Tasapainotilan laskeminen [1]

Timo Lilja
timo.lilja@hut.fi
47989s

7. maaliskuuta 2002
21. maaliskuuta 2002

[1] T–79.194 Tietojenkäsittelyteorian seminaari

Sisältö

1 Johdanto

Tämä tutkielma käsittelee osia McKelveyn ja McLennanin paperista *Computation of Equilibria in Finite Games* [1]. Tässä paperissa esitellään normaalimuotoisen pelin formaalia määrittely todennäköisyysjakaumien avulla ja Nashin tasapainotila määrittely. Lewken ja Hawsonin algoritmi esitellään kohtuullisen kattavasti ja osittain sivutaan muita normaalimuotoisten pelien laskentamenetelmiä. XXX Ekstensiivisisiä (*extensive form games*) pelejä käsitellään hyvin yleisesti ilman tarkkaa formalismia.

2 Notaatiosta

Määritellään Nashin tasapainotila n:n pelaajan normaalimuodon (*normal form*) pelille: On olemassa **pelaajien** joukko $N = \{1, \ldots, n\}$. Jokaisella pelaajalla on **strategiajoukko** $S_i = \{s_{i1}, \ldots, s_{im_i}\}$. Jokaiselle pelaajalle $i \in N$ määritellään **hyötyfunktio** $u_i : S \mapsto \mathbb{R}$ missä $S = \prod_{i \in N} S_i = S_1 \times \ldots \times S_n$.

Olkoon $\mathcal{P}_i = \{f | f : S_i \mapsto \mathbb{R}\}$ eli $\mathcal{P}_i$ on funktio strategiajoukosta S_i reaaliluvuille. Joukon $\mathcal{P}_i$ jäsenille $p_i \in \mathcal{P}_i$ käytetään merkintää $p_{ij} = p_i(s_{ij})$. Muodostetaan yksittäisistä $\mathcal{P}_i$ joukoista yhdistelmä seuraavasti: $\mathcal{P} = \prod_{i \in N} \mathcal{P}_i$ ja $m = \sum_{i \in N} m_i$.

Joukko $\mathcal{P}$ koostuu siis n-jonoista, jotka puolestaan koostuvat reaaliarvoisista funktioista $\mathcal{P}_i$. S.o. $p \in \mathcal{P}$:lle pätee $p = (p_1, \ldots, p_n)$, missä $p_i = (p_{i1}, \ldots, p_{im_i})$

Jos $p \in \mathcal{P}$ ja $p'_i \in \mathcal{P}_i$, merkitään alkiota $q \in \mathcal{P}$ (p', p_{-i}):lla, kun pätee $q_i = p'_i$ ja $q_j = p_j$, jos $j \neq i$.

Hyötyfunktion määrittelyjoukko laajennetaan $\mathbb{R}^m$:ään seuraavasti:

$$u_i(p) = \sum_{s \in S} p(s) u_i(s), \tag{2.1}$$

missä

$$p(s) = \prod_{i \in N} p_i(s_i) \tag{2.2}$$

Tässä $u_i(p)$ on *hyödyn odotusarvo* eli se hyöty, jonka pelaaja i tulee pitkällä aikavälillä saamaan itselleen. Todennäköisyys $p(s)$ on kuvaa sitä todennäköisyyttä, jolla kaikki pelaajat tekevät s:n mukaiset valinnat.

Olkoon $\Delta_i = \{p_i \in \mathcal{P}_i | \sum_j p_{ij} = 1, p_i \geq 0\}$ joukolle S_i todennäköisyysmittojen virittävä joukko joukolle $\mathcal{P}_i$.

Laajennetaan tämä todennäköisyysmittojen joukko kattamaan kaikkien pelaajien valinnat eli sisältämään jouko S: $\Delta = \prod_{i \in N} \Delta_i \subseteq \mathbb{R}^m$. Merkintä s_{ij} tarkoittaa alkiota $p_i \in \Delta_i$ s.e. $p_{ij} = p_i(s_{ij}) = 1$. Siis merkintä (s_{ij}, p_{-i}) tarkoittaa kokonaisstrategiaa, jossa pelaaja i valitsee strategian s_{ij} ja muut pelaajat valitsevat omat komponentinsa p:stä.

Määritelmä 1 *Piste $p \in \mathcal{P}$ on **Nashin tasapainotila**, jos $p^* \in \Delta$ ja kaikilla $i \in N$, $p_i \in \Delta_i$ pätee $u_i(p_i, p^*_{-i}) \leq u_i(p^*)$*

2.1 Esimerkki ja tulkintaa

Seuraavaksi mietitään, miten vangin ongelman (*The Prisoner's Dilemma*, McKelvey [1]) voisi mallintaa edellä esitetyllä formalismilla.

