EGÉSZÉRTÉKŰ PROGRAMOZÁS

Dr. Nagy Tamás
egyetemi docens

Miskolci Egyetem

Alkalmazott Matematikai Tanszék

Tartalomjegyzék

1. Bevezetés

1.1. Matematikai programozási (MP) feladat

A matematikai programozási feladatot a következőképpen definiáljuk.

Meghatározandó az $\mathbf{x} = (x_1, \ldots, x_n)$ vektor, amely minimalizálja az

$$f(x_1, \ldots, x_n)$$

függvényt az alábbi feltételek mellett

$$\begin{aligned} g_i(x_1, \ldots, x_n) &\leqq 0, \quad i = 1, \ldots, m \\ h_i(x_1, \ldots, x_n) &= 0, \quad i = 1, \ldots, k \\ \mathbf{x} &\in X \end{aligned}$$

ahol az $\mathbf{x} \in \mathbb{R}^n$ a döntési változó vektor, az $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, $g_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \ldots, m)$, $h_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \ldots, k)$ n változós valós függvények, amelyeket rendre célfüggvénynek, egyenlőtlenséges ill. egyenlőséges feltételi függvényeknek nevezünk. Az $X \subseteq \mathbb{R}^n$ halmaz pedig nyílt halmaz. Számos optimalizálási modellben $m = k = 0$ és $X \equiv \mathbb{R}^n$, az ilyen feladatot feltétel nélküli optimalizálási feladatnak nevezzük. Egyéb optimalizálási modelleket pedig feltételes optimalizálási feladatnak nevezünk.

1.2. Lineáris programozási (LP) feladat

A legegyszerűbb feltételes optimalizálási feladatban a célfüggvény és a feltételi függvények a döntési változónak lineáris függvényei, ezt az optimalizálási problémát lineáris programozási feladatnak nevezzük. A lineáris programozási feladat standard formája a következő:

(i) Skalár formában:

Minimalizálandó a

$$\sum_{j=1}^{n} c_j x_j$$

a következő feltételek mellett

$$\begin{aligned} \sum_{j=1}^{n} a_{ij} x_j &= b_i, \quad i = 1, \ldots, m \\ x_j &\geqq 0, \quad j = 1, \ldots, n \end{aligned}$$

(ii) Mátrix-vektor formában:

$$\begin{aligned} \mathbf{cx} &\to \min! \\ \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geqq \mathbf{0} \end{aligned}$$

ahol $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$ ismert konstans mátrix ill. vektorok, $\mathbf{x} \in \mathbb{R}^n$ pedig a döntési változó vektor. A $\mathbf{cx}$ a $\mathbf{c}$ és az $\mathbf{x}$ vektorok skaláris szorzatát jelöli.

1.3. Egészértékű lineáris programozási (ILP) feladat

Számos optimalizálási feladatban megköveteljük, hogy a döntési változó értéke csak egész szám lehet, ezeket a feladatokat egészértékű programozási feladatoknak nevezzük, szokásos elnevezés még az integer programozás is.

Például, ha a döntési változó egy befektetési összeget vagy egy diétás étel kalóriamennyiségét, stb. jelenti, akkor ezek az értékek tört értékek is lehetnek, viszont, ha a döntési változó egy vállalat gépparkjának létesítéséhez megvásárolandó gépkocsik darabszámát vagy egy munkavégzéshez igénybe veendő munkások számát jelenti, akkor szükséges megkövetelni annak egész voltát.

A következőkben az integer programozásban használatos elnevezéseket ismertjük. Ha egy optimalizálási probléma minden döntési változójáról megköveteljük az egészértékűséget, akkor azt tiszta integer programozási feladatnak nevezzük, amennyiben nem mindegyik döntési változó egész, akkor vegyes (mixed) integer programozási feladatnak nevezzük. Abban az esetben, amikor az integer döntési változó csak 0 vagy 1 értéket vehet fel, akkor tiszta (vegyes) 0-1 programozási feladatról beszélünk, szokásos még a tiszta (vegyes) bináris programozási feladat elnevezés is. Szigorú értelemben minden integer programozási feladat nemlineáris feladat, mivel a probléma függvényei csak diszkrét értékeknél vannak értelmezve. Ha eltekintünk a döntési változók egészértékűségétől és az így keletkező feladat lineáris programozási feladat, azaz a probléma összes függvénye lineáris függvény, akkor az integer programozási feladatot integer lineáris programozási feladatnak nevezzük.

A következő fejezetekben az integer lineáris programozási feladatokkal fogunk foglalkozni, azok néhány ismert modelljét és a leginkább alkalmazott megoldási módszereket mutatjuk be. Az integer lineáris programozási feladat standard alakja a következő:

$$\begin{aligned} \mathbf{cx} &\rightarrow \text{min!} \\ \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geqq \mathbf{0}, \text{ egész} \end{aligned}$$

2. Egészértékű optimalizálási modellek

Ebben a fejezetben néhány olyan modellt ismertetünk, amelyekben a döntési változó csak egész szám lehet.

2.1. Befektetési modellek

A következő két alfejezetben különböző befektetési/beruházási lehetőségeket vizsgálunk olyan szempontból, hogy kiválasszuk azokat, amelyek megvalósításához elegendő pénzeszközök állnak rendelkezésünkre és a hozamuk minél nagyobb legyen. Egy- és többperiódusos modelleket különböztetünk meg.

2.1.1. Egyperiódusos befektetési modell

Tegyük fel, hogy legfeljebb 18.7 millió Ft-ot akarunk befektetni. Négy beruházási/befektetési lehetőséget vizsgálunk, amelyekre vonatkozóan az alábbi ismereteink vannak. Az első befektetési lehetőség megvalósítási költsége 6.1 millió Ft, a többié pedig rendre 8.4, 5.6 és 4.1

millió Ft. Az egyes befektetési lehetőségek jelenértékre diszkontált hozama rendre az alábbi: 9.3, 12.5, 7.9 és 5.8 millió Ft. Melyik beruházási lehetőséget tudjuk teljes mértékben megvalósítani a rendelkezésre álló pénzkeretből úgy, hogy a hozam maximális legyen?

Kézenfekvő az alábbi döntési változó alkalmazása:

$$x_j = \begin{cases} 1, & \text{ha a } j\text{-edik beruházást megvalósítjuk,} \\ 0, & \text{ha a } j\text{-edik beruházást nem valósítjuk meg.} \end{cases}$$

Ez az alábbi 0-1 programozási feladatra vezet:

$$\begin{array}{rcrcrcrcl} 9.3x_1 & + & 12.5x_2 & + & 7.9x_3 & + & 5.8x_4 & \rightarrow & \max! \\ 6.1x_1 & + & 8.4x_2 & + & 5.6x_3 & + & 4.1x_4 & \leqq & 18.7 \end{array}$$

$$x_j \in \{0,1\}, \quad j = 1,2,3,4$$

2.1.2. Többperiódusos befektetési modell

Tekintsünk szintén négy befektetési/beruházási lehetőséget, de most több időszakot vegyünk figyelembe, nevezetesen hármat. Mindhárom időszakban ismert, hogy legfeljebb mennyi pénzkeretet vagyunk hajlandók a beruházásra fordítani, ezek rendre legyenek a következők: 18, 16 és 19 millió Ft. Az első beruházás az egyes időszakokban rendre az alábbi pénzbefektetéseket követeli meg: 6, 9 és 3 millió Ft-ot. A második beruházás az egyes időszakokban rendre az alábbi pénzbefektetéseket követeli meg: 8, 0 és 11 millió Ft-ot. A harmadik beruházás az egyes időszakokban rendre az alábbi pénzbefektetéseket követeli meg: 0, 5 és 7 millió Ft-ot. A negyedik beruházás pedig az egyes időszakokban rendre az alábbi pénzbefektetéseket követeli meg: 4, 5 és 6 millió Ft-ot. Az egyes befektetési lehetőségek hozama a három időszak végén rendre az alábbi: 16, 22, 12 és 8 millió Ft.

Ennél a modellnél is a 0-1 típusú döntési változót használhatjuk, amely a következő

$$x_j = \begin{cases} 1, & \text{ha a } j\text{-edik befektetésbe/beruházásba invesztálunk,} \\ 0, & \text{ha a } j\text{-edik befektetésbe/beruházásba nem invesztálunk.} \end{cases}$$

Az alábbi 0-1 programozási feladatot kapjuk:

$$\begin{array}{rcrcrcrcl} 16x_1 & + & 22x_2 & + & 12x_3 & + & 8x_4 & \rightarrow & \max! \\ 6x_1 & + & 8x_2 & + & & + & 4x_4 & \leqq & 18 \\ 9x_1 & + & & + & 5x_3 & + & 5x_4 & \leqq & 16 \\ 3x_1 & + & 11x_2 & + & 7x_3 & + & 6x_4 & \leqq & 19 \end{array}$$

$$x_j \in \{0,1\}, \quad j = 1,2,3,4$$

2.2. Logikai feltételek kezelése 0-1 döntési változókkal

2.2.1. Egyszerű logikai feltételek

A feladatokban megfogalmazott logikai feltételek könnyen kezelhetők 0-1 változókkal. Erre mutatunk két példát.

Példa:

Tekintsük a fentebb ismertetett egyperiódusos befektetési modellt, amelyben legyen további három megkötés. Ezek legyenek az alábbiak:

1. Legfeljebb csak két befektetést valósíthatunk meg.
2. Ha a második befektetést megvalósítjuk, akkor a negyediket is meg kell valósítani.
3. Ha az első befektetést megvalósítjuk, akkor a harmadikat nem valósíthatjuk meg.

Megoldás:
Ezek a logikai feltételek az alábbi módon építhetők be a modellünkbe:

1. $x_1 + x_2 + x_3 + x_4 \leqq 2$
2. $x_2 - x_4 \leqq 0$, mert
   ha $x_2 = 1$, akkor $x_4 = 1$
   ha $x_2 = 0$, akkor $x_4 = 0$ vagy 1
3. $x_1 + x_3 \leqq 1$, mert
   ha $x_1 = 1$, akkor $x_3 = 0$
   ha $x_1 = 0$, akkor $x_3 = 0$ vagy 1.

Példa:
Egy olajkutató vállakozás 10 lehetséges fúrási hely közül akarja kiválasztani a számára legjobb (legtöbb profitot szolgáltató) ötöt. Jelölje az egyes lehetséges fúrási helyeket $h_1, h_2, \ldots, h_{10}$ és legyen a várható profit az egyes fúrási helyeken $p_1, p_2, \ldots, p_{10}$. Ismert továbbá a kiválasztás három fontos szabálya (előírása), amelyek a következők:

1. Ha a h_2 helyet megkutatjuk, akkor a h_3 helyet is meg kell kutatni.
2. Ha a h_1 és a h_7 helyet is megkutatjuk, akkor nincs szükség a h_8 hely megkutatására.
3. Ha a h_3 vagy a h_4 helyet megkutatjuk, akkor nincs szükség a h_5 hely megkutatására.

Megoldás:
Legyenek $x_1, x_2, \ldots, x_{10}$ a döntési változók, amelyeket a következőképpen definiálunk:

$$x_j = \begin{cases} 1, & \text{ha a } h_j \text{ helyet megkutatjuk,} \\ 0, & \text{ha a } h_j \text{ helyet nem kutatjuk meg.} \end{cases}$$

A fentieket figyelembe véve az alábbi integer programozási feladatot kapjuk:

$$\sum_{j=1}^{10} p_j x_j \to \max!$$

$$\begin{aligned} \text{5 fúróhely kiválasztás:} &\quad \textstyle\sum_{j=1}^{10} x_j = 5 \\ \text{1. előírás:} &\quad x_2 - x_3 \leqq 0 \\ \text{2. előírás:} &\quad x_1 + x_7 + x_8 \leqq 2 \\ \text{3. előírás:} &\quad \begin{cases} x_3 + x_5 \leqq 1 \\ x_4 + x_5 \leqq 1 \end{cases} \end{aligned}$$

2.2.2. Vagy-vagy feltételek

(*i*) Vannak olyan gyakorlati problémák, amelyekben azt írjuk elő, hogy két feltétel közül csak egy teljesülhet. Tekintsük az alábbi két feltételt

$$\begin{aligned} \mathbf{a}_1\mathbf{x} &\leqq b_1 \\ \mathbf{a}_2\mathbf{x} &\leqq b_2 \end{aligned}$$

ahol $\mathbf{a}_1, \mathbf{a}_2, \mathbf{x} \in \mathbb{R}^n$ vektorok, $b_1, b_2 \in \mathbb{R}$ számok. Azt az előírást, hogy két feltétel közül csak egy teljesülhet egy új 0-1 változó és egy nagyon nagy pozitív szám bevezetésével az alábbi módon kényszeríthetjük ki. Jelölje a bináris változót y, az adott nagy számot pedig M.

Az előírás a következő feltételekkel adható meg:

$$\begin{aligned} \mathbf{a}_1\mathbf{x} &\leqq b_1 + My \\ \mathbf{a}_2\mathbf{x} &\leqq b_2 + M(1-y) \\ y &= 0 \text{ vagy } 1 \end{aligned}$$

A megoldás helyessége könnyen ellenőrizhető, hiszen

ha $y = 0$, akkor az első feltétel válik aktívvá, a második feltétel passzív,

ha $y = 1$, akkor a második feltétel válik aktívvá, az első feltétel passzív.

Az $y = 0$ esetben az első feltétel jobboldala az eredeti marad, míg a második feltétel jobboldala nagyon nagy szám lesz, így a második feltétel nem jelent megszorítást. Az $y = 1$ eset hasonlóan értelmezhető.

(*ii*) A gyakorlati problémákban az is előfordulhat, hogy m feltételből csak k ($k < m$) feltétel teljesedhet.

Legyen az m feltétel a következő:

$$\sum_{j=1}^{n} a_{ij}x_j \leqq b_i, \quad i = 1, \ldots, m$$

Hasonlóan az előzőhöz, itt is bináris változók bevezetésével lehet az előírást kezelni, de itt minden feltételhez rendelni kell egy 0-1 váltózót. Az alábbi összefüggésekkel lehet kikényszeríteni, hogy csak k feltétel teljesüljön:

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}x_j &\leqq b_i + My_i, \quad i = 1, \ldots, m \\ \sum_{i=1}^{m} y_i &= m - k \\ y_i &= 0 \text{ vagy } 1, \quad i = 1, \ldots, m \end{aligned}$$

A bináris változók összegére vonatkozó előírásból láthatjuk, hogy $(m - k)$ darab y_i változó értéke lesz 1, ami azt jelenti, hogy ennyi feltétel válik passzívvá, k darab feltétel pedig aktív lesz.

(*iii*) A fentiekhez hasonlóan bináris változók bevezetésével kezelhető a következő megkötés is. Legyen a feladat egy adott feltétele a következő:

$$\sum_{j=1}^{n} a_j x_j = b$$

Adott k darab lehetőség a jobboldali b értékre, ezek legyenek rendre: $b_1, \ldots, b_k$. A megkötés az, hogy a $\sum_{j=1}^{n} a_j x_j = b_i$ $(i = 1, \ldots, k)$, azaz k darab feltétel közül csak egy teljesedjen. Ezt az alábbi módon valósíthatjuk meg:

$$\begin{aligned} \sum_{j=1}^{n} a_j x_j &= \sum_{i=1}^{k} b_i y_i \\ \sum_{i=1}^{k} y_i &= 1 \\ y_i &= 0 \text{ vagy } 1, \qquad i = 1, \ldots, k \end{aligned}$$

Könnyen ellenőrizhető, hogy a bináris változók összegére tett feltétel miatt pontosan egy y_i változó értéke lesz 1.

2.3. Transzformálás bináris programozási feladattá

A bináris (más elnevezéssel 0-1) programozási problémák fontosságát az adja, hogy minden integer programozási feladat megfogalmazható ekvivalens bináris programozási feladatként is.

Legyen x egy integer nemnegatív változó és legyen ismert a változónak egy felső korlátja, jelölje ezt az u egész szám, azaz $0 \leqq x \leqq u$. Ekkor az x egész változó felírható az $y_0, y_1, y_2, \ldots, y_u$ 0-1 változók összegeként, azaz

$$x = y_0 + y_1 + y_2 + \ldots + y_u$$

Ha az u felső korlát nagy, akkor elég sok bináris változót kell bevezetni. Ennél sokkal hatékonyabb módszer az alábbi transzformáció (kettes számrendszerben való felírás):

$$x = y_0 + 2y_1 + 2^2 y_2 + \ldots + 2^k y_k$$

ahol k a $2^{k+1} - 1 \geqq u$ egyenlőségből határozható meg. Ez sokkal kevesebb y_i bináris változót követel meg.

Ha például $u = 18$, akkor az első transzformáció esetén 19 darab bináris változó bevezetésére kerül sor, a kettes számrendszerbeli transzformáció esetén pedig csak 5 darab bináris változót kell bevezetni. Egy integer problémánál, ha az x változó helyébe behelyettesítjük az y_i bináris változókat, akkor vagy egy tiszta 0-1 feladatot vagy egy vegyes 0-1 feladatot kapunk. A transzformálásnak általában az a hátránya, hogy túl sok bináris változó keletkezhet, ami a megoldáshoz szükséges számítási időt erőteljesen megnövelheti. Nem mindig célszerű tehát az alkalmazása.

2.4. Hátizsák feladat

A feladat elnevezése a következő megfogalmazásra utal. Egy turista hátizsákjában n különböző hasznos holmit szeretne a túrájára vinni. Az egyes tárgyak súlya legyen $a_1, a_2, \ldots, a_n$ és a hátizsákjában legfeljebb b súlyt tud elvinni. Az összes tárgy nem fér a hátizsákjába, ezért a turista szelektálni kénytelen. Minden tárgynak ad egy, a tárgynak a túra során betöltendő hasznosságát mérő számértéket, amelyek legyenek $c_1, c_2, \ldots, c_n$. A válogatást úgy végzi,

hogy a súlykorlátozás betartása mellett a magával vitt tárgyak összértéke (összhasznossága) minél nagyobb legyen.

A válogatást tárgyanként egy 0 vagy 1 értéket felvehető döntési változóval írhatjuk le. Legyen a döntési változó x_j, amelynek értéke legyen az alábbi

$$x_j = \begin{cases} 1, & \text{ha a } j\text{-edik tárgyat a túrista beteszi a hátizsákjába,} \\ 0, & \text{ha a } j\text{-edik tárgyat a túrista nem teszi be a hátizsákjába.} \end{cases}$$

A $\sum_{j=1}^{n} a_j x_j$ mennyiség azt jelenti, hogy mennyi a hátizsákba behelyezett tárgyak összes súlya, a $\sum_{j=1}^{n} c_j x_j$ mennyiség pedig azt jelenti, hogy mennyi a hátizsákba behelyezett tárgyak összes hasznossága. Így a fenti probléma matematikai megfogalmazása a következő:

Legyenek adottak a $\mathbf{c} = (c_1, c_2, \ldots, c_n), \mathbf{a} = (a_1, a_2, \ldots, a_n)$ vektorok és a b szám, amelyek pozitív egészek és $\sum_{j=1}^{n} a_j > b$. Meghatározandó az $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ vektor úgy, hogy

$$\sum_{j=1}^{n} c_j x_j \to \max!$$

$$\sum_{j=1}^{n} a_j x_j \leqq b$$

$$x_j = 0 \text{ vagy } 1, \quad j = 1, 2, \ldots, n$$

Vektoros megfogalmazásban:

$$\begin{aligned} \mathbf{cx} &\to \max! \\ \mathbf{ax} &\leqq b \\ \mathbf{x} &\in \{0,1\}^n \end{aligned}$$

Általában minden olyan integer programozási feladatot hátizsák feladatnak neveznek, amelynek csak egyetlen feltétele van. A fentebb megfogalmazott hátizsák feladatot bináris hátizsák feladatnak is nevezhetjük, amelynek két fő jellemző tulajdonsága van:

- csak egy feltételt tartalmaz,
- a benne szereplő összes alapadat (a_j, c_j, b) pozitív egész szám.

