

Mikroøkonomi B Eksamen

Eksamensnummer 30

2023

1 Matrix Games

1.1

Rafinering: Nash-ligevægt.

For at bestemme Nash-ligevægte i spillet når $x \in \{1, 4, 10\}$ bruges `Support_enumeration` fra `nashpy`, som får de rene strategier (ingen MSNE):

$x = 1$:

$$PSNE = \{(I, I)\}$$

$x = 4$:

$$PSNE = \{(I, I), (I, U), (U, I)\}$$

$x = 10$:

$$PSNE = \{(I, U), (U, I), (U, U)\}$$

1.2

Rafinering: Nash-ligevægt.

For at bestemme alle Nash-ligevægte når $x \in \mathbb{R}$ kan vi blot se på hvornår ligevægtene skifter når x ændres.

For at bestemme dette, kan der ses på strengt dominerede strategier:

$x < 4$: Når x er mindre end 4, så strengt dominerer strategi I for Anders U. Herfra ved vi, at Benjamin kun vil spille I, da $4 - x > 0$ når $x < 4$. Vi ender derfor med Nash-ligevægten $\{(I, I)\}$ når $x < 4$.

$x = 4$: Ved vi fra opgave 1.1 har Nash-ligevægtene $\{(I, I), (I, U), (U, I)\}$.

$10 > x > 4$: Når x er mellem 4 og 10 har vi at $10 - x > 0$ og $0 > 4 - x$, og vi ender derfor med de Nash-ligevægtene i rene strategier til $\{(U, I), (I, U)\}$. Da vi ikke har brugt den strengt dominerede metode her, og vi som tommelfingerregel ved at der næsten altid findes et ulige antal Nash-ligevægte skal vi også kontrollere for blandede strategier:

For at bestemme hvad Anders skal blande med af sandsynlighed, så opstilles ligningerne der gør Benjamin indifferent:

$$\mathbb{E}U_I = \mathbb{E}U_U$$

$$\mathbb{E}U_I = f(\sigma_I)$$

$$\mathbb{E}U_U = f(\sigma_I)$$

Herfra løses σ_I for som Anders skal blande med

$$\mathbb{E}U_I = \sigma_I(4 - x) + (1 - \sigma_I)(10 - x)$$

¹Medmindre andet er opgivet, er kode til alle opgaver er lavet med udgangspunkt i udleveret kode fra github, fra tidligere hjemmeopgaver med min gruppe eller selvlavet.

$$\mathbb{E}U_U = \sigma_I \cdot 0 + (1 - \sigma_I) \cdot 0 = 0$$

$$\mathbb{E}U_I = \mathbb{E}U_U \iff \sigma_I(4 - x) + (1 - \sigma_I)(10 - x) = 0 \iff \sigma_I = \frac{10 - x}{6}$$

Og pga. symmetri skal Benjamin også blande med samme.

Dvs. den MSNE er $\{((\frac{10-x}{6}, 1 - \frac{10-x}{6}), (\frac{10-x}{6}, 1 - \frac{10-x}{6}))\}$.

$x > 10$: Når x er over 10 dominerer U strengt I for Benjamin, da $4 - 10 < 0$ og $10 - x < 0$. Det samme gælder for Anders, og den eneste Nash-ligevægt bliver derfor $\{(U, U)\}$.

Opsamlet fås alle Nash-ligevægte i både rene og blandede strategier for alle x i tabel 1.

x	PSNE	MSNE
$x < 4$	$\{(I, I)\}$	-
$x = 4$	$\{(I, I), (I, U), (U, I)\}$	-
$4 < x < 10$	$\{(U, I), (I, U)\}$	$\{((\frac{10-x}{6}, 1 - \frac{10-x}{6}), (\frac{10-x}{6}, 1 - \frac{10-x}{6}))\}$
$x = 10$	$\{(I, U), (U, I), (U, U)\}$	-
$x > 10$	$\{(U, U)\}$	-

Tabel 1: Alle ligevægte for alle x -værdier

1.3

Raffinering: Underspilsperfekt Nash-ligevægt (SPNE).

For at se hvad der sker i det sekventielle spil, skal der igen ses på hvad der sker når x er forskellige værdier. Når $x < 4$ vil spillet have SPNE $\{(I, (I, I))\}$, da Benjamin enten får $4 - x$ eller $10 - x$. Dette observerer Anders, og vælger derfor også at gå ind, da $4 - x > 0$.

Når $10 > x > 4$ ved Benjamin at Anders ikke vil kunne få over 0 ved at gå ind, og vil derfor udeblive. Dette sikrer Benjamin monopolet og får $10 - x$, og derfor SPNE $\{(I, (U, I))\}$. Når $10 > x$ vil begge udeblive, da det ikke er profitabelt længere, og derfor SPNE $\{(U, (U, U))\}$.

Vi kan nu sammenligne hvad Anders stod til at få når han ikke kendte til Benjamins handling og til han nu gør, da vi har de simultane ligevægte fra 1.2:

$x < 4$: Ligevægten ender begge med at de indtræder, og Anders er derfor indifferent om han får den information.

$10 > x > 4$: Her vil Benjamin altid indtræde i det sekventielle spil og Anders ender derfor altid med at skulle udeblive. Men før da det var simultan var der ligevægte hvor Anders stod med en positiv profit, og derfor ville Anders hellere have været uden informationen.

$x > 10$: Her ender de i samme ligevægt som ved det simultane, og Anders er derfor indifferent om han får den information.