Ongelmassa on kyse siitä, että kahta vankia epäillään rikoksesta. Jos molemmat tunnustavat, he saavat kolme vuotta vankeutta. Jos toinen tunnustaa ja toinen ei, tunnustanut osapuoli vapautetaan ja ja syyllisyytensä kieltänyt tuomitaan vankeuteen neljäksi vuodeksi. Mikäli kumpikaan osapuolista ei tunnusta, tuomitaan molemmat yhden vuoden tuomioon. Koska Nashin tasapainotila on määritelty hyödyn maksimointina, ei tässä pelissä voida suoraan käyttää vankeusrangaistusten pituuksia hyötyinä. Sen sijaan tulee ajatella, että hyöty on *saavutettavissa olevan vapauden* määrä. Näin päädytään alla olevaan kaavion (Kuva 1).

	älä tunnusta	tunnusta
älä tunnusta	3,3	0,4
tunnusta	4,0	1,1

Kuva 1: Vangin ongelma

Ensimmäinen alkio kertoo pystyrivillä olevan pelaajan 1 hyödyn, toinen alkio kertoo vastaavasti vaakarivillä olevan pelaajan 2 hyödyn. Esimerkiksi tilanteessa, jossa pelaaja 1 ei tunnusta ja pelaaja 2 tunnustaa, hyöty luetaan oikeasta yläsarakkeesta. Hyöty tässä tapauksessa on siis ensimmäiselle pelaajalle 0 vuotta ja toisella pelaajalle 4 vuotta, mikä tarkoittaa sitä, että ensimmäinen saa neljä vuotta vankeutta ja toinen vapautetaan.

Pelaajien lukumäärä on kaksi: $N = \{1, 2\}$. Molempien strategiajoukko koostuu toiminnoista: älä tunnusta (D) ja tunnusta (C). Strategiajoukot ovat:

$$S_1 = S_2 = \{D, C\}$$

Molempien vankien hyötyfunktiot ovat muotoa $u_i : S \mapsto \mathbb{R}$, missä $S = S_1 \times S_2 = \{(D, D),\ (D, C),\ (C, D),\ (C, C)\}$ ja $i \in \{1, 2\}$. Kuvan (1) perusteella hyötyfunktion arvoiksi saadaan:

$$\begin{array}{ll} u_1(D,D) = 3 & u_2(D,D) = 3 \\ u_1(D,C) = 0 & u_2(D,C) = 4 \\ u_1(C,D) = 4 & u_2(C,D) = 0 \\ u_1(C,C) = 1 & u_2(C,C) = 1 \end{array}$$

Nyt määritellään reaaliarvoiset todennäköisyysfunktiot $p_{11} = p_1(s_{11})$, $p_{12} = p_1(s_{12})$, $p_{21} = p_2(s_{21})$, $p_{22} = p_2(s_{22})$.

Nashin tasapainotilan laskeminen tarkoittaa edellä mainittujen p-funktioiden arvojen laskemista. Funktio $p_i(s_{ij})$ kertoo todennäköisyyden, jolla i:s pelaaja valitsee strategian s_{ij}.

Nashin tasapainotilan laskeminen kiinnittää todennäköisyysfunktioille p_{ij} optimaaliset arvot. Peliteoreettisesti järkevä agentti valitsee strategiansa noudattaen funktioiden p_i jakaumaa

Laajennettu hyötyfunktiot u_i voidaan nyt laskea seuraavasti kaavoista (2.1) ja (2.2) seuraavasti esimerkiksi pelaajalle $i = 1$:

$$\begin{aligned} u_1(p) &= \sum_{s\in S}\left[\prod_{i\in N} p_i(s_i)\right] u_1(s) \\ &= \sum_{s\in S} p_1(s_1)\cdot p_2(s_2)\cdot u_1(s) \\ &= p_1(D)\cdot p_2(D)\cdot u_1(D,D) + \ldots + p_1(C)\cdot p_2(C)\ u_1(C,C) \end{aligned}$$

Laajennettuun hyötyfunktio on alkuperäisen hyötyfunktion odotusarvo kun pelaajat toisistaan riippumattomasti noudattavat omia strategioitaan (todennäköisyysjakaumiaan).