Speciális hátizsák feladat az ún. **pénzváltási probléma**. Ebben a feladatban egy adott pénzösszeget akarunk pontosan kifizetni egy adott pénzrendszer címleteivel úgy, hogy minimális számú pénzdarabot használunk fel. Ilyen feladat volt a régebbi időkben a készpénzes bérkifizetés, amikor a munkavállaló borítékban kapta meg a fizetését, mégpedig a legkevesebb pénzdarabbal. A jelen korban is még aktuális a feladat, gondoljunk az üzletekben történő készpénzes vásárlásokra, amikor a pénztárgép automatikusan a legkevesebb darabszámú pénzt adja vissza.

Legyen egy pénzrendszer címleteinek száma n, az egyes címletek értékei pedig $a_1, a_2, \ldots, a_n$. Legyen b a kifizetendő pénzösszeg.

Legyen a döntési változó az x_j egész szám, amelynek értéke azt mutatja, hogy a j-edik címletből hány darabot használunk fel a kifizetésnél.

A $\sum_{j=1}^{n} a_j x_j$ mennyiség azt jelenti, hogy mennyi pénzt fizetünk ki összesen.

A $\sum_{j=1}^{n} x_j$ mennyiség a pénzdarabok számát jelenti.

Ezek alapján a pénzváltási probléma matematikai megfogalmazása a következő:

$$\sum_{j=1}^{n} x_j \to \min!$$

$$\sum_{j=1}^{n} a_j x_j = b$$

$$x_j \geqq 0, \text{ egész} \quad j = 1, 2, \ldots, n$$

A következő példában egy hátizsák feladatra vezető példát mutatunk be.

Példa:
Valamely beruházási program megvalósítására összesen 14 millió Ft áll rendelkezésünkre. Négy beruházási javaslatot vizsgálunk, amelyek rendre 7, 3, 5, 4 millió Ft-ba kerül. Az egyes beruházási lehetőségek rendre 11, 4, 8, 6 millió Ft hasznot hoznak. Mely beruházási javaslatokat célszerű megvalósítani, hogy a rendelkezésre álló pénzkeretet ne lépjük túl és az összes haszon maximális legyen?

Legyen a döntési változó x_j, amelynek értéke legyen 1, ha a j-edik beruházási javaslatot megvalósítjuk, ill. legyen az értéke 0, ha a j-edik beruházási javaslatot nem valósítjuk meg.
A feltétel és a célfüggvény felírása nem jelent különösebb problémát. A feladat matematikai modellje az alábbi:

$$\begin{array}{rcrcrcrcl} 11x_1 & + & 4x_2 & + & 8x_3 & + & 6x_4 & \to & \max! \\ 7x_1 & + & 3x_2 & + & 5x_3 & + & 4x_4 & \leqq & 14 \end{array}$$

$$x_j \in \{0, 1\}, \quad j = 1, 2, 3, 4$$

2.5. Fix költséges termelési modell

Tekintsünk egy céget, amely n különböző terméket gyárt. Legyen c_j a j-edik termék egységnyi előállításának költsége, legyen $k_j > 0$ a j-edik termék gyártásának beindításával kapcsolatos fix költség.

Legyen a feladat döntési változója x_j, amely a j-edik termékből előállított mennyiséget jelenti. Ekkor a j-edik termék előállításának teljes költsége:

$$C_j = \begin{cases} k_j + c_j x_j, & \text{ha } x_j > 0 \\ 0, & \text{ha } x_j = 0 \end{cases}$$

Ha ugyanis egy terméket nem gyártunk, akkor az előkészítési költsége sem merül fel. Ahhoz, hogy a költségre vonatkozó célfüggvényt fel tudjuk írni, be kell vezetni minden termékhez egy bináris változót, legyen ez az y_j és jelentse a következőket

$$y_j = \begin{cases} 1, & \text{ha a } j\text{-edik terméket gyártjuk } (x_j > 0), \\ 0, & \text{ha a } j\text{-edik terméket nem gyártjuk } (x_j = 0). \end{cases}$$

Legyen M_j az x_j változó egy felső korlátja. Ekkor könnyen belátható, hogy az

$$x_j \leqq M_j y_j, \quad j = 1, \ldots, n$$

feltételek hozzáadásával biztosítani tudjuk, hogy $x_j > 0$ esetén $y_j = 1$ legyen. Ekkor az összköltséget leíró kettős célfüggvényt a

$$\sum_{j=1}^{n}(k_j y_j + c_j x_j)$$

formában írhatjuk fel. A bevezetett feltételek azonban nem biztosítják azt, hogy $x_j = 0$ esetén $y_j = 0$ legyen. A feltétel megengedi ugyanis, hogy $x_j = 0$ esetén $y_j = 1$. Mivel a célunk az összköltség minimalizálása, így optimális esetben $x_j = 0$ esetén $y_j = 0$ fog adódni, hiszen $y_j = 1$ nagyobb költséget eredményezne. Tehát nem kell újabb feltételt bevezetni, mivel a minimalizálás megoldja az $x_j = 0$, $y_j = 1$ esetet.

Összefoglalva tehát a fix költséges feladatot a következőképpen kezelhetjük.

$$\sum_{j=1}^{n}(k_j y_j + c_j x_j) \to \min!$$

$$\begin{array}{ll} x_j \leqq M y_j, & j = 1, \ldots, n \\ x_j \geqq 0, & j = 1, \ldots, n \\ y_j = 0 \text{ vagy } 1, & j = 1, \ldots, n \end{array}$$

ahol M olyan nagy pozitív szám, amely az x_j változók felső korlátja közül a legnagyobb vagy annál is nagyobb.

Természetesen egyéb feltételek, előírások is vannak egy problémában, de ezekkel most nem foglalkoztunk, hiszen fő feladatunk csupán a kettős célfüggvény egységbe foglalása volt.

2.6. Halmazlefedési, halmazfelbontási és halmazkitöltési feladatok

A halmazlefedési, a halmazfelbontási és a halmazkitöltési feladatok mindegyikének az a jellemzője, hogy bináris döntési változókkal vannak megfogalmazva és a feltételek együtthatói szintén binárisak, azaz 0 vagy 1 értékek. A feltételek jobboldalai és a célfüggvény együtthatói általában tetszőleges egész számok. Sok esetben azonban ezek az értékek is binárisak.

2.6.1. Halmazlefedési feladat (HL)

Legyen adott egy S alaphalmaz m elemmel és adott továbbá az S halmaznak n részhalmaza. A halmazlefedési feladat röviden a következő: fedjük le az alaphalmazt részhalmazaival. Bővebben kifejtve: a részhalmazokból válasszunk ki minimális számút úgy, hogy az alaphalmaz minden eleme le legyen fedve, más szavakkal minden alaphalmazbeli elem legalább egyszer szerepeljen a kiválasztott részhalmazok egyesítésében. A következőkben nézzünk erre egy példát:

Példa:
Legyen $m = 5,\ S = \{s_1, s_2, s_3, s_4, s_5\}$

és $n = 7$, a részhalmazok pedig a következők:

$$\begin{aligned} S_1 &= \{s_1, s_3, s_4\} \\ S_2 &= \{s_2, s_3\} \\ S_3 &= \{s_2, s_5\} \\ S_4 &= \{s_1, s_4\} \\ S_5 &= \{s_3, s_5\} \\ S_6 &= \{s_2, s_3, s_5\} \\ S_7 &= \{s_3, s_4\} \end{aligned}$$

A feladat matematikai megfogalmazásához szerkesszük meg az alábbi $\mathbf{A} \in \mathbb{R}^{m \times n}$ ún. incidencia (tartalmazási) mátrixok, amelynek elemei az alábbiak:

$$a_{ij} = \begin{cases} 1, & \text{ha az } S_j \text{ részhalmaz tartalmazza az } s_i \text{ elemet,} \\ 0, & \text{egyébként.} \end{cases}$$

Tehát az $\mathbf{A}$ mátrix sorai az elemeket, oszlopai pedig a részhalmazokat reprezentálják, így a példabeli mátrix 5×7 méretű, amelyet az alábbiak szerint írhatunk:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Az $\mathbf{A}$ mátrix sorainak és oszlopainak összegére az alábbi megjegyzéseket tehetjük:
$\sum_{j=1}^{n} a_{ij}$ jelentése (sorösszeg): azon részhalmazok száma, amelyek tartalmazzák az s_i elemet.
$\sum_{i=1}^{m} a_{ij}$ jelentése (oszlopösszeg): az S_j részhalmazban lévő elemek száma.

A feladat döntési változóját (x_j) a következőképpen definiáljuk:

$$x_j = \begin{cases} 1, & \text{ha az } S_j \text{ részhalmazt kiválasztjuk,} \\ 0, & \text{ha az } S_j \text{ részhalmazt nem választjuk ki.} \end{cases}$$

A célfüggvényünk a feladat szerint a kiválasztott részhalmazok száma, ezt pedig a döntési változók összege adja, amelyet minimalizálni kell, azaz

$$x_1 + x_2 + \ldots + x_n \to \min!$$

A feltételeket pedig az alábbiak szerint írhatjuk. Akkor mondhatjuk, hogy az alaphalmazt lefedtük a részhalmazaival, ha az alaphalmaz minden eleme legalább egy részhalmazzal le van fedve. A $\sum_{j=1}^{n} a_{ij} x_j$ mennyiség azt jelenti, hogy hány darab **kiválasztott részhalmaz** tartalmazza az s_i elemet, ennek pedig legalább 1-nek kell lennie.

Összefoglalva tehát a halmazlefedési feladat a következőképpen fogalmazható meg:

$$\sum_{j=1}^{n} x_j \to \min!$$

$$\sum_{j=1}^{n} a_{ij} x_j \geqq 1, \quad i = 1, \ldots, m$$

$$x_j = 0 \text{ vagy } 1, \quad j = 1, \ldots, n$$

Általánosított halmazlefedési feladat (ÁHL)

A standard halmazlefedési feladat adatain felül legyen adott a részhalmazok $c_1, c_2, \ldots, c_n$ költsége, valamint legyen megadva, hogy az egyes elemek legalább $b_1, b_2, \ldots, b_m$-szer legyenek lefedve. Most a feladatunk úgy kiválasztani a részhalmazokat, hogy a kiválasztott részhalmazok összköltsége minél kisebb legyen és az egyes elemeknek legalább $b_1, b_2, \ldots, b_m$ számú lefedése legyen. A feladat matematikai formája:

$$\sum_{j=1}^{n} c_j x_j \to \min!$$

$$\sum_{j=1}^{n} a_{ij} x_j \geqq b_i, \quad i = 1, \ldots, m$$
$$x_j = 0 \text{ vagy } 1, \quad j = 1, \ldots, n$$

Halmazlefedési feladat gráfokon Adott egy $G = (N, E)$ gráf az N ponthalmazzal és az E élhalmazzal, a pontok száma legyen m, az élek száma pedig n.

Ennek a feladatnak a megfogalmazása nagyon hasonlít a halmazlefedési feladatra. Most az $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n) \in \mathbb{R}^{m \times n}$ mátrixot a gráf mátrixa reprezentálja úgy, hogy minden oszlop egy gráfélre, minden sor pedig egy gráfpontra vonatkozik. Az $\mathbf{A}$ mátrix j-edik oszlopában a j-edik él kezdőpontjánál és végpontjánál van 1-es. Ebben a feladatban az $\mathbf{A}$ mátrix minden $\mathbf{a}_j$ oszlopa tehát pontosan kettő 1-est tartalmaz, a többi oszlopbeli elem zérus. Az $\mathbf{A}$ mátrix i-edik sorában lévő 1-esek pedig azt mutatják, hogy mely élek illeszkednek az i-edik pontra.

Feladatunk kiválasztani azt a legkevesebb számú élet, amely az összes gráfpontot legalább egyszer lefedi.

A feladat x_j döntési változója legyen

$$x_j = \begin{cases} 1, & \text{ha a } j\text{-edik élet kiválasztjuk,} \\ 0, & \text{ha a } j\text{-edik élet nem választjuk ki.} \end{cases}$$

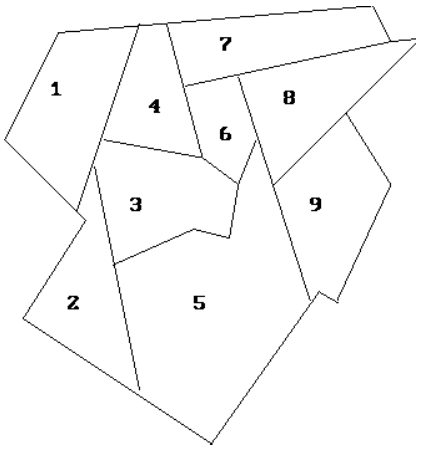
A feladat célfüggvénye:

$$x_1 + x_2 + \ldots + x_n \to \min!$$

A feltételek:

$$\sum_{j=1}^{n} a_{ij} x_j \geqq 1, \quad i = 1, \ldots, m$$

Halmazlefedési feladat elhelyezési feladatra Tekintsük az alábbi elhelyezési problémát. Legyen egy város kilenc kerületre felosztva, amelyet az alábbi ábra illusztrál.



A fenti felosztás alapján szerkesszünk meg egy $\mathbf{A} \in \mathbb{R}^{9\times 9}$ ún. szomszédossági mátrixot, amelynek elemei a következők:

$$a_{ij} = \begin{cases} 1, & \text{ha az } i\text{-edik kerület szomszédos a } j\text{-edik kerülettel vagy } i = j, \\ 0, & \text{egyébként.} \end{cases}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

A város tűzoltó állomások létesítését tervezi. Az egyes kerületekben létesítendő tűzoltó állomás a szomszédos kerületekben előforduló tűzesetek oltását is el tudja végezni.

Az $\mathbf{A}$ mátrix szimmetrikus mátrix, az i-edik sora (j-edik oszlopa) azt mutatja, hogy az i-edik (j-edik) kerületben létesítendő tűzoltó állomás mely kerületek tűzeseteit tudja ellátni. Feladatunk kiválasztani azokat a kerületeket, amelyekben létesítenünk kell tűzoltó állomást, hogy bármelyik kerületben is keletkezzen tűz, azt el lehessen oltani. Tehát minden kerület tűzesete le legyen fedve legalább egy tűzoltó állomással. Célunk az, hogy minél kevesebb számú tűzoltó állomást létesítsünk.

Az x_j döntési változót a következőképpen definiáljuk:

$$x_j = \begin{cases} 1, & \text{ha a } j\text{-edik kerületben létesítünk tűzoltó állomást,} \\ 0, & \text{egyébként.} \end{cases}$$

A feladatunk a következőképpen fogalmazható meg matematikai formában:

$$\begin{array}{ccccccccccccccccccc}
x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & + & x_8 & + & x_9 & \to & \min! \\
 & & & & & & & & & & & & & & & & & & \\
x_1 & + & x_2 & + & x_3 & + & x_4 & & & & & & & & & & & \geqq & 1 \\
x_1 & + & x_2 & + & x_3 & & & + & x_5 & & & & & & & & & \geqq & 1 \\
x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & & & & & & & \geqq & 1 \\
x_1 & & & + & x_3 & + & x_4 & & & + & x_6 & + & x_7 & & & & & \geqq & 1 \\
 & & x_2 & + & x_3 & & & + & x_5 & + & x_6 & & & + & x_8 & + & x_9 & \geqq & 1 \\
 & & & & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & + & x_8 & & & \geqq & 1 \\
 & & & & & & x_4 & & & + & x_6 & + & x_7 & + & x_8 & & & \geqq & 1 \\
 & & & & & & & & x_5 & + & x_6 & + & x_7 & + & x_8 & + & x_9 & \geqq & 1 \\
 & & & & & & & & x_5 & & & & & + & x_8 & + & x_9 & \geqq & 1
\end{array}$$

$$x_j \in \{0,1\}, \quad j = 1, \ldots, 9$$

Megjegyezzük, hogy a feladat egy optimális megoldása: $x_3 = x_8 = x_9 = 1$, a többi $x_j = 0$.

2.6.2. Halmazfelbontási feladat (HF)

A halmazfelbontási feladat egy speciális halmazlefedési feladat, itt azonban nem lefedésről van szó valójában, hanem az alaphalmaz részekre bontásáról. A kiválasztott részhalmazoknak tehát diszjunknak kell lennie, azaz minden elem pontosan egyszer legyen lefedve. A két feladat között csupán az a különbség, hogy a feltételek relációja "$\geqq$" helyett "$=$".

$$\sum_{j=1}^{n} x_j \to \min!$$

$$\begin{aligned}
\sum_{j=1}^{n} a_{ij} x_j &= 1, \quad i = 1, \ldots, m \\
x_j &= 0 \text{ vagy } 1, \quad j = 1, \ldots, n
\end{aligned}$$

Általánosított halmazfelbontási feladat (ÁHF)

A halmazfelbontási feladatot ugyanúgy általánosítjuk, mint a halmazlefedési feladatot, amely a következőképpen írható le:

$$\sum_{j=1}^{n} c_j x_j \to \min!$$

$$\begin{aligned}
\sum_{j=1}^{n} a_{ij} x_j &= b_i, \quad i = 1, \ldots, m \\
x_j &= 0 \text{ vagy } 1, \quad j = 1, \ldots, n
\end{aligned}$$

Halmazfelbontási feladat kiszolgálási problémára Tekintsünk egy áruházat, amely m különböző megrendelőtől kap szállítási megbízást. A szállító kombinálhatja a megrendeléseket, legfeljebb k darab megrendelést elégíthet ki egyszerre.

Definiáljunk n féle tevékenységet, amelyek egy-egy lehetséges kombinációját jelentik az egy, kettő, ... , vagy k darab megrendelés egyszerre történő kielégítésének. Foglaljuk ezt egy $\mathbf{A} \in \mathbb{R}^{m\times n}$ mátrixba, amelynek elemei a következők:

$$a_{ij} = \begin{cases} 1, & \text{ha a } j\text{-edik tevékenység szerint szállítunk az } i\text{-edik megrendelőhöz,} \\ 0, & \text{egyébként.} \end{cases}$$

Legyen c_j a j-edik tevékenység költsége. Tervezzük meg a megrendelők legkisebb költséggel történő kiszolgálását.

Az x_j döntési változót a következőképpen definiáljuk:

$$x_j = \begin{cases} 1, & \text{ha a } j\text{-edik tevékenységet választjuk,} \\ 0, & \text{egyébként.} \end{cases}$$

A $\sum_{j=1}^{n} a_{ij}x_j$ mennyiség azt mutatja, hogy az i-edik megrendelést hányféle kiválasztott tevékenységgel elégíthetjük ki. Mivel a megrendelésnek csak egyféle megvalósítása szükséges, így ennek minden megrendelés esetén egynek kell lennie.

A kiszolgálási feladat tehát matematikai formában a következőképpen írható:

$$\sum_{j=1}^{n} c_j x_j \to \min!$$

$$\sum_{j=1}^{n} a_{ij}x_j = 1, \quad i = 1, \ldots, m$$
$$x_j = 0 \text{ vagy } 1, \quad j = 1, \ldots, n$$

Halmazfelbontási feladat légitársaság személyzetének ütemezésére Egy légitársaság minden járatán egy bizonyos feladatra alkalmazni kell egy megfelelő személyt. A légitársaságnak legyen m járata és n személy közül választhat. Az $\mathbf{A} \in \mathbb{R}^{m\times n}$ mátrix segítségével adjuk, meg, hogy mely járaton mely személy végezheti el a feladatot időbeli elfoglaltságát figyelembe véve, az $\mathbf{A} \in \mathbb{R}^{m\times n}$ mátrix elemei:

$$a_{ij} = \begin{cases} 1, & \text{ha az } i\text{-edik járatnál a } j\text{-edik személy el tudja látni a feladatot,} \\ 0, & \text{egyébként.} \end{cases}$$

Legyen x_j döntési változó a következőképpen definiálva:

$$x_j = \begin{cases} 1, & \text{ha a } j\text{-edik személyt alkalmazzuk,} \\ 0, & \text{egyébként.} \end{cases}$$

A $\sum_{j=1}^{n} a_{ij}x_j$ mennyiség azt mutatja, hogy az i-edik járaton hány kiválasztott személy tudja ellátni a feladatot. Mivel minden járatra egy személyt kell beosztani, így ennek minden járat esetén egynek kell lennie.

