Egészértékű programozás jegyzet

– az előadás és a kiadott szakirodalom alapján –

Készítette: *Schmidt Péter*

Alk. Mat., II. évf.

2010-2011

TARTALOM

EGÉSZÉRTÉKŰ PROGRAMOZÁS 3

MODELLEZÉS EGÉSZÉRTÉKŰ VÁLTOZÓKKAL 3

Vagy-vagy feltételek modellezése 3

Kapacitási korlátokra vonatkozó feltételek kölcsönös kizárása 3

Változókra vonatkozó feltételek kölcsönös kizárása 4

Ha-akkor feltétel modellezése 4

Kapacitási korlátokra vonatkozó feltételek implikációja 4

Változókra vonatkozó feltételek implikációja 4

MEGOLDÁSI MÓDSZEREK 5

Valós relaxáció: az EP-feladat LP-lazítása 5

A szétválasztás és korlátozás módszere (Branch & Bound) 5

Hátizsák feladat megoldása a korlátozás és szétválasztás módszerével 6

A korlátozás és szétválasztás alkalmazása az utazó ügynök problémára 11

Az implicit leszámolás módszere 13

Implicit leszámolás minimum feladatra 13

Hátizsák feladat megoldása implicit leszámolással 15

A vágások módszere (Gomory) 17

Hátizsák-feladat megoldása Gomory-féle vágással 17

Duális teljesen egész vágás 20

Dinamikus programozás 22

Hátizsák feladat megoldása dinamikus programozással 22

Fiktív lejátszás 25

LP-probléma modellezése kétszemélyes zérusösszegű játékkal 25

Kétszemélyes zérusösszegű játék megoldása fiktív lejátszásokkal 27

PREKONDICIONÁLÁS 29

Standard LP-probléma prekondicionálása 29

Infizibilitási teszt: kielégíthetetlen feltétel keresése 30

Redundancia teszt: elhagyható feltétel keresése 30

Változók értékének optimális rögzítése 30

Változók korlátainak erősítése 31

Példa 32

Prekondicionálás bináris változók esetén *34*
Infizibilitási teszt: kielégíthetetlen feltétel keresése 34
Redundancia teszt: elhagyható feltétel keresése 34
Változók korlátainak erősítése, változók optimális rögzítése 35
Bináris prekondicionálásra épülő megoldó algoritmus 35
Bináris prekondicionálás kiterjesztése konfliktus-gráffal *35*
Konfliktus-gráf 36
Bővítési szabály 37
Rögzítési szabály 37
Példa 37

mailto: szabos@ttk.pte.hu

Egészértékű programozás

Egészértékű LP-feladatról beszélünk, ha a változók (vagy egy részük) csak nemnegatív egész értékeket vehetnek fel. Azokat a változókat, amelyek csak a 0 vagy 1 értéket vehetik fel, bináris változóknak nevezzük. Az egészértékű programozási feladat általában nem oldható meg egy LP-feladat optimális megoldásának a kerekítésével, mert a kerekítéssel olykor nem lehetséges megoldást kapunk, olykor pedig a kapott megoldás nem optimális.

Modellezés egészértékű változókkal

Egész értékű változókkal tipikusan **fix költségeket modellezünk**, amelyek megjelenése pusztán bizonyos feltételek teljesülésétől függ.

Tételezzük fel például, hogy egy x_i termék gyártása automatikusan s_i fix költséget generál.

Ezt a tényt a következő feltételekkel modellezhetjük:

$$x_i \le M_i y_i$$
$$y_i \in \{0,1\}$$

ahol M_i az x_i változó egy felső korlátja.

Továbbá, a keresett optimum típusától (maximum vagy minimum) függően a célfüggvény a következőképpen módosul:

$$z = \sum_j c_j x_j - s_i y_i \to \max$$
$$z = \sum_j c_j x_j + s_i y_i \to \min$$

Vagy-vagy feltételek modellezése

Kapacitási korlátokra vonatkozó feltételek kölcsönös kizárása

Ha a

$$\mathbf{a}_1\mathbf{x} \le b_1$$
$$\mathbf{a}_2\mathbf{x} \le b_2$$

feltételeket szeretnénk a feladathoz adni úgy, hogy közülük pontosan az egyik teljesüljön, akkor a megoldás:

$$\mathbf{a}_1\mathbf{x} \le b_1 + My$$
$$\mathbf{a}_2\mathbf{x} \le b_2 + M(1-y)$$

ahol M az $\mathbf{a}_1\mathbf{x}$ és $\mathbf{a}_2\mathbf{x}$ függvények közös felső korlátja, és y bináris változó.

Változókra vonatkozó feltételek kölcsönös kizárása

Ha a

$$x_i = 0$$
$$x_i \geq d$$

két egymást kizáró feltételt szeretnénk a feladathoz adni, akkor a megoldás:

$$x_i \leq My$$
$$x_i \geq dy$$

ahol $M \geq d$ az x_i felső korlátja, és y bináris változó.

Ha-akkor feltétel modellezése

Kapacitási korlátokra vonatkozó feltételek implikációja

Ha a

$$\mathbf{a}_1\mathbf{x} \leq b_1$$
$$\mathbf{a}_2\mathbf{x} \leq b_2$$

feltételeket szeretnénk a feladathoz adni úgy, hogy az első implikálja a másodikat, akkor a megoldás:

$$\mathbf{a}_1\mathbf{x} \leq b_1 + My$$
$$\mathbf{a}_2\mathbf{x} \geq b_2 - M(1-y)$$

ahol M az $\mathbf{a}_1\mathbf{x}$ és $\mathbf{a}_2\mathbf{x}$ függvények közös felső korlátja, és y bináris változó.

Változókra vonatkozó feltételek implikációja

Ha a

$$x_i > 0$$
$$x_j \geq d$$

feltételeket szeretnénk a feladathoz adni úgy, hogy az első implikálja a másodikat, akkor a megoldás:

$$x_i \leq My$$
$$x_j \geq d - M(1-y)$$

ahol $M \geq d$ az x_i felső korlátja, és y bináris változó.

Megoldási módszerek

Valós relaxáció: az EP-feladat LP-lazítása

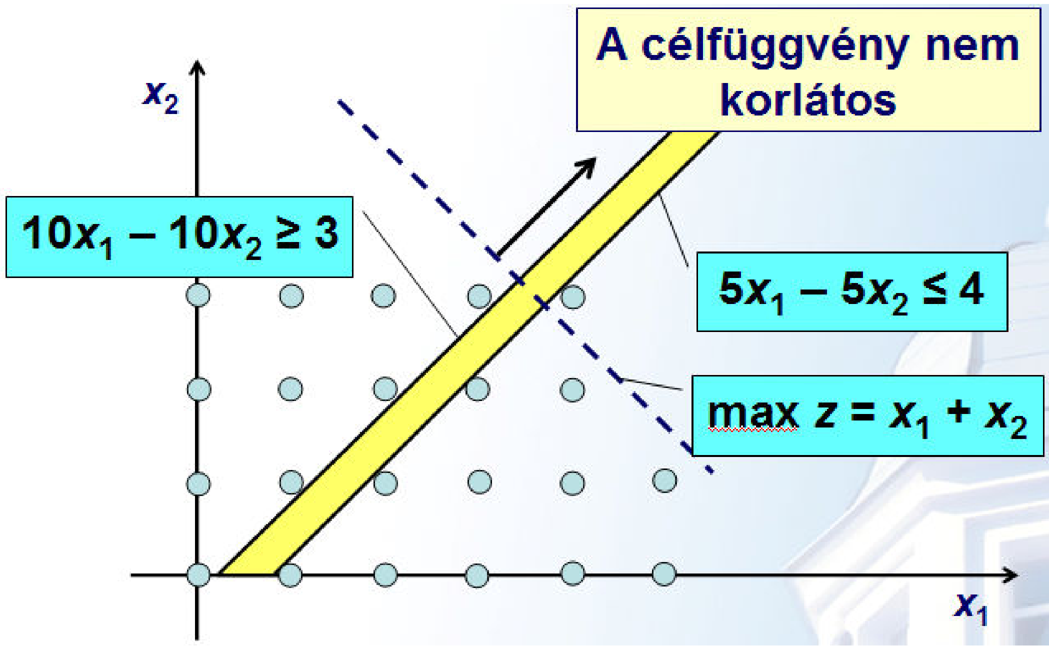
Egy EP feladat LP lazítását úgy kapjuk, hogy nem követelünk meg egészértékűséget a változóktól.

Ha az LP lazítás optimális megoldása egész, akkor megkaptuk az egészértékű feladat optimális megoldását.

Az LP lazítás optimális célfüggvényértéke legalább olyan jó, mint az EP feladaté.

Ha az LP lazításnak nincs megengedett megoldása, akkor az EP-nek sincs.

Lehetséges, hogy az EP megoldás-halmaza üres, az LP lazításé viszont nem korlátos, például:



A szétválasztás és korlátozás módszere (*Branch & Bound*)

A **szétválasztás** arra utal, hogy az EP feladatot részfeladatokra bontjuk úgy, hogy a részfeladatok célfüggvénye azonos legyen az EP feladatéval, és a lehetséges megoldásainak halmaza részhalmaza legyen az EP feladatének. A **korlátozás** azt jelenti, a részfeladatok optimális célfüggvényértékére az LP lazítás megoldásával korlátot határozunk meg és a korlát alapján döntjük el, hogy érdemes-e egy részfeladatot felbontani.

Felbontási eljárás

Ha az LP lazítás optimális megoldásában az x_i változó optimális értéke x_i^*, akkor két részfeladatot képezünk az alábbi feltételek csatolásával:

1. részfeladat: $x_i \le \lfloor x_i^* \rfloor$

2. részfeladat: $x_i \ge \lfloor x_i^* \rfloor + 1$

Célszerű azt a tört értékű változót választani, amelynek törtrésze a legnagyobb.

Felbontási stratégiák

Legjobb korlát stratégia

A legjobb korlát stratégiával mindig azt a rész-feladatot bontjuk fel, amelynek korlátja a legjobb megoldást ígéri.

A kapott részfeladatok LP lazítását azonnal megoldjuk, azaz két ágat képezünk.

LIFO stratégia

A LIFO (*last in first out*) stratégiával a legutolsó két részfeladat egyikét bontjuk fel.

A kapott részfeladatok egyikének LP lazítását oldjuk meg azonnal, azaz csak egy ágat képezünk. A másikat később vizsgáljuk.

Lezárási feltételek

Egy részfeladatot lezárunk, ha az LP lazítás:

- lehetséges megoldásainak halmaza üres;
- vagy optimális megoldása egészértékű;
- vagy optimális célfüggvényértékének egészrésze nem jobb az addig talált legjobb egészértékű megoldás célfüggvényértékénél.

Hátizsák feladat megoldása a korlátozás és szétválasztás módszerével

A probléma leírása

Adott kapacitású hátizsákot töltsünk meg a lehető legnagyobb összértékű tárgyakkal.

Optimalizálási modell

Indexek

Tárgyak száma: n

Változók

Tárgyak kiválasztása: $x_i \in \{0,1\} \ (i = 1,...,n)$

Paraméterek

Tárgyak súlya (költség): $w_i \in \mathbf{R} \ (i = 1,...,n)$

Tárgyak értéke (haszon): $p_i \in \mathbf{R} \ (i = 1,...,n)$

Hátizsák kapacitása: $c \in \mathbf{R}$

Feltételek

A hátizsákba tett tárgyak összsúlya nem haladhatja meg a hátizsák kapacitását:

$$\sum_{i=1}^{n} w_i x_i \leq c$$

Célfüggvény

A hátizsákba tett tárgyak összértéke legyen a lehető legnagyobb:

$$\sum_{i=1}^{n} p_i x_i \rightarrow \max$$

A megoldás módja

A feladat valós relaxáltjának megoldása

1. Rendezzük a tárgyakat csökkenő haszon/költség $\left(\frac{p_i}{w_i}\right)$ sorrendbe.

2. Tegyük a zsákba a tárgyakat sorban egészen addig, amíg túl nem lépjük a kapacitási korlátot.

3. Az utolsóként választott tárgynak csak a kapacitást éppen kitöltő töredékét lehetne a hátizsákba tenni.

A fenti eljárás szimplex módszer esetében azt jelenti, hogy a csökkenő haszon/költség sorrendű változókra felírt szimplex táblában mindig balról az első lehetséges oszlopban választok pivot-elemet.

