

Micro Economics B Exam

Exam nr: 60

June 2025

Use of AI

I have used GitHub Copilot as support for writing code. I have used ChatGPT in the writing process by drafting the text myself and then asking the algorithm to suggest improvements, some of which I have implemented.

1.1 Matrix Games

1.1.1

RESULT

We consider both Pure Strategy Nash Equilibria (PSNE) and Mixed Strategy Nash Equilibria (MSNE) since the game is simultaneous with complete information.

For $\theta = 0$:

- PSNE: $\{(M, M), (H, H)\}$
- MSNE: $\{(\sigma_1 = (0, 0.8, 0.2), \sigma_2 = (0, 0.8, 0.2))\}$

For $\theta = 3$:

- PSNE: $\{(M, M), (H, H)\}$
- MSNE: $\{(\sigma_1 = (0, 0.2, 0.8), \sigma_2 = (0, 0.8, 0.2))\}$

METHOD

We used Nashpy's support enumeration method to find all Nash equilibria. This method is guaranteed to find all PSNE and MSNE in finite games, though it may be computationally intensive for larger strategy spaces. Since this is a small 3x3 bimatrix game, it is well-suited.

1.1.2

RESULT

Refinement: Subgame-Perfect Nash Equilibrium (SPE) obtained by backward induction.

- For both $\theta = 0$ and $\theta = 3$, Birk's best reply to V, M, H is H, M, H respectively. Anticipating this, Ask chooses H .
- Therefore the unique SPE in each case is (H, H) .

First-mover advantage.

$\theta = 0$: Sequential payoff to Ask = 5, Simultaneous (best PSNE) = 5 \Rightarrow no advantage.

$\theta = 3$: Sequential payoff to Ask = 5, Simultaneous PSNE payoffs $\{2, 5\}$ \Rightarrow **advantage of moving**

METHOD

When $\theta = 0$, the game is symmetric, and the pure strategy Nash equilibria— (M, M) and (H, H) —yield identical payoffs to both players: $(2, 2)$ and $(5, 5)$, respectively. In the sequential version of the game, Player 1 (Ask) anticipates that Player 2 (Birk) will respond with H if he plays H , making (H, H) the preferred equilibrium. Since both equilibria benefit the players equally and the game is symmetric, there is no first-mover advantage.

When $\theta = 3$, the game becomes asymmetric. The two pure strategy Nash equilibria now yield unequal payoffs: (M, M) results in $(2, 5)$, while (H, H) results in $(5, 2)$. In the sequential version, Ask can anticipate Birk's best response in each subgame and will choose H , securing the $(5, 2)$ outcome. This results in a first-mover advantage, as Ask avoids the risk of ending up with only 2 (as could happen in the simultaneous game). Thus, while the equilibrium action remains the same, the distribution of payoffs creates

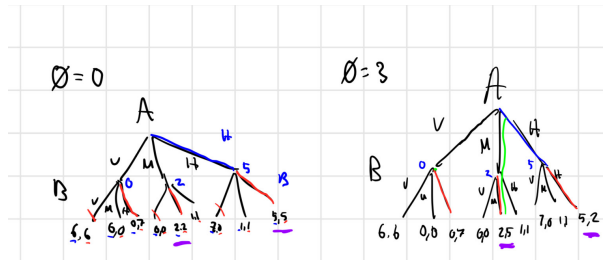


Figure 1: Backwards Induction

a strategic advantage for moving first.

1.1.3

RESULT

Equilibrium refinement: Perfect Bayesian Equilibrium (PBE), we have a Bayesian game.

$$BNE = \{(M, MM), (H, HH)\}$$

Mixed BNE:

$$P(a_1 = M) = 0.8, P(a_1 = H) = 0.2,$$

$$P(a_2 = MM) = 0.6, P(a_2 = HM) = 0.4$$

METHOD

We write the game on wide matrix form, we run IESDS and get:

| | MM | HM | HH |
|---|------------|------------|------------|
| M | (2.0, 3.5) | (1.5, 3.0) | (1.0, 1.0) |
| H | (1.0, 1.0) | (3.0, 3.0) | (5.0, 3.5) |

We solve this using support enumeration and find the above result. IESDS removes the strictly dominated strategies and reduces the problem for nash py support enumeration to solve.

1.1.4

RESULT

Refinement: Subgame Perfect Nash Equilibrium (SPNE). Yes, there exists an SPNE in the repeated game $G(2)$ with $\theta = 0$ where (V, V) is played in the first period.

Although (V, V) is not a Nash equilibrium of the stage game, it can be sustained by punishing any deviation with (M, M) in the second period. Since both (M, M) and (H, H) are valid stage-game equilibria, this threat is credible. Given the payoffs, no player gains from deviating, so (V, V) in period 1 is part of an SPNE.

