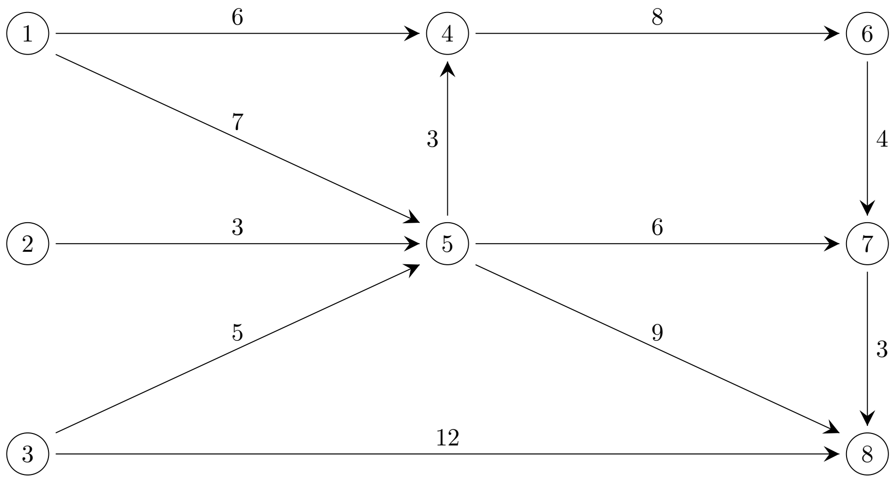
Ágoston Kolos Csaba – Gyetvai Márton

Hálózati feladatok megoldása lineáris programozási feladatok segítségével



Budapesti Corvinus Egyetem

Matematikai és Statisztikai Modellezés Intézet

lektor: Sziklai Balázs

Budapest, 2022

Elektronikus jegyzet

Kérjük, ne nyomtassa ki!

Kedves Hallgatók!

Ezen jegyzetet elektronikus jegyzetnek szánjuk. A lineáris programozási feladatok CPLEX LP kódját sok feladat esetében megadjuk. Ezek a kódok változtatás nélkül futtathatóak, megkönnyítve az eredmények reprodukálását. Ez a módszer a tanulást megkönnyíti, viszont a jegyzet méretét megnöveli. Ezért nyomatékosan kérjük Önöket, hogy ne nyomtassák ki ezt a jegyzetet, használják elektronikus változatban.

Kérjük, óvja környezetét!

Ne nyomtassa ki ezt a jegyzetet!

Hálózati feladatok megoldása lineáris programozási feladatok segítségével

Jegyzet

Ágoston Kolos Csaba, Gyetvai Márton

Budapesti Corvinus Egyetem

Matematikai és Statisztikai Modellezés Intézet

2022

ISBN 978-963-503-903-6

Tartalomjegyzék

1. fejezet

Bevezetés

Közgazdász hallgatók régóta tanulnak fejezeteket az operációkutatás tudományterület témaköreiből. Az évek során több könyv és feladatgyűjtemény is született, amelyek a hallgatókat segítik a vizsgára való felkészülés során. Ezért, amikor átvettem gazdaságinformatikus szakon az operációkutatás tárgy oktatását, úgy gondoltam, hogy nem szükséges újabb tankönyv vagy jegyzet írása, a meglévők bőségesen elegendőek.

Időközben azonban kiderült, hogy ez nem feltétlenül van így. Minden szak speciális valamilyen szempontból, más szakokhoz írt segédanyagok nem helyettesíthetik teljes mértékben a szükséges, releváns jegyzeteket.

A gazdaságinformatikus szak abból a szempontból speciális, hogy az operációkutatás során oktatott algoritmusokat az ide felvett hallgatók más tárgyakból már tanulták. Tehát ezen algoritmusok megismertetése nem kell, hogy a félév anyagát képezze. Érdemes helyette viszont ezen feladatok LP felírásait alaposabban tanulmányozni. Emellett külön hangsúly van a géptermi oktatáson: a felírt feladatok esetén kíváncsiak vagyunk az optimális megoldásra (akár nagyméretűekére is). A bemutatandó problémák a mindennapi élet során is előkerülnek, de sokszor nem tiszta formájukban. Az összes lehetséges bővítés ismertetése lehetetlen vállalkozás, a jegyzetben csak ötleteket szeretnék adni; a konkrét feladatok esetén mindenkinek saját magának kell majd az alapfeladatot a konkrét üzleti szituációhoz igazítania.

A jegyzet írásakor feltételezem a lineáris programozás elméletének, és a szimplex módszernek az ismeretét (lásd pl.: Wayne L. Winston: Operációkutatás, Aula, 2003.). A hálózati feladatok LP felírásai megtalálhatóak a szokásos bevezető operációkutatással foglalkozó tankönyvekben (lásd pl. ugyanott), mégis visszatérő igény, hogy ezeket összefoglalva, kicsit részletesebben tárgyalva is kézhez kaphassák a hallgatók. Reményeim szerint ezt tartalmazza a jegyzet.

A felírt modelleket meg is szeretnénk oldani, ehhez megoldóra (solver) van szükségünk. A legjobb programok (Cplex, Gurobi) kereskedelmi termékek, amelyeknek komoly áruk van. A kereskedelmi termékek mellett vannak 'open source' programok is, ilyenek pl. a CBC és a GLPK. Mi a GLPK használata mellett döntöttünk, két okból: egyrészt a hallgatók ezt a programot később is használni tudják jogi/erkölcsi aggályok nélkül, másrészt az alkalmazás könnyűszerrel beágyazható más rendszerekbe is. Az LP feladatokat ún. CPLEX LP formátumban írtuk fel. Ezt a formátumot a legtöbb megoldó ismeri, tehát a jegyzetben szereplő kódok más programokkal is futtathatóak, de elképzelhető, hogy más struktúrájú outputot fog kapni a felhasználó. A GLPK használata esetén az output állomány tartalmazza a döntési változók optimális értékeit, információkat a korlátokról és a duálváltozók optimális értékeit. Néha szükségünk lesz az érzékenységvizsgálat egyéb összetevőire is (érzékenységvizsgálati intervallumok), amit az ún. LP Sensitivity Analysis tartalmaz.

A jegyzetben a GLPK program inputjait és outputjait `verbatim` stílusban adjuk meg. Nehézséget jelent a tizedesvessző kérdése. Folyó szövegben a magyar helyesírás szabályainak megfelelően tizedesvesszőt használunk, de a GLPK program inputjai tizedespontot követelnek meg és az outputok is tizedespontot tartalmaznak. Ezt nem változtattuk meg, tehát `verbatim` stílusban szedett szöveg esetén tizedespontot fog találni az olvasó.

A jegyzetet egyedül kezdtem írni, de a munka meghaladta a rendelkezésemre álló erőforrásokat. Gyetvai Márton felajánlotta segítségét, amit én örömmel elfogadtam.

Kettőnk nevében szeretném megköszönni feleségem, Ágoston Andrea segítségét a szöveg korrektúrájáért. Szintén szeretnénk megköszönni Sziklai Balázsnak a jegyzet

lektorálását. Igyekeztünk az elvárható gondossággal eljárni a jegyzet írásakor, ha valami hiba/pontatlanság maradt a szövegben az nekünk róható fel.

Ágoston Kolos Csaba

Budapest, 2022. március 3.

2. fejezet

Lineáris programozás emlékeztető

Mint a bevezetőben említettük a jegyzet olvasásához feltételezzük, hogy a hallgatók ismerik az LP programozás elméletét. Ebben a témában nagyon sok jó könyv érhető el (lásd pl.: Wayne L. Winston: Operációkutatás, Aula, 2003.), ezért a téma átfogó ismertetését kontraproduktívnek tartjuk. Ebben a fejezetben csak arra vállalkozunk, hogy a legfontosabb fogalmakra emlékeztetjük az olvasót. Emellett a GLPK/GUSEK program használatát is bemutatjuk.

A tárgyalni kívánt fogalmakat talán egy termelési példán keresztül tudjuk a legszemléletesebben bemutatni.

2.1. példa. *Van egy vállalkozásunk, ami 3 különböző terméket állít elő, legyenek x_1, x_2 és x_3 (a termékeket most nem nevesítjük). A termékek előállításához erőforrásokra van szükségünk (pl.: nyersanyag, munkaerő, energia, stb ...), de a szükszéges erőforrásokat sem nevesítjük. Az erőforrások korlátozott mértékben állnak rendelkezésre, az első erőforrásból 10000, a másodikból 25000, a harmadikból pedig 110000 egység áll rendelkezésre. Ismert továbbá, hogy a termékek előállításához az erőforrásokból mennyit kell elhasználni. Ezeket az értékeket egy mátrixba rendezzük:*

$$A = \begin{pmatrix} 11 & 30 & 39 \\ 41 & 60 & 73 \\ 112 & 135 & 194 \end{pmatrix},$$

tehát az első erőforrásból az első termék előállításához 11 egység kell, a második *erőforrásból 41, a harmadikból pedig 112. Hasonlóan a második termék előállításához az első erőforrásból 30 egység kell, a másodikból 60, a harmadikból pedig 135, stb...*

Az első termék darabonként 21 profitot eredményez, a második 32-t, a harmadik pedig 50-t. A feladat innentől kezdve egyértelmű: szeretnénk a maximális profitot eredményező termelési tervet előállítani.

Megoldás.

Könnyen fel tudjuk írni a probléma LP modelljét:

2.2. LP felírás.

$$\begin{array}{rcl} 21x_1 + 32x_2 + 50x_3 & \rightarrow & \max \\ \textit{feltéve hogy:} & & \\ 11x_1 + 30x_2 + 39x_3 & \leq & 10000 \\ 41x_1 + 60x_2 + 73x_3 & \leq & 25000 \\ 112x_1 + 135x_2 + 194x_3 & \leq & 110000 \\ x_1, x_2, x_3 & \geq & 0 \end{array}$$

A lineáris programozási feladatot néha tömörebb formában is felírjuk. A döntési változókat az $\mathbf{x}$ vektorba rendezzük, a célfüggvény együtthatókat (jelen példában a profitokat) a $\mathbf{c}$ vektorba, a kapacitásokat (az erőforrások rendelkezésre álló mennyiségét) pedig a $\mathbf{b}$ vektorba. Tehát

$$\mathbf{c} = \begin{pmatrix} 21 \\ 32 \\ 50 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 10000 \\ 25000 \\ 110000 \end{pmatrix}.$$

Használva a bevezetett jelöléseket a 2.2. LP felírást tömörebben is megfogalmazhatjuk:

2.3. LP felírás.

$$\begin{array}{rcl} \mathbf{c}^T\mathbf{x} & \rightarrow & \max \\ \textit{feltéve hogy:} & & \\ A\mathbf{x} & \leq & \mathbf{b} \\ \mathbf{x} & \geq & \mathbf{0} \end{array}.$$

2.1. A GLPK megoldó és a Gusek program használata

Oldjuk meg a 2.2. LP felíráshoz tartozó feladatot. LP feladatokat jellemzően szimplex módszerrel oldunk meg. Régebbi időkben még előfordult, hogy a szimplex módszert kézzel, papíron számolták végig, manapság ez már elképzelhetetlen. Több program is elérhető, ami képes LP feladatot megoldani, mi ezek közül a GLPK-t választottuk (`https://www.gnu.org/software/glpk/`). A választásban az motivált minket, hogy a GLPK ún. szabad szoftver, amelyet a hallgatók későbbi munkájuk során is használhatnak majd, emellett a GLPK megoldó könnyen implementálható más rendszerekbe is. Emellett hangsúlyozzuk azt is, hogy a GLPK nem a leghatékonyabb megoldó, nagyméretű, egészértékű változókat is tartalmazó modellek esetén érezhetően hosszabb időbe telik az optimális megoldás megtalálása, mint a kereskedelmi programok (pl. CPLEX, Gurobi) esetén. A GLPK alapvetően egy parancssorból vezérelhető program, a Gusek (`http://gusek.sourceforge.net/gusek.html`) program a GLPK használata kényelmesebbé tehető.

A GLPK számára több formátumban is meg lehet adni a modellt. Mi a lehetséges változatok közül a CPLEX LP formátumot használjuk. Amint a neve is mutatja ezt a formátumot a CPLEX programhoz fejlesztették ki, de mára sztenderddé vált, az LP megoldók általában ismerik ezt a formátumot. A CPLEX LP kód ráadásul nagyon intuitív.

A CLPLEX LP kód ún. szekciókra oszlik, amelyek kulcsszavakkal vannak bevezetve. A szóközök tekintetében elég rugalmas a CPLEX LP formátum, de a kulcsszavaknak mindig a sor elején kell kezdődniük. További fontos észrevétel, hogy a CPLEX LP formátum *'case sensitive'*, azaz a kis és nagybetű két különböző karakter. Az első szekció a célfüggvény, amelyet a `min` vagy `max` kulcsszó vezet be (további lehetséges kulcsszavak: a `min`-nel egyenértékű a `minimize` és a `minimum`; illetve a `max`-szal a `maximize` és a `maximum`). A kulcsszó után következik egy lineáris kifejezés scx formában, ahol s egy előjel, azaz egy + vagy − jel; c egy konstans (szám),

x a változó neve. A 2.2. LP felírás esetén a célfüggvény: `21x1 + 32x2 + 50x3`. A GLPK rugalmasan kezeli a szóközöket/sortöréseket, a mi esetünkben a célfüggvény lehetne a `21x1+32x2+50x3` vagy a `21 x1 + 32 x2 + 50 x3`, de akár a

```
21
x1
+
32

x2
+
50x3
```

is. A sor elejére a pozitív előjelet nem kell feltétlenül kitenni, de megengedett a használata.

Pár dologra érdemes fehívni a figyelmet:

- Ha nem adunk meg konstanst, akkor 1-nek tekinti, tehát az `x1+x2` kifejezés egyenértékű a `+ 1 x1 + 1 x2` kifejezéssel.
- A CPLEX LP formátumban nem tudjuk kiértékelni a kifejezéseket. Tehát a $2(x_1 + x_2)$ kifejezés lehetne célfüggvény, de a `2(x_1+x_2)` szöveg szintaktikailag helytelen, pontosabban ezt a szöveget úgy értelmezi a GLPK, hogy van egy változónk, aminek "`(x_1`" a neve, ennek 2 a célfüggvény együtthatója, van egy másik változónk, aminek "`x_2)`" a neve és nincs megadva a célfüggvény együtthatója, tehát 1-nek tekinti.
- A $+ - x_1$ kifejezés matematikailag értelmezhető, de ez szintaktikailag hibás. Ekkor a `missing variable name` hibaüzenetet kapjuk. Két előjel között mindig kell változónévnek lennie. $+ - x_1$ kifejezés helyett egyszerűen `-x1` kifejezést kell írni. Ez nem tűnik nehéznek, de ha valamilyen programmal (akár MS Excellel) generáljuk az CPLEX LP kódot figyelni kell, mert a programok a pozitív számok elé nem teszik ki az előjelet, a negatív számok elé viszont kiteszik.
- Egy változó egy kifejezésben egyszer szerepelhet csak. Az $x_1 + 3x_2 + 4x_1$ kifejezéssel sincs semmi probléma matematikai

szövegben, de az `x1+3x2+4x1` kifejezés használata esetén a `multiple use of variable 'x1' not allowed` hibaüzenetet kapjuk.

- Mint hangsúlyoztuk, a CPLEX LP 'case sensitive', tehát az `x1` és `X1` két külön változó. Az `x1+3x2+4x1` kifejezés használata esetén hibaüzenetet kapunk, de az `x1+3x2+4X1` kifejezés szintaktikailag helyes, igaz ugyan, hogy ekkor 3 változó szerepel a kifejezésben: `x1`, `X1` és `x2`.
- A CPLEX LP formátumban tizedespontot kell használni: `2.1x1`. Ha tizedesvesszőt használunk (`2,1x1`) használunk, akkor a szöveg szintaktikailag helyes lesz, de úgy 'értelmezi', hogy van egy változónk, aminek `,1x1` a neve, ennek a változónak az együtthatója pedig 2.

A következő szekció a korlátokat írja le. Ezt a szekciót a `subject to` kulcsszó vezeti be (megengedettek még a `such that`, `s.t.`, `st.` és a `st` kulcsszavak). A korlátok is egy lineáris kifejezéssel kezdődnek (ugyanazok a szabályok érvényesek, mint amit a célfüggvény esetén leírtunk). A lineáris kifejezés után egy relációjel szerepel (<=, = vagy >=), majd egy konstans. Megengedett, hogy a <= helyett =< vagy akár < karaktereket használjunk, de a < karaktert is $\leq$ relációnak fogja értelmezni a fordító.

A mi esetünkben a korlátokat leíró szekció a következő:

```
subject to

11x1  +  30x2 +  39x3 <=  10000
41x1  +  60x2 +  73x3 <=  25000
112x1 + 135x2 + 194x3 <= 110000
```

Az előforduló hibákat a célfüggvény megadásánál már leírtuk, a korlátok esetén ezeken túlmenően egy újabb hiba is előfordul: matematikailag helyes az $x_1 \leq x_2$ kifejezés, de az `x1<=x2` kifejezés használata esetén a `missing right-hand side` hibaüzenetet fogjuk kapni. A CPLEX LP formátum esetén a változókat egy oldalra kell rendezni, a 'jobboldalon' csak konstans lehet. Tehát az `x1<=x2` szöveg helyett az `x1-x2<=0` szöveget kell használni (de használhatnánk a `-x1+x2>=0` változatot is).

A `bound` szekcióban lehet a változókra határokat megadni. A mi esetünkben a változók nemnegatívak, amit pl.: `x1>=0` módon rögzíthetünk. Ha a `bound` szekció hiányzik (vagy valamely változókra nem szerepel korlát), akkor alapértelmezésként nemnegatív változót fog definiálni. Ez a példafeladathoz pont megfelelő, ezért nem szerepel a modellben `bound` szekció. Amennyiben van egészértékű vagy bináris változó a modellben, ezeket a változókat az `integer` vagy `binary` szekcióban fel kell sorolni, de erre később még visszatérünk.

A modellünket az `end` kulcsszóval zárjuk le. A kulcsszó után még sortörést is be kell szúrni, különben a `missing final end of line` üzenetet kapjuk. Ez nem hibaüzenet, a kód értelmezhető még ettől.

A 2.2. LP felírás CPLEX LP formátumban:

2.4. kód.

```
max
21x1 + 32x2 + 50x3

subject to
11x1  +  30x2 +  39x3 <=  10000
41x1  +  60x2 +  73x3 <=  25000
112x1 + 135x2 + 194x3 <= 110000

end
```

Most már csak az van hátra, hogy lefuttassuk ezt a kódot. Töltsük le a GLPK programot. A 2.4 kódot mentsük el egy text állományba, pl. model.txt. Adjuk ki az alábbi parancsot (ha a glpsol parancs nincs benne az alapértelmezett elérési utakban, akkor teljes elérési utat kell használni):

`glpsol.exe --cpxlp --output output.txt model.txt.`

Ekkor az eredményt a GLPK kiírja az output.txt állományba.

A parancssorok használata kényelmetlen a mai gyakorlatban, szerencsére ezt ki lehet váltani a Gusek (GLPK Under Scite Extended Kit) programmal. Ez lényegében egy szövegszerkesztő, amely a kód futtatását és az eredmény beolvasását is támogatja. Csomagoljuk ki a programot. A Gusek magában foglalja a GLPK programot is, nem kell külön telepíteni. A GLPK számára az LP modellt több módon is meg lehet adni, mi korábban a CPLEX LP formátumot ismertettük.

A Gusek számára az állomány kiterjesztéséből derül ki, hogy melyik formátumot használjuk. A CPLEX LP formátumhoz a .lp kiterjesztés tartozik, amely előtt tetszőleges modellnév állhat.

Továbbá fontos beállítás indítás után, hogy a modell futtatásának eredményét is szeretnénk majd látni. Ehhez a Tools menüben klikkeljünk az ‘Generate Output File on Go’ szövegre (ezt csak akkor tudjuk megtenni, ha az aktív ablak egy mentett .lp állomány, az LP modellek mentéséről a következő bekezdésben ejtünk szót). Ha utána még egyszer a Tools menüt megtekintjük, akkor a szöveg mellett látszik a kék pipa.

A program első indításakor egy üres felületet látunk magunk előtt. Első lépésként kell egy új modellt kérnünk (File menü → New parancs vagy CTRL+N billentyűkombináció). Célszerű már az üres ablakot a modell beillesztése előtt névvel és megfelelő kiterjesztéssel elmentenünk, tegyük ezt meg a File menü → Save As parancs vagy CTRL+SHIFT+S billentyűkombináció segítségével, és legyen ez most a termeles.lp nevű file. Másoljuk be a 2.4. kódot az termeles.lp fülre. A modellt most már lehet futtatni is. Amennyiben több fülön több modell vagy több eredményfül is van, úgy figyeljünk rá, hogy futtatáshoz az legyen aktív, amelyikre szükségünk van. Futtatni is többféleképpen lehet: Tools menü → Go parancs vagy F5 billentyű vagy a ‘tappancs’ ikon a menüsor alatt.

Ha minden rendben akkor az adott .lp fül jobb oldali osztatában megjelenik a ‘log’ fül. A mi esetünkben:

```
>C:\gusek\glpsol.exe --cover --clique --gomory --mir --lp "termeles.lp" -o "termeles.out"
GLPSOL: GLPK LP/MIP Solver, v4.60
Parameter(s) specified in the command line:
 --cover --clique --gomory --mir --lp termeles.lp -o termeles.out
Reading problem data from '.lp'...
3 rows, 3 columns, 9 non-zeros
9 lines were read
GLPK Simplex Optimizer, v4.60
3 rows, 3 columns, 9 non-zeros
Preprocessing...
3 rows, 3 columns, 9 non-zeros
Scaling...
 A: min|aij| =  1.100e+01  max|aij| =  1.940e+02  ratio =  1.764e+01
GM: min|aij| =  8.154e-01  max|aij| =  1.226e+00  ratio =  1.504e+00
EQ: min|aij| =  6.648e-01  max|aij| =  1.000e+00  ratio =  1.504e+00
Constructing initial basis...
Size of triangular part is 3
*     0: obj =  -0.000000000e+00 inf =   0.000e+00 (3)
*     2: obj =   1.494346734e+04 inf =   0.000e+00 (0)
OPTIMAL LP SOLUTION FOUND
Time used:   0.0 secs
Memory used: 0.0 Mb (29985 bytes)
Writing basic solution to 'termeles.out'...
>Exit code: 0    Time: 0.229
```

Az első sorban meghívja a Gusek a GLPK-t. Látható, hogy azzal a `glpsol` paranccsal, amelyet mi is korábban használtunk. A `--cover --clique --gomory --mir` opciók ún. metszősíkok generálását engedélyezik, amelyek hatékonyabbá teszik az egészértékű modellek megoldását. A mi modellünkben nincsenek egészértékű változók, így számunkra közömbösek ezek az opciók. A következő sorban a GLPK verziószáma látható, utána a parancssorban kiadott opciókat ismétli meg, majd a beolvasott modellről szerepel néhány adat. Utána elkezdődik a tényleges optimalizálás. Fontos, hogy nem szerepel hibaüzenet a 'log' állományban, és még fontosabb az `OPTIMAL LP SOLUTION FOUND` szöveg, ami azt takarja, hogy optimális megoldást talált a solver. Végül a futási idő szerepel másodpercben, majd a használt memória mérete.

Mint a 'log' állományból láthatjuk optimális megoldást találtunk. A Gusek az optimális megoldást rögtön be is olvassa, a termeles.out fül lesz aktív. Az fül tartalma:

```
Problem:
Rows:       3
Columns:    3
Non-zeros:  9
Status:     OPTIMAL
Objective:  obj = 14943.46734 (MAXimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 r.5          NU         10000                       10000      0.649497
     2 r.6          NU         25000                       25000       0.33794
     3 r.7          B        67374.4                      110000

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 x1           B        307.789             0
     2 x2           NL             0             0                   -7.76131
     3 x3           B        169.598             0

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err = 3.64e-12 on row 2
        max.rel.err = 7.28e-17 on row 2
        High quality

KKT.PB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality
```

```
KKT.DE: max.abs.err = 0.00e+00 on column 0
        max.rel.err = 0.00e+00 on column 0
        High quality

KKT.DB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

End of output
```

Az eredményfül (output) struktúrája a következő: az első pár sor összefoglalót tartalmaz az LP feladatról. Jelen feladatban 3 korlát szerepel (`Rows: 3`), és 3 döntési változó (`Columns: 3`). A feladat együttható mátrixában 9 nemnulla elem szerepel (`Non-zeros: 9`).

A következő sor (`Status: OPTIMAL`) nagyon lényeges, ez mutatja, hogy a megoldó talált optimális megoldást. Ha a `Status` nem `OPTIMAL`, akkor az eredményfül többi része értelmezhetetlen lenne a számunkra, meg kell keresni a hibát és újra futtatni a modellt. A következő sor a célfüggvény (optimális) értékét mutatja.

Az eredményfül további részében két táblázat szerepel, ezekben információkat kapunk a korlátokról (`Row`), és a döntési változókról (`Column`). Az eredményfül utolsó része (`Karush-Kuhn-Tucker optimality conditions`) az optimális megoldás 'minőségéről' ad információt, de a feltételek pontos ismertetése meghaladja egy bevezető operációkutatás tárgy tematikáját. Annyi látszik, hogy a mi esetünkben `High quality` szerepel, tehát a megoldó nem észlelt problémára utaló jelet.

Nézzük a feladat optimális megoldását. A döntési változók az 'oszlopok', tehát a `Column name` alatt kell keresni őket. Az optimális értékeket az `Activity` oszlopban találjuk meg: az első termékből 307,789 darabot kell gyártani, a másodikból nem érdemes gyártani semennyit, a harmadikból pedig 169,598 darabot. Most fogadjuk el, hogy lehetséges tört értéket gyártani a termékekből (később megvizsgáljuk majd az optimális megoldást abban az esetben is, ha nem oszthatóak a termékek, azaz csak egész szám fogadható el megoldásként). Amelyik terméket gyártjuk, ott az `St` oszlopban `B` betű található, ezek a bázisváltozók. Az `x2` változó oszlopában `NL` betűk szerepelnek, ebből az `N` betű azt jelenti, hogy az `x2` változó nembázis változó, az `L` betű pedig arra utal, hogy az alsó (Lower) korlátba ütközik bele, azaz a 'default' nemnegativitási korlátba. Megjegyezzük, hogy olyan változó is lehet bázisváltozó, ahol az optimális érték 0, ezt a jelenséget degenerációnak hívjuk. A degeneráció jelensége a valós üzleti problémák esetén igen gyakori.

Az első termékből 307,789 darabot gyártunk, és egységenként 21 (Ft) bevételre teszünk szert. A harmadik termékből 169,598 darabot gyártunk, és egységenként 50

(Ft) bevételre teszünk szert. Az összes bevételünk: $21 \times 307{,}789 + 50 \times 169{,}598 = 14943{,}47$, ez a célfüggvény értéke (némi kerekítési pontatlansággal).

A korlátok az együtthatómátrix sorai, a korlátok megnevezései a `Row name` oszlopban található. Az `r.5` név azt takarja, hogy a kód 5. sorában kezdődő korlát, esetünkben ez a `11x1 + 30x2 + 39x3 <= 10000` korlát. Kicsit szerencsétlen ez a megnevezés. Ha akarjuk, saját nevet adhatunk a korlátoknak (vagy akár a célfüggvénynek is), és akkor jobban olvasható az eredményfül. Ennek módja: a sor elején megadjuk a korlát nevét (ami nem kezdődhet számmal), majd a név után kettőspontot teszünk. Lássuk ezt a példában: a célfüggvénynek a `profit` nevet adjuk, az erőforrásokra vonatkozó korlátoknak pedig az `EF1`, `EF2` és `EF3` neveket.

2.5. kód.
```
max
profit: 21x1 + 32x2 + 50x3

subject to
EF1:  11x1  +  30x2 +  39x3 <=  10000
EF2:  41x1  +  60x2 +  73x3 <=  25000
EF3: 112x1  + 135x2 + 194x3 <= 110000

end
```

Az eredményfül a nevek használata esetén így néz ki (A Karush-Kuhn-Tucker optimalitási feltételeket lehagyva az állomány végéről):

Problem:
Rows: 3
Columns: 3
Non-zeros: 9
Status: OPTIMAL
Objective: profit = 14943.46734 (MAXimum)

No.	Row name	St	Activity	Lower bound	Upper bound	Marginal
1	EF1	NU	10000		10000	0.649497
2	EF2	NU	25000		25000	0.33794
3	EF3	B	67374.4		110000	

No.	Column name	St	Activity	Lower bound	Upper bound	Marginal
1	x1	B	307.789	0		
2	x2	NL	0	0		-7.76131
3	x3	B	169.598	0		

Látható, hogy a fejlécben az `Objective` mellett megjelent a `profit` felirat, és az első korlátnál nem az `r.5` elnevezés szerepel, hanem az `EF1`. Nézzük a táblázatban található adatokat. Érdemes előre venni a `Lower bound` és `Upper bound` oszlopokat. A mi esetünkben az `Lower bound` oszlop üres, csak az `Upper bound` oszlopban vannak értékek, pl. az `EF1` korlát esetén 10000. Ez azt jelenti, hogy a korlátnak (a `11x1+30x2+39x3` kifejezésnek) nincs alsó korlátja, a felső korlátja pedig 10000. Nézzük az `Activity` oszlopot. Ez az oszlop mutatja az erőforrásból használt mennyiséget, amely az `EF1` korlát esetén 10000 lesz. Lássuk, hogy hogyan áll össze ez a szám. Az első termékből 307,789 mennyiséget gyártunk, egy termékhez 11 egység kell az első erőforrásból, ez összesen: $11 \times 307{,}789 = 3385{,}679$. A második termékből nem gyártunk, így ehhez nem is használunk semmit az első erőforrásból. A harmadik termékből 169,598-at gyártunk, egy termékhez 39 egység kell az első erőforrásból, $39 \times 169{,}598 = 6614{,}322$. A három termékhez összesen $3385{,}679+0+6614{,}322=10000$ (kerekítési hibával), ez az érték szerepel az `Activity` oszlopban. Az első erőforrásból tehát az összes rendelkezésre álló mennyiséget elhasználtuk, ebben az esetben `N` szerepel az `St` (státusz) oszlopban (a korláthoz tartozó eltérésváltozó nembázis változó lesz). Az `St` oszlopban az `N` mellett szerepel az `U` betű is, ez arra utal, hogy a `11x1+30x2+39x3` kifejezés a felső (upper) korlátjába ütközik bele. A mi esetünkben ez triviális, az alsó korlátba nem is tudna beleütközni, mert nincs neki.

Az optimális termelési tervhez a második erőforrásból is elhasználjuk a teljes rendelkezésre álló mennyiséget, ezért az `St` oszlopban itt is `NU` karakterek szerepelnek. Megjegyezzük, hogy degeneráció jelensége korlátok esetén is felléphet, tehát előfordulhat, hogy a korlát egyenlőség formájában teljesül, az `St` oszlopban mégis `B` betű található.

A harmadik erőforrásból nem használjuk el a teljes rendelkezésre álló mennyiséget, csak 67374,4-t ($110000-67374{,}4=42625{,}6$ egység még marad). Látható, hogy ebben az esetben az `St` oszlopban `B` betű szerepel (a korláthoz tartozó eltérésváltozó bázisváltozó).

2.2. Duálfeladat fogalma és az LP feladatok érzékenységvizsgálata

Az eredményfülön szerepel még a `Marginal` oszlop. Az oszlop tartalmának megértéséhez át kell ismételni a dualitás elméletet. Az alfejezet további részében nem fogjuk a teljes eredményfület megadni, csak azt a részét, ami a vizsgált problémánál releváns. A duálfeladat azért hasznos a számunkra, mert a duálváltozókhoz közgazdasági interpretáció társítható, segítségükkel több és mélyebb információra tehetünk szert a primál feladat kapcsán is.

2.6. példa. *Adjuk meg a 2.2. LP felírás duálfeladatát, illetve a duálfeladat optimális megoldását!*

Megoldás:

A 2.7. LP felírásesetén megadtuk a vizsgált termelési feladatot mátrixalgebrai jelölésekkel:

$$\begin{array}{rcl} \mathbf{c}^T\mathbf{x} & \to & \max \\ \text{feltéve hogy:} & & \\ A\mathbf{x} & \leq & \mathbf{b} \\ \mathbf{x} & \geq & \mathbf{0} \end{array} .$$

Fontos hangsúlyozni, hogy a $\mathbf{b}$ vektorról nem követeljük meg, hogy nemnegatív legyen, így minden LP feladat ilyen alakra hozható.

A 2.7. LP felírás feladatához tartozó duálfeladat:

2.7. LP felírás.

$$\begin{array}{rcl} \mathbf{y}^T\mathbf{b} & \to & \min \\ \textit{feltéve hogy:} & & \\ \mathbf{y}^TA & \geq & \mathbf{c}^T \\ \mathbf{y} & \geq & \mathbf{0} \end{array} .$$

A duálfeladatban $\mathbf{y}$ vektor tartalmazza a döntési változókat. Látható, hogy a primnál feladatban a kapacitások ($\mathbf{b}$ vektor) a duálfeladatban célfüggvény együtthatók lesznek, a primál feladat célfüggvény együtthatói pedig ($\mathbf{c}$ vektor) kapacitások.

A konkrét termelési feladat esetén a duálfeladat:

2.8. kód.
```
min
profit: 10000y1 + 25000y2 + 110000y3

subject to

Dualk1:  11y1  +  41y2 +  112y3 >=  21
Dualk2:  30y1  +  60y2 +  135y3 >=  32
Dualk3:  39y1  +  73y2 +  194y3 >=  50

end
```

A 2.8. kódhoz tartozó LP feladat optimális megoldása:

```
Status:     OPTIMAL
Objective:  profit = 14943.46734 (MINimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 Dualk1       NL            21       307.789
     2 Dualk2       B        39.7613
     3 Dualk3       NL            50       169.598

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 y1           B       0.649497
     2 y2           B        0.33794
     3 y3           NL             0       42625.6
```

Látható, hogy a célfüggvény értéke ugyanannyi, mint korábban. A első és harmadik termékből gyártott mennyiség feltűnik a megfelelő duálkorlátok `Marginal` oszlopában, és a 3. duálváltozó `Marginal` oszlopában található a harmadik erőforrásból el nem használt mennyiség. További kapcsolat a két feladat között, hogy a duálváltozók optimális értéke (`Activity` oszlop) megjelenik a 2.5. kódhoz tartozó feladat optimális megoldásában, a `Marginal` oszlopban. Innen nem nehéz megállapítani, hogy az alapfeladat megoldásában a `Marginal` oszlop a duálváltozók optimális értékeit adja meg.

2.9. példa. *Végezzük el a 2.1. példa kapcsán az LP feladatok érzékenységvizsgálatát!*

Megoldás.

A duálváltozóknak közgazdasági jelentést is tudunk adni. Nézzük először a korlátokat. Térjünk vissza a 2.5. kódhoz és vizsgáljuk az első korlátot:

```
   No.   Row name    St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 EF1          NU         10000      0.649497
```

A célfüggvény értéke 14943,46734. Nézzük, mi történik, ha az első erőforrásból rendelkezésre álló mennyiséget 1 egységgel csökkentjük. Az egyértelműség kedvéért az új feladat kódját is teljes egészében megadjuk:

2.10. kód.

```
max
profit: 21x1 + 32x2 + 50x3

subject to
EF1:  11x1  +  30x2 +  39x3 <=   9999
EF2:  41x1  +  60x2 +  73x3 <=  25000
EF3: 112x1  + 135x2 + 194x3 <= 110000

end
```

Az optimális megoldás ebben az esetben:

```
Objective:  profit = 14942.81784 (MAXimum)

  Row name    St   Activity     Upper bound    Marginal
------------ -- ------------- ------------- -------------
EF1          NU          9999          9999      0.649497
EF2          NU         25000         25000       0.33794
EF3          B        67374.7        110000

Column name  St   Activity     Lower bound    Marginal
------------ -- ------------- ------------- -------------
x1           B        307.881             0
x2           NL             0             0      -7.76131
x3           B        169.546             0
```

Az optimális megoldással kapcsolatban a következő tényekre érdemes felhívni a figyelmet:

- Ugyanazokat a termékeket gyártjuk, mint korábban (bár a mennyiségek változtak).
- Ugyanazokat az erőforrásokat használjuk ki teljes mértékben (`EF1`, `EF2`), mint korábban. A harmadik erőforrást nem használjuk ki teljesen most sem.
- Nem változtak a `Marginal` oszlopban szereplő értékek.
- Az előző pontokat úgy szoktuk összefoglalni, hogy a bázis változatlan maradt.

Nézzük a célfüggvény változását: ha az első erőforrásból 10000 állt rendelkezésre, akkor a célfüggvény értéke 14943,46734, ha csak 9999 áll rendelkezésre, akkor 14942,81784. A különbség: 0,64950, ami nem más mint az első korláthoz tartozó duálváltozó értéke (kis numerikus pontatlansággal). Tehát az első erőforrás csökkenése (növelése) egységenként 0,64950-nel csökkenti (növeli) a profitot. Ezért szoktuk a korlátokhoz tartozó duálváltozót **árnyékár**nak hívni.

A `EF2` korlát esetén az árnyékár 0,33794, tehát a második erőforrás csökkenése (növelése) egységenként 0,33794-gyel csökkenti (növeli) a profitot.

A `EF3` korlát esetén az árnyékár 0, ami logikus is, hiszen semmi probléma nem származik abból, ha csökken a rendelkezésre álló mennyiség (legfeljebb kevesebb megy veszendőbe). Tehát ha a korlát 'státusza' `B` (a korláthoz tartozó eltérésváltozó bázisváltozó), akkor a duálváltozó értéke 0 (ezt hívjuk az LP feladat komplementaritásának).

Ha az első erőforrás 'értéke' 0,64950, és 10000 egység áll rendelkezésre, akkor a célfüggvény értéke miért nem $0{,}64950 \times 10000 = 6495{,}0$? Hogy megértsük ezt, állítsuk az első erőforrásból rendelkezésre álló mennyiséget kicsi értékre, mondjuk 1-re. Nézzük az optimális megoldást ebben az esetben:

```
Objective:  profit = 1.909090909 (MAXimum)

  Row name   St   Activity     Upper bound    Marginal
------------ -- ------------- ------------- -------------
EF1          NU             1             1       1.90909
EF2          B        3.72727         25000
EF3          B        10.1818        110000
```

```
Column name  St   Activity     Lower bound    Marginal
------------ -- ------------- ------------- -------------
x1           B      0.0909091             0
x2           NL             0             0      -25.2727
x3           NL             0             0      -24.4545
```

Amit érdemes észrevenni az optimális megoldás kapcsán:

- Csak az első terméket gyártjuk.
- Az első és második erőforrásból is van felesleg.
- Megváltoztak az árnyékárak.
- Az előző pontokat együttesen úgy is mondhatjuk, hogy megváltozott a bázis.

Tehát az árnyékár csak addig érvényes, amíg a bázis változatlan marad. És honnan tudhatjuk, hogy meddig marad változatlan a bázis? Ezt az ún. érzékenységvizsgálati határok adják meg. Az érzékenységvizsgálati határok nem szerepelnek az eredmény fülön, ezt egy másik fülön találjuk meg. Tools menüben klikkeljünk az 'Generate LP Sensitivity Analysis' szövegre, majd futtassuk megint a modellt. Látjuk, hogy nem csak egy er4edmény fül keletkezik, hanem egy `termeles_sens.out` fül is.

```
GLPK 4.60 - SENSITIVITY ANALYSIS REPORT                                                         Page   1

Problem:
Objective:  profit = 14943.46734 (MAXimum)

   No. Row name     St      Activity         Slack   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
     1 EF1          NU   10000.00000          .             -Inf     6707.31707       -.64950   12804.87805 x3
                                            .64950   10000.00000    13356.16438          +Inf   17123.28767 x1

     2 EF2          NU   25000.00000          .             -Inf    18717.94872       -.33794   12820.51282 x1
                                            .33794   25000.00000    37272.72727          +Inf   19090.90909 x3

     3 EF3          BS   67374.37186   42625.62814          -Inf    49743.58974       -.12041    6830.79678 EF2
                                              .     110000.00000    68292.68293       2.32883  171846.84685 EF1

GLPK 4.60 - SENSITIVITY ANALYSIS REPORT                                                         Page   2

Problem:
Objective:  profit = 14943.46734 (MAXimum)

   No. Column name  St      Activity      Obj coef   Lower bound       Activity      Obj coef  Obj value at Limiting
                                          Marginal   Upper bound          range         range   break point variable
------ ------------ -- ------------- ------------- -------------  ------------- ------------- ------------- ------------
     1 x1           BS     307.78894      21.00000          .              -Inf      14.10256   12820.51282 EF2
                                              .             +Inf      609.75610      28.08219   17123.28767 EF1

     2 x2           NL          .         32.00000          .       -1703.31325          -Inf   28163.40361 EF3
                                          -7.76131          +Inf      236.84211      39.76131   13105.26316 x3

     3 x3           BS     169.59799      50.00000          .       -1000.00000      39.16140   13105.26316 x2
                                              .             +Inf      256.41026      74.45455   19090.90909 EF2

End of report
```

Az állomány struktúrája hasonló az eredményfülhöz: az első táblázat a korlátokról ad információkat, a második a változókról. Az `St` és `Activity` oszlopok az eredményfülön is szerepeltek, így ezekre nem térünk ki. A következő oszlop a `Slack/Marginal`. Minden korláthoz két sor tartozik ebben az outputban. A felső sorban van a `Slack` érték, az alsóban a `Marginal`. A `Marginal` címke alatt a duálváltozót érti, jelen esetben az árnyékárat jelenti. A `Slack` érték az adott erőforrásból el nem használt mennyiséget jelöli (A hiányzó érték 0-t jelent). Döntési változók esetén a `Obj coef` oszlopban az adott döntési váltózó célfüggvény együtthatója szerepel. A következő oszlop a `Lower bound/Upper bound`. Ez a korlátra vagy a döntési változóra vonatkozó határokat adja meg, ezt is ismertettük már korábban. A következő oszlop a `Activity range`, ez adja meg, hogy milyen határok között marad változatlan a bázis. Az `Obj coef range` oszlopot döntési változók esetén fogjuk értelmezni, a célfüggvény együtthatók változtatására ad olyan intervallumot, amin belül a bázis változatlan marad. Az `Obj value at break point` oszlopban a célfüggvény optimális értéke látható az érzékenységvizsgálati intervallum alsó és felső határa esetén. A `Limiting variable` oszlop pedig azt mutatja meg, hogyan változik a bázis, ha kilépünk az érzékenységvizsgálati intervallumból.

Nézzük konkrétan a `EF1` korlátot. Az érzékenységvizsgálati intervallum alsó és felső határa 6707,31707 és 13356,16438. Amennyiben az első erőforrásból rendelkezésre álló mennyiség ezen két érték közé esik, akkor a bázis változatlan marad (azaz érvényes marad 0,64950-es árnyékár is). Nézzük mi történik, ha megközelítjük ezt a határt: állítsuk az első erőforrásból rendelkezésre álló mennyiséget 6708-ra. Az eredményfül:

```
Objective:  profit = 12805.32161 (MAXimum)

   Row name    St   Activity      Upper bound    Marginal
------------ -- ------------- ------------- -------------
EF1          NU          6708          6708      0.649497
EF2          NU         25000         25000       0.33794
EF3          B        68292.5        110000

Column name  St   Activity     Lower bound    Marginal
```

```
------------ -- ------------- ------------- -------------
x1           B         609.693             0
x2           NL              0             0      -7.76131
x3           B       0.0351759             0
```

Látható, hogy a bázis tényleg nem változik, az árnyékárak ugyanazok mint korábban. A harmadik termékből viszont már alig gyártunk valamit. Ha tovább csökkentjük az erőforrás mennyiségét, `x3` változó értéke 0-vá válik, és kikerül a bázisból. Látható, hogy a `Limiting variable` oszlopban, a felső sorban is az `x3` változó tűnik fel. A célfüggvény optimális értéke (12805,32161) is elég közel került az `Obj value at break point` oszlop felső sorában szereplő értékhez.

Ha az erőforrás felső határához állítjuk be a rendelkezésre álló mennyiséget (13356), akkor az optimális megoldás:

```
Objective:  profit = 17123.1809 (MAXimum)

   Row name   St   Activity     Upper bound    Marginal
------------ -- ------------- ------------- -------------
EF1          NU          13356         13356      0.649497
EF2          NU          25000         25000       0.33794
EF3          B         66438.4        110000

Column name   St   Activity     Lower bound    Marginal
------------ -- ------------- ------------- -------------
x1           B       0.0150754             0
x2           NL              0             0      -7.76131
x3           B         342.457             0
```

Most pedig az első termékből állítunk elő nagyon kevés mennyiséget, és ha tovább növeljük az erőforrás rendelkezésre álló mennyiségét, akkor az első terméket nem fogjuk gyártani, ez a változó tűnik fel a `Limiting variable` oszlopban, az alsó sorban.

Az `EF2` korlát esetén az `Activity range` alsó és felső határa 18717,94872 és 37272,72727, ha az erőforrás rendelkezésre álló mennyisége ezen két érték között van, akkor a bázis változatlan marad.

Az `EF3` korlát esetén óvatosan kell eljárni. Ez a korlát egyenlőtlenség formájában teljesül. Ha növeljük a korlát jobboldalát, akkor egyszerűen nő az el nem használt mennyiség, tehát bármeddig növelhetjük a rendelkezésre álló mennyiséget, a bázis változatlan marad. Csökkenteni pedig annyit tudjuk, amennyi az el nem használt mennyiség. Ha ennél is tovább csökkentjük, ez az erőforrás is szűkössé válik, tehát megváltozik a bázis. `EF3` korlát esetén az érzékenységvizsgálati intervallum alsó és felső határa: 110000-42625,62814=67374,37186 és végtelen. Az érzékenységvizsgálati outputban az `Activity range` oszlopban nem ezek az értékek szerepelnek. Az `Activity range` oszlop csak egyenlőséggel teljesülő korlátok esetén adja meg az érzékenységvizsgálati intervallum határait.

Az árnyékárak idáig pozitívak voltak, de ez feltétlenül van így. Módosítsuk a példát úgy, hogy kormányzati előírásnak megfelelően a második termékből legalább 10 egységet gyártani kell:

2.11. kód.
```
max
profit: 21x1 + 32x2 + 50x3

subject to
EF1:  11x1  +  30x2 +  39x3 <=  10000
EF2:  41x1  +  60x2 +  73x3 <=  25000
EF3: 112x1  + 135x2 + 194x3 <= 110000
KE:               x2        >=     10

end
```

Az optimális megoldás:

```
Objective:  profit = 14865.20477 (MAXimum)

  Row name   St   Activity      Marginal
------------ -- ------------- -------------
EF1          NU         10000      0.649497
EF2          NU         25000       0.33794
EF3          B        67124.4
KE           NL            10      -7.76131

Column name  St   Activity      Marginal
------------ -- ------------- -------------
x1           B        305.905
x2           B             10
x3           B        162.437
```

Látható, hogy a `KE` korlát árnyékára negatív, ami logikus is: minél magasabb a kormányzati előírás, annál kisebb a profit.

Térjünk vissza az eredeti feladathoz (kormányzati előírás nélküli feladat). A döntési változókhoz kapcsolódóan milyen kérdéseket tudunk megválaszolni az LP feladat érzékenységvizsgálata segítségével?

Látható, hogy a második terméket nem gyártjuk, nem elég profitábilis. Mennyit kellene kérni a második termékért, hogy megérje gyártani? Erre ad választ az `x2` változóhoz tartozó duál eltérésváltozó (`Marginal` oszlop): -7,76131. Ez az ún. **redukált költség**. Ha ennyivel csökkentjük a termék árát (ha negatív értékkel csökkentjük, az ténylegesen emelkedést jelent), akkor megérné gyártani. Nézzük, mi történik, ha a második termék árát 7,76131 egységgel 39,76131-ra növeljük. Hogy egyértelmű legyen a különbség egy kicsit alatta maradunk ennek a határnak: 39,76:

2.12. kód.

```
max
profit: 21x1 + 39.76x2 + 50x3

subject to
EF1:  11x1  +  30x2 +  39x3 <=  10000
EF2:  41x1  +  60x2 +  73x3 <=  25000
EF3: 112x1  + 135x2 + 194x3 <= 110000

end
```

Az optimális megoldás:

```
Objective:  profit = 14942.81784 (MAXimum)

  Row name   St   Activity     Upper bound    Marginal
------------ -- ------------- ------------- -------------
EF1          NU         10000         10000      0.649497
EF2          NU         25000         25000       0.33794
EF3          B        67374.4        110000

Column name  St   Activity     Lower bound    Marginal
------------ -- ------------- ------------- -------------
x1           B        307.789             0
x2           NL             0             0   -0.00130653
x3           B        169.598             0
```

Látható, hogy ugyanazokat a termékeket gyártjuk, mint korábban, ugyanolyan mennyiségben. Továbbra is az első és második erőforrást használjuk ki teljes mértékben és a harmadik erőforrásból van felesleg. Összefoglalóan: nem változott a bázis. De látjuk azt is, hogy a redukált költség (abszolút értékben) lecsökkent. Ha a termék árát még egy kicsit megemelnénk, akkor már az első és második terméket érné meg gyártani.

Térjünk vissza az eredeti értékekre. Jelenleg az első terméket 21 egységért lehet értékesíteni, és ennyi pénzért érdemes is gyártani ezt a terméket. Milyen határok között változhat az első termékért kapott pénz, hogy ne változzon ez a gyártási terv? Erre a kérdésre megint csak az érzékenységvizsgálati határok adnak választ. A `Sensitivity analysis` outputon az `x1` változóhoz tartozó két sor:

```
Column name  St      Obj coef  Obj value at Limiting
                        range   break point variable
------------ -- ------------- ------------- ------------
x1           BS      14.10256   12819.23077 EF2
                     28.08219   17123.28767 EF1
```

Az `Obj coef range` oszlop intervallumának alsó határa 14,10256, a felső határa pedig 28,08219. Amennyiben az első termék ára ezen két határ között van, nem változik az optimális gyártási terv. Próbáljuk ki! Állítsuk az első termék árát 14,11-re. Ekkor az optimális megoldás:

```
Objective:  profit = 12821.52014 (MAXimum)

  Row name   St   Activity     Upper bound    Marginal
------------ -- ------------- ------------- -------------
EF1          NU         10000         10000       1.28137
EF2          NU         25000         25000   0.000364322
EF3          B        67374.4        110000

Column name  St   Activity     Lower bound    Marginal
------------ -- ------------- ------------- -------------
x1           B        307.789             0
x2           NL             0             0      -6.46294
x3           B        169.598             0
```

Ugyanazokat a termékeket gyártjuk, mint korábban, és továbbra is az első és a második erőforrás rendelkezésre álló mennyiségét használjuk ki teljesen. Megváltoztak viszont az árnyékárak: az `EF2` erőforrás árnyékára közel 0. Ha az első termék ára tovább csökken, akkor nem használjuk el az összes rendelkezésre álló mennyiséget a második erőforrásból. Ezt az információt mutatja az érzékenységvizsgálat `Limiting variable` oszlopában a felső, `EF2` kifejezés. Ha ellenben az első termék ára 28,08219 felé emelkedik, akkor már csak egyes terméket fogunk gyártani, és az első erőforrásból fog felesleg maradni (lásd: `Limiting variable` oszlopban az alsó, `EF1` kifejezés).

A harmadik termék esetén az érzékenységvizsgálati intervallum (`Obj coef range`) alsó határa 39,16140, a felső határa pedig 74,45455. Amennyiben a harmadik termék ára ezen két érték közé esik, nem változik a termelési terv, azaz nem változik az optimális bázis.

A második termék nembázis változó, ezért az `Obj coef range` értékeknek nincs számunkra jelentősége (nembázis változó esetén a redukált költséget kell nézni). Természetesen egyszerre csak 1 termék árának változását tudjuk vizsgálni, több termék árának együttes változása már bonyolultabb kérdés.

2.3. Speciális esetek: nincs lehetséges megoldás és nemkorlátos célfüggvény

2.13. példa. *Módosítsunk a 2.1 példa szövegén. Mindegyik termékből legalább 1000 egységet le kell gyártani, mert ekkora mennyiségre korábban már leszerződtünk.*

Megoldás.

A 2.5. kódot könnyen át tudjuk írni, hogy szerepeljenek benne a termékekre vonatkozó alsó korlátok:

2.14. kód.
```
max
profit: 21x1 + 32x2 + 50x3

subject to
EF1:  11x1  +  30x2 +  39x3 <=  10000
EF2:  41x1  +  60x2 +  73x3 <=  25000
EF3: 112x1  + 135x2 + 194x3 <= 110000
x1also: x1 >= 1000
x2also: x2 >= 1000
x3also: x3 >= 1000

end
```

A 2.14. kód futtatásakor a következő üzenetet kapjuk: `PROBLEM HAS NO PRIMAL FEASIBLE SOLUTION`, ami annyit tesz, hogy nincs olyan termelési terv, ami az összes feltételnek eleget tesz. Ezt könnyen ellenőrizhetjük is: 1000 db első termékhez az első erőforrásból 11000 egységre lenne szükség, de ez már önmagában meghaladja a rendelkezésre álló 10000-es keretet. Fontos hangsúlyozni, hogy outputot ilyenkor is kapunk, de a status `UNDEFINED`, tehát nem a feladat megoldását mutatja.

2.15. példa. *Módosítsuk más irányba a 2.1. példát. Az együttható mátrixot (A) cseréljük ki az*

$$A = \begin{pmatrix} 11 & -70 & 39 \\ 41 & -73 & 73 \\ 112 & 135 & -200 \end{pmatrix}$$

mátrixra. Elsőre furcsák lehetnek a negatív előjelek a mátrixban. A negatív együtthatók azt jelentik, hogy egy adott termék során keletkezik olyan melléktermék (pl. hulladékhő), amit másik termék előállításánál fel lehet használni.

Mennyi ebben az esetben az elérhető maximális bevétel?

Megoldás.

A feladat CPLEX LP kódja:

2.16. kód.

```
max
profit: 21x1 + 32x2 + 50x3

subject to
EF1:  11x1  -  70x2 +  39x3 <=  10000
EF2:  41x1  -  73x2 +  73x3 <=  25000
EF3: 112x1  + 135x2 - 200x3 <= 110000

end
```

A 2.16. kód futtatásakor `LP HAS UNBOUNDED PRIMAL SOLUTION` üzenetet kapjuk, azaz nemkorlátos a célfüggvény. Bármilyen nagy nyereségre szert tudunk tenni, amit ebben az esetben is könnyen ellenőrizhetünk: gyártsunk a második és harmadik termékből 1-1 darabot, ami 82 Ft profitot eredményez számunkra.

Ekkor az első erőforrásból -70+39=-31 darabot használunk, tehát nemhogy fogyna az erőforrás rendelkezésre álló mennyisége, hanem még nő is. Hasonlóan a második erőforrásból -73+73=0 mennyiséget használunk el, a harmadikból pedig 135-200=-65 mennyiséget. Látható, hogy egyik erőforrás mennyisége sem csökkent. Gyártsunk n-n darabot a második és harmadik termékből. Ekkor $82n$ profitra tehetünk szert anélkül, hogy csökkenne bármelyik erőforrás mennyisége is. Az n értéket tetszőlegesen nagyra választhatjuk, tehát 'végtelen' nagyságú profitra tehetünk szert.

Természetesen üzleti problémák esetén nehezen hihető, hogy végtelen nagyságú profitra lehet szert tenni. Ezekben az esetekben a felírt modell nem felel meg teljesen a valóságnak. A konkrét példa esetén érdemes lenne a paramétereket még egyszer ellenőrizni, de még ha a paraméterek helyesek is, valószínüleg akkor sem igaz, hogy tetszőleges mennyiséget értékesíteni lehet a megadott árakon.

Bizonyos esetekben elképzelhető, hogy nemkorlátos célfüggvény esetén a `PROBLEM HAS NO DUAL FEASIBLE SOLUTION` üzenetet kapjuk, de a dualitási tétel értelmében ez logikus is (ha a primál feladat célfüggvénye nem korlátos, akkor a duál feladatnak nincs lehetséges megoldása).

2.4. Egészértékű változók

Bizonyos feladatok esetén elfogadhatatlan, ha az optimális megoldásban tört szám szerepel. Nem tudunk 0,72 darab repülőgépet gyártani. Ráadásul egészértékű változók segítségével logikai korlátokat is elő lehet írni, ebben az esetben különösen fontos a változók egészértékűsége, tört értékű változóknak semmi értelmet nem tudunk adni.

2.17. példa. *Adja meg a 2.1. példa optimális megoldását abban az esetben is, ha a termelési mennyiség csak egész érték lehet.*

Megoldás.

LP feladatok esetén lehetőségünk van egészértékűségi kikötésre. Egészértékű feladatokat az ún. korlátozás és szétválasztás (branch and bound) algoritmussal oldunk meg. Elsőként megoldjuk a feladatot egészértékűségi kikötések nélkül. Ha valamelyik változó értékére tört érték adódott, akkor két részre bontjuk a feladatot, az egyik ágon megoldjuk úgy, hogy az adott változó értéke kisebb vagy egyenlő, mint az optimális megoldás alsó egész része; a másik ágon a változó értéke nagyobb, mint az optimális megoldás felső egész része. Eltároljuk a megoldáshoz tartozó célfüggvény értéket is. Ha a két ág valamelyikén újfent tört érték adódik bármelyik változóra, akkor iteráljuk ezt a folyamatot. Mindig azt a lehetséges egészértékű megoldásvektort tartjuk optimálisnak, amelyik a legjobb célfüggvényértékkel rendelkezik. Szerencsés esetben bizonyos LP feladatokat (elágazást) nem kell megoldani, mert már találtunk egy egészértékű (lehetséges) megoldást, ahol a célfüggvény értéke már magasabb, mint ami az elágazásban elérhető, ezt hívjuk az algoritmus korlátozás részének. A folyamat akkor ér véget, ha minden ágra teljesül, hogy végigszámoltuk, vagy az algoritmus korlátozása miatt lezártuk.

A mai LP megoldók a korlátozás és szétválasztás algoritmusát nem a tiszta formájában használják, hanem mielőtt elvégeznék a szétválasztást, ún. metszősíkokat generálnak. A metszősíkok levágnak a lehetséges megoldások tartományából egy részt, ami biztosan nem tartalmaz egészértékű megoldást. A metszősíkok használatával jelentősen felgyorsítható az egészértékű feladatok megoldása. A korlátozás és szétválasztás és a metszősíkok együttes használatát korlátozás és vágásnak hívjuk (angolul ‘’ branch and cut’).

Egészértékű változó használatával a futási idő jelentősen megnövekedhet, akár olyan mértékben is, hogy praktikusan a feladatok megoldhatatlanná válik számunkra. Ha tudjuk, kerüljük el a használatukat. Természetesen ez sokszor nem lehetséges. Egészértékű feladatok megoldása esetén a modell felírása is nagyon sokat számít. Ugyanannak az egészértékű feladatnak a különböző felírásai között nagyságrendnyi különbség is lehet futási idő tekintetében.

Bár az egészértékű feladatok megoldása bonyolult, a megadásuk a CPLEX LP kódban viszont egyszerű: egy új szekcióban, amit az `integer` kulcsszó vezet be (megengedettek még `general`, `int` és `gen` kulcsszavak), felsoroljuk az egészértékű változókat (mindegyiket új sorban, vagy szóközzel elválasztva).

2.18. kód.
```
max
profit: 21x1 + 32x2 + 50x3

subject to
EF1:  11x1  +  30x2 +  39x3 <=  10000
EF2:  41x1  +  60x2 +  73x3 <=  25000
EF3: 112x1  + 135x2 + 194x3 <= 110000

integer
x1
x2
x3

end
```

Az optimális megoldás:

```
Problem:
Rows:        3
Columns:     3 (3 integer, 0 binary)
Non-zeros:   9
Status:      INTEGER OPTIMAL
Objective:   profit = 14929 (MAXimum)

  Row name        Activity     Lower bound   Upper bound
------------    ------------- ------------- -------------
EF1                      9998                       10000
EF2                     24984                       25000
EF3                     67305                      110000

Column name       Activity     Lower bound   Upper bound
------------    ------------- ------------- -------------
x1           *            307             0
x2           *              1             0
x3           *            169             0
```

Az optimális megoldásban az első termékből 307-et, a második termékből 1-et, a harmadik termékből pedig 169-et érdemes gyártani.

Egészérték változók esetén az eredményfül is megváltozik valamelyest. A fejlécben nem csak az oszlopok (változók) száma szerepel, hanem az is, hogy a változók közül hány egészértékű. Jelen feladatban mindegyik változó egészértékű, ezt egészértékű programozási feladatnak (angolul IP=Integer Programming) nevezzük. Nem szükséges, hogy a modellben szereplő összes változó egészértékű legyen, ezt vegyes egészértékű feladatnak (angolul MILP=Mixed Integer Linear Programming) nevezzük.

A fejlécben a státusz `INTEGER OPTIMAL`. Az egészértékű változók mellett $\star$ szerepel. További fontos következmény, hogy nincs `Marginal` oszlop, azaz duálváltozók. Az árnyékár és redukált költség egészértékű feladatok esetén is releváns mutatók, de nem tudjuk őket meghatározni az optimális megoldásból. Ilyen jellegű kérdéseket csak újabb futtatásokkal tudjuk megválaszolni.

2.5. Bináris változók

Ha egy egészértékű (nemnegatív) változóról kikötjük, hogy értéke kisebb vagy egyenlő, mint 1, akkor ez a változó csak 0 vagy 1 értéket vehet fel. Ezeket a változókat bináris változóknak hívjuk. Bináris változók használatával a modellezési lehetőségek kitágulnak, de sokszor a futási idő is megnövekszik.

Bináris változókat megadhatnánk úgy is, hogy egy egészértékű változóra kikötjük, hogy értéke nem lehet nagyobb, mint 1, de érdemes ezeket a változókat külön kezelni. A CPLEX LP formátumban erre szolgál a `binary` szekció. Az `integer` szekcióhoz hasonlóan egyszerűen fel kell sorolni a bináris változókat (a bináris változókat nem kell/nem szabad az `integer` szekcióban is megadni).

2.19. példa. *A 2.1. példa szövegén módosítunk egy kicsit. Egészértékű megoldást keresünk, de méretgazdaságossági megfontolásokból egy termékből vagy nem gyártunk, vagy legalább 10-t. Továbbá a második terméket csak akkor lehet gyártani, ha gyártunk az elsőből is, a harmadik terméket pedig csak akkor lehet gyártani, ha gyártunk az első kettőből is. Ezeket a korlátozó feltételeket betartva mekkora maximális nyereség érhető el?*

Megoldás.

Vegyük először a méretgazdaságossági korlátokat. A korlátok felírásához be kell vezetni bináris változókat. Legyenek `b1`, `b2` és `b3` bináris változók, amelyeknek 0 értéke azt jelenti, hogy nem gyártjuk az adott terméket, 1 értéke pedig azt, hogy gyártjuk.

Nem elég bevezetni a `b1`, `b2` és `b3` változókat, ezek a változók nem függetlenek `x1`, `x2` és `x3` változóktól. Nehezen tudnánk egy olyan megoldást értelmezni, ahol `x1` értéke 10, `b1` értéke pedig 0. Korlátokkal ki kell kényszeríteni a változók koherenciáját, azaz ha `x1` változó pozitív, akkor `b1` változó értéke 1, és fordítva, ha `b1` változó értéke 0, akkor `x1` értékének is 0-nak kell lenni. Ezt a kapcsolatot ún. 'nagy M' korláttal (angolul big M) tudjuk megoldani:

$$x_i \leq M b_i \ ,$$

ahol M egy kellően nagy szám. Ha b_i változó 0, akkor abból következik, hogy x_i változó értéke is 0. Ha b_i értéke 1, akkor x_i értéke 0 és M között bármi lehet. Ha M értékét olyan nagyra állítjuk, hogy x_i értéke annál nagyobb úgysem lehetne, akkor ezt azt jelenti, hogy amennyiben b_i változó értéke 1, akkor nincs ténylegesen korlátozva x_i változó.

A következő kérdés, hogy honnan tudunk ilyen M értékre szert tenni? Ez problémáról problémára változik. A konkrét feladat esetén a rendelkezésre álló erőforrások korlátozzák a termelést. Biztosnak vehetjük, hogy bármelyik erőforrást választjuk is ki az M értékének megbecsléséhez, az egy valós termelési korlátot ad. Lehet, hogy nem a legjobb (legalacsonyabb) korlátot, de egy érvényeset. Válasszunk hát egy tetszőleges erőforrást, és takarítsuk meg magunknak azt az erőfeszítést, amit a legoptimálisabb M keresése igényelne. Példánkban tekintsük az első erőforrást. Az első termék előállításához 11 egységre van szükség a rendelkezésre álló 10000 egységből, így 10000/11=909,1 egységnél többet biztos nem tudunk gyártani az első termékből. Az M értékek a második termék esetén 10000/30=333,4 (felfele kerekítve), a harmadik termék esetén pedig: 256,5 (szintén felfele kerekítve).

Nagy M korlátok esetén, ha matematikai oldalról nézzük a problémát, nem kell a legjobb felső korlátot megtalálni. Tehát a 909,1 helyett nyugodtan használhatunk

1000-es értéket vagy akár 1000000-t is. Numerikus oldalról nézve problémás lehet, ha nagyon nagy számot választunk M értékének. Ezen feladatok megoldása az egyik legnagyobb kihívás a MILP feladatok esetén. Igyekezzünk kerülni ezeket a 'nagy M' korlátokat. Vannak esetek, amikor ez nem lehetséges, de vannak olyan jó ismert trükkök, amik segíthetnek. (Próbálkozhatunk például a folytonos döntési változók diszkretizálásával.)

Az $x_i \leq Mb_i$ korlátokkal ki tudjuk kényszeríteni az x_i és b_i változók közötti koherenciát. A bevezetett b_i változóval fel lehet írni a méretgazdaságossági feltételt is:

$$x_i \geq 10b_i$$

Ha b_i értéke 0, akkor x_i értéke lehet 0 (a $x_i \leq Mb_i$ korlátok kizárják, hogy x_i értéke pozitív legyen). Ha b_i értéke 1, akkor x_i értéke legalább 10. Azaz a két korlát együtt azt kényszeríti ki, hogy x_i-ből legalább 10 darabot gyártsunk, ha a gyártás mellett döntünk, de engedélyezi azt a lehetőséget is, hogy egyáltalán ne késztsünk belőle egy darabot sem.

A további korlátokat már könnyen fel lehet írni. A második terméket csak akkor tudjuk gyártani, ha gyártjuk az első terméket is: $b_2 \leq b_1$. A harmadik terméket csak akkor tudjuk gyártani, ha gyártjuk az első kettőt is: $2b_3 \leq b_1 + b_2$. Összefoglalva:

2.20. kód.

```
max
profit: 21x1 + 32x2 + 50x3

subject to
EF1:  11x1  +  30x2 +  39x3 <=  10000
EF2:  41x1  +  60x2 +  73x3 <=  25000
EF3: 112x1  + 135x2 + 194x3 <= 110000

Koh1: x1  - 910b1 <= 0
Koh2: x2  - 334b2 <= 0
Koh3: x3  - 257b3 <= 0

MG1: x1  - 10b1 >= 0
MG2: x2  - 10b2 >= 0
MG3: x3  - 10b3 >= 0

T2: b2 - b1 <= 0
```

```
T3: 2b3 - b1 - b2 <= 0

integer
x1
x2
x3

binary
b1
b2
b3

end
```

Az optimális megoldás:

```
Status:     INTEGER OPTIMAL
Objective:  profit = 14849 (MAXimum)

Column name       Activity
------------    -------------
x1           *           307
x2           *            11
x3           *           161
b1           *             1
b2           *             1
b3           *             1
```

Tehát az első termékből 307 darabot gyártunk, a másodikból 11-t, a harmadikból pedig 161-t. Ekkor 14849 egység bevételre teszünk szert.

2.6. Gyakorló feladatok

2.6.1. Egyszerűbb feladatok

2.21. példa. *Egy gyógyszergyár egy új típusú multivitamint szeretne piacra dobni, amelyet négy különböző, már forgalomban lévő, saját fejleszésű termékből kívánnak összeállítani. Az új termék tervezésekor 3 ásványi anyag és 1 vitamin értékét állítják be előre meghatározott szintekre.*

Azt szeretnék elérni, hogy az új termék 1 grammjában a kalcium legalább 100 mg, a magnézium legalább 90 mg, a foszfor pedig 85 és 95 mg között lenne. A vitaminok *közül egyedül a K-vitamin szintje fontos, amelyből legalább 60 μg-ot kell tartalmaznia az új multivitaminnak.*

A négy eddigi termékükben 1 gramm tartalmaz rendre 140, 105, 99 és 90 mg kalciumot, valamint 65, 72, 100 és 123 mg magnéziumot, valamint 152, 115, 81 és 98 mg foszfort, illetve 71, 42, 41 és 83 μg K-vitamint.

A négy régebbi termékük előállítási költsége rendre 67, 83, 68 illetve 55 Ft grammonként. A gyógyszergyár szeretné az új terméket a lehető legkevesebb előállítási költséggel kikeverni. Adjon meg egy LP-t amelyben a legolcsóbb keverék meghatározása a cél, és érzékenységvizsgálat segítségével válaszoljon a következő kérdésekre:

a) *Mennyibe kerül az új termékből egy 1,8 grammos tabletta?*

b) *Hogyan változik egy tabletta ára, ha a foszfor 84 és 96 mg között lehet?*

c) *Hogyan változik egy tabletta ára, ha a foszfor 80 és 115 mg között lehet?*

d) *Ha 10 mg-al növeljük a magnézium mennyiségét egy gramm új termékben, akkor hogyan változik az új termék előállítási költsége?*

e) *A következő hónapban egy új gépet fog beüzemelni a gyógyszergyár, aminek köszönhetően a második termék előállítási költsége 20 forinttal olcsóbb lesz. Az új gép mennyivel csökkenti az új termék előállítási költségét?*

Megoldás a 206. oldalon.

2.22. példa. *Egy üdítőital gyártó vállalat alma, barack és alma-barack ízesítésű ivóleveket állít elő. Egy liter alma ízesítésű ital előállításához 10 dkg cukorra, 770 ml vízre és 230 ml almasűrítményre van szükség. Egy liter barackos ívólé 8 dkg cukorból, 6,5 dl vízből és 350 ml baracksűrítményből készül. A vegyes ízű üdítőital elkészítéséhez 7 dkg cukrot, 450 ml vizet, 260 ml almasűrítményt és 290 ml baracksűrítményt használnak fel.*

Egy nap alatt 30 kg cukrot, 150 liter vizet, 60 liter almasűrítményt és a 40 liter baracksűrítményt tud a vállalat feldolgozni. Az üdítőitalokat rendre 300, 380 és 400 *forinton árulják. Mindegyik üdítőből legalább 20 litert kell naponta gyártani. Adjon meg egy LP-t, amely esetén a napi gyártás bevétele maximalizálható, és egy IP-t arra az esetre, ha a gyártás a palackozással ér véget a nap végén, ahol minden üdítőitalt 1 literes palackba töltenek. Mennyivel növelné a napi bevételt mind a két esetben, ha egy olyan beszállítót választanának, ahol a felhasználható baracksűrítmény naponta 45 liter lenne?*

Megoldás a 210. oldalon.

2.23. példa. *Egy vállalat marketingkampányba kezd, amelyre 50 millió forintot szánnak. A kampány során a következőképpen lehetőségek közül választhatnak:*

1. *Hirdetést adhatnak fel az A televízióban, 15 millió forintért, amely várhatóan 400 ezer nézőt ér el.*
2. *Hirdethetnek a konkurens B televízióban is, amely 12 millió forintba kerül, azonban ezzel csak várhatóan 340 ezer embert érnek el.*
3. *Reklámozhatják magukat a megyei rádióban 3 millió forintért, amivel 60 ezer embert érnek el.*
4. *Választhatják az országos rádiót is, amellyel várhatóan 200 ezer hallgató figyel fel rájuk, 8 millió forintért cserébe.*
5. *Nyereményjátékot indíthatnak, amelynek a költsége 16 millió forint, és várhatóan 350 ezer ember figyelmét kelti fel.*
6. *A város több pontján plakátokat helyezhetnek el, amellyel várhatóan 130 ezerrel növelik az ügyfeleik számát, 8 millió forint költség mellett.*
7. *Internetes hirdetést is választhatnak, amelyre 5, 10, vagy 20 millió forintot költve rendre várhatóan 100, 220 és 500 ezer embert érnek el.*

A két televízióban egyszerre nem hirdethetnek, valamint az internetes kampányból is csak az egyik futhat. Azonban a vállalat ragaszkodik az internetes hirdetéshez, ezért valamelyiket a három közül mindenképpen választani kell.

A megyei rádió az A televízió leányvállalata, tehát amennyiben ezt a tévét választják, akkor a megyei rádióban is kapányolniuk kell, viszont fordítva nem kötelező.

A két rádióban hirdethetnek egyszerre, viszont ebben az esetben a legdrágább internetes opciót nem választhatják, mert ekkor az előrejelzésük pontossága csökken.

Nyereményjátékot csak megfelelő marketingkampány mellett indíthatnak, tehát ha az országos rádióban és valamelyik televízióban is futnak hirdetések.

Adjon meg egy IP-t, amellyel a várhatóan elérhető emberek számát maximalizálhatjuk.

Megoldás a 213. oldalon.

2.6.2. Nehezebb feladatok

2.24. példa. *A Balatonba a Zala folyóból érkező vízmennyiség az év 52 hetére a következő értékekkel növeli meg a vízszintet (mm-ben).*

hét	*Zala hozama*	*hét*	*Zala hozama*	*hét*	*Zala hozama*	*hét*	*Zala hozama*
1	21	14	34	27	19	40	15
2	18	15	11	28	17	41	17
3	16	16	13	29	11	42	29
4	13	17	7	30	13	43	28
5	12	18	10	31	18	44	21
6	11	19	8	32	13	45	10
7	14	20	13	33	11	46	16
8	13	21	15	34	10	47	14
9	26	22	10	35	12	48	12
10	33	23	19	36	8	49	28
11	37	24	20	37	12	50	25
12	40	25	24	38	13	51	16
13	31	26	23	39	16	52	18

A Balatonban felgyülemlett vizet a Sió csatornában le lehet engedni, hetente maximum 20 (mm-t). A Balaton (átlagos) vízszintje induláskor 3996 (mm).

a) A vízügyi igazgatóság a vízszintet a $(4000-d; 4000+d)$ *intervallumban szeretné tartani (minden hét végén). Mi a legkisebb d érték, amire teljesíthető a vállalás?*

b) A Sió Szivattyútelep Kft. opciós joggal rendelkezik egy szivattyú egyheti használatára. A szivattyú a beüzemelt héten (plusz) 1 mm-rel csökkenti a vízszintet. Melyik hétre kell a szivattyú használatát időzíteni? Honnan tudjuk?

c) Az első heti vízhozamot nem tudjuk pontosan megmondani (de a többit igen). A Sió Szivattyútelep Kft. a dolgozók szabadságolását szeretné tervezni, azokra a hetekre időzítve, amikor nem működik a szivattyútelep. Milyen határok között változhat a Zala hozama az első héten, hogy ne változzon a szivattyútelep (heti) ütemterve? Honnan tudjuk?

d) Változik-e a helyzet (és ha igen hogyan), ha a Sió csatorna zsilipje egész évben maximum 5 héten lehet nyitva?

e) Változik-e a helyzet (és ha igen hogyan), ha a Sió csatorna zsilipje legfeljebb 4 hétig lehet folyamatosan nyitva, az azt követő ötödik héten a zsilipet be kell zárni (azután újra megnyitható)?

Megoldás a 216. oldalon.

2.25. példa. *Gotham városnak nyolc kerülete van. A táblázat mutatja, hogy hány percig tart, amíg egy mentőautó az egyik kerületből a másikba ér.*

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1	*0*	*3*	*4*	*6*	*8*	*9*	*8*	*10*
2	*3*	*0*	*5*	*4*	*8*	*6*	*12*	*9*
3	*4*	*5*	*0*	*2*	*2*	*3*	*5*	*7*
4	*6*	*4*	*2*	*0*	*3*	*2*	*5*	*4*
5	*8*	*8*	*2*	*3*	*0*	*2*	*2*	*4*
6	*9*	*6*	*3*	*2*	*2*	*0*	*3*	*3*
7	*8*	*12*	*5*	*5*	*2*	*3*	*0*	*2*
8	*10*	*9*	*7*	*4*	*4*	*3*	*2*	*0*

Tehát pl. az első kerületből a második kerületbe 3 perc alatt ér oda a mentő.

Az egyes kerületek lakossága (ezer főben megadva) a következő: 1. kerület 40; 2. kerület 30; 3. kerület 35; 4. kerület 20; 5. kerület 15; 6. kerület 50; 7. kerület 45; 8. kerület 60. A városnak csak két mentőautója van.

- Hol állomásoztassák a mentőautókat, hogy maximalizálják azok számát, akikhez a mentőkocsik 2 percen belül kiérnek?
- Melyik kerületbe lenne érdemes telepíteni egy harmadik mentőautót? Ezzel hány új lakót lehetne elérni?
- Térjünk vissza az eredeti feladathoz! Hány mentőautó szükséges az összes kerület 2 percen belüli eléréséhez?

Megoldás a 233. oldalon.

3. fejezet

Szállítási feladat

3.1. Egyszerű szállítási feladat

A szállítási feladat esetén az alapfeladat a következőképpen néz ki: adott a raktárak egy halmaza $R = \{r_1, r_2, ..., r_n\}$ és az áruházak egy halmaza $A = \{a_1, a_2, ..., a_m\}$. Emellett adottak a raktárakban tárolt árumennyiségek $(s_1, s_2, ..., s_n)$ (s=supply) és az áruházak igényei $(d_1, d_2, ..., d_m)$ (d=demand). A tárolt árumennyiségekre kapacitásokként is szoktunk hivatkozni. A c_{ij} költségmátrix elemei az egységnyi áru elszállításának költségét adják meg az r_i raktárból az a_j áruházba. A célunk az, hogy kielégítsük az áruházak igényeit a lehető legkisebb szállítási költséggel.

Jelölje x_{ij} döntési változó az r_i raktárból az a_j áruházba elszállított árumennyiséget. (A bevezetett jelöléshez jobban igazodna az $x_{r_i a_j}$ változó, de így a képletek válnának áttekinthetetlenné. Az egyszerű szállítási feladatoknál nem okoz gondot ez a jelölés, az átrakodásos szállítási feladatnál további magyarázattal szolgálunk majd a változónevekkel kapcsolatban.) Ekkor a feladat LP felírása a következő:

3.1. LP felírás.

$$\sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \rightarrow \min$$

feltéve hogy:

$$\begin{aligned} \sum_{j=1}^{m} x_{ij} &\leq s_i && \forall i \\ \sum_{i=1}^{n} x_{ij} &\geq d_i && \forall j \\ x_{ij} &\geq 0 && \forall i,j \end{aligned}$$

Az LP felírásban az utolsó sort nemnegativitási korlátnak hívjuk. Bár elvi lehetőség van rá, hogy másképp definiáljunk változókat (pl.: előjelkötetlenül), de ezzel a lehetőséggel most nem fogunk élni. Helytakarékosság miatt a továbbiakban feltesszük, hogy $x_{ij} \geq 0$ minden (i,j)-re.

Nézzünk egy konkrét példát!

3.2. példa. *Van 5 raktárunk és 6 áruházunk. Minden raktárban 100 egységnyi áru van, az áruházak igényei pedig rendre 98, 136, 46, 52, 83, és 85. A költségmátrix legyen:*

$$\begin{matrix} 17 & 26 & 25 & 28 & 34 & 25 \\ 35 & 21 & 21 & 14 & 13 & 11 \\ 27 & 23 & 12 & 18 & 31 & 14 \\ 18 & 21 & 29 & 39 & 13 & 17 \\ 35 & 37 & 29 & 30 & 15 & 12 \end{matrix} \quad .$$

Legkevesebb mekkora költséget igényel, hogy kielégítsük az áruházak igényeit?

Megoldás.

Az LP felírás a konkrét feladat esetén:

$$
\begin{array}{ll}
+17x_{1\_1}+26x_{1\_2}+25x_{1\_3}+28x_{1\_4}+34x_{1\_5}+25x_{1\_6} & \\
+35x_{2\_1}+21x_{2\_2}+21x_{2\_3}+14x_{2\_4}+13x_{2\_5}+11x_{2\_6} & \\
+27x_{3\_1}+23x_{3\_2}+12x_{3\_3}+18x_{3\_4}+31x_{3\_5}+14x_{3\_6} & \\
+18x_{4\_1}+21x_{4\_2}+29x_{4\_3}+39x_{4\_4}+13x_{4\_5}+17x_{4\_6} & \\
+35x_{5\_1}+37x_{5\_2}+29x_{5\_3}+30x_{5\_4}+15x_{5\_5}+12x_{5\_6} & \rightarrow \quad \min
\end{array}
$$

f.h.:

$$
\begin{array}{rrrrrrcl}
+x_{1\_1} & +x_{1\_2} & +x_{1\_3} & +x_{1\_4} & +x_{1\_5} & +x_{1\_6} & \leq & 100 \\
+x_{2\_1} & +x_{2\_2} & +x_{2\_3} & +x_{2\_4} & +x_{2\_5} & +x_{2\_6} & \leq & 100 \\
+x_{3\_1} & +x_{3\_2} & +x_{3\_3} & +x_{3\_4} & +x_{3\_5} & +x_{3\_6} & \leq & 100 \\
+x_{4\_1} & +x_{4\_2} & +x_{4\_3} & +x_{4\_4} & +x_{4\_5} & +x_{4\_6} & \leq & 100 \\
+x_{5\_1} & +x_{5\_2} & +x_{5\_3} & +x_{5\_4} & +x_{5\_5} & +x_{5\_6} & \leq & 100 \\
\\
 & +x_{1\_1} & +x_{2\_1} & +x_{3\_1} & +x_{4\_1} & +x_{5\_1} & \geq & 98 \\
 & +x_{1\_2} & +x_{2\_2} & +x_{3\_2} & +x_{4\_2} & +x_{5\_2} & \geq & 136 \\
 & +x_{1\_3} & +x_{2\_3} & +x_{3\_3} & +x_{4\_3} & +x_{5\_3} & \geq & 46 \\
 & +x_{1\_4} & +x_{2\_4} & +x_{3\_4} & +x_{4\_4} & +x_{5\_4} & \geq & 52 \\
 & +x_{1\_5} & +x_{2\_5} & +x_{3\_5} & +x_{4\_5} & +x_{5\_5} & \geq & 83 \\
 & +x_{1\_6} & +x_{2\_6} & +x_{3\_6} & +x_{4\_6} & +x_{5\_6} & \geq & 85 \\
 & & & & & x_{ij} & \geq & 0
\end{array}
$$

Ugyanez az LP felírás CPLEX LP formátumban:

3.3. kód.
```
min
ktg:
+17x1_1 +26x1_2 +25x1_3 +28x1_4 +34x1_5 +25x1_6
+35x2_1 +21x2_2 +21x2_3 +14x2_4 +13x2_5 +11x2_6
+27x3_1 +23x3_2 +12x3_3 +18x3_4 +31x3_5 +14x3_6
+18x4_1 +21x4_2 +29x4_3 +39x4_4 +13x4_5 +17x4_6
+35x5_1 +37x5_2 +29x5_3 +30x5_4 +15x5_5 +12x5_6

subject to
r1:+x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 <= 100
r2:+x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 <= 100
r3:+x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 <= 100
r4:+x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 <= 100
r5:+x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 <= 100

a1:+x1_1 +x2_1 +x3_1 +x4_1 +x5_1 >= 98
a2:+x1_2 +x2_2 +x3_2 +x4_2 +x5_2 >= 136
a3:+x1_3 +x2_3 +x3_3 +x4_3 +x5_3 >= 46
a4:+x1_4 +x2_4 +x3_4 +x4_4 +x5_4 >= 52
a5:+x1_5 +x2_5 +x3_5 +x4_5 +x5_5 >= 83
a6:+x1_6 +x2_6 +x3_6 +x4_6 +x5_6 >= 85

end
```

A 3.3. kódhoz tartozó LP output állomány (alternatív megoldás is van, másik output is elképzelhető, amit az output állományhoz fűzött megoldás után adunk meg):

```
Status:     OPTIMAL
Objective:  ktg = 8049 (MINimum)

   No.   Row name   St   Activity
------ ------------ -- -------------
     1 r1           B            100
     2 r2           NU           100
     3 r3           NU           100
     4 r4           NU           100
     5 r5           NU           100
     6 a1           NL            98
     7 a2           NL           136
     8 a3           NL            46
     9 a4           NL            52
```

```
    10 a5           NL            83
    11 a6           NL            85

   No. Column name  St   Activity
------ ------------ -- -------------
     1 x1_1         B             98
     2 x1_2         B              2
     3 x1_3         NL             0
     4 x1_4         NL             0
     5 x1_5         NL             0
     6 x1_6         NL             0
     7 x2_1         NL             0
     8 x2_2         B             48
     9 x2_3         NL             0
    10 x2_4         B             52
    11 x2_5         NL             0
    12 x2_6         NL             0
    13 x3_1         NL             0
    14 x3_2         B             54
    15 x3_3         B             46
    16 x3_4         NL             0
    17 x3_5         NL             0
    18 x3_6         NL             0
    19 x4_1         NL             0
    20 x4_2         B             32
    21 x4_3         NL             0
    22 x4_4         NL             0
    23 x4_5         B             68
    24 x4_6         NL             0
    25 x5_1         NL             0
    26 x5_2         NL             0
    27 x5_3         NL             0
    28 x5_4         NL             0
    29 x5_5         B             15
    30 x5_6         B             85
```

Az optimális megoldás esetén a szállítási összköltség 8049. A következő viszonylatokban történik szállítás: első raktárból az első áruházba 98 egység (x1_1=98), az első raktárból a második áruházba 2 egység (x1_2=2), a második raktárból a második áruházba 48 egység (x2_2=48), a második raktárból a negyedik áruházba 52 egység (x2_4=52), a harmadik raktárból a második áruházba 54 egység (x3_2=54), a harmadik raktárból a harmadik áruházba 46 egység (x3_3=46), a negyedik raktárból a második áruházba 32 egység (x4_2=32), a negyedik raktárból az ötödik áruházba 68 egység (x4_5=68), az ötödik raktárból az ötödik áruházba 15 egység és végül az ötödik raktárból a hatodik áruházba 85 egység (x5_6=85). Ellenőrzésül: 17×98+26×2+21×48+14×52+23×54+12×46+21×32+13×68+15×15+12×85=8049.