A korlátozás és szétválasztás módszere

1. Oldjuk meg a feladat valós relaxáltját a fenti módszerrel.

2. Határozzuk meg az utolsóként választott tárgy változójának 0 ill. 1 értékéhez tartozó célfüggvény-értékeket, vagyis az ún. **optimumhatárokat**.

3.a.) Ha a kapott felső optimumhatár kisebb, mint az eddigi alsó optimumhatárok legnagyobbika, akkor ez a megoldási ág optimumhoz nem vezethet, ezért lezárjuk.

b.) Ha a kapott alsó optimumhatár nagyobb, mint az eddigi alsó optimumhatárok legnagyobbika, akkor újra ellenőrizzük az eddigi nyitott ágakat, és lezárjuk közülük azokat, ahol a felső optimumhatár kisebb, mint az új alsó optimumhatár.

c.) Ha a valós relaxált optimális megoldásának változói egészértékűek, akkor az alapfeladat egy lehetséges (de nem biztos, hogy optimális) megoldásához jutottunk.

d.) Máskülönben az utolsóként választott tárgy változójának értékét rendre 0 ill. 1 értéken rögzítve a problémát két kisebb méretű feladatra bontjuk, és visszalépünk az 1. pontra.

A fenti eljárásnak akkor van **vége**, ha már nincs nyitott részfeladat.

A probléma optimális **megoldása** a legnagyobb alsó optimumhatárhoz tartozó lehetséges megoldás.

Példa

Feladat

A változók száma: $n = 8$

A hátizsák kapacitása: $c = 102$

A tárgyak értéke és súlya csökkenő haszon/költség sorrendben:

$\mathbf{p}:$	15	100	90	60	40	15	10	1
$\mathbf{w}:$	2	20	20	30	40	30	60	10

Modell

$$\begin{cases} 15x_1 + 100x_2 + 90x_3 + 60x_4 + 40x_5 + 15x_6 + 10x_7 + x_8 \to \max \\ \text{-----------------------------------} \\ 2x_1 + 20x_2 + 20x_3 + 30x_4 + 40x_5 + 30x_6 + 60x_7 + 10x_8 \le 102 \\ \text{-----------------------------------} \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 \in \{0,1\} \end{cases}$$

Megoldás

Az alapprobléma valós relaxáltjának megoldása:

$\mathbf{x}: \; 1 \quad 1 \quad 1 \quad 1 \quad {}^3\!/_4 \quad 0 \quad 0 \quad 0$

A megoldáshoz tartozó optimumhatárok: $[265,295]$

Legjobb alsó optimumhatár: 265

1.1 részfeladat

Rögzített paraméterek: $x_5 = 0$

A változók száma: $n = 7$

A hátizsák kapacitása: $c = 102$

A tárgyak értéke és súlya csökkenő haszon/költség sorrendben:

$$\begin{array}{lcccccccc} \mathbf{p}: & 15 & 100 & 90 & 60 & (40) & 15 & 10 & 1 \\ \mathbf{w}: & 2 & 20 & 20 & 30 & (40) & 30 & 60 & 10 \end{array}$$

A részfeladat valós relaxáltjának megoldása:

$\mathbf{x}: \; 1 \quad 1 \quad 1 \quad 1 \quad (0) \quad 1 \quad 0 \quad 0$

A megoldáshoz tartozó optimumhatárok: $[280,280]$

Legjobb alsó optimumhatár: 280

1.2 részfeladat

Rögzített paraméterek: $x_5 = 1$

A változók száma: $n = 7$

A hátizsák kapacitása: $c = 62$

A tárgyak értéke és súlya csökkenő haszon/költség sorrendben:

$$\begin{array}{lcccccccc} \mathbf{p}: & 15 & 100 & 90 & 60 & (40) & 15 & 10 & 1 \\ \mathbf{w}: & 2 & 20 & 20 & 30 & (40) & 30 & 60 & 10 \end{array}$$

A részfeladat valós relaxáltjának megoldása:

$\mathbf{x}: \; 1 \quad 1 \quad 1 \quad {}^2\!/_3 \quad (1) \quad 0 \quad 0 \quad 0$

A megoldáshoz tartozó optimumhatárok: $[245,285]$

Legjobb alsó optimumhatár: 280

1.2.1 részfeladat

Rögzített paraméterek:

$$x_5 = 1,\ x_4 = 0$$

A változók száma: $n = 6$

A hátizsák kapacitása: $c = 62$

A tárgyak értéke és súlya csökkenő haszon/költség sorrendben:

$$\begin{array}{lcccccccc} \mathbf{p}: & 15 & 100 & 90 & (60) & (40) & 15 & 10 & 1 \\ \mathbf{w}: & 2 & 20 & 20 & (30) & (40) & 30 & 60 & 10 \end{array}$$

A részfeladat valós relaxáltjának megoldása:

$$\mathbf{x}: \ 1 \ \ 1 \ \ 1 \ \ (0) \ \ (1) \ \ \tfrac{2}{3} \ \ 0 \ \ 0$$

A megoldáshoz tartozó optimumhatárok: $[245,255]$

A felső optimumhatár kisebb, mint a legjobb alsó optimumhatár.

1.2.2 részfeladat

Rögzített paraméterek:

$$x_5 = 1,\ x_4 = 1$$

A változók száma: $n = 6$

A hátizsák kapacitása: $c = 32$

A tárgyak értéke és súlya csökkenő haszon/költség sorrendben:

$$\begin{array}{lcccccccc} \mathbf{p}: & 15 & 100 & 90 & (60) & (40) & 15 & 10 & 1 \\ \mathbf{w}: & 2 & 20 & 20 & (30) & (40) & 30 & 60 & 10 \end{array}$$

A részfeladat valós relaxáltjának megoldása:

$$\mathbf{x}: \ 1 \ \ 1 \ \ \tfrac{1}{2} \ \ (1) \ \ (1) \ \ 0 \ \ 0 \ \ 0$$

A megoldáshoz tartozó optimumhatárok: $[215,260]$

A felső optimumhatár kisebb, mint a legjobb alsó optimumhatár.

Válasz

Az optimális megoldást az 1.1 részfeladat szolgáltatja.

A korlátozás és szétválasztás alkalmazása az utazó ügynök problémára

A probléma leírása

Milyen sorrendben keressük fel egy hálózat csúcspontjait, hogy a megtett út összhossza minimális legyen?

Mivel a lehetséges körutak száma $(n-1)!$, ezért a teljes leszámlálás lehetetlen. Ehelyett EP modellt írunk fel, és azt oldjuk meg a szétválasztás és korlátozás módszerével.

EP modell

Változók bevezetése

Jelölje $x_{ij}=1$, ha az *i*-edik városból a *j*-edikbe megyünk, illetve $x_{ij}=0$, ha nem.

Feltételek felírása

1. Minden városból pontosan egyszer el kell indulni: $\sum_j x_{ij}=1$

2. Minden városba pontosan egyszer meg kell érkezni: $\sum_i x_{ij}=1$

3. Ne keletkezzenek részkörutak, hanem egyetlen teljes körút:

$$\forall j_1,\dots,j_k : x_{ij_1}+\sum_{s=1}^{k-1} x_{j_s j_{s+1}}+x_{j_k i}\le k$$

Célfüggvény

A körút hosszát minimalizálni kell

$$\sum_i\sum_j c_{ij}x_{ij}\to \min$$

A szétválasztás és korlátozás módszere hozzárendelési feladat (HF-) lazítással

1. Oldjuk meg az utazó ügynök probléma HF-lazítását.

2a) Ha az optimális megoldás körút, akkor ez a TSP optimális megoldása.

2b) Ha részkörutak keletkeztek, akkor válasszunk egyet. Ha ennek *s* éle van, akkor képezzünk *s* számú részfeladatot (ágat).

3. Oldjuk meg a részfeladatok HF-lazítását.

4. Ha az optimális megoldás körút, akkor zárjuk le. Ha e körút hossza az eddigi legrövidebb, akkor ezt jegyezzük fel és zárjuk le a magasabb korlátú aktív részfeladatokat.

5. Válasszuk ki a legkisebb korlátú részfeladatot, oldjuk meg annak HF-lazítását és menjünk a 2b) lépésre.

Az eljárás vége

Ha már nincs aktív részfeladat, akkor az eljárás befejeződött.

Járatszervezési probléma visszavezetése az utazó ügynök problémára

A járatszervezési probléma az utazó ügynök probléma általánosítása, amelyet a következőképpen szokás bemutatni.

Egy bizonyos termék iránt adott városokban meghatározott kereslet mutatkozik. A termék kiszállítására az egyik városban található központi raktárból korlátos kapacitású járművek állnak a rendelkezésünkre. A feladat a keresleti igények kielégítése a minimális összhosszúságú szállítási útvonalakon.

A probléma azért általánosítása az utazó ügynök problémának, mert a járművek korlátozott kapacitása nem teszi lehetővé a szükséges termékmennyiség egyszerre történő kiszállítását.

A járatszervezési probléma ugyanakkor visszavezethető az utazó ügynök problémára a következőképpen:

1. Virtuálisan sokszorozzuk meg a központi raktárt: vezessünk be helyette annyi különálló pontot, ahány jármű szükséges a szállítások lebonyolítására.

2. Az új pontokat összekötő éleket tekintsük végtelen hosszúnak. (Raktáron belül nem szállítunk.)

3. Az új pontokból a régiekbe vezető élek hossza legyen azonos az eredeti központi raktárból a keresleti pontokba vezető élek hosszával.

Az implicit leszámolás módszere

Implicit leszámolás minimum feladatra

Feladat

$$
\begin{cases}
3x_1 + 2x_2 + 5x_3 + 2x_4 + 3x_5 \to \min \\
------------------- \\
-x_1 - x_2 + x_3 + 2x_4 - x_5 \le 1 \\
-7x_1 + 3x_3 - 4x_4 - 3x_5 \le -2 \\
11x_1 - 6x_2 - 3x_4 - 3x_5 \le -1 \\
------------------- \\
x_1, x_2, x_3, x_4, x_5 \in \{0,1\}
\end{cases}
$$

Megoldás

Írjuk fel a probléma induló szimplex tábláját maximum feladatra:

x_1	x_2	x_3	x_4	x_5	u_1	u_2	u_3	**b**
-1	-1	1	2	-1	1			1
-7	0	3	-4	-3		1		-2
11	-6	0	-3	-3			1	-1
-3	-2	-5	-2	-3				0

A segédváltozókról csak annyi teszünk fel, hogy nemnegatív számok.

Az általános algoritmus szerint a maximum feladatra felírt célfüggvény negatív együtthatóihoz tartozó változókat 0-ra, a pozitív együtthatókhoz tartozó változókat 1-re állítjuk, és kiolvassuk az induló bázismegoldást. Jelen esetben az összes változó 0 értékéből indulunk ki:

$$(x_1, x_2, x_3, x_4, x_5, u_1, u_2, u_3) = (0, 0, 0, 0, 0, 1, -2, -1)$$

Ehhez a bázismegoldáshoz a $z = 0$ célfüggvényérték tartozik.

