Baffling Raffling Debaffled [version 2011-12-17 08:07]

Preston McAfee
Yahoo! Research

[Editor's Note: The mechanism described in the puzzle is sometimes known as a Chinese Auction. It is also equivalent, as McAfee points out, to a special case of a Cournot problem. An alternative formulation is: I decide a bid x, pay it in full, and then win the good with probability x/X where X is the sum of all the bids. Generalizing the question in the original puzzle, this solves the game for an arbitrary vector of common-knowledge valuations, i.e., the complete-information case with n agents.]

Notation: x_i is i's bid (the number of tickets bought by i) and v_i is the value of i, indexed so that $v_1 \geq v_2 \geq \ldots$. Let

$$X_{-i} = \sum_{j \neq i} x_j \text{ and } X = \sum_j x_j.$$

First, note that the payoff to i, given the choices of others, is $\dfrac{x_i}{x_i + X_{-i}} v_i - x_i$. The choice of x_i is restricted to $x_i \geq 0$, and probably should be restricted to integers. I will ignore this constraint. [This turns out to be moot for the specific (carefully constructed) valuations given in the puzzle.] Note that the individual maximization problem is equivalent to maximizing

$$\frac{x_i}{x_i + X_{-i}} - \frac{1}{v_i} x_i \equiv p(X) x_i - c_i x_i,$$

where $p(X) = \frac{1}{X}$ and $c_i = \frac{1}{v_i}$. The solution to the problem is just the solution to the standard constant marginal cost Cournot problem, with a unitary elasticity demand curve and asymmetric firms. While this is a common graduate student exercise, the solution isn't necessarily well-behaved.

To characterize the equilibria, return to the profit functions $\frac{x_i}{x_i + X_{-i}} - c_i x_i$. This function is concave, so the first order conditions characterize the maximum. The first derivative is $\frac{X_{-i}}{(x_i + X_{-i})^2} - c_i = \frac{X - x_i}{X^2} - c_i$. As the values of c_i increase in i (being the reciprocals of the v's), there will be a value n so that the first n have $x_i > 0$ and all others have $x_i = 0$. Note that all the agents with positive production have a zero first order condition, or $\frac{X - x_i}{X^2} - c_i = 0$. Summing these gives

$$0 = \frac{nX - X}{X^2} - \sum_{i=1}^{n} c_i,$$

which solves for

$$\frac{1}{X} = \frac{1}{n-1} \sum_{i=1}^{n} c_i,$$

and note immediately from the first order conditions that $n > 1$. An equilibrium has been achieved if, given this value of X, the first n firms want to enter and produce positive amounts and no others do, which is equivalent to

$$\frac{1}{X} - c_n \geq 0 \geq \frac{1}{X} - c_{n+1} \text{ or}$$
$$c_n \leq \frac{1}{X} \leq c_{n+1} \text{ or}$$
$$c_n \leq \frac{1}{n-1} \sum_{i=1}^{n} c_i \leq c_{n+1}.$$

Once we have an equilibrium number of agents and $\frac{1}{X} = \frac{1}{n-1} \sum_{i=1}^{n} c_i$, we can use the first order conditions $0 = \frac{X - x_i}{X^2}$, or $x_i = X - c_i X^2$ to generate the number of tickets purchased.

Using the Mathematica functions below, that yields $\langle 119, 77, 21, 0 \rangle$ with profits of $\langle 144.5, 42.35, 2.25, 0 \rangle$.

Is the solution unique? Let $p_n = \sum_{i=1}^{n} c_i$. The coumputation given shows

$$c_n \leq p_n$$
$$\iff c_n \leq p_{n-1}$$
$$\implies c_{n-1} \leq p_{n-1}.$$

Thus take the largest equilibrium n^{**}. For all k smaller,

$$c_k \leq p_k.$$

But consider any hypothetical smaller equilibrium n^*. As shown it satisfies

$$p_{n^*+1} \leq c_{n^*+1}$$

This would be a contradiction except for ties. If the c's were strictly increasing we would have the first inequality strictly and be done. If some c's are equal, the additional firms/agents produce/bid zero (since we are satisfying the price inequality with equality) and can be safely ignored.

The final question: how did the profits compare? The profit vector (seller, buyers) was $\langle 336.35, 144.15, 0, 0, 0, 0 \rangle$, and under the raffle it is $\langle 217, 144.5, 42.35, 2.25, 0, 0 \rangle$. So Nora gained the most.

Implementation of the solution in Mathematica

Following is Mathematica code to compute the equilibrium bids and profits. The helper function `bz` gives the hypothetical equilibrium bids (as a function of the vector of values) if we knew all agents would, in equilibrium, participate. Another helper function, `bs`, gives the equilibrium bids, without assuming full participation, if the values are in ascending order, which is of course WLOG. The `bs` function works by recursively re-solving for equilibrium bids with the subset of agents for which `bz` yields positive bids. Finally, `bids` gives the equilibrium bids for arbitrary values (by just sorting, calling `bs`, and then unsorting). Additionally, `prof` gives the expected profit to each agent in equilibrium.

```
bz[v_]:= With[{n = Length[v], r = Total[1/v]}, (n-1)(r-(n-1)/v)/r^2]
bs[v_]:= With[{x = bz[v]}, If[x[[1]]<0, Prepend[bs[Rest[v]], 0], x]]
bids[v_]:= bs[Sort@v][[Ordering@Ordering@v]]
prof[v_]:= With[{b = bids[v]}, v*b/Total[b] - b]
```