Vangin ongelmassa on yksi tasapainopiste: kumpikin vanki tunnustaa. Tämä voidaan mallintaa seuraavanlaisena todennäköisyysjakaumana:

$$\begin{array}{ll} p_1(D) = 0 & p_2(D) = 0 \\ p_1(C) = 1 & p_2(C) = 1 \end{array}$$

Tällöin pätee $p^* \in \mathcal{P}$:

$$p^* = \begin{pmatrix} p_1(D) & p_1(C) \\ p_2(D) & p_2(C) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$

Jotta p^* olisi Nashin tasapainopiste, täytyy $p^* \in \Delta$, mikä pitää paikkansa, koska $i \in \{1, 2\}$:

$$p_i(D) + p_i(C) = 1$$

Tämän lisäksi tulee olla $u_i(p_i, p^*_{-i}) \leq u_i(p^*)$. Tämä on intuitiivisesti selvää vangin ongelmassa: kumpi tahansa vanki voi riippumatta siitä, mitä toinen tekee, parantaa asemaansa myöntämällä syyllisyytensä. Esimerkiksi:

$$u_1(D, D) = 3 \leq u_1(C, D) = 4$$

ja

$$u_1(D, C) = 0 \leq u_1(C, C) = 1$$

Vastaava pätee vangille kaksi.

3 Tasapainotilan laskeminen

Tasapainotilojen laskeminen jakautuu useaan osaongelmaan. On päätettävä, riittääkö yksi edustava tasapainotila (*sample equilibrum*) vai halutaanko löytää kaikki tasapainotilat.

Pelaajien lukumäärällä on merkityksensä: Kahden pelaajan tapauksissa tasapainotila-algoritmit pysyvät ratkaisujoukot pysyvät *siisteinä*[1], jos syöte on rationaalinen. Useamman kuin kahden pelaajan tapauksissa näin ei välttämättä ole.

[1] Ratkaisujoukko on unioni konvekseista monitahokkaista (*polyhedron*), jotka siis voidaan määrittää rationaalisten ääripisteittensä avulla.

Tasapainotilan *tyypillä* voi myös olla väliä. Kaikki tasapainotilat eivät ole pelaajan kannalta yhtä hyviä. Yksi tasapainotila saattaa olla ylitse muiden jollain mittarilla mitattuna.

Globaalisti suppenevat (*globally convergent*) menetelmät löytävät ratkaisun lähtien mistä tahansa pisteestä. Ei-globaalisti suppenevat menetelmät vastaavasti eivät löydä ratkaisua kuin lähtemällä tietyistä pisteistä. Tällaiset menetelmät ovat yleensä huomattavasti nopeampia kuin globaalisti suppenevat, minkä takia niiden käyttö voi joissain tilanteissa olla mielenkiintoista.

3.1 Lemken ja Howsonin algoritmi kahdelle pelaajalle

Lemken ja Howsonin algoritmi on ns. **polun seuranta -algoritmi** (*path following algorithm*) ja se kehitettiin aluksi kahden pelaajan pelejä varten. Myöhemmin sitä lavennettiin ratkaisemaan yleisempiä lineaarisia komplementtiongelmia

Käsitellään siis kahden pelaajan tapausta. Olkoon annettuna pelaajille $i = 1, 2$ strategiajoukot $S_i = \{s_{1i}, \ldots, s_{im_i}\}$. Olkoon U_i pelaajan i hyötymatriisi s.e. vaakarivit ovat indeksoitu pelaajan i strategioiden ja pystyrivit pelaajan $\ell \neq i$ mukaan. Toisin sanoen $U_i(j,k) = u_i(s_{ij}, s_{-i,k})$.

Nashin tasapainotila on pystyvektoripari $p_i \in \mathbb{R}^{m_i}$, jolle pätee

$$U_i \cdot p_{-i} + r_i = v_i \cdot \mathbf{1} \tag{3.1}$$

missä $p_i \geq 0$, $r_i \geq 0$, $p_i \cdot 1 = 1$, ja $p_i \cdot r_i = 0$ kummalakin $i = 1, 2$. Vakio v_i on skalaari, joka edustaa pelaajan i hyötyä ja $r_i \in \mathbb{R}^{m_i}$ on pystyvektori, jonka alkioiden tulee olla 0, jos i käyttää strategiaa s_{ij} positiivisella todennäköisyydellä.