Ismerjük továbbá a j-edik személy alkalmazásának c_j költségét. Feladatunk minden járatra egy személy kiválasztása úgy, hogy a személyek alkalmazásának összköltsége minimális legyen. Könnyen látható, hogy ez a probléma egy halmazfelbontási feladatra vezet.

Halmazfelbontási feladat információ visszakeresésre Tegyük fel, hogy n féle különböző fájlból nyerhetünk információt. Legyen c_j a j-edik file hossza. A feladatunk m információ megkeresése a fájlokban. Feltesszük, hogy mindegyik információ legalább egy fájlban fellelhető. Definiáljuk az $\mathbf{A} \in \mathbb{R}^{m \times n}$ mátrix elemeit a következőképpen:

$$a_{ij} = \begin{cases} 1, & \text{ha az } i\text{-edik információ megtalálható a } j\text{-edik fájlban,} \\ 0, & \text{egyébként.} \end{cases}$$

Legyen x_j döntési változó a következőképpen definiálva:

$$x_j = \begin{cases} 1, & \text{ha a } j\text{-edik fájlban keressük az információt,} \\ 0, & \text{egyébként.} \end{cases}$$

A $\sum_{j=1}^{n} a_{ij}x_j$ mennyiség azt mutatja, hogy az i-edik információ hány kiválasztott fájlban található meg. Mivel minden információt egy fájlból akarunk megkeresni, így ennek minden információ esetén egynek kell lennie. Feladatunk meghatározni azokat a fájlokat, amelyekben keressük az információt úgy, hogy a fájlok összhossza minimális. Könnyen látható, hogy ez a probléma is egy halmazfelbontási feladatra vezet.

2.6.3. Halmazkitöltési feladat (HK)

Ebben a problémában az alaphalmazba minél több diszjunk részhalmazát akarjuk betenni. Válasszuk ki tehát a maximális számú diszjunkt részhalmazt. Ebben a feladatban nem követeljük meg, hogy minden elem le legyen fedve, viszont a diszjunkság miatt legfeljebb egyszer lehet lefedve.

A halmazkitöltési feladat matematikai formában a következőképpen írható:

$$\sum_{j=1}^{n} x_j \to \max!$$

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}x_j &\leqq 1, \quad i = 1, \ldots, m \\ x_j &= 0 \text{ vagy } 1, \quad j = 1, \ldots, n \end{aligned}$$

Általánosított halmazkitöltési feladat (ÁHK)

A halmazkitöltési feladatot az eddigiekhez hasonlóan általánosítjuk és a következőképpen írható le:

$$\sum_{j=1}^{n} c_j x_j \to \max!$$

$$\begin{aligned} \sum_{j=1}^{n} a_{ij}x_j &\leqq b_i, \quad i = 1, \ldots, m \\ x_j &= 0 \text{ vagy } 1, \quad j = 1, \ldots, n \end{aligned}$$

2.6.4. Kapcsolat az ÁHL és az ÁHK feladatok között

Az ÁHK feladatot egyszerűen transzformálhatjuk az ÁHL feladatba. Vezessünk be új bináris döntési változókat a következőképpen: $y_j = 1 - x_j$ minden j indexre.

Ekkor az ÁHK feladat célfüggvénye

$$\sum_{j=1}^{n} c_j x_j = \sum_{j=1}^{n} c_j(1 - y_j) = \sum_{j=1}^{n} c_j - \sum_{j=1}^{n} c_j y_j \to \max!$$

amely ekvivalens a $\sum_{j=1}^{n} c_j y_j \to \min!$ célfüggvénnyel, azaz az ÁHL feladat célfüggvényével.

A feltételek átírása pedig a következőképpen történik:

$$\sum_{j=1}^{n} a_{ij} x_j = \sum_{j=1}^{n} a_{ij}(1 - y_j) = \sum_{j=1}^{n} a_{ij} - \sum_{j=1}^{n} a_{ij} y_j \leqq b_i$$

amely ekvivalens a

$$\sum_{j=1}^{n} a_{ij} y_j \geqq \sum_{j=1}^{n} a_{ij} - b_i$$

feltétellel. Ha $\sum_{j=1}^{n} a_{ij} > b_i$, akkor az i-edik jobboldal pozitív egész szám, egyébként pedig az i-edik feltétel felesleges.

Az eredeti (standard) feladatokban a transzformáció akkor hajtható végre, ha $\sum_{j=1}^{n} a_{ij} - 1 = 1$, ekkor az i-edik feltétel $\sum_{j=1}^{n} a_{ij} = 2$, azaz ekkor minden elemet pontosan két részhalmaznak kell tartalmaznia.

Végül összefoglalásképpen közöljük a három általánosított feladatot mátrixos-vektoros formában:

Adott az $\mathbf{A} \in \mathbb{R}^{m \times n}$ mátrix és a $\mathbf{b} \in \mathbb{R}^m, \mathbf{c} \in \mathbb{R}^n$ vektorok, ahol $a_{ij} \in \{0, 1\}$, a b_i, c_j pedig pozitív egész számok.

Halmazlefedési feladat	Halmazfelbontási feladat	Halmazkitöltési feladat
$\mathbf{cx} \to \min!$	$\mathbf{cx} \to \min!$	$\mathbf{cx} \to \max!$
$\mathbf{Ax} \geqq \mathbf{b}$	$\mathbf{Ax} = \mathbf{b}$	$\mathbf{Ax} \leqq \mathbf{b}$
$\mathbf{x} \in \{0, 1\}^n$	$\mathbf{x} \in \{0, 1\}^n$	$\mathbf{x} \in \{0, 1\}^n$

3. Integer lineáris programozás megoldási módszerei

Az előző fejezetben bemutattunk néhány integer problémát, ebben a fejezetben pedig a megoldási módszereket ismertetjük. Az integer problémák megoldására két módszercsalád, a **Vágási módszer** és a **Szétválasztás és korlátozás módszer** a leginkább elterjedt. Röviden a két módszer lényege a következő.

A **Vágási módszer** (cutting planes) lényege, hogy új feltételek bevezetésével kényszerítjük ki az egészértékűséget. Ez a módszer lett először kifejlesztve.

A **Szétválasztás és korlátozás módszer** (branch and bound) lényege pedig abban áll, hogy a feladatot felbontjuk kisebb feladatokra.

Az alábbiakban a módszerek részleteit példával illusztrálva mutatjuk be.

3.1. Integer és folytonos lineáris programozás kapcsolata

Tekintsük az alábbi integer lineáris programozási feladatot (ILP):

$$\left.\begin{array}{l} \mathbf{cx} \rightarrow \text{min! (vagy max!)} \\ \mathbf{Ax} = \mathbf{b} \\ \mathbf{x} \geqq \mathbf{0}, \quad \text{egész} \end{array}\right\} ILP$$

Az ILP feladat egészértékűségi megkötését hagyjuk el, így egy, az eredeti feladathoz rendelt lineáris programozási feladatot kapunk, amelyet nevezzünk folytonos lineáris programozási feladatnak (FLP), amely a következő:

$$\left.\begin{array}{l} \mathbf{cx} \rightarrow \text{min! (vagy max!)} \\ \mathbf{Ax} = \mathbf{b} \\ \mathbf{x} \geqq \mathbf{0} \end{array}\right\} FLP$$

Mivel az FLP feltételi halmaza bővebb, mint az ILP feladaté, ebből a tényből azonnal következnek az alábbi állítások:

1. Ha az ILP minimalizálási feladat, akkor az FLP feladat célfüggvényének optimális (minimális) értéke nem lehet nagyobb (vagyis kisebb vagy egyenlő), mint az ILP feladat célfüggvényének optimális (minimális) értéke.

2. Ha az ILP maximalizálási feladat, akkor az FLP feladat célfüggvényének optimális (maximális) értéke nem lehet kisebb (vagyis nagyobb vagy egyenlő), mint az ILP feladat célfüggvényének optimális (maximális) értéke.

3. Ha az FLP feladat feltételi halmaza üres, akkor az ILP feladaté is az.

4. Ha az FLP feladat optimális megoldásában a változók egészek, akkor ez az optimális megoldás az ILP feladatnak is optimális megoldása.

5. Ha a célfüggvény együtthatói egész számok, akkor az FLP feladat optimális célfüggvényértékének egészre kerekített értékére is igaz az 1. és 2. állítás. Minimum feladat esetén felfelé kell kerekíteni, maximum feladat esetén pedig lefelé kell kerekíteni.

Ha tehát megoldjuk az FLP feladatot, akkor ez számunkra értékes információkat ad. Szerencsés esetben azonnal megkapjuk az integer feladat optimális megoldását. Ha ez nem is következik be, akkor viszont az 1. és 2. állítások értelmében az ILP feladat célfüggvényének optimális értékére kapunk korlátokat, minimalizálási feladat esetén alsó korlátot, maximalizálási feladat esetén pedig felső korlátot.

3.2. Szétválasztás és korlátozás módszer (Branch and Bound)

3.2.1. A Szétválasztás és korlátozás módszer alapjai

A módszer lényegét a modelleknél bemutatott Hátizsák feladat megoldásán keresztül mutatjuk be.

Példa:
Tekintsük a Hátizsák feladat tárgyalásánál bemutatott beruházási feladatot, de a célfüggvény együtthatóinak vegyük a tízszeresét. A megoldandó feladat tehát a következő:

$$\begin{array}{rcrcrcrcl} 110y_1 & + & 40y_2 & + & 80y_3 & + & 60y_4 & \rightarrow & \max! \\ 7y_1 & + & 3y_2 & + & 5y_3 & + & 4y_4 & \leqq & 14 \end{array}$$
$$y_j \in \{0,1\}\,, \quad j = 1,2,3,4$$

A feladathoz rendelt folytonos lineáris programozási feladat (FLP) a következő:

$$\begin{array}{rcrcrcrcl} 110y_1 & + & 40y_2 & + & 80y_3 & + & 60y_4 & \rightarrow & \max! \\ 7y_1 & + & 3y_2 & + & 5y_3 & + & 4y_4 & \leqq & 14 \end{array}$$
$$0 \leqq y_j \leqq 1, \quad j = 1,2,3,4$$

Mivel ez a feladat csak egyetlen feltételt tartalmaz, így az FLP feladat egyszerűen megoldható, nem kell hozzá szimplex módszer. Az egyszerű megoldáshoz rendezzük át a feladatot a c_j/a_j (célfüggvény együttható/feltétel együttható) hányadosok **csökkenő sorrendjében**. Példánkban a hányadosok rendre:

$$\frac{110}{7} = 15.7, \quad \frac{40}{3} = 13.\dot{3}, \quad \frac{80}{5} = 16, \quad \frac{60}{4} = 15\ .$$

Mint látjuk az y_j változók szerint a hányadosok nem csökkenő sorrendben vannak, ezeket csökkenő sorrendbe rendezve az alábbi adódik:

$$\frac{80}{5}, \quad \frac{110}{7}, \quad \frac{60}{4}, \quad \frac{40}{3}\ .$$

Az y_j változókról térjünk át az új x_j változókra úgy, hogy az x_j változók tekintetében már a fenti rendezettség fennálljon. A változócsere a következő lesz: $x_1 = y_3,\ x_2 = y_1,\ x_3 = y_4,\ x_4 = y_2$. Az új változókkal a megoldandó FLP feladat az alábbi:

$$\begin{array}{rcrcrcrcl} 80x_1 & + & 110x_2 & + & 60x_3 & + & 40x_4 & \rightarrow & \max! \\ 5x_1 & + & 7x_2 & + & 4x_3 & + & 3x_4 & \leqq & 14 \end{array}$$
$$0 \leqq x_j \leqq 1, \quad j = 1,2,3,4$$

Bizonyítható, hogy ennek a feladatnak az optimális megoldása mindig egyetlen tört értéket tartalmazhat. Ha a változók a fentebb közölt sorrendben követik egymást, akkor az **első** k darab ismeretlen optimális értéke $x_j = 1$, a $k+1$-edik ismeretlen optimális értéke x_{k+1} tört (lehet egész is), a többi ismeretlen optimális értéke $x_j = 0$. Ehhez tehát sorra kell venni a változókat az elsőtől kezdve és amíg a feltétel teljesül addig $x_j = 1$. A következő változó olyan tört lesz, amellyel a feltétel egyenlőséggel teljesül. Tehát a hátizsák terminológiával elmondva az FLP megoldása a következő: amíg a soron következő tárgy belefér a hátizsákba, beletesszük, amelyik már nem fér bele azt törtértékkel "tesszük bele", a többit pedig nem tesszük bele.

A példabeli FLP feladat megoldása az alábbi: Az $x_1 = 1$, mert az 1. tárgy belefér a hátizsákba (teljesül a feltétel, $5 < 14$), az $x_2 = 1$, mert a 2. tárgy is belefér (teljesül a feltétel, $5 + 7 < 14$). Az x_3 változó már nem lehet 1, csupán $(14 - 12)/4$, azaz 0.5 lehet, az x_4 pedig 0. A FLP feladat optimális megoldása tehát: $x_1 = 1,\ x_2 = 1,\ x_3 = 0.5,\ x_4 = 0$, a célfüggvény maximális értéke pedig 220. Ez utóbbi szerint az eredeti integer feladat (ILP) célfüggvényének maximuma nem lehet 220-nál nagyobb, tehát az ILP feladat célfüggvényének felső korlátja 220. Ez a tény adja a módszer elnevezésében a **korlátozást**. Mivel az optimális megoldás nem egész, így tovább kell folytatni az eljárást.

Most az eredeti feladatot két részfeladatra bontjuk, ez a felbontás adja a módszer elnevezésében a **szétválasztást**. Mivel az optimális megoldásban egyetlen változó értéke, az x_3 értéke tört, ezért az egyik részfeladatban legyen $x_3 = 0$, a másikban pedig $x_3 = 1$ legyen, azaz oldjuk meg a feladatokat úgy, hogy az 3. tárgyat nem tesszük bele a hátizsákba ill. beletesszük.

1. részfeladat: $x_3 = 0$
Ekkor az FLP feladat a következő:

$$\begin{array}{rcrcrcl} 80x_1 & + & 110x_2 & + & 40x_4 & \rightarrow & \max! \\ 5x_1 & + & 7x_2 & + & 3x_4 & \leqq & 14 \end{array}$$
$$0 \leqq x_j \leqq 1, \quad j = 1, 2, 4$$

A fentebb ismertetett egyszerű módon meghatározható az optimális megoldás, amely a következő:

$$x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ x_4 = \frac{2}{3} = 0.667,\ z = 216.67$$

2. részfeladat: $x_3 = 1$
Ekkor az FLP feladat a következő:

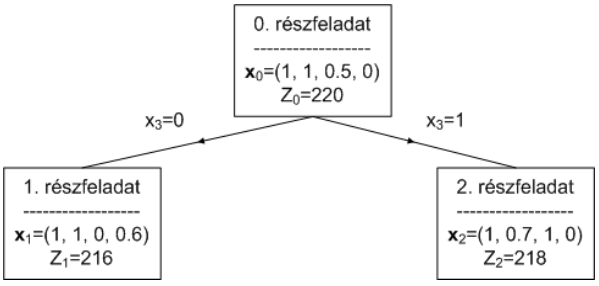
$$\begin{array}{rcrcrcrcl} 60 & + & 80x_1 & + & 110x_2 & + & 40x_4 & \rightarrow & \max! \\ & & 5x_1 & + & 7x_2 & + & 3x_4 & \leqq & 10 \end{array}$$
$$0 \leqq x_j \leqq 1, \quad j = 1, 2, 4$$

Az optimális megoldás:

$$x_1 = 1,\ x_2 = \frac{5}{7} = 0.714,\ x_3 = 1,\ x_4 = 0,\ z = 218.57$$

A megoldás menetét egy fagráfon célszerű illusztrálni. A fagráf pontjai reprezeltálják a részfeladatokat. A szétbontás előtti feladatot 0. részfeladatnak nevezzük. A pontokban az FLP optimális megoldását és annak optimális célfüggvényértékéből az ILP feladatra kapott

célfüggvény korlátot (felső korlát) tüntetjük fel, jelölje ezt a Z nagybetű. Mivel maximum feladatunk van és a célfüggvény együtthatók egészek, így a korlát meghatározásánál lefelé kerekítünk.



A fenti fagráfból leolvashatjuk, hogy az eredeti feladat célfüggvénye 220-nál nem lehet nagyobb, a részfeladatokból pedig azt, hogy $x_3 = 0$ esetén nem lehet 216-nál, $x_3 = 1$ esetén pedig nem lehet 218-nál nagyobb a célfüggvény maximuma.

Most újabb részfeladatokat határozunk meg, amelyet az alábbi elvek szerint végzünk:

- Szétválasztás elve: olyan részfeladaton ágaztatunk el, amelynél a megoldás nem integer (aktív részfeladat).
- Korlátozás elve: olyan részfeladaton ágaztatunk el, amelynél a célfüggvénykorlát értéke a legnagyobb (minimum feladatnál a legkisebb).

Esetünkben az 1. és a 2. részfeladat egyikében sem kaptunk egész megoldást, mindegyik aktív. Ezek közül a 2. részfeladatot választjuk, mert a célfüggvénykorlát itt a legnagyobb. Mivel a 2. részfeladatban az x_2 változó értéke tört, ezért a 2. részfeladat két elágaztatásában $x_2 = 0$, ill. $x_2 = 1$. A 3. és a 4. részfeladatokat és azok megoldását az alábbiak mutatják:

3. részfeladat: $x_3 = 1,\ x_2 = 0$
Ekkor az FLP feladat a következő:

$$\begin{aligned} 60 \ + \ 80x_1 \ + \ 40x_4 \ &\rightarrow \ \max! \\ 5x_1 \ + \ 3x_4 \ &\leqq \ 10 \\ 0 \leqq x_j \leqq 1, \quad j = 1,4 \end{aligned}$$

Az optimális megoldás:

$$x_1 = 1,\ x_2 = 0,\ x_3 = 1,\ x_4 = 1,\ z = 180$$

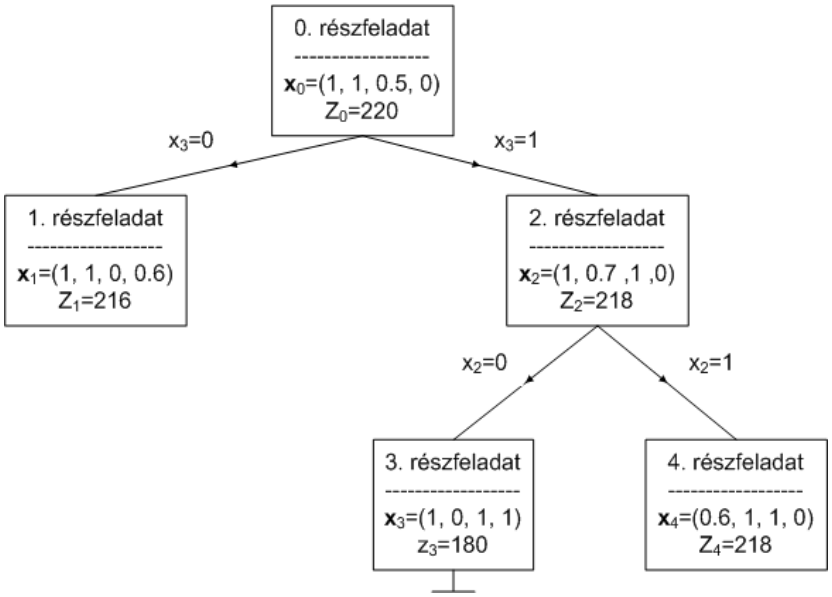
4. részfeladat: $x_3 = 1,\ x_2 = 1$
Ekkor az FLP feladat a következő:

$$\begin{aligned} 170 \ + \ 80x_1 \ + \ 40x_4 \ &\rightarrow \ \max! \\ 5x_1 \ + \ 3x_4 \ &\leqq \ 3 \\ 0 \leqq x_j \leqq 1, \quad j = 1,4 \end{aligned}$$

Az optimális megoldás:

$$x_1 = 0.6,\ x_2 = 1,\ x_3 = 1,\ x_4 = 0,\ z = 218$$

A feladatmegoldásnak ebben a fázisában a fagráf a következő:



A 3. részfeladatban egész értékű megoldás adódott (maximum érték 180), így ezt az ágat lezárjuk, vagyis ebből nem végzünk elágaztatást. A lezárást a fagráfban a $\perp$ szimbólummal jelöltük. A megoldást azonban tovább kell folytatni, mivel az 1. és a 4. részfeladatokban tört megoldás adódott (aktív részfeladatok), de még elképzelhető ezeken az ágakon olyan integer megoldás, amelyben a célfüggvény maximuma 180 vagy annál nagyobb.

