STAHL JÁNOS

Algoritmus poliéderjátékok megoldására

Bevezetés

Az alábbiakban tárgyalandó algoritmus tulajdonképpen rendkívül kézenfekvő módja poliéderjátékok megoldásának: noha a [10]-ben bevezetett lineáris programozási feladat explicite általában nem ismert, megoldása — amennyiben egyáltalán létezik — többnyire lényegesen kisebb méretű lineáris programozási feladatok egy sorozatának megoldásával meghatározható.

Az 1. fejezetben ezt az eljárást tárgyaljuk, 2-ben pedig két olyan speciális poliéderjátékra történő alkalmazásával foglalkozunk röviden, amelyek ekvivalensek a lineáris programozási feladat megoldásával: így különféle dekompozíciós eljáráshoz jutunk.

Az alkalmazott jelölésmód: félkövér kisbetűvel oszlopvektort, felül vesszővel sorvektort, nagybetűvel mátrixot jelölünk. Az elemek valós számok, a méreteket, dimenziókat külön nem hangsúlyozzuk.

1.

Az $(\mathcal{K}_1, \mathcal{K}_2, C)$ hármast, ahol $\mathcal{K}_1 = \{\mathbf{x}' | \mathbf{x}' A_1 \geq \mathbf{a}_1'\} \neq \emptyset$ és $\mathcal{K}_2 = \{\mathbf{y} | A_2\mathbf{y} \leq \mathbf{a}_2\} \neq \emptyset$ konvex poliéderek, poliéderjátéknak nevezzük.

Legyen $v_1 = \sup\limits_{\mathcal{K}_2} \inf\limits_{\mathcal{K}_1} \mathbf{x}' C\mathbf{y}$ és $v_2 = \inf\limits_{\mathcal{K}_1} \sup\limits_{\mathcal{K}_2} \mathbf{x}' C\mathbf{y}$. Mint az belátható ([10], [11]), mindig fennáll a következő négy eset valamelyike:

$$(1.1.)\quad -\infty < v_1 = v_2 < +\infty$$

$$(1.2.)\quad -\infty = v_1 = v_2$$

$$(1.3.)\quad v_1 = v_2 = +\infty$$

$$(1.4.)\quad v_1 = -\infty,\ v_2 = +\infty$$

(1.1.) esetén v_1 és v_2 definíciójában inf és sup helyett min és max írható és van olyan $\overline{\mathbf{x}}' \in \mathcal{K}_1$, $\overline{\mathbf{y}} \in \mathcal{K}_2$ pár, amelyre $\min\limits_{\mathcal{K}_1} \mathbf{x}' C\overline{\mathbf{y}} = \overline{\mathbf{x}}' C\overline{\mathbf{y}} = \max\limits_{\mathcal{K}_2} \overline{\mathbf{x}}' C\mathbf{y}$. (Egyébként a következő algoritmus, illetve annak verifikálása is lényegében kiadja a fentieket).

A továbbiakban definiáljuk az (1.5.)–(1.8.) eljárást, amely véges számú lépésben eldönti, hogy egy adott poliéderjáték esetén (1.1.)-gyel vagy valamelyik további „kivételes" esettel van-e dolgunk, illetve (1.1.) teljesülése esetén megadja a fenti $\overline{\mathbf{x}}' \in \mathcal{K}_1$ és $\overline{\mathbf{y}} \in \mathcal{K}_2$ párt.

(1.5.) Legyen $\widetilde{x}_1$ tetszőleges $\mathcal{K}_1^4$-beli extremális elem, ahol $\mathcal{K}_1$ kanonikus felbontásában ([4], [8]) $\mathcal{K}_1^4$ jelöli a korlátos, $\mathcal{K}_1^<$ pedig a kúp komponenst. Minthogy $\mathcal{K}_1 \neq \emptyset$, ilyen van.

