Arrow-Debreu piacmodell és megoldása

Tardos Zsófia

Témavezető: *Illés Tibor*, docens
ELTE TTK Operációkutatási Tanszék

Budapest, 2010.

Tartalomjegyzék

1. Bevezető

A szakdolgozat témája egy közgazdaságtani probléma különböző modellezése, illetve az egyes modellekre vonatkozó megoldási módszererek bemutatása.

A kiinduló problémát L. Walras egy 1900-as cikkében vetette fel: egy piacon úgy szeretnénk beárazni a termékeket, hogy ha mindenki a lehető leggazdaságosabban vásárol be, akkor pont fogyjanak el a termékek. Azt, hogy mennyire gazdaságos egy vásárlás egy adott szereplő számára, úgy mérjük, hogy az i. szereplőhöz rendelünk egy u_i hasznossági függvényt, amely egy adott vásárlás-vektorhoz hozzárendel egy értéket, ami azt mutatja, hogy az illetőnek mennyire hasznosak a megvásárolt termékek.

Arrow és Debreu egy 1954-ben megjelent cikkben mutattak olyan feltételeket, amelyek teljesülése mellett biztosan létezni fog ilyen egyensúlyi ár. E szerint, ha a piac minden szereplője rendelkezik valamilyen tőkével (ez lehet pénz, termék, munkaerő), a hasznossági függvénynek nincs maximuma, teljesül, hogy ha

$u_i(x_i) > u_i(x_i')$, és $0 < t < 1$, akkor $u_i[tx_i + (1-t)x_i'] > u_i(x_i')$, valamint minden szereplő tőkéje egy korlátos mennyiség, akkor létezni fog egyensúlyi ár. Ebben a cikkben erre egy Banach-fixponttételen alapuló egzisztencia- bizonyítás szerepel, de az egyensúlyi ár konkrét kiszámítását nem adja meg. [1]

Ahhoz, hogy ezt a problémát modellezzük, el kell döntenünk, hogy a piacot úgy szeretnénk elképzelni, hogy vannak külön termelők és külön vásárlók: ez a *Fisher-modell*, vagy úgy, hogy a piacon lévő szereplők egyszerre árulják is a termékeiket, és egyszerre vásárolnak is: ez az *Arrow-Debreu-modell*.

Mindkét modellre megvizsgálunk két speciális esetet: azt, amikor a hasznossági függvények lineárisak, illetve azt, amikor a hasznossági függvények *Leontief-*

függvények, azaz $u_i(x) = min \frac{x_i}{f_{ij}}$, adott f_{ij} súlyokra.

A Fisher-modellben lineáris hasznossági függvények mellett bemutatunk két polinomiális idejű módszert az egyensúlyi ár kiszámítására. Ezek után megmutatjuk, hogy az Arrow-Debreu-modellben lévő egyensúlyi ár számítási problémát lineáris hasznossági függvények mellett hogyan tudjuk visszavezetni polinomiális idő alatt a Fischer-modellben lévő problémára.

A Fisher-modellben lévő Leontief-hasznossági függvények melletti egyensúlyi ár számítási problémát nem részletezzük külön, egy az 1. részben bemutatott nyomkövetési algoritmushoz hasonló polinomiális algoritmus található [7]-ben

Végül megmutatjuk, hogy az Arrow-Debreu-modellben lévő egyensúlyi ár számítási probléma NP-teljes, ha a hasznossági függvények Leontief-függvények, mert a nem nullösszegű Nash-egyensúly számítás visszavezethető rá, amiről tudjuk, hogy egy NP-teljes probléma.

2. Az egyensúlyi ár kiszámítására vonatkozó algoritmusok

2.1. 1. eset: Fisher-modell lineáris hasznossági függvényekkel

Adott n vásárló, és m különböző termék a piacon (mindegyikből 1 egység, és olyan termékeket képzelünk el, amely tetszőleges módon részekre osztható), az i. vásárló w_i kezdeti tőkével, és egy u_i lineáris hasznossági függvénnyel rendelkezik, amely egy x_i vásárlási terv-vektorhoz, (melynek j. eleme azt mutatja meg, hogy az illető mennyit vásárol a j. termékből), $\sum_{j=1}^{m} u_{ij}x_{ij}$ értéket rendeli, ahol u_{ij} adott hasznossági együttható az i. vásárlóra és j. termékre vonatkozóan.

Egy adott p árvektor esetén a vásárlók célja maximalizálni a hasznossági függvényeiket amellett a korlát mellett, hogy csak bizonyos pénzmennyiség áll rendelkezésükre.

$$max \quad \sum_{j=1}^{m} u_{ij}x_{ij}$$

$$\sum_{j=1}^{m} x_{ij}p_j = w_i \qquad (1)$$

$$x_{ij} \geq 0$$

Az egyensúlyi ár esetében a piacon lévő összes terméket felvásárolják a vásárlók, azaz olyan p árvektort keresünk, amelyre léteznek olyan $x(p)_i$ vektorok, amelyek az előző feladatnak optimális megoldásai, illetve amelyekkel minden j-re teljesül a következő egyenlőség:

$$\sum_{i=1}^{n} x(p)_{ij} = 1$$

Feltehető, hogy minden j-re legalább egy u_{ij} pozitív, különben a terméket törölhetnénk.

1. Állítás. *A fenti n darab optimalizációs feladat összefoglalható az alábbi konvex optimalizációs feladatban.*

$$max \sum_{i=1}^{n} w_i \, log \sum_{j=1}^{m} u_{ij} y_{ij}$$

$$\sum_{i=1}^{n} y_{ij} = 1 \quad \forall j \quad (2)$$

$$y_{ij} \geq 0 \quad \forall (i, j)$$

Bizonyítás. Azt kell bebizonyítanunk, hogy azon $(p, x_1, ..., x_n)$ vektor $(n + 1)$-esekre, amelyeket az alábbi rendszer megad, teljesül, hogy p egyensúlyi ár, azaz $x_i(p)$ optimális megoldása (1)-nek.

$$p_j = max_i \frac{w_i u_{ij}}{\sum_{j=1}^{m} u_{ij} {y^*}_{ij}} \quad (3)$$

$$x_{ij} = {y^*}_{ij}$$

ahol y^* az (1) egy optimális megoldását jelöli, és az első egyenlet jobb oldalán a (2)-ben maximalizálandó függvény parciális deriváltjait jelöli a maximumhelyen.

Ehhez következő három feltételt kell ellenőriznünk:

1. $\sum_{j=1}^{m} x_{ij} p_j = w_i \quad \forall i$

2. $\sum_{i=1}^{n} x(p)_{ij} = 1 \quad \forall j$

3. az i. vásárló maximalizálja a hasznosságát a p ár mellett, azaz pontosan olyan termékek közül vásárol, amelyekre $\frac{u_{ij}}{p_j}$ maximális: ha $\mu_i = max_s \frac{u_i s}{p_s}$ és $x_{ij} > 0$, akkor $\mu_i = \frac{u_{ij}}{p_j}$

2. teljesül, mert, ha összeadjuk (3) 2. egyenleteit minden i-re, akkor (2) második egyenletei alapján teljesülni fog az egyenlőség

2. Állítás. *Ha* $y^*{}_{ij} > 0$*, akkor*

$$p_j = \frac{w_i u_{ij}}{\sum_{j=1}^{m} u_{ij} y^*{}_{ij}}$$

Bizonyítás. Tegyük fel, hogy $y^*{}_{ij} > 0$, és $p_j > \frac{w_i u_{ij}}{\sum_{j=1}^{n} u_{ij} y^*{}_{ij}}$. Ekkor $\exists k$ hogy $\frac{w_k u_{kj}}{\sum_{j=1}^{m} u_{kj} y^*{}_{kj}} > \frac{w_i u_{ij}}{\sum_{j=1}^{m} u_{ij} y^*{}_{ij}}$, azaz az kj-edik parciális derivált értéke nagyobb lenne, mint a ki-edik, és így y_{kj} növelésével, és y_{ki} csökkentésével az optimalizálandó függvény értékét növelni tudnánk, ami lehetetlen. □

Az 1. feltétel az alábbi, a 2. állításból következő egyenlőségből következik:

$$x_{ij} p_j = x_{ij} \frac{w_i u_{ij}}{\sum_{j=1}^{m} u_{ij} y^*{}_{ij}}$$

ezeket az egyenleteket minden j-re összeadva pont az 1. feltételt kapjuk.