METHOD

We use the following strategy:

- Period 1: play (V, V)
- Period 2: play (H, H) if no deviation, otherwise (M, M)

We find out if this is sustainable by comparing payoffs for deviating.

| Period 1 Action | P1 Payoff | P2 Payoff | Total |
|----------------------|-----------|-----------|----------------|
| Both play V | (6, 6) | (5, 5) | (11,11) |
| Ask deviates to H | (7, 0) | (2, 2) | (9,2) |
| Birk deviates to H | (0, 7) | (2, 2) | (2,9) |

Table 1: Total payoffs with and without deviation from (V, V)

Deviating gives at most 1 extra point in period 1 but leads to a 2-point loss in period 2. Thus, no profitable deviation exists, and the strategy forms an SPNE.

1.2 Signaling Games

1.2.1

RESULT

There exists an equilibrium where the student always reports and the insurance company always accepts:

$$\text{PBSE} = \left\{ (JJ, AA, -45 > 9B - 10K, q = \frac{9}{10}) \right\}$$

Solving for K we get that for the equilibrium to exist we need:

$$K > 0.9B + 4.5$$

In other words the cost of looking into the claim is potentially larger than the potential fine for insurance fraud, f.ex $B=10,000$ dkk gives $K \geq 13,500$ dkk. This seems expensive since I imagine looking into bike claims is fairly standard and straight forward. Hence it is not realistic.

METHOD

Separating NJ

(2) P_2 expected payoff

$$1 - q = \mathbb{P}(\emptyset_S | J) = 1 \quad (\text{since NJ})$$

$$\mathbb{E}[u_2(A | J)] = -5 \quad \leftarrow \text{BR A}$$

$$\mathbb{E}[u_2(V | J)] = -5 - K$$

(3) P_1 , Deviators

| P_1 type | P_1 Action | P_2 Action | P_1 dev > Not dev |
|---------------|--------------|--------------|---------------------------------|
| \emptyset_S | N | A | $5 > 0 = T \rightarrow$ deviate |

Not part of equilibrium

Pooling JJ**(2) Payoffs P_2**

$$q = \mathbb{P}(\emptyset_E | J) = \frac{9}{10}$$

$$\mathbb{E}[u_2(A | J)] = \frac{9}{10}(-5) + \frac{1}{10}(-5) = -5$$

$$\mathbb{E}[u_2(U | J)] = \frac{9}{10}(B - K) + \frac{1}{10}(-5 - K) = \frac{9B - 10K - 5}{10}$$

J chooses A iff

$$-5 > \frac{9B - 10K - 5}{10} \implies -45 > 9B - 10K$$

J chooses U iff

$$\frac{9B - 10K - 5}{10} > -5 \implies 9B - 10K > -45$$

(3) Profitable deviations P_1

| P_1 type | P_1 act. | P_2 act. | Ineq. | dev > no-dev | Profitable? |
|---------------|------------|------------|------------------|--------------|-------------|
| \emptyset_E | N | A | $-45 > 9B - 10K$ | $0 > 5$ | F |
| \emptyset_E | N | U | $9B - 10K > -45$ | $0 > -B$ | $B > 0$ (T) |

Pooling PBSE

$$\text{PBSE} = \left\{ (JJ, AA, -45 > 9B - 10K, q = \frac{9}{10}) \right\}$$

$$= \frac{\sigma_1 \cdot \frac{9}{10}}{\sigma_1 \cdot \frac{9}{10} + 1 \cdot \frac{1}{10}}$$

P_2 's Updated Belief

$$q = \mathbb{P}(\emptyset_E | J) = \frac{K}{B + 5}$$

$$\frac{K}{B + 5} = \frac{\sigma_1 \cdot \frac{9}{10}}{\sigma_1 \cdot \frac{9}{10} + 1 \cdot \frac{1}{10}}$$

$$\sigma_1 = \frac{K}{9(B - K + 5)}$$

Mixed PBSE

$$\text{PBSE} = \left\{ \left(\sigma_1 = \frac{K}{9(B - K + 5)}, \sigma_2 = \frac{B}{5 + B}, q = \frac{K}{B + 5} \right) \right\}$$

1.2.3

The semi-separating PBE depends on the fine B and investigation cost K as follows:

$$\sigma_1 = \frac{K}{9(B - K + 5)}, \quad \sigma_2 = \frac{B}{B + 5}, \quad q = \frac{K}{B + 5}$$

Effect of increasing B : A higher fine reduces the incentive to file false claims, hence σ_1 decreases. Since fraud becomes rarer, the insurer is more willing to accept claims ($\sigma_2 \uparrow$), and its belief that a claim is fraudulent falls ($q \downarrow$).