1.4

a

Når Anders kender x_B på forhånd kan spillet blot løses som et simultan spil, som kan gøres ved brug af `support_enumeration` fra `nashpy`. Hertil findes også det forventede payoff, som kan bruges til sammenligning senere:

$x_B = 3$:

$$\{(U, I)\}$$

$$\mathbb{E}U((U, I)) = 0$$

$x_B = 7$:

$$\{(I, U), (U, I), ((0.5, 0.5), (0.833, 0.167))\}$$

$$\mathbb{E}U((I, U)) = 0$$

$$\mathbb{E}U((U, I)) = 5$$

$$\mathbb{E}U((0.5, 0.5), (0.833, 0.167)) = 0$$

b

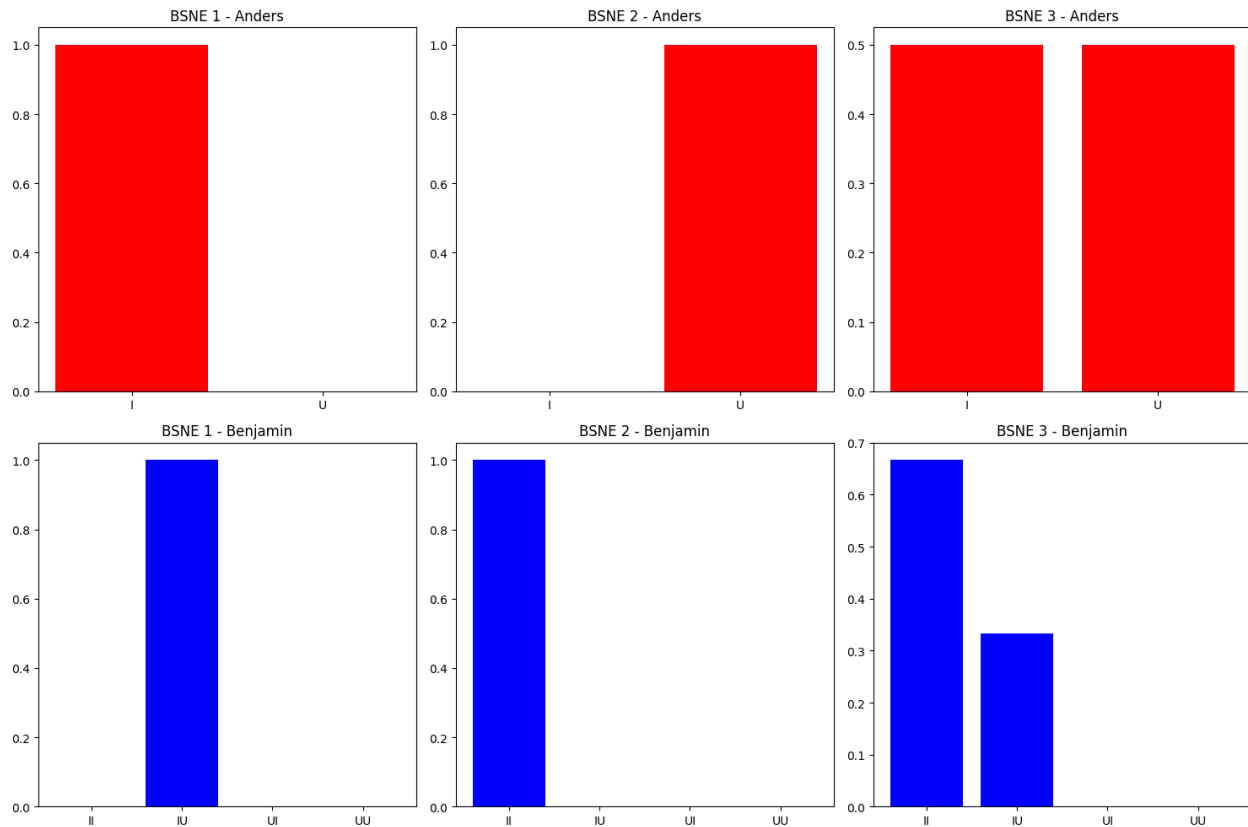
Da Anders ikke kender hvad type Benjamin er, men ved at han med 50% ssh. er $x_B = 3$ eller $x_B = 7$, kan spillet løses for en BSNE ved at opstille spillet på bred matrix form (se tabel 2) og derefter løse den med `support_enumeration` fra `nashpy`, som fandt BSNE til:

$$\{(I, IU), (U, (II)), ((0.5, 0.5), (0.667, 0.333, 0, 0))\}$$

Visuelt er BSNE også vist i figur 1.

	II	IU	UI	UU
I	(-1.0, -1.0)	(2.0, 0.5)	(2.0, -1.5)	(5.0, 0.0)
U	(0.0, 5.0)	(0.0, 3.5)	(0.0, 1.5)	(0.0, 0.0)

Tabel 2: Bred matrix form



Figur 1: BSNE visuelt.

Forventet nytte for Anders for BSNE:

$$EU((I, IU)) = 2$$

$$EU((U, (II))) = 0$$

$$EU((0.5, 0.5), (0.667, 0.333, 0, 0)) = 0$$

For at bestemme om information om Benjamins type er en fordel, skal de forventede nyttier ved de forskellige ligevægte sammenlignes. Her ses det, at kun en ligevægt når Anders kender typen gav et payoff på 5, og resten 0. Hvorimod når han ikke kender typen, så får han maks 2, og 0 på resten. Så Det kommer an på hvilken ligevægt de ender i, dog kan han ende på et bedre payoff ved at kende typen først på 5, men det kan være svært at forudsige hvilken ligevægt de ender i.

2 Signaleringspil

2.1

For at vise, at der ikke er ligevægte i rene strategier, findes først strengt dominerede strategier, herefter undersøges både poolende og separerende PBE.

Strengt dominerede strategier:

Hvis $\theta = \theta_H$, så vil Volodomir altid vælge at Angribe, da $4 > -1$ og $1 > -1$ og derfor $A \succ U$ når $\theta = \theta_H$. U er derfor strengt domineret, og da Volodomir antages for at være rationel, vil han aldrig spille U .

Poolende PBE:

Da (U, U) inkluderer en strengt domineret strategi, vil (A, A) være den eneste poolende strategi.

Når Volodomir spiller (A, A) , gør det $q = \frac{1}{2}$, og Vladimir kan derfor regne sin forventede nytte ud for M og U :

$$\mathbb{E}(u_2(A, M)) = \frac{1}{2} \cdot 4 + \frac{1}{2} \cdot -4 = 0$$

$$\mathbb{E}(u_2(A, U)) = \frac{1}{2} \cdot -1 + \frac{1}{2} \cdot -1 = -1$$

og da $0 > -1$ er $M \succ U$ for Vladimir i en poolende strategi.