Mivel ebben negatív segédváltozók is szerepelnek, így ez a megoldás nem megengedett (*infeasible*). A negatív segédváltozók összegét szokás a megengedhetetlenség mértékének (*infeasibility measure*) nevezni, ami a jelen esetben: $(-2) + (-1) = -3$

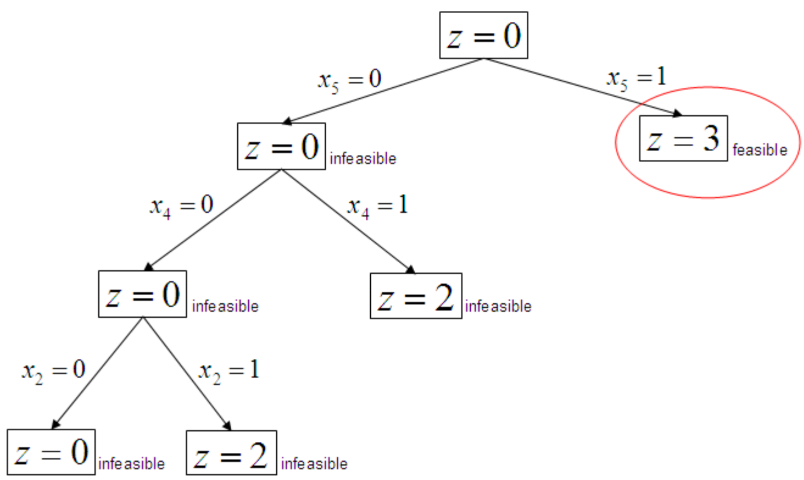
Ezek után egyenként megvizsgálom, hogy a meg nem engedett bázismegoldás (bináris) változóit 0-ról 1-re növelve hogyan változik a helyzet. Ezzel az esetvizsgálattal egy olyan fát generálok, amelynek csúcsaira a célfüggvény értékét, ágaira pedig az adott elágazáshoz tartozó döntést – egy bináris változó beállított értékét – írom.

A változók értékének beállításával kapott esetek a következők:

Döntés	Bázismegoldás	Célérték	*Infeasibility*
$x_1 = 1$	$(1, 0, 0, 0, 0, 2, 5, -12)$	$z = 3$	-12
$x_2 = 1$	$(0, 1, 0, 0, 0, 2, -2, 5)$	$z = 2$	-2
$x_3 = 1$	$(0, 0, 1, 0, 0, 0, -5, -1)$	$z = 5$	-6
$x_4 = 1$	$(0, 0, 0, 1, 0, -1, 2, 2)$	$z = 2$	-1
$x_5 = 1$	$(0, 0, 0, 0, 1, 2, 1, 2)$	$z = 3$	0

Vegyük észre, hogy az utolsó eset *infeasibility* mértéke 0, vagyis ott egy megengedett megoldást kapunk. Ezt az ágat tovább nem vizsgáljuk, mert ha további változók értékét módosítanánk, azzal csak rontanánk a célfüggvény értékén. A megtalált megengedett megoldások viszont egyben korlátot is jelentenek a célértékre nézve, így viszonyítási alapként szolgálnak a még le nem zárt ágak értékeléséhez. Az első és a harmadik eset célértéke például eleve rosszabb, mint a már megtalált megoldásé, így azoknak az ágaknak a további vizsgálata felesleges. Az általános szabály az, hogy le kell zárni minden olyan ág vizsgálatát, ahol rosszabb célértéket kaptunk, mint az eddig elért legjobb megengedhető megoldásnál, vagy azonos célértékkel ugyan, de nem megengedhető megoldáshoz jutottunk.

Az algoritmus szerint az esetvizsgálattal azon a meg nem engedett ágon kell továbblépni, ahol a megengedhetetlenség mértéke a legkisebb. Amennyiben több minimális jelölt is van, akkor közülük azt választjuk, amelyiknek kedvezőbb a célértéke. A jelen esetben az $x_4 = 1$ döntés esetében van remény a célérték javítására. Folytatva az iterációt az alábbi megoldási fához jutunk:



Hátizsák feladat megoldása implicit leszámolással

Feladat

A változók száma: $n = 4$

A hátizsák kapacitása: $c = 7$

A tárgyak értéke és súlya csökkenő haszon/költség sorrendben:

$$\begin{array}{llllll} \mathbf{p}: & 3 & 5 & 2 & 1 \\ \mathbf{w}: & 1 & 4 & 3 & 2 \end{array}$$

Modell

$$\begin{cases} 3x_1 + 5x_2 + 2x_3 + x_4 \to \max \\ ----------------- \\ x_1 + 4x_2 + 3x_3 + 2x_4 \le 7 \\ ----------------- \\ x_1, x_2, x_3, x_4 \in \{0,1\} \end{cases}$$

Megoldás

Írjuk fel a probléma induló szimplex tábláját maximum feladatra:

x_1	x_2	x_3	x_4	u_1	**b**
1	4	3	2	1	7
3	5	2	1	0	0

A segédváltozókról csak annyi teszünk fel, hogy nemnegatív számok.

Az általános algoritmus szerint a maximum feladatra felírt célfüggvény negatív együtthatóihoz tartozó változókat 0-ra, a pozitív együtthatókhoz tartozó változókat 1-re állítjuk, és kiolvassuk az induló bázismegoldást. Jelen esetben az összes változó 1 értékéből indulunk ki: $(x_1, x_2, x_3, x_4, u_1) = (1, 1, 1, 1, -3)$

Ehhez a bázismegoldáshoz a $z = 11$ célfüggvényérték tartozik.

Mivel ebben a segédváltozó negatív, így ez a megoldás nem megengedett (*infeasible*). A megengedhetetlenség mértéke (*infeasibility measure*) a jelen esetben: $u_1 = -3$

Ezek után egyenként megvizsgálom, hogy a meg nem engedett bázismegoldás (bináris) változóit 1-ről 0-ra csökkentve hogyan változik a helyzet. Ezzel az esetvizsgálattal egy olyan fát generálok, amelynek csúcsaira a célfüggvény értékét, ágaira pedig az adott elágazáshoz tartozó döntést – egy bináris változó beállított értékét – írom.

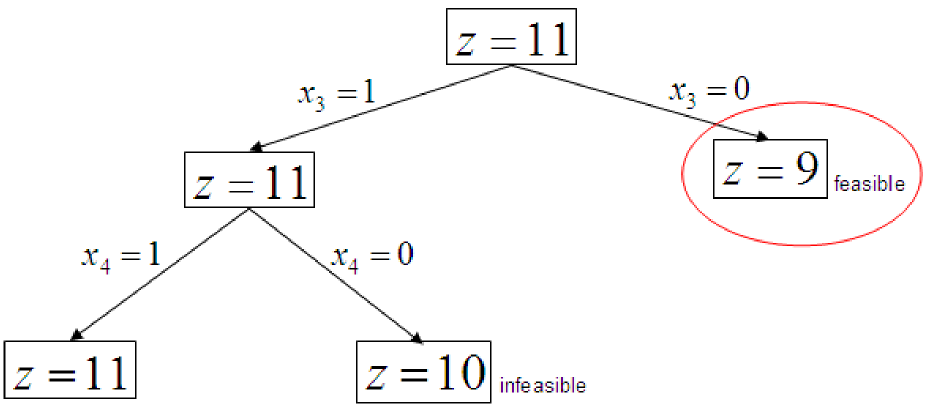
A változók értékének beállításával kapott esetek a következők:

Döntés	Bázismegoldás	Célérték	*Infeasibility*
$x_1 = 0$	$(0,1,1,1,-2)$	$z = 8$	-2
$x_2 = 0$	$(1,0,1,1,1)$	$z = 6$	0
$x_3 = 0$	$(1,1,0,1,0)$	$z = 9$	0
$x_4 = 0$	$(0,1,1,0,-1)$	$z = 10$	-1

Vegyük észre, hogy a második és harmadik eset *infeasibility* mértéke 0, vagyis ott megengedett megoldásokat kapunk. Ezek közül az $x_3 = 0$ döntéssel jutunk jobb célértékhez, ami viszonyítási alapként szolgál a még le nem zárt ágak értékeléséhez. Az első eset célértéke például eleve rosszabb, mint a már megtalált megoldásé, így annak az ágaknak a további vizsgálata felesleges. Az általános szabály az, hogy le kell zárni minden olyan ág vizsgálatát, ahol rosszabb célértéket kaptunk, mint az eddig elért legjobb megengedhető megoldásnál, vagy azonos célértékkel ugyan, de nem megengedhető megoldáshoz jutottunk.

Az algoritmus szerint az esetvizsgálattal azon a meg nem engedett ágon kell továbblépni, ahol a megengedhetetlenség mértéke a legkisebb. Amennyiben több minimális jelölt is van, akkor közülük azt választjuk, amelyiknek kedvezőbb a célértéke. A jelen esetben az $x_4 = 0$ döntés esetében van remény a célérték javítására.

Folytatva az iterációt az alábbi megoldási fához jutunk:



A vágások módszere (Gomory)

Hátizsák-feladat megoldása Gomory-féle vágással

Ezt a módszert akkor vetjük be, amikor a relaxált változat eléri az optimumát a szimplex táblában.

Feladat

A változók száma: $n = 4$

A hátizsák kapacitása: $c = 7$

A tárgyak értéke és súlya csökkenő haszon/költség sorrendben:

$$\begin{array}{lcccc} \mathbf{p}: & 3 & 5 & 2 & 1 \\ \mathbf{w}: & 1 & 4 & 3 & 2 \end{array}$$

Modell

$$\begin{cases} 3x_1 + 5x_2 + 2x_3 + x_4 \to \max \\ ---------------- \\ x_1 + 4x_2 + 3x_3 + 2x_4 \leq 7 \\ ---------------- \\ x_1, x_2, x_3, x_4 \in \{0,1\} \end{cases}$$

A valós relaxált megoldása szimplex módszerrel

x_1	x_2	x_3	x_4	u_1	u_2	u_3	u_4	u_5	c
1	4	3	2	1					7
$\underline{\underline{1}}$					1				1
	1					1			1
		1					1		1
			1					1	1
3	5	2	1						0

x_1	x_2	x_3	x_4	u_1	u_2	u_3	u_4	u_5	c
	4	3	2	1	-1				6
1					1				1
	$\underline{\underline{1}}$					1			1
		1					1		1
			1					1	1
	5	2	1		-3				-3

x_1	x_2	x_3	x_4	u_1	u_2	u_3	u_4	u_5	c
		$\underline{\underline{3}}$	2	1	−1	−4			2
1					1				1
	1					1			1
		1					1		1
			1					1	1
		2	1		−3	−5			−8

x_1	x_2	x_3	x_4	u_1	u_2	u_3	u_4	u_5	c
		1	$2/3$	$1/3$	$-1/3$	$-4/3$			$2/3$
1					1				1
	1					1			1
			$-2/3$	$-1/3$	$1/3$	$4/3$	1		$1/3$
			1					1	1
			$-1/3$	$-2/3$	$-7/3$	$-7/3$			$-28/3$

A fenti tábla optimális.

Gomory-féle vágás

Ha a valós relaxált megoldása egészértékű lenne, akkor az alapfeladatnak is optimális megoldása lenne. Azonban a relaxált megoldásvektor egyik komponense nem egész: $x_3 = 2/3$

Amennyiben a relaxált megoldásvektor egyik komponense nem egész, akkor van az optimális táblában olyan sor, amiben ennek a komponensnek az együtthatója nem nulla. Írjuk fel ennek a sornak az egyenletét úgy, hogy az egyenlet tagjait egy egész és egy **nemnegatív törtrész** összegére bontjuk:

$$(1+0)\cdot x_3 + (0+2/3)\cdot x_4 + (0+1/3)\cdot u_1 + (-1+2/3)\cdot u_2 + (-2+2/3)\cdot u_3 = 2/3$$

Rendezzük egy oldalra az egész, illetve a tört tagokat:

$$x_3 - u_2 - 2\cdot u_3 = 2/3 - 2/3\cdot x_4 - 1/3\cdot u_1 - 2/3\cdot u_2 - 2/3\cdot u_3$$

Mármost amennyiben egy egészértékű lehetséges megoldást helyettesítünk a fenti egyenletbe, úgy a baloldalon egész érték adódik, következésképpen a jobboldalon is egészértékű kifejezés áll.

Tekintettel arra, hogy a változók a feladat szerint nemnegatív egész számok, a jobb oldalon csak egy nempozitív egész szám állhat, azaz a feladat egészértékű megoldásai kielégítik a következő feltételt:

$$2/3 - 2/3\cdot x_4 - 1/3\cdot u_1 - 2/3\cdot u_2 - 2/3\cdot u_3 \le 0$$

Ez a pótlólagos feltétel a Gomory-féle **vágás**: egy metszősík a problématérben.