Effect of increasing K : A higher investigation cost weakens the insurer's threat, encouraging more false claims ($\sigma_1 \uparrow$). The insurer's strategy σ_2 remains unaffected, but its belief that a claim is fraudulent rises ($q \uparrow$).

The equilibrium reflects a trade-off between deterrence (via B) and enforcement costs (via K). Stronger punishment deters fraud; higher enforcement costs embolden it.

2 Hearth Stone

2.1

0.0.1 RESULT

| k | Action | Deck |
|----------|---------------|----------------|
| 1 | 17 | Fishing Priest |
| 2 | 8 | Taunt Warrior |
| 3 | 30 | Space Warlock |
| 4 | 1 | Imbue Hunter |
| 5 | 17 | Fishing Priest |
| 6 | 8 | Taunt Warrior |
| 7 | 30 | Space Warlock |
| 8 | 1 | Imbue Hunter |
| 9 | 17 | Fishing Priest |
| 10 | 8 | Taunt Warrior |

Table 2: Deck performance data with actions and winrates

We see a repeating pattern of Fishing Priest, Taunt Warrior, Space Warlock and Imbue Hunter. That consecutively are the best responses to each other.

METHOD

Overall K 'th level action is just K consecutive best responses, I implemented this recursively. A level-0 player picks actions uniformly at random. A level-1 player best-responds to this, choosing the action with the highest expected payoff against a uniform opponent. A level- k player, for $k > 1$, assumes the opponent is level- $(k-1)$, recursively predicts their action, and best-responds to that. This yields a chain of nested best responses, where each level refines its strategy based on deeper reasoning. The algorithm returns the pure strategy maximizing utility given the predicted opponent move. This way we can compute the K 'th level reasoning.

2.2

RESULT

The best strategy against an online opponent who follows the empirical distribution of deck frequencies, σ_{data} , is to play **Fishing Priest**. This deck yields the highest expected win probability at 59.464%.

METHOD

The expected win probability for each deck was calculated as the dot product of the winrate matrix U and the empirical strategy vector σ_{data} . The deck with the highest expected payoff was identified as the best pure strategy.

2.3

RESULT

| Deck | Equilibrium probability | Best response within support |
|--------------------|-------------------------|------------------------------|
| Imbue Paladin | 0.157 | Fishing Priest |
| Leech Death Knight | 0.133 | Space Warlock |
| Terran Warrior | 0.095 | Space Warlock |
| Fishing Priest | 0.502 | Leech Death Knight |
| Protoss Priest | 0.019 | Imbue Paladin |
| Space Death Knight | 0.003 | Terran Warrior |
| Space Warlock | 0.091 | Imbue Paladin |

Table 3: NE and BR in EQ support.

METHOD

Why Hearthstone is a zero-sum game

In the considered format of Hearthstone, each match is a one-on-one contest with a binary outcome: one player wins and the other loses. There are no draws in the dataset, and no partial wins or points awarded to both players. Therefore, any gain in win probability for one player corresponds directly to a loss for the opponent. This strict competition structure justifies modeling the interaction as a zero-sum game, where the payoffs to the two players always sum to one. The winrate matrix U naturally represents this by

recording player one's win probabilities, and the opponent's payoff is simply $-U^T$. TODO: double check if 1-U or -U.

Why use linear programming

In a two-player zero-sum game, the minimax theorem guarantees that a Nash equilibrium exists and can be found by solving a linear optimization problem. The objective is to find a mixed strategy that maximizes the player's minimum expected payoff against any strategy of the opponent. Since the problem involves optimizing a linear function (expected winrate) subject to linear constraints (probabilities summing to one and non-negativity), linear programming provides an efficient and exact method for computing the equilibrium. This approach is especially suitable for games with a large but finite number of actions, such as Hearthstone with 32 decks.

2.4

RESULT

Hypothesis: I believe that the pairs of decks d1, d2 that will have a high unreliability in the winning probability and that i expect will affect the calculated NE the most are:

- Decks with high simply unreliability: Those with few samples.
- Deck affecting eq the most: d1 will be a deck in the NE support and d2 will be some deck against which d1s win rate that is crucial for keeping that deck in the support. I believe that there could be "tipping points" where small changes in single win rates could cause large overall changes, especially if being included hinges on this small delta.