Herefter undersøges om Volodomir vil afvige for denne strategi: Hvis $\theta = \theta_L$ ser Volodomir at han altid vil få -4 , da Vladimir altid går til Modangreb, og vil derfor hellere Undvige og altså afvige, da $-1 > -4$. Altså er der ingen poolende PBE.

Separerende PBE: Den eneste separerende strategi er (U, A) , da (A, U) vil inkludere den strengt dominerede strategi nævnt tidligere.

Vladimir er også klar over det faktum, og ved derfor at Vladimir altid vil Angribe hvis han er den stærke type, og Undvige hvis han er svag. Vladimir behøver altså kun at kigge på når Volodomir har $\theta = \theta_H$. Her vil Vladimir vælge U , da $-1 > -4$.

Volodomir er dog også klar over at Vladimir vil vælge U , og kan derfor afvige når han er den svage type, da $1 > -1$ og derfor er der ingen separerende PBE, og derfor **ingen PBE i rene strategier**.

2.2

For at bestemme en semi-separerende PBE, lader vi de subjektive forventninger være

$$q \equiv \mu(\theta = \theta_H | a_1 = \text{Angrib}) \quad (1)$$

og de blandede strategier

$$\sigma_1 \equiv \Pr(A | \theta_L) \quad (2)$$

$$\sigma_2 \equiv \Pr(M | A) \quad (3)$$

hvor der så løses for σ_1 , σ_2 og q :

Indifferens for Volodomir:

Løser ligning (4) nedenfor for σ_2

$$\mathbb{E}[u_1(A | \theta_L)] = \mathbb{E}[u_1(U | \theta_L)] \quad (4)$$

$$\mathbb{E}[u_1(A | \theta_L)] = \sigma_2 \cdot -4 + (1 - \sigma_2) \cdot 1 = 1 - 5\sigma_2$$

$$\mathbb{E}[u_1(U \mid \theta_L)] = -1$$

$$1 - 5\sigma_2 = -1 \iff \sigma_2 = \frac{2}{5}$$

Dvs. Volodomir er indifferent om at Angribe eller Udeblive når han er svag og Vladimir går til Modangreb $\frac{2}{5}$ af gangene og Udebliver $\frac{3}{5}$ af gangene.

Indifferens for Vladimir:

Løser ligning (5) nedenfor for q

$$\mathbb{E}[u_2(M \mid A)] = \mathbb{E}[u_2(U \mid A)] \quad (5)$$

$$\mathbb{E}[u_2(M \mid A)] = q \cdot 4 + (1 - q) \cdot -4 = 8q - 4$$

$$\mathbb{E}[u_2(U \mid A)] = q \cdot -1 + (1 - q) \cdot -1 = -1$$

$$8q - 4 = -1 \iff q = \frac{3}{8}$$

Dvs. Vladimir er indifferent om at gå til Modangreb eller Afvige når $q = \frac{3}{8}$.

Vladimirs subjektive forventninger skal følge Bayes regel

Fra Bayes regel ved vi at

$$Pr(\theta_L|A) = \frac{\sigma_1}{\sigma_1 + 1}$$

Ydermere vides det at

$$q = Pr(\theta_L|A)$$

og at $q = \frac{3}{8}$, og derfra kan σ_1 bestemmes

$$\frac{3}{8} = \frac{\sigma_1}{\sigma_1 + 1} \iff \sigma_1 = \frac{3}{5}$$

Den semi-separerende PBE er derfor

$$\text{PBE} = \left(\left(A, \sigma_1 = \frac{3}{5} \right), \left(\sigma_2 = \frac{2}{5} \right), \left(q = \frac{3}{8} \right) \right) \quad (6)$$

Volodomirs strategi i ligevægten er, at altid at gå til Angreb hvis han er stærk, og $\frac{3}{5}$ af gangene gå til angreb når han er svag. Han går til angreb nogle gange, selvom han er svag, da han ved at Vladimir nogle gange Udebliver, i frygt for at Volodomir er stærk (Volodomir bluffer altså).

Vladimir gør ikke noget hvis Volodomir Udebliver, men hvis han går til Angreb, så går Vladimir til Modangreb $\frac{2}{5}$ af gangene, da han bliver nødt til at forhindre Volodomir i altid at gå til angreb når han er svag, selvom han nogle gange ender med at gå til Modangreb med en stærk Volodomir. Vladimirs subjektive forventning er, at når Volodomir går til Angreb, så er der $q = \frac{3}{8}$ ssh. for at han er svag.

2.3

For at bestemme de værdier af p hvor der findes en PBE i rene strategier, kan der ses tilbage på problemet i den poolende strategi i 2.1, hvor Volodomir ville afvige. Grunden til Volodomir afvigede var, at Vladimir altid gik til Modangreb, hvilket får den svage Volodomir til at kunne afvige. Vi kan derfor i stedet bestemme de p -værdier, så Vladimir i stedet altid udebliver, så Volodomir ikke vil afvige.

$$\mathbb{E}(u_2(A, M)) = p \cdot 4 + (1 - p) \cdot -4 = 8p - 4$$

$$\mathbb{E}(u_2(A, U)) = p \cdot -1 + (1 - p) \cdot -1 = -1$$

Dette vil ske når $\mathbb{E}(u_2(A, M)) < \mathbb{E}(u_2(A, U))$, hvortil der kan løses for p :

$$8p - 4 < -1 \iff p < \frac{3}{8}$$

Altså når $p < \frac{3}{8}$ vil Vladimir altid Udeblive, og Volodomir kan ikke afvige, da $1 > -1$ og der vil derfor være en poolende ren strategi.

3 Hearthstone

3.1

Først findes hvad man spiller som første ordens rationel spiller, og herefter kan der blot best responses ud fra det. Dette er implementeret i python, og er angivet i tabel 3. Her ses det tydeligt, at den går i ring mellem "Frost Death Knight", "Thaddius Warlock" og "Pure Paladin".

k	Deck
1	Frost Death Knight
2	Thaddius Warlock
3	Pure Paladin
4	Frost Death Knight
5	Thaddius Warlock
6	Pure Paladin
7	Frost Death Knight
8	Thaddius Warlock
9	Pure Paladin
10	Frost Death Knight

Tabel 3: Tabel over hvad en k 'te ordens spiller vil spille.

