Korlátozás és szétválasztás módszere

Algoritmusok és adatszerkezetek 2.

2012. 05. 02.

A módszert

Imreh Balázs, Imreh Csanád:

Kombinatorikus optimalizálás

Novadat, Győr, 2005

egyetemi tankönyve alapján,

kisebb változtatásokkal fogjuk

bemutatni.

I. Az eljárás

II. Hátizsák feladat

III. Halmazlefedési feladat

1. **A feladat**
2. Leszámlálás
3. A szétválasztó függvény
4. A korlátozó függvény
5. A B&B eljárás

Legyen L egy véges halmaz és

z: $L \to R$ egy függvény.

Keressük:

- min $\{z(x): x \in L\}$ -t és azt az
- x értéket, melyre ez megvalósul.

Akkor z-t célfüggvénynek fogjuk nevezni.

1. A feladat
2. **Leszámlálás**
3. A szétválasztó függvény
4. A korlátozó függvény
5. A B&B eljárás

Mivel L véges, ezért a fenti feladatnak biztosan van optimális megoldása.

Legegyszerűbb megoldás: végignézzük L minden elemét és kiválasztunk egy olyant, melyre z(x) a legkisebb.

Ezt nevezik leszámlálási eljárásnak. Csakhogy L elemszáma nagyon nagy lehet!

Hogyan tudjuk a teljes leszámlálást elkerülni?

Többféleképpen, de ha jobbat nem tudunk, akkor egy módszer:

Korlátozás és szétválasztás módszere

(Branch and Bound, B&B)

- a leszámlálás egy hatékonyabb változata
- általában NP nehéz feladatoknál szokták használni

Korlátozás és szétválasztás módszere

Legfontosabb elemei:

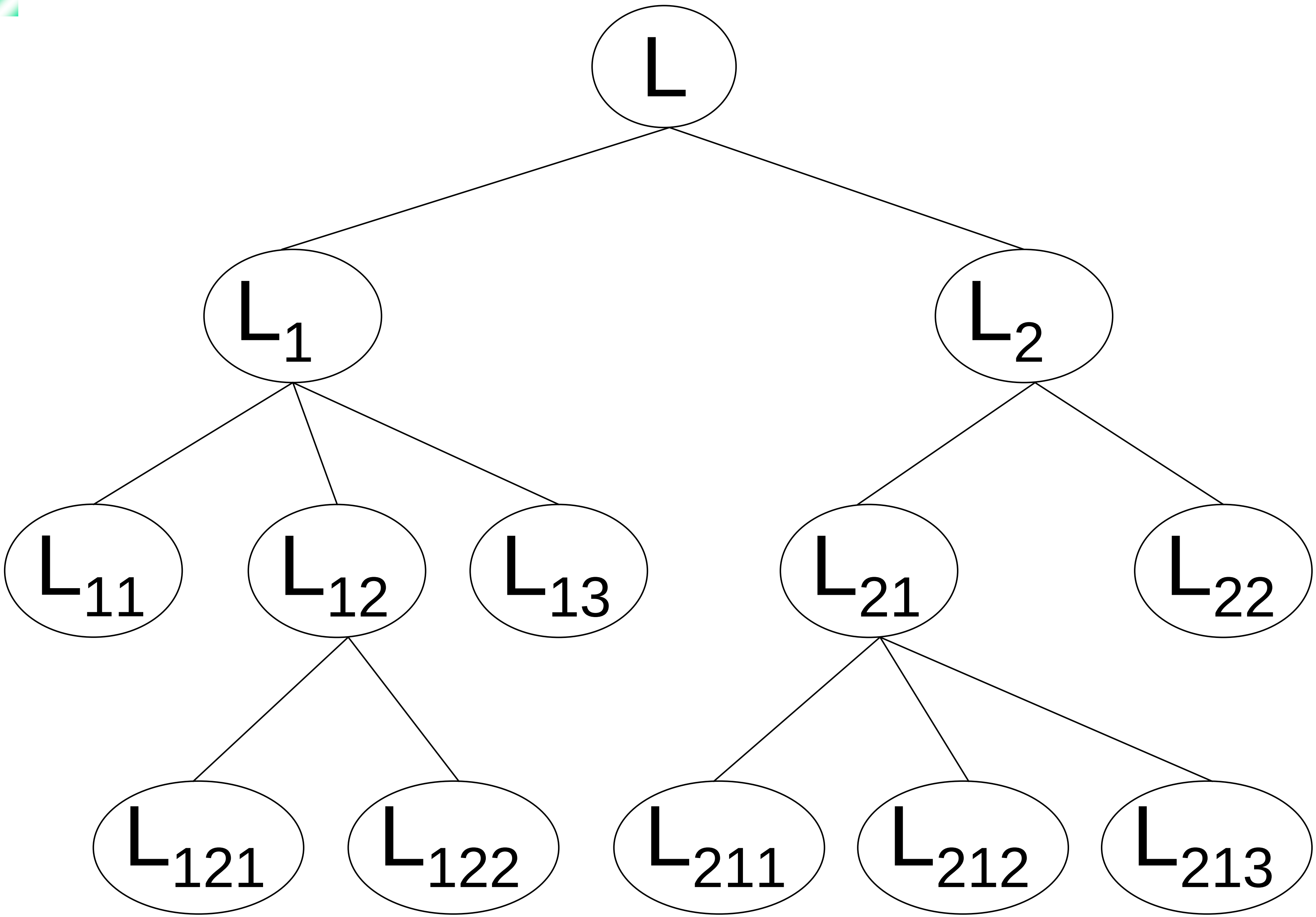
- szétválasztó függvény
- korlátozó függvény
- levélválasztási stratégia.