Quantitative measure for "affecting the NE the most": I choose to measure the absolute total change in the probability vector for selecting a deck when in NE, given some change δ in the percentage point winrate of d_1 against d_2 , making us go from $p_{currentEQ}$ to p_{newEQ} . So we measure how sensitive the EQ is to a change for a given win rate. The change is in percentage points not percent, this is to take into account a lower sample quality for decks with low playrate:

$$L_1 = abs(p_{currentEQ} - p_{newEQ})$$

The Quatitative measure for affecting EQ "the most" is hence:

$$\operatorname{argmax}_{d_1, d_2}(L_1)$$

A table with the 15 decks with lowest palyrates is included. It is plausible that we find a relatively larger percent change when adding delta percentage points for these decks resulting a larger chain reaction of changes for everything that hinges on those probabilities.

| Deck | Playrate |
|----------------------------|----------|
| FFU Menagerie Death Knight | 0.007755 |
| Space Warlock | 0.008219 |
| Space Death Knight | 0.011386 |
| Templar Rogue | 0.011415 |
| Protoss Priest | 0.012138 |
| Dollhouse Druid | 0.012464 |
| Imbue Mage | 0.013257 |
| Menagerie Paladin | 0.013664 |
| Spell Mage | 0.014441 |
| Zerg Token Hunter | 0.015240 |
| Handbuff Death Knight | 0.015270 |
| Evolve Shaman | 0.015392 |
| Terran Shaman | 0.015929 |
| Protoss Imbue Mage | 0.016325 |
| Fishing Priest | 0.016413 |

Table 4: Least played Hearthstone decks by playrate

TODO: add method?

0.1 2.5

RESULT

After running a simulation we conclude that a percentage point $\delta = +3\%$ change in the winrate of Protoss Priest against Fishing Priest can cause a total absolute value of changes in the NE selection of decks beeing $L_1 = 0.86$, and it cuases Leech Death Knight to leave the support.

We observe that this largest L1 change did cause a deck to leave the support, which was our hypothesis.

We also observe that both Protoss Priest and Fishing Priest has low playrates, which could mean that the sample quality is lower. This is a cause for worry for Ali, since it makes the type of change we saw seem quite plausible and it casts uncertainty over how much we can conclude based on this type of analysis. We do observe in the table from 2.4 that both Fishing Priest and Protoss priest can be found in the table of decks with the lowest playrate, although this alone is not the cause for the effect, but somewhat confirms that the larger relative change when adding delta has some effect for causing more overall change in NE probabilities.

METHOD

I ran a test where I loop through the decks in the NE support and change all other decks winrate by 3 percent against them.

Table 5: Most influential change: Protoss Priest til Fishing Priest

| Parameter | Værdi |
|--------------------------|----------------|
| Deck i (from support) | Protoss Priest |
| Deck j (receives change) | Fishing Priest |
| L1-change | 0.860711 |
| Original winrate | 0.4841 |
| New win rate | 0.5141 |

Table 6: Probability changes in the support

| Deck | Original | New | Change |
|--------------------|----------|--------|---------|
| Imbue Paladin | 0.1574 | 0.0272 | -0.1302 |
| Leech Death Knight | 0.1329 | 0.0000 | -0.1329 |
| Terran Warrior | 0.0954 | 0.1489 | +0.0535 |
| Fishing Priest | 0.5018 | 0.3568 | -0.1451 |
| Protoss Priest | 0.0188 | 0.1172 | +0.0985 |
| Space Death Knight | 0.0028 | 0.1255 | +0.1227 |
| Space Warlock | 0.0910 | 0.0687 | -0.0223 |

The simulation The simulation analyzes the sensitivity of the symmetric zero-sum game equilibrium to local changes in the winrate matrix. The original winrates are normalized and used to compute a mixed-strategy Nash equilibrium via linear programming. This is done by maximizing the minimum guaranteed payoff using standard simplex constraints.

For each deck in the support of the equilibrium (probability ≥ 0.001), the winrate against all other decks is increased by a fixed amount (here, $\Delta = 0.03$), and the winrate matrix is

updated to maintain symmetry. A new equilibrium is computed, and the total change in the strategy distribution is measured using the L1 norm. The pair of decks that induces the largest shift in equilibrium is identified.

3 Dynamic Price Competition

| Market | Nash_p | Nash_q | Nash_π | Cartel_p | Cartel_q | Cartel_π | Dev_p | Dev_π | δ* |
|----------|--------|--------|--------|----------|----------|----------|-------|-------|-------|
| A | 2.227 | 0.185 | 0.227 | 2.463 | 0.158 | 0.232 | 2.236 | 0.236 | 0.480 |
| B | 1.496 | 0.496 | 0.246 | 2.312 | 0.405 | 0.531 | 1.956 | 0.706 | 0.380 |
| Combined | 1.630 | 0.750 | 0.473 | 2.326 | 0.574 | 0.761 | 1.981 | 0.937 | 0.379 |

Table 7: Market outcomes under Nash, Cartel, and Deviation strategies

3.1

RESULTS

For each market $m \in \{A, B\}$ the numerically computed symmetric Nash-equilibrium prices and quantities are

$$(p_{1,m}^*, p_{2,m}^*) = (2.227, 2.227) \text{ in A, } (1.496, 1.496) \text{ in B.}$$

Corresponding quantities and per-firm profits are $q_{i,A}^* = 0.185$, $\pi_{i,A}^* = 0.227$ and $q_{i,B}^* = 0.496$, $\pi_{i,B}^* = 0.246$ (see Table ??).