A vágás (és a hozzá tartozó újabb segédváltozó) hozzávételével a valós relaxált optimális tábláját elrontjuk ugyan, de egészértékű megoldásokat nem veszítünk.

x_1	x_2	x_3	x_4	u_1	u_2	u_3	u_4	u_5	$\boldsymbol{u_6}$	c
		1	$\frac{2}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$	$-\frac{4}{3}$				$\frac{2}{3}$
1					1					1
	1					1				1
			$-\frac{2}{3}$	$-\frac{1}{3}$	$\frac{1}{3}$	$\frac{4}{3}$	1			$\frac{1}{3}$
			1					1		1
			$\boldsymbol{-\frac{2}{3}}$	$\boldsymbol{-\frac{1}{3}}$	$\boldsymbol{-\frac{2}{3}}$	$\boldsymbol{-\frac{2}{3}}$			**1**	$\boldsymbol{-\frac{2}{3}}$
			$-\frac{1}{3}$	$-\frac{2}{3}$	$-\frac{7}{3}$	$-\frac{7}{3}$				$-\frac{28}{3}$

A pótlólagos feltétel sora tehát a törtrészek negatívjait, és a pótlólagos segédváltozót tartalmazza. A pótlólagos feltétel kapacitásváltozója negatív, ezért a tábla primál nem optimális. Duál szimplex lépés következik: a pótlólagos feltétel sorából választunk pivot-elemet úgy, hogy a célsor változói negatívak maradjanak.

x_1	x_2	x_3	x_4	u_1	u_2	u_3	u_4	u_5	u_6	c
		1	$\frac{2}{3}$	$\frac{1}{3}$	$-\frac{1}{3}$	$-\frac{4}{3}$				$\frac{2}{3}$
1					1					1
	1					1				1
			$-\frac{2}{3}$	$-\frac{1}{3}$	$\frac{1}{3}$	$\frac{4}{3}$	1			$\frac{1}{3}$
			1					1		1
			$-\frac{2}{3}$	$-\frac{1}{3}$	$-\frac{2}{3}$	$-\frac{2}{3}$			1	$-\frac{2}{3}$
			$-\frac{1}{3}$	$-\frac{2}{3}$	$-\frac{7}{3}$	$-\frac{7}{3}$				$-\frac{28}{3}$

x_1	x_2	x_3	x_4	u_1	u_2	u_3	u_4	u_5	u_6	c
		1			-1	-2			1	
1						1			0	1
	1					0			0	1
					1	2	1		-1	1
				$-\frac{1}{2}$	-1	-1		1	$\frac{3}{2}$	
			1	$\frac{1}{2}$	1	1			$-\frac{3}{2}$	1
				$-\frac{1}{2}$	-2	-2			$-\frac{1}{2}$	-9

Ez a tábla megint primál optimális, és ráadásul egészértékű megoldást ad, tehát egyben az alapfeladatnak is megoldása. Ha nem így lenne, további duál szimplex lépéseket tehetnénk a tábla optimalizálása érdekében, illetve további vágásokkal biztosíthatnánk a megoldásvektor komponenseinek egészértékűségét.

Duális teljesen egész vágás

Duális teljesen egész vágás esetében nem garantált, hogy csökken a feltételi halmaz mérete.

Feladat

$$\begin{cases} 2x_1 + x_2 + 2x_3 \to \min \\ ------------ \\ x_1 + 2x_2 - x_3 \le 1 \\ x_1 - 2x_2 - 3x_3 \le -2 \\ ------------ \\ x_1, x_2, x_3 \in \mathbf{N} \end{cases}$$

Megoldás

Írjuk fel az induló szimplex táblát negatív célfüggvénnyel, maximum feladatra:

x_1	x_2	x_3	u_1	u_2	**b**
1	2	−1	1	0	1
1	−2	−3	0	1	−2
−2	−1	−2	0	0	0

Mivel a második sorban negatív kapacitás szerepel, így a tábla nem optimális.

Legyen λ a nem megengedhető sor legkisebb együtthatójának abszolút értéke, azaz a negatív együtthatók abszolút értékeinek legnagyobbika: $\lambda = \max\{|-2|, |-3|\} = 3$

Osszuk el a nem megengedhető sort λ-val, és írjuk fel a sor egyenletét úgy, hogy az egyenlet tagjait egy egész és egy **nemnegatív törtrész** összegére bontjuk:

$$\left(0 + \tfrac{1}{3}\right) \cdot x_1 + \left(-1 + \tfrac{1}{3}\right) \cdot x_2 + \left(-1 + 0\right) \cdot x_3 + \left(0 + \tfrac{1}{3}\right) \cdot u_2 = \left(-1 + \tfrac{1}{3}\right)$$

Rendezzük egy oldalra az egész, illetve a tört tagokat:

$$-x_2 - x_3 + 1 = \tfrac{1}{3} - \tfrac{1}{3} \cdot x_1 - \tfrac{1}{3} \cdot x_2 - \tfrac{1}{3} \cdot u_2$$

Mármost amennyiben egy egészértékű lehetséges megoldást helyettesítünk a fenti egyenletbe, úgy a baloldalon egész érték adódik, következésképpen a jobboldalon is egészértékű kifejezés áll.

Tekintettel arra, hogy a változók a feladat szerint nemnegatív egész számok, a jobb oldalon csak egy nempozitív egész szám állhat, így a baloldalon is, azaz a feladat egészértékű megoldásai kielégítik a következő feltételt:

$$-x_2 - x_3 \le -1$$

Ez a pótlólagos feltétel a duális teljesen egész vágás, aminek a hozzávételével a szimplex tábla így alakul:

x_1	x_2	x_3	u_1	u_2	u_3	**b**
1	2	-1	1	0	0	1
1	-2	-3	0	1	0	-2
0	$\underline{\underline{-1}}$	-1	0	0	1	-1
-2	-1	-2	0	0	0	0

A pótlólagos feltétel sora tehát az egészrészeket és a pótlólagos segédváltozót tartalmazza.

Duális pivotálás következik a vágás sorában. (Maximumfeladat esetében garantáltan -1, minimumfeladat esetében garantáltan 1 lesz a pivot-elem.)

x_1	x_2	x_3	u_1	u_2	u_3	**b**
1	0	-3	1	0	2	-1
1	0	-1	0	1	-2	0
0	1	1	0	0	-1	1
-2	0	-1	0	0	-1	1

Most az első sor negatív kapacitása miatt nem optimális a tábla, ezért újabb duális teljesen egész vágás következik:

$$\lambda = \max\{|-3|\} = 3$$

Az λ-val való osztás után fennmaradó egészrészekkel és egy újabb segédváltozóval képezzük a vágást:

x_1	x_2	x_3	u_1	u_2	u_3	u_4	**b**
1	0	-3	1	0	2	0	-1
1	0	-1	0	1	-2	0	0
0	1	1	0	0	-1	0	1
0	0	$\underline{\underline{-1}}$	0	0	0	1	-1
-2	0	-1	0	0	-1	0	1

Duális pivotálás következik a vágás sorában:

x_1	x_2	x_3	u_1	u_2	u_3	u_4	**b**
1	0	0	1	0	2	-3	2
1	0	0	0	1	-2	-1	1
0	1	0	0	0	-1	1	0
0	0	1	0	0	0	-1	1
-2	0	0	0	0	-1	-1	2

Ez a tábla már optimális és egész értékű megoldást ad, ami tehát az eredeti feladatnak is optimális megoldása: (**0, 0, 1**, 2, 1, 0, 0)

A duális teljesen egész vágás egy determinisztikus véges eljárást ad egész értékű LP-problémák megoldásához.

Dinamikus programozás

Dirichlet-elv

Skatulya-elv

Pontrjagin-Bellman optimalitási elv

Optimális útnak minden rész-útja optimális.

Hátizsák feladat megoldása dinamikus programozással

Feladat

$$\begin{cases} 3x_1 + 5x_2 + 2x_3 + x_4 \to \max \\ ---------------- \\ x_1 + 4x_2 + 4x_3 + 2x_4 \leq 7 \\ ---------------- \\ x_1, x_2, x_3, x_4 \in \{0,1\} \end{cases}$$

Megoldás

Felállítok egy diszkrét koordináta-rendszert a változók száma és a kapacitási korlát méretében. Több feltétel esetén minden egyes feltételnek egy külön dimenziót biztosítok.

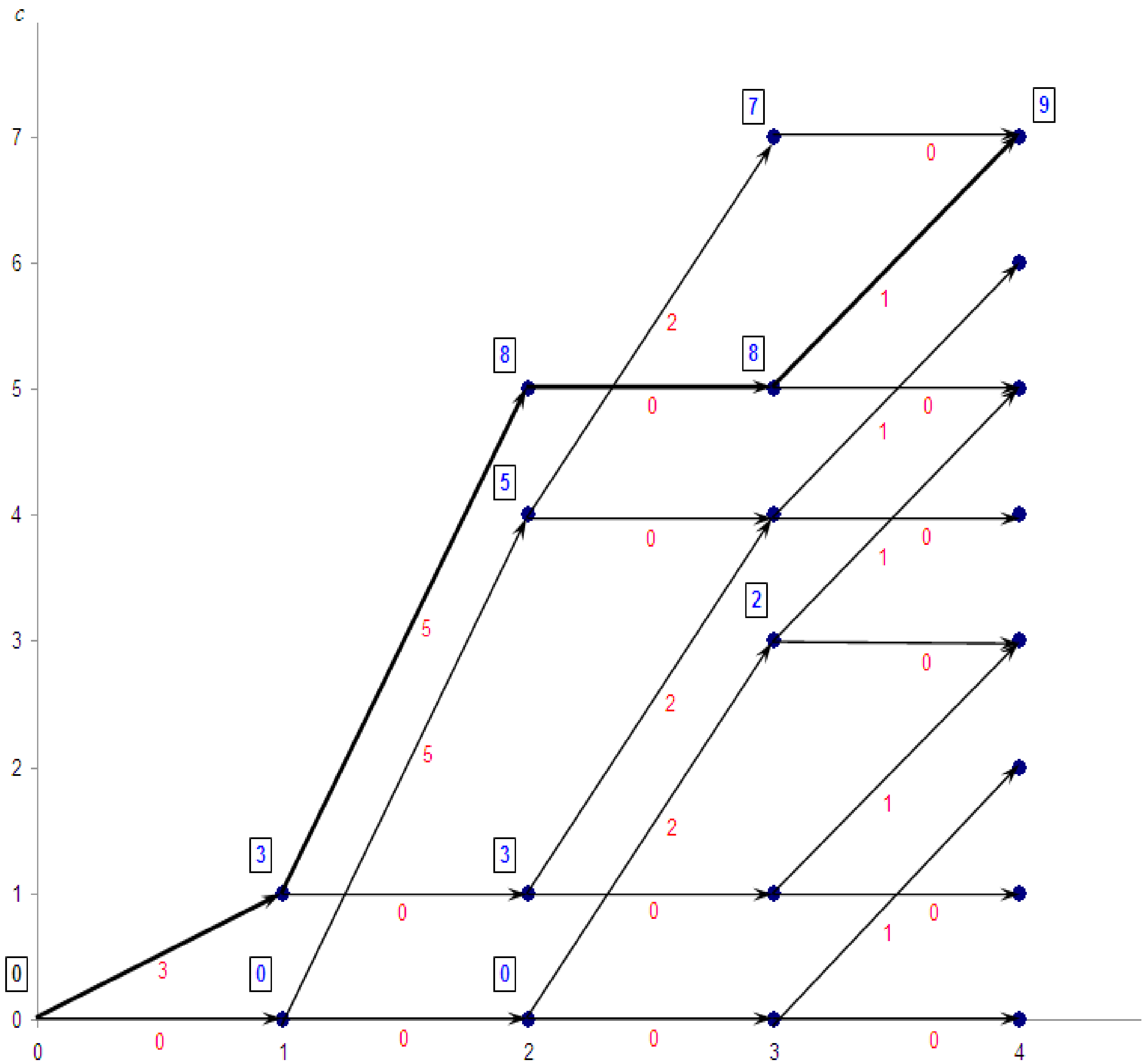
Előremenő irány

Előremenő irányban az origóból indulok, ahol a változók értéke zérus, és minden egyes lépésben az összes eddig elért pontból a soron következő változó kapacitási együtthatójának megfelelő meredekségű nyilakat állítok, ha a változó értékét 1-re emelem, illetve vízszintesen megyek tovább, ha a változó értékét 0-ban rögzítem.