(1.6.) Tegyük fel, $\widetilde{\mathbf{x}}_1' \ldots \widetilde{\mathbf{x}}_{k-k'}' \in \mathcal{K}_1^4$ és $\widetilde{\widetilde{\mathbf{x}}}_1', \ldots \widetilde{\widetilde{\mathbf{x}}}_{k'}' \in \mathcal{K}_1^<$ extremális elemek már ismertek.

Tekintsük a következő lineáris programozási feladatot:

P. 1.
$$\left.\begin{array}{c} -\tilde{\mathbf{x}}_j' C\mathbf{y} + v \leq 0 \\ j = 1, 2, \dots k - k' \\ -\tilde{\tilde{\mathbf{x}}}_j' C\mathbf{y} \leq 0 \\ j = 1, 2, \dots k' \\ A_2\mathbf{y} \leq a_2. \end{array}\right| v \to \max.$$

Ha ezen feladatnak nincs lehetséges megoldása, az eljárás végetér, az (1.2.) vagy (1.4.) esettel van dolgunk.

Ha a feladat nem korlátos, legyen $\mathbf{y}_k \in \mathcal{R}_2^{\leq}$ olyan extremális elem, melyre $\tilde{\mathbf{x}}_j' C\mathbf{y}_k > 0$ és $\tilde{\tilde{\mathbf{x}}}_j' C\mathbf{y}_k \geq 0$ minden P.1.-ben szereplő $\tilde{\mathbf{x}}_j$ és $\tilde{\tilde{\mathbf{x}}}_j'$-re és $v_1^{(k)} = +\infty$.

Ha P.1.-nek létezik optimális megoldása, legyen $\mathbf{y}_k \in \mathcal{R}_2^A$ egy extremális optimális megoldás és jelöljük $v_1^{(k)}$-val az optimumértéket.

(1.7.) Tekintsük a következő lineáris programozási feladatot:

P. 2. $\mathbf{x}' A_1 \geq \mathbf{a}_1' \}$ $\mathbf{x}' C\mathbf{y}_k \to \min$

Ha ezen feladat nem korlátos, legyen $v_2^{(k)} = -\infty$ és $\mathbf{x}_{k+1}' \in \mathcal{R}_1^{\leq}$ olyan extremális elem, hogy $\mathbf{x}_{k+1}' C\mathbf{y}_k < 0$. Ellenkező esetben legyen $\mathbf{x}_{k+1}' \in \mathcal{R}_1^A$ P.2. egy optimális extremális megoldása és $v_2^{(k)}$ az optimumérték.

(1.8.) Ha $v_1^{(k)} = +\infty$ és $v_2^{(k)} > 0$, az eljárás végetér, az (1.3.) vagy (1.4.) esettel van dolgunk.

Ha $v_1^{(k)} = v_2^{(k)}$ véges érték, az eljárás végetér, az (1.1.) esettel van dolgunk.

Más esetben legyen $\tilde{\mathbf{x}}_{k-k'+1}' = \mathbf{x}_{k+1}'$ ha $\mathbf{x}_{k+1}' \in \mathcal{R}_1^A$, $\tilde{\tilde{\mathbf{x}}}_{k'+1}' = \mathbf{x}_{k+1}'$, ha $\mathbf{x}_{k+1}' \in \mathcal{R}_1^{\leq}$. Folytassuk az eljárást (1.6.)-tól k helyett $k+1$-gyel.

A fenti állítások a következőképpen láthatók be:

ad (1.6.) Ha a P.1, feladat nem oldható meg, akkor tetszőleges $\mathbf{y} \in \mathcal{R}_2$-re valamely $1 \leq j \leq k - k'$-vel $\tilde{\tilde{\mathbf{x}}}_j' A\mathbf{y} < 0$. Minthogy $\tilde{\tilde{\mathbf{x}}}_j \in \mathcal{R}_1^{\leq}$, $\inf\limits_{\mathcal{R}_1} \mathbf{x}' C\mathbf{y} = -\infty$ minden $\mathbf{y} \in \mathcal{R}_2$-re, tehát $v_1 = -\infty$.

