

Egy elágaztatási módszer diszkrét változókat is tartalmazó nem-lineáris programozási feladatok megoldására

Bevezetés

Az elágaztatási módszerek a diszkrét változókra vonatkozó optimálási problémák megoldásánál indokolt népszerűségnek örvendenek. Ugyanakkor még a legutóbbi publikációkban sem oldották meg; sőt nem is szenteltek kellő figyelmet annak a problémának, amely az elágaztatási módszerek keretében definiálásra kerülő irányított, ültetett fa (a továbbiakban csak fának nevezzük) méreteinek ellenőrizhetetlen növekedéséből származik (Hervé, 1968; Roy, Benayoun és Tergny, 1970). Ez a nehézség a számológépen csak külső tárolók révén hidalható át. Ugyanakkor számos, néhány száz (egész vagy nem egész értékű) változót és hasonló nagyságrendű feltételt (amelyekhez a pozitivitási vagy általánosabban a változók korlátossági feltételei csatlakoznak) tartalmazó probléma mégis teljesen megoldható lenne a központi tárolóban, ha biztosítható lenne, hogy a fa hasznos része ne növekedjék bizonyos határon túl. Maradjon mondjuk N és $2N$ között, ahol N a diszkrét értékeket felvevő x_j -k száma (ami egyébként nincs korlátozva).

A fent jelzett nehézség egy sereg ún. „heurisztikus” módszer kibontakozását segítette elő. A jelen összefüggésben a „heurisztikus”-t olyan módszerek elnevezésére használjuk, amelyeknél semmi bizonyosat sem tudunk arról, hogy az optimális megoldás elérhető-e, de amelyek mégis az optimális megoldást — pontatlanul szólva — „nagy valószínűséggel megközelítő” pontokat szolgáltatnak. Az időzjelekbe tett szavak a legtöbb esetben egyébként nincsenek is definiálva. Ezeknek a heurisztikus módszereknek van bizonyos jelentőségük, ugyanis meg kell elégednünk egy probléma „majdnem megoldásával”, ha a megoldása lehetetlen vagy nagyon költséges. Idézzünk három ilyen heurisztikus eljárást:

1. oldjuk meg a felvetett problémának megfelelő „folytonos” optimálási problémát, azután az így kapott megoldásban „a lehető legjobban” kerekítsük azokat a változókat, amelyeknek egész értékűeknek kellett lenniök (ebben a tanulmányban a „folytonos” gyakran „nem diszkrét”-et jelent); ezt követően a nem egész értékű változókat tekintve optimalizáljuk azt a problémát, amelyet az előbbi problémából úgy nyerünk, hogy rögzítjük abban az eredeti probléma egész értékű változóira előzően nyert értékeket (ez a rögzített egész értékekhez tartozó ún. *redukált* probléma);

2. kezdjük a fenti heurisztikus eljárással, azután változtassuk meg egy egész értékű változónak az értékét mégpedig úgy, hogy növeljük vagy csökkentjük 1-gyel és oldjuk meg az ennek megfelelő redukált problémát; ismételjük meg ezt az eljárást ugyanezzel az egész értékű változóval mindaddig, amíg a minimalizálandó $\varphi(x)$ függvény csökken, ezután változtassuk meg egy másik egész értékű változó értékét; állítsuk meg az eljárást egy olyan ponton, hogy

a külön választott egész értékű változók ± 1 -gyel való megváltoztatásai ne tegyék többé lehetővé a $\varphi(x)$ csökkenését. Ez azzal a kockázattal jár, hogy elhanyagoljuk azokat a szimultán megváltoztatásokat, amelyek csökkenthetnék a $\varphi(x)$ -et, márpedig, ha N egész értékű változó van, akkor ezeknek a megváltoztatásoknak a száma — ami 2^N — nagyon nagy lehet;

3. úgy alkalmazzuk az elágaztatási eljárást, hogy a fa tárolandó csúcsainak a számát N' -ben korlátozzuk. Valahányszor a csúcsok száma $N' + 1$ -re nő, a „legrosszabb” csúcsot töröljük; kockázta az, hogy ez utóbbi egy optimális megoldáshoz vezető ágon van, amelyet ezáltal biztosan nem tudunk majd megtalálni.

Egyéb, sokkal jobban kidolgozott heurisztikus eljárásokat is használnak. Ezeket nem ismertetjük, mert kifejtésük hosszadalmas lenne. A jól használható heurisztikus eljárásoknak közös tulajdonságuk, hogy elég gyorsan adnak egy olyan diszkrét megoldást, ami még ha nem is optimális, a „gyakorlatban elfogadható”.

Az egzakt módszerekre vonatkozó eddigi tapasztalatok (ellentétben a heurisztikus módszerekkel) különösen a diszkrét lineáris problémák területéről valók. Azonban egy általános kísérleti kód megadásához feltétlenül túl kell lépni a lineáris problémákon; ez tökéletesen megvalósítható: Colville (1968) munkája mintegy harminc nem-lineáris, folytonos optimálási módszerre vonatkozó gépi összehasonlító tesztet tartalmaz (az említett problémákkal foglalkoznak még: Abadie (1967), (1969), (1970); Abadie és Carpentier (1966 és 1969), Abadie és Guigon (1969)). Egyébként e tanulmány tárgyától eltérő elágaztatási módszerek is léteznek, a nem-lineáris programozási feladatok megoldásának általános módszereivel vannak párosítva (Carpentier (1969)), sőt vannak speciális problémák számára létesített kódok is (Carpentier, Cassapoglou és mások (1968)).

Az az elágaztatási módszer, amelyet most leírunk (és amelyet mi a „Bounded Branch and Bound” után BBB-nek kereszteltünk el) egy rokon változata annak a módszernek, amit Land és Doig (1960) javasoltak részlegesen diszkrét változókat is tartalmazó lineáris programozási feladat megoldására. A módszer beletartozik a Hervé-féle axiómatikába, amely magában foglalja az összes ismert elágaztatási eljárást, beleértve a SEP (B. Roy) eljárást is. Rendelkezik mindazokkal az alapvető tulajdonságokkal, amelyeknek, úgy hisszük, érzékelteztük szükségességét és megvalósíthatóságát:

1. a fa tárolandó csúcsainak a száma elenyésző ($2N - 2$, sőt néha N);
2. alkalmazható, ha a célfüggvény és a feltételek nem lineárisak (tetszőleges meglévő nem lineáris programozási gépi kódot használhat szubrutinként);
3. rendelkezik a heurisztikus eljárásoknak azzal az — ámbár nem precízen definiált — alapvető sajátosságával, hogy elég gyorsan ad egy „elfogadható” diszkrét megoldást.

A következőkben jelbeszédet (sőt jeleket) fogunk használni. A tanulmánynak célja, hogy azok az olvasók is megértsék, akik nem járatosak az elágaztatási eljárásokban. Módszerünket egy teljes részletességgel tárgyalt numerikus példával illusztráljuk: nem azért választottuk ezt a példát, hogy — az iterációk számát minimálisra csökkentve — csillogtassuk módszerünk használhatóságát; a látszat ellenére egy nehéz példáról van szó. A legtöbb heurisztikus módszer nem ad rá ésszerűen jó megoldást, a mi algoritmusunkhoz szükséges iterációk száma pedig abnormális módon meg van benne emelve. A példa azonban olyan, hogy illusztrálja a legtöbb előforduló jelenséget.

I. A P probléma és a hozzátartozó G gráf

Tekintsük a következő részlegesen diszkrét változókra vonatkozó programozási feladatot:

határozzuk meg a $\varphi(x)$ minimumát az alábbi feltételek mellett:

$$x \in K$$

$$x_j \in F_j, \forall j \in E$$

$$a_j \leq x_j \leq b_j, \forall j \in E',$$

ahol x az $(x_j)_{j \in J}$ vektor, $j = \{1, 2, \dots, n\}$;

K az R^n térnek egy részhalmaza;

E a J -nek egy részhalmaza; $E' = J - E$;

F_j valós számokból álló véges halmaz.