A pozitív meredekségű nyilakra felírom a változó célfüggvénybeli együtthatóját (a vízszintes nyilakon a változó 0 értékének megfelelően nincs haszon). Minden ponthoz felírom az oda vezető irányított utak közül a maximális hasznú út összértékét, így a következő lépésben elegendő csak az előző állapotot vizsgálni, ami programozási szempontból előnyös.

A programozásnál további előny jelenthet, ha számon tartjuk a hátralévő változók által még megszerezhető összes lehetséges hasznot, és ha egy pont eddigi értékéhez ezt a hasznot hozzáadva nem érjük el az eddigi maximális értékű részút hasznát, akkor ebből a pontból már nem is érdemes folytatni az utak növelését.

A jelenlegi feladat esetén előremenő irányban a következő úthálózathoz jutunk, amiből a legfelső, 9 hasznú út mutatkozik maximálisnak értékűnek:

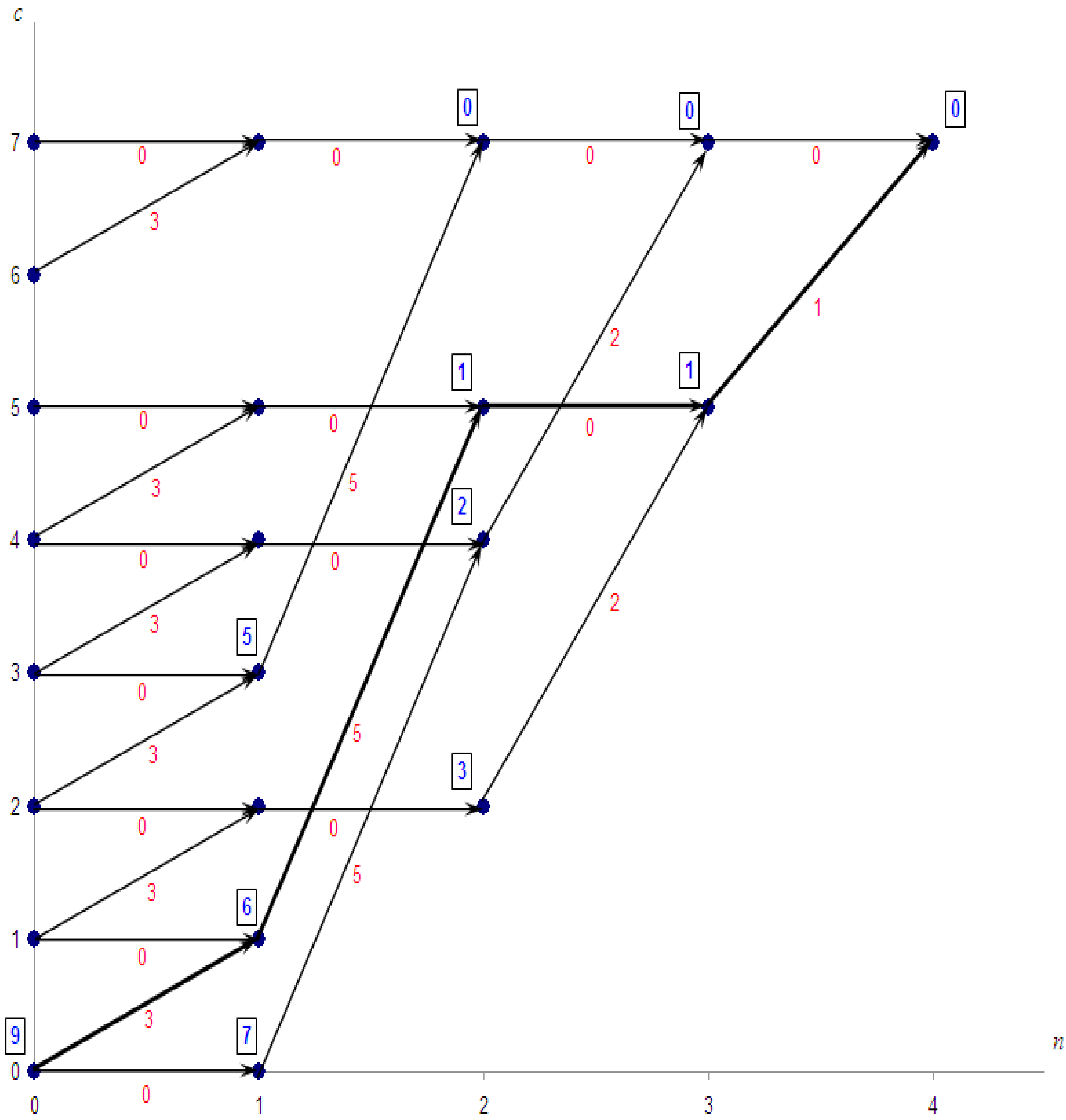


A maximális hasznú út egyes állomásairól leolvasható, hogy a változók milyen értékének a választása vezetett a haszon maximalizálásához:

$$x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ x_4 = 1$$

Visszamenő irány

Visszamenő irányú programozás esetén a változók csupa 1-es állapotából és a maximális kapacitásból indulok ki, majd ugyanazon elvek szerint szerkesztem meg az úthálózatot:



Ha jól dolgoztunk, akkor előremenő és visszamenő irányban is ugyanazt a maximális hasznú utat kapjuk végeredménynek, természetesen ugyanazokkal a változóértékekkel.

Fiktív lejátszás

A fiktív lejátszások módszere a kétszemélyes zérusösszegű játékok megoldásának egy lassú, de rendkívül egyszerű módja, amely egészértékű LP-problémák megoldására is alkalmas. A módszer előnye, hogy számítógépes implementációja teljesen megbízható: a valós relaxáción alapuló szimplex módszerek kerekítésből eredő bizonytalanságai itt nem fordulhatnak elő. A módszer alkalmazhatóságának az alapja az LP-problémák és a kétszemélyes zérusösszegű játékok közötti átjárhatóság, az első lépés tehát az LP-probléma ekvivalens átalakítása egy kétszemélyes zérusösszegű játékká.

LP-probléma modellezése kétszemélyes zérusösszegű játékkal

Tekintsünk egy standard alakú LP-feladatot és annak duálisát a következő jelölésekkel:

primál feladat

$$\begin{cases} \mathbf{cx} \to \max \\ \mathbf{Ax} \le \mathbf{b} \\ \mathbf{x} \ge \mathbf{0} \end{cases}$$

duál feladat

$$\begin{cases} \mathbf{yb} \to \min \\ \mathbf{yA} \ge \mathbf{c} \\ \mathbf{y} \ge \mathbf{0} \end{cases}$$

Tudjuk, hogy a primál feladat bármely lehetséges $\mathbf{x}$ megoldása és a duál feladat bármely lehetséges $\mathbf{y}$ megoldása esetén $\mathbf{cx} \le (\mathbf{yA})\mathbf{x} \le \mathbf{y}(\mathbf{Ax}) \le \mathbf{yb}$ (gyenge dualitás), és az egyenlőség pontosan akkor teljesül, ha mindkét megoldás optimális (erős dualitás).

Tekintsük most azt a kétszemélyes zérusösszegű játékot, amelynek a kifizetési mátrixa

$$\mathbf{P} = \left(\begin{array}{c|c|c} & \mathbf{A} & -\mathbf{b} \\ \hline -\mathbf{A}^{\mathrm{T}} & & \mathbf{c}^{\mathrm{T}} \\ \hline \mathbf{b}^{\mathrm{T}} & -\mathbf{c} & \end{array} \right) \qquad (C \text{ fizet } R\text{-nek } p_{ij} \text{ összeget}).$$

Minthogy a mátrix antiszimmetrikus, ezért ez egy szimmetrikus játék, az optimális stratégiavektorok tehát egyformák. Legyen az R játékos kevert stratégiája $\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ w \end{bmatrix}$. Ezzel szorozva a kifizetési mátrixot (C játékát optimálisnak feltételezve) nem kaphatunk pozitív értéket, azaz

$$\left(\begin{array}{c:c:c} & \mathbf{A} & -\mathbf{b} \\ \hdashline -\mathbf{A}^{\mathrm{T}} & & \mathbf{c}^{\mathrm{T}} \\ \hdashline \mathbf{b}^{\mathrm{T}} & -\mathbf{c} & \end{array}\right) \cdot \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ w \end{bmatrix} \le \mathbf{0}$$

A szorzást kifejtve az alábbi egyenlőtlenség-rendszerhez jutunk:

$$\begin{cases} \mathbf{A}\mathbf{v} - \mathbf{b}w \le \mathbf{0} \\ -\mathbf{A}^{\mathrm{T}}\mathbf{u} + \mathbf{c}^{\mathrm{T}} w \le \mathbf{0} \\ \mathbf{b}^{\mathrm{T}}\mathbf{u} - \mathbf{c}\mathbf{v} \le \mathbf{0} \end{cases}$$

Tételezzük fel, hogy a w skalár értéke nem nulla, osszuk végig vele az egyenlőtlenségeket, a szorzatok megfordításával küszöböljük ki a transzponálásokat, és átrendezéssel szüntessük meg a negatív előjeleket:

$$\begin{cases} \mathbf{A}\left[\dfrac{1}{w}\mathbf{v}\right] \le \mathbf{b} \\ \left[\dfrac{1}{w}\mathbf{u}^{\mathrm{T}}\right]\mathbf{A} \ge \mathbf{c} \\ \left[\dfrac{1}{w}\mathbf{u}^{\mathrm{T}}\right]\mathbf{b} \le \mathbf{c}\left[\dfrac{1}{w}\mathbf{v}\right] \end{cases}$$

Vezessük be az $\mathbf{x} = \left[\dfrac{1}{w}\mathbf{v}\right]$ illetve $\mathbf{y} = \left[\dfrac{1}{w}\mathbf{u}^{\mathrm{T}}\right]$ vektorjelöléseket, amikkel a fenti egyenlőtlenségrendszer az alábbi, ismerős formát nyeri:

$$\begin{cases} \mathbf{A}\mathbf{x} \le \mathbf{b} \\ \mathbf{y}\mathbf{A} \ge \mathbf{c} \\ \mathbf{y}\mathbf{b} \le \mathbf{c}\mathbf{x} \end{cases}$$

Vegyük észre, hogy az egyenlőtlenség-rendszer első két tagja a primál és a duál LP-problémák feltételeivel azonos, míg a harmadik egyenlőtlenség éppen a gyenge dualitással ellenkező irányú, ami csak abban az esetben lehetséges, ha itt az egyenlőség teljesül – ekkor viszont az erős dualitás tétele szerint $\mathbf{x}$ ill. $\mathbf{y}$ optimális megoldásai a primál ill. duál LP-feladatnak.

Összefoglalva tehát: ha az LP-feladat elemeiből a fenti módon összeállított kifizetési mátrixú kétszemélyes zérusösszegű játék optimális $\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ w \end{bmatrix}$ kevert stratégiájában $w \ne 0$, akkor $\mathbf{x} = \left[\dfrac{1}{w}\mathbf{v}\right]$ a primál feladatnak, $\mathbf{y} = \left[\dfrac{1}{w}\mathbf{u}^{\mathrm{T}}\right]$ pedig a duál feladatnak optimális megoldása.