ad (1.8.) Ha $v_2^{(k)} > 0$, akkor tetszőleges $\mathbf{x}' \in \mathcal{R}_1$-re $\mathbf{x}' C\mathbf{y}_k > 0$. Minthogy $v_1^{(k)} = +\infty$ folytán $\mathbf{y}_k \in \mathcal{R}_2^{\leq}$, azért $\sup\limits_{\mathcal{R}_2} \mathbf{x}' C\mathbf{y} = +\infty$ tetszőleges $\mathbf{x}' \in \mathcal{R}_1$-re, így $v_2 = +\infty$.

A $v_1^{(k)} = v_2^{(k)}$ véges érték esetben vezessük be a következő jelöléseket. Legyen $\bar{\mathbf{x}}' = \mathbf{t}' X$, ahol X az $\tilde{\mathbf{x}}_1', \dots \tilde{\mathbf{x}}_{k-k'}'$ $\tilde{\tilde{\mathbf{x}}}_1' \dots \tilde{\tilde{\mathbf{x}}}_{k'}'$ sorvektorokból alkotott mátrix, $\mathbf{t}'$ a P.1. feladat egy duális optimális megoldásának első k komponenséből alkotott vektor és $\bar{\mathbf{y}} = \mathbf{y}_k$. Ekkor $\bar{\mathbf{x}}' \in \mathcal{R}_1$ és P.2. alapján $\min\limits_{\mathcal{R}_1} \mathbf{x}' C\bar{\mathbf{y}} = v_2^{(k)}$. Másrészt P.1.-re a kiegészítő eltérések tételét ([8]) alkalmazva, $\bar{\mathbf{x}}' C\bar{\mathbf{y}} = v_1^{(k)}$, és tetszőleges $y \in \mathcal{R}_2$-re $\bar{\mathbf{x}}' C\mathbf{y} \leq v_1^{(k)}$, igy $\max\limits_{\mathcal{R}_2} \bar{\mathbf{x}}' C\mathbf{y} = v_1^{(k)}$. Tehát a most definiált $\bar{\mathbf{x}}'$ és $\bar{\mathbf{y}}$ valóban szolgáltatja az (1.1.)-gyel kapcsolatban említett elempárt.

Minthogy P.2. extremális megoldásainak száma véges és egy extremális megoldás újbóli megjelenése olyan esetre vezet, mikor az eljárás befejeződik, az (1.5.)—(1.8.) eljárás véges számú lépésben végetér.

Ezzel kapcsolatban egyrészt azt kell meggondolnunk, hogy egy P.1.-ben szerepelt $\tilde{\tilde{\mathbf{x}}}' \in \mathcal{A}_1^{<}$ nem adódhat újra P.2. megoldásakor, hiszen ekkor $\tilde{\tilde{\mathbf{x}}}' C\mathbf{y}_k < 0$-nak kellene teljesülnie, míg P.1. szerint $\tilde{\tilde{\mathbf{x}}} C\mathbf{y}_k \geq 0$. Másrészt, ha egy P.1.-ben már szerepelt $\tilde{\mathbf{x}} \in \mathcal{A}_1^{\Delta}$ adódna újra, akkor P.2. alapján $v_2^{(k)} = \tilde{\mathbf{x}}' C\mathbf{y}_k \leq \tilde{\mathbf{x}}'_j C\mathbf{y}_k$, $j = 1, 2, \ldots k-k'$-ra. Amennyiben most $\mathbf{y}_k \in \mathcal{A}_2^{\Delta}$ akkor P.1. alapján ebből $v_1^{(k)} = \tilde{\tilde{\mathbf{x}}}' C\mathbf{y}_k$ is adódik, azaz $v_1^{(k)} = v_2^{(k)}$ véges érték. Ha viszont $\mathbf{y}_k \in \mathcal{A}_2^{<}$, akkor (1.6.) szerint $v_1^{(k)} = +\infty$ és $\tilde{\mathbf{x}}' C\mathbf{y}_k > 0$, azaz $v_1^{(k)} = +\infty$ és $v_2^{(k)} > 0$ teljesül. (1.8.) szerint mindkét esetben befejeződik az eljárás.