Amelyik viszonylatban történik szállítás, ott a megfelelő x változók bázisváltozók. Ezt onnan látjuk, hogy az eredményfül St oszlopában B betű szerepel (pl. 1 x1_1 B).

Az első raktárból a harmadik áruházba nem történik szállítás. Ez nembázis változó. Ekkor az St oszlopban NL szerepel (3 x1_3 NL). Az N mutatja, hogy nembázis változó, az L betű, hogy az alsó korlátba ütközünk bele (emlékeztetőül: minden változó esetén van egy nemnegativitási korlát, ezt nem kellett külön megadni, ez az alapértelmezés).

Az Activity oszlop mutatja a korlát baloldalát, az első korlát esetén a x1_1+x1_2+x1_3+x1_4+x1_5+x1_6 összeget. Az első korlát esetén ez is 100, tehát a korlát egyenlőség formájában teljesül. Az első korlát esetén az St oszlopban B szerepel a többi oszlopban NU vagy NL. Az N a nembázis változóra utal (emlékeztetőül: minden korláthoz tartozik egy eltérésváltozó, és ez az eltérésváltozó nembázis változó, azaz 0, tehát a korlát egyenlőség formájában teljesül) <= korlátok esetén NU szerepel, tehát itt a kifejezés a felső korlátjába ütközik bele; >= korlátok esetén NL szerepel, ebben az esetben a kifejezés az alsó korlátjába ütközik bele. Érdekes az első korlát esete: ez is egyenlőség formájában teljesül, de a St oszlopban B szerepel, ami bázisváltozóra utal. Itt az a helyzet, hogy az eltérésváltozó tényleg a bázisban szerepel, de értéke 0. Ezt hívjuk degenerációnak. A kérdésről később még szót ejtünk.

Érdemes megemlíteni, hogy a feladatnak van egy alternatív megoldása is (ami az eredményfülön nem látszik). Hogy lássuk ezt, adjuk hozzá a 3.3. kódhoz az x2_2=0 korlátot. Ekkor az optimális megoldás:

```
   No. Column name  St   Activity
------ ------------ -- -------------
     1 x1_1         B             98
```

```
 2 x1_2          B              2
10 x2_4          B             52
11 x2_5          B             48
14 x3_2          B             54
15 x3_3          B             46
20 x4_2          B             80
23 x4_5          B             20
29 x5_5          B             15
30 x5_6          B             85
```

Látható, hogy a célfüggvény értéke most is 8049, tehát ez is optimális megoldás. Az előző megoldáshoz képest az a különbség, hogy `x2_2` változó értéke 0, 'cserébe' viszont `x2_5` változó értéke 48. Változik még továbbá `x4_2` és `x4_5` változó értéke.

Amikor a raktárak kapacitásának összege és az áruházak igényeinek összege megegyezik, akkor a feladatot *kiegyensúlyozott szállítási feladat*nak nevezzük. Ha a szállítási feladatot LP feladatként oldjuk meg, nincs különösebb jelentősége, hogy a feladat kiegyensúlyozott-e. A szállítási feladatot megoldhatjuk ún. huroktranszformációs módszerrel is, de ezen módszerrel már csak kiegyensúlyozott feladatot tudunk megoldani (mint látni fogjuk, sok nem kiegyensúlyozott feladat visszavezethető kiegyensúlyozott feladatra). Szimplex algoritmus használatával kellően nagyméretű feladat is megoldható (pl. 1000 raktár 1000 áruház), de nagyméretű feladatok esetén már érzékelhető lehet a huroktranszformációs módszer előnye.

Pár megjegyzést érdemes tenni a kiegyensúlyozott szállítási feladattal kapcsolatban. Amikor a raktárak kapacitásainak összege megegyezik az áruházak igényeinek összegével, akkor minden korlát egyenlőség formájában teljesül, tehát minden relációjel egyenlőség is lehetne. Ekkor, ha a raktárakra valamint az áruházakra vonatkozó korlátokat összeadjuk (külön-külön), akkor ugyanazt az összeget kapjuk. Másképpen megfogalmazva: kiszemeljük valamelyik áruházat (mondjuk a 6. áruházat), majd összeadjuk a *raktárakra* vonatkozó korlátokat, és levonjuk belőle az első 5 *áruházra* vonatkozó korlátok (a 6. korlátot nem tartalmazó áruházi korlátok) összegét. Ezáltal pont a 6. áruházra vonatkozó korlátot kapjuk meg.

Az előző gondolatmenetet általánosíthatjuk is minden kiegyensúlyozott feladatra.

Tehát ha n raktárra teljesül az egyenlőség, továbbá $m-1$ áruházra is, akkor teljesül az m. áruházra is. Nyilván ez fordítva is igaz lesz: ha $n-1$ raktárra és m áruházra teljesül az egyenlőség, akkor az n. raktárra is teljesül, tehát az egyenletrendszer redundáns.

Cseréljük ki az összes relációjelet egyenlőségre a 3.3. kódban. Ekkor az `r1` korláthoz tartozó eltérésváltozó bázisváltozó lesz 0 értékkel. Másrészről válasszuk ki valamelyik raktárat (vagy áruházat) és hagyjuk el a hozzá tartozó korlátot. Ekkor ugyanazt a megoldást kapjuk.

3.1.1. Bővítési lehetőségek

Az alapfeladatot többféle irányban bővíthetjük. Egyik lehetséges irány, ha a raktárakban több áru van, mint amennyi az áruházak igénye. Ekkor valamennyi áru ott marad a raktárakban, a kérdés csak az, hogy melyikben és mekkora mennyiség.

3.4. példa. *A 3.2. példa szövegén változtassunk annyit, hogy ne 100, hanem 120 egységnyi áru legyen minden raktárban. Mennyi ebben az esetben a szállítási összköltség, valamint melyik raktárakban marad áru és mennyi?*

Megoldás.

Többféle megoldás is elképzelhető. Egyik lehetséges megoldás, hogy pl. a 3.3. kódban kicseréljük a raktárakra vonatkozó korlátokat `<=100`-ról `<=120`-ra.

3.5. kód.

```
min
ktg:
+17x1_1 +26x1_2 +25x1_3 +28x1_4 +34x1_5 +25x1_6
+35x2_1 +21x2_2 +21x2_3 +14x2_4 +13x2_5 +11x2_6
+27x3_1 +23x3_2 +12x3_3 +18x3_4 +31x3_5 +14x3_6
+18x4_1 +21x4_2 +29x4_3 +39x4_4 +13x4_5 +17x4_6
+35x5_1 +37x5_2 +29x5_3 +30x5_4 +15x5_5 +12x5_6

subject to
r1:+x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 <= 120
r2:+x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 <= 120
r3:+x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 <= 120
r4:+x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 <= 120
r5:+x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 <= 120
```

```
a1:+x1_1 +x2_1 +x3_1 +x4_1 +x5_1 >= 98
a2:+x1_2 +x2_2 +x3_2 +x4_2 +x5_2 >= 136
a3:+x1_3 +x2_3 +x3_3 +x4_3 +x5_3 >= 46
a4:+x1_4 +x2_4 +x3_4 +x4_4 +x5_4 >= 52
a5:+x1_5 +x2_5 +x3_5 +x4_5 +x5_5 >= 83
a6:+x1_6 +x2_6 +x3_6 +x4_6 +x5_6 >= 85

end
```

Az eredményfülön a korlátokra vonatkozó részből állapítható meg egyszerűen, hogy melyik raktárban mennyi áru marad.

```
No.   Row name St   Activity     Lower bound   Upper bound
--- ---------- -- ------------- ------------- -------------
  1 r1         B             98                         120
  2 r2         NU           120                         120
  3 r3         B             77                         120
  4 r4         NU           120                         120
  5 r5         B             85                         120
  6 a1         NL            98            98
  7 a2         NL           136           136
  8 a3         NL            46            46
  9 a4         NL            52            52
 10 a5         NL            83            83
 11 a6         NL            85            85
```

Az r_2 raktárhoz tartozó korlát esetén a felső korlát 120, ami megegyezik az egyenlet baloldalának értékével (`Activity` oszlop), tehát ebből a raktárból 120 egységnyi árut fognak elszállítani.

Az r_1 raktárhoz tartozó korlát esetén a felső korlát szintén 120, de az egyenlet baloldala csak 98, vagyis csak 98 egységnyi árut szállítanak el ebből a raktárból, tehát a maradék 22 egység áru ebben a raktárban marad. Hasonlóan meg belátható, hogy a harmadik raktárban 43, az ötödik raktárban pedig 35 egységnyi áru marad.

Megoldás #2.

A feladatot visszavezethetjük kiegyensúlyozott feladatra is. Bevezetünk egy fiktív 7. áruházat (a_f), amelynek igénye a raktárakban tárolt árumennyiségek és az áruházak igényeinek különbsége, jelen példában 5×120-$500=100$. Ebbe az 'áruházba' történő 'szállítás' az áruk raktárban hagyását jelenti, amelynek ebben a példában nincs költsége, ezért a költségmátrix:

$$\begin{matrix} 17 & 26 & 25 & 28 & 34 & 25 & 0 \\ 35 & 21 & 21 & 14 & 13 & 11 & 0 \\ 27 & 23 & 12 & 18 & 31 & 14 & 0 \\ 18 & 21 & 29 & 39 & 13 & 17 & 0 \\ 35 & 37 & 29 & 30 & 15 & 12 & 0 \end{matrix} .$$

Tehát ebben a megoldásban az a_f áruházba 'szállított' mennyiségek fogják megmutatni, hogy melyik raktárban mennyi áru marad (a fiktív áruházhoz tartozó döntési változóknak 0 az együtthatójuk a célfüggvényben, így akár el is lehetne hagyni őket a célfüggvényből).

3.6. kód.
```
min
ktg:
+17x1_1 +26x1_2 +25x1_3 +28x1_4 +34x1_5 +25x1_6 +0x1_f
+35x2_1 +21x2_2 +21x2_3 +14x2_4 +13x2_5 +11x2_6 +0x2_f
+27x3_1 +23x3_2 +12x3_3 +18x3_4 +31x3_5 +14x3_6 +0x3_f
+18x4_1 +21x4_2 +29x4_3 +39x4_4 +13x4_5 +17x4_6 +0x4_f
+35x5_1 +37x5_2 +29x5_3 +30x5_4 +15x5_5 +12x5_6 +0x5_f

subject to
r1:+x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_f = 120
r2:+x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 +x2_f = 120
r3:+x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 +x3_f = 120
r4:+x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 +x4_f = 120
r5:+x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 +x5_f = 120

a1:+x1_1 +x2_1 +x3_1 +x4_1 +x5_1 = 98
a2:+x1_2 +x2_2 +x3_2 +x4_2 +x5_2 = 136
a3:+x1_3 +x2_3 +x3_3 +x4_3 +x5_3 = 46
a4:+x1_4 +x2_4 +x3_4 +x4_4 +x5_4 = 52
a5:+x1_5 +x2_5 +x3_5 +x4_5 +x5_5 = 83
a6:+x1_6 +x2_6 +x3_6 +x4_6 +x5_6 = 85
af:+x1_f +x2_f +x3_f +x4_f +x5_f = 100

end
```

A korlátok mind egyenlőség formájában teljesülnek, de ami számunkra most érdekes, hogy az `x1_f`, …, `x5_f` változókra milyen optimális értékeket kapunk.

```
   No. Column name  St   Activity
------ ------------ -- -------------
     7 x1_f         B             22
    14 x2_f         NL             0
    21 x3_f         B             43
    28 x4_f         NL             0
    35 x5_f         B             35
```

Tehát az r_1 raktárban 22, az r_3 raktárban 43, az r_5 raktárban pedig 35 egység áru marad. Ugyanazt az eredményt kaptuk, mint a korábban bemutatott másik megközelítésű megoldásban.

Ennek a feladatnak is van alternatív optimuma:

```
   No. Column name  St   Activity
------ ------------ -- -------------
     7 x1_f         B             22
    14 x2_f         NL             0
    21 x3_f         B             74
    28 x4_f         NL             0
    35 x5_f         B              4
```

Ezen bázismegoldás esetén annyi a változás, hogy a r_3 raktárban 74, r_5 raktárban pedig 4 egységnyi áru marad.

3.7. példa. *A 3.4. példa szövegén változtassunk annyit, hogy a raktárakban maradó mennyiség után tárolási költséget kell fizetni. A raktározási költség az első raktárban 21, a másodikban 12, a harmadikban 27, a negyedikben 5, az ötödikben pedig 19. Mennyi most a minimális összköltség (szállítási+raktározási)?*

Megoldás.

Ebben az esetben az a könnyebben járható út, ha bevezetünk egy fiktív áruházat, amelyik a tárolandó mennyiségeket rögzíti majd (most is ez a `f` indexű). A 3.6. kódhoz képest most csak annyi a különbség, hogy a fiktív áruházba nem 0 költséggel tudunk 'szállítani', hanem a raktározási költségekkel.

3.8. kód.

```
min
ktg:
+17x1_1 +26x1_2 +25x1_3 +28x1_4 +34x1_5 +25x1_6 +21x1_f
```

```
+35x2_1 +21x2_2 +21x2_3 +14x2_4 +13x2_5 +11x2_6 +12x2_f
+27x3_1 +23x3_2 +12x3_3 +18x3_4 +31x3_5 +14x3_6 +27x3_f
+18x4_1 +21x4_2 +29x4_3 +39x4_4 +13x4_5 +17x4_6 +5x4_f
+35x5_1 +37x5_2 +29x5_3 +30x5_4 +15x5_5 +12x5_6 +19x5_f

subject to
r1:+x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_f = 120
r2:+x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 +x2_f = 120
r3:+x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 +x3_f = 120
r4:+x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 +x4_f = 120
r5:+x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 +x5_f = 120

a1:+x1_1 +x2_1 +x3_1 +x4_1 +x5_1 = 98
a2:+x1_2 +x2_2 +x3_2 +x4_2 +x5_2 = 136
a3:+x1_3 +x2_3 +x3_3 +x4_3 +x5_3 = 46
a4:+x1_4 +x2_4 +x3_4 +x4_4 +x5_4 = 52
a5:+x1_5 +x2_5 +x3_5 +x4_5 +x5_5 = 83
a6:+x1_6 +x2_6 +x3_6 +x4_6 +x5_6 = 85
af:+x1_f +x2_f +x3_f +x4_f +x5_f = 100

end
```

Nézzük az optimális tárolási tervet a fiktív áruházra!

```
   No. Column name  St   Activity
------ ------------ -- -------------
     7 x1_f         NL             0
    14 x2_f         NL             0
    21 x3_f         NL             0
    28 x4_f         B            100
    35 x5_f         NL             0
```

Ebben az esetben csak a 4. raktárból történik szállítás a fiktív áruházba, tehát csak ott marad áru. A szállítási összköltség 8729.

Megoldás #2.

Ezt a feladatot is meg lehet oldani fiktív áruház bevezetése nélkül. Az r_1 raktár esetén az el nem szállított mennyiség `120-x1_1-x1_2-x1_3-x1_4-x1_5-x1_6` lesz, ennek raktározási költsége pedig 21 (egységenként). Ezt a raktározási összköltséget (amely ennél a raktárnál `21(120-x1_1-x1_2-x1_3-x1_4-x1_5-x1_6)`), hozzáadjuk a célfüggvényhez. A 21*120 fix költség, az optimum helyét nem befolyásolja, ezért nem kell szerepeltetni (viszont majd ha megkapjuk az optimális megoldást a célfüggvény értékéhez hozzá kell adni). A CPLEX LP formátumban egy változó csak egyszer szerepelhet, ezért az x változók együtthatóit össze kell vonni, tehát az r_1 rakár esetében a célfüggvény `-4x1_1+5x1_2+4x1_3+7x1_4+13x1_5+4x1_6`. Ugyanezt a műveletsort elvégezzük a többi raktárra is.

3.9. kód.

```
min
ktg:
- 4x1_1 + 5x1_2 + 4x1_3 + 7x1_4 +13x1_5 + 4x1_6
+23x2_1 + 9x2_2 + 9x2_3 + 2x2_4 + 1x2_5 - 1x2_6
+ 0x3_1 - 4x3_2 -15x3_3 - 9x3_4 + 4x3_5 -13x3_6
+13x4_1 +16x4_2 +24x4_3 +34x4_4 + 8x4_5 +12x4_6
+16x5_1 +18x5_2 +10x5_3 +11x5_4 - 4x5_5 - 7x5_6

subject to
r1:+x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 <= 120
r2:+x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 <= 120
r3:+x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 <= 120
r4:+x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 <= 120
r5:+x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 <= 120

a1:+x1_1 +x2_1 +x3_1 +x4_1 +x5_1 >= 98
a2:+x1_2 +x2_2 +x3_2 +x4_2 +x5_2 >= 136
a3:+x1_3 +x2_3 +x3_3 +x4_3 +x5_3 >= 46
a4:+x1_4 +x2_4 +x3_4 +x4_4 +x5_4 >= 52
a5:+x1_5 +x2_5 +x3_5 +x4_5 +x5_5 >= 83
a6:+x1_6 +x2_6 +x3_6 +x4_6 +x5_6 >= 85

end
```

A célfüggvény optimális értéke -1351. Ehhez hozzá kell adni azt a részt, amitől korábban eltekintettünk, nevezetesen `120(21+12+27+5+19)=10080`. Ekkor `-1351+10080=8729`, azaz ugyanakkora szállítási költséget kapunk, mint a korábbi megoldásban.

Érdemes nagy körültekintéssel eljárni, mert veszélyeket is rejt magában ez a felírás: legyen a raktározási költség egységesen 30. Ekkor a 3.9. kódban csak a célfüggvény változik a fent részletezett műveletsor alapján:

```
min
ktg:
```

```
-13x1_1 -4x1_2 - 5x1_3 - 2x1_4 + 4x1_5 - 5x1_6
+ 5x2_1 -9x2_2 - 9x2_3 -16x2_4 -17x2_5 -19x2_6
- 3x3_1 -7x3_2 -18x3_3 -12x3_4 + 1x3_5 -16x3_6
-12x4_1 -9x4_2 - 1x4_3 + 9x4_4 -17x4_5 -13x4_6
+ 5x5_1 +7x5_2 - 1x5_3 + 0x5_4 -15x5_5 -18x5_6
```

Az optimális megoldás:

```
  Row name   St   Activity     Lower bound   Upper bound
------------ -- ------------- ------------- -------------
r1           NU           120                         120
r2           NU           120                         120
r3           NU           120                         120
r4           NU           120                         120
r5           NU           120                         120
a1           NL            98            98
a2           NL           136           136
a3           B            111            46
a4           NL            52            52
a5           NL            83            83
a6           B            120            85

Column name  St   Activity
------------ -- -------------
x1_1         B             98
x1_2         B             22
x2_2         B             68
x2_4         B             52
x3_2         B              9
x3_3         B            111
x4_2         B             37
x4_5         B             83
x5_6         B            120
```

Az összes raktárból elszállítottuk az árut; ezzel párhuzamosan a_3 és a_6 áruházak esetén túlteljesítettük az igényt. Ez valahol logikus, olcsóbb elszállítani a raktárakból bizonyos költség mellett az árut (és az áruházakban raktározni 0 költséggel) mint a raktárakban hagyni ennél drágábbáron. Kérdés, hogy ez megtehető-e? Erre a feladat szövegéből kell választ kapnunk, és amennyiben nem egyértelmű, akkor pontosítást kell eszközölni (ez nagyon sokszor az üzleti projekteknél is így működik). A mi szempontunkból az a lényeg, hogy ez a felírás akkor helyes, ha túltejesíthetőek az áruházak igényei.

Ha nem teljesíthetőek túl akkor a fiktív áruházas megoldást kell alkalmazni. Vagy az is egy lehetséges megoldás, ha a 3.9. kódban kicseréljük az áruházakra vonatkozó korlátokat >=-ről =-re.

3.10. példa. *Tekintsük a 3.7. példát! Látjuk, hogy a raktárakból elszállítjuk az összes árut, mert magas a szállítási költség. Tegyük fel, hogy mi vezetjük a 2. raktárat. Mennyit kellene kérni a raktározási díjért (mennyivel kellene csökkenteni a jelenlegi díjat), hogy ebben a raktárban valamennyi árut tároljon a cég (azaz valamekkora raktározási díjra szert tegyen a raktárt üzemeltető cég)?*

Megoldás.

Ha jobban belegondolunk ez egy viszonylag egyszerű kérdés, tulajdonképpen a 2. raktár redukált költségére vagyunk kíváncsiak, ami az `x2_f` változó redukált költsége, ami az eredményfülön a `Marginal` oszlopban található.

```
   No. Column name  St   Activity
------ ------------ -- -------------
    14 x2_f         NL             0

 Lower bound   Upper bound    Marginal
------------- ------------- -------------
            0                           7
```

Tehát 7 egységgel kellene csökkenteni a raktározási költséget, hogy itt is raktározzon a cég, azaz az új díj maximum 12-7=5 lehet.

3.11. példa. *Tekintsük a 3.4. példát (raktározási költség 0)! A használt útvonalakat előre be kell jelenteni, viszont a második áruház keresletében nem vagyunk teljesen biztosak, elképzelhető, hogy valamivel kisebb vagy nagyobb lesz, mint a jelenlegi igény. Meg tudjuk-e mondani, hogy milyen határok között változhat az a_2 áruház kereslete ahhoz, hogy ugyanazokat az útvonalakat használjuk, mint az eredeti optimális szállítási terv esetén (a szállított mennyiség változhat)?*

Megoldás.

Természetesen meg lehet mondani, mivel most egy érzékenységvizsgálati határra kérdez rá a feladat. Ahhoz, hogy a feladatot meg tudjuk válaszolni, most nem célszerű fiktív áruházat bevezetni, mivel akkor a kereslet megváltozásával nem csak egy korlát változik (az a_2 áruház kereslete), hanem a fiktív áruház igénye is. Amennyiben nem vezetünk be fiktív áruházat (3.5. kód esete), akkor a második áruház érzékenységvizsgálati határa a kérdés. Ez megtalálható a 'sensitivity analysis' output állomány `activity range` oszlopában.

```
   No. Row name      St       Activity         Slack
                                            Marginal
------ ------------ -- ------------- -------------
     7 a2           NL     136.00000         .
                                            23.00000

Lower bound        Activity   Limiting
Upper bound           range   variable
-------------   ------------- ------------
    136.00000       105.00000 x3_2
         +Inf       179.00000 r3
```

Tehát a kereslet 105 és 179 között lehet, hogy ugyanezt az úthálózatot használjuk.

Viszont a 3.5. kódhoz tartozó LP feladatnak is van alternatív megoldása. Ebben az esetben a `x3_2` változó értéke 0. Ebben az alternatív optimumban az érzékenységi jelentés is változik, a 2. áruház igénye 105 és 140 között lehet. Sajnos azt, hogy van alternatív megoldása a feladatnak, az output állományból nem láthatjuk, így azt sem tudhatjuk, hogy ha másik bázist használnánk, akkor más (adott esetben nagyobb) lenne a kérdezett intervallum.

3.12. példa. *Tekintsük a 3.7. példát (raktározási költség nem 0)! Tegyük fel, hogy a használt útvonalakat előre be kell jelenteni a közlekedési hatóságnál, azaz csak azokban a viszonylatokban tudok szállítani, amit lejelentettem. A szálított mennyiség változhat de a használt utak halmaza nem (pl. forgalomtervezési okokból). A kelleténél több utat viszont nem lehet bejelenteni, mert ezért büntetés jár. Ezért a 3.7. példához tartozó optimális megoldást lejelentjük, viszont a második áruház keresletében nem vagyunk teljesen biztosak, elképzelhető, hogy valamivel kisebb vagy nagyobb lesz. Meg tudjuk-e mondani, hogy milyen határok között változhat az a_2 áruház igénye ahhoz, hogy ugyanazokat az utakat használjuk az optimális szállítási terv esetén (a szállított mennyiség változhat)?*

Megoldás.

Meg lehet mondani, de óvatosan kell eljárni. Meg lehetne oldani a feladatot fiktív áruház nélkül is, de most a gyakorlás kedvéért oldjuk meg a feladatot a bevezetésével. Arra kell figyelni, hogy ha megváltozik a második áruház kereslete, akkor ezzel együtt a fiktív áruház 'kereslete' is változik. Két korlát együttes megváltozása esetén nincs kész módszertanunk arra, hogy ekkor milyen határok között marad a bázis változatlan. Egy lehetséges út az, hogy a második áruház igényét külön korlátként írjuk elő (`kapa2=136`), és a `kapa2` változónak a segítségével adjuk meg a második és a fiktív áruház keresletét szabályozó egyenleteket. Ezek után a `kapa2=136` korlát érzékenységvizsgálati határa adja meg a kérdésre a választ.

3.13. kód.

```
min
ktg:
+17x1_1 +26x1_2 +25x1_3 +28x1_4 +34x1_5 +25x1_6 +21x1_f
+35x2_1 +21x2_2 +21x2_3 +14x2_4 +13x2_5 +11x2_6 +12x2_f
+27x3_1 +23x3_2 +12x3_3 +18x3_4 +31x3_5 +14x3_6 +27x3_f
+18x4_1 +21x4_2 +29x4_3 +39x4_4 +13x4_5 +17x4_6 +5x4_f
+35x5_1 +37x5_2 +29x5_3 +30x5_4 +15x5_5 +12x5_6 +19x5_f

subject to
kapa2:kapa2=136

r1:+x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_f = 120
r2:+x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 +x2_f = 120
r3:+x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 +x3_f = 120
r4:+x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 +x4_f = 120
r5:+x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 +x5_f = 120

a1:+x1_1 +x2_1 +x3_1 +x4_1 +x5_1 = 98
a2:+x1_2 +x2_2 +x3_2 +x4_2 +x5_2 -kapa2 = 0
a3:+x1_3 +x2_3 +x3_3 +x4_3 +x5_3 = 46
a4:+x1_4 +x2_4 +x3_4 +x4_4 +x5_4 = 52
a5:+x1_5 +x2_5 +x3_5 +x4_5 +x5_5 = 83
a6:+x1_6 +x2_6 +x3_6 +x4_6 +x5_6 = 85
af:+x1_f +x2_f +x3_f +x4_f +x5_f + kapa2 = 236

end
```

Most csak az érzékenységvizsgálati output érdekel minket.

```
Row name      St       Activity      Activity
                                        range
------------ -- ------------- -------------
kapa2         NS     136.00000      116.00000
                                    164.00000
```

Tehát 116 és 164 között változhat a második áruház igénye ahhoz, hogy ne változzon a bázis (a használt útvonalak). Megint hangsúlyozzuk, hogy az eredmény erre a bázisra vonatkozik, lehet most is alternatív optimum, aminek a bázisára másik intervallum vonatkozna.

Térjünk át egy más típusú probléma vizsgálatára. Mi a helyzet akkor, ha az áruházak igényeinek összege nagyobb, mint az összes raktár kapacitása? A kérdést matematikai szempontból elég röviden meg lehet válaszolni: nincs lehetséges megoldás. Az üzleti élet viszont kezeli ezeket a helyzeteket is, ilyenkor kötbért kell fizetni, amelyeket szokás büntető tarifának is nevezni. A kötbér összege különbözhet áruházanként. A kérdés ebben az esetben is az, hogy mekkora az összköltség, ami viszont most a szállítási költség és a kötbér összege.

Ha a szállítási költség magasabb, mint a büntetőtarifa, akkor elképzelhető, hogy érdemesebb kifizetni a kötbér összegét, még akkor is, ha egyébként legalább részben ki tudnánk elégíteni a keresletet. Kérdés, hogy ez megtehető-e? Megvizsgáljuk a kérdést mind a két esetben.

3.14. példa. *Tekintsük a következő költségmátrixot:*

$$\begin{array}{ccccc}
17 & 30 & 28 & 12 & 27 \\
30 & 16 & 21 & 18 & 18 \\
26 & 27 & 20 & 29 & 26 \\
16 & 19 & 10 & 23 & 13 \\
19 & 19 & 26 & 10 & 21 \\
13 & 22 & 20 & 13 & 19
\end{array} .$$

Minden raktárban 100 egységnyi áru van, az áruházak igényei rendre 146, 151, 69, 235 és 174. A ki nem elégített igények után kötbér fizetésre kötelezett a cég. Az áruházaknak fizetendő kötbér összege (egységenként) rendre 40, 41, 57, 49, 43. A *raktárakban nem maradhat áru, az igényeket legalább addig ki kell elégíteni, amíg lehetséges. Mennyi az összköltség minimuma?*

Megoldás.

Ha a raktárakban nem maradhat áru, akkor a szokásos megoldás az, hogy bevezetünk egy fiktív raktárat (most is `f` index jelenti a fiktív raktárt). Ennek a raktárnak a kapacitása megint az áruházok keresletének összege minusz a raktárok kapacitásának összege (esetünkben 146+146+151+69+235+174-6*100=175). Ezen fiktív raktárból 'elégítjük ki' a kötbéreket, ennek költsége így épp a kötbér.

3.15. kód.

```
min
ktg:
+17x1_1 +30x1_2 +28x1_3 +12x1_4 +27x1_5
+30x2_1 +16x2_2 +21x2_3 +18x2_4 +18x2_5
+26x3_1 +27x3_2 +20x3_3 +29x3_4 +26x3_5
+16x4_1 +19x4_2 +10x4_3 +23x4_4 +13x4_5
+19x5_1 +19x5_2 +26x5_3 +10x5_4 +21x5_5
+13x6_1 +22x6_2 +20x6_3 +13x6_4 +19x6_5
+40xf_1 +41xf_2 +57xf_3 +49xf_4 +43xf_5

subject to
r1: +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 = 100
r2: +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 = 100
r3: +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 = 100
r4: +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 = 100
r5: +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 = 100
r6: +x6_1 +x6_2 +x6_3 +x6_4 +x6_5 = 100
rf: +xf_1 +xf_2 +xf_3 +xf_4 +xf_5 = 175

a1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1 +xf_1 = 146
a2: +x1_2 +x2_2 +x3_2 +x4_2 +x5_2 +x6_2 +xf_2 = 151
a3: +x1_3 +x2_3 +x3_3 +x4_3 +x5_3 +x6_3 +xf_3 =  69
a4: +x1_4 +x2_4 +x3_4 +x4_4 +x5_4 +x6_4 +xf_4 = 235
a5: +x1_5 +x2_5 +x3_5 +x4_5 +x5_5 +x6_5 +xf_5 = 174

end
```

A feladat optimális megoldása:

```
Column name  St   Activity     Lower bound   Upper bound
------------ -- ------------- ------------- -------------
x1_4         B            100             0
x2_2         B            100             0
x3_3         B             69             0
x3_5         B             31             0
x4_5         B            100             0
x5_4         B            100             0
x6_1         B             65             0
x6_4         B             35             0
xf_1         B             81             0
xf_2         B             51             0
xf_5         B             43             0
```

A költségminimum pedig 15766. Az első, a második és az ötödik áruház keresletét nem elégítjük ki.

3.16. példa. *A 3.14. példa szövegében ejtsünk annyi változást, hogy a kötbérek mértéke legyen egységesen 25, illetve legyen lehetséges, hogy a raktárakban maradhat áru, ha kifizetjük a kötbért helyette. Mennyi most az összköltség minimuma?*

Megoldás.

A 3.15. kódon nem sok változást kell eszközölni. A célfüggvény esetében aktualizáljuk az együtthatókat, a raktárakra vonatkozó korlátokat pedig megváltoztatjuk =-ről <=-re. A fiktív raktár esetében nincs szükség korlátra, mert az összeg lehet nagyobb is, mint 175.

3.17. kód.

```
min
ktg:
+17x1_1 +30x1_2 +28x1_3 +12x1_4 +27x1_5
+30x2_1 +16x2_2 +21x2_3 +18x2_4 +18x2_5
+26x3_1 +27x3_2 +20x3_3 +29x3_4 +26x3_5
+16x4_1 +19x4_2 +10x4_3 +23x4_4 +13x4_5
+19x5_1 +19x5_2 +26x5_3 +10x5_4 +21x5_5
+13x6_1 +22x6_2 +20x6_3 +13x6_4 +19x6_5
+25xf_1 +25xf_2 +25xf_3 +25xf_4 +25xf_5

subject to
r1: +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 <= 100
r2: +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 <= 100
r3: +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 <= 100
r4: +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 <= 100
r5: +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 <= 100
r6: +x6_1 +x6_2 +x6_3 +x6_4 +x6_5 <= 100

a1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1 +xf_1 = 146
a2: +x1_2 +x2_2 +x3_2 +x4_2 +x5_2 +x6_2 +xf_2 = 151
a3: +x1_3 +x2_3 +x3_3 +x4_3 +x5_3 +x6_3 +xf_3 =  69
a4: +x1_4 +x2_4 +x3_4 +x4_4 +x5_4 +x6_4 +xf_4 = 235
a5: +x1_5 +x2_5 +x3_5 +x4_5 +x5_5 +x6_5 +xf_5 = 174

end
```

Az optimális megoldás:

```
  Row name    St   Activity     Lower bound   Upper bound
------------ -- ------------- ------------- -------------
r1           NU           100                         100
r2           NU           100                         100
r3           B             69                         100
r4           NU           100                         100
r5           NU           100                         100
r6           NU           100                         100

Column name  St   Activity
------------ -- -------------
xf_1         B             46
xf_2         B             51
xf_3         NL             0
xf_4         B             35
xf_5         B             74
```

Tehát a 3. raktárban marad áru; egyedül a harmadik áruház keresletét elégítjük ki teljes egészében, a többiét csak részben, tehát valamennyi kötbért fizetünk. Az összköltség 12930.

3.18. példa. *A 3.16. példa szövegéhez még annyit hozzáteszünk, hogy maximum két áruház esetében engedhetjük meg, hogy kötbért fizetünk. Hogyan változik a költségminimum?*

Megoldás.

Az `xf` változók közül csak kettő lehet pozitív. Ehhez bináris döntési változókat kell bevezetni. Legyen B_i változók értéke 1, ha az a_i áruház esetében fizetünk kötbért, és 0, ha nem. Most B_i és x_{fi} változók jelentése összefügg, a köztük lévő koherenciát korlátokkal kell kiegészíteni: $x_{fi} \leq M \times B_i$ (M értékének az adott áruház keresletét tehetjük). Ezek után már csak azt kell előírni, hogy a B bináris változók összege maximum 2 lehet.

3.19. kód.

```
min
ktg:
+17x1_1 +30x1_2 +28x1_3 +12x1_4 +27x1_5
+30x2_1 +16x2_2 +21x2_3 +18x2_4 +18x2_5
```

```
+26x3_1 +27x3_2 +20x3_3 +29x3_4 +26x3_5
+16x4_1 +19x4_2 +10x4_3 +23x4_4 +13x4_5
+19x5_1 +19x5_2 +26x5_3 +10x5_4 +21x5_5
+13x6_1 +22x6_2 +20x6_3 +13x6_4 +19x6_5
+25xf_1 +25xf_2 +25xf_3 +25xf_4 +25xf_5

subject to
r1: +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 <= 100
r2: +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 <= 100
r3: +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 <= 100
r4: +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 <= 100
r5: +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 <= 100
r6: +x6_1 +x6_2 +x6_3 +x6_4 +x6_5 <= 100

a1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1 +xf_1 = 146
a2: +x1_2 +x2_2 +x3_2 +x4_2 +x5_2 +x6_2 +xf_2 = 151
a3: +x1_3 +x2_3 +x3_3 +x4_3 +x5_3 +x6_3 +xf_3 =  69
a4: +x1_4 +x2_4 +x3_4 +x4_4 +x5_4 +x6_4 +xf_4 = 235
a5: +x1_5 +x2_5 +x3_5 +x4_5 +x5_5 +x6_5 +xf_5 = 174

b1: xf_1 - 146B1 <= 0
b2: xf_2 - 151B2 <= 0
b3: xf_3 -  69B3 <= 0
b4: xf_4 - 235B4 <= 0
b5: xf_5 - 174B5 <= 0

bsum: B1 + B2 + B3 + B4 + B5 <=2

binary
B1
B2
B3
B4
B5

end
```

Az outputot nem szerepeltetjük, de ismertetjük a feladat optimális megoldását: az első két áruház esetében fizetünk kötbért. Az összköltség így 13047 lesz. A raktárakból az összes árut elszállítjuk (ezt nem kellett volna feltétlenül megtenni de az összköltség így lesz minimális).

3.20. példa. *Az alábbi mátrix egy szállítási feladat során a* **nyereségeket**

tartalmazza:

$$\begin{array}{rrrrrr} -7 & 2 & 8 & 28 & 29 & 17 \\ -9 & -3 & 20 & 11 & 14 & 24 \\ -2 & 10 & 12 & 17 & 2 & 3 \\ -1 & 30 & 27 & 0 & 6 & 12 \\ 14 & 29 & 0 & 24 & 16 & 1 \\ 1 & 19 & 4 & 19 & -4 & 14 \end{array}.$$

A mátrixban a negatív értékek azt jelentik, hogy az adott viszonylatban nem lehetséges nyereségesen szállítani. Minden raktárban 100 egységnyi áru van, minden áruház igénye 90. Adja meg a nyereségmaximumot!

Megoldás.

A feladat szövege nincs egyértelműen megfogalmazva. Kérdés, hogy az igényeket ki kell-e elégíteni feltétlenül és lehet-e túlteljesíteni őket. Mi most azt feltételezzük, hogy az áruházak igényét pontosan ki kell elégíteni, sem kevesebb, sem több nem lehet. Ekkor a célfüggvényt maximalizáljuk, az áruházakra vonatkozó korlátok egyenlőség formájában teljesülnek, a raktárakra vonatkozó korlátok pedig kisebb vagy egyenlő korlátok lesznek.

3.21. kód.

```
max
nyer:
- 7x1_1 + 2x1_2 + 8x1_3 +28x1_4 +29x1_5 +17x1_6
- 9x2_1 - 3x2_2 +20x2_3 +11x2_4 +14x2_5 +24x2_6
- 2x3_1 +10x3_2 +12x3_3 +17x3_4 + 2x3_5 + 3x3_6
- 1x4_1 +30x4_2 +27x4_3 + 0x4_4 + 6x4_5 +12x4_6
+14x5_1 +29x5_2 + 0x5_3 +24x5_4 +16x5_5 + 1x5_6
+ 1x6_1 +19x6_2 + 4x6_3 +19x6_4 - 4x6_5 +14x6_6

subject to
r1: +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 <= 100
r2: +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 <= 100
r3: +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 <= 100
r4: +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 <= 100
r5: +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 <= 100
r6: +x6_1 +x6_2 +x6_3 +x6_4 +x6_5 +x6_6 <= 100

a1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1 = 90
```

```
a2: +x1_2 +x2_2 +x3_2 +x4_2 +x5_2 +x6_2 = 90
a3: +x1_3 +x2_3 +x3_3 +x4_3 +x5_3 +x6_3 = 90
a4: +x1_4 +x2_4 +x3_4 +x4_4 +x5_4 +x6_4 = 90
a5: +x1_5 +x2_5 +x3_5 +x4_5 +x5_5 +x6_5 = 90
a6: +x1_6 +x2_6 +x3_6 +x4_6 +x5_6 +x6_6 = 90

end
```

A maximális nyereség ebben az esetben 12140.

3.22. példa. *Az alábbi mátrix egy szállítási feladat során a nyereségeket tartalmazza:*

$$\begin{matrix} 29 & 56 & 53 & M & 29 & 15 \\ 16 & M & 12 & 41 & 64 & M \\ 27 & 21 & 40 & M & 41 & 20 \\ M & 18 & 37 & 40 & 17 & 53 \end{matrix}$$

A mátrixban szereplő M értékek azt jelzik, hogy az adott viszonylatban nem lehetséges a szállítás. Ezt szokás tiltótarifaként is említeni. Minden raktárban 150 egységnyi áru van, minden áruház igénye 100. Adja meg a nyereségmaximumot biztosító szállítási tervet!

Megoldás.

A tiltótarifák kezelése LP felírás esetén nem okoz gondot. Két lehetőség is kínálkozik. Vagy utólag kinullázzuk azokat az x változókat, ahol nem lehetséges a szállítás (tehát a mi esetünkben `x1_4=0`, `x2_2=0`, stb ...) vagy eleve nem is használjuk ezeket a változókat. Nézzük ez utóbbi megközelítést:

3.23. kód.

```
max
nyer:
+29x1_1 +56x1_2 +53x1_3         +29x1_5 +15x1_6
+16x2_1         +12x2_3 +41x2_4 +64x2_5
+27x3_1 +21x3_2 +40x3_3         +41x3_5 +20x3_6
        +18x4_2 +37x4_3 +40x4_4 +17x4_5 +53x4_6

subject to
r1: +x1_1 +x1_2 +x1_3       +x1_5 +x1_6 = 150
r2: +x2_1       +x2_3 +x2_4 +x2_5       = 150
```

```
r3: +x3_1 +x3_2 +x3_3       +x3_5 +x3_6 = 150
r4:       +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 = 150

a1: +x1_1 +x2_1 +x3_1       = 100
a2: +x1_2       +x3_2 +x4_2 = 100
a3: +x1_3 +x2_3 +x3_3 +x4_3 = 100
a4:       +x2_4       +x4_4 = 100
a5: +x1_5 +x2_5 +x3_5 +x4_5 = 100
a6: +x1_6       +x3_6 +x4_6 = 100

end
```

Az optimális megoldás:

```
   No. Column name  St   Activity
------ ------------ -- -------------
     2 x1_2         B            100
     3 x1_3         B             50
     8 x2_4         B             50
     9 x2_5         B            100
    10 x3_1         B            100
    12 x3_3         B             50
    17 x4_4         B             50
    19 x4_6         B            100
```

Fontos hangsúlyozni, hogy tiltótarifák megjelenésével nem lehet garantálni, hogy mindig lesz lehetséges szállítási terv, nézzük mintaként a 3.24. példát.

3.24. példa. *Az alábbi mátrix egy szállítási feladat során a költségeket tartalmazza:*

$$\begin{matrix} 29 & 56 \\ 16 & M \end{matrix}$$

Az első raktárban 100 egységnyi áru van, a másodikban 200. Az áruházak igényei 150 egység áruházanként.

Megoldás.

3.25. kód.

```
min
ktg:
+29x1_1 +56x1_2
+16x2_1
```

```
subject to
r1: +x1_1 +x1_2 = 100
r2: +x2_1       = 200

a1: +x1_1 +x2_1 = 150
a2: +x1_2       = 150

end
```

Nem kapunk outputot, nincs lehetséges megoldás.

3.26. példa. *Az alábbi mátrix egy szállítási feladat során a költségeket tartalmazza:*

$$
\begin{matrix}
\mathbf{43} & 50 & 125 & 50 & 60 & 90 \\
39 & \mathbf{64} & 113 & 109 & \mathbf{47} & \mathbf{72} \\
80 & 74 & \mathbf{44} & 90 & \mathbf{53} & 47 \\
\mathbf{90} & \mathbf{38} & 52 & \mathbf{35} & 103 & \mathbf{86} \\
\mathbf{126} & 43 & 75 & 95 & 43 & 68 \\
76 & 82 & 44 & \mathbf{59} & 121 & 77
\end{matrix} .
$$

Minden raktárban 100 egység áru van, és ennyi minden áruház igénye is. A mátrixban vastaggal jelzett számok azt jelentik, hogy az adott viszonylatban autópálya is elérhető. Autópálya használata esetén a költségek 20%-kal kisebbek, de autópályát csak akkor lehet igénybe venni, ha beruházunk autópálya matricába, aminek ára 4000 egység (ami feljogosít minket arra, hogy az összes autópályát használjuk). Adja meg az összköltség minimumát (ha igénybe veszünk autópályát, akkor az összköltség tartalmazza az autópálya matrica árát is)!

Megoldás.

A feladatot természetesen meg lehet oldani úgy is, hogy két szállítási feladatot oldunk meg. Egyet autópálya használat nélkül, egyet pedig az igénybevételével, és megnézzük, hogy melyik esetben lesz kisebb az összköltség. Kérdés, hogy tudjuk egyetlen futtatással megoldani a feladatot?

Jelöljék x_{ij} változók az i raktárból a j áruházba szállított mennyiséget közúton. Jelöljék y_{ij} változók az i raktárból a j áruházba szállított mennyiséget autópályán (természetesen csak azokban a viszonylatokban, ahol van autópálya). Jelölje B bináris változó, hogy veszünk-e autópálya matricát (1, ha veszünk; 0 különben). A költségfüggvényt könnyen fel lehet írni. Viszont problémát okozhat, ha B változó értéke 0, és emellett valamelyik y_{ij} változó pozitív. Ekkor az autópályát igénybe vesszük anélkül, hogy kifizetnénk a érte a matrica árát. Ez természetesen nem megengedett, korláttal kell kizárni ezt az esetet:

$$\sum y_{ij} \leq 600B$$

Most már felírható a feladat CPXLP formátumban (a CPLEXLP kódban `B` nem lehet változónév, ezért `BB` változónevet fogunk használni):

3.27. kód.

```
min
ktg:
+ 43x1_1 +50x1_2 +125x1_3 + 50x1_4 + 60x1_5 +90x1_6
+ 39x2_1 +64x2_2 +113x2_3 +109x2_4 + 47x2_5 +72x2_6
+ 80x3_1 +74x3_2 + 44x3_3 + 90x3_4 + 53x3_5 +47x3_6
+ 90x4_1 +38x4_2 + 52x4_3 + 35x4_4 +103x4_5 +86x4_6
+126x5_1 +43x5_2 + 75x5_3 + 95x5_4 + 43x5_5 +68x5_6
+ 76x6_1 +82x6_2 + 44x6_3 + 59x6_4 +121x6_5 +77x6_6

+ 34.4y1_1
           +51.2y2_2           +37.6y2_5 +57.6y2_6
                     +35.2y3_3 +42.4y3_5
+ 72  y4_1 +30.4y4_2 +28  y4_4           +68.8y4_6
+100.8y5_1
                     +47.2y6_4
+4000BB

subject to
r1: +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6
    +y1_1                          = 100

r2: +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6
          +y2_2             +y2_5 +y2_6 = 100

r3: +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6
                +y3_3       +y3_5       = 100

r4: +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6
```

```
    +y4_1 +y4_2       +y4_4       +y4_6 = 100

r5: +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6
    +y5_1                               = 100

r6: +x6_1 +x6_2 +x6_3 +x6_4 +x6_5 +x6_6
                      +y6_4             = 100

a1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1
    +y1_1             +y4_1 +y5_1       = 100

a2: +x1_2 +x2_2 +x3_2 +x4_2 +x5_2 +x6_2
          +y2_2 +y4_2                   = 100

a3: +x1_3 +x2_3 +x3_3 +x4_3 +x5_3 +x6_3
                +y3_3                   = 100

a4: +x1_4 +x2_4 +x3_4 +x4_4 +x5_4 +x6_4
                +y4_4 +y6_4             = 100

a5: +x1_5 +x2_5 +x3_5 +x4_5 +x5_5 +x6_5
          +y2_5 +y3_5                   = 100

a6: +x1_6 +x2_6 +x3_6 +x4_6 +x5_6 +x6_6
          +y2_6 +y4_6                   = 100

+y1_1 +y2_2 +y2_5 +y2_6 +y3_3 +y3_5
+y4_1 +y4_2 +y4_4 +y4_6 +y5_1 +y6_4
-600BB <=0

binary
BB

end
```

Nem érdemes igénybe venni az autópályát, közúton szállítunk, az összköltség 25800. Újabb futtatások segítségével megállapítható, hogy ha az autópálya matrica ára csak 2300 lenne, akkor viszont már érdemes lenne azon is szállítani.

3.2. Összetett vagy átrakodásos szállítási feladat

Összetett vagy átrakodásos feladat esetén –amint a neve is mutatja– vannak átrakodási pontok is t_1, t_2, ..., t_k. Az átrakodási pontok halmazát T-vel jelöljük. Alapesetben szállítás a raktárak és az átrakodási pontok, illetve az átrakodási pontok és az áruházak között lehetséges.

Összetett feladatok felírására két lehetőséget is megmutatunk. Első lehetőségként bemutatjuk, hogyan lehet visszavezetni az átrakodásos feladatot egyszerű szállítási feladattá. Ennek a módja az, hogy az átrakodási pontokat szerepeltetjük az áruházak valamint a raktárak között is. Ekkor kapunk egy nagy költségmátrixot, de most bizonyos viszonylatokban nem lehetséges a szállítás. Egyetlen kérdés van még, mennyi legyen az átrakodási pontok kapacitása (és igénye)? Gond-e a túl nagyra méretezett kapacitás? A t_i átrakodási pontból, mint raktárból a t_i átrakodási pontba, mint áruházba 0 költséggel tudunk 'szállítani'. Ha esetleg elvétettük a kapacitást, akkor a 'felesleg' itt fog lecsapódni, ami így nem fog költségemelkedéssel járni. Tehát, ha nagyobb számot írunk mint szükséges, nem fog problémát okozni. A raktárakban tárolt árumennyiség összegénél (vagy esetleg az áruházak igényeinek összegénél, ha büntetőtarifákról is szó van) biztos nem fog több áru keresztüláramolni egyetlen átrakodási ponton sem. Ez a mennyiség fog szerepelni a az átrakodási pontok kapacitásánál (és igényénél).

Második lehetőség, ha az átrakodási pontokra egyszerűen megadunk egy csomóponti feltételt. Ez a feltétel azt mondja ki, hogy a csomópontba beérkező árumennyiségnek, és az innen kiáramló mennyiségnek meg kell egyeznie.

Mielőtt képletszerűen is felírnánk ezt a feltételt vissza kell térnünk a jelölés kérdésére. Az egyszerű szállítási feladatnál megengedhettük, hogy az $x_{r_i a_j}$ jelölés helyett egyszerűen a x_{ij} jelölést használjuk, de átrakodásos szállítási feladatnál ez már nem ilyen egyszerű. Az $x_{1,2}$ változó jelentheti az $x_{r_1 t_2}$ változót is, tehát az első raktárból az első átrakodási pontba szállított mennyiséget, és jelentheti $x_{t_1 a_2}$ változót is, tehát az első átrakodási pontból a második áruházba szállított mennyiséget. Szeretnénk továbbra is a kettős indexet elkerülni, ezért az az raktárak indexei: 1, 2, ..., n, az átrakodási pontok indexei $n+1$, $n+2$, ..., $n+k$, az áruházak indexei pedig

$n+k+1$, $n+k+2$, ..., $n+k+m$. Tehát minden szám csak egyszer fog előfordulni. Valószínűleg $x_{1,2}$ változónk nem lesz, hacsak nem egyetlen raktár szerepel csak a feladatban. Az egyszerűség (és a jobb követhetőség) kedvéért ezeket az indexeket fogjuk a visszavezetett feladat esetén is használni.

Ezek után a csomóponti feltétel képlettel:

$$\sum_{i=1}^{n} x_{i\ell} = \sum_{j=n+k+1}^{n+k+m} x_{\ell j} \qquad \ell = n+1,\ \ldots,\ n+k\ .$$

3.28. példa. *Van 3 raktárunk, 4 átrakodási pontunk és 5 áruházunk. A C mátrix megadja a szállítási költségeket a raktárak és az átrakodási pontok között, a D mátrix pedig az átrakodási pontok és az áruházak közötti szállítási költségeket adja meg.*

$$C = \begin{pmatrix} 28 & 27 & 28 & 35 \\ 34 & 28 & 31 & 33 \\ 16 & 31 & 14 & 19 \end{pmatrix}, \qquad D = \begin{pmatrix} 21 & 32 & 33 & 57 & 31 \\ 59 & 55 & 29 & 39 & 64 \\ 38 & 54 & 66 & 46 & 49 \\ 52 & 25 & 64 & 45 & 25 \end{pmatrix}.$$

A raktárakban 200 egység áru van, az áruházak igénye pedig 120. Adja meg a minimális költségű szállítási tervet!

Megoldás (visszavezetés egyszerű szállítási feladattá).

A raktárak lesznek az 1-3, az átrakodási pontok a 4-7, az áruházak pedig 8-12 csomópontok. A visszavezetett feladatban a költségmátrixnak 7 sora és 5 oszlopa lesz. Az első 3 sor a raktárakra vonatkozik, a második 4 sor pedig az átrakodási pontokra. Az első 4 oszlop szintén az átrakodási pontokra vonatkozik, a második 5 oszlop pedig az áruházakra.

28	27	28	35	M	M	M	M	M
34	28	31	33	M	M	M	M	M
16	31	14	19	M	M	M	M	M
0	M	M	M	21	32	33	57	31
M	0	M	M	59	55	29	39	64
M	M	0	M	38	54	66	46	49
M	M	M	0	52	25	64	45	25

A sorok kapacitása rendre 200, 200, 200, 600, 600, 600 és 600. Az oszlopok igénye rendre 600, 600, 600, 600, 120, 120, 120, 120 és 120. Most már csak fel kell írni a feladatot CPXLP formátumban:

3.29. kód.

```
min
ktg:
+28x1_4 +27x1_5 +28x1_6 +35x1_7
+34x2_4 +28x2_5 +31x2_6 +33x2_7
+16x3_4 +31x3_5 +14x3_6 +19x3_7
+0x4_4      +21x4_8 +32x4_9 +33x4_10 +57x4_11 +31x4_12
 +0x5_5     +59x5_8 +55x5_9 +29x5_10 +39x5_11 +64x5_12
  +0x6_6    +38x6_8 +54x6_9 +66x6_10 +46x6_11 +49x6_12
   +0x7_7   +52x7_8 +25x7_9 +64x7_10 +45x7_11 +25x7_12

subject to
r1:  +x1_4 +x1_5 +x1_6 +x1_7                      = 200
r2:  +x2_4 +x2_5 +x2_6 +x2_7                      = 200
r3:  +x3_4 +x3_5 +x3_6 +x3_7                      = 200
t1r: +x4_4    +x4_8 +x4_9 +x4_10 +x4_11 +x4_12 = 600
t2r:  +x5_5   +x5_8 +x5_9 +x5_10 +x5_11 +x5_12 = 600
t3r:   +x6_6  +x6_8 +x6_9 +x6_10 +x6_11 +x6_12 = 600
t4r:    +x7_7 +x7_8 +x7_9 +x7_10 +x7_11 +x7_12 = 600

t1a: +x1_4 +x2_4 +x3_4 +x4_4                          = 600
t2a: +x1_5 +x2_5 +x3_5       +x5_5                    = 600
t3a: +x1_6 +x2_6 +x3_6             +x6_6              = 600
t4a: +x1_7 +x2_7 +x3_7                   +x7_7        = 600
a1:                    +x4_8  +x5_8  +x6_8  +x7_8     = 120
a2:                    +x4_9  +x5_9  +x6_9  +x7_9     = 120
a3:                    +x4_10 +x5_10 +x6_10 +x7_10    = 120
```

```
a4:                    +x4_11 +x5_11 +x6_11 +x7_11 = 120
a5:                    +x4_12 +x5_12 +x6_12 +x7_12 = 120

end
```

A feladat optimális megoldása: az összköltség 31800, a szállított mennyiségek pedig:

```
   No. Column name  St   Activity
------ ------------ -- -------------
     1 x1_4         B            120
     2 x1_5         B             80
     6 x2_5         B            160
     8 x2_7         B             40
    12 x3_7         B            200
    13 x4_4         B            480
    14 x4_8         B            120
    19 x5_5         B            360
    22 x5_10        B            120
    23 x5_11        B            120
    25 x6_6         B            600
    31 x7_7         B            360
    33 x7_9         B            120
    36 x7_12        B            120
```

Tehát az első raktárból az első átrakodási pontra szállítunk 120 egységet, ugyanebből a raktárból a második átrakodási pontra 80 egységet, stb. A 120 egységet ami az első átrakodási pontra érkezik továbbszállítjuk az első áruházba (`x4_8=120`). `x4_4` változó értéke 480, ami logikus is, hiszen a 600-as kapacitásból csak 120 egységet 'használunk' el; 480 pedig kihasználatlan marad. Ezt a mennyiséget 'elszállítjuk' az első átrakodási pontból az első átrakodási pontba, tehát itt csapódik le a felesleges kapacitás. Ha az átrakodási pont kapacitását nem 600-nak, hanem 700-nak állítottuk volna be, akkor egyszerűen csak az `x4_4` változó is 580-ra növekedne.

Megoldás (csomóponti feltételek).

Nézzük meg, miképpen alakul a felírás, ha nem vezetjük vissza egyszerű szállítási feladattá, hanem az ún. csomóponti feltételeket használjuk. A célfüggvény lényegileg ugyanaz, mint a 3.29. kód esetén. Az `r1`, `r2`, `r1` raktárakra vonatkozó korlátok, azt fejezik ki, hogy a raktárakból nem szállíthatunk el több árut, mint ami ott van. Hasonlóan az `a8-a12` áruházakra vonatkozó korlátok, azt fejezik ki, hogy az áruházak legalább annyit kapjanak, mint amennyit igényeltek. A csomóponti korlátok (`t4-t7`) azt fejezik ki, hogy egy adott átrakodási pontban nem keletkezhet és szűnhet meg áru, tehát amennyi árut odaszállítunk, annyit el is kell szállítani. Az első átrakodási pont esetében $x_{1,4} + x_{2,4} + x_{3,4}$ mennyiséget szállítunk oda, és $x_{4,8} + x_{4,9} + x_{4,10} + x_{4,11} + x_{4,12}$ mennyiséget szállítunk el. A két összegnek meg kell egyeznie.

3.30. kód.

```
min
ktg:
+28x1_4 +27x1_5 +28x1_6 +35x1_7
+34x2_4 +28x2_5 +31x2_6 +33x2_7
+16x3_4 +31x3_5 +14x3_6 +19x3_7
+21x4_8 +32x4_9 +33x4_10 +57x4_11 +31x4_12
+59x5_8 +55x5_9 +29x5_10 +39x5_11 +64x5_12
+38x6_8 +54x6_9 +66x6_10 +46x6_11 +49x6_12
+52x7_8 +25x7_9 +64x7_10 +45x7_11 +25x7_12

subject to
r1: +x1_4 +x1_5 +x1_6 +x1_7 = 200
r2: +x2_4 +x2_5 +x2_6 +x2_7 = 200
r3: +x3_4 +x3_5 +x3_6 +x3_7 = 200

t4: +x1_4 +x2_4 +x3_4
      -1x4_8 -x4_9 -x4_10 -x4_11 -x4_12 = 0

t5: +x1_5 +x2_5 +x3_5
      -x5_8 -x5_9 -x5_10 -x5_11 -x5_12 = 0

t6: +x1_6 +x2_6 +x3_6
      -x6_8 -x6_9 -x6_10 -x6_11 -x6_12 = 0

t7: +x1_7 +x2_7 +x3_7
      -x7_8 -x7_9 -x7_10 -x7_11 -x7_12 = 0

a8:  +x4_8  +x5_8  +x6_8  +x7_8  = 120
a9:  +x4_9  +x5_9  +x6_9  +x7_9  = 120
a10: +x4_10 +x5_10 +x6_10 +x7_10 = 120
a11: +x4_11 +x5_11 +x6_11 +x7_11 = 120
a12: +x4_12 +x5_12 +x6_12 +x7_12 = 120

end
```

Lényegileg most is ugyanazt az eredményt kaptuk, mint az előző esetben, csak az indexek különböznek.

```
   No. Column name  St   Activity
------ ------------ -- -------------
     1 x1_4         B            120
     2 x1_5         B             80
     6 x2_5         B            160
     8 x2_7         B             40
    12 x3_7         B            200
    13 x4_8         B            120
    20 x5_10        B            120
    21 x5_11        B            120
    29 x7_9         B            120
    32 x7_12        B            120
```

Tehát pl. most is az első raktárból szállítunk az első átrakodási pontra 120 egységet, amit továbbszállítunk az első áruházba.

Nyilván átrakodásos feladat esetén is lehet kezelni az összes olyan jelenséget (nem kiegyensúlyozott feladat, büntetőtarifa, nyereségmaximalizálás, tiltótarifa stb.) amit egyszerű szállítási feladat esetén bemutattunk; közülük párat mutat be a 3.31. példa.

3.31. példa. *Változtassunk a 3.28. példa szövegén. Az áruházak igénye legyen csak 110. Továbbá az első raktárból tudunk az első áruházba szállítani közvetlenül is 70-es egységáron. Az első átrakodási pontban is van 20 egységnyi áru, amit el lehet szállítani, de nem kötelező. Adja meg az összköltség minimumát!*

Megoldás (visszavezetés).

A következő átalakításokat kell elvégezni. Mivel nem kiegyensúlyozott a feladat, ezért egy fiktív áruházat be kell vezetni (a költségmátrix 8. oszlopa fog hozzá tartozni) . Ebbe a fiktív áruházba 0 költséggel szállíthatunk. Nyilván az a legegyszerűbb, ha közvetlenül (átrakodási pont használata nélkül) engedélyezzük a szállítást a raktárak és a fiktív áruház között. Hasonlóképpen engedélyezzük a szállítást az első raktár és az első áruház között, 70-es költséggel.

Kérdés, hogy mit tudunk kezdeni azzal, hogy az első átrakodási pontban is van 20 egységnyi áru? Egyrészt, mivel nem kötelező elszállítani, ebből az átrakodási pontból is mutat él 0 költséggel a fiktív áruházba. Másrészt, így már ez az átrakodási pont nem csak átrakodási pont, hanem raktár is, tehát az első átrakodási sor esetén (és csak a sorban, az oszlopban nem!) a kapacitást megnöveljük 20 egységgel.

Mennyi lesz a fiktív áruház igénye? 620 egységnyi készlet van raktáron (az átrakodási pontban lévő 20 egységet is bele kell számolnunk), az áruházak igénye pedig összesen 550. Tehát a fiktív áruház igénye 70.

Így a költségmátrix:

$$\begin{array}{cccc:cccccc}
28 & 27 & 28 & 35 & 70 & M & M & M & M & 0 \\
34 & 28 & 31 & 33 & M & M & M & M & M & 0 \\
16 & 31 & 14 & 19 & M & M & M & M & M & 0 \\
\hdashline
0 & M & M & M & 21 & 32 & 33 & 57 & 31 & 0 \\
M & 0 & M & M & 59 & 55 & 29 & 39 & 64 & M \\
M & M & 0 & M & 38 & 54 & 66 & 46 & 49 & M \\
M & M & M & 0 & 52 & 25 & 64 & 45 & 25 & M
\end{array} .$$

A sorok kapacitásai rendre: 200, 200, 200, 620, 600, 600 és 600. Az oszlopok igényei rendre 600, 600, 600, 600, 110, 110, 110, 110, 110 és 70. A feladathoz tartozó kód az eddigiek alapján már könnyen felírható. Megoldva a feladatot kapjuk az összköltség minimumára a 28320 értéket.

Megoldás (csomóponti feltételek).

Nézzük most azt, hogy lehet csomóponti feltételekkel megoldani a feladatot. Raktárak esetén a korlátok nem egyenlőségekkel adottak, hanem <= korlátok lesznek.

Az első raktárból tudunk az első áruházba szállítani, ezt jelölje egyszerűen egy `x1_8` változó.

Hogyan kezeljük azt, hogy az első átrakodási pontban van 20 egységnyi áru, ami elszállítható, de nem kell feltétlenül elszállítani? Korábban az első átrakodási pontra felírt csomóponti feltétel az alábbi formában szerepelt:

$$x_{14} + x_{24} + x_{34} - (x_{48} + x_{49} + x_{4,10} + x_{4,11} + x_{4,12}) = 0 \ .$$

Ez most nem helytálló. A vizsgált esetben az a helyzet, hogy kevesebbet nem lehet továbbszállítani, mint amennyi áru beérkezett, de maximum 20 egységgel többet viszont lehet. Ezt két korláttal tudjuk felírni:

$$x_{14} + x_{24} + x_{34} - (x_{48} + x_{49} + x_{4,10} + x_{4,11} + x_{4,12}) >= -20$$

és

$$x_{14} + x_{24} + x_{34} - (x_{48} + x_{49} + x_{4,10} + x_{4,11} + x_{4,12}) <= 0 \ .$$

Ha belegondolunk, akkor a második korlát a mi esetünkben nem lényeges, nem éri meg pozitív költséggel elszállítani az átrakodási pontba az árut és otthagyni. Olcsóbb egyszerűen ekkor a raktárban hagyni az árut. Ha lennének pl. raktározási költségek is, akkor adott esetben már fontos feltétel lenne a második korlát is.

A feladat CPLEX LP formátumban:

3.32. kód.

```
min
ktg:
+28x1_4 +27x1_5 +28x1_6 +35x1_7 +70x1_8
+34x2_4 +28x2_5 +31x2_6 +33x2_7
+16x3_4 +31x3_5 +14x3_6 +19x3_7
+21x4_8 +32x4_9 +33x4_10 +57x4_11 +31x4_12
+59x5_8 +55x5_9 +29x5_10 +39x5_11 +64x5_12
+38x6_8 +54x6_9 +66x6_10 +46x6_11 +49x6_12
+52x7_8 +25x7_9 +64x7_10 +45x7_11 +25x7_12

subject to
r1: +x1_4 +x1_5 +x1_6 +x1_7 + x1_8 <= 200
r2: +x2_4 +x2_5 +x2_6 +x2_7 <= 200
r3: +x3_4 +x3_5 +x3_6 +x3_7 <= 200

t4a: +x1_4 +x2_4 +x3_4
      -x4_8 -x4_9 -x4_10 -x4_11 -x4_12 >= -20

t4f: +x1_4 +x2_4 +x3_4
      -x4_8 -x4_9 -x4_10 -x4_11 -x4_12 <= 0

t5: +x1_5 +x2_5 +x3_5
      -x5_8 -x5_9 -x5_10 -x5_11 -x5_12 = 0

t6: +x1_6 +x2_6 +x3_6
```

```
      -x6_8 -x6_9 -x6_10 -x6_11 -x6_12 = 0

t7: +x1_7 +x2_7 +x3_7
      -x7_8 -x7_9 -x7_10 -x7_11 -x7_12 = 0

a8 : +x1_8  +x4_8  +x5_8  +x6_8  +x7_8  = 110
a9 : +x4_9  +x5_9  +x6_9  +x7_9  = 110
a10: +x4_10 +x5_10 +x6_10 +x7_10 = 110
a11: +x4_11 +x5_11 +x6_11 +x7_11 = 110
a12: +x4_12 +x5_12 +x6_12 +x7_12 = 110

end
```

Az optimális megoldás (a számunkra lényeges részei):

```
  Row name   St   Activity     Lower bound   Upper bound
------------ -- ------------- ------------- -------------
r1           NU           200                         200
r2           B            130                         200
r3           NU           200                         200
t4a          NL           -20           -20
t4f          B            -20                           0


Column name  St   Activity
------------ -- -------------
x1_4         B             90
x1_8         NL             0
x2_4         NL             0
x3_4         NL             0
x4_8         B            110
x4_9         NL             0
x4_10        NL             0
x4_11        NL             0
x4_12        NL             0
```

Tehát az első raktárban nem marad áru, a másodikban marad 70 egység, a harmadikban sem marad. Az első átrakodási pontban lévő 20 egységnyi árut érdemes elszállítani; konkrétan az első átrakodási pontba szállítunk 90 egységnyi árut és továbbszállítunk 110 egységet, amint az az eredményfülön is látható.

3.3. Gyakorló feladatok

3.3.1. Egyszerűbb feladatok

3.33. példa. *Adva van az alábbi költségmátrix:*

$$\begin{array}{ccccccc}
53 & 71 & 77 & 74 & 17 & 82 & 80 \\
72 & 41 & 90 & 85 & 72 & 32 & 86 \\
89 & 41 & 49 & 11 & 51 & 35 & 41 \\
66 & 1 & 41 & 38 & 44 & 64 & 68 \\
72 & 20 & 11 & 86 & 24 & 53 & 8 \\
81 & 74 & 31 & 2 & 39 & 90 & 91 \\
28 & 83 & 25 & 31 & 69 & 59 & 90 \\
19 & 38 & 58 & 56 & 65 & 67 & 4 \\
15 & 62 & 23 & 98 & 2 & 69 & 16 \\
16 & 37 & 89 & 76 & 24 & 61 & 67 \\
23 & 68 & 60 & 49 & 56 & 11 & 72
\end{array} \quad .$$

A raktárak kapacitásai rendre: 9, 5, 13, 5, 2, 18, 11, 10, 19, 15, 10; az áruházak igényei pedig: 20, 15, 10, 20, 10, 11 és 31.

Mekkora a minimális összköltség, amivel az áru a raktárakból az áruházakba elszállítható?

Megoldás a 236. oldalon.

3.34. példa. *Tekintsük az alábbi mátrixot, amely egy szállítási feladat költségeit* *tartalmazza:*

13	5	3	5	9	2	4
13	7	3	6	10	2	5
16	10	13	9	13	9	10
25	19	18	12	17	14	11
39	21	22	12	19	21	18
42	23	27	23	21	26	19
43	27	27	23	26	30	38
43	34	27	26	30	43	45
46	39	28	26	33	49	48
51	42	39	27	33	49	48

.

A raktárak kapacitásai rendre: 10, 20, 15, 40, 50, 20, 10, 15, 15, 35; az áruházak igényei pedig: 30, 40, 10, 20, 50, 40 és 40.

Mekkora a minimális összköltség, amivel az áru a raktárakból az áruházakba elszállítható?

Megoldás a 238. oldalon.

3.35. példa. *Az alábbi mátrix jelentse egy szállítási feladat költségeit:*

100	30	14	24	125	120
37	100	33	41	137	104
27	9	100	10	124	118
3	10	18	100	118	115

.

Mind a 4 raktárban 20 egységnyi áru van, az áruházak igényei pedig rendre: 18, 20, 15, 7, 23 és 17. A ki nem elégített szállítások után kötbért kell fizetni, aminek összege az áruházak sorrendjében 11, 17, 21, 9, 13 és 15. Adja meg a minimális összköltséget biztosító szállítási tervet!

Megoldás a 239. oldalon.

3.36. példa. *Az alábbi mátrix jelentse egy szállítási feladat költségeit:*

$$\begin{matrix} 82 & 28 & 37 & 43 & 38 & 78 & 43 \\ 29 & 51 & 89 & 96 & 94 & 67 & 87 \\ 59 & 37 & 97 & 60 & 64 & 49 & 74 \\ 52 & 80 & 55 & 65 & 27 & 30 & 65 \\ 27 & 16 & 68 & 71 & 90 & 13 & 98 \\ 28 & 11 & 42 & 32 & 67 & 42 & 25 \\ 29 & 23 & 97 & 20 & 11 & 41 & 69 \\ 66 & 67 & 80 & 10 & 93 & 53 & 76 \end{matrix} \quad .$$

Minden egyes raktárban 8 egységnyi áru van és minden áruház igénye 7 egység.

Adva van továbbá egy átrakodási pont. A raktárakból közvetlenül lehet szállítani az áruházakba, illetve az átrakodási ponton keresztül is. Ebbe az átrakodási pontba be-illetve kiszállítás költségei azonosak, amit jelöljünk α-val. Legyen az induláskor ennek az értéke 100. Mennyire kell lecsökkenteni α értékét, hogy használjuk az átrakodási pontot? Mennyire kell lecsökkenteni α értékét, hogy minden árut az átrakodási ponton keresztül szállítsuk?

Megoldás a 241. oldalon.

3.3.2. Nehezebb feladatok

3.37. példa. *Egészítsük ki a 3.34. példát: bizonyos viszonylatokban megoldható a vízi szállítás is az alábbi költségeken:*

$$
\begin{array}{rrrrrrr}
M & 2 & 3 & 2 & M & 2 & M \\
6 & M & M & 5 & 8 & 7 & 2 \\
6 & 8 & 5 & 8 & M & 9 & M \\
15 & M & 6 & 9 & 12 & 20 & 4 \\
16 & 14 & 8 & M & 20 & 21 & M \\
M & 14 & M & 17 & 21 & 23 & 9 \\
25 & 21 & 10 & M & 24 & 25 & 11 \\
27 & M & 10 & 26 & 26 & M & 20 \\
30 & 22 & 11 & 27 & 29 & 29 & M \\
30 & 23 & 22 & M & M & 29 & 25
\end{array}
.
$$

A hajópark kiépítésének fix költsége 500 egység, kikötő építésének költsége 100 egység (természetesen vízi szállítás csak akkor lehetséges, ha ki van építve hozzá a vízi szállítás infrastruktúrája: van hozzá hajópark és a kezdőállomáson és a végállomáson is van kikötő). Írjon fel egy vegyes egészértékű lineáris programozási feladatot, amely megadja, hogy érdemes-e vízen szállítani, és ha igen hol!

Megoldás a 245. oldalon.

3.38. példa. *Tekintsük az alábbi költségmátrixot:*

$$
\begin{array}{rrrrrr}
86 & 67 & 90 & 77 & 84 & 87 \\
64 & 98 & 58 & 55 & 65 & 76 \\
60 & 97 & 92 & 61 & 89 & 53 \\
70 & 60 & 82 & 73 & 92 & 87 \\
99 & 53 & 61 & 87 & 91 & 70
\end{array}
$$

Minden raktárban 150 egységnyi áru van, minden áruház igénye 100. Az el nem szállított árumennyiség után raktározási költséget kell fizetni. A raktározási költségek *a raktárak esetében rendre 66, 59, 69, 55 és 80. Ha az áruházakba több árut szállítunk, mint a kapacitása, akkor a fennmaradó mennyiség után is raktározási költséget kell fizetni. A raktározási költség az áruházak esetén (egységenként) rendre 21, 17, 5, 21, 17 és 19. Adja meg a minimális összköltséget (szállítási+raktározási)!*

Megoldás a 251. oldalon.

3.39. példa. *Egy mezőgazdasági vállalat földterületekről (sorok) szeretné elszállítani a megtermelt kukoricát raktárakba (oszlopok). A földterületeken rendre 40, 70, 30, 100, 80 és 20 mázsa kukorica található. Mindegyik raktár kapacitása 80 mázsa. A földekről a raktárakba szállítás költségei az alábbi táblázatban láthatóak:*

70	109	106	99	88
56	**25**	**22**	105	21
56	28	**32**	119	**32**
114	94	73	**33**	116
69	90	56	74	32
85	**30**	87	79	76

A **vastag** *betűvel szedett értékek azt jelentik, hogy ott földút található, az itt szállított kukorica 5%-a elveszik szállítás közben. A mezőgazdasági vállalatnak lehetősége van a raktárakat egy külső piacról is közvetlenül feltölteni, ahol a kukorica mázsája 100 egységbe kerül. Adjon meg egy olyan szállítási tervet, amely minimalizálja az összköltséget!*
Hogyan változik a szállítási terv, ha a földekről mázsákra kerekítve lehet csak meghatározni a szállítást?
Hogyan változik a szállítási terv, ha a vállalat vezetése szerint legalább 330 mázsa kukoricára szükség lesz a következő évben? Hogyan változik, ha 340 mázsa kukoricára lesz szükség a következő évben?

Megoldás a 253. oldalon.

4. fejezet

Maximális folyam feladat

4.1. Alapfeladat

A maximális folyam feladat azt vizsgálja, hogy mekkora egy hálózat áteresztő képessége. A maximális folyam feladat felírását a 4.7. példában adjuk meg. Ahhoz, hogy jobban megértsük a maximális feladatot, érdemes megmutatni a kapcsolódását az átrakodásos (vagy összetett) szállítási feladathoz. Ezért térjünk vissza a szállítási feladatokhoz, tekintsük a 3.28. példát és az ehhez tartozó 'csomópontos' felírást (a 3.30. kódot)! Amikor a szállítási feladatokat modelleztük, akkor egy-egy viszonylatban korlátlan mennyiséget tudtunk szállítani, de a valós problémák megoldásakor ez nem feltétlenül igaz. Például, ha egy teherautó közlekedik két csomópont között, akkor ebbe a teherautóba nem lehet korlátlan mennyiséget rakni, van valamekkora kapacitása a rakterének. Ezért vezessünk be korlátokat arra vonatkozóan, hogy egy-egy viszonylatban mennyi árut lehet szállítani!

4.1. példa. *Bővítsük 3.28. példát a következő korlátozásokkal:*

Honnan	*Hova*	*Kapacitás*
1	*4*	*80*
1	*6*	*60*
2	*7*	*50*
3	*5*	*90*
4	*8*	*100*
5	*9*	*70*
6	*12*	*80*

Tehát az 1-es pontból (ami egy raktár) a 4-es pontba (ami egy átrakodási pont) maximum 80 egységnyi árut szállíthatunk. Azt is látjuk, hogy nem minden viszonylatban korlátozott az elszállítandó áru mennyisége, az 1-es pontból az 5-ös pontba korlátlan mennyiségű áru szállítható. Továbbá az is látható, hogy a korábbi (kapacitáskorlátok nélküli) feladat optimális megoldása most nem lesz lehetséges megoldás, hiszen abban az 1-es pontból a 4-es pontba 120 egységnyi áru szállítása volt az optimális, most viszont maximum 80 egység lehetséges.

Mennyi a minimális szállítási költség, és mi az ehhez kapcsolódó szállítási terv?

Megoldás.

Elég egyszerű a feladat LP felírása: az adott szállítási viszonylatokban a változó értékére korlátot kell szabni. Pl. az 1-es pontból a 4-es pontba maximum 80 egységnyi áru szállítható, amit könnyen elő tudunk írni, az `x1_4<=80` korláttal.

4.2. kód.

```
min
ktg:
+28x1_4 +27x1_5 +28x1_6  +35x1_7
+34x2_4 +28x2_5 +31x2_6  +33x2_7
+16x3_4 +31x3_5 +14x3_6  +19x3_7
+21x4_8 +32x4_9 +33x4_10 +57x4_11 +31x4_12
+59x5_8 +55x5_9 +29x5_10 +39x5_11 +64x5_12
+38x6_8 +54x6_9 +66x6_10 +46x6_11 +49x6_12
+52x7_8 +25x7_9 +64x7_10 +45x7_11 +25x7_12

subject to
r1: +x1_4 +x1_5 +x1_6 +x1_7 = 200
r2: +x2_4 +x2_5 +x2_6 +x2_7 = 200
r3: +x3_4 +x3_5 +x3_6 +x3_7 = 200
```

```
a1: +x4_8  +x5_8  +x6_8  +x7_8  = 120
a2: +x4_9  +x5_9  +x6_9  +x7_9  = 120
a3: +x4_10 +x5_10 +x6_10 +x7_10 = 120
a4: +x4_11 +x5_11 +x6_11 +x7_11 = 120
a5: +x4_12 +x5_12 +x6_12 +x7_12 = 120

t1: +x1_4 +x2_4 +x3_4
      -1x4_8 -x4_9 -x4_10 -x4_11 -x4_12 = 0

t2: +x1_5 +x2_5 +x3_5
      -x5_8 -x5_9 -x5_10 -x5_11 -x5_12 = 0

t3: +x1_6 +x2_6 +x3_6
      -x6_8 -x6_9 -x6_10 -x6_11 -x6_12 = 0

t4: +x1_7 +x2_7 +x3_7
      -x7_8 -x7_9 -x7_10 -x7_11 -x7_12 = 0

kap1_4:  x1_4  <=  80
kap1_6:  x1_6  <=  60
kap2_7:  x2_7  <=  50
kap3_5:  x3_5  <=  90
kap4_8:  x4_8  <= 100
kap5_9:  x5_9  <=  70
kap6_12: x6_12 <=  80

end
```

Az eredmény pedig:

```
Status:     OPTIMAL
Objective:  ktg = 32190 (MINimum)

   No. Column name  St   Activity
------ ------------ -- -------------
     1 x1_4         B             80
     2 x1_5         B             90
     3 x1_6         B             20
     4 x1_7         B             10
     5 x2_4         NL             0
     6 x2_5         B            150
     7 x2_6         NL             0
     8 x2_7         B             50
     9 x3_4         B             20
    10 x3_5         NL             0
    11 x3_6         NL             0
```

```
12 x3_7         B            180
13 x4_8         B            100
14 x4_9         NL             0
15 x4_10        NL             0
16 x4_11        NL             0
17 x4_12        NL             0
18 x5_8         NL             0
19 x5_9         NL             0
20 x5_10        B            120
21 x5_11        B            120
22 x5_12        NL             0
23 x6_8         B             20
24 x6_9         NL             0
25 x6_10        NL             0
26 x6_11        NL             0
27 x6_12        NL             0
28 x7_8         NL             0
29 x7_9         B            120
30 x7_10        NL             0
31 x7_11        NL             0
32 x7_12        B            120
```

Tehát az 1-es pontból szállítunk a 4-es, 5-ös, 6-os és 7-es pontokba is, rendre 80, 90, 20 és 10 egységnyi árut. A 2-es pontból az 5-ös pontba szállítunk 150 egységnyit és a 7-es pontba 50 egységnyit. A 3-as pontból a 4-es pontba szállítunk 20 egységet és a 7-esbe 180-t.

A 4-es pontból csak a 8-as pontba szállítunk 100 egységet, az 5-ösből a 10-es és 11-es pontokba 120-120 egységet, a 6-osból csak a 8-asba 20, a 7-esből pedig a 9-esbe és 12-esbe 120-120 egységet.

Természetesen az összköltség is megnövekedett: a korábbi 31800 helyett most 32190.

A 4.2. kód esetén a kapacitáskorlátok a `subject to` szekcióban szerepelnek. Lehetőségünk van az egy változóra vonatkozó korlátot a `bound` szekcióban is megadni:

4.3. kód.

```
min
ktg:
+28x1_4 +27x1_5 +28x1_6 +35x1_7
+34x2_4 +28x2_5 +31x2_6 +33x2_7
```

```
+16x3_4 +31x3_5 +14x3_6 +19x3_7
+21x4_8 +32x4_9 +33x4_10 +57x4_11 +31x4_12
+59x5_8 +55x5_9 +29x5_10 +39x5_11 +64x5_12
+38x6_8 +54x6_9 +66x6_10 +46x6_11 +49x6_12
+52x7_8 +25x7_9 +64x7_10 +45x7_11 +25x7_12

subject to
r1: +x1_4 +x1_5 +x1_6 +x1_7 = 200
r2: +x2_4 +x2_5 +x2_6 +x2_7 = 200
r3: +x3_4 +x3_5 +x3_6 +x3_7 = 200

a1: +x4_8  +x5_8  +x6_8  +x7_8  = 120
a2: +x4_9  +x5_9  +x6_9  +x7_9  = 120
a3: +x4_10 +x5_10 +x6_10 +x7_10 = 120
a4: +x4_11 +x5_11 +x6_11 +x7_11 = 120
a5: +x4_12 +x5_12 +x6_12 +x7_12 = 120

t1: +x1_4 +x2_4 +x3_4
      -1x4_8 -x4_9 -x4_10 -x4_11 -x4_12 = 0

t2: +x1_5 +x2_5 +x3_5
      -x5_8 -x5_9 -x5_10 -x5_11 -x5_12 = 0

t3: +x1_6 +x2_6 +x3_6
      -x6_8 -x6_9 -x6_10 -x6_11 -x6_12 = 0

t4: +x1_7 +x2_7 +x3_7
      -x7_8 -x7_9 -x7_10 -x7_11 -x7_12 = 0

bounds
x1_4  <=  80
x1_6  <=  60
x2_7  <=  50
x3_5  <=  90
x4_8  <= 100
x5_9  <=  70
x6_12 <=  80

end
```

Természetesen most is ugyanazt a megoldást kapjuk.

4.4. példa. *Vezessünk be további kapacitáskorlátokat is (az eddigiek mellé) a 4.1. példa esetén:*

Honnan	*Hova*	*Kapacitás*
1	*5*	*10*
1	*7*	*30*
4	*9*	*20*
4	*10*	*40*
7	*9*	*40*

Mennyi lesz most a minimális szállítási költség?

Megoldás.

Könnyen ellenőrizhető, hogy ebben az esetben nincs lehetséges megoldás. Elég, ha ehhez megnézzük az 1. raktárból elszállítandó mennyiségeket. A 4, 5, 6 és 7 átrakodási pontokba összesen 200 egységnek kellene megérkeznie, de a kapacitáskorlátok miatt csak 180 egységet tudunk elszállítani. Tehát a raktárakból elszállítandó 600 egység árut ebben az esetben nem tudjuk elszállítani, ehhez túl szűk a hálózat.

4.5. példa. *Maximum mennyi egységet lehet elszállítani a 4.4. példa keretei között?*

Megoldás.

A megoldás könnyen megadható, a célfüggvényt meg kell változtatni: a költségek minimalizálása helyett az elszállítható mennyiségeket maximalizáljuk. Emellett a raktárakra és az áruházakra vonatkozó korlátoknál meg kell engedni, hogy kevesebb mennyiség is kielégíthesse a feltételeket.

4.6. kód.

```
max
z:
+x1_4 +x1_5 +x1_6 +x1_7
+x2_4 +x2_5 +x2_6 +x2_7
+x3_4 +x3_5 +x3_6 +x3_7

subject to
r1: +x1_4 +x1_5 +x1_6 +x1_7 <= 200
r2: +x2_4 +x2_5 +x2_6 +x2_7 <= 200
r3: +x3_4 +x3_5 +x3_6 +x3_7 <= 200

a1: +x4_8  +x5_8  +x6_8  +x7_8  <= 120
a2: +x4_9  +x5_9  +x6_9  +x7_9  <= 120
```

```
a3: +x4_10 +x5_10 +x6_10 +x7_10 <= 120
a4: +x4_11 +x5_11 +x6_11 +x7_11 <= 120
a5: +x4_12 +x5_12 +x6_12 +x7_12 <= 120

t1: +x1_4 +x2_4 +x3_4
      -1x4_8 -x4_9 -x4_10 -x4_11 -x4_12 = 0

t2: +x1_5 +x2_5 +x3_5
      -x5_8 -x5_9 -x5_10 -x5_11 -x5_12 = 0

t3: +x1_6 +x2_6 +x3_6
      -x6_8 -x6_9 -x6_10 -x6_11 -x6_12 = 0

t4: +x1_7 +x2_7 +x3_7
      -x7_8 -x7_9 -x7_10 -x7_11 -x7_12 = 0

kap1_4:  x1_4  <=  80
kap1_6:  x1_6  <=  60
kap2_7:  x2_7  <=  50
kap3_5:  x3_5  <=  90
kap4_8:  x4_8  <= 100
kap5_9:  x5_9  <=  70
kap6_12: x6_12 <=  80

kap1_5:  x1_5  <=  10
kap1_7:  x1_7  <=  30
kap4_9:  x4_9  <=  20
kap4_10: x4_10 <=  40
kap7_9:  x7_9  <=  40

end
```

Az optimális esetben a célfüggvény értéke 580.

