q y^q_{j'} - \sum_q y^q_{j'+1}\right)$ (if $j' \neq N$) or $\sum_q y^q_N$ (if $j' = N$) has remainder η. So if we allocate in full on all steps on all goods $j' \notin J$, then the non-integral part of the RHS of (2) is $(N - |J|)\eta$, and so also the LHS of (2) is non-integral. So in this case for some bid i,

Case (i) $0 < \sum_{j' \notin J} x^i_{j'} < k^i$ for some i.

Moreover, if case (i) does not hold, the RHS of (2) must be integral. There is therefore some step on which we have only partially allocated, that is, some $j' \notin J$ with $0 < y^q_{j'} < s^q_{j'}$ or $0 < y^q_{j'+1} < s^q_{j'+1}$ Furthermore, if both $j' - 1 \notin J$ and $j' \notin J$, then the two occurrences of $\sum_q y^q_{j'}$ in the RHS of (2) would cancel out. Similarly, if both $j' \notin J$ and $j'+1 \notin J$ then the two occurrences of $\sum_q y^q_{j'+1}$ would cancel out. So there must be a partially allocated step on some $j' \notin J$ for which such a cancellation does *not* apply. That is, if case (i) does not hold, then one of

Case (ii) $0 < y^q_{j'} < s^q_{j'}$ for some q and some $j' \notin J$, such that either $j' - 1 \in J$ or $j' = 1$.

Case (ii') $0 < y^q_{j'+1} < s^q_{j'+1}$ for some q and some $j' \notin J$, such that $j' + 1 \in J$.

We can now consider each of the posssible cases in turn:

Case (i). Let bid i be such that $0 < \sum_{j' \notin J} x^i_{j'} < k^i$. Now:

Either $\sum_j x^i_j = \sum_{j' \notin J} x^i_{j'}$, so $0 < \sum_j x^i_j < k^i$. In this case CC 6 and CC 5 jointly imply that $\max_j\{v^i_j - z_j\} = 0$. But since $\sum_j x^i_j = \sum_{j' \notin J} x^i_{j'}$, only goods $j' \notin J$ are being allocated to bid i, so $\max_{j' \notin J}\{v^i_{j'} - z_{j'}\} = \max_j\{v^i_j - z_j\} = 0$. Moreover, it now follows from CC 5 that all goods that are allocated to bid i have $\{v^i_j - z_j\} = 0$. So the price $z_{j'}$ of at least one good $j' \notin J$ is now known without ambiguity.

Or $0 < \sum_{j' \notin J} x^i_{j'} < \sum_j x^i_j \leq k^i$, so bid i must be being rationed across one or more goods $\notin J$ and other goods $\in J$. By (the contrapositive of) CC 5 the surplus is the same for all such goods. That is, there exists some good $j' \notin J$ and a good $j \in J$ with $v^i_{j'} - z_{j'} = v^i_j - z_j$. Since z_j is already known without ambiguity, this fixes the price of the good $j' \notin J$ without ambiguity.

Case (ii). The contrapositives of CC 7 and CC 8 tell us that $\mu^q_{j'} = z_{j'} - z_{j'-1}$. Since $j' - 1 \in J$, the price $z_{j'-1}$ is already known without ambiguity, so this fixes the price of $j' \notin J$ without ambiguity.

Case (ii'). As in case (ii) above, CC 7 and CC 8 $\Rightarrow z_{j'+1} - z_{j'}$; Since $j'+1 \in J$, this fixes the price of $j' \notin J$ without ambiguity.

This completes the proof $\square$.

7.2 Proof that the use of η "tweaks" to resolve ambiguity in prices, selects the *lowest* possible prices, when all the goods are sold