Ez a probléma megoldható — a φ -re és K -ra vonatkozó bizonyos feltevések mellett — módszerünk segítségével. A módszer bemutatása abban a speciális esetben a legegyszerűbb, amikor a következő a probléma (megjegyezzük, hogy az általános eset hasonlóképpen tárgyalható):

P : minimalizáljuk a $\varphi(x)$ -et az alábbi feltételek mellett:

$$x \in K \tag{1}$$

$$a_j \leq x_j \leq b_j, \forall j \in J \tag{2}$$

$$x_j \text{ egész, } \forall j \in E; \tag{3}$$

feltesszük, hogy az a_j és b_j egészek $\forall j \in E$. (Vagyis részlegesen egészértékű feladattal van dolgunk.)

A bizonyítások egyszerűsége kedvéért feltesszük, hogy a (2) által definiált Π paralelotop korlátos (ami a gyakorlatban sohasem okoz zavart) és feltesszük, hogy a φ függvény szigorúan kvázi-konvex a Π -n. Emléztetünk arra, hogy akkor mondjuk a φ -t szigorúan kvázi-konvexnek a Π -n, ha nincs olyan Π -hez tartozó intervallum, amelynek belső pontjában a φ eléri a maximumát. Hogy a legegyszerűbb példát idézzük, minden olyan konvex függvény szigorúan kvázi-konvex, amely nem éri el több pontban a minimumát (speciálisan így van ez a nem azonosan konstans lineáris függvények esetében). A K halmazról feltesszük, hogy konvex.

Legyen A egy tetszőleges részhalmaz $A \subset E$, és $x_A = (x_j)_{j \in A}$ az x -nek egy olyan részvektora, amelynek komponensei az x azon komponensei, melyek indexei az A -ból valók. A következőkben az $\bar{x}_A, \tilde{x}_A, \hat{x}_A$ olyan x_A vektorokat jelölnek, amelyeknek a komponensei egészek és értékeik — a (2)-nek megfelelően — vannak rögzítve.

Egy adott $S = (A, \bar{x}_A)$ párnak megfeleltetjük a következőképpen definiált $P(S)$ problémát:

$P(S)$: minimalizáljuk a $\varphi(x)$ -et az alábbi feltételek mellett:

$$x \in K \tag{1}$$

$$a \leq x \leq b \tag{2}$$

$$x_j = \bar{x}_j, \forall j \in A \tag{4}$$

Legyen $\hat{x}(S)$ a $P(S)$ probléma megoldása (amelyről az egyszerűség kedvéért feltesszük, hogy egyértelmű) és legyen $\hat{\varphi}_S$ a φ minimális értéke a $P(S)$ problémában, ahol a szokásos megegyezés szerint $\varphi_S = +\infty$, ha az (1), (2), (4) feltételek ellentmondóak. Az előzőkből következik, hogy $\bar{x}_j = \hat{x}_j(S)$, $\forall j \in A$. Legyen $E(S)$ azon $j \in E$ indexek halmaza, amelyekre $\hat{x}_j(S)$ egész.

Tekintsük a következőképp definiált G irányított gráfot:

1. G csúcsai a fent definiált különböző $S = (A, \bar{x}_A)$ párok.
2. Minden S csúcshoz a $\hat{\varphi}_S$ értéke van hozzárendelve.
3. Amikor A üres, a megfelelő csúcsot a G S_0 gyökerének nevezzük.
4. Ha $E(S) = E$, akkor azt mondjuk, hogy az $S = (A, \bar{x}_A)$ csúcs végpont.
5. Minden $S = (A, \bar{x}_A)$ csúcsnak megfeleltetjük az $\hat{\varphi}_S$ -el jelölt emeletét, ami az A elemeinek a száma (az emelet Roy (1969) terminológiájában a rendnek felel meg).

6. Tegyük fel, hogy az $S = (A, \bar{x}_A)$ nem végpont és legyen $\beta \in E(S) - A$.

Definiáljuk az S -nek β -ra vonatkozó egy „magasabb emeletű” utódát a következőképp:

$$T = (B, \bar{x}_B),$$

ahol:

$$B = A \cup \{\beta\}$$

$$\bar{x}_j = \bar{x}_j, \forall j \in A$$

$$\bar{x}_\beta = [\hat{x}_\beta(A)] \text{ vagy } [\hat{x}_\beta(A)] + 1$$

(a $[\lambda]$ a szokásnak megfelelően a λ valós szám egész része). Az S -nek a β -ra vonatkozólag két utóda létezik.

7. Általánosabban, ha adott az $S = (A, \bar{x}_A)$ csúcs és az nem végpont, akkor a $(t_S + 1)$ -ik emelet csúcsait definiáljuk a következőképp:

$$\beta \in E(S) - A$$

$$B = A \cup \{\beta\}$$

$$\bar{x}_j = \bar{x}_j, \forall j \in A$$

$$\bar{x}_\beta = [\hat{x}_\beta(A)] + \gamma$$

$$T = (B, \bar{x}_B),$$

ahol γ nulla vagy tetszőleges előjelű egész szám úgy, hogy a (2) feltétel teljesüljön.

A $\gamma = 0, -1, -2, \dots$ -nek megfelelő csúcsokat egymás utáni irányított élekkel (a továbbiakban csak élt mondunk) összekötjük, ezek alkotják a balszárnyat, amelynek kezdőpontja a $\gamma = 0$ -nak megfelelő csúcs. Hasonlóképp egymás utáni élekkel összekötjük a $\gamma = 1, 2, \dots$ -nek megfelelő csúcsokat, ezek alkotják a jobbszárnyat, amelynek kezdőpontja a $\gamma = 1$ -nek megfelelő csúcs. A két szárnyat ellentétesnek és az S -ből keletkezőnek mondjuk.

Egy csúcsnak szárnyában való eltávolodása $|\gamma|$ a balszárnyra és $\gamma - 1$ a jobbszárnyra.

Példa: Minimalizáljuk a

$$2(x_1^2 + x_2^2 + x_3^2) - (x_1 + x_2 + 9x_3)$$

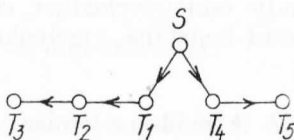
kifejezést az alábbi feltételek mellett:

$$-2 \leq x_j \leq 2, \forall j \in J = \{1, 2, 3\}$$

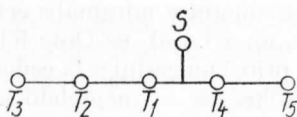
$$x_j \text{ egész, } \forall j \in E = J$$

Ha az $S = (A, \bar{x}_A)$ csúcsot az $A = \{1\}$, $\bar{x}_1 = 2$ -vel definiáljuk, akkor a $P(A)$ probléma megoldása $\hat{x}(A) = (2; 1/4; 2)$.

A G -nek az $S = (A, \bar{x}_A)$ -ból, ennek a $\beta = 2$ -re vonatkozó 2-ik emeleti utódaiból és a megfelelő két ellentétes szárnyból álló parciális részgráfja az 1a ábrán látható (az 1a ábrát mindig az 1b ábra szerint fogjuk rajzolni, ahol az élék irányát mutató nyilak odaértendőek).