1. A feladat
2. Leszámlálás
3. **A szétválasztó függvény**
4. A korlátozó függvény
5. A B&B eljárás

Szétválasztó függvény

Az L halmaz tetszőleges |L'|>1 részhalmazához hozzárendeli L' egy valódi osztályozását.

A szétválasztó függvényt φ-vel fogjuk jelölni és egy fával ábrázolható.


L
L1
L2
L11
L12
L13
L21
L22
L121
L122
L211
L212
L213

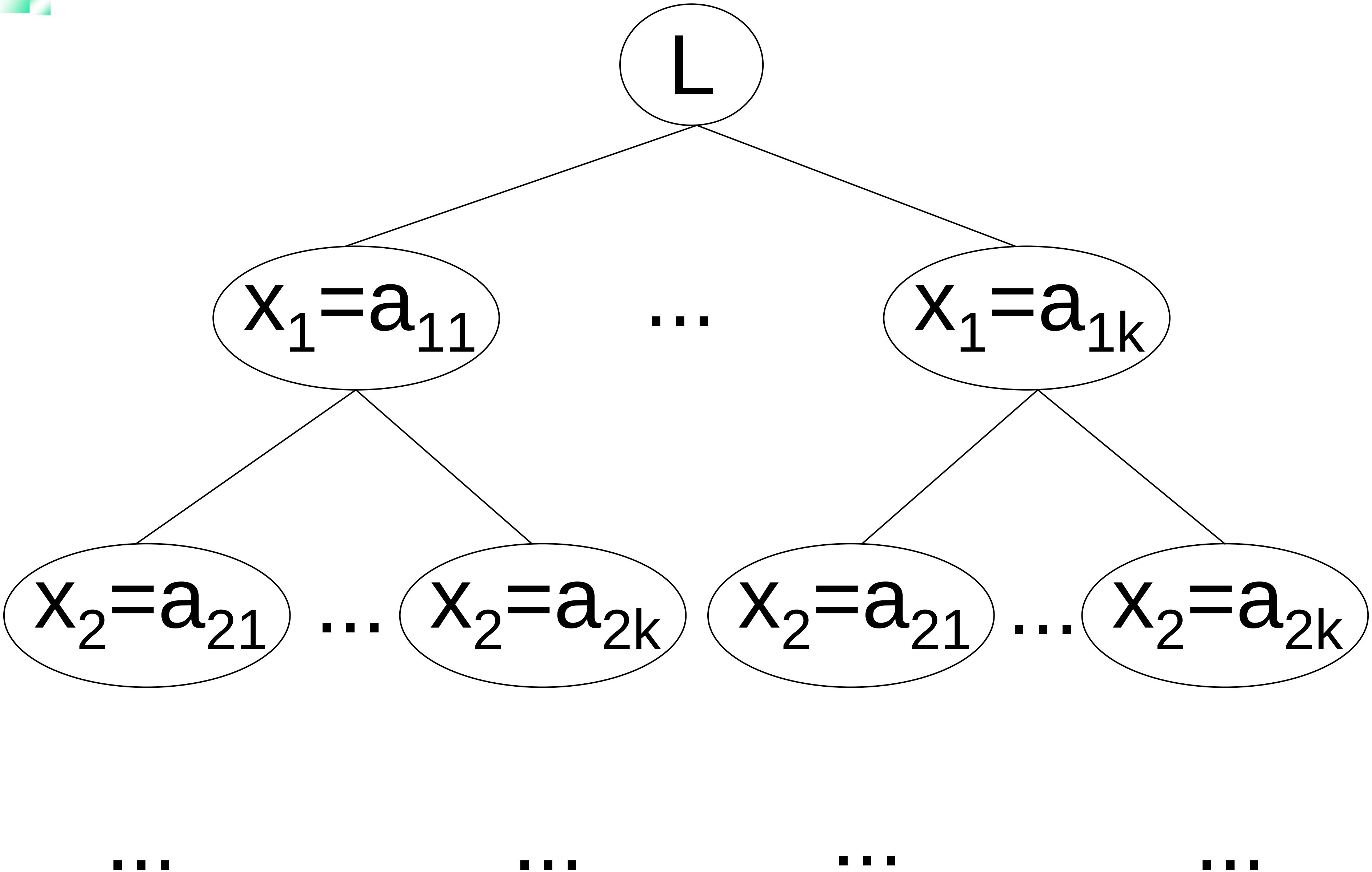
Pl. - legkevésbé önző ember (ha az önzőséget mérni tudnánk)

L = emberek; L1 = férfiak; L2 = nők stb. De mi van, ha:

- legkevésbé önző 5 fős csoport, melynek összsúlya legfeljebb 600, IQ összege legalább 500.

Sok esetben olyan a feladat, hogy L vektorok halmaza, tehát minden $x \in L$-re $x = (x_1, x_2, \dots, x_n)$.

Hogyha $x_i \in \{a_{i1}, a_{i2}, \dots, a_{ik}\}$, akkor természetesen adódik az i-dik szinten az x_i értéke szerinti szétválasztás.


L
x1=a11
...
x1=a1k
x2=a21
...
x2=a2k
x2=a21
...
x2=a2k
...
...
...
...