Kétszemélyes zérusösszegű játék megoldása fiktív lejátszásokkal

A fiktív lejátszások lényege az, hogy a játékosok felváltva módosíthatnak stratégiát minden egyes lejátszás után. A játékosok nyilvántartják a lejátszások összes kifizetéseit, és a következő lépésben úgy választanak stratégiát, hogy a teljes kifizetés a számukra legkedvezőbben alakuljon. Julia Robinson 1951-ben igazolta, hogy a fiktív lejátszások során követett stratégiák gyakoriságát véve egy konvergens kevert stratégia-sorozathoz jutunk, amelynek a határértéke éppen az optimális stratégia lesz.

A módszer egy konkrét példán szemléltetjük. Legyen a játék kifizetési mátrixa:

$$\begin{array}{cc} & C \\ R & \begin{bmatrix} 0 & -1 & 2 & 0 & 3 \\ 1 & 1 & 2 & -1 & -1 \\ 0 & 1 & 0 & 1 & 3 \\ 4 & 2 & -1 & 0 & 1 \end{bmatrix} \end{array} \qquad (C \text{ fizet } R\text{-nek } p_{ij} \text{ összeget})$$

Az elinduláshoz R kezdjen az első stratégiával. A stratégiaválasztást csillagozással jelöljük. A mátrix első sora C teljes kifizetési sorát tartalmazza az első lejátszásra nézve: ezt a kifizetési mátrix alatt tartjuk nyilván. A teljes kifizetések optimális elemét elosztva a lejátszások számával egy közelítő becslést kaphatunk majd a játék értékére nézve.

$$\begin{array}{c|ccccc|l} & & & C & & & \\ \hline & 0 & -1 & 2 & 0 & 3 & {}^{*} \\ & 1 & 1 & 2 & -1 & -1 & \\ R & 0 & 1 & 0 & 1 & 3 & \\ & 4 & 2 & -1 & 0 & 1 & \\ \hline & 0 & -1 & 2 & 0 & 3 & \end{array}$$

A következő lépésben C stratégiát választhat. Számára a legkedvezőbb az, ha a teljes kifizetési sorában a legkisebb elemhez tartozó stratégiát, vagyis a mátrix második oszlopát választja. Ez lesz most R teljes kifizetési oszlopa, amit a mátrixtól jobbra kezdünk nyilvántartani:

$$\begin{array}{c|ccccc|cr} & & & C & & & & \\ \hline & 0 & -1 & 2 & 0 & 3 & {}^{*} & -1 \\ & 1 & 1 & 2 & -1 & -1 & & 1 \\ R & 0 & 1 & 0 & 1 & 3 & & 1 \\ & 4 & 2 & -1 & 0 & 1 & & 2 \\ \hline & 0 & -1^{*} & 2 & 0 & 3 & & \end{array}$$

Most R következik, akinek a teljes kifizetési oszlopában a maximális elemhez tartozó, utolsó stratégiát érdemes választani. A mátrix utolsó sorát hozzá kell adnunk tehát C teljes kifizetési sorához.

	C						
R	0	−1	2	0	3	*	−1
	1	1	2	−1	−1		1
	0	1	0	1	3		1
	4	2	−1	0	1		2^*
	0	-1^*	2	0	3		
	4	1	1	0	4		

Most C válthat stratégiát, aki megint a teljes kifizetési sor minimális eleméhez tartozó negyedik stratégiát választja: a mátrix negyedik oszlopát a következő lépésben hozzáadjuk tehát R teljes kifizetési oszlopához, és így tovább.

Amennyiben a stratégiaváltáskor egy játékos több optimális kifizetéssel is rendelkezik, a fiktív lejátszások módszerének különféle variánsai szerint választhatja a sorban első optimális kifizetéshez tartozó stratégiát, vagy azt, amelyikre nézve a partner a következő lépésben a legkisebb előnyhöz juthat stb.

A sorban mindig az első optimális kifizetéshez tartozó stratégia választásával a fiktív lejátszások következő sorozatához jutunk:

	C														
R	0	−1	2	0	3	*	−1	−1	1	1	1	1	3	5^*
	1	1	2	−1	−1		1	0	2^*	1	0	−1	2	4
	0	1	0	1	3		1	2^*	2	3^*	4^*	5^*	5^*	5
	4	2	−1	0	1		2^*	2	1	1	1	1	0	−1
	0	-1^*	2	0	3									
	4	1	1	0^*	4									
	4	2	1^*	1	7									
	5	3	3	0^*	6									
	5	4	3	1^*	9									
	5	5	3	2^*	12									
	5	6	3^*	3	15									
	5	7	3^*	4	18									
	5	6	5	4^*	21									

Az eddigi lejátszásokat összesítve a következő becsléseket kapjuk R illetve C optimális kevert stratégiájára:

$$S_R = \left(\frac{2}{9}, \frac{1}{9}, \frac{5}{9}, \frac{1}{9}\right) \qquad\qquad S_C = \left(0, \frac{1}{9}, \frac{3}{9}, \frac{5}{9}, 0\right)$$

A játék értéke az utolsó kifizetések alapján $\frac{4}{9}$ és $\frac{5}{9}$ között van.

Megjegyzés a fiktív lejátszások számítógépes implementációjához

Az algoritmus implementációjánál elegendő 5 adatstruktúrát tárolni: a kifizetési mátrixot, valamint az egyes játékosok stratégiaválasztásainak gyakoriságát és teljes kifizetési sorát. A pontosság maximalizálása érdekében a struktúrák egész értékű komponenseit korlátlan méretű egész típusokkal is reprezentálhatjuk.

Prekondicionálás

A prekondicionálás (*preconditioning, preprocessing, inspection*): egy probléma előkészítése a megoldásra. Olyan hatékony előzetes lépések és eljárások sorozata, amelyek olcsón és gyorsan azonosítják a felállított modell vagy a konkrét probléma megformulázásában található hibát; kimutatják a probléma esetleges megoldhatatlanságát; a felesleges feltételek kiküszöbölésével csökkentik a feladat méretét; erősítik az ismeretlen változók korlátjait; illetve behúzzák a feladatot a numerikus stabilitás tartományába, kiküszöbölve ezáltal a nagyságrendi eltérésekből adódó nehézségeket.

Standard LP-probléma prekondicionálása

A prekondicionálás elemeinek bemutatásához tekintsük az alábbi, standard alakú LP-problémát:

$$\begin{cases} c_1x_1 + \ldots + c_nx_n \to \max \\ \text{-------------} \\ a_{11}x_1 + \ldots + a_{1n}x_n \le b_1 \\ \vdots \qquad\qquad\qquad \vdots \\ a_{m1}x_1 + \ldots + a_{mn}x_n \le b_m \\ \text{-------------} \\ 0 \le l_1 \le x_1 \le u_1 \le \infty \\ \quad \vdots \qquad \vdots \\ 0 \le l_n \le x_n \le u_n \le \infty \end{cases}$$

A prekondicionálás során a kulcslépés az együtthatók előjel szerinti szétválasztása lesz, amely utána az LP-modell a következő alakba írható:

$$\begin{cases} c_1x_1 + \ldots + c_nx_n \to \max \\ --------------- \\ \sum\limits_{a_{1j}>0} a_{1j}x_j + \sum\limits_{a_{1j}<0} a_{1j}x_j \le b_1 \\ \vdots \qquad\qquad\qquad \vdots \\ \sum\limits_{a_{mj}>0} a_{mj}x_j + \sum\limits_{a_{mj}<0} a_{mj}x_j \le b_m \\ --------------- \\ 0 \le l_1 \le x_1 \le u_1 \le \infty \\ \quad \vdots \quad \vdots \\ 0 \le l_n \le x_n \le u_n \le \infty \end{cases}$$

Infizibilitási teszt: kielégíthetetlen feltétel keresése

Egy feltétel kielégíthetetlen (és ezáltal a probléma megoldhatatlan), ha baloldalának egy alsó korlátja nagyobb, mint a jobboldala. Formálisan, az i feltétel kielégíthetetlen, ha

$$\sum_{a_{ij}>0} a_{ij}l_j + \sum_{a_{ij}<0} a_{ij}u_j > b_i$$

Megjegyezzük, hogy egyenlőség esetén a feltétel az összes változót a megfelelő korlátján rögzíti, így egyetlen lehetséges megoldást ad, amit csak ellenőrizni kell.

Redundancia teszt: elhagyható feltétel keresése

Egy feltétel elhagyható (és ezáltal a probléma mérete csökkenthető), ha baloldalának egy felső korlátja nem nagyobb, mint a jobboldala. Formálisan, az i feltétel elhagyható, ha

$$\sum_{a_{ij}>0} a_{ij}u_j + \sum_{a_{ij}<0} a_{ij}l_j \le b_i$$

Megjegyezzük, hogy ha a jobboldal nagyobb, mint a felső korlát, akkor a feltétel a probléma minden lehetséges megoldására nézve inaktív.

Változók értékének optimális rögzítése

Ha az LP-probléma eredeti alakjában, az együtthatók mátrixának egy adott változóhoz tartozó oszlopában valamennyi együttható negatív, ugyanakkor a változó célegyütthatója pozitív, akkor a változó optimális értéke annak felső korlátján rögzíthető. Megfordítva: ha az együtthatók mátrixának egy adott változóhoz tartozó oszlopában valamennyi együttható pozitív, ugyanakkor a változó célegyütthatója negatív, akkor a változó optimális értéke annak alsó korlátján rögzíthető.

A változók értékének optimális rögzíthetősége egy nagyon hatékony vizsgálat és jelentős előrelépés a probléma méretének csökkentése irányában, ezért ezt akár legelső lépésként is végrehajthatjuk a prekondicionálásban.

Változók korlátainak erősítése

Egy feltétel minden nemnulla együtthatójú változójára nézve korlátot generál, mégpedig felső vagy alsó korlátot rendre aszerint, hogy az adott változó együtthatója pozitív vagy negatív. Formálisan, az *i* feltétel a *k* változóra nézve a következő korlátot generálja:

Ha $a_{ik} > 0$, akkor

$$x_k \le \frac{1}{a_{ik}}\left[b_i - \sum_{\substack{j \ne k \\ a_{ij} > 0}} a_{ij}x_j - \sum_{\substack{j \ne k \\ a_{ij} < 0}} a_{ij}x_j\right] \le \frac{1}{a_{ik}}\left[b_i - \sum_{\substack{j \ne k \\ a_{ij} > 0}} a_{ij}l_j - \sum_{\substack{j \ne k \\ a_{ij} < 0}} a_{ij}u_j\right]$$

felső korlátja x_k -nak.

Ha $a_{ik} < 0$, akkor

$$x_k \ge \frac{1}{a_{ik}}\left[b_i - \sum_{\substack{j \ne k \\ a_{ij} > 0}} a_{ij}x_j - \sum_{\substack{j \ne k \\ a_{ij} < 0}} a_{ij}x_j\right] \ge \frac{1}{a_{ik}}\left[b_i - \sum_{\substack{j \ne k \\ a_{ij} > 0}} a_{ij}l_j - \sum_{\substack{j \ne k \\ a_{ij} < 0}} a_{ij}u_j\right]$$

alsó korlátja x_k -nak.

Amennyiben valamelyik feltétel által generált korlát jobb, mint a változó eddig ismert korlátja, akkor az a korlát javítható. Egy korlát erősítése után a prekondicionálás eddigi lépései rekurzívan megismételhetők, célszerűen csak azokra az esetekre nézve, amelyekben az adott korlát előfordul.

Vegyük észre, hogy a korlátok generálásánál (a kapcsos zárójelben) az aktuális feltétel baloldalának alsó korlátját előállító kifejezésből rendre elhagytuk az aktuális változóhoz tartozó tagot, és az így kapott összeget vontuk ki az aktuális feltétel jobboldalából. Mivel az alsó korlátokat az infizibilitási teszt során már egyszer előállítottuk, célszerű azokat menteni, hogy aztán a korlátok generálását gyorsabban elvégezhessük.