Ezzel lényegileg el is érkeztünk a maximális folyam feladathoz: adott egy (jellemzően irányított) gráf, ahol az élekhez kapacitások tartoznak. Általában egy kibocsájtó van, amit **forrásnak** hívunk (a szállítási feladat kontextusában ez egy raktár) és jellemzően egy felhasználó, amit **nyelőnek** hívunk (a szállítási feladat kontextusában áruház). Az a kérdés, hogy mekkora a hálózat kapacitása, amit ebben a kontextusban maximális folyamnak hívunk. Bár elképzelhetünk szállítási feladatot is, mint korábban, de egy csővezeték talán szemléletesebb ebben az esetben. Ekkor az élek értékei a csövek kapacitásait adják meg, a csúcspontokat pedig a csövek közötti csomópontoknak tekinthetjük. Ebben a kontextusban a cél, hogy a lehető legnagyobb mennyiségű folyadékot juttassuk át a forrásból a nyelőbe.

4.7. példa. *Tekintsük az alábbi hálózatot:*

	2	*3*	*4*	*5*	*6*	*7*
1	*13*	*23*		*44*	*16*	*32*
2		*26*		*36*		*11*
3			*37*		*11*	*37*
4	*38*			*34*	*23*	*23*
5		*23*				*11*
6				*11*		*32*

Tehát a hálózat (gráf) 1-es pontjából mutat egy él a 2-es pontba, amelynek a kapacitása 13 egység. A hálózat 1-es pontjából a 4-es pontba nem mutat él (de tekinthetjük úgy is, hogy van, csak 0 a kapacitása). Fontos megjegyzés, hogy irányított élekkel dolgozunk, tehát az 1-es pontból a 2-es pontba áramolhat áru, de visszafelé nem.

Maximum mennyi áru áramolhat az 1-es pontból a 7-es pontba?

Megoldás.

A megoldást az átrakodásos szállítási feladatból nagyrészt le lehet vezetni. Egyetlen lényeges ötlet, hogy bevezetünk egy fiktív élt a nyelőből a forrásba, jelen esetben a 7-es pontból az 1-es pontba (jelöljük ezt `f7_1` módon). Ennek a fiktív élnek nincs kapacitása (vagy másképpen 'végtelen nagy' a kapacitása). Ily módon létre tudunk hozni egy áramot, vagy más néven (kör)folyamot (bizonyos helyeken a cirkuláció kifejezés is használatos). Ennek a fiktív élnek az áteresztő képessége megadja a folyam méretét, ezért ezt maximalizáljuk. Írjuk fel tehát a csomóponti feltételeket a gráf minden pontjára, és adjuk hozzá a kapacitáskorlátokat is!

4.8. kód.

```
max
f7_1

subject to
csp1: +f7_1-x1_2-x1_3-x1_5-x1_6-x1_7         = 0
csp2: +x1_2+x4_2-x2_3-x2_5-x2_7              = 0
```

```
csp3: +x1_3+x2_3+x5_3-x3_4-x3_6-x3_7       = 0
csp4: +x3_4-x4_2-x4_5-x4_6-x4_7            = 0
csp5: +x1_5+x2_5+x4_5+x6_5-x5_3-x5_7       = 0
csp6: +x1_6+x3_6+x4_6-x6_5-x6_7            = 0
csp7: +x1_7+x2_7+x3_7+x4_7+x5_7+x6_7-f7_1 = 0

kap1_2: x1_2<= 13
kap1_3: x1_3<= 23
kap1_5: x1_5<= 44
kap1_6: x1_6<= 16
kap1_7: x1_7<= 32
kap2_3: x2_3<= 26
kap2_5: x2_5<= 36
kap2_7: x2_7<= 11
kap3_4: x3_4<= 37
kap3_6: x3_6<= 11
kap3_7: x3_7<= 37
kap4_2: x4_2<= 38
kap4_5: x4_5<= 34
kap4_6: x4_6<= 23
kap4_7: x4_7<= 23
kap5_3: x5_3<= 23
kap5_7: x5_7<= 11
kap6_5: x6_5<= 11
kap6_7: x6_7<= 32

end
```

A feladat optimális megoldása:

```
Status:     OPTIMAL
Objective:  obj = 118 (MAXimum)

   No.   Row name    St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             0             1
     2 csp2         NS             0         < eps
     3 csp3         NS             0         < eps
     4 csp4         NS             0         < eps
     5 csp5         NS             0             1
     6 csp6         NS             0         < eps
     7 csp7         B              0
     8 kap1_2       NU            13             1
     9 kap1_3       NU            23             1
    10 kap1_5       B             34
```

```
    11 kap1_6       NU            16             1
    12 kap1_7       NU            32             1
    13 kap2_3       B              2
    14 kap2_5       B              0
    15 kap2_7       NU            11         < eps
    16 kap3_4       B             11
    17 kap3_6       B              0
    18 kap3_7       NU            37         < eps
    19 kap4_2       B              0
    20 kap4_5       B              0
    21 kap4_6       B              0
    22 kap4_7       B             11
    23 kap5_3       NU            23             1
    24 kap5_7       NU            11             1
    25 kap6_5       B              0
    26 kap6_7       B             16

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 f7_1         B            118
     2 x1_2         B             13
     3 x1_3         B             23
     4 x1_5         B             34
     5 x1_6         B             16
     6 x1_7         B             32
     7 x4_2         NL             0         < eps
     8 x2_3         B              2
     9 x2_5         NL             0            -1
    10 x2_7         B             11
    11 x5_3         B             23
    12 x3_4         B             11
    13 x3_6         NL             0         < eps
    14 x3_7         B             37
    15 x4_5         NL             0            -1
    16 x4_6         NL             0         < eps
    17 x4_7         B             11
    18 x6_5         NL             0            -1
    19 x5_7         B             11
    20 x6_7         B             16
```

Vegyünk egy adott csomópontot, mondjuk a 2-es pontot. A beérkező árumennyiség 13, ami teljes egészében az egyes pontból érkezik (`x1_2=13`), ez két irányban megy tovább: 2 egység a 3-as csúcs irányába (`x2_3=2`), 11 egység pedig a 7-es csúcs irányába (`x2_7=11`). Hasonlóan számolható a többi csomópont is. Ezáltal a folyam értéke 118 lesz. Ezt két oldalról is ellenőrizhetjük. Egyrészt a kiinduló 1-es pontból ennyi áru áramlik szét: `x1_2=13`, `x1_3=23`, `x1_5=34`, `x1_6=16` és `x1_7=33`, azaz 13+23+34+16+33=118. Másrészt a végső 7-es pontba is ugyanennyi áru érkezik meg: `1_7=32`, `2_7=11`, `x3_7=37`, `x4_7=11`, `x5_7=11` és `x6_7=16`, azaz 32+11+37+11+11+16=118.

Tehát ebben a hálózatban akár az `x1_2+x1_3+x1_5+x1_6`, vagy az `x1_7+x2_7+x3_7+x4_7+x5_7+x6_7` kifejezés is lehetne a célfüggvény. Óvatosságra intünk azonban, mivelhogy más feladatban lehetnek olyan élek, amelyek a kiinduló pontba visszamutatnak, vagy olyanok, amelyek a végső pontból indulnak ki (nem a kezdőpontba mutató fiktív él, amelyet segítségképpen mi vezettünk be). Ilyen esetben a fenti kifejezések nem feltétlenül a maximális folyam értékét adják meg. A fiktív él használatával elkerülhetjük ezeket a nehézségeket.

4.2. A hálózat bővítése

4.9. példa. *Tekintsük a 4.7. példát! Melyik élek kapacitását éri meg bővíteni? Milyen mértékben?*

Megoldás.

Tekintsük a 4.8. kódhoz tartozó outputot. Minden élhez tartozik egy korlát. Először is, amelyik élen nem áramlik annyi áru, mint amennyi az él kapacitása, azt biztos nem érdemes tovább növelni (mivel az eddigi kapacitást sem tudjuk teljesen kihasználni). Ilyen él például az 1-es csúcsból az 5-ös csúcsba mutató él, amelyen keresztül 34 egység áru halad át, de lehetséges lenne akár 44 is. Vannak olyan élek is, amelyeknek kapacitását ugyan teljesen kihasználjuk, ellenben mégsem lenne érdemes azt növelni. Ilyen él például a 2-es csúcsból a 7-es csúcsba mutató él. Ennek az élnek a kapacitása 11, és 11-et is szállítunk, de hiába növeljük a kapacitását, nem lesz nagyobb a folyam értéke (próbáljuk ki!).

Olyan éleket keresünk, ahol a kapacitás növekedésével nő a folyam mérete is. A kapacitás értéke a korlátok jobboldalán szerepel, a folyam mérete pedig a célfüggvény értéke is egyben, tehát a kapacitáskorlátokhoz tartozó árnyékárak pont azt adják meg, hogy a kapacitás (kis mértékű) emelésével hogyan változik a folyam mérete. Csak azon élek kapacitását érdemes növelni, ahol az árnyékár értéke 1 (korlátok `Marginal` oszlopbán szereplő érték). Az árnyékár alapvetően azt mutatja meg, hogyha egységnyivel növeljük a korlát jobb oldali konstansát, akkor mennyivel növekszik a célfüggvény értéke. Tehát jelen modellben, ha növeljük a vezeték kapacitását, akkor mennyivel nő a folyam értéke. A maximális folyam feladat esetében ezeknek a korátoknak az árnyékára kizárólag 1 vagy 0 lehet.

Ilyen él például az 1-es csúcsból a 2-es csúcsba mutató él. Ezen élek a minimális vágás részei, amivel részletesebben a 4.18 példa foglalkozik.

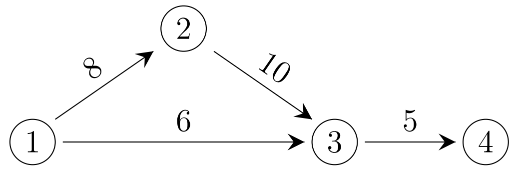
Milyen határok között érdemes változtatni az adott él kapacitását? A kérdés emlékeztet minket az érzékenységvizsgálati határokra. Lássuk hát a megfelelő eredményfület: az `activity range` a `kap1_2` korlát esetén:

```
   No. Row name      St       Activity
                                 range
------ ------------ -- -------------
     8 kap1_2       NU       11.00000
                             25.00000
```

Tehát ha az 1-es csúcsból a 2-es csúcsba mutató él kapacitását megemeljük 25-ig, akkor a bázis változatlan marad. Ha ezen él kapacitását 25-ig fel lehet emelni (ha van rá lehetőség), akkor a folyam mértéke eléri a 130-as értéket.

Kérdés, hogy mi történik, ha ennél tovább emeljük? Ha kipróbáljuk és felemeljük 26-ra az értékét, akkor azt látjuk, hogy a folyam mértéke tovább nő. Nézzünk egy egyszerűbb, jobban átlátható hálózatot, hogy megértsük ezt a jelenséget!

Tekintsünk egy 4 csúcsból álló hálózatot az alábbi ábra szerint.



Tehát az 1-es csúcsból a 2-es csúcsba mutat él, aminek kapacitása 8 egység, a 2-es csúcsból pedig a 3-as csúcsba, 10 egységnyi kapacitással. Az 1-es pontból is mutat él a 3-as csúcsba, amelynek kapacitása 6 egység. Végül a 3-as csúcsból a 4-esbe is mutat egy él, 5 egység kapacitással.

Ehhez a kis hálózathoz tartozó LP felírás:

4.10. kód.

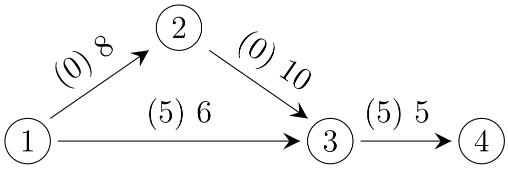
```
max
f4_1

subject to
csp1: +f4_1-x1_2-x1_3       = 0
csp2: +x1_2-x2_3            = 0
csp3: +x1_3+x2_3-x3_4       = 0
csp4: +x3_4-f4_1            = 0

kap1_2: x1_2 <=  8
kap1_3: x1_3 <=  6
kap2_3: x2_3 <= 10
kap3_4: x3_4 <=  5

end
```

A feladat optimális megoldása:

```
Status:     OPTIMAL
Objective:  obj = 5 (MAXimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             0         < eps
     2 csp2         NS             0         < eps
     3 csp3         B              0
     4 csp4         NS             0            -1
     5 kap1_2       B              0
     6 kap1_3       B              5
     7 kap2_3       B              0
     8 kap3_4       NU             5             1

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 f4_1         B              5
     2 x1_2         NL             0         < eps
     3 x1_3         B              5
     4 x2_3         B              0
     5 x3_4         B              5
```

Tehát a folyam mérete 5, az 1-es csúcsból a 3-as csúcsba áramlik 5 egységnyi áru, ami továbbmegy a 4-es végpontba.

Grafikusan szemléltetve (az élek kapacitása előtt zárójelben szerepel a ténylegesen szállított mennyiség).

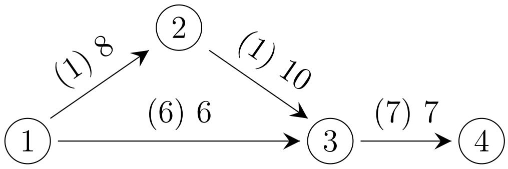


Ekkor csak a `kap3_4` korlát árnyékára 1. Az ehhez tartozó érzékenységvizsgálati határ:

```
   No. Row name      St       Activity
                                 range
------ ------------ -- -------------
     8 kap3_4       NU            .
                               6.00000
```

Tehát a korlát jobboldala 0 és 6 között változhat, a bázis változatlansága nélkül. De mi történik, ha a kapacitást 6 felé emeljük? Mondjuk 7-re:

```
Status:     OPTIMAL
Objective:  obj = 7 (MAXimum)

   No.   Row name    St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             0         < eps
     2 csp2         NS             0         < eps
     3 csp3         B              0
     4 csp4         NS             0            -1
     5 kap1_2       B              1
     6 kap1_3       NU             6         < eps
     7 kap2_3       B              1
     8 kap3_4       NU             7             1

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 f4_1         B              7
     2 x1_2         B              1
     3 x1_3         B              6
     4 x2_3         B              1
     5 x3_4         B              7
```



A folyam mérete tovább nő. Az 1-es csúcsból a 3-as csúcsba mutató él telítődik (ezért megváltozik a bázis). A 2-es csúcson keresztül tudunk további egy egységet szállítani, de a szűk kapacitás továbbra is a 3-as csúcsból a 4-es csúcsba mutató él, amelynek az árnyékára 1. Az `activity range` ebben az esetben a [6;14] intervallum, ami érthető is: az 1-es csúcsból az alsó ágon el tudunk szállítani 6 egységet, a felső ágon 8-at, ami összesen 14 egység. Ha ennél tovább növeljük a 3-as csúcsból a 4-es csúcsba mutató él kapacitását, azt már nem fogjuk tudni kihasználni.

Térjünk vissza a 4.7. példa adataihoz! Első lépésként csak annyit látunk, hogy az 1-es csúcsból a 2-es csúcsba mutató él kapacitását 25-ig fel lehet emelni. Menjünk tovább egy kicsit ezen a határon, emeljük fel a kapacitás mértékét 25,1-re. Ekkor a `kap1_2` korlát árnyékára továbbra is 1 marad, azaz új bázisra tért át az algoritmus. Ehhez a bázishoz tartozó `activity range` a [25;36] intervallum. Nézzük tovább! Emeljük a kapacitás értékét egy kicsit tovább, 36,1-re! Ekkor a `kap1_2` korlát árnyékára továbbra is 1 marad, ehhez a bázishoz a [36;37] intervallum tartozik. Ha ennél is tovább növeljük a kapacitást, akkor az új bázisban a `kap1_2` korlát árnyékára már nem 1 lesz, tehát 37-ig érdemes emelni az 1-es csúcsból a 2-es csúcsban mutató korlát kapacitását.

Az érzékenységvizsgálati határoknak az lenne az igazi értelme, hogy újabb futtatás nélkül meg tudunk bizonyos kérdéseket válaszolni. Láthatjuk azonban, hogy egy él kapacitásának lehetséges bővítését nem feltétlenül tudjuk meg az érzékenységvizsgálati táblázatból. Felmerülhet bennünk, hogy esetleg egyetlen (újabb) futtatással meg tudjuk-e válaszolni a kérdést?

A válasz igenlő. Két dolog szükségeltetik hozzá: egyrészt vezessünk be egy `b1_2` változót, ami azt mutatja, hogy mennyivel bővítjük az 1-esből a 2-es pontba mutató él kapacitását. Eredetileg az él kapacitása 13, amit bővítünk `b1_2` egységgel, tehát `x1_2<=13+b1_2`, amit a `x1_2-b1_2<=13` formára kell átalakítani. Hogy tudjuk elérni, hogy ne bővítsük a kelleténél tovább az él kapacitását? Módosítsuk a célfüggvényt az `f7_1-0.1b1_2` értékre. Ha az él kapacitásának növekedésével a folyam mérete is növekszik, akkor a célfüggvény is nőni fog. Pl.: ha 1 egységgel növeljük a kapacitást, aminek hatására 1 egységgel növekszik a folyam értéke is, akkor a célfüggvény 1-0,1=0,9-cel fog nőni. Ha az él kapacitásának a növekedése nem jár együtt a folyam méretének növekedésével, akkor a célfüggvény 0,1-gyel csökken.

Megjegyezzük, hogy a 0,1-es együttható a `b1_2` változó esetén bármely tetszőlegesen kiválasztott érték lehet, amely szigorúan nagyobb, mint 0 és szigorúan kisebb, mint 1.

4.11. kód.

```
max
f7_1-0.1b1_2

subject to
csp1: +f7_1-x1_2-x1_3-x1_5-x1_6-x1_7      = 0
csp2: +x1_2+x4_2-x2_3-x2_5-x2_7           = 0
csp3: +x1_3+x2_3+x5_3-x3_4-x3_6-x3_7      = 0
csp4: +x3_4-x4_2-x4_5-x4_6-x4_7           = 0
csp5: +x1_5+x2_5+x4_5+x6_5-x5_3-x5_7      = 0
csp6: +x1_6+x3_6+x4_6-x6_5-x6_7           = 0
csp7: +x1_7+x2_7+x3_7+x4_7+x5_7+x6_7-f7_1 = 0

kap1_2: x1_2-b1_2<= 13
kap1_3: x1_3<= 23
kap1_5: x1_5<= 44
kap1_6: x1_6<= 16
kap1_7: x1_7<= 32
kap2_3: x2_3<= 26
kap2_5: x2_5<= 36
kap2_7: x2_7<= 11
kap3_4: x3_4<= 37
kap3_6: x3_6<= 11
kap3_7: x3_7<= 37
kap4_2: x4_2<= 38
kap4_5: x4_5<= 34
kap4_6: x4_6<= 23
kap4_7: x4_7<= 23
kap5_3: x5_3<= 23
kap5_7: x5_7<= 11
kap6_5: x6_5<= 11
kap6_7: x6_7<= 32
```

```
end
```

Az eredményfülről számunkra most csak a `b1_2` változó értéke releváns, ami 24. Tehát maximum 24 egységgel érdemes növelni a kapacitást, azaz 37 (13+24) egységig. Így ugyanazt kaptuk, mint korábban az iterációkkal.

Idáig az 1-es csúcsból a 2-es csúcsba vezető élen a kapacitás bővítésének lehetőségét vizsgáltuk. Hasonló módon megvizsgálhatjuk a többi élt is, ahol az árnyékár 1 (ilyenek az 1-ből 3-ba; 1-ből 6-ba; 1-ből 7-be; 5-ből 3-ba illetve az 5-ből 7-be mutató élek). Természetesen mindig csak egy helyen bővítjük a hálózatot, és futtatjuk a 4.11. kódot, átírva a vizsgált viszonylatra. A következő példa szól arról, ha egyszerre több helyen is bővítjük a hálózatot.

4.12. példa. *Tekintsük megint a 4.7. példát! Tudjuk, hogy ebben az esetben a maximális folyam mérete 118. A következő évben a megnövekedett keresletnek megfelelőn biztosítani kellene, hogy legalább 200 egységnyi árut el tudjunk szállítani az 1-es pontból a 7-es pontba. Ehhez nyilvánvalóan bővíteni kell a hálózatot, a bővítés viszont költségekkel jár. Az alábbi táblázat mutatja, hogy adott él kapacitásának egységnyi bővítése mekkora költséggel jár.*

Él	*Bővítési egységköltség*
1→2	*30*
1→5	*18*
2→7	*51*
4→7	*37*
5→3	*29*
6→7	*13*

Hogyan tudjuk a hálózat áteresztő kapacitását a legolcsóbban bővíteni 200 egységre?

Megoldás.

Vezessünk be most is b (bővítési) változókat azokhoz az élekhez, amelyek kapacitását tudjuk bővíteni. Tehát pl. `b1_2` változó azt jelenti, hogy az 1-es pontból a 2-es pontba mutató él kapacitását mennyivel bővítjük. Vezessünk be egy korlátot, a fiktív élen szállított mennyiségre (legalább 200), és a célfüggvényben az ehhez szükséges bővítési költséget szeretnénk minimalizálni.

4.13. kód.

```
min
+30b1_2 +29b5_3 +18b1_5 +51b2_7 +37b4_7 +13b6_7

subject to

csp1: +f7_1 -x1_2 -x1_3 -x1_5 -x1_6 -x1_7       = 0
csp2: +x1_2 +x4_2 -x2_3 -x2_5 -x2_7             = 0
csp3: +x1_3 +x2_3 +x5_3 -x3_4 -x3_6 -x3_7       = 0
csp4: +x3_4 -x4_2 -x4_5 -x4_6 -x4_7             = 0
csp5: +x1_5 +x2_5 +x4_5 +x6_5 -x5_3 -x5_7       = 0
csp6: +x1_6 +x3_6 +x4_6 -x6_5 -x6_7             = 0
csp7: +x1_7 +x2_7 +x3_7 +x4_7 +x5_7 +x6_7 -f7_1 = 0

mfolyam: f7_1       >= 200
kap1_2:  x1_2 -b1_2 <=13
kap1_3:  x1_3       <=23
kap1_5:  x1_5 -b1_5 <=44
kap1_6:  x1_6       <=16
kap1_7:  x1_7       <=32
kap2_3:  x2_3       <=26
kap2_5:  x2_5       <=36
kap2_7:  x2_7 -b2_7 <=11
kap3_4:  x3_4       <=37
kap3_6:  x3_6       <=11
kap3_7:  x3_7       <=37
kap4_2:  x4_2       <=38
kap4_5:  x4_5       <=34
kap4_6:  x4_6       <=23
kap4_7:  x4_7 -b4_7 <=23
kap5_3:  x5_3 -b5_3 <=23
kap5_7:  x5_7       <=11
kap6_5:  x6_5       <=11
kap6_7:  x6_7 -b6_7 <=32

end
```

Az optimális megoldás:

```
Status:     OPTIMAL
Objective:  obj = 4913 (MINimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             0           -81
     2 csp2         NS             0           -51
```

3	csp3	NS	0	-34
4	csp4	NS	0	-13
5	csp5	NS	0	-63
6	csp6	NS	0	-13
7	csp7	B	0	
8	mfolyam	NL	200	81
9	kap1_2	NU	13	-30
10	kap1_3	NU	23	-47
11	kap1_5	NU	44	-18
12	kap1_6	NU	16	-68
13	kap1_7	NU	32	-81
14	kap2_3	NU	26	-17
15	kap2_5	B	0	
16	kap2_7	NU	11	-51
17	kap3_4	NU	37	-21
18	kap3_6	NU	11	-21
19	kap3_7	NU	37	-34
20	kap4_2	B	0	
21	kap4_5	B	0	
22	kap4_6	B	14	
23	kap4_7	NU	23	-13
24	kap5_3	NU	23	-29
25	kap5_7	NU	11	-63
26	kap6_5	B	0	
27	kap6_7	NU	32	-13

No.	Column name	St	Activity	Marginal
1	b1_2	B	69	
2	b5_3	B	13	
3	b1_5	B	3	
4	b2_7	B	45	
5	b4_7	NL	0	24
6	b6_7	B	9	
7	f7_1	B	200	
8	x1_2	B	82	
9	x1_3	B	23	
10	x1_5	B	47	
11	x1_6	B	16	
12	x1_7	B	32	
13	x4_2	NL	0	38
14	x2_3	B	26	
15	x2_5	NL	0	12
16	x2_7	B	56	
17	x5_3	B	36	

```
18 x3_4          B             37
19 x3_6          B             11
20 x3_7          B             37
21 x4_5          NL             0             50
22 x4_6          B             14
23 x4_7          B             23
24 x6_5          NL             0             50
25 x5_7          B             11
26 x6_7          B             41
```

Tehát az 1-es pontból a 2-es pontba mutató él kapacitását növeljük meg 69 egységgel, az 5-ös csúcsból a 3-asba menő él kapacitását 13 egységgel, az 1-es pontból az 5-ös pontba mutató él kapacitását 3 egységgel, a 2-es pontból a 7-es pontba vezető él kapacitását 45-el, valamint a 6-os pontból a 7-es pontba mutató él kapacitását 9 egységgel, így a bővítési költség összesen 4913 lesz.

4.3. Gyakorló feladatok

4.3.1. Egyszerűbb feladatok

4.14. példa. *A következő táblázatban a hálózat csúcspontjai közötti irányított élek árukra vonatkozó kapacitásai láthatóak:*

	2	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
1	*72*	*30*	*80*						
2		*90*	*74*			*70*			
3			*88*			*33*		*50*	
4				*42*			*73*		
5		*12*			*120*				
6	*30*					*25*	*75*		
7				*30*					*90*
8									*100*
9						*23*	*84*		

Maximálisan mennyi áru áramolhat az első csúcsból a 10. csúcsba?

Megoldás a 258. oldalon.

4.15. példa. *Tekintsük a következő táblázatot.*

	2	3	4	5	6	7	8	9	10	11	12
1	45	42	38								
2		92		71	66						
3			48		30						
4						66	88				15
5								58	67		
6						41			94		
7				77				58			
8								32		96	
9											32
10										100	
11											60

Mekkora a maximálisan átfolyatható mennyiség, az 1. és a 12. csúcs között?

Megoldás a 261. oldalon.

4.16. példa. *Tekintsük a 4.15. kódban meghatározott hálózatot. Ezen hálózat meglévő éleinek a kapacitását most bővíthetjük az alábbi költségek mellett:*

- *Ha kiinduló és a beérkező csúcs is páros, akkor a bővítési költség egységenként 10.*
- *Ha mind a kettő páratlan, akkor viszont csak 8.*
- *Ha az él egy páros csúcsból páratlanba érkezik, akkor 16 lesz az egy egységre jutó bővítési költség, míg fordított esetben 20.*

Mekkora lenne a bővítési költség, ha azt szeretnénk, hogy a 12. csúcsba összesen 200 egység érkezzen meg?

4.3.2. Nehezebb feladatok

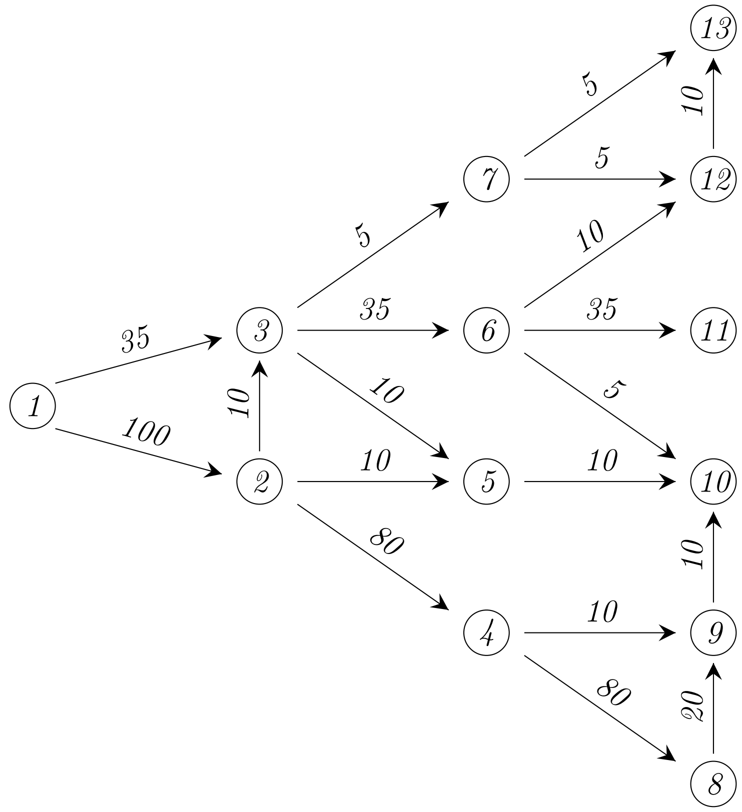
4.17. példa. *Tekintsük a 4.15 példa hálózatát. Hogyan módosul a megoldás, amennyiben az éleken akár visszafele is lehet a folyadékot folyatni? Természetesen egyszerre csak egy irányba lehet a csövön folyatni.*

Megoldás a 267. oldalon.

4.18. példa. *Tekintsük a 4.7. példát. A 4.8. kód segítségével megkaphatjuk az optimális megoldást is. Maximális folyam esetén az LP feladat alternatívája a Ford-Fulkerson algoritmus. Az optimalitás kritériuma az ún. minimális vágás. Az LP feladat eredményfüléről leolvasható-e a minimális vágás?*

Megoldás a 278. oldalon.

4.19. példa. *A következő gráf egy város bizonyos részének stilizált úthálózatát mutatja. Egy vállalat, amelynek központi irodája az 1 csúcsban található, szolgálati lakásokat akar bérelni a dolgozóinak a 8-13 csúcsokkal jelölt lakóparkokban. A gráf élein jelölt városi úthálózat kapacitásai alapján dönt a bérleményekről, mivel szeretné, hogy a dolgozói ne töltsenek sok időt az irodába utazással.*



A kapacitások azt jelzik, hogy hány autót képes elakadásmentesen átereszteni az adott útszakasz. A vállalat úgy tervezi, hogy minden átengedhető autó után bérel egy lakást. Ezen a szabály alapján mennyi lakást bérelhet maximum a vállalat, a 8-13 lakóparkokban?

Változtassunk annyit a feladaton, hogy a vállalat összesen 100 lakást akar bérelni, és szeretné az összes bérleti díjat minimalizálni! A következő táblázatban látható az adott lakóparkok egy-egy lakásának bérleti díjai:

	bérleti díj
13	*15*
12	*18*
11	*20*
10	*12*
9	*18*
8	*25*

Mekkora összköltséggel bérli a lakásokat, és melyik lakóparkokban?

Megoldás a 282. oldalon.

4.20. példa. *Adott egy csővezeték hálózata:*

	2	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
1	$\underline{30}$	*50*	***40***						
2		$\underline{20}$		$\underline{52}$					
3				$\underline{21}$	*60*			*10*	
4		***10***			***13***	***25***			
5					$\underline{10}$		$\underline{36}$		$\underline{10}$
6							*30*	*30*	
7					***10***			***15***	***5***
8									*40*
9									*40*

A hálózaton keresztül A és B típusú olajat szeretnénk eljuttatni az első csúcsból a 10. csúcsba. A táblázatban jelölt számok az elosztó-állomások közötti csövek kapacitásait mutatják. Az $\underline{\textit{aláhúzott}}$ *éleken kizárólag A típusú, míg a* ***vastagított*** *számokkal jelölt éleken kizárólag B típusú olaj szállítható. A jelöletlen élek mind a két típusú olaj szállítására alkalmasak, viszont a két olaj nem keveredhet össze, ezért ezeken az éleken is egyszerre csak az egyik szállítható. Mennyi olajat szállíthatunk A, és B típusból a 10. csúcsba, feltéve, hogy azt szeretnénk, hogy mind a kettő típusból legalább 20 egység beérkezzen, de úgy, hogy A-ból legalább annyi legyen, mint B-ből?*

Megoldás a 288. oldalon.

4.21. példa. *A következő 5 napra egy kórház 5 ügyeletes orvosának (*`A,B,C,D,E`*) beosztását akarjuk megszervezni. Egy nap három ügyeleti időszak van, egy reggeli, egy délutáni és egy esti. Mindegyik orvosnak 3 ügyeletet kell vállalnia a héten, mégpedig úgy, hogy minden egyes napszakban legalább legyen egy ügyelete. Kórházi szabály, hogy egyik orvos sem vállalhat egy napon két egymást követő ügyeletet, és a reggeli- és a délutáni ügyelet napja között legfeljebb 2 nap lehet, viszont a délutáni és esti műszakok nem lehetnek egymást követő napokon.*

Az orvosok egyedül a reggeli időpontokról mondhatják meg, hogy mikor nem alkalmas nekik:

`A` *nem ér rá az első és harmadik reggeli ügyelet alatt,* `B`*, és* `C` *orvosoknak a 4. délelőtt nem alkalmas. A* `D` *orvosnak a 3.-5. napok reggelei,* `E`*-nek pedig az 1.-3. reggelek nem alkalmasak ügyelet tartására.*

Használjon maximális folyam feladatot, egy lehetséges beosztás meghatározásához!

Megoldás a 293. oldalon.

5. fejezet

Legrövidebb út feladat

5.1. Csomópontos formalizáció

Az összetett szállítási feladat egy általánosításának tekinthető a legrövidebb út feladat is. Adott egy gráf, és kiválasztjuk a gráf két tetszőleges pontját és az a célunk, hogy ezen két pont között a legrövidebb utat megtaláljuk. Szállítási feladat kontextusában: van egy egységnyi áru a kiinduló pontban és ezt szeretnénk elszállítani a végpontba. A szállítási költségek ebben az esetben nem mások, mint az útvonalak hosszai. Érdemes emellett a feladat duálját is alaposabban megvizsgálni, mert véleményünk szerint segít megérteni a primál-duál feladatpárok közötti összegüggést.

Ha nem LP feladatként tekintünk a legrövidebb út feladatra, akkor a klasszikus feladatot megoldhatjuk a Dijkstra algoritmussal is. A Dijkstra algoritmus implementációja sokkal gyorsabb kódot eredményez, mint az LP felírás, de az előbbi esetben csak a klasszikus feladatot tudjuk megoldani, az utóbbiban viszont lehetséges egyedi korlátok hozzáadása a feladathoz.

5.1. példa. *Adott egy gráf 7 ponttal, a köztük lévő távolságot az alábbi mátrix* *mutatja:*

	1	2	3	4	5	6	7
1		$22,9$	$59,3$	$56,3$	$96,8$	$15,6$	$100,1$
2			$67,4$	$24,7$	$44,5$	$38,5$	$97,6$
3				$104,7$	$49,7$	$46,8$	$103,2$
4					$53,9$	$88,0$	$105,6$
5						$17,6$	$15,2$
6							$130,6$
7							

Szeretnénk a legrövidebb utat megkeresni az 1-es csúcspontból a 7-es csúcspontba. A megoldást most határozzuk meg úgy, mintha egy átrakodásos szállítási feladatot oldanánk meg!

Megoldás.

Mielőtt megadnánk a feladat LP felírását, érdemes kihangsúlyozni, hogy a legrövidebb út feladat esetében általában irányított gráffal dolgozunk. Tehát jelen példában az 1-es pontból el tudunk jutni a 2-es csúcsba, de a 2-esből nem tudunk eljutni az 1-es csúcsba, mert második sor első eleme nincs megadva.

Azonban nem lenne nehéz irányítatlan gráffal dolgozni, hiszen ekkor a költségek mátrixát egyszerűen tükrözni kellene a diagonálisra, tehát a második sor első cellájába be kellene írni a 22,9-es értéket, a harmadik sor első cellájába a 59,3-as értéket, és így tovább. Irányítatlan gráf esetét a 5.5. példában vizsgáljuk majd meg.

De olyan is előfordulhat, hogy két csúcs között mind a két irányba lehet menni, de eltérő költséggel, például hegynek felfelé vagy hegyről lefelé nem ugyanannyi a benzinköltség. Ekkor célszerű irányított gráfot használni, ahol a két csúcs között kettő él megy, ellentétes irányokban.

A jelen példában tehát az 1-es pontban van egységnyi áru, amit el kell szállítanunk a 7-es pontba. A 2-es, 3-as, ..., 6-os csúcspontok lehetséges átrakodási pontok. Ezekre a pontokra fel kell írni a csomóponti feltételeket: a bejövő árumennyiségnek meg kell egyeznie a kimenő árumennyiséggel. Jelölje x_{ij} az i csúcsból a j csúcsba szállított árumennyiséget.

5.2. kód.

```
min
ktg:
+22.9x1_2+59.3x1_3+ 56.3x1_4+ 96.8x1_5+ 15.6x1_6+100.1x1_7
         +67.4x2_3+ 24.7x2_4+ 44.5x2_5+ 38.5x2_6+ 97.6x2_7
                  +104.7x3_4+ 49.7x3_5+ 46.8x3_6+103.2x3_7
                            + 53.9x4_5+ 88.0x4_6+105.6x4_7
                                      + 17.6x5_6+ 15.2x5_7
                                                +130.6x6_7
subject to
csp1:        +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_7 = 1
csp2: -x1_2        +x2_3 +x2_4 +x2_5 +x2_6 +x2_7 = 0
csp3: -x1_3 -x2_3        +x3_4 +x3_5 +x3_6 +x3_7 = 0
csp4: -x1_4 -x2_4 -x3_4        +x4_5 +x4_6 +x4_7 = 0
csp5: -x1_5 -x2_5 -x3_5 -x4_5        +x5_6 +x5_7 = 0
csp6: -x1_6 -x2_6 -x3_6 -x4_6 -x5_6        +x6_7 = 0
csp7: -x1_7 -x2_7 -x3_7 -x4_7 -x5_7 -x6_7        =-1

end
```

Értelmezzük egy kicsit részletesebben a 5.2. kódot! A célfüggvényben a költségminimumot keressük, és a feltételek a csúcspontokra vannak felírva. Az 1-es csúcspontban van egységnyi áru, amit el kell szállítani (`csp1` korlát). A 2-es csúcspontra felírjuk a szokásos csomóponti feltételt: a bejövő árumennyiség egyezzen meg a kimenő árumennyiséggel. A 2-es csúcsba csak az 1-es csúcspontból tudunk mennyi. A 2-es csúcsból viszont 3-as, 4-es, 5-ös, 6-os és 7-es csúcspontokba is mehetünk. Tehát ennek a két mennyiségnek meg kell egyezni: `x1_2=x2_3+x2_4+x2_5+x2_6+x2_7`. Átrendezés után kapjuk a `csp2` korlátot. Hasonló korlátokat fel kell írni a többi csúcspontra is, az utolsó csúcspont kivételével. Erre a csúcspontra vonatkozó korlát (`csp7`) azt mondja, hogy 1 egységnyi árunak meg kell érkeznie a 7-es csúcsba is. Ezt írhatnánk a szokásos `x1_7+x2_7+x3_7+x4_7+x5_7+x6_7=1` formában is, de ha ennek ellentettjét szerepeltetjük, akkor szebb struktúrája lesz az egyenletrendszernek. Megjegyezzük, hogy ez a korlátot akár el is lehetne hagyni, hiszen a többi folyammegmaradási korlátból következik.

Az optimális megoldás:

```
Status:     OPTIMAL
```

```
Objective:  ktg = 82.6 (MINimum)

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 x1_2         B              1
     2 x1_3         NL             0          41.6
     3 x1_4         NL             0          42.8
     4 x1_5         NL             0          29.4
     5 x1_6         NL             0          63.6
     6 x1_7         NL             0          17.5
     7 x2_3         NL             0          72.6
     8 x2_4         NL             0          34.1
     9 x2_5         B              1
    10 x2_6         NL             0         109.4
    11 x2_7         NL             0          37.9
    12 x3_4         NL             0         108.9
    13 x3_5         B              0
    14 x3_6         NL             0         112.5
    15 x3_7         NL             0          38.3
    16 x4_5         B              0
    17 x4_6         NL             0         149.5
    18 x4_7         NL             0          36.5
    19 x5_6         NL             0           133
    20 x5_7         B              1
    21 x6_7         B              0
```

Az eredményfülről leolvasható, hogy a legrövidebb út az $1 \to 2 \to 5 \to 7$.

Látható, hogy —akárcsak korábban— most sem szükségesek az egészértékűségi kikötések, mivel enélkül is egészértékű megoldást kapunk. Ez egyben azt is jelenti, hogy rendelkezésünkre állnak az árnyékárak, a redukált költségek, és az érzékenységvizsgálati határok is. Ezek a mennyiségek abban nyújtanak segítséget, hogy meg tudjuk határozni, hogy a gráf élei milyen mértékben változhatnak, hogy a legrövidebb út változatlan maradjon, vagy éppen megváltozzon.

5.3. példa. *Az 5.1. példa esetén mennyivel lehet megváltoztatni az 1-es pontból a 3-as pontba mutató él hosszát úgy, hogy a legrövidebb út változatlan maradjon? Mennyivel kell megváltoztatni a 3-as pontból a 4-es pontba mutató él hosszát, hogy a legrövidebb út része legyen?*

Megoldás.

Az 1-es csúcsból a 3-as csúcsba mutató él hossza 59,3, és az ehhez tartozó redukált költség 41,6 az output szerint. A redukált költség azt jelenti, hogy ha a célfüggvény együttható ennél jobban csökken, akkor az élhez tartozó `x1_3` változó a bázis része lesz. Most ‘szerencsés’ helyzet áll elő: 59,3-41,6=17,7. Ha az él hosszát ennél valamivel kisebb értékre állítjuk —mondjuk 17,69-re— akkor az 1-es pontból a 3-as pontba mutató él a legrövidebb út része lesz:

No.	Column name	St	Activity	Marginal
1	x1_2	NL	0	0.01
2	x1_3	B	1	
3	x1_4	NL	0	42.81
4	x1_5	NL	0	29.41
5	x1_6	NL	0	63.61
6	x1_7	NL	0	17.51
7	x2_3	NL	0	72.6
8	x2_4	NL	0	34.1
9	x2_5	B	0	
10	x2_6	NL	0	109.4
11	x2_7	NL	0	37.9
12	x3_4	NL	0	108.9
13	x3_5	B	1	
14	x3_6	NL	0	112.5
15	x3_7	NL	0	38.3
16	x4_5	B	0	
17	x4_6	NL	0	149.5
18	x4_7	NL	0	36.5
19	x5_6	NL	0	133
20	x5_7	B	1	
21	x6_7	B	0	

Viszont, ha 17,7-re állítjuk akkor a 1-es csúcs és a 3-as csúcs közötti él még nem lesz a legrövidebb út része:

No.	Column name	St	Activity	Marginal
1	x1_2	B	1	
2	x1_3	NL	0	< eps
3	x1_4	NL	0	42.8
4	x1_5	NL	0	29.4
5	x1_6	NL	0	63.6
6	x1_7	NL	0	17.5
7	x2_3	NL	0	72.6

8	x2_4	NL	0	34.1
9	x2_5	B	1	
10	x2_6	NL	0	109.4
11	x2_7	NL	0	37.9
12	x3_4	NL	0	108.9
13	x3_5	B	0	
14	x3_6	NL	0	112.5
15	x3_7	NL	0	38.3
16	x4_5	B	0	
17	x4_6	NL	0	149.5
18	x4_7	NL	0	36.5
19	x5_6	NL	0	133
20	x5_7	B	1	
21	x6_7	B	0	

Nézzük az x3_4 változót. Az eredeti adatok esetén a redukált költség 108,9. Vonjuk le a költségelemből a redukált költséget: 104,7-108,9=-4,2. Cseréljük le a x3_4 változó célfüggvény együtthatóját erre az értékre, egészen pontosan egy kicsit kisebbre (-4,21), hogy elkerüljük az alternatív megoldást (tényleges utak esetén furcsának hat a negatív érték, de bizonyos —pl. gazdasági— feladatok esetén elképzelhető a negatív érték —veszteség— is egy út hosszára; most elfogadjuk ezt az absztrakciót, hogy egy él hossza lehet negatív szám is). Ekkor az optimális megoldás:

No.	Column name	St	Activity	Marginal
1	x1_2	B	1	
2	x1_3	NL	0	41.59
3	x1_4	NL	0	42.8
4	x1_5	NL	0	29.4
5	x1_6	NL	0	63.6
6	x1_7	NL	0	17.5
7	x2_3	NL	0	72.59
8	x2_4	NL	0	34.1
9	x2_5	B	1	
10	x2_6	NL	0	109.4
11	x2_7	NL	0	37.9
12	x3_4	B	0	
13	x3_5	NL	0	0.01
14	x3_6	NL	0	112.51
15	x3_7	NL	0	38.31
16	x4_5	B	0	
17	x4_6	NL	0	149.5
18	x4_7	NL	0	36.5

19	x5_6	NL	0	133
20	x5_7	B	1	
21	x6_7	B	0	

Az x3_4 változó bázisváltozó lett, de az értéke 0, tehát nem került be a legrövidebb útba. Mivel a feladat degenerált, ezért a bázisba kerülés nem jelenti feltétlenül azt is, hogy az adott él a legrövidebb út része lesz. Ahhoz, hogy ez megtörténjen, tovább kell csökkenteni az célfüggvény együtthatót. De van jobb módszer annál, hogy egyszerűen próbálkozunk: nézzük meg ebben a módosított feladatban az érzékenységvizsgálati határokat!

No.	Column name	St	Obj coef range
12	x3_4	BS	-45.80000
			-4.20000

Az x3_4 változó célfüggvény együtthatója -45,8 és -4,2 között van, akkor a bázis nem változik. Állítsuk tehát a célfüggvény együtthatót egy picivel a -45,8-as érték alá (-45,81)! Ekkor az optimális megoldás:

No.	Column name	St	Activity	Marginal
1	x1_2	NL	0	0.01
2	x1_3	B	1	
3	x1_4	NL	0	42.81
4	x1_5	NL	0	29.41
5	x1_6	NL	0	63.61
6	x1_7	NL	0	17.51
7	x2_3	NL	0	30.99
8	x2_4	NL	0	34.1
9	x2_5	B	0	
10	x2_6	NL	0	109.4
11	x2_7	NL	0	37.9
12	x3_4	B	1	
13	x3_5	NL	0	41.61
14	x3_6	NL	0	154.11
15	x3_7	NL	0	79.91
16	x4_5	B	1	
17	x4_6	NL	0	149.5
18	x4_7	NL	0	36.5
19	x5_6	NL	0	133

```
   20 x5_7          B               1
   21 x6_7          B               0
```

Látható, hogy most már a legrövidebb út része a 3-as csúcsból a 4-es csúcsba mutató él.

5.4. példa. *Tekintsük megint az 5.1. példát. Milyen határok között változhatnak a legrövidebb utat alkotó élek hosszai, hogy ezek az élek továbbra is a legrövidebb út részei legyenek?*

Megoldás.

Ez a kérdés nagyon emlékeztet minket az érzékenységvizsgálati határokra, amelyek megadják, hogy a célfüggvény együttható (az él hossza jelen esetben) milyen határok között változhat úgy, hogy ne változzon a bázis. Ebben az esetben is igaz, hogy a bázist nem csak a legrövidebb utat alkotó élek alkotják, ezért a bázis megváltozhatna anélkül is, hogy a legrövidebb utat alkotó élek halmaza megváltozik. De most nem ez a helyzet: ha kilépünk a legrövidebb utak esetén az érzékenységvizsgálati határok alkotta intervallumból, akkor változik a legrövidebb út is.

Az legrövidebb utat alkotó éleket reprezentáló változók esetén az érzékenységvizsgálati határok:

```
   No. Column name  St       Obj coef
                                range
------ ------------ -- -------------
     1 x1_2         BS           -Inf
                             40.40000
     9 x2_5         BS           -Inf
                             62.00000
    20 x5_7         BS           -Inf
                             32.70000
```

Az érzékenységvizsgálati határok alsó értéke `-Inf` minden esetben. Ezen nagyon nem is lepődhetünk meg, mivel már így is a legrövidebb utat alkotják, ha csökkennek, még rövidebbé válik ez az út. Tehát számunkra csak a felső határ érdekes. Az 1-es csúcsból a 2-es csúcsba mutató él esetén ez az érték 40,4. Állítsuk be a célfüggvény együtthatót (az út hosszát) 40,41 értékre, és nézzük meg, mi történik! Az optimális megoldás ebben az esetben:

```
   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 x1_2         NL             0          0.01
     2 x1_3         NL             0          24.1
     3 x1_4         NL             0          25.3
     4 x1_5         NL             0          11.9
     5 x1_6         NL             0          46.1
     6 x1_7         B              1
     7 x2_3         NL             0          72.6
     8 x2_4         NL             0          34.1
     9 x2_5         B              0
    10 x2_6         NL             0         109.4
    11 x2_7         NL             0          37.9
    12 x3_4         NL             0         108.9
    13 x3_5         B              0
    14 x3_6         NL             0         112.5
    15 x3_7         NL             0          38.3
    16 x4_5         B              0
    17 x4_6         NL             0         149.5
    18 x4_7         NL             0          36.5
    19 x5_6         NL             0           133
    20 x5_7         B              0
    21 x6_7         B              0
```

Ekkor a várakozásunknak megfelelően `x1_2` változó már nem lesz bázisváltozó, azaz az 1-es csúcsból a 2-es csúcsba mutató él már nem lesz a legrövidebb út része, az a jelen esetben 1→7-re változik. Hasonlóan kell vizsgálni a legrövidebb út másik két élét is.

5.5. példa. *Térjünk vissza az eredeti feladatra (5.1. példa). Hogyan változik a legrövidebb út, ha irányítatlan gráffal dolgozunk?*

Megoldás.

A helyzetet viszonylag egyszerűen tudjuk kezelni, a költségmátrixban használjuk az alsó háromszög mátrixot is. Ebben az esetben a egyszerűen tükrözzük a diagonálisra a költségelemeket, azaz a 2-es csúcsból a 3-as csúcsba eljutni ugyanannyi ideig tart, mint fordítva.

Ezt a kibővítést a feladat jellegétől függően lehet elvégezni, ugyanis ha ez egy lejtős utca lenne, akkor lejtőn felfelé menni hosszabb ideig tartana (főleg gyalog), mint lefelé. Ebben az esetben a költségmátrix nem lenne szimmetrikus a diagonálisra, tehát a 2. sor 3. cellájában például nagyobb érték szerepelne, mint a 3. sor 2. cellájában. De mint mondtuk, a mi esetünkben nem ez a helyzet:

	1	2	3	4	5	6	7
1		22,9	59,3	56,3	96,8	15,6	100,1
2	22,9		67,4	24,7	44,5	38,5	97,6
3	59,3	67,4		104,7	49,7	46,8	103,2
4	56,3	24,7	104,7		53,9	88	105,6
5	96,8	44,5	49,7	53,9		17,6	15,2
6	15,6	38,5	46,8	88	17,6		130,6
7	100,1	97,6	103,2	105,6	15,2	130,6	

Ebben az esetben az LP felírás:

5.6. kód.

```
min
ktg:
+ 22.9x1_2 +59.3x1_3 + 56.3x1_4 + 96.8x1_5 +15.6x1_6 +100.1x1_7
+ 22.9x2_1 +67.4x2_3 + 24.7x2_4 + 44.5x2_5 +38.5x2_6 + 97.6x2_7
+ 59.3x3_1 +67.4x3_2 +104.7x3_4 + 49.7x3_5 +46.8x3_6 +103.2x3_7
+ 56.3x4_1 +24.7x4_2 +104.7x4_3 + 53.9x4_5 +88.0x4_6 +105.6x4_7
+ 96.8x5_1 +44.5x5_2 + 49.7x5_3 + 53.9x5_4 +17.6x5_6 + 15.2x5_7
+ 15.6x6_1 +38.5x6_2 + 46.8x6_3 + 88.0x6_4 +17.6x6_5 +130.6x6_7
+100.1x7_1 +97.6x7_2 +103.2x7_3 +105.6x7_4 +15.2x7_5 +130.6x7_6

subject to
csp1:       +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_7
                 -x2_1 -x3_1 -x4_1 -x5_1 -x6_1 -x7_1      = 1
csp2: +x2_1       +x2_3 +x2_4 +x2_5 +x2_6 +x2_7
        -x1_2       -x3_2 -x4_2 -x5_2 -x6_2 -x7_2      = 0
csp3: +x3_1 +x3_2       +x3_4 +x3_5 +x3_6 +x3_7
        -x1_3 -x2_3       -x4_3 -x5_3 -x6_3 -x7_3      = 0
csp4: +x4_1 +x4_2 +x4_3       +x4_5 +x4_6 +x4_7
        -x1_4 -x2_4 -x3_4       -x5_4 -x6_4 -x7_4      = 0
csp5: +x5_1 +x5_2 +x5_3 +x5_4       +x5_6 +x5_7
        -x1_5 -x2_5 -x3_5 -x4_5       -x6_5 -x7_5      = 0
csp6: +x6_1 +x6_2 +x6_3 +x6_4 +x6_5       +x6_7
        -x1_6 -x2_6 -x3_6 -x4_6 -x5_6       -x7_6      = 0
csp7: +x7_1 +x7_2 +x7_3 +x7_4 +x7_5 +x7_6
```

```
        -x1_7 -x2_7 -x3_7 -x4_7 -x5_7 -x6_7          =-1
end
```

Az optimális megoldás:

```
Status:     OPTIMAL
Objective:  ktg = 48.4 (MINimum)

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 x1_2         B              0
     2 x1_3         B              0
     3 x1_4         NL             0           8.7
     4 x1_5         NL             0          33.6
     5 x1_6         B              1
     6 x1_7         NL             0          51.7
     7 x2_1         NL             0          45.8
     8 x2_3         NL             0            31
     9 x2_4         B              0
    10 x2_5         NL             0          34.2
    11 x2_6         NL             0          45.8
    12 x2_7         NL             0          72.1
    13 x3_1         NL             0         118.6
    14 x3_2         NL             0         103.8
    15 x3_4         NL             0         116.4
    16 x3_5         NL             0          75.8
    17 x3_6         NL             0          90.5
    18 x3_7         NL             0         114.1
    19 x4_1         NL             0         103.9
    20 x4_2         NL             0          49.4
    21 x4_3         NL             0            93
    22 x4_5         NL             0          68.3
    23 x4_6         NL             0           120
    24 x4_7         NL             0         104.8
    25 x5_1         NL             0           100
    26 x5_2         NL             0          54.8
    27 x5_3         NL             0          23.6
    28 x5_4         NL             0          39.5
    29 x5_6         NL             0          35.2
    30 x5_7         B              1
    31 x6_1         NL             0          31.2
    32 x6_2         NL             0          31.2
    33 x6_3         NL             0           3.1
    34 x6_4         NL             0            56
    35 x6_5         B              1
    36 x6_7         NL             0          97.8
```

```
37 x7_1          NL          0          148.5
38 x7_2          NL          0          123.1
39 x7_3          NL          0           92.3
40 x7_4          NL          0          106.4
41 x7_5          NL          0           30.4
42 x7_6          NL          0          163.4
```

Látható, hogy rövidebb lett a 'legrövidebb' út. Az út hossza most csak 48,4 (a korábbi 82,6 helyett). Az út maga pedig: 1→6→5→7. Látható, hogy az 5-ös pontból a 6-os pontba mutató élen 'visszafelé' érdemes haladni, ezért nem találtuk meg ezt az utat korábban, egészen pontosan ez abban a felírásban nem volt megengedett.

5.2. Gyűrűs formalizáció

Az előzőekben megmutattuk hogyan tudjuk felírni a legrövidebb utat átrakodásos szállítási feladatként. Létezik azonban egy másik LP felírás is, ami ugyancsak megadja a legrövidebb utat, de jobban a gráfra koncentrál, mint a szállítási feladat formalizáció. Érdekes, hogy ez a formalizáció éppen a csomópontos felírásnak a duál feladata, ezért először írjuk fel a duál feladatot.

5.7. példa. *Tekintsük megint az 5.1. példát. Írjuk fel a szállítási feladat formalizáció duálját, és értelmezzük azt!*

Megoldás.

Ahhoz, hogy 'szép' alakban kapjuk meg a duál feladatot, érdemes pár algebrai változtatást tenni. Az eredeti feladat minimumfeladat volt, ahol a korlátok a következőek voltak (5.2. kód):

```
subject to
csp1:       +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_7 = 1
csp2: -x1_2       +x2_3 +x2_4 +x2_5 +x2_6 +x2_7 = 0
csp3: -x1_3 -x2_3       +x3_4 +x3_5 +x3_6 +x3_7 = 0
csp4: -x1_4 -x2_4 -x3_4       +x4_5 +x4_6 +x4_7 = 0
csp5: -x1_5 -x2_5 -x3_5 -x4_5       +x5_6 +x5_7 = 0
csp6: -x1_6 -x2_6 -x3_6 -x4_6 -x5_6       +x6_7 = 0
csp7: -x1_7 -x2_7 -x3_7 -x4_7 -x5_7 -x6_7       =-1
```

A továbbiak szempontjából kényelmesebb lesz, ha mindegyik korlátot beszorozzuk -1-gyel. Mivel a korlátok egyenlőség formájában teljesülnek, ezért ez az algebrai művelet megtehető:

```
subject to
csp1:        -x1_2 -x1_3 -x1_4 -x1_5 -x1_6 -x1_7 = -1
csp2: +x1_2        -x2_3 -x2_4 -x2_5 -x2_6 -x2_7 =  0
csp3: +x1_3 +x2_3        -x3_4 -x3_5 -x3_6 -x3_7 =  0
csp4: +x1_4 +x2_4 +x3_4        -x4_5 -x4_6 -x4_7 =  0
csp5: +x1_5 +x2_5 +x3_5 +x4_5        -x5_6 -x5_7 =  0
csp6: +x1_6 +x2_6 +x3_6 +x4_6 +x5_6        -x6_7 =  0
csp7: +x1_7 +x2_7 +x3_7 +x4_7 +x5_7 +x6_7        =  1
```

Az első sorban szereplő `csp1` korlát azt fejezi ki, hogy (pontosan) egy egységnyi áru útnak indul az 1-es pontból. Az utolsó sorban szereplő `csp7` korlát pedig azt fejezi ki, hogy (pontosan) egy egységnyi áru érkezik meg a 7-es (csomó)pontba. Megtehető, hogy mind a két korlát esetén kicseréljük a relációt >=-re. Ekkor az első korlát azt fogja kifejezni, hogy maximum 1 egységnyi áru indul útnak, az utolsó korlát pedig azt, hogy legalább egy egységnyi áru érkezik meg. Ez nyilván csak úgy lehet, ha pontosan egy áru indul útnak és pontosan egy áru érkezik meg.

Tudva azt, hogy pontosan egy áru indul útnak, és pontosan egy áru érkezik meg, a `csp2-csp6` korlátok mindegyike esetén kicserélhetjük a relációt >=-re. Ez a korlát azt jelenti, hogy minden csomópontba több áru érkezhet, mint amennyi továbbmegy, tehát valamennyi ottmaradhat. De ha maximum 1 árut indítunk útnak, és minimum 1-nek meg kell érkeznie, akkor sehol sem hagyhatunk el árukat, tehát hiába lesz `>=0` a korlát, valójában ezek a korlátok egyenlőség formájában teljesülnek. Azért lényeges számunkra ez az átalakítás, mert egyenlőséggel adott korláthoz tartozó duál-változó előjelkötetlen lenne —ami kezelhető ugyan LP feladatok esetén—, de mi nem foglalkoztunk ezzel az esettel. Ha >= relációval adott egy korlát, akkor (minimumfeladat esetén) nemnegatív duál-változó tartozik hozzá, ami számunkra kedvezőbb.

5.8. kód.

```
min
ktg:
```

```
+22.9x1_2+59.3x1_3+ 56.3x1_4+ 96.8x1_5+ 15.6x1_6+100.1x1_7
         +67.4x2_3+ 24.7x2_4+ 44.5x2_5+ 38.5x2_6+ 97.6x2_7
                  +104.7x3_4+ 49.7x3_5+ 46.8x3_6+103.2x3_7
                            + 53.9x4_5+ 88.0x4_6+105.6x4_7
                                      + 17.6x5_6+ 15.2x5_7
                                                +130.6x6_7
subject to
csp1:       -x1_2 -x1_3 -x1_4 -x1_5 -x1_6 -x1_7 >= -1
csp2: +x1_2       -x2_3 -x2_4 -x2_5 -x2_6 -x2_7 >= 0
csp3: +x1_3 +x2_3       -x3_4 -x3_5 -x3_6 -x3_7 >= 0
csp4: +x1_4 +x2_4 +x3_4       -x4_5 -x4_6 -x4_7 >= 0
csp5: +x1_5 +x2_5 +x3_5 +x4_5       -x5_6 -x5_7 >= 0
csp6: +x1_6 +x2_6 +x3_6 +x4_6 +x5_6       -x6_7 >= 0
csp7: +x1_7 +x2_7 +x3_7 +x4_7 +x5_7 +x6_7       >= 1

end
```

Az 5.8. kód esetén a duál feladatot könnyen fel tudjuk írni (vegyük észre hogy minimumfeladatról van szó). A `csp1-csp7` korlátokhoz tartozó duál-változókat `y1-y7` változók jelentik. Mivel mindegyik korlát `>=`, ezért a duál-változók nemnegatívok. Az `x` primálváltozók is nemnegatívok, ezért a duál feladatban `<=` korlátok lesznek.

5.9. kód.

```
max
y7-y1

subject to
el1_2: +y2-y1 <=  22.9
el1_3: +y3-y1 <=  59.3
el2_3: +y3-y2 <=  67.4
el1_4: +y4-y1 <=  56.3
el2_4: +y4-y2 <=  24.7
el3_4: +y4-y3 <= 104.7
el1_5: +y5-y1 <=  96.8
el2_5: +y5-y2 <=  44.5
el3_5: +y5-y3 <=  49.7
el4_5: +y5-y4 <=  53.9
el1_6: +y6-y1 <=  15.6
el2_6: +y6-y2 <=  38.5
el3_6: +y6-y3 <=  46.8
el4_6: +y6-y4 <=  88.0
el5_6: +y6-y5 <=  17.6
```

```
el1_7: +y7-y1 <= 100.1
el2_7: +y7-y2 <=  97.6
el3_7: +y7-y3 <= 103.2
el4_7: +y7-y4 <= 105.6
el5_7: +y7-y5 <=  15.2
el6_7: +y7-y6 <= 130.6

end
```

Az `el1_2` korlát úgy jön ki, hogy az 5.9. kód esetén az `x1_2` változó a `csp1` korlátban -1 együtthatóval szerepel, `csp2` korlátban pedig 1 együtthatóval. A `csp1` korláthoz `y1` duál-változó tartozik, `csp2` korláthoz pedig az `y2`. Így `y2-y1>=22.9`. A korlát jobboldala nem más, mint az `x1_2` változóhoz tartozó célfüggvény-együttható. Az 5.9. kód esetén a célfüggvényt úgy kapjuk meg, hogyha az 5.8. kód esetén a `csp1-csp7` korlátok jobboldalát beszorozzuk a duál-változókkal. A `csp1` korlát esetén a jobboldal -1, tehát `-y1`. A `csp2-csp6` korlátok esetén a jobboldal 0, tehát `0y2`, `0y3`, ... `0y6`. A `csp7` korlát esetén a jobboldal `1`, tehát `+1y7`. Ha összegezzük őket, megkapjuk a `y7-y1` célfüggvényt.

A továbbiakban az 5.8. kódhoz tartozó feladatra 'csomópontos' formalizációként (vagy csomópontos felírásként) fogunk hivatkozni (természetesen az 5.2. kód és az 5.6. kódhoz tartozó feladat is csomópontos formalizáció), az 5.9. kódhoz tartozó feladatra pedig 'gyűrűs' formalizációként.

Az 5.9. kód felíráshoz nagyon szemléletes magyarázat társítható: a gráfnak a csúcspontjai legyenek gyűrűk. A gráf élei pedig legyenek cérnaszálak, amelyek hossza az élek hosszának felel meg. A 1-es és 7-es csúcsnak megfelelő gyűrűt pedig elkezdjük ellentétes irányba húzni. Addig húzzuk, ameddig lehet, tehát a cérnaszálak egyszer csak megfeszülnek. A feszes cérnaszálak alkotják a legrövidebb utat.

Tehát másképpen megfogalmazva a duálváltozók az 1-es csúcsból az adott csúcsba vezető legrövidebb út hosszát adják meg. Így a duál feladat korlátait is tudjuk értelmezni. Például az `el2_3: +y3-y2 <=  67.4` korlátot átírhatjuk `+y3 <= 67.4 + y2` alakba, ami azt fejezi ki, hogy a 3-as csúcsba vezető legrövidebb út nem lehet hosszabb, mint a 2-es csúcsba vezető legrövidebb út és még 67.4, ami a 2-es csúcsból a 3-asba vezető él hossza.

Az eredmény:

```
Status:     OPTIMAL
Objective:  obj = 82.6 (MAXimum)

   No.   Row name   St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 el1_2        NU          22.9             1
     2 el1_3        B           17.7
     3 el2_3        B           -5.2
     4 el1_4        B           13.5
     5 el2_4        B           -9.4
     6 el3_4        B           -4.2
     7 el1_5        B           67.4
     8 el2_5        NU          44.5             1
     9 el3_5        NU          49.7         < eps
    10 el4_5        NU          53.9         < eps
    11 el1_6        B              0
    12 el2_6        B          -22.9
    13 el3_6        B          -17.7
    14 el4_6        B          -13.5
    15 el5_6        B          -67.4
    16 el1_7        B           82.6
    17 el2_7        B           59.7
    18 el3_7        B           64.9
    19 el4_7        B           69.1
    20 el5_7        NU          15.2             1
    21 el6_7        B           82.6

   No. Column name  St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 y7           B           82.6
     2 y1           NL             0         < eps
     3 y2           B           22.9
     4 y3           B           17.7
     5 y4           B           13.5
     6 y5           B           67.4
     7 y6           NL             0         < eps
```

A célfüggvény értéke megint csak 82,6, mint a csomópontos felírás esetén. Kérdés, hogy ezen formalizáció esetén honnan látjuk, hogy melyik élek alkotják a legrövidebb utat? Ha visszagondolunk ez előbbi szemléltetésre, akkor azok az élek alkotják, ahol a cérnaszál megfeszül. Ha feszülő cérnák hosszát kicsivel rövidebbre vesszük, akkor a legrövidebb út hossza is csökkenni fog. Ezt mi úgy mondjuk, hogy a korlát árnyékára

1. És valóban, az 1-es csúcsból a 2-es csúcsba mutató él (egészen pontosan ehhez az élhez tartozó korlát), valamint a 2-es csúcsból az 5-ös csúcsba, valamint az 5-ös csúcsból a 7-es csúcsba mutató él árnyékárai 1-ek. Tehát a legrövidebb út ugyanaz, mint korábban volt: 1→2→5→7.

Érdemes még pár pillanatig elidőzni az eredménynél. Ha alaposabban szemügyre vesszük az eredményeket, akkor a primál-duál feladatpár is jobban megértető ezen feladaton keresztül.

A komplementaritási tétel alapján tudjuk, hogy bázisváltozók esetén a duál feladatban a korlát egyenlőség formájában teljesül. A legrövidebb utat alkotó élek esetén ez triviális, hiszen az árnyékár 1, de a csomópontos felírás esetén az `x3_5` és az `x4_5` változók is bázisváltozók, viszont csak 0 értékkel. Ellenőrizzük le, hogy a gyűrűs felírásban `el3_5` és `el4_5` korlátok is egyenlőség formájában teljesülnek (mivel erősen degenerált a probléma nem biztos, hogy mindig ilyen szépen kijön, mint most, de elméletileg ez a megfeleltetés mindig megtehető). Miért pont `x3_5` és `x4_5` változók lesznek bázisváltozók a csomópontos felírás esetén, és lehetnek-e mások is? A gyűrűs felírásban az `y7` változó értéke 82,6 az `y1` változó értéke pedig 0. Ezt úgy tudjuk elképzelni, hogy az 1-es (csomó)pontnak megfelelő gyűrű a számegyenes 0 pontjához van kötve, a 7-es pontot pedig a számegyenes 82,6 értékéig tudjuk húzni. Az 5-ös pontból a 7-es pontba mutató él hossza 15,2, tehát az 5-ös gyűrű a 82,6-15,2=67,4 értéknél van (`y5=67.4`). Hasonlóan: a 2-es pontból az 5-ös pontba mutató él hossza 44,5, tehát a 2-es ponthoz tartozó gyűrű a 67,4-44,5=22,9 értéknél lesz, már csak azért is, mert az 1-es pontból a 2-es pontba mutató él hossza 22,9. De hogy magyarázzuk az `y3=17.7` értéket? Úgy lehet ezt szemléletesen magyarázni, hogy a számegyenes ne vízszintesen menjen, hanem függőlegesen (7-es csomópontot szemléltető karikát felvesszük az asztalról és szép lassan addig emeljük, amíg megfeszül a cérnaszálak rendszere). A 7-es gyűrűt rögzítjük a 82,6-os értéknél (vagy más megfogalmazásban eddig tudjuk húzni), a nehézségi erő pedig lefele húzza a gyűrűket. Ekkor a 3-as csúcsból az 5-ös csúcsba mutató élt jelképező cérnaszál is megfeszül. A 3-as ponthoz tartozó gyűrű az 5-ös ponthoz tartozó gyűrűnél (ami a legrövidebb út része) 49,7-tel lesz lentebb, tehát a 67,4-49,7=17,7-es értéknél. Hasonló a 4-es karika esete. Az `x4_5` változó bázisváltozó a csomópontos felírásban (`el4_5` korlát egyenlőség formájában teljesül), tehát a 4-es gyűrű az 5-ös gyűrűhöz képest 53,9 értékkel lesz lentebb: 67,4-53,9=13,5. Hátra van még `y6=0` érték magyarázata. Fontos látni, hogy ez nem bázisváltozó (`St=NL`). Tehát a 6-os ponthoz tartozó gyűrű a 0 pontban a földön lesz anélkül, hogy bármelyik cérnaszál is feszes lenne, amelyik hozzá van kötve. Ha megengednénk, hogy a 0 pontnál lejjebb legyen, akkor előbb-utóbb valamelyik cérnaszál megfeszülne és a ponthoz tartozó karika valamely negatív értékhez kerülne. Ha kíváncsiak vagyunk, akkor a -48-as értéknél lesz a gyűrű és a 6-os pontból a 7-es pontba mutató élt reprezentáló cérna feszülne meg ekkor (de ez nem olvasható le közvetlenül az eredményfülből).

Mit jelent az hogy a bázis változása nem jelenti a legrövidebb út változását. A 3-as és 7-es gyűrű jelen esetben 64,9 távolságra van egymástól, de ezt a két gyűrűt összekötő cérnaszál hossza 103,2. Ha az él hosszát lecsökkentjük 64,9 alá, akkor ez a cérnaszál feszes lesz, a 3-as gyűrű kicsit feljebb kerül, de a legrövidebb út nem változik. Próbáljuk ki, legyen a 3-as pontból a 7-es pontba mutató él hossza 64,8. Ekkor az optimális megoldás:

```
Status:     OPTIMAL
Objective:  obj = 82.6 (MAXimum)

   No.   Row name   St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 el1_2        NU          22.9             1
     2 el1_3        B           17.8
     3 el2_3        B           -5.1
     4 el1_4        B           13.5
     5 el2_4        B           -9.4
     6 el3_4        B           -4.3
     7 el1_5        B           67.4
     8 el2_5        NU          44.5             1
     9 el3_5        B           49.6
    10 el4_5        NU          53.9         < eps
    11 el1_6        B              0
    12 el2_6        B          -22.9
    13 el3_6        B          -17.8
    14 el4_6        B          -13.5
    15 el5_6        B          -67.4
    16 el1_7        B           82.6
```

```
    17 el2_7        B            59.7
    18 el3_7        NU           64.8              < eps
    19 el4_7        B            69.1
    20 el5_7        NU           15.2                  1
    21 el6_7        B            82.6

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 y7           B            82.6
     2 y1           NL              0              < eps
     3 y2           B            22.9
     4 y3           B            17.8
     5 y4           B            13.5
     6 y5           B            67.4
     7 y6           NL              0              < eps
```

Látható, hogy a legrövidebb út nem változott (a célfüggvény értéke ugyanannyi, mint korábban, továbbá ugyanazon korlátok esetén lesz az árnyékár 1), de a bázis változott: `el3_7` korlát korábban egyenlőtlenség formájában teljesült (`St=B`), most viszont egyenlőség formájában (`St=NU`).

Térjünk vissza az eredeti paraméterezéshez (5.9. kód). Kérdés, hogy ugyanehhez a legrövidebb úthoz van-e másik bázis? Erre nagyon szemléletes magyarázat adható. A számegyenes ne felfele menjen, hanem lefele (a gyűrűk helye negatív számoknál lesz, de az abszolút értéket tekintjük). A nehézségi erő most fordítva hat, most is bizonyos cérnák megfeszülnek, de nem azok, mint korábban. Ez is meghatároz egy bázist. De mondhatjuk azt is, hogy 'meggörbítjük a számegyenest (szemléletesen mindkét végét felfele húzzuk). Ekkor is megfeszülnek bizonyos cérnaszálak, ennek az állapotnak is megfeleltethető egy bázis.

Az érzékenységvizsgálat kérdését a csomópontos felírásnál már tárgyaltuk, de beszéljünk róla most is, ez is segíti a primál-duál feladatpár jobb megismerését. Induljunk ki megint az 5.9. kódhoz tartozó feladatból. Korábban megvizsgáltuk, hogy mennyivel kell csökkenteni egy adott él hosszát ahhoz, hogy a legrövidebb út része legyen. Az 1-es pontból a 3-as pontba mutató él esetén ez az érték 17,7 volt. A duál-feladat esetén hogyan tudjuk ezt leolvasni? Az `el1_3` korlát egyenlőtlenség formájában teljesül, a korlát értéke (`Activity` oszlopban szereplő érték) 17,7 (szemléletesen a két gyűrű távolsága). Ha az él hossza ezen érték alá csökken, akkor a korlát egyensúly formájában fog teljesülni. Próbáljuk, ki: csökkentsük az 1-es pontból a 3-as pontba mutató él hosszát 17,69-re. Az eredmény:

```
Status:     OPTIMAL
Objective:  obj = 82.59 (MAXimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 el1_2        B          22.89
     2 el1_3        NU         17.69             1
     3 el2_3        B           -5.2
     4 el1_4        B          13.49
     5 el2_4        B           -9.4
     6 el3_4        B           -4.2
     7 el1_5        B          67.39
     8 el2_5        NU          44.5         < eps
     9 el3_5        NU          49.7             1
    10 el4_5        NU          53.9         < eps
    11 el1_6        B              0
    12 el2_6        B         -22.89
    13 el3_6        B         -17.69
    14 el4_6        B         -13.49
    15 el5_6        B         -67.39
    16 el1_7        B          82.59
    17 el2_7        B           59.7
    18 el3_7        B           64.9
    19 el4_7        B           69.1
    20 el5_7        NU          15.2             1
    21 el6_7        B          82.59

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 y7           B          82.59
     2 y1           NL             0         < eps
     3 y2           B          22.89
     4 y3           B          17.69
     5 y4           B          13.49
     6 y5           B          67.39
     7 y6           NL             0         < eps
```

A legrövidebb út megváltozott (most is, akárcsak a csomópontos felírásnál): 1→3→5→7.

Térjünk vissza az eredeti feladathoz (5.9. kód). Nézzük meg mi történik, ha a 3-as pontból a 4-es pontba mutató élt csökkentjük. Az `el3_4` korlát esetén az activity érték -4,2, tehát erre az értékre kell lecsökkenteni az él hosszát. Az él hosszára negatív hosszúságot kapunk, amit egy kis absztrakcióval tudunk értelmezni. Csökkentsük tehát az el3_4 korlát jobboldalát a -4,2-es érték alá egy kicsivel (-4,21). Az eredmény:

```
Status:     OPTIMAL
Objective:  obj = 82.6 (MAXimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 el1_2        NU          22.9             1
     2 el1_3        B          17.71
     3 el2_3        B          -5.19
     4 el1_4        B           13.5
     5 el2_4        B           -9.4
     6 el3_4        NU         -4.21         < eps
     7 el1_5        B           67.4
     8 el2_5        NU          44.5             1
     9 el3_5        B          49.69
    10 el4_5        NU          53.9         < eps
    11 el1_6        B              0
    12 el2_6        B          -22.9
    13 el3_6        B         -17.71
    14 el4_6        B          -13.5
    15 el5_6        B          -67.4
    16 el1_7        B           82.6
    17 el2_7        B           59.7
    18 el3_7        B          64.89
    19 el4_7        B           69.1
    20 el5_7        NU          15.2             1
    21 el6_7        B           82.6

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 y7           B           82.6
     2 y1           NL             0         < eps
     3 y2           B           22.9
     4 y3           B          17.71
     5 y4           B           13.5
     6 y5           B           67.4
     7 y6           NL             0         < eps
```

Látható, hogy az el3_4 korlát esetén az St érték megváltozott B-ről NU-ra, tehát megváltozott a bázis is. Az el3_4 korláthoz tartozik is árnyékár, de ez 0

(< `eps`), tehát a 3-as pontból a 4-es csúcspontba mutató él nem lett a legrövidebb út része. Az érzékenységvizsgálati határokból leolvasható, hogy ez a bázis változatlan marad, ha az él hossza -45,8 és -4,2 között változik. Ugyanazt kaptuk most is, mint a csomópontos felírás esetén, csak most az `Activity range` oszlop a mérvadó a számunkra.

```
   No. Column name  St      Activity
                               range
------ ------------ -- -------------
     6 el3_4        NU     -45.80000
                            -4.20000
```

A teljesség kedvéért most is csökkentsük az él hosszát a -45,8 érték alá egy kicsivel (-45,81). Így a 3-as pontból a 4-es pontba mutató él a legrövidebb út része lesz (megegyezően a csomóponti felírással).

Hátramaradt még annak kérdése, hogy a legrövidebb utat alkotó élek milyen határok között változhatnak, hogy a legrövidebb út változatlan maradjon. Ez a kérdés is emlékezetet minket az érzékenységvizsgálati határokra, de az eddigiek ismeretében óvatosak vagyunk. Vizsgáljuk meg az eredeti feladat (5.9. kód) érzékenységvizsgálati határait!

```
   No. Column name  St      Activity
                               range
------ ------------ -- -------------
     1 el1_2        NU       9.40000
                            40.40000
     8 el2_5        NU      31.00000
                            62.00000
    20 el5_7        NU     -67.40000
                            32.70000
```

Az intervallumok felső határa ugyanaz, mint korábban, viszont korábban az alsó határ `-Inf` volt, ami logikus is, hiszen ha már így is a legrövidebb út része volt, ha rövidítjük, akkor még inkább az lesz. Miért nem ez az érték szerepel akkor az intervallumok alsó határánál? A válasz egyszerű, ha az 1-es pontból a 2-es pontba mutató él 9,4-es érték alá csökken, akkor a 4-es gyűrű eléri a földet, és azért nem lesz feszes a cérnaszál. Gondoljunk bele, hogy a 4-es gyűrű 13,5-ös értéknél van, és az

1-es pontból a 2-es pontba mutató él hosszát (22,9) pont ennyivel kell csökkenteni, a 9,4-es értékig (22,9-13,5=9,4). Tehát ebben az esetben is a bázis változása nem feltétlenül jelenti a legrövidebb út változását. De ezt könnyen tudjuk kezelni, az alsó határokat mindig kicseréljük a `-Inf` értékre, és akkor megkapjuk a helyes intervallumot. De látszik az is, hogy erre a kérdésre célszerű a csomópontos felírás segítségével választ adni.

Érdemes egy két modellezési kérdésről is szót ejteni. A gyűrűs formalizáció esetén a célfüggvény `y7-y1`. Ezt úgy tudjuk interpretálni, hogy az 1-es és 7-es gyűrűt egymástól ellentétes irányba húzzuk. Az első gyűrű a 0 pontban van, a 7-es gyűrű pedig a 82,6 pontban. De ez nem törvényszerű, ugyanolyan optimális megoldás az is, ha az 1-es gyűrű a 1-es pontban van, a 7-es gyűrű pedig a 83,6 pontban. De számunkra kellemesebb az előző eset (azaz egyes gyűrű a 0 pontban) vizsgálat. Ezt explicitté is tehetjük a felírás esetén a `y1<=0` korláttal (mivel `y1` nemnegatív változó az `y1<=0` egyenértékű a `y1=0` korláttal, de a primál-duál feladatpár esetén egyszerűbb a `y1<=0` korlátot tekinteni). Ekkor viszont a célfüggvényt is megváltoztathatjuk egyszerűen `y7-re`.

5.10. kód.
```
max
y7

subject to
kezd:  +y1    <=     0
el1_2: +y2-y1 <=  22.9
el1_3: +y3-y1 <=  59.3
el2_3: +y3-y2 <=  67.4
el1_4: +y4-y1 <=  56.3
el2_4: +y4-y2 <=  24.7
el3_4: +y4-y3 <= 104.7
el1_5: +y5-y1 <=  96.8
el2_5: +y5-y2 <=  44.5
el3_5: +y5-y3 <=  49.7
el4_5: +y5-y4 <=  53.9
el1_6: +y6-y1 <=  15.6
el2_6: +y6-y2 <=  38.5
el3_6: +y6-y3 <=  46.8
el4_6: +y6-y4 <=  88.0
el5_6: +y6-y5 <=  17.6
```

```
el1_7: +y7-y1 <= 100.1
el2_7: +y7-y2 <=  97.6
el3_7: +y7-y3 <= 103.2
el4_7: +y7-y4 <= 105.6
el5_7: +y7-y5 <=  15.2
el6_7: +y7-y6 <= 130.6

end
```

Az előzőekben megvizsgáltuk a kérdést, hogy a 3-as csúcsból a 4-es csúcsba mutató élnek mennyivel kell rövidebbnek lennie ahhoz, hogy a legrövidebb út része legyen. Láttuk, hogy az érzékenységvizsgálati határokkal nem feltétlenül kapunk választ a kérdésre, mert a bázis változása nem feltétlenül jár együtt a legrövidebb útváltozásával. Egy iteratív eljárással megkaphatjuk a kérdésre a választ. De a kérdést megválaszolhatjuk egyetlen (újabb) futtatással is.

5.11. példa. *Adjunk meg egy olyan LP felírást, amely esetén egyetlen futtatással megállapítható, hogy egy olyan él hosszát, ami nem a legrövidebb út része, meddig kell csökkenteni ahhoz, hogy a legrövidebb út része legyen!*

Megoldás.

Vizsgáljuk megint a 3-as csúcsból a 4-es csúcsba mutató élt. A legrövidebb utakhoz tartozó korlátok esetén (de csak ott) a <= korlátot lecseréljük egyenlőségre, a célfüggvényt pedig a 3-as és 4-es gyűrű távolságára (`y4-y3`) cseréljük, amelyet most minimalizálnunk kell.

5.12. kód.

```
min
y4-y3

subject to
kezd:  +y1    <=     0
el1_2: +y2-y1  =  22.9
el1_3: +y3-y1 <=  59.3
el2_3: +y3-y2 <=  67.4
el1_4: +y4-y1 <=  56.3
el2_4: +y4-y2 <=  24.7
el3_4: +y4-y3 <= 104.7
el1_5: +y5-y1 <=  96.8
```

```
el2_5: +y5-y2  =  44.5
el3_5: +y5-y3 <=  49.7
el4_5: +y5-y4 <=  53.9
el1_6: +y6-y1 <=  15.6
el2_6: +y6-y2 <=  38.5
el3_6: +y6-y3 <=  46.8
el4_6: +y6-y4 <=  88.0
el5_6: +y6-y5 <=  17.6
el1_7: +y7-y1 <= 100.1
el2_7: +y7-y2 <=  97.6
el3_7: +y7-y3 <= 103.2
el4_7: +y7-y4 <= 105.6
el5_7: +y7-y5  =  15.2
el6_7: +y7-y6 <= 130.6

end
```

A célfüggvény értéke -45,8, tehát ha a 3-as pontból a 4-es pontba mutató él hosszát -45,8 alá csökkentjük, akkor a legrövidebb út részévé válik. Amennyiben az él hosszára a negatív szám nem elfogadható, akkor nem tudjuk olyan rövidre venni ezt az élt, hogy a legrövidebb út része legyen.

5.13. példa. *Hogyan lehet irányítatlan gráfon a gyűrűs formalizációt használni, a legrövidebb út keresésére?*

Megoldás.

Első ránézésre úgy tűnhet, mintha már eddig is irányítatlan gráffal dolgoztunk volna. Ennek azaz oka, hogy a cérnaszálas szemléltetés megvezet minket. Ha ez a szemléltetés pontos lenne, akkor akár ellenkező irányban is dolgozhatnánk: a 7-es gyűrűt rögzítjük a 0 ponthoz és az 1-es gyűrűt húzzuk pozitív irányban. Nyilván a cérnaszálak esetén ugyanazt a legrövidebb utat kapnánk, de ha matematikailag is végigvezetjük, akkor a következő változtatásokat kell eszközölni a 5.10. kódon: a célfüggvényt cseréljük ki a `y1` változóra, a `kezd` korlátot pedig az `y7<=0` kifejezésre. Így viszont a célfüggvény nem lesz korlátos, mivel ebben a felírásban a cérnaszálak az egyik irányba korlátlanul nyúlhatnak, a másik irányba viszont nem. Ezt viszont nehezen lehet a cérnaszálas szemléltetéssel értelmezni.