Jelenleg azt tudjuk tenni, hogy bejárjuk a fát (pl. mélységi bejárással).

Ennek hatékonysága növelhető lenne azzal, ha bizonyos részfák bejárását elhagyhatnánk.

Erre egy lehetőség lenne az, ha lennének olyan megkötések, melyek szerint x_1, x_2, ..., x_i adott értékei esetén x_{i+1} bizonyos értékeket nem vehet fel.

De ha a feladatban nincsenek ilyen megkötések, akkor ez nem segít.

Egy másik lehetőség lenne az, ha a fa bejárása során egy adott időpontig összegyűjtött ismeretek birtokában el tudnánk dönteni azt, hogy adott irányba már nem érdemes tovább menni.

Erre lesz jó a korlátozó függvény.

1. A feladat
2. Leszámlálás
3. A szétválasztó függvény
4. **A korlátozó függvény**
5. A B&B eljárás

Korlátozó függvény

Egy olyan g függvény, mely L tetszőleges $L' \neq \emptyset$ részhalmazához hozzárendeli a $\{z(x'): x' \in L'\}$ függvényértékek egy alsó korlátját, továbbá

- ha $L'=\{x'\}$, akkor $g(L') = z)x')$.

Hogyan használjuk a g-t?

Tételezzük fel, hogy a fa bejárása során már rendelkezünk egy x^* nem feltétlen optimális lehetséges megoldással.

Hogyan kaphatunk ilyent?

A fa bejárása során többnyire előállnak lehetséges megoldások.

Továbbá, számos feladat esetében heurisztikus eljárással elő lehet állítani viszonylag jó célfüggvény-értékkel rendelkező x* lehetséges megoldást.

A fa bejárása során az aktuális L’ csúcspontra kiszámoljuk g(L’)-t, és ha $z(x^*) \leq g(L')$, akkor mivel $g(L') \leq \min \{z(x'): x' \in L'\}$, ezért nyilvánvaló, hogy L’-ben nem fogunk jobb megoldást találni, így ezt a részfát nem érdemes bejárni.

Ilyenkor azt a részfát nem fogjuk tovább vizsgálni, azt mondjuk, hogy a fa megfelelő ágát, illetve levelét lezárjuk.

A fa többi csúcspontját élő csúcspontoknak nevezzük.

Hogyha g(L’) kiszámolása bonyolultabb, mint az L’ gyökerű részfa bejárása, akkor a B&B eljárás nem fogja növelni a hatékonyságot, ezért fontos, hogy g(L’) könnyen meghatározható legyen minden L’-re.

Természetesen

- minél jobb a rendelkezésre álló x^* lehetséges megoldás,
- annál több ágat tudunk lezárni,
- így annál kisebb fát kell bejárjunk.

Ezért a bejárás során minél hamarabb el szeretnénk jutni minél jobb lehetséges x* megoldáshoz.

Ehhez a mélységi bejárás helyett jobb lenne a fában mindig a legígéretesebb irányba menni.

A fát dinamikusan fogjuk felépíteni oly módon, hogy az adott lépésben mindig a legígéretesebb csúcspontot fogjuk megvizsgálni és szükség szerint tovább bontani.

Van azonban egy olyan probléma, hogy sok esetben nehéz megfelelő szétválasztó függvényt definiálni. Ezért általában egy $\Omega \supseteq L$ halmazból szoktunk kiindulni és arra definiáljuk a szétválasztó függvényt.

Ebben az esetben a g korlátozó függvényre vonatkozóan a következő feltételeket írjuk elő.

Az Ω tetszőleges nem üres Ω' részhalmazára:

G1. $g(\Omega')=z(x')$, ha

$$|\Omega'|=1 \text{ és } \Omega' \cap L = \{ x' \}$$

G2. $g(\Omega') = W$,

$$\text{ha } |\Omega'| = 1 \text{ és } \Omega \cap L = \emptyset$$

ahol W a gépen ábrázolható leg-nagyobb számot jelöli, melyhez konstans hozzáadásának vagy kivonásának eredményét úgy definiáljuk, hogy az ismét W.

G3. $g(\Omega') \le \min\{ z(x) : x \in \Omega' \cap L \}$,

ha $|\Omega'| > 1$ és $\Omega' \cap L \neq \emptyset$.

G4. $g(\Omega')$ tetszőleges, ha

$|\Omega'| > 1$ és $\Omega' \cap L = \emptyset$.

Ha a G4 esetet könnyen felismer-

jük, akkor a $g(\Omega') = W$ még jobb.

1. A feladat
2. Leszámlálás
3. A szétválasztó függvény
4. A korlátozó függvény
5. **A B&B eljárás**

A B&B eljárás

Az eljárás során

- z^*-al fogjuk jelölni az adott pillanatig megtalált legjobb optimum értéket, és
- x^*-al az ehhez tartozó lehetséges megoldást, azaz amire $z(x^*)=z^*$.

Előkészítő rész

Heurisztikus módszerrel határozzunk meg egy minél jobb megoldást, legyen ez x*, és z*=z(x*).

Ha nem tudunk ilyen heurisztikus módszert, akkor legyen z*=W és x* definiálatlan.

Határozzuk meg $g(\Omega)$-t. Ennek során minden esetlegesen előálló x' lehetséges megoldásra:

- ha $z(x')<z^*$, akkor legyen

$$x^* = x' \text{ és } z^* = z(x').$$

Legyen $F_0 = \{ \Omega \}$, ha $g(\Omega) < z^*$, és $F_0 = \emptyset$ különben. Legyen $r = 0$.

r-dik iteráció

Ha $F_r = \emptyset$, akkor

- x* optmális megoldás,
- z* az optimum értéke és
- vége az eljárásnak.