Már csak a 3. feltételt kell bebizonyítani. Mivel feltettük, hogy minden j-re legalább egy u_{ij} pozitív, ezért tudjuk, hogy minden p_j pozitív. (3) első egyenletéből kapjuk, hogy

$$\frac{u_{ij}}{p_j} \leq \frac{\sum_{j=1}^{m} u_{ij} y_{ij}}{w_i}$$

A 2. állítás szerint, ha x_{ij} pozitív, akkor a fenti egyenlőtlenség egyenlőséggel is teljesül. Azaz, ha $x_i j$ pozitív, akkor $\mu_i = \frac{u_i j}{p_j}$.

$\square$

Így a következőkben elég az alábbi feladattal foglalkoznunk:

$$max \sum_{i=1}^{n} w_i \, log \sum_{j=1}^{m} u_{ij} x_{ij}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j$$

$$x_{ij} \geq 0 \forall (i, j)$$

Ezt a rendszert átírva kapjuk:

$$max \sum_{i=1}^{n} w_i \, log \, u_i$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j$$

$$u_i - \sum_{j=1}^{m} u_{ij} x_{ij} \quad \forall i$$

$$u_i, x_{ij} \geq 0 \quad \forall (i, j)$$

Ennek a problémának egy általánosabb verziójával fogunk foglalkozni:

$$max \sum_{i=1}^{n} w_i \, log \, u_i$$

$$Ax = b$$

$$x \geq 0$$

ahol A egy $m \times n$-es maximális sorrangú mátrix, b egy n-dimenziós vektor, w_j-k pedig nemnegatív súlyok.

Azt a diagonális mátrixot, melynek átlóelemei egy adott vektor elemei, a vektort jelölő betű nagybetű megfelelőjével jelöljük.

A Karush-Kuhn-Tucker tétel alapján a fenti rendszer optimalitási feltétele:

$$Sx = w$$

$$Ax = b$$

$$-A^T y + s = 0 \quad (4)$$

$$x \geq 0 \quad s \geq 0$$

Mutatunk egy algoritmust, ami tetszőlegesen megközelíti a fenti rendszer megoldását polinomiális időben. Az algoritmus lényege, hogy kiindulunk egy belső pontból, majd iterációs lépéseket végzünk: egy iterációs lépésben egy olyan belső pont környezetében keresünk egy belső pontot, amelyről tudjuk, hogy közelebb van a pontos megoldáshoz, mint ahol állunk.

Tegyük fel, hogy az iterációs lépés előtt az $(\bar{x}, \bar{s}, \bar{y})$ pontban állunk:

$$A\bar{x} = b$$

$$-A^T\bar{y} + \bar{s} + = 0$$

$$\|\bar{S}\bar{x} - \bar{w}\| \leq \eta\mu$$

$$\bar{x} \geq 0 \quad \bar{s} \geq 0$$

ahol η egy adott 1-nél kisebb konstans, $\mu \geq 0$, és $\bar{w}_j \leq \max\{\mu, w_j\}$

Szeretnénk úgy elmozdulni $(\bar{x}, \bar{s}, \bar{y})$-ból, hogy továbbra is belső pontban maradjunk, de közelebb legyünk a megoldáshoz:

$$A(\bar{x} + \Delta x) = b$$

$$-A^T\bar{y} + \Delta y + \bar{s} + \Delta s = 0$$

$$(\bar{s} + \Delta s)(\bar{x} + \Delta x) = \tilde{w}$$

$$\bar{x} + \Delta x \geq 0$$

$$\bar{s} + \Delta s \geq 0$$

$$\tilde{w}_j = max\{1 - \frac{\eta}{\sqrt{n}\mu} w_j\}$$

$(\bar{s} + \Delta s)(\bar{x} + \Delta x) = \tilde{w} \Leftrightarrow \bar{s}\bar{x} + \bar{s}\Delta x + \bar{x}\Delta s + \Delta x \Delta s = \tilde{w} \quad \bar{s}\bar{x} = \bar{w}$, így $\bar{s}\Delta x + \bar{x}\Delta s + \Delta x \Delta s = \tilde{w} - \bar{w}$

Azért, hogy lineáris egyenletrendszert kelljen megoldanunk, töröljük a nemlineáris $\Delta x \Delta s$ tagot. A kapott rendszert már meg tudjuk oldani, majd megvizsgáljuk, hogy mekkora eltérést okoztunk azzal, hogy töröltük $\Delta x \Delta s$-et.

$\Delta x \Delta s$ törlése, és $A\bar{x} = b$ illetve $-A^T\bar{y} + \bar{s} + = 0$ után kapjuk, hogy

$$A\Delta x = 0$$

$$-A^T \Delta y + \Delta s = 0$$

$$\bar{S}\Delta x + \bar{X}\Delta s = \tilde{w} - \bar{w}$$

$$\bar{x} + \Delta x \geq 0$$

$$\bar{s} + \Delta s \geq 0$$

A 3. egyenletet $\bar{S}^{-1}$-vel balról szorozva: ($\bar{S}^{-1}$ létezik, hiszen $\bar{S}$ pozitív diagonális)

$$\Delta x + \bar{S}^{-1}\bar{X}\Delta s = \bar{S}^{-1}(\tilde{w} - \bar{w})$$

Ezt A-val balról szorozva, és $A\Delta x = 0$-t felhasználva:

$$A\bar{S}^{-1}\bar{X}\Delta s = A\bar{S}^{-1}(\tilde{w} - \bar{w})$$

$A\bar{S}^{-1}\bar{X}\Delta s = A\bar{S}^{-1}(\tilde{w} - \bar{w}) \Leftrightarrow A\bar{S}^{-1}\bar{X}A^T\Delta y = A\bar{S}^{-1}(\tilde{w} - \bar{w})$

$D := \bar{S}^{-1}\bar{X}$ pozitív diagonális mátrix.

$$ADA^T\Delta y = A\bar{S}^{-1}(\tilde{w} - \bar{w})$$

Ennek az egyenletrendszernek egyértelműen létezik megoldása, mivel A maximális sorrangú, így ADA^T invertálható.

Δy-ból Δs is egyértelmű, Δx-et pedig a $\Delta x = -\bar{S}^{-1}\Delta s + \bar{S}^{-1}(\tilde{w} - \bar{w})$ egyenletből számolhatjuk ki.