Ha irányítatlan gráffal dolgozunk, akkor a megoldás teljesen analóg a csomóponti felírással. Ha például az 1-es és 2-es pont közötti él hossza 22,9, és ezen az élen mind a két irányban közlekedhetünk, akkor fel kell venni az `y2-y1<=22.9` és `y1-y2<=22.9` korlátokat is. Természetesen itt is előfordulhat, hogy az oda és visszaút különböző költséggel járnak.

5.14. kód.

```
max
y7-y1

subject to
kezd:  +y1    <=     0
el1_2: +y2-y1 <=  22.9
el2_1: +y1-y2 <=  22.9
el1_3: +y3-y1 <=  59.3
el3_1: +y1-y3 <=  59.3
el2_3: +y3-y2 <=  67.4
el3_2: +y2-y3 <=  67.4
el1_4: +y4-y1 <=  56.3
el4_1: +y1-y4 <=  56.3
el2_4: +y4-y2 <=  24.7
el4_2: +y2-y4 <=  24.7
el3_4: +y4-y3 <= 104.7
el4_3: +y3-y4 <= 104.7
el1_5: +y5-y1 <=  96.8
el5_1: +y1-y5 <=  96.8
el2_5: +y5-y2 <=  44.5
el5_2: +y2-y5 <=  44.5
el3_5: +y5-y3 <=  49.7
el5_3: +y3-y5 <=  49.7
el4_5: +y5-y4 <=  53.9
el5_4: +y4-y5 <=  53.9
el1_6: +y6-y1 <=  15.6
el6_1: +y1-y6 <=  15.6
el2_6: +y6-y2 <=  38.5
el6_2: +y2-y6 <=  38.5
el3_6: +y6-y3 <=  46.8
el6_3: +y3-y6 <=  46.8
el4_6: +y6-y4 <=  88.0
el6_4: +y4-y6 <=  88.0
el5_6: +y6-y5 <=  17.6
el6_5: +y5-y6 <=  17.6
el1_7: +y7-y1 <= 100.1
el7_1: +y1-y7 <= 100.1
el2_7: +y7-y2 <=  97.6
el7_2: +y2-y7 <=  97.6
el3_7: +y7-y3 <= 103.2
```

```
el7_3: +y3-y7 <= 103.2
el4_7: +y7-y4 <= 105.6
el7_4: +y4-y7 <= 105.6
el5_7: +y7-y5 <=  15.2
el7_5: +y5-y7 <=  15.2
el6_7: +y7-y6 <= 130.6
el7_6: +y6-y7 <= 130.6

end
```

Az optimális megoldás:

```
Status:     OPTIMAL
Objective:  obj = 48.4 (MAXimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 kezd         NU             0         < eps
     2 el1_2        B              0
     3 el2_1        B              0
     4 el1_3        B              0
     5 el3_1        B              0
     6 el2_3        B              0
     7 el3_2        B              0
     8 el1_4        B              0
     9 el4_1        B              0
    10 el2_4        B              0
    11 el4_2        B              0
    12 el3_4        B              0
    13 el4_3        B              0
    14 el1_5        B           33.2
    15 el5_1        B          -33.2
    16 el2_5        B           33.2
    17 el5_2        B          -33.2
    18 el3_5        B           33.2
    19 el5_3        B          -33.2
    20 el4_5        B           33.2
    21 el5_4        B          -33.2
    22 el1_6        NU          15.6             1
    23 el6_1        B          -15.6
    24 el2_6        B           15.6
    25 el6_2        B          -15.6
    26 el3_6        B           15.6
    27 el6_3        B          -15.6
    28 el4_6        B           15.6
```

```
    29 el6_4         B           -15.6
    30 el5_6         B           -17.6
    31 el6_5         NU           17.6              1
    32 el1_7         B            48.4
    33 el7_1         B           -48.4
    34 el2_7         B            48.4
    35 el7_2         B           -48.4
    36 el3_7         B            48.4
    37 el7_3         B           -48.4
    38 el4_7         B            48.4
    39 el7_4         B           -48.4
    40 el5_7         NU           15.2              1
    41 el7_5         B           -15.2
    42 el6_7         B            32.8
    43 el7_6         B           -32.8

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 y7           B            48.4
     2 y1           B               0
     3 y2           NL              0         < eps
     4 y3           NL              0         < eps
     5 y4           NL              0         < eps
     6 y5           B            33.2
     7 y6           B            15.6
```

A legrövidebb út hossza 48,4 (akárcsak a 5.5 példa esetén) és az 1→6→5→7 csúcsok között megy (akárcsak a 5.5. példa esetén).

5.3. Gyakorló feladatok

5.3.1. Egyszerűbb feladatok

5.15. példa. *Tekintsük a következő táblázatot, amelyben egy gráf 10 csúcspontja közötti időbeni távolságok vannak megadva, ahol a két csúcs közötti út hossza függ* *az út irányától:*

	1	2	3	4	5	6	7	8	9	10
1			26	19	29		51	75		350
2	72			39	72	52	80	45	79	
3	30	15		46	50	80	17	79	48	
4	26				36	56	65	50	68	
5	66	40	16	67		57		79	76	
6	13		31					75	35	65
7		51	11		43	51		53	12	45
8	45	10	58	10	67	75	59		56	22
9		35	17	36		17	62	10		55
10	13	47	63	66	34		76	58	63	

Adja meg a csomópontos és a gyűrűs formalizációval is a feladat megoldását!

Megoldás a 300. oldalon.

5.16. példa. *Tekintsük az előző feladatot. Mekkora a legrövidebb út a 7-es pontból az 1-es pontba (visszafele)?*

Megoldás a 306. oldalon.

5.3.2. Nehezebb feladatok

5.17. példa. *Tekintsük a 5.10. Kódban megadott gyűrűs felírást! Adjuk meg az ehhez tartozó duál feladatot (csomóponti) felírást!*

Megoldás a 310. oldalon.

5.18. példa. *Irányítatlan gráf esetén, hogyan tudjuk meghatározni, hogy egy él milyen határok között változhat úgy, hogy a legrövidebb út nem változik?*

Megoldás a 310. oldalon.

6. fejezet

Kritikus út feladat

Projektütemezésben játszik fontos feladatot az ún. kritikus út feladat. Magát a módszert amerikai kutatók fejlesztették ki, és meghatározó szerepet játszott pl. abban, hogy a Polaris rakéták a határidő előtt 2 évvel működőképesek voltak.

Manapság a kritikus út feladat (vagy angolul Critical Path Method, CPM) elterjedté vált, és már nem csak az operációkutatási kurzusok része, hanem menedzsment szakokon is elengedhetetlen. Megmutatjuk, hogy operációkutatási szempontból a CPM feladat szoros összefüggésben van a legrövidebb út feladattal, akár nevezhetnénk ezt a problémát 'leghosszabb' út feladatnak is.

6.1. Alapfeladat

6.1. példa. *Egy építési vállalkozó egy épület felépítésének projektje során az alábbi táblázatban szereplő fázisokat (tevékenységeket) tudja elkülöníteni. A táblázat megadja a fázis hosszát (napokban) és az előfeltételeket is, amelyeknek teljesülnie kell ahhoz, hogy az adott tevékenység elkezdődhessen.*

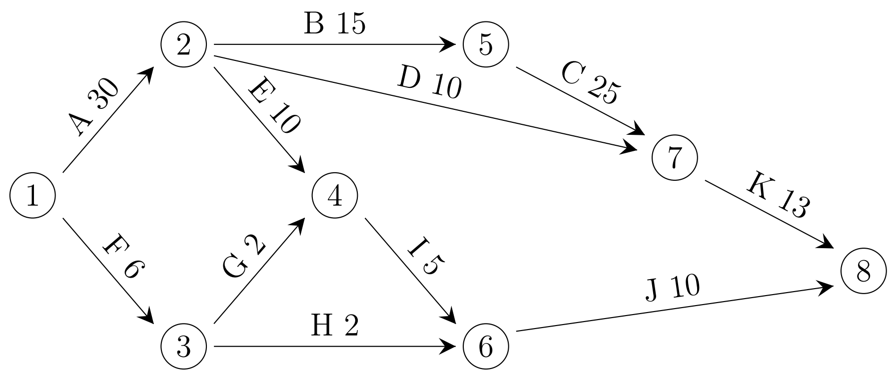
Fázis neve	*Fázis kódja*	*Fázis hossza*	*Előfeltételek*
Falazás	*A*	*30*	-
Gázszerelés	*B*	*15*	*A*
Fűtésszerelés	*C*	*25*	*B*
Villanyszerelés	*D*	*10*	*A*
Nyílászárók beépítése	*E*	*10*	*A*
Hirdetési piac feltérképezése	*F*	*6*	-
Fényképész kiválasztása	*G*	*2*	*F*
Nyomda kiválasztása	*H*	*2*	*F*
Fényképezés	*I*	*5*	*E,G*
Prospektus elkészítése	*J*	*10*	*I,H*
Festés	*K*	*13*	*C,D*

Adjunk meg egy gráfot, ami reprezentálja a tevékenységeket!

Megoldás.

A tevékenységeket reprezentálhatják a gráfban az élek, vagy a csúcspontok is. Mi csak azzal az esettel foglalkozunk, amikor élek jelentik a tevékenységeket. Ebben az esetben az él 'súlya' a tevékenység időtartama.

A gráf felrajzolása nem mindig egyszerű feladat. Mivel az LP feladat felírásához nem feltétlenül szükséges a gráf felrajzolása, ezért csak megadjuk a gráfot, de nem mutatjuk be részletesen, hogy milyen módszerrel lehet ezt megtenni. Ezt a gráfot elég egyszerű felrajzolni, bonyolultabb eset szerepel a 6.13. példában.



6.2. példa. *Tekintsük a 6.1. példában megadott feladatot! Legkevesebb mennyi idő alatt alatt lehet befejezni a projektet? Melyik tevékenységgel nem lehet késni ahhoz, hogy a projekt idejében befejeződjön?*

Megoldás.

A feladat megoldásához használjuk a legrövidebb út feladatoknál megismert gyűrűs formalizációt. Természetesen bizonyos pontokon módosítást kell eszközölni. Mielőtt megadjuk ezt a formalizációt, bemutatunk megint egy szemléltetést, ami hozzásegít a feladat megértéséhez. Vegyük elő megint a legrövidebb út feladatnál megismert gyűrűket! A gyűrűk most nem egyszerűen cérnaszálakkal vannak összekötve, hanem a gyűrűk között mindig van egy rugó, aminek nyugalmi hossza megegyezik az él hosszával (a tevékenység időtartamával). Azonban a rugók hossza (most feltesszük, hogy bármeddig) nyúlhat.

Ezeket a gyűrűkből és a rugókból készített szerkezetet berakjuk egy csőbe és a két végén elkezdjük összenyomni. Előbb-utóbb a rugók összetömörülnek és nem tudjuk tovább nyomni ezt a szerkezetet. Tehát lesz (legalább egy) olyan egymást követő rugók sorozata, amelyek nyugalmi állapotban maradnak. Ezek a rugók (élek) alkotják a kritikus utat. Több rugó pedig ki fog nyúlni, ami azt jelenti, hogy azokra a tevékenységekre, amelyeket ezek a rugók szemléltetnek, a szükségesnél több idő áll a rendelkezésre.

Nézzük meg ezt a számpéldával. Az 1-es pontból a 2-es pontba mutató él hossza 30 egység, tehát az 1-es gyűrű és a kettes gyűrű nem lehet közelebb egymáshoz, mint 30 egység: `y2-y1>=30`. Az 1-es és 3-as pont közötti él hossza 6 egység, tehát az 1-es és 3-as pont nem lehet egymáshoz közelebb, mint 6 egység: `y3-y1>=6`, stb... Szeretnénk a két szélső karikát a lehető legközelebb nyomni egymáshoz (a projektet a lehető leghamarabb befejezni), tehát a célfüggvény: `min y8-y1`. Most már fel tudjuk írni a teljes feladatot:

6.3. kód.

```
min
y8-y1

subject to
el1_2: y2 - y1 >= 30
el1_3: y3 - y1 >=  6
el2_4: y4 - y2 >= 10
el2_5: y5 - y2 >= 15
el2_7: y7 - y2 >= 10
el3_4: y4 - y3 >=  2
el3_6: y6 - y3 >=  2
```

```
el4_6: y6 - y4 >=  5
el5_7: y7 - y5 >= 25
el6_8: y8 - y6 >= 10
el7_8: y8 - y7 >= 13

end
```

A feladat optimális megoldása:

```
Status:     OPTIMAL
Objective:  obj = 83 (MINimum)

   No.   Row name   St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 el1_2        NL            30             1
     2 el1_3        NL             6         < eps
     3 el2_4        NL            10         < eps
     4 el2_5        NL            15             1
     5 el2_7        B             40
     6 el3_4        B             34
     7 el3_6        B             39
     8 el4_6        NL             5         < eps
     9 el5_7        NL            25             1
    10 el6_8        B             38
    11 el7_8        NL            13             1

   No. Column name  St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 y8           B             83
     2 y1           NL             0         < eps
     3 y2           B             30
     4 y3           B              6
     5 y4           B             40
     6 y5           B             45
     7 y7           B             70
     8 y6           B             45
```

A projekt befejezésének legrövidebb ideje 83 nap. Melyek azok a tevékenységek, amelyekkel nem lehet késni? Ezeket a tevékenységeket hívjuk kritikus tevékenységeknek. Ha egy tevékenység kritikus tevékenység, akkor a legkisebb késése is a projekt késedelmét okozza. Pl.: ha az A tevékenységgel (az 1-es pontból a 2-es pontba mutató él) késünk, akkor az `el1_2` korlát jobboldala megnövekszik (mondjuk 1 egységgel). Amennyiben az A kritikus tevékenység, akkor a hosszának egy egységgel növelésével a projektet csak késedelmesen tudjuk befejezni, vagyis nő a célfüggvény értéke. Ha egy adott korlát jobboldalának egy egységnyi növekedése a célfüggvény 1 egységnyi növekedését vonja maga után, akkor azt mondjuk, hogy az árnyékára 1. Tehát egy tevékenység akkor lesz kritikus, ha a hozzá tartozó korlát árnyékára 1.

Az eredményfülről leolvasható, hogy az `el1_2`, `el2_5`, `el5_7` és `el7_8` korlátok esetén lesz az árnyékár 1. Látható, hogy ez meghatároz egy utat: 1→2→5→7→8. Ehhez a korlátokhoz tartozó tevékenységek: A, B, C és K tevékenységek.

Megoldás #2.

Írjuk fel a 6.3. kódban megadott LP feladat duálját (lásd 6.14. példa). Ekkor (átalakítások után) megint egy (összetett) szállítási feladatot kapunk, amit a legrövidebb út feladatnál csomópontos formalizációnak hívtunk.

6.4. kód.
```
max

+ 30 x1_2 +  6 x1_3 + 10 x2_4 + 15 x2_5 + 10 x2_7 +  2 x3_4
+  2 x3_6 +  5 x4_6 + 25 x5_7 + 10 x6_8 + 13 x7_8 =

subject to
csp1: + x1_2 + x1_3 = 1
csp2: - x1_2 + x2_4 + x2_5 + x2_7 = 0
csp3: - x1_3 + x3_4 + x3_6 = 0
csp4: - x2_4 - x3_4 + x4_6 = 0
csp5: - x2_5 + x5_7 = 0
csp6: - x3_6 - x4_6 + x6_8 = 0
csp7: - x2_7 - x5_7 + x7_8 = 0
csp8: - x6_8 - x7_8 = -1

end
```

A formalizációt nehéz közgazdaságilag interpretálni. Olyan, mintha a tevékenységek hossza költség helyett bevétel lenne, és szeretnénk 1 egységet elszállítani a legnagyobb bevétellel az 1-es pontból a 8-as pontba.

Ha a közgazdasági interpretáció nem is nagyon könnyű, a legrövidebb út és a kritikus út közötti hasonlóság jól látszik: a korlátok megegyeznek mind a két esetben, egyedül a célfüggvény különbözik. De talán az is jól látszik, hogy a kritikus út 'leghosszabb út' feladat.

A leghosszabb út azonban nem csak az LP modellekből látható, a feladatot úgy is lehet interpretálni, hogy ahhoz, hogy a projekt elkészüljön minden tevékenységet el kell végezni. Másképpen fogalmazva az összes utat be kell járni a gráfban. Így értelemszerűen a leghosszabb út, tehát a kritikus út fogja meghatározni a projekt befejezésének idejét.

A csomópontos formalizáció esetén az optimális megoldás:

```
Status:      OPTIMAL
Objective:   obj = 83 (MAXimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             1            73
     2 csp2         NS             0            43
     3 csp3         NS             0             7
     4 csp4         NS             0             5
     5 csp5         NS             0            28
     6 csp6         B              0
     7 csp7         NS             0             3
     8 csp8         NS            -1           -10

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 x1_2         B              1
     2 x1_3         NL             0           -60
     3 x2_4         NL             0           -28
     4 x2_5         B              1
     5 x2_7         NL             0           -30
     6 x3_4         B              0
     7 x3_6         NL             0            -5
     8 x4_6         B              0
     9 x5_7         B              1
    10 x6_8         B              0
    11 x7_8         B              1
```

Összességében ugyanazt a megoldást kaptuk: a kritikus út az 1→2→5→7→8 pontokon átvezető út, a projekt 'legrövidebb' időtartama pedig 83 nap.

6.5. példa. *Tekintsük a 6.1. példában megadott feladatot! Meg tudjuk-e adni a kritikus utat anélkül is, hogy felrajzolnánk a gráfot?*

Megoldás.

A válasz igenlő. Ekkor mindegyik tevékenység kezdetéhez bevezetünk egy döntési változót: `yA`, `yB`, stb ... Jelölje `yVeg` a projekt befejezésének időpontját.

A B tevékenységet csak akkor tudjuk elkezdeni, ha az A tevékenységet befejeztük. Tehát elő kell írni, hogy a B tevékenység elkezdésének ideje nem lehet előbb, mint az A tevékenység megkezdésének ideje plusz az A tevékenység hossza. Erre a tevékenységre tehát `yA+30<=yB`, amely korlátot át kell rendezni a `yA-yB<=-30` alakra. Ha több előfeltétel is van (pl. az I tevékenység esetén), akkor minden előfeltételhez külön korlát tartozik (konkrétan az I tevékenység esetén `yE-yI<=-10` és `yG-yI<=-2`). Továbbá a projekt akkor fejeződik be, ha minden tevékenység befejeződött. Tehát `yA-yVeg<=-30`, `yB-yVeg<=-15`, stb ... Egészen pontosan ezen utóbbi korlátokat elég lenne csak azokra a tevékenységekre felírni, amik nem valamelyik másik tevékenység előfeltétele. Jelen esetben elég lenne a J és K tevékenységeket felírni, de adott esetben nehezen állapítható meg, hogy melyek azok a tevékenységek, amik nem előfeltételei másik tevékenységnek. Ha az összes tevékenységre felírjuk, akkor (valószínűleg) redundanciát viszünk a rendszerbe, de ez az optimum értékét és magát a kritikus utat nem befolyásolja.

Az LP felírás tehát:

6.6. kód.
```
min
yVeg

subject to
kB_A: yA -yB <= -30
kC_B: yB -yC <= -15
kD_A: yA -yD <= -30
kE_A: yA -yE <= -30
kG_F: yF -yG <= - 6
kH_F: yF -yH <= - 6
kI_E: yE -yI <= -10
kI_G: yG -yI <= - 2
kJ_I: yI -yJ <= - 5
```

```
kJ_H: yH -yJ <= - 2
kK_C: yC -yK <= -25
kK_D: yD -yK <= -10

tA: yA - yVeg <= -30
tB: yB - yVeg <= -15
tC: yC - yVeg <= -25
tD: yD - yVeg <= -10
tE: yE - yVeg <= -10
tF: yF - yVeg <= - 6
tG: yG - yVeg <= - 2
tH: yH - yVeg <= - 2
tI: yI - yVeg <= - 5
tJ: yJ - yVeg <= -10
tK: yK - yVeg <= -13

end
```

Ebben az esetben az optimális megoldás:

```
Status:     OPTIMAL
Objective:  obj = 83 (MINimum)

   No.   Row name   St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 kB_A         NU           -30            -1
     2 kC_B         NU           -15            -1
     3 kD_A         B            -60
     4 kE_A         NU           -30         < eps
     5 kG_F         NU            -6         < eps
     6 kH_F         NU            -6         < eps
     7 kI_E         NU           -10         < eps
     8 kI_G         B            -34
     9 kJ_I         NU            -5         < eps
    10 kJ_H         B            -39
    11 kK_C         NU           -25            -1
    12 kK_D         NU           -10         < eps
    13 tA           B            -83
    14 tB           B            -53
    15 tC           B            -38
    16 tD           B            -23
    17 tE           B            -53
    18 tF           B            -83
    19 tG           B            -77
    20 tH           B            -77
    21 tI           B            -43
```

```
    22 tJ           B            -38
    23 tK           NU           -13            -1

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 yVeg         B             83
     2 yA           NL             0             1
     3 yB           B             30
     4 yC           B             45
     5 yD           B             60
     6 yE           B             30
     7 yF           NL             0         < eps
     8 yG           B              6
     9 yH           B              6
    10 yI           B             40
    11 yJ           B             45
    12 yK           B             70
```

Az eddigi ismereteink alapján már könnyen beazonosíthatjuk, hogy a projekt befejezésének legrövidebb időtartama 83 nap, a kritikus tevékenységek pedig az A, B, C és K tevékenységek együttese (itt -1 az árnyékár).

Megoldás #2.

A 6.6. kód esetén problémát jelenthet, hogy egy-egy korlát felírása 'adatbázis' műveletet igényel. Tehát amikor a `kB_A` korlátot írjuk fel akkor látjuk, hogy a B tevékenység előfeltétele az A tevékenység, de a korlát felírásához szükséges az A tevékenység hossza is, amely adat egy másik sorban található.

Bemutatunk egy olyan LP felírást is, amihez nem szükséges 'adatbázis' művelet, egy adott sorban megtalálható információk alapján fel lehet írni a modellt. Ehhez az szükséges, hogy ne a tevékenység megkezdésének időpontjához vezessünk be változókat, hanem a tevékenység befejezésének időpontjához. Jelölje tehát `vA`, `vB`, stb ... döntési változó az adott tevékenység befejezésének időpontját. Ekkor a korlátok úgy fognak kinézni, hogy a tevékenység befejezésének időpontjának nagyobbnak kell lennie, mint a tevékenység kezdete (az előfeltétel tevékenység befejezése) plusz az esemény hossza. Tehát a B tevékenység esetén `vA+15<=vB`, ami átrendezve, a `vA-vB<=-15` formában fog megjelenni az LP felírásban. További változás, hogy ezeket a korlátokat fel kell írni azokra a tevékenységekre is, aminek nincs előfeltétele. Tehát pl. az *A* tevékenység esetén: az *A* végének nagyobbnak kell lennie, mint a kezdetének és hosszának az összege. Mivel az *A* tevékenységen nincs előfeltétele, ezért a 0-adik időpontban pontban is elkezdődhet, tehát a korlát egyszerűen `vA>=30`. A korlátot inkább a `-vA<=-30` formában fogjuk felírni, mivel így jobban hasonlít a többi korlátra. A projekt akkor fejeződik be, ha minden tevékenység befejeződik, tehát mindegyik tevékenység befejezésének időpontjánál nagyobb a `vVeg` változó értéke.

6.7. kód.
```
min
vVeg

subject to
kA_0:    -vA <= -30
kB_A: vA -vB <= -15
kC_B: vB -vC <= -25
kD_A: vA -vD <= -10
kE_A: vA -vE <= -10
kF_0:    -vF <= - 6
kG_F: vF -vG <= - 2
kH_F: vF -vH <= - 2
kI_E: vE -vI <= - 5
kI_G: vG -vI <= - 5
kJ_I: vI -vJ <= -10
kJ_H: vH -vJ <= -10
kK_C: vC -vK <= -13
kK_D: vD -vK <= -13

tA: vA - vVeg <= 0
tB: vB - vVeg <= 0
tC: vC - vVeg <= 0
tD: vD - vVeg <= 0
tE: vE - vVeg <= 0
tF: vF - vVeg <= 0
tG: vG - vVeg <= 0
tH: vH - vVeg <= 0
tI: vI - vVeg <= 0
tJ: vJ - vVeg <= 0
tK: vK - vVeg <= 0

end
```

Az optimális megoldás:

```
Status:     OPTIMAL
Objective:  obj = 83 (MINimum)

   No.   Row name   St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 kA_0         NU           -30            -1
     2 kB_A         NU           -15            -1
     3 kC_B         NU           -25            -1
     4 kD_A         NU           -10         < eps
     5 kE_A         NU           -10         < eps
     6 kF_0         NU            -6         < eps
     7 kG_F         NU            -2         < eps
     8 kH_F         NU            -2         < eps
     9 kI_E         NU            -5         < eps
    10 kI_G         B            -37
    11 kJ_I         NU           -10         < eps
    12 kJ_H         B            -47
    13 kK_C         NU           -13            -1
    14 kK_D         B            -43
    15 tA           B            -53
    16 tB           B            -38
    17 tC           B            -13
    18 tD           B            -43
    19 tE           B            -43
    20 tF           B            -77
    21 tG           B            -75
    22 tH           B            -75
    23 tI           B            -38
    24 tJ           B            -28
    25 tK           NU             0            -1

   No. Column name  St   Activity
------ ------------ -- -------------
     1 vVeg         B             83
     2 vA           B             30
     3 vB           B             45
     4 vC           B             70
     5 vD           B             40
     6 vE           B             40
     7 vF           B              6
     8 vG           B              8
     9 vH           B              8
    10 vI           B             45
    11 vJ           B             55
```

```
12 vK                B                83
```

Természetesen megoldásként ugyanazt kaptunk, mint korábban: a projekt időtartama 83 nap, a kritikus út pedig az A, B, C és K tevékenységekből tevődik össze.

6.2. Projekt lerövidítése

Üzleti problémák során sokszor szembesülünk azzal, hogy az elsőre elgondolt változat valamiért nem megfelelő. Például ha egy új terméket szeretnénk bevezetni, akkor az időzítés nagyon fontos. Ha későn jövünk ki az új termékkel, akkor a versenytársak megelőznek minket, tehát sok esetben le kell rövidíteni valahogy a projekt időtartamát. Ez viszont költséggel járhat, például elrendelhetünk túlórát, de ezért az alkalmazottaknak fizetni kell, jellemzően többet, mint a főmunkaidőben végzett munkáért. Másik példa, az engedélyeztetési eljárás. Például útlevél kérelem esetén például lehetőségünk van arra, hogy gyorsabban hozzájussunk az útlevélhez, de ez is többletköltséggel jár.

A projekt időtartamának lerövidítésénél azonban figyelembe kell venni, hogy lehetnek olyan tevékenységek is, amelyek hosszát nem lehet lerövidíteni. Például egy építési munkálat során a beton kötési idejét nem lehet meggyorsítani.

6.8. példa. *Tekintsük ismét a 6.2. példában megadott feladatot, azzal a módosítással, hogy a projektet 62 nap alatt be akarjuk fejezni. Az alábbi táblázat mutatja, hogy melyik tevékenység mennyi idővel rövidíthető le, és ez hetenként mennyi többletköltséget jelent:*

Fázis neve	*Lehetséges rövidítés*	*Költség naponként*
Falazás	*2*	*50*
Gázszerelés	*8*	*70*
Fűtésszerelés	*19*	*30*
Villanyszerelés	*3*	*25*
Nyílászárók beépítése	*2*	*15*
Hirdetési piac feltérképezése	-	-
Fényképész kiválasztása	-	-
Nyomda kiválasztása	-	-
Fényképezés	*1*	*55*
Prospektus elkészítése	*5*	*120*
Festés	*8*	*110*

Adja meg hogyan lehetséges a legkisebb költséggel 62 napra redukálni a projekt időtartamát!

Megoldás.

A korábban bemutatott 6.3., 6.4., 6.6. és 6.7. kódok mindegyike használható lenne ennek a feladat megoldásának meghatározására, mivel az alkalmazott ötlet azonosan felhasználható mindegyikben. Azonban az LP feladat felírása a 6.7. kód módosítása esetén szemléletesebb. Ekkor pl. az 'A' tevékenység esetén a korlát: `kA_0: -vA <= -30`, ami azt mondja ki, hogy az 'A' tevékenység befejezésének ideje nem lehet hamarabb, mint az előzmény tevékenység befejezési ideje (ami történetesen a projekt kezdete) plusz a tevékenység hossza. Ebbe a korlátba kell 'beleszerkeszteni' a lehetséges rövidítést. Az 'A' tevékenység esetén a rövidítést jelöljük `rA` módon. Ekkor a korábban bemutatott korlát a `kA_0: -vA-rA <= -30` formára változik. Emellett ki kell még kötni, hogy az `rA` változó értéke maximum 2 lehet (maximum 2 nappal tudjuk lerövidíteni a tevékenység hosszát): `rA: rA <= 2`. Hasonlóan az összes többi tevékenységesetén is el kell végezni ezt az átalakítást, a célfüggvény pedig a rövidítés összköltsége lesz, amit minimalizálunk. Ezen kívül meg kell adni, hogy a projekt hossza ne legyen több, mint 62 hét, tehát `vVeg <= 62` korlátot is hozzá kell adni a modellhez.

6.9. kód.

```
min
+  50 rA +  70 rB +  30 rC +  25 rD +  15 rE +  55 rI
```

```
+ 120 rJ + 110 rK

subject to
vVeg <= 62

kA_0:     -vA -rA <= -30
kB_A: vA -vB -rB <= -15
kC_B: vB -vC -rC <= -25
kD_A: vA -vD -rD <= -10
kE_A: vA -vE -rE <= -10
kF_0:     -vF     <= - 6
kG_F: vF -vG      <= - 2
kH_F: vF -vH      <= - 2
kI_E: vE -vI -rI <= - 5
kI_G: vG -vI -rI <= - 5
kJ_I: vI -vJ -rJ <= -10
kJ_H: vH -vJ -rJ <= -10
kK_C: vC -vK -rK <= -13
kK_D: vD -vK -rK <= -13

rA: rA <=  2
rB: rB <=  8
rC: rC <= 19
rD: rD <=  3
rE: rE <=  2
rI: rI <=  1
rJ: rJ <=  5
rK: rK <=  8


tA: vA - vVeg <= 0
tB: vB - vVeg <= 0
tC: vC - vVeg <= 0
tD: vD - vVeg <= 0
tE: vE - vVeg <= 0
tF: vF - vVeg <= 0
tG: vG - vVeg <= 0
tH: vH - vVeg <= 0
tI: vI - vVeg <= 0
tJ: vJ - vVeg <= 0
tK: vK - vVeg <= 0

end
```

Az optimális megoldás:

```
Status:     OPTIMAL
Objective:  obj = 670 (MINimum)

Column name  St   Activity      Marginal
------------ -- ------------- -------------
rA           B              2
rB           B              0            20
rC           B             19
rD           NL             0            25
rE           B              0            15
rI           NL             0            55
rJ           NL             0           120
rK           B              0            60
vVeg         B             62
vA           B             28
vB           B             43
vC           B             49
vD           B             38
vE           B             38
vF           B              6
vG           B              8
vH           B              8
vI           B             43
vJ           B             53
vK           B             62
```

A szükséges rövidítést 670 költséggel tudjuk megvalósítani: az ‘A’ tevékenységet rövidítjük 2 nappal és a ‘C’ fázist pedig 19 nappal. Észrevehetjük, hogy most elég volt a kritikus út menti tevékenységeket rövidíteni, ez a feladat adott esetben LP modell nélkül is viszonylag könnyen megkapható lenne. De amennyiben a projekt időtartamát tovább rövidítjük 10 nappal, akkor a következő eredményt kapjuk:

```
Status:     OPTIMAL
Objective:  obj = 1465 (MINimum)

Column name  St   Activity  Marginal
------------ -- ------------- -------------
rA           B              2
rB           B              8
rC           B             19
rD           NL             0            25
rE           B              1
rI           NL             0            40
```

```
rJ           NL            0           105
rK           B             2
vVeg         B            52
vA           B            28
vB           B            35
vC           B            41
vD           B            38
vE           B            37
vF           B             6
vG           B             8
vH           B             8
vI           B            42
vJ           B            52
vK           B            52
```

Egyrészt az összköltség megemelkedett (1465-re), ami nyilván logikus. Ami viszont számunkra fontos, hogy az 'E' tevékenység hosszát is le kell rövidíteni egységnyivel, ami viszont már nem a kritikus út része; egészen pontosan az eredeti feladatban nem volt a kritikus út része, de a rövidítés után ez a tevékenység is a kritikus út részévé válik. Ebben az esetben nem lesz egyértelmű a kritikus út, mivel az 'A' ←'B'←'C'←'K' és az 'A' ←'E'←'I'←'J' utak is kritikusak lesznek.

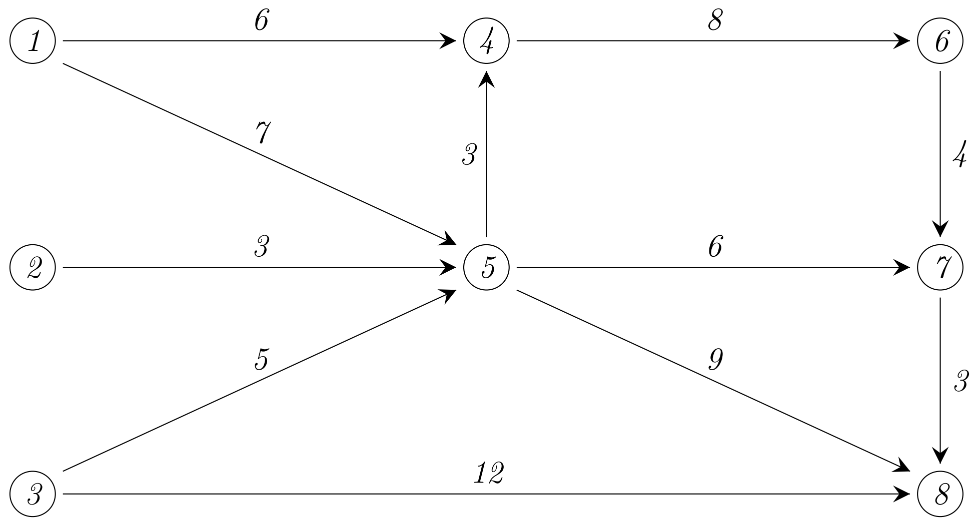
6.3. Gyakorló feladatok

6.3.1. Egyszerűbb feladatok

6.10. példa. *Tekintsük a 6.8. példát. A projekt lerövidítését felírtuk a 6.7. kód módosításával (6.9. kód). Adja meg a lerövidítések LP felírásait, a 6.3. kód, 6.4. kód ás a 6.6. kód módosításával is.*

Megoldás a 311. oldalon.

6.11. példa. *Az alábbi gráf egy vasúttársaság sematikus hálózatát reprezentálja. A nyilak reprezentálnak egy-egy járatot, a hozzátartozó számok pedig a becsült menetidőket.*



A vasúttársaság bizonyos esetekben átszállási lehetőséget szeretne biztosítani. Ez azt jelenti hogy egy adott járat csak akkor tud elindulni, ha az összes csatlakozás beérkezett. Adjon meg egy menetrendet (vonatok legkorábbi lehetséges indulását) úgy, hogy minden csúcsban van átszállási lehetőség. Melyik vonatok késése kritikus a vasúttársaság számára?

Megoldás a 313. oldalon.

6.3.2. Nehezebb feladatok

6.12. példa. *Adja meg, hogy a 6.2. példában milyen határok között változhat a B, D és H tevékenység időszükséglete úgy, hogy ne változzon a kritikus út hossza!*

Megoldás a 316. oldalon.

6.13. példa. *Egy szakács versenyre készül. Számára különösen fontos az időzítés és ha valamelyik feladattal megcsúszik, akkor nem lesz képes időben bemutatni a főztjét. A szakács az alábbi fázisokat tudja elkülöníteni:*

Fázis neve	*Fázis azonosítója*	*Szükséges idő*	*Előfeltételek*
Zöldséghámozás	*A*	*15*	-
Rizsfőzés	*B*	*20*	-
Hús előkészítése	*C*	*10*	-
Hús sütése	*D*	*30*	*A, C*
Zöldség párolás	*E*	*20*	*A*
Rizi-bizi	*F*	*10*	*B, E*
Mártás	*G*	*15*	*D, E*
Fehérje felverése	*H*	*5*	-
Rizsfelfújt	*I*	*30*	*B, H*
Tálalás	*J*	*10*	*F, G, I*

Adjunk meg egy gráfot, ami az elvégzendő fázisokat reprezentálja!

Megoldás a 318. oldalon.

6.14. példa. *Tekintsük a 6.3. kódban megadott LP feladatot. Írjuk fel ennek a duál feladatát és vezessük le ebből a csomópontos formalizációt!*

Megoldás a 319. oldalon.

6.15. példa. *A következő táblázatban egy projekt elvégzéséhez tartozó tevékenységek, és azok időigénye látható:*

Tevékenység	*Szükséges idő*	*Előfeltételek*
A	*3*	-
B	*29*	-
C	*13*	-
D	*33*	*A*
E	*12*	*A*
F	*6*	*B,D*
G	*2*	*B,D*
H	*25*	*E,F*
I	*7*	*C*

Félidő környékén tartunk egy időegységnyi szünetet, hogy a dolgozók meg a gépek pihenjenek. Semelyik tevékenységet nem szabad félbeszakítani. Mikor legyen ez a szünet, ha azt szeretnénk, hogy a projekt első fele pontosan olyan hosszú legyen, mint a második, és a végső befejezési idő minél hamarabb legyen? Melyik tevékenységek esnek az első félidőbe és melyikek a másodikba?

Megoldás a 320. oldalon.

6.16. példa. *A következő táblázatban egy összeszerelési projekt elvégzéséhez tartozó tevékenységek, és azok időigénye látható:*

Tevékenység	*Szükséges idő*	*Előfeltételek*
A	10	-
B	8	-
C	7	A
D	5	B
E	8	B
F	9	A,
G	4	C,D
H	6	C,D
I	3	H,E
J	6	F,G

A tevékenységek közül a C, D, E, F, I és J tevékenység végzéséhez speciális gépek szükségesek. A C és D tevékenységeket csak az első típusú gépen lehet végezni, az E és F típusút csak második típusú gépen, a I és J tevékenységet pedig harmadik típusú gépen. Ezekből a gépekből csak 1-1 áll rendelkezésre, és egy adott gépet egyszerre csak 1 tevékenységhez lehet használni, de nincs külön megkötve, hogy melyik tevékenységet kell előbb elvégezni.

Adjon meg egy vegyes egészértékű LP-t, amellyel meghatározható a projekt legkorábbi befejezésének időpontja!

Megoldás a 323. oldalon.

7. fejezet

Hozzárendelési feladat

Hozzárendelési feladatok esetén szintén adott egy költségmátrix, ebben az esetben jellemzően négyzetes mátrixról van szó. A sorok például munkafeladatokat jelentenek, az oszlopok pedig munkásokat. A feladat célja, hogy a munkafeladatokat hozzá tudjuk rendelni a munkásokhoz úgy hogy az összköltség minimális legyen.

7.1. példa. *Adott az alábbi költségmátrix:*

$$\begin{array}{cccc} 23 & 31 & 13 & 27 \\ 11 & 39 & 16 & 60 \\ 37 & 50 & 37 & 54 \\ 60 & 30 & 24 & 33 \end{array}$$

Adjuk meg az összes lehetséges hozzárendelést!

Megoldás.

A feladatok munkásokhoz rendelése azt jelenti, hogy megmondjuk az adott feladatra, hogy melyik munkás végzi el. Ezt legegyszerűbben úgy tudjuk jelölni, hogy bekeretezzük azokat a számokat, ahol a hozzárendelések vannak. Pl.:

$$\begin{array}{cccc} \boxed{23} & 31 & 13 & 27 \\ 11 & \boxed{39} & 16 & 60 \\ 37 & 50 & \boxed{37} & 54 \\ 60 & 30 & 24 & \boxed{33} \end{array}$$

Ebben az esetben az első feladatot az első munkás végzi, a másodikat a második munkás, a harmadik feladatot a harmadik munkás, a negyedik feladatot pedig a negyedik munkás. Ekkor az összköltség: 23+39+37+33=132

Természetesen többféle hozzárendelés is lehetséges, csak az a lényeg, hogy minden sorban és minden oszlopban pontosan 1 bekeretezett szám legyen. Könnyen láthatjuk, hogy egy nxn-es mátrix esetén $n!$ hozzárendelés lehetséges: a mátrix első sorában n számot választhatok; a másodikban azt az oszlopot nem választhatom, amit az első sorban választottam. A harmadik sorban nem választhatom azokat az oszlopokat, amit az első vagy második sorban választottam, így $n-2$ választási lehetőség marad. Folytatva az eljárást, az utolsó sorban már csak egy választási lehetőség marad. Összesen tehát $n(n-1)\cdots 1 = n!$ választási lehetőség. Jelölje pl.: F1M3 azt, ha az első feladatot a 3. munkás végzi. Ekkor egy 4x4-es mátrix esetén az összes hozzárendelés:

F1M1 F2M2 F3M3 F4M4	F1M1 F2M2 F4M3 F3M4	F1M1 F3M2 F2M3 F4M4	F1M1 F3M2 F4M3 F2M4	F1M1 F4M2 F2M3 F3M4	F1M1 F4M2 F3M3 F2M4
F2M1 F1M2 F3M3 F4M4	F2M1 F1M2 F4M3 F3M4	F2M1 F3M2 F1M3 F4M4	F2M1 F3M2 F4M3 F1M4	F2M1 F4M2 F1M3 F3M4	F2M1 F4M2 F3M3 F1M4
F3M1 F1M2 F2M3 F4M4	F3M1 F1M2 F4M3 F2M4	F3M1 F2M2 F1M3 F4M4	F3M1 F2M2 F4M3 F1M4	F3M1 F4M2 F1M3 F2M4	F3M1 F4M2 F2M3 F1M4
F4M1 F1M2 F2M3 F3M4	F4M1 F1M2 F3M3 F2M4	F4M1 F2M2 F1M3 F3M4	F4M1 F2M2 F3M3 F1M4	F4M1 F3M2 F1M3 F2M4	F4M1 F3M2 F2M3 F1M4

A mátrix méretének növekedésével az összes hozzárendelés száma exponenciálisan növekszik. Nagyméretű feladatok esetén már szinte lehetetlen az összes hozzárendelést számba venni.

7.2. példa. *Tekintsük a 7.1. példát. Adjuk meg a minimális költségű hozzárendelést!*

Megoldás.

Természetesen a feladatot úgy is meg lehetne oldani, hogy sorra vesszük az összes lehetséges hozzárendelést, és mindegyik esetben kiszámoljuk a hozzárendelés költségét, majd ezek közül kiválasztjuk a minimálisat. Mint láttuk nagyméretű feladatok esetén ez problematikus. Szerencsére erre nincs is szükség.

A hozzárendelési feladatot tekinthetjük speciális szállítási feladatnak is. A szállítási mennyiségek legyenek minden sorban és minden oszlopban egységnyiek. Mielőtt ezt megtennénk, érdemes feltenni a kérdést, hogy miért foglalkozunk külön a hozzárendelési feladattal, ha az lényegében szállítási feladat? A válasz az, hogy azért, mert mivel speciális struktúrája van a feladatnak létezik olyan algoritmus (az ún. magyar módszer), ami hatékonyabban meg tudja oldani a hozzárendelési feladatot, mint a szimplex módszer. Ez az előny azonban csak nagyon nagyméretű feladatok esetén jelentkezik.

Írjuk fel így a hozzárendelési feladatot.

7.3. kód.

```
min
ktg:
+23x1_1 +31x1_2 +13x1_3 +27x1_4
+11x2_1 +39x2_2 +16x2_3 +60x2_4
+37x3_1 +50x3_2 +37x3_3 +54x3_4
+60x4_1 +30x4_2 +24x4_3 +33x4_4

subject to
F1: +x1_1 +x1_2 +x1_3 +x1_4 = 1
F2: +x2_1 +x2_2 +x2_3 +x2_4 = 1
F3: +x3_1 +x3_2 +x3_3 +x3_4 = 1
F4: +x4_1 +x4_2 +x4_3 +x4_4 = 1

M1: +x1_1 +x2_1 +x3_1 +x4_1 = 1
M2: +x1_2 +x2_2 +x3_2 +x4_2 = 1
M3: +x1_3 +x2_3 +x3_3 +x4_3 = 1
M4: +x1_4 +x2_4 +x3_4 +x4_4 = 1

end
```

Fontos megjegyzés, hogy a 7.3. kód esetén a változókra nincsenek egészértékű megkötések. Első ránézésre erre szükség lenne, hiszen ha `x1_1` változó értéke 0.2, akkor nem tudjuk értelmezni a feladatot. De ebben az esetben az együttható (A) mátrix ún. TU (teljesen unimoduláris) mátrix, ami biztosítja, hogy ha a kapacitásvektor egészértékű, akkor a bázismegoldások is egészértékűek lesznek. A mi esetünkben is ez a helyzet, hiszen a kapacitásvektor kizárólag 1-esből áll, és az a A mátrix TU mátrix.Természetesen ez igaz a szállítási feladat esetében is. Tehát amennyiben a keresletek és a kínálatok egészértékűek, a változókra nem kell az egészértékűséget külön kikötni.

A feladat optimális megoldása:

```
Objective:  ktg = 105 (MINimum)

No.   Row name   St   Act. L. bound U. bound Marg.
--- ------------ -- ------ -------- -------- -----
  1 F1           NS      1        1        =    27
  2 F2           NS      1        1        =    32
  3 F3           NS      1        1        =    53
  4 F4           NS      1        1        =    33
  5 M1           NS      1        1        =   -21
  6 M2           NS      1        1        =    -3
  7 M3           NS      1        1        =   -16
  8 M4           B       1        1        =

No. Column name  St   Act. L. bound U. bound Marg.
--- ------------ -- ------ -------- -------- -----
  1 x1_1         NL      0        0             17
  2 x1_2         NL      0        0              7
  3 x1_3         NL      0        0              2
  4 x1_4         B       1        0
  5 x2_1         B       1        0
  6 x2_2         NL      0        0             10
  7 x2_3         B       0        0
  8 x2_4         NL      0        0             28
  9 x3_1         NL      0        0              5
 10 x3_2         B       0        0
 11 x3_3         B       1        0
 12 x3_4         NL      0        0              1
 13 x4_1         NL      0        0             48
 14 x4_2         B       1        0
 15 x4_3         NL      0        0              7
 16 x4_4         B       0        0
```

Az x1_4, x2_1, x3_3 és x4_2 változók értéke 1, tehát az optimális hozzárendelés: F1M4, F2M1, F3M3 és F4M2. Ezen hozzárendelés költsége: 27+11+37+30=105.

A hozzárendelési feladatnál lényegileg ugyanazok a speciális esetek, mint a szállítási feladatnál (nyereségek maximalizálása, büntetőtarifa, tiltótarifa stb...), amelyeket most nem ismételünk meg. A kitűzött feladatok között szerepelnek ilyen specialitások.

7.1. Gyakorló feladatok

7.1.1. Egyszerűbb feladatok

7.4. példa. *7 munkással szeretnénk elvégeztetni 7 munkagépen 7 alkatrész legyártását. A következő táblázatban a munkásoknak az adott munkagépeken a feladat elvégzéséhez szükséges várható ideje látható, percre kerekítve:*

	1	2	3	4	5	6	7
1	64	83	67	55	60	41	35
2	32	44	31	60	99	38	61
3	56	92	97	68	75	33	85
4	40	46	60	39	79	97	58
5	59	61	91	54	66	31	83
6	45	36	72	66	51	68	99
7	52	75	53	87	96	99	79

Azt szeretnénk, hogy a gépek a lehető legkevesebb ideig legyenek használva. Mekkora lesz az összesített gépműködési idő, és várhatóan mennyi idő szükséges ahhoz, hogy mindegyik alkatrész elkészüljön?

Megoldás a 325. oldalon.

7.5. példa. *Egy taxitársaságtól ugyanabban az időpontban 6 autót rendelnek, a város különböző pontjain. A taxitársaságnak jelenleg 8 szabad taxija van. A következő táblázatban láthatóak a taxik az utasokhoz várható érkezési idejei:*

	U1	U2	U3	U4	U5	U6
T1	5	12	20	8	11	4
T2	7	4	5	2	21	4
T3	5	14	2	2	21	11
T4	9	14	14	8	4	3
T5	8	8	4	12	7	10
T6	11	5	10	5	6	7
T7	3	3	20	9	21	8
T8	4	9	16	9	13	7

A taxitársaság szeretné minimalizálni az összes utas taxira várakozásának idejét. Melyik taxis, melyik utashoz induljon, és melyik két taxis marad utas nélkül? Melyik utasoknak lenne közelebbi taxi az optimálisan hozzájuk rendelt taxinál?

Megoldás a 327. oldalon.

7.6. példa. *Egy egyetemi kurzuson, a hallgatóknak csoportokba rendeződve feladatokat kell elvégezniük. Összesen 7 csoport van, és 7 különböző témájú feladatot közül választhatnak. Minden csoport 1 témát választhat, és 1 témát kizárólag egy csoport kaphatja feladatként. A csoportok egy 3 elemű skálán rangsorolták a számukra leginkább kedvező témákat (1-es jelöli, amit a legjobban szeretnének).*

	t1	t2	t3	t4	t5	t6	t7
cs1		2	1				3
cs2	3	1	2				
cs3			3		1	2	
cs4	2	3			1		
cs5	1			2			3
cs6		2	1		3		
cs7	3					1	2

A célunk, hogy úgy rendeljük a témákat a csoportokhoz, hogy a hallgatók összessége számára legyen megfelelő a hozzárendelés, mégpedig úgy, hogy a csoportok csak olyan témát kaphatnak, amely számunkra benne van a választott első háromban.

Megoldás a 328. oldalon.

7.1.2. Nehezebb feladatok

7.7. példa. *9 barát elhatározza, hogy váltóban elindulnak a hétvégi terepfutóversenyen. Az alábbi táblázat mutatja, hogy melyik szakaszt hány perc alatt tudják lefutni. Kilencük közül Anna fut a legjobban. Annyira jól fut hogy akár két különböző szakaszt is képes lefutni (a szabályok ezt megengedik, és kocsival át is tudják szállítani a másik indulási pontra). Viszont abban az esetben ha kétszer is futnia kell, a második futására már egy kicsit fáradtabb, és csak 4 perccel lassabban tudja lefutni, mint pihenten. Hogy állítsák össze a váltót, ha a lehető leggyorsabban akarják teljesíteni a távot?*

	1.sz.	*2.sz.*	*3.sz.*	*4.sz.*	*5.sz.*	*6.sz.*	*7.sz.*	*8.sz.*	*9.sz.*
Anna	*11*	*13*	*13*	*16*	*14*	*12*	*12*	*11*	*13*
Béla	*18*	*16*	*19*	*25*	*23*	*19*	*20*	*12*	*14*
Cili	*16*	*24*	*14*	*15*	*18*	*16*	*23*	*17*	*14*
Dávid	*17*	*23*	*17*	*16*	*18*	*25*	*14*	*13*	*17*
Eszter	*17*	*17*	*12*	*12*	*18*	*13*	*17*	*24*	*18*
Feri	*18*	*16*	*17*	*22*	*13*	*14*	*15*	*16*	*19*
Géza	*19*	*15*	*15*	*20*	*14*	*15*	*14*	*21*	*18*
Hedvig	*22*	*21*	*14*	*15*	*15*	*19*	*13*	*13*	*18*
Imre	*20*	*24*	*14*	*24*	*18*	*21*	*21*	*18*	*16*

Megoldás a 330. oldalon.

7.8. példa. *Egy vállalat egy három emeletes épület 7 lakását árulja. Minden emeleten 2 lakás van, kivéve a legfelsőn, ahol csak 1. Az ingatlanokra 8 vevő érdeklődik, amelyek közül 2 üzletet akar kialakítani a lakásból, 1 pedig irodát szeretne az épületben nyitni, a többi pedig a lakásokban lakni szeretne. Az üzletek csak a földszinten lehetnek. Az a vevő, aki irodát akar nyitni a házban, két lakást vásárolna, viszont csak úgy, hogy a lakások ugyanazon az emeleten lehetnek. A táblázatban láthatóak a vállalat rezervációs árai, (az összeg, ami alatt egyetlen lakást sem hajlandó eladni), és a vevők lakásokra tett ajánlatai, millió forintban:*

lakás	*Rezerv. ár*	*lakó1*	*lakó2*	*lakó3*	*lakó4*	*lakó5*	*üzlet1*	*üzlet2*	*iroda*
0.A	*35*	*40*	*44*				*44*	*46*	*45*
0.B	*33*	*40*	*44*				*43*	*44*	*45*
1.A	*40*	*42*	*46*	*50*	*50*				*52*
1.B	*38*	*42*	*46*	*50*	*50*				*49*
2.A	*46*	*42*	*48*	*50*	*52*	*63*			*53*
2.B	*44*	*42*	*48*	*50*	*52*	*63*			*50*
3.A	*60*	*42*	*48*	*50*	*65*	*70*			

A vállalatnak nem kell minden lakást eladnia, ingatlanpiaci becslések alapján, ha nem adja el bármely lakást, akkor annak a rezervációs ára felett várhatóan 2 millió forintos áron tudja majd eladni. Melyik lakást melyik vevőnek adja el vállalat, ha a cél, hogy az ajánlati ár, és a rezervációs ár közötti összeg legyen maximális?

Megoldás a 334. oldalon.

7.9. példa. *Egy alkatrészek összeszerelésével foglalkozó vállalatnak kettő futószalagja, és kilenc munkása van. Egy futószalagon gyártott alkatrész elkészítése három szakaszból áll, amihez a munkások szaktudása szükséges, tehát a gyártás adott szakaszát csak néhány munkás képes elvégezni. A következő táblázatban látható, hogy a munkás melyik munkafolyamatot képes elvégezni:*

szakasz	*m1*	*m2*	*m3*	*m4*	*m5*	*m6*	*m7*	*m8*	*m9*
1	*1*	*1*	*1*				*1*		
2	*1*			*1*	*1*	*1*		*1*	
3					*1*	*1*	*1*	*1*	*1*

A vállalat a munkások tapasztalata alapján akarja meghatározni, hogy melyik munkás melyik futószalag mellett, és melyik szakaszban dolgozzon. A következő táblázatban látható, hogy a munkások eddig mennyi alkatrész gyártásában vettek részt:

	m1	*m2*	*m3*	*m4*	*m5*	*m6*	*m7*	*m8*	*m9*
legyártott alkatrész	*117*	*121*	*95*	*73*	*140*	*110*	*75*	*180*	*130*

A futószalagon készített alkatrészekből a vállalat három terméket (A,B,C) gyárt. Egy ilyen alkatrész összeszereléséhez 1 munkás szükséges, és a hatékonyság meghatározásához a vállalat egy 100 pontos rangsorban pontozta a munkásokat. Ez a következő táblázatban látható:

	m1	*m2*	*m3*	*m4*	*m5*	*m6*	*m7*	*m8*	*m9*
A	*36*	*61*	*15*	*28*	*36*	*13*	*98*	*53*	*79*
B	*97*	*96*	*85*	*30*	*85*	*12*	*49*	*69*	*41*
C	*69*	*66*	*85*	*19*	*66*	*11*	*18*	*68*	*28*

A vállalat célja, hogy úgy ossza be a munkásokat, hogy az A,B,C termékek gyártásához rendelt munkások hatékonysága maximális legyen. Viszont a két futószalag mellé úgy kell a munkásokat beosztani, hogy az általuk eddig összesen legyártott alkatrészek száma legalább 300 legyen.

Megoldás a 337. oldalon.

8. fejezet

Stabil párosítások

A hozzárendelési feladat egyfajta bővítése a stabil párosítási feladat. A stabil párosítások legismertebb kiinduló példája a stabil házasság probléma: adott n férfi és n nő. Mind a férfiak, mind a nők megadják a másik nem iránti preferenciájukat. Ezeket a preferenciákat két mátrixba rendezzük, az első mátrix a férfiak preferenciáit mutatja, a másik pedig a nőkét. A preferenciák megadásánál több lehetséges út is lehetne, mi ebben a jegyzetben a következő módon járunk el: a férfiak sorokban szerepelnek, a nők oszlopokban; tehát a férfiak mátrixa esetén a az i. sor azt mutatja meg, hogy az i. férfi hogyan rangsorolja a nőket. A nők mátrixában a j. oszlop pedig azt adja meg, hogy a a j. nő hogyan rangsorolja a férfiakat. Példaként tekintsük a következő preferenciákat:

	$F1$	$F2$	$F3$	$F4$	$F5$
$M1$	1	5	4	2	3
$M2$	2	4	1	3	5
$M3$	2	1	3	5	4
$M4$	5	3	2	4	1
$M5$	4	3	5	1	2

és

	$F1$	$F2$	$F3$	$F4$	$F5$
$M1$	5	1	2	3	3
$M2$	2	4	5	4	1
$M3$	3	5	4	1	4
$M4$	1	3	3	2	5
$M5$	4	2	1	5	2

Ebben a konkrét esetben az első férfi legjobban az első nőt kedveli, utána a negyediket, az ötödiket, majd a harmadikat végül pedig a másodikat.

Az első nő legjobban a negyedik férfit kedveli, utána a másodikat, a harmadikat, az ötödiket és legkevésbé pedig az elsőt.

Stabil párosítási feladatok esetén is egy hozzárendelést kell megadni a férfiak és a nők között, viszont teljesülnie kell a stabilitásnak, azaz a hozzárendelésben nincs ún. blokkoló pár. A blokkoló párnak olyan férfi-nő párt hívunk, akik egymást jobban szeretnék párnak, mint azt, aki a jelenlegi párosításban éppen hozzájuk van rendelve.

A mátrixokban a sorokat M betűvel indexeljük (M1, M2, ... , M=male), az oszlopokat pedig F betűvel (F1, F2, ..., F=female). Egy hozzárendelést most is az M és F betűkkel tudunk megadni, tehát pl. a diagonális elemek alkotta hozzárendelés: M1F1, M2F2, M3F3, M4F4 és M5F5.

$$\begin{pmatrix} \boxed{1} & 5 & 4 & 2 & 3 \\ 2 & \boxed{4} & 1 & 3 & 5 \\ 2 & 1 & \boxed{3} & 5 & 4 \\ 5 & 3 & 2 & \boxed{4} & 1 \\ 4 & 3 & 5 & 1 & \boxed{2} \end{pmatrix} \quad \text{és} \quad \begin{pmatrix} \boxed{5} & 1 & 2 & 3 & 3 \\ 2 & \boxed{4} & 5 & 4 & 1 \\ 3 & 5 & \boxed{4} & 1 & 4 \\ 1 & 3 & 3 & \boxed{2} & 5 \\ 4 & 2 & 1 & 5 & \boxed{2} \end{pmatrix}$$

8.1. példa. *Mutassa meg, hogy a diagonális elemek alkotta párosítás nem stabil!*

Megoldás.

A megoldásban egyedül az M2F1 párra mutatjuk meg, hogy blokkol. A második férfi a a második nőt kapná, ebben a hozzárendelésben, viszont neki az első nő jobban tetszene (a második nő a negyedik helyen áll a második férfi rangsorában, az első a második helyen). Az első nő az első férfit kapná, de neki a második férfi jobban tetszene (az első férfi az ötödik helyen áll az első nő rangsorában, a második férfi a második helyen). Tehát mind a ketten hajlandóak lennének elcserélni a jelenlegi párjukat egymásra.

Nézzünk meg egy másik esetet is. Ebben a példában az M1F2 pár nem blokkoló pár: a második nőnek jobban tetszene az első férfi, de az első férfinak nem tetszik jobban a második nő, mint akihez hozzá van rendelve.

A stabil párosítási feladatokat az ún. Gale-Shapley (más néven késleltetett elfogadási) algoritmussal lehet hatékonyan megoldani. Ez az algoritmus viszonylag gyorsan (polinomiális időben) talál stabil megoldást, ugyanakkor ezzel az algoritmussal csak két nevezetes párosítást lehet megtalálni, egy férfi-optimális, illetve egy nő-optimális párosítást (amik akár megegyezhetnek). A férfi-optimális párosítás esetén a férfiak a lehető legjobban járnak, tehát nincs egyetlen másik stabil párosítás sem, ahol valamelyik férfi jobban járna, mint a férfi-optimális megoldás esetén. Analóg módon a nő-optimális megoldás esetén a nők járnak a lehető legjobban.

A férfi-optimális és nő-optimális megoldás esetén jellemzően valaki nagyon rosszul jár. A férfi- és nő-optimális megoldásokon kívül lehetnek még egyéb stabil párosítások, ami kiegyenlítetettebbek a férfi-nő viszonylatban.

A stabil párosítás (vagy más néven stabil házasság) feladat kicsit erőltetett, módosítások nélkül kevés gyakorlati alkalmazása van. Egyfajta módosítása a feladatnak, ha az egyik típusú szereplőhöz egyszerre több másik típusút rendelnénk hozzá. Ilyen például az (egyetemi) felvételi rendszer, ahol a férfiak és nők helyett szakok és diákok vannak. Ebben az esetben a szakok jellemzően nem egy embert választanak, a diákokat pontszámok alapján rangsorolják, majd előre meghatározott számú jelentkezőt vesznek fel.

A felvételi rendszereknél könnyen előfordulhat, hogy két diáknak ugyanannyi pontja van, viszont mind a két diákot nem tudja felvenni az intézmény, mert azzal túljelentkezés lenne az adott szakon. Kérdés, hogy mit lehet kezdeni az ilyen szituációkkal.

Szintén gyakori, hogy egy szak csak akkor indul, ha elegendő jelentkezőt tudunk oda allokálni. Ezek a specialitások nehézzé vagy lehetetlenné teszik, hogy a feladat megoldására a Gale-Shapley algoritmust használjuk, viszont LP feladatként elképzelhető, hogy a feladat megoldható.

8.2. példa. *Adjunk megy egy LP felírást a stabil házasság problémára!*

Megoldás.

Első megközelítésben a férfiak és nők száma megegyezik, ezt jelöljük n-el. Az LP felírás esetén jelölje x_{ij} döntési változó, ha az i férfi a j nőhöz van rendelve. Minden férfi csak egy nőhöz lehet rendelve, ami fordítva is igaz. Erre a hozzárendelési feladatból ismert korlátokat kell használni:

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \text{és} \qquad \sum_{j=1}^{n} x_{ij} = 1$$

Azonban nem minden hozzárendelés lesz stabil, ezért ki kell válogatni a lehetséges hozzárendelések közül a stabilokat. Vizsgáljunk meg egy konkrét párt, amit az x_{ij} változó fejez ki. Azt akarjuk elérni, hogy ez a pár ne legyen blokkoló pár. Azaz, ha i férfihez nem rendeltünk olyan nőt, aki jobban tetszik neki mint j nő; illetve j nőhöz nem rendeltünk olyan férfit aki jobban tetszik neki, mint i férfi, akkor i férfit és j nőt egymáshoz kell rendelni. Matematikailag megfogalmazva:

$$\sum_{k:r_k^i<r_j^i}^{n} x_{ik} + \sum_{\ell:s_\ell^j<s_i^j}^{n} x_{\ell j} + x_{ij} \geq 1 \ ,$$

ahol r_j^i jelöli j nő rangsorszámát i férfi rangsorában. Hasonlóan s_i^j jelöli i férfi rangsorszámát j nő rangsorában. Esetünkben pl.: $r_1^1 = 1$; $r_2^1 = 5$; $r_3^1 = 4$ stb ... Hasonlóan $s_1^1 = 5$; $s_2^1 = 2$; $s_3^1 = 3$ stb ...
Természetesen minden x_{ij} változóhoz tartozik egy ilyen stabilitási korlát.

A célfüggvényt a célnak megfelelően tetszőlegesen alakíthatjuk. Amennyiben

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

kifejezést minimalizáljuk megkapjuk a férfi-optimális megoldást, míg a

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

kifejezés minimalizálásával a nő-optimális megoldást kapjuk.

Nézzük most a konkrét LP felírást a férfi-optimális feladatra!

8.3. kód.
```
min
ffi:
+1x1_1+5x1_2+4x1_3+2x1_4+3x1_5
+2x2_1+4x2_2+1x2_3+3x2_4+5x2_5
```

```
+2x3_1+1x3_2+3x3_3+5x3_4+4x3_5
+5x4_1+3x4_2+2x4_3+4x4_4+1x4_5
+4x5_1+3x5_2+5x5_3+1x5_4+2x5_5

subject to
+x1_1+x1_2+x1_3+x1_4+x1_5=1
+x2_1+x2_2+x2_3+x2_4+x2_5=1
+x3_1+x3_2+x3_3+x3_4+x3_5=1
+x4_1+x4_2+x4_3+x4_4+x4_5=1
+x5_1+x5_2+x5_3+x5_4+x5_5=1

+x1_1+x2_1+x3_1+x4_1+x5_1=1
+x1_2+x2_2+x3_2+x4_2+x5_2=1
+x1_3+x2_3+x3_3+x4_3+x5_3=1
+x1_4+x2_4+x3_4+x4_4+x5_4=1
+x1_5+x2_5+x3_5+x4_5+x5_5=1

+x1_1                              +x2_1+x3_1+x4_1+x5_1>=1
+x1_2+x1_1     +x1_3+x1_4+x1_5                         >=1
+x1_3+x1_1          +x1_4+x1_5                    +x5_3>=1
+x1_4+x1_1                                   +x3_4+x4_4     >=1
+x1_5+x1_1          +x1_4          +x2_5          +x5_5>=1
+x2_1          +x2_3                         +x4_1     >=1
+x2_2+x2_1     +x2_3+x2_4     +x1_2          +x4_2+x5_2>=1
+x2_3                         +x1_3     +x3_3+x4_3+x5_3>=1
+x2_4+x2_1     +x2_3          +x1_4     +x3_4+x4_4     >=1
+x2_5+x2_1+x2_2+x2_3+x2_4                              >=1
+x3_1     +x3_2                    +x2_1     +x4_1     >=1
+x3_2                         +x1_2+x2_2     +x4_2+x5_2>=1
+x3_3+x3_1+x3_2               +x1_3          +x4_3+x5_3>=1
+x3_4+x3_1+x3_2+x3_3     +x3_5                         >=1
+x3_5+x3_1+x3_2+x3_3          +x1_5+x2_5          +x5_5>=1
+x4_1     +x4_2+x4_3+x4_4+x4_5                         >=1
+x4_2          +x4_3     +x4_5+x1_2               +x5_2>=1
+x4_3                    +x4_5+x1_3               +x5_3>=1
+x4_4     +x4_2+x4_3     +x4_5          +x3_4          >=1
+x4_5                         +x1_5+x2_5+x3_5     +x5_5>=1
+x5_1     +x5_2     +x5_4+x5_5     +x2_1+x3_1+x4_1     >=1
+x5_2               +x5_4+x5_5+x1_2                    >=1
+x5_3+x5_1+x5_2     +x5_4+x5_5                         >=1
+x5_4                         +x1_4+x2_4+x3_4+x4_4     >=1
+x5_5               +x5_4          +x2_5               >=1

end
```

Vizsgáljuk meg alaposabban a stabilitási korlátokat a 8.3 kód esetén! Az első férfi és első nő összepárosítására vonatkozó korlát a `+x1_1+x2_1+x3_1+x4_1+x5_1>=1` összefüggés. Az első férfi rangsorában az első nő az első helyen áll, tehát ennek a férfinek nem áll érdekében más nővel blokkolni. Az első nő rangsorában az első férfi viszont az ötödik helyen áll, tehát ő csak akkor akar párban lenni ezzel a férfivel, ha a többi férfi nem választja őt. Másképpen, ha a második, harmadik, negyedik vagy az ötödik férfivel lehet párban, akkor nem kell neki az első férfi. Matematikailag ha `x2_1`, `x3_1`, `x4_1` vagy `x5_1` változók bármelyike is 1, akkor a korlát már teljesül. Ha mindegyik változó 0, akkor viszont `x1_1` változónak kell 1-nek lennie.

Nézzük az első férfi és negyedik nő hozzárendeléséhez tartozó stabilitási korlátot: `+x1_4+x1_1+x3_4+x4_4>=1`. Az első férfi rangsorában a negyedik nő a második helyen áll, tehát ha ő nem kapja meg a rangsorában első helyen szereplő (első) nőt, akkor szeretné a negyediket. A negyedik nő rangsorában az első férfi a harmadik helyen áll, tehát ő vagy a harmadik-, vagy a negyedik férfit szeretné, az első férfit pedig csak akkor, ha a harmadikat vagy negyediket nem kapja meg. Érdekessége ennek a formalizációnak, hogy az x változókra, ugyanúgy, mint a hozzárendelési feladat esetében, nem kell egészértékűségi kikötés. Azonban ebben az esetben az együttható mátrix nem teljesen unimoduláris mátrix, mégis fennáll, hogy a megoldás egészértékű lesz.

Az optimális megoldás:

```
   No. Column name  St   Activity
------ ------------ -- -------------
     1 x1_1         B              1
     8 x2_3         B              1
    12 x3_2         B              1
    20 x4_5         B              1
    24 x5_4         B              1
```

mátrixokban bejelölve

$$\begin{pmatrix} \boxed{1} & 5 & 4 & 2 & 3 \\ 2 & 4 & \boxed{1} & 3 & 5 \\ 2 & \boxed{1} & 3 & 5 & 4 \\ 5 & 3 & 2 & 4 & \boxed{1} \\ 4 & 3 & 5 & \boxed{1} & 2 \end{pmatrix} \quad \text{és} \quad \begin{pmatrix} \boxed{5} & 1 & 2 & 3 & 3 \\ 2 & 4 & \boxed{5} & 4 & 1 \\ 3 & \boxed{5} & 4 & 1 & 4 \\ 1 & 3 & 3 & 2 & \boxed{5} \\ 4 & 2 & 1 & \boxed{5} & 2 \end{pmatrix}$$

Látható, hogy mindegyik férfi a rangsorukban első helyen szereplő nőt kapta, viszont a nők korántsem jártak ilyen jól, számukra mindig a lista végén szereplő férfi jutott.

Nézzük a nő-optimális megoldást! Ebben az esetben a 8.3. kódban a célfüggvényt ki kell cserélni a

```
+5x1_1 +1x1_2 +2x1_3 +3x1_4 +3x1_5
+2x2_1 +4x2_2 +5x2_3 +4x2_4 +1x2_5
+3x3_1 +5x3_2 +4x3_3 +1x3_4 +4x3_5
+1x4_1 +3x4_2 +3x4_3 +2x4_4 +5x4_5
+4x5_1 +2x5_2 +1x5_3 +5x5_4 +2x5_5
```

kifejezésre.

A nő-optimális párosítás:

```
   No. Column name  St   Activity
------ ------------ -- -------------
     2 x1_2         B              1
    10 x2_5         B              1
    14 x3_4         B              1
    16 x4_1         B              1
    23 x5_3         B              1
```

mátrixokban bejelölve:

$$\begin{pmatrix} 1 & \boxed{5} & 4 & 2 & 3 \\ 2 & 4 & 1 & 3 & \boxed{5} \\ 2 & 1 & 3 & \boxed{5} & 4 \\ \boxed{5} & 3 & 2 & 4 & 1 \\ 4 & 3 & \boxed{5} & 1 & 2 \end{pmatrix} \quad \text{és} \quad \begin{pmatrix} 5 & \boxed{1} & 2 & 3 & 3 \\ 2 & 4 & 5 & 4 & \boxed{1} \\ 3 & 5 & 4 & \boxed{1} & 4 \\ \boxed{1} & 3 & 3 & 2 & 5 \\ 4 & 2 & \boxed{1} & 5 & 2 \end{pmatrix}$$

8.4. példa. *A korábbi megoldás során megkaptuk a férfi- és nő-optimális megoldást, amiket a Gale-Shapley algoritmussal is megkaphatnánk. Tudunk-e ezeken kívül olyan stabil párosítást találni, amelyben a férfiak és nők szempontjai nem ennyire szélsőségesen vannak figyelembe véve?*

Megoldás.

Egyik megoldás az lehet, hogy a célfüggvényben nem csak a férfiak vagy nők rangsorát vesszük figyelembe, hanem a kettő összegét (Lényegében a társadalmi összhasznosságot maximalizáljuk):

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(r_j^i + s_i^j)x_{ij} \ ,$$

ami a mi esetünkben:

```
+6x1_1 +6x1_2 +6x1_3 +5x1_4 +6x1_5
+4x2_1 +8x2_2 +6x2_3 +7x2_4 +6x2_5
+5x3_1 +6x3_2 +7x3_3 +6x3_4 +8x3_5
+6x4_1 +6x4_2 +5x4_3 +6x4_4 +6x4_5
+8x5_1 +5x5_2 +6x5_3 +6x5_4 +4x5_5
```

Ebben az esetben az optimális megoldás

```
   No. Column name  St   Activity
------ ------------ -- -------------
     4 x1_4         B              1
     6 x2_1         B              1
    12 x3_2         B              1
    18 x4_3         B              1
    25 x5_5         B              1
```

mátrixokban bejelölve

$$\begin{pmatrix} 1 & 5 & 4 & \boxed{2} & 3 \\ \boxed{2} & 4 & 1 & 3 & 5 \\ 2 & \boxed{1} & 3 & 5 & 4 \\ 5 & 3 & \boxed{2} & 4 & 1 \\ 4 & 3 & 5 & 1 & \boxed{2} \end{pmatrix} \quad \text{és} \quad \begin{pmatrix} 5 & 1 & 2 & \boxed{3} & 3 \\ \boxed{2} & 4 & 5 & 4 & 1 \\ 3 & \boxed{5} & 4 & 1 & 4 \\ 1 & 3 & \boxed{3} & 2 & 5 \\ 4 & 2 & 1 & 5 & \boxed{2} \end{pmatrix}$$

Megoldás #2.

Látható, hogy az előző a megoldás sokkal kiegyenlítettebb a férfiak és nők tekintetében. De még ebben az esetben is a második nő nagyon rosszul jár. Kérdés, hogy van-e olyan megoldás ahol senki sem jár nagyon rosszul? Konkrétabban, van-e olyan megoldás, ahol senki sem kap olyan személyt, aki a listájában az ötödik helyen van. Ezt viszonylag egyszerűen kideríthetjük: az M1F1, M1F2, M2F3, M2F5, M3F2, M3F4, M4F1, M4F5, M5F3 és M5F4 azok a párosítások, ahol valamelyik fél az legrosszabb helyen szereplő társat kapja. Kérdés, hogy lehet-e ez az összeg 0. Ezt legkönnyebben úgy deríthetjük ki, hogy a célfüggvényt kicseréljük a

```
+x1_1+x1_2
+x2_3+x2_5
+x3_2+x3_4
+x4_1+x4_5
+x5_3+x5_4
```

célfüggvényre. Ebben az esetben a célfüggvény optimális értéke 1 lesz, tehát nem tudjuk elérni azt, hogy senki se kapja párba a listája legrosszabb helyén található párt.

Az operációkutatásban több helyen is felmerül, amikor a legrosszabb esetet (pl. soronként a minimumot) akarjuk maximalizálni. Igazából itt is ez a helyzet: a férfiak és nők esetében a legrosszabb esetet akarjuk maximalizálni. Erre a kérdésre a vizsgált probléma esetén már tudjuk a választ, de érdemes a kérdést általánosan is feltenni: melyik hozzárendelés az, amikor a legrosszabbul járó férfi vagy nő is a lehető legjobban jár.

Másképpen: ha lehet kerüljük el, hogy bárki is az 5. helyen álló párt kapja. Ha ez lehetséges, akkor lehetőleg kerüljük el azt, hogy bárki is a listája 4. helyén álló párt kapja. Ha ez is lehetséges, akkor lehetőleg kerüljük el, hogy bárki is a listája 3. helyén álló párt kapja...

Ha valaki a listája 5. helyén álló párt kapja, akkor

```
+x1_1+x1_2
+x2_3+x2_5
+x3_2+x3_4
+x4_1+x4_5
+x5_3+x5_4
```

összeg legalább 1 lesz. Elvi maximuma ennek az összegnek 10 (bár lehetne ennél jobb felső határt is megadni, számunkra nem lesz lényeges).

Ha valaki a listája 4. helyén álló párt kapja, akkor

```
+x1_3
+x2_2+x2_4
+x3_3+x3_5
+x4_4
+x5_1
```

összeg legalább 1 lesz. Hasonlóan meg lehet adni, a kifejezést a 3. és 2. helyre is (az első helyre nem lesz szükségünk).

Ezek a célfüggvényeket ún. 'lexikografikusan' szeretnénk rendezni, tehát először az `+x1_1+x1_2+x2_3+x2_5+x3_2+x3_4+x4_1+x4_5+x5_3+x5_4` kifejezést szeretnénk minimalizálni. Utána ennek a kifejezésnek az értékét rögzítjük az elért szinten, utána ezen feltétel hozzáadása mellett minimalizáljuk a `+x1_3+x2_2+x2_4+x3_3+x3_5+x4_4+x5_1` kifejezést. Ezt több lépésben is megtehetjük, de erre nincs feltétlenül szükség, egyetlen lépésben is megtehető. Azt kell csak elérni, hogy a `+x1_3+x2_2+x2_4+x3_3+x3_5+x4_4+x5_1` kifejezés változása ne tudja megváltoztatni `+x1_1+x1_2+x2_3+x2_5+x3_2+x3_4+x4_1+x4_5+x5_3+x5_4` kifejezést, azaz ne lehessen az, hogy pl. csökkentjük 3-mal a negyedik helyet kapó személyek számát, cserébe nő eggyel az ötödik helyet kapó személyek száma. A `+x1_3+x2_2+x2_4+x3_3+x3_5+x4_4+x5_1` maximális száma most 7, de bizonyos esetekben a 4. helyezések egybeestek. De ez a kifejezés sem lehet 10-nél több, vagy általánosan: ha n férfi és n nő szerepel a példában akkor $2n$-nél több nem lehet. Ha az ötödik helyet kapók számát beszorozzuk $2n + 1$-gyel, akkor el tudjuk érni, hogy ne állhasson elő a fent leírt helyzet.

Nyilván nem csak a 4. és 5. helyet kapók számát kell figyelembe venni, hanem az összes többit is. Tehát a 2. helyet kapók számát beszorozzuk $2n + 1$-gyel. A harmadik helyet kapók számát $(2n + 1)^2$-nel, negyedik helyet kapók számát $(2n + 1)^3$-nal, stb... Ha az elemszám nagyon nagy, akkor ez csak egy elméleti konstrukció, a gyakorlati kivitelezhetősége csekély, viszont az utolsó, utolsó előtti helyekre könnyen alkalmazható. Másrészt a $(2n+1)^k$ egy olyan szám, ami minden eshetőség esetén működik, azaz a konkrét példa esetén lehet, hogy lehet csökkenteni ezt a számot.

Ha alkalmazzuk $(2n+1)^k$ 'szabályt' a mi konkrét esetünkben ez azt jelenti, hogy 11-gyel kell szorozni a 2. helyes hozzárendeléseket, 121-gyel a harmadik helyes hozzárendeléseket, 1331-gyel a 4. helyes hozzárendeléseket és 14641-gyel az ötödik helyes hozzárendeléseket. Nyilván egy döntési változóhoz két helyezés is tartozik, a CPXLP felírásban viszont egy változó csak egyszer szerepelhet a célfüggvényben. Így ezt a két értéket össze kell adni: tehát ha az $FiMj$ hozzárendelés esetén (i férfi a j nőt kapja, a j nő az i férfi rangsorában a k helyen áll, az i férfi pedig a j nő rangsorában a ℓ helyen áll, úgy x_{ij} változó együtthatója a célfüggvényben $(2n+1)^{k-1}+(2n+1)^{\ell-1}$ lesz. Az F1M1 hozzárendelés esetén az első nő az első férfi rangsorában első helyen áll, az első férfi az első nő rangsorában viszont az ötödik helyen, így `x1_1` változó együtthatója $11^0+11^4=14652$.

```
+14642x1_1 +14642x1_2 + 1342x1_3 +  132x1_4 +  242x1_5
+   22x2_1 + 2662x2_2 +14642x2_3 + 1452x2_4 +14642x2_5
+  132x3_1 +14642x3_2 + 1452x3_3 +14642x3_4 + 2662x3_5
+14642x4_1 +  242x4_2 +  132x4_3 + 1342x4_4 +14642x4_5
+ 2662x5_1 +  132x5_2 +14642x5_3 +14642x5_4 +   22x5_5
```

Ezzel a célfüggvénnyel sem kapunk újabb megoldást. Érdemes azonban még egy módszert megismerni, ami gyakran használatos az LP modellezés során. Legyen y változó értéke a legrosszabb sorszám, ami szerepel a párosítások között, majd pedig ezt az értéket szeretnénk minimalizálni. Minden cella esetén megadunk két korlátot: $r^i_j x_{ij} \leq y$ és $s^j_i x_{ij} \leq y$. Ha $x_{ij}=0$, akkor ezek a korlátok csak azt fogják jelenteni, hogy $y \geq 0$ ami triviálisan teljesülni fog. Ha $x_{ij}=1$, akkor viszont ezek korlátok elő fogják írni, hogy y legyen nagyobb vagy egyenlő, mint az x_{ij} párosításhoz tartozó rangszámok. Egyetlen hátulütője van ennek a megközelítésnek: így már nem fogunk egészértékű megoldást, kapni, tehát elő kell írni, az x változókra, hogy egészértékűek.

8.5. kód.

```
min
y

subject to
+x1_1+x1_2+x1_3+x1_4+x1_5=1
```

```
+x2_1+x2_2+x2_3+x2_4+x2_5=1
+x3_1+x3_2+x3_3+x3_4+x3_5=1
+x4_1+x4_2+x4_3+x4_4+x4_5=1
+x5_1+x5_2+x5_3+x5_4+x5_5=1

+x1_1+x2_1+x3_1+x4_1+x5_1=1
+x1_2+x2_2+x3_2+x4_2+x5_2=1
+x1_3+x2_3+x3_3+x4_3+x5_3=1
+x1_4+x2_4+x3_4+x4_4+x5_4=1
+x1_5+x2_5+x3_5+x4_5+x5_5=1

+x1_1                              +x2_1+x3_1+x4_1+x5_1>=1
+x1_2+x1_1     +x1_3+x1_4+x1_5                         >=1
+x1_3+x1_1          +x1_4+x1_5                    +x5_3>=1
+x1_4+x1_1                              +x3_4+x4_4     >=1
+x1_5+x1_1          +x1_4          +x2_5          +x5_5>=1
+x2_1          +x2_3                         +x4_1     >=1
+x2_2+x2_1     +x2_3+x2_4     +x1_2          +x4_2+x5_2>=1
+x2_3                         +x1_3     +x3_3+x4_3+x5_3>=1
+x2_4+x2_1     +x2_3          +x1_4     +x3_4+x4_4     >=1
+x2_5+x2_1+x2_2+x2_3+x2_4                              >=1
+x3_1     +x3_2                    +x2_1     +x4_1     >=1
+x3_2                         +x1_2+x2_2     +x4_2+x5_2>=1
+x3_3+x3_1+x3_2               +x1_3          +x4_3+x5_3>=1
+x3_4+x3_1+x3_2+x3_3     +x3_5                         >=1
+x3_5+x3_1+x3_2+x3_3          +x1_5+x2_5          +x5_5>=1
+x4_1     +x4_2+x4_3+x4_4+x4_5                         >=1
+x4_2          +x4_3     +x4_5+x1_2               +x5_2>=1
+x4_3                    +x4_5+x1_3               +x5_3>=1
+x4_4     +x4_2+x4_3     +x4_5          +x3_4          >=1
+x4_5                         +x1_5+x2_5+x3_5     +x5_5>=1
+x5_1     +x5_2     +x5_4+x5_5     +x2_1+x3_1+x4_1     >=1
+x5_2               +x5_4+x5_5+x1_2                    >=1
+x5_3+x5_1+x5_2     +x5_4+x5_5                         >=1
+x5_4                         +x1_4+x2_4+x3_4+x4_4     >=1
+x5_5               +x5_4          +x2_5               >=1

+1x1_1-y<=0
+2x2_1-y<=0
+2x3_1-y<=0
+5x4_1-y<=0
+4x5_1-y<=0
+5x1_2-y<=0
+4x2_2-y<=0
+1x3_2-y<=0
```

```
+3x4_2-y<=0
+3x5_2-y<=0
+4x1_3-y<=0
+1x2_3-y<=0
+3x3_3-y<=0
+2x4_3-y<=0
+5x5_3-y<=0
+2x1_4-y<=0
+3x2_4-y<=0
+5x3_4-y<=0
+4x4_4-y<=0
+1x5_4-y<=0
+3x1_5-y<=0
+5x2_5-y<=0
+4x3_5-y<=0
+1x4_5-y<=0
+2x5_5-y<=0

+5x1_1-y<=0
+2x2_1-y<=0
+3x3_1-y<=0
+1x4_1-y<=0
+4x5_1-y<=0
+1x1_2-y<=0
+4x2_2-y<=0
+5x3_2-y<=0
+3x4_2-y<=0
+2x5_2-y<=0
+2x1_3-y<=0
+5x2_3-y<=0
+4x3_3-y<=0
+3x4_3-y<=0
+1x5_3-y<=0
+3x1_4-y<=0
+4x2_4-y<=0
+1x3_4-y<=0
+2x4_4-y<=0
+5x5_4-y<=0
+3x1_5-y<=0
+1x2_5-y<=0
+4x3_5-y<=0
+5x4_5-y<=0
+2x5_5-y<=0

binary
```

```
x1_1
x2_1
x3_1
x4_1
x5_1
x1_2
x2_2
x3_2
x4_2
x5_2
x1_3
x2_3
x3_3
x4_3
x5_3
x1_4
x2_4
x3_4
x4_4
x5_4
x1_5
x2_5
x3_5
x4_5
x5_5

end
```

A 8.5. kód futtatása során a célfüggvény 5, tehát bármelyik stabil megoldás esetén lesz olyan személy aki az lista legrosszabb helyén szereplő párt kapja. Ez az eredmény kijött már korábban is, de megnyugtató, hogy konzisztens eredményeket kapunk.