1a ábra



1b ábra

A feltevések miatt:

$$T_i = (B_i, \bar{x}_{B_i}^i),$$

ahol:

$$B_i = \{1, 2\}, \quad i = 1, 2, \dots, 5$$

$$(\bar{x}_1^i, \bar{x}_2^i) = \begin{cases} (2, 1 - i), & i = 1, 2, 3 \\ (2, i - 3), & i = 4, 5 \end{cases}$$

T_1 és T_4 a két ellentétes szárny kezdőpontjai. T_i -nek az eltávolodása:

$$i - 1, \quad i = 1, 2, 3$$

$$i - 4, \quad i = 4, 5.$$

A G gráfban az „utód” fogalmán kívül (a T akkor utód, ha az ST egy él; az utód ugyanazt jelenti, mint a „követő” (Roy (1969) terminológiájában) még létezik a „leszármazott” (Roy, 1969) fogalma is. A T az S -nek egy leszármazottja, ha létezik az S -et a T -vel összekötő irányított út (a későbbiekben csak utat mondunk). Nyilvánvaló, hogy egy utód az egy olyan leszármazott, amelyet egyetlen élből alkotott út hoz létre.

1. Tétel. Ha T az S -nek egy leszármazottja a G -ben, akkor $\hat{\varphi}_T \geq \hat{\varphi}_S$.

Bizonyítás. Vegyük figyelembe a G definíciója 6. részében szereplő fogalmakat és tegyük fel, hogy a T az S -ből közvetlenül keletkező jobbszárnnyhoz tartozik. Az (\hat{x}_S, \hat{x}_T) szakasz metszi az $x_\beta = [\hat{x}_\beta(A)] + \gamma$ hipersíkot az x^γ pontokban mégpedig úgy, hogy a $\varphi(x^\gamma)$ növekszik γ -val. Ebből mindenestre következik, hogy $\hat{\varphi}_T \geq \hat{\varphi}_S$, s ezért a bizonyítást befejeztük, ha $\gamma = 1$; ha $\gamma > 1$, akkor tekintve a $(\gamma - 1)$ -nek megfelelő T' -t adódik: $\hat{\varphi}_{T'} \leq \hat{\varphi}_T$ azaz a $\hat{\varphi}_T$ növekszik, amikor az eltávolodás növekszik a jobbszárnnyon. Hasonló okoskodás alkalmazható a balszárnnyra is.

2. Tétel. A G gráfban nincs körút.

Bizonyítás. Egy csúcs rendje csak növekedhet, vagy konstans marad. Csak azokban a szárnnyakban marad konstans, amelyekben nincs körút.

2. A módszer leírása

Látni fogjuk, hogy a Land- és Doig-féle módszer (Land és Doig, 1960; Balinski, 1967; Hervé, 1968) abban áll, hogy szerkesztünk az előző G gráfban egy olyan utat, amely az S_0 gyökeret a minimális értékű végponttal köti össze. Az alkalmazások szempontjából a Land- és Doig-féle konstrukciónak van egy lényeges hibája: nem lehet a priori megadni a fa csúcsainak a számára — a közbeeső lépések tárolásához szükséges — megfelelő korlátot. Egyetlen S csúcs tárolása — néhány egységtől eltekintve — egyenértékű $\hat{x}(S)$ nyilvántartásával, ami n rekeszt igényel.

Nyilvánvaló, hogy megelégedhetünk a G végpontjainak a megvizsgálásával és a kapott csúcsok közül elég a legkisebb értékűt megtartani (ha $E = J$, akkor ez arra vezet, hogy a $\varphi(x)$ értékét a $p = p_1 \times p_2 \times \dots \times p_n$ pontokra kell kiszámítani, ahol $p_j = [b_j - a_j]$, pl. $p = 10^n$, ha $p_j = 10, \forall j$). Egyébként léteznek olyan esetek, amikor ez a legjobb eljárás (pl. ha $E = \{1\}$, $a_1 = 0$, $b_1 = 1$, akkor a Land- és Doig-féle eljárás, illetve a következőkben leírt, három optimalizációra vezet; a végpontok teljes vizsgálata e három közül csak kettőre).

A továbbiakban bemutatunk egy olyan iteratív módszert, amely a G -ben egy fát szerkeszt (Berge, 1958; Hervé, 1968; Roy, 1969); ebből bennünket az érdekel, hogy a szerkesztett fa parciális részgráfja lesz G -nek. Tartalmazza az S_0 gyökeret és az S_0 kivételével minden csúcs egy és csak egy élnek a végpontja lesz. A gráfnak minden pontjához hozzárendelünk egy bizonyos számú jelet (l. az illusztráló példát).

Minden iteráció kezdetén a fa egy bizonyos állapotban van és van egy becslésünk a φ P -ben való minimális értékére. Ezt a becslést $\hat{\varphi}$ -nek nevezzük ($\hat{\varphi} = +\infty$ az első iteráció kezdetén, kivéve, ha ismeretes egy az (1), (2), (3) feltételnek eleget tevő megoldás: ekkor a $\hat{\varphi}$ értékének az ismert megoldásnak megfelelő φ értéket vesszük).

Vizsgáljunk egy olyan $S = (A, \bar{x}_A)$ csúcsot, amely megoldása a $P(S)$ problémának, azaz:

vagy ellentmondóak a $P(S)$ (1), (2), (4) feltételei és S -nek a $\hat{\varphi}_S = +\infty$ értéket adjuk;

vagy $\hat{x}(S)$ a $P(S)$ megoldása és S -nek a $\hat{\varphi}_S = p(\hat{x}(S))$ értéket adjuk.

Egy S csúcsot elutasítunk, ha $\hat{\varphi}_S \geq \hat{\varphi}$, vagy ha az S minden leszármazottjának az értéke legalább $\hat{\varphi}_S$. Az 1. tétel miatt egy elutasított csúcs minden leszármazottja elutasított lesz. Az elutasított csúcsokat nem tároljuk a tárlóban (l. még később az Sz. 6 szabályt).

A csúcsok megvizsgálása után minden csúcs vagy elfogadott, vagy elutasított lesz. Mindannyiszor elfogadunk egy csúcsot, amikor az nem elutasított. Ez az elfogadás ideiglenes, mivel a $\hat{\varphi}$ iterációról iterációra csökken, ami azt eredményezi, hogy egy bizonyos iterációnál elfogadott csúcs egy későbbi iterációnál elutasított lehet (maga az elutasítás végleges, mert nem lehet az eredeti P probléma optimális megoldása egyetlen olyan végpont sem, amely egy elutasított csúcs leszármazottja). Egy elfogadott csúcsot mindig tárolunk a tárolóban, egészen az esetleges elutasításig.

A következő jelbeszédet fogjuk alkalmazni:

\circ t a t -ik emelet (ideiglenesen) elfogadott csúcsa;

\times t a t -ik emelet (az összes leszármazottaival együtt véglegesen) elutasított csúcsa;

$\begin{array}{c} \circ \\ | \\ \times \end{array}$ t a t -ik emelet olyan (ideiglenesen) elfogadott csúcsa, amelynek a t -nél nagyobb sorszámú emeleteken levő leszármazottjai elutasítottak;

$\times \text{---} \circ$ t a t -ik emelet olyan (ideiglenesen) elfogadott csúcsa, amelynek a — szárnyában (itt a bal) levő — nála nagyobb eltávolodású leszármazottjai elutasítottak;

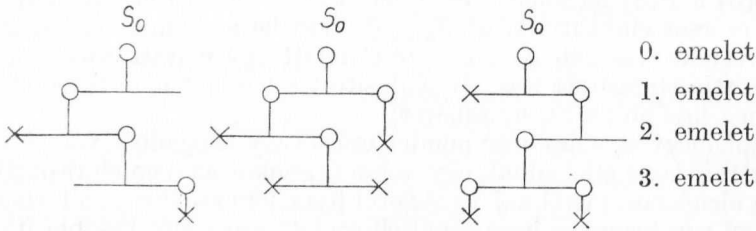
$\circ \text{---} \times$ t jelentése hasonló (jobb szárny a bal helyett);

$\begin{array}{c} S \\ | \\ \circ \\ | \\ T \text{---} \circ \end{array}$ t a t -ik emelet (ideiglenesen) elfogadott S csúcsa, valamint a $(t + 1)$ -ik emelet ugyancsak ideiglenesen elfogadott T csúcsa; a T csúcs az S egy olyan leszármazottja a G -ben, amely az S -ből keletkező balszárnyhoz tartozik (a T nem feltétlen lesz az S utóda a G -ben); az ellentétes szárny egyetlen csúcsát sem vizsgáltuk meg;

$\begin{array}{c} S \\ | \\ \circ \\ | \\ T \text{---} \times \end{array}$ t jelentése hasonló, de az ellentétes szárny elutasított (valamint minden olyan csúcsnak a leszármazottja, amely ebben a szárnyban szerepel);

$\begin{array}{c} S \\ | \\ \circ \\ | \\ \times \text{---} \times \end{array}$ t az S csúcs (ideiglenesen) elfogadott a t -ik emeleten; a két ellentétes szárny visszautasított a $(t + 1)$ -ik emeleten.