Tegyük fel, hogy az iterációs lépés előtt $\|\bar{S}\bar{x} - \bar{w}\| \leq \eta\mu$ állt fenn, ahol $\eta \in$ $\in (0{,}1)$. Bebizonyítjuk, hogy ekkor az iterációs lépéssel kapott

$$x^* = x + \Delta x$$

$$y^* = y + \Delta y$$

$$s^* = s + \Delta s$$

értékekre

$$\|S^*x^* - w^*\| \le \|\eta\mu^*\|$$

ahol $\mu^* = (1 - \frac{\eta}{\sqrt{n}})\mu$

$$p := \bar{X}^{-\frac{1}{2}}\bar{S}^{\frac{1}{2}}\Delta x$$

$$q := \bar{X}^{\frac{1}{2}}\bar{S}^{-\frac{1}{2}}\Delta s$$

$$r := (\bar{X}\bar{S})^{-\frac{1}{2}}(w^* - \bar{w})$$

$$p + q = r \text{ és } p^T q = 0$$

Az iteráció előtti állapotra vonatkozó feltételekből

$$\bar{x}_j\bar{s}_j \ge \bar{w}_j - \eta\mu \ge (1-\eta)\mu$$

És

$$\|w^* - Xs\| \le \|w^* - \bar{w}\| + \|\bar{w} - Xs\| \le 2\eta\mu$$

Így

$$\|r\| \le \frac{2\eta\sqrt{\mu}}{\sqrt{1-\eta}}$$

3. Állítás. [2]

$$\|p\|^2 + \|q\|^2 = \|r\|^2$$

$$|Pq| \leq \frac{\sqrt{2}}{4}|r|^2$$

Így

$$\|S^*x^* - w^*\|^2 = \|Pq\|^2 \leq \left(\frac{\sqrt{2}}{4}\|r\|^2\right)^2$$

Ez pedig tovább becsülhető az $\|r\|$-re kapott becslés alapján

$$\|S^*x^* - w*\|^2 \leq \left(\frac{\sqrt{2}\eta^2}{1-\eta}\mu\right)^2 \leq \frac{\sqrt{2}\eta^2}{((1-\eta)^2\mu^*)^2}$$

Így $\|S^*x^* - w*\| \leq \eta\mu^*\|$, $\mu^* = (1 - \frac{\eta}{\sqrt{n}})\mu$ teljesül, ha η-t úgy válastzjuk, hogy $\frac{\sqrt{2}\eta^2}{(1-\eta)^2\mu^*} \leq \eta$ (pl. $\eta = \frac{1}{4}$).

Még azt kell megvizsgálnunk, hogy az új x^*, s^*-ra $x^* \geq 0$ illetve $s^* \geq 0$ teljesül-e.

$$\|\bar{X}^{-1}(x^* - \bar{x})\| = \|(\bar{x}\bar{s})^{-\frac{1}{2}}p\| \leq \|(\bar{x}\bar{s})^{-\frac{1}{2}}\|\|p\| \leq \frac{\|r\|}{\sqrt{(1-\eta)\mu}} \leq \frac{2\eta}{1-\eta} < 1$$

Másrészt $\|\bar{X}^{-1}(x^* - \bar{x})\| = \||\bar{X}^{-1}x^* - e\|$, ahol $\bar{X}$ egy nemnegatív elemekből álló diagonális mátrix, így $x^* \geq 0$, és $\|\bar{S}^{-1}(s^* - \bar{s})\| = \|(\bar{X}\bar{s})^{-\frac{1}{2}}q$ amiből ugyanígy $s^* \geq 0$.

Induljunk olyan kezdeti pontból, amelyre $\mu_0 = \max(w)$! Kiszámoljuk, hogy hány iterációs lépést kell ahhoz végeznünk, hogy elérjük $\|Sx - w\| \leq \epsilon$-t. Ehhez elég, ha $s_j x_j \leq \frac{\epsilon}{\sqrt{n}}$. Ez teljesül, ha $\mu \leq \frac{\epsilon}{\sqrt{n}(1+\eta)}$, mivel $s_j x_j \leq (\eta + 1)\mu$.

4. Állítás. *A közelítő algoritmus futásideje polinomiális*

Bizonyítás. Egy iterációs lépés polinomiális idő alatt fut, mert egy lineáris egyenletrendszert kell megoldani hozzá. Induljunk olyan kezdeti pontból, amelyre ($\mu_0 = = \max(w_i)$)! Kiszámoljuk, hogy hány iterációs lépést kell ahhoz végeznünk, hogy elérjük $\|Sx - w\| \leq \epsilon$-t. Ehhez elég, ha $s_j x_j \leq \frac{\epsilon}{\sqrt{n}}$. Ez teljesül, ha $\mu \leq \leq \frac{\epsilon}{\sqrt{n}(1+\eta)}$, mivel $s_j x_j \leq (\eta + 1)\mu$. Azaz $O(\sqrt{n}\, log(\frac{n\, max(w_i)}{\epsilon}))$ iterációs lépésre van szükség.

□

Ezek után kell találnunk egy polinomiális időben futó algoritmust, ami egy kellően közelítő belső pontból kiszámítja a pontos megoldást.

Az Eisenberg-Gale-féle átírás bizonyításában láttuk, hogy (2) megoldásaira teljesül, hogy

$$p_j = \frac{w_i u_{ij}}{\sum_{k=1}^{m} u_{ik\}x_{ik}}}, x_{ij} > 0, \forall(i,j) \in B^*$$

$$p_j = \frac{w_i u_{ij}}{\sum_{k=1}^{m} u_{ik\}x_{ik}}}, x_{ij} = 0, \forall(i,j) \in Z^*$$

$$p_j = \frac{w_i u_{ij}}{\sum_{k=1}^{m} u_{ik\}x_{ik}}}, x_{ij} > 0, \forall(i,j) \in N^*$$

ahol B^* azon x_{ij} változók indexeit tartalmazza, amelyek pozitívak lehetnek egy optimális megoldásnál, N^* azokat, amelyekre $s_{ij} = p_j - \frac{w_i u_{ij}}{\sum_{k=1}^{m} u_{ik} x_{ik}}$ pozitív lehet egy optimális duális megoldásnál, Z^* pedig a maradékot.

Tetszőleges $(i,j) \in B^*$ indexekre $u_{ij} > 0$, és ha $(i,k) \in B^*$, akkor

$$\frac{u_{ij}}{p_j} = \frac{u_{ik}}{p_j}$$

ha pedig $(i,k) \in Z^*$ vagy $(i,k) \in N^*$, akkor

$$\frac{u_{ij}}{p_j} \geq \frac{u_{ik}}{p_j}$$

Így $\lambda_i = \frac{p_k}{u_{ik}}, \forall (i,k) \in B^*$ jóldefiniált.

Minden i-re teljesül,

$$\sum_{k=1}^{m} u_{ik}x_{ik} = \sum_{k=1}^{m} \frac{u_{ik}p_k}{p_k x_{ik}} = \sum_{k:(i,k)\in B^*} \frac{u_i k}{p_k} p_k x_{ik} =$$

$$= \sum_{k:(i,k)\in B^*} \frac{1}{\lambda_i} p_k x_{ik} = \frac{1}{\lambda_i} \sum_{k:(i,k)\in B^*} p_k x_{ik} = \frac{1}{\lambda_i} \sum_{k=1}^{m} p_k x_{ik}$$

Vezessük be az $y_{ij} = p_j x_{ij}$ változót!

Így a következő rendszert kapjuk:

$$u_{ij}\lambda_i = p_j, y_{ij} > 0, \forall (i,j) \in B*$$

$$u_{ij}\lambda_i = p_j, y_{ij} = 0, \forall (i,j) \in Z*$$

$$u_{ij}\lambda_i < p_j, y_{ij} = 0, \forall (i,j) \in N*$$

$$\sum_{i=1}^{n} y_{ij} = w_i$$

$$\sum_{j=1}^{m} = p_j$$

5. Állítás. *Ennek a rendszernek létezik y^*_{ij},p^*_j,λ^*_i megoldása, melynek elemei racionálisak lesznek, és a bithosszaik hossza maximum L, ahol L u_{ij} és w_i bithosszainak lineáris függvénye. [5]*

Mivel $y^*_{ij} \geq 0\, \forall (i,j) \in B^*$, és y^* bithossza maximum L, így

$$y^*_{ij} \leq 2^{-L} \quad \forall (i,j) \in B^*$$

és mivel ${p^*}_j > u_{ij}{\lambda^*}_i \quad \forall (i,j) \in N^*$, és p^* bithossza is maximum L, így $2^L \geq$ $\geq p^*_j \geq u_{ij}\lambda^*_i + 2^{-L} \quad \forall (i,j) \in N^*$.