Az elágaztatást a 4. részfeladatnál végezzük, mert az aktív részfeladatok közül itt a legnagyobb a célfüggvény felső korlátja. E részfeladatban az x_1 változó tört, így az elágaztatásban $x_1 = 0$, ill. $x_1 = 1$. A további két részfeladat a következő:

5. részfeladat: $x_3 = 1,\ x_2 = 1,\ x_1 = 0$
Ekkor az FLP feladat a következő:

$$\begin{array}{rcrl} 170 & + & 40x_4 & \rightarrow \text{ max!} \\ & & 3x_4 & \leqq 3 \\ & & 0 \leqq x_4 \leqq 1 & \end{array}$$

Az optimális megoldás:

$$x_1 = 0,\ x_2 = 1,\ x_3 = 1,\ x_4 = 1,\ z = 210$$

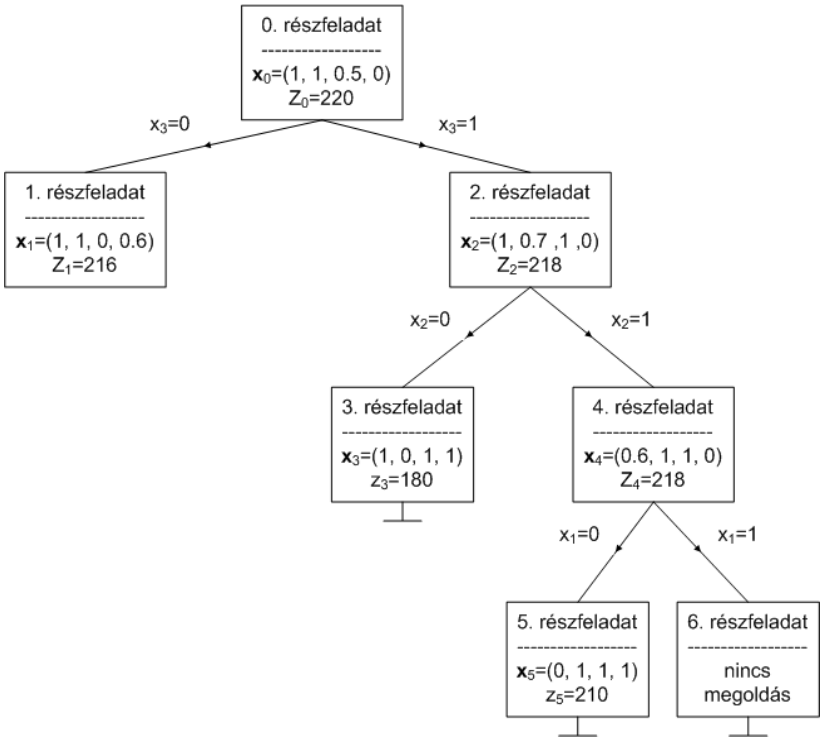
Ebben a részfeladatban egész megoldás adódott, tehát ezt az ágat lezárhatjuk.

6. részfeladat: $x_3 = 1,\ x_2 = 1,\ x_1 = 1$
Ekkor az FLP feladat a következő:

$$\begin{array}{rcrl} 250 & + & 40x_4 & \rightarrow \text{ max!} \\ & & 3x_4 & \leqq -2 \\ & & 0 \leqq x_4 \leqq 1 & \end{array}$$

A folytonos feladatnak nincs lehetséges megoldása, így az integer feladatnak sincs. Ez az ág tehát lezárható.

A feladatmegoldásnak ebben a fázisában a fagráf a következő:



A fagráf azt mutatja, hogy egy aktív részfeladatunk (1. számú) van, amelyben a célfüggvény maximuma nem lehet nagyobb 216-nál. A legjobb integer megoldás eddig 210. Tehát még elképzelhető, hogy az 1. részfeladatban található 210 vagy annál nagyobb célfüggvényértékű integer megoldás. A következő elágaztatás tehát az 1. részfeladatnál lesz az x_4 változó 0, ill. 1 értéke szerint.

7. részfeladat: $x_3 = 0,\ x_4 = 0$
Ekkor az FLP feladat a következő:

$$\begin{aligned} 80x_1 \ &+\ 110x_2 \ \rightarrow\ \text{max!} \\ 5x_1 \ &+\ 7x_2 \ \leqq\ 14 \\ 0 \leqq x_j &\leqq 1,\ \ j = 1, 2 \end{aligned}$$

Az optimális megoldás:

$$x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ x_4 = 0,\ z = 190$$

Ebben a részfeladatban egész megoldás adódott, tehát ezt az ágat lezárhatjuk.

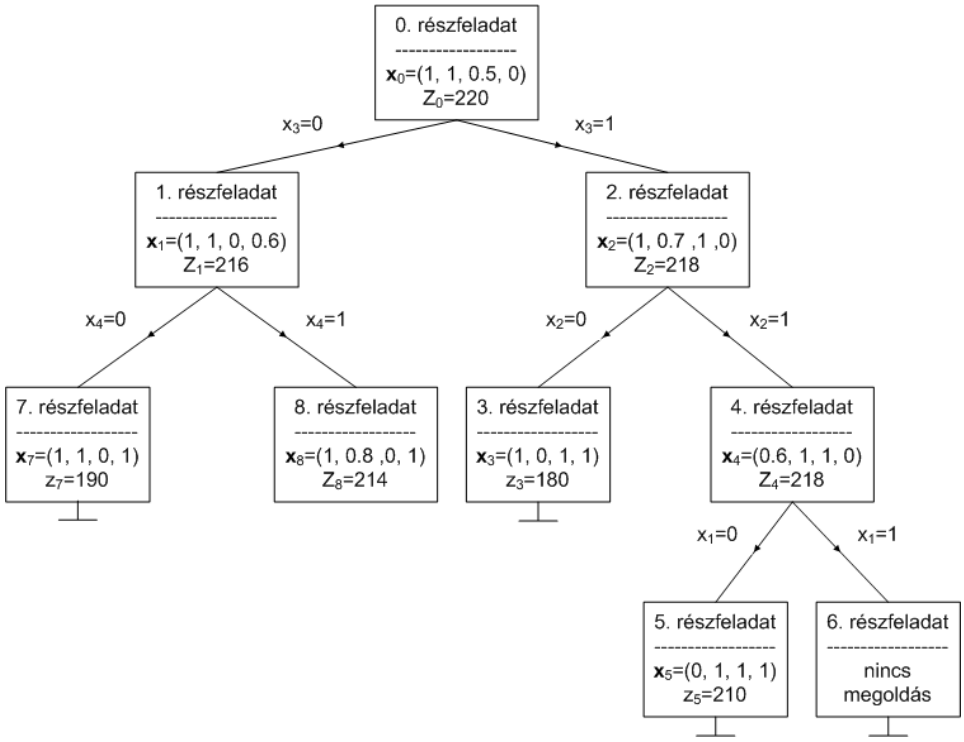
8. részfeladat: $x_3 = 0,\ x_4 = 1$
Ekkor az FLP feladat a következő:

$$\begin{aligned} 40 \ +\ 80x_1 \ &+\ 110x_2 \ \rightarrow\ \text{max!} \\ 5x_1 \ &+\ 7x_2 \ \leqq\ 11 \\ 0 \leqq x_j &\leqq 1,\ \ j = 1, 2 \end{aligned}$$

Az optimális megoldás:

$$x_1 = 1,\ x_2 = \frac{6}{7} = 0.857,\ x_3 = 0,\ x_4 = 1,\ z = 214.29$$

A feladatmegoldásnak ebben a fázisában a fagráf a következő:



A 8. részfeladat aktív és itt a célfüggvénykorlát nagyobb, mint az eddigi legnagyobb egész célfüggvényérték, tehát elképzelhető ezen az ágon 210 vagy annál jobb integer megoldás. Az x_2 szerint ágaztatunk el, a két részfeladat a következő:

9. részfeladat: $x_3 = 0,\ x_4 = 1,\ x_2 = 0$
Ekkor az FLP feladat a következő:

$$\begin{aligned} 40 \ + \ 80x_1 \ &\rightarrow \ \max! \\ 5x_1 \ &\leqq \ 11 \\ 0 \leqq x_1 &\leqq 1 \end{aligned}$$

Az optimális megoldás:

$$x_1 = 1,\ x_2 = 0,\ x_3 = 0,\ x_4 = 1,\ z = 120$$

Ebben a részfeladatban egész megoldás adódott, tehát ezt az ágat lezárhatjuk.

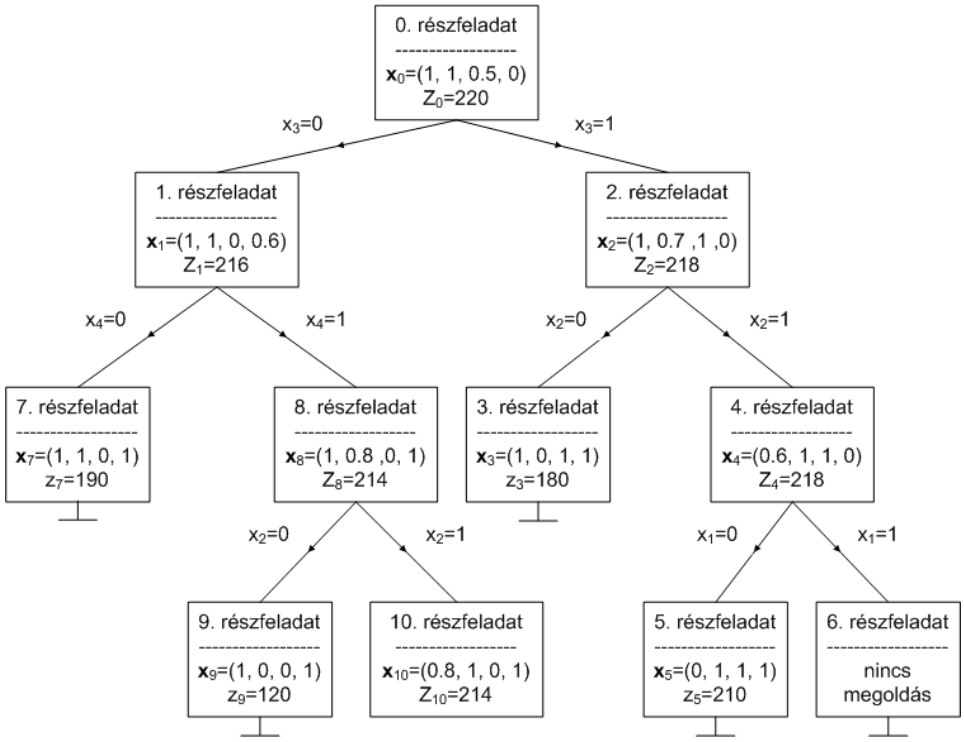
10. részfeladat: $x_3 = 0,\ x_4 = 1,\ x_2 = 1$
Ekkor az FLP feladat a következő:

$$\begin{aligned} 150 \ + \ 80x_1 \ &\rightarrow \ \max! \\ 5x_1 \ &\leqq \ 4 \\ 0 \leqq x_1 &\leqq 1 \end{aligned}$$

Az optimális megoldás:

$$x_1 = \frac{4}{5} = 0.8,\ x_2 = 1,\ x_3 = 0,\ x_4 = 1,\ z = 214$$

A feladatmegoldásnak ebben a fázisában a fagráf a következő:



A fagráfban a 10. részfeladat aktív és a célfüggvénykorlát nagyobb, mint 210, tehát itt kellene elágaztatni. Azonban a 10. részfeladatban már csak egy szabad változó van és tudjuk, hogy az nem lehet 1, így ez az ág is lezárható $x_1 = 0$ értékkel. Ez pedig már integer megoldás 120 célfüggvénnyel. Megállapíthatjuk, hogy az optimális megoldás az 5. részfeladat megoldása, ami

$$x_1 = 0,\ x_2 = 1,\ x_3 = 1,\ x_4 = 1,\ z = 210.$$

A feladat optimális megoldása az eredeti változókkal:

$$y_1 = 1,\ y_2 = 1,\ y_3 = 0,\ y_4 = 1,\ z = 210.$$

A fentiekben tehát bemutattuk a hátizsák feladat Szétválasztás és korlátozás módszerével történő megoldását. Az algoritmus lépései az alábbi pontokban foglalható össze:

1. Megoldjuk a feladathoz rendelt folytonos lineáris programozási feladatot. Ha a megoldás egész, akkor készen vagyunk. Egyéb esetben két új részfeladatot fogalmazunk meg az egyetlen tört értékű változó 0, ill. 1 értéken való rögzítésével.

2. Egy részfeladatot a következő négy esetben **nem** tekintünk **aktív**nak:

   - Ha a részfeladaton már végeztünk elágaztatást, azaz, ha nem a fagráf végső pontján van.
   - Ha a részfeladat megoldása integer (ekkor ezt az ágat lezárhatjuk).
   - Ha a részfeladat nem megoldható (ekkor ezt az ágat lezárhatjuk).
   - Ha a részfeladat célfüggvényértéke (eredeti feladat célfüggvényének felső korlátja) kisebb, mint az egész megoldásokhoz tartozó legnagyobb célfüggvényérték (ekkor ezt az ágat is lezárhatjuk).

3. Kiválasztunk egy aktív részfeladatot és elágaztatunk a törtértékű változó szerint. Azt az aktív részfeladatot választjuk ki, amelynek célfüggvénye a legnagyobb. Az eljárást akkor fejezzük be, ha már nincs aktív részfeladat.

Megjegyezzük, hogy gyorsíthatunk az algoritmuson. Ezt tettük például a 10. részfeladat lezárásával. További hasonló lezárások is eszközölhetők, de ezekre nem térünk ki, mivel a Szétválasztás és korlátozás módszer alapelveit kívántuk bemutatni.

A Szétválasztás és korlátozás módszere egy módszercsalád, a megoldandó feladat jellege szabja meg, hogy milyen elven választjuk meg a szétválasztás szabályát, ill. hogyan határozzuk meg a célfüggvény korlátját. Például egy részfeladatban egy integer változó értéke $x_j = 3.41$, akkor az $x_j \leqq 3$, ill. $x_j \geqq 4$ feltételekkel bontjuk ketté a feladatot. Ezt az elvet követi a következőkben bemutatandó DAKIN algoritmus.

3.2.2. Dakin algoritmus

A DAKIN algoritmus tiszta vagy vegyes integer lineáris programozási feladatot old meg szétválasztás és korlátozás módszerével. Az algoritmus első lépéseként az integer feladathoz rendelt folytonos feladatot oldjuk meg. Ha ez teljesíti az egészértékűségi megkötéseket, akkor készen vagyunk. Egyébként a folytonos megoldásból kiválasztjuk azt a **törtértékű** változót, amelyre egészértékűségi megkötés vonatkozik, legyen ez az $x_r = x_r^B$ változó. Az x_r^B mennyiség egész értékének ($\left[x_r^B\right]$) megfelelően két részfeladatra bontjuk a problémát, egyiket az $x_r \leqq \left[x_r^B\right]$, a másikat az $x_r \geqq \left[x_r^B\right] + 1$ előírásoknak az eredeti feltételekhez való hozzáadásával kapjuk. Ez a felbontás természetes, hiszen egész megoldást nem kapunk a lefelé és a felfelé kerekített egész értékek közötti tartományban. Az új feladatokat vagy az induló táblából kiindulva oldjuk meg szimplex módszerrel vagy az optimális szimplex táblázatba beépítjük az új feltételt és duál módszerrel folytatjuk a megoldást.

Az alábbiakban röviden megmutatjuk, hogyan építhetjük be az optimális szimplex táblába az egyenlőtlenséges új feltételt. Az érdeklődő olvasó a Gazdaságmatematika tananyagban erről részletesen olvashat. Legyen az új feltétel a következő:

$$\mathbf{ax} \leqq b,$$

ahol $\mathbf{a}, \mathbf{x}$ vektorok, b valós szám. Tekintsük az optimális szimplex táblát, benne a $\mathbf{T}$ mátrixot és az $\mathbf{x}^B$ vektort, ez utóbbi az $\mathbf{x}$ megoldásvektor bázisbeli része:

$\mathbf{T}$	$\mathbf{x}^B$
$\ominus$ ... $\ominus$	

A feltétel együtthatóvektorát ($\mathbf{a}$) partícionáljuk bázis ($\mathbf{a}^B$) és nembázis részre ($\mathbf{a}^N$). Az optimális szimplex táblába az új feltételt az alábbi szimplex táblán látható módon építhetjük be. Ha a tábla szegélyeire felírjuk az új feltétel adatait, akkor a számolást könnyebbé tehetjük. A szimplex táblában az új feltételhez számított új sorban lévő $\mathbf{a}^N - \mathbf{a}^B\mathbf{T}$ és $b - \mathbf{a}^B\mathbf{x}^B$ műveletek

elvégzése így kényelmesebbé válik. A feltétel hiányváltozója legyen u, ami bázisváltozóként szerepel a módosított szimplex táblában.

		$\mathbf{a}^N$	b
$\mathbf{a}^B$		$\mathbf{T}$	$\mathbf{x}^B$
	u	$\mathbf{a}^N - \mathbf{a}^B\mathbf{T}$	$b - \mathbf{a}^B\mathbf{x}^B$
		$\ominus \quad \dots \quad \ominus$	

Mivel ez a szimplex tábla duál megengedett, így a duál módszerrel folytathatjuk a megoldást.

Az $\mathbf{ax} \geqq b$ típusú feltételek beépítését (-1)-el való beszorzással vezethetjük vissza a bemutatott $\mathbf{ax} \leqq b$ esetre.

Példa:

Tekintsük az alábbi tiszta integer lineáris programozási feladatot és oldjuk meg Dakin algoritmussal.

$$\begin{array}{rcrcl} 10x_1 & + & 30x_2 & \rightarrow & \max! \\ 2x_1 & + & x_2 & \leqq & 40 \\ 2x_1 & + & 4x_2 & \leqq & 51 \end{array}$$
$$x_1, x_2 \geqq 0, \quad \text{egész}$$

Vezessük be az u_1, u_2 hiányváltozókat, mivel a feltételek együtthatói és a jobboldal egészek, így a hiányváltozókra is érvényesíthetjük az egészértékűséget, azaz a standard alakú LP is tiszta integer feladat lesz, amely a következőképpen írható:

$$\begin{array}{rcrcrcrcl} 10x_1 & + & 30x_2 & & & & & \rightarrow & \max! \\ 2x_1 & + & x_2 & + & u_1 & & & = & 40 \\ 2x_1 & + & 4x_2 & & & + & u_2 & = & 51 \end{array}$$
$$x_1, x_2, u_1, u_2 \geqq 0, \quad \text{egész}$$

Először megoldjuk a folytonos feladatot, a megoldást szimplex módszerrel végezzük, a kezdő és az optimális szimplex tábla a következő:

	x_1	x_2	
u_1	2	1	40
u_2	2	4	51
	10	30	0

	x_1	u_2	
u_1	3/2	-1/4	109/4
x_2	1/2	1/4	51/4
	-10/2	-30/4	-1530/4

A folytonos feladat optimális megoldása (0. részfeladat):

$$x_1 = 0, \; x_2 = 12.75, \; z_0 = 382.5, \; Z_0 = 382$$

A célfüggvény felső korlátját, azaz a lefelé kerekített célfüggvényértéket a Z nagybetűvel fogjuk jelölni. Az x_2 változó értéke tört, így e változó szerint határozzuk meg a két részfeladatot.