Az LP felírás azért (is) érdekes, mert nem csak a sztenderd feladatot tudjuk megoldani, hanem tudjuk a feladatot bővíteni is. Nézzük először milyen specialitások szoktak előfordulni.

8.6. példa. *Egy adott férfi számára lehetnek olyan nők, akire azt mondja, hogy semmi esetre sem felel meg a számára – ami azt is jelenti, hogy ha kell inkább az egyedüllétet választja (ha belegondolunk felvételi feladatok esetében ez szinte törvényszerű, a diákok nem rangsorolják az összes felsőoktatási képzést, hanem csak néhányra jelentkeznek).*

Tekintsük a következő preferencia mátrixokat:

$$\begin{pmatrix} 1 & & 4 & 2 & 3 \\ 2 & 3 & 1 & & 4 \\ 2 & 1 & 3 & 5 & 4 \\ 5 & 3 & 2 & 4 & 1 \\ & 3 & 4 & 1 & 2 \end{pmatrix} \quad \textit{és} \quad \begin{pmatrix} & 1 & 2 & 2 & \\ 2 & 4 & 5 & 3 & 1 \\ 3 & 5 & 4 & 1 & \\ 1 & 3 & 3 & & \\ & 2 & 1 & 4 & 2 \end{pmatrix}$$

A mátrixokban szereplő hiányzó értékek azt jelenti, az adott férfi/nő számára a hozzárendelés elfogadhatatlan. Pl.: az első férfi számára a második nő elfogadhatatlan, inkább az egyedüllétet választja.

Megoldás.

Ha nem teljes a rangsor, akkor nem biztos, hogy mindenkinek jut pár, de annyi állítható, hogy minden stabil párosítás esetén ugyanannyi pár lesz. Akinek nem jut pár valamelyik stabil párosítás esetén, annak semelyik esetben sem fog jutni. Tehát továbbra is beszélhetünk férfi- és nő-optimális párosításokról, stb ...

Az LP felírás sem bonyolult ebben az esetben. Ha valamelyik mátrix esetén az i sor j oszlopa hiányzik, akkor töröljük a kapcsolódó x_{ij} változót (nem kell, hogy mind a két helyen hiányozzon; elég ha egyik mátrixból hiányzik).

Nézzük meg férfi- és nő-optimális megoldást ebben az esetben!

8.7. kód.
```
min
ffi:
              +4x1_3+ 2x1_4
+2x2_1 +3x2_2 +1x2_3        +4x2_5
+2x3_1 +1x3_2 +3x3_3 +5x3_4
+5x4_1 +3x4_2 +2x4_3
       +3x5_2 +4x5_3 +1x5_4 +2x5_5

subject to

+x1_3 +x1_4             = 1
+x2_1 +x2_2 +x2_3 +x2_5 = 1
+x3_1 +x3_2 +x3_3 +x3_4 = 1
+x4_1 +x4_2 +x4_3       = 1
+x5_2 +x5_3 +x5_4 +x5_5 = 1
```

```
+x2_1 +x3_1 +x4_1             = 1
+x2_2 +x3_2 +x4_2 +x5_2       = 1
+x1_3 +x2_3 +x3_3 +x4_3 +x5_3 = 1
+x1_4 +x3_4 +x5_4             = 1
+x2_5 +x5_5                   = 1

x1_3               +x1_4                         +x5_3 >= 1
x1_4                                   +x3_4           >= 1
x2_1          +x2_3                         +x4_1      >= 1
x2_2+x2_1     +x2_3                         +x4_2+x5_2 >= 1
x2_3                         +x1_3     +x3_3+x4_3+x5_3 >= 1
x2_5+x2_1+x2_2+x2_3                                    >= 1
x3_1     +x3_2                    +x2_1     +x4_1      >= 1
x3_2     +x2_2                              +x4_2+x5_2 >= 1
x3_3+x3_1+x3_2               +x1_3          +x4_3+x5_3 >= 1
x3_4+x3_1+x3_2+x3_3                                    >= 1
x4_1     +x4_2+x4_3                                    >= 1
x4_2          +x4_3                              +x5_2 >= 1
x4_3                         +x1_3               +x5_3 >= 1
x5_2               +x5_4+x5_5                          >= 1
x5_3     +x5_2     +x5_4+x5_5                          >= 1
x5_4                         +x1_4     +x3_4           >= 1
x5_5          +x5_4               +x2_5                >= 1

end
```

Az optimális megoldás:

```
   No. Column name  St   Activity
------ ------------ -- -------------
     2 x1_4         B              1
     3 x2_1         B              1
     8 x3_2         B              1
    13 x4_3         B              1
    17 x5_5         B              1
```

$$\begin{pmatrix} 1 & & 4 & \boxed{2} & 3 \\ \boxed{2} & 3 & 1 & & 4 \\ 2 & \boxed{1} & 3 & 5 & 4 \\ 5 & 3 & \boxed{2} & 4 & 1 \\ & 3 & 4 & 1 & \boxed{2} \end{pmatrix} \quad \text{és} \quad \begin{pmatrix} & 1 & 2 & \boxed{2} & \\ \boxed{2} & 4 & 5 & 3 & 1 \\ 3 & \boxed{5} & 4 & 1 & \\ 1 & 3 & \boxed{3} & & \\ & 2 & 1 & 4 & \boxed{2} \end{pmatrix}$$

A nő-optimális esetben megint egyedül csak a célfüggvény különbözik:

```
no:
            +1x1_3+2x1_4
+2x2_1+4x2_2+5x2_3      +1x2_5
+3x3_1+5x3_2+4x3_3+1x3_4
+1x4_1+3x4_2+3x4_3
      +2x5_2+1x5_3+4x5_4+2x5_5
```

Ebben az esetben a stabil párosítás:

$$
\begin{pmatrix}
1 & & \boxed{4} & 2 & 3 \\
2 & 3 & 1 & & \boxed{4} \\
2 & 1 & 3 & \boxed{5} & 4 \\
\boxed{5} & 3 & 2 & 4 & 1 \\
 & \boxed{3} & 4 & 1 & 2
\end{pmatrix}
\quad \text{és} \quad
\begin{pmatrix}
 & 1 & \boxed{2} & 2 & \\
2 & 4 & 5 & 3 & \boxed{1} \\
3 & 5 & 4 & \boxed{1} & \\
\boxed{1} & 3 & 3 & & \\
 & \boxed{2} & 1 & 4 & 2
\end{pmatrix}
$$

8.1. Gyakorló feladatok

8.1.1. Egyszerűbb feladatok

8.8. példa. *Tekintsük a következő preferencia mátrixokat:*

$$
\begin{pmatrix}
3 & 1 & & 2 & 4 & 5 \\
5 & & 3 & 1 & 2 & 4 \\
4 & 5 & & 2 & 1 & 3 \\
 & 5 & 1 & 2 & 3 & 4 \\
 & 2 & 3 & 4 & 5 & 1 \\
5 & 3 & 2 & & 4 & 1
\end{pmatrix}
\quad \textit{és} \quad
\begin{pmatrix}
2 & 3 & 3 & 1 & 1 & 5 \\
 & 2 & 4 & 5 & 3 & 1 \\
3 & 4 & 2 & 4 & 2 & 3 \\
1 & 1 & 5 & 2 & 5 & 2 \\
5 & & 1 & 3 & & 4 \\
4 & 5 & & & 4 &
\end{pmatrix}
$$

Adja meg a férfi- illetve nő-optimális stabil párosítást.

Megoldás a 341. oldalon.

8.9. példa. *A következő házasítási piacon 8 férfi és 5 nő van. A szereplők* *preferenciái:*

$$\begin{pmatrix} 4 & 3 & 2 & 5 & 1 \\ 3 & 4 & 2 & 5 & 1 \\ 4 & 5 & 1 & 3 & 2 \\ 3 & 1 & 4 & 5 & 2 \\ 3 & 2 & 4 & 1 & 5 \\ 4 & 3 & 2 & 1 & 5 \\ 2 & 1 & 4 & 3 & 5 \\ 3 & 5 & 1 & 2 & 4 \end{pmatrix} \quad \textit{és} \quad \begin{pmatrix} 6 & 5 & 2 & 1 & 1 \\ 3 & 6 & 5 & 7 & 5 \\ 1 & 7 & 4 & 8 & 2 \\ 4 & 8 & 6 & 6 & 6 \\ 2 & 3 & 1 & 2 & 4 \\ 5 & 1 & 7 & 4 & 8 \\ 7 & 2 & 8 & 3 & 7 \\ 8 & 4 & 3 & 5 & 3 \end{pmatrix}$$

Adja meg a férfi- illetve nő-optimális stabil párosítást. Ezekben az esetekben melyik férfiak maradnak pár nélkül?

Megoldás a 345. oldalon.

8.10. példa. *3 szakra 9 diák jelentkezik. A következő táblázatban látható a diákok preferenciái (oszloponként):*

$$\begin{pmatrix} 2 & 2 & 2 & 1 & 2 & 3 & 1 & 3 & 3 \\ 1 & 3 & 1 & 3 & 1 & 1 & 2 & 1 & 2 \\ 3 & 1 & 3 & 2 & 3 & 2 & 3 & 2 & 1 \end{pmatrix}$$

A következő táblázatban pedig a diákok pontjai láthatóak, amit a maximálisan megszerezhető 500 pontból szereztek, minden egyes szakra:

$$\begin{pmatrix} 391 & 375 & 477 & 379 & 486 & 476 & 454 & 481 & 486 \\ 352 & 407 & 490 & 427 & 371 & 470 & 424 & 435 & 473 \\ 393 & 351 & 355 & 374 & 391 & 482 & 456 & 440 & 431 \end{pmatrix}$$

A magasabb pontszám jobb eredményt jelent. Minden szakra legfeljebb 4 diák kerülhet be.

Adjon meg egy szak- illetve diák-optimális stabil párosítást.

Megoldás a 348. oldalon.

8.1.2. Nehezebb feladatok

8.11. példa. *Tekintsük ismét a 8.8. példában megadott preferenciákat, de most kizárjuk azokat a hozzárendeléseket, amikor mind a két fél értékelése a másikról magasabb, mint 3. Amennyiben valakit olyannal hozunk össze, akit nem az első 3 között értékelt, akkor kompenzációként fizetünk neki 50 pénzegységet. Amennyiben valaki nem kap párt, akkor neki pedig 100 pénzegységet fizetünk.*

Adjon meg egy olyan stabil párosítást, amely mellett a fizetendő összeg minimális.

Megoldás a 352. oldalon.

8.12. példa. *(Stabil 'szobatárs' probléma) 8 hallgatónak 2 fős csoportokban kell megoldania 4 feladatot. Mindenki értékeli a többi hallgatót, az alapján, hogy mennyire szeretne az illetővel együtt lenni a csoportmunkában (A sorban lévő hallgató értékeli az oszlopban lévő hallgatót):*

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1		*4*	*3*	*1*	*6*	*7*	*2*	*5*
2	*5*		*6*	*4*	*7*	*3*	*2*	*1*
3	*3*	*6*		*5*	*2*	*7*	*4*	*1*
4	*1*	*6*	*5*		*3*	*4*	*7*	*2*
5	*7*	*6*	*1*	*2*		*3*	*4*	*5*
6	*1*	*4*	*5*	*7*	*2*		*3*	*6*
7	*6*	*5*	*1*	*3*	*2*	*7*		*4*
8	*1*	*4*	*3*	*5*	*6*	*2*	*7*	

Adjon meg egy stabil párosítást!

Megoldás a 355. oldalon.

9. fejezet

Játékelmélet

A jegyzet ezen fejezete némiképp különbözik a korábbi fejezetektől. A korábbi fejezetekben jellemzően valamilyen gráf játszotta a főszerepet, és ezen a gráfon végeztünk el valamilyen optimalizálási feladatot. Játékelmélet esetén nem gráfok játszák a főszerepet.

Mindennapi helyzetünkben tapasztaljuk, hogy az általunk elért eredmény nem csak saját döntésünktől függ, hanem másokétól is. Hiába tartom tisztán az utcát, ha a szomszéd szemetel. Egy egyetemi szak csak akkor indul el, ha megfelelő számú hallgató jelentkezik rá. Egy bolt nyeresége függ attól, hogy van-e a közelében másik hasonló profilú üzlet, és sorolhatnánk tovább. A közgazdaságtanban a játékelmélet foglalkozik ilyen szituációk elemzésével, ezáltal a szereplők motivációit jobban megértjük, viselkedésüket jobban előre tudjuk jelezni (emlékezzünk vissza pl. a duopólium modellekre).

Játékelmélet látóköre meglehetősen széles, akár több félévig is lehetne tárgyalni. Így vizsgálódásaink körét meglehetősen leszűkítjük, de az alapvető fogalmakat így is szemléltetni tudjuk. Ebben a fejezetben csak az ún. mátrixjátékok témakörét érintjük. Mátrixjátékok esetén a játékosok száma pontosan 2. Mindkét játékosnak véges sok stratégiája van. További megkötés, hogy zéróösszegű vagy konstansösszegű játékokról van szó. Zéróösszegű játék esetén a nyeremények összege 0, tehát amennyit nyer az egyik fél, annyit veszít a másik. Általában a szerencsejátékok ilyenek. Kicsit bővebb kategória a konstansösszegű játékok, amiben a két játékos kifizetésének összege konstans. Például egy adott nagyságú piacon osztozik két vállalat. Minél nagyobb a piaci részesedése az egyiknek, annál kisebb a másiknak.

Azért hívjuk ezeket a játékokat mátrixjátéknak, mert egyetlen mátrixszal le lehet írni a teljes játékot. Az első játékos stratégiái a sorokban szerepelnek (ezért is fogjuk ezt a játkéost sorjátékosnak hívni), a második játékos (oszlopjátékos) stratégiái oszlopokban. A mátrix elemei a sorjátékos kifizetéseit mutatják, ebből már következtetni lehet az oszlopjátékos kifizetésére is. Például legyen a kifizetési mátrix:

$$\begin{pmatrix} -1 & 9 & 0 \\ 3 & -2 & -1 \end{pmatrix}$$

Ekkor a sorjátékosnak két stratégiája van, az oszlopjátékosnak 3. Ha a sorjátkos az első stratégiáját játssza, az oszlopjátékos pedig a másodikat, akkor a sorjátékos kifizetése 9. Ha pl. zéróösszegű játékról van szó, akkor ez azt jelenti, hogy az oszlopjátékos kifizetése -9.

9.1. Mátrixjátékok

9.1. példa. *Tekintsünk egy egyszerű kártyajátékot. Az első játékosnak van 3 lapja: 4-es, 5-ös és 5-ös. A második játékosnak is van 3 lapja van 5-ös, 5-ös és 6-os. A játékosok meghatározhatják a kártyák sorrendjét, majd a kártyákat egymásra rakva fejjel lefelé leteszik az asztalra. Tehát minkét játékos előtt 3-3 lefordított kártya van. Majd egyesével felfordítják. Az első kört a sorjátékos kezdi. Minden kört a nagyobb lap viszi (szín nem számít), ha két egyforma lapot tesznek az viszi, aki hívta. Aki elvitte a kört, az kezdi a következőt. A játékosok kifizetése az elvitt körök számával egyezik meg. Adja meg a kifizetési mátrixot!*

Megoldás

Mind a két játékosnak 3-3 stratégiája van, ami azt jelenti, hogy milyen sorrendben rakja le a kártyákat (a két négyes kártyát nem különböztetjük meg). A sorjátékos esetén 455 jelölje azt az esetet, amikor egy 4-es lap van legfelül (ezt fogja először kijátszani), után a 5-ös lap, végül megint egy 5-ös lap. Ezen jelölések esetén a sorjátékos 3 stratégiája: 455, 545 és 554. Ezeket rendre első, második és harmadik stratégiának fogjuk hívni. Az oszlopjátékos 3 stratégiája: 556, 565 és 655. Ha mindkét játékos az első stratégiáját játssza, azaz (455 és 556), akkor az első kört a második (oszlop) játékos viszi (5-ös lap nagyobb értékű, mint a 4-es), ő hívja a következő kört is, amit megint csak ő visz (mind a két lap 5-ös, ezért az viszi, aki hívta), ezért ő kezdi a harmadik kört, amit megint csak ő visz (6-os lap nagyobb, mint a 5-ös lap). Tehát ebben az esetben a sorjátékos kifizetése 0 (kör). Hasonlóan végig kell számolni a többi stratégiaegyüttest is:

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & 1 & 0 \end{pmatrix}$$

9.2. példa. *A 9.1. példa esetén melyik játékosnak milyen stratégiát lenne érdemes játszania?*

Megoldás

Nem túl nagy erőfeszítés árán rá lehet jönni, hogy az első játékosnak először mindenképpen 5-öst kell hívnia. Ha 4-est hív, akkor biztosan elveszíti ezt a kört, ha 5-öst hív, akkor ha a másik játékos is 5-öst hív, akkor nyer. Az oszlopjátékosnak viszont mindig 6-os lappal kell kezdeni. A 6-os lap mindenképpen ütni fog, és akkor utána az 5-ös lapok is. De ha 5-ös lappal kezd, akkor azt esetleg elveszítheti. Tehát ha jól játszanak a játékosok (főleg a második), akkor az első játékosnak nem lesz ütése.

De próbáljunk játékelméleti oldalról érvelni. Azt mondjuk, hogy az első játékosnak a harmadik stratégiája dominálja az első stratégiáját. Ez azért van így, mert a sorjátékos a harmadik stratégiájával nem jár rosszabbul, mint az első stratégiájával, függetlenül attól, hogy mit játszik a másik játékos. Konkrétabban: tegyük fel, hogy a második játékos az első stratégiáját játssza, tehát a kifizetési mátrix első oszlopát kell tekintenünk. Ha a sorjátékos az első stratégiáját játssza, akkor a kifizetése 0 egység lesz, ha a harmadik stratégiáját játssza, akkor 2 lesz a kifizetése. Nyilván a sorjátékos számára a 2 jobb, mint a 0. Ha az oszlopjátékos a második stratégiáját játssza, akkor a sorjátékos megint csak a harmadik stratégiájával jobban jár, mint az elsővel; az egységnyi kifizetés jobb, mint a semmi. Ha az oszlopjátékos a harmadik stratégiáját játssza, akkor az első és harmadik stratégia ugyanazt a kifizetést fogja eredményezni (0-t), de ekkor is állíthatjuk, hogy a harmadik stratégiájával nem jár rosszabbul, mint az elsővel. Tehát összesítve: függetlenül az oszlopjátékos stratégiájától, a sorjátékos harmadik stratégiája sohasem eredményez kisebb kifizetést, mint az első stratégiája (és többször kifejezetten jobb a harmadik stratégia). Tehát a sorjátékos harmadik stratégiája dominálja a sorjátékos első stratégiáját. Racionális sorjátékos az első stratégiáját nem fogja játszani, helyette érdemesebb a harmadikat játszania.

$$\begin{array}{ccc} \cancel{0} & \cancel{0} & \cancel{0} \\ 1 & 1 & 0 \\ 2 & 1 & 0 \end{array}$$

Hasonlóan rá lehet jönni, hogy a sorjátékos harmadik stratégiája dominálja a második stratégiáját is.

$$\begin{array}{ccc} \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{1} & \cancel{1} & \cancel{0} \\ 2 & 1 & 0 \end{array}$$

De a játékosokról nem csak azt tételezzük fel, hogy racionálisak, hanem azt is, hogy egymást is racionálisnak tekintik. Tehát ha az oszlopjátékos ránéz a kifizetésekre, rögtön látja, hogy a sorjátékos nem fogja játszani az első és második stratégiáját. Tehát a kifizetési mátrix az eredeti mátrix harmadik sorára redukálódik. De ekkor az oszlopjátékos harmadik stratégiája dominálja az elsőt és a másodikat (az oszlopjátékos minimalizál!).

$$\begin{array}{ccc} \cancel{0} & \cancel{0} & \cancel{0} \\ \cancel{1} & \cancel{1} & \cancel{0} \\ \cancel{2} & \cancel{1} & 0 \end{array}$$

Tehát kialakult egy ún. domináns stratégiai egyensúly: az első játékos a harmadik stratégiáját játssza, a második szintén a harmadikat.

9.3. példa. *Tekintsük az alábbi kifizetési mátrixot (ami nem a korábban leírt kártyajátékhoz tartozik, hanem valami másik játékhoz):*

$$\begin{pmatrix} 5 & 1 & 0 \\ 4 & 2 & 3 \\ 0 & 1 & 4 \end{pmatrix}$$

Kialakul-e egyensúly ebben az esetben?

Megoldás

Könnyen ellenőrizhetjük, hogy ebben az esetben egyik stratégia sem dominál valamelyik másikat. Tehát nincs domináns stratégiai egyensúly. Ugyanakkor ebben az esetben is kialakul ún. Nash-egyensúly. A Nash-egyensúly azt jeleni, hogy abból az állapotból nem érdemes senkinek eltérnie. Tegyük fel, hogy valahogy kialakult, hogy mind a két játékos a második stratégiáját játssza. Ebből az állapotból egyoldalúan senkinek sem érdemes elérnie.

Ha sorjátékos eltérne (egyoldalúan) és nem a második stratégiáját játszaná, hanem az elsőt, akkor a kifizetése nem 2 lenne, hanem 1, tehát rosszabbul jár. Hasonlóan, ha nem a második, hanem a harmadik stratégiáját játssza, akkor szintén a két egységnyi kifizetést cserélné el 1 egységnyi kifizetésre, tehát nem járna jobban.

Nézzük az oszlopjátékost. Ha ő nem a második stratégiáját játszaná, hanem az elsőt, akkor a sorjátékos kifizetése nem 2 lenne, hanem 4, ami számára hátrányos lenne (oszlopjátékos minimalizál). Hasonlóan, ha az oszlopjátékos nem a második stratégiáját játszaná, hanem a harmadikat, akkor a sorjátékos kifizetése nem 2 lenne, hanem 3, ami az oszlopjátékosra nézve megint csak hátrányos.

Másképpen úgy is fogalmazhatunk, hogy a kifizetési mátrix második sorában, második oszlopában lévő érték ún. nyeregpont. Ez azt is jelenti, hogy ez az érték sorában minimális és oszlopában maximális.

A nyeregpontnak mátrixjátékok esetén más jelentést is adhatunk: nézzük meg a játékosok ún. biztonsági szintjét. Mi az a kifizetés, amit a játékosok garantálni tudnak maguknak? A sorjátékos esetén ez 2. Ha ő a második stratégiáját játssza, akkor az ő kifizetése 2-nél nem lehet kevesebb, függetlenül attól, hogy melyik stratégiáját játssza az oszlopjátékos. Ellenőrizzük le: ha az oszlopjátékos az első stratégiáját játssza, akkor a sorjátékos kifizetése 4, ha a másodikat, akkor 2, ha a harmadikat, akkor 3. Tehát a legrosszabb esetben is 2 a kifizetés. Nézzük meg mi lenne, ha a sorjátékos az első stratégiáját játszaná! Ekkor a legrosszabb esetben (ha az oszlopjátékos a harmadik stratégiáját játssza) 0 lenne a kifizetése, ami rosszabb, mint a korábban megkapott 2.

Másként úgy is mondhatjuk, hogy a választás *maximin*. Tehát a minden sorban megnézzük, hogy hogy mi a minimális érték és ezek közül vesszük a legjobbat. A sorjátékos esetében ez a 2.

Nézzük most az oszlopjátékost! Az oszlopjátékos esetén a biztonsági szint megint csak 2. Amennyiben a második stratégiáját választja, akkor a (sorjátékos) kifizetés(e) nem lesz 2-nél nagyobb (oszlopjátékos minimalizál!). Ha az első stratégiáját játssza, akkor a kifizetés 5 is lehet, ami most is rosszabb, mint a korábban megkapott 2. Ez a *minimax* választás. Minden oszlopban megnézzük mi a maximális érték, és ezek közül vesszük a legkisebbet. Ez megint csak 2. Tehát a maximin választás megegyezik minimax választással.

9.4. példa. *A 9.1. példa adatain módosítsunk egy kicsit: az első játékosnak továbbra is 3 kártyája van, de ezek most: 4-es, 5-ös és 7-es (a második játékos kártyái változatlanok). Mit mondhatunk ebben az esetben a Nash-egyensúlyról?*

Megoldás

Az oszlopjátékos stratégiái változatlanok. A sorjátékosnak viszont most már 6 stratégiája van: 457, 475, 547, 574, 745 és 754. Legyenek ezek rendre a sorjátékos első, második, ..., hatodik stratégiája. Kis számolás után megkaphatjuk a kifizetési mátrixot:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \\ 2 & 1 & 2 \end{pmatrix}$$

Könnyen látható, hogy a sorjátékos (pl.) második stratégiája dominálja az első és ötödik stratégiát is. Iletve a sorjátékos számára a harmadik és negyedik stratégia megegyezik, ezek közül elég egyet szerepeltetni:

$$\begin{pmatrix} \cancel{1} & \cancel{1} & \cancel{1} \\ 1 & 2 & 2 \\ \cancel{2} & \cancel{2} & \cancel{1} \\ 2 & 2 & 1 \\ \cancel{1} & \cancel{1} & \cancel{1} \\ 2 & 1 & 2 \end{pmatrix}$$

Ez egyértelműség kedvéért külön is leírjuk a redukált kifizetési mátrixot, a továbbiakban ezt a mátrixot tekintjük csak.

$$\begin{pmatrix} 1 & 2 & 2 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \end{pmatrix}$$

Nyilván ezt az eredményt egy kis okoskodással is ki lehet hozni. Egyszerűen arról van szó, hogy a sorjátékosnak a 4-es lapját nem érdemes közvetlenül az 5-ös lap előtt kijátszania: a 4-es lapot az oszlopjátékos biztos elüti, így nála lesz a hívás; ezért az 5-ös lap sohasem üthet. 5-ös lapot csak akkor érdemes hívnia, ha nála van a hívás (induláskor, vagy közvetlenül a 7-es lap kijátszása után).

Könnyen ellenőrizhetjük, hogy a játékban nincs nyeregpont, nincs olyan érték, ami sorában minimális, oszlopában maximális. A sorjátékos biztonsági szintje 1 (maximin választás értéke), az oszlopjátékos biztonsági szintje 2 (minimax választás értéke). Ebben az esetben nem egyezik meg a két biztonsági szint, ezt úgy mondjuk, hogy *tiszta stratégiák* esetén nincs Nash-egyensúly. Tiszta stratégián azt értjük, hogy a lehetséges stratégiák közül kiválasztunk egyet, és azt játsszuk (1 valószínűséggel).

9.5. példa. *A 9.4. példa esetén van-e optimum kevert stratégia esetén?*

Megoldás

Először a *kevert stratégiákat* kell megértenünk. Gondoljunk bele egy kő-papír-olló játékba. Mindenki tudja, hogy ebben a játékban 1/3-1/3 valószínűséggel kell követ, papírt és ollót mutatni. Lényegileg ez a kevert stratégia, azaz megadunk egy valószínűség-eloszlást a stratégiák felett. Tehát például a sorjátékos

$$\begin{pmatrix} 0{,}2 \\ 0{,}7 \\ 0{,}1 \end{pmatrix}$$

(kevert) stratégiája azt jelenti, hogy az első stratégiáját játssza 0,2 (20%) valószínűséggel, a másodikat 0,7, a harmadikat pedig 0,1 valószínűséggel. Természetesen valamelyik stratégiát játszania kell, tehát a valószínűségek összegének 1-nak kell lenni. Jelen esetben 0,2+0,7+0,1=1.

Kevert stratégiák esetén a játékosok a várható kifizetést tekintik. Tehát ha a sorjátékos a fentebb megadott (kevert) stratégiáját játssza, akkor várható kifizetése rendre 0,2*1+0,7*2+0,1*2=1,8 (első oszlop), 0,2*2+0,7*2+0,1*1=1,9 (második oszlop) 0,2*2+0,7*1+0,1*2=1,3 (harmadik oszlop). Tehát a sorjátékos várható kifizetése a megadott (kevert) stratégia esetén: (1,8; 1,9; 1,3). Látható, hogy ha ezt a stratégiát játssza, akkor a sorjátékos biztonsági szintje 1,3-ra változik.

Nézzük az oszlopjátékost is. A példa kedvéért az oszlopjátékos a (0,4; 0,4; 0,2) stratégiáját játssza. Ekkor az oszlopjátékos várható kifizetése

$$\begin{pmatrix} 1{,}6 \\ 1{,}8 \\ 1{,}6 \end{pmatrix}$$

lesz, tehát az oszlopjátékos biztonsági szintje is lecsökken a korábbi 2-ről 1,8-ra. Tehát kevert stratégiák esetén a játékosok lehetősége megnő. Számítsuk ki a biztonsági szintjüket. Mivel végtelen sok (pontosan kontinuum sok) lehetőség van, ezeket egyesével végignézni nincs sok értelme, de fel tudunk írni egy LP feladatot, ami megadja a játékosok biztonsági szintjeit.

Érdemes az oszlopjátékos feladatát tekinteni először. Az oszlopjátékos `p1`, `p2` és `p3` valószínűséggel játssza az első, második és harmadik stratégiáját. Természetesen `p1`, `p2` és `p3` változóknak valószínűségeloszlást kell, hogy adjanak, tehát `p1+p2+p3=1`.

Jelölje `v` az oszlopjátékos biztonsági szintjét. Ez az érték azt jelent, hogy az oszlopjátékos várható kifizetése mindig ez alatt van, függetlenül attól, hogy melyik stratégiáját játssza a sorjátékos. Ha a sorjátékos minden tiszta stratégiája esetén igaz lesz, hogy az oszlopjátékos várható kifizetése `v` alatt marad, akkor a sorjátékos bármilyen kevert stratégiájára is igaz lesz.

Ha az oszlopjátékos `p1`, `p2` és `p3` valószínűségekkel játssza a stratégiáit, akkor a várható kifizetése `p1+2p2+2p3` lesz, ha a sorjátékos az első stratégiáját játssza. Ennek az értéknek kisebbnek kell lennie, mint `v`, különben `v` nem lenne biztonsági szint. Matematikailag: `p1+2p2+2p3<=v`, amit nyilván a `p1+2p2+2p3-v<=0` formában kell szerepeltetni. Hasonló korlátot kapunk a sorjátékos második és harmadik stratégiája esetén is. Az oszlopjátékos szeretné a legjobb (jelen esetben legkisebb) biztonsági szintet megkapni. Most már fel tudjuk írni az LP feladatot:

9.6. kód.
```
min
v

subject to
s1: 1p1 + 2p2 + 2p3 - v <= 0
s2: 2p1 + 2p2 + 1p3 - v <= 0
s3: 2p1 + 1p2 + 2p3 - v <= 0

ve:  p1 +  p2 +  p3      = 1

end
```

A megoldás:

```
Status:      OPTIMAL
Objective:   obj = 1.666666667 (MINimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 s1           NU             0     -0.333333
     2 s2           NU             0     -0.333333
     3 s3           NU             0     -0.333333
     4 ve           NS             1       1.66667

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 v            B        1.66667
     2 p1           B       0.333333
     3 p2           B       0.333333
     4 p3           B       0.333333
```

Az optimális megoldást ebben a feladatban akár ki is lehetne találni: az oszlopjátékosnak 1/3 valószínűséggel kell játszania mindhárom stratégiáját, ekkor a biztonsági szint, amit el tud érni 5/3.

Nézzük a sorjátékost. Igazából a sorjátékosra nem kell újabb LP feladatot felírni, a sorjátékos kevert stratégiáját, és biztonsági szintjét az oszlopjátékosra felírt LP feladatból le tudjuk olvasni. Azért, hogy lássuk ezt, érdemes kicsit átalakítani a feladatot. Induljunk ki a korlátokból:

```
1p1 + 2p2 + 2p3  <= v
2p1 + 2p2 + 1p3  <= v
2p1 + 1p2 + 2p3  <= v

 p1 +  p2 +  p3   = 1
```

Minden korlátot osszunk v-vel (így átmenetileg nem lesznek a korlátjaink lineárisok, de kezelni fogjuk ezt a helyzetet):

```
1p1/v + 2p2/v + 2p3/v  <= 1
2p1/v + 2p2/v + 1p3/v  <= 1
2p1/v + 1p2/v + 2p3/v  <= 1

 p1/v +  p2/v +  p3/v   = 1/v
```

A célfüggvényben `v-t` minimalizáljuk, ami ugyanaz, mintha `1/v-t` maximalizálnánk. Vezessük be továbbá a `p1/v` kifejezésre az `x1` változót, `p2/v` kifejezésre az `x2` változót, stb... Az `1/v` kifejezést szeretnénk maximalizálni, de `1/v` kifejezés egyenlő `p1/v+p2/v+p3/v` kifejezéssel, ami a bevezetett jelöléssel `x1+x2+x3`. Így már fel is tudunk írni egy újabb LP feladatot:

9.7. kód.

```
max
 x1 +  x2 +  x3

subject to
s1: 1x1 + 2x2 + 2x3 <= 1
s2: 2x1 + 2x2 + 1x3 <= 1
s3: 2x1 + 1x2 + 2x3 <= 1

end
```

A 9.7. Kódban szereplő feladat már egy normál feladat, aminek a duálját könnyű lesz felírni. Mielőtt ezt megtennénk, vessünk egy pillantást, a feladat optimális megoldására:

```
Status:     OPTIMAL
Objective:  obj = 0.6 (MAXimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 s1           NU             1           0.2
     2 s2           NU             1           0.2
     3 s3           NU             1           0.2

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 x1           B            0.2
     2 x2           B            0.2
     3 x3           B            0.2
```

Most a célfüggvény optimális értéke a biztonsági szint reciproka, tehát a biztonsági szint a célfüggvény optimális értékének reciproka lesz: $\frac{1}{0{,}6} = \frac{5}{3}$ ugyanúgy, mint korábban.

Hasonlóan `x1` értéke az első stratégia valószínűsége osztva a biztonsági szinttel. Átrendezéssel megkaphatjuk az első stratégia valószínűségét: `x1` optimális értéke osztva a biztosági szinttel. $\frac{0{,}2}{0{,}6} = \frac{1}{3}$, megint csak úgy, mint korábban. Hasonlóan meg lehet kapni a második és harmadik stratégia valószínűségeit is, amik megint csak $\frac{1}{3}$ lesznek.

Ahhoz, hogy megkapjuk az oszlopjátékos biztonsági szintjét, a duálfeladat felírása szükséges. A 9.7. Kódban szereplő LP feladat normál feladat, így a duálja viszonylag könnyen felírható:

9.8. kód.
```
min
 y1 +  y2 +  y3

subject to
o1: 1y1 + 2y2 + 2y3 >= 1
o2: 2y1 + 2y2 + 1y3 >= 1
o3: 2y1 + 1y2 + 2y3 >= 1

end
```

Ahhoz, hogy világosabb legyen számunkra az, amit kaptunk, megint csak végezzünk pár átalakítást: a célfüggvényben szereplő kifejezés legyen `1/w`. Szorozzuk be a korlátokat ezzel a `w` értékkel:

```
1y1w + 2y2w + 2y3 >= w
2y1w + 2y2w + 1y3 >= w
2y1w + 1y2w + 2y3 >= w

 y1w +  y2w +  y3  = 1
```

Ezek után vezessük be a `q1=y1*w` változótranszformációt. Ekkor eljutunk egy újabb LP feladathoz:

9.9. kód.
```
max
w

subject to
o1: 1q1 + 2q2 + 2q3 - w>= 0
o2: 2q1 + 2q2 + 1q3 - w>= 0
o3: 2q1 + 1q2 + 2q3 - w>= 0
```

```
ve:  q1 +  q2 +  q3      = 1

end
```

Ezzel megkaptuk a sorjátékos biztonsági szintjét leíró LP feladatot. A feladatban `w` a biztonsági szint, ennél az értéknél a sorjátékos kifizetése nem lesz kisebb, mindegy melyik stratégiáját játssza az oszlopjátékos. Tehát az oszlopjátékosra felírt feladatból leolvasható a sorjátékos feladatának optimális értéke is. Nézzük először a 9.7. Kódhoz tartozó LP feladat optimális megoldását. A biztonsági szint értéke megint csak a célfüggvény optimális értékének reciproka, $\frac{5}{3}$. A duálváltozók értékei a `Marginal` oszlopban találhatóak most is. Az `s1` korláthoz tartozó duálváltozó értéke (ami a mi jelölésünkben `y1`) 0,2, ami nem más mint `q1*w`, amiből `q1` értéke $\frac{1}{3}$. Hasonlóan megkapható, hogy a második és harmadik stratégiát is $\frac{1}{3}$-$\frac{1}{3}$ valószínűséggel kell játszani.

Annyit érdemes még megjegyezni, hogy ezek az értékek szerepelnek már a 9.11. Kódhoz tartozó LP feladatban is. Vegyük észre, hogy ebben a feladatban az `s1` korláthoz tartozó duálváltozó értéke $\frac{1}{3}$ és a többi duálváltozónak is ennyi lesz az értéke.

Tehát az erős dualitási tételből kifolyólag a sorjátékos és oszlopjátékos biztonsági szintje megegyezik, tehát a maximin és a minimax választás érétke ugyanaz lesz, ami azt is jelenti, hogy kevert stratégiák esetén mindig van Nash-egyensúly. Ezt az állítást általánosan is kijelenthetjük: mátrixjátékok estén a kevert stratégiák halmazán mindig van Nash-egyensúly. Az optimális stratégiák pedig elolvashatóak a megoldott LP feladatokból.

9.10. példa. *Tekintsük az alábbi kifizetési mátrixot (amihez szöveges magyarázatot nem adunk):*

$$\begin{pmatrix} -5 & -2 & 4 \\ 4 & -1 & 3 \\ 0 & 6 & -4 \\ -2 & -3 & 7 \end{pmatrix}$$

Adjuk meg a Nash-egyensúlyi stratégiákat!

Megoldás

LP feladatok segítségével mindig megtalálható a Nash-egyensúlyi stratégia, akkor is, ha tiszta stratégiák halmazán van Nash-egyensúly. Ezért egyből nekiállunk az LP feladat felírásának. De a kódolás megkezdése előtt érdemes felhívni a figyelmet arra, hogy a mátrixban negatív értékek is szerepelnek. Ez azért probléma, mert a Nash-egyensúly értéke így lehet negatív szám, tehát a biztonsági szint is negatív lesz (ami egyébként az oszlopjátékosnak nyereséget jelent). Ha a biztonsági szint negatív, akkor a v változónak is negatívnak kellene lennie, de LP feladatban a döntési változók nemnegatívak, így az LP feladat optimális megoldása nem a Nash-egyensúly értékét adja meg. Szerencsére a helyzet könnyen kezelhető. A mátrix minden eleméhez ugyanazt a konstans értéke adjuk, akkor az optimális stratégiák nem változnak. Esetünkben adjunk hozzá a kifizetési mátrix minden eleméhez 5-t, így már csak nemnegatív értékek szerepelnek benne:

$$\begin{pmatrix} 0 & 3 & 9 \\ 9 & 4 & 8 \\ 5 & 11 & 1 \\ 3 & 2 & 12 \end{pmatrix}$$

Ezen kifizetési mátrix esetén már fel lehet írni a LP feladatot, a biztonsági szint biztos, hogy nemnegatív lesz.

9.11. kód.

```
min
v

subject to
s1:  0p1 +  3p2 +  9p3 - v <= 0
s2:  9p1 +  4p2 +  8p3 - v <= 0
s3:  5p1 + 11p2 +  1p3 - v <= 0
s4:  3p1 +  2p2 + 12p3 - v <= 0
ve:  p1 +  p2 +  p3      = 1

end
```

A megoldás:

```
Status:     OPTIMAL
Objective:  obj = 6.213114754 (MINimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 s1           B       -1.09836
     2 s2           NU             0     -0.409836
     3 s3           NU             0     -0.377049
     4 s4           NU             0     -0.213115
     5 ve           NS             1       6.21311

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 v            B        6.21311
     2 p1           B       0.114754
     3 p2           B        0.47541
     4 p3           B       0.409836
```

A (módosított) játék értéke 6,213114754, de ne felejtsük el, hogy a kifizetési mátrix minden eleméhez hozzáadtunk 5-t, tehát az eredeti játék esetén a játék értéke 6,213114754-5=1,213114754. Az az érték pozitív, ami azt jelenti, hogy a játék a sorjátékosnak kedvező. Az oszlopjátékos optimális kevert stratégiája: az első stratégiát kb. 11,48% valószínűséggel játssza, a másodikat 47,54% valószínűséggel a harmadikat pedig 40,98%-kal. A sorjátékos optimális stratégiáját a duálfeladat adja meg. A duálváltozók optimális értékét a primál feladatból is leolvashatjuk, a `marginal` oszlopból: az első startégiát 0 valószínűséggel játssza, a másodikat kb. 40,98%-kal, a harmadikat 37,70%-kal, a negyediket pedig 21,31%-kal.

Megoldás #2

A 9.11. kód esetén a kifizetési mátrixot átírtuk, hogy biztosítsuk a LP változók nemnegativitását. 'Alapesetben' az LP feladatokat úgy szoktuk tárgyalni, hogy a döntési változók nemnegatívok. De lehetséges az LP feladat olyan bővítési is, ahol bizonyos változók előjelkötetlenek. A CPXLP kód esetén ezt úgy érhetjük el, hogy a `bound` szekcióban megadjuk mely változók előjelkötetlenek. Jelen esetben a `p` változók valószínűséget fejeznek ki, ezek mindig nemnegatívak. A `v` változó pedig a biztonságiszint értéke, ami előjelkötetlen. Ebben az esetben a kód:

9.12. kód.

```
min
```

```
v

subject to
s1:  - 5p1 - 2p2 +  4p3 - v <= 0
s2:  + 4p1 -  p2 +  3p3 - v <= 0
s3:  + 0p1 + 6p2 -  4p3 - v <= 0
s4:  - 2p1 - 3p2 +  7p3 - v <= 0
ve:  p1 +  p2 +  p3      = 1

bound
v free

end
```

Az eredmény ugyanaz, mint korábban, egyedül a célfüggvény értéke változott, most ‘egyből’ megkapjuk a 1,213114754 értéket.

9.13. példa. *A 9.10. feladatban megadott kifizetési mátrix esetén a játék a sorjátékosnak kedvez. Mennyivel kell csökkenteni az első sor első sor első elemét, hogy a játék igazságossá váljék? Hát a második sor első elemét?*

Megoldás

Az első kérdés könnyen megválaszolható. Az első stratégiáját a sorjátékos 0 valószínűséggel játssza. Ha csökkentjük ebben a cellában a játék értékét, akkor továbbra is 0 valószínűséggel fogja játszani ezt a stratégiát, tehát semmi érdemi változás nem történik (nem változik a játék értéke). Az első sor első cellában szereplő érték változtatásával nem tudjuk igazságossá tenni a rendszert.

A második sor első cellája már izgalmasabb kérdés, hiszen a második stratégiát mind a két játékos pozitív valószínűséggel játssza. Első ránézésre, ez is egy érzékenységvizsgálati határnak tűnik. Az részben igaz is, de a kifizetési mátrix második sorának első cellájában szereplő érték a LP feladat ‘A’ mátrixának eleme. Ezekre az értékekre nem tudunk érzékenységvizsgálati határokat meghatározni. ‘Próbálgatással jutunk csak eredményre. Ha cella értéke 0-vá válik, akkor a játék értéke is 0 lesz, azaz fair lesz a játék.

9.2. Gyakorló feladatok

9.2.1. Egyszerűbb feladatok

9.14. példa. *A következő kifizetésmátrixú konstans 10 összegű játékot játszák ketten:*

	X	Y	Z
A	5	4	8
B	5	6	6
C	3	6	4

Ha mind a két játékos racionális, akkor a domináns stratégiákat figyelembe véve mi lesz a játék kimenetele?

Megoldás a 360. oldalon.

9.15. példa. *Két játékos a következő játékot játssza: Mind a ketten mondanak egy számot 1 és 3 között. Ha ugyanazt mondják, akkor az oszlopjátékos kifizeti sorjátékosnak a bemondott számot. Ha 1 a különbség, akkor pedig az oszlopjátékosnak fizet a sorjátékos, a nagyobb számmal megegyező összeget. Ha a különbség kettő, akkor az oszlopjátékos fizet a sorjátékosnak kettőt.*

Egy LP modell felírásával határozza meg, hogy melyik játékosnak kedvez ez a játékszabály.

Megoldás a 361. oldalon.

9.16. példa. *Határozza meg a következő konstans 100 összegű játék kevert stratégiáját egy LP modell segítségével:*

	1	2	3	4	5
A	80	40	20	50	80
B	10	90	60	90	30
C	50	40	40	60	50
D	50	90	10	90	60
E	40	10	80	90	50

Mekkora valószínűséggel fognak 50-50 arányban osztozni a 100 összegen?

Megoldás a 362. oldalon.

9.2.2. Nehezebb feladatok

9.17. példa. *András és Bálint egy módosított kő-papír-olló játékot játszanak. A játék zéróösszegű, azaz amit az egyik játékos megnyer a másik elveszít. Az eredeti választások (kő, papír, olló) esetén a nyertes 100 Ft-ot kap.*

Bálint a sima kő/papír/olló jel felmutatása mellett még további három lehetséges akcióval rendelkezik: mutathat egyszerre két jelet is (tehát kő+ollót, kő+papírt, és olló+papírt). Ha olyan kombinációt választ, ami nem tartalmaz olyan jelet, amit András felmutatott, akkor nyer 200 Ft-ot, különben 50 vagy 0 Ft-ot kap annak függvényében, hogy az egyik általa mutatott jel le tudja-e győzni András által mutatott jelet (pl. kő+olló legyőzi az ollót (50), de a követ nem (0)).

Andrásnak van egy további választási lehetősége, a Kő, Papír és az Ollón túl: a Joker. A Joker stratégia esetén András 400 Ft-ot nyer, ha Bálint egy kombinációt mutat. Viszont, ha Bálint csak egy lehetőséget választ, akkor a Joker esetén András fizet 100 Ft-ot Bálintnak. Adja meg a játékosok kevert stratégiáit és a várható kifizetéseit.

Megoldás a 363. oldalon.

9.18. példa. *Egy futballcsapat a következő meccsének a stratégiáját játékelméleti módszerekkel akarja meghatározni. Arról akarnak dönteni, hogy a két félidőben támadó vagy védekező stratégiával játszanak. A következő táblázat mutatja meg, hogy feltételezéseik szerint milyen kimenetre számítanak a különböző stratégiák kiválasztása esetén.*

stratégia		győzelmi esély	
saját	ellenfél	1.félidő	2.félidő
Támadó	Támadó	50	50
Támadó	Védekező	70	40
Védekező	Védekező	50	50
Védekező	Támadó	30	60

Ezeket az értékeket úgy kell értelmezni, hogy például, ha az első félidőben a saját csapat támadó stratégiát választja, az ellenfél pedig védekező stratégiát, akkor 70% esélyét látják arra, hogy a saját csapat jobban fog állni a félidő végén. A második *félidőben fáradtabbak a játékosok, ezért a támadó stratégia a védekezővel szemben csak 40% esélyt ad a vezetésre. A csapat célja, hogy az összesített győzelmi esélyt maximalizálják, amit a két félidő esélyeinek az átlagolásával határoznak meg.*

A csapatnak lehetősége van az eredeti felállástól eltérni. Több csatárt is tud alkalmazni (A felállás), ekkor az első félidőben a támadás stratégia 10 százalékponttal hatékonyabb, viszont a védekezés 10 százalékponttal gyengébb mind a két félidőben. A második félidőben a támadás is gyengébb ebben az esetben 5 százalékponttal.

Ezen kívül akár több hátvéddel is játszhatnak (B felállás), ekkor az első félidőben támadás gyengébb 5 százalékponttal, de a védekezés mind a két félidőben erősebb 5 százalékponttal.

Határozza meg LP modell felírásával, hogy melyik stratégiát használja a csapat és melyik felállást (eredeti, A vagy B).

Megoldás a 365. oldalon.

10. fejezet

Megoldások

10.1. Lineáris programozás emlékeztető

2.21. példa megoldása.

Jelölje `x1`, `x2`, `x3` és `x4` a négy termék arányát 1 gramm új termékben. Nyilván ezeknek az összege pontosan 1:

```
gr:    +   x1 +   x2 +   x3 +   x4  =   1
```

Ezen kívül két alsó korlátunk van: egy a kalciumra (`Ca`), és egy a magnéziumra (`Mg`):

```
Ca:    +140x1 +105x2 + 99x3 + 90x4 >= 100
Mg:    + 65x1 + 72x2 +100x3 +123x4 >=  90
```

A foszfor (`P`) értéke az új termékben a 85-95 mg közötti intervallumban mozoghat, tehát egy alsó- és egy felső korlátot kell a modellhez adni:

```
P_a:  +152x1 +115x2 + 81x3 + 98x4 >=  85
P_f:  +152x1 +115x2 + 81x3 + 98x4 <=  95
```

A K-vitamin esetében a korlát mértékegysége ugyan eltér a korábbi korlátokétól, de ez a felírást nem módosítja:

```
Kvit: + 71x1 + 42x2 + 41x3 + 83x4 >=  60
```

A célfüggvényben az egy grammra jutó előállítási költséget minimalizáljuk:

```
Min
+67x1 +83x2 +68x3 +55x4
```

Természetesen az összes változó nemnegatív, viszont a CPLEX-LP formátumban ezt nem kell külön kikötni. Tehát a feladat teljes kódja:

10.1. kód.

```
Min
        67x1 + 83x2 + 68x3 + 55x4

Subject to
gr:   +   x1 +   x2 +   x3 +   x4  =   1
Ca:   +140x1 +105x2 + 99x3 + 90x4 >= 100
Mg:   + 65x1 + 72x2 +100x3 +123x4 >=  90
P_a:  +152x1 +115x2 + 81x3 + 98x4 >=  85
P_f:  +152x1 +115x2 + 81x3 + 98x4 <=  95
Kvit: + 71x1 + 42x2 + 41x3 + 83x4 >=  60

end
```

Az optimális megoldás:

```
Objective:  obj = 62.99850299 (MINimum)

   No.   Row name    St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 gr           NS             1       33.7246
     2 Ca           NL           100      0.678144
     3 Mg           B        104.913
     4 P_a          B             95
     5 P_f          NU            95     -0.405689
     6 Kvit         B         60.024

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 x1           B       0.107036
     2 x2           NL             0       24.7246
     3 x3           B       0.516467
     4 x4           B       0.376497
```

A Sensitivity Analysis (érzékenységvizsgálati jelentés) fontosabb oszlopai:

```
Row name      St      Activity         Slack       Activity
                                    Marginal          range
```

```
----------- -- ------------ ------------ ------------
gr          NS      1.00000       .             .99977
                              33.72455        1.05929

Ca          NL    100.00000       .           91.58824
                                .67814      100.01294

Mg          BS    104.91317    -14.91317    104.91317
                                  .         104.89159

P_a         BS     95.00000    -10.00000     95.00000
                                  .          94.98394

P_f         NU     95.00000       .          94.98394
                               -.40569      108.80000

Kvit        BS     60.02395      -.02395     60.02395
                                  .          45.11377


Column name  St     Activity     Obj coef     Obj coef
                                 Marginal        range
----------- -- ------------ ------------ ------------
x1          BS       .10704     67.00000     13.70588
                                  .         127.22222

x2          NL        .         83.00000     58.27545
                              24.72455          +Inf

x3          BS       .51647     68.00000     57.16000
                                  .             +Inf

x4          BS       .37650     55.00000        -Inf
                                  .          67.76056
```

a) Egy gramm előállítának az optimuma 62,99850299, ezt 1,8-cal beszorozva megkapjuk 1 tablettának az előállítási költségét, ami tehát 113,4 Ft.

b) Mivel az optimális megoldásban a foszfor felső korlátja aktív, ezért az alsó korlátozás változása nem fogja befolyásolni az optimumot. A felső korlát aktív feltétel, tehát egységnyi növelése esetén a korlát árnyékárát kell figyelembe venni, ami `-.40569`. A korlát addig marad aktív, amíg a felső korlátozás a 108,8-as foszfor mennyiséget el nem éri (jelen esetben 96 lesz ezen érték maximuma). Mivel az árnyékárat egy mg növelése esetén tudjuk értelmezni, így az egy tablettára jutó költség $0,40569 \times 1,8$ értékkel fog csökkenni, azaz 112,67 Ft lesz.

c) Az alsó korlátot ebben az esetben sem kell figyelembe venni, mivel nem aktív feltétel. A foszfor felső korlátját azonban csak $108,8$-ig tudjuk ezzel az érzékenységvizsgálattal vizsgálni, tehát a 115 mg-os felső korlát esetén a jelenleg használt árnyékár nem lesz érvényes, azaz a költség nem $0,40569 \times (115 - 95)$ értékkel fog csökkenni. Ezen kérdés megválaszolásához a modellt mindenképpen újra kell futtatni.

   Állítsuk a modellben a foszfor felső korlátjának jobb oldalát 115-re, és futtassuk újra a modellt. Ekkor a következő eredményt kapjuk:

```
Objective:  obj = 57.4 (MINimum)

   No.   Row name   St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 gr           NS             1          33.4
     2 Ca           NL           100          0.24
     3 Mg           B          111.4
     4 P_a          B          108.8
     5 P_f          B          108.8
     6 Kvit         B           80.6

   No. Column name  St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 x1           B            0.2
     2 x2           NL             0          24.4
     3 x3           NL             0         10.84
     4 x4           B            0.8
```

   Tehát egy tabletta ára $57,4 \times 1,8 = 103,32$ lesz. Az eredményből látható, hogy ekkor a foszfor felső korlátja nem aktív feltétel.

d) Az eredeti esetben a magnézium nem aktív feltétel, az új termék 1 grammjába összesen 104,913 mg magnézium kerül, miközben a korlát arra vonatkozik, hogy legalább 90-nek kell benne lennie. Ha a korlát jobb oldalát 10 egységgel növeljük, a feltétel akkor sem lesz aktív, tehát az optimumot nem fogja megváltoztatni.

e) A második terméket jelenleg nem használjuk fel az új termék keverésénél, azonban az előállítási költség csökkentése eredményezheti azt, hogy ez a változó is bekerüljön a bázisba. Azt, hogy meddig kell a költséget ehhez csökkenteni, az eredményben a redukált költség mutatja meg (`marginal` oszlop), ami a második változó esetén $24,7246$. Tehát ahhoz, hogy a második terméket is felhasználjuk az új termék keverésénél, legalább $24,7246$ forintos árcsökkenés szükséges. Mivel $20 < 24,7246$, ezért a második termék áresése nem fogja csökkenteni az új termék előállítási költségét.

Amennyiben a költség jobban csökkenne, mint $24,7246$, akkor az új viszonyokról csakis a modell újrafuttatása esetén tudnánk biztosat mondani.

2.22. példa megoldása.

Ez a feladat a 2.1 példához hasonlóan egy termelési feladat. Legyenek az `x1,x2,x3>= 0` folytonos változók rendre az alma, a barack és az alma-barack üdítőkből készített mennyiségek, amelyek a napi termelést adják meg literben.

Az LP felírása esetén négy termelési korlátot kell figyelembe venni: cukorra, vízre, almasűrítményre és baracksűrítményre vonatkozót. A víz és sűrítmények esetén az 1 liter üdítőkhöz szükséges mennyiségek ml-ben vannak megadva, ezért a korlátozó mennyiségeket is át kell alakítani literből ml-re.

Ezeken kívül szükségünk van továbbá a minimális gyártási feltételeket leíró egyenletekre. A célfüggvényben a bevételt maximalizáljuk. Tehát a kód:

10.2. kód.

```
max
bev:   +300x1 +380x2 +400x3

Subject to
cukor: + 10x1 +   8x2 +   7x3 <=  30000
viz:   +770x1 +650x2 +450x3 <= 150000
alma:  +230x1        +260x3 <=  60000
barack:       +350x2 +290x3 <=  40000
```

```
m1:    +  1x1              >=     20
m2:          +  1x2        >=     20
m3:                 +  1x3 >=     20
end
```

Az optimális megoldás pedig:

```
Objective:  bev = 86543.12584 (MAXimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 cukor        B        2070.75
     2 viz          NU        150000       0.38961
     3 alma         B        55212.7
     4 barack       NU         40000      0.774742
     5 m1           B         111.42
     6 m2           NL            20      -144.407
     7 m3           B        113.793

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 x1           B         111.42
     2 x2           B             20
     3 x3           B        113.793
```

Amennyiben a feladatban arra keressük a választ, hogy 1 literes palackokban mennyit termeljünk, akkor a változókról fel kell tenni az egészértékűséget, tehát a kódhoz hozzá kell adni az

```
integer
 x1
 x2
 x3
```

sorokat.

Ebben az esetben az eredmény:

```
Objective:  bev = 86100 (MAXimum)

   No.   Row name        Activity
------ ------------    -------------
     1 cukor                    2061
     2 viz                    149320
     3 alma                    54910
```

```
     4 barack                       39770
     5 m1                             111
     6 m2                              20
     7 m3                             113

   No. Column name        Activity
------ ------------     -------------
     1 x1           *             111
     2 x2           *              20
     3 x3           *             113
```

Tehát az első, folytonos esetben a bevétel 86543,12584 forint lett, míg az egészértékű megoldás esetén 86100 forint. A folytonos esetben rendre (111,42; 20; 113,793) litert kell készíteni az üdítőkből, míg a második esetben (111; 20; 113) palackot.

Ha a `barack:` korlát jobb oldalát 5000 egységgel növeljük, akkor ennek a változásnak a bevételre gyakorolt hatását az árnyékárból tudjuk megmondani, amennyiben a korlát aktív marad. Ezt az érzékenységvizsgálati jelentés mutatja:

```
Row name      St       Activity          Slack      Activity
                                       arginal         range
------------ -- ------------- ------------- -------------
barack        NU   40000.00000           .       12800.00000
                                     .77474    51054.80869
```

Az érzékenységvizsgálati jelentés alapján a korlát jobb oldalát `51054.80869`-as értékig lehet növelni, addig aktív marad. Tehát amennyiben a baracksűrítményből 45 liter állna rendelkezésre, akkor a bevétel $86543,12584 + 5000 \times 0,77474 = 90416,83584$ Ft lenne.

Az egészértékű modell esetén nem tudjuk meghatározni az árnyékárat, sőt az érzékenységvizsgálati jelentést sem tudjuk elkészíteni. A feladat eredménytáblázatában látható, hogy ez a korlát nem egyenlőségként teljesül, tehát $40000 - 39770 = 230$ ml baracksűrítmény marad a nap végén a raktárban. Azonban mivel a növekedés 5000 ml, ami a fenti kimaradó értéknél több, ezért a célfüggvény értékét biztosan növelni fogja. Azonban erről nem tudunk semmit sem a jelenlegi ismereteinkből, ezért a modellt újra kell futtatni. Ekkor a következő eredményt kapjuk:

```
Objective:  bev = 90300 (MAXimum)

   Row name        Activity     Lower bound   Upper bound
------------    ------------- ------------- -------------
cukor                  2087                         30000
viz                  149720                        150000
alma                  57290                         60000
barack                44990                         45000
m1                      101            20
m2                       20            20
m3                      131            20

Column name        Activity     Lower bound   Upper bound
------------    ------------- ------------- -------------
x1           *          101             0
x2           *           20             0
x3           *          131             0
```

Tehát ebben az esetben a bevétel 90300 ft lesz.

2.23. példa megoldása.

A következő bináris változókat használjuk:

- x_1: A vállalat az A televízióban hirdet.
- x_2: A vállalat az B televízióban hirdet.
- x_3: A vállalat a megyei rádióban hirdet.
- x_4: A vállalat az országos rádióban hirdet.
- x_5: A vállalat nyereményjátékot indít.
- x_6: A vállalat plakátokkal hirdet.
- x_7, x_8, x_9: A vállalat internetes hirdetései, növekvő költségek szerint.

Minden változó bináris, ahol az $x_i = 1$ jelenti az i-edik hirdetés elindítását. Az IP-ben az elérhető emberek számát szeretnénk maximalizálni:

$$\max \sum_{i=1}^{9} c_i x_i \ ,$$

ahol a c_i az i-edik hirdetés által elérhető emberek száma. Továbbá jelölje a_i az i-edik hirdetés költségét, és B azt az összeget, amit a vállalat a kampányra szán. Ekkor marketingkampány során a következő költségvetési korlátnak kell teljesülnie:

$$\sum_{i=1}^{9} a_i x_i \leq B \ .$$

A két televízióban egyszerre nem hirdethet a vállalat, azaz másképpen az x_1 és az x_2 egyszerre nem veheti fel az 1-es értéket:

$$x_1 + x_2 \leq 1 \ .$$

Ezen kívül pontosan egy internetes hirdetést választaniuk kell:

$$x_7 + x_8 + x_9 = 1 \ .$$

Amennyiben az A tv-ben hirdetnek, akkor a megyei rádióban is hirdetniük kell. Másképpen fogalmazva amennyiben $x_1 = 1$, akkor $x_3 = 1$, azonban ha $x_1 = 0$, akkor x_3 lehet nulla és egy is. Az IP-be ezt a következőképpen tudjuk implementálni:

$$x_1 \leq x_3 \ .$$

Amennyiben mind a két rádióban egyszerre hirdetnek, akkor a legdrágább internetes hirdetést nem választhatják. Tehát amennyiben $x_3 + x_4 = 2$, akkor $x_9 = 0$. Ehhez vezessük be az y_{34} bináris változót, ami akkor 1, ha $x_3 + x_4 = 2$. Tehát másképpen a két korlátot kell a modellhez adni:

$$2y_{34} \leq x_3 + x_4 \leq 1 + y_{34}$$

Ezzel a változóval pedig könnyen korlátozhatjuk az x_9 változót:

$$x_9 \leq 1 - y_{34}$$

Nyereményjátékot csak akkor indíthatnak, ha országos rádióban, és valamelyik televízióban hirdetnek. Tehát x_5 csak akkor lehet 1, ha $x_4 = 1$ és $x_1 + x_2 = 1$. Ehhez megint bevezetünk egy bináris segédváltozót: legyen $s_{12} = 1$, ha $x_1 + x_2 = 1$, különben pedig 0.

$$s_{12} \leq x_1 + x_2 \leq s_{12}$$

Ekkor tehát az $x_5 = 1$ csakis akkor teljesülhet, ha $x_4 + s_{12} = 2$, amit a következőképpen illeszthetünk a modellbe:

$$2x_5 \leq x_4 + s_{12} \leq 1 + x_5 \ .$$

A CPLEX-LP kód:

10.3. kód.

```
max
400x1 +340x2 +60x3 +200x4 +350x5 +130x6 +100x7 +220x8 +500x9

subject to
ktg: +15x1 +12x2 +3x3 +8x4 +16x5 +8x6 +5x7 +10x8 +20x9 <=50
tv: x1 +x2 <= 1
inter: x7 +x8 +x9 = 1
Ao: x1-x3 <= 0
yf:  x3 +x4     - y34 <= 1
ya: -x3 -x4     +2y34 <= 0
yx9:          x9 + y34 <= 1
sf:  x1 +x2         -s12 <= 0
sa:  x1 +x2         -s12 >= 0
sx4a:       x4 -2x5 +s12 >= 0
sx4f:       x4 - x5 +s12 <= 1

binary
x1 x2 x3 x4 x5 x6 x7 x8 x9
y34
s12

end
```

Az optimális megoldás pedig:

```
Objective:  obj = 1170 (MAXimum)

  Row name        Activity     Lower bound   Upper bound
```

```
------------    ------------- ------------- -------------
ktg                        49                          50
tv                          1                           1
inter                       1             1             =
Ao                         -1                           0
yf                          1                           1
ya                          0                           0
yx9                         1                           1
sf                          0                           0
sa                          0             0
sx4a                        0             0
sx4f                        1                           1

Column name        Activity     Lower bound   Upper bound
------------    ------------- ------------- -------------
x1           *              0             0             1
x2           *              1             0             1
x3           *              1             0             1
x4           *              1             0             1
x5           *              1             0             1
x6           *              0             0             1
x7           *              0             0             1
x8           *              1             0             1
x9           *              0             0             1
y34          *              1             0             1
s12          *              1             0             1
```

Várhatóan 1170 embert érhetnek el a marketingkampánnyal, és összesen 49 millió forintba fog kerülni.

2.24. példa megoldása.

Jelölje x_i az i-edik héten a Balaton vízállását mutató nemnegatív, folytonos változót. Valamint jelölje y_i az i-edik héten a Sió csatornával leengedett víz mennyiségét. Mivel ez a mennyiség minden időszakban legfeljebb 20 mm lehet, ezért:

$$y_i \leq 20 \quad i = 1, \ldots 52 \ .$$

Jelöljük a Zala folyó hozamát az i-edik héten z_i-vel, amely értékeket a feladat kiírásában megtalálunk. A Balaton vízállása az i-edik héten a korábbi hét vízállásával, a Zala folyó hozamával, és a Sió leengedésével egyezik meg:

$$x_i = x_{i-1} + z_i - y_i \quad i = 1, \ldots 52 \ ,$$

amit természetesen át kell rendeznünk úgy, hogy a változók az egyenlőség bal oldalán legyenek:

$$x_i - x_{i-1} + y_i = z_i \quad i = 1, \ldots 52 \ .$$

A 0-adik időszakban a Balaton a 3996-os vízszinten áll, tehát $x_0 = 3996$. A Balaton vízszint-ingadozásának modellezésére bevezetünk egy d változót is, amelyre a következő korlátoknak kell teljesülniük:

$$4000 - d \leq x_i \leq 4000 + d \quad i = 1, \ldots 52 \ ,$$

amelyek átrendezve:

$$x_i - d \leq 4000 \quad i = 1, \ldots 52$$

és

$$x_i + d \geq 4000 \quad i = 1, \ldots 52 \ .$$

A célfüggvényben egyedül a d változót kell minimalizálni. A CPLEX-LP kód:

10.4. kód.

```
Min
d

Subject to

y1k:  y1  <= 20
y2k:  y2  <= 20
y3k:  y3  <= 20
y4k:  y4  <= 20
y5k:  y5  <= 20
y6k:  y6  <= 20
y7k:  y7  <= 20
y8k:  y8  <= 20
y9k:  y9  <= 20
y10k: y10 <= 20
y11k: y11 <= 20
```

```
y12k: y12 <= 20
y13k: y13 <= 20
y14k: y14 <= 20
y15k: y15 <= 20
y16k: y16 <= 20
y17k: y17 <= 20
y18k: y18 <= 20
y19k: y19 <= 20
y20k: y20 <= 20
y21k: y21 <= 20
y22k: y22 <= 20
y23k: y23 <= 20
y24k: y24 <= 20
y25k: y25 <= 20
y26k: y26 <= 20
y27k: y27 <= 20
y28k: y28 <= 20
y29k: y29 <= 20
y30k: y30 <= 20
y31k: y31 <= 20
y32k: y32 <= 20
y33k: y33 <= 20
y34k: y34 <= 20
y35k: y35 <= 20
y36k: y36 <= 20
y37k: y37 <= 20
y38k: y38 <= 20
y39k: y39 <= 20
y40k: y40 <= 20
y41k: y41 <= 20
y42k: y42 <= 20
y43k: y43 <= 20
y44k: y44 <= 20
y45k: y45 <= 20
y46k: y46 <= 20
y47k: y47 <= 20
y48k: y48 <= 20
y49k: y49 <= 20
y50k: y50 <= 20
y51k: y51 <= 20
y52k: y52 <= 20

c0:  x0 = 3996
c1:  x1  -x0  +y1  = 21
c2:  x2  -x1  +y2  = 18
```

```
c3:  x3  -x2  +y3  = 16
c4:  x4  -x3  +y4  = 13
c5:  x5  -x4  +y5  = 12
c6:  x6  -x5  +y6  = 11
c7:  x7  -x6  +y7  = 14
c8:  x8  -x7  +y8  = 13
c9:  x9  -x8  +y9  = 26
c10: x10 -x9  +y10 = 33
c11: x11 -x10 +y11 = 37
c12: x12 -x11 +y12 = 40
c13: x13 -x12 +y13 = 31
c14: x14 -x13 +y14 = 34
c15: x15 -x14 +y15 = 11
c16: x16 -x15 +y16 = 13
c17: x17 -x16 +y17 =  7
c18: x18 -x17 +y18 = 10
c19: x19 -x18 +y19 =  8
c20: x20 -x19 +y20 = 13
c21: x21 -x20 +y21 = 15
c22: x22 -x21 +y22 = 10
c23: x23 -x22 +y23 = 19
c24: x24 -x23 +y24 = 20
c25: x25 -x24 +y25 = 24
c26: x26 -x25 +y26 = 23
c27: x27 -x26 +y27 = 15
c28: x28 -x27 +y28 = 17
c29: x29 -x28 +y29 = 29
c30: x30 -x29 +y30 = 28
c31: x31 -x30 +y31 = 21
c32: x32 -x31 +y32 = 10
c33: x33 -x32 +y33 = 16
c34: x34 -x33 +y34 = 14
c35: x35 -x34 +y35 = 12
c36: x36 -x35 +y36 = 28
c37: x37 -x36 +y37 = 25
c38: x38 -x37 +y38 = 16
c39: x39 -x38 +y39 = 18
c40: x40 -x39 +y40 = 19
c41: x41 -x40 +y41 = 17
c42: x42 -x41 +y42 = 11
c43: x43 -x42 +y43 = 13
c44: x44 -x43 +y44 = 18
c45: x45 -x44 +y45 = 13
c46: x46 -x45 +y46 = 11
c47: x47 -x46 +y47 = 10
```

```
c48: x48 -x47 +y48 = 12
c49: x49 -x48 +y49 =  8
c50: x50 -x49 +y50 = 12
c51: x51 -x50 +y51 = 13
c52: x52 -x51 +y52 = 16

x1_f:  x1  -d <= 4000
x2_f:  x2  -d <= 4000
x3_f:  x3  -d <= 4000
x4_f:  x4  -d <= 4000
x5_f:  x5  -d <= 4000
x6_f:  x6  -d <= 4000
x7_f:  x7  -d <= 4000
x8_f:  x8  -d <= 4000
x9_f:  x9  -d <= 4000
x10_f: x10 -d <= 4000
x11_f: x11 -d <= 4000
x12_f: x12 -d <= 4000
x13_f: x13 -d <= 4000
x14_f: x14 -d <= 4000
x15_f: x15 -d <= 4000
x16_f: x16 -d <= 4000
x17_f: x17 -d <= 4000
x18_f: x18 -d <= 4000
x19_f: x19 -d <= 4000
x20_f: x20 -d <= 4000
x21_f: x21 -d <= 4000
x22_f: x22 -d <= 4000
x23_f: x23 -d <= 4000
x24_f: x24 -d <= 4000
x25_f: x25 -d <= 4000
x26_f: x26 -d <= 4000
x27_f: x27 -d <= 4000
x28_f: x28 -d <= 4000
x29_f: x29 -d <= 4000
x30_f: x30 -d <= 4000
x31_f: x31 -d <= 4000
x32_f: x32 -d <= 4000
x33_f: x33 -d <= 4000
x34_f: x34 -d <= 4000
x35_f: x35 -d <= 4000
x36_f: x36 -d <= 4000
x37_f: x37 -d <= 4000
x38_f: x38 -d <= 4000
x39_f: x39 -d <= 4000
```

```
x40_f: x40 -d <= 4000
x41_f: x41 -d <= 4000
x42_f: x42 -d <= 4000
x43_f: x43 -d <= 4000
x44_f: x44 -d <= 4000
x45_f: x45 -d <= 4000
x46_f: x46 -d <= 4000
x47_f: x47 -d <= 4000
x48_f: x48 -d <= 4000
x49_f: x49 -d <= 4000
x50_f: x50 -d <= 4000
x51_f: x51 -d <= 4000
x52_f: x52 -d <= 4000

x1_a:  x1  +d >= 4000
x2_a:  x2  +d >= 4000
x3_a:  x3  +d >= 4000
x4_a:  x4  +d >= 4000
x5_a:  x5  +d >= 4000
x6_a:  x6  +d >= 4000
x7_a:  x7  +d >= 4000
x8_a:  x8  +d >= 4000
x9_a:  x9  +d >= 4000
x10_a: x10 +d >= 4000
x11_a: x11 +d >= 4000
x12_a: x12 +d >= 4000
x13_a: x13 +d >= 4000
x14_a: x14 +d >= 4000
x15_a: x15 +d >= 4000
x16_a: x16 +d >= 4000
x17_a: x17 +d >= 4000
x18_a: x18 +d >= 4000
x19_a: x19 +d >= 4000
x20_a: x20 +d >= 4000
x21_a: x21 +d >= 4000
x22_a: x22 +d >= 4000
x23_a: x23 +d >= 4000
x24_a: x24 +d >= 4000
x25_a: x25 +d >= 4000
x26_a: x26 +d >= 4000
x27_a: x27 +d >= 4000
x28_a: x28 +d >= 4000
x29_a: x29 +d >= 4000
x30_a: x30 +d >= 4000
x31_a: x31 +d >= 4000
```

```
x32_a: x32 +d >= 4000
x33_a: x33 +d >= 4000
x34_a: x34 +d >= 4000
x35_a: x35 +d >= 4000
x36_a: x36 +d >= 4000
x37_a: x37 +d >= 4000
x38_a: x38 +d >= 4000
x39_a: x39 +d >= 4000
x40_a: x40 +d >= 4000
x41_a: x41 +d >= 4000
x42_a: x42 +d >= 4000
x43_a: x43 +d >= 4000
x44_a: x44 +d >= 4000
x45_a: x45 +d >= 4000
x46_a: x46 +d >= 4000
x47_a: x47 +d >= 4000
x48_a: x48 +d >= 4000
x49_a: x49 +d >= 4000
x50_a: x50 +d >= 4000
x51_a: x51 +d >= 4000
x52_a: x52 +d >= 4000

end
```

a) A megoldásban `d = 40.5`, tehát a Balaton vízállása a $3959,5-4040,5$ mm-es intervallumban fog maradni.

b) Ez a kérdést másképpen úgy lehetne átfogalmazni, hogy melyik y_i változó kapacitásának egységnyi növelése csökkentené leginkább az optimális d értékét. Mivel az y_i kapacitását az

$$y_i \leq 20$$

korlátok adják meg, d értéke pedig megegyezik a célfüggvény értékével, ezért ezeknek a korlátoknak az árnyékárait kell figyelembe venni. A szivattyú használatát pedig arra hétre kell időzíteni, amikor ennek a korlátnak az árnyékára a legkisebb (=a legjobban csökken a célfüggvény értéke, azaz d).

A korlátok árnyékárai:

```
   No.   Row name    St   Activity       Marginal
------ ------------ -- ------------- -------------
```

1	y1k	NU	20	< eps
2	y2k	NU	20	< eps
3	y3k	NU	20	< eps
4	y4k	NU	20	< eps
5	y5k	NU	20	< eps
6	y6k	NU	20	< eps
7	y7k	NU	20	< eps
8	y8k	B	14.5	
9	y9k	NU	20	-0.5
10	y10k	NU	20	-0.5
11	y11k	NU	20	-0.5
12	y12k	NU	20	-0.5
13	y13k	NU	20	-0.5
14	y14k	NU	20	-0.5
15	y15k	NU	20	< eps
16	y16k	NU	20	< eps
17	y17k	NU	20	< eps
18	y18k	NU	20	< eps
19	y19k	NU	20	< eps
20	y20k	NU	20	< eps
21	y21k	NU	20	< eps
22	y22k	NU	20	< eps
23	y23k	NU	20	< eps
24	y24k	NU	20	< eps
25	y25k	NU	20	< eps
26	y26k	NU	20	< eps
27	y27k	NU	20	< eps
28	y28k	NU	20	< eps
29	y29k	NU	20	< eps
30	y30k	NU	20	< eps
31	y31k	NU	20	< eps
32	y32k	NU	20	< eps
33	y33k	NU	20	< eps
34	y34k	NU	20	< eps
35	y35k	B	16	
36	y36k	NU	20	< eps
37	y37k	NU	20	< eps
38	y38k	NU	20	< eps
39	y39k	NU	20	< eps
40	y40k	NU	20	< eps
41	y41k	NU	20	< eps
42	y42k	B	14	
43	y43k	B	13	
44	y44k	B	18	
45	y45k	B	13	

```
46 y46k          B              11
47 y47k          B              10
48 y48k          B              12
49 y49k          B               8
50 y50k          B              12
51 y51k          B              13
52 y52k          B              16
```

A táblázat alapján a 9-14 hét között javít a d értékén a szivattyú alkalmazása, minden héten ugyanakkora mértékben. Ezen hetek bármelyikén lehet a szivattyút alkalmazni.

c) A Zala folyó első heti vízhozama egyedül a `c1:` korlátban szerepel. Erre a kérdésre a korlát jobb oldalának megengedhető növelése és csökkentése ad választ, ami 19,5–26,5.

```
Row name  St   Activity          Slack      Activity
                              Marginal         range
--------- -- ---------- -------------  ------------
c1        NS   21.00000            .       19.50000
                                   .       26.50000
```

Tehát amíg ez a két érték között marad a vízhozam, addig az ütemterv sem változik.

d) Ebben az esetben a modellen változtatni kell, hozzáadunk a modellhez s_i bináris változókat. Amennyiben $s_i = 1$, akkor a Sió zsilipje nyitva van. Nyilván ebben az esetben $y_i > 0$, amit most az egyszerűbb felírás kedvéért $y_i \geq 0{,}001$-re egyszerűsítünk. Ekkor az y-ra a következő korlátok vonatkoznak:

$$0,001 s_i \leq y_i \leq 20 s_i \ .$$

(Az eredeti feladat megoldásából tudjuk, hogy valójában az alsó korlát jelen esetben nem szükséges, azonban ha más paraméterekre futtatnánk ezt a modellt, akkor külön ki kell kötni, hogy a zsilip nyitása pozitív mennyiséget jelent.)

Ezen kívül fel kell tenni, hogy a zsilipek nyitásának száma legfeljebb 5 lehet:

$$\sum_{i=1}^{50} s_i \leq 5 \ .$$

A módosított kód:

10.5. kód.

```
Min
d

Subject to


y1f:  y1  -20s1  <= 0
y2f:  y2  -20s2  <= 0
y3f:  y3  -20s3  <= 0
y4f:  y4  -20s4  <= 0
y5f:  y5  -20s5  <= 0
y6f:  y6  -20s6  <= 0
y7f:  y7  -20s7  <= 0
y8f:  y8  -20s8  <= 0
y9f:  y9  -20s9  <= 0
y10f: y10 -20s10 <= 0
y11f: y11 -20s11 <= 0
y12f: y12 -20s12 <= 0
y13f: y13 -20s13 <= 0
y14f: y14 -20s14 <= 0
y15f: y15 -20s15 <= 0
y16f: y16 -20s16 <= 0
y17f: y17 -20s17 <= 0
y18f: y18 -20s18 <= 0
y19f: y19 -20s19 <= 0
y20f: y20 -20s20 <= 0
y21f: y21 -20s21 <= 0
y22f: y22 -20s22 <= 0
y23f: y23 -20s23 <= 0
y24f: y24 -20s24 <= 0
y25f: y25 -20s25 <= 0
y26f: y26 -20s26 <= 0
y27f: y27 -20s27 <= 0
y28f: y28 -20s28 <= 0
y29f: y29 -20s29 <= 0
y30f: y30 -20s30 <= 0
y31f: y31 -20s31 <= 0
```

```
y32f: y32 -20s32 <= 0
y33f: y33 -20s33 <= 0
y34f: y34 -20s34 <= 0
y35f: y35 -20s35 <= 0
y36f: y36 -20s36 <= 0
y37f: y37 -20s37 <= 0
y38f: y38 -20s38 <= 0
y39f: y39 -20s39 <= 0
y40f: y40 -20s40 <= 0
y41f: y41 -20s41 <= 0
y42f: y42 -20s42 <= 0
y43f: y43 -20s43 <= 0
y44f: y44 -20s44 <= 0
y45f: y45 -20s45 <= 0
y46f: y46 -20s46 <= 0
y47f: y47 -20s47 <= 0
y48f: y48 -20s48 <= 0
y49f: y49 -20s49 <= 0
y50f: y50 -20s50 <= 0
y51f: y51 -20s51 <= 0
y52f: y52 -20s52 <= 0

y1a:  y1  -0.001s1  >= 0
y2a:  y2  -0.001s2  >= 0
y3a:  y3  -0.001s3  >= 0
y4a:  y4  -0.001s4  >= 0
y5a:  y5  -0.001s5  >= 0
y6a:  y6  -0.001s6  >= 0
y7a:  y7  -0.001s7  >= 0
y8a:  y8  -0.001s8  >= 0
y9a:  y9  -0.001s9  >= 0
y10a: y10 -0.001s10 >= 0
y11a: y11 -0.001s11 >= 0
y12a: y12 -0.001s12 >= 0
y13a: y13 -0.001s13 >= 0
y14a: y14 -0.001s14 >= 0
y15a: y15 -0.001s15 >= 0
y16a: y16 -0.001s16 >= 0
y17a: y17 -0.001s17 >= 0
y18a: y18 -0.001s18 >= 0
y19a: y19 -0.001s19 >= 0
y20a: y20 -0.001s20 >= 0
y21a: y21 -0.001s21 >= 0
y22a: y22 -0.001s22 >= 0
y23a: y23 -0.001s23 >= 0
```

```
y24a: y24 -0.001s24 >= 0
y25a: y25 -0.001s25 >= 0
y26a: y26 -0.001s26 >= 0
y27a: y27 -0.001s27 >= 0
y28a: y28 -0.001s28 >= 0
y29a: y29 -0.001s29 >= 0
y30a: y30 -0.001s30 >= 0
y31a: y31 -0.001s31 >= 0
y32a: y32 -0.001s32 >= 0
y33a: y33 -0.001s33 >= 0
y34a: y34 -0.001s34 >= 0
y35a: y35 -0.001s35 >= 0
y36a: y36 -0.001s36 >= 0
y37a: y37 -0.001s37 >= 0
y38a: y38 -0.001s38 >= 0
y39a: y39 -0.001s39 >= 0
y40a: y40 -0.001s40 >= 0
y41a: y41 -0.001s41 >= 0
y42a: y42 -0.001s42 >= 0
y43a: y43 -0.001s43 >= 0
y44a: y44 -0.001s44 >= 0
y45a: y45 -0.001s45 >= 0
y46a: y46 -0.001s46 >= 0
y47a: y47 -0.001s47 >= 0
y48a: y48 -0.001s48 >= 0
y49a: y49 -0.001s49 >= 0
y50a: y50 -0.001s50 >= 0
y51a: y51 -0.001s51 >= 0
y52a: y52 -0.001s52 >= 0


c0:  x0 = 3996
c1:  x1  -x0  +y1  = 21
c2:  x2  -x1  +y2  = 18
c3:  x3  -x2  +y3  = 16
c4:  x4  -x3  +y4  = 13
c5:  x5  -x4  +y5  = 12
c6:  x6  -x5  +y6  = 11
c7:  x7  -x6  +y7  = 14
c8:  x8  -x7  +y8  = 13
c9:  x9  -x8  +y9  = 26
c10: x10 -x9  +y10 = 33
c11: x11 -x10 +y11 = 37
c12: x12 -x11 +y12 = 40
c13: x13 -x12 +y13 = 31
```

```
c14: x14 -x13 +y14 = 34
c15: x15 -x14 +y15 = 11
c16: x16 -x15 +y16 = 13
c17: x17 -x16 +y17 =  7
c18: x18 -x17 +y18 = 10
c19: x19 -x18 +y19 =  8
c20: x20 -x19 +y20 = 13
c21: x21 -x20 +y21 = 15
c22: x22 -x21 +y22 = 10
c23: x23 -x22 +y23 = 19
c24: x24 -x23 +y24 = 20
c25: x25 -x24 +y25 = 24
c26: x26 -x25 +y26 = 23
c27: x27 -x26 +y27 = 15
c28: x28 -x27 +y28 = 17
c29: x29 -x28 +y29 = 29
c30: x30 -x29 +y30 = 28
c31: x31 -x30 +y31 = 21
c32: x32 -x31 +y32 = 10
c33: x33 -x32 +y33 = 16
c34: x34 -x33 +y34 = 14
c35: x35 -x34 +y35 = 12
c36: x36 -x35 +y36 = 28
c37: x37 -x36 +y37 = 25
c38: x38 -x37 +y38 = 16
c39: x39 -x38 +y39 = 18
c40: x40 -x39 +y40 = 19
c41: x41 -x40 +y41 = 17
c42: x42 -x41 +y42 = 11
c43: x43 -x42 +y43 = 13
c44: x44 -x43 +y44 = 18
c45: x45 -x44 +y45 = 13
c46: x46 -x45 +y46 = 11
c47: x47 -x46 +y47 = 10
c48: x48 -x47 +y48 = 12
c49: x49 -x48 +y49 =  8
c50: x50 -x49 +y50 = 12
c51: x51 -x50 +y51 = 13
c52: x52 -x51 +y52 = 16

x1_f:  x1  -d <= 4000
x2_f:  x2  -d <= 4000
x3_f:  x3  -d <= 4000
x4_f:  x4  -d <= 4000
x5_f:  x5  -d <= 4000
```

```
x6_f:  x6  -d <= 4000
x7_f:  x7  -d <= 4000
x8_f:  x8  -d <= 4000
x9_f:  x9  -d <= 4000
x10_f: x10 -d <= 4000
x11_f: x11 -d <= 4000
x12_f: x12 -d <= 4000
x13_f: x13 -d <= 4000
x14_f: x14 -d <= 4000
x15_f: x15 -d <= 4000
x16_f: x16 -d <= 4000
x17_f: x17 -d <= 4000
x18_f: x18 -d <= 4000
x19_f: x19 -d <= 4000
x20_f: x20 -d <= 4000
x21_f: x21 -d <= 4000
x22_f: x22 -d <= 4000
x23_f: x23 -d <= 4000
x24_f: x24 -d <= 4000
x25_f: x25 -d <= 4000
x26_f: x26 -d <= 4000
x27_f: x27 -d <= 4000
x28_f: x28 -d <= 4000
x29_f: x29 -d <= 4000
x30_f: x30 -d <= 4000
x31_f: x31 -d <= 4000
x32_f: x32 -d <= 4000
x33_f: x33 -d <= 4000
x34_f: x34 -d <= 4000
x35_f: x35 -d <= 4000
x36_f: x36 -d <= 4000
x37_f: x37 -d <= 4000
x38_f: x38 -d <= 4000
x39_f: x39 -d <= 4000
x40_f: x40 -d <= 4000
x41_f: x41 -d <= 4000
x42_f: x42 -d <= 4000
x43_f: x43 -d <= 4000
x44_f: x44 -d <= 4000
x45_f: x45 -d <= 4000
x46_f: x46 -d <= 4000
x47_f: x47 -d <= 4000
x48_f: x48 -d <= 4000
x49_f: x49 -d <= 4000
x50_f: x50 -d <= 4000
```

```
x51_f: x51 -d <= 4000
x52_f: x52 -d <= 4000

x1_a:  x1  +d >= 4000
x2_a:  x2  +d >= 4000
x3_a:  x3  +d >= 4000
x4_a:  x4  +d >= 4000
x5_a:  x5  +d >= 4000
x6_a:  x6  +d >= 4000
x7_a:  x7  +d >= 4000
x8_a:  x8  +d >= 4000
x9_a:  x9  +d >= 4000
x10_a: x10 +d >= 4000
x11_a: x11 +d >= 4000
x12_a: x12 +d >= 4000
x13_a: x13 +d >= 4000
x14_a: x14 +d >= 4000
x15_a: x15 +d >= 4000
x16_a: x16 +d >= 4000
x17_a: x17 +d >= 4000
x18_a: x18 +d >= 4000
x19_a: x19 +d >= 4000
x20_a: x20 +d >= 4000
x21_a: x21 +d >= 4000
x22_a: x22 +d >= 4000
x23_a: x23 +d >= 4000
x24_a: x24 +d >= 4000
x25_a: x25 +d >= 4000
x26_a: x26 +d >= 4000
x27_a: x27 +d >= 4000
x28_a: x28 +d >= 4000
x29_a: x29 +d >= 4000
x30_a: x30 +d >= 4000
x31_a: x31 +d >= 4000
x32_a: x32 +d >= 4000
x33_a: x33 +d >= 4000
x34_a: x34 +d >= 4000
x35_a: x35 +d >= 4000
x36_a: x36 +d >= 4000
x37_a: x37 +d >= 4000
x38_a: x38 +d >= 4000
x39_a: x39 +d >= 4000
x40_a: x40 +d >= 4000
x41_a: x41 +d >= 4000
x42_a: x42 +d >= 4000
```

```
x43_a: x43 +d >= 4000
x44_a: x44 +d >= 4000
x45_a: x45 +d >= 4000
x46_a: x46 +d >= 4000
x47_a: x47 +d >= 4000
x48_a: x48 +d >= 4000
x49_a: x49 +d >= 4000
x50_a: x50 +d >= 4000
x51_a: x51 +d >= 4000
x52_a: x52 +d >= 4000

+s1  +s2  +s3  +s4
+s5  +s6  +s7  +s8
+s9  +s10 +s11 +s12
+s13 +s14 +s15 +s16
+s17 +s18 +s19 +s20
+s21 +s22 +s23 +s24
+s25 +s26 +s27 +s28
+s29 +s30 +s31 +s32
+s33 +s34 +s35 +s36
+s37 +s38 +s39 +s40
+s41 +s42 +s43 +s44
+s45 +s46 +s47 +s48
+s49 +s50 +s51 +s52 <= 5


binary
s1  s2  s3  s4
s5  s6  s7  s8
s9  s10 s11 s12
s13 s14 s15 s16
s17 s18 s19 s20
s21 s22 s23 s24
s25 s26 s27 s28
s29 s30 s31 s32
s33 s34 s35 s36
s37 s38 s39 s40
s41 s42 s43 s44
s45 s46 s47 s48
s49 s50 s51 s52
end
```

Ebben az esetben `d = 810`, és a zsilip az 1-5 heteken lesz nyitva.

e) Ebben az esetben is használni kell az s_i bináris változókat, és a $\sum_{i=1}^{52} s_i \leq 5$

korlátot le kell cserélni másikra.