Egyébként folytassuk.

Valamilyen levélválasztási stratégia szerint válasszunk ki egy $\Omega' \in F_r$ elemet.

Legyen $\varphi(\Omega') = \{ \Omega'_1, \Omega'_2, \ldots, \Omega'_k \}$.

Határozzuk meg rendre a $g(\Omega'_1), g(\Omega'_2), \ldots, g(\Omega'_k)$ korlátokat.

A korlátok meghatározása során minden esetlegesen előálló x’ lehetséges megoldásra:

- ha z(x’)<z*, akkor legyen

  x*= x’ és z* = z(x’).

Legyen

$$F_{r+1} = \{ \Omega_i : \Omega_i \in (F_r \setminus \{\Omega'\}) \cup \varphi(\Omega') \ \& \ g(\Omega_i) < z^* \}$$

Növeljük r értékét 1-el.

Térjünk rá a következő iterációs lépésre.

Érdemes megfigyelni, hogy φ-nek és g-nek nem muszáj minden $\Omega' \subseteq \Omega$-re értelmezve lenni, elegendő ha a B&B fa élő levelein értelmezve vannak.

Az eljárás nem csak minimum, hanem maximum feladatokra is használható oly módon, hogy:

- W helyett –W-t,
- < helyett >-at, ≤ helyett ≥-t,

használunk, és g a halmazokhoz felső korlátot rendel.

Amikor B&B eljárással oldunk meg egy feladatot, akkor meg kell adni:

- az Ω halmazt
- a φ szétválasztó függvényt
- a g korlátozó függvényt
- a levélválasztási stratégiát.

A megoldás hatékonysága attól függ, hogy ezeket hogyan adjuk meg.

A továbbiakban gyakorlati példákon keresztül szemléltetjük ezek lehetséges megadását.

I. Az eljárás

II. Hátizsák feladat

III. Halmazlefedési feladat

IV. Hálózati folyamatok szintézise

I. Az eljárás

II. Hátizsák feladat

III. Halmazlefedési feladat

1. **A feladat**
2. Megoldás dinamikus programozással
3. A hátizsák feladat lineáris programozási relaxációja

A hátizsák feladatnak több formája van. Mi most a bináris hátizsák feladattal fogunk foglalkozni, melyről bizonyították, hogy NP nehéz, tehát nem várható hatékony eljárás a megoldására.

Adott egy hátizsák és különböző tárgyak. Legyen:

- m a tárgyak száma
- a_j a j-dik tárgy súlya (j = 1, ..., m)
- c_j a j-dik tárgy értéke (j = 1, ..., m)
- b a hátizsák súlykorlátja
- x_j = 1, ha a j-dik tárgy bekerül a zsákba, és x_j = 0 ha nem.

Feltételezzük, hogy minden tárgyból csak egy van. Ha nem, akkor a példányokat különböző tárgyaknak tekintjük.

A cél az, hogy a tárgyakból olyan rakományt állítsunk össze, melyet a hátizsák megbír, és összértéke maximális.

Formálisan:

$$a_1x_1 + a_2x_2 + \ldots + a_mx_m \leq b$$

$$x_j \in \{0,1\},\ j = 1, \ldots, m$$

$$\max\{\ c_1x_1 + c_2x_2 + \ldots + c_mx_m\ \}$$

a_j, c_j, b egészek

Tétel.

Az optimális megoldás létezését és meghatározását illetően elegendő olyan feladatok vizsgálatára szorítkozni, melyekben minden a_j, c_j együttható pozitív egész.

Bizonyítás Az előző feladatból kiindulva felépítünk egy olyan feladatot, melyben a_j, c_j pozitív egészek, és melynek optimális megoldásából közvetlenül származtatható az előző feladat optimális megoldása.

1. lépés: Minden olyan j indexre, melyre $a_j \leq 0$ és $c_j \leq 0$, helyettesítsük x_j-t $(1-x'_j)$-vel.
2. lépés: Minden olyan j-re, melyre $a_j > 0$ és $c_j \leq 0$, legyen $x_j = 0$.
3. lépés: Minden olyan j-re, melyre $a_j \leq 0$ és $c_j > 0$, legyen $x_j = 1$.

Világos, hogy ha a_j és c_j pozitív,

akkor b is pozitív kell legyen,

különben vagy nincs megoldás,

vagy az a 0 vektor.

A továbbiakban tehát feltételezzük,

hogy a_j, c_j, és b pozitív egészek.

1. A feladat
2. **Megoldás dinamikus programozással**
3. A hátizsák feladat lineáris programozási relaxációja

Jelölje f(k,r) az alábbi feladat optimális megoldását:

$$a_1x_1 + a_2x_2 + \ldots + a_kx_k \le r$$

$$x_j \in \{0,1\},\ j = 1, \ldots, k$$

$$\max\{\ c_1x_1 + c_2x_2 + \ldots + c_kx_k\ \}$$

a_j, c_j, b egészek

Akkor: $f(k,0) = 0$, $k = 1, 2, ..., m$

$f(1,r) = c_1$, ha $a_1 \le r$, $r = 0, 1, ..., b$

$f(1,r) = 0$, ha $a_1 > r$.