1. részfeladat : $x_2 \leqq 12$

$$\begin{array}{rcrcl} 10x_1 & + & 30x_2 & \rightarrow & \max! \\ 2x_1 & + & x_2 & \leqq & 40 \\ 2x_1 & + & 4x_2 & \leqq & 51 \\ & & x_2 & \leqq & 12 \end{array}$$
$$x_1, x_2 \geqq 0$$

Mint említettük, ezt az új feladatot megoldhatjuk szimplex módszerrel az induló szimplex táblájából kiindulva. A gyorsabb módszert javasoljuk, amely szerint az új feltételt az optimális szimplex táblába építünk be. Ezt a beépítést, azaz az új feltételnek megfelelő új sorral való bővítést a következő szimplex tábla mutatja. A feltétel hiányváltozója legyen u_3, amelyre szintén érvényes az egészértékűség.

		0	0	12
		x_1	u_2	
0	u_1	3/2	-1/4	109/4
1	x_2	1/2	1/4	51/4
	u_3	**-1/2**	**-1/4**	**-3/4**
		-10/2	-30/4	-1530/4

Az új szimplex tábla duál megengedett, a duál módszer szerint a pivotelem kiválasztása a következők szerint történik: a megoldásoszlopban megkeressük melyik sorban van negatív szám, ezt választjuk pivotsornak. Példánkban ez az u_3 bázisváltozó sora. Ebben a pivotsorban **negatív pivotelem**et választunk a hányadoskritérium alapján, a vizsgálósorbeli értékek és a pivotsorbeli negatív értékek hányadosát vesszük és ahol ez a hányados a legkisebb ott választunk pivotelemet. Példánkban a $\frac{-\frac{10}{2}}{-\frac{1}{2}} = 10$ és a $\frac{-\frac{30}{4}}{-\frac{1}{4}} = 30$ a hányadosok, ezek közül pedig az első a legkisebb, így a pivotelem $-\frac{1}{2}$ lesz.

	u_3	u_2	
u_1	3	-1	25
x_2	1	0	12
x_1	-2	1/2	3/2
	-10	-10/2	-375

Az 1. részfeladat optimális megoldása:

$$x_1 = 1.5,\ x_2 = 12,\ z_1 = 375,\ Z_1 = 375$$

2. részfeladat : $x_2 \geqq 13$

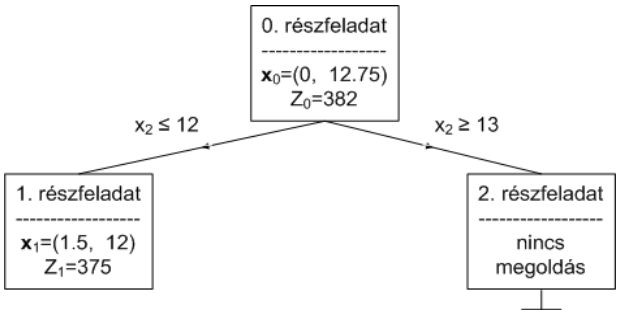
$$\begin{array}{rcrcl} 10x_1 & + & 30x_2 & \rightarrow & \max! \\ 2x_1 & + & x_2 & \leqq & 40 \\ 2x_1 & + & 4x_2 & \leqq & 51 \\ & & x_2 & \geqq & 13 \end{array}$$
$$x_1, x_2 \geqq 0$$

A feltétel $\geqq$ típusú, így a $-x_2 \leqq -13$ feltételt építjük be a szimplex táblázatba. Az új szimplex tábla:

		0	0	-13
		x_1	u_2	
0	u_1	3/2	-1/4	109/4
-1	x_2	1/2	1/4	51/4
	u_3	**1/2**	**1/4**	**-1/4**
		-10/2	-30/4	-1530/4

Az u_3 sorában nincs negatív szám, ez pedig azt jelenti, hogy a 2. részfeladatnak nincs lehetséges megoldása, így az integer feladatnak sincs, tehát ez az ág lezárható.

A Dakin módszernél is célszerű az egyes lépések eredményét egy fagráfon szemléltetni. A megoldás ezen fázisában az eredményeket az alábbi fagráf mutatja:



Mivel csak az 1. részfeladat aktív, ezért az elágaztatást ennél a részfeladatnál kell végrehajtani, mégpedig az $x_1 = 1.5$ törtmegoldást figyelembe véve. A két feladat új feltétele: $x_1 \leqq 1$, ill. $x_1 \geqq 2$.

3. részfeladat : $x_1 \leqq 1$

$$\begin{array}{rcrcl} 10x_1 & + & 30x_2 & \rightarrow & \max! \\ 2x_1 & + & x_2 & \leqq & 40 \\ 2x_1 & + & 4x_2 & \leqq & 51 \\ & & x_2 & \leqq & 12 \\ x_1 & & & \leqq & 1 \end{array}$$

$$x_1, x_2 \geqq 0$$

Az új szimplex táblát az 1. részfeladat optimális szimplex táblájából nyerhetjük, a következőkben a szegélyek felrajzolásától eltekintünk, mivel az új feltételek nagyon speciálisak, így azok nélkül is egyszerűen számolható az új sor, ennek átgondolását az olvasóra bízzuk. Az új szimplex tábla a következő:

	u_3	u_2	
u_1	3	-1	25
x_2	1	0	12
x_1	-2	1/2	3/2
u_4	**2**	**-1/2**	**-1/2**
	-10	-10/2	-375

A duál módszer végrehajtása után az alábbi optimális szimplex tábla adódik:

	u_3	u_4	
u_1	-1	-2	26
x_2	1	0	12
x_1	0	1	1
u_2	-4	-2	1
	-30	-10	-370

A 2. részfeladat optimális megoldása (egész megoldás adódott):

$$x_1 = 1,\ x_2 = 12,\ z_3 = 370.$$

4. részfeladat : $x_1 \geqq 2$

Az új szimplex tábla, amely az 1. részfeladat optimális szimplex táblájából építhető fel:

	u_3	u_2	
u_1	3	-1	25
x_2	1	0	12
x_1	-2	1/2	3/2
u_4	**-2**	**1/2**	**-1/2**
	-10	-10/2	-375

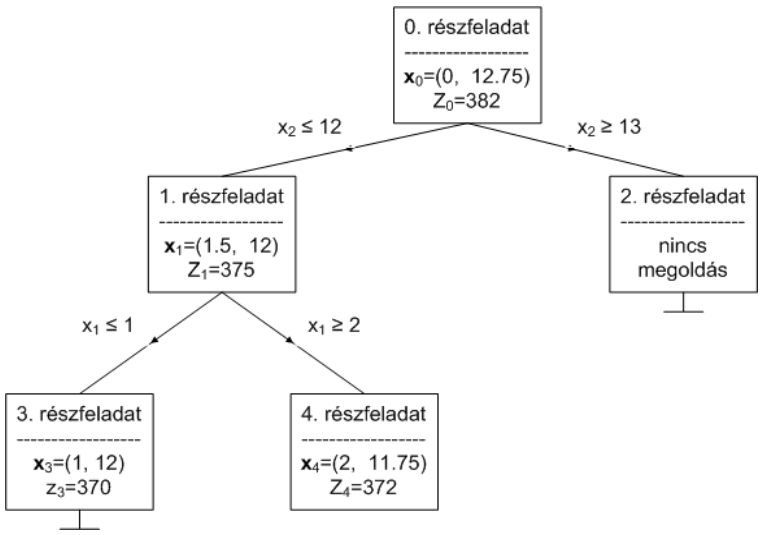
A duál módszer végrehajtása után az alábbi optimális szimplex tábla adódik:

	u_4	u_2	
u_1	3/2	-1/4	24.15
x_2	1/2	1/4	11.75
x_1	-1	0	2
u_3	-1/2	-1/4	0.25
	-10/2	-30/4	-372.5

A 4. részfeladat optimális megoldása:

$$x_1 = 2,\ x_2 = 11.75,\ z_4 = 372.5,\ Z_4 = 372$$

A megoldás ezen fázisában az eredményeket az alábbi fagráf mutatja:



Mivel egyetlen aktív (4. számú) részfeladatunk van és annál a célfüggvénykorlát nagyobb mint az integer megoldáshoz tatozó célfüggvény maximum, ezért ezt az aktív részfeladatot kell tovább bontani, mégpedig az $x_2 = 11.75$ törtváltozónak megfelelően az $x_2 \leqq 11$ és az $x_2 \geqq 12$ részfeladatokra.

5. részfeladat : $x_2 \leqq 11$

Az új szimplex táblát a 4. részfeladat optimális szimplex táblájának módosításával állítjuk elő:

	u_4	u_2	
u_1	3/2	-1/4	24.25
x_2	1/2	1/4	11.75
x_1	-1	0	2
u_3	-1/2	-1/4	0.25
u_5	**-1/2**	**-1/4**	**-0.75**
	-10/2	-30/4	-372.5

A duál módszer végrehajtása után az alábbi optimális szimplex tábla adódik:

	u_5	u_2	
u_1	3	-1	22
x_2	1	0	11
x_1	-2	1/2	3.5
u_3	-1	0	1
u_4	–2	1/2	0.5
	-10	-10/2	-365

Az 5. részfeladat optimális megoldása:

$$x_1 = 3.5,\ x_2 = 11,\ z_5 = 365,\ Z_5 = 365$$

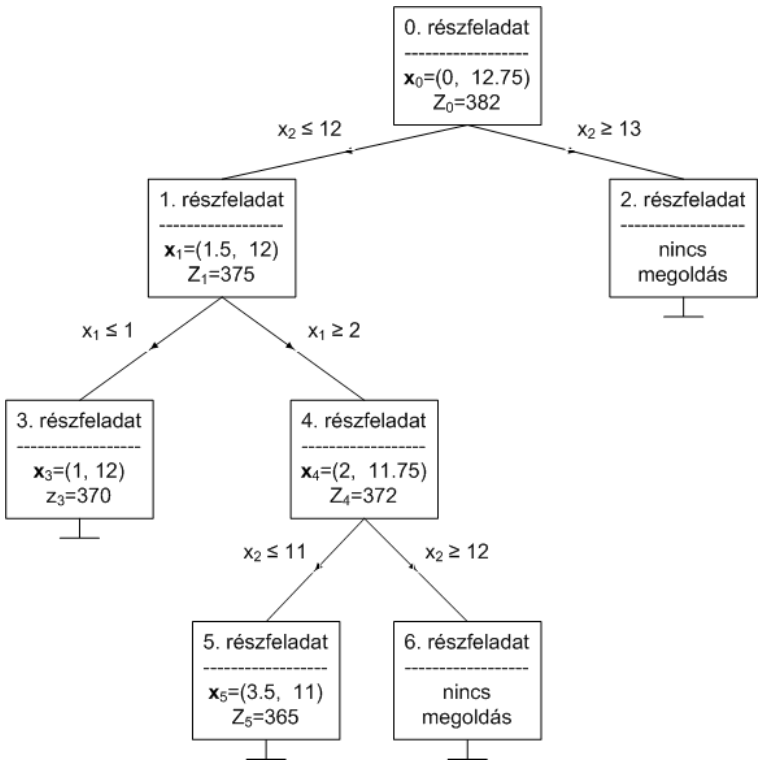
6. részfeladat : $x_2 \geqq 12$

Az új szimplex tábla, amely szintén a 4. részfeladat optimális szimplex táblájából építhető fel:

	u_4	u_2	
u_1	3/2	-1/4	24.15
x_2	1/2	1/4	11.75
x_1	-1	0	2
u_3	-1/2	-1/4	0.25
u_5	**1/2**	**1/4**	**-0.25**
	-10/2	-30/4	-372.5

Az u_5 változó sorában nem választható negatív pivotelem, ezért a 6. részfeladatnak nincs lehetséges megoldása, így az integer feladatnak sincs, ezért ez az ág lezárható.

Most tekinsük az eddigi munkánk során kapott fagráfot.



Mivel egyetlen aktív részfeladatunk van (5. sorszámú) és ennél a célfüggvénykorlát (365) kisebb mint az integer megoldáshoz tatozó célfüggvény maximum (370), ezért ezt az aktív részfeladatot nem érdemes tovább bontani, tehát az algoritmus befejeződött.

Az eredeti feladat optimális megoldása:

$$x_1 = 1,\ x_2 = 12,\ z_{\max} = 370.$$

3.3. Vágási módszerek (Cutting Plane)

3.3.1. A vágási módszer alapgondolata

Ebben a fejezetben megismerkedünk az integer lineáris programozási feladatok megoldására szolgáló vágási módszercsaláddal.

Az alábbiakban röviden ismertetjük a vágási mószerek alapgondolatát. Legyen adott egy integer lineáris programozási feladat. Ehhez hozzárendeljük az egészértékűség megkötése nélkül nyert folytonos lineáris programozási feladatot. A folytonos lineáris programozási feladat feltételi halmaza egy konvex poliéder. Az integer lineáris programozási feladat feltételi halmaza pedig a konvex poliéder diszkrét pontjai, amelyeket **rácspontoknak** nevezünk.

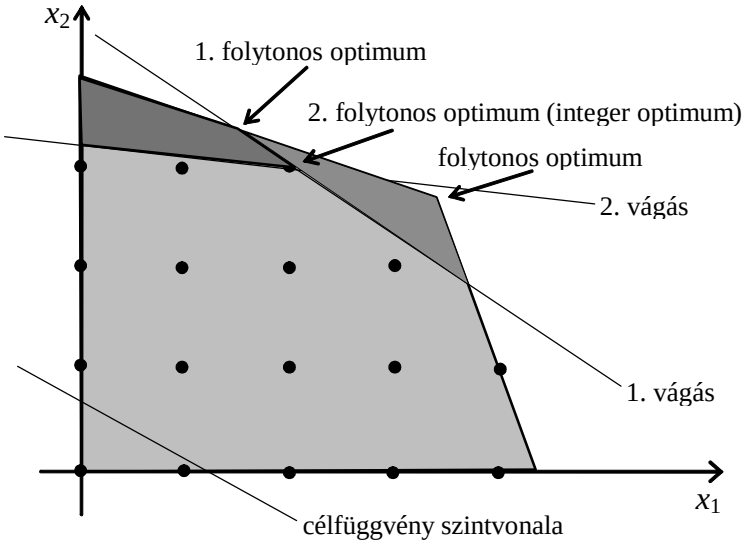
A vágási módszer lényege a következő:

Induló lépésként a folytonos lineáris programozási feladatot megoldjuk. Mint tudjuk az optimális megoldás a konvex poliéder egy extremális pontja (**csúcspont**) lesz.

Ha ez az optimális csúcspont egyben rácspont is, akkor ez az optimális megoldás egyben az integer lineáris programozási feladat optimális megoldása is.

Ha viszont a folytonos optimum nem rácspont, akkor a **folytonos** lineáris programozási feladat lehetséges megoldásainak halmazát szűkítjük úgy, hogy az integer lineáris programozási feladat feltételi halmaza ne változzon. Ezt egy új feltétel bevezetésével valósítjuk

meg, amely **levág** egy, az optimumot is tartalmazó szeletet a folytonos tartományból, azaz a konvex poliéderből. Innen a módszer elnevezése. A következő lépésben újra megoldjuk a módosított folytonos feladatot és ezt ismételjük mindaddig, amíg a folytonos feladat optimális megoldása rácspont nem lesz. Az elmondottakat szemlélteti az alábbi ábra. Az ábrában a közepesen árnyalt tartományt az 1. vágással, a sötétebben árnyalt tartományt pedig a 2. vágással metszettük le. A világosabban árnyalt tartománybeli optimális csúcspont egyben rácspont is.



3.3.2. Gomory vágás

Tekintsük a tiszta integer lineáris programozási feladat standard alakját:

$$\begin{aligned} \mathbf{cx} &\rightarrow \text{min!} \\ \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geqq \mathbf{0}, \text{ egész} \end{aligned}$$

Megoldjuk a fenti feladatnak megfelelő folytonos feladatot szimplex, duál vagy criss-cross módszerek valamelyikével. Ha e folytonos feladat optimális megoldásában minden döntési változó egész, akkor az optimum egyben integer optimum is. Ha nem, akkor tekintsük az optimális szimplex táblának azt a sorát, amelyben a megoldás tört érték, legyen ez az r indexhez tartozó sor. A szimplex táblának ezt a sorát **forrás sor**nak nevezzük, mert a vágási feltételt ebből a sorból felírható egyenlet segítségével alkotjuk meg.

Az optimális szimplex tábla legyen a következő, amelyben a forrás sort fel is tüntettük, a táblában az x_r^B értéknek törtek kell lennie.

	x_j	
x_r	t_{rj}	x_r^B
$\ominus$	$\ldots$	$\ominus$

Az eredeti feladat r-edik egyenletének a pivótálások utáni transzformált változata, a forrás sorból olvasható ki, az egyenlet a következő alakban írható:

$$x_r + \sum_{j \in NB} t_{rj} x_j = x_r^B,$$

ahol NB a nembázis változók indexhalmazát jelenti, az x_r és az x_j ($j \in NB$) az egyenletben szereplő változók. Az egyenletben szereplő együtthatókat (t_{rj}) és a jobboldalt (x_r^B) írjuk fel egy egész szám és egy valódi tört összegeként.

Mielőtt ezt megtennénk definiáljuk egy tetszőleges valós szám egész részét. Legyen w egy valós szám. A w szám egész része alatt a w-nél nem nagyobb legnagyobb egész számot értjük és $[w]$-vel jelöljük. Egy szám egész része a számegyenesen ábrázolva a számtól balra lévő első egész szám. Ennek megfelelően a w szám tört része $w - [w]$ és jelöljük a tört részt f betűvel. Tehát a tört rész mindig 1-nél kisebb nemnegatív szám.

Például:
$w = +2.72$ esetén $[w] = 2, \quad f = 0.72,$
$w = -2.72$ esetén $[w] = -3, \; f = 0.28,$
$w = -4.00$ esetén $[w] = -4, \; f = 0.$

Ennek megfelelően a forrássorból származtatott egyenlet az alábbiak szerint írható:

$$x_r + \sum_{j \in NB} \left([t_{rj}] + f_{rj} \right) x_j = \left[x_r^B \right] + f_r,$$

ahol $[t_{rj}]$ a t_{rj} táblabeli értékek egész része, az f_{rj} a t_{rj} táblabeli értékek tört része, hasonlóan az $\left[x_r^B \right]$ az x_r^B táblabeli érték egész része, az f_r az x_r^B táblabeli érték tört része. A fentiek miatt igaz, hogy $0 < f_r < 1$ és $0 \leqq f_{rj} < 1$.