Oletetaan, että U_i:n alkiot ovat positiivisiä. Tällöin v_i on myös positiivinen. Nyt edellä oleva kaava (3.1) voidaan uudelleen formuloida, kun $p'_i = p_i / v_{-i}$ ja $r'_i = r_i / v_i$:

$$U_i \cdot p'_{-i} + r'_{-i} = \mathbf{1} \tag{3.2}$$

missä $p'_i \geq 0$, $r'_i \geq 0$, $p'_i \cdot \mathbf{1} = 1/v_{-i}$, ja $p'_i \cdot r'_i = 0$, kun $i = 1, 2$. Jos ehdosta $p'_i \cdot \mathbf{1} = 1/v_{-i}$ luovutaan, syntyy yhtälöön yksi ylimääräinen ratkaisu $p'_1 = p'_2 = 0$. Tätä kutsutaan **ylimääräiseksi ratkaisuksi** (*extraneous solution*).

Olkoon $m = \sum_{i \in N} m_i$ eli molempien pelaajien strategioiden lukumäärän summa. Määritellään yhdistetty $m \times m$-hyötymatriisi U seuraavasti:

$$U = \begin{bmatrix} 0 & U_1 \\ U_2 & 0 \end{bmatrix} \tag{3.3}$$

Määritellään myös m-kokoiset pystyvektorit x, y, ja q:

$$x = \begin{bmatrix} r'_1 \\ r'_2 \end{bmatrix}, \; y = \begin{bmatrix} p'_1 \\ p'_2 \end{bmatrix}, \; q = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix} \tag{3.4}$$

Nyt kaava (3.2) voidaan kirjoittaa lineaarisena komplementtiongelmana:

$$Uy + x = q \tag{3.5}$$

missä $x \geq 0$, $y \geq 0$ ja $x \cdot y = 0$. Koska vaaditaan $x \cdot y = 0$, ei voida soveltaa perinteistä käänteismatriisilla kertomista ratkaisujen x ja y hakemiseeen, vaan tarvitaan monimutkaisempia menetelmiä. Koska $x \geq 0$ ja $y \geq 0$ tulee vähintään puolet vektorien x ja y komponenteista olla arvoltaan 0, jotta pistetuloehto täyttyisi.

Vektoripari $x, y \in \mathbb{R}^m$ on **ratkaisu** yhtälölle (3.5), jos $Uy + x = q$. Koska ratkaisuvektorien x ja y komponenttien tulee olla ei-negatiivisia ja jokainen U:n alkio sekä q on ei-negatiivinen, on ratkaisujoukko äärellisen monen muotoa $\{\xi \in \mathbb{R}^{2m} : \nu \cdot \xi\}$ olevan puoliavaruuden leikkaus. Tästä seuraa, että ratkaisujen joukko on suljettu, konveksi ja rajoitettu monitahokas (koska kyseessä on äärellisen monta. Edellä mainitut ominaisuudet takaavat, että algoritmi pysähtyy varmasti.

Yhtälö (3.5) voidaan kirjoittaa:

$$\begin{bmatrix} U & I_m \end{bmatrix} \cdot \begin{bmatrix} y \\ x \end{bmatrix} = q \tag{3.6}$$

Merkitään $A = \begin{bmatrix} U & I_m \end{bmatrix} \cdot$ ja $z = \begin{bmatrix} y \\ x \end{bmatrix}$. Nyt A on $m \times 2m$-matriisi ja z on $2m$-alkioinen pystyvektori. Matriisille A määritellään joukko indeksejä $\beta = \{b_1, \ldots, b_m\}$, missä $1 \leq b_i \leq 2m$. Joukossa β voi esiintyä sama indeksi useamman kerran. Määritellään myös $m \times m$-matriisi B^β, jonka i:s pystyrivi on b_i:s A-matriisin pystyrivi.

Yhtälöä (3.6) ei voida ratkaista vain laskemalla A:n käänteismatriisi ja kertomalla yhtälö sillä, koska näin saatu ratkaisu ei välttämättä täytä pistetuloehtoa $x \cdot y = 0$. Ratkaisuja kyseiseen ongelmaan on ilmeisesti äärellinen määrä.

Joukko β on **kanta** (*basis*), jos B^β ei ole singulaarinen eli matriisi, jolle ei ole käänteismatriisia. **Perusratkaisu** (*basic solution*) on sellainen yhtälön (3.6) ratkaisu z, jolle pätee, että $z_i = 0$ kaikilla $i \notin \beta$.

Jokainen kanta β määrittää yksikäsitteisen ratkaisun yhtälöön (3.6):

$$z = (B^\beta)^{-1} q \tag{3.7}$$

missä joukon β määrittämien indeksien osoittamat matriisin A pystyrivit muodostavat matriisin B^β. Yhtälön (3.7) ratkaisu on myös alkuperäisen lineaarisen komplementtiongelman ratkaisu (3.6), koska matriisi B^β on muodostettu niin että siihen on poimittu matriisista A β-joukossa esiintyvät indeksit. Jos z on yhtälön (3.7) ratkaisu, on se myös yhtälön (3.6) ratkaisu, jos muiden indeksien osoittamat paikat asetetaan nolliksi.