Ezen rész lezárásaként a következőket jegyezzük meg:

(1.9) Nyilván $v_1^{(k)} \geq v_1^{(k+1)} \ldots = v_1 = v_2^{(k)}$.

(1.10.) A k. képésben szereplő P.1. feladatból elhagyhatók a $k+1$. lépésre mindazon feltételek, amelyek nem egyenlőségként teljesültek a k. lépésbeli optimális megoldásnál. (Ha a k. lépésben ilyen egyáltalán volt).

(1.11.) Ha megengedjük, hogy $\mathcal{A}_1 = \emptyset$ vagy $\mathcal{A}_2 = \emptyset$, akkor az üres számhalmaz supremumára, illetve infimumára vonatkozó szokásos definíció alapján meg kell engednünk azt a lehetőséget is, hogy $v_1 = +\infty$ és $v_2 = -\infty$. Ekkor (1.5.)-ben egy lineáris programozási feladat megoldásával $\mathbf{x}_1$ meghatározható, vagy az adódik, hogy $\mathcal{A}_1 = \emptyset$, amivel a most bevezetett esettel az eljárás véget is ér. Ez utóbbi eset állhat fenn akkor is, ha $k = 1$ esetén P.1.-nek nincs lehetséges megoldása.

(1.12.) Az eljárással kapcsolatban elmondottakon nem változtat, ha $v_1^{(k)} < +\infty$ esetén 1.7. helyett $\mathbf{x}'_{k+1}$-t $\mathcal{A}_1$ olyan extremális elemeként definiáljuk, amelyre $\mathbf{x}'_{k+1} C\mathbf{y}_k \leq v_1^{(k)}$. Ekkor $v_2^{(k)} = \mathbf{x}'_{k+1} C\mathbf{y}_k$ szükséges a korábbi definíció helyett.

(1.13.) A k. lépésben szereplő P.1. feladat bővíthető a $k+1$. lépésre – az $\mathbf{x}'_{k+1}$-hoz tartozó feltételen kívül – tetszőleges $\mathbf{x}' \in \mathcal{A}_1$-nek megfelelő $-\mathbf{x}' C\mathbf{y} + v \leq 0$ feltétellel, illetve tetszőleges sok ilyen feltétellel.

(1.14.) Ha $\mathcal{A}_1$ és $\mathcal{A}_2$ korlátos poliéderek, azaz $\mathcal{A}_1^{<} = \emptyset$ és $\mathcal{A}_2^{<} = \emptyset$, akkor az algoritmusból ezen utóbbi halmazokkal kapcsolatos lehetőségek elhagyhatók és amennyiben nem üres poliéderekről van szó, mindig az (1.1.) esettel van dolgunk.

2.

Tekintsük a $\sup\{\mathbf{c}'\mathbf{x} \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$ lineáris programozási feladatot.

(2.1.) Ha $A = [A_1 \ldots A_n]$, $\mathbf{c}' = [\mathbf{c}'_1, \ldots \mathbf{c}'_n]$ a feladat paramétereinek egy particiója és $\mathbf{x} = [\mathbf{x}_1, \ldots \mathbf{x}_n]$ a változók megfelelő particiója, akkor a fenti feladat nyilván ekvivalens a következővel ([7]):

$$\sup\Big\{ \sum_i \mathbf{c}'_i \mathbf{x}_i \,\Big|\, \sum_i A_i \mathbf{x}_i = \mathbf{b},\ \mathbf{x}_i \geq 0 \Big\} =$$

$$= \sup\Big\{ \sum_i \sup\{\mathbf{c}'_i \mathbf{x}_i \mid \mathbf{A}_i \mathbf{x}_i = \mathbf{b}_i,\ \mathbf{x}_i \geq 0\} \,\Big|\, \sum_i \mathbf{b}_i = \mathbf{b} \Big\} =$$