$f(k,r) = \max\{ f(k-1,r), c_k + f(k-1, r-a_k) \}$

Az $f(k,r)$ értékeket egy $m \times (b+1)$-es táblázatban soronként számítjuk ki.

Az optimum értékét az f(m,b) értéke adja, az optimális megoldások pedig a rekurzív egyenletet használva visszafelé határozhatók meg.

1. A feladat
2. Megoldás dinamikus programozással
3. **A hátizsák feladat lineáris programozási relaxációja**

Az eredeti feladatban egy tárgyat vagy teljesen beleteszünk a hátizsákba, vagy nem. Tekintsük most azt a változatot, melyben a tárgyakat részben is beletehetjük a hátizsákba. Ez lesz a feladat lineáris programozási relaxációja.

Formálisan:

$$a_1x_1 + a_2x_2 + \dots + a_mx_m \le b$$

$$0 \le x_j \le 1,\ j = 1, \dots, m$$

$$\max\{ c_1x_1 + c_2x_2 + \dots + c_mx_m \}$$

a_j, c_j, b pozitív egészek

Ha

- L a bináris hátizsák feladat lehetséges megoldásainak halmaza, és
- L* a relaxáció lehetséges megoldásainak halmaza,

akkor:

- $L \subseteq L^*$ és
- $\max\{z(x): x \in L\} \leq \max\{z(x): x \in L^*\}$

Sőt, a relaxáció optimumának egész része is felső korlátja a bináris hátizsák feladat optimális megoldásának.

Hogyan oldjuk meg a feladatot?

Rendezzzük át az indexeket úgy,

hogy teljesüljön a

$$c_1/a_1 \geq c_2/a_2 \geq \ldots \geq c_m/a_m$$

reláció.

Akkor teljesül G. B. Dantzig

következő tétele:

Legyen k a legnagyobb olyan egész szám, amire

$a_1 + a_2 + \ldots + a_k \leq b.$

Akkor az

$x'_j = 1, j = 1, \ldots, k$

$x'_{k+1} = (b-(a_1 + a_2 + \ldots + a_k))/a_{k+1}$

$x'_l = 0, l = k+2, \ldots, m$

egy optimális megoldása a hátizsák feladat lineáris programozási relaxációjának.

A tétel könnyen belátható.

Az is nyilvánvaló, hogy ezáltal egy hatékony megoldást adtunk.

Amennyiben a lineáris programozási relaxáció optimális megoldása egész, akkor egy bináris vektort kapunk, mely optimális megoldása a bináris hátizsák feladatnak is.

Ha a lineáris programozási relaxáció optimális megoldása nem egész, akkor abban az egyetlen nem egész komponenst 0-ra változtatva, a bináris hátizsák feladat egy lehetséges megoldását kapjuk.

4. Megoldás korlátozás és szétválasztás módszerével

a. **Az Ω halmaz**

b. A g korlátozó függvény

c. A φ szétválasztó függvény

d. A levélválasztási stratégia

e. Az eljárás

Az Ω halmaz

Legyen $\Omega = \{ (x_1, \ldots, x_m),$

$$x_j \in \{0,1\}, j = 1, \ldots, m \}.$$

Látható, hogy $L \subseteq \Omega$.

Továbbá $|\Omega| = 2^m$, tehát Ω véges.

a. Az Ω halmaz

b. A φ szétválasztó függvény

c. A g korlátozó függvény

d. A levélválasztási stratégia

e. Az eljárás

Adott halmazra a szétválasztást úgy fogjuk megvalósítani, hogy valamely még nem rögzített x_i értékét rögzítjük 0-ra, illetve 1-re, és ily módon az eredeti halmazt két részhalmazra bontjuk.

Legyen egy adott pillanatban

- $I, J \subseteq \{1, ..., m\}$, $I \cap J = \emptyset$,
- I azon i indexek halmaza,

  melyekre x_i-t 1-re rögzítettük,
- J azon j indexek halmaza,

  melyekre x_j-t 0-ra rögzítettük, és

$\Omega_{I,J} = \{ x : x \in \Omega \; \& \; x_i = 1, \text{ ha } i \in I$

$\& \; x_j = 0, \text{ ha } j \in J \}.$

Akkor: $|\Omega_{I,J}| > 1 \quad \Leftrightarrow \quad |I| + |J| < m.$

Ebben az esetben létezik

$k \in \{1, ..., m\}$ amire $k \notin I \cup J$.

Legyen $I1 = I \cup \{k\}$, $J1 = J$ és

$I2 = I, \qquad J2 = J \cup \{k\}.$

Ha több ilyen k érték is van, akkor legyen k azon legkisebb komponens indexe, amely az $\Omega_{I,J}$ által meghatározott részprobléma relaxációjának optimális megoldásában nem egész értéket kap.

Akkor legyen

$\varphi(\Omega_{I,J}) = \{\Omega_{I1,J1}, \Omega_{I2,J2}\}$.

Ez azt jelenti, hogy:

- rögzítjük x_k-t 1-re, illetve 0-ra,
- és ennek megfelelően bontjuk fel

  osztályokra az $\Omega_{I,J}$ -t.

a. Az Ω halmaz

b. A φ szétválasztó függvény

c. A g korlátozó függvény

d. A levélválasztási stratégia

e. Az eljárás

Az $\Omega_{I,J}$ halmazban rögzítettünk bizonyos indexű komponenseket 0-ra, illetve 1-re. Hogyha ezeket behelyettesítjük az eredeti feladatban, akkor annak egy (I,J)-részproblémáját kapjuk, ahol a lehetséges megoldásokba a rögzített változókat is beleértjük.