3.2

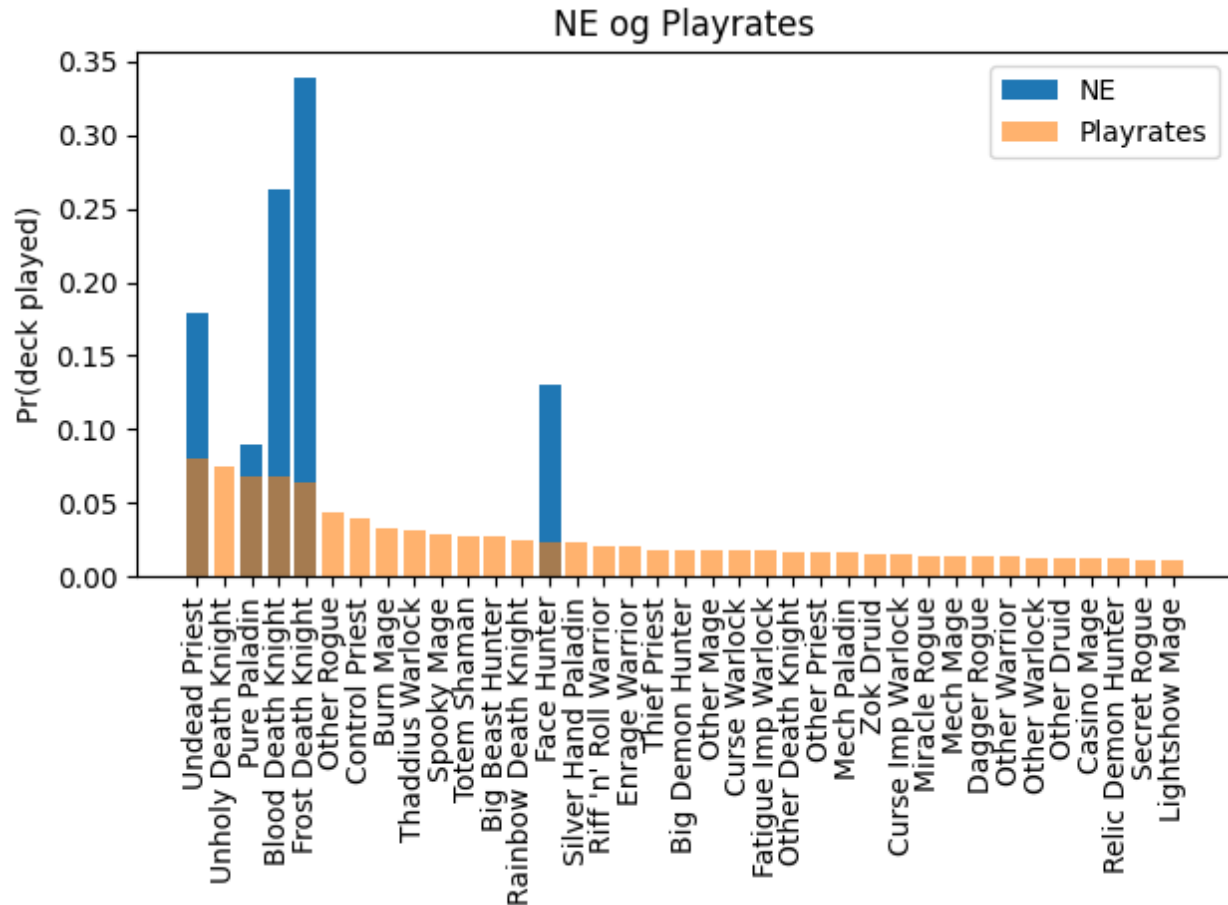
For at finde den strategi der er bedst mod en tilfældig modstander online bestemmes den gennemsnitlige winrate ud fra frekvensen af hvilke deck der bliver spillet. Dette er implementeret i python, som fandt frem til at "Pure Paladin"er bedst at spille, da det deck har den højeste forventede winrate på 62.89%.

3.3

For at finde Nash-ligevægten i spillet kan den lineære programmeringsfunktionen opgivet i kurset bruges, da det er et symmetrisk nulsumspil. Algoritmen fandt ligevægten, som kan ses i tabel 4 med de givne decks empiriske frekvenser samt i figur 2. Her ses det, at de decks der har højeste empiriske frekvenser næsten alle indgår i NE, dog på nær "Unholy Death knight"og så indgår "Face Hunter"selvom den har en langt lavere frekvens.

Deck	NE	Playrates
Undead Priest	0.179	0.080
Pure Paladin	0.089	0.068
Blood Death Knight	0.263	0.068
Frost Death Knight	0.339	0.065
Face Hunter	0.130	0.024

Tabel 4: NE og empiriske frekvenser for de deck i støtten.



Figur 2: NE og de empiriske frekvenser.

3.4

Fra sidste opgave fandtes det, at "Frost Death Knight" havde størst vægt i støtten, og vi starter derfor med det deck. Derfra benytttes IBR algoritmen (på samme måde som i 3.1), blot med kun de decks der er i støtten. Se tabel 5 for outputtet. Her ses det, at "Frost Death Knight" og "Pure Paladin" er i begge "mønstre". Det kan ses, at best response for "Pure Paladin" stadig er "Frost Death Knight" hvilket det burde, da deres best response ikke burde ændres, hvis det deck der plejede at være med stadig er i sættet.

Iteration	Deck
1	Face Hunter
2	Blood Death Knight
3	Undead Priest
4	Pure Paladin
5	Frost Death Knight
6	Face Hunter
7	Blood Death Knight
8	Undead Priest
9	Pure Paladin
10	Frost Death Knight

Tabel 5: Output fra IBR algoritmen.