Példa

Feladat

Hajtsuk végre a prekondicionálás lépéseit a következő LP-problémán rekurzívan addig, amíg minden teszt inkonkluzívvá nem válik.

$$\begin{cases} 2x_1 + x_2 - x_3 \to \max \\ ------------- \\ 5x_1 - 2x_2 + 8x_3 \le 15 \\ 8x_1 + 3x_2 - x_3 \ge 9 \\ x_1 + x_2 + x_3 \le 6 \\ ------------- \\ 0 \le x_1 \le 3 \\ 0 \le x_2 \le 1 \\ 1 \le x_3 \le \infty \end{cases}$$

Megoldás

1) Hozzuk a feladatot standard alakra megszorozva a második feltételt (-1) -gyel:

$$\begin{cases} 2x_1 + x_2 - x_3 \to \max \\ ------------- \\ 5x_1 - 2x_2 + 8x_3 \le 15 \\ -8x_1 - 3x_2 + x_3 \le -9 \\ x_1 + x_2 + x_3 \le 6 \\ ------------- \\ 0 \le x_1 \le 3 \\ 0 \le x_2 \le 1 \\ 1 \le x_3 \le \infty \end{cases}$$

2) Hajtsuk végre a prekondicionálás lépéseit:

- Infizibilitási teszt: inkonkluzív.
- Redundancia teszt: inkonkluzív.
- Változók optimális rögzítése: inkonkluzív.
- Változók korlátainak erősítése:

$$a_{11} = 5 > 0 \Rightarrow x_1 \le \frac{1}{a_{11}}\left[b_1 - a_{12}u_2 - a_{13}l_3\right] = \frac{1}{5}\left[15 - (-2)(1) - (8)(1)\right] = \frac{9}{5}$$

Ez a felső korlát kisebb, mint az eddigi legjobb, így u_1 azonnal javítható!

Legyen $u_1 = \frac{9}{5}$, és kezdjük elölről a prekondicionálást!

3) Hajtsuk végre a prekondicionálás lépéseit:

- Infizibilitási teszt: inkonkluzív.
- Redundancia teszt: inkonkluzív.
- Változók optimális rögzítése: inkonkluzív.
- Változók korlátainak erősítése:

$$a_{11} = 5 > 0 \Rightarrow x_1 \le \frac{1}{a_{11}}\left[b_1 - a_{12}u_2 - a_{13}l_3\right] = \frac{1}{5}\left[15 - (-2)(1) - (8)(1)\right] = \frac{9}{5}$$

$$a_{12} = -2 < 0 \Rightarrow x_2 \ge \frac{1}{a_{12}}\left[b_1 - a_{11}l_1 - a_{13}l_3\right] = \frac{1}{5}\left[15 - (5)(0) - (8)(1)\right] = -\frac{7}{2}$$

$$a_{13} = 8 > 0 \Rightarrow x_3 \le \frac{1}{a_{13}}\left[b_1 - a_{11}l_1 - a_{12}u_2\right] = \frac{1}{5}\left[15 - (5)(0) - (-2)(1)\right] = \frac{17}{8}$$

Ez a felső korlát kisebb, mint az eddigi legjobb, így u_3 javítható!

Legyen $u_3 = \frac{17}{8}$, és kezdjük elölről a prekondicionálást!

4) Hajtsuk végre a prekondicionálás lépéseit:

- Infizibilitási teszt: inkonkluzív.
- Redundancia teszt: a harmadik feltételre konkluzív, így az elhagyható.

A redukált probléma:

$$\begin{cases} 2x_1 + x_2 - x_3 \to \max \\ -------------- \\ 5x_1 - 2x_2 + 8x_3 \le 15 \\ -8x_1 - 3x_2 + x_3 \le -9 \\ -------------- \\ 0 \le x_1 \le \frac{9}{5} \\ 0 \le x_2 \le 1 \\ 1 \le x_3 \le \frac{17}{8} \end{cases}$$

- Változók optimális rögzítése: a probléma redukált modelljében a második és harmadik változó optimálisan rögzíthető $x_2 = u_2 = 1$ és $x_3 = l_3 = 1$ értéken, amiből azután az első változóra már adódik az $x_1 = u_1 = \frac{9}{5}$ optimális megoldás.

Prekondicionálás bináris változók esetén

Amennyiben a standard LP-probléma megoldását bináris változókon keressük, azaz 0-1 programozási feladatot kívánunk megoldani, úgy a modell alkalmas bővítésével a prekondicionálás lépései rendkívüli módon leegyszerűsödnek. A modell alkalmas bővítése pedig nem más, mint a változók komplementereinek a bevezetése azoknál a tagoknál, ahol az eredeti változó együtthatója negatív. A komplementer változók bevezetésével a standard alakra hozott 0-1 programozási modell minden együtthatója nemnegatívvá tehető.

$$\begin{cases} c_1x_1 + \bar{c}_1\bar{x}_1 + \ldots + c_nx_n + \bar{c}_n\bar{x}_n \to \max \\ -------------------- \\ a_{11}x_1 + \bar{a}_{11}\bar{x}_1 + \ldots + a_{1n}x_n + \bar{a}_{1n}\bar{x}_n \le b_1 \\ \vdots \qquad\qquad\qquad \vdots \\ a_{m1}x_1 + \bar{a}_{m1}\bar{x}_1 + \ldots + a_{mn}x_n + \bar{a}_{mn}\bar{x}_n \le b_m \\ -------------------- \\ x_1,\ldots,x_n \in \{0,1\} \\ \bar{x}_j = 1 - x_j \quad (j = 1,\ldots,n) \\ c_j, \bar{c}_j \ge 0 \quad (j = 1,\ldots,n) \\ a_{ij}, \bar{a}_{ij} \ge 0 \quad (i = 1,\ldots,m;\ j = 1,\ldots,n) \end{cases}$$

Vegyük észre, hogy a fenti modellben a komplementer változók megfelelő együtthatói közül legalább az egyik biztosan 0, így a célfüggvény és a feltételek pozitív együtthatójú tagjainak a száma nem több, mint az eredeti modell nemnulla együtthatóinak a száma.

Infizibilitási teszt: kielégíthetetlen feltétel keresése

Egy feltétel kielégíthetetlen (és ezáltal a probléma megoldhatatlan), ha a jobboldala negatív. Formálisan, az *i* feltétel kielégíthetetlen, ha $0 > b_i$.

Redundancia teszt: elhagyható feltétel keresése

Egy feltétel elhagyható (és ezáltal a probléma mérete csökkenthető), ha baloldali együtthatóinak összege nem nagyobb, mint a jobboldala. Formálisan, az *i* feltétel elhagyható, ha

$$\sum_{j=1}^{n}\left(a_{ij} + \bar{a}_{ij}\right) \le b_i$$

Változók korlátainak erősítése, változók optimális rögzítése

Vegyük észre, hogy a nemnegatív együtthatójú 0-1 programozási modellben a feltételek kizárólag egy változó felső korlátjának a javítására adhatnak lehetőséget. Ha pedig az lehetséges, akkor az nem más, mint a bináris változó optimális értékének a rögzítése a 0 helyen. Formálisan, az i feltétel a k változót a 0 optimális értéken rögzíti, ha $\frac{b_i}{a_{ik}} < 1$.

Megjegyezzük, hogy egy változó rögzítése a komplementerének az értékét is rögzíti.

Bináris prekondicionálásra épülő megoldó algoritmus

Bináris változók esetén a prekondicionálásra egyszerű megoldó algoritmus építhető ágaztatással, a következő elv szerint. Amennyiben a prekondicionálási tesztek valamelyike konkluzív, akkor a probléma mérete csökken, amennyiben viszont minden teszt inkonkluzív, akkor egy (még nem rögzített) bináris változó lehetséges értékei mentén a feladatot két kisebb méretű problémára bontjuk fel, amelyeket külön-külön megoldunk ugyanezen elv alapján.

Bináris prekondicionálás kiterjesztése konfliktus-gráffal

Lehetséges, hogy a 0-1 programozási modellen végrehajtott standard prekondicionálási tesztek inkonkluzívak, mégis könnyen kimutatható bizonyos változók értékeinek erős összefüggése a feltételek alapján. Az erős összefüggés kimutatása itt azt jelenti, hogy változópárok rögzítése révén a feltételek között ellentmondást detektálunk, ezáltal a változók vizsgált értékét egymást kizárónak állapítjuk meg. Ezek a változópárok értékére vonatkozó kizárási információk, amelyeket a változópárok közötti konfliktusnak nevezünk, elvezethetnek bizonyos változók értékének rögzítéséig, amivel a probléma mérete tovább csökken. A változópárok értékére vonatkozó kizárási információt, ami lényegében egy reláció a változók felett, egy speciális irányítatlan gráffal reprezentáljuk, amelyet konfliktus-gráfnak nevezünk.

Konfliktus-gráf

(Egyenlőtlenség-feltételes) konfliktus-gráf, NAND gráf

(Egyenlőtlenség-feltételes) konfliktus-gráfot úgy szerkesztünk, hogy a változópárokat az 1 értéken rögzítjük, és ha ez kielégíthetetlen feltételrendszert eredményez, akkor az konfliktust jelent.

Formálisan: a $\Gamma_{\leq} = \left(V, E_{\leq}\right)$ irányítatlan gráf akkor és csak akkor konfliktus-gráfja a komplementer-változókkal bővített standard alakú 0-1 programozási modellnek, ha $V = \left\{x_j \mid 1 \leq j \leq n\right\} \cup \left\{\bar{x}_j \mid 1 \leq j \leq n\right\}$ és $\{x_j, x_k\} \in E_{\leq} \Rightarrow x_j + x_k \leq 1$.

A konfliktus-gráf pontjait tehát a változók és azok komplementerei alkotják. Ha pedig a konfliktus-gráf két pontja között él fut, akkor a vonatkozó két változó értéke nem lehet egyszerre 1.

Megjegyzések

1. Vegyük észre, hogy egy konfliktus-gráf nem tartalmazza szükségképpen az összes konfliktust, információtartalma tehát nem feltétlenül teljes.

Ez azt jelenti, hogy egy problémának több különböző konfliktus-gráfja is lehet, speciálisan: az üres gráf biztosan konfliktus-gráf.

Könnyű megmutatni, hogy konfliktus-gráfok él-uniója is konfliktus-gráfot eredményez.

2. Az (egyenlőtlenség-feltételes) konfliktus-gráfban detektált **klikk** egy mély kombinatorikus vágást jelent a problémára nézve.

3. Létezik **egyenlőség-feltételes konfliktus-gráf** (XOR-gráf) is, amelynek élei a változópárok egyenlő értékének a kizárását jelentik: $\{x_j, x_k\} \in E_{=} \Rightarrow x_j + x_k = 1$

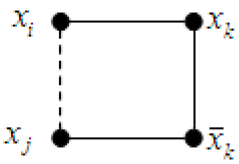
Könnyű megmutatni, hogy minden egyenlőség-feltételes konfliktus-gráf egyben egyenlőtlenség-feltételes is, visszafelé azonban ez általában nem igaz: $E_{=} \subseteq E_{\leq}$

A teljes egyenlőség-feltételes konfliktus-gráf tehát mindig részgráfja a teljes egyenlőtlenség-feltételes konfliktus-gráfnak.

4. Vegyük észre, hogy mindkét típusú konfliktus-gráf tartalmazza a változók és komplementereik között futó éleket: $\{x_k, \bar{x}_k\} \in E_{=} \cap E_{\leq}$

Bővítési szabály

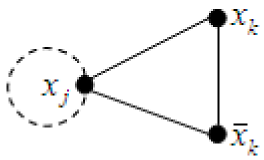
Egy konfliktus-gráfban éllel összeköthetők azok a csúcsok, amelyekből él fut egy változóhoz és annak komplementeréhez is: $\{x_i, x_k\}, \{x_j, \bar{x}_k\} \in E_{\leq} \Rightarrow \{x_i, x_j\} \in E_{\leq}$



A bővítési szabály kiterjeszthető, ha az egyenlőség-feltételes konfliktus gráf információi is a rendelkezésünkre állnak. Eszerint, az egyenlőtlenség-feltételes gráfban éllel összeköthetők azok a csúcsok, amelyekből él fut egy olyan változópárhoz, akik az egyenlőség-feltételes konfliktus gráfban éllel vannak összekötve: $\{x_i, x_k\}, \{x_j, x_l\} \in E_{\leq} \wedge \{x_k, x_l\} \in E_{=} \Rightarrow \{x_i, x_j\} \in E_{\leq}$

Rögzítési szabály

A bővítési szabály speciális esete, amikor egy változóhoz és annak komplementeréhez is ugyanattól a harmadik csúcstól fut él, azaz a gráf olyan 3 pontú teljes gráfot (3-klikket) tartalmaz, amelyben egy változó és komplementere is szerepel. Ebben az esetben a harmadik csúcs változóját a 0 értéken rögzíthetjük: $\{x_j, x_k\}, \{x_j, \bar{x}_k\} \in E_{\leq} \Rightarrow x_j = 0$



Az előzőhöz hasonló módon a rögzítési szabály is kiterjeszthető, ha az egyenlőség-feltételes konfliktus gráf információi is a rendelkezésünkre állnak: $\{x_j, x_k\}, \{x_j, x_l\} \in E_{\leq} \wedge \{x_k, x_l\} \in E_{=} \Rightarrow x_j = 0$

Egy változó rögzítése azt jelenti, hogy a neki megfelelő csúcs az (egyenlőtlenség-feltételes) konfliktus gráfban maximális fokú, telített.