A fa minden állapotának megfelel egy bizonyos grafikus séma. A következő példákban az emeletek száma négy:





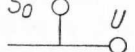
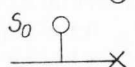
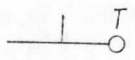
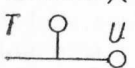
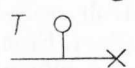
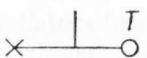
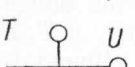
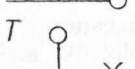
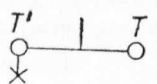
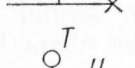
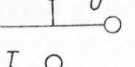
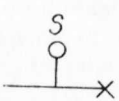
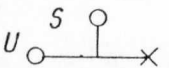
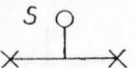
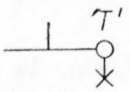
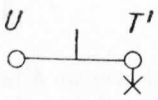
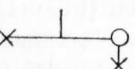
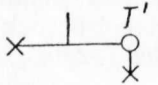
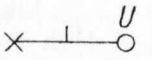
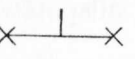
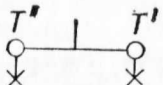
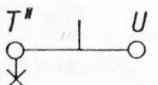
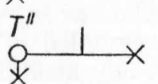
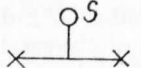

2. ábra.

A legmagasabb emelet sorszámá, amelyet t_m -mel jelölünk, legfeljebb akkora, mint az E elemeinek a száma. A séma minden elfogadott $S = (A, \bar{x}_A)$ csúcsához hozzátartozik az $\hat{\varphi}_S$ értéke és a $P(S) : \hat{x}(S)$ megoldása; ezenkívül tároló rekeszek vannak fenntartva a $\hat{\varphi}$ és az (1), (2), (3)-nak — az előző iterációk során kapott — legjobb \hat{x} megoldása számára.

A javasolt algoritmushoz egy bizonyos számú szabály tartozik. Ezeket a szabályokat (Sz. 1-től Sz. 6-ig) szukcesszíve kell alkalmazni, sematikus illusztrálásukat — a legmagasabb t_m -ik emeletre vonatkozólag a 0, 1, ..., 9 állapotokra — l. később. Minden emeleten, mint majd meglátjuk, csak két elfogadott csúcs lehet, és ha kettő van, akkor ezek a két ellentétes szárnyhoz tartoznak.

Az algoritmus szabályai.

- Sz. 0. Tekintsük a legmagasabb t_m -ik emeletet (az első iterációnál $t_m = 0$, vizsgáljuk az S_0 kezdőpontot).
- Sz. 1. Ha az S_0 -nál magasabb emelet leszármazottai mind elutasítottak, VÉGE a problémának: az (1), (2), (3) legjobb megoldására fenntartott tároló rekeszek tartalmazzák a P megoldását, ha korábban ide (1), (2), (3)-nak valamilyen megoldása bekerült. Ha nem: (1), (2) és (3) ellentmondóak.
- Sz. 2. Ha van a t_m -ik emeleten olyan elfogadott T csúcs, amelynek felsőbb emeletéről való leszármazottai között van nem elutasított, akkor vizsgáljuk a $(t_m + 1)$ -ik emeleten a T egy U utódát és fogadjuk el, vagy utasítjuk el. Az emelet t_m sorszámá egy egységgel nő (1, 2, 3, 4 állapotok).
- Sz. 3. Ellenkező esetben, amikor ez lehetséges, vizsgáljuk a t_m -et alkotó két ellentétes szárny halmazában azt a csúcsot, amelynek az eltávolodása a legkisebb és fogadjuk el vagy utasítjuk el (5, 6, 7, 8 állapotok). Ha ez az utóbbi csúcs ugyanazon szárnyban egy másiknak az utóda, akkor ez az utóbbi törlődött (7, 8 állapotok).
- Sz. 4. Ha a t_m -ik ($t_m > 0$) emelet két ellentétes szárnya elutasított, akkor elutasítjuk annak az S csúcsnak valamennyi felsőbb emeletű utódát, amelyből ez a két ellentétes szárny eredt. Az emelet t_m sorszámá egy egységgel csökkentendő (9-es állapot).
- Sz. 5. Az iteráció végén a következő áll:
 $\hat{\varphi}$ új, $\leq \hat{\varphi}$ régi.
 Ha:
 $\hat{\varphi}$ új, $< \hat{\varphi}$ régi,

Előtte				Utána		
sorszám	állapot	emelet	szabály	sorszám	állapot	emelet
0		0	Sz. 1		vége	
1		0	Sz. 2 Elf.	2		1
			Sz. 2 Elut.	5		1
2		t_m	Sz. 2 Elf.	2		$t_m = t_m + 1$
			Sz. 2 Elut.	5		$t_m = t_m + 1$
3		t_m	Sz. 2 Elf.	2		$t_m = t_m + 1$
			Sz. 2 Elut.	5		$t_m = t_m + 1$
4		t_m	Sz. 2 Elf.	2		$t_m = t_m + 1$
			Sz. 2 Elut.	5		$t_m = t_m + 1$
5		t_m	Sz. 3 Elf.	3		$t_m = t_m$
			Sz. 3 Elut.	9		$t_m = t_m$
6		t_m	Sz. 3 Elf.	4		$t_m = t_m$
			Sz. 3 Elut.	7		$t_m = t_m$
7		t_m	Sz. 3 Elf.	3		$t_m = t_m$
			Sz. 3 Elut.	9		$t_m = t_m$
8		t_m	Sz. 3 Elf.	4		$t_m = t_m$
			Sz. 3 Elut.	7		$t_m = t_m$
9		t_m	Sz. 4	0		$t_m = t_m - 1$

tekintsük át a tárolóban tárolt valamennyi $\hat{\varphi}_S$ -t és utasítsuk el (a tároló rekeszek törlésével) azokat az S csúcsokat, amelyekre:

$$\hat{\varphi}_S \geq \hat{\varphi} \text{ új.}$$

Sz. 6. Ha $\hat{x}_j(S)$ minden $j \in E$ -re egész, akkor az S -et elutasítjuk. Ha ezenkívül $\hat{\varphi}_S < \hat{\varphi}$, akkor $\hat{x}(S)$ a P jelenlegi legjobb megoldása és ez helyettesíti az előzőt az erre a célra fenntartott tároló rekeszekben, amikor is $\hat{\varphi} := \hat{\varphi}_S$. Az ellenkező esetben, azaz ha $\hat{\varphi}_S \geq \hat{\varphi}$, akkor az előző legjobb megoldás a helyén marad.

Az Sz. 1–Sz. 4. szabályok az utolsó emelet átalakulását írják le, sematikus ábrázolásukat l. a továbbiakban. Az Sz. 5. és Sz. 6. szabályok lehetővé teszik a tárolóban tárolt csúcsok számának csökkentését, ezeket nem ábrázoljuk. Az Sz. i. Elf. (Sz. i. Elut.) azt jelenti, hogy az Sz. i. szabály alkalmazásával megvizsgált csúcs elfogadott (elutasított).