Legyen az (I,J)-részprobléma:

- lehetséges megoldásainak halmaza $L_{I,J}$,
- lineáris programozási relaxációjának optimuma $z_{I,J}$, annak egész része $\lfloor z_{I,J} \rfloor$.

Akkor: $L_{I,J} = L \cap \Omega_{I,J}$.

Továbbá $\lfloor z_{I,J} \rfloor$ felső korlátja a

$\{ z(x) : x \in L_{I,J} \}$ célfüggvény-

értékeknek. Akkor legyen

- $g(\Omega_{I,J}) = \lfloor z_{I,J} \rfloor$ ha az (I,J)-részprobléma lineáris programozási relaxációjának van lehetséges megoldása, és
- $g(\Omega_{I,J}) = -W$, ha nincs.

A lineáris programozási relaxáció megoldásánál tárgyaltak értelmében, ha az (I,J)-részprobléma lineáris programozási relaxációjának optimális megoldása egész, akkor az a részproblémának is optimális megoldása. Egyébként tudunk képezni egy lehetséges megoldást.

a. Az Ω halmaz

b. A φ szétválasztó függvény

c. A g korlátozó függvény

d. **A levélválasztási stratégia**

e. Az eljárás

Osztályozás céljából minden lépésben egy legnagyobb korláttal rendelkező levelet fogunk választani. Ha több ilyen is van, akkor választhatjuk pl. azt, amelyiket hamarabb megtaláljuk az F_r halmazban.

a. Az Ω halmaz

b. A φ szétválasztó függvény

c. A g korlátozó függvény

d. A levélválasztási stratégia

e. Az eljárás

Az eredeti feladat lineáris programozási relaxációjának megoldásából kaphatunk egy lehetséges megoldást.

Az eljárás az ismertetett módon működik, annyi különbséggel, hogy itt maximumot keresünk, ezért:

- minden esetlegesen előálló x’ lehetséges megoldásra, ha $z(x') > z^*$, akkor $x^* = x'$ és $z^* = z(x')$,

és

- $F_{r+1} = \{ \Omega_i : \Omega_i \in (F_r \setminus \{\Omega'\}) \cup \varphi(\Omega') \ \& \ g(\Omega_i) > z^* \}.$

I. Az eljárás

II. Hátizsák feladat

III. Halmazlefedési feladat

I. Az eljárás

II. Hátizsák feladat

III. Halmazlefedési feladat

IV. Hálózati folyamatok szintézise

1. **A feladat**
2. Alkalmazások
3. Redukciók
4. Redundáns és prím fedőrendszerek

Adott egy $I = \{1, ..., n\}$ halmaz és I nemüres részhalmazainak egy P_1, ..., P_m rendszere c_1, ..., c_m pozitív súlyokkal. Határozzuk meg I-nek egy minimális súlyú fedőrendszerét, ahol egy fedőrendszer súlyát a benne szereplő részhalmazok súlyainak összege adja.

$\{P_{j1}, \ldots P_{jk}\}$ fedőrendszere I-nek, ha $P_{j1} \cup \ldots \cup P_{jk} \supseteq I$.

Formálisan, tetszőleges $i \in I$, $1 \leq j \leq m$ indexekre legyen

$a_{ij} = 1$, ha $i \in P_j$ és $a_{ij} = 0$ különben,

$x_j = 1$, ha P_j eleme a fedő-rendszernek és $x_j = 0$ különben.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
$a^{(1)}$	1	0	0	1	1	0	1	1	0	0	0
$a^{(2)}$	0	1	0	1	0	0	1	0	1	0	0
$a^{(3)}$	1	1	1	0	0	1	1	1	0	1	0
$a^{(4)}$	0	1	0	1	0	0	1	0	0	0	0
$a^{(5)}$	1	0	1	0	0	0	0	0	0	0	1
$a^{(6)}$	0	0	1	1	1	1	0	0	1	0	1
$a^{(7)}$	0	1	0	0	1	1	0	1	0	1	0
c	6	4	6	3	3	4	5	2	3	4	3

$$a_{i1}x_1 + a_{i2}x_2 + \dots\ a_{im}x_m \geq 1,$$

$$i=1, \dots, n$$

$$a_{ij} \in \{0,1\},\ \ x_j \in \{0,1\},\ \ c_j > 0,$$

$$j = 1, \dots, m$$

$$c_1x_1 + c_2x_2 + c_mx_m = z \text{ -> } \min$$

A feladatnak akkor és csakis akkor van lehetséges megoldása, ha minden egyenlőtlenség baloldala tartalmaz legalább egy 0-tól különböző együtthatót.

Ezt a továbbiakban feltételezni fogjuk.

A lehetséges megoldások halmaza $L \subseteq \{0,1\}^m$, így L véges, ezért létezik optimális megoldás.

Megjegyzendő, hogy a

$$c_j > 0, j = 1,..., m$$

feltétel nem jelent tényleges megszorítást.

Könnyen belátható, hogy ha $x_j=1$-et veszünk minden olyan j indexre, melyre $c_j \leq 0$, akkor az így keletkezett csak pozitív súlyokkal rendelkező probléma megoldása az eredeti feladat optimális megoldását adja.

1. A feladat
2. **Alkalmazások**
3. Redukciók
4. Redundáns és prím fedőrendszerek

Információ kigyűjtés

Adott m fájl különböző információ-tartalmakkal. Fájlok összemásolásával egy minél rövidebb olyan új fájlt akarunk képezni, mely n féle, az eredeti fájlokban megtalálható információt tartalmaz.