3.5

Jeg har valgt at angive et kvantativt mål for hvor god en balancen er ved hvor mange deck der er i støtten i Nash ligevægten. Dvs. jo flere deck i støtten desto bedre, da der derfor kommer flere deck i spil. Dette kan gøres ved at fjerne alle deck en af gangen, og så se hvor mange deck der er i støtten, når det deck er væk. Som anden ordens betingelse (hvis flere giver samme antal deck i støtten), så vil vi gerne have den NE, som giver den mest uniforme fordeling (da vi ikke vil have over- eller underspillede decks). Dette gøres ved tage summen af de kvadrerede afstande mellem sandsynlighederne og $\frac{1}{\# \text{deck i støtten}}$ (den uniforme fordeling). Dette er implementeret i python, som angiver at vi skal fjerne "Unholy Death Knight", da det giver 5 deck i støtten og en sum af de kvadrerede afstande til den uniforme fordeling på 0.041². Kedeligt nok giver det den samme Nash-ligevægt (se tabel 4 og figur 2), og man kan derfor argumentere for, at spillet allerede var balanceret ud fra de kriterier jeg har angivet.

4 Dynamisk Priskonkurrence

	p^*	$\bar{p}$	$\hat{p}$	δ -Tærskel	π^D	π^C	$\hat{\pi}$
$\gamma_1 = 1, \gamma_2 = 0.1$	2.233	2.679	2.318	0.454	0.549	0.579	0.604
$\gamma_1 = 0.1, \gamma_2 = 0.411$	2.233	2.701	2.313	0.455	0.504	0.535	0.561

Tabel 6: Alle fundne priser og profiler i opgaven.

²Bemærk, at min algoritme blot finder det første deck, men at der godt kan være deck der er "ligeså gode" at fjerne ud fra disse kriterier.

	$q^* = q(p^*, p^*)$	$\bar{q} = q(\bar{p}, \bar{p})$	$c(q^*)$	$c(\bar{q})$
$\gamma_1 = 1, \gamma_2 = 0.1$	0.307	0.252	0.443	0.381
$\gamma_1 = 0.1, \gamma_2 = 0.411$	0.307	0.249	0.589	0.552

Tabel 7: Efterspørgsel og omkostninger ved Nash-prisen og kartel-prisen

4.1

For at bestemme den symmetriske Nash-ligevægt (p^*, p^*) benytter jeg IBR algoritmen, indtil den rammer en absolut tolerance på 0.000001 eller ved 100 iterationer. For at benytte algoritmen skal der opstilles best response funktioner for de to firmaer. Best response funktionerne er symmetriske, og tager blot det andet firmas pris, og maksimerer herefter egen profit. IBR fandt efter 6 iterationer frem til

$$(p^*, p^*) = (2.233, 2.233)$$

4.2

For at bestemme det statiske kartels pris $(\bar{p}, \bar{p})$, ved vi at et kartel løser for

$$\max_{p \geq 0} \pi_1(p, p) + \pi_2(p, p) \quad (7)$$

Hvilket er implementeret i python og løst til

$$(\bar{p}, \bar{p}) = (2.679, 2.679)$$

4.3

For at finde den tærskel værdi for $\bar{\delta}$ der gør at Trigger-strategien er en underspil-perfekt Nash-ligevægt, da kan formen³

$$\delta \geq \frac{\hat{\pi} - \pi^C}{\hat{\pi} - \pi^D} \quad (8)$$

benyttes, hvor $\hat{\pi}$ er profitten ved at afvige for strategien, hvor man så straffes med Nash prisen resten af tiden. π^C er profitten for kartellet og π^D er profitten ved Nash. Formlen implementeres i python, som får

$$\delta \geq 0.454 \quad (9)$$

4.4

For at bestemme den γ_2 værdi der gør, at Nash-ligevægten kommer tættest på $(p^*, p^*) = (2.233, 2.233)$, har jeg lavet en funktion der giver den absolutte afstanden fra en Nash ligevægt

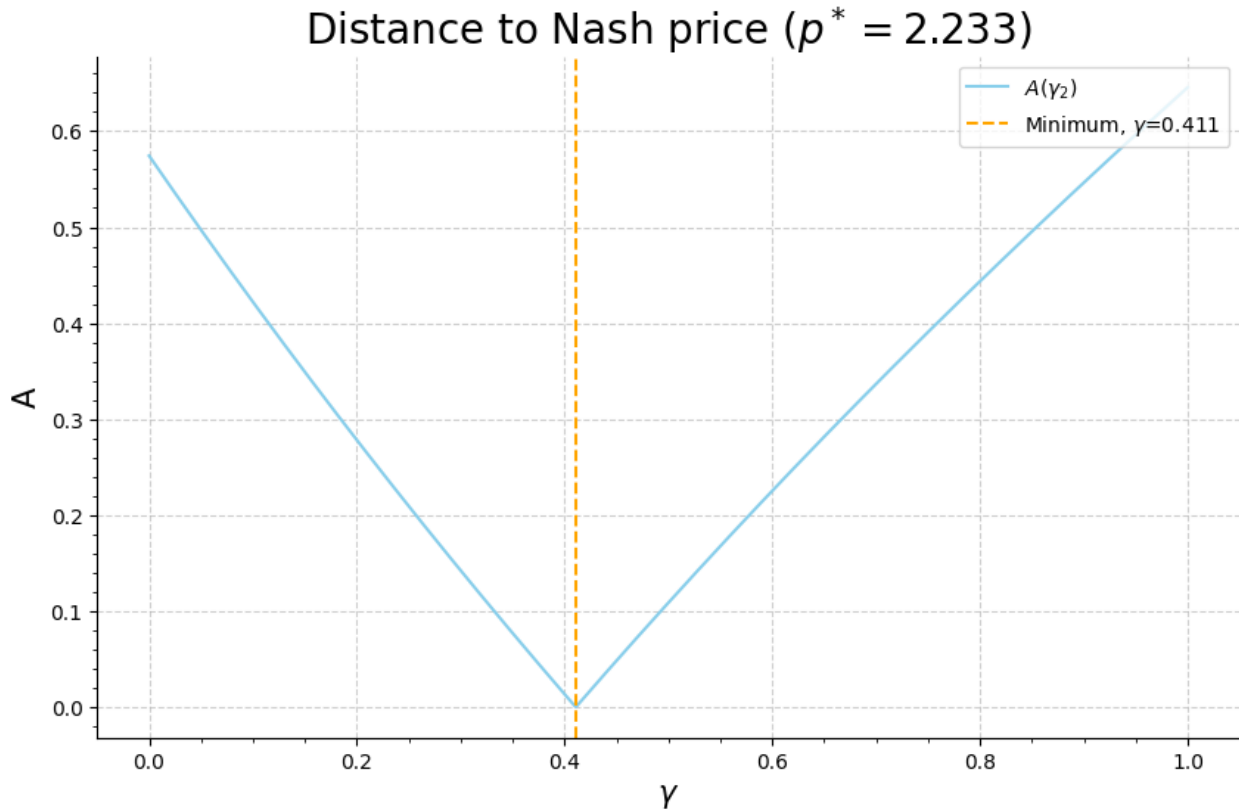
³Ligning 10.5 i bogen, kapitel 11.4.