Tehát $x^*_{ij} = \frac{{y^*}_{ij}}{{p^*}_{ij}} \geq 2^{-2L} \quad \forall (i,j) \in B^*$, és

$${s^*}_{ij} = {p^*}_j - p_j - \frac{w_i u_{ij}}{\sum_{k=1}^{m} u_{ik\}x_{ik}}} \geq 2^{-2L} \quad \forall (i,j) \in N^*$$

A megoldandó rendszert átalakítjuk egy lineáris egyenletekből egyenlőtlenségekből álló rendszerré.

Legyen $P^k = \{j : x^k_j \geq s^k_j\}$, ahol x^k és s^k a k. iterációban kapott értékek. η-t válasszuk $\frac{1}{4}$-nek! Bebizonyítjuk, hogy egy adott k-ra elérhető, hogy $B^* \subset$ $\subset P^k$ és $N^* \cap P^k = \emptyset$. Ha ez teljesül, akkor a megoldandó rendszer átalakítható a következő rendszerré, amely polinomiális időben megoldható.

$$u_{ij}\lambda_i = p_j,\ y_{ij} \geq 0,\ \forall(i,j) \in P^k$$

$$u_{ij}\lambda_i \leq p_j,\ y_{ij} = 0,\ \forall(i,j) \notin P^k$$

$$\sum_{i=1}^{n} y_{ij} = w_i\ \forall i$$

$$\sum_{j=1}^{m} = p_j$$

Legyen $W := \{j : w_j > 0\}$, és x, s valamelyik iterációs lépés után kapott értékek, μ pedig a hozzájuk tartozó hibabecslő. Kiszámoljuk, hogy milyen kicsinek kell lennie μ-nek ahhoz, hogy $B^* \subset P^k$ és $N^* \cap P^k = \emptyset$ teljesüljön.

$$w_j - \frac{1}{4}\mu \leq x_j s_j \leq w_j + \frac{1}{4}\mu \quad \forall j \in W$$

$$(1-\mu)\frac{1}{4} \leq x_j s_j \leq (1+\mu)\frac{1}{4}\forall j \notin W$$

Legyen $(\dot{x}, \dot{s})$ egy pontos megoldás!

Mivel

$$(x - x^*)^T(s - s^*) = 0$$

ezért

$$s^T\dot{x} + x^T\dot{s} = x^T s + \dot{x}^T\dot{s} \leq \sum_{j\in W} 2w_j + \mu\frac{1}{4} + \sum_{j\notin W}(1+\frac{1}{4})\mu$$

Azaz

$$\sum_{j\in W}(\dot{x}_j s_j + \dot{s}_j x_j) + \sum_{j\notin W}(\dot{x}_j s_j + \dot{s}_j x_j) \leq \sum_{j\in W} 2w_j + \frac{1}{4}\mu + \sum_{j\notin W}(1+\frac{1}{4})\mu$$

A számtani-mértani közepek közti egyenlőtlenséget felhasználva $\forall j \in W$-re kapjuk, hogy

$$\dot{x}_j s_j + \dot{s}_j x_j \geq 2\sqrt{x_j s_j \dot{x}_j \dot{s}_j} \geq 2w_j\sqrt{1-\frac{\frac{1}{4}\mu}{w_j}} \geq 2w_j\left(1-\frac{\frac{1}{4}\mu}{w_j}\right) = 2w_j - \frac{1}{2}\mu$$

Ezekből pedig

$$\sum_{j\notin W}\dot{x}_j s_j + \dot{s}_j x_j \leq \sum_{j\in W} 2w_j + \frac{1}{4}\mu + \sum_{j\notin W}\left(1+\frac{1}{4}\right)\mu) - \sum_{j\in W}(2w_j - \frac{1}{2}\mu) =$$

$$= \sum_{j\in W}\frac{3}{4}\mu + \sum_{j\notin W}\frac{5}{4}\mu \leq \frac{5}{4}\mu n$$

Most kiszámoljuk, hogy mekkora μ esetén fog teljesülni j-re, hogy $B^* \subset P^k$ és $N^* \cap P^k = \emptyset$.

Feltesszük, hogy $1 \notin W$, ha ez nem teljesülne, akkor egy nagyobb indexre, ami teljesíti a feltételt végig lehetne mondani az alábbi gondolatmenetet.

Ha $x_1^* > 0$ és $s^* = 0$, akkor $s_1 x_1^* = s_1 x_1^* + x_1 s_1^* \leq \frac{5}{4}\mu n$ az előző becslés alapján.

Ebből, illetve az $x_j s_j$ szorzatra vonatkozó becslésből kapjuk, hogy $\frac{5}{4}x_1\mu n \geq$ $\geq x_1 s_1 x_1^* \geq \frac{3}{4}\mu x_1^*$.

Így ebben az esetben:

$$x_1 \geq \frac{3x_1^*}{5n}$$

$$s_1 \geq \frac{5n\mu}{4x_1^*}$$

Ha pedig $s_1^* > 0$ és $x_1^* = 0$, akkor ugyanígy kapjuk, hogy

$$s_1 \geq \frac{3s_1^*}{5n}$$

$$x_1 \geq \frac{5n\mu}{4s_1^*}$$

Ezek szerint, ha

$$\mu < \frac{12}{25n^2} \min\{(x_j^* + s_j^*)^2 : j \in B^* \cup N^*\}$$

akkor teljesülni fog, hogy $B^* \subset P^k$ és $N^* \cap P^k = \emptyset$.

A pontos megoldás bithosszára vonatkozó állítás következményeként

$\min\{(x_j^* + s_j^*)^2 : j \in B^* \cup N^*\} \geq 2^{-2L}$. Így $O(\sqrt{mn}L)$ iterációs lépés után elég kicsi lesz μ ahhoz, hogy már egy lineáris egyenlet megoldásával megkapható legyen a pontos megoldás.

Összefoglalva a (4)-es rendszer megoldására vonatkozó algoritmus a következő:

Bemenő adatok: A, b, w

1. Keresünk az alábbi lineáris rendszernek egy (x^0, y^0, s^0) megoldását.

$$Ax = b$$

$$-A^T y + s = 0$$

$$x \geq 0$$

$$s \geq 0$$

2. Kiszámítjuk a közelítő értékeket.

   Begin

   $\overline{x} := x^0, \overline{y} := y^0, \overline{s} := s^0$

   while $\mu \geq \frac{12}{25n^2} \min\{(x_j^* + s_j^*)^2 : j \in B^* \cup N^*\}$ **do**

   Kiszámítjuk Δx-et Δy-t és Δs-t.

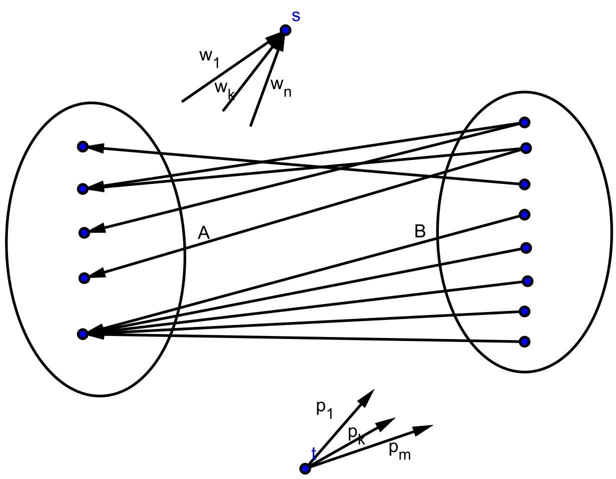
   $\overline{x} := \overline{x} + \Delta x, \overline{y} := \overline{y} + \Delta y, \overline{s} := \overline{s} + \Delta s.$