A négy egymást követő hét utáni szünetet másképpen úgy tudjuk megfogalmazni, hogy az egymást követő 5 hétből legfeljebb csak négyben lehet nyitva a zsilip. Tehát

$$s_i + s_{i+1} + s_{i+2} + s_{i+3} + s_{i+4} \leq 4 \ .$$

Amennyiben minden $i = 1, \ldots 48$ esetén megadjuk a modellben ezeket a korlátokat, akkor sehol sem lehet 5 egymást követő héten nyitva a zsilip. Tehát a következő sorokat kell a modellhez hozzáadni:

```
s1  +s2  +s3  +s4  +s5  <= 4
s2  +s3  +s4  +s5  +s6  <= 4
s3  +s4  +s5  +s6  +s7  <= 4
s4  +s5  +s6  +s7  +s8  <= 4
s5  +s6  +s7  +s8  +s9  <= 4
s6  +s7  +s8  +s9  +s10 <= 4
s7  +s8  +s9  +s10 +s11 <= 4
s8  +s9  +s10 +s11 +s12 <= 4
s9  +s10 +s11 +s12 +s13 <= 4
s10 +s11 +s12 +s13 +s14 <= 4
s11 +s12 +s13 +s14 +s15 <= 4
s12 +s13 +s14 +s15 +s16 <= 4
s13 +s14 +s15 +s16 +s17 <= 4
s14 +s15 +s16 +s17 +s18 <= 4
s15 +s16 +s17 +s18 +s19 <= 4
s16 +s17 +s18 +s19 +s20 <= 4
s17 +s18 +s19 +s20 +s21 <= 4
s18 +s19 +s20 +s21 +s22 <= 4
s19 +s20 +s21 +s22 +s23 <= 4
s20 +s21 +s22 +s23 +s24 <= 4
s21 +s22 +s23 +s24 +s25 <= 4
s22 +s23 +s24 +s25 +s26 <= 4
s23 +s24 +s25 +s26 +s27 <= 4
s24 +s25 +s26 +s27 +s28 <= 4
s25 +s26 +s27 +s28 +s29 <= 4
s26 +s27 +s28 +s29 +s30 <= 4
s27 +s28 +s29 +s30 +s31 <= 4
s28 +s29 +s30 +s31 +s32 <= 4
s29 +s30 +s31 +s32 +s33 <= 4
s30 +s31 +s32 +s33 +s34 <= 4
```

```
s31 +s32 +s33 +s34 +s35 <= 4
s32 +s33 +s34 +s35 +s36 <= 4
s33 +s34 +s35 +s36 +s37 <= 4
s34 +s35 +s36 +s37 +s38 <= 4
s35 +s36 +s37 +s38 +s39 <= 4
s36 +s37 +s38 +s39 +s40 <= 4
s37 +s38 +s39 +s40 +s41 <= 4
s38 +s39 +s40 +s41 +s42 <= 4
s39 +s40 +s41 +s42 +s43 <= 4
s40 +s41 +s42 +s43 +s44 <= 4
s41 +s42 +s43 +s44 +s45 <= 4
s42 +s43 +s44 +s45 +s46 <= 4
s43 +s44 +s45 +s46 +s47 <= 4
s44 +s45 +s46 +s47 +s48 <= 4
s45 +s46 +s47 +s48 +s49 <= 4
s46 +s47 +s48 +s49 +s50 <= 4
s47 +s48 +s49 +s50 +s51 <= 4
s48 +s49 +s50 +s51 +s52 <= 4
```

Ebben az esetben `d =116`.

2.25. példa megoldása.

Első nekifutásra vezessük be az x_i $(i = 1, \dots 8\ )$ bináris változókat, ami a két mentőautó elhelyezését mutatja. Ha $x_i = 1$, akkor egy mentőautót az i-edik kerületben állomásoztatunk. Ha az első kerületben állomásoztatunk mentőautót, akkor innen csak az első kerület lakossága érhető el, 40 ezer fő. Ha a második kerülteben állomásoztatunk egy mentőautót, akkor innen is csak a második kerület lakossága érhető el, 30 ezer fő. A harmadik kerületben állomásoztatott mentőautó segítségével elérhető a 3., a 4. és az 5. kerület lakossága is, ami összesen 70 ezer fő. A többi kerületre is kiszámítjuk ezeket az értékeket, így megkapjuk a célfüggvényt. Mivel két mentőutó van, az x_i változók összege 2.

10.6. kód.

```
max
+40x1 +30x2 +70x3 +105x4 +145x5 +85x6 +120x7 +105x8

subject to
sx: x1 +x2 +x3 +x4 +x5 +x6 +x7 +x8 = 2
```

```
binary
x1 x2 x3 x4 x5 x6 x7 x8
end
```

Az optimális megoldás esetén az 5. és a 7. kerületben állomásoztatunk mentőautókat, a célfüggvény értéke pedig 265. Ha az ötödik és a hetedik kerületben állomásoztatunk mentőautókat akkor ezekből a kerületekből a 3., 5., 6., 7. és 8. kerületek lakossága érhető el, ami 205 ezer fő. Miért nem egyezik meg a kettő? Azert mert pl. az 5. kerület lakosságát kétszer vettük figyelembe: egyszer úgy, hogy az 5. kerületből elérhető két percen belül, egyszer pedig úgy, hogy a 7. kerülteből elérhető két percen belül. De amiatt, hogy egy kerületet két mentőautóval is el lehet érni 2 percen belül, még nem lehet a lakosságszámot kétszeresen beszámítani.

A problémát úgy korrigálhatjuk, hogy bevezetjük az y_i $(i = 1, \ldots 8\ )$ bináris változókat, amelyek 1-es értéke azt jelzi, hogy az i-edik kerület lakosságát elérjük-e már egy telepített mentőállomással valahonnan 2 percen belül.

Természetesen x_i és y_j változók között szoros kapcsolat van. Az y_1 értéke például csak akkor lehet 1, ha az első kerületben állomásoztatunk mentőautót: $y_1 \leq x_1$. Ennél érdekesebb például a 3. kerület: y_3 változó csak akkor lehet 1, ha a 3., a 4. vagy az 5. kerületben állomásoztatunk mentőautót: $y_3 \leq x_3 + x_4 + x_5$. Ezeket a korlátokat az összes kerületre felírjuk. Az y_I változókat beszorozzuk a kerület lakosságával, ez lesz a célfüggvény.

10.7. kód.

```
max
+40y1 +30y2 +35y3 +20y4 +15y5 +50y6 +45y7 +60y8

subject to
sx: x1 +x2 +x3 +x4 +x5 +x6 +x7 +x8 = 2
y1-x1 <= 0
y2-x2 <= 0
y3-x3-x4-x5 <= 0
y4-x3-x4-x6 <= 0
y5-x3-x5-x6-x7 <= 0
y6-x4-x5-x6 <= 0
y7-x5-x7-x8 <= 0
y8-x7-x8 <= 0

binary
```

```
x1 x2 x3 x4 x5 x6 x7 x8
y1 y2 y3 y4 y5 y6 y7 y8

end
```

Az optimális megoldás:

```
Column name        Activity
------------     -------------
y1              *              0
y2              *              0
y3              *              1
y4              *              1
y5              *              1
y6              *              1
y7              *              1
y8              *              1
x1              *              0
x2              *              0
x3              *              0
x4              *              1
x5              *              0
x6              *              0
x7              *              1
x8              *              0
```

- A negyedik és hetedik kerületben állomásoztatunk mentőautót. Így elérhetővé válik az első két kerületen kívül az összes kerület lakossága, ami 225 ezer fő.

- A $\sum_{i=1}^{8} x_i \leq 2$ korlát jobboldalának egységnyi növelésével, majd az IP újrafuttatásával tudnánk erre általános esetben választ adni. Azonban ebben az esetben látható, hogy egyedül az 1. és a 2. kerületet nem éri el a mentőautó 2 percen belül, és egyik kerületet sem érheti el ilyen rövid idő alatt semelyik másik kerületből. Tehát a harmadik mentőautó biztosan ezek közül valamelyikbe kerül. Ha a mentő az első kerületbe kerül, akkor további 40 ezer embert ér el 2 percen belül, ha a másodikba, akkor csak 30 ezret. Ezért az elsőbe kell, hogy kerüljön.

- Az előző pont megoldásából következik, hogy összesen feltehetően 4 mentőautó lenne elegendő ahhoz, hogy a teljes várost lefedjük. Azonban ezt egy új LP bevezetésével is meghatározhatjuk. A 10.8. kód estén vegyünk be egy olyan korlátot, hogy az y_i változók összege 8, és a célfüggvényt cseréljük le az x_I változók összegére, és az `sx` korlátot töröljük.

10.8. kód.

```
min
x1 +x2 +x3 +x4 +x5 +x6 +x7 +x8

subject to
y1 +y2 +y3 +y4 +y5 +y6 +y7 +y8 = 8
y1-x1 <= 0
y2-x2 <= 0
y3-x3-x4-x5 <= 0
y4-x3-x4-x6 <= 0
y5-x3-x5-x6-x7 <= 0
y6-x4-x5-x6 <= 0
y7-x5-x7-x8 <= 0
y8-x7-x8 <= 0

binary
x1 x2 x3 x4 x5 x6 x7 x8
y1 y2 y3 y4 y5 y6 y7 y8

end
```

Az optimális megoldás esetén a célfüggvény értéke 4, tehát valóban csak 4 mentőautóval lehet lefedni a várost.

10.2. Szállítási feladatok

3.33. példa megoldása.

Ebben a feladatban egy egyszerű kiegyensúlyozott szállítási feladatot kell megoldani, 11 raktárra, illetve 7 áruházra.

10.9. kód.

```
min
ktg:
+53x1_1 +71x1_2 +77x1_3 +74x1_4 +17x1_5 +82x1_6 +80x1_7
+72x2_1 +41x2_2 +90x2_3 +85x2_4 +72x2_5 +32x2_6 +86x2_7
+89x3_1 +41x3_2 +49x3_3 +11x3_4 +51x3_5 +35x3_6 +41x3_7
+66x4_1 + 1x4_2 +41x4_3 +38x4_4 +44x4_5 +64x4_6 +68x4_7
```

```
+72x5_1 +20x5_2 +11x5_3 +86x5_4 +24x5_5 +53x5_6 + 8x5_7
+81x6_1 +74x6_2 +31x6_3 + 2x6_4 +39x6_5 +90x6_6 +91x6_7
+28x7_1 +83x7_2 +25x7_3 +31x7_4 +69x7_5 +59x7_6 +90x7_7
+19x8_1 +38x8_2 +58x8_3 +56x8_4 +65x8_5 +67x8_6 + 4x8_7
+15x9_1 +62x9_2 +23x9_3 +98x9_4 + 2x9_5 +69x9_6 +16x9_7
+16x10_1+37x10_2+89x10_3+76x10_4+24x10_5+61x10_6+67x10_7
+23x11_1+68x11_2+60x11_3+49x11_4+56x11_5+11x11_6+72x11_7

subject to
r1:  +x1_1  +x1_2  +x1_3  +x1_4  +x1_5  +x1_6  +x1_7  = 9
r2:  +x2_1  +x2_2  +x2_3  +x2_4  +x2_5  +x2_6  +x2_7  = 5
r3:  +x3_1  +x3_2  +x3_3  +x3_4  +x3_5  +x3_6  +x3_7  =13
r4:  +x4_1  +x4_2  +x4_3  +x4_4  +x4_5  +x4_6  +x4_7  = 5
r5:  +x5_1  +x5_2  +x5_3  +x5_4  +x5_5  +x5_6  +x5_7  = 2
r6:  +x6_1  +x6_2  +x6_3  +x6_4  +x6_5  +x6_6  +x6_7  =18
r7:  +x7_1  +x7_2  +x7_3  +x7_4  +x7_5  +x7_6  +x7_7  =11
r8:  +x8_1  +x8_2  +x8_3  +x8_4  +x8_5  +x8_6  +x8_7  =10
r9:  +x9_1  +x9_2  +x9_3  +x9_4  +x9_5  +x9_6  +x9_7  =19
r10: +x10_1 +x10_2 +x10_3 +x10_4 +x10_5 +x10_6 +x10_7 =15
r11: +x11_1 +x11_2 +x11_3 +x11_4 +x11_5 +x11_6 +x11_7 =10

a1: +x1_1  +x2_1  +x3_1  +x4_1  +x5_1  +x6_1
                  +x7_1  +x8_1  +x9_1  +x10_1  +x11_1 =20
a2: +x1_2  +x2_2  +x3_2  +x4_2  +x5_2  +x6_2
                  +x7_2  +x8_2  +x9_2  +x10_2  +x11_2 =15
a3: +x1_3  +x2_3  +x3_3  +x4_3  +x5_3  +x6_3
                  +x7_3  +x8_3  +x9_3  +x10_3  +x11_3 =10
a4: +x1_4  +x2_4  +x3_4  +x4_4  +x5_4  +x6_4
                  +x7_4  +x8_4  +x9_4  +x10_4  +x11_4=20
a5: +x1_5  +x2_5  +x3_5  +x4_5  +x5_5  +x6_5
                  +x7_5  +x8_5  +x9_5  +x10_5  +x11_5=10
a6: +x1_6  +x2_6  +x3_6  +x4_6  +x5_6  +x6_6
                  +x7_6  +x8_6  +x9_6  +x10_6  +x11_6=11
a7: +x1_7  +x2_7  +x3_7  +x4_7  +x5_7  +x6_7
                  +x7_7  +x8_7  +x9_7  +x10_7  +x11_7=31
end
```

A szállítások:

```
 No.  Column name St   Activity
---- ------------ -- -------------
   5  x1_5         B              9
   9  x2_2         B              4
  13  x2_6         B              1
  16  x3_2         B              6
  18  x3_4         B              2
```

```
  21  x3_7         B            5
  23  x4_2         B            5
  35  x5_7         B            2
  39  x6_4         B           18
  43  x7_1         B            1
  45  x7_3         B           10
  56  x8_7         B           10
  57  x9_1         B            4
  61  x9_5         B            1
  63  x9_7         B           14
  64  x10_1        B           15
  76  x11_6        B           10
```

A szállítás összköltsége: 1833.

3.34. példa megoldása.

Ez a feladat is egy egyszerű kiegyensúlyozott szállítási feladat, az CPLEX LP kód hasonló az előző feladathoz.

10.10. kód.

```
min
ktg:
+13x1_1 + 5x1_2 + 3x1_3 + 5x1_4 + 9x1_5 + 2x1_6 + 4x1_7
+13x2_1 + 7x2_2 + 3x2_3 + 6x2_4 +10x2_5 + 2x2_6 + 5x2_7
+16x3_1 +10x3_2 +13x3_3 + 9x3_4 +13x3_5 + 9x3_6 +10x3_7
+25x4_1 +19x4_2 +18x4_3 +12x4_4 +17x4_5 +14x4_6 +11x4_7
+39x5_1 +21x5_2 +22x5_3 +12x5_4 +19x5_5 +21x5_6 +18x5_7
+42x6_1 +23x6_2 +27x6_3 +23x6_4 +21x6_5 +26x6_6 +19x6_7
+43x7_1 +27x7_2 +27x7_3 +23x7_4 +26x7_5 +30x7_6 +38x7_7
+43x8_1 +34x8_2 +27x8_3 +26x8_4 +30x8_5 +43x8_6 +45x8_7
+46x9_1 +39x9_2 +28x9_3 +26x9_4 +33x9_5 +49x9_6 +48x9_7
+51x10_1+42x10_2+39x10_3+27x10_4+33x10_5+49x10_6+48x10_7

subject to
r1: +x1_1  +x1_2  +x1_3  +x1_4  +x1_5  +x1_6  +x1_7  =10
r2: +x2_1  +x2_2  +x2_3  +x2_4  +x2_5  +x2_6  +x2_7  =20
r3: +x3_1  +x3_2  +x3_3  +x3_4  +x3_5  +x3_6  +x3_7  =15
r4: +x4_1  +x4_2  +x4_3  +x4_4  +x4_5  +x4_6  +x4_7  =40
r5: +x5_1  +x5_2  +x5_3  +x5_4  +x5_5  +x5_6  +x5_7  =50
r6: +x6_1  +x6_2  +x6_3  +x6_4  +x6_5  +x6_6  +x6_7  =20
r7: +x7_1  +x7_2  +x7_3  +x7_4  +x7_5  +x7_6  +x7_7  =10
r8: +x8_1  +x8_2  +x8_3  +x8_4  +x8_5  +x8_6  +x8_7  =15
r9: +x9_1  +x9_2  +x9_3  +x9_4  +x9_5  +x9_6  +x9_7  =15
```

```
r10:+x10_1 +x10_2+ x10_3 +x10_4 +x10_5 +x10_6 +x10_7 =35

a1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1
                            +x7_1 +x8_1 +x9_1 +x10_1 =30
a2: +x1_2 +x2_2 +x3_2 +x4_2 +x5_2 +x6_2
                            +x7_2 +x8_2 +x9_2 +x10_2 =40
a3: +x1_3 +x2_3 +x3_3 +x4_3 +x5_3 +x6_3
                            +x7_3 +x8_3 +x9_3 +x10_3 =10
a4: +x1_4 +x2_4 +x3_4 +x4_4 +x5_4 +x6_4
                            +x7_4 +x8_4 +x9_4 +x10_4 =20
a5: +x1_5 +x2_5 +x3_5 +x4_5 +x5_5 +x6_5
                            +x7_5 +x8_5 +x9_5 +x10_5 =50
a6: +x1_6 +x2_6 +x3_6 +x4_6 +x5_6 +x6_6
                            +x7_6 +x8_6 +x9_6 +x10_6 =40
a7: +x1_7 +x2_7 +x3_7 +x4_7 +x5_7 +x6_7
                            +x7_7 +x8_7 +x9_7 +x10_7 =40
end
```

Az optimális megoldás:

```
 No. Column name  St   Activity
---- ------------ -- -------------
   6 x1_6          B           10
   8 x2_1          B           15
  13 x2_6          B            5
  15 x3_1          B           15
  27 x4_6          B           25
  28 x4_7          B           15
  30 x5_2          B           30
  32 x5_4          B           15
  35 x5_7          B            5
  42 x6_7          B           20
  44 x7_2          B           10
  54 x8_5          B           15
  59 x9_3          B           10
  60 x9_4          B            5
  68 x10_5         B           35
```

A szállítás összköltsége: 4545.

3.35. példa megoldása.

Az 5. raktár ebben az esetben a ki nem elégített áruházi igényeket jelenti, az x5_1, x5_2, x5_3, x5_4, x5_5, x5_6 változók pedig az adott áruház fennmaradó szükségleteit adják meg.

10.11. kód.

```
min
ktg:
+100x1_1 + 30x1_2 + 14x1_3 + 24x1_4 +125x1_5 +120x1_6
+ 37x2_1 +100x2_2 + 33x2_3 + 41x2_4 +137x2_5 +104x2_6
+ 27x3_1 +  9x3_2 +100x3_3 + 10x3_4 +124x3_5 +118x3_6
+  3x4_1 + 10x4_2 + 18x4_3 +100x4_4 +118x4_5 +115x4_6
+ 11x5_1 + 17x5_2 + 21x5_3 +  9x5_4 + 13x5_5 + 15x5_6

subject to
r1: +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 =20
r2: +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 =20
r3: +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 =20
r4: +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 =20
r5: +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 =20

a1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 =18
a2: +x1_2 +x2_2 +x3_2 +x4_2 +x5_2 =20
a3: +x1_3 +x2_3 +x3_3 +x4_3 +x5_3 =15
a4: +x1_4 +x2_4 +x3_4 +x4_4 +x5_4 = 7
a5: +x1_5 +x2_5 +x3_5 +x4_5 +x5_5 =23
a6: +x1_6 +x2_6 +x3_6 +x4_6 +x5_6 =17

end
```

Az optimális megoldás:

```
No.  Column name St  Activity
---- ------------ -- -------------
 3    x1_3         B      15
 4    x1_4         B       5
11    x2_5         B       3
12    x2_6         B      17
14    x3_2         B      18
16    x3_4         B       2
19    x4_1         B      18
20    x4_2         B       2
29    x5_5         B      20
```

Az összes kötbért tehát az 5. áruház esetében kell fizetni, a teljes szállítási költség pedig 3025.

3.36. példa megoldása.

Legyen a_8 egy fiktív áruház, amelybe szállítás költsége minden raktárból 0, és a kereslete 15. Az átrakodási pontba szállítást jelölje a 9. áruház (a_9), és onnan elszállítást pedig a 9. raktár (r_9). Az átrakodási pontba legfeljebb 64 árucikket lehet szállítani, ezért ennyi lesz az a_9 kereslete, és az r_9 kínálata. A kiinduló eset:

10.12. kód.

```
min
ktg:
+ 82x1_1 +28x1_2 +37x1_3 +43x1_4 +38x1_5
                         +78x1_6 +43x1_7 +0x1_8 +100x1_9
+ 29x2_1 +51x2_2 +89x2_3 +96x2_4 +94x2_5
                         +67x2_6 +87x2_7 +0x2_8 +100x2_9
+ 59x3_1 +37x3_2 +97x3_3 +60x3_4 +64x3_5
                         +49x3_6 +74x3_7 +0x3_8 +100x3_9
+ 52x4_1 +80x4_2 +55x4_3 +65x4_4 +27x4_5
                         +30x4_6 +65x4_7 +0x4_8 +100x4_9
+ 27x5_1 +16x5_2 +68x5_3 +71x5_4 +90x5_5
                         +13x5_6 +98x5_7 +0x5_8 +100x5_9
+ 28x6_1 +11x6_2 +42x6_3 +32x6_4 +67x6_5
                         +42x6_6 +25x6_7 +0x6_8 +100x6_9
+ 29x7_1 +23x7_2 +97x7_3 +20x7_4 +11x7_5
                         +41x7_6 +69x7_7 +0x7_8 +100x7_9
+ 66x8_1 +67x8_2 +80x8_3 +10x8_4 +93x8_5
                         +53x8_6 +76x8_7 +0x8_8 +100x8_9
+100x9_1+100x9_2+100x9_3+100x9_4+100x9_5
                        +100x9_6+100x9_7 +0x9_8 +  0x9_9

subject to
r1: +x1_1 +x1_2 +x1_3 +x1_4 +x1_5
                      +x1_6 +x1_7 +x1_8+x1_9 =  8
r2: +x2_1 +x2_2 +x2_3 +x2_4 +x2_5
                      +x2_6 +x2_7 +x2_8+x2_9 =  8
r3: +x3_1 +x3_2 +x3_3 +x3_4 +x3_5
                      +x3_6 +x3_7 +x3_8+x3_9 =  8
r4: +x4_1 +x4_2 +x4_3 +x4_4 +x4_5
                      +x4_6 +x4_7 +x4_8+x4_9 =  8
r5: +x5_1 +x5_2 +x5_3 +x5_4 +x5_5
                      +x5_6 +x5_7 +x5_8+x5_9 =  8
r6: +x6_1 +x6_2 +x6_3 +x6_4 +x6_5
                      +x6_6 +x6_7 +x6_8+x6_9 =  8
r7: +x7_1 +x7_2 +x7_3 +x7_4 +x7_5
                      +x7_6 +x7_7 +x7_8+x7_9 =  8
r8: +x8_1 +x8_2 +x8_3 +x8_4 +x8_5
```

```
                    +x8_6 +x8_7 +x8_8+x8_9 =  8
r9: +x9_1 +x9_2 +x9_3 +x9_4 +x9_5
                    +x9_6 +x9_7 +x9_8+x9_9 = 64

a1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1
                    +x6_1 +x7_1 +x8_1 +x9_1 =  7
a2: +x1_2 +x2_2 +x3_2 +x4_2 +x5_2
                    +x6_2 +x7_2 +x8_2 +x9_2 =  7
a3: +x1_3 +x2_3 +x3_3 +x4_3 +x5_3
                    +x6_3 +x7_3 +x8_3 +x9_3 =  7
a4: +x1_4 +x2_4 +x3_4 +x4_4 +x5_4
                    +x6_4 +x7_4 +x8_4 +x9_4 =  7
a5: +x1_5 +x2_5 +x3_5 +x4_5 +x5_5
                    +x6_5 +x7_5 +x8_5 +x9_5 =  7
a6: +x1_6 +x2_6 +x3_6 +x4_6 +x5_6
                    +x6_6 +x7_6 +x8_6 +x9_6 =  7
a7: +x1_7 +x2_7 +x3_7 +x4_7 +x5_7
                    +x6_7 +x7_7 +x8_7 +x9_7 =  7
a8: +x1_8 +x2_8 +x3_8 +x4_8 +x5_8
                    +x6_8 +x7_8 +x8_8 +x9_8 = 15
a9: +x1_9 +x2_9 +x3_9 +x4_9 +x5_9
                    +x6_9 +x7_9 +x8_9 +x9_9 = 64
end
```

A példa optimális megoldása:

```
 No.  Column name    Activity
---- ------------  -------------
   2  x1_2                   1
   3  x1_3                   7
  10  x2_1                   7
  17  x2_8                   1
  26  x3_8                   8
  33  x4_6                   3
  35  x4_8                   5
  38  x5_2                   4
  42  x5_6                   4
  47  x6_2                   1
  52  x6_7                   7
  56  x7_2                   1
  59  x7_5                   7
  67  x8_4                   7
  71  x8_8                   1
  81  x9_9                  64
```

A szállítás összköltsége 1052, és az átrakodási pontot nem használjuk ezen költségek mellett.

Ahhoz, hogy meghatározzuk a legnagyobb olyan α-t, amely esetén már használjuk az átrakodási pontot, ahhoz az ehhez a ponthoz kapcsolódó redukált költségeket kell megnéznünk (`Marginal` oszlop):

```
   No. Column name  St   Activity        Marginal
------ ------------ -- -------------  -------------
    73   x9_1        NL            0              71
    74   x9_2        NL            0              67
    75   x9_3        NL            0              58
    76   x9_4        NL            0              90
    77   x9_5        NL            0              79
    78   x9_6        NL            0              70
    79   x9_7        NL            0              53
    80   x9_8        B             0
    81   x9_9        B            64
     9   x1_9        NL            0             105
    18   x2_9        NL            0             100
    27   x3_9        NL            0             100
    36   x4_9        NL                          100
    45   x5_9        NL            0             117
    54   x6_9        NL            0             122
    63   x7_9        NL            0             110
    72   x8_9        NL            0             100
```

Azonban a feladat ilyen felírásában a be- és a kiszállítás redukált költségeit egyszerre kellene vizsgálni, ezért így nem tudjuk meghatározni a kérdéses α értéket. Változtassuk meg úgy a kódot, hogy az átrakodási pont szállításainak a költségeit egyedül az odaszállításkor vesszük figyelembe, az elszállításkor nem. Tehát a módosított kódban a költségmátrix 9. sorában csak 0 értékek szerepelnek, a 9. oszlopban pedig 2α értékek (kivéve a (9,9), mert az marad 0).

10.13. kód.

```
min
ktg:
+ 82x1_1 +28x1_2 +37x1_3 +43x1_4 +38x1_5
                         +78x1_6 +43x1_7 +0x1_8 +200x1_9
+ 29x2_1 +51x2_2 +89x2_3 +96x2_4 +94x2_5
                         +67x2_6 +87x2_7 +0x2_8 +200x2_9
+ 59x3_1 +37x3_2 +97x3_3 +60x3_4 +64x3_5
                         +49x3_6 +74x3_7 +0x3_8 +200x3_9
+ 52x4_1 +80x4_2 +55x4_3 +65x4_4 +27x4_5
```

```
                        +30x4_6 +65x4_7 +0x4_8 +200x4_9
+ 27x5_1 +16x5_2 +68x5_3 +71x5_4 +90x5_5
                        +13x5_6 +98x5_7 +0x5_8 +200x5_9
+ 28x6_1 +11x6_2 +42x6_3 +32x6_4 +67x6_5
                        +42x6_6 +25x6_7 +0x6_8 +200x6_9
+ 29x7_1 +23x7_2 +97x7_3 +20x7_4 +11x7_5
                        +41x7_6 +69x7_7 +0x7_8 +200x7_9
+ 66x8_1 +67x8_2 +80x8_3 +10x8_4 +93x8_5
                        +53x8_6 +76x8_7 +0x8_8 +200x8_9
+  0x9_1+  0x9_2+  0x9_3+  0x9_4+  0x9_5
                       +  0x9_6 + 0x9_7 +0x9_8 +  0x9_9

subject to
r1:+x1_1 +x1_2 +x1_3 +x1_4 +x1_5
                          +x1_6 +x1_7 +x1_8 +x1_9 =  8
r2:+x2_1 +x2_2 +x2_3 +x2_4 +x2_5
                          +x2_6 +x2_7 +x2_8 +x2_9 =  8
r3:+x3_1 +x3_2 +x3_3 +x3_4 +x3_5
                          +x3_6 +x3_7 +x3_8 +x3_9 =  8
r4:+x4_1 +x4_2 +x4_3 +x4_4 +x4_5
                          +x4_6 +x4_7 +x4_8 +x4_9 =  8
r5:+x5_1 +x5_2 +x5_3 +x5_4 +x5_5
                          +x5_6 +x5_7 +x5_8 +x5_9 =  8
r6:+x6_1 +x6_2 +x6_3 +x6_4 +x6_5
                          +x6_6 +x6_7 +x6_8 +x6_9 =  8
r7:+x7_1 +x7_2 +x7_3 +x7_4 +x7_5
                          +x7_6 +x7_7 +x7_8 +x7_9 =  8
r8:+x8_1 +x8_2 +x8_3 +x8_4 +x8_5
                          +x8_6 +x8_7 +x8_8 +x8_9 =  8
r9:+x9_1 +x9_2 +x9_3 +x9_4 +x9_5
                          +x9_6 +x9_7 +x9_8 +x9_9 = 64

a1:+x1_1 +x2_1 +x3_1 +x4_1 +x5_1
                          +x6_1 +x7_1 +x8_1 +x9_1 =  7
a2:+x1_2 +x2_2 +x3_2 +x4_2 +x5_2
                          +x6_2 +x7_2 +x8_2 +x9_2 =  7
a3:+x1_3 +x2_3 +x3_3 +x4_3 +x5_3
                          +x6_3 +x7_3 +x8_3 +x9_3 =  7
a4:+x1_4 +x2_4 +x3_4 +x4_4 +x5_4
                          +x6_4 +x7_4 +x8_4 +x9_4 =  7
a5:+x1_5 +x2_5 +x3_5 +x4_5 +x5_5
                          +x6_5 +x7_5 +x8_5 +x9_5 =  7
a6:+x1_6 +x2_6 +x3_6 +x4_6 +x5_6
                          +x6_6 +x7_6 +x8_6 +x9_6 =  7
a7:+x1_7 +x2_7 +x3_7 +x4_7 +x5_7
```

```
                             +x6_7 +x7_7 +x8_7 +x9_7 =  7
a8:+x1_8 +x2_8 +x3_8 +x4_8 +x5_8
                             +x6_8 +x7_8 +x8_8 +x9_8 = 15
a9:+x1_9 +x2_9 +x3_9 +x4_9 +x5_9
                             +x6_9 +x7_9 +x8_9 +x9_9 = 64
end
```

Ekkor az átrakodási pontba szállítások redukált költségei:

```
   No. Column name  St   Activity        Marginal
------ ------------ -- -------------  -------------
     9   x1_9         NL             0            158
    18   x2_9         NL             0            153
    27   x3_9         NL             0            153
    36   x4_9         NL             0            153
    45   x5_9         NL             0            170
    54   x6_9         NL             0            175
    63   x7_9         NL             0            163
    72   x8_9         NL             0            153
    81   x9_9         B             64
```

Azt találjuk, hogy legkisebb redukált költség 153, tehát ha $\alpha = \frac{200-153}{2} = 23,5$, akkor már lesz olyan szállítás, ami az átrakodási ponton keresztül valósul meg.

A legnagyobb olyan α meghatározásához, amely esetén minden szállítás az átrakodási ponton keresztül valósul meg, elég a célfüggvényt vizsgálni. Kiinduló esetben minden raktárból az átrakodási ponton keresztül a szállítás 200, azaz 2α. A közvetett szállítások közül a legolcsóbb a 8. raktár és a 4. áruház közötti szállítás, amely összesen 10. Ahhoz, hogy a közvetett szállítás legalább olyan jó legyen ebben az esetben, mint a közvetett, ahhoz fel kell tenni, hogy $2\alpha \leq 10$. Tehát akkor lehet olyan optimális szállítási terv, amelyben minden szállítás az átrakodási ponton keresztül történik, ha $\alpha \leq 5$.

3.37. példa megoldása.

Jelölje x_{ij} a szárazföldi-, y_{ij} pedig a vízi szállításokat az i. raktár és a j. áruház között. Legyen H bináris változó, amely akkor lehet 1, ha megépül a hajópark, különben pedig 0. Valamint legyenek K_i bináris változók, amely akkor 1, ha megépül az i. kezdőállomás, tehát az i. raktár kikötője, és 0, ha nem. Ezen kívül legyenek

V_j bináris változók, amely akkor 1, ha megépül a j. végállomás, azaz a j. áruház kikötője, és 0, ha nem.

A célfüggvényben az összesített költségeket minimalizáljuk, ahol a c_{ij} jelöli a szárazföldi szállítási költségeit, b_{ij} pedig a vízi szállításét. A szállítási költségeken kívül az építési költségeket is szerepeltetni kell a célfüggvényben:

$$\min \quad \sum_{i=1}^{10}\sum_{j=1}^{7}\left(c_{ij}x_{ij}+b_{ij}y_{ij}\right)+\sum_{i=1}^{10}100K_i+\sum_{j=1}^{7}100V_j+500H \ .$$

Az r_i : feltételek a raktárakból elszállítandó mennyiségre adnak korlátot:

$$r_i : \qquad \sum_{j=1}^{7} x_{ij}+y_{ij}=S_i \qquad \forall i \ ,$$

ahol S_i az i. raktár kínálata. Az a_j : feltételek az áruházakba szállítandó mennyiségre adnak korlátot:

$$a_j : \qquad \sum_{j=1}^{10} x_{ij}+y_{ij}=D_j \qquad \forall j \ ,$$

ahol D_j az j. áruház kereslete. Ezen kívül fel kell tenni, hogy ha legalább egy vízi szállítás megvalósul, akkor meg kell építeni a hajóparkot:

$$h : \qquad \sum_{i=1}^{10}\sum_{j=1}^{7} y_{ij} \le MH \ ,$$

ahol M egy kellően nagy konstans, jelen példában $M \ge 230$, mivel $\sum\limits_{i=1}^{10} S_i = \sum\limits_{j=1}^{7} D_j = 230$. Amennyiben az i. raktárból szállítunk a vízen keresztül, akkor a kikötőt meg kell építeni, ehhez elég a következő korlátokat hozzáadni a feladathoz:

$$k_i : \qquad \sum_{j=1}^{7} y_{ij} \le MK_i \qquad \forall i \ .$$

Ugyanúgy minden áruházhoz tartozó kikötőről is fel kell tenni:

$$v_j : \qquad \sum_{i=1}^{10} y_{ij} \le MV_j \qquad \forall j \ .$$

10.14. kód.

```
min
ktg:
+13x1_1 + 5x1_2 + 3x1_3 + 5x1_4 + 9x1_5 + 2x1_6 + 4x1_7
+13x2_1 + 7x2_2 + 3x2_3 + 6x2_4 +10x2_5 + 2x2_6 + 5x2_7
+16x3_1 +10x3_2 +13x3_3 + 9x3_4 +13x3_5 + 9x3_6 +10x3_7
+25x4_1 +19x4_2 +18x4_3 +12x4_4 +17x4_5 +14x4_6 +11x4_7
+39x5_1 +21x5_2 +22x5_3 +12x5_4 +19x5_5 +21x5_6 +18x5_7
+42x6_1 +23x6_2 +27x6_3 +23x6_4 +21x6_5 +26x6_6 +19x6_7
+43x7_1 +27x7_2 +27x7_3 +23x7_4 +26x7_5 +30x7_6 +38x7_7
+43x8_1 +34x8_2 +27x8_3 +26x8_4 +30x8_5 +43x8_6 +45x8_7
+46x9_1 +39x9_2 +28x9_3 +26x9_4 +33x9_5 +49x9_6 +48x9_7
+51x10_1+42x10_2+39x10_3+27x10_4+33x10_5+49x10_6+48x10_7

        + 2y1_2 + 3y1_3 + 2y1_4         + 2y1_6
+ 6y2_1                 + 5y2_4 + 8y2_5 + 7y2_6 + 2y2_7
+ 6y3_1 + 8y3_2 + 5y3_3 + 8y3_4         + 9y3_6
+15y4_1         + 6y4_3 + 9y4_4 +12y4_5 +20y4_6 + 4y4_7
+16y5_1 +14y5_2 + 8y5_3         +20y5_5 +21y5_6
        +14y6_2         +17y6_4 +21y6_5 +23y6_6 + 9y6_7
+25y7_1 +21y7_2 +10y7_3         +24y7_5 +25y7_6 +11y7_7
+27y8_1         +10y8_3 +26y8_4 +26y8_5         +20y8_7
+30y9_1 +22y9_2 +11y9_3 +27y9_4 +29y9_5 +29y9_6
+30y10_1+23y10_2+22y10_3                +29y10_6+25y10_7

+100K1 +100K2 +100K3 +100K4 +100K5 +100K6
                             +100K7 +100K8 +100K9+100K10
+100V1  +100V2  +100V3  +100V4  +100V5  +100V6  +100V7
+500H

subject to
r1:  +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_7
     +y1_2       +y1_3 +y1_4       +y1_6       = 10
r2:  +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 +x2_7
     +y2_1             +y2_4 +y2_5 +y2_6 +y2_7 = 20
r3:  +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 +x3_7
     +y3_1 +y3_2 +y3_3 +y3_4       +y3_6       = 15
r4:  +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 +x4_7
     +y4_1       +y4_3 +y4_4 +y4_5 +y4_6 +y4_7 = 40
r5:  +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 +x5_7
     +y5_1 +y5_2 +y5_3       +y5_5 +y5_6       = 50
r6:  +x6_1 +x6_2 +x6_3 +x6_4 +x6_5 +x6_6 +x6_7
           +y6_2       +y6_4 +y6_5 +y6_6 +y6_7 = 20
r7:  +x7_1 +x7_2 +x7_3 +x7_4 +x7_5 +x7_6 +x7_7
     +y7_1 +y7_2 +y7_3       +y7_5 +y7_6 +y7_7 = 10
```

```
r8:  +x8_1 +x8_2 +x8_3 +x8_4 +x8_5 +x8_6 +x8_7
     +y8_1       +y8_3 +y8_4 +y8_5       +y8_7 = 15
r9:  +x9_1 +x9_2 +x9_3 +x9_4 +x9_5 +x9_6 +x9_7
     +y9_1 +y9_2 +y9_3 +y9_4 +y9_5 +y9_6       = 15
r10: +x10_1+x10_2+x10_3+x10_4+x10_5+x10_6+x10_7
     +y10_1+y10_2+y10_3             +y10_6+y10_7= 35

a1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1
                      +x6_1 +x7_1 +x8_1 +x9_1 +x10_1
          +y2_1 +y3_1 +y4_1 +y5_1
                            +y7_1 +y8_1 +y9_1 +y10_1 = 30
a2: +x1_2 +x2_2 +x3_2 +x4_2 +x5_2
                      +x6_2 +x7_2 +x8_2 +x9_2 +x10_2
    +y1_2       +y3_2       +y5_2
                      +y6_2 +y7_2       +y9_2 +y10_2 = 40
a3: +x1_3 +x2_3 +x3_3 +x4_3 +x5_3
                      +x6_3 +x7_3 +x8_3 +x9_3 +x10_3
    +y1_3       +y3_3 +y4_3 +y5_3
                            +y7_3 +y8_3 +y9_3 +y10_3 = 10
a4: +x1_4 +x2_4 +x3_4 +x4_4 +x5_4
                      +x6_4 +x7_4 +x8_4 +x9_4 +x10_4
    +y1_4 +y2_4 +y3_4 +y4_4
                      +y6_4       +y8_4 +y9_4        = 20
a5: +x1_5 +x2_5 +x3_5 +x4_5 +x5_5
                      +x6_5 +x7_5 +x8_5 +x9_5 +x10_5
          +y2_5       +y4_5 +y5_5
                      +y6_5 +y7_5 +y8_5 +y9_5        = 50
a6: +x1_6 +x2_6 +x3_6 +x4_6 +x5_6
                      +x6_6 +x7_6 +x8_6 +x9_6 +x10_6
    +y1_6 +y2_6 +y3_6 +y4_6 +y5_6
                      +y6_6 +y7_6       +y9_6 +y10_6 = 40
a7: +x1_7 +x2_7 +x3_7 +x4_7 +x5_7
                      +x6_7 +x7_7 +x8_7 +x9_7 +x10_7
          +y2_7       +y4_7
                      +y6_7 +y7_7 +y8_7       +y10_7 = 40

h:     +y1_2  +y1_3  +y1_4        +y1_6
+y2_1                +y2_4  +y2_5 +y2_6  +y2_7
+y3_1  +y3_2  +y3_3  +y3_4        +y3_6
+y4_1         +y4_3  +y4_4  +y4_5 +y4_6  +y4_7
+y5_1  +y5_2  +y5_3         +y5_5 +y5_6
       +y6_2         +y6_4  +y6_5 +y6_6  +y6_7
+y7_1  +y7_2  +y7_3         +y7_5 +y7_6  +y7_7
+y8_1         +y8_3  +y8_4  +y8_5        +y8_7
+y9_1  +y9_2  +y9_3  +y9_4  +y9_5 +y9_6
```

```
+y10_1 +y10_2 +y10_3              +y10_6 +y10_7
-230H <= 0

k1:        +y1_2  +y1_3  +y1_4         +y1_6
                                              -230K1  <= 0
k2:  +y2_1               +y2_4  +y2_5  +y2_6  +y2_7
                                              -230K2  <= 0
k3:  +y3_1  +y3_2  +y3_3  +y3_4         +y3_6
                                              -230K3  <= 0
k4:  +y4_1         +y4_3  +y4_4  +y4_5  +y4_6  +y4_7
                                              -230K4  <= 0
k5:  +y5_1  +y5_2  +y5_3         +y5_5  +y5_6
                                              -230K5  <= 0
k6:        +y6_2         +y6_4  +y6_5  +y6_6  +y6_7
                                              -230K6  <= 0
k7:  +y7_1  +y7_2  +y7_3         +y7_5  +y7_6  +y7_7
                                              -230K7  <= 0
k8:  +y8_1         +y8_3  +y8_4  +y8_5         +y8_7
                                              -230K8  <= 0
k9:  +y9_1  +y9_2  +y9_3  +y9_4  +y9_5  +y9_6
                                              -230K9  <= 0
k10: +y10_1 +y10_2 +y10_3               +y10_6 +y10_7
                                              -230K10 <= 0

v1:       +y2_1 +y3_1 +y4_1 +y5_1       +y7_1
                            +y8_1 +y9_1 +y10_1 -230V1 <= 0
v2: +y1_2       +y3_2       +y5_2 +y6_2 +y7_2
                                  +y9_2 +y10_2 -230V2 <= 0
v3: +y1_3       +y3_3 +y4_3 +y5_3       +y7_3
                            +y8_3 +y9_3 +y10_3 -230V3 <= 0
v4: +y1_4 +y2_4 +y3_4 +y4_4       +y6_4
                            +y8_4 +y9_4        -230V4 <= 0
v5:       +y2_5       +y4_5 +y5_5 +y6_5 +y7_5
                            +y8_5 +y9_5        -230V5 <= 0
v6: +y1_6 +y2_6 +y3_6 +y4_6 +y5_6 +y6_6 +y7_6
                                  +y9_6 +y10_6 -230V6 <= 0
v7:       +y2_7       +y4_7       +y6_7 +y7_7
                            +y8_7       +y10_7 -230V7 <= 0
binary
K1
K2
K3
K4
K5
K6
```

```
K7
K8
K9
K10
V1
V2
V3
V4
V5
V6
V7
H

end
```

Ekkor az összköltség 4545, a szállítások pedig:

```
   No. Column name    Activity
------ ------------ -------------
     1 x1_1                   10
    13 x2_6                   20
    15 x3_1                   15
    27 x4_6                   20
    28 x4_7                   20
    30 x5_2                   30
    32 x5_4                   15
    33 x5_5                    5
    42 x6_7                   20
    44 x7_2                   10
    50 x8_1                    5
    54 x8_5                   10
    59 x9_3                   10
    60 x9_4                    5
    68 x10_5                  35
    75 K1          *           0
    81 K2          *           0
    87 K3          *           0
    94 K4          *           0
   100 K5          *           0
   106 K6          *           0
   113 K7          *           0
   119 K8          *           0
   126 K9          *           0
   132 K10         *           0
   133 V1          *           0
   134 V2          *           0
```

```
135 V3          *             0
136 V4          *             0
137 V5          *             0
138 V6          *             0
139 V7          *             0
140 H           *             0
```

Mivel az összes (csillagozott) bináris változó értéke 0, ezért a vízi szállítás nem optimális.

3.38. példa megoldása.

Az `a7` áruház jelöli a raktárakban hagyott mennyiségeket, a megadott költségekkel, az `a8-a13` pedig a 6 áruházba szállított raktározandó mennyiségeket, a megadott raktározási költségekkel, és az oda szállítás költségeivel. Mindegyik "fiktív" áruház igénye 150. Ezen kívül egy `r6` nevű raktárat kell hozzáadni az LP feladathoz, 900 egység kínálattal a kiegyensúlyozottság miatt. Ebből a raktárból az `a7-a13` áruházakba 0 költséggel szállíthatunk, ami a raktárak kihasználatlanságát jelenti, viszont az `a1-a6` áruházakba pedig nem lehet szállítani, mivel az igényeket teljesen ki kell elégíteni.

10.15. kód.

```
min
ktg:
+ 86x1_1+ 67x1_2+90x1_3 + 77x1_4 + 84x1_5 + 87x1_6 +66x1_7
+107x1_8+ 84x1_9+95x1_10+ 98x1_11+101x1_12+106x1_13
+ 64x2_1+ 98x2_2+58x2_3 + 55x2_4 + 65x2_5 + 76x2_6 +59x2_7
+ 85x2_8+115x2_9+63x2_10+ 76x2_11+ 82x2_12+ 95x2_13
+ 60x3_1+ 97x3_2+92x3_3 + 61x3_4 + 89x3_5 + 53x3_6 +69x3_7
+ 81x3_8+114x3_9+97x3_10+ 82x3_11+106x3_12+ 72x3_13
+ 70x4_1+ 60x4_2+82x4_3 + 73x4_4 + 92x4_5 + 87x4_6 +55x4_7
+ 91x4_8+ 77x4_9+87x4_10+ 94x4_11+109x4_12+106x4_13
+ 99x5_1+ 53x5_2+61x5_3 + 87x5_4 + 91x5_5 + 70x5_6 +80x5_7
+120x5_8+ 70x5_9+66x5_10+108x5_11+108x5_12+ 89x5_13
+  0x6_7+  0x6_8+0x6_9  +  0x6_10+  0x6_11+  0x6_12+ 0x6_13

subject to
r1: +x1_1 +x1_2 +x1_3 +x1_4  +x1_5  +x1_6
    +x1_7 +x1_8 +x1_9 +x1_10 +x1_11 +x1_12 +x1_13 = 150
r2: +x2_1 +x2_2 +x2_3 +x2_4 +x2_5  +x2_6
```

```
    +x2_7 +x2_8 +x2_9 +x2_10 +x2_11 +x2_12 +x2_13 = 150
r3: +x3_1 +x3_2 +x3_3 +x3_4  +x3_5  +x3_6
    +x3_7 +x3_8 +x3_9 +x3_10 +x3_11 +x3_12 +x3_13 = 150
r4: +x4_1 +x4_2 +x4_3 +x4_4  +x4_5  +x4_6
    +x4_7 +x4_8 +x4_9 +x4_10 +x4_11 +x4_12 +x4_13 = 150
r5: +x5_1 +x5_2 +x5_3 +x5_4  +x5_5  +x5_6
    +x5_7 +x5_8 +x5_9 +x5_10 +x5_11 +x5_12 +x5_13 = 150
r6: +x6_7 +x6_8 +x6_9 +x6_10 +x6_11 +x6_12 +x6_13 = 900

a1:  +x1_1  +x2_1  +x3_1  +x4_1  +x5_1         = 100
a2:  +x1_2  +x2_2  +x3_2  +x4_2  +x5_2         = 100
a3:  +x1_3  +x2_3  +x3_3  +x4_3  +x5_3         = 100
a4:  +x1_4  +x2_4  +x3_4  +x4_4  +x5_4         = 100
a5:  +x1_5  +x2_5  +x3_5  +x4_5  +x5_5         = 100
a6:  +x1_6  +x2_6  +x3_6  +x4_6  +x5_6         = 100
a7:  +x1_7  +x2_7  +x3_7  +x4_7  +x5_7  +x6_7  = 150
a8:  +x1_8  +x2_8  +x3_8  +x4_8  +x5_8  +x6_8  = 150
a9:  +x1_9  +x2_9  +x3_9  +x4_9  +x5_9  +x6_9  = 150
a10: +x1_10 +x2_10 +x3_10 +x4_10 +x5_10 +x6_10 = 150
a11: +x1_11 +x2_11 +x3_11 +x4_11 +x5_11 +x6_11 = 150
a12: +x1_12 +x2_12 +x3_12 +x4_12 +x5_12 +x6_12 = 150
a13: +x1_13 +x2_13 +x3_13 +x4_13 +x5_13 +x6_13 = 150

end
```

Az optimális megoldás:

```
No.  Column name     Activity
---- ------------  -------------
   2  x1_2                   50
   5  x1_5                   50
   7  x1_7                   50
  17  x2_4                  100
  18  x2_5                   50
  27  x3_1                   50
  32  x3_6                  100
  40  x4_1                   50
  46  x4_7                  100
  54  x5_2                   50
  55  x5_3                  100
  67  x6_8                  150
  68  x6_9                  150
  69  x6_10                 150
  70  x6_11                 150
  71  x6_12                 150
  72  x6_13                 150
```

Tehát minden fennmaradó mennyiség a raktárakban marad, és semmilyen többlet nem raktározódik az áruházakban. Az összköltség: 45650.

3.39. példa megoldása.

Jelölje x_{ij} az i. földről a j. raktárba szállított kukorica kezdeti mennyiségét, azaz amennyit a szállító járműbe a kukoricaföldön bepakolnak. Jelölje y_{ij} az i. földről a j. raktárba végül beérkező kukorica-mennyiséget. Legyen F a földutak halmaza, tehát amennyiben $(i, j) \in F$, akkor az i. föld és a j. raktár közötti út földút. Az eddigiekből tudjuk, hogy amennyiben a szállítás földúton valósul meg ($(i, j) \in F$), akkor $y_{ij} =$ 0,95 x_{ij}, különben pedig $y_{ij} = x_{ij}$. Jelölje megint c_{ij} a szállítási költségeket az i. földtől a j. raktárba, S_i pedig az i. földön megtermelt mennyiséget. Ezenkívül pedig legyen p_j a j. raktárba a piacról vásárolt búza mennyisége.

10.16. LP felírás.

$$
\begin{array}{rll}
\min \quad \sum_{i=1}^{6}\sum_{j=1}^{5} c_{ij}x_{ij} + 100p_j & & \\
\sum_{j=1}^{5} x_{ij} & = S_i & \forall i \\
\sum_{i=1}^{6} y_{ij} + p_j & \leq 80 & \forall j \\
0,95x_{ij} - y_{ij} & = 0 & \forall (i,j) \in F \\
x_{ij} - y_{ij} & = 0 & \forall (i,j) \notin F \\
\forall i,j : x_{ij}, y_{ij}, p_j & \geq 0 &
\end{array}
$$

A kódban az y_{ij} változókat egyedül a földutak esetén vezettük be, a rendes utak esetén a kifelé szállításokat is x_{ij}-vel jelöltük.

10.17. kód.

```
min
ktg:
+ 70x1_1 +109x1_2 +106x1_3 + 99x1_4 + 88x1_5
+ 56x2_1 + 25x2_2 + 22x2_3 +105x2_4 + 21x2_5
+ 56x3_1 + 28x3_2 + 32x3_3 +119x3_4 + 32x3_5
+114x4_1 + 94x4_2 + 73x4_3 + 33x4_4 +116x4_5
+ 69x5_1 + 90x5_2 + 56x5_3 + 74x5_4 + 32x5_5
+ 85x6_1 + 30x6_2 + 87x6_3 + 79x6_4 + 76x6_5
```

```
+100p1   +100p2   +100p3   +100p4   +100p5

subject to
f1: +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 = 40
f2: +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 = 70
f3: +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 = 30
f4: +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 =100
f5: +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 = 80
f6: +x6_1 +x6_2 +x6_3 +x6_4 +x6_5 = 20

r1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1+p1 <=80
r2: +x1_2 +y2_2 +x3_2 +x4_2 +x5_2 +y6_2+p2 <=80
r3: +x1_3 +y2_3 +y3_3 +x4_3 +x5_3 +x6_3+p3 <=80
r4: +x1_4 +x2_4 +x3_4 +y4_4 +x5_4 +x6_4+p4 <=80
r5: +x1_5 +x2_5 +y3_5 +x4_5 +x5_5 +x6_5+p5 <=80

v2_2: 0.95x2_2 -y2_2 = 0
v2_3: 0.95x2_3 -y2_3 = 0
v3_3: 0.95x3_3 -y3_3 = 0
v3_5: 0.95x3_5 -y3_5 = 0
v4_4: 0.95x4_4 -y4_4 = 0
v6_2: 0.95x6_2 -y6_2 = 0

end
```

Az optimális megoldás:

```
 No. Column name     Activity
---- ------------  -------------
   1 x1_1                    40
   7 x2_2               2.40997
   8 x2_3                 67.59
  12 x3_2                    30
  18 x4_3               15.7895
  19 x4_4               84.2105
  25 x5_5                    80
  27 x6_2                    20
  36 y2_2               2.28947
  37 y6_2                    19
  38 y2_3               64.2105
  40 y4_4                    80
```

Tehát alapesetben optimálisan 174,21 mázsa kukoricát szállítunk földúton, amiből összesen 165,5 mázsa érkezik meg a raktárakba (8,71 mázsa veszik el út közben). A raktárakba összesen 331,29 mázsa érkezik meg, 12278,81-es költséggel.

Az első módosításban feltesszük, hogy az összes földekről induló szállítás (tehát minden x_{ij}) egészértékű. Ezért a CPLEX LP kód végéhez (az `end` elé közvetlenül) a következőket kell hozzáadni:

```
Integer
x1_1
x2_1
x3_1
x4_1
x5_1
x6_1
x1_2
x2_2
x3_2
x5_2
x6_2
x1_3
x2_3
x3_3
x4_3
x5_3
x6_3
x1_4
x2_4
x3_4
x4_4
x5_4
x6_4
x1_5
x2_5
x3_5
x4_5
x5_5
x6_5
```

Ekkor az optimális megoldás a következőképpen alakul:

```
 No.  Column name    Activity
---- ------------  -------------
   1 x1_1                    40
   7 x2_2                     3
   8 x2_3                    67
  12 x3_2                    30
  18 x4_3                    16
```

```
19 x4_4                    84
25 x5_5                    80
27 x6_2                    20
36 y2_2                  2.85
37 y6_2                    19
38 y2_3                 63.65
40 y4_4                  79.8
```

Tehát most 174 mázsa kukorica indul el földúton, amiből 8,7 mázsa veszik el, így 165,3 érkezik meg a raktárakba. Összesen 331,3 mázsa kukoricát tudtunk szállítani, 12289 összköltséggel.

Mivel már ekkor is 331,3 mázsa kukorica kerül a raktárakba, ezért a legalább 330 mázsás előírás esetén a szállítási terv nem változik. Elég, ha a 340-es esettel foglalkozunk.

Ehhez egy olyan korlátot kell hozzáadni a kódhoz, hogy az összes beérkezett kukorica mennyisége legyen több mint 340:

$$\sum_{i=1}^{6}\sum_{j=1}^{5} y_{ij} + p_j \geq 340 \ .$$

10.18. kód.

```
min
ktg:
+ 70x1_1 +109x1_2 +106x1_3 + 99x1_4 + 88x1_5
+ 56x2_1 + 25x2_2 + 22x2_3 +105x2_4 + 21x2_5
+ 56x3_1 + 28x3_2 + 32x3_3 +119x3_4 + 32x3_5
+114x4_1 + 94x4_2 + 73x4_3 + 33x4_4 +116x4_5
+ 69x5_1 + 90x5_2 + 56x5_3 + 74x5_4 + 32x5_5
+ 85x6_1 + 30x6_2 + 87x6_3 + 79x6_4 + 76x6_5
+100p1   +100p2   +100p3   +100p4   +100p5

subject to
f1: +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 = 40
f2: +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 = 70
f3: +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 = 30
f4: +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 =100
f5: +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 = 80
f6: +x6_1 +x6_2 +x6_3 +x6_4 +x6_5 = 20

r1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1+p1 <=80
r2: +x1_2 +y2_2 +x3_2 +x4_2 +x5_2 +y6_2+p2 <=80
```

```
r3: +x1_3 +y2_3 +y3_3 +x4_3 +x5_3 +x6_3+p3 <=80
r4: +x1_4 +x2_4 +x3_4 +y4_4 +x5_4 +x6_4+p4 <=80
r5: +x1_5 +x2_5 +y3_5 +x4_5 +x5_5 +x6_5+p5 <=80

v2_2: 0.95x2_2 -y2_2 = 0
v2_3: 0.95x2_3 -y2_3 = 0
v3_3: 0.95x3_3 -y3_3 = 0
v3_5: 0.95x3_5 -y3_5 = 0
v4_4: 0.95x4_4 -y4_4 = 0
v6_2: 0.95x6_2 -y6_2 = 0


mm:  +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1 +p1
     +x1_2 +y2_2 +x3_2 +x4_2 +x5_2 +y6_2 +p2
     +x1_3 +y2_3 +y3_3 +x4_3 +x5_3 +x6_3 +p3
     +x1_4 +x2_4 +x3_4 +y4_4 +x5_4 +x6_4 +p4
     +x1_5 +x2_5 +y3_5 +x4_5 +x5_5 +x6_5 +p5
>= 340

Integer
x1_1
x_1
x3_1
x4_1
x5_1
x6_1
x1_2
x2_2
x3_2
x4_2
x5_2
x6_2
x1_3
x2_3
x3_3
x4_3
x5_3
x6_3
x1_4
x2_4
x3_4
x4_4
x5_4
x6_4
x1_5
```

```
x2_5
x3_5
x4_5
x5_5
x6_5

end
```

Az optimális megoldás:

```
 No.  Column name     Activity
---- ------------  -------------
   1 x1_1                     40
   7 x2_2                      3
   8 x2_3                     67
  12 x3_2                     30
  18 x4_3                     16
  19 x4_4                     84
  25 x5_5                     80
  27 x6_2                     20
  32 p2                      8.7
  36 y2_2                   2.85
  37 y6_2                     19
  38 y2_3                  63.65
  40 y4_4                   79.8
```

Most ugyanaz marad a szállítási terv, mint az előző megoldásban, annyi módosítással, hogy a 8,7 elveszett búza mennyiségét a piacról pótolják, és a 2. raktárban helyezik el, így az összköltség 13159 lesz.

10.3. Maximális folyam feladatok

4.14. példa megoldása.

Ezt a feladatot hasonlóan kell megoldani, mint a 4.7. példát.

10.19. kód.

```
Max
f10_1

subject to

csp1:  +f10_1 -x1_2  -x1_3 -x1_4              = 0
```

```
csp2:  +x1_2  +x6_2  -x2_3 -x2_4 -x2_7        = 0
csp3:  +x1_3  +x2_3  +x5_3 -x3_4 -x3_7 -x3_9  = 0
csp4:  +x1_4  +x2_4  +x3_4 -x4_5 -x4_8        = 0
csp5:  +x4_5  +x7_5  -x5_3 -x5_6              = 0
csp6:  +x5_6  -x6_2  -x6_7 -x6_8              = 0
csp7:  +x2_7  +x3_7  +x6_7 +x9_7 -x7_5 -x7_10 = 0
csp8:  +x4_8  +x6_8  +x9_8 -x8_10             = 0
csp9:  +x3_9  -x9_7  -x9_8                    = 0
csp10: +x7_10 +x8_10 -f10_1                   = 0

kap1_2:  x1_2  <=  72
kap1_3:  x1_3  <=  30
kap1_4:  x1_4  <=  80
kap2_3:  x2_3  <=  90
kap2_4:  x2_4  <=  74
kap2_7:  x2_7  <=  70
kap3_4:  x3_4  <=  88
kap3_7:  x3_7  <=  33
kap3_9:  x3_9  <=  50
kap4_5:  x4_5  <=  42
kap4_8:  x4_8  <=  73
kap5_3:  x5_3  <=  12
kap5_6:  x5_6  <= 120
kap6_2:  x6_2  <=  30
kap6_7:  x6_7  <=  25
kap6_8:  x6_8  <=  75
kap7_5:  x7_5  <=  30
kap7_10: x7_10 <=  90
kap8_10: x8_10 <= 100
kap9_7:  x9_7  <=  23
kap9_8:  x9_8  <=  84

end
```

Az optimális megoldás:

```
Status:     OPTIMAL
Objective:  obj = 182 (MAXimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             0             1
     2 csp2         NS             0         < eps
     3 csp3         NS             0         < eps
     4 csp4         NS             0         < eps
     5 csp5         NS             0         < eps
```

```
     6 csp6         NS             0            < eps
     7 csp7         NS             0            < eps
     8 csp8         NS             0            < eps
     9 csp9         NS             0            < eps
    10 csp10        B              0
    11 kap1_2       NU            72                1
    12 kap1_3       NU            30                1
    13 kap1_4       NU            80                1
    14 kap2_3       B              2
    15 kap2_4       B              0
    16 kap2_7       NU            70            < eps
    17 kap3_4       B              0
    18 kap3_7       B             12
    19 kap3_9       B             20
    20 kap4_5       B              7
    21 kap4_8       NU            73            < eps
    22 kap5_3       B              0
    23 kap5_6       B              7
    24 kap6_2       B              0
    25 kap6_7       B              0
    26 kap6_8       B              7
    27 kap7_5       B              0
    28 kap7_10      B             82
    29 kap8_10      NU           100            < eps
    30 kap9_7       B              0
    31 kap9_8       B             20

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 f10_1        B            182
     2 x1_2         B             72
     3 x1_3         B             30
     4 x1_4         B             80
     5 x6_2         NL             0            < eps
     6 x2_3         B              2
     7 x2_4         NL             0            < eps
     8 x2_7         B             70
     9 x5_3         NL             0            < eps
    10 x3_4         NL             0            < eps
    11 x3_7         B             12
    12 x3_9         B             20
    13 x4_5         B              7
    14 x4_8         B             73
    15 x7_5         NL             0            < eps
    16 x5_6         B              7
```

```
   17 x6_7          NL             0         < eps
   18 x6_8          B              7
   19 x9_7          NL             0         < eps
   20 x7_10         B             82
   21 x9_8          B             20
   22 x8_10         B            100
```

Tehát a maximális folyam értéke 182, a szállítások pedig az eredményfülről leolvashatóak.

4.15. példa megoldása.

Ezt a feladatot is ugyanúgy kell megoldani, mint az előző 4.14. példát.

10.20. kód.

```
Max
f12_1

subject to

csp1:  +f12_1 -x1_2  -x1_3   -x1_4          = 0
csp2:  +x1_2  -x2_3  -x2_5   -x2_6          = 0
csp3:  +x1_3  +x2_3  -x3_4   -x3_6          = 0
csp4:  +x1_4  +x3_4  -x4_7   -x4_8  -x4_12 = 0
csp5:  +x2_5  +x7_5  -x5_9   -x5_10         = 0
csp6:  +x2_6  +x3_6  -x6_7   -x6_10         = 0
csp7:  +x4_7  +x6_7  -x7_5   -x7_9          = 0
csp8:  +x4_8  -x8_9  -x8_11                 = 0
csp9:  +x5_9  +x7_9  +x8_9   -x9_12         = 0
csp10: +x5_10 +x6_10 -x10_11                = 0
csp11: +x8_11 +x10_11-x11_12                = 0
csp12: +x4_12 +x9_12 +x11_12 -f12_1         = 0

kap1_2:   x1_2   <=  45
kap1_3:   x1_3   <=  42
kap1_4:   x1_4   <=  38
kap2_3:   x2_3   <=  92
kap2_5:   x2_5   <=  71
kap2_6:   x2_6   <=  66
kap3_4:   x3_4   <=  48
kap3_6:   x3_6   <=  30
kap4_7:   x4_7   <=  66
kap4_8:   x4_8   <=  88
kap4_12:  x4_12  <=  15
```

```
kap5_9:    x5_9   <=  58
kap5_10:   x5_10  <=  67
kap6_7:    x6_7   <=  41
kap6_10:   x6_10  <=  94
kap7_5:    x7_5   <=  77
kap7_9:    x7_9   <=  58
kap8_9:    x8_9   <=  32
kap8_11:   x8_11  <=  96
kap9_12:   x9_12  <=  32
kap10_11:  x10_11 <= 100
kap11_12:  x11_12 <=  60

end
```

Az optimális megoldás:

```
Status:     OPTIMAL
Objective:  obj = 107 (MAXimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             0         < eps
     2 csp2         NS             0         < eps
     3 csp3         NS             0         < eps
     4 csp4         NS             0         < eps
     5 csp5         NS             0         < eps
     6 csp6         NS             0         < eps
     7 csp7         NS             0         < eps
     8 csp8         NS             0         < eps
     9 csp9         NS             0         < eps
    10 csp10        NS             0         < eps
    11 csp11        B              0
    12 csp12        NS             0            -1
    13 kap1_2       NU            45         < eps
    14 kap1_3       B             24
    15 kap1_4       NU            38         < eps
    16 kap2_3       B              0
    17 kap2_5       B              0
    18 kap2_6       B             45
    19 kap3_4       B              0
    20 kap3_6       B             24
    21 kap4_7       B             23
    22 kap4_8       B              0
    23 kap4_12      NU            15             1
    24 kap5_9       B              0
```

```
    25 kap5_10      B              0
    26 kap6_7       B              9
    27 kap6_10      B             60
    28 kap7_5       B              0
    29 kap7_9       B             32
    30 kap8_9       B              0
    31 kap8_11      B              0
    32 kap9_12      NU            32             1
    33 kap10_11     B             60
    34 kap11_12     NU            60             1

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 f12_1        B            107
     2 x1_2         B             45
     3 x1_3         B             24
     4 x1_4         B             38
     5 x2_3         NL             0         < eps
     6 x2_5         NL             0         < eps
     7 x2_6         B             45
     8 x3_4         NL             0         < eps
     9 x3_6         B             24
    10 x4_7         B             23
    11 x4_8         NL             0         < eps
    12 x4_12        B             15
    13 x7_5         NL             0         < eps
    14 x5_9         B              0
    15 x5_10        NL             0         < eps
    16 x6_7         B              9
    17 x6_10        B             60
    18 x7_9         B             32
    19 x8_9         NL             0         < eps
    20 x8_11        B              0
    21 x9_12        B             32
    22 x10_11       B             60
    23 x11_12       B             60
```

A maximális mennyiség, amit átáramolhatunk az 1. csúcsból a 12. csúcsba 107 egység.

4.16. példa megoldása.

Ekkor a következők lesznek a bővítési költségek:

	2	3	4	5	6	7	8	9	10	11	12
1	20	8	20								
2		16		16	10						
3			20		20						
4						16	10				10
5								8	20		
6						16			10		
7				8				8			
8								16		16	
9											20
10										16	
11											20

10.21. kód.

```
Min

+20b1_2  + 8b1_3 +20b1_4 +16b2_5 +10b2_6  +16b4_7 +10b4_8
+ 8b5_9  +20b5_10+16b8_11+10b4_12+16b2_3  +20b3_4 + 8b7_5
+20b3_6  +16b6_7 + 8b7_9 +10b6_10+16b10_11+20b9_12+16b8_9
+20b11_12

subject to

csp1:  +f12_1 -x1_2  -x1_3  -x1_4        = 0
csp2:  +x1_2  -x2_3  -x2_5  -x2_6        = 0
csp3:  +x1_3  +x2_3  -x3_4  -x3_6        = 0
csp4:  +x1_4  +x3_4  -x4_7  -x4_8 -x4_12 = 0
csp5:  +x2_5  +x7_5  -x5_9  -x5_10       = 0
csp6:  +x2_6  +x3_6  -x6_7  -x6_10       = 0
csp7:  +x4_7  +x6_7  -x7_5  -x7_9        = 0
csp8:  +x4_8  -x8_9  -x8_11              = 0
csp9:  +x5_9  +x7_9  +x8_9  -x9_12       = 0
csp10: +x5_10 +x6_10 -x10_11             = 0
csp11: +x8_11 +x10_11-x11_12             = 0
csp12: +x4_12 +x9_12 +x11_12-f12_1       = 0

mfolyam: f12_1 >= 200
kap1_2:   x1_2  -b1_2   <=45
kap1_3:   x1_3  -b1_3   <=42
```

```
kap1_4:   x1_4  -b1_4   <=38
kap2_3:   x2_3  -b2_3   <=92
kap2_5:   x2_5  -b2_5   <=71
kap2_6:   x2_6  -b2_6   <=66
kap3_4:   x3_4  -b3_4   <=48
kap3_6:   x3_6  -b3_6   <=30
kap4_7:   x4_7  -b4_7   <=66
kap4_8:   x4_8  -b4_8   <=88
kap4_12:  x4_12 -b4_12  <=15
kap5_9:   x5_9  -b5_9   <=58
kap5_10:  x5_10 -b5_10  <=67
kap6_7:   x6_7  -b6_7   <=41
kap6_10:  x6_10 -b6_10  <=94
kap7_5:   x7_5  -b7_5   <=77
kap7_9:   x7_9  -b7_9   <=58
kap8_9:   x8_9  -b8_9   <=32
kap8_11:  x8_11 -b8_11  <=96
kap9_12:  x9_12 -b9_12  <=32
kap10_11: x10_11-b10_11 <=100
kap11_12: x11_12-b11_12 <=60

end
```

Az optimális megoldás:

```
Status:     OPTIMAL
Objective:  obj = 1998 (MINimum)

   No.   Row name    St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             0           -20
     2 csp2         NS             0         < eps
     3 csp3         NS             0           -12
     4 csp4         NS             0         < eps
     5 csp5         NS             0         < eps
     6 csp6         NS             0         < eps
     7 csp7         NS             0         < eps
     8 csp8         NS             0         < eps
     9 csp9         NS             0         < eps
    10 csp10        NS             0         < eps
    11 csp11        B              0
    12 csp12        NS             0            10
    13 mfolyam      NL           200            30
    14 kap1_2       NU            45           -20
    15 kap1_3       NU            42            -8
```

16	kap1_4	NU	38	-20
17	kap2_3	B	0	
18	kap2_5	B	0	
19	kap2_6	B	62	
20	kap3_4	NU	48	-12
21	kap3_6	NU	30	-12
22	kap4_7	B	0	
23	kap4_8	B	0	
24	kap4_12	NU	15	-10
25	kap5_9	B	0	
26	kap5_10	B	0	
27	kap6_7	B	32	
28	kap6_10	B	60	
29	kap7_5	B	0	
30	kap7_9	B	32	
31	kap8_9	B	0	
32	kap8_11	B	0	
33	kap9_12	NU	32	-10
34	kap10_11	B	60	
35	kap11_12	NU	60	-10

No.	Column name	St	Activity	Marginal
1	b1_2	B	17	
2	b1_3	B	36	
3	b1_4	B	22	
4	b2_5	NL	0	16
5	b2_6	NL	0	10
6	b4_7	NL	0	16
7	b4_8	NL	0	10
8	b5_9	NL	0	8
9	b5_10	NL	0	20
10	b8_11	NL	0	16
11	b4_12	B	93	
12	b2_3	NL	0	16
13	b3_4	NL	0	8
14	b7_5	NL	0	8
15	b3_6	NL	0	8
16	b6_7	NL	0	16
17	b7_9	NL	0	8
18	b6_10	NL	0	10
19	b10_11	NL	0	16
20	b9_12	NL	0	10
21	b8_9	NL	0	16
22	b11_12	NL	0	10

```
23 f12_1        B            200
24 x1_2         B             62
25 x1_3         B             78
26 x1_4         B             60
27 x2_3         NL             0             12
28 x2_5         NL             0          < eps
29 x2_6         B             62
30 x3_4         B             48
31 x3_6         B             30
32 x4_7         NL             0          < eps
33 x4_8         NL             0          < eps
34 x4_12        B            108
35 x7_5         NL             0          < eps
36 x5_9         NL             0          < eps
37 x5_10        B              0
38 x6_7         B             32
39 x6_10        B             60
40 x7_9         B             32
41 x8_9         NL             0          < eps
42 x8_11        B              0
43 x9_12        B             32
44 x10_11       B             60
45 x11_12       B             60
```

Tehát a bővítési összköltség 1998.

4.17. példa megoldása.

Ekkor a következők lesznek az átfolyatható kapacitások:

	1	2	3	4	5	6	7	8	9	10	11	12
1		45	42	38								
2	45		92		71	66						
3	42	92		48		30						
4	38		48				66	88				15
5		71					77		58	67		
6		66	30				41			94		
7				66	77	41			58			
8				88					32		96	
9					58		58	32				32
10					67	94					100	
11								96		100		60
12				15					32		60	

A modellhez ebben az esetben figyelembe kell venni, hogy egy csövön (élen) egyszerre csak egyik irányba lehet a folyadékot folyatni. Bár nem eredményezhet jobb megoldást az oda-vissza folyatás egy élen, alternatív optimális megoldás még lehet.

A felmerülő problémát kétféleképpen lehet feloldani. Egyrészt nettósíthatjuk a szállításokat. Ezen azt értjük, hogyha pl. két csúcspont között odafele 10 egység áramlik, visszafele pedig 3, akkor átrendezzük az átfolyó mennyiségeket: odafele 7 egység áramlik, visszafele pedig semmi. Könnyen belátható, hogy ez a művelet nem érinti a többi élen átfolyó mennyiséget. Másik lehetséges megoldásként előírhatjuk az LP modellben, hogy minden viszonylatban csak maximum az egyik irányban használhatjuk a csővezetéket. Azt, hogy ezt megtegyük egész értékű változókat kell bevezetni; ezt a modell nagy hálózatokra már nem biztos, hogy belátható időn belül meg tudjuk oldani. Gyakorlásképpen mégis megadjuk ezt a modellt is.

Ebben a modellben minden él esetén korlátozni kell, hogy a folyadék csak egy irányba haladhasson. Ezért minden élhez bevezetünk egy bináris változót, ami akkor veszi fel az 1-et, ha az adott élen folyik valamilyen mennyiség. Jelölje y_{ab} az ab élhez tartozó bináris változót:

$$y_{ab} \leq x_{ab} \leq M y_{ab} \quad \forall a, b$$

Ahol M egy kellően nagy szám, és x_{ab} jelöli a csövön átfolyatott mennyiséget. Ahhoz, hogy egyszerre csak egy irányba lehessen folyatni, a bináris változók segítségével lehet korlátozni, hogy legfeljebb csak az él egyik irányán legyen pozitív mennyiség:

$$y_{ab} + y_{ba} \leq 1 \quad \forall a, b$$

Ekkor a feladat kódja:

10.22. kód. `Max`
```
f12_1

subject to

csp1:  +f12_1 -x1_2  -x1_3  -x1_4                              = 0
csp2:  +x1_2  +x3_2  +x5_2  +x6_2  -x2_3  -x2_5  -x2_6         = 0
csp3:  +x1_3  +x2_3  +x4_3  +x6_3  -x3_2  -x3_4  -x3_6         = 0
csp4:  +x1_4  +x3_4  +x7_4  +x8_4  -x4_3  -x4_7  -x4_8 -x4_12 = 0
csp5:  +x2_5  +x7_5  +x9_5  +x10_5 -x5_2  -x5_7  -x5_9 -x5_10 = 0
csp6:  +x2_6  +x3_6  +x7_6  +x10_6 -x6_2  -x6_3  -x6_7 -x6_10 = 0
csp7:  +x4_7  +x5_7  +x6_7  +x9_7  -x7_4  -x7_5  -x7_6 -x7_9  = 0
csp8:  +x4_8  +x9_8  +x11_8 -x8_4  -x8_9  -x8_11               = 0
csp9:  +x5_9  +x7_9  +x8_9  -x9_5  -x9_7  -x9_8  -x9_12        = 0
csp10: +x5_10 +x6_10 +x11_10-x10_5 -x10_6 -x10_11              = 0
csp11: +x8_11 +x10_11-x11_8 -x11_10-x11_12                     = 0
csp12: +x4_12 +x9_12 +x11_12-f12_1                             = 0


kap1_2:   x1_2   <= 45
kap1_3:   x1_3   <= 42
kap1_4:   x1_4   <= 38
kap2_3:   x2_3   <= 92
kap2_5:   x2_5   <= 71
kap2_6:   x2_6   <= 66
kap3_2:   x3_2   <= 92
kap3_4:   x3_4   <= 48
kap3_6:   x3_6   <= 30
kap4_3:   x4_3   <= 48
kap4_7:   x4_7   <= 66
```

```
kap4_8:    x4_8   <= 88
kap4_12:   x4_12  <= 15
kap5_2:    x5_2   <= 71
kap5_7:    x5_7   <= 77
kap5_9:    x5_9   <= 58
kap5_10:   x5_10  <= 67
kap6_2:    x6_2   <= 66
kap6_3:    x6_3   <= 30
kap6_7:    x6_7   <= 41
kap6_10:   x6_10  <= 94
kap7_4:    x7_4   <= 66
kap7_5:    x7_5   <= 77
kap7_6:    x7_6   <= 41
kap7_9:    x7_9   <= 58
kap8_4:    x8_4   <= 88
kap8_9:    x8_9   <= 32
kap8_11:   x8_11  <= 96
kap9_5:    x9_5   <= 58
kap9_7:    x9_7   <= 58
kap9_8:    x9_8   <= 32
kap9_12:   x9_12  <= 32
kap10_5:   x10_5  <= 67
kap10_6:   x10_6  <= 94
kap10_11:  x10_11 <= 100
kap11_8:   x11_8  <= 96
kap11_10:  x11_10 <= 100
kap11_12:  x11_12 <= 60

y2_3_al:    y2_3   -x2_3    <= 0
y2_5_al:    y2_5   -x2_5    <= 0
y2_6_al:    y2_6   -x2_6    <= 0
y3_2_al:    y3_2   -x3_2    <= 0
y3_4_al:    y3_4   -x3_4    <= 0
y3_6_al:    y3_6   -x3_6    <= 0
y4_3_al:    y4_3   -x4_3    <= 0
y4_7_al:    y4_7   -x4_7    <= 0
y4_8_al:    y4_8   -x4_8    <= 0
y5_2_al:    y5_2   -x5_2    <= 0
y5_7_al:    y5_7   -x5_7    <= 0
y5_9_al:    y5_9   -x5_9    <= 0
y5_10_al:   y5_10  -x5_10   <= 0
y6_2_al:    y6_2   -x6_2    <= 0
y6_3_al:    y6_3   -x6_3    <= 0
y6_7_al:    y6_7   -x6_7    <= 0
y6_10_al:   y6_10  -x6_10   <= 0
```

```
y7_4_al:   y7_4   -x7_4   <= 0
y7_5_al:   y7_5   -x7_5   <= 0
y7_6_al:   y7_6   -x7_6   <= 0
y7_9_al:   y7_9   -x7_9   <= 0
y8_4_al:   y8_4   -x8_4   <= 0
y8_9_al:   y8_9   -x8_9   <= 0
y8_11_al:  y8_11  -x8_11  <= 0
y9_5_al:   y9_5   -x9_5   <= 0
y9_7_al:   y9_7   -x9_7   <= 0
y9_8_al:   y9_8   -x9_8   <= 0
y10_5_al:  y10_5  -x10_5  <= 0
y10_6_al:  y10_6  -x10_6  <= 0
y10_11_al: y10_11 -x10_11 <= 0
y11_8_al:  y11_8  -x11_8  <= 0
y11_10_al: y11_10 -x11_10 <= 0

y2_3_fel:  1000 y2_3   -x2_3   >=0
y2_5_fel:  1000 y2_5   -x2_5   >=0
y2_6_fel:  1000 y2_6   -x2_6   >=0
y3_2_fel:  1000 y3_2   -x3_2   >=0
y3_4_fel:  1000 y3_4   -x3_4   >=0
y3_6_fel:  1000 y3_6   -x3_6   >=0
y4_3_fel:  1000 y4_3   -x4_3   >=0
y4_7_fel:  1000 y4_7   -x4_7   >=0
y4_8_fel:  1000 y4_8   -x4_8   >=0
y5_2_fel:  1000 y5_2   -x5_2   >=0
y5_7_fel:  1000 y5_7   -x5_7   >=0
y5_9_fel:  1000 y5_9   -x5_9   >=0
y5_10_fel: 1000 y5_10  -x5_10  >=0
y6_2_fel:  1000 y6_2   -x6_2   >=0
y6_3_fel:  1000 y6_3   -x6_3   >=0
y6_7_fel:  1000 y6_7   -x6_7   >=0
y6_10_fel: 1000 y6_10  -x6_10  >=0
y7_4_fel:  1000 y7_4   -x7_4   >=0
y7_5_fel:  1000 y7_5   -x7_5   >=0
y7_6_fel:  1000 y7_6   -x7_6   >=0
y7_9_fel:  1000 y7_9   -x7_9   >=0
y8_4_fel:  1000 y8_4   -x8_4   >=0
y8_9_fel:  1000 y8_9   -x8_9   >=0
y8_11_fel: 1000 y8_11  -x8_11  >=0
y9_5_fel:  1000 y9_5   -x9_5   >=0
y9_7_fel:  1000 y9_7   -x9_7   >=0
y9_8_fel:  1000 y9_8   -x9_8   >=0
y10_5_fel: 1000 y10_5  -x10_5  >=0
y10_6_fel: 1000 y10_6  -x10_6  >=0
```

```
y10_11_fel:1000 y10_11 -x10_11 >=0
y11_8_fel: 1000 y11_8  -x11_8  >=0
y11_10_fel:1000 y11_10 -x11_10 >=0

el2_3:   y2_3   + y3_2   <= 1
el2_5:   y2_5   + y5_2   <= 1
el2_6:   y2_6   + y6_2   <= 1
el3_4:   y3_4   + y4_3   <= 1
el3_6:   y3_6   + y6_3   <= 1
el4_7:   y4_7   + y7_4   <= 1
el4_8:   y4_8   + y8_4   <= 1
el5_7:   y5_7   + y7_5   <= 1
el5_9:   y5_9   + y9_5   <= 1
el5_10:  y5_10  + y10_5  <= 1
el6_7:   y6_7   + y7_6   <= 1
el6_10:  y6_10  + y10_6  <= 1
el7_9:   y7_9   + y9_7   <= 1
el8_9:   y8_9   + y9_8   <= 1
el8_11:  y8_11  + y11_8  <= 1
el10_11: y10_11 + y11_10 <= 1

Binary

y2_3
y2_5
y2_6
y3_2
y3_4
y3_6
y4_3
y4_7
y4_8
y5_2
y5_7
y5_9
y5_10
y6_2
y6_3
y6_7
y6_10
y7_4
y7_5
y7_6
y7_9
y8_4
```

```
y8_9
y8_11
y9_5
y9_7
y9_8
y10_5
y10_6
y10_11
y11_8
y11_10


end
```

Az optimális megoldás:

```
Status:     INTEGER OPTIMAL
Objective:  obj = 107 (MAXimum)

No.      Row name        Activity     Upper bound
------ ------------   ------------- -------------
     1 csp1                       0             =
     2 csp2                       0             =
     3 csp3                       0             =
     4 csp4                       0             =
     5 csp5                       0             =
     6 csp6                       0             =
     7 csp7                       0             =
     8 csp8                       0             =
     9 csp9                       0             =
    10 csp10                      0             =
    11 csp11                      0             =
    12 csp12                      0             =
    13 kap1_2                    45            45
    14 kap1_3                    24            42
    15 kap1_4                    38            38
    16 kap2_3                     0            92
    17 kap2_5                     0            71
    18 kap2_6                    45            66
    19 kap3_2                     0            92
    20 kap3_4                     0            48
    21 kap3_6                    24            30
    22 kap4_3                     0            48
    23 kap4_7                     0            66
    24 kap4_8                    23            88
```

25	kap4_12	15	15
26	kap5_2	0	71
27	kap5_7	0	77
28	kap5_9	0	58
29	kap5_10	0	67
30	kap6_2	0	66
31	kap6_3	0	30
32	kap6_7	32	41
33	kap6_10	37	94
34	kap7_4	0	66
35	kap7_5	0	77
36	kap7_6	0	41
37	kap7_9	32	58
38	kap8_4	0	88
39	kap8_9	0	32
40	kap8_11	23	96
41	kap9_5	0	58
42	kap9_7	0	58
43	kap9_8	0	32
44	kap9_12	32	32
45	kap10_5	0	67
46	kap10_6	0	94
47	kap10_11	37	100
48	kap11_8	0	96
49	kap11_10	0	100
50	kap11_12	60	60
51	y2_3_al	0	0
52	y2_5_al	0	0
53	y2_6_al	44	0
54	y3_2_al	0	0
55	y3_4_al	0	0
56	y3_6_al	-23	0
57	y4_3_al	0	0
58	y4_7_al	0	0
59	y4_8_al	-22	0
60	y5_2_al	0	0
61	y5_7_al	0	0
62	y5_9_al	0	0
63	y5_10_al	0	0
64	y6_2_al	0	0
65	y6_3_al	0	0
66	y6_7_al	-31	0
67	y6_10_al	-36	0
68	y7_4_al	0	0
69	y7_5_al	0	0

70	y7_6_al	0	0
71	y7_9_al	-31	0
72	y8_4_al	0	0
73	y8_9_al	0	0
74	y8_11_al	-22	0
75	y9_5_al	0	0
76	y9_7_al	0	0
77	y9_8_al	0	0
78	y10_5_al	0	0
79	y10_6_al	0	0
80	y10_11_al	-36	0
81	y11_8_al	0	0
82	y11_10_al	0	0
83	y2_3_fel	0	
84	y2_5_fel	0	
85	y2_6_fel	955	
86	y3_2_fel	0	
87	y3_4_fel	0	
88	y3_6_fel	976	
89	y4_3_fel	0	
90	y4_7_fel	0	
91	y4_8_fel	977	
92	y5_2_fel	0	
93	y5_7_fel	0	
94	y5_9_fel	0	
95	y5_10_fel	0	
96	y6_2_fel	0	
97	y6_3_fel	0	
98	y6_7_fel	968	
99	y6_10_fel	963	
100	y7_4_fel	0	
101	y7_5_fel	0	
102	y7_6_fel	0	
103	y7_9_fel	968	
104	y8_4_fel	0	
105	y8_9_fel	0	
106	y8_11_fel	977	
107	y9_5_fel	0	
108	y9_7_fel	0	
109	y9_8_fel	0	
110	y10_5_fel	0	
111	y10_6_fel	0	
112	y10_11_fel	963	
113	y11_8_fel	0	
114	y11_10_fel	0	

115	el2_3	0	1
116	el2_5	0	1
117	el2_6	1	1
118	el3_4	0	1
119	el3_6	1	1
120	el4_7	0	1
121	el4_8	1	1
122	el5_7	0	1
123	el5_9	0	1
124	el5_10	0	1
125	el6_7	1	1
126	el6_10	1	1
127	el7_9	1	1
128	el8_9	0	1
129	el8_11	1	1
130	el10_11	1	1

No.	Column name	Activity
1	f12_1	107
2	x1_2	45
3	x1_3	24
4	x1_4	38
5	x3_2	0
6	x5_2	0
7	x6_2	0
8	x2_3	0
9	x2_5	0
10	x2_6	45
11	x4_3	0
12	x6_3	0
13	x3_4	0
14	x3_6	24
15	x7_4	0
16	x8_4	0
17	x4_7	0
18	x4_8	23
19	x4_12	15
20	x7_5	0
21	x9_5	0
22	x10_5	0
23	x5_7	0
24	x5_9	0
25	x5_10	0
26	x7_6	0

```
27 x10_6                     0
28 x6_7                     32
29 x6_10                    37
30 x9_7                      0
31 x7_9                     32
32 x9_8                      0
33 x11_8                     0
34 x8_9                      0
35 x8_11                    23
36 x9_12                    32
37 x11_10                    0
38 x10_11                   37
39 x11_12                   60
40 y2_3         *            0
41 y2_5         *            0
42 y2_6         *            1
43 y3_2         *            0
44 y3_4         *            0
45 y3_6         *            1
46 y4_3         *            0
47 y4_7         *            0
48 y4_8         *            1
49 y5_2         *            0
50 y5_7         *            0
51 y5_9         *            0
52 y5_10        *            0
53 y6_2         *            0
54 y6_3         *            0
55 y6_7         *            1
56 y6_10        *            1
57 y7_4         *            0
58 y7_5         *            0
59 y7_6         *            0
60 y7_9         *            1
61 y8_4         *            0
62 y8_9         *            0
63 y8_11        *            1
64 y9_5         *            0
65 y9_7         *            0
66 y9_8         *            0
67 y10_5        *            0
68 y10_6        *            0
69 y10_11       *            1
70 y11_8        *            0
71 y11_10       *            0
```

4.18. példa megoldása.