$$= \sup\Big\{ \sum_i \inf\{\mathbf{p}'_i \mathbf{b}_i \mid \mathbf{p}'_i A_i \geq \mathbf{c}'_i\} \,\Big|\, \sum_i \mathbf{b}_i = \mathbf{b} \Big\} = \sup_{\mathcal{B}} \inf_{\mathcal{P}} \hat{\mathbf{p}}' \hat{\mathbf{b}}$$

ahol

$$\mathscr{B}=\left\{\hat{\mathbf{b}}=\begin{pmatrix}\vdots\\ \mathbf{b}_i\\ \vdots\end{pmatrix}\middle|\ \sum_i \mathbf{b}_i=\mathbf{b}\right\}$$

és

$$\mathscr{P}=\{\hat{\mathbf{p}}'=(\dots\mathbf{p}_i'\dots)\mid \forall_i:\mathbf{p}_i'A_i=\mathbf{c}_i'\}.$$

Hallgatólag feltettük, hogy $+\infty+(-\infty)=-\infty$.

$\mathscr{B}$ nyilván nem üres és mint egyszerűen belátható $\mathscr{P}\neq\emptyset$ teljesül, ha egyetlen i-re sincs olyan $\mathbf{x}_i$, hogy $A_i\,\mathbf{x}_i=\mathbf{0}$ és $\mathbf{c}_i'\mathbf{x}_i>0$, ami gyengébb, mintha az eredeti lineáris programozási feladat korlátos voltát tennénk fel. (Egyébként lásd az (1.11.) alatti megjegyzést).

Ha az (1.5.)–(1.8.) algoritmust alkalmazzuk ezen poliéderjáték megoldására, akkor az (1.7.) lépésben a min $\{\mathbf{p}_i'\mathbf{b}_i|\mathbf{p}_i'A_i=\geq\mathbf{c}_1'\}$ feladatokat, vagy max $\{\mathbf{c}_i'\mathbf{x}_i|A_i\mathbf{x}_i=\mathbf{b}_i,\ \mathbf{x}_i\geq\mathbf{0}\}$ duálisaikat kell megoldanunk, míg az (1.6.) lépésben szereplő P.2. feladat $\max\{v|\mathbf{p}_i'\mathbf{b}\geq v,\dots\overline{\overline{\mathbf{p}}}_1'\mathbf{b}\geq 0,\dots,\mathbf{b}\in\mathscr{B}\}$. (A duálisra való áttérés lehetőségét most és a következőkben mint egy konkrét esetben alkalmazható eszközt tekintjük).

Ha az eredeti feladat a Dantzig—Wolfe-féle dekompozíciós eljárás ([3]) tárgyalásánál „szokásos" — de mindenképpen érdekes és fontos — max $\left\{\sum_i \mathbf{c}_i'\mathbf{x}_i|\sum_i A_i\mathbf{x}_i=\mathbf{b}_i,\ D_i\mathbf{x}_i=\mathbf{d}_i,\ \mathbf{x}_i\geq\mathbf{0}\right\}$ feladat és az előbbi partíció az $\mathbf{x}_i$-knek megfelelően történik, a fenti $\mathscr{B}$ definíciójához nyilván elegendő csak a „közös" feltételekhez tartozó jobboldalt felbontani. A „szektorfeladatok" mérete az (1.5.)–(1.8.) algoritmusnál nagyobb, mint a Dantzig—Wolfe algoritmusnál, a különbség a közös feltételek száma. A központi feladat is nagyobb ennél az algoritmusnál és általában egy feltétellel még növekszik is minden lépésben. [Lásd még az (1.10.) és (1.12.) megjegyzéseket].