At any solution of our ("untweaked") program, let the total quantity of good j allocated to the bidders be $X_j = \sum_i x_j^i$, and the prices be $\{z_j\}$. Let $Y_j = \sum_q y_j^q$, so in the solution $Y_j = X_j + X_{j+1} + ...+ X_N$. (We noted above that $j = 1$ is an exception to the rule that compatibility constraints bind in the solution–but we can think of the Y_1 that satisfies this equality as being the "correct" Y_1, so this is w.l.o.g.). So $X_j = Y_j - Y_{j+1}$ for $j < N$, and $X_N = Y_N$. Now "tweak" the program by adding $(N + 1 - j)\eta$ to s_j^1. At the original prices, $\{z_j\}$, the supply conditions would be satisfied by quantities $\left\{\widetilde{X}_j\right\}$ that satisfy $Y_j+(N+1-j)\eta = \widetilde{X}_j+\widetilde{X}_{j+1}+...+\widetilde{X}_N$, so $\widetilde{X}_j = Y_j-Y_{j+1}+\eta$ for $j < N$, and $\widetilde{X}_N = Y_N + \eta$. That is, they would be satisfied by $\left\{\widetilde{X}_j\right\}$ such that $\widetilde{X}_j - X_j = \eta > 0$. So excess supply, $\widetilde{X}_j - X_j$, would be weakly positive for every good j. So since both the new supply curves and bids represent ordinary substitute preferences, it follows from Theorem 23 of Milgrom and Strulovici (2009) that there exists an equilibrium of the "tweaked" program at which all prices are weakly lower than $\{z_j\}$. But we have already established that the prices of the "tweaked" program are uniquely determined, so they are therefore weakly lower than any equilibrium prices of the "untweaked" program. □

7.3 Proof that, without paired bids, "tweaking" the bids by adding $a_j\epsilon$ to all bids on the jth good, in which the $a_j \in \{1, ..., N\}$, selects $\{y_j^q\}$ and $\sum_i x_j^i$ uniquely.

We "tweak" the bids by adding $a_j\epsilon$ to all bids on the jth good, in which the $a_j \in \{1, ..., N\}$are such that j is the $(N + 1 - a_j)$th preference good (so the good we most prefer to allocate to has $a_j = N$, and the good we allocate to last has $a_j = 1$), and $\epsilon < \frac{1}{2N}$ (assuming all bids are integers, e.g., whole numbers of basis points). That is, the objective function becomes

$$\max_{\mathbf{x}^1,...,\mathbf{x}^M,\{y_j^q\}} \left[\sum_{i,j}(v_j^i + a_j\epsilon)x_j^i - \sum_{j,q}\mu_j^q y_j^q\right]$$

Let the shadow prices with the new objective function be $z'_1, \ldots, z'_N$, and the shadow prices rounded down to the nearest integers be $z_1, \ldots, z_N$.

We show that $\{\mathbf{x}^i\}$ are optimal choices for the bidders in the "old" problem, if the prices are $\{z_j\}$. There are three cases.

1. If $x_j^i = k^i$, then $v_j^i + a_j\epsilon \geq z'_j$ so, since all the v_j^i and z_j are integer, and all the $a_j\epsilon < 1$, it follows that $v_j^i \geq z_j$ (the cases $>$ and $=$ yield $\geq$ and $=$ respectively), so $x_j^i = k^i$ is an optimal choice (in the "old" problem).

2. If $x_j^i = 0$, then $v_j^i + a_j\epsilon \leq z'_k$ so (since all the v_j^i and z_j are integer, and all the $a_j\epsilon < 1$) we have $v_j^i \leq z_j$, so $x_j^i = 0$ is an optimal choice.

3. If $x_j^i \in (0, k^i)$, then $v_j^i + a_j\epsilon = z'_j$ so (since all the v_j^i and z_j are integer, and all the $a_j\epsilon < 1$) we have $v_j^i = z_j$, so any $x_j^i \in [0, k^i]$ (consistent with the constraints) is an optimal choice.

Now we show that $\{y_j^q\}$ are optimal choices for the auctioneer in the "old" problem if the prices are $\{z_j\}$. For convenience, we write $z'_0 = z_0 = 0$ for the case $j = 1$).

1. If $y_j^q = s_j^q$, then $z'_j - z'_{j-1} \geq \mu_j^q$ so, since all the μ_j^q and z_j are integer, it follows that $z_j - z_{j-1} \geq \mu_j^q$, so $y_j^q = s_j^q$ is an optimal choice (in the "old" problem).

2. If $y_j^q = 0$, then $z'_j - z'_{j-1} \leq \mu_j^q$ so (since all the μ_j^q and z_j are integer) we have $z_j - z_{j-1} \leq \mu_j^q$, so $y_j^q = 0$ is an optimal choice.