A vágás lényegében az élek egy halmaza. Ez a vágás a hálózatot kétfelé osztja. A forrás oldaláról a nyelő oldalára kétféle él lehet. A forrás oldaláról a nyelő oldalára mutató él, ahol a szállítás mennyisége megegyezik az él kapacitásával. Ha van olyan él, ami a nyelő oldaláról a forrás oldalára mutat, akkor ezen az élen semmit sem szállítunk. Ha találunk egy egy ilyen vágást, akkor az egyben bizonyíték is arra, hogy a folyam értéke nem lehet nagyobb.

Ha a forrás oldaláról a nyelő oldalára mutató élek kapacitását meg tudnánk növelni, azzal növelnénk a folyam kapacitását is. Tehát ezen élek árnyékára 1. De ez még nem elég. Ha előírnánk, hogy a nyelő oldaláról a forrás oldalára mutató éleken (ahol semmit sem szállítunk) valami kis mennyiséget kellene szállítani, akkor viszont csökkentenénk a folyam mértékét, tehát az árnyékár -1 lenne. Ilyet viszont nem látunk. Ennek az az oka, hogy vannak nemnegativitási korlátaink is, és ezekhez tartozó árnyékárakra vagyunk kíváncsiak. Ezt adott esetben `sens.out` állományból ki is lehetne deríteni, de egyszerűbb, ha előírjuk, hogy valamely csekély mennyiséget minden élen szállítani kell. Ekkor a 4.8. Kód az alábbiak szerint változik:

10.23. kód.

```
max
f7_1

subject to
csp1: +f7_1-x1_2-x1_3-x1_5-x1_6-x1_7       = 0
csp2: +x1_2+x4_2-x2_3-x2_5-x2_7            = 0
csp3: +x1_3+x2_3+x5_3-x3_4-x3_6-x3_7       = 0
csp4: +x3_4-x4_2-x4_5-x4_6-x4_7            = 0
csp5: +x1_5+x2_5+x4_5+x6_5-x5_3-x5_7       = 0
csp6: +x1_6+x3_6+x4_6-x6_5-x6_7            = 0
csp7: +x1_7+x2_7+x3_7+x4_7+x5_7+x6_7-f7_1 = 0

kap1_2: x1_2<= 13
kap1_3: x1_3<= 23
kap1_5: x1_5<= 44
kap1_6: x1_6<= 16
kap1_7: x1_7<= 32
kap2_3: x2_3<= 26
kap2_5: x2_5<= 36
kap2_7: x2_7<= 11
```

```
kap3_4: x3_4<= 37
kap3_6: x3_6<= 11
kap3_7: x3_7<= 37
kap4_2: x4_2<= 38
kap4_5: x4_5<= 34
kap4_6: x4_6<= 23
kap4_7: x4_7<= 23
kap5_3: x5_3<= 23
kap5_7: x5_7<= 11
kap6_5: x6_5<= 11
kap6_7: x6_7<= 32

kapa1_2: x1_2>= 0.001
kapa1_3: x1_3>= 0.001
kapa1_5: x1_5>= 0.001
kapa1_6: x1_6>= 0.001
kapa1_7: x1_7>= 0.001
kapa2_3: x2_3>= 0.001
kapa2_5: x2_5>= 0.001
kapa2_7: x2_7>= 0.001
kapa3_4: x3_4>= 0.001
kapa3_6: x3_6>= 0.001
kapa3_7: x3_7>= 0.001
kapa4_2: x4_2>= 0.001
kapa4_5: x4_5>= 0.001
kapa4_6: x4_6>= 0.001
kapa4_7: x4_7>= 0.001
kapa5_3: x5_3>= 0.001
kapa5_7: x5_7>= 0.001
kapa6_5: x6_5>= 0.001
kapa6_7: x6_7>= 0.001

end
```

Az optimális megoldás:

```
Status:     OPTIMAL
Objective:  obj = 117.997 (MAXimum)

   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             0             1
     2 csp2         NS             0         < eps
     3 csp3         NS             0         < eps
     4 csp4         NS             0         < eps
```

```
     5 csp5         NS             0             1
     6 csp6         NS             0         < eps
     7 csp7         B              0
     8 kap1_2       NU            13             1
     9 kap1_3       NU            23             1
    10 kap1_5       B         33.997
    11 kap1_6       NU            16             1
    12 kap1_7       NU            32             1
    13 kap2_3       B              2
    14 kap2_5       B          0.001
    15 kap2_7       NU            11         < eps
    16 kap3_4       B         10.999
    17 kap3_6       B          0.001
    18 kap3_7       NU            37         < eps
    19 kap4_2       B          0.001
    20 kap4_5       B          0.001
    21 kap4_6       B          0.001
    22 kap4_7       B         10.996
    23 kap5_3       NU            23             1
    24 kap5_7       NU            11             1
    25 kap6_5       B          0.001
    26 kap6_7       B         16.001
    27 kapa1_2      B             13
    28 kapa1_3      B             23
    29 kapa1_5      B         33.997
    30 kapa1_6      B             16
    31 kapa1_7      B             32
    32 kapa2_3      B              2
    33 kapa2_5      NL         0.001            -1
    34 kapa2_7      B             11
    35 kapa3_4      B         10.999
    36 kapa3_6      NL         0.001         < eps
    37 kapa3_7      B             37
    38 kapa4_2      NL         0.001         < eps
    39 kapa4_5      NL         0.001            -1
    40 kapa4_6      NL         0.001         < eps
    41 kapa4_7      B         10.996
    42 kapa5_3      B             23
    43 kapa5_7      B             11
    44 kapa6_5      NL         0.001            -1
    45 kapa6_7      B         16.001

   No. Column name  St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 f7_1         B        117.997
```

```
 2 x1_2          B              13
 3 x1_3          B              23
 4 x1_5          B          33.997
 5 x1_6          B              16
 6 x1_7          B              32
 7 x4_2          B           0.001
 8 x2_3          B               2
 9 x2_5          B           0.001
10 x2_7          B              11
11 x5_3          B              23
12 x3_4          B          10.999
13 x3_6          B           0.001
14 x3_7          B              37
15 x4_5          B           0.001
16 x4_6          B           0.001
17 x4_7          B          10.996
18 x6_5          B           0.001
19 x5_7          B              11
20 x6_7          B          16.001
```

Látható, hogy az alsó korlátok miatt a folyam mérete egy kicsit csökkent, de az is látszik, hogy azért lényegében ugyanannyi, mint eddig. Most már meg tudjuk adni a minimális vágáshoz tartozó éleket: 1-2, 1-3, 1-6, 1-7, 5-3, 5-7, valamint az 2-5, 4-5, és 6-5 élek. Az utolsó 3 'visszafele' mutató él.

De tovább mehetünk, és a vágás egyik illetve másik oldalán levő pontokat is meg tudjuk adni az eredményfül segítségével. Ehhez a csomóponti feltételekhez tartozó árnyékárak fognak segítséget jelenteni. Mivel az árnyékárak egyenlőségekhez tartoznak az interpretálásuk kicsit nehéz jelen formában. De első lépésként töröljük ki a `csp7` korlátot. Mivel a feltételrendszer túlhatározott volt, ezért a feladat megoldásában semmi érdemi változást nem veszünk észre.

Következő lépésként cseréljük ki a `csp1` korlát jobb oldalát 0-ról 1-re. Ez azt jelenti, hogy 1 egységnyi áru eltűnik ebben a pontban. Gondolhatunk erre például adó- vagy vámfizetési kötelezettségként. Nézzük meg mi történik: a folyam mérete megnőtt 1 egységgel. A 7-es csomópontban keletkezik 1 egységnyi áru, ezzel megnöveltük a fiktív élen áthaladó árumennyiséget, tehát a célfüggvény értékét is. Ez a keletkezett 1 egységnyi áru megszűnik az 1-es csomópontban. Így érthetővé válik a 1 egységnyi árnyékár a `csp1` korlát esetén. Nézzük most a `csp6` korlátot.

Ez a vágás túloldalán van. Mi történik, ha itt cseréljük le a korlát jobboldalát 0-ról 1-re. Ekkor nem változik a célfüggvény értéke: a vágáson nem tudunk több árut átküldeni, mint korábban. A vágáson átmenő árumennyiségből 1 egység eltűnik a 6-os csomópontban, ezt az eltűnő mennyiséget tudjuk pótolni a 7-es csomópontban. De a fiktív élen áthaladó árumennyiség változatlan marad, tehát ehhez a korláthoz tartozó árnyékár 0.

Tehát azok a csomópontok lesznek a vágás egyik oldalán, ahol az árnyékár 1, a túloldalon pedig azok, ahol az árnyékár 0. A mi esetünkben az 1-es és 5-ös csomópont van az egyik oldalon, a többi pedig a túloldalon.

4.19. példa megoldása.

Az eredeti hálózathoz hozzá kell adni egy fiktív csúcsot, amelybe a `8-13` pontokból lehet közvetlenül kerülni. Ezeknek az éleknek nincs kapacitása. A hálózat többi éleinek a kapacitásai táblázatba rendezve:

	2	3	4	5	6	7	8	9	10	11	12	13
1	100	35										
2		10	80	10								
3				10	35	5						
4							80	10				
5									10			
6									5	35	10	
7											5	5
8								20				
9									10			
10												
11												
12												10
13												

Ezután a feladat első részét ugyanúgy kell megoldani, mint az eddigi esetekben.

10.24. kód.

```
Max
f14_1
subject to
```

```
csp1:  +f14_1 -x1_2   -x1_3                  = 0
csp2:  +x1_2  -x2_3   -x2_4  -x2_5           = 0
csp3:  +x1_3  +x2_3   -x3_5  -x3_6-x3_7      = 0
csp4:  +x2_4  -x4_8   -x4_9                  = 0
csp5:  +x2_5  +x3_5   -x5_10                 = 0
csp6:  +x3_6  -x6_10  -x6_11 -x6_12          = 0
csp7:  +x3_7  -x7_12  -x7_13                 = 0
csp8:  +x4_8  -x8_9   -x8_14                 = 0
csp9:  +x4_9  +x8_9   -x9_10 -x9_14          = 0
csp10: +x5_10 +x6_10  +x9_10 -x10_14         = 0
csp11: +x6_11 -x11_14                        = 0
csp12: +x6_12 +x7_12  -x12_13 -x12_14        = 0
csp13: +x7_13 +x12_13 -x13_14                = 0
csp14: +x8_14 +x9_14  +x10_14
              +x11_14 +x12_14 +x13_14 -f14_1 = 0

kap1_2:   x1_2   <= 100
kap1_3:   x1_3   <=  35
kap2_3:   x2_3   <=  10
kap2_4:   x2_4   <=  80
kap2_5:   x2_5   <=  10
kap3_5:   x3_5   <=  10
kap3_6:   x3_6   <=  35
kap3_7:   x3_7   <=   5
kap4_8:   x4_8   <=  80
kap4_9:   x4_9   <=  10
kap5_10:  x5_10  <=  10
kap6_10:  x6_10  <=   5
kap6_11:  x6_11  <=  35
kap6_12:  x6_12  <=  10
kap7_12:  x7_12  <=   5
kap7_13:  x7_13  <=   5
kap8_9:   x8_9   <=  20
kap9_10:  x9_10  <=  10
kap12_13: x12_13 <=  10

end
```

A feladat optimális megoldása:

```
   No.   Row name    St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             0             1
```

```
     2 csp2         NS             0             1
     3 csp3         NS             0             1
     4 csp4         NS             0         < eps
     5 csp5         NS             0             1
     6 csp6         NS             0         < eps
     7 csp7         NS             0         < eps
     8 csp8         NS             0         < eps
     9 csp9         NS             0         < eps
    10 csp10        NS             0         < eps
    11 csp11        NS             0         < eps
    12 csp12        NS             0         < eps
    13 csp13        NS             0         < eps
    14 csp14        B              0
    15 kap1_2       B             95
    16 kap1_3       NU            35         < eps
    17 kap2_3       B              5
    18 kap2_4       NU            80             1
    19 kap2_5       NU            10         < eps
    20 kap3_5       B              0
    21 kap3_6       NU            35             1
    22 kap3_7       NU             5             1
    23 kap4_8       B             70
    24 kap4_9       NU            10         < eps
    25 kap5_10      NU            10             1
    26 kap6_10      NU             5         < eps
    27 kap6_11      B             20
    28 kap6_12      NU            10         < eps
    29 kap7_12      B              0
    30 kap7_13      B              5
    31 kap8_9       B              0
    32 kap9_10      B              0
    33 kap12_13     B              0

   No. Column name  St   Activity
------ ------------ -- -------------
     1 f14_1        B            130
     2 x1_2         B             95
     3 x1_3         B             35
     4 x2_3         B              5
     5 x2_4         B             80
     6 x2_5         B             10
     7 x3_5         B              0
     8 x3_6         B             35
     9 x3_7         B              5
    10 x4_8         B             70
```

```
11 x4_9         B            10
12 x5_10        B            10
13 x6_10        B             5
14 x6_11        B            20
15 x6_12        B            10
16 x7_12        NL            0
17 x7_13        B             5
18 x8_9         NL            0
19 x8_14        B            70
20 x9_10        NL            0
21 x9_14        B            10
22 x10_14       B            15
23 x11_14       B            20
24 x12_13       NL            0
25 x12_14       B            10
26 x13_14       B             5
```

Tehát összesen 130 lakást tud bérelni a vállalat, a lakóparkok szerinti bérlendő lakások száma a következő táblázatban látható:

	bérelt lakások
13	5
12	10
11	20
10	15
9	10
8	70

Ahhoz, hogy a bérleti díjakat minimalizáljuk, a célfüggvényt kell megváltoztatni, a következőképpen:

`+15x13_14 +18x12_14 +20x11_14 +12x10_14 +18x9_14 +25x8_14`

Ezen kívül fel kell tenni, hogy összesen 100 lakást bérel a vállalat, tehát a korlátokhoz hozzá kell adni, hogy :

`f14_1 = 100`

10.25. kód.

```
min
+15x13_14 +18x12_14 +20x11_14
+12x10_14 +18x9_14 +25x8_14

subject to
```

```
csp1:  +f14_1 -x1_2   -x1_3                  = 0
csp2:  +x1_2  -x2_3   -x2_4  -x2_5           = 0
csp3:  +x1_3  +x2_3   -x3_5  -x3_6-x3_7      = 0
csp4:  +x2_4  -x4_8   -x4_9                  = 0
csp5:  +x2_5  +x3_5   -x5_10                 = 0
csp6:  +x3_6  -x6_10  -x6_11 -x6_12          = 0
csp7:  +x3_7  -x7_12  -x7_13                 = 0
csp8:  +x4_8  -x8_9   -x8_14                 = 0
csp9:  +x4_9  +x8_9   -x9_10 -x9_14          = 0
csp10: +x5_10 +x6_10  +x9_10 -x10_14         = 0
csp11: +x6_11 -x11_14                        = 0
csp12: +x6_12 +x7_12  -x12_13 -x12_14        = 0
csp13: +x7_13 +x12_13 -x13_14                = 0
csp14: +x8_14 +x9_14  +x10_14
              +x11_14 +x12_14 +x13_14 -f14_1 = 0

kap1_2:   x1_2   <= 100
kap1_3:   x1_3   <=  35
kap2_3:   x2_3   <=  10
kap2_4:   x2_4   <=  80
kap2_5:   x2_5   <=  10
kap3_5:   x3_5   <=  10
kap3_6:   x3_6   <=  35
kap3_7:   x3_7   <=   5
kap4_8:   x4_8   <=  80
kap4_9:   x4_9   <=  10
kap5_10:  x5_10  <=  10
kap6_10:  x6_10  <=   5
kap6_11:  x6_11  <=  35
kap6_12:  x6_12  <=  10
kap7_12:  x7_12  <=   5
kap7_13:  x7_13  <=   5
kap8_9:   x8_9   <=  20
kap9_10:  x9_10  <=  10
kap12_13: x12_13 <=  10

lakas: f14_1 = 100

end
```

A feladat optimális megoldása:

```
   No.   Row name   St   Activity      Marginal
```

```
------ ------------ -- ------------- -------------
     1 csp1         NS             0           -10
     2 csp2         NS             0           -10
     3 csp3         NS             0           -10
     4 csp4         NS             0           -10
     5 csp5         NS             0           -10
     6 csp6         NS             0            -5
     7 csp7         NS             0         < eps
     8 csp8         NS             0           -10
     9 csp9         NS             0            -3
    10 csp10        NS             0             3
    11 csp11        NS             0            -5
    12 csp12        NS             0         < eps
    13 csp13        B              0
    14 csp14        NS             0            15
    15 kap1_2       B             65
    16 kap1_3       NU            35         < eps
    17 kap2_3       B              5
    18 kap2_4       B             50
    19 kap2_5       NU            10         < eps
    20 kap3_5       B              0
    21 kap3_6       NU            35            -5
    22 kap3_7       NU             5           -10
    23 kap4_8       B             40
    24 kap4_9       NU            10            -7
    25 kap5_10      NU            10           -13
    26 kap6_10      NU             5            -8
    27 kap6_11      B             20
    28 kap6_12      NU            10            -5
    29 kap7_12      B              0
    30 kap7_13      B              5
    31 kap8_9       NU            20            -7
    32 kap9_10      NU            10            -6
    33 kap12_13     B             10
    34 lakas        NS           100            25

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 x13_14       B             15
     2 x12_14       NL             0             3
     3 x11_14       B             20
     4 x10_14       B             25
     5 x9_14        B             20
     6 x8_14        B             20
     7 f14_1        B            100
```

```
  8 x1_2         B             65
  9 x1_3         B             35
 10 x2_3         B              5
 11 x2_4         B             50
 12 x2_5         B             10
 13 x3_5         B              0
 14 x3_6         B             35
 15 x3_7         B              5
 16 x4_8         B             40
 17 x4_9         B             10
 18 x5_10        B             10
 19 x6_10        B              5
 20 x6_11        B             20
 21 x6_12        B             10
 22 x7_12        NL             0         < eps
 23 x7_13        B              5
 24 x8_9         B             20
 25 x9_10        B             10
 26 x12_13       B             10
```

A következő táblázatban látható, hogy a lakóparkokban így mennyit bérel ekkor a vállalat:

	bérelt lakások
13	15
12	0
11	20
10	25
9	20
8	20

A bérleti díj pedig 1785.

4.20. példa megoldása.

Most jelölje `xAi_Aj` az i. csúcs és j. csúcs közötti éleken átfolyatott `A` olaj mennyiségét, és `xBi_Bj` ugyanezen az élen az átfolyatott `B` mennyiséget. Ebben az esetben a `csp:` korlátokat mind a két típusú olajra meg kell adni, külön-külön, tehát kellenek külön `cspA:` és `cspB:` korlátok is. Azokon az éleken, ahol csak az egyik típusú olaj szállítható, ott a kapacitás korlátok nem változnak. Ahol azonban mind a két típus áramoltatható, ott a korlát (az `i,j` élen) a következőképpen alakul:

`kapi_j: xAi_Aj + xBi_Bj <= K(i,j)`

`K(i,j)` az adott él kapacitását jelöli. Ezen kívül fel kell tennünk, hogyha `A` áramlik ezen az élen, akkor a `B` típusban nem áramolhat ezen a csővezetéken keresztül. Azaz egy olyan korlátot kell felírnunk, hogyha `xAi_Aj`>0, akkor `xBi_Bj`=0. Ehhez bevezetünk egy `Yi_j` bináris változót, ami esetén ha `Yi_j`=1, akkor az élen `A` olajat szállítunk, ha `Yi_j`=0, akkor viszont `B`-t. Ehhez a következő korlátokat kell hozzáadni a modellhez:

`fkAi_j: xAi_Aj - M Yi_j <= 0`

`fkBi_j: xBi_Bj + M Yi_j <= M`

`M` egy kellően nagy konstans, ami ebben a példában megfelelő, ha `M` a `K(i,j)` értéket veszi fel.

Ahhoz, hogy feltegyük, hogy legalább 20 egység kerüljön A-ból, és B-ből a 10. csúcsba, a `fA10_1>=20`, és `fB10_1>=20` korlátokat kell a kódhoz hozzáadni. Ahhoz, hogy mindenképpen több `A` legyen elszállítva, mint `B`, ahhoz pedig a `fA10_1>=fB10_1` korlát szükséges.

10.26. kód.

```
Max

fA10_1 +fB10_1

subject to

csp1A:  +fA10_1  -x1_A2   -x1_A3                         = 0
csp2A:  +x1_A2   -xA2_A3  -xA2_A5                        = 0
csp3A:  +x1_A3   +xA2_A3  -xA3_A5  -xA3_A6 -xA3_A9       = 0
csp5A:  +xA2_A5  +xA3_A5  -xA5_A6  -xA5_A8 -xA5_A10      = 0
csp6A:  +xA3_A6  +xA5_A6  -xA6_A8  -xA6_A9               = 0
csp8A:  +xA5_A8  +xA6_A8  -xA8_A10                       = 0
csp9A:  +xA3_A9  +xA6_A9  -xA9_A10                       = 0
csp10A: +xA5_A10 +xA8_A10 +xA9_A10 -fA10_1               = 0

csp1B:  +fB10_1  -x1_B3   -x1_B4                         = 0
csp3B:  +x1_B3   +xB4_B3  -xB3_B6  -xB3_B9               = 0
csp4B:  +x1_B4   -xB4_B3  -xB4_B6  -xB4_B7               = 0
csp6B:  +xB3_B6  +xB4_B6  +xB7_B6  -xB6_B8   -xB6_B9 = 0
csp7B:  +xB4_B7  -xB7_B6  -xB7_B9  -xB7_B10              = 0
```

```
csp8B:  +xB6_B8  -xB8_B10                           = 0
csp9B:  +xB3_B9  +xB6_B9  +xB7_B9  -xB9_B10         = 0
csp10B: +xB7_B10 +xB8_B10 +xB9_B10 -fB10_1          = 0

kap1_A2:   x1_A2   <= 30
kapA2_A3:  xA2_A3  <= 20
kapA2_A5:  xA2_A5  <= 52
kapA3_A5:  xA3_A5  <= 21
kapA5_A6:  xA5_A6  <= 10
kapA5_A8:  xA5_A8  <= 36
kapA5_A10: xA5_A10 <= 10

kap1_B4:   x1_B4   <= 40
kapB4_B3:  xB4_B3  <= 10
kapB4_B6:  xB4_B6  <= 13
kapB4_B7:  xB4_B7  <= 25
kapB7_B6:  xB7_B6  <= 10
kapB7_B9:  xB7_B9  <= 15
kapB7_B10: xB7_B10 <=  5

kap1_3:  x1_A3   +x1_B3   <= 50
kap3_6:  xA3_A6  +xB3_B6  <= 60
kap3_9:  xA3_A9  +xB3_B9  <= 10
kap6_8:  xA6_A8  +xB6_B8  <= 30
kap6_9:  xA6_A9  +xB6_B9  <= 30
kap8_10: xA8_A10 +xB8_B10 <= 40
kap9_10: xA9_A10 +xB9_B10 <= 40

fkA1_3:  x1_A3   -50Y1_3  <= 0
fkA3_6:  xA3_A6  -60Y3_6  <= 0
fkA3_9:  xA3_A9  -10Y3_9  <= 0
fkA6_8:  xA6_A8  -30Y6_8  <= 0
fkA6_9:  xA6_A9  -30Y6_9  <= 0
fkA8_10: xA8_A10 -40Y8_10 <= 0
fkA9_10: xA9_A10 -40Y9_10 <= 0

fkB1_3:  x1_B3   +50Y1_3  <= 50
fkB3_6:  xB3_B6  +60Y3_6  <= 60
fkB3_9:  xB3_B9  +10Y3_9  <= 10
fkB6_8:  xB6_B8  +30Y6_8  <= 30
fkB6_9:  xB6_B9  +30Y6_9  <= 30
fkB8_10: xB8_B10 +40Y8_10 <= 40
fkB9_10: xB9_B10 +40Y9_10 <= 40

AN20: fA10_1 >= 20
```

```
BN20: fB10_1 >= 20
ANB:  fA10_1 - fB10_1 >= 0
Binary

Y1_3
Y3_6
Y3_9
Y6_8
Y6_9
Y8_10
Y9_10

end
```

A feladat optimális megoldása:

```
   No.   Row name         Activity
------ ------------    -------------
     1 csp1A                      0
     2 csp2A                      0
     3 csp3A                      0
     4 csp5A                      0
     5 csp6A                      0
     6 csp8A                      0
     7 csp9A                      0
     8 csp10A                     0
     9 csp1B                      0
    10 csp3B                      0
    11 csp4B                      0
    12 csp6B                      0
    13 csp7B                      0
    14 csp8B                      0
    15 csp9B                      0
    16 csp10B                     0
    17 kap1_A2                    0
    18 kapA2_A3                   0
    19 kapA2_A5                   0
    20 kapA3_A5                  21
    21 kapA5_A6                   0
    22 kapA5_A8                  11
    23 kapA5_A10                 10
    24 kap1_B4                   40
    25 kapB4_B3                  10
```

```
    26 kapB4_B6                    5
    27 kapB4_B7                   25
    28 kapB7_B6                   10
    29 kapB7_B9                   10
    30 kapB7_B10                   5
    31 kap1_3                     50
    32 kap3_6                     29
    33 kap3_9                     10
    34 kap6_8                     29
    35 kap6_9                     15
    36 kap8_10                    40
    37 kap9_10                    35
    38 fkA1_3                      0
    39 fkA3_6                    -31
    40 fkA3_9                      0
    41 fkA6_8                     -1
    42 fkA6_9                      0
    43 fkA8_10                     0
    44 fkA9_10                     0
    45 fkB1_3                     50
    46 fkB3_6                     60
    47 fkB3_9                     10
    48 fkB6_8                     30
    49 fkB6_9                     15
    50 fkB8_10                    40
    51 fkB9_10                    35
    52 AN20                       50
    53 BN20                       40
    54 ANB                        10

   No. Column name       Activity
------ ------------    -------------
     1 fA10_1                     50
     2 fB10_1                     40
     3 x1_A2                       0
     4 x1_A3                      50
     5 xA2_A3                      0
     6 xA2_A5                      0
     7 xA3_A5                     21
     8 xA3_A6                     29
     9 xA3_A9                      0
    10 xA5_A6                      0
    11 xA5_A8                     11
    12 xA5_A10                    10
    13 xA6_A8                     29
```

```
14 xA6_A9                        0
15 xA8_A10                      40
16 xA9_A10                       0
17 x1_B3                         0
18 x1_B4                        40
19 xB4_B3                       10
20 xB3_B6                        0
21 xB3_B9                       10
22 xB4_B6                        5
23 xB4_B7                       25
24 xB7_B6                       10
25 xB6_B8                        0
26 xB6_B9                       15
27 xB7_B9                       10
28 xB7_B10                       5
29 xB8_B10                       0
30 xB9_B10                      35
31 Y1_3         *                1
32 Y3_6         *                1
33 Y3_9         *                0
34 Y6_8         *                1
35 Y6_9         *                0
36 Y8_10        *                1
37 Y9_10        *                0
```

Tehát összesen 90 egységet tudunk az 1. csúcsból a 10-es csúcsba szállítani, amelyből 50 egység `A` típusú olaj, 40 pedig `B` típusú.

4.21. példa megoldása.

4 egymást követő "oszlopokba" rendezzük a hálózat csúcsait. Az első oszlopban 5 csúcs szerepel, amelyek az orvosokat jelölik, a következő 3 oszlopban pedig a napszakokhoz tartozó 5-5 darab csúcs szerepel. Az egyes oszlopok a reggeli- (`r1, r2, r3, r4, r5`), délutáni- (`d1, d2, d3, d4, d5`), és esti ügyeleteket (`e1, e2, e3, e4, e5`) jelentik, a bennük lévő a csúcsok pedig a napokat.

Az első oszlopba tartozó csúcsokba egy-egy kapacitású élek tartanak, egy "fiktív" kifolyóból (`F`). Ezekből az élekből a következő csúcsok oszlopába haladó élek jelentik, hogy az adott orvos melyik reggel lehet ügyeletes. Tehát a következő 1 kapacitású élek mennek ez a két oszlop között:

	r1	r2	r3	r4	r5
A		1		1	1
B	1	1	1		1
C	1	1	1		1
D	1	1			
E				1	1

A következő csúcsok oszlopa a délutáni ügyeletet jelenti, tehát az itteni 1 kapacitású élek a két napaszak közötti lehetséges reggel-délutáni ügyeletpárokat adja:

	d1	d2	d3	d4	d5
r1		1	1		
r2	1		1	1	
r3	1	1		1	1
r4		1	1		1
r5			1	1	

Az utolsó ilyen csúcsokkal kapcsolatos élek pedig a délután-este közötti kapcsolatot adják meg, természetesen itt is 1 kapacitással.

	e1	e2	e3	e4	e5
d1			1	1	1
d2				1	1
d3	1				1
d4	1	1			
d5	1	1	1		

Minden esti ügyelethez tartozik egy 1 kapacitású él, amelyik a végső csúcsba, az `N`-be vezet. A feladatot annyiban módosítani kell a hagyományos Maximális folyam feladathoz képest, hogy fel kell tenni, hogy minden csomópontba bemenő összes él kapacitásának összege 1 (elegendő, ha csak a reggeli, délutáni, és az esti időpontokhoz tartozó csúcsokba bemenő élekhez tesszük fel ezeket a korlátokat). Ezáltal érhető el, hogy minden ügyeleten kizárólag egy orvos legyen jelen. Ezek a korlátok a kódban `Okap:` néven vannak jelölve.

10.27. kód.

```
Max
fN_F

subject to

csp1:  +fN_F   -xF_A   -xF_B   -xF_C   -xF_D   -xF_E   = 0
csp2:  +xF_A   -xA_r2  -xA_r4  -xA_r5                  = 0
csp3:  +xF_B   -xB_r1  -xB_r2  -xB_r3  -xB_r5          = 0
csp4:  +xF_C   -xC_r1  -xC_r2  -xC_r3  -xC_r5          = 0
csp5:  +xF_D   -xD_r1  -xD_r2                          = 0
csp6:  +xF_E   -xE_r4  -xE_r5                          = 0
csp7:  +xB_r1  +xC_r1  +xD_r1  -xr1_d2 -xr1_d3         = 0
csp8:  +xA_r2  +xB_r2  +xC_r2
                       +xD_r2  -xr2_d1 -xr2_d3 -xr2_d4 = 0
csp9:  +xB_r3  +xC_r3  -xr3_d1 -xr3_d2 -xr3_d4 -xr3_d5 = 0
csp10: +xA_r4  +xE_r4  -xr4_d2 -xr4_d3 -xr4_d5         = 0
csp11: +xA_r5  +xB_r5  +xC_r5  +xE_r5  -xr5_d3 -xr5_d4 = 0
csp12: +xr2_d1 +xr3_d1 -xd1_e3 -xd1_e4 -xd1_e5         = 0
csp13: +xr1_d2 +xr3_d2 +xr4_d2 -xd2_e4 -xd2_e5         = 0
csp14: +xr1_d3 +xr2_d3 +xr4_d3 +xr5_d3 -xd3_e1 -xd3_e5 = 0
csp15: +xr2_d4 +xr3_d4 +xr5_d4 -xd4_e1 -xd4_e2         = 0
csp16: +xr3_d5 +xr4_d5 -xd5_e1 -xd5_e2 -xd5_e3         = 0
csp17: +xd3_e1 +xd4_e1 +xd5_e1 -xe1_N                  = 0
csp18: +xd4_e2 +xd5_e2 -xe2_N                          = 0
csp19: +xd1_e3 +xd5_e3 -xe3_N                          = 0
csp20: +xd1_e4 +xd2_e4 -xe4_N                          = 0
csp21: +xd1_e5 +xd2_e5 +xd3_e5 -xe5_N                  = 0
csp22: +xe1_N  +xe2_N  +xe3_N  +xe4_N  +xe5_N -fN_F    = 0

Okap_r1: +xB_r1  +xC_r1  +xD_r1          = 1
Okap_r2: +xA_r2  +xB_r2  +xC_r2  +xD_r2  = 1
Okap_r3: +xB_r3  +xC_r3                  = 1
Okap_r4: +xA_r4  +xE_r4                  = 1
Okap_r5: +xA_r5  +xB_r5  +xC_r5  +xE_r5  = 1
Okap_d1: +xr2_d1 +xr3_d1                 = 1
Okap_d2: +xr1_d2 +xr3_d2 +xr4_d2         = 1
Okap_d3: +xr1_d3 +xr2_d3 +xr4_d3 +xr5_d3 = 1
Okap_d4: +xr2_d4 +xr3_d4 +xr5_d4         = 1
Okap_d5: +xr3_d5 +xr4_d5                 = 1
Okap_e1: +xd3_e1 +xd4_e1 +xd5_e1         = 1
Okap_e2: +xd4_e2 +xd5_e2                 = 1
Okap_e3: +xd1_e3 +xd5_e3                 = 1
Okap_e4: +xd1_e4 +xd2_e4                 = 1
Okap_e5: +xd1_e5 +xd2_e5 +xd3_e5         = 1
```

```
kapF_A:    xF_A   <= 1
kapF_B:    xF_B   <= 1
kapF_C:    xF_C   <= 1
kapF_D:    xF_D   <= 1
kapF_E:    xF_E   <= 1
kapA_r2:   xA_r2  <= 1
kapA_r4:   xA_r4  <= 1
kapA_r5:   xA_r5  <= 1
kapB_r1:   xB_r1  <= 1
kapB_r2:   xB_r2  <= 1
kapB_r3:   xB_r3  <= 1
kapB_r5:   xB_r5  <= 1
kapC_r1:   xC_r1  <= 1
kapC_r2:   xC_r2  <= 1
kapC_r3:   xC_r3  <= 1
kapC_r5:   xC_r5  <= 1
kapD_r1:   xD_r1  <= 1
kapD_r2:   xD_r2  <= 1
kapE_r4:   xE_r4  <= 1
kapE_r5:   xE_r5  <= 1
kapr1_d2:  xr1_d2 <= 1
kapr1_d3:  xr1_d3 <= 1
kapr2_d1:  xr2_d1 <= 1
kapr2_d3:  xr2_d3 <= 1
kapr2_d4:  xr2_d4 <= 1
kapr3_d1:  xr3_d1 <= 1
kapr3_d2:  xr3_d2 <= 1
kapr3_d4:  xr3_d4 <= 1
kapr3_d5:  xr3_d5 <= 1
kapr4_d2:  xr4_d2 <= 1
kapr4_d3:  xr4_d3 <= 1
kapr4_d5:  xr4_d5 <= 1
kapr5_d3:  xr5_d3 <= 1
kapr5_d4:  xr5_d4 <= 1
kapd1_e3:  xd1_e3 <= 1
kapd1_e4:  xd1_e4 <= 1
kapd1_e5:  xd1_e5 <= 1
kapd2_e4:  xd2_e4 <= 1
kapd2_e5:  xd2_e5 <= 1
kapd3_e1:  xd3_e1 <= 1
kapd3_e5:  xd3_e5 <= 1
kapd4_e1:  xd4_e1 <= 1
kapd4_e2:  xd4_e2 <= 1
kapd5_e1:  xd5_e1 <= 1
kapd5_e2:  xd5_e2 <= 1
```

```
kapd5_e3: xd5_e3 <= 1
kape1_N:  xe1_N  <= 1
kape2_N:  xe2_N  <= 1
kape3_N:  xe3_N  <= 1
kape4_N:  xe4_N  <= 1
kape5_N:  xe5_N  <= 1

end
```

Az optimális megoldás:

```
   No.    Row name    St   Activity    Marginal
------ ------------ -- ------------- -------------
     1 csp1         NS             0             1
     2 csp2         NS             0             1
     3 csp3         NS             0             1
     4 csp4         NS             0             1
     5 csp5         NS             0             1
     6 csp6         NS             0             1
     7 csp7         NS             0         < eps
     8 csp8         NS             0         < eps
     9 csp9         NS             0         < eps
    10 csp10        NS             0         < eps
    11 csp11        NS             0         < eps
    12 csp12        NS             0         < eps
    13 csp13        NS             0         < eps
    14 csp14        NS             0         < eps
    15 csp15        NS             0         < eps
    16 csp16        NS             0         < eps
    17 csp17        NS             0         < eps
    18 csp18        B              0
    19 csp19        NS             0         < eps
    20 csp20        NS             0         < eps
    21 csp21        NS             0         < eps
    22 csp22        NS             0         < eps
    23 Okap_r1      NS             1             1
    24 Okap_r2      NS             1             1
    25 Okap_r3      NS             1             1
    26 Okap_r4      NS             1             1
    27 Okap_r5      NS             1             1
    28 Okap_d1      NS             1         < eps
    29 Okap_d2      NS             1         < eps
    30 Okap_d3      NS             1         < eps
    31 Okap_d4      NS             1         < eps
    32 Okap_d5      B              1
```

33	Okap_e1	NS	1	< eps
34	Okap_e2	B	1	
35	Okap_e3	NS	1	< eps
36	Okap_e4	NS	1	< eps
37	Okap_e5	NS	1	< eps
38	kapF_A	NU	1	< eps
39	kapF_B	B	1	
40	kapF_C	B	1	
41	kapF_D	B	1	
42	kapF_E	B	1	
43	kapA_r2	NU	1	< eps
44	kapA_r4	B	0	
45	kapA_r5	B	0	
46	kapB_r1	B	0	
47	kapB_r2	B	0	
48	kapB_r3	B	0	
49	kapB_r5	NU	1	< eps
50	kapC_r1	B	0	
51	kapC_r2	B	0	
52	kapC_r3	B	1	
53	kapC_r5	B	0	
54	kapD_r1	B	1	
55	kapD_r2	B	0	
56	kapE_r4	B	1	
57	kapE_r5	B	0	
58	kapr1_d2	NU	1	< eps
59	kapr1_d3	B	0	
60	kapr2_d1	B	0	
61	kapr2_d3	B	0	
62	kapr2_d4	B	1	
63	kapr3_d1	B	1	
64	kapr3_d2	B	0	
65	kapr3_d4	B	0	
66	kapr3_d5	B	0	
67	kapr4_d2	B	0	
68	kapr4_d3	B	0	
69	kapr4_d5	B	1	
70	kapr5_d3	B	1	
71	kapr5_d4	B	0	
72	kapd1_e3	B	0	
73	kapd1_e4	B	0	
74	kapd1_e5	B	1	
75	kapd2_e4	B	1	
76	kapd2_e5	B	0	
77	kapd3_e1	B	1	

```
    78 kapd3_e5     B              0
    79 kapd4_e1     B              0
    80 kapd4_e2     B              1
    81 kapd5_e1     B              0
    82 kapd5_e2     B              0
    83 kapd5_e3     B              1
    84 kape1_N      B              1
    85 kape2_N      B              1
    86 kape3_N      B              1
    87 kape4_N      B              1
    88 kape5_N      B              1

   No. Column name  St   Activity
------ ------------ -- -------------
     1 fN_F         B              5
     2 xF_A         B              1
     3 xF_B         B              1
     4 xF_C         B              1
     5 xF_D         B              1
     6 xF_E         B              1
     7 xA_r2        B              1
     8 xA_r4        B              0
     9 xA_r5        NL             0
    10 xB_r1        NL             0
    11 xB_r2        NL             0
    12 xB_r3        NL             0
    13 xB_r5        B              1
    14 xC_r1        NL             0
    15 xC_r2        NL             0
    16 xC_r3        B              1
    17 xC_r5        NL             0
    18 xD_r1        B              1
    19 xD_r2        B              0
    20 xE_r4        B              1
    21 xE_r5        B              0
    22 xr1_d2       B              1
    23 xr1_d3       B              0
    24 xr2_d1       NL             0
    25 xr2_d3       NL             0
    26 xr2_d4       B              1
    27 xr3_d1       B              1
    28 xr3_d2       NL             0
    29 xr3_d4       NL             0
    30 xr3_d5       B              0
    31 xr4_d2       B              0
```

```
32 xr4_d3        B             0
33 xr4_d5        B             1
34 xr5_d3        B             1
35 xr5_d4        B             0
36 xd1_e3        NL            0
37 xd1_e4        NL            0
38 xd1_e5        B             1
39 xd2_e4        B             1
40 xd2_e5        B             0
41 xd3_e1        B             1
42 xd3_e5        B             0
43 xd4_e1        NL            0
44 xd4_e2        B             1
45 xd5_e1        B             0
46 xd5_e2        B             0
47 xd5_e3        B             1
48 xe1_N         B             1
49 xe2_N         B             1
50 xe3_N         B             1
51 xe4_N         B             1
52 xe5_N         B             1
```

A végeredmény, azaz egy lehetséges ügyeleti beosztás a következő táblázatban látható:

	reggel	délután	este
A	2. nap	4. nap	2. nap
B	5. nap	3. nap	1. nap
C	3. nap	1. nap	5. nap
D	1. nap	2. nap	4. nap
E	4. nap	5. nap	3. nap

10.4. Legrövidebb út feladatok

5.15. példa megoldása.

Először a csomópontos formalizációt adjuk meg. A megoldás most a 5.6. kódhoz hasonlít leginkább, mivel a legtöbb csúcs között két irányba is lehet haladni. Annyiban tér el viszont a mostani feladat, hogy nem azonosak az oda-vissza tartó utak hosszai. Ez azt jelenti, hogy valójában most is irányított gráfon kell meghatározni a legrövidebb utat, viszont sok csúcs között kettő irányított él is megy, ellenkező irányba.

10.28. kód.

```
Min
tav:
  +26x1_3 +19x1_4 +29x1_5 +51x1_7 +75x1_8 +350x1_10
  +72x2_1 +39x2_4 +72x2_5 +52x2_6 +80x2_7 +45x2_8  +79x2_9
  +30x3_1 +15x3_2 +46x3_4 +50x3_5 +80x3_6 +17x3_7  +79x3_8
  +48x3_9
  +26x4_1 +36x4_5 +56x4_6 +65x4_7 +50x4_8 +68x4_9
  +66x5_1 +40x5_2 +16x5_3 +67x5_4 +57x5_6 +79x5_8  +76x5_9
  +13x6_1 +31x6_3 +75x6_8 +35x6_9 +65x6_10
  +51x7_2 +11x7_3 +43x7_5 +51x7_6 +53x7_8 +12x7_9  +45x7_10
  +45x8_1 +10x8_2 +58x8_3 +10x8_4 +67x8_5 +75x8_6  +59x8_7
  +56x8_9 +22x8_10
  +35x9_2 +17x9_3 +36x9_4 +17x9_6 +62x9_7 +10x9_8  +55x9_10
  +13x10_1+47x10_2+63x10_3+66x10_4+34x10_5+76x10_7 +58x10_8
  +63x10_9

Subject to

csp1:+x1_3+x1_4+x1_5+x1_7+x1_8+x1_10
     -x2_1-x3_1-x4_1-x5_1-x6_1-x8_1-x10_1             =  1
csp2:+x2_1+x2_4+x2_5+x2_6+x2_7+x2_8+x2_9
     -x3_2-x5_2-x7_2-x8_2-x9_2-x10_2                  =  0
csp3:+x3_1+x3_2+x3_4+x3_5+x3_6+x3_7+x3_8+x3_9
     -x1_3-x5_3-x6_3-x7_3-x8_3-x9_3-x10_3             =  0
csp4:+x4_1+x4_5+x4_6+x4_7+x4_8+x4_9
     -x1_4-x2_4-x3_4-x5_4-x8_4-x9_4-x10_4             =  0
csp5:+x5_1+x5_2+x5_3+x5_4+x5_6+x5_8+x5_9
     -x1_5-x2_5-x3_5-x4_5-x7_5-x8_5-x10_5             =  0
csp6:+x6_1+x6_3+x6_8+x6_9+x6_10
     -x2_6-x3_6-x4_6-x5_6-x7_6-x8_6-x9_6              =  0
csp7:+x7_2+x7_3+x7_5+x7_6+x7_8+x7_9+x7_10
     -x1_7-x2_7-x3_7-x4_7-x8_7-x9_7-x10_7             =  0
csp8:+x8_1+x8_2+x8_3+x8_4+x8_5+x8_6+x8_7+x8_9+x8_10
     -x1_8-x2_8-x3_8-x4_8-x5_8-x6_8-x7_8-x9_8-x10_8   =  0
csp9:+x9_2+x9_3+x9_4+x9_6+x9_7+x9_8+x9_10
     -x2_9-x3_9-x4_9-x5_9-x6_9-x7_9-x8_9-x10_9        =  0
csp10:+x10_1+x10_2+x10_3+x10_4+x10_5+x10_7+x10_8+x10_9
      -x1_10-x6_10-x7_10-x8_10-x9_10                  = -1

End
```

A megoldás:

```
   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 x1_3         B              1
    19 x3_7         B              1
    45 x7_9         B              1
    55 x8_10        B              1
    61 x9_8         B              1
```

A $1 \to 3 \to 7 \to 9 \to 8 \to 10$ út lesz a legrövidebb út, amelynek hossza 87.

A gyűrűs felírásban a következőképpen néz ki a feladat

10.29. kód. `max`

```
  y10 -y1

Subject to

el2_1:  y1  -y2  <=  72
el3_1:  y1  -y3  <=  30
el4_1:  y1  -y4  <=  26
el5_1:  y1  -y5  <=  66
el6_1:  y1  -y6  <=  13
el8_1:  y1  -y8  <=  45
el10_1: y1  -y10 <=  13
el3_2:  y2  -y3  <=  15
el5_2:  y2  -y5  <=  40
el7_2:  y2  -y7  <=  51
el8_2:  y2  -y8  <=  10
el9_2:  y2  -y9  <=  35
el10_2: y2  -y10 <=  47
el1_3:  y3  -y1  <=  26
el5_3:  y3  -y5  <=  16
el6_3:  y3  -y6  <=  31
el7_3:  y3  -y7  <=  11
el8_3:  y3  -y8  <=  58
el9_3:  y3  -y9  <=  17
el10_3: y3  -y10 <=  63
el1_4:  y4  -y1  <=  19
el2_4:  y4  -y2  <=  39
el3_4:  y4  -y3  <=  46
el5_4:  y4  -y5  <=  67
el8_4:  y4  -y8  <=  10
el9_4:  y4  -y9  <=  36
el10_4: y4  -y10 <=  66
el1_5:  y5  -y1  <=  29
el2_5:  y5  -y2  <=  72
```

```
el3_5:  y5  -y3  <=  50
el4_5:  y5  -y4  <=  36
el7_5:  y5  -y7  <=  43
el8_5:  y5  -y8  <=  67
el10_5: y5  -y10 <=  34
el2_6:  y6  -y2  <=  52
el3_6:  y6  -y3  <=  80
el4_6:  y6  -y4  <=  56
el5_6:  y6  -y5  <=  57
el7_6:  y6  -y7  <=  51
el8_6:  y6  -y8  <=  75
el9_6:  y6  -y9  <=  17
el1_7:  y7  -y1  <=  51
el2_7:  y7  -y2  <=  80
el3_7:  y7  -y3  <=  17
el4_7:  y7  -y4  <=  65
el8_7:  y7  -y8  <=  59
el9_7:  y7  -y9  <=  62
el10_7: y7  -y10 <=  76
el1_8:  y8  -y1  <=  75
el2_8:  y8  -y2  <=  45
el3_8:  y8  -y3  <=  79
el4_8:  y8  -y4  <=  50
el5_8:  y8  -y5  <=  79
el6_8:  y8  -y6  <=  75
el7_8:  y8  -y7  <=  53
el9_8:  y8  -y9  <=  10
el10_8: y8  -y10 <=  58
el2_9:  y9  -y2  <=  79
el3_9:  y9  -y3  <=  48
el4_9:  y9  -y4  <=  68
el5_9:  y9  -y5  <=  76
el6_9:  y9  -y6  <=  35
el7_9:  y9  -y7  <=  12
el8_9:  y9  -y8  <=  56
el10_9: y9  -y10 <=  63
el1_10: y10 -y1  <= 350
el6_10: y10 -y6  <=  65
el7_10: y10 -y7  <=  45
el8_10: y10 -y8  <=  22
el9_10: y10 -y9  <=  55

end
```

A gyűrűs felírás megoldása:

No.	Row name	St	Activity	Marginal
1	el2_1	B	-20	
2	el3_1	B	-26	
3	el4_1	B	-15	
4	el5_1	B	-10	
5	el6_1	B	-22	
6	el8_1	B	-65	
7	el10_1	B	-87	
8	el3_2	B	-6	
9	el5_2	B	10	
10	el7_2	B	-23	
11	el8_2	B	-45	
12	el9_2	B	-35	
13	el10_2	B	-67	
14	el1_3	NU	26	1
15	el5_3	NU	16	< eps
16	el6_3	B	4	
17	el7_3	B	-17	
18	el8_3	B	-39	
19	el9_3	B	-29	
20	el10_3	B	-61	
21	el1_4	B	15	
22	el2_4	B	-5	
23	el3_4	B	-11	
24	el5_4	B	5	
25	el8_4	B	-50	
26	el9_4	B	-40	
27	el10_4	B	-72	
28	el1_5	B	10	
29	el2_5	B	-10	
30	el3_5	B	-16	
31	el4_5	B	-5	
32	el7_5	B	-33	
33	el8_5	B	-55	
34	el10_5	B	-77	
35	el2_6	B	2	
36	el3_6	B	-4	
37	el4_6	B	7	
38	el5_6	B	12	
39	el7_6	B	-21	
40	el8_6	B	-43	
41	el9_6	B	-33	
42	el1_7	B	43	
43	el2_7	B	23	

```
    44 el3_7        NU            17             1
    45 el4_7        B             28
    46 el8_7        B            -22
    47 el9_7        B            -12
    48 el10_7       B            -44
    49 el1_8        B             65
    50 el2_8        NU            45         < eps
    51 el3_8        B             39
    52 el4_8        NU            50         < eps
    53 el5_8        B             55
    54 el6_8        B             43
    55 el7_8        B             22
    56 el9_8        NU            10             1
    57 el10_8       B            -22
    58 el2_9        B             35
    59 el3_9        B             29
    60 el4_9        B             40
    61 el5_9        B             45
    62 el6_9        B             33
    63 el7_9        NU            12             1
    64 el8_9        B            -10
    65 el10_9       B            -32
    66 el1_10       B             87
    67 el6_10       NU            65         < eps
    68 el7_10       B             44
    69 el8_10       NU            22             1
    70 el9_10       B             32

   No. Column name  St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 y10          B             87
     2 y1           NL             0         < eps
     3 y2           B             20
     4 y3           B             26
     5 y4           B             15
     6 y5           B             10
     7 y6           B             22
     8 y8           B             65
     9 y7           B             43
    10 y9           B             55
```

A legrövidebb út természetesen itt is ugyanaz, mint a másik felírás esetén.

5.16. példa megoldása.

Mivel az előző feladatban az 1. csúcsból a 10-esbe már felírtuk mind a két formalizált LP-t, ezért egy egyszerű megoldás, ha azok valamelyikét módosítjuk. A másik irányú legrövidebb út keresésére egyszerűbb a gyűrűs formalizációt módosítani, mivel ott csak a célfüggvény arra módosul, hogy `y1 -y10`, a `y10 -y1` helyett. A korlátok ugyanazok maradnak, mint az előző esetben.

10.30. kód.

```
max
y1 -y10

Subject to

el2_1:  y1  -y2  <=  72
el3_1:  y1  -y3  <=  30
el4_1:  y1  -y4  <=  26
el5_1:  y1  -y5  <=  66
el6_1:  y1  -y6  <=  13
el8_1:  y1  -y8  <=  45
el10_1: y1  -y10 <=  13
el3_2:  y2  -y3  <=  15
el5_2:  y2  -y5  <=  40
el7_2:  y2  -y7  <=  51
el8_2:  y2  -y8  <=  10
el9_2:  y2  -y9  <=  35
el10_2: y2  -y10 <=  47
el1_3:  y3  -y1  <=  26
el5_3:  y3  -y5  <=  16
el6_3:  y3  -y6  <=  31
el7_3:  y3  -y7  <=  11
el8_3:  y3  -y8  <=  58
el9_3:  y3  -y9  <=  17
el10_3: y3  -y10 <=  63
el1_4:  y4  -y1  <=  19
el2_4:  y4  -y2  <=  39
el3_4:  y4  -y3  <=  46
el5_4:  y4  -y5  <=  67
el8_4:  y4  -y8  <=  10
el9_4:  y4  -y9  <=  36
el10_4: y4  -y10 <=  66
el1_5:  y5  -y1  <=  29
el2_5:  y5  -y2  <=  72
el3_5:  y5  -y3  <=  50
```

```
el4_5:  y5  -y4  <=  36
el7_5:  y5  -y7  <=  43
el8_5:  y5  -y8  <=  67
el10_5: y5  -y10 <=  34
el2_6:  y6  -y2  <=  52
el3_6:  y6  -y3  <=  80
el4_6:  y6  -y4  <=  56
el5_6:  y6  -y5  <=  57
el7_6:  y6  -y7  <=  51
el8_6:  y6  -y8  <=  75
el9_6:  y6  -y9  <=  17
el1_7:  y7  -y1  <=  51
el2_7:  y7  -y2  <=  80
el3_7:  y7  -y3  <=  17
el4_7:  y7  -y4  <=  65
el8_7:  y7  -y8  <=  59
el9_7:  y7  -y9  <=  62
el10_7: y7  -y10 <=  76
el1_8:  y8  -y1  <=  75
el2_8:  y8  -y2  <=  45
el3_8:  y8  -y3  <=  79
el4_8:  y8  -y4  <=  50
el5_8:  y8  -y5  <=  79
el6_8:  y8  -y6  <=  75
el7_8:  y8  -y7  <=  53
el9_8:  y8  -y9  <=  10
el10_8: y8  -y10 <=  58
el2_9:  y9  -y2  <=  79
el3_9:  y9  -y3  <=  48
el4_9:  y9  -y4  <=  68
el5_9:  y9  -y5  <=  76
el6_9:  y9  -y6  <=  35
el7_9:  y9  -y7  <=  12
el8_9:  y9  -y8  <=  56
el10_9: y9  -y10 <=  63
el1_10: y10 -y1  <= 350
el6_10: y10 -y6  <=  65
el7_10: y10 -y7  <=  45
el8_10: y10 -y8  <=  22
el9_10: y10 -y9  <=  55

end
```

Az optimális megoldás pedig:

No.	Row name	St	Activity	Marginal
1	el2_1	B	13	
2	el3_1	B	13	
3	el4_1	B	13	
4	el5_1	B	13	
5	el6_1	NU	13	< eps
6	el8_1	B	13	
7	el10_1	NU	13	1
8	el3_2	B	0	
9	el5_2	B	0	
10	el7_2	B	0	
11	el8_2	B	0	
12	el9_2	B	0	
13	el10_2	B	0	
14	el1_3	B	-13	
15	el5_3	B	0	
16	el6_3	B	0	
17	el7_3	B	0	
18	el8_3	B	0	
19	el9_3	B	0	
20	el10_3	B	0	
21	el1_4	B	-13	
22	el2_4	B	0	
23	el3_4	B	0	
24	el5_4	B	0	
25	el8_4	B	0	
26	el9_4	B	0	
27	el10_4	B	0	
28	el1_5	B	-13	
29	el2_5	B	0	
30	el3_5	B	0	
31	el4_5	B	0	
32	el7_5	B	0	
33	el8_5	B	0	
34	el10_5	B	0	
35	el2_6	B	0	
36	el3_6	B	0	
37	el4_6	B	0	
38	el5_6	B	0	
39	el7_6	B	0	
40	el8_6	B	0	
41	el9_6	B	0	
42	el1_7	B	-13	
43	el2_7	B	0	

```
    44 el3_7        B              0
    45 el4_7        B              0
    46 el8_7        B              0
    47 el9_7        B              0
    48 el10_7       B              0
    49 el1_8        B            -13
    50 el2_8        B              0
    51 el3_8        B              0
    52 el4_8        B              0
    53 el5_8        B              0
    54 el6_8        B              0
    55 el7_8        B              0
    56 el9_8        B              0
    57 el10_8       B              0
    58 el2_9        B              0
    59 el3_9        B              0
    60 el4_9        B              0
    61 el5_9        B              0
    62 el6_9        B              0
    63 el7_9        B              0
    64 el8_9        B              0
    65 el10_9       B              0
    66 el1_10       B            -13
    67 el6_10       B              0
    68 el7_10       B              0
    69 el8_10       B              0
    70 el9_10       B              0

   No. Column name  St   Activity     Marginal
------ ------------ -- ------------- -------------
     1 y1           B             13
     2 y10          NL             0         < eps
     3 y2           NL             0         < eps
     4 y3           NL             0         < eps
     5 y4           NL             0         < eps
     6 y5           NL             0         < eps
     7 y6           B              0
     8 y8           NL             0         < eps
     9 y7           NL             0         < eps
    10 y9           NL             0         < eps
```

A legrövidebb út most $10 \leftarrow 1$, aminek a hossza 13.

5.17. példa megoldása.

A példában meghatározott feladatra a duál feladat:

10.31. kód.

```
Min

+22.9x1_2+59.3x1_3 + 56.3x1_4+96.8x1_5+15.6x1_6 +100.1x1_7
         +67.4x2_3 + 24.7x2_4+44.5x2_5+38.5x2_6 + 97.6x2_7
                   +104.7x3_4+49.7x3_5+46.8x3_6 +103.2x3_7
                             +53.9x4_5+88x4_6   +105.6x4_7
                                      +17.6x5_6 + 15.2x5_7
                                                +130.6x6_7

subject to

csp1: +x1   -x1_2 -x1_3 -x1_4 -x1_5 -x1_6 -x1_7 >= 0
csp2: +x1_2       -x2_3 -x2_4 -x2_5 -x2_6 -x2_7 >= 0
csp3: +x1_3 +x2_3       -x3_4 -x3_5 -x3_6 -x3_7 >= 0
csp4: +x1_4 +x2_4 +x3_4       -x4_5 -x4_6 -x4_7 >= 0
csp5: +x1_5 +x2_5 +x3_5 +x4_5       -x5_6 -x5_7 >= 0
csp6: +x1_6 +x2_6 +x3_6 +x4_6 +x5_6       -x6_7 >= 0
csp7: +x1_7 +x2_7 +x3_7 +x4_7 +x5_7 +x6_7       >= 1

end
```

5.18. példa megoldása.

A megoldáshoz nézzük a 5.5. példa megoldását. Mivel ebben az esetben ha egy él hosszát megváltoztatjuk, akkor a másik irányba is meg kéne változtatni az élnek a hosszát, ami azt jelentené, hogy a célfüggvényben két együttható együttes megváltoztatását kéne vizsgálni. Az érzékenységvizsgálat segítségével kizárólag egy paraméter megváltozása esetén tudunk biztosat mondani.

Mivel azonban ebben az esetben a két együttható összefügg, ezért a lehetséges határokat meg tudjuk határozni. Ha olyan élt vizsgálunk, amely a legrövidebb út része, akkor azt kell vizsgálnunk, hogy meddig növelhető az él hossza, úgy, hogy bennmaradjon az adott él a bázisban. Mivel ekkor a másik irányba mutató él is ugyanannyival lesz hosszabb, ezért nem alakulhat ki olyan, hogy az eredeti irány helyett a másik irány legyen a legrövidebb út része. Tehát elég, ha az érzékenységi jelentésben bázisváltozó esetén az `Obj coef range` oszlopot vizsgáljuk.

A 5.5. példában nézzük például az x6_5 bázisváltozót:

```
   No. Column name  St Obj coef range
------ ------------ -- -------------
    35 x6_5         BS      -6.00000
                            51.80000
```

Ebben az esetben tehát ha az él hossza 51,8 alatt van, akkor nem változik a legrövidebb út, felette viszont igen.

10.5. Kritikus út feladatok

6.10. példa megoldása.

A gondolatmenet mindegyik kód esetében azonos, ahogy a megoldás is, ezért a felírásokat csak felsorolva adjuk meg. A 6.3. kód esetén a következőképpen néz ki az LP feladat módosítása:

10.32. kód.

```
min
+ 50rA + 70rB + 30rC + 25rD+ 15rE + 55rI
+120rJ +110rK

subject to

 y8 - y1 <= 62

el1_2: y2 - y1 + rA >= 30
el1_3: y3 - y1      >=  6
el2_4: y4 - y2 + rE >= 10
el2_5: y5 - y2 + rB >= 15
el2_7: y7 - y2 + rD >= 10
el3_4: y4 - y3      >=  2
el3_6: y6 - y3      >=  2
el4_6: y6 - y4 + rI >=  5
el5_7: y7 - y5 + rC >= 25
el6_8: y8 - y6 + rJ >= 10
el7_8: y8 - y7 + rK >= 13

rA: rA <=  2
rE: rE <=  2
rB: rB <=  8
rD: rD <=  3
```

```
rI: rI <=  1
rC: rC <= 19
rJ: rJ <=  5
rK: rK <=  8

end
```

A 6.6. kód esetén pedig a következőképpen néz ki az LP feladat:

10.33. kód.

```
min
+ 50rA + 70rB + 30rC + 25rD+ 15rE + 55rI
+120rJ +110rK

subject to

yVeg<=62

kB_A: yA -yB -rA <= -30
kC_B: yB -yC -rB <= -15
kD_A: yA -yD -rA <= -30
kE_A: yA -yE -rA <= -30
kG_F: yF -yG     <= - 6
kH_F: yF -yH     <= - 6
kI_E: yE -yI -rE <= -10
kI_G: yG -yI     <= - 2
kJ_I: yI -yJ -rI <= - 5
kJ_H: yH -yJ     <= - 2
kK_C: yC -yK -rC <= -25
kK_D: yD -yK -rD <= -10

tA: yA -yVeg -rA <= -30
tB: yB -yVeg -rB <= -15
tC: yC -yVeg -rC <= -25
tD: yD -yVeg -rD <= -10
tE: yE -yVeg -rE <= -10
tF: yF -yVeg     <= - 6
tG: yG -yVeg     <= - 2
tH: yH -yVeg     <= - 2
tI: yI -yVeg -rI <= - 5
tJ: yJ -yVeg -rJ <= -10
tK: yK -yVeg -rK <= -13

rA: rA <=  2
rE: rE <=  2
rB: rB <=  8
```

```
rD: rD <=  3
rI: rI <=  1
rC: rC <= 19
rJ: rJ <=  5
rK: rK <=  8

end
```

A 6.4. kód esetén a felírás bonyolultabb, mivel ekkor az eddig használt célfüggvényből a következő korlátnak kéne a modellbe kerülnie:

```
+ (30-rA) x1_2 +       6 x1_3 + (10-rE) x2_4
+ (15-rB) x2_5 + (10-rD) x2_7 +       2 x3_4
+       2 x3_6 +  (5-rI) x4_6 + (25-rC) x5_7
+ (10-rJ) x6_8 + (13-rK) x7_8                <= 62
```

Ami viszont nem lineáris. Természetesen megadhatjuk a 10.32. kódhoz tartozó LP feladat duálját is, ami lineáris feladat lesz, de ebben az esetben nehéz értelmezni a döntési változókat.

6.11. példa megoldása.

A megoldás ebben az esetben is bármelyik formalizációval is megadható, viszont talán a 6.7. kódban szereplő felírás esetén a legegyszerűbb a menetrend meghatározása. Először a gráf éleinek meghatározzuk az előzményeit:

Vonat	Várható menetidő	Korábbi vonatok
14	6	
15	7	
25	3	
35	5	
38	12	
46	8	14, 54
54	3	15, 25, 35
57	6	15, 25, 35
58	9	15, 25, 35
67	4	46
78	3	57, 67

Az LP kód pedig:

10.34. kód.

```
Min

vVeg

Subject to
k14_0:      -v14 <= - 6
k15_0:      -v15 <= - 7
k25_0:      -v25 <= - 3
k35_0:      -v35 <= - 5
k38_0:      -v38 <= -12
k46_14: v14 -v46 <= - 8
k46_54: v54 -v46 <= - 8
k54_15: v15 -v54 <= - 3
k54_25: v25 -v54 <= - 3
k54_35: v35 -v54 <= - 3
k57_15: v15 -v57 <= - 6
k57_25: v25 -v57 <= - 6
k57_35: v35 -v57 <= - 6
k58_15: v15 -v58 <= - 9
k58_25: v25 -v58 <= - 9
k58_35: v35 -v58 <= - 9
k67_46: v46 -v67 <= - 4
k78_57: v57 -v78 <= - 3
k78_67: v67 -v78 <= - 3

t14: v14 -vVeg <= 0
t15: v15 -vVeg <= 0
t25: v25 -vVeg <= 0
t35: v35 -vVeg <= 0
t38: v38 -vVeg <= 0
t46: v46 -vVeg <= 0
t54: v54 -vVeg <= 0
t57: v57 -vVeg <= 0
t58: v58 -vVeg <= 0
t67: v67 -vVeg <= 0
t78: v78 -vVeg <= 0

end
```

Az optimális megoldás pedig:

```
   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 k14_0        NU            -6         < eps
```

```
     2 k15_0        NU            -7            -1
     3 k25_0        B             -7
     4 k35_0        NU            -5         < eps
     5 k38_0        NU           -12         < eps
     6 k46_14       B            -12
     7 k46_54       NU            -8            -1
     8 k54_15       NU            -3            -1
     9 k54_25       NU            -3         < eps
    10 k54_35       B             -5
    11 k57_15       NU            -6         < eps
    12 k57_25       B             -6
    13 k57_35       B             -8
    14 k58_15       B            -18
    15 k58_25       B            -18
    16 k58_35       B            -20
    17 k67_46       NU            -4             -1
    18 k78_57       B            -12
    19 k78_67       NU            -3             -1
    20 t14          B            -19
    21 t15          B            -18
    22 t25          B            -18
    23 t35          B            -20
    24 t38          B            -13
    25 t46          B             -7
    26 t54          B            -15
    27 t57          B            -12
    28 t58          NU             0          < eps
    29 t67          B             -3
    30 t78          NU             0             -1

   No. Column name  St   Activity
------ ------------ -- -------------
     1 vVeg         B             25
     2 v14          B              6
     3 v15          B              7
     4 v25          B              7
     5 v35          B              5
     6 v38          B             12
     7 v46          B             18
     8 v54          B             10
     9 v57          B             13
    10 v58          B             25
    11 v67          B             22
    12 v78          B             25
```

Ebben az esetben a változók értékei a vonatok megérkezését adják, tehát a menetrend megadásához ezekből a menetidőt kivonva kaphatunk egy lehetséges menetrendet:

Vonat	Várható menetidő	Érkezés	Indulás
14	6	6	0
15	7	7	0
25	3	7	4
35	5	5	0
38	12	12	0
46	8	18	10
54	3	10	7
57	6	13	7
58	9	25	16
67	4	22	18
78	3	25	22

6.12. példa megoldása.

Hasonló jellegű kérdéseket megvizsgáltunk a legrövidebb út feladat esetén 5.3. példa, 5.4. példa, 5.7. példa és 5.11. példa esetén. Mivel a kritikus út feladat felfogható leghosszabb út feladatként is, ezért a kérdésre adott válasz teljesen analóg a legrövidebb út feladat esetével.

Kezdjük a feladat megoldását a B tevékenységgel. A B tevékenység a kritikus út része, ezért ebben az esetben használhatjuk az érzékenységvizsgálati határokat a kérdés megválaszolásához. A B tevékenység hossza a 6.3. kódban `el2_7` korlát jobboldala. Ezért ehhez a korláthoz tartozó `Activity range` érdekel bennünket:

```
   No. Row name     St Activity range
------ ------------ -- --------------
     4 el2_5        NL      -13.00000
                                 +Inf
```

Tehát a B tevékenység hossza tetszőleges mértékben növelhető, a kritikus út tagja marad; amely megállapítás valahol trivialitás. Érdekesebb, hogy mennyivel rövidíthető, hogy a kritikus út éle maradjon. Az `Activity range` értékei alapján a tevékenység hossza -13 egységnél hosszabb, akkor a kritikus út része marad. A negatív érték megint kicsit gondolkodóba ejthet minket, de megérthetjük, ha a B, C és K tevékenység együttesét nézzük: ha a B tevékenység hossza -13, akkor a B, C és K tevékenységek együttes hossza 25, pont annyi mint az E, I és J tevékenység együttes hossza. Tehát ha a B tevékenység hossza -13 egység alá esik a kritikus tevékenységek láncolata a $A \to B \to C \to K$ útról a $A \to E \to I \to J$ útra változik.

Nézzük a D tevékenységet. A 6.3. kódban a D tevékenységet az `el2_7` korlát adja meg. Ezen korlát esetén az `Activity` értéke 40, a `Lower bound` pedig 10. A korlát nem egyenlőség formájában teljesül, tehát nem lehet az érzékenységvizsgálati határokat figyelembe venni. Ha a D tevékenység hosszát 10 egység alá csökkentjük, akkor a hozzá tartozó eltérésváltozó kikerül a bázisból. A legrövidebb út feladatnál is láttuk, hogy az a tény, hogy egy élhez tartozó eltérésváltozó kikerül a LP feladat bázisába nem egyenlő azzal, hogy a legrövidebb út része lesz. Most is óvatosan kell eljárni, növeljük a tevékenység hosszát 40 egység fölé egy kicsivel, és nézzük meg mi történik. Az új feladat esetén már a legrövidebb út része lesz. De ez nem törvényszerű, erre mutat példát a H tevékenység. A H tevékenység esetén az `Activity` érték 39. Ha a tevékenység hosszát 39 felé emeljük (39,1-re), akkor a tevékenységhez tartozó eltérésváltozó kikerül a bázisból, de nem a tevékenység a kritikus út része. Mivel ebben az új feladatban a korláthoz tartozó eltérésváltozó nem bázisváltozó, ezért tekinthetjük az `Activity range` értékeket:

```
   No. Row name     St Activity range
------ ------------ -- --------------
     7 el3_6        NL       39.00000
                             67.00000
```

Tehát ez a bázis érvényben marad addig (= a H tevékenység nem lesz a legrövidebb út része) amíg a H hossza 39 és 67 között lesz. Amennyiben a tevékenység hossza 67 felé emelkedik, akkor a tevékenység már kritikus tevékenységgé válik.

A kérdést a csomópontos formalizáció (pl. 6.4. kód) segítségével is meg lehetne válaszolni. Itt a D tevékenységet az `x2_7` változó jelenti. Most a redukált költség és a `Obj coef range` oszlop lenne segítségünkre.

Azt a kérdést, hogy a H tevékenység hosszát mennyivel kell növelni, hogy kritikus tevékenységgé váljon, egyetlen (újabb) futtatással is meg tudjuk mondani. A feladat megint csak analóg lesz a legrövidebb út feladattal: kritikus tevékenységek esetén a korlátokat cseréljük egyenlőségre, és a 3-as és 6-os gyűrűket húzzuk minél távolabbra:

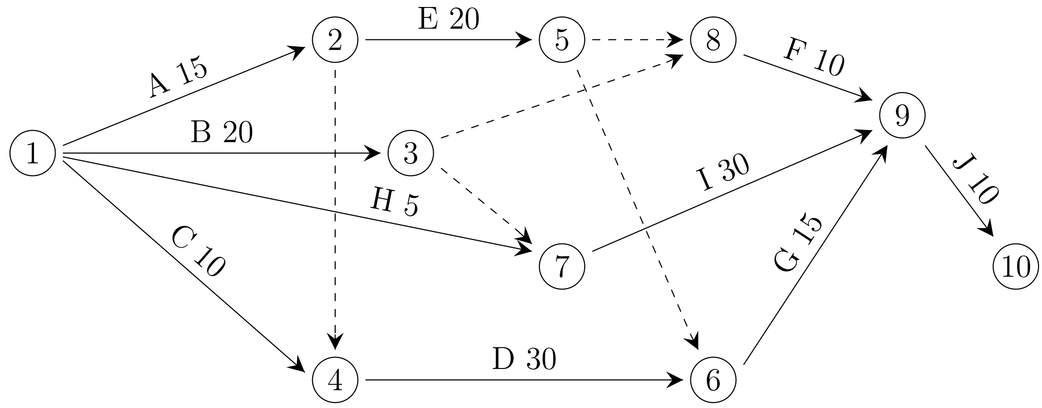
10.35. kód.

```
max
y6-y3

subject to
el1_2: y2 - y1  = 30
el1_3: y3 - y1 >=  6
el2_4: y4 - y2 >= 10
el2_5: y5 - y2  = 15
el2_7: y7 - y2 >= 10
el3_4: y4 - y3 >=  2
el3_6: y6 - y3 >=  2
el4_6: y6 - y4 >=  5
el5_7: y7 - y5  = 25
el6_8: y8 - y6 >= 10
el7_8: y8 - y7  = 13

end
```

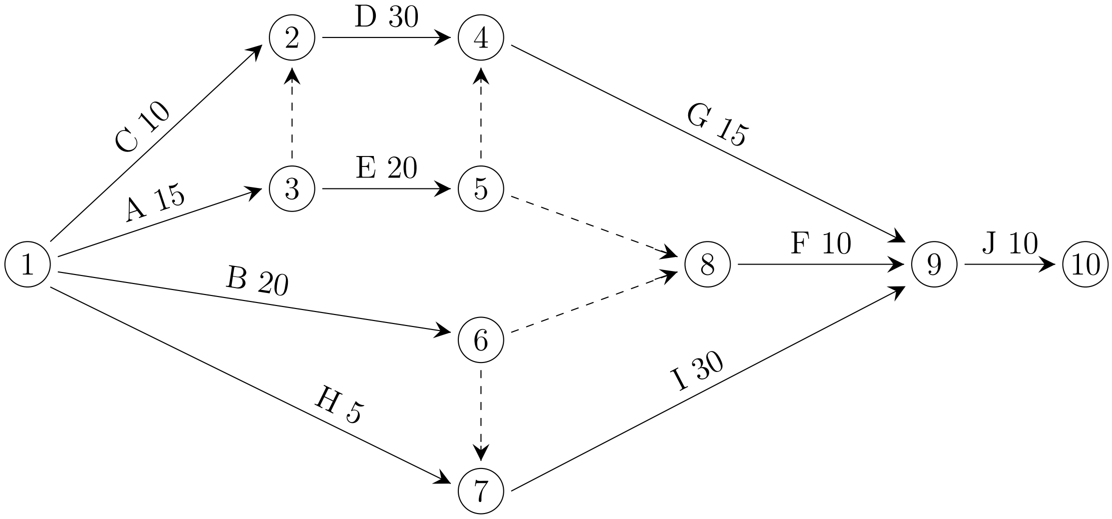
A 10.35. kódhoz tartozó feladat esetén a célfüggvény optimális értéke 67. Tehát a H tevékenység hossza maximum 67egységig növelhető úgy, hogy ne változzon kritikus tevékenységgé.

6.13. példa megoldása.

A gráf egy lehetséges felrajzolása a következő:



Amit átalakítva egy másmilyen rendezési formában is megadhatunk:



6.14. példa megoldása.

A primál feladat `eli_j:` korlátjához tartozó duálváltozót `xi_j`-vel jelöljük. Ebben az esetben a duál feladatban 11 változó lesz, a célfüggvényben pedig a primál korlátok jobboldala lesz az együtthatójuk. A duál feladatban 8 korlát lesz, az első és az utolsó kivételével ezeknek 0 lesz a jobboldalán. Tehát a következőképpen néz ki a duál feladat:

10.36. kód.

```
max
+30x1_2 + 6x1_3 +10x2_4 +15x2_5 +10x2_7 + 2x3_4 + 2x3_6
+ 5x4_6 +25x5_7 +10x6_8 +13x7_8

subject to
y1: -x1_2 -x1_3                <=-1
y2: +x1_2 -x2_4 -x2_5 -x2_7 <= 0
y3: +x1_3 -x3_4 -x3_6       <= 0
y4: +x2_4 +x3_4 -x4_6       <= 0
y5: +x2_5 -x5_7             <= 0
y6: +x3_6 +x4_6 -x6_8       <= 0
y7: +x2_7 +x5_7 -x7_8       <= 0
y8: +x6_8 +x7_8             <= 1

end
```

Amennyiben az első és az utolsó korlátot -1-el megszorozzuk, akkor olyan LP-t kapunk, ami annyiban tér el a csomópontos felírástól, hogy egyenlőtlenségekkel vannak a korlátok korlátozva. Azonban könnyen belátható, hogy mivel az első korlát `x1_2 +x1_3 >= 1`és az utolsó `x6_8 +x7_8 <= 1`, ezért az összes korlát csakis egyenlőség formájában teljesülhet.

6.15. példa megoldása.

Amennyiben nem kellene a félidő, akkor a 6.7. kódban szereplő felírás esetén az LP kód a következőképpen nézne ki:

10.37. kód.

```
Min

vVeg

Subject to
kA_0:    -vA <= - 3
kB_0:    -vB <= -29
kC_0:    -vC <= -13
kD_A: vA -vD <= -33
kE_A: vA -vE <= -12
kF_B: vB -vF <= - 6
kF_D :vD -vF <= - 6
kG_B: vB -vG <= - 2
kG_D: vD -vG <= - 2
kH_E: vE -vH <= -25
kH_F: vF -vH <= -25
kI_C: vC -vI <= - 7

tA: vA -vVeg <= 0
tB: vB -vVeg <= 0
tC: vC -vVeg <= 0
tD: vD -vVeg <= 0
tE: vE -vVeg <= 0
tF: vF -vVeg <= 0
tG: vG -vVeg <= 0
tH: vH -vVeg <= 0
tI: vI -vVeg <= 0

end
```

Ehhez a felíráshoz be kell vezetnünk egy `vFel` változót, amely az első félidő végét adja meg. Mivel mind a kettő félidő ugyanolyan hosszú, és a kettő közötti szünet egységnyi, ezért a következő korlátot kell a kódhoz adni: `2vFel +1 = vVeg`, ami átrendezve: `2vFel -vVeg = -1`.

Ezen kívül meg kell határozni minden tevékenységre, hogy melyik félidőbe tartozik. Ehhez bevezetünk `y` bináris változókat minden tevékenységhez, amely ha 1, akkor a tevékenységet az első félidőben végezzük el. Másképpen, ha például az `A` tevékenység esetén, ha `yA=1`, akkor `vA <= vFel`, különben pedig `3+vFel +1 <= vA`

Tehát az LP feladathoz az `A` tevékenység esetén a következő korlátokat kell adni: `vA -vFel + MyA <= M`, illetve `vA -vFel +MyA >= 3 + 1`, ahol `M` egy kellően nagy szám.