METHOD

For each market we solved the fixed-point problem $p_i = \text{BR}_i(p_j)$ by iterated best response. Each best-response BR_i is obtained from a one-dimensional profit maximisation

$$\max_{p_i \geq 0} (p_i - 1) q_i(p_i, p_j),$$

evaluated with `scipy.optimize.minimize` (L-BFGS-B). Convergence to tolerance 10^{-8} was reached in under 15 iterations.

3.2

RESULTS

Joint-profit maximisation yields cartel prices

$$(p_{1,m}, p_{2,m}) = (2.463, 2.463) \text{ in A,} \quad (2.312, 2.312) \text{ in B,}$$

with quantities $q_{i,A} = 0.158$, $\pi_{i,A} = 0.232$ and $q_{i,B} = 0.405$, $\pi_{i,B} = 0.531$.

METHOD

The cartel solves $\max_{p_1, p_2} \pi_1(p_1, p_2) + \pi_2(p_1, p_2)$, again via `minimize` (negated objective, L-BFGS-B). Symmetry of the demand system implies $p_1 = p_2$ in optimum, but symmetry was *not* imposed—numerics confirm it.

3.3

RESULTS

The critical discount factors are

$$\delta_A^* = 0.480, \quad \delta_B^* = 0.380.$$

METHOD

Under the grim-trigger strategy GT, unilateral deviation is profitable only if

$$\pi_D + \delta \frac{\pi_N}{1 - \delta} > \pi_C + \delta \frac{\pi_C}{1 - \delta},$$

which rearranges to the threshold $\delta^* = (\pi_D - \pi_C)/(\pi_D - \pi_N)$. Here π_N and π_C are Nash and cartel profits, while π_D is the deviator's one-shot best response against the rival's cartel price. All three profits are taken from the previous steps; π_D is computed with the same BR routine.

Intuition. Market B exhibits a larger profit gap ($\pi_C - \pi_N$) and thus a harsher future punishment for deviation, lowering the patience required for collusion. Hence $\delta_B^* < \delta_A^*$.

3.4

RESULTS

With one common price the outcomes are

$$\begin{aligned}\text{Nash: } p^* &= 1.630, & q^* &= 0.750, & \pi^* &= 0.473; \\ \text{Cartel: } p &= 2.326, & q &= 0.574, & \pi &= 0.761; \\ & & \delta^* &= 0.379.\end{aligned}$$

METHOD

Demand aggregates to $q_i(p_i, p_j) = q_{i,A} + q_{i,B}$. All three steps (Nash, cartel, deviation) are repeated with the aggregated profit function. Computation is fully analogous to the individual-market analysis.

3.5

Markets: Market A ($\alpha_A = 1, \gamma_A = -1$) is small and only mildly elastic, whereas Market B ($\alpha_B = 10, \gamma_B = -4$) is large and highly elastic. Because $|\gamma_B| > |\gamma_A|$, customers in B react much more to price changes.

Nash pricing: With separate prices, firms charge more in A ($p_A^* = 2.23$) than in B ($p_B^* = 1.50$). If a single price is required for both markets, the equilibrium settles at $p^* = 1.63$, pulled toward B's level because B delivers most of the volume.

Cartel pricing: A cartel allowed to price-discriminate chooses $p_A = 2.46$ and $p_B = 2.31$. Under one uniform price, it picks $p = 2.33$, again leaning toward the profitable level for the larger market B.

In general, uniform pricing leaves money on the table by ignoring each market's specific demand, yet it also shrinks the benefit from cheating, so collusion is easier to maintain (lower required δ^*).

4. Auction Formats

4.1

RESULT

| N | $\mathbb{E}[p_t]$ | $\text{sd}(p_t)$ | $ \Delta $ |
|----------|-------------------|------------------|---------------|
| 3 | 1.6667 | 1.2021 | 0.6179 |
| 4 | 2.1692 | 1.3053 | 0.0123 |
| 5 | 2.5683 | 1.3654 | 0.4470 |
| 6 | 2.9002 | 1.4014 | 0.8148 |
| 7 | 3.1892 | 1.4329 | 1.1354 |
| 8 | 3.4391 | 1.4550 | 1.4073 |

$$\boxed{\hat{N} = 4}$$

METHOD

We simulate 10^5 second-price sealed-bid (SPSB) auctions for each $N \in \{3, \dots, 8\}$, where each bidder's valuation v_i is drawn i.i.d. from a $\Gamma(1, 2)$ distribution. The auction payment in each simulation is the second-highest valuation.

