Dynamic auctions

A dynamic auction procedure:

1. Initially set the price vector $\mathbf{p}$ to a starting price vector $\mathbf{p}^0$.
2. Ask each buyer i to report her demand correspondence $D_i(\mathbf{p})$.
3. Test if there is excess demand. If no, stop. Otherwise, raise the price of each item with excess demand by one and go to step (2).

Question: If the procedure guarantees to converge to an Walrasian equilibrium?

Some results

- Walrasian equilibria do not always exist [Gul and Stacchetti 1999].
- Walrasian equilibria exist if each buyer's valuation function satisfies the following GS condition [Gul and Stacchetti 2000]:

  Gross substitutes condition: *if the price of some items was increased, the demand for the items which price has not been increased remains the same.*
- This condition implies substitutability.

Question: If each bidder's valuation satisfies GS condition, whether does a dynamic auction procedure converges to an Walrasian equilibrium? **Yes**

Gross substitutes and complements

- Gross substitutes and complements (GSC) condition: *if all the items can be divided into two categories, say software and hardware, increasing the prices of items in one category and decreasing the prices of items in the other category would not affect the demand of the items that prices are not changed.* [Sun and Yang 2006].

Double-track auction

1. Let $X = X_1 \cup X_2$ and $X_1 \cap X_2 = \emptyset$.
2. Set initial price of each item in X_1 to be zero and the initial price of each item in X_2 to be an artificial high price.
3. Ask each buyer i to submit her demand correspondence $D_i(\mathbf{p})$.
4. If there is no excess demand for items in X_1 and there are no residual items in X_2, stop. Otherwise, increase the price of each item in X_1 with excess demand by 1 and decrease the price of each item in X_2 with no demand by 1. Go to step (2).

Theorem

If each bidder's valuation satisfies GSC with the items of two categories X_1 and X_2, the above algorithm converges to a Walrasian equilibrium [Sun and Yang 2009]. The complexity of the algorithm is polynomial [Zhang *et al.* 2010]