Példa

Feladat (Vízvári Béla)

Bizonyítsuk be, hogy az alábbi feladatnak a $(0,0,1,1,1,1,1)$ vektor egy optimális megoldása!

$$\begin{cases} 15x_1 + 12x_2 + 10x_3 + 9x_4 + 7x_5 + 4x_6 + 3x_7 \to \max \\ ------------------------------ \\ 4x_1 + 4x_2 + 3x_3 + 2x_4 + 2x_5 + 2x_6 + x_7 \le 10 \\ 8x_1 + 3x_2 + 6x_3 + 5x_4 - 3x_5 - 4x_6 - 5x_7 \le 2 \\ -------------------------- \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7 \in \{0,1\} \end{cases}$$

Megoldás

A $(0,0,1,1,1,1,1)$ vektor esetében a feltételek teljesülnek (az első aktív, a második inaktív); a célfüggvény értéke 33.

Alakítsuk át a célfüggvény sorát egyenlőtlenséggé, írjunk az egyenlőtlenség jobb oldalára 34-et, majd – szükség esetén konfliktus-gráffal kiterjesztett – prekondicionálással mutassuk meg, hogy az így kapott feltételrendszer kielégíthetetlen.

$$\begin{cases} 15x_1 + 12x_2 + 10x_3 + 9x_4 + 7x_5 + 4x_6 + 3x_7 \ge 34 \\ 4x_1 + 4x_2 + 3x_3 + 2x_4 + 2x_5 + 2x_6 + x_7 \le 10 \\ 8x_1 + 3x_2 + 6x_3 + 5x_4 - 3x_5 - 4x_6 - 5x_7 \le 2 \end{cases}$$

A feltételeket tegyük egyirányúvá, és a komplementer változók bevezetésével minden együtthatót alakítsunk nemnegatívvá:

$$\begin{cases} 15\bar{x}_1 + 12\bar{x}_2 + 10\bar{x}_3 + 9\bar{x}_4 + 7\bar{x}_5 + 4\bar{x}_6 + 3\bar{x}_7 \le 26 \\ 4x_1 + 4x_2 + 3x_3 + 2x_4 + 2x_5 + 2x_6 + x_7 \le 10 \\ 8x_1 + 3x_2 + 6x_3 + 5x_4 + 3\bar{x}_5 + 4\bar{x}_6 + 5\bar{x}_7 \le 14 \end{cases}$$

A fenti feltételrendszeren a prekondicionálási tesztek inkonkluzívak, és egyik feltétel sem rögzít változót.

A változópárok értékének hipotetikus rögzítésével és a feltételek újraellenőrzésével szerkesszük meg a probléma konfliktus-gráfjának adjacencia-mátrixát!

Az első feltétel az első két változó beállítására azonnal ellentmondásos, így $\{\bar{x}_1, \bar{x}_2\} \in E_{\le}$

Az első feltételben rögzítsük az első és harmadik változót 1 értéken: $\bar{x}_1 = \bar{x}_3 = 1$; ekkor a többi változó értéke már csak $\bar{x}_2 = \bar{x}_4 = \bar{x}_5 = \bar{x}_6 = \bar{x}_7 = 0$ lehet. Az így rögzített változók azonban nem elégítik ki a második feltételt, amiből

$\{\overline{x}_1, \overline{x}_3\} \in E_{\leq}$ következik. Ezt a gondolatsort a következőképpen rövidíthetjük: $1F : \overline{x}_1 = \overline{x}_3 = 1 \Rightarrow \overline{x}_2 = \overline{x}_4 = \overline{x}_5 = \overline{x}_6 = \overline{x}_7 = 0 \leftrightarrow 2F$

További változópárok tesztelésével a következő eredményekre jutunk:

- $1F : \overline{x}_1 = \overline{x}_4 = 1 \Rightarrow \overline{x}_2 = \overline{x}_3 = \overline{x}_5 = \overline{x}_6 = \overline{x}_7 = 0 \leftrightarrow 2F$
- $1F : \overline{x}_1 = \overline{x}_5 = 1 \Rightarrow \overline{x}_2 = \overline{x}_3 = \overline{x}_4 = 0 \Rightarrow 2F : x_6 = 0 \leftrightarrow 1F$
- $1F : \overline{x}_2 = \overline{x}_3 = 1 \Rightarrow \overline{x}_1 = \overline{x}_4 = \overline{x}_5 = 0 \Rightarrow 1F : \overline{x}_6 = \overline{x}_7 = 0 \leftrightarrow 2F$
- $3F : x_1 = x_3 = 1 \Rightarrow x_2 = x_4 = \overline{x}_5 = \overline{x}_6 = \overline{x}_7 = 0 \leftrightarrow 2F$
- $3F : x_1 = x_4 = 1 \Rightarrow x_2 = x_3 = \overline{x}_5 = \overline{x}_6 = \overline{x}_7 = 0 \leftrightarrow 2F$
- $3F : x_1 = \overline{x}_7 = 1 \Rightarrow x_2 = x_3 = x_4 = \overline{x}_5 = \overline{x}_6 = 0 \leftrightarrow 1F$
- $3F : x_3 = x_4 = 1 \Rightarrow x_1 = \overline{x}_6 = \overline{x}_7 = 0 \Rightarrow 2F : x_2 = 0 \leftrightarrow 1F$

Ábrázoljuk a kapott eredményeket a konfliktus-gráf adjacencia-mátrixán:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\overline{x}_1$	$\overline{x}_2$	$\overline{x}_3$	$\overline{x}_4$	$\overline{x}_5$	$\overline{x}_6$	$\overline{x}_7$
x_1			•	•				•						•
x_2									•					
x_3	•			•						•				
x_4	•		•								•			
x_5												•		
x_6													•	
x_7														•
$\overline{x}_1$	•								•	•	•	•		
$\overline{x}_2$		•						•		•				
$\overline{x}_3$			•					•	•					
$\overline{x}_4$				•				•						
$\overline{x}_5$					•			•						
$\overline{x}_6$						•								
$\overline{x}_7$	•						•							

Alkalmazzuk a bővítési szabályt az első változóra (és komplementerére):

- x_1 szomszédjai: x_3, x_4, $\overline{x}_7$
- $\overline{x}_1$ szomszédjai: $\overline{x}_2$, $\overline{x}_3$, $\overline{x}_4$, $\overline{x}_5$

További élek a gráfban: $\{x_3, \overline{x}_2\}$, $\{x_3, \overline{x}_4\}$, $\{x_3, \overline{x}_5\}$, $\{x_4, \overline{x}_2\}$, $\{x_4, \overline{x}_3\}$, $\{x_4, \overline{x}_5\}$, $\{\overline{x}_7, \overline{x}_2\}$, $\{\overline{x}_7, \overline{x}_3\}$, $\{\overline{x}_7, \overline{x}_4\}$, $\{\overline{x}_7, \overline{x}_5\}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\overline{x}_1$	$\overline{x}_2$	$\overline{x}_3$	$\overline{x}_4$	$\overline{x}_5$	$\overline{x}_6$	$\overline{x}_7$
x_1			•	•				•						•
x_2									•					
x_3	•			•					•	•	•	•		
x_4	•		•						•	•	•	•		
x_5												•		
x_6													•	
x_7														•
$\overline{x}_1$	•								•	•	•	•		
$\overline{x}_2$		•	•	•				•		•				•
$\overline{x}_3$			•	•				•	•					•
$\overline{x}_4$			•	•				•						•
$\overline{x}_5$			•	•	•			•						•
$\overline{x}_6$						•								
$\overline{x}_7$	•						•		•	•	•	•		

Alkalmazzuk a bővítési szabályt a harmadik változóra (és komplementerére):

- x_3 szomszédjai: x_1, $\overline{x}_2$, x_4, $\overline{x}_4$, $\overline{x}_5$
- $\overline{x}_3$ szomszédjai: x_4, $\overline{x}_2$, $\overline{x}_1$, $\overline{x}_7$

További élek a gráfban: $\{x_1,\overline{x}_2\}$, $\{\overline{x}_2,\overline{x}_2\}$, $\{x_4,x_4\}$, $\{x_4,\overline{x}_1\}$, $\{x_4,\overline{x}_7\}$, $\{\overline{x}_4,\overline{x}_2\}$, $\{\overline{x}_5,\overline{x}_2\}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\overline{x}_1$	$\overline{x}_2$	$\overline{x}_3$	$\overline{x}_4$	$\overline{x}_5$	$\overline{x}_6$	$\overline{x}_7$
x_1			•	•				•	•					•
x_2									•					
x_3	•			•					•	•	•	•		
x_4	•		•	•				•	•	•	•	•		•
x_5												•		
x_6													•	
x_7														•
$\overline{x}_1$	•			•					•	•	•	•		
$\overline{x}_2$	•	•	•	•				•	•	•	•	•		•
$\overline{x}_3$			•	•				•	•					•
$\overline{x}_4$			•	•				•	•					•
$\overline{x}_5$			•	•	•			•	•					•
$\overline{x}_6$						•								
$\overline{x}_7$	•			•			•		•	•	•	•		

Vegyük észre, hogy hurokél keletkezett az x_4 és az $\overline{x}_2$ csúcson, ami azt jelenti, hogy ezek a változók speciális esetei voltak az előző bővítési szabálynak, és a rögzítési szabály szerint 0 értéket kapnak. A gráfban ez a rögzítés azt jelenti, hogy az x_4 és az $\overline{x}_2$ csúcs automatikusan telítetté válik.

A két változó (és komplementereik) rögzítésével csökken a feladat mérete:

$$\begin{cases} 15\overline{x}_1 + 10\overline{x}_3 + 7\overline{x}_5 + 4\overline{x}_6 + 3\overline{x}_7 \le 17 \\ 4x_1 + 3x_3 + 2x_5 + 2x_6 + x_7 \le 6 \\ 8x_1 + 6x_3 + 3\overline{x}_5 + 4\overline{x}_6 + 5\overline{x}_7 \le 11 \end{cases}$$

Vegyük észre, hogy a redukált első feltételből azonnal következnek az $\{\overline{x}_1, \overline{x}_6\}$, $\{\overline{x}_1, \overline{x}_7\}$ új élek.

A redukált második feltétel vizsgálata elvezet az $\{x_1, x_5\}$, $\{x_1, x_6\}$ élekhez.

A redukált harmadik feltétel vizsgálata alapján kapjuk a további $\{x_1, \overline{x}_6\}$ élt, amire a rögzítési szabály alapján $x_1 = 0$ következik.

Az $x_1 = 0$ rögzítésre az első feltétel csak $\overline{x}_3 = \overline{x}_5 = \overline{x}_6 = \overline{x}_7 = 0$ értékekkel elégíthető ki, az így kapott $(0,1,1,0,1,1,1)$ vektor viszont a második feltételt nem elégíti ki.

A feltételrendszer ellentmondásos, tehát a célfüggvény értéke nem növelhető, így az eredeti megoldás valóban optimális volt.