Kanta on **käyttökelpoinen** (*feasible*), jos sitä vastaava ratkaisun z komponentit ovat positiivisia.

Kanta β on **komplementaarinen** (*complementary*), jos jokaisesta indeksistä $1 \leq j \leq m$ joko j tai $m + j$ on β:ssa, mutta ei molemmat. Kanta on i-**melkein komplementaarinen** (*i almost complementary*), jos jokaisella j: $1 \leq j \leq m$, $j \neq i$, joko $j \neq \beta$ tai $m + j \neq \beta$, mutta ei molemmat. Komplementaariset kannat saadaan suoraan yhtälön (3.6) lineaarisen komplementtiongelman ehdot toteuttavia ratkaisuja, mutta nämä eivät välttämättä ole kelvollisia.

Algoritmin toimii siten, että lähdetään *käyttökelpoisesta komplementaarisesta* ratkaisusta ja siirrytään *i-käyttökelpoisten melkein komplementaaristen ratkaisujen* kautta toiseen *käyttökelpoiseen komplementaariseen ratkaisuun.*

Siis matriisista A valitaan m-pystyrivin indeksia kannaksi β. Tämän avulla muodostetaan matriisi B^β, johon poimitaan matriisista A joukon β-indeksien määräämät pystyrivit ja vieläpä niin, että matriisista B^β ei saa muodostoa singulaarimatriisia.

Laskenta etenee seuraavasti: Olkoon kanta β ja $q^\beta = (B^\beta)^{-1}q$ ja $A^\beta = (B^\beta)^{-1}A$. Määritellään kannalle β matriisi T^β:

$$T^\beta = \begin{bmatrix} -q^\beta & A^\beta \end{bmatrix} = (B^\beta)^{-1} \begin{bmatrix} -q & A \end{bmatrix} \tag{3.8}$$

Tämä on algoritmin "ensimmäinen kierros". Annetut A ja q siis sijoitetaan matriisiin ja kerrotaan matriisin B^β käänteismatriisilla.

Nyt algoritmia jatketaan siten, että määritetään uusi kanta $\gamma = \{g_1, \ldots, g_n\}$. Erityisesti sellaiset kannat, joissa β ja γ eroavat vain yhdellä elementillä ovat mielenkiintoisia, koska tällöin komplementaarisuus tai melkein komplementaarisuus ominaisuus säilyy (s.o. ratkaisu on edelleen ratkaisu alkuperäiseen LCP-ongelmaan.). Nyt voidaan laskea kannalle γ matriisi T^γ:

$$T^\gamma = \begin{bmatrix} -q^\gamma & A^\gamma \end{bmatrix} = (P^{\gamma\beta})^{-1} \begin{bmatrix} -q^\beta & A^\beta \end{bmatrix} \tag{3.9}$$

Tässä siis q^β ja A^β on laskettu kaavalla (3.8). $P^{\gamma\beta} = (B^\gamma)^{-1}B^\beta$ on ns. **pivot-matriisi**.

Edellä mainittujen kaavojen (3.8) ja (3.9) avulla algoritmia voidaan suorittaa, kunhan löydetään uusi kanta. Kannalta edellytetään, että se on:

- käyttökelpoinen
- i-melkein komplementaarinen
- eri kuin se kanta, josta tähän kantaan saavuttiin

Seuraavaa kantaa valittaessa tulee kiinnittää huomiota siihen, että onko lähtökanta komplementaarinen vai i-melkein komplementaarinen. Seuraava kanta valitaan siten, että kannat pysyvät komplementaarisina tai i-melkein komplementaarisina (jotka myös ovat komplementaarisia).

Kullakin kierroksella kannat valitaan pivot-matriisia P ja sekä matriisia T. Siis, ainoastaan ensimmäisellä kierroksella kannat valitaan yhtälön (3.6) matriisista A, seuraavilla kierroksilla aina lasketusta matriisista T.

Reaalinumeroista koostuva matriisi on **lex-negatiivinen**, jos jokaisen rivin ensimmäinen ei-nolla arvoinen alkio on negatiivinen (positiivinen vast.). Kanta β on **lex-käyttökelpoinen** (*lex feasible*), jos T^β on lex-negatiivinen. Eli uuden kannan γ hakeminen edellyttää, että lähtökannasta β pudotetaan joku alkio pois. McKelvey esittelee paperinsa [1] luvussa 3.1.1 tarkemman kuvauksen ja matemaattisen pohjan sille, että uusi kanta on aina olemassa.