10.38. kód.

```
Min

vVeg

Subject to
kA_0:     -vA <= - 3
kB_0:     -vB <= -29
kC_0:     -vC <= -13
kD_A: vA -vD <= -33
kE_A: vA -vE <= -12
kF_B: vB -vF <= - 6
kF_D: vD -vF <= - 6
kG_B: vB -vG <= - 2
kG_D: vD -vG <= - 2
kH_E: vE -vH <= -25
kH_F: vF -vH <= -25
kI_C: vC -vI <= - 7

tA: vA -vVeg <= 0
tB: vB -vVeg <= 0
tC: vC -vVeg <= 0
tD: vD -vVeg <= 0
tE: vE -vVeg <= 0
tF: vF -vVeg <= 0
tG: vG -vVeg <= 0
tH: vH -vVeg <= 0
tI: vI -vVeg <= 0
```

```
felido: 2vFel -vVeg = -1

yA_f: vA -vFel +100yA <= 100
yB_f: vB -vFel +100yB <= 100
yC_f: vC -vFel +100yC <= 100
yD_f: vD -vFel +100yD <= 100
yE_f: vE -vFel +100yE <= 100
yF_f: vF -vFel +100yF <= 100
yG_f: vG -vFel +100yG <= 100
yH_f: vH -vFel +100yH <= 100
yI_f: vI -vFel +100yI <= 100

yA_a: vA -vFel +100yA >=  4
yB_a: vB -vFel +100yB >= 30
yC_a: vC -vFel +100yC >= 14
yD_a: vD -vFel +100yD >= 34
yE_a: vE -vFel +100yE >= 13
yF_a: vF -vFel +100yF >=  7
yG_a: vG -vFel +100yG >=  3
yH_a: vH -vFel +100yH >= 26
yI_a: vI -vFel +100yI >=  8


Binary
yA
yB
yC
yD
yE
yF
yG
yH
yI

end
```

Az optimális megoldás pedig:

```
   No. Column name       Activity
------ ------------    -------------
     1 vVeg                       73
     2 vA                          3
     3 vB                         36
     4 vC                         50
     5 vD                         36
     6 vE                         15
```

```
  7 vF                            48
  8 vG                            39
  9 vH                            73
 10 vI                            57
 11 vFel                          36
 12 yA             *               1
 13 yB             *               1
 14 yC             *               0
 15 yD             *               1
 16 yE             *               1
 17 yF             *               0
 18 yG             *               0
 19 yH             *               0
 20 yI             *               0
```

Tehát a félidő hossza 36 egység, és az *A*,*B*, *D*,*E* tevékenységeket kell elvégezni a szünet előtt.

6.16. példa megoldása.

Itt is a legegyszerűbb, ha a 6.7. kódban szereplő felírást használjuk. A megoldásban azokra a párokra (i, j), amelyeket nem végezhetjük egyszerre, bevezetünk egy y_{ij} bináris változókat. Amennyiben $y_{ij} = 1$, akkor az i tevékenységet végezzük el előbb, amennyiben $y_{ij} = 0$, akkor pedig a j-t. Tehát matematikailag azt kell meghatároznunk, hogy:

$$\begin{aligned} y_{ij} = 1 \Leftrightarrow \quad & v_j \geq v_i + h_j \\ y_{ij} = 0 \Leftrightarrow \quad & v_i \geq v_j + h_i \end{aligned}$$

Ahol h_i az i-edik tevékenység hossza. Az LP kódban tehát a következő korlátokat kell megadni:

$$\begin{aligned} M(1 - y_{ij}) \quad & \geq v_i + h_j - v_j \\ My_{ij} \quad & \geq v_j + h_i - v_i \end{aligned}$$

Ami így teljesíti az i, j párra vonatkozó feltételeket. Az egyenlőtlenségekben M egy kellően nagy szám (A kódban 100-ra állítottuk be).

10.39. kód.

```
Min
```

```
vVeg

Subject to
kA_0:    -vA <= -10
kB_0:    -vB <=  -8
kC_A: vA -vC <=  -7
kD_B: vB -vD <=  -5
kE_B: vB -vE <=  -8
kF_A: vA -vF <=  -9
kG_C: vC -vG <=  -4
kG_D: vD -vG <=  -4
kH_C: vC -vH <=  -6
kH_D: vD -vH <=  -6
kI_H: vH -vI <=  -3
kI_E: vE -vI <=  -3
kJ_F: vF -vJ <=  -6
kJ_G: vG -vJ <=  -6

tA: vA -vVeg <= 0
tB: vB -vVeg <= 0
tC: vC -vVeg <= 0
tD: vD -vVeg <= 0
tE: vE -vVeg <= 0
tF: vF -vVeg <= 0
tG: vG -vVeg <= 0
tH: vH -vVeg <= 0
tI: vI -vVeg <= 0
tJ: vJ -vVeg <= 0

eC: vC -vD -100yD_C >= -93
eD: vD -vC +100yD_C >=   5

eE: vE -vF -100yF_E >= -92
eF: vF -vE +100yF_E >=   9

eI: vI -vJ -100yJ_I >= -94
eJ: vJ -vI +100yJ_I >=   3

Binary
yD_C
yF_E
yJ_I

end
```

Az optimális megoldás pedig:

```
   No. Column name       Activity
------ ------------    -------------
     1 vVeg                     32
     2 vA                       10
     3 vB                        8
     4 vC                       20
     5 vD                       13
     6 vE                       16
     7 vF                       26
     8 vG                       26
     9 vH                       26
    10 vI                       29
    11 vJ                       32
    12 yD_C         *            1
    13 yF_E         *            0
    14 yJ_I         *            0
```

Tehát a párok közül a D tevékenységet végezzük el először, majd a C-t, az E-t az F előtt, és a J-t az I előtt. A projekt időtartama így 32 időegység.

10.6. Hozzárendelési feladatok

7.4. példa megoldása.

Ez egy egyszerűen felírható hozzárendelési feladat, ahol most nem költséget, hanem időt akarunk minimalizálni.

10.40. kód.

```
Min
ido:
+64x1_1 +83x1_2 +67x1_3 +55x1_4 +60x1_5 +41x1_6 +35x1_7
+32x2_1 +44x2_2 +31x2_3 +60x2_4 +99x2_5 +38x2_6 +61x2_7
+56x3_1 +92x3_2 +97x3_3 +68x3_4 +75x3_5 +33x3_6 +85x3_7
+40x4_1 +46x4_2 +60x4_3 +39x4_4 +79x4_5 +97x4_6 +58x4_7
+59x5_1 +61x5_2 +91x5_3 +54x5_4 +66x5_5 +31x5_6 +83x5_7
+45x6_1 +36x6_2 +72x6_3 +66x6_4 +51x6_5 +68x6_6 +99x6_7
+52x7_1 +75x7_2 +53x7_3 +87x7_4 +96x7_5 +99x7_6 +79x7_7

subject to
G1:  +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_7 = 1
G2:  +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 +x2_7 = 1
```

```
G3:  +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 +x3_7 = 1
G4:  +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 +x4_7 = 1
G5:  +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 +x5_7 = 1
G6:  +x6_1 +x6_2 +x6_3 +x6_4 +x6_5 +x6_6 +x6_7 = 1
G7:  +x7_1 +x7_2 +x7_3 +x7_4 +x7_5 +x7_6 +x7_7 = 1

M1:  +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1 +x7_1 = 1
M2:  +x1_2 +x2_2 +x3_2 +x4_2 +x5_2 +x6_2 +x7_2 = 1
M3:  +x1_3 +x2_3 +x3_3 +x4_3 +x5_3 +x6_3 +x7_3 = 1
M4:  +x1_4 +x2_4 +x3_4 +x4_4 +x5_4 +x6_4 +x7_4 = 1
M5:  +x1_5 +x2_5 +x3_5 +x4_5 +x5_5 +x6_5 +x7_5 = 1
M6:  +x1_6 +x2_6 +x3_6 +x4_6 +x5_6 +x6_6 +x7_6 = 1
M7:  +x1_7 +x2_7 +x3_7 +x4_7 +x5_7 +x6_7 +x7_7 = 1

end
```

Az optimális megoldás:

```
   No. Column name  St   Activity
------ ------------ -- -------------
     7 x1_7         B              1
    10 x2_3         B              1
    20 x3_6         B              1
    25 x4_4         B              1
    33 x5_5         B              1
    37 x6_2         B              1
    43 x7_1         B              1
```

A következő táblázatban látható, hogy az optimális megoldásban az adott feladatot végül melyik munkás, mennyi idő alatt végzi el:

Gép	Munkás	Idő
1	7	35
2	3	31
3	6	33
4	4	39
5	5	66
6	2	36
7	1	52

A gépek összesített működési ideje 292 perc. Ahhoz, hogy az összes alkatrész elkészüljön, összesen 66 perc szükséges, mivel, ha a munkások egyszerre kezdik el az alkatrészek gyártását, akkor az 5. gépen az 5. munkás fejezi be legkésőbb a számára kijelölt feladatot.

7.5. példa megoldása.

Mivel több taxis van, mint ahány utas, ezért a feladatot ki kell egyensúlyozni, tehát két fiktív utast adunk a feladathoz. Amelyik taxis egy fiktív utashoz rendelődik hozzá, az valójában nem kap „rendes" utast, tehát számára az utazási idő 0. Ezzel a két fiktív utassal a feladatra kaptunk egy négyzetes mátrixot, tehát innentől kezdve ugyanúgy kell a feladatot megoldani, mint az eddigi példákban.

10.41. kód.

```
Min
ido:
+ 5x1_1+12x1_2+20x1_3+ 8x1_4+11x1_5+ 4x1_6+0x1_7+0x1_8
+ 7x2_1+ 4x2_2+ 5x2_3+ 2x2_4+21x2_5+ 4x2_6+0x2_7+0x2_8
+ 5x3_1+14x3_2+ 2x3_3+ 2x3_4+21x3_5+11x3_6+0x3_7+0x3_8
+ 9x4_1+14x4_2+14x4_3+ 8x4_4+ 4x4_5+ 3x4_6+0x4_7+0x4_8
+ 8x5_1+ 8x5_2+ 4x5_3+12x5_4+ 7x5_5+10x5_6+0x5_7+0x5_8
+11x6_1+ 5x6_2+10x6_3+ 5x6_4+ 6x6_5+ 7x6_6+0x6_7+0x6_8
+ 3x7_1+ 3x7_2+20x7_3+ 9x7_4+21x7_5+ 8x7_6+0x7_7+0x7_8
+ 4x8_1+ 9x8_2+16x8_3+ 9x8_4+13x8_5+ 7x8_6+0x8_7+0x8_8

subject to
T1:  +x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_7 +x1_8 = 1
T2:  +x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 +x2_7 +x2_8 = 1
T3:  +x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 +x3_7 +x3_8 = 1
T4:  +x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +x4_6 +x4_7 +x4_8 = 1
T5:  +x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +x5_6 +x5_7 +x5_8 = 1
T6:  +x6_1 +x6_2 +x6_3 +x6_4 +x6_5 +x6_6 +x6_7 +x6_8 = 1
T7:  +x7_1 +x7_2 +x7_3 +x7_4 +x7_5 +x7_6 +x7_7 +x7_8 = 1
T8:  +x8_1 +x8_2 +x8_3 +x8_4 +x8_5 +x8_6 +x8_7 +x8_8 = 1

U1:  +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1 +x7_1 +x8_1 = 1
U2:  +x1_2 +x2_2 +x3_2 +x4_2 +x5_2 +x6_2 +x7_2 +x8_2 = 1
U3:  +x1_3 +x2_3 +x3_3 +x4_3 +x5_3 +x6_3 +x7_3 +x8_3 = 1
U4:  +x1_4 +x2_4 +x3_4 +x4_4 +x5_4 +x6_4 +x7_4 +x8_4 = 1
U5:  +x1_5 +x2_5 +x3_5 +x4_5 +x5_5 +x6_5 +x7_5 +x8_5 = 1
U6:  +x1_6 +x2_6 +x3_6 +x4_6 +x5_6 +x6_6 +x7_6 +x8_6 = 1
Uf7: +x1_7 +x2_7 +x3_7 +x4_7 +x5_7 +x6_7 +x7_7 +x8_7 = 1
Uf8: +x1_8 +x2_8 +x3_8 +x4_8 +x5_8 +x6_8 +x7_8 +x8_8 = 1
```

```
end
```

Az optimális megoldás:

```
No. Column name  St   Activity
  6 x1_6         B           1
 12 x2_4         B           1
 19 x3_3         B           1
 29 x4_5         B           1
 39 x5_7         B           1
 48 x6_8         B           1
 50 x7_2         B           1
 57 x8_1         B           1
```

Tehát a következő hozzárendelések valósulnak meg:

Taxis	Utas	Idő
1	6	4
2	4	2
3	3	2
4	5	4
5	7	0
6	8	0
7	2	3
8	1	4

Az 5. és 6. taxisnak nem marad utas, így az ő utazási idejük 0, míg az összesített utazási idő összesen 19 perc. Az 1. és a 6. utasnak lenne közelebbi taxis, mind a kettőjüknek 4 percet kell a taxira várniuk, pedig lenne 3 percnyire is egy autó.

7.6. példa megoldása.

Ekkor csak olyan változókat kell bevezetni, amelyeket megjelölték a rangsorolásban a csoportok. A célfüggvény együtthatói a megadott mátrix értékei lesznek, és most is minimalizálni akarjuk a feladatot.

10.42. kód.

```
Min
pont:
+2x1_2 +1x1_3 +3x1_7
+3x2_1 +1x2_2 +2x2_3
```

```
+3x3_3 +1x3_5 +2x3_6
+2x4_1 +3x4_2 +1x4_5
+1x5_1 +2x5_4 +3x5_7
+2x6_2 +1x6_3 +3x6_5
+3x7_1 +1x7_6 +2x7_7

subject to
cs1: +x1_2 +x1_3 +x1_7 = 1
cs2: +x2_1 +x2_2 +x2_3 = 1
cs3: +x3_3 +x3_5 +x3_6 = 1
cs4: +x4_1 +x4_2 +x4_5 = 1
cs5: +x5_1 +x5_4 +x5_7 = 1
cs6: +x6_2 +x6_3 +x6_5 = 1
cs7: +x7_1 +x7_6 +x7_7 = 1

t1: +x2_1 +x4_1 +x5_1 +x7_1 = 1
t2: +x1_2 +x2_2 +x4_2 +x6_2 = 1
t3: +x1_3 +x2_3 +x3_3 +x6_3 = 1
t4: +x5_4                   = 1
t5: +x3_5 +x4_5 +x6_5       = 1
t6: +x3_6 +x7_6             = 1
t7: +x1_7 +x5_7 +x7_7       = 1

end
```

Az optimális megoldás:

```
   No. Column name  St   Activity
------ ------------ -- -------------
     3 x1_7         B              1
     5 x2_2         B              1
     8 x3_5         B              1
    10 x4_1         B              1
    14 x5_4         B              1
    17 x6_3         B              1
    20 x7_6         B              1
```

Tehát a csoportok a következő témákat kapják:

Csoport	Téma
1	7
2	2
3	5
4	1
5	4
6	3
7	6

7.7. példa megoldása.

Ahhoz, hogy a megadott táblázatból a célfüggvény együtthatóinak a mátrixát kapjuk, hozzá kell adnunk egy sort, amely Anna második futását adja meg. Ebben a sorban minden szakasz ideje 4 perccel több, mint az Anna eredeti sorában lévő értékek, kivéve az első szakaszban, mivel az csak Anna első futása esetén lehet. Mivel így kiegyensúlyozatlan lesz a feladat, mivel egyel több futó (mivel így Anna kettőnek számít) lesz, mint szakasz, ezért a feladathoz hozzá kell adni egy fiktív szakaszt. Amelyik futó ehhez a fiktív szakaszhoz rendelődik hozzá, annak nem kell futnia a váltóban. Tehát ekkor a következőképpen alakul az eredeti táblázat:

	1.sz.	2.sz.	3.sz.	4.sz.	5.sz.	6.sz.	7.sz.	8.sz.	9.sz.	f.sz.
Anna	11	13	13	16	14	12	12	11	13	0
Anna2		17	17	20	18	16	16	15	17	0
Béla	18	16	19	25	23	19	20	12	14	0
Cili	16	24	14	15	18	16	23	17	14	0
Dávid	17	23	17	16	18	25	14	13	17	0
Eszter	17	17	12	12	18	13	17	24	18	0
Feri	18	16	17	22	13	14	15	16	19	0
Géza	19	15	15	20	14	15	14	21	18	0
Hedvig	22	21	14	15	15	19	13	13	18	0
Imre	20	24	14	24	18	21	21	18	16	0

A megoldáshoz pedig erre a mátrixra kell keresnünk a minimális hozzárendelést. De további megfontolás is szükséges: nem lehetséges pl. ha ‘Anna2’-höz a 2. szakaszt rendeljük, ‘Anna’-hoz pedig a 3. szakaszt. Ebben az esetben ténylegesen a 2. szakaszt fogja először futni a 3. szakaszt pedig másodszorra. Ezen konkrét feladat esetén érvelhetünk úgy, hogy Anna másodszorra futott szakasza mindig 4 perccel több, mint az első; tehát ha Annához két szakasz is lesz rendelve, akkor kicserélhető, hogy melyket futja 'elsőnek' és 'másodiknak' az össz idő nem változik. Ha Anna csak egy szakaszt fut, akkor nyilván azt 'előszörre' kell futni, különben nem lenne optimális a hozzárendelés.

10.43. kód.

```
Min
ido:
+11x1_1   +13x1_2   +13x1_3   +16x1_4   +14x1_5
          +12x1_6   +12x1_7   +11x1_8   +13x1_9   +0x1_10
          +17x2_2   +17x2_3   +20x2_4   +18x2_5
          +16x2_6   +16x2_7  +15x2_8    +17x2_9   +0x2_10
+18x3_1   +16x3_2   +19x3_3   +25x3_4   +23x3_5
          +19x3_6   +20x3_7   +12x3_8   +14x3_9   +0x3_10
+16x4_1   +24x4_2   +14x4_3   +15x4_4   +18x4_5
          +16x4_6   +23x4_7   +17x4_8   +14x4_9   +0x4_10
+17x5_1   +23x5_2   +17x5_3   +16x5_4   +18x5_5
          +25x5_6   +14x5_7   +13x5_8   +17x5_9   +0x5_10
+17x6_1   +17x6_2   +12x6_3   +12x6_4   +18x6_5
          +13x6_6   +17x6_7   +24x6_8   +18x6_9   +0x6_10
+18x7_1   +16x7_2   +17x7_3   +22x7_4   +13x7_5
          +14x7_6   +15x7_7   +16x7_8   +19x7_9   +0x7_10
+19x8_1   +15x8_2   +15x8_3   +20x8_4   +14x8_5
          +15x8_6   +14x8_7   +21x8_8   +18x8_9   +0x8_10
+22x9_1   +21x9_2   +14x9_3   +15x9_4   +15x9_5
          +19x9_6   +13x9_7   +13x9_8   +18x9_9   +0x9_10
+20x10_1 +24x10_2 +14x10_3 +24x10_4 +18x10_5
          +21x10_6 +21x10_7 +18x10_8 +16x10_9 +0x10_10

subject to
Anna:    +x1_1   +x1_2   +x1_3   +x1_4   +x1_5
                 +x1_6   +x1_7   +x1_8   +x1_9   +x1_10   = 1
Anna2:           +x2_2   +x2_3   +x2_4   +x2_5
                 +x2_6   +x2_7   +x2_8   +x2_9   +x2_10   = 1
Bela:    +x3_1   +x3_2   +x3_3   +x3_4   +x3_5
                 +x3_6   +x3_7   +x3_8   +x3_9   +x3_10   = 1
Cili:    +x4_1   +x4_2   +x4_3   +x4_4   +x4_5
                 +x4_6   +x4_7   +x4_8   +x4_9   +x4_10   = 1
David:   +x5_1   +x5_2   +x5_3   +x5_4   +x5_5
                 +x5_6   +x5_7   +x5_8   +x5_9   +x5_10   = 1
Eszter: +x6_1    +x6_2   +x6_3   +x6_4   +x6_5
                 +x6_6   +x6_7   +x6_8   +x6_9   +x6_10   = 1
Feri:    +x7_1   +x7_2   +x7_3   +x7_4   +x7_5
```

```
               +x7_6  +x7_7  +x7_8  +x7_9  +x7_10  = 1
Geza:   +x8_1  +x8_2  +x8_3  +x8_4  +x8_5
               +x8_6  +x8_7  +x8_8  +x8_9  +x8_10  = 1
Hedvig: +x9_1  +x9_2  +x9_3  +x9_4  +x9_5
               +x9_6  +x9_7  +x9_8  +x9_9  +x9_10  = 1
Imre:   +x10_1 +x10_2 +x10_3 +x10_4 +x10_5
               +x10_6 +x10_7 +x10_8 +x10_9 +x10_10 = 1

sz1:   +x1_1         +x3_1  +x4_1  +x5_1  +x6_1
                     +x7_1  +x8_1  +x9_1  +x10_1  = 1
sz2:   +x1_2  +x2_2  +x3_2  +x4_2  +x5_2  +x6_2
                     +x7_2  +x8_2  +x9_2  +x10_2  = 1
sz3:   +x1_3  +x2_3  +x3_3  +x4_3  +x5_3  +x6_3
                     +x7_3  +x8_3  +x9_3  +x10_3  = 1
sz4:   +x1_4  +x2_4  +x3_4  +x4_4  +x5_4  +x6_4
                     +x7_4  +x8_4  +x9_4  +x10_4  = 1
sz5:   +x1_5  +x2_5  +x3_5  +x4_5  +x5_5  +x6_5
                     +x7_5  +x8_5  +x9_5  +x10_5  = 1
sz6:   +x1_6  +x2_6  +x3_6  +x4_6  +x5_6  +x6_6
                     +x7_6  +x8_6  +x9_6  +x10_6  = 1
sz7:   +x1_7  +x2_7  +x3_7  +x4_7  +x5_7  +x6_7
                     +x7_7  +x8_7  +x9_7  +x10_7  = 1
sz8:   +x1_8  +x2_8  +x3_8  +x4_8  +x5_8  +x6_8
                     +x7_8  +x8_8  +x9_8  +x10_8  = 1
sz9:   +x1_9  +x2_9  +x3_9  +x4_9  +x5_9  +x6_9
                     +x7_9  +x8_9  +x9_9  +x10_9  = 1
sz10:  +x1_10 +x2_10 +x3_10 +x4_10 +x5_10 +x6_10
                     +x7_10 +x8_10 +x9_10 +x10_10 = 1
end
```

Az optimális megoldás:

```
   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 x1_1         B              1
    15 x2_6         B              1
    27 x3_8         B              1
    38 x4_9         B              1
    49 x5_10        B              1
    53 x6_4         B              1
    64 x7_5         B              1
    71 x8_2         B              1
    86 x9_7         B              1
    92 x10_3        B              1
```

Tehát a következőképpen osztják szét a szakaszokat:

szakasz	futó
1.sz.	Anna
2.sz.	Géza
3.sz.	Imre
4.sz.	Eszter
5.sz.	Feri
6.sz.	Anna2
7.sz.	Hedvig
8.sz.	Béla
9.sz.	Cili

Dávid nem fog futni a futóversenyen, és Anna az első, majd a 6. szakaszt fogja futni. Összesen a váltó végigfutásához 120 perc szükséges.

Ha nem mindig 4 perc a különbség Anna elsőre és másodszorra futott távja között (hanem a külöbnség változik), akkor a kapott minimális össz idő nem biztos, hogy megvalósítható. Ebben az esetben külön elő kell írni, hogy Anna második szakasza későbbinek kell lennie, mint az elsőnek. Tehát Anna csak akkor futhatja a másodiknak a második szakaszt, ha az elsőt futja elsőnek (ha Anna nincs hozzárendelve az első szakaszhoz, akkor a másodikat nem másodszorra futja, hanem elsőnek). Ezt a feltételt a `x2_2-x1_1<=0` korláttal tudjuk biztosítani. Hasonlóan: Anna csak akkor futhatja a 3. szakaszt másodiknak, ha az elsőt vagy másodikat már lefutotta: `x2_3-x1_1-x1_2<=0`. Természetesen az összes szakaszra fel kell írni ezt az összefüggést:

```
x2_2 - x1_1 <= 0
x2_3 - x1_1 - x1_2 <= 0
x2_4 - x1_1 - x1_2 - x1_3 <= 0
x2_5 - x1_1 - x1_2 - x1_3 - x1_4 <= 0
x2_6 - x1_1 - x1_2 - x1_3 - x1_4 - x1_5 <= 0
x2_7 - x1_1 - x1_2 - x1_3 - x1_4 - x1_5 - x1_6 <= 0
x2_8 - x1_1 - x1_2 - x1_3 - x1_4 - x1_5 - x1_6 - x1_7 <= 0
x2_9 - x1_1 - x1_2 - x1_3 - x1_4 - x1_5 - x1_6
                                          - x1_7 - x1_8 <= 0
```

Ebben a konkrét feladatban továbbra is egészértékű megoldást kapunk, de előfordulhat, hogy hasonló jellegű korlátok hozzáadása a modellhez nem eredményez egyészértékű megoldást. Ha ez lenne a helyzet, akkor az x változókra kik kell kötni, hogy egészértékűek lehetnek csak, ami kikötés adott esetben jelentősen megnövelheti a futási időt is.

7.8. példa megoldása.

Most a profitot kell maximalizálni, tehát a célfüggvénybe a vevők árajánlatai és a vállalat rezervációs árainak különbségei kerülhetnek, azonban csak olyan helyen lehet hozzárendelés, ahol ez az érték pozitív.

Az iroda bérlését két külön oszlop szerint kell rendezni, az egyik az adott szinten az A, a másik pedig a B lakás vásárlására vonatkozik. Ezekre fel kell tenni, hogy az adott szinten az egyik lakás megvétele esetén az emelet másik lakását is meg kell venni. Azaz a következő korlátokat kell a feladathoz hozzáadni: $xA_i - xB_i = 0$, ahol $i = 0, 1, 2$ index a szintet jelöli, A, és B pedig a lakást.

A vállalatnak lehetősége van arra, hogy az adott lakásokat ne adja el, ekkor mindenhol 2 millió forint profittal kell számolni és az erre vonatkozó hozzárendelések száma legfeljebb 7 lehet. Ugyanakkor be kell vezetni fiktív lakásokat, amikhez azok a vevők rendelődnek hozzá, akik végül nem vásárolnak lakást. Ezek a hozzárendelések száma legfeljebb 9 lehet. Ezen kívül, amennyiben nem adunk külön 7 oszlopot, és 9 sort a feladathoz, akkor a változókról külön fel kell tenni, hogy binárisak.

Tehát a végső mátrix, ami alapján a maximális hozzárendeléseket keressük:

lakás	l1	l2	l3	l4	l5	ü1	ü2	iroda(A)	iroda(B)	nincs üzlet
0.A	5	9				9	11	10		2
0.B	7	11				10	11		12	2
1.A	2	6	10	10				12		2
1.B	4	8	12	12					11	2
2.A		2	4	6	17			7		2
2.B		4	6	8	19				6	2
3.A				5	10					2
fiktív	0	0	0	0	0	0	0	0	0	0

10.44. kód.

```
Max
profit:
```

```
+ 5x1_1 + 9x1_2 + 9x1_6 +11x1_7 +10x1_8 + 2x1_10
+ 7x2_1 +11x2_2 +10x2_6 +11x2_7 +12x2_9 + 2x2_10
+ 2x3_1 + 6x3_2 +10x3_3 +10x3_4 +12x3_8 + 2x3_10
+ 4x4_1 + 8x4_2 +12x4_3 +12x4_4 +11x4_9 + 2x4_10
+ 2x5_2 + 4x5_3 + 6x5_4 +17x5_5 + 7x5_8 + 2x5_10
+ 4x6_2 + 6x6_3 + 8x6_4 +19x6_5 + 6x6_9 + 2x6_10
+ 5x7_4 +10x7_5 + 2x7_10
+ 0x8_1 + 0x8_2 + 0x8_3 + 0x8_4 + 0x8_5 + 0x8_6
+ 0x8_7 + 0x8_8 + 0x8_9 + 0x8_10

subject to
lak1:    +x1_1 +x1_2 +x1_6  +x1_7 +x1_8 +x1_10  = 1
lak2:    +x2_1 +x2_2 +x2_6  +x2_7 +x2_9 +x2_10  = 1
lak3:    +x3_1 +x3_2 +x3_3  +x3_4 +x3_8 +x3_10  = 1
lak4:    +x4_1 +x4_2 +x4_3  +x4_4 +x4_9 +x4_10  = 1
lak5_    +x5_2 +x5_3 +x5_4  +x5_5 +x5_8 +x5_10  = 1
lak6:    +x6_2 +x6_3 +x6_4  +x6_5 +x6_9 +x6_10  = 1
lak7:    +x7_4 +x7_5 +x7_10                     = 1
fiktiv8: +x8_1 +x8_2 +x8_3  +x8_4 +x8_5
               +x8_6 +x8_7  +x8_8 +x8_9 +x8_10 <= 9

lako1: +x1_1  +x2_1  +x3_1  +x4_1  +x8_1               = 1
lako2: +x1_2  +x2_2  +x3_2  +x4_2  +x5_2  +x6_2  +x8_2 = 1
lako3: +x3_3  +x4_3  +x5_3  +x6_3  +x8_3               = 1
lako4: +x3_4  +x4_4  +x5_4  +x6_4  +x7_4  +x8_4        = 1
lako5: +x5_5  +x6_5  +x7_5  +x8_5                      = 1
uzlet1:+x1_6  +x2_6  +x8_6                             = 1
uzlet2:+x1_7  +x2_7  +x8_7                             = 1
irodaA:+x1_8  +x3_8  +x5_8  +x8_8                      = 1
irodaB:+x2_9  +x4_9  +x6_9  +x8_9                      = 1
n:     +x1_10 +x2_10 +x3_10 +x4_10
                     +x5_10  +x6_10 +x7_10 +x8_10     <= 7

iroda_sz0: +x1_8 -x2_9 = 0
iroda_sz1: +x3_8 -x4_9 = 0
iroda_sz2: +x5_8 -x6_9 = 0

Binary
x1_1
x2_1
x3_1
x4_1
x8_1
x1_2
x2_2
```

```
x3_2
x4_2
x5_2
x6_2
x8_2
x3_3
x4_3
x5_3
x6_3
x8_3
x3_4
x4_4
x5_4
x6_4
x7_4
x8_4
x5_5
x6_5
x7_5
x8_5
x1_6
x2_6
x8_6
x1_7
x2_7
x8_7
x1_8
x3_8
x5_8
x8_8
x2_9
x4_9
x6_9
x8_9
x1_10
x2_10
x3_10
x4_10
x5_10
x6_10
x7_10
x8_10

end
```

Az optimális megoldás:

```
   No. Column name       Activity
------ ------------    -------------
     4 x1_7         *              1
     8 x2_2         *              1
    17 x3_8         *              1
    23 x4_9         *              1
    26 x5_3         *              1
    34 x6_5         *              1
    37 x7_4         *              1
    40 x8_1         *              1
    45 x8_6         *              1
```

Tehát a földszint A lakásába a 2. üzlet költözik, a B lakásba pedig a 2. lakó. Az 1. emeleten lesz az iroda, a 2. emelet A lakásába a 3. lakó költözik, míg a B-be az ötödik, a 3. emeletre a 4. lakó. Az 1. lakó, és az 1. üzletnek nem jut a társasházban lakás.

7.9. példa megoldása.

A következő bináris változókat használjuk (i =1, ..., 9), ami akkor 1, ha:

x1_i	Az i. munkás az A terméket gyártja.
x2_i	Az i. munkás a B terméket gyártja.
x3_i	Az i. munkás a C terméket gyártja.
x4_i	Az i. munkás az első futószalag mellett az első szakaszt végzi.
x5_i	Az i. munkás az első futószalag mellett a második szakaszt végzi.
x6_i	Az i. munkás az első futószalag mellett a harmadik szakaszt végzi.
x7_i	Az i. munkás a második futószalag mellett az első szakaszt végzi.
x8_i	Az i. munkás a második futószalag mellett a második szakaszt végzi.
x9_i	Az i. munkása második futószalag mellett a harmadik szakaszt végzi.

A célfüggvényben csak az A, B, C termékekhez vonatkozó korlátok kerülnek. Az eredeti hozzárendelési feladathoz képest még olyan korlátokat kell hozzáadni a feladathoz, ami biztosítja, hogy mind a két futószalag mellett olyan munkások dolgozzanak, akik eddig összesen 300 alkatrészt gyártottak le, tehát: A célfüggvényben csak az A, B, C termékekhez vonatkozó korlátok kerülnek. Az eredeti hozzárendelési feladathoz képest még olyan korlátokat kell hozzáadni a feladathoz, ami biztosítja, hogy mind a két futószalag mellett olyan munkások dolgozzanak, akik eddig összesen 300 alkatrészt gyártottak le, tehát:

```
om1: +117x4_1 +121x4_2 + 95x4_3 + 75x4_7
+117x5_1 + 73x5_4 +140x5_5 +110x5_6 +180x5_8
+140x6_5 +110x6_6 + 75x6_7 +180x6_8 +130x6_9 >= 300
```

Ugyanezt a második futószalagra is fel kell tenni. A kódban ezeket a korlátokat `om1:`, illetve `om2:` névvel jelöltük.

10.45. kód.

```
Max
pont:
+36x1_1 +61x1_2 +15x1_3 +28x1_4 +36x1_5
+13x1_6 +98x1_7 +53x1_8 +79x1_9
+97x2_1 +96x2_2 +85x2_3 +30x2_4 +85x2_5
+12x2_6 +49x2_7 +69x2_8 +41x2_9
+69x3_1 +66x3_2 +85x3_3 +19x3_4 +66x3_5
+11x3_6 +18x3_7 +68x3_8 +28x3_9

subject to
A1:  +x1_1+x1_2+x1_3+x1_4+x1_5+x1_6+x1_7+x1_8+x1_9 = 1
B2:  +x2_1+x2_2+x2_3+x2_4+x2_5+x2_6+x2_7+x2_8+x2_9 = 1
C3:  +x3_1+x3_2+x3_3+x3_4+x3_5+x3_6+x3_7+x3_8+x3_9 = 1
f1_4:+x4_1+x4_2+x4_3+x4_7                          = 1
f1_5:+x5_1+x5_4+x5_5+x5_6+x5_8                     = 1
f1_6:+x6_5+x6_6+x6_7+x6_8+x6_9                     = 1
f2_7:+x7_1+x7_2+x7_3+x7_7                          = 1
f2_8:+x8_1+x8_4+x8_5+x8_6+x8_8                     = 1
f2_9:+x9_5+x9_6+x9_7+x9_8+x9_9                     = 1

m1: +x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x7_1 +x8_1 = 1
m2: +x1_2 +x2_2 +x3_2 +x4_2 +x7_2             = 1
m3: +x1_3 +x2_3 +x3_3 +x4_3 +x7_3             = 1
m4: +x1_4 +x2_4 +x3_4 +x5_4 +x8_4             = 1
m5: +x1_5 +x2_5 +x3_5 +x5_5 +x6_5 +x8_5 +x9_5 = 1
m6: +x1_6 +x2_6 +x3_6 +x5_6 +x6_6 +x8_6 +x9_6 = 1
m7: +x1_7 +x2_7 +x3_7 +x4_7 +x6_7 +x7_7 +x9_7 = 1
m8: +x1_8 +x2_8 +x3_8 +x5_8 +x6_8 +x8_8 +x9_8 = 1
m9: +x1_9 +x2_9 +x3_9 +x6_9 +x9_9             = 1

om1:+117x4_1 +121x4_2 + 95x4_3 + 75x4_7
+117x5_1 + 73x5_4 +140x5_5 +110x5_6 +180x5_8
+140x6_5 +110x6_6 + 75x6_7 +180x6_8 +130x6_9 >= 300
```

```
om2:+117x7_1 +121x7_2 + 95x7_3 + 75x7_7
+117x8_1 + 73x8_4 +140x8_5 +110x8_6 +180x8_8
+140x9_5 +110x9_6 + 75x9_7 +180x9_8 +130x9_9 >= 300

Binary
x1_1
x1_2
x1_3
x1_4
x1_5
x1_6
x1_7
x1_8
x1_9
x2_1
x2_2
x2_3
x2_4
x2_5
x2_6
x2_7
x2_8
x2_9
x3_1
x3_2
x3_3
x3_4
x3_5
x3_6
x3_7
x3_8
x3_9
x4_1
x4_2
x4_3
x4_4
x4_5
x4_6
x4_7
x4_8
x4_9
x5_1
x5_2
x5_3
x5_4
```

```
x5_5
x5_6
x5_7
x5_8
x5_9
x6_1
x6_2
x6_3
x6_4
x6_5
x6_6
x6_7
x6_8
x6_9
x7_1
x7_2
x7_3
x7_4
x7_5
x7_6
x7_7
x7_8
x7_9
x8_1
x8_2
x8_3
x8_4
x8_5
x8_6
x8_7
x8_8
x8_9
x9_1
x9_2
x9_3
x9_4
x9_5
x9_6
x9_7
x9_8
x9_9

End
```

Az optimális megoldás:

```
No.    Row name        Activity
-- ------------    -----------
19 om1                      427
20 om2                      304

No. Column name        Activity
-- ------------    -------------
 7 x1_7         *              1
14 x2_5         *              1
21 x3_3         *              1
28 x4_1         *              1
36 x5_8         *              1
41 x6_9         *              1
43 x7_2         *              1
47 x8_4         *              1
52 x9_6         *              1
```

Tehát a végeredmény:

feladat	munkás
A	m7
B	m5
C	m3
futószalag1: 1. szakasz	m1
futószalag1: 2. szakasz	m8
futószalag1: 3. szakasz	m9
futószalag2: 1. szakasz	m2
futószalag2: 2. szakasz	m4
futószalag2: 3. szakasz	m6

Az A, B, C termék összeszereléséhez az összesített pontszám pedig 268.

10.7. Stabil párosítás feladatok

8.8. példa megoldása.

Mivel ebben az esetben 6 férfi és 6 nő között keressük a stabil párosításokat, és a szereplők csak az első 5 legszimpatikusabbat rangsorolták, ezért az LP felírás a 8.7. kódhoz hasonló:

10.46. kód.

```
Min
```

```
ffi:
+3x1_1 +1x1_2        +2x1_4 +4x1_5 +5x1_6
              +3x2_3 +1x2_4 +2x2_5 +4x2_6
+4x3_1 +5x3_2        +2x3_4 +1x3_5 +3x3_6
       +5x4_2 +1x4_3 +2x4_4 +3x4_5 +4x4_6
              +3x5_3 +4x5_4        +1x5_6
+5x6_1 +3x6_2               +4x6_5

subject to

+x1_1 +x1_2 +x1_4 +x1_5 +x1_6 = 1
+x2_3 +x2_4 +x2_5 +x2_6       = 1
+x3_1 +x3_2 +x3_4 +x3_5 +x3_6 = 1
+x4_2 +x4_3 +x4_4 +x4_5 +x4_6 = 1
+x5_3 +x5_4 +x5_6             = 1
+x6_1 +x6_2 +x6_5             = 1

+x1_1 +x3_1 +x6_1             =1
+x1_2 +x3_2 +x4_2 +x6_2       =1
+x2_3 +x4_3 +x5_3             =1
+x1_4 +x2_4 +x3_4 +x4_4 +x5_4 =1
+x1_5 +x2_5 +x3_5 +x4_5 +x6_5 =1
+x1_6 +x2_6 +x3_6 +x4_6 +x5_6 =1

x1_1     +x1_2     +x1_4
                                                        >=1
x1_2
                                         +x4_2          >=1
x1_4     +x1_2
                                                        >=1
x1_5+x1_1+x1_2     +x1_4
                                                        >=1
x1_6+x1_1+x1_2     +x1_4+x1_5
                               +x2_6+x3_6+x4_6+x5_6     >=1
x2_3               +x2_4+x2_5
                                              +x5_3     >=1
x2_4
                          +x1_4     +x3_4+x4_4+x5_4     >=1
x2_5               +x2_4
                          +x1_5     +x3_5               >=1
x2_6          +x2_3+x2_4+x2_5
                                                        >=1
x3_1               +x3_4+x3_5+x3_6
                          +x1_1                         >=1
x3_2+x3_1          +x3_4+x3_5+x3_6
```

```
                          +x1_2           +x4_2           >=1
x3_4                    +x3_5
                          +x1_4           +x4_4+x5_4      >=1
x3_5
                          +x1_5                           >=1
x3_6               +x3_4+x3_5
                               +x2_6+      x4_6           >=1
x4_2          +x4_3+x4_4+x4_5+x4_6
                                                          >=1
x4_3
                               +x2_3           +x5_3      >=1
x4_4          +x4_3
                          +x1_4                           >=1
x4_5          +x4_3+x4_4
                          +x1_5+x2_5+x3_5          +x6_5>=1
x4_6          +x4_3+x4_4+x4_5
                               +x2_6                      >=1
x5_3                         +x5_6
                                                          >=1
x5_4          +x5_3          +x5_6
                          +x1_4           +x4_4           >=1
x5_6
                               +x2_6+x3_6+x4_6            >=1
x6_1     +x6_2          +x6_5
                          +x1_1     +x3_1                 >=1
x6_2
                          +x1_2     +x3_2+x4_2            >=1
x6_5     +x6_2
                          +x1_5+x2_5+x3_5                 >=1
end
```

Az optimális megoldás pedig:

```
   No. Column name      St    Activity
------ ------------     --  -------------
     2 x1_2             B               1
     7 x2_4             B               1
    13 x3_5             B               1
    16 x4_3             B               1
    22 x5_6             B               1
    23 x6_1             B               1
```

A mátrixokba bejelölve:

$$
\begin{pmatrix}
3 & \boxed{1} & & 2 & 4 & 5 \\
5 & & 3 & \boxed{1} & 2 & 4 \\
4 & 5 & & 2 & \boxed{1} & 3 \\
& 5 & \boxed{1} & 2 & 3 & 4 \\
& 2 & 3 & 4 & 5 & \boxed{1} \\
\boxed{5} & 3 & 2 & & 4 & 1
\end{pmatrix}
\quad \text{és} \quad
\begin{pmatrix}
2 & \boxed{3} & 3 & 1 & 1 & 5 \\
& 2 & 4 & \boxed{5} & 3 & 1 \\
3 & 4 & 2 & 4 & \boxed{2} & 3 \\
1 & 1 & \boxed{5} & 2 & 5 & 2 \\
5 & & 1 & 3 & & \boxed{4} \\
\boxed{4} & 5 & & & 4 &
\end{pmatrix}
$$

A nő-optimális esetben a célfüggvény:

```
no:
+2x1_1+3x1_2      +1x1_4+1x1_5+5x1_6
            +4x2_3+5x2_4+3x2_5+1x2_6
+3x3_1+4x3_2      +4x3_4+2x3_5+3x3_6
      +1x4_2+5x4_3+2x4_4+5x4_5+2x4_6
            +1x5_3+3x5_4      +4x5_6
+4x6_1+5x6_2            +4x6_5
```

Az optimális megoldás pedig:

```
   No. Column name      St    Activity
------ ------------     --  -------------
     3 x1_4             B               1
     9 x2_6             B               1
    13 x3_5             B               1
    15 x4_2             B               1
    20 x5_3             B               1
    23 x6_1             B               1
```

A mátrixokba bejelölve:

$$
\begin{pmatrix}
3 & 1 & & \boxed{2} & 4 & 5 \\
5 & & 3 & 1 & 2 & \boxed{4} \\
4 & 5 & & 2 & \boxed{1} & 3 \\
& \boxed{5} & 1 & 2 & 3 & 4 \\
& 2 & \boxed{3} & 4 & 5 & 1 \\
\boxed{5} & 3 & 2 & & 4 & 1
\end{pmatrix}
\quad \text{és} \quad
\begin{pmatrix}
2 & 3 & 3 & \boxed{1} & 1 & 5 \\
& 2 & 4 & 5 & 3 & \boxed{1} \\
3 & 4 & 2 & 4 & \boxed{2} & 3 \\
1 & \boxed{1} & 5 & 2 & 5 & 2 \\
5 & & \boxed{1} & 3 & & 4 \\
\boxed{4} & 5 & & & 4 &
\end{pmatrix}
$$

8.9. példa megoldása.

Mivel ebben az esetben több férfi van, mint nő, ezért az LP felírásban bevezetünk minden férfira egy `f` fiktív bináris változót. Amennyiben `f_i=1`, akkor az `i`-edik férfi egyedül marad. Ezeket a változókat a hozzárendelési korlátokban vesszük figyelembe. A stabilitási korlátokat nem kell külön ezekre az esetekre felírni, mivel a bármilyen más hozzárendelést jobban értékelnek, mint ezt.

A kód, a férfi-optimális megoldáshoz:

10.47. kód.

```
Min
ffi:
+4x1_1 +3x1_2 +2x1_3 +5x1_4 +1x1_5
+3x2_1 +4x2_2 +2x2_3 +5x2_4 +1x2_5
+4x3_1 +5x3_2 +1x3_3 +3x3_4 +2x3_5
+3x4_1 +1x4_2 +4x4_3 +5x4_4 +2x4_5
+3x5_1 +2x5_2 +4x5_3 +1x5_4 +5x5_5
+4x6_1 +3x6_2 +2x6_3 +1x6_4 +5x6_5
+2x7_1 +1x7_2 +4x7_3 +3x7_4 +5x7_5
+3x8_1 +5x8_2 +1x8_3 +2x8_4 +4x8_5

subject to

+x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +f_1 = 1
+x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +f_2 = 1
+x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +f_3 = 1
+x4_1 +x4_2 +x4_3 +x4_4 +x4_5 +f_4 = 1
+x5_1 +x5_2 +x5_3 +x5_4 +x5_5 +f_5 = 1
+x6_1 +x6_2 +x6_3 +x6_4 +x6_5 +f_6 = 1
+x7_1 +x7_2 +x7_3 +x7_4 +x7_5 +f_7 = 1
+x8_1 +x8_2 +x8_3 +x8_4 +x8_5 +f_8 = 1

+x1_1 +x2_1 +x3_1 +x4_1 +x5_1 +x6_1 +x7_1 +x8_1 = 1
+x1_2 +x2_2 +x3_2 +x4_2 +x5_2 +x6_2 +x7_2 +x8_2 = 1
+x1_3 +x2_3 +x3_3 +x4_3 +x5_3 +x6_3 +x7_3 +x8_3 = 1
+x1_4 +x2_4 +x3_4 +x4_4 +x5_4 +x6_4 +x7_4 +x8_4 = 1
+x1_5 +x2_5 +x3_5 +x4_5 +x5_5 +x6_5 +x7_5 +x8_5 = 1

x1_1     +x1_2+x1_3     +x1_5
         +x2_1+x3_1+x4_1+x5_1+x6_1           >= 1
x1_2          +x1_3     +x1_5
                        +x5_2+x6_2+x7_2+x8_2 >= 1
x1_3                    +x1_5
                        +x5_3                >= 1
```

```
x1_4+x1_1+x1_2+x1_3     +x1_5                     >= 1
x1_5                                              >= 1
x2_1          +x2_3     +x2_5
              +x3_1     +x5_1                     >= 1
x2_2+x2_1     +x2_3     +x2_5
    +x1_2               +x5_2+x6_2+x7_2+x8_2 >= 1
x2_3     +x2_5
    +x1_3     +x3_3     +x5_3          +x8_3 >= 1
x2_4+x2_1+x2_2+x2_3     +x2_5
    +x1_4          +x4_4+x5_4+x6_4+x7_4+x8_4 >= 1
x2_5
    +x1_5     +x3_5     +x5_5          +x8_5 >= 1
x3_1          +x3_3+x3_4+x3_5                     >= 1
x3_2+x3_1     +x3_3+x3_4+x3_5
    +x1_2+x2_2          +x5_2+x6_2+x7_2+x8_2 >= 1
x3_3
    +x1_3               +x5_3          +x8_3 >= 1
x3_4          +x3_3     +x3_5
    +x1_4+x2_4     +x4_4+x5_4+x6_4+x7_4+x8_4 >= 1
x3_5          +x3_3
    +x1_5                                    >= 1
x4_1     +x4_2          +x4_5
         +x2_1+x3_1     +x5_1                     >= 1
x4_2     +x1_2
         +x2_2+x3_2     +x5_2+x6_2+x7_2+x8_2 >= 1
x4_3+x4_1+x4_2          +x4_5
    +x1_3+x2_3+x3_3     +x5_3          +x8_3 >= 1
x4_4+x4_1+x4_2+x4_3     +x4_5
    +x1_4               +x5_4+x6_4+x7_4+x8_4 >= 1
x4_5     +x4_2
    +x1_5+x2_5+x3_5     +x5_5          +x8_5 >= 1
x5_1     +x5_2     +x5_4
              +x3_1                               >= 1
x5_2               +x5_4
                             +x6_2+x7_2      >= 1
x5_3+x5_1+x5_2     +x5_4                          >= 1
x5_4
    +x1_4                                    >= 1
x5_5+x5_1+x5_2+x5_3+x5_4
    +x1_5     +x3_5                    +x8_5 >= 1
x6_1     +x6_2+x6_3+x6_4
         +x2_1+x3_1+x4_1+x5_1                     >= 1
x6_2          +x6_3+x6_4                          >= 1
x6_3               +x6_4
    +x1_3+x2_3+x3_3+x4_3+x5_3          +x8_3 >= 1
```

```
x6_4
    +x1_4               +x5_4     +x7_4      >= 1
x6_5+x6_1+x6_2+x6_3+x6_4
    +x1_5+x2_5+x3_5+x4_5+x5_5     +x7_5+x8_5 >= 1
x7_1     +x7_2
    +x1_1+x2_1+x3_1+x4_1+x5_1+x6_1           >= 1
x7_2
                         +x6_2               >= 1
x7_3+x7_1+x7_2     +x7_4
    +x1_3+x2_3+x3_3+x4_3+x5_3+x6_3      +x8_3 >= 1
x7_4+x7_1+x7_2
    +x1_4               +x5_4                >= 1
x7_5+x7_1+x7_2+x7_3+x7_4
    +x1_5+x2_5+x3_5+x4_5+x5_5           +x8_5 >= 1
x8_1          +x8_3+x8_4
    +x1_1+x2_1+x3_1+x4_1+x5_1+x6_1+x7_1      >= 1
x8_2+x8_1     +x8_3+x8_4+x8_5
                    +x5_2+x6_2+x7_2      >= 1
x8_3
    +x1_3               +x5_3                >= 1
x8_4          +x8_3
    +x1_4               +x5_4+x6_4+x7_4      >= 1
x8_5+x8_1     +x8_3+x8_4
    +x1_5     +x3_5                          >= 1

end
```

Az optmiális megoldás pedig:

```
   No. Column name      St    Activity
------ ------------     --  -------------
     5 x1_5             B               1
    11 x3_1             B               1
    24 x5_4             B               1
    27 x6_2             B               1
    38 x8_3             B               1
    42 f_2              B               1
    44 f_4              B               1
    47 f_7              B               1
```

A mátrixokba bejelölve:

$$\begin{pmatrix} 4 & 3 & 2 & 5 & \boxed{1} \\ 3 & 4 & 2 & 5 & 1 \\ \boxed{4} & 5 & 1 & 3 & 2 \\ 3 & 1 & 4 & 5 & 2 \\ 3 & 2 & 4 & \boxed{1} & 5 \\ 4 & \boxed{3} & 2 & 1 & 5 \\ 2 & 1 & 4 & 3 & 5 \\ 3 & 5 & \boxed{1} & 2 & 4 \end{pmatrix} \quad \text{és} \quad \begin{pmatrix} 6 & 5 & 2 & 1 & \boxed{1} \\ 3 & 6 & 5 & 7 & 5 \\ \boxed{1} & 7 & 4 & 8 & 2 \\ 4 & 8 & 6 & 6 & 6 \\ 2 & 3 & 1 & \boxed{2} & 4 \\ 5 & \boxed{1} & 7 & 4 & 8 \\ 7 & 2 & 8 & 3 & 7 \\ 8 & 4 & \boxed{3} & 5 & 3 \end{pmatrix}$$

A nő-optimális megoldásban megint egyedül csak a célfüggvény változik:

```
no:
+6x1_1 +5x1_2 +2x1_3 +1x1_4 +1x1_5
+3x2_1 +6x2_2 +5x2_3 +7x2_4 +5x2_5
+1x3_1 +7x3_2 +4x3_3 +8x3_4 +2x3_5
+4x4_1 +8x4_2 +6x4_3 +6x4_4 +6x4_5
+2x5_1 +3x5_2 +1x5_3 +2x5_4 +4x5_5
+5x6_1 +1x6_2 +7x6_3 +4x6_4 +8x6_5
+7x7_1 +2x7_2 +8x7_3 +3x7_4 +7x7_5
+8x8_1 +4x8_2 +3x8_3 +5x8_4 +3x8_5
```

Ezen célfüggvény esetén a hozzárendelések megegyeznek a férfi-optimálissal, tehát ebben a feladatban kizárólag ez az egy stabil páros létezik. Végezetül: a 2-es, a 4-es és 7-es férfi marad pár nélkül.

8.10. példa megoldása.

Ebben az esetben az `xi_j` változásoknál az első index a szakot, a második pedig a diákot jelöli. A szak esetében most nem csak egy párt keresünk, ezért a hozzárendelési korlát változik: szakoknént legfeljebb 4 diákot tudunk felvenni.

A feladatban probléma még, hogy a diákok pontszámainál a maximális érték a legjobb, míg a diákok értékeléseinél a minimális. Ennek a problémának a legegyszerűbb megoldása, ha a pontszámok helyett a maximálisan elérhető 500 ponthoz szükséges pontokat vesszük figyelembe:

$$\begin{pmatrix} 109 & 125 & 23 & 121 & 14 & 24 & 46 & 19 & 14 \\ 148 & 93 & 10 & 73 & 129 & 30 & 76 & 65 & 27 \\ 107 & 149 & 145 & 126 & 109 & 18 & 44 & 60 & 69 \end{pmatrix}$$

Ebben az esetben minél kisebb az érték, a szak számára annál értékesebb a hallgató. Tehát ezekkel a változtatásokkal ugyanúgy meghatározhatjuk az LP-t, mint korábban. A szak-optimális megoldás:

10.48. kód.

```
Min
+109x1_1 +125x1_2 + 23x1_3 +121x1_4 + 14x1_5
+ 24x1_6 + 46x1_7 + 19x1_8 + 14x1_9
+148x2_1 + 93x2_2 + 10x2_3 + 73x2_4 +129x2_5
+ 30x2_6 + 76x2_7 + 65x2_8 + 27x2_9
+107x3_1 +149x3_2 +145x3_3 +126x3_4 +109x3_5
+ 18x3_6 + 44x3_7 + 60x3_8 + 69x3_9

subject to

+x1_1 +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_7 +x1_8 +x1_9 <= 4
+x2_1 +x2_2 +x2_3 +x2_4 +x2_5 +x2_6 +x2_7 +x2_8 +x2_9 <= 4
+x3_1 +x3_2 +x3_3 +x3_4 +x3_5 +x3_6 +x3_7 +x3_8 +x3_9 <= 4

+x1_1 +x2_1 +x3_1 = 1
+x1_2 +x2_2 +x3_2 = 1
+x1_3 +x2_3 +x3_3 = 1
+x1_4 +x2_4 +x3_4 = 1
+x1_5 +x2_5 +x3_5 = 1
+x1_6 +x2_6 +x3_6 = 1
+x1_7 +x2_7 +x3_7 = 1
+x1_8 +x2_8 +x3_8 = 1
+x1_9 +x2_9 +x3_9 = 1

x1_1          +x1_3      +x1_5+x1_6+x1_7+x1_8+x1_9
                                             +x2_1      >=1
x1_2+x1_1     +x1_3+x1_4+x1_5+x1_6+x1_7+x1_8+x1_9
                                                  +x3_2>=1
x1_3                     +x1_5          +x1_8+x1_9
                                             +x2_3      >=1
x1_4+x1_1     +x1_3      +x1_5+x1_6+x1_7+x1_8+x1_9
                                                        >=1
x1_5
                                             +x2_5      >=1
```

```
x1_6           +x1_3     +x1_5           +x1_8+x1_9
                                               +x2_6+x3_6>=1
x1_7           +x1_3     +x1_5+x1_6      +x1_8+x1_9
                                                         >=1
x1_8                     +x1_5                +x1_9
                                               +x2_8+x3_8>=1
x1_9
                                               +x2_9+x3_9>=1
x2_1      +x2_2+x2_3+x2_4+x2_5+x2_6+x2_7+x2_8+x2_9
                                                         >=1
x2_2           +x2_3+x2_4      +x2_6+x2_7+x2_8+x2_9
                                          +x1_2      +x3_2>=1
x2_3
                                                         >=1
x2_4           +x2_3           +x2_6     +x2_8+x2_9
                                          +x1_4      +x3_4>=1
x2_5      +x2_2+x2_3+x2_4      +x2_6+x2_7+x2_8+x2_9
                                                         >=1
x2_6           +x2_3                          +x2_9
                                                         >=1
x2_7           +x2_3+x2_4      +x2_6     +x2_8+x2_9
                                         +x1_7           >=1
x2_8           +x2_3           +x2_6          +x2_9
                                                         >=1
x2_9           +x2_3
                                                     +x3_9>=1
x3_1                           +x3_6+x3_7+x3_8+x3_9
                                          +x1_1+x2_1     >=1
x3_2+x3_1      +x3_3+x3_4+x3_5+x3_6+x3_7+x3_8+x3_9
                                                         >=1
x3_3+x3_1           +x3_4+x3_5+x3_6+x3_7+x3_8+x3_9
                                          +x1_3+x2_3     >=1
x3_4+x3_1                +x3_5+x3_6+x3_7+x3_8+x3_9
                                          +x1_4          >=1
x3_5+x3_1                      +x3_6+x3_7+x3_8+x3_9
                                          +x1_5+x2_5     >=1
x3_6
                                               +x2_6     >=1
x3_7                           +x3_6
                                          +x1_7+x2_7     >=1
x3_8                           +x3_6+x3_7
                                               +x2_8     >=1
x3_9                           +x3_6+x3_7+x3_8
                                                         >=1
end
```

Az optimális megoldás pedig:

```
   No. Column name      St    Activity
------ ------------     --  -------------
     5 x1_5             B              1
     8 x1_8             B              1
     9 x1_9             B              1
    11 x2_2             B              1
    12 x2_3             B              1
    13 x2_4             B              1
    19 x3_1             B              1
    24 x3_6             B              1
    25 x3_7             B              1
```

A diák-optimális megoldásban egyedül a célfüggvény változik:

```
+2x1_1 +2x1_2 +2x1_3 +1x1_4 +2x1_5
              +3x1_6+ 1x1_7 +3x1_8 +3x1_9
+1x2_1 +3x2_2 +1x2_3 +3x2_4 +1x2_5
              +1x2_6 +2x2_7 +1x2_8 +2x2_9
+3x3_1 +1x3_2 +3x3_3 +2x3_4 +3x3_5
              +2x3_6 +3x3_7 +2x3_8 +1x3_9
```

Aminek a megoldása pedig:

```
   No. Column name      St    Activity
------ ------------     --  -------------
     4 x1_4             B              1
     5 x1_5             B              1
     7 x1_7             B              1
    10 x2_1             B              1
    12 x2_3             B              1
    15 x2_6             B              1
    17 x2_8             B              1
    20 x3_2             B              1
    27 x3_9             B              1
```

Tehát a szak- illetve diák-optimális megoldásokban a következő jelentkezőket vették fel:

szak	diák (szak-optimális)	diák (diák-optimális)
1	5,8,9	4,5,7
2	2,3,4	1,3,6,8
3	1,6,7	2,9

8.11. példa megoldása.

A következő esetek azok, amiket ki kell zárnunk, mivel a két értékelés több mint 3: `x1_6`, `x3_2`, `6_1`, `6_5`. Ezeket a változókat töröljük ki a preferencia-mátrixból:

$$\begin{pmatrix} 3 & 1 & & 2 & 4 & \\ 5 & & 3 & 1 & 2 & 4 \\ 4 & & & 2 & 1 & 3 \\ & 5 & 1 & 2 & 3 & 4 \\ & 2 & 3 & 4 & 5 & 1 \\ & 3 & 2 & & & 1 \end{pmatrix} \quad \text{és} \quad \begin{pmatrix} 2 & 3 & 3 & 1 & 1 & \\ & 2 & 4 & 5 & 3 & 1 \\ 3 & & 2 & 4 & 2 & 3 \\ 1 & 1 & 5 & 2 & 5 & 2 \\ 5 & & 1 & 3 & & 4 \\ & 5 & & & & \end{pmatrix}$$

Természetesen ezen cellákhoz tartozó stabilitási korlátokat is el kell hagynia a 10.46 kódból.

A célfüggvény két részből áll, egyrészt azok a kifizetések, amikor az egyik fél értékelése nagyobb, mint 3:

```
                            +50x1_5
              +50x2_3+50x2_4       +50x2_6
+50x3_1              +50x3_4
       +50x4_2+50x4_3       +50x4_5+50x4_6
                     +50x5_4       +50x5_6
       +50x6_2
```

illetve, amikor valakit nem tudunk mással összepárosítani. Az első férfit akkor nem tudjuk mással párosítani, ha `x1_1`, `x1_2`, `x1_4` és `x1_5` változók mindegyike 0. Másképpen megfogalmazva a `1-(x1_1+x1_2+x1_4+x1_5)` kifejezés 1, lesz, ha az első férinek nem jut pár, és 0 lesz, ha jut neki. Ezek után a költséget már fel tudjuk írni: `100(1-(x1_1+x1_2+x1_4+x1_5))`, amely kifejezés kibontva: `100-100x1_1-100x1_2-100x1_4-100x1_5`. Minden férfire és nőre elvégezve ezt a műveletet:

```
1200
     -200x1_1 -200x1_2                   -200x1_4 -200x1_5
                        -200x2_3 -200x2_4 -200x2_5 -200x2_6
     -200x3_1                   -200x3_4 -200x3_5 -200x3_6
              -200x4_2 -200x4_3 -200x4_4 -200x4_5 -200x4_6
                        -200x5_3 -200x5_4          -200x5_6
              -200x6_2
```

kifejezéshez jutunk. A kifejezésben az első tag (`1200`) konstans, így az optimális hozzárendelést nem befolyásolja, ezért elhagyjuk a célfüggvényből. Adjuk össze a kétféle költséget, és lényegileg kész is a kód:

10.49. kód.

```
Min
     -200x1_1 -200x1_2          -200x1_4 -150x1_5
                       -150x2_3 -150x2_4 -200x2_5 -150x2_6
     -150x3_1                   -150x3_4 -200x3_5 -200x3_6
              -150x4_2 -150x4_3 -200x4_4 -150x4_5 -150x4_6
                       -200x5_3 -150x5_4          -150x5_6
              -150x6_2

subject to

+x1_1 +x1_2 +x1_4 +x1_5       <= 1
+x2_3 +x2_4 +x2_5 +x2_6       <= 1
+x3_1       +x3_4 +x3_5 +x3_6 <= 1
+x4_2 +x4_3 +x4_4 +x4_5 +x4_6 <= 1
+x5_3 +x5_4 +x5_6             <= 1
      +x6_2                   <= 1

+x1_1 +x3_1                   <= 1
+x1_2       +x4_2 +x6_2       <= 1
+x2_3 +x4_3 +x5_3             <= 1
+x1_4 +x2_4 +x3_4 +x4_4 +x5_4 <= 1
+x1_5 +x2_5 +x3_5 +x4_5       <= 1
      +x2_6 +x3_6 +x4_6 +x5_6 <= 1

x1_1    +x1_2    +x1_4
                                                      >=1
x1_2
                                        +x4_2         >=1
x1_4    +x1_2
                                                      >=1
x1_5+x1_1+x1_2   +x1_4
                                                      >=1
x2_3             +x2_4+x2_5
                                             +x5_3    >=1
x2_4
                        +x1_4     +x3_4+x4_4+x5_4     >=1
x2_5             +x2_4
                        +x1_5     +x3_5               >=1
x2_6         +x2_3+x2_4+x2_5
                                                      >=1
```

```
x3_1                  +x3_4+x3_5+x3_6
                             +x1_1                        >=1
x3_4                       +x3_5
                             +x1_4          +x4_4+x5_4    >=1
x3_5
                             +x1_5                        >=1
x3_6                  +x3_4+x3_5
                                  +x2_6+     x4_6         >=1
x4_2             +x4_3+x4_4+x4_5+x4_6
                                                          >=1
x4_3
                                  +x2_3          +x5_3    >=1
x4_4             +x4_3
                             +x1_4                        >=1
x4_5             +x4_3+x4_4
                             +x1_5+x2_5+x3_5              >=1
x4_6             +x4_3+x4_4+x4_5
                                  +x2_6                   >=1
x5_3                            +x5_6
                                                          >=1
x5_4             +x5_3          +x5_6
                             +x1_4          +x4_4         >=1
x5_6
                                  +x2_6+x3_6+x4_6         >=1
x6_2
                             +x1_2          +x4_2         >=1
end
```

Az optimális megoldás:

```
   No. Column name   St     Activity
------ ------------  --  -------------
     2 x1_2          B               1
     8 x2_6          B               1
    11 x3_5          B               1
    15 x4_4          B               1
    18 x5_3          B               1
```

A célfüggvény értéke -950, de ne felejtsük el, hogy ehhez 1200-t hozzá kell adni: -950+1200=250. Ez az érték már teljesen jól magyarázható: a hatodik férfi, és az első nő marad pár nélkül, aminek költsége 200, ezen felül a második férfinek kell fizetni 50 pénzegységet, mivel ő a hatodik nőt kapja, aki a listájában a negyedik helyen szerepel.

$$\begin{pmatrix} 3 & \boxed{1} & & 2 & 4 & 5 \\ 5 & & 3 & 1 & 2 & \boxed{4} \\ 4 & 5 & & 2 & \boxed{1} & 3 \\ & 5 & 1 & \boxed{2} & 3 & 4 \\ & 2 & \boxed{3} & 4 & 5 & 1 \\ 5 & 3 & 2 & & 4 & 1 \end{pmatrix} \quad \text{és} \quad \begin{pmatrix} 2 & \boxed{3} & 3 & 1 & 1 & 5 \\ & 2 & 4 & 5 & 3 & \boxed{1} \\ 3 & 4 & 2 & 4 & \boxed{2} & 3 \\ 1 & 1 & 5 & \boxed{2} & 5 & 2 \\ 5 & & \boxed{1} & 3 & & 4 \\ 4 & 5 & & & 4 & \end{pmatrix}$$

8.12. példa megoldása.

Ebben az esetben csak egy típusú szereplője van a feladatnak, viszont ugyanúgy meghatározható a stabil párosítás, mint azokban az esetekben, amikor két típus is volt. Jelölje $x_{ij} = 1$ változó megint a hozzárendelést, az i-edik hallgató, és a j-edik hallgató között. Ebben az esetben $x_{ij} = x_{ji}$, tehát a kódban a következő korlátokat kell a feladathoz adni:

```
xi_j-xj_i = 0
```

A feladatban hozzárendelési korlátok nem változnak:

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

Azonban a stabilitási korlátok ebben az esetben módosulnak. Legyen t_j^i a j-edik hallgató rangsorszáma a i-edik hallgató rangsorában. Ekkor a stabilitási korlátok a következőképpen néznek ki:

$$x_{ij} + \sum_{k:t_k^i < t_j^i}^{n} x_{ik} + \sum_{l:t_l^j < t_i^j}^{n} x_{jl} \geq 1 \qquad \forall i, j$$

A célfüggvényben pedig az értékeléseket kell minimalizálnunk. A kód:

10.50. kód.

```
Min
        +4x1_2 +3x1_3 +1x1_4 +6x1_5 +7x1_6 +2x1_7 +5x1_8
+5x2_1         +6x2_3 +4x2_4 +7x2_5 +3x2_6 +2x2_7 +1x2_8
+3x3_1 +6x3_2         +5x3_4 +2x3_5 +7x3_6 +4x3_7 +1x3_8
+1x4_1 +6x4_2 +5x4_3         +3x4_5 +4x4_6 +7x4_7 +2x4_8
```

```
+7x5_1 +6x5_2 +1x5_3 +2x5_4        +3x5_6 +4x5_7 +5x5_8
+1x6_1 +4x6_2 +5x6_3 +7x6_4 +2x6_5        +3x6_7 +6x6_8
+6x7_1 +5x7_2 +1x7_3 +3x7_4 +2x7_5 +7x7_6        +4x7_8
+1x8_1 +4x8_2 +3x8_3 +5x8_4 +6x8_5 +2x8_6 +7x8_7

subject to
      +x1_2 +x1_3 +x1_4 +x1_5 +x1_6 +x1_7 +x1_8 = 1
+x2_1       +x2_3 +x2_4 +x2_5 +x2_6 +x2_7 +x2_8 = 1
+x3_1 +x3_2       +x3_4 +x3_5 +x3_6 +x3_7 +x3_8 = 1
+x4_1 +x4_2 +x4_3       +x4_5 +x4_6 +x4_7 +x4_8 = 1
+x5_1 +x5_2 +x5_3 +x5_4       +x5_6 +x5_7 +x5_8 = 1
+x6_1 +x6_2 +x6_3 +x6_4 +x6_5       +x6_7 +x6_8 = 1
+x7_1 +x7_2 +x7_3 +x7_4 +x7_5 +x7_6       +x7_8 = 1
+x8_1 +x8_2 +x8_3 +x8_4 +x8_5 +x8_6 +x8_7       = 1

x1_2 -x2_1 = 0
x1_3 -x3_1 = 0
x1_4 -x4_1 = 0
x1_5 -x5_1 = 0
x1_6 -x6_1 = 0
x1_7 -x7_1 = 0
x1_8 -x8_1 = 0
x2_3 -x3_2 = 0
x2_4 -x4_2 = 0
x2_5 -x5_2 = 0
x2_6 -x6_2 = 0
x2_7 -x7_2 = 0
x2_8 -x8_2 = 0
x3_4 -x4_3 = 0
x3_5 -x5_3 = 0
x3_6 -x6_3 = 0
x3_7 -x7_3 = 0
x3_8 -x8_3 = 0
x4_5 -x5_4 = 0
x4_6 -x6_4 = 0
x4_7 -x7_4 = 0
x4_8 -x8_4 = 0
x5_6 -x6_5 = 0
x5_7 -x7_5 = 0
x5_8 -x8_5 = 0
x6_7 -x7_6 = 0
x6_8 -x8_6 = 0
x7_8 -x8_7 = 0

x1_2          +x1_3+x1_4          +x1_7
```

```
                   +x2_4     +x2_6+x2_7+x2_8 >= 1
x1_3               +x1_4          +x1_7
                        +x3_5          +x3_8 >= 1
x1_4
                                             >= 1
x1_5     +x1_2+x1_3+x1_4          +x1_7+x1_8
         +x5_2+x5_3+x5_4     +x5_6+x5_7+x5_8 >= 1
x1_6     +x1_2+x1_3+x1_4+x1_5     +x1_7+x1_8
                                             >= 1
x1_7               +x1_4
         +x7_2+x7_3+x7_4+x7_5          +x7_8 >= 1
x1_8     +x1_2+x1_3+x1_4          +x1_7
                                             >= 1
x2_1               +x2_4     +x2_6+x2_7+x2_8
              +x1_3+x1_4          +x1_7      >= 1
x2_3+x2_1          +x2_4     +x2_6+x2_7+x2_8
    +x3_1          +x3_4+x3_5     +x3_7+x3_8 >= 1
x2_4                         +x2_6+x2_7+x2_8
    +x4_1     +x4_3     +x4_5+x4_6     +x4_8 >= 1
x2_5+x2_1     +x2_3+x2_4     +x2_6+x2_7+x2_8
              +x5_3+x5_4     +x5_6+x5_7+x5_8 >= 1
x2_6+                              x2_7+x2_8
    +x6_1               +x6_5     +x6_7      >= 1
x2_7                                   +x2_8
              +x7_3+x7_4+x7_5          +x7_8 >= 1
x2_8
    +x8_1     +x8_3          +x8_6           >= 1
x3_1                    +x3_5          +x3_8
                   +x1_4          +x1_7      >= 1
x3_2+x3_1          +x3_4+x3_5     +x3_7+x3_8
    +x2_1          +x2_4     +x2_6+x2_7+x2_8 >= 1
x3_4+x3_1               +x3_5     +x3_7+x3_8
    +x4_1               +x4_5+x4_6     +x4_8 >= 1
x3_5                                   +x3_8
                                             >= 1
x3_6+x3_1+x3_2     +x3_4+x3_5     +x3_7+x3_8
    +x6_1+x6_2          +x6_5     +x6_7      >= 1
x3_7+x3_1               +x3_5          +x3_8
                                             >= 1
x3_8
    +x8_1                    +x8_6           >= 1
x4_1
                                             >= 1
x4_2+x4_1     +x4_3     +x4_5+x4_6     +x4_8
                             +x2_6+x2_7+x2_8 >= 1
```

```
x4_3+x4_1               +x4_5+x4_6     +x4_8
    +x3_1               +x3_5     +x3_7+x3_8 >= 1
x4_5+x4_1                              +x4_8
              +x5_3                          >= 1
x4_6+x4_1               +x4_5          +x4_8
    +x6_1+x6_2+x6_3     +x6_5     +x6_7+x6_8 >= 1
x4_7+x4_1+x4_2+x4_3     +x4_5+x4_6     +x4_8
              +x7_3     +x7_5                >= 1
x4_8+x4_1
    +x8_1+x8_2+x8_3          +x8_6           >= 1
x5_1     +x5_2+x5_3+x5_4     +x5_6+x5_7+x5_8
         +x1_2+x1_3+x1_4          +x1_7+x1_8 >= 1
x5_2          +x5_3+x5_4     +x5_6+x5_7+x5_8
    +x2_1     +x2_3+x2_4     +x2_6+x2_7+x2_8 >= 1
x5_3
                                       +x3_8 >= 1
x5_4          +x5_3
    +x4_1                              +x4_8 >= 1
x5_6          +x5_3+x5_4
    +x6_1                                    >= 1
x5_7          +x5_3+x5_4     +x5_6
              +x7_3                          >= 1
x5_8          +x5_3+x5_4     +x5_6+x5_7
    +x8_1+x8_2+x8_3+x8_4     +x8_6           >= 1
x6_1
         +x1_2+x1_3+x1_4+x1_5     +x1_7+x1_8 >= 1
x6_2+x6_1               +x6_5     +x6_7
                                  +x2_7+x2_8 >= 1
x6_3+x6_1+x6_2          +x6_5     +x6_7
    +x3_1+x3_2     +x3_4+x3_5     +x3_7+x3_8 >= 1
x6_4+x6_1+x6_2+x6_3     +x6_5     +x6_7+x6_8
    +x4_1               +x4_5          +x4_8 >= 1
x6_5+x6_1
              +x5_3+x5_4                     >= 1
x6_7+x6_1               +x6_5
    +x7_1+x7_2+x7_3+x7_4+x7_5          +x7_8 >= 1
x6_8+x6_1+x6_2+x6_3     +x6_5     +x6_7
    +x8_1                                    >= 1
x7_1     +x7_2+x7_3+x7_4+x7_5          +x7_8
                   +x1_4                     >= 1
x7_2          +x7_3+x7_4+x7_5          +x7_8
                                       +x2_8 >= 1
x7_3
    +x3_1               +x3_5          +x3_8 >= 1
x7_4          +x7_3     +x7_5
```

```
    +x4_1+x4_2+x4_3     +x4_5+x4_6     +x4_8 >= 1
x7_5          +x7_3
              +x5_3+x5_4     +x5_6           >= 1
x7_6+x7_1+x7_2+x7_3+x7_4+x7_5          +x7_8
    +x6_1                +x6_5               >= 1
x7_8          +x7_3+x7_4+x7_5
    +x8_1+x8_2+x8_3+x8_4+x8_5+x8_6           >= 1
x8_1
         +x1_2+x1_3+x1_4          +x1_7      >= 1
x8_2+x8_1     +x8_3          +x8_6
                                             >= 1
x8_3+x8_1                    +x8_6
                                             >= 1
x8_4+x8_1+x8_2+x8_3          +x8_6
    +x4_1                                    >= 1
x8_5+x8_1+x8_2+x8_3+x8_4     +x8_6
              +x5_3+x5_4     +x5_6+x5_7      >= 1
x8_6+x8_1
    +x6_1+x6_2+x6_3     +x6_5     +x6_7      >= 1
x8_7+x8_1+x8_2+x8_3+x8_4+x8_5+x8_6
              +x7_3+x7_4+x7_5                >= 1

end
```

Az optimális megoldás:

```
   No. Column name   St      Activity
------ ------------  --   -------------
     3 x1_4          B                1
    13 x2_7          B                1
    21 x3_8          B                1
    22 x4_1          B                1
    33 x5_6          B                1
    40 x6_5          B                1
    44 x7_2          B                1
    52 x8_3          B                1
```

A mátrixba bejelölve:

	1	2	3	4	5	6	7	8
1		4	3	$\boxed{1}$	6	7	2	5
2	5		6	4	7	3	$\boxed{2}$	1
3	3	6		5	2	7	4	$\boxed{1}$
4	$\boxed{1}$	6	5		3	4	7	2
5	7	6	1	2		$\boxed{3}$	4	5
6	1	4	5	7	$\boxed{2}$		3	6
7	6	$\boxed{5}$	1	3	2	7		4
8	1	4	$\boxed{3}$	5	6	2	7	

Megjegyezzük, hogy a stabil szobatárs problémának (ellentétben a stabil házastárs problémával) nem mindig létezik megoldása.

10.8. Mátrixjátékok

9.14. példa megoldása.

A sorjátékos számára a B tevékenység dominálja a C-t, ezért ezt biztosan nem fogja választani. Ekkor a kifizetésmátrix:

	X	Y	Z
A	5	4	8
B	5	6	6

Amennyiben a sorjátékos a C-t biztosan nem játssza, akkor az oszlopjátékosnak sem éri meg a Z stratégiát választania, mert minden esetben jobban jár az X választásával.

	X	Y
A	5	4
B	5	6

Ha az oszlopjátékos racionális és a Z-t biztosan nem választja, akkor a sorjátékos a B választásával mindegyik kimenet esetén jobban járna, mint az A választása esetén. Ha a sorjátékos a B-t választja, akkor az oszlopjátékos az X választása estén

5 egységet kap, az Y esetén pedig csak 4-et. Ezért a domináns stratégiák miatt a B-X lesz a játék kimenete, ha a játékosok racionálisan döntenek.

Az is könnyen belátható, hogy ez a pont Nash-egyensúly, ha eltérnek ettől a stratégiától, akkor nem érnek el jobb eredményt.

9.15. példa megoldása.

Ennek a zéró-összegű játéknak a sorjátékos kifizetés-mátrixa:

	1	2	3
1	1	-2	2
2	-2	2	-3
3	2	-3	3

Az LP modell, az oszlopjátékos várható kifizetésének meghatározására:

10.51. kód.
```
Min
  v
St
s1:  +1x1 -2x2 +2x3 -v <= 0
s2:  -2x1 +2x2 -3x3 -v <= 0
s3:  +2x1 -3x2 +3x3 -v <= 0
szum: +x1  +x2  +x3 = 1
bounds
v free
end
```

Aminek az eredménye:

10.52. kód.
```
   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 s1           NU             0         -0.25
     2 s2           NU             0          -0.5
     3 s3           NU             0         -0.25
     4 szum         NS             1         -0.25

   No. Column name  St   Activity      Marginal
```

```
------ ------------ -- ------------- -------------
     1 v            B           -0.25
     2 x1           B            0.25
     3 x2           B             0.5
     4 x3           B            0.25
```

A célfüggvény értéke $-0,25$, ami azt jelenti, hogy az oszlopjátékos várható kifizetése $0,25$, a sorjátékosé pedig $-0,25$, tehát a játék az oszlopjátékosnak kedvez.

9.16. példa megoldása.

A sorjátékos várható kifizetésére vonatkozó LP modell:

10.53. kód.

```
Max
   w

St
o1: +80 yA +10 yB +50 yC +50 yD +40 yE -w>=0
o2: +40 yA +90 yB +40 yC +90 yD +10 yE -w>=0
o3: +20 yA +60 yB +40 yC +10 yD +80 yE -w>=0
o4: +50 yA +90 yB +60 yC +90 yD +90 yE -w>=0
o5: +80 yA +30 yB +50 yC +60 yD +50 yE -w>=0

szum: +yA+yB+yC+yD+yE=1

end
```

Aminek az eredménye:

10.54. kód.

```
   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 o1           NL             0     -0.385321
     2 o2           NL             0     -0.238532
     3 o3           NL             0     -0.376147
     4 o4           B        24.8624
     5 o5           B        8.80734
     6 szum         NS             1       47.8899

   No. Column name  St   Activity      Marginal
```

```
------ ----------- -- ------------ ------------
     1 w           B       47.8899
     2 yA          B      0.431193
     3 yB          B      0.311927
     4 yC          NL            0       -4.0367
     5 yD          NL            0       -3.3945
     6 yE          B      0.256881
```

Amiből következik, hogy a sorjátékos a C és D stratégiát biztosan nem fogja választani, az A, B és E stratégiákat pedig a következő valószínűséggel fogja választani:

stratégia	valószínűség
B	43,12%
C	31,19%
E	25,69%

A korlátok árnyékáraiból meghatározható a duál feladat változói, amik az oszlopjátékos stratégiaválasztásainak a valószínűségei. Ezek alapján az oszlopjátékos a 4-es és az 5-ös stratégiát nem fogja választani, a többi stratégiát pedig a következő valószínűségekkel választja:

stratégia	valószínűség
1	38,53%
2	23,85%
3	37,61%

Ami alapjána kimenetek valószínűségei:

	1	2	3
B	16,61%	10,29%	16,22%
C	12,02%	7,44%	11,73%
E	9,90%	6,13%	9,66%

Mivel az 50-50-es kifizetést egyedül a C-1 stratégiák adják, ezért ez a kifizetés $12,02\%$-os valószínűséggel fog kialakulni.

9.17. példa megoldása.

András (sorjátékos) kifizetés-mátrixa:

	Kő	**Papír**	**Olló**	**Kő-Papír**	**Kő-Olló**	**Papír-Olló**
Kő	0	-100	100	-50	0	-200
Papír	100	0	-100	0	-200	-50
Olló	-100	100	0	-200	-50	0
Joker	-100	-100	-100	400	400	400

András várható kifizetésére vonatkozó LP modell:

10.55. kód.

```
Max
   w
subject to

O_K:  +  0xK +100xP -100xO -100xJ -w>=0
O_P:  -100xK +  0xP +100xO -100xJ -w>=0
O_O:  +100xK -100xP +  0xO -100xJ -w>=0
O_KP: - 50xK +  0xP -200xO +400xJ -w>=0
O_KO: +  0xK -200xP - 50xO +400xJ -w>=0
O_PO: -200xK - 50xP +  0xO +400xJ -w>=0
szum: +   xK +   xP +   xO +   xJ =1

Bounds

w free

end
```

Aminek az eredménye:

10.56. kód.

```
   No.   Row name   St   Activity      Marginal
------ ------------ -- ------------- -------------
     1 O_K          NL             0     -0.161905
     2 O_P          NL             0     -0.361905
     3 O_O          NL             0     -0.304762
     4 O_KP         NL             0     -0.171429
     5 O_KO         B              0
     6 O_PO         B              0
     7 szum         NS             1      -14.2857

   No. Column name  St   Activity      Marginal
------ ------------ -- ------------- -------------
```

```
1 w             B        -14.2857
2 xK            B        0.285714
3 xP            B        0.285714
4 xO            B        0.285714
5 xJ            B        0.142857
```

Tehát a játék Bálintnak kedvez, mert a várható kifizetése Andrásnak $-14,2857$.

9.18. példa megoldása.

A meccs kimeneteleinek mátrixa:

			Ellenfél (1,2 félidő)			
			T	T	V	V
felállás	1. félidő	2. félidő	T	V	V	T
E	T	T	50	50	55	55
E	T	V	50	50	55	55
E	V	V	45	45	50	50
E	V	T	45	45	50	50
A	T	T	52,5	52,5	57,5	57,5
A	T	V	52,5	52,5	57,5	57,5
A	V	V	35	35	40	40
A	V	T	35	35	40	40
B	T	T	47,5	47,5	52,5	52,5
B	T	V	47,5	47,5	52,5	52,5
B	V	V	50	50	55	55
B	V	T	50	50	55	55

Az LP modell:

10.57. kód.

```
Max
   w
St

T_T: +  50xETT +  50xETV +45xEVV +45xEVT
     +52.5xATT +52.5xATV +35xAVV +35xAVT
     +47.5xBTT +47.5xBTV +50xBVV +50xBVT -w >= 0
T_V: +  50xETT +  50xETV +45xEVV +45xEVT
     +52.5xATT +52.5xATV +35xAVV +35xAVT
     +47.5xBTT +47.5xBTV +50xBVV +50xBVT -w >= 0
```

```
V_V: +  55xETT +  55xETV +50xEVV +50xEVT
     +57.5xATT +57.5xATV +40xAVV +40xAVT
     +52.5xBTT +52.5xBTV +55xBVV +55xBVT -w >= 0
V_T: +  55xETT +  55xETV +50xEVV +50xEVT
     +57.5xATT +57.5xATV +40xAVV +40xAVT
     +52.5xBTT +52.5xBTV +55xBVV +55xBVT -w >= 0

szum: +xETT +xETV +xEVV +xEVT
      +xATT +xATV +xAVV +xAVT
      +xBTT +xBTV +xBVV +xBVT = 1

end
```

Aminek az eredménye:

10.58. kód.

```
   No.   Row name   St   Activity       Marginal
------ ------------ -- -------------  -------------
     1 T_T          NL             0              -1
     2 T_V          B              0
     3 V_V          B              5
     4 V_T          B              5
     5 szum         NS             1            52.5

   No. Column name  St   Activity       Marginal
------ ------------ -- ------------- -------------
     1 w            B           52.5
     2 xETT         NL             0          -2.5
     3 xETV         NL             0          -2.5
     4 xEVV         NL             0          -7.5
     5 xEVT         NL             0          -7.5
     6 xATT         B              1
     7 xATV         NL             0         < eps
     8 xAVV         NL             0         -17.5
     9 xAVT         NL             0         -17.5
    10 xBTT         NL             0            -5
    11 xBTV         NL             0            -5
    12 xBVV         NL             0          -2.5
    13 xBVT         NL             0          -2.5
```

Tehát a csapatnak az A felállással kell felállnia az ellenféllel szemben és végig támadniuk kell a modell alapján. Arra számítanak, hogy az ellenfél is végig támadni fog a mérkőzésen.