Ezzel kapcsolatban — számítástechnikai tapasztalatok vagy további elmélet híján — csak annyit, hogy ha a közös feltételek száma nem túl nagy a szektorfeladat(ok) feltételei számához képest, a Dantzig—Wolfe algoritmushoz képest nagyobb szektorfeladatok kezelése nem jelent lényeges különbséget és többnyire csak ilyen esetekben alkalmazták sikerrel a Dantzig—Wolfe algoritmust is ([1]). Ide kívánkozik még az is, hogy több esetben az A_i mátrixok egy részében viszonylag alacsony a nemzérus sorok aránya.

Vegyük majd észre még, hogy a (2.2.) alatti megközelítésben a szektorfeladatok méreteivel nincs ilyen „baj".

(2.2.) Gyakorlatban egy másmilyen visszavezetés után alkalmaztuk az algoritmust lineáris programozási feladat megoldására ([5]). Az adódó algoritmus lényegében a Benders-féle eljárás ([2]) lineáris esetben.

Hasonló „mélységben" végezve ezen algoritmusnak a [7]-ben javasolt eljárással történő összevetését azt mondhatjuk, hogy a bonyolultabb központi feladat kezelése az a többletráfordítás, amellyel eljárásunk esetében a végességet elérjük.

Ha a lineáris programozási feladat paramétereinek $A=[A_1A_2]$, $\mathbf{c}'=[\mathbf{c}_1',\mathbf{c}_2']$, illetve $\mathbf{x}$ megfelelő $[\mathbf{x}_1,\ \mathbf{x}_2]$ partícióját tekintjük, akkor

$$\sup\{\mathbf{c}_1'\mathbf{x}_1 + \mathbf{c}_2'\mathbf{x}_2 | A_1\mathbf{x}_1 + A_2\mathbf{x}_2 = \mathbf{b}, \mathbf{x}_1, \mathbf{x}_2 \geqq \mathbf{0}\} =$$
$$= \sup\{\mathbf{c}_1'\mathbf{x}_1 + \sup\{\mathbf{c}_2'\mathbf{x}_2 | A_2\mathbf{x}_2 = \mathbf{b} - A_1\mathbf{x}_1, \mathbf{x}_2 \geqq 0\} | \mathbf{x}_1 \geqq 0\} =$$
$$= \sup\{\mathbf{c}_1'\mathbf{x}_1 + \inf\{\mathbf{p}'(\mathbf{b} - A_1\mathbf{x}_1) | \mathbf{p}'A_2 \geqq \mathbf{c}_2'\} | \mathbf{x}_1 \geqq 0\} =$$
$$= \sup_{\mathscr{X}} \inf_{\mathscr{P}} \hat{\mathbf{p}}' C \hat{\mathbf{x}}$$

$$\mathscr{X} = \{\hat{\mathbf{x}} = (\mathbf{x}_1, \mathbf{0}) | \mathbf{x}_1 \geqq 0\},$$
$$\mathscr{P} = \{\hat{\mathbf{p}} = (\mathbf{p}', \mathbf{0}) | \mathbf{p}'A_2 \geqq \mathbf{c}_2'\}$$

és

$$C = \begin{bmatrix} 0 & \mathbf{c}_1' \\ \mathbf{b} & -A_1 \end{bmatrix}.$$

Ezen megközelítést nyilván akkor célszerű alkalmazni, ha a $\mathbf{p}'A_2 \geqq \mathbf{c}_2'$ feltételrendszer vagy duálisa jól kezelhető (pl. [5] esetén ez egy szállítási feladat volt), vagy

$$A_2 = \begin{bmatrix} A_{21} & & 0 \\ & A_{22} & \\ 0 & & \ddots \end{bmatrix},$$

mikoris az (1.7.)-beli P.2. feladat megoldása több kisebb méretű feladat megoldását jelenti.