For each N , we compute the sample mean and standard deviation of the payments. We then compute the total absolute error relative to the observed auction data:

$$|\bar{p}_{\text{sim}} - \bar{p}_{\text{obs}}| + |\text{sd}_{\text{sim}} - \text{sd}_{\text{obs}}|.$$

The value \hat{N} that minimizes this total error is selected as the best estimate of the true number of bidders. The simulation uses fixed random seeds to ensure comparability across N .

4.2

RESULT

The function $b^*(v)$ represents the optimal bidding strategy in a first-price auction. A bidder with valuation v should bid $b^*(v) < v$ to balance two competing forces: bidding

high enough to win against competitors, but low enough to earn positive profit. The integral term captures the expected highest competing bid, ensuring the strategy forms a Nash equilibrium where no bidder benefits from deviating.

| v | $b^*(v)$ |
|-----|----------|
| 1 | 0.7105 |
| 2 | 1.3357 |
| 3 | 1.8693 |
| 4 | 2.3107 |

METHOD

In a first-price auction with N bidders who are risk-neutral and know only their own value, the symmetric Bayesian–Nash equilibrium bidding strategy is given by

$$b^*(v) = v - \frac{\int_0^v F(x)^{N-1} dx}{F(v)^{N-1}}, \quad v > 0,$$

where F is the cumulative distribution function (CDF) of the valuation distribution.

In our case, valuations follow a Gamma(1, 2) distribution, so $F = \Gamma(1, 2)$, and we use the estimated $\hat{N} = 4$. The integral is calculated using numerical integration (specifically `scipy.integrate.quad` in Python). We then compute $b^*(v)$ for $v = 1, 2, 3, 4$. This method does not require any optimisation, only integration.

4.3

RESULT

| N | $\mathbb{E}[\text{Rev} \mid \text{SPSB}]$ | $\mathbb{E}[\text{Rev} \mid \text{FPSB}]$ | Diff | Ratio |
|-----|---|---|---------|-------|
| 3 | 1.6779 | 1.6547 | −0.0232 | 0.986 |
| 4 | 2.1506 | 2.1638 | +0.0132 | 1.006 |
| 5 | 2.5523 | 2.5687 | +0.0164 | 1.006 |
| 6 | 2.9077 | 2.8878 | −0.0199 | 0.993 |
| 7 | 3.1848 | 3.1727 | −0.0121 | 0.996 |
| 8 | 3.4302 | 3.4468 | +0.0166 | 1.005 |

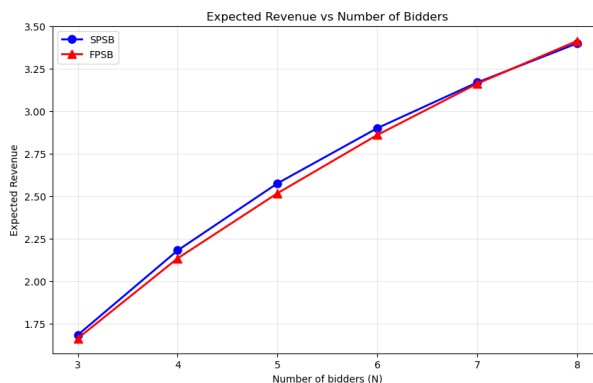


Figure 3: Revenue Expected to be the same

METHOD

For each $N \in \{3, \dots, 8\}$ I simulate 10 000 auctions under:

- the SPSB rule (payment = second-highest valuation),
- the FPSB rule using the equilibrium bid function from part 2 (payment = winning bid).

Recommendation. Given the simulation yielding the above table we can conclude that revenue equivalence holds, meaning that we expect the same revenue regardless of whether we are using FPSB or SPSB, see in the figure how the curves follow each other. The auction format in itself is not going to affect the expected return, however it might cause difference in variance, and different auction types can also be preferred by different bidders in different markets. If the price of the good is more or less objective and known SPSB can work well, it is also well suited for inexperienced bidders since it does not require any bid shading like in FPSB that relies on some distribution assumptions. SPSB may also help a bit against winners curse by picking the second highest bid, although one has to expect that there can be two bidders doing the wrong calculation and hence this is not a guarantee. Overall I would recommend to use the auction type that will attract the most bidders, and be most well received by them, since the nr of bidders, not like auction format, can increase the expected return to the auctioneer.