fundet for en γ_2 ($\hat{p}_{\gamma_2}^*$) værdi til den gamle pris (p^*). Denne funktion kan herefter minimeres. Dette kan formelt opskrives som

$$\hat{\gamma}_2 = \arg \min_{\gamma_2} A(\gamma_2) \quad (10)$$

$$A(\gamma) = |p^* - \hat{p}_{\gamma_2}^*| \quad (11)$$

Dette er implementeret i python, hvor $\hat{\gamma}_2 = 0.411$, hvilket også kan ses visuelt i figur 3 nedenfor.



Figur 3: Funktionen A plottet.

Har herefter igen løst for kartelprisen $\bar{p}$ og tærskelværdien δ i python og fundet

$$(\bar{p}, \bar{p}) = (2.701, 2.701)$$

$$\delta \geq 0.455$$

4.5

Det kan ses i tabellerne givet i starten (6 og 7), at profitterne er faldet, hvilket hænger sammen med at omkostningerne er steget. Dette giver også god mening, da vi har forøget γ_2 ,

som har en større indflydelse på omkostningerne end γ_1 :

$$\frac{\partial c}{\partial \gamma_1} = q \quad (12)$$

$$\frac{\partial c}{\partial \gamma_2} = \exp(q) \quad (13)$$

og da $\exp(q) > q$ ændres omkostningerne mere ved at øge γ_2 , og derfor ser vi faldende profiler, men nogenlunde samme efterspørgsel og tærskel, da efterspørgslen ikke afhænger af γ_1 eller γ_2 .

5 Auktionsformater

5.1

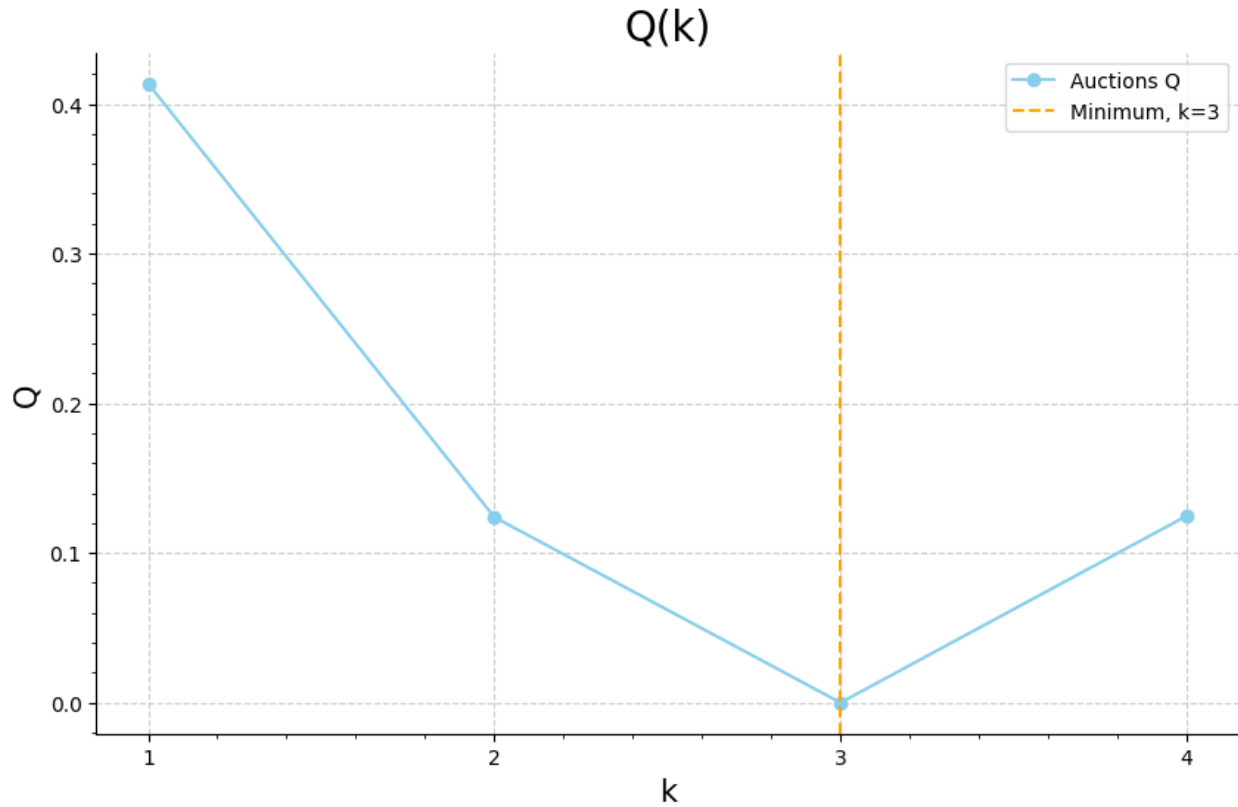
For at estimere k laver jeg først en simuleret fordeling for $k \in \{1, 2, 3, 4\}$ af SPSB auktioner. Dvs. jeg bruger beta fordelingen til at finde deres valueringer, sortere dem herefter og tager den anden højeste. Det giver for hvert k en fordeling af vinder-budende. Herefter kan jeg benytte modulet `Statsmodel` og dens funktion `ECDF` til at lave en estimeret CDF fordeling, $\hat{F}^{\text{sim}}$. Dette bruger jeg til at lave en Q funktion, som igen laver en estimeret CDF fordeling af data, $\hat{F}^{\text{data}}$. Herefter finder jeg den gennemsnitlige kvadrerede afstand mellem 100 punkter mellem minimum budet og maksimum budet på de to estimerede CDF'er⁴. Herefter kan Q minimeres ud fra $k \in \{1, 2, 3, 4\}$. Minimeringsproblemet kan formaliseres til:

$$\hat{k} = \arg \min_{k \in \{1, 2, 3, 4\}} Q(k) \quad (14)$$

$$Q(k) = G^{-1} \sum_{g=1}^G \left[\hat{F}^{\text{data}}(x_g) - \hat{F}^{\text{sim}}(x_g) \right]^2 \quad (15)$$

Her er x et lineært grid af 100 punkter mellem $p_{\min}$ og $p_{\max}$, og derfra $G = 100$. Hertil fandtes det, at Q er minimum ved $\hat{k} = 3$, hvilket også kan ses grafisk på figur 4.