Magyarázatok. A T, T', T'' az Sz. 1-től az Sz. 4-ig terjedő szabályok alkalmazása előtt elfogadott csúcsok a t_m -ik ($t_m > 0$) legmagasabb emeleten. A vizsgálandó csúcsot — ha van ilyen U — jelöli; amikor ez elutasított, akkor nem jelöljük azért, hogy emlékeztessünk arra, hogy az elutasított csúcsokat nem tároljuk a tárolóban. A 7. és 8. állapotokban U a T' utóda; ha ebben a két állapotban az U elutasított, akkor egy közbeeső állapotot a következőképpen ábrázoltunk:





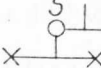

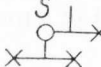

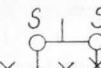

ha pedig U elfogadott, a közbeeső állapot a következő:



Ez a táblázat végső állapotaiba van közvetlenül átalakítva. Az S a t_m -ik emelet egyetlen (közvetlen) elődjét jelöli (feltéve, hogy van ilyen). A $t_m := t_m + \varepsilon$, ($\varepsilon = 0, 1$ vagy -1) azt jelenti, hogy az Sz. 1-től Sz. 4-ig terjedő szabályok alkalmazása előtti legmagasabb emelet a t_m -ik volt, alkalmazásuk utáni pedig a $(t_m + \varepsilon)$ -ik. A 9-es állapotban olyan esetek lehetnek a t_m -re, amelyek a $(t_m - 1)$ -ik emeletnek az Sz. 1-től Sz. 4-ig terjedő szabályok alkalmazása előtti állapotától függenek.

Egy állapottól egy másikba való átmenet

1. Ha a t_m -ik emelet t_m sorszámú egy egységgel nő (1, 2, 3, 4 sorszámú állapotok), akkor az x_β változónak megváltozik az értéke. Ha az $\hat{x}(T)$ -ből kiindulva, az egyetlen x_β koordináta megváltoztatásával nyert pont kielégíti a feltételeket, úgy kitűnő kiinduló megoldásul szolgál a $P(U)$ feladat iteratív megoldásához. Ha ez a pont nem megengedett; még mindig jó induló megoldást jelent, ha a $P(U)$ feladat megoldására „pönálás” módszerrel működő algoritmust használunk. Ha ezzel szemben olyan eljárással dolgozunk, amelyenél minden iterált pont megengedett kell hogy legyen; akkor egy előkészítő

Előtte				Utána		
sorszám	állapot	emelet	szabály	sorszám	állapot	emelet
9.1		0 1	Sz. 4	0		0
9.2		$t_m - 1$ t_m	Sz. 4	6		$t_m = t_m - 1$
9.3		$t_m - 1$ t_m	Sz. 4	7		$t_m = t_m - 1$
9.4		$t_m - 1$ t_m	Sz. 4	8		$t_m = t_m - 1$

fázisra van szükség (amelyben természetesen magát ezt az eljárást használjuk fel). Ha előre gondoskodunk az előkészítő fázisról, akkor egy automatikus indulási lehetőséget biztosítottunk. Ugyanakkor a jelen esetben célszerűbbnek véljük a következő módon eljárni. Kiindulunk az $\hat{x}(T)$ -ből és maximalizáljuk (vagy minimalizáljuk) az x_β -t, egészen addig, amíg eléri a kívánt értéket (hogy ezt megteheszük, (2)-t a kívánt x_β értékkel megegyező felső vagy alsó korláttal vesszük). Az x_β ezen maximalizálásának eredménye egy, a $P(T)$ korlátozásainak eleget tevő pont (ha van ilyen; ellenkező esetben azt a következtetést vonjuk le, hogy $\hat{\varphi}_T = +\infty$).

2. Ha a t_m -ik emelet t_m sorszáma változatlan marad, akkor hasonlóképpen járunk el, az S -ből (5-ös állapot) vagy a T' -ből (5, 7, 8-as állapotok) kiindulva.

3. Ha a t_m -ik emelet t_m sorszáma csökken (9-es állapot vagy az Sz. 6 szabály alkalmazása), akkor nem merül fel az a kérdés, hogy bizonyos feltételeknek eleget tevő pontot kell kapni.

3. Elméleti eredmények

Megmutatjuk, hogy az algoritmus rendelkezik a tároló rekeszek számára (4. tétel) és a konvergenciára (5. tétel) vonatkozó tételekben kimondott tulajdonságokkal. Előbb bebizonyítjuk a 3. tételt, amely éppúgy használható a két további tétel bizonyítására, mint a módszer gépi programozásának megkönnyítésére.

3. Tétel. *Az algoritmus alkalmazása során a legmagasabb emelet — minden iteráció kezdetén, a fenti 0-tól 9-ig terjedő sorszámú állapotok egyikében, az alacsonyabb emeletek pedig a 0, 1, 2, 3, 4, 6, 7, 8-as állapotok egyikében találhatók.*

Bizonyítás. Az Sz. 1—Sz. 4 szabályok alkalmazásakor az 0—9 állapotok csak 0—9 állapotot hozhatnak létre. Az alsóbb emeleteknek van egy elfogadott pontjuk; számukra az 5 és 9-es állapotok ki vannak zárva. Az Sz. 5 és Sz. 6 szabályok alkalmazása — az 1. tétel miatt — nem változtat a helyzeten.

4. Tétel. *A fa állapota, bármely iterációnál, legfeljebb $2N - 1$ elfogadott csúcsot vesz számításba; N az egész értékű változók számát (azaz E elemeinek a számát) jelenti.*

A 3. tétel miatt ez a szám legfeljebb $2N + 1$, ami az Sz. 6 szabályt alkalmazva $(2N - 1)$ -re redukálódik.

Megjegyzések

1. Tekintheszük a fa csúcsainak maximális számát $(2N - 2)$ -nek is megjegyezve, hogy az S_0 gyökér feleslegessé válik, mielőtt megkezdjük a két ellentétes szárny vizsgálatát.

2. Ezt a maximális számot elérjük, ha $t_m = N - 1$ és az 1-től $(N - 2)$ -ig terjedő sorszámú emeletek a 4-es állapotban, az $(N - 1)$ -ik emelet pedig a 4-es vagy 8-as állapotban van.

5. Tétel. *Véges számú iteráció után az algoritmus a P egy megoldását adja, vagy kiderül, hogy a P feltételei ellentmondóak.*

Bizonyítás. Az iterációk száma véges, mivel a G csúcsainak száma véges és egyiket sem érintjük kétszer.

A két lehetséges befejezés a következő:

VÉGE 1: 0 állapot és $\hat{\varphi} < +\infty$ (a probléma meg van oldva),

VÉGE 2: 0 állapot és $\hat{\varphi} = +\infty$ (a korlátozások ellentmondóak).

Az elutasított $S = (A, \bar{x}_A)$ csúcsokra vonatkozólag kétféle eset van: vagy azért lettek elutasítva, mert azoknak a $P(S)$ problémáknak felelnek meg, amelyek korlátozásai ellentmondóak (ami azt jelzi, hogy \bar{x}_A nem lehet része P egyetlen megengedett megoldásának sem) vagy azért, mert $\hat{\varphi}_S \geq \hat{\varphi} \geq \geq \min \{\varphi / (1), (2), (3)\}$. Az 1. tétel miatt ugyanez áll az S -ből leszármazó végpontokra is. Ebből következik, hogy a VÉGE 1 és VÉGE 2 a fenti jelentéssel bír.