4.4

RESULT

The revenue-maximising reserve price in a second-price sealed-bid (SPSB) auction with the estimated bidder count $\hat{N} = 4$ is

$$R_{\hat{N}}^* = 2.00 \quad (\text{units of value}), \quad \Pi(R_{\hat{N}}^*, \hat{N}) = 2.25.$$

Figure displays the criterion function $\Pi(R, \hat{N})$ and highlights $R_{\hat{N}}^*$.

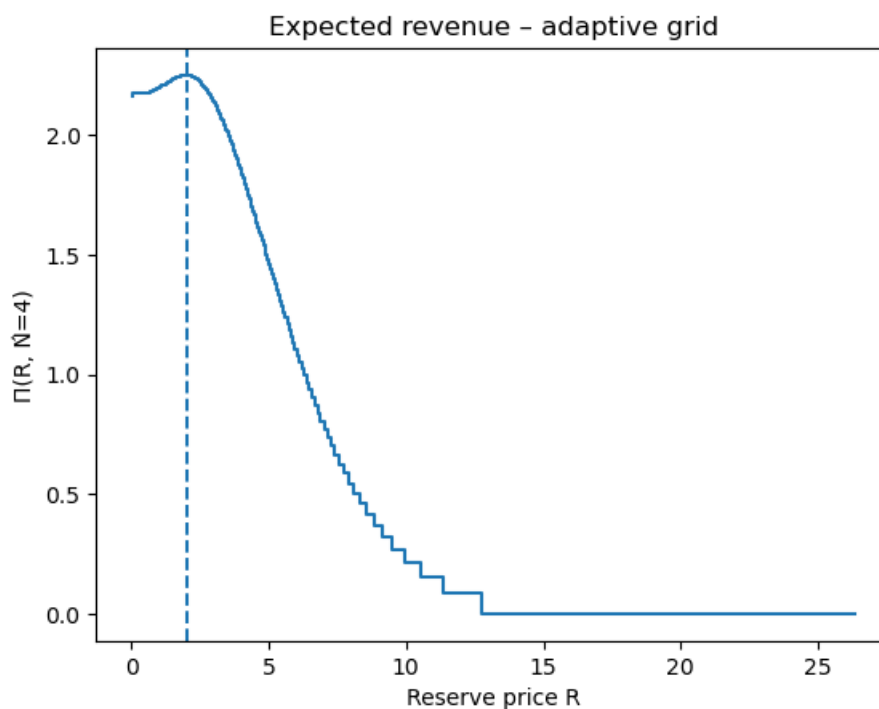


Figure 4: Criterion function

METHOD

- (i) **Monte-Carlo framework.** Expected revenue $\Pi(R, N)$ was evaluated by simulation: 5×10^5 independent auction draws with valuations $\{v_i\}_{i=1}^N \stackrel{\text{i.i.d.}}{\sim} \Gamma(1, 2)$.
- (ii) **Adaptive grid search.** When building this simulation some numerical instability was observed, as different runs with different random seeds would yield different results. I adopted a grid with less points in the flat sections and more point in the curved areas, so more detail where I would be doing my evaluation for max. The grid was built on empirical quantiles of the *maximum* order statistic $v_{(1)}$. Specifically,

150 quantiles of $v_{(1)}$ spanning 0–100% form R -candidates. This allocates more grid points in the economically relevant part of the support and may provide more detail and numerical stability around the value in question.

Revenue calculation. (From lecture Auctions III p.21) Let $v_{(1)} = \min\{v_1, \dots, v_N\}$ and $v_{(2)} = \max\{v_1, \dots, v_N\}$ denote the order statistics of the valuations. With a reserve price R the winner's payment is

$$B(R, v_{(1)}, v_{(2)}) = \begin{cases} v_{(1)}, & R \in [0, v_{(1)}), \\ R, & R \in [v_{(1)}, v_{(2)}), \\ 0, & R \geq v_{(2)}. \end{cases}$$

Hence the expected revenue is

$$\Pi(R, N) = \Pr(v_{(2)} < R) \cdot 0 + \Pr(v_{(1)} < R < v_{(2)}) \cdot R + \Pr(v_{(1)} \geq R) \mathbb{E}[v_{(1)} \mid v_{(1)} \geq R].$$

For each R we compute payments average over simulations, and pick the R with the highest mean. As the optimisation is a one-dimensional discrete arg-max on a compact grid, convergence is guaranteed and no further numerical optimiser is needed.