Rendezzük át ezt az egyenletet úgy, hogy a baloldalon a felbontásból kapott egész adatok, a jobboldalon pedig a tört adatok legyenek:

$$x_r - \left[x_r^B \right] + \sum_{j \in NB} [t_{rj}] \, x_j = f_r - \sum_{j \in NB} f_{rj} x_j,$$

Ha az egyenletben szereplő összes változóról feltesszük az egészértékű megkötést, akkor az egyenlet baloldalának értéke egész szám. Mivel az egyenlőségnek teljesedni kell, ezért az egyenlet jobboldalának is egésznek kell lennie. Vizsgáljuk meg közelebbről a jobboldalt, amely a következő:

$$f_r - \sum_{j \in NB} f_{rj} x_j.$$

A szummás tag mindegyik tagja nemnegatív (mivel a tényezők is nemnegatívok), így maga a szumma is az, ebből pedig következik, hogy a fenti jobboldal kisebb vagy egyenlő mint f_r, tehát írható, hogy

$$f_r - \sum_{j \in NB} f_{rj} x_j \leqq f_r < 1.$$

Tehát a jobboldalról eddig az derült ki, hogy $f_r - \sum_{j \in NB} f_{rj} x_j < 1$, amennyiben kikötjük az egészértékűséget. Ugyanakkor a jobboldalnak egyenlőnek kell lennie a baloldallal, amiről pedig tudjuk, hogy egész szám. Ezek alapján csak a $0, -1, -2, \ldots$ egészek jöhetnek szóba. Annak tehát, hogy a feladat megoldása egész legyen, **szükséges feltétele**, hogy a jobboldal nemnegatív legyen, azaz képletben

$$f_r - \sum_{j \in NB} f_{rj} x_j \leqq 0.$$

Ezt a feltételt hívjuk **Gomory vágás**nak. Egy nemnegatív u hiányváltozó bevezetésével a következőképpen is írhatjuk a vágást:

$$-\sum_{j\in NB} f_{rj}x_j + u = -f_r.$$

Ebből adódik, hogy $-u = f_r - \sum_{j\in NB} f_{rj}x_j$, tehát a $-u$ nem más mint a forrás sorból levezetett egyenlet baloldala, amely az egészértékűséget előírva csak $0, -1, -2, \ldots$ értékeket vehet fel, ebből pedig az következik, hogy a bevezetett nemnegatív u hiányváltozóra is ugyanazok az előírások érvényesek, mint az x_j döntési változókra, azaz $u \geqq 0$ és egész, így a vágással kiegészült feladat is tiszta integer feladat marad. Mivel az optimális megoldásban minden nembázis változó zérus, így a vágási feltételből az következik, hogy $u = -f_r$, ami ellentmondás, tehát az optimális megoldás (optimális csúcspont) nem lehet lehetséges megoldása a vágási feltétellel kiegészített folytonos feladatnak, azaz a vágás levágja az optimális csúcspontot. Emellett még más pontokat is levág a folytonos tartományból, de rácspontot semmiképpen nem vág le, tehát csak a folytonos feladat tartományát szűkítjük, az integer feladat tartománya változatlan marad minden vágás esetén.

A későbbiekben még fogunk látni más vágásokat is, azért, hogy ezeket egységes szerkezetben lássuk, célszerű átírni a vágást az alábbi alakra:

$$-\sum_{j\in NB} q_j x_j \leqq -f_r,$$

ahol

$$q_j = f_{rj}$$

A vágás tehát olyan új egyenlőtlenséges feltétel, amelyben csak nembázis változók szerepelnek.

Példa:
Oldjuk meg az alábbi programozási feladatot Gomory vágási módszerrel!

$$\begin{array}{rcrcl} x_1 & + & 3x_2 & \rightarrow & \max! \\ 2x_1 & + & x_2 & \leqq & 40 \\ x_1 & + & 2x_2 & \leqq & 51/2 \end{array}$$
$$x_1, x_2 \geqq 0, \quad \text{egész}$$

A második feltétel jobboldala tört, így a bevezetendő u_2 hiányváltozóra nem írhatnánk elő az egészértékűséget. Ezért először szorozzuk be a második egyenlőtlenséget 2-vel, ekkor mindkét hiányváltozóra előírható az egészértékűség, azaz tiszta integer feladatot kapunk:

$$\begin{array}{rcrcrcrcl} x_1 & + & 3x_2 & & & & & \rightarrow & \max! \\ 2x_1 & + & x_2 & + & u_1 & & & = & 40 \\ 2x_1 & + & 4x_2 & & & + & u_2 & = & 51 \end{array}$$
$$x_1, x_2, u_1, u_2 \geqq 0, \quad \text{egész}$$

A folytonos feladatot megoldjuk, az induló és az optimális szimplex táblát közöljük csupán, amelyek az alábbiak:

	x_1	x_2	
u_1	2	1	40
u_2	2	4	51
	1	3	0

	x_1	u_2	
u_1	3/2	-1/4	109/4
x_2	1/2	1/4	51/4
	-1/2	-3/4	-153/4

Mivel az optimális megoldás nem integer, ezért vágni kell. Válasszuk ki forrás sornak az u_1 bázisváltozó sorát, amelyből az alábbi vágási feltétel írható fel:

$$-1/2x_1 - 3/4u_2 \leqq -1/4$$

A vágási feltétel együtthatói és jobboldala tehát a forrás sorbeli adatok törtrészei lesznek ellenkező (negatív) előjellel. A vágási feltétel az u_3 hiányváltozó bevezetésével:

$$-1/2x_1 - 3/4u_2 + u_3 = -1/4$$

Ezt a vágást, mint új feltételt a korábban megismert módon építjük be az optimális szimplex táblába.

			-1/2	-3/4	-1/4
			x_1	u_2	
0	u_1		3/2	-1/4	109/4
0	x_2		1/2	1/4	51/4
	u_3		**-1/2**	**-3/4**	**-1/4**
			-1/2	-3/4	-153/4

Mivel a vágás feltétele speciális, nem tartalmaz ugyanis bázisváltozót, ezért nagyon egyszerűen beépíthetjük a vágást a szimplex táblába. Nem kell mást tenni, mint a forrás sorban lévő elemek törtrészének (-1)-szeresét beírni az új sorba, a szegélyek használata így feleslegessé válik.

A duál módszer végrehajtása után az alábbi optimális szimplex tábla adódik:

	u_3	u_2	
u_1	3	-5/2	53/2
x_2	1	-1/2	25/2
x_1	-2	3/2	1/2
	-1	0	-38

Mivel az optimális megoldás nem egész, ezért újabb vágást kell alkalmazni, legyen a forrás sor az x_1 változó sora, ekkor a második vágási feltétel

$$-1/2u_2 + u_4 = -1/2$$

Ezt az alábbi módon építhetjük be az előző optimális szimplex táblába:

	u_3	u_2	
u_1	3	-5/2	53/2
x_2	1	-1/2	25/2
x_1	-2	3/2	1/2
u_4	**0**	**-1/2**	**-1/2**
	-1	0	-38

A duál módszer végrehajtása: Itt most nem kapunk egyetlen pivotálással optimális szimplex táblát, de a szimplex tábla duál megengedett marad. A pivotálás során keletkező szimplex tábla és az optimális szimplex tábla az alábbi:

	u_3	u_4	
u_1	3	-5	29
x_2	1	-1	13
x_1	-2	3	-1
u_2	0	-2	-1
	-1	0	-38

	x_1	u_4	
u_1	3/2	-1/2	55/2
x_2	1/2	1/2	25/2
u_3	-1/2	-3/2	1/2
u_2	0	-2	1
	-1/2	-3/2	-75/2

Most sem kaptunk integer optimumot, ezért a harmadik vágást is végre kell hajtani. Legyen a forrás sor az x_2 változó sora, amelyből a harmadik vágás

$$-1/2x_1 - 1/2u_4 + u_5 = -1/2$$

A harmadik vágás beépítése az előző optimális szimplex táblába:

	x_1	u_4	
u_1	3/2	-1/2	55/2
x_2	1/2	1/2	25/2
u_3	-1/2	-3/2	1/2
u_2	0	-2	1
u_5	**-1/2**	**-1/2**	**-1/2**
	-1/2	-3/2	-75/2

A duál módszer végrehajtása után az alábbi optimális szimplex tábla adódik:

	u_5	u_4	
u_1	3	-2	26
x_2	1	0	12
u_3	-1	-1	1
u_2	0	-2	1
x_1	-2	1	1
	-1	-1	-37

A harmadik vágás után a folytonos optimum egészre adódott, így befejezzük az eljárást. Az eredeti feladat optimális megoldása:

$$x_1 = 1,\ x_2 = 12,\ z_{\max} = 37.$$

Végezetül az algoritmus szemléltetése végett nézzük meg geometriailag is a megoldás menetét. Ahhoz, hogy szemléltethessük a vágásokat az (x_1, x_2) síkon, a vágási feltételeket az x_1, x_2 ismeretlenekkel kell felírni.

1. vágás eredeti alakja: $-1/2x_1 - 3/4u_2 \leqq -1/4$

Az eredeti feladat 2. feltételi egyenletéből, a $2x_1 + 4x_2 + u_2 = 51$ egyenletből, az u_2-t kifejezve és behelyettesítve a fenti vágási feltételbe, az 1. vágásra az alábbi ekvivalens vágási feltételt kapjuk, amelyet már ábrázolhatunk az (x_1, x_2) síkon:

1. vágás ábrázolható alakja: $x_1 + 3x_2 \leqq 38$.

2. vágás eredeti alakja: $-1/2u_2 \leqq -1/2$

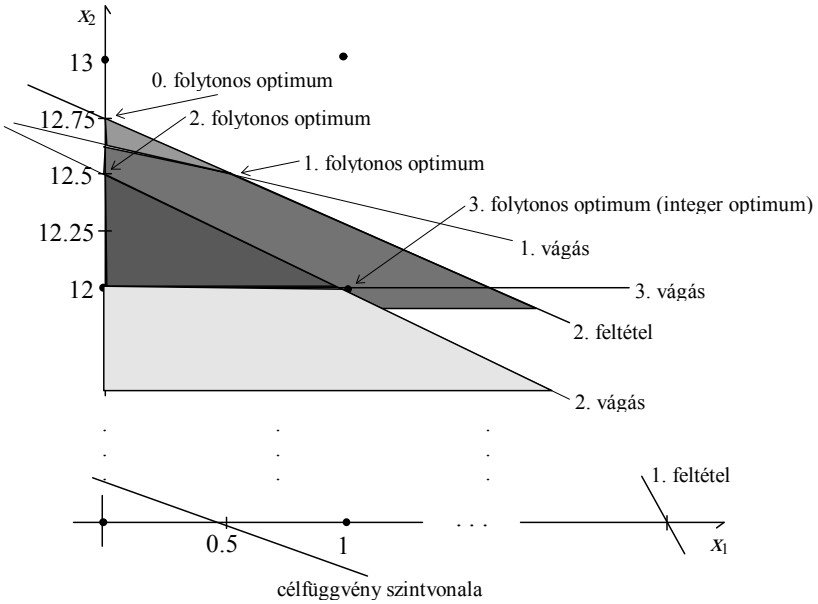
2. vágás ábrázolható alakja: $x_1 + 2x_2 \leqq 25$.

3. vágás eredeti alakja: $-1/2x_1 - 1/2u_4 \leqq -1/2$

Az u_4 változó a $-1/2u_2 + u_4 = -1/2$ egyenletből nyerhető, az u_2 változó pedig a $2x_1 + 4x_2 + u_2 = 51$ egyenletből, ezeket figyelembe véve kapjuk a 3. vágás ábrázolható feltételét.

3. vágás ábrázolható alakja: $x_2 \leqq 12$.

A vágások menetét szemlélteti az alábbi ábra, a folytonos feladatoknak csak azon résztartományát mutattuk, ahol a vágások elhelyezkednek. A vágásokkal lemetszett tartományokat különböző árnyalással szemléltettük.



Az alábbiakban javaslatot teszünk az egyes lépések gyorsabb végrehajtására. Célszerű a célfüggvény információit tartalmazó alsó sort a szimplex tábla első sorába helyezni. Ekkor a vágások azonnal beépíthetők a szimplex tábla alsó sorába, így nem kell mégegyszer felírni a szimplex táblát, kihagyva a helyet vágások beépítéséhez.

A megoldás ilyenfajta végrehajtását az alábbiakban mutatjuk be:

Induló tábla:

	x_1	x_2	
	1	3	0
u_1	2	1	40
u_2	2	4	51

Az optimális szimplex tábla, amelybe azonnal beépíthető az első vágás. A forrás sor (az u_1 bázisváltozó sora) adatainak tört részét kell az alsó sorba írni és folytatható a megoldás duál módszerrel.

	x_1	u_2	
	-1/2	-3/4	-153/4
u_1	3/2	-1/4	109/4
x_2	1/2	1/4	51/4
u_3	**-1/2**	**-3/4**	**-1/4**

Az optimális szimplex tábla, a második vágás beépítése és pivotálás duál módszerrel:

	u_3	u_2	
	-1	0	-38
u_1	3	-5/2	53/2
x_2	1	-1/2	25/2
x_1	-2	3/2	1/2
u_4	**0**	**-1/2**	**-1/2**

	u_3	u_4	
	-1	0	-38
u_1	3	-5	29
x_2	1	-1	13
x_1	-2	3	-1
u_2	0	-2	-1

Az optimális szimplex tábla, a harmadik vágás beépítése és pivotálás duál módszerrel:

	x_1	u_4	
	-1/2	-3/2	-75/2
u_1	3/2	-1/2	55/2
x_2	1/2	1/2	25/2
u_3	-1/2	-3/2	1/2
u_2	0	-2	1
u_5	**-1/2**	**-1/2**	**-1/2**

Az optimális szimplex tábla:

	u_5	u_4	
	-1	-1	-37
u_1	3	-2	26
x_2	1	0	12
u_3	-1	-1	1
u_2	0	-2	1
x_1	-2	1	1

Az algoritmus befejeződött, mert egész megoldást kaptunk.

Végezetül néhány megjegyzést teszünk a Gomory vágással kapcsolatban. Láttuk, hogy a forrás sor meghatározása nem egyértelmű, így más-más vágásokkal juthatunk az optimális megoldáshoz. Kívánatos lenne olyan vágásokat eszközölni, amely a legmélyebben belevág a folytonos halmazba. Ahhoz, hogy megállapíthassuk melyik a legmélyebb vágás, definiálnunk kell azt, hogy két vágás közül melyik a mélyebb. Tekintsünk egy optimális szimplex táblában két forrás sort, legyenek ezek az r-edik és az s-edik sor, ezekben felírt vágások:

$$\sum_{j \in NB} (-f_{rj}) x_j \leqq -f_r,$$

és

$$\sum_{j \in NB} (-f_{sj}) x_j \leqq -f_s.$$

Az r-edik vágás akkor mélyebb az s-edik vágásnál, ha $f_r \geqq f_s$ és $f_{rj} \leqq f_{sj}$ minden $j \in NB$ esetén fennáll, de legalább egy esetben a szigorú egyenlőtlenség áll fenn.

A mélység ezen definíciója sok esetben nem tud dönteni két vágás közül, ezért inkább az alábbi két tapasztalati módszer szerint döntenek a forrás sorról. Az egyik szerint azt a sort választják, ahol a megoldás törtrésze a legnagyobb, azaz

$$\max_{i \in BT} \{f_i\}$$

vagy pedig az alábbi szerint döntenek

$$\max_{i \in BT} \left\{ \frac{f_i}{\sum_{j \in NB} f_{ij}} \right\}$$

ahol $BT = \left\{ i \in B : x_i^B \text{ tört} \right\}$ és B a bázisváltozók indexét jelenti.

Az utóbbi módszer a hatékonyabb, mivel sokkal közelebb van az eredeti mélységi definícióhoz.

3.3.3. Teljesen egész primál vágás

Tekintsük a **normál** tiszta integer lineáris programozási feladatot. Tegyük fel, hogy a feladat összes együtthatója és a jobboldala **egész szám**. A normál feladat azt jelenti, hogy maximum feladatról van szó, minden feltétel $\leqq$ típusú és minden jobboldal nemnegatív. Az induló szimplex tábla tehát primál megengedett és minden adata egész szám.

Az induló szimplex táblában a szimplex módszer alapján kiválasztjuk a pivotelemet. Ha a pivotelem értéke 1, akkor a pivotálás elvégzése utáni szimplex tábla minden eleme egész marad. Amennyiben a pivotelem egynél nagyobb egész, akkor nem végezzük el a pivotálást, hanem meghatározunk egy vágást, amelyet beépítünk a szimplex táblába és utána végezzük el a pivotálást.

A vágást az alábbiak szerint határozzuk meg. Tekintsük forrás sornak a pivotsor jelöltet, legyen ez az x_r változóhoz tartozó sor, amelyből a szokásos módon írható fel a megfelelő egyenlet:

$$x_r + \sum_{j \in NB} t_{rj} x_j = x_r^B.$$

Osszuk el az egyenletet a $t_{rs} > 1$ pivotelem jelölttel és írjuk fel az osztás után kapott egyenletet az egész és a tört részekre bontással:

$$\frac{1}{t_{rs}} x_r + \sum_{j \in NB} \left(\left[\frac{t_{rj}}{t_{rs}} \right] + f_{rj} \right) x_j = \left[\frac{x_r^B}{t_{rs}} \right] + f_r.$$

Rendezzük a fenti egyenletet az alábbi formára:

$$\sum_{j \in NB} \left[\frac{t_{rj}}{t_{rs}} \right] x_j - \left[\frac{x_r^B}{t_{rs}} \right] = f_r - \sum_{j \in NB} f_{rj} x_j - \frac{1}{t_{rs}} x_r.$$

Mivel az egyenletben szereplő összes változó nemnegatív, így a jobboldal kisebb vagy egyenlő mint f_r, tehát írható, hogy

$$\sum_{j \in NB} \left[\frac{t_{rj}}{t_{rs}} \right] x_j - \left[\frac{x_r^B}{t_{rs}} \right] \leqq f_r < 1.$$

Ha megköveteljük a változók egészértékűségét, akkor az egyenlőtlenség baloldalának egésznek kell lennie, de az egyenlőtlenség miatt csak nempozitív $(0, -1, -2, \dots)$ lehet a baloldal, így az egészértékűség **szükséges feltétele** a következő egyenlőtlenség:

$$\sum_{j \in NB} \left[\frac{t_{rj}}{t_{rs}} \right] x_j \leqq \left[\frac{x_r^B}{t_{rs}} \right]$$

Ezt az egyenlőlenséget nevezzük **teljesen egész primál vágás**nak. Az u nemnegatív hiányváltozó bevezetésével a vágás:

$$\sum_{j \in NB} \left[\frac{t_{rj}}{t_{rs}} \right] x_j + u = \left[\frac{x_r^B}{t_{rs}} \right].$$

Könnyen ellenőrizhető, hogy u is csak egész lehet, így a feladat tiszta integer feladat marad. A vágás beépítése után kapott szimplex táblában újra pivotelemet választunk, de az eredeti pivotoszlopban. Az is könnyen ellenőrizhető, hogy az új pivotelem sora az újonnan beépített sor lesz, a pivotelem értéke pedig 1 lesz.

E módszernél tehát nem a folytonos feladat optimális megoldása ismeretében kezdjük el a vágásokat, hanem minden olyan pivotáláskor vágunk, amikor a pivotelem értéke nem 1.

Példa:
Oldjuk meg az alábbi feladatot teljesen egész primál vágás módszerrel!

$$\begin{array}{rcrcl} x_1 & + & 3x_2 & \rightarrow & \max! \\ 2x_1 & + & x_2 & \leqq & 40 \\ x_1 & + & 2x_2 & \leqq & 51/2 \end{array}$$
$$x_1, x_2 \geqq 0, \quad \text{egész}$$

A második feltétel 2-vel történő beszorzása után olyan normál feladatot kapunk, amelyben minden adat egész, tehát a módszer alkalmazható. A hiányváltozókkal felírt tiszta integer feladat:

$$\begin{array}{rcrcrcrcl} x_1 & + & 3x_2 & & & & & \rightarrow & \max! \\ 2x_1 & + & x_2 & + & u_1 & & & = & 40 \\ 2x_1 & + & 4x_2 & & & + & u_2 & = & 51 \end{array}$$
$$x_1, x_2, u_1, u_2 \geqq 0, \quad \text{egész}$$

Az induló szimplex tábla

	x_1	x_2	
u_1	2	1	40
u_2	2	4	51
	1	3	0

A második oszlopban választunk pivotelemet, értéke 4. Mivel ez a pivotelem jelölt nem 1, így azonnal vágni kell. A pivotsor jelölt az u_2 változó sora, tehát ez lesz a forrássor. A vágás

$$\left[\frac{2}{4} \right] x_1 + \left[\frac{4}{4} \right] x_2 + u_3 = \left[\frac{51}{4} \right].$$

Mivel a vágás itt is csak nembázis változókat tartalmaz, ezért egyszerűen beépíthető a szimplex táblázatba az új feltétel, itt a **pivotelemmel való osztás utáni egészeket** kell az új sorba beírni, az új szimplex tábla

	x_1	x_2	
u_1	2	1	40
u_2	2	4	51
u_3	0	[1]	12
	1	3	0

A pivotálás végrehajtása után az alábbi szimplex tábla adódik:

	x_1	u_3	
u_1	2	-1	28
u_2	2	-4	3
x_2	0	1	12
	1	-3	-36

A tábla nem optimális, most a pivot jelölt az első oszlop, a pivotelem jelölt pedig az u_2 sorbeli 2. Mivel a pivotelem jelölt nem 1, így vágni kell. A vágás

$$1x_1 - 2x_2 + u_4 = 1,$$

amelynek beépítése után kapott szimplex tábla

	x_1	u_3	
u_1	2	-1	28
u_2	2	-4	3
x_2	0	1	12
u_4	[1]	-2	1
	1	-3	-36

A pivotálás végrehajtása után az alábbi szimplex tábla adódik:

	u_4	u_3	
u_1	-2	3	26
u_2	-2	0	1
x_2	0	1	12
x_1	1	-2	1
	-1	-1	-37

Az eljárás befejeződött, mert optimál szimplex tábla adódott. Az eredeti feladat optimális megoldása:

$$x_1 = 1,\ x_2 = 12,\ z_{\max} = 37.$$

Ennél a megoldási módszernél is javasoljuk a vizsgálósornak (alsó sor) az első sorban történő szerepeltetését.