Az az eset, amikor a P probléma konvex és differenciálható

Tegyük fel, hogy a φ függvény konvex és differenciálható a Π parallelogramon és hogy a K konvex halmaz a következő egyenlőtlenségek által van definiálva:

$$f_i(x) \leq 0, \quad i \in \{1, 2, \dots, m\},$$

ahol az f_i függvények konvexek és differenciálhatók a Π -n. Legyen $U = (B, \hat{x}_B)$ a $T = (B, \hat{x}_B)$ utóda az $S = (A, \bar{x}_A)$ -ből keletkező valamelyik szárnyban. Emlékeztetünk arra, hogy

$$B = A \cup \{\beta\}, \quad \text{ahol } \beta \in E(S) - A$$

$$\bar{x}_j = \hat{x}_j = \bar{x}_j, \quad \forall j \in A$$

$$\hat{x}_\beta = [\hat{x}_\beta(S)] + \gamma$$

$$\hat{x}_\beta = \bar{x}_\beta + \varepsilon$$

$$\varepsilon = \begin{cases} +1 & \text{a jobb szárnyra} \\ -1 & \text{a bal szárnyra} \end{cases}$$

A $P(T)$ megoldása a $\lambda_i(T)$, $\mu_k(T)$ multiplikátorokat adja, amelyekre:

$$\left(\frac{\partial \varphi}{\partial x_k} + \sum_{i=1}^m \lambda_i(T) \frac{\partial f_i}{\partial x_k} \right)_T = \mu_k(T), \quad k \in J - B$$

$$\lambda_i(T) \geq 0, \quad \lambda_i(T) f_i(\hat{x}(T)) = 0, \quad i = 1, 2, \dots, m$$

$$\mu_k(T) \geq 0, \quad \text{ha} \quad \hat{x}_k(T) = a_k$$

$$\mu_k(T) \leq 0, \quad \text{ha} \quad \hat{x}_k(T) = b_k$$

$$\mu_k(T) = 0, \quad \text{ha} \quad a_k < \hat{x}_k(T) < b_k,$$

ahol a $()_T$ jelölés azt jelenti, hogy a zárójelben levő kifejezést $x = \hat{x}(T)$ -re kell venni.

A konvexitás egyenlőtlensége:

$$\varphi(\hat{x}(U)) + \sum_{i=1}^m \lambda_i(T) f_i(\hat{x}(U)) \geq \varphi(\hat{x}(T)) + \sum_{i=1}^m \lambda_i(T) f_i(\hat{x}(T)) +$$

$$+ \sum_{k \in J-B} \left(\frac{\partial \varphi}{\partial x_k} + \sum_{i=1}^m \lambda_i(T) \frac{\partial f_i}{\partial x_k} \right)_T (\hat{x}(U) - \hat{x}(T)) +$$

$$+ \varepsilon \left(\frac{\partial \varphi}{\partial x_\beta} + \sum_{i=1}^m \lambda_i(T) \frac{\partial f_i}{\partial x_\beta} \right)_T$$

azt adja, hogy:

$$\hat{\varphi}_U \geq \hat{\varphi}_T + \varepsilon \left(\frac{\partial \varphi}{\partial x_\beta} + \sum_{i=1}^m \lambda_i(T) \frac{\partial f_i}{\partial x_\beta} \right)_T = \tilde{\varphi}_T$$

Ez az összefüggés lehetővé teszi az U elutasítását akkor, ha $\tilde{\varphi}_T \geq \hat{\varphi}$, anélkül, hogy azt előzetesen megvizsgáltuk volna. Ha nem teljesül az összefüggés, akkor a T -vel egyidejűleg a $\tilde{\varphi}_T$ -t is tárolni fogjuk: mivel a $\hat{\varphi}$ csak csökkenhet, lehetséges, hogy az U -t egy későbbi iterációnál fogjuk elutasítani.

Másrészt, a $P(S)$ probléma megoldásakor minden — a $P(S)$ probléma megoldására használt — iterációnál — a konvexitás egyenlőtlenségeiből — a $\hat{\varphi}_S$ -re egy alsó korlát adódik, legalábbis ha, amit mi felteszünk, egy olyan iteratív módszert használunk, amely a $P(S)$ feltételeinek eleget tevő x^n pontokat ad. Legyen φ^n az x^n -höz tartozó alsó korlát. Ekkor:

$$\varphi^n \uparrow \hat{\varphi}_S, \quad \text{ha} \quad v \rightarrow +\infty$$

Ha tehát $\hat{\varphi}_S > \hat{\varphi}$ (S elutasítandó), akkor a számítások ténylegesen leállíthatók, mihelyt $\varphi^n > \hat{\varphi}$.

A kétértékű változók esete

Lehetnek olyan egészértékű változók, amelyek csak két értéket vesznek fel (a továbbiakban ezeket kétértékű változóknak nevezzük): feltesszük, hogy az a két érték, amit felvehetnek a 0 és az 1. Ebben az esetben a módszer általános leírásából nyilvánvaló, hogy $a_j = 0$ és $b_j = 1$ minden kétértékű x_j változóra. Megjegyezzük, hogy minden egyes — kétértékű változónak megfelelő — szárny a kezdőpontjára redukálódik.

Az a legérdekesebb eset, amikor mind az N egész értékű változó kétértékű, mert ekkor a gépi-programozás egyszerűsödik, minthogy minden gráfcsúcsnak legfeljebb két utódja lehet. Jóval kevesebb tároló rekeszre lesz tehát szükség, mivel N bit elegendő az N db kétértékű változó tárolására, s így lehetővé válik egy sereg ilyen típusú probléma gyors megoldása. Megjegyezzük, hogy az általános eset is visszavezethető erre azáltal, hogy az egész értékű változókat felírjuk a 2-es alapú számrendszerben:

$$x_j = a_j + \sum_{k=0}^{n_j} 2^k x_{jk}, \quad 0 \leq x_{jk} \leq 1,$$

ahol az n_j -t a b_j határozza meg. Ezen átalakítás előnyei és hátrányaira nem fogunk kitérni. Mégis megjegyezzük, hogy konvex programoknál valószínűleg nem lépnek fel speciális elutasítások.

4. Illusztráló példa

A következőkben teljes egészében tárgyaljuk a következő P problémát:
 P : minimalizáljuk a

$$\varphi = (5x - 13)^2 + 15(130x - 100y - 23)^2 + 30(20x + 10y - 20z - 10)$$

függvényt az alábbi feltételek mellett:

$$0 \leq x_j \leq 6, \quad j \in J = \{1, 2, 3\} \quad (2)$$

$$x_j \text{ egész}, \quad j \in E = J \quad (3)$$

(a K konvex halmaz legyen a teljes R^3 tér). Könnyen látható, hogy a φ függvény konvex és differenciálható.

Ha a megoldandó problémák „kicsinyek”, az ésszerű megoldás, legalábbis egy gép számára (az ember esetében ez nem így van), az lenne, hogy teljes leszámolással járjon el, esetünkben ez azt jelenti, hogy $7^3 = 343$ ponttal számoljon.

A (3) elhanyagolásával kapott $P(\emptyset)$ probléma (ami annak felel meg, hogy a változókat nem egész értékűeknek tekintjük, hanem valós változóknak) megoldása a következő:

$$\hat{x}(\emptyset) = (2,6; 3,15; 3,7575), \quad \varphi(\hat{x}(\emptyset)) = 0.$$

A P probléma megoldása pedig:

$$\hat{x} = (1; 1; 1), \quad \varphi(\hat{x}) = 829.$$

A fentiekből a „legjobb kerekítéssel” nyert megoldás:

$$x^0 = (3; 3; 4), \quad \varphi(x^0) = 67609.$$

Azon nyolc pont közül, amelyeket úgy nyertünk, hogy az $\hat{x}_j(\emptyset) - t$ ($j = 1, 2, 3$) bármely lehetséges módon kerekítettük, a legjobb az:

$$x^1 = (3; 4; 4), \quad \varphi(x^1) = 21219.$$

Így tehát a „legjobb kerekítéssel” kapott megoldás (67609) egyáltalán „nem kielégítő”, sőt még „az összes lehetséges kerekítések legjobbika” (21219) is alig javasolható. Ha az algoritmust az elsőnek talált megoldásnál megállítjuk, akkor már valamivel jobb eredményt kapunk (17419 a harmadik iterációnál), a másodsorra talált egész pontnál való megállás pedig egy olyan megközelítést (934, a második iterációnál) ad, amelynek az az érdeme, hogy hasonló nagyságrendű, mint az optimális érték (829, a tizennegyedik iterációnál). Három kiegészítő iteráció szükséges, hogy meggyőződhessünk: a már korábban megkapott, optimális megoldás tényleg a P probléma optimális megoldása.