3. If $y_j^q \in (0, s_j^q)$, then $z'_j - z'_{j-1} = \mu_j^q$ so (since all the μ_j^q and z_j are integer) we have $z_j - z_{j-1} = \mu_j^q$, so any $y_j^q \in [0, s_j^q]$ (consistent with the constraints) is an optimal choice.

Therefore, since $\{\mathbf{x}^i\}$ and $\{y_j^q\}$ are also consistent with the constraints of the old problem (the constraints to the new and old problems are the same!), they are a solution to the old problem.

Absent lexicographic preferences, the auctioneer might be indifferent between allocating or holding back units of goods, and there might be bids on different goods that are "tied". But in the solution given here, the allocation of any good to some bidder is always preferred, and bids on higher preference goods are always allocated in preference to "tied" bids on lower preference goods, so this is a solution to the lexicographic-preference problem. Since it elementary that the set of optimal solutions to a linear programme is a convex set, it is straightforward that this solution is also the unique solution to the lexicographic-preference problem (see formal proof in next subsection of Appendix). □

7.4 Proof that solving the lexicographic preferences determines $\sum_i x_j^i$ uniquely, for all goods j.

*Proof.*The set X of solutions $\{x_j^i\}, \{y_j^q\}$ optimising the objective function is convex, by standard theory of linear programming. It follows that its image $\bar{X}$ under the linear transformation

$$\{x_j^i\}_{i,j}, \{y_j^q\}_{q,j} \mapsto \left\{ \sum_i x_j^i \right\}_j$$

is convex also. We thus have a convex subset of N-dimensional space (recall N is the number of goods) representing all possible total allocations against each

good which optimise the objective function. Let the axes of this space be t_j for $j = 1, \ldots, N$ (t_j being the total allocation against good j. Note that $\bar{X}$ has dimension at most N.

The subset $\bar{X}^1$ of $\bar{X}$ for which $\sum_j t_j$ is maximised is again a convex set, because it is the intersection of a convex set and a hyperplane ($\sum_j t_j =$ constant). It has dimension at most $N - 1$, as it lies in this hyperplane.

Next, if j_1 is the good first in order of priority, the subset $\bar{X}^2$ of $\bar{X}^1$ for which t_{j_1} is maximised is again a convex set, being again the intersection of a convex set and a hyperplane. It has dimension at most $N - 2$ since the hyperplanes $\sum_j t_j =$ constant and $t_{j_1} =$ constant intersect in this dimension (they have linearly independent normals).

We continue this argument for each subsequent good in order of priority, obtaining at each step a new convex set constrained by a smaller dimension. Finally, when we distinguish between the $(N-1)$th good and the Nth good, we obtain a single point.

So the image of the lexicographic preference problem in this N-dimensional space is unique – that is, $\sum_i x_j^i$ is uniquely specified by the lexicographic preferences. □

7.5 Alternatives approaches to rationing

A. We simply let the linear programme do what it would do. This is of course, efficient, and allocates the maximum amount possible (if we already imposed tweaks to implement the auctioneer's preference to allocate the maximum possible) but is completely arbitrary, and may look unfair. So the following variant is probably a considerable improvement:

We could tweak the multiply-marginal bids by increasing their prices on good j by $j\delta$. (We could increase the prices by a slightly larger amount on triply-marginal than doubly-marginal, etc.) We then rerun the LP. This should favour the multiply-marginal bids in the allocation, while still selecting a solution (after rounding prices back down).

We could further allow paired bidders to specify their own lexicographic preferences about where they are used, by allowing them to specify component(s) of their bid to be tweaked more than other components if they turn out to be multiply-marginal. (See variant D(iii), in the main text.)

We can also then go through the goods in turn, reallocating to treat all the other rationed bids equally, which is easy to do since they are all "singly-marginal" (while leaving the allocations to multiply-marginal bids unchanged).

The (?only) remaining problem is that the allocations of multiply-marginal bids may look strange. So we should finally reallocate, so that identical multiply-marginal bids are treated identically. (This is easy to do, by just rationing equally among all identical bids.[48])

[This doesn't quite solve all problems–e.g., a multiply-marginal bid may by chance be treated "worse" than another which is a multiple of it.]

[48] See note 26, in main text.