   end

3. Kiszámítjuk a pontos megoldást egy lineáris egyenletrendszer megoldásával.

2.2. DPSV algoritmus

Mutatunk egy másik polinomiális algoritmust az egyensúlyi ár kiszámítá- sára a Fishher-modellben, ahol a hasznossági függvények lineárisak. Megint van n vásárlónk w_i kezdeti tőkékkel és u_i hasznossági függvényekkel illetve m termékünk. Világos, hogy egy adott p ár mellett az i. vásárló akkor maximalizálja a hasznosságát, ha olyan termékekből vásárol, amelyekre $\frac{u_{ij}}{p_j}$ maximális. Definiáljuk a következő páros gráfot egy adott p árvektorhoz! Az A csúcsosztályban lesznek a vevőknek megfelelő csúcsok, a B csúcsosztályban a termékeknek megfelelő csúcsok. Az $i \in$ $\in A$ és $j \in B$ csúcsok között akkor fut él, ha $\frac{u_{ij}}{p_j}$ maximális. Irányítsuk meg ezeket az éleket úgy, hogy B-ből A-ba mutassanak, az élek kapacitása legyen végtelen. Vegyünk fel egy s forráspontot és egy t nyelőt. Az s pontból fusson minden $j \in$ $\in B$-hez él, és ezek kapacitása legyen p_j. A t pontba fusson minden $i \in A$-ból él, és ezek kapacitása legyen w_i. Így kapunk minden p-hez egy $N(p)$ hálózatot.



Vegyük észre, hogy az $N(p)$ hálózatban egy maximális folyam megfelel egy olyan vásárlásnak, amelyben maximális mennyiségű terméket vásárolnak fel a vevők a következő feltételek mellett: 1. nem haladja meg az elköltött pénzmennyiség a kezdőtőkéjüket 2. csak olyan termékből vásárolnak, amelyre maximális az érték/ár arány

1. Tétel. *(MFMC): Egy hálózatban a maximális folyam értéke megegyezik a minimális vágás értékével.*

6. Állítás. *Ha egy p árra $N(p)$-ben az $(s, A \cup B \cup t)$ egy minimális vágás, akkor emellett az ár mellett a vevők fel fognak vásárolni minden terméket. (Ezt a tulajdonságot nevezzük *-tulajdonságnak.)*

Az algoritmusban végig *-tulajdonságú p árakon fogunk lépkedni. Azt kell elérnünk, hogy addig növeljük az árat, amíg találunk ezek között a p-k között egy olyat, amelyikre minden vevő elkölti az összes pénzét. Ez azt jelenti, hogy $N(p)$-ben a maximális folyam értéke meg fog egyezni a vásárlók össztőkéjével.

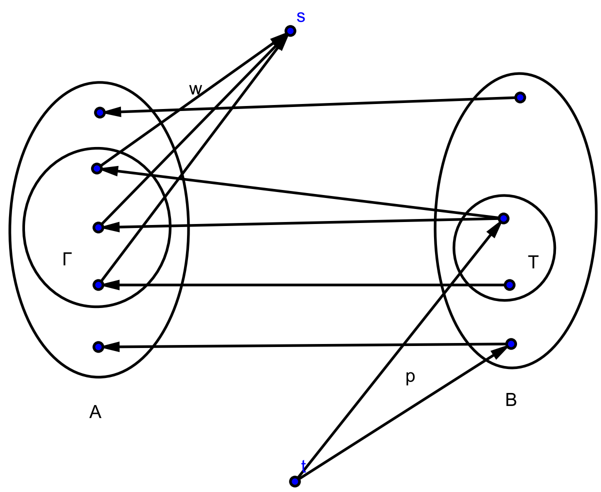
Ahhoz, hogy el tudjuk kezdeni az algoritmust szükségünk van egy *-tulajdonságú árra. Ehhez először legyen $p_j = \frac{min\, w_i}{m}$, így minden vevő tetszőleges mennyiségben tud vásárolni. Még arra van szükség, hogy $N(p)$-ben minden termékből mutasson él. Ha valamelyik termékből nem mutat él, akkor az ár megfelelő csökkentésével elérhető, hogy mutasson bele is él, de ne is törlődjön máshonnan él. Így eljutunk egy megfelelő p_0 kezdőárhoz.

Keresünk egy olyan tulajdonságát $N(p)$-nek, amellyel leírható, hogy p *-tulajdonságú-e.

$$S \subseteq A \;\; m(S) := \sum_{i \in S} w_i$$

$$T \subseteq B \; q(T) := \sum_{j \in T} p_j$$

$T \subseteq B \Gamma(T)$: azon csúcsok A-ban, amelyekbe megy él T valamelyik pontjából.



7. Állítás. *p *-tulajdonságú* $\Leftrightarrow \forall T \subseteq B \; q(T) \leq m(\Gamma(s))$

Bizonyítás. $\Rightarrow$: Ha p *-tulajdonságú, akkor minden terméket feltudnak vásárolni a vevők. Ez lehetetlen lenne, ha lenne olyan halmaza a termékekeknek, hogy az ezen termékekben érdekelt vásárlóknak kevesebb össztőkéje lenne, mint a termékek árának összege.

$\Leftarrow$: Tegyük fel, hogy $N(p)$-ben $(s \cup A_1 \cup B_1, A_2 \cup B_2 \cup t)$ egy minimális vágás. Ekkor $\Gamma(B_1) \subseteq A_1$, különben futna B_1 és A_2 között él, így a vágás értéke végtelen lenne. A feltevés szerint, ha B_1-et és $\Gamma(B_1)$-et áttesszük a másik csúcshalmazba, úgy a vágás értéke nem nő, azaz $(s, A \cup B \cup t)$ egy minimális vágás.

$\square$

Az algoritmus lényege a következő: megkeressük azt a legnagyobb T halmazt, amelyre $m(\Gamma(T)) = q(T)$ (azokat az S halmazokat, amelyek $m(\Gamma(S)) = q(S)$

tulajdonsággal rendelkeznek, *feszes halmazoknak* fogjuk nevezni) . Világos, hogy a T-beli termékekre nem szabad növelnünk az árat, mert akkor megszűnik a *-tulajdonság. $T \cup \Gamma(T)$ részgráfot passzív részgráfnak, a $(B \setminus T \cup A \setminus \Gamma(T))$ részgráfot aktív részgráfnak nevezzük. A megmaradt termékek árát elkezdjük egy 1-től induló és folyamatosan növekvő x számmal szorozni addig, amíg a következő események valamelyike be nem következik:

1. esemény: Keletkezik egy $R \neq \emptyset$ halmaz az aktív termékekek között, amelyre $m(\Gamma(T)) = q(T)$

2. esemény: Keletkezik egy él $i \in A \setminus \Gamma(T)$ és $j \in T$ között.

Ha 1. következik be, akkor R-et illetve $\Gamma(R)$-et hozzávesszük a passzív részgráfhoz. Ha 2. következik be, akkor vizsgáljuk meg, hogy melyik az a legkisebb T-n belüli T_j halmaz, amely tartalmazza j-t, és amelyre teljesül, hogy $m(\Gamma(T_j)) =$ $= q(T_j)$ volt még az előző lépésben. Az új él behúzása után ez az egyenlőség már nem fog teljesülni, tehát $\Gamma(T_j)$-t és T_j-t át kell helyeznünk az aktív részgráfba.

Azt az árat, ami mellett 2. következik be, ki tudjuk számítani polinomiális idő alatt. Kell találnunk egy polinomiális algoritmust arra, hogy ki tudjuk számítani az a legkisebb x^* értéket, amelyre keletkezik az $N(px^*)$ aktív részgráfjában egy nemüres feszes halmaz.