A számításokat egy tízoszlopos táblázat formájában mutatjuk be.

1. oszlop: a fában megvizsgált $T = (B, \bar{x}_B)$ csúcs megnevezése (0, 1, 2, ...); ez megegyezik az iteráció sorszámával.

2. oszlop: a B halmaz; az egyszerűség kedvéért feltesszük, hogy ez mindig a következő négy halmaz valamelyike: \emptyset , $\{1\}$, $\{1, 2\}$, $\{1, 2, 3\}$; nyilvánvaló, hogy egyéb szabályok is lehetségesek: miután az x_1 -et az \bar{x}_1 egész értékkel rögzítettük, az x_2 rögzítése helyett rögzíthetjük az x_2 és x_3 közül azt, amely a legmesszebb (vagy a legközelebb) van egy egész számhoz.

Mindez összhangban van az Sz. 2 szabályban biztosított bizonyos fajta szabadsággal.

3. oszlop: Az $i.b$ jelölés azt jelenti, hogy a vizsgált csúcs az i csúcs utóda az \bar{o} szárnyában, ami a balszárny; az $i.fb$ jelölés azt jelenti, hogy a vizsgált csúcs az i következő emeleti utóda és hogy \bar{o} egy balszárny kezdőpontja (az $i.j$ és $i.fj$ ugyanazt jelentik, csak a „bal” a „jobb”-al helyettesítendő).

4., 5., 6. oszlopok: a $P(T)$ optimális megoldásának a koordinátái az x_1, x_2, x_3 értékek; az, hogy egy x_j érték alá van húzva, azt jelenti, hogy egész értékre rögzítettük azért, mert $j \in A$; amikor ezen kívül még egy csillag is van, ez azt jelenti, hogy $j = \beta$; az $x_j = 6^\circ$ csak annak a ténynek kifejezésére szolgál, hogy $x_j = 6$ és így az $x_j \leq 6$ eredeti feltétel: hatékony; hasonlóként jelent az $x_j = 0^\circ$, zéró alsó korlát esetén.

7., 8. oszlopok: a $\hat{\varphi}_T, \tilde{\varphi}_T$, amiket már előzőleg definiáltunk; a 7. oszlopban aláhúzott számnak a legjobb egész értékű pont felel meg.

9. oszlop: a fa állapota a folyamatban levő iteráció végén (a fa gyökerének a megjelölését mellőztük; minthogy a 9-es állapot a fa legmagasabb emeletének állapota: közvetlenül az általa létrehozott 0, 6, 7, 8-as állapotok valamelyikébe megyünk).

10. oszlop: t_m , a legmagasabb emelet sorszám.

Az $\hat{x}_j(T)$, $\hat{\varphi}_T, \tilde{\varphi}_T$ értékei pontos értékek, kivéve, ha $T = 9$, ahol az értékek (+)-al vagy (-)-al vannak jelölve. Az algoritmusnál a három változóra való minimalizálás után 17 iteráció szükséges; ebből két változóra 5, egy változóra 8 és a $\varphi(x)$ számítására 4 esik.

A bemutatott módszerrel szemben gyakran ajánlható heurisztikus eljárások alkalmazása, feltéve, ha ezek egyszerűek és kevés számolásra vezetnek. Az (1), (2), (3) egy \hat{x} megoldásának ismeretében, a $P(S)$ -nek az $S = (E, \hat{x}_E)$ -vel való megoldása valóban ad a $\hat{\varphi}$ -re egy olyan értéket, amelyet elég kicsinek remélhetünk ahhoz, hogy lehetővé tegye a G elég sok csúcsának az elutasítását; ez viszont azt adja, hogy minél kisebb a $\hat{\varphi}$, annál több olyan csúcsa lesz a G -nek, amelyeket nem kell megvizsgálni. Nem mond eljárt ezeknek az a tény sem, hogy a jelen példában a legtöbb heurisztikus eljárás csődöt mond.

1	2	3	4	5	6	7	8	9	10
T	B	S	$\hat{x}(T)$			$\hat{\varphi}_T$	$\tilde{\varphi}_T$	a fa állapota	t_m
			x_1	x_2	x_3				
0	\emptyset		2.6	3.15	3.7575	0			Kezdés 0
1	1	0. fj	3.*	3.67	4.5185	4	44		1
2	1, 2	1. fj	3.	4.*	4.7	16 339	115 379		2
3	1, 2, 3	2. fj	3.	4.	5.*	17 419			3
4	1, 2, 3	2. fb	3.	4.	4.*	21 219			2
5	1, 2	1. fb	3.	3.*	4.15	67 339			1
6	1	0. fb	2.*	2.37	2.8035	9	39		1
7	1, 2	6. fb	2.	2.*	2.06	20 544			2
8	1, 2	6. fj	2.	3.*	3.15	59 544			1

Nevezzük 1. példának az előző példát, 2-nak pedig azt, amit ebből úgy kapunk, hogy elhagyjuk a (2) feltételeket. Nevezzük a fent kifejtett módszert M1-nek és M2-nek a következő módszert: a tárolt csúcsok közül minden iterációnál tekintjük azt az S csúcsot, amelyen a $\hat{\varphi}_S$ értéke a legkisebb és vizsgáljuk a három utódot (kettő a felsőbb emeletről való, egy az S szárnyában van — ez utóbbi nem létezik, ha S a gyökér —, mivel az utód a szárnyban mindig a harmadik); alkalmazzuk a 2. § elutasítási eljárásait és ha sor kerül rá, a konvexitásból levezetett elutasítási eljárást. Egyébként a Land- és Doig-féle eljárásnak a nem-lineáris programozásra való kiterjesztett változatáról van szó. A számításokat nem fogjuk részletezni. Az érdeklődő olvasó megkaphatja ezeket a szerzőtől.

Az összehasonlítás a következő pontokra vonatkozik: az első változó minimalizációinak száma (1. tábla), a második változó minimalizációinak száma (2. tábla), a tárolandó csúcsok száma (3. tábla), az egészértékű optimális megoldásokat adó iterációk száma (4. tábla, ahol a zárójelben levő számok a φ -nek a megfelelő értékeit jelentik).

1	2	3	x(T)			7	8	9	10
T	B	S	x ₁	x ₂	x ₃	$\hat{\varphi}_T$	$\tilde{\varphi}_T$	a fa állapota	t _m
9	1	1. j	4.*	4.9600+	6.	187+	6 808-		1
10	1, 2	9. fj	4.	5.*	6.	934			2
11	1, 2	9. fb	4.	4.*	5.7	141 184			1
12	1	6. b	1.*	1.07	1.0885	64	144		1
13	1, 2	12. fb	1.	1.*	1.05	799	21 799		2
14	1, 2, 3	13. fb	1.	1.	1*	829			3
15	1, 2, 3	13. fj	1.	1.	2*	14 029			2
16	1, 2	12. fj	1.	2.*	1.6	129 735			1
17	1	12. b	0.*	0.°	0.°	11 104			0

VÉGE

Megoldás: T = 14

	M1	M2	M1	M2	M1	M2
1. Példa	8	8	5	6	3	8
2. Példa	24	26	12	12	3	15
	1. tábla		2. tábla		3. tábla	

M1

M2

1. Példa

3 (17 419), 10 (934), 14 (829)

11 (11 104), 12 (934), 15 (829)

2. Példa

3 (17 419), 11 (934), 19 (829)

21 (934), 40 (829)

4. tábla

Így az adódik, legalábbis e példák láttán, hogy az M1-től az M2-höz viszonyítva a következő előnyöket várhatjuk:

1. kevesebb a minimalizációk száma (1. és 2. tábla),
2. kevesebb a tárolandó csúcs (3. tábla),
3. kevesebb időre van szükség ahhoz, hogy ugyanazokat az egész megoldásokat kapjuk (4. tábla) (ez az előny akkor áll fenn, ha elhagyjuk a befejezés előtti számításokat).