B. We could divide up the multiply-marginal bids (doubly-marginal into two bids for 1/2 unit, triply-marginal into three bids for 1/3 unit, etc). We are then in a no "OR" bids situation, and simply ration fairly between all the bids. This is clear and straightforward and fair, but may allocate less than it could do, and we assume we have lexicographic preferences for allocating more rather than less. [The problem arises if there are too many multiply-marginal bids, e.g., if there are $m = 3$ 11s paired with $n = 3$ 30s, we're not allocating the full 2 units of 11s, even though this would be possible.]

(Another objection is, that as this scheme stands, multiply-marginal bidders may feel hard done by – they may not like receiving little bits [and we want to favour paired bidders])

More problematic, this can be inefficient if there are multiply-marginal bids, where the marginal parts are (all) above the allocation prices: assume we will sell two units at 30 and two at 11. But assume we have a multiply-marginal bid at (40, 21) in addition to two bids at 30 and one bid at 11. Then efficiency requires (40, 21) gets a full unit. Moreover, we can't just divide the bid into two halves, since the program may need to allocate to one side or the other–in this case, the side that has price 21 must be the side that is allocated. Since we may have a cycle of multiply-marginal bids, e.g., one may be marginal between A and C; a second marginal between A and B; and the third marginal between B and C, it seems hard to guarantee efficiency with this approach.

So this approach (B) doesn't seem appealing unless either (i) we have no multiply-marginal bids, or (ii) the auctioneer has the flexibility to vary its supply a little so that multiply-marginal bids, where the marginal parts are (all) above the allocation prices, can be given the choice of which part they would like filled, prior to rationing the remaining marginal bids.[49]

[49]There are some natural variants of this approach, but they suffer from the same basic problem

(i) Having divided the multiply-marginal bids into halves, or thirds, etc., we could favour them in the straightforward way of filling them first on each good (rationing among them if necessary), before rationing any remainder among other bids. (Or fill/ration among triply-marginal, then doubly-marginal, etc., before other bids.)

(ii) Not quite the same as B, but similar:

Go through the goods in the same lexicographic order that the auctioneer used earlier. Assume, w.l.o.g. that good 2 (the 30s) comes before good 1 (the 11s) in this order.

Then *either* (a) each of the bids of 30 gets $1/n$ units, whether or not they are parts of multiply-marginal bids.

or (a') allocate to the 30s those bids on 30s that are parts of multiply-marginal bids (rationing among them if necessary), before rationing any remainder among other bids. [this is as in B(i), but without initially having divided pairs into smaller bids]

Now either (b) any parts of multiply-marginal bids on 11 are then reduced by the amount allocated to a 30. (In case (a) that means reduce to a bid of $(1 - 1/n)$ on 11.) The 2 units of 11 are now allocated in proportion. (In case (a), if e.g., just 1 of the m bids was part of a multiply-marginal bid, then all the bids on 11 now get $2/(m - 1/n)$.)

or (b') any of the k bids on 11 that haven't yet received anything (the non multiply-marginal bids in this example, but would include multiply-marginal bids that are marginal with lexicographically still-lower goods) get $1/n$ if $k/n < 2$ (or $2/k$ otherwise), then all m bids share any remainder. [(b') may seem fairer than (b), or may seem "hard" on the multiply-marginal bids.]

As in other B-schemes, we may be allocating fewer units than we could allocate (by "wast-

(This issue raises no problems for the other broad approaches we consider.)

C. The auctioneer makes additional units available to allocate all the bids (perhaps giving multiply-marginal bids the choice). Or the auctioneer might do a limited amount of this (perhaps favoring multiply-marginal bids). (Or the auctioneer could allocate fewer units.)

7.6 Including a Normalised TQSS[50]

We assume there are $l = 1, ..., N$ horizontal goods, i.e. N columns with only one good each. The normalised TQSS is defined as a step function and TQSS steps are denoted by n. Each step n has length τ_n and height κ_n. The height of a given step n represents the seller's cost of supplying a unit of any good on that step, i.e., the seller's marginal cost of supplying an additional unit of any good (in addition to the marginal cost specified by the respective good's supply curve) when the total number of units of all goods sold, $\sum_{q,l} y_l^q$, satisfies $\sum_{s=1}^{n-1} \tau_s \leq \sum_{q,l} y_l^q < \sum_{s=1}^{n} \tau_s$. Let t_n denote the quantity allocated on step n. We incorporate the normalised TQSS in the linear programme by subtracting $\sum_n \kappa_n t_n$ from the objective and by including the total supply constraint $\sum_{q,l} y_l^q \leq \sum_n t_n$ and the step constraint $0 \leq t_n \leq \tau_n$.