3.3.4. Vegyes vágás

Az előző két alfejezetben tiszta integer lineáris programozási feladatokra vonatkozó vágásokat ismertettünk. Ebben az alfejezetben vegyes integer lineáris programozási feladatokra vonatkozó vágást, az ún. vegyes vágást mutatjuk be.

Hasonlóan a Gomory vágásnál elmondottakhoz, először meghatározzuk a folytonos feladat optimális megoldását és az optimális szimplex táblát. Legyen az x_r bázisváltozó olyan, amelyre elő van írva az egészértékűség, de az x_r^B optimális megoldás tört. Az x_r bázisváltozó sora a forrás sor, amelyből a már jól ismert egyenlet olvasható ki:

$$x_r + \sum_{j \in NB} t_{rj} x_j = x_r^B.$$

Az x_r^B megoldásnak egész és tört részre való bontása és rendezés után kapjuk, hogy

$$x_r - \left[x_r^B\right] = f_r - \sum_{j \in NB} t_{rj} x_j.$$

Két diszjunkt esetet különböztetünk meg, egyik amikor az x_r változó értékére az $x_r \leqq \left[x_r^B\right]$, a másik pedig amikor az $x_r \geqq \left[x_r^B\right] + 1$ összefüggés áll fenn.

a) eset: $x_r \leqq \left[x_r^B\right]$, ekkor a fenti egyenletből az alábbi egyenlőtlenség következik:

$$-\sum_{j \in NB} t_{rj} x_j \leqq -f_r.$$

b) eset: $x_r \geqq \left[x_r^B\right] + 1$, ekkor pedig az egyenletből az alábbi egyenlőtlenség következik:

$$\sum_{j \in NB} t_{rj} x_j \leqq f_r - 1.$$

Mivel előre nem tudjuk, hogy a két eset közül melyik következik be, ezért a két feltételt megpróbáljuk egyetlen feltételbe egyesíteni, amelyből majd meghatározzuk a vágást. Ehhez definiáljunk két indexhalmazt, a nembázis változók indexeinek két diszjunkt halmazát, amelyek legyenek a következők:

$$NB^+ = \{j \in NB : t_{rj} > 0\} \quad \text{és} \quad NB^- = \{j \in NB : t_{rj} < 0\}$$

Ekkor az a) esetbeli egyenlőtlenségünk alakja

$$-\sum_{j \in NB^+} t_{rj} x_j - \sum_{j \in NB^-} t_{rj} x_j \leqq -f_r.$$

Az egyenlőlenségben szereplő második tag $\left(-\sum_{j \in NB^-} t_{rj} x_j\right)$ az ismeretlenek nemnegativitása és a t_{rj} negativitása miatt pozitív vagy zérus, így ha ezt a tagot elhagyjuk, akkor is fennáll az egyenlőtlenség, azaz írható, hogy

$$(*) \qquad -\sum_{j \in NB^+} t_{rj} x_j \leqq -f_r.$$

Hasonlóan kezeljük a b) esetbeli egyenlőtlenséget, így a

$$\sum_{j \in NB^+} t_{rj} x_j + \sum_{j \in NB^-} t_{rj} x_j \leqq f_r - 1$$

egyenlőlenségben szereplő első tag $\left(\sum_{j\in NB^+} t_{rj}x_j\right)$ az ismeretlenek nemnegativitása és a t_{rj} pozitivitása miatt pozitív vagy zérus, így ha ezt a tagot elhagyjuk, akkor is fennáll az egyenlőtlenség, azaz írható, hogy

$$\sum_{j\in NB^-} t_{rj}x_j \leqq f_r - 1.$$

Célszerűségi okokból szorozzuk be ezt az utóbbi egyenlőtlenséget az $f_r/(1-f_r) > 0$ mennyiséggel, ekkor

$$(**) \qquad \sum_{j\in NB^-} \frac{f_r}{1-f_r} t_{rj}x_j \leqq -f_r.$$

Mivel a $(*)$ és a $(**)$ egyenlőtlenségek egyszerre nem teljesedhetnek, azaz kölcsönösen kizárják egymást, a kettőt egyetlen egyenlőtlenségbe lehet összevonni az alábbiak szerint

$$\sum_{j\in NB^-} \frac{f_r}{1-f_r} t_{rj}x_j - \sum_{j\in NB^+} t_{rj}x_j \leqq -f_r.$$

Ezt az egyenlőtlenséget nevezzük **vegyes vágás**nak. A vegyes vágás a már korábban bevezetett egységesített vágási képlettel az alábbiak szerint fogalmazható meg:

$$-\sum_{j\in NB} q_j x_j \leqq -f_r,$$

ahol

$$q_j = \begin{cases} t_{rj} & \text{ha } t_{rj} \geqq 0 \\ \frac{f_r}{1-f_r}(-t_{rj}) & \text{ha } t_{rj} < 0 \end{cases}$$

Példa:

Oldjuk meg az alábbi vegyes integer lineáris programozási feladatot vegyes vágás módszerrel!

$$\begin{array}{rcrcl} 3x_1 & + & 5x_2 & \to & \max! \\ x_1 & + & 2x_2 & \leqq & 9.75 \\ 3x_1 & + & 4x_2 & \leqq & 21.75 \end{array}$$

$$x_1, x_2 \geqq 0$$

$$x_2 \text{ egész}$$

A standard feladathoz bevezetett $u_1, u_2 \geqq 0$ hiányváltozók lehetnek törtek is.

A folytonos feladat induló és optimális szimplex táblája a következő:

	x_1	x_2	
u_1	1	2	9.75
u_2	3	4	21.75
	3	5	0

	u_2	u_1	
x_2	-1/2	3/2	3.75
x_1	1	-2	2.25
	-1/2	-3/2	-25.5

Csak az x_2 változó sora választható forrás sornak, mivel csak az x_2 változóra kötöttük ki az egészértékűséget és a megoldásban értéke tört. A vegyes vágás

$$-\frac{0.75}{1-0.75}(-(-1/2))u_2 - 3/2u_1 + u_3 = -0.75$$

Ne feledkezzünk meg, hogy itt a bevezetett u_3 nemnegatív hiányváltozóra már nem írható elő az egészértékűség. A vegyes vágás beépítése után kapott szimplex tábla

	u_2	u_1	
x_2	–1/2	3/2	3.75
x_1	1	-2	2.25
u_3	-3/2	-3/2	-0.75
	-1/2	-3/2	-153/4

A duál módszer végrehajtása után az alábbi optimális szimplex tábla adódik:

	u_3	u_1	
x_2	-1/3	2	4
x_1	2/3	-3	1.75
u_2	-2/3	1	0.5
	-1/3	-1	-25.25

Az optimális táblából látható, hogy az x_2 változóra előírt egészértékűségi feltétel teljesül, így az algoritmus befejeződött. Az eredeti feladat optimális megoldása:

$$x_1 = 1.75,\ x_2 = 4,\ z_{\max} = 25.25$$

3.3.5. Mélyebb vegyes vágás

A vegyes vágás esetén mélyebben belevághatunk a folytonos feltételi halmazba, ha az optimális szimplex tábla egyes **nembázis változóira** elő van írva az egészértékűség. A mélyebb vegyes vágás levezetése az alábbiak szerint történik. Tekintsük az optimális szimplex tábla forrás sorát és abból a már jól ismert egyenletet:

$$x_r + \sum_{j \in NB} t_{rj} x_j = x_r^B.$$

A mélyebb vegyes vágás levezetése sok hasonlóságot mutat a vegyes vágásnál látottakhoz. Most is definiáljuk a nembázis változók két indexhalmazát, de másféle módon.

Legyen NBE azon nembázis változók indexeinek halmaza, amely nembázis változókra az egészértékűség ki van kötve, képletben $NBE = \{j \in NB : x_j$ változó egész$\}$.

Legyen NBT azon nembázis változók indexeinek halmaza, amely nembázis változókra az egészértékűség nincs kikötve, képletben $NBT = \{j \in NB : x_j$ változó lehet tört is$\}$, tehát $NBT = NB \setminus NBE$.

Most vezessük be az x_r **bázisbeli egész megkötésű változó** helyett a szintén egész megkötésű $\hat{x}_r$ változót a következő módon

$$\hat{x}_r = x_r + \sum_{j \in NBE} \lambda_j x_j,$$

ahol λ_j rögzített egész számok. Tehát az x_r változóhoz adjuk hozzá azon nembázis változók lineáris kombinációját, amelyekre az egészértékűség meg van kötve. A λ_j egész számok értékének megadását később ismertetjük. Mivel $x_r,\ \lambda_j, x_j, j \in NBE$ egészek, az $\hat{x}_r$ változónak is

egésznek kell lennie. Figyelembe véve a fentieket, a forrás sorbeli egyenletet átrendezhetjük az alábbi módon

$$\hat{x}_r - \left[x_r^B\right] = f_r - \sum_{j \in NBE} (t_{rj} - \lambda_j) x_j - \sum_{j \in NBT} t_{rj} x_j.$$

A vegyes vágáshoz hasonlóan szintén két esetet különböztetünk meg.

a) eset: $\hat{x}_r \leqq \left[x_r^B\right]$, ekkor az egyenletből az alábbi egyenlőtlenség következik

$$- \sum_{j \in NBE} (t_{rj} - \lambda_j) x_j - \sum_{j \in NBT} t_{rj} x_j \leqq -f_r.$$

b) eset: $\hat{x}_r \geqq \left[x_r^B\right] + 1$, ekkor pedig az egyenletből az alábbi egyenlőtlenség következik

$$\sum_{j \in NBE} (t_{rj} - \lambda_j) x_j + \sum_{j \in NBT} t_{rj} x_j \leqq f_r - 1.$$

Vezessük be az alábbi módon a következő indexhalmazokat:

$$\begin{array}{lll} NBE^+ = \{j \in NBE : t_{rj} - \lambda_j > 0\} & \text{és} & NBE^- = \{j \in NBE : t_{rj} - \lambda_j < 0\} \\ NBT^+ = \{j \in NBT : t_{rj} > 0\} & \text{és} & NBT^- = \{j \in NBT : t_{rj} < 0\} \end{array}$$

Hasonlóan a vegyes vágásnál ismertetettekhez, elhagyható az egyenlőtlenségekből egy-egy tag, így az a) és a b) esetekben az alábbi egyenlőtlenségek adódnak.

a) esetben

$$- \sum_{j \in NBE^+} (t_{rj} - \lambda_j) x_j - \sum_{j \in NBT^+} t_{rj} x_j \leqq -f_r$$

b) esetben

$$\sum_{j \in NBE^-} (t_{rj} - \lambda_j) x_j + \sum_{j \in NBT^-} t_{rj} x_j \leqq f_r - 1,$$

amely az $f_r/(1 - f_r) > 0$ mennyiséggel megszorozva az alábbi alakot ölti

$$\sum_{j \in NBE^-} \frac{f_r}{1 - f_r} (t_{rj} - \lambda_j) x_j + \sum_{j \in NBT^-} \frac{f_r}{1 - f_r} t_{rj} x_j \leqq -f_r.$$

Mivel a két esetbeli egyenlőtlenségek egyszerre nem teljesedhetnek, azaz kölcsönösen kizárják egymást, a kettőt egyetlen egyenlőtlenségbe lehet összevonni az alábbiak szerint

$$- \sum_{j \in NBE^+} (t_{rj} - \lambda_j) x_j + \sum_{j \in NBE^-} \frac{f_r}{1 - f_r} (t_{rj} - \lambda_j) x_j - \sum_{j \in NBT^+} t_{rj} x_j + \sum_{j \in NBT^-} \frac{f_r}{1 - f_r} t_{rj} x_j \leqq -f_r.$$

Eddig a λ_j ($j \in NBE$) adott értékekről csupán annyit követeltünk meg, hogy egész számok. Válasszuk a λ_j egész értékeket olyanra, hogy az x_j változók együtthatói **abszolút érték**ben minél kisebbek legyenek.

A $j \in NBE^+$, azaz a $t_{rj} - \lambda_j > 0$ eset vizsgálata**:**

Az x_j változó együtthatója akkor a legkisebb, ha $\lambda_j = [t_{rj}]$, ebben az esetben az x_j együtthatója $t_{rj} - \lambda_j = f_{rj} > 0$, függetlenül attól, hogy a t_{rj} táblázatbeli értékeknek milyen előjele van.

A $j \in NBE^-$, azaz a $t_{rj} - \lambda_j < 0$ eset vizsgálata:
Az x_j változó együtthatója abszolút értékben akkor a legkisebb, ha $\lambda_j = [t_{rj}]+1$, ebben az esetben az x_j együtthatójának $t_{rj} - \lambda_j$ tényezője $t_{rj} - \lambda_j = f_{rj} - 1 < 0$, függetlenül attól, hogy a t_{rj} táblázatbeli értékeknek milyen előjele van.

A fentiekből kiolvasható, hogy az x_j egész megkötésű nembázis változó együtthatójának **abszolút értéke** az indexhalmazoktól függően:

$$\begin{aligned} f_{rj}, &\quad \text{ha } j \in NBE^+, \\ \frac{f_r}{1-f_r}(1-f_{rj}), &\quad \text{ha } j \in NBE^-. \end{aligned}$$

Válasszuk az x_j változó együtthatójának abszolút értékét a két érték közül a kisebbre. Könnyen ellenőrizhető, hogy

ha $f_{rj} < f_r$, akkor az f_{rj} a kisebb érték,
ha $f_{rj} > f_r$, akkor pedig $\frac{f_r}{1-f_r}(1-f_{rj})$ a kisebb érték.

Összefoglalva tehát, a mélyebb vegyes vágás az alábbiak szerint írható fel:

$$-\sum_{j \in NB} q_j x_j \leqq -f_r$$

ahol

$$q_j = \begin{cases} t_{rj} & \text{ha } t_{rj} \geqq 0 \text{ és } x_j \text{ nembázis változóra } \textbf{nincs} \text{ egész megkötés} \\ \frac{f_r}{1-f_r}(-t_{rj}) & \text{ha } t_{rj} < 0 \text{ és } x_j \text{ nembázis változóra } \textbf{nincs} \text{ egész megkötés} \\ f_{rj} & \text{ha } f_{rj} \leqq f_r \text{ és } x_j \text{ nembázis változóra } \textbf{van} \text{ egész megkötés} \\ \frac{f_r}{1-f_r}(1-f_{rj}) & \text{ha } f_{rj} > f_r \text{ és } x_j \text{ nembázis változóra } \textbf{van} \text{ egész megkötés} \end{cases}$$

Példa:

Tekintsünk egy vegyes integer feladat optimális szimplex táblájából egy részletet. Legyen az x_2, x_3, x_4 változókra megkötve az egészértékűség, a többi változó értéke lehet tört is.

a) Határozzuk meg a vegyes vágást!

b) Határozzuk meg a mélyebb vegyes vágást!

	x_2	x_4	x_5	x_7	
$\vdots$					$\vdots$
x_3	$\frac{29}{6}$	$-\frac{17}{6}$	$-\frac{13}{6}$	$\frac{34}{6}$	$\frac{152}{6}$
$\vdots$					$\vdots$

A forrás sor adatainak a vágások meghatározásához szükséges törtrészei a következők:

$$f_{32} = \frac{5}{6}, \quad f_{34} = \frac{1}{6}, \quad f_3 = \frac{2}{6}.$$

a) A vegyes vágás:

A vágásbeli q_j együtthatók és a vágás:

$$\begin{aligned} q_2 &= \tfrac{29}{6}, && \text{mert } t_{32} > 0 \\ q_4 &= \tfrac{\frac{2}{6}}{1-\frac{2}{6}}\tfrac{17}{6}, && \text{mert } t_{34} < 0 \\ q_5 &= \tfrac{\frac{2}{6}}{1-\frac{2}{6}}\tfrac{13}{6}, && \text{mert } t_{35} < 0 \\ q_7 &= \tfrac{34}{6}, && \text{mert } t_{37} > 0 \end{aligned}$$

$$-\frac{29}{6}x_2 - \frac{\frac{2}{6}}{1-\frac{2}{6}}\frac{17}{6}x_4 - \frac{\frac{2}{6}}{1-\frac{2}{6}}\frac{13}{6}x_5 - \frac{34}{6}x_7 \leqq -\frac{2}{6}.$$

b) A mélyebb vegyes vágás:

A vágásbeli q_j együtthatók és a vágás:

$$\begin{aligned}
q_2 &= \tfrac{\frac{2}{6}}{1-\frac{2}{6}}(1-\tfrac{5}{6}), && \text{mert } f_{32} > f_3 \text{ és } x_2 \text{ egész kötésű}\\
q_4 &= \tfrac{1}{6}, && \text{mert } f_{34} < f_3 \text{ és } x_4 \text{ egész kötésű}\\
q_5 &= \tfrac{\frac{2}{6}}{1-\frac{2}{6}}\tfrac{13}{6}, && \text{mert } t_{35} < 0 \text{ és } x_5 \text{ nem egész kötésű}\\
q_7 &= \tfrac{34}{6}, && \text{mert } t_{37} > 0 \text{ és } x_7 \text{ nem egész kötésű}
\end{aligned}$$

$$-\frac{\frac{2}{6}}{1-\frac{2}{6}}(1-\frac{5}{6})x_2 - \frac{1}{6}x_4 - \frac{\frac{2}{6}}{1-\frac{2}{6}}\frac{13}{6}x_5 - \frac{34}{6}x_7 \leqq -\frac{2}{6}.$$

3.4. Utazó ügynök feladat (TSP)

3.4.1. A TSP feladat megfogalmazása

Legyen adott n város és jelöljük ezeket az $1, 2, \ldots, n$ számokkal. Legyenek ismertek az egyes városok közötti nemnegatív utazási költségek (vagy távolságok, vagy idők), jelölje ezeket c_{ij}, amely értékeket egy $\mathbf{C} = (c_{ij})$ mátrix segítségével adunk meg. Egy ügynöknek munkája során minden városba el kell mennie. Az utazó ügynök feladat (traveling salesman problem, TSP) a **legkisebb költségű** (vagy legrövidebb távolságú, vagy legrövidebb idejű) **körút** meghatározása oly módon, hogy **minden várost csak egyszer** látogathat meg az ügynök a körútja során. Feladatunk tehát megállapítani, hogy milyen sorrendben látogassa meg az ügynök a városokat.

Mielőtt matematikailag megfogalmaznánk a TSP feladatot, definiáljuk a teljes körút (röviden **körút**) és a kis körút (**alkörút**) fogalmát. Válasszunk ki egy várost tetszőlegesen, legyen ez az i_1. Egy körút akkor teljes, ha minden várost érintünk és pontosan egyszer érintünk mielőtt visszaérkezünk az i_1 városba. A körút az alábbi ún. **indexlánc**cal írható le: $i_1, i_2, \ldots i_{n-1}, i_n, i_1$, ahol i_j a meglátogatott városok és minden j-re ($j = 2, 3, \ldots, n$) különböző. Ha k várost, ahol $k < n$, pontosan egyszer érintve térünk vissza, akkor ez egy alkörút.