Termelésszervezés

Adott n elvégzendő feladat és m munkás, melyek mindegyike a feladatok közül bizonyosokat el tud végezni. Válasszunk ki minél kevesebb (összköltségű) munkást, akik együtt az n munkát elvégzik.

Játékszervezés

Egy társaságban adott kétszemélyes játékot mindenki hajlandó játszani valakikkel, de nem mindenkivel.

Feltételezzük, hogy a hajlandóság kölcsönös.

Hogyan kell minimális számú játszmát lejátszani úgy, hogy mindenki játsszon legalább egyszer?

Itt a fedőrendszer elemei az egymással játszani hajlandó emberek-nek megfelelő kételemű halmazok.

1. A feladat
2. Alkalmazások
3. **Redukciók**
4. Redundáns és prím fedőrendszerek

Bizonyos feltételek teljesülése esetén az eredeti feladat mérete csökkenthető. Ezeket a feltételeket írják le a redukciós szabályok. Jelölje $a^{(i)}$, illetve $a_{(j)}$ az A együtthatómátrix i-dik sorát, illetve j-dik oszlopát.

R1. Ha $a^{(i)}=e_k$, ahol e_k a k-dik egységvektor, akkor bármelyik $x' \in L$ lehetséges megoldásra $x'_k=1$. Akkor minden $t \in P_k$-ra $a_{tk}x'_k=1$, ezért a t-dik feltétel teljesül, így elhagyható. Továbbá a k-adik oszlop is törölhető.

R2. Ha $a^{(r)} \leq a^{(s)}$, $r, s \in \{1, \ldots, n\}$,

akkor $\forall x' \in L$-re $a^{(r)}x' \geq 1 \Rightarrow a^{(s)}x' \geq 1$,

ezért az s-edik feltétel elhagyható.

R3. Ha $\exists J \subseteq \{1, \ldots, m\}$, $\exists k \in \{1, \ldots, m\}$

$\sum_{t \in J} a_{(t)} \geq a_{(k)}$ és $\sum_{t \in J} c_t \leq c_k$, akkor a

k-adik oszlop elhagyható.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
$a^{(1)}$	1	0	0	1	1	0	1	1	0	0	0
$a^{(2)}$	0	1	0	1	0	0	1	0	1	0	0
$a^{(3)}$	1	1	1	0	0	1	1	1	0	1	0
$a^{(4)}$	0	1	0	1	0	0	1	0	0	0	0
$a^{(5)}$	1	0	1	0	0	0	0	0	0	0	1
$a^{(6)}$	0	0	1	1	1	1	0	0	1	0	1
$a^{(7)}$	0	1	0	0	1	1	0	1	0	1	0
c	6	4	6	3	3	4	5	2	3	4	3

1. A feladat
2. Alkalmazások
3. Redukciók
4. **Redundáns és prím fedőrendszerek**

Legyen $\Phi = \{P_1, \ldots, P_k\}$ egy fedőrendszer.

P_s redundáns elem, ha $P \setminus P_s$ is fedőrendszer.

Φ redundáns fedőrendszer, ha van redundáns eleme. Ellenkező esetben Φ prím fedőrendszer.

P1. Redundáns fedőrendszerből elhagyva egy redundáns elemet kisebb súlyú fedőrendszert kapunk.

P2. A halmazfeledési probléma bármely optimális megoldása egy prím fedőrendszert ad.

P3. A $\Phi = \{P_1, \ldots, P_k\}$ fedőrendszer P_j eleme akkor és csak akkor redundáns, ha $\forall i \in P_j$-re

$a_{i1} + a_{i2} + \ldots + a_{ik} \geq 2$

teljesül. Ily módon tetszőleges fedőrendszerről és elemről eldönthető, hogy redundáns-e.

Egy $x' \in L$ által meghatározott redundáns fedőrendszerből a redundáns elemek elhagyásával kapott prím fedőrendszert jelöljük prim(x')-vel. A kapott prím fedőrendszer függ a redundáns elemek elhagyásának sorrendjétől.

5. **Lineáris programozási relaxáció**

6. Megoldás korlátozás és szétválasztás módszerével

A lineáris relaxációban az eredeti feladat

$x_j \in \{0,1\}$, $j = 1, ..., m$

feltételt helyettesítjük

$x_j \geq 0$, $j = 1, ..., m$ feltétellel.

Tehát a feladat:

$$a_{i1}x_1 + a_{i2}x_2 + \ldots\ a_{im}x_m \geq 1,$$

$$i=1, \ldots, n$$

$$a_{ij} \in \{0,1\},\quad \mathbf{x_j \geq 0},\quad c_j > 0,$$

$$j = 1, \ldots, m$$

$$c_1x_1 + c_2x_2 + c_mx_m = z \text{ -> } \min$$

Ha egy x' optimális megoldásban lenne $x'_j>1$, akkor x'_j-t 1-re változtatva is lehetséges megoldást kapnánk és a célfüggvény értéke csökkenne, ami ellentmondás lenne azzal, hogy x' optimális megoldás.

L1. A relaxáció bármely x’ optimális megoldására $0 \leq x'_j \leq 1$.