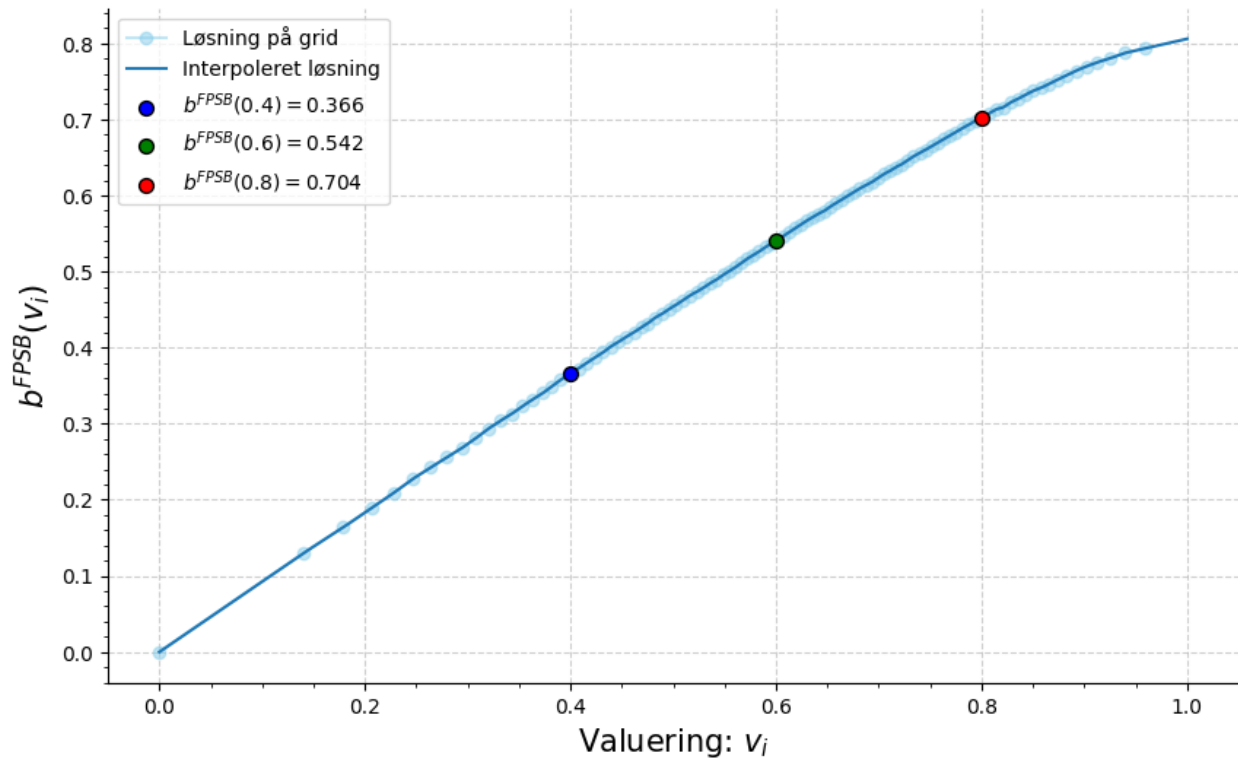
⁴I ord giver Q funktionen et tal på hvor meget fordelingerne ligner hinanden. Jo lavere jo bedre.



Figur 4: Q plottet.

5.2

For at bestemme $b^*(v)$ numerisk, kan der benyttes provenuækvivalens, da valueringer er efter samme fordeling, samt at en byder med 0 i valuering betaler 0 og derfor har FPSB og SPSB samme forventet betaling. Herfra vil $b^*(v)$ blive bestemt for SPSB, da den er nemmere at regne med. Herfra kan funktionen `Ev_largest` opgivet i kurset benyttes hvorefter der kan interpoleres mellem punkterne for at bestemme $b^*(v)$ numerisk. Se figur 5 for at funktionen grafisk samt funktionsværdierne for $v \in \{0.4, 0.6, 0.8\}$.



Figur 5: b^* funktionen samt funktionsværdierne for $v \in \{0.4, 0.6, 0.8\}$.

5.3

FPSB

Fordel

FPSB har oftest en lavere varians over vinder budende, og det kan derfor være til fordel at have denne auktionsformat hvis auktionarius er risikoavers (hvilket man oftest vil antage den her).

Ulempe

"The winners curse" er et problem (i common value auktioner), når en byder fx har en alt for ekstrem valuering ift. hvad et objekt i virkeligheden er værd, og vinderen ender derfor oftest med at tabe på auktionen (fx ved oliefeltsauktionerne i USA). Er dog afhjulpel hvis byderne ikke byder naivt.

Eksempel

I offentlige kontrakter, hvor det offentlige gerne vil have en så billig kontrakt som muligt. Det er dog ikke altid godt, da byderen oftest har en urealistisk valuering (og går derfor oftest over budget, se Niels Bohr bygningen).

SPSB

Fordel

Byderne kan byde deres valuering, og være sikre på det er bedst, da det er den dominante strategi.

Ulempe

Ofte en større varians over vinder budende, og er derfor ikke godt for en risikoavers auktionarius.

Eksempel

Hvis en virksomhed gerne vil vide de hvad deres produkt er værd, så kunne det være smart at lave en SPSB auktion, da byderne der vil byde deres reelle valueringer. Dette vil kunne give dyrebar information til virksomheden, som så herefter ville kunne bruge det til at prissætte deres produkt.