4.5

RESULT

| N | R_N^* | $\Pi(R_N^*, N)$ | $\Pi(R_N^*, N) - \Pi(0, N)$ |
|-----|---------|-----------------|-----------------------------|
| 3 | 1.92 | 1.837 | 0.170 |
| 4 | 2.00 | 2.255 | 0.089 |
| 5 | 2.04 | 2.615 | 0.050 |
| 6 | 2.03 | 2.930 | 0.030 |
| 7 | 2.08 | 3.200 | 0.012 |
| 8 | 2.03 | 3.447 | 0.010 |

Table 8: Optimal reserve and incremental revenue in SPSB

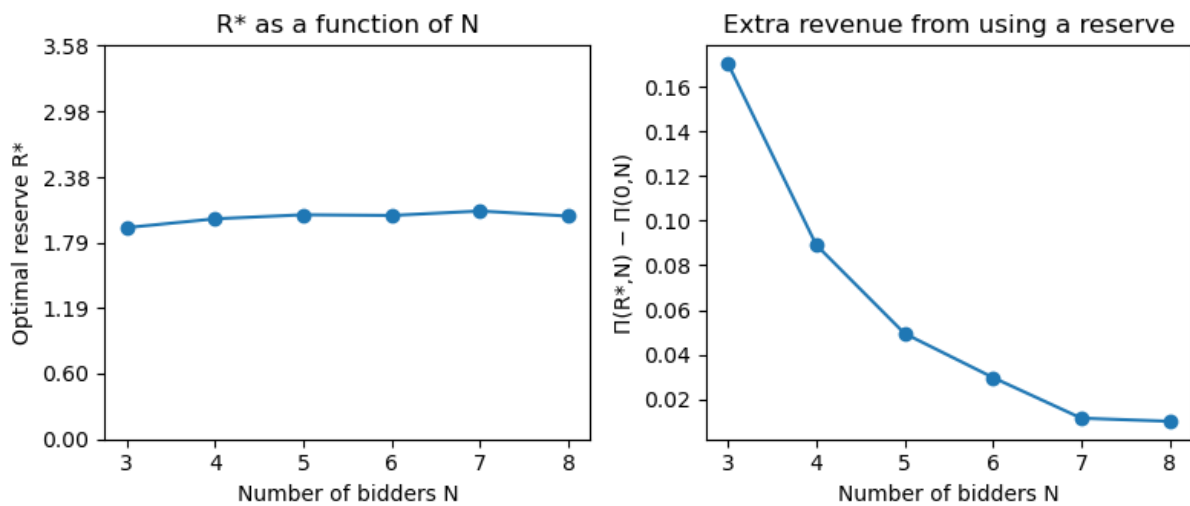


Figure 5: Optimal reserve and incremental revenue in SPSB plotted

Observations.

- Revenue vs N** The expected revenue depends on N in the expected way: more bidders imply more competition. As N increases, it becomes more likely that we observe all types of bids, including high ones. The range fills up with bids all the way to the top. In this way, a lower N might yield a lower probability of a very high bid, while a high N increases the likelihood that we see any given bid. That said, this is only probabilistic or we might get a high bid even with a low N ; it's just more certain with a higher N . This is why our function seems to jump around at

random, because it is partly random. However, our more effective grid search allows more runs in the simulation, which helps mitigate this slightly.

2. R^* vs N Observed R^* does not seem to depend on N . It appears constant around 2, regardless of N . Some googling and back-and-forth with GPT led me to this paper: <https://www.cs.princeton.edu/courses/archive/spr09/cos444/papers/myerson81.pdf>. The author is also mentioned in our book, though not this particular formula (5.1):

$$r^* = \text{solution to } c_i(t) = 0 \iff t = \frac{1 - F_i(t)}{f_i(t)}.$$

This may support the observed outcome that R^* does not depend on N , since the formula itself does not contain N . However, this is "above my pay grade", and I leave it as trivia. I cannot personally confirm whether this is true, but I found it interesting.

3. The *incremental* value of a reserve, $\Pi(R_N^*, N) - \Pi(0, N)$, declines sharply with N . As competition intensifies, the expected second-highest valuation rises, so the seller captures most surplus without any reserve. Conversely, with few bidders ($N = 3$), there is a sizable 9% revenue boost from screening out low-value contests. In other words, the reserve is only effective when there is a sizable risk of getting only low or zero bids. If we get bids above the reserve, it has no effect. So what we see is that the number of auctions with low or zero bids decline when N increases and hence the utility of our reserve price declines as well.

METHOD

For each $N \in \{3, \dots, 8\}$, we simulated 5×10^5 auctions and tested 150 different reserve prices using a grid based on the highest bids.

The revenue function $\Pi(R, N)$ changes only at certain points, so a step plot is enough. We avoid smoothing to keep results clear and stable.

Table 8 shows the best reserve price and the revenue for each N .