Meg tudjuk mondani, hogy $x^* = min_{0 \neq S \subseteq B} \frac{m(\Gamma(s))}{q(S)}$, csak ezzel az a baj, hogy exponenciálisan sok részhalmazra kellene kiszámítanunk a $\frac{m(\Gamma(s))}{q(S)}$ értéket, hogy megtaláljuk a minimálist.

Adunk egy minimális vágásokkal való leírását az x^* értéknek, amely lehetővé teszi, hogy m darab minimális vágás kereséssel ki tudjuk számítani x^*-ot. (A minimális vágás keresése pedig polinomiális idő alatt megoldható a Ford-Fulkerson algoritmussal.)

8. Állítás. *1. Ha $x \leq x^*$, akkor $N(px)$-ben $(s, A \cup B \cup t)$ egy minimális vágás.*

2. Ha $x > x^*$, akkor $(s, A \cup B \cup t)$ nem minimális vágás. Sőt, ha $(s \cup A_1 \cup \cup B_1, A_2 \cup B_2 \cup t)$ egy minimális vágás, akkor az $N(px^*)$-ban feszessé váló S^*-ra teljesülni fog, hogy $S^* \subseteq B_1$.

Bizonyítás. 1. $x^* = min_{\emptyset \neq S \subseteq B} \frac{m(\Gamma(s))}{q(S)}$, azaz ha $x \leq x^*$, akkor $\forall S \subseteq B : xq(T) \leq \leq m(\Gamma(S))$ Ezek szerint px *-tulajdonságú, tehát $(s, A \cup B \cup t)$ egy minimális vágás.

2. Ha $x > x^*$, akkor $xq(S^*) > x^*q(S^*) = m(\Gamma(S^*))$, így az

$(s \cup S^* \cup \Gamma(S^*), t \cup A \backslash \Gamma(S^*) \cup B \backslash S^*)$ vágás értéke szigorúan kisebb, mint az $(s, A \cup B \cup t)$ vágás értéke, így ez utóbbi nem lehet minimális vágás.

$S_2 =: S^* \cap B_2$ és $S_1 := S^* - S_2$. Azt kell bebizonyítanunk, hogy $S_2 = \emptyset$. Tegyük fel, hogy $S_2 \neq \emptyset$. $\Gamma(S_1) \subseteq A_1$, különben végtelen lenne a minimális vágás értéke. $(s \cup A_1 \cup B_1, A_2 \cup B_2 \cup t)$ vágás minimialitása miatt $m(\Gamma(S_2 \cap A_2) < xq(S_2)$. Így $S_2 \neq S^*$, azaz $S_1 \neq \emptyset$. A minimalitásból $m(\Gamma(S_2) \cap A_2) \geq xq(S_2) = x^*q(S_2)$ is következik.

Másrészt $m(\Gamma(S_2) \cap A_2) + m(\Gamma(S_1)) \leq x^*(m(S_2) + m(S_1))$. Ezekből $x^* < < \frac{m(\Gamma(S_1))}{q(S_1)}$ következne, ami lehetetlen, mivel $x^* = \frac{m(\Gamma(S_1))}{q(S_1)}$.

□

Így, x^*-ot a következő módon tudjuk kiszámítani:

1. $x_0 = \frac{q(B)}{m(A)}$ értékből indulunk. Világos, hogy $x \geq x^*$. Így, ha $N(px_0)$-ban $(s, A \cup B \cup t)$ egy minimális vágás, akkor $x = x^*$.

2. Ha $(s \cup A_1 \cup B_1, A_2 \cup B_2 \cup t)$ egy minimális vágás, akkor B_1 valódi részhalmaza B-nek. (Ha $B_1 = B$ lenne, akkor $A_1 = A$ is, különben végtelen nagy lenne a vágás értéke.) Így most a fenti állítás szerint a feszes halmazt elég a

kisebb B_1 halmazban keresni. Azaz visszatérünk az 1. lépéshez az $s, B_1, \Gamma(B_1), t$ csúcsokat tartalmazó gráffal.

Ezek szerint maximum m-szer kell lefuttatnunk egy minimális vágás keresést, azaz az algoritmus polinomiális.

Legyen $\gamma_i(p, f)$ az i. vevő megmaradó pénze $N(p)$ hálózatban folyó f folyam esetén, $\|.\|$ pedig jelentse az euklideszi vektornormát. Egy maximális folyamot kiegyensúlyozottnak fogunk nevezni, ha minimális lesz rá N(p)-ben $\|\gamma(p, f)\|$ értéke.

Az a célunk tehát, hogy N(p)-ben az $(s, A \cup B \cup t)$ minimális vágáshoz tartozó maximális folyamra $\|\gamma(p, f)\| = 0$ legyen.

A kiegyensúlyozott folyam megtalálására található egy polinomiális algoritmus [3]-ben, amely az alábbi állításon alapszik.

Jelölje $R(p, f)$ az $N(p, f)$ megmaradó hálózatát az f folyam mellett.

9. Állítás. *Egy maximális folyam pontosan akkor kiegyensúlyozott $N(p)$-ben, ha nincsenek olyan i, j vevők, hogy $\gamma_i(p, f) \geq \gamma_j(p, f)$, és i-ből van út j-be $R(p, f)$-ben.*

A teljes algoritmus:

1. Kiszámítunk egy kiinduló *-tulajdonságú árat.
2. Megkeressük az aktív részgráfot $N(p)$-n belül.
3. Növelni kezdjük p-t az aktív részgráfon belül.
4. Ha az 1. esemény következik be, visszalépünk az 1. lépéshez.
5. Ha a 2. esemény következik be, akkor bevesszük az új élet, majd kiszámítunk egy kiegyensúlyozott folyamot, és hozzávesszük az aktív részgráfhoz azokat

a csúcsokat $R(p, f) \backslash s, t$-ből, amelyekből irányított úton elérhető valamelyik aktív részgráfbeli termék, majd visszalépünk az 1. lépéshez.

Nevezzük fázisnak azokat a részeit egy algoritmusnak, amik egy 1. esemény bekövetkezésével kezdődnek, és addig tartanak, amíg be nem következik egy újabb 1. esemény.

Világos, hogy a 4. lépésben mindig hozzáveszünk legalább egy csúcsot az aktív részgráfhoz, így legfeljebb csúcsszámnyi lépésenként be fog következni az 1. esemény.

Azt kell bebizonyítanunk, hogy egy fázis alatt kellő mértékben nő az aktuális p érték.

10. Állítás. *Egy fázis alatt lesz legalább egy i vevő, amelyre $\gamma_i(p, f)$ legalább $\frac{\delta}{m}$-mel csökken, ahol δ a maximális megmaradó tőkét jelöli a fázis elején.*

Bizonyítás. Biz: Mivel a fázis elején minden aktív részgráfbeli vevő megmaradó tőkéje δ, és a fázis végére kell lennie egy i vevőnek, akinek 0-ra csökken a megmaradó tőkéje, ezért van egy olyan lépés ebben a fázisban, ahol legalább $\frac{\delta}{m}$-mel csökken ennek a vevőnek a megmaradó tőkéje. $\square$

11. Állítás. *Ha f egy megengedett folyam, f' pedig egy kiegyensúlyozott folyam, és az i. vevőre $\gamma_i(p, f) = \gamma_i(p, f') + \delta$, akkor $\|\gamma(p, f)\|^2 \geq \|\gamma(p, f')\|$.*

A fenti állításból kiszámítható, hogy

12. Állítás. *Ha p az ár egy fázis elején és p' pedig a következő fázis elején, akkor $\|\gamma(p')\|^2 \preceq \|\gamma(p)\|^2(1 - \frac{1}{n^3})$.*

Ez az eredmény garantálja, hogy az algoritmus polinomiális időben kiszámolja az egyensúlyi árat.