Siis voidaan sanoa, että tietyin edellytyksin voidaan löytää kanta $\gamma = (\beta - \{r\}) \cup \{s\}$, missä γ on lex-käyttökelpoinen.

Olkoon $\mathcal{B}^*$ kaikkien komplementaaristen lex-käyttökelpoisten kantojen joukko ja $\mathcal{B}_i^*$ kaikkien i-melkein komplementaaristen kantojen joukko. Näiden joukkojen alkiolle voidaan määritellä järjestys ja alla olevaa algoritmia voidaan käyttää hakemaan Nashin tasapainopisteet. Algoritmi lähtee komplementaarisesta lex-käyttökelpoisesta perusratkaisusta, esimerkiksi *ylimääräisestä ratkaisusta.*

(a) Valitaan $\beta_0 \in \mathcal{B}^*$ ja $1 \leq i \leq m$. Etsitään uusi kanta β_1 lisäämällä joko i tai $m+i$ kantaan β_0. Asetetaan $k=1$.

(b) Jos $\beta_k \in \mathcal{B}^*$ pysähdytään, muuten siirrytään kohtaan 3.

(c) Annetusta β_k:sta, josta j (tai $m+j$) on pudotettu pois. Lisätään $m+j$ (j) ja etsitään uusi kanta β_{k+1}. Asetetaan $k=k+1$. Palataan kohtaan 2.

Periaattessa mikä tahansa ratkaisu käy algoritmin lähtöpisteeksi. Ongelma on siinä, kuinka ratkaisu löydetään. Lemken ja Hawsonin algoritmi ei takaa, että kaikki tasapainotilat tulevat löydetyiksi.

Algoritmin läpikäymät pisteet muodostavat monitahokkaan, jonka reunapisteet ovat haettuja Nashin tasapainotiloja eli yhtälön (3.5) ratkaisuja. Koska ratkaisujoukko on rajoitettu ja edellä oleva algoritmi huolehtii siitä, ettei samaan pisteeseen palata useampaan kertaan, ei tasapainotilan haku voi jäädä ikisilmukkaan.

Algoritmi käy kantoja läpi vaihdellen komplementaarisen ja melkein komplementaaristen ratkaisujen välillä ja pitää kirjaa käymistään kannoista. Kantoja on äärellinen määrä. Näistä seuraa se, että algoritmi väistämättä pysähtyy.

3.2 Muita tasapainotilan laskentamenetelmiä

3.2.1 Kolmionti

Kolmionti (*fixed triangualation*) perustuu Nashin tasapainon esittämiseen funktion kiintopisteen avulla. Nashin tasapainopiste esitetään funktion kiintopisteenä seuraavasti: Olkoon $y : \Delta \mapsto \Delta$, s.e. i, j:s komponentti on

$$y_{ij}(p) = \frac{p_{ij} + g_{ij}(p)}{1 + \sum_j g_{ij}(p)} \tag{3.10}$$

missä

$$\begin{aligned} x_{ij}(p) &= u_i(s_{ij}, p_{-i}) \\ z_{ij}(p) &= x_{ij}(p) - u_i(p) \\ g_{ij}(p) &= \max[z_{ij}(p), 0] \end{aligned}$$

$p^* \in \Delta$ on Nashin tasapainopiste, jos se on y:n kiintopiste.

Kolmiointi on laskennallisesti melko vaativa ja McKelveyn mukaan sitä ei ole implementoitu tietokokoneelle.

Algoritmin tarkempi kuvaus löytyy Scarfin papereista [3] ja [4]. Hirsch, Papdimitriou ja Vavasis [5] käsittelevät algoritmin kompleksisuutta.

3.2.2 Tasapaino funktion miniminä

Nashin tasapainoehto voidaan määritellä funktion minimin avulla. Tällöin voidaan tilan etsimiseen käyttää rajoitetun optimoinnin (*constrained optimization*) menetelmiä. Tällaiset menetelmät saattavat kuitenkin päätyä sellaisiin ratkaisuihin, jotka eivät ole Nashin tasapainopisteitä.