Könnyű megmutatni, hogy az így adódó eljárás — és így a Benders algoritmus is — olymódon is származtatható, hogy a $\sup\{\mathbf{c}_1'\mathbf{x}_1 + \mathbf{c}_2'\mathbf{x}_2 | A_1\mathbf{x}_1 + A_2\mathbf{x}_2 = \mathbf{b}; \mathbf{x}_1, \mathbf{x}_2 \geqq 0\}$ feladat helyett duálisát, az $\inf\{\mathbf{p}'\mathbf{b} | \mathbf{p}'A_1 \geqq \mathbf{c}_1', \mathbf{p}'A_2 \geqq \mathbf{c}_2'\}$ feladatot oldjuk meg az eredeti Dantzig—Wolfe algoritmus egy némileg módosított változatával, mely alkalmazásnál a $\mathbf{p}'A_2 \geqq \mathbf{c}_2'$ feladat a szektorfeladat. A módosítás abban áll, hogy a központi feladat helyett mindig annak duálisát tekintjük és a központi feladatban megőrizzük azon változókat is, melyek elhagyták a bázist. Ennek következtében a Dantzig—Wolfe algoritmus központi feladatának egy bázistranszformációja helyett egy lineáris programozási feladat megoldása szerepel.

[1] szerint a Dantzig—Wolfe eljárásnál a központi feladat „régi" változóinak megőrzése, illetve minél több változó figyelembevétele a számoláshoz szükséges idő csökkenéséhez vezet. Valószínűleg ebben van az [5] -beli számítás „sikerének titka" is. Mindenesetre jó lenne ezt pontosan tudni.

(Beérkezett: 1970. szeptember 1.)

IRODALOM

[1] BEALE, E. L. M.: Mathematical programming in practice. Pitman, 1969.
[2] BENDERS, J. F.: Partitioning Procedures for Solving Mixed-Variable Programming Problems. Numerische Mathematik, 1962. 4. sz. 238—252. o.
[3] DANTZIG, G. B.—WOLFE, P.: The Decomposition Algorithm for Linear Programs. Econometrica, 1961. 29. sz. 767—778. o.

[4] Goldman, A. I.: Resolution and Separation Theorems for Polyhedral Convex Sets. Ld. [6].
[5] Kovács, Á.—Stahl, J.: Dekompozíciós eljárás egy széntermelést és elosztást optimalizáló modell megoldására. Szigma, 1970. 2. sz. 97—107. o.
[6] Kuhn, H. W.—Tucker, A. W. (eds): Linear Inequalities and Related Systems. Princeton, 1956. Princeton University Press.
[7] Lipták, T.: Two-level Programming. Ld. [9]
[8] Prékopa, A.: Lineáris programozás. Bolyai János Matematikai Társulat kiadványa, 1968.
[9] Prékopa, A.: (ed): Colloquium on Applications of Mathematics to Economics. Budapest, 1965. Akadémiai Kiadó.
[10] Stahl, J.: An Existence Theorem for Polyhedral Games. Ld. [9]
[11] Wolfe, P.: Determinateness of Polyhedral Games. Ld. [6]

AN ALGORITHM FOR THE SOLUTION OF POLYHEDRAL GAMES

The paper deals with the solution of the so-called polyhedral game being a sort of generalization of the notion of the two-person 0-sum matrix game. The algorithm suggested in the first part of the paper is a series of solutions of linear programming problems. The second part is a certain inverse of the foregoing: the application of the algorithm to the solution of polyhedral games which are equivalent to the linear programming problem and some statements in connection with decomposition methods having derived from the above algorithm.

АЛГОРИТМ ДЛЯ РЕШЕНИЯ ПОЛИЭДЭРНЫХ ИГР

Статья занимается решением так называемых полиэдэрных игр, происходящих как некое обобщение понятия матричной игры двух лиц с нулевой суммой. Предлагаемый в первой части статьи алгоритм является серией решения задач линейного программирования. Вторая часть является определенным поворачиванием первого: применение алгоритма для решения таких полиэдэрных игр, которые эквивалентны задачей линейного программирования и несколько установлений в связи с таким образом появляющимися декомпозиционными методами.