2.3. 2. eset: Arrow-Debreu-modell lineáris hasznossági függvényekkel

Adott n kereskedő és m termék a piacon, az i-edik kereskedő b_{ij} mennyiséggel rendelkezik a j-edik termékből. Feltesszük, hogy minden termékből 1 egységnyi szerepel összesen a piacon. Az i-edik kereskedő lineáris hasznossági függvénye u_i. Egy kereskedő annyi pénzért tud vásárolni a többiektől, amennyiért ő el tudja adni a termékeit, azaz egy adott p árvektorra az $i.$ kereskedő p által indukált kezdőtőkéje $w_i(p) = \sum_j p_j b_{ij}$

$$\sum_{j=1}^{m} p_j x_{ij} \leq \sum_{j=1}^{m} p_j b_{ij} \quad \forall i$$

ahol x_{ij} jelöli azt, hogy az i-edik kereskedő mekkora mennyiséget vásárol a j-edik termékből.

Azaz a fő különbség a Fisher-modell-beli problémához képest az, hogy itt nem egy adott kezdőtőkével rendelkeznek a kereskedők, hanem az elkölthető pénzük is az árvektor függvénye. A célunk olyan árvektort találni, amely olyan elkölthető pénzt indukál a kereskedők számára, hogy ha mindenki optimalizálja a hasznossági függvényét, akkor minden terméket pont felvásároljanak.

Azaz, $\overline{p}$ egyensúlyi ár, ha a következő rendszernek van megoldása $\overline{p}$ mellett:

$$\sum_{j=1}^{m} u_{ij} x_{ij} = max_{y \geq 0} \sum_{j=1}^{m} u_{ij} y_{ij} \quad \forall i$$

$$\sum_{j=1}^{m} \overline{p}_j x_{ij} \leq \sum_{j=1}^{m} \overline{p}_j b_{ij} \quad \forall i$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j$$

$$x_{ij} \geq 0 \quad \forall (i,j)$$

Mutatunk az egyensúlyi ár kiszámítására egy polinomiális ϵ-közelítő algoritmust, amely Fischer-modell-beli egyensúlyi ár kiszámítására vezeti vissza a problémát. $N(p)$ azt a hálózatot jelenti, amit a p által indukált kezdőtőkével kapunk.

Definiáljuk $T \subseteq B$-re $\varphi(T)$-t: $\varphi(T) = q(T) - m(\Gamma(T))$.

$max\varphi(p) =: max_{T \subseteq B} \varphi(T)$.

13. Állítás. *Ha $N(p)$-ben egy minimális vágás egyik oldala $s \cup S \cup T$, akkor $S = \Gamma(T)$, és $\varphi(T) = max\varphi(p)$.*

Ebből fakadóan egy olyan T halmaz találása, melyre $\varphi(T) = max\varphi(p)$ teljesül, megoldható egy minimális vágás kereséssel, ami polinomiális idejű.

Az algoritmus:

1. Kiindulunk egy tetszőleges p árvektorból, kiszámítjuk minden vevő p által indukált kezdőtőkéjét ($w_i(p)$), és elkészítjük $N(p)$-t. (Legelső körben a csupa-1 árvektorból indulunk.)

2. Keresünk egy olyan T halmazt, amelyre $\varphi(T) = max\varphi(p)$ teljesül.

   $D =: \varphi(T)$. Ha $D = 0$, akkor készen vagyunk, mert p egyensúlyi ár.

3. Beveszünk egy $n+1$-edik fantomvásárlót a piacba: az ő kezdőtőkéje D lesz, és $u_{n+1,j} = p_j$, azaz a fantomvásárló számára megegyezik minden termékre az érték/ár arány. Az így kapott hálózatot $N^*(p) - vel$ jelöljük.

4. Lefuttatjuk az előző fejezetben leírt DPSV algoritmust az így kapott Fischer-modell-beli piacon, amely kiszámít egy p' egyensúlyi árat.

5. Kiszámoljuk, hogy p' mellett mennyi lenne a vevők kezdőtőkéje: $w_i(p')$. Ha $\forall i \frac{w_i(p')}{w_i(p)} \leq 1+\epsilon$, akkor megállunk. Ha nem, akkor p'-vel újraindulunk.

Három dolgot kell ellenőriznünk:

1. A 4. lépésben lefuttatunk egy DPSV algoritmust. A DPSV algoritmus kiinduló árának *-tulajdonságúganak kell lennie, így le kell ellenőriznünk, hogy ez teljesül-e itt.

2. Ez az algoritmus egy ϵ-közelítő algoritmus, azaz, ha az output $\overline{p}$, akkor teljesülnie kell, hogy van olyan vásárlás (x_{ij}: ennyit vesz az i-edik vevő a j-edik termékből), hogy az alábbi két feltétel teljesül (azaz majdnem minden termék elfogy, és mindenki majdnem maximalizálja a haszmosságát).

a)

$$(1-\epsilon)\sum_{i=1}^{n} b_{ij} \leq \sum_{i=1}^{n} x_{ij} \forall\ j$$

b)

$$\sum_{j=1}^{m} u_{ij}x_{ij} \geq max_{y\geq 0}\sum_{j=1}^{m} u_{ij}y_{ij}\forall i$$

3. Az algoritmus polinomális.

1. p *-tulajdonságú $N^*(p)$-ben, mivel a fantom vásárló minden termékkel össze van kötve, így $\forall T \subseteq Bq(T) \leq m(\Gamma(s))$, hiszen az $n+1$. vásárló benne van $\Gamma(s) - ben.$

Tegyük fel, hogy $p^* = p^n$. (p^n: az n-edik kiszámított p).

2.a) bizonyítása

Azt bizonyítjuk be, hogy minden szereplő elkölti a pénzének legalább az $(1-\epsilon)$-szorosát, ebből pedig már következik az állítás, hiszen mivel a hasznossági függvények lineárisak, ezért mindenki pontosan azokból a termékekből vásárol, amelyekre az érték/ár arány maximális, ez azt jelenti, hogy mindenki felvásárolja legalább az $(1-\epsilon)$-od részét annak az árumennyiségnek, amit akkor vásárolna meg, ha az összes pénzét elköltve maximalizálhatná szabadon a hasznossági függvényét.

Az, hogy minden kereskedő elkölti a pénzének legalább az $(1-\epsilon)$-szorosát, abból látszik, hogy $p^* = p^n$ egyensúlyi ár volt abban a Fischer-modellben, ahol az $i.$ vásárló kezdeti tőkéje $\sum_{j=1}^{n} p_j^{n-1} w_{ij}$.

2.b) bizonyítása

Mivel mindenki mindenki felvásárolja legalább az $(1-\epsilon)$-od részét annak az árumennyiségnek, amit akkor vásárolna meg, ha az összes pénzét elköltve maximalizálhatná szabadon a hasznossági függvényét, ezért mindenki számára az elért hasznosság legalább az $(1-\epsilon)$-od része annak, mintha szabadon maximalizálhatná az összes pénze elköltése mellett a hasznossági függvényét.

3. bizonyítása

14. Állítás. *D értéke nem nő az algoritmus futása alatt*

Bizonyítás. p-ből indulunk, és a 4. lépésben kapott új érték p': a DPSV algoritmus végig *-tulajdonságú árakon lépked, így fennáll

$q^{*\prime}_p(T) \leq m^*(\Gamma^{*\prime}_p(T)) \forall T \subseteq B$ (A csillaggal ellátott betűk ugyanazt jelentik az $N^*(p)$ hálózatban, mint a csillag nélküli megfelelőjük az N(p) hálózatban.) Így, ha kivesszük a fantomvevőt: $q^{*\prime}_p(T) - m^*_p(\Gamma^{*\prime}_p(T)\{n+1\}$, és $q^{*\prime}_p(T) = q'_p(T)$.