Funktion minimi -algoritmi on laskennallisesti vaativa algoritmi ja sopivan lähtöpisteen löytäminen voi olla hankalaa. Joissain tilanteissa minimi-algoritmi voi kuitenkin olla hyödyllinen: Polkua seuraavat algoritmit eivät välttämättä löydä kaikkia tasapainopisteitä. Minimi-algoritmi voi ainakin teoriassa löytää kaikki Nashin tasapainopisteet, koska jokaisella tasapainopisteellä on suppenemissäde joka on suurempi kuin 0. Tämä tietysti vaatii yleensä tarkkaan valitut lähtöpisteet, joiden määrittäminen saattaa olla työlästä.

Nashin tasapainotilan määrittämistä funktion miniminä kuvataan tarkemmin McKelveyn papereissa [1] ja [6].

4 Ekstensiiviset pelit

Ekstensiivinen peli (*extensive form game*) on peliteorian malli, joka kuvaa eksplisiittisesti sekventiaalisia pelejä. Normaalimuotoisissa peleissä pelaajan tulee valita strategiansa pelin alussa ja itse peli koostuu käytännössä tämän valinnan tekemisessä. Ekstensiivisissä peleissä pelaaja tekee valintoja jatkuvasti pelin edetessä.

Ekstensiivisessä pelissä pelaaja voi harkita strategiaansa uudelleen riippuen pelitilanteesta. Ekstensiiviset pelit jakautuvat täydellisen (*perfect information*) ja epätäydellisen informaation sisältäviin peleihin.

Täydellisen informaation peleissä pelaaja tietää kaikki aikaisemmat siirtonsa, kun taas epätäydellisessä pelaaja ei vastaavasti tiedä (tai muista) kaikkia aikaisempia siirtojaan.

Formaalimmin pelin tapahtumia kuvataan **puina** tai **metsinä**. Olkoon $(T, \prec)$ pari, joka koostuu äärellisistä joukosta T **solmuja**. Relaatio $\prec$ on binäärinen joukolle T. Merkintä $x \prec t$ tarkoittaa, että solmu x **edeltää** solmua t. Relaatiolta $\prec$ edellytetään, että se on transitiivinen ja syklitön, eli se määrittää normaalin puun. Tämän lisäksi vaaditaan, että jokaisella $t \in T$, joukko $P(t) = \{x \in T : x \prec t\ \}$ eli t:n edeltäjien joukko on täysin järjestetty relaatiolla $\prec$, mikä tarkoittaa sitä, että samaan lopputulokseen johtavat siirrot eivät ole sama asia, vaikka ne pelin kannalta muuten saattaisivatkin olla.

Alkusolmujen joukko on $W = \{w \in T : P(w) = \emptyset\}$, joukko $Y = T - W$ on **ei-alkusolmujen** joukko. Vastaavasti, joukko $Z = \{z \in T : P^{-1}(z) = \emptyset\}$ on **loppusolmujen** joukko ja $X = T - Z$ **ei-loppusolmujen** eli **strategisten solmujen** joukko. Solmuja kutsutaan strategisiksi, koska vain niissä pelaaja voi tehdä jonkun valinnan; loppusolmuissahan ei valinnanvaraa enää ole.

Oletuksesta, että $P(t)$ (solmun t edeltäjien joukko) on täysin järjestetty, seuraa että **välitön edeltäjä** -funktio $p_1 : Y \to X$, $p_1(y) = maxP(y)$ on hyvin määritelty. (Välitön edeltäjä alkiolle t on siis se joukon $P(t)$ alkio, joka on "suurin".) Olkoon $\ell(t)$ joukon $P(t)$ mahtavuus (*cardinality*) ja jos $\ell \geq 2$, määritellään

$p_\ell : \{y \in Y : \ell(y) \geq \ell\} \to X$ olemaan ℓ-taitos p_1:n yhdistelmästä itselleen. Käytännössä p_ℓ ottaa argumentikseen puun solmun ja palauttaa sen alkion, jonka ℓ-seuraajajoukosta tämä puun solmu on.

Jos $P(t)$ on äärellinen, on mahtavuus $\ell(t)$ joukon $P(t)$ alkioiden lukumäärä. Muussa tapauksessa mahtavuutta määriteltäessä joudutaan miettimään, voidaanko muodostaa bijektio luonnollisten lukujen ja $P(t)$:n välille tai voidaanko $P(t)$ samaistaa reaalilukuihin. Funktio p_ℓ on kuvaus jokaiselta vähintään ℓ edeltäjää sisältävältä ei-loppusolmulta strategisille solmuille. Määritellään p_0 olemaan identiteettifunktio joukolle T. Solmun $t \in T$ edeltäjien lukumäärä on siis $\ell(t)$ eli suurin kokonaisluku, jolle $p_{\ell(t)}(t)$ on määritelty.