3.4.2. A TSP feladat matematikai megfogalmazása Hozzárendelési feladat segítségével

Legyen a feladat döntési változója x_{ij}, ahol

$$x_{ij} = \begin{cases} 1, & \text{ha az ügynök az } i \text{ városból a } j \text{ városba utazik} \\ 0, & \text{egyébként} \end{cases}$$

A körúthoz szükséges feltétel, hogy egy tetszőleges i városból egy és csak egy városba mehet az ügynök, ezt az alábbi összefüggéssel írhatjuk le és ennek minden i városra igaznak kell lennie:

$$x_{i1} + x_{i2} + \ldots + x_{in} = 1, \quad i = 1, \ldots, n$$

A körúthoz szükséges feltétel továbbá az is, hogy egy tetszőleges j városba egy és csak egy városból érkezhet az ügynök, ezt az alábbi összefüggéssel írhatjuk le és ennek minden j városra igaznak kell lennie:

$$x_{1j} + x_{2j} + \ldots + x_{nj} = 1, \quad j = 1, \ldots, n$$

A fenti két feltétel azonban nem biztosít minden esetben körútat. Gondoljuk arra, hogy pl. $n = 3$ esetén az alábbi három megoldás mindegyike teljesíti a feltételeket, de csak a harmadik ad körútat:
$x_{11} = x_{22} = x_{33} = 1$, a többi $x_{ij} = 0$,
$x_{12} = x_{21} = x_{33} = 1$, a többi $x_{ij} = 0$,
$x_{13} = x_{32} = x_{21} = 1$, a többi $x_{ij} = 0$.

A fenti feltételek tehát **nem elégségesek**, kell valami előírás az $x_{ij} = 1$, azaz az utat definiáló döntési változókra. Ez pedig az $x_{ij} = 1$ változók indexeire vonatkozó **indexlánc** feltétel, amely biztosítja, hogy a megoldás körút.

A feladat célfüggvénye, azaz a körút végrehajtásának költsége:

$$c_{11}x_{11} + c_{12}x_{12} + \ldots + c_{ij}x_{ij} + \ldots + c_{nn}x_{nn}.$$

A c_{ii} költségértékeknek és az x_{ii} döntési változóknak a TSP feladatnál nincs értelme. Elviekben az összes eddigi képletünkben nem szabadott volna használni őket. Könnyen észrevehetjük, hogy a $c_{ii} = \infty$ $(i = 1, 2, \ldots n)$ választással a minimalizálás miatt mindig biztosítható az $x_{ii} = 0$ megoldás. Tehát a következőkben mindig feltesszük, hogy $c_{ii} = \infty$ (vagy $c_{ij} = M$, ahol M egy elegendően nagy adott számot jelöl).

Az utazó ügynök feladat tehát egy olyan speciális Hozzárendelési feldat, amelynek a megoldása körút, matematikai megfogalmazása a következő:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min!$$

$$\begin{aligned}
\sum_{j=1}^{n} x_{ij} &= 1, \quad i = 1, \ldots, n \\
\sum_{i=1}^{n} x_{ij} &= 1, \quad j = 1, \ldots, n \\
x_{ij} &= 0 \text{ vagy } 1, \quad i, j = 1, \ldots, n \\
&\quad \text{és a megoldás körút}
\end{aligned}$$

3.4.3. A TSP feladat megoldása Szétválasztás és korlátozás mószerrel

Az utazó ügynök feladat elvileg tényleges leszámlálással is megoldható. Könnyen észrevehetjük, hogy n város esetén $(n-1)!$ lehetséges körút létezik, ezek mindegyikét előállítjuk (**leszámláljuk**) és közülük kiválasztjuk a legkedvezőbb költségű körútat. Nagy n esetén azonban ezt a módszert nehézkes használni, mivel a lehetséges körútak száma nagyon nagy

(például, ha $n = 11$, akkor $10! = 3\,628\,800$). A TSP feladat integer programozási feladat, így az integer feladatok megoldására megismert Vágási módszer és Szétválasztás és korlátozás módszer egyaránt alkalmas. A probléma megoldására kidolgozott Vágási módszereknél sokkal szemléletesebbek a Szétválasztás és korlátozás módszerek, ezért ezek közül választottunk ki kettőt, amelyeket az alábbi alfejezetekben ismertetünk.

Alkörút elimináló algoritmus Az algoritmus minden lépésében egy Hozzárendelési feladatot oldunk meg, tehát elhagyjuk a körút feltételt. Ennek a feladatnak az optimális célfüggvényértéke az eredeti feladat célfüggvényének egy alsó korlátja lesz. Az alkörút elimináló algoritmust Eastman dolgozta ki 1958-ban.

Első lépésben megoldjuk az eredeti TSP feladathoz rendelt Hozzárendelési feladatot. Amennyiben ennek megoldása körút, akkor befejezzük az algoritmust, mert megkaptuk a TSP feladat optimális megoldását. Ha nem körutat kaptunk, akkor legalább két alkörút adódik. Kiválasztjuk azt az alkörutat, amelyik a legkevesebb várost tartalmazza. Legyenek ebben az alkörútban a városok a következők: $i_1, i_2, \ldots, i_k$, ekkor $x_{i_1 i_2} = x_{i_2 i_3} = \ldots = x_{i_k i_1} = 1$. Az elágaztatást úgy végezzük, hogy eliminájuk (kiküszöböljük) a kapott alkörutat. Ezt úgy végezhetjük, hogy k ágra (részfeladatra) bontjuk a probléma megoldását és az egyes ágakra előírjuk, hogy az egyes ágakon a megfelelő $x_{i_1 i_2}, x_{i_2 i_3}, \ldots, x_{i_k i_1}$ döntési változó értéke 0 legyen. Ezután minden egyes ágon újra megoldunk egy Hozzárendelési feladatot, amelyben az ágat meghatározó döntési változó értéke zérus ($x_{ij} = 0$), ezt úgy valósítjuk meg, hogy a $\mathbf{C}$ mátrixban a megfelelő költséget végtelenre módosítjuk, azaz legyen $c_{ij} = \infty$ (vagy $c_{ij} = M$).

Ezekután már csak azt kell megválaszolni, hogy az aktív részfeladatok közül melyiket választjuk elágaztatásra. Természetesen azt célszerű választani, amelyben az alsó korlát a legkisebb. Ha mindegyik aktív részfeladat alsó korlátja nagyobb mint egy már meglévő körútmegoldás célfüggvényértéke, akkor megállunk, mert megoldottuk az utazó ügynök feladatot.

Példa:

Oldjuk meg az alábbi $\mathbf{C}$ költségmátrix-szal adott TSP feladatot az alkörút eliminációs módszerrel!

$$\mathbf{C} = \begin{bmatrix} M & 5 & 9 & 2 & 10 & 13 \\ 12 & M & 3 & 13 & 10 & 6 \\ 1 & 6 & M & 3 & 10 & 13 \\ 5 & 4 & 10 & M & 12 & 2 \\ 11 & 14 & 7 & 4 & M & 5 \\ 8 & 3 & 11 & 13 & 3 & M \end{bmatrix}$$

A következőkben a Szétválasztás és korlátozás módszerének egyes részfeladatait és azok megoldását közöljük:

0. részfeladat:

Az első lépésben a költségmátrixot jelölje $\mathbf{C}_0$, ahol $\mathbf{C}_0 = \mathbf{C}$. A további lépésekben a részfeladat sorszámával fogjuk jelölni a részfeladat költségmátrixát. Megoldjuk a Hozzárendelési feladatot a $\mathbf{C}_0$ segítségével. A Hozzárendelési feladat megoldásának lépéseit nem közöljük, az megtalálható többek között a Hálózati folyamok tananyagban is.

A Hozzárendelési feladat (0. részfeladat) optimális megoldása:

$$x_{14} = x_{23} = x_{31} = x_{42} = x_{56} = x_{65} = 1, \text{ a többi } x_{ij} = 0.$$

A célfüggvény optimális értéke 18. Mivel nem körút adódott ezért ez a célfüggvényérték a TSP feladat célfüggvényének alsó korlátjaként használható, jelöljük ezt a továbbiakban aláhúzással ($\underline{z}$), azaz $\underline{z}_0 = 18$.

A megoldásból látható, hogy két alkörút adódott:
— egyik alkörút: $1 \to 4 \to 2 \to 3 \to 1$, vagy egyszerűbben jelölve $1, 4, 2, 3, 1$
— másik alkörút: $5 \to 6 \to 5$, vagy egyszerűbben jelölve $5, 6, 5$

Mivel a második alkörút tartalmaz kevesebb várost, ezért annak élei szerint ágaztatunk el. Két részfeladatot fogalmazunk meg, az egyik részfeladatban előírjuk, hogy $x_{56} = 0$, a másik részfeladatban pedig $x_{65} = 0$.

1. részfeladat : $x_{56} = 0$

A részfeladat költségmátrixa legyen $\mathbf{C}_1$, amelyet a $\mathbf{C}_0$ mátrixból kapunk a $c_{56} = M$ választással:

$$\mathbf{C}_1 = \begin{bmatrix} M & 5 & 9 & 2 & 10 & 13 \\ 12 & M & 3 & 13 & 10 & 6 \\ 1 & 6 & M & 3 & 10 & 13 \\ 5 & 4 & 10 & M & 12 & 2 \\ 11 & 14 & 7 & 4 & M & M \\ 8 & 3 & 11 & 13 & 3 & M \end{bmatrix}$$

Az 1. részfeladat optimális megoldása:

$$\underline{z}_1 = 18,\ x_{12} = x_{23} = x_{31} = x_{46} = x_{54} = x_{65} = 1, \text{ a többi } x_{ij} = 0.$$

Két alkörút adódott, ezek
$1 \to 2 \to 3 \to 1$, másképpen $(1, 2, 3, 1)$
$4 \to 6 \to 5 \to 4$, másképpen $(4, 6, 5, 4)$

2. részfeladat : $x_{65} = 0$

A $\mathbf{C}_2$ költségmátrixot a $\mathbf{C}_0$ mátrixból kapjuk a $c_{65} = M$ választással:

$$\mathbf{C}_2 = \begin{bmatrix} M & 5 & 9 & 2 & 10 & 13 \\ 12 & M & 3 & 13 & 10 & 6 \\ 1 & 6 & M & 3 & 10 & 13 \\ 5 & 4 & 10 & M & 12 & 2 \\ 11 & 14 & 7 & 4 & M & 5 \\ 8 & 3 & 11 & 13 & M & M \end{bmatrix}$$

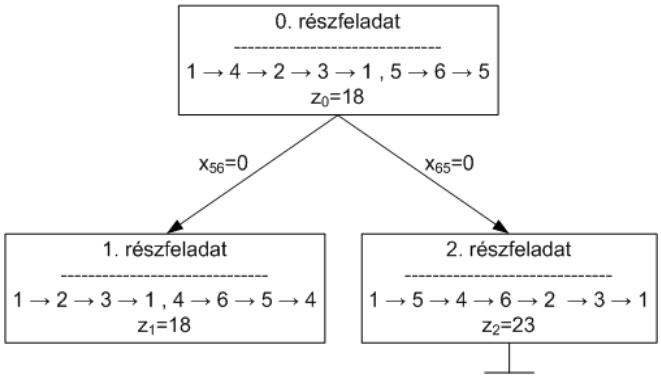
A 2. részfeladat optimális megoldása:

$$z_2 = 23,\ x_{15} = x_{23} = x_{31} = x_{46} = x_{54} = x_{62} = 1, \text{ a többi} x_{ij} = 0.$$

Körút adódott: $1 \to 5 \to 4 \to 6 \to 2 \to 3 \to 1$, másképpen $(1, 5, 4, 6, 2, 3, 1)$.

A 2. részfeladat lezárható, hisz körutat kaptunk $z_2 = 23$ célfüggvényértékkel. Mivel ez nem korlát, hanem pontos célfüggvény érték, ezért nem használjuk az aláhúzást.

A feladat megoldásának eddigi fázisában az eredményeket az alábbi fagráf tartalmazza:



Az 1. és a 2. részfeladatok megoldása után azt kaptuk, hogy csak egy aktív részfeladatunk (1. sorszámú) van és itt a célfüggvénykorlát kisebb, mint a 2. részfeladat célfüggvény értéke, azaz $\underline{z}_1 < z_2$, így az 1. részfeladatnál ágaztatunk el. Az 1. részfeladatnak két alkörútja van és mindegyik ugyanannyi (három) várost tartalmaz, ezért tetszőlegesen választunk az alkörutak közül. Válasszuk az $1 \to 2 \to 3 \to 1$ alkörutat ennek megfelelően ágaztassunk el három felé.

3. részfeladat : $x_{12} = 0$
A $\mathbf{C}_3$ költségmátrixot a $\mathbf{C}_1$ mátrixból kapjuk a $c_{12} = M$ választással:

$$\mathbf{C}_3 = \begin{bmatrix} M & M & 9 & 2 & 10 & 13 \\ 12 & M & 3 & 13 & 10 & 6 \\ 1 & 6 & M & 3 & 10 & 13 \\ 5 & 4 & 10 & M & 12 & 2 \\ 11 & 14 & 7 & 4 & M & M \\ 8 & 3 & 11 & 13 & 3 & M \end{bmatrix}$$

A 3. részfeladat optimális megoldása:

$$z_3 = 23,\ x_{15} = x_{23} = x_{31} = x_{46} = x_{54} = x_{62} = 1, \text{ a többi } x_{ij} = 0.$$

Körút adódott: $1 \to 5 \to 4 \to 6 \to 2 \to 3 \to 1$, másképpen $(1, 5, 4, 6, 2, 3, 1)$.
3. részfeladat lezárható, hisz körutat kaptunk $z_3 = 23$ célfüggvényértékkel.

4. részfeladat : $x_{23} = 0$
A $\mathbf{C}_4$ költségmátrixot a $\mathbf{C}_1$ mátrixból kapjuk a $c_{23} = M$ választással:

$$\mathbf{C}_4 = \begin{bmatrix} M & 5 & 9 & 2 & 10 & 13 \\ 12 & M & M & 13 & 10 & 6 \\ 1 & 6 & M & 3 & 10 & 13 \\ 5 & 4 & 10 & M & 12 & 2 \\ 11 & 14 & 7 & 4 & M & M \\ 8 & 3 & 11 & 13 & 3 & M \end{bmatrix}$$

A 4. részfeladat optimális megoldása:

$$z_4 = 23,\ x_{14} = x_{26} = x_{31} = x_{42} = x_{53} = x_{65} = 1, \text{ a többi } x_{ij} = 0.$$

Körút adódott: $1 \to 4 \to 2 \to 6 \to 5 \to 3 \to 1$, másképpen $(1, 4, 2, 6, 5, 3, 1)$.
A 4. részfeladat lezárható, hisz körutat kaptunk $z_4 = 23$ célfüggvényértékkel.

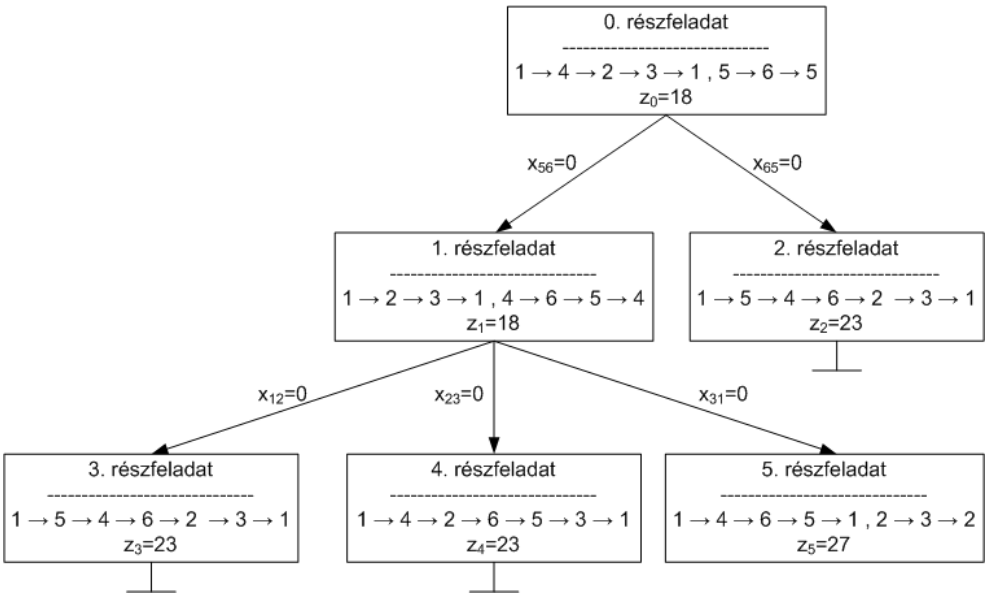
5. részfeladat : $x_{31} = 0$
A $\mathbf{C}_5$ költségmátrixot a $\mathbf{C}_1$ mátrixból kapjuk a $c_{31} = M$ választással:

$$\mathbf{C}_5 = \begin{bmatrix} M & 5 & 9 & 2 & 10 & 13 \\ 12 & M & 3 & 13 & 10 & 6 \\ M & 6 & M & 3 & 10 & 13 \\ 5 & 4 & 10 & M & 12 & 2 \\ 11 & 14 & 7 & 4 & M & M \\ 8 & 3 & 11 & 13 & 3 & M \end{bmatrix}$$

Az 5. részfeladat optimális megoldása:

$$\underline{z}_5 = 27,\ x_{14} = x_{23} = x_{32} = x_{46} = x_{51} = x_{65} = 1, \text{ a többi } x_{ij} = 0.$$

Két alkörút adódott, ezek
$1 \to 4 \to 6 \to 5 \to 1$, másképpen $(1, 4, 6, 5, 1)$
$2 \to 3 \to 2$, másképpen $(2, 3, 2)$.
A feladat megoldásának eddigi fázisában az eredményeket az alábbi fagráf tartalmazza:



A fagráfon látható, hogy egy aktív részfeladat van (5. sorszámú) és itt a célfüggvénykorlát ($\underline{z}_5 = 27$) nagyobb, mint az eddig talált körutak közül a legjobb célfüggvényérték (23), így nincs szükség további elágaztatásra, befejeztük a feladat megoldását.

Az utazó ügynök feladat optimális megoldásához tartozó célfüggvényérték $z_{\min} = 23$. Az optimális körutat a 2., 3. és a 4. részfeladatok megoldásai adják. Mivel a 2. és a 3. részfeladat ugyanazt az optimális körutat adta, így a TSP feladatnak két optimális megoldása van, az optimális körutak pedig a következők:

$$1 \to 5 \to 4 \to 6 \to 2 \to 3 \to 1, \text{ másképpen } (1, 5, 4, 6, 2, 3, 1),$$
$$1 \to 4 \to 2 \to 6 \to 5 \to 3 \to 1, \text{ másképpen } (1, 4, 2, 6, 5, 3, 1).$$

Körút építő algoritmus Az alkörút eliminációs algoritmus minden egyes részfeladatánál egy Hozzárendelési feladatot kellett megoldani, ami számítástechnikai szempontból elég költséges. Ebben az algoritmusban nem oldunk meg Hozzárendelési feladatot, így tehát a célfüggvény korlátot is más módon fogjuk megállapítani. Az alkörút építő algoritmust Little és munkatársai (Murty, Sweeney, Karel) dolgozták ki 1963-ban. Az alábbiakban megmutatjuk, hogy egyszerű számolással hogyan tudunk becslést (alsó korlátot) adni a TSP feladat célfüggvényének értékére. Legyen

$$\begin{aligned} u_i &= \min_j c_{ij} \\ v_j &= \min_i (c_{ij} - u_i) \end{aligned}$$

azaz u_i a $\mathbf{C}$ költségmátrix i-edik sorának legkisebb értéke, a v_j pedig $\mathbf{C}$ mátrixból az