Másrészt mivel a DPSV algoritmus sosem csökkenti semelyik termék árát, így $\forall j p'_j \geq p_j$. Így

$$m^*(\Gamma^{*}{}'_p(T)\backslash n+1 = \sum_{i \in Gamma^{*}{}'_p(T)} \sum_j p'_j b_i j \leq \sum_{i \in Gamma^{*}{}'_p(T)} \sum_j p'_j b_i j = q_{p'}(T)$$

Ebből

$$\varphi_{p'}(T) = q_{p'}(T) - m_{p'}(\Gamma_{p'}(S) \leq q^{*}{}'_p(T) - m^*{}_p(\Gamma^{*}{}'_p(T)\backslash\{n+1\}) \leq D$$

$\square$

15. Állítás. *Az algoritmus véget ér $O(\frac{n}{\epsilon} log(\frac{m}{\epsilon min_i \sum_j b_{ij}}))$ lépés alatt.*

Bizonyítás. Egy kiszámított ár legyen p, a következő lépésben kiszámított ár pedig p'. Mivel minden termékből pontosan egy egységnyi van a piacon, ezért az aktuális ár által indukált összkezdőtőke megegyezik a termékek árának összegével:

$\sum_i w_i(p') = \sum_j p'_j = \sum_j p_j + D = \sum_i w_i(p) + D$. Tudjuk, hogy az ár semelyik lépésben sem csökken, így a kezdőtőkék sem csökkenhetnek, azaz $w_i(p') -$ $- w_i(p) \leq D$. Tudjuk, hogy D egyik lépésben sem nő, és az első lépésben pedig $\leq m$, hiszen akkor ennyi a termékek összára. Így $w_i(p') - w_i(p) \leq D$ is következik. Ebből: $\frac{w_i(p')}{w_i(p)} \leq 1 + \frac{m}{w_i(p')}$. Az algoritmusunk akkor áll le, ha $\forall i \frac{w_i(p')}{w_i(p)} \leq 1 + \epsilon$. Ez csak akkor nem következik be, ha $\exists i w_i(p) < \frac{m}{\epsilon}$. Ekkor $\frac{w_i(p')}{w_i(p)} \leq 1 + \epsilon$, azaz az i. játékos kezdőtőkéje $(1 + \epsilon)$-szorosára nő, ami, ha bekövetkezik $O(\frac{n}{\epsilon} log(\frac{m}{\epsilon min_i \sum_j b_{ij}}))$-szor, akkor $\exists i w_i(p) < \frac{m}{\epsilon}$ nem teljesülhet már, így az algoritmus leáll.

$\square$

2.4. 3. eset: Arrow-Debreu-modell, Leontief hasznossági függvényekkel

Ebben az esetben egy egyensúlyi ár kiszámítása olyan nehézségű, mint egy kétszemélyes nem nullösszegű játék esetében egy Nash-egyensúly megtalálása, azaz NP-teljes probléma. Megmutatjuk, hogy egy nem nullösszegű játék Nash-egyensúly számítási feladata visszavezethető egy egyensúlyi ár számítási feladatra az Arrow-Debrue-modellben.

Legyen a kétszemélyes játékban az egyik fél hasznossági mátrixsza A, a másiké B. Ezek a mátrixok legyenek mxn-esek. Feltehető, hogy ezen mátrixok minden mezője pozitív, hiszen a játék nem változik attól, ha a mátrixok minden mezőjét megnöveljük egy konstanssal. Egy (x, y) stratégiavektor-pár akkor Nash-egyensúly, ha tetszőleges x' illetve y' vektorokra teljesül, hogy $x^T Ay \geq x'^T Ay$ és $x^T By \geq$ $\geq x^T By'$. Ezek szerint az (x, y) Nash-egyensúlyra teljesül, ha x'-nek illetve y'-nek azt a stratégiát választjuk, amikor az i. illetve j. lehetőséget választjuk 1 valószínűséggel, a többit pedig 0-val, hogy $x^T Ay \geq \sum_{j=1}^{n} a_{ij} y_j$ és $x^T By \geq$ $\geq \sum_{i=1}^{m} b_{ij} x_i$. Ha bevezetjük u és v normált vektorokat: $u_i = \frac{y_i}{x^T Ay}$, $v_i = \frac{x_i}{x^T By}$. Így a fenti egyenlőtlenségekből a következőket kapjuk minden i-re és j-re: $1 \geq$ $\geq \sum_{i=1}^{m} b_{ij} v_i$ és $1 \geq \sum_{j=1}^{n} a_{ij} u_j$. Ezekből egy-egy slack-változó bevezetésével az alábbi egyenleteket kapjuk (e a csupa 1 vektort jelöli):

$$Au + r = e \quad r \geq 0$$

$$B^T + t = e \quad t \geq 0$$

Ezen kívül még annak kell teljesülnie egy (x, y) Nash-egyensúlyra, hogy csak azokat a lehetőségeket választják pozitív valószínűséggel a felek, amelyekre igaz, hogy ha adottnak tekintjük az ellenfél x illetve y stratégiaválasztását, akkor ezek a lehetőségek legjobb tiszta válaszok lennének. Azaz

$$x_i \geq 0 \Rightarrow \sum_{j=1}^{n} a_{ij} y_j = x^T A y$$

$$y_i \geq 0 \Rightarrow \sum_{j=1}^{n} a_{ij} x_j = x^T B y$$

Azaz az r és t slack-változók csak ott lehetnek pozitívak, ahol az x illetve y stratégiavektorok az adott lehetőséget 0 valószínűséggel választják.

Legyen $H = \begin{pmatrix} 0 & A \\ B^T & 0 \end{pmatrix}$

és $z = (r, t)$

Így a Nash-egyensúly keresési probléma átírható egy LCP problémává:

$$Hw + z = e$$

$$w^T z = 0$$

Legyen most ez a H mátrix egy Arrow-Debreu piac Leontief hasznossági mátrixa. Azaz az i. játékos hasznossága az l_i vásárlás esetén: $u_i(l_i) = min_j \frac{l_{ij}}{h_{ij}}$.

16. Állítás. *Ha p ennek a mátrixnak egyensúlyi ára, és a vásárlók ezen ár mellett maximalizált hasznosságai: $\beta_1, ..., \beta_{n+m}$, akkor $\beta = \beta_1, ..., \beta_{n+m}$ megoldása a fenti LCP problémának.*

Bizonyítás. $Hw + z = e \Leftrightarrow \sum_j h_{ij}\beta_j \leq 1 \forall i \in (1, n+m)$, ez pedig nyilván teljesül, hiszen a kereslet nem haladhatja meg a kínálatot. $w^T z = 0$ pedig azért teljesül, mert az egyensúlyi ár esetében csak akkor lehet $\beta_i > 0$, ha $\sum_{j=1}^{n} h_{ij}\beta_j = 1$. □

Hivatkozások

[1] Arrow K. J. and G. Debreu "The Existence of an Equilibrium for a Competitive Economy" Econometrica, vol. XXII, 265-90 (1954)

[2] B. Codenotti : The complexity of equlibria : Hardness results for economies via a correspondence with games, Theoretical Computer Science doi :10.1016/j,tcs.2008.08.007 (2008)

[3] Nikhil. R. Devanur, Christos Papadimitrou, Amin Saberi, Vijay V. Vazirani : Market equilibrium via a primal-dual-type algorithm. The 43rd Annual IEEE Symposium on Foundations of Computer Science (2002)

[4] Eisenberg, E., Gale, D. : Consensus of subjective probabilities : the pari-mutuel method. Ann. Mathe. Statist. 30, 165-168 (1959)

[5] Illés Tibor : Megoldások mérete (2010)

[6] K. Jain, M. Mahdian, A. Saberi, Approximating market equilibrium, in : Workshop on Approximation Algorithms for Combinatorial Optimization, APPROX 2003 (2003)

[7] Y. Ye, A Path to the Arrow-Debreu Competitive Market Equilibrium. Stanford University working paper (2004)