Määritellään 0 esittämään ”luontoa” tai ”sattumaa” ja äärelliset joukot ja niille funktiot:

Joukko	Merkintä	Funktio
informaatio	H	$\eta : X \to H$
toiminto	A	$\alpha : Y \to A$
pelaaja	$I = \{1, \ldots, n\}$	$\iota : H \to \{0\} \cup I$

Funktio η kuvaa strategisen solmun informaatiojoukolle eli η:n avulla pelaaja ”tulkitsee” pelin tilanteen. Määritellään **informaatio-osiointi** η:n käänteisfunktion avulla: $\eta^{-1}(h)$, $h \in H$. Nyt η^{-1} on siis kuvaus informaatiojoukolta pelipuun solmuille (tilanteille). Usein merkitään $x \in h$ tarkoittamaan $\eta(x) = h$.

Funktio ι on kuvaus informaatiojoukolta pelaajien joukolle ja kertoo mikä pelaaja valitsee toiminnon, kun informaatio $h \in H$ tapahtuu. Käänteiskuvaus $H^i = \iota^{-1}(i)$ on niiden informaatijoukkojen joukko joista pelaaja i valitsee toimintonsa.

Kun pelaaja i valitsee toimintonsa saatuaan informaation $h \in H^i$, hän tietää vain, että jokin $\eta^{-1}(h)$:n määräämistä pelipuun solmuista on tapahtunut.

Jokaiselle $y \in Y$, $\alpha(y)$ tulkitaan tarkoittamaan toimitoa, joka valittiin y:n välittömässä edeltäjäsolmussa. Siispä $A(x) = \alpha(p_1^{-1}(x))$ on niiden toimintojen joukko, jotka voidaan tehdä puun solmussa $x \in X$.

Jotta valintaongelmista saisi mielekkäitä täytyy olettaa, että:

(a) Jos $y, y' \in Y$ ovat erilliset ja $p_1(y) = p_1(y') = x$ (sama edeltäjäsolmu), niin $\alpha(y) \neq \alpha(y')$ (toiminnot ovat eri).

(b) $A(x) = A(x')$ kaikilla h ja $x, x' \in h$.

Jokaiselle $h \in H$ olkoon $A(h)$ niiden toimintojen joukko, jotka valitaan h:n ilmetessä. Yleensä on kätevää olettaa, että samoja toimintoja ei valita saman informaation ilmetessä eli $A(h) \cap A(h') = \emptyset$, kun $h, h' \in H$ ovat erillisiä.

Peleissä voidaan ottaa huomioon myös satunnaisuus ja luonnon valinta. Mallista tulee kuitenkin yksinkertaisempi jos nämä jätetään huomiotta.

Tässä esitelty malli on ns. **täydellisen muistamisen** peli eli pelaajat tietävät/muistavat minkä pelipuun solmujen kautta he ovat päätyneet tähän tilaan.

Käyttäytymisstrategia on todennäköisyysmittojen joka määrittää sen jakauman, jolla agentti valitsee annetun toiminnon joukosta A saatuaan informaation

$h \in H$. Formaalimpi määritelmä, joka esitellään McKelveyn paperin [1] sivulla 22 mahdollistaa myös todennäköisyyden laskemisen sille, että tietty pelipuun solmu saavutetaan.

Jokaiselle pelipuunsolmulle voidaan määritellä **hyöty**. pelaajan strategia on siis todennäköisyysjakauma, jonka perusteella pelaaja tekee valintansa saatuaan informaation. Nashin tasapainotila on sellainen todennäköisyysjakauma, jonka avulla laskettu hyödyn odotusarvo on suurin siten, että vaihtamalla muiden agenttien strategiatodennäköisyysjakaumia, heidän asemansa ei voi parantua. Formaalimi kuvaus löytyy McKelveyn paperista [1] sivuilta 23—24.

Viitteet

[1] McKelvey, Richard D ja McLennan, Andrew. *Computation of Equilibria in Finite Games.*

[2] Osborne, Martin J. ja Rubinstein, Ariel. *A Course in Game Theory.*

[3] Scarf, H. *The approximation of fixed points of a continious mapping*, SIAM Journal of Applied Mathematics, 15 (1967): 1328–1343.

[4] Scarf, H. *The Computation of Economic Equilibria*, New Haven: Ylae Unviersity Press, 1973.

[5] Hirsch, M. D., Papdimitiriou C. H. *Exponential lower bounds for finding Brouwer fixed points* Journal of Complexity, 5 (1989): 379–416.

[6] McKelvey, R. D. *A Liapunov function for Nash equilibria*, mimeo, California Institute of Technology (1992).