5.4

For at vise, at det stadig er dominant at byde sin valuering på trods af en reservationspris, så kan vi angive det i 3 forskellige tilfælde:

1. $b_{(1)} > \bar{p}$ og $b_{(2)} < \bar{p}$: Her betaler vinderen $p = \bar{p}$, og får derfor $b_{(1)} - \bar{p} > 0$ i nytte.
2. $b_{(1)} > \bar{p}$ og $b_{(2)} > \bar{p}$: Her betaler vinderen $p = b_{(1)}$, og får derfor $b_{(1)} - b_{(2)} \geq 0$ i nytte.
3. $b_{(1)} \leq \bar{p}$: Her vinder byderen ikke, men får 0 i nytte.

Det ses, at det ikke kan betale sig at afvige fra at byde sin valuering. Det kan ikke betale sig at "bid shade", da byderen derfra kan risikere ikke at vinde, selvom byderen kunne få positiv nytte ved blot at byde sin valuering. Det kan ses, at det hellere ikke kan betale sig at byde over sin valuering, da man derfor kan risikere at skulle betale over sin valuering, og derfor få en negativ nytte.

5.5

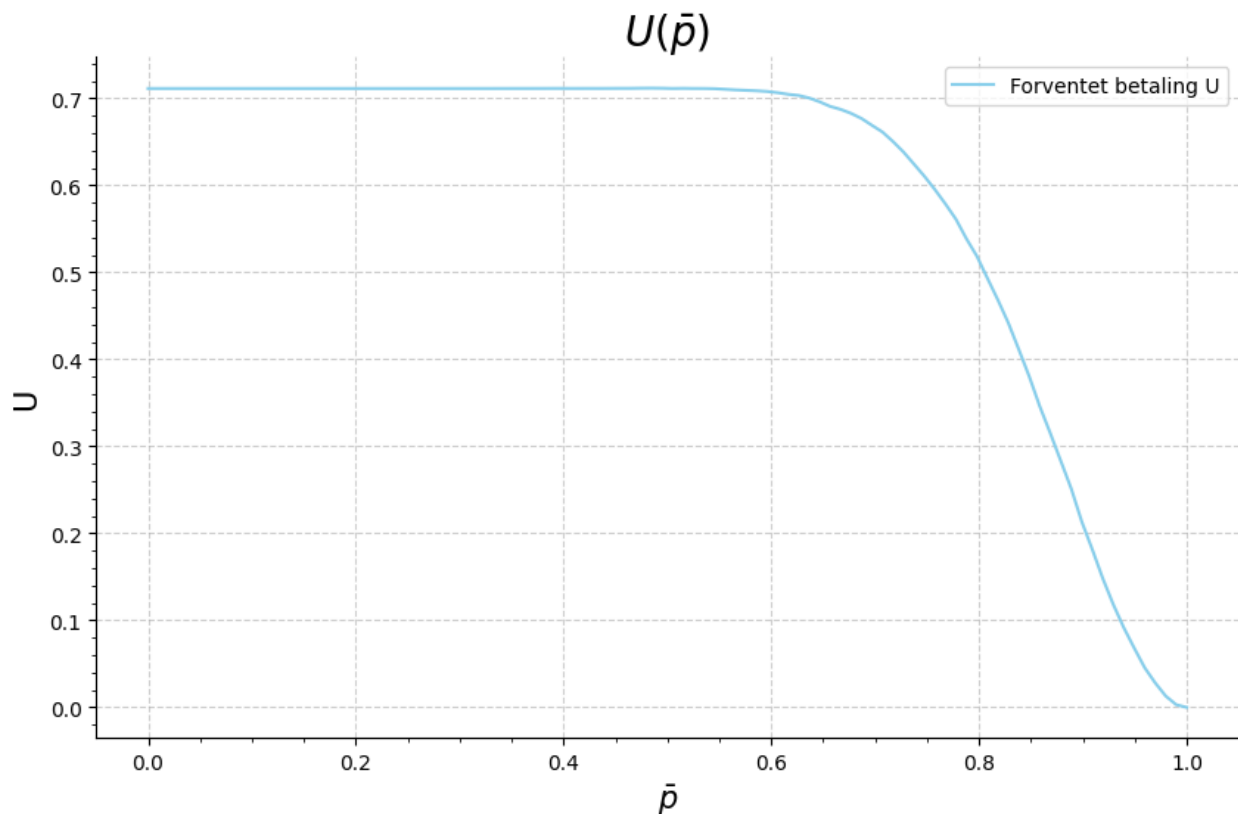
For at bestemme den reservationspris ($\bar{p}$), som maksimerer auktionarius' forventede bud har jeg lavet $U(\bar{p}) = \mathbb{E}[P(b_1, \dots, b_N, \bar{p})]$ funktionen ud fra simulationer. Funktionen tager en reservationspris som input, laver 10.000 simulerede auktioner efter samme parametre tidligere i opgaven. For hver simulerede auktion benyttes da P funktionen, som er angivet nedenfor i

(16).

$$P(b_1, \dots, b_N, \bar{p}) = \begin{cases} \max(b_{(2)}, \bar{p}) & \text{hvis } b_{(1)} > \bar{p} \\ 0 & \text{hvis } b_{(1)} \leq \bar{p} \end{cases} \quad (16)$$

Herefter har jeg plottet $U(\bar{p})$ for 1000 værdier af $\bar{p}$ mellem 0 og 1 i figur 6. Her ses det, at så længe $\bar{p} < 0.6$ får auktionarius det samme forventede bud på 0.711. Den kan ikke maksimeres numerisk til et enkelt toppunkt, da forskellene blot er simulationsstøj.

Valget af $\bar{p}$ indebærer de tradeoffs, at jo højere reservationsprisen er, desto større er sandsynligheden for at ingen bud er over prisen. Det gør derimod også, at hvis der er langt fra $b_{(1)}$ og $b_{(2)}$, og at $b_{(2)} < \bar{p}$, så får auktionarius et højere vinderbud.



Figur 6: Auktionarius målfunktion $U(\bar{p})$ plottet.