Tehát az optimális megoldás szempontjából a relaxációs feladatba $x_j \geq 0$ helyett írhatunk

$$0 \leq x'_j \leq 1\text{-et.}$$

L2. A lineáris programozási relaxáció bármely x’ optimális megoldásában minden 0-tól különböző komponens helyébe 1-et írva egy fedőrendszerhez jutunk, melyet int(x’)-vel jelölünk.

Ebből a redundáns elemek elhagyásával egy prim(int(x’)) fedőrendszerhez jutunk.

5. Lineáris programozási relaxáció

6. **Megoldás korlátozás és szétválasztás módszerével**

a. **Az Ω halmaz**

b. A g korlátozó függvény

c. A φ szétválasztó függvény

d. A levélválasztási stratégia

e. Az eljárás

Az Ω halmaz

Legyen $\Omega = \{ (x_1, \ldots, x_m),$

$$x_j \in \{0,1\},\ j = 1, \ldots, m \}.$$

Látható, hogy $L \subseteq \Omega$.

Továbbá $|\Omega| = 2^m$, tehát Ω véges.

a. Az Ω halmaz

b. A φ szétválasztó függvény

c. A g korlátozó függvény

d. A levélválasztási stratégia

e. Az eljárás

Adott halmazra a szétválasztást úgy fogjuk megvalósítani, hogy valamely még nem rögzített x_i értékét rögzítjük 0-ra, illetve 1-re, és ily módon az eredeti halmazt két részhalmazra bontjuk.

Legyen egy adott pillanatban

- $I, J \subseteq \{1, ..., m\}, I \cap J = \emptyset$,
- I azon i indexek halmaza, melyekre x_i-t 1-re rögzítettük,
- J azon j indexek halmaza, melyekre x_j-t 0-ra rögzítettük, és

$\Omega_{I,J} = \{ x : x \in \Omega \;\&\; x_i = 1, \text{ ha } i \in I$

$\qquad\qquad\qquad\qquad \& \; x_j = 0, \text{ ha } j \in J \}.$

Akkor: $|\Omega_{I,J}| > 1 \quad \Leftrightarrow \quad |I| + |J| < m.$

Ebben az esetben létezik

$k \in \{1, ..., m\}$ amire $k \notin I \cup J$.

Legyen $I1 = I \cup \{k\}$, $J1 = J$ és

$I2 = I, \qquad J2 = J \cup \{k\}.$

Ha több ilyen k érték is van, akkor legyen k azon komponens indexe, melyhez a legnagyobb értékű célfüggvény együttható tartozik. Ha több ilyen is van, akkor ezek közül válasszuk a legkisebb indexűt.

Akkor legyen

$\varphi(\Omega_{I,J}) = \{\Omega_{I1,J1}, \Omega_{I2,J2}\}.$

Ez azt jelenti, hogy:

- rögzítjük x_k-t 1-re, illetve 0-ra,
- és ennek megfelelően bontjuk fel

  osztályokra az $\Omega_{I,J}$ -t.

a. Az Ω halmaz

b. A φ szétválasztó függvény

c. A g korlátozó függvény

d. A levélválasztási stratégia

e. Az eljárás

Az $\Omega_{I,J}$ halmazban rögzítettünk bizonyos indexű komponenseket 0-ra, illetve 1-re. Hogyha ezeket behelyettesítjük az eredeti feladatban, akkor annak egy (I,J)-részproblémáját kapjuk, ahol a lehetséges megoldásokba a rögzített változókat is beleértjük.

Legyen az (I,J)-részprobléma:

- lehetséges megoldásainak halmaza $L_{I,J}$,
- lineáris programozási relaxációjának optimuma $z_{I,J}$, annak egész része $\lfloor z_{I,J} \rfloor$.

Akkor: $L_{I,J} = L \cap \Omega_{I,J}$.

Továbbá $z_{I,J}$ alsó korlátja a

$\{ z(x) : x \in L_{I,J} \}$ célfüggvény-

értékeknek. Akkor legyen

- $g(\Omega_{I,J}) = z_{I,J}$.

Ha c_j, $j = 1, \ldots, m$ egész és $z_{I,J}$ nem

egész, akkor lehet $g(\Omega_{I,J}) = \lfloor z_{I,J} \rfloor + 1$.

A lineáris programozási relaxáció megoldásánál tárgyaltak értelmé-ben, ha az (I,J)-részprobléma lineáris programozási relaxációjának optimális megoldása egész, akkor egy fedőrendszerhez jutunk.

Egyébként a prim(int(x’) fedő-
rendszer képzésével kaphatunk
egy lehetséges megoldást.

Így minden korlát meghatározá-
sánál elő tudunk állítani egy
lehetséges megoldást.

a. Az Ω halmaz

b. A φ szétválasztó függvény

c. A g korlátozó függvény

d. **A levélválasztási stratégia**

e. Az eljárás

Osztályozás céljából minden lépésben egy legkisebb korláttal rendelkező levelet fogunk választani. Ha több ilyen is van, akkor választhatjuk pl. azt, amelyiket hamarabb megtaláljuk az F_r halmazban.

a. Az Ω halmaz

b. A φ szétválasztó függvény

c. A g korlátozó függvény

d. A levélválasztási stratégia

e. Az eljárás

Egy induló lehetséges megoldás meghatározásához használjuk a következő heurisztikát.

Rendezzük növekvő sorrendbe a célfüggvény együtthatókat.

A rendezett sorozatnak megfelelően a változók addig kapnak 1-et, amíg el nem érjük, hogy minden feltétel teljesül.

Megjegyzendő, hogy hatékonyabb heurisztikák is léteznek.