Primal problem

$$\max_{\{x_l^i\},\{y_l^q\}} \left[\sum_{i,l} v_l^i x_l^i - \sum_{l,q} \mu_l^q y_l^q - \sum_n \kappa_n t_n \right]$$

such that (corresponding shadow values on the right)

$$\begin{array}{rcll} \sum_l x_l^i & \leq & k_i & \forall i \quad b_i, \text{ constraint on size of bids} \\ y_l^q & \leq & s_l^q & \forall q, \forall l \quad u_l^q, \text{ constraint on length of steps} \\ \sum_i x_l^i - \sum_q y_l^q & \leq & 0 & \forall l \quad z_l, \text{ good } l \text{ consistency constraint} \\ \sum_{q,l} y_l^q - \sum_n t_n & \leq & 0 & \quad w, \text{ total supply constraint} \\ t_n & \leq & \tau_n & \forall n \quad \rho_n, \text{ constraint on length of TQSS step} \end{array}$$

and all variables are non-negative.

Dual problem

---

ing" units on the 30s, when they could have been allocated to 11s). The more we favour the multiply-marginal bids (as in a'+b), the more we run the risk of "inefficiently" allocating fewer units than we could allocate.

Another issue with B(ii) is that a multiply-marginal bid may regret being allocated on the 30, since it may lexicographically prefer the 11 for some reason. We're not too troubled about this, because it has already faced the risk of this happening in the much more likely event that there was a tie between prices $30 - x$ and $11 - x$ for some $x > 0$, in which case it would have been decided on the basis of the auctioneer's needs (or possibly auctioneer's lexicographic preferences).

[50] **This Appendix was written by Simon Finster.**

$$\min_{\{b^i\},\{z_l\},\{u_l^q\},w,\{\rho_n\}} \left[\sum_i k^i b^i + \sum_{q,l} s_l^q u_l^q + \sum_n \tau_n \rho_n \right]$$

such that

$$\begin{array}{rcll} b^i + z_l & \geq & v_l^i & \forall i, l \quad \text{corresponding } x_l^i \\ u_l^q - z_l + w & \geq & -\mu_l^q & \forall q, \forall l \quad \text{corresponding } y_l^q \\ -w + \rho_n & \geq & -\kappa_n & \forall n \quad \text{corresponding } t_n \end{array}$$

and all the variables b^i, u_l^q, z_l, w, and ρ_n are non-negative.

Shadow prices
The interpretation of the shadow price b^i is unchanged. The shadow price u_l^q, corresponding to allocating on the qth step of the good l supply curve, is the marginal value of relaxing the constraint $y_l^q \leq s_l^q$. It is interpreted as the marginal value of being able to sell an additional unit of good l when the total supply constraint is not binding. When the total supply constraint is binding, it must be interpreted as the marginal value of being able to sell an additional unit of good l instead of a unit of any other good $l' \neq l$. The shadow price on the lth consistency constraint, z_l, is interpreted as the price of good l. w is the marginal value of being able to sell an additional unit by relaxing the TQSS constraint. ρ_n is the marginal value of being able of sell an additional unit on the same TQSS step (the marginal value of relaxing the TQSS step length constraint).

Complementary slackness

CS 1. If $x_l^i \neq 0$, then $b^i = v_l^i - z_l$.

CS 2. If $\sum_l x_l^i < k_i$ then $b^i = 0$.

CS 3. If $\sum_i x_l^i < \sum_q y_l^q$ then $z_l = 0$.

CS 4. If $y_l^q \neq 0$, then $u_l^q + \mu_l^q = z_l - w$.

CS 5. If $y_l^q < s_l^q$ then $u_l^q = 0$.

CS 6. If $\sum_{q,l} y_l^q < \sum_n t_n$ then $w = 0$.

CS 7. If $t_n < \tau_n$ then $\rho_n = 0$.

CS 8. If $t_n \neq 0$ then $w = \rho_n + \kappa_n$.