Mindezekhez hozzátehetjük, hogy a tároló rekeszek számának maximuma $2N - 2 = 4$ az M1 módszer esetében és ennél több az M2 módszer (3. tábla) esetében, ami ennél a módszernél a priori nem is korlátozható. Ez arra vezet, hogy amikor az M2 módszert alkalmazzuk, vagy fenntartandó N' tároló rekesz a központi tárolóban és kell a külső tárolóban is tárolni, ha a tárolandó csúcsok száma N -nél nagyobb (így a programozás nehézkes és rendkívül lassú lesz), vagy amikor $N' + 1$ csúcs van, akkor el kell hagyni a legrosszabbat (kockáztatva azt, hogy nem találjuk meg a felvetett probléma optimális megoldását).

5. Az N csoportos elágaztatási módszer

Néhány „nagy” problémánál előfordul, hogy a $2N - 2$ csúcs tárolása is túl sok rekeszt követel meg. Megmutatjuk, hogy hogyan kell irányítani a műveleteket ahhoz, hogy a $2N - 2$ helyett N legyen a szükséges rekeszek maximuma. Elég ha elérjük azt, hogy az összes $t < t_m$ sorszámú emeleten egy és csak egy elfogadott csúcs legyen, miközben a t_m -ik emelet 0 vagy 1 elfogadott csúcsot tartalmaz (ha $t_m = N$, ez a szám a 0 lesz). E célból elég, ha az Sz. 3 szabályt a következő Sz. 3a szabállyal helyettesítjük:

Sz. 3a: Ha a t_m -ik emelet a 6-os vagy 7-es állapotban van, akkor vizsgáljuk meg a T' utódát az ő szárnyában és fogadjuk el vagy utasítsuk el (ez a 2, 5 vagy 9 állapothoz vezet). Ha a t_m -ik emelet az 5-ös állapotban van, akkor vizsgáljuk meg az ellenkező szárny kezdőpontját és fogadjuk el vagy utasítsuk el (ez a 3 vagy a 9-es állapothoz vezet).

Most — bizonyítás nélkül — felsorolunk néhány tételt.

6. Tétel. *Az algoritmus alkalmazása során az iterációk kezdetén a legmagasabb emelet a 0, 1, 2, 3, 5, 6, 7, 9 állapotok valamelyikében van és a kevésbé magas emeletek a 0, 1, 2, 3 állapotok egyikében vannak.*

7. Tétel. *Bármely iterációnál a fa állapota legfeljebb N elfogadott csúcsot tartalmaz.*

8. Tétel. *Véges számú iteráció után a módosított algoritmus a P -nek egy megoldását adja, vagy kiderül, hogy a P feltételei ellentmondóak.*

Érdeemes megnézni, hogy mit ad a módosított algoritmus az előző paragrafus numerikus példája esetén. Pontosan ugyanazokat a csúcsokat adja, de más sorrendben. A csúcsokat ugyanúgy jelölve, mint ahogy ezt tettük, ez a sorrend a következő:

0, 1, 2, 3, 4, 5, 9, 10, 11, 6, 7, 8, 12, 13, 14, 15, 16, 17

(az aláhúzott csúcsok az egész megoldásokat jelentik).

A módosított módszer érdekes. Nincs módunk azonban arra, hogy a tároló rekeszek számának a második paragrafusban megadott vizsgálatával való összevetésén kívül egyéb szempontokkal is összevessük. Valószínű, hogy első-sorban az a hátránya ennek a módszernek, hogy túlságosan megvizsgál egy szárnyat, még mielőtt az ellenkező szárny megvizsgálását elkezdené.

Fordította: *Baróti György*

(*Beérkezett: 1970. február 5.*)

IRODALOM

- ABADIE, J. (1967), ed. *Nonlinear Programming*, North Holland Publishing Company.
- ABADIE, J. (1969), On the GRG method for nonlinear optimization and its application to control problems, EDF Note HI 141/00 (to appear, slightly modified, in Abadie, 1970).
- ABADIE, J. (1970), ed., *Integer and Nonlinear Programming*, North Holland Publishing Company.
- ABADIE, J. et J. CARPENTIER (1966), Généralisation de la méthode du Gradient Réduit de Wolfe au cas des contraintes non-linéaires, pages 1047–1953 de Proceedings of the Fourth IFORS Conference, D. B. Hertz and J. Mélése (editors), Wiley — Interscience.
- ABADIE J., and J. CARPENTIER (1969), Generalization of the Wolfe Reduced Gradient method to the case of nonlinear constraints, in: *Optimization*, R. Fletcher (editor), Academic Press.
- ABADIE, J., et J. GUIGOU (1969), Gradient Réduit Généralisé, Note EDF HI 069/02.
- BALINSKI, M. L. (1967), Some general methods in integer programming, Chap. IX of Abadie, 1967.
- BERGE, C. (1958), *Théorie des Graphes et ses applications*, Dunod.
- CARPENTIER, J. (1969), Generalized Reduced Gradient algorithm, combination with integer programming, application to an electric power system problem (mimeographed), paper presented to the NATO Advanced Study Institute on Integer and Nonlinear Programming, Bandol 8–20 June 1969.
- CARPENTIER, J., C. CASSAPOGLOU, et. al. (1968), Une méthode générale de résolution des problèmes de dispatching économique avec variables entières, tenant compte des coûts de marche à vide des groupes, paper presented at the International Conference on Operational Research & Power Systems, Athens, 1968.
- COLVILLE, A. R. (1968), A comparative study of nonlinear programming codes, IBM NYSC Report 320–2949.
- HERVE, P. (1968), Les procédures arborescentes d'optimisation, R. I. R. O., 14, 69–80.
- LAND, A. H., and A. G. DOIG (1960), An automatic method of solving discrete programming problems, *Econometrica*, 28, 497–520.
- ROY, B. (1969), *Algèbre moderne et théorie des graphes*, Dunod.
- ROY, B., R. BENAYOUN, and J. TERGNY (1970), From S.E.P. procedure to the mixed Ophélie programme, in Abadie, 1970.
- ROY, B., P. BERTIER et P. T. NGHIEM (1965), Programmes linéaires en nombres entiers et procédure SEP. *Metra*, IV, n° 3, 1965.

A BRANCH AND BOUND METHOD FOR MIXED INTEGER NON-LINEAR PROGRAMMING

The paper presents the „BBB algorithm” (Bounded Branch and Bound), and illustrates it by mean of a numerical example. This algorithm possesses, with respect to other Branch and Bound methods, the following advantages: (a) there exists an *a priori* upper bound to the number of edges of the tree to be memorized; (b) this bound is reasonably small ($2N-2$, or even N , where N is the number of integer valued variables); (c) „acceptable” solution are quicker to obtain (this advantage comes to work in the case where computations must be stopped before completion). Proofs are given under quasi-convexity hypothesis.

МЕТОД ОТВЕТВЛЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ, СОДЕРЖАЩИХ ДИСКРЕТНЫЕ ПЕРЕМЕННЫЕ

В статье представляется «Метод-ВВВ» («Bounded Branch and Bound»), иллюстрируемый на нумерическом примере. Этот метод по сравнению с другими методами ответвления имеет следующие преимущества: (а) число попадающих в память вершин управляемого дерева априори ограничено сверху; (б) ограничение является рационально малым ($2N-2$ или иногда N , где N = число целозначных переменных); (в) можно быстрее получить приемлемое решение (это является благоприятным в том случае, если расчеты следует остановить перед окончанием).

В доказательстве предполагается квази-выпуклость.