Auction Prices
If q is the final step on which good l is allocated and n is the final TQSS step on which any good is allocated, then

(i) if $0 < y_l^q < s_l^q$, then $z_l = w + \mu_l^q \ \forall l$

(ii) if $y_l^q = s_l^q$ (and $y_l^{q+1} = 0$), then $z_l \in [w + \mu_l^q, w + \mu_l^{q+1}] \ \forall l$

(iii) if $0 < t_n < \tau_n$, then $w = \kappa_n$

(iv) if $t_n = \tau_n$ (and $t_{n+1} = 0$), then $w \in [\kappa_n, \kappa_{n+1}]$

Prices are typically set equal to the "cost" of the final supplied unit which is μ_l^q plus an increment w, which is typically strictly positive. Depending on the bids, prices may also be higher than $\mu_l^q + w$. The price of good l is z_l, when this is uniquely defined. When the $\{z_l\}$ are not unique, we will select the lowest possible values such that the CS constraints hold. The same principle applies to the normalised price (or the TQSS shadow price) w itself. Whenever w is not unique, we select the lowest possible.

Tweaks and dummies to determine correct and unique prices
In the case where (i) no bids for good l are made OR (ii) bids for good l are made but no positive amount is allocated on good l, we need to employ a dummy bid at an arbitrarily high price in order to (a) fix prices *uniquely* and (b) determine prices *correctly* by adjusting for the shadow value of the supply constraint. The size of this dummy bid is $\eta/2$, where $0 < \eta < \frac{1}{2N}$. Remember that the supply curves are tweaked by adding $(N + 1 - l)\eta$ to s_l^1 in order to uniquely determine the lowest prices possible.

Additionally, we need to tweak the first step of the TQSS by adding $2N\eta$ to it (so the dummy bids don't fill up the tweak and the overall quantity constraint doesn't mess with the tweaks on the supply curves). Then we always have $w = \kappa_n$ with n being the last step of the TQSS with $t_n > 0$.

8 References

Baldwin, E., Goldberg, P., Klemperer, P. and Lock, E. (in preparation). 'Solving Strong-Substitutes Product-Mix Auctions'.

Baldwin, E., and Klemperer, P. (in preparation). 'Implementing Walrasian Equilibrium –the Language of Product-Mix Auctions'.

Frost, T., Govier N., and Horn T. (2015). 'Innovations in the Bank's provision of liquidity insurance via indexed long-term repo (ILTR) operations'. Bank of England Quarterly Bulletin, 55/2: 181-188.

Klemperer, P. (2008). 'A New Auction for Substitutes: Central Bank Liquidity Auctions, the U.S. TARP, and Variable Product-Mix Auctions'. Mimeo: Oxford University.

Klemperer, P. (2010). 'The Product-Mix Auction: a New Auction Design for Differentiated Goods'. Journal of the European Economic Association, 8: 526-36.

Klemperer, P. (2018). 'Product-Mix Auctions'.Nuffield College Economics Discussion Paper 2018-W07

LIST OF MAIN VARIABLES

- x_j^i allocation of bid i to good j (bid size k_i, values v_j^i)
- y_j^q allocation to qth step of goods j *and above* (step length s_j^q, height μ_j^q)

dual variables

- b_i marginal value of bid i
  (marginal cost of k_i constraint)
- u_j^q marginal value of using qth step for goods $\geq j$
  (marginal cost of s_j^q constraint)
- z_j (shadow) price (= marginal value) of good j
  (marginal cost of jth "consistency constraint")

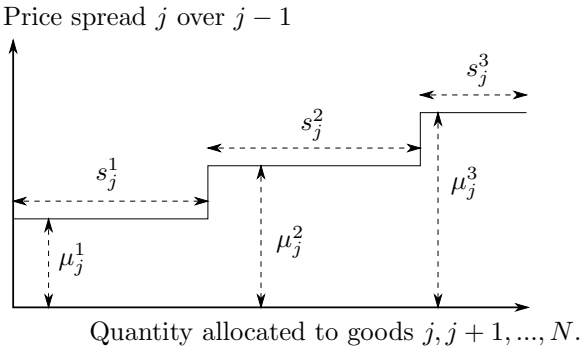

Figure 1: The jth "supply curve", which relates price spread of the jth good over the next *lower-quality* good, with the quantity of all goods *superior* or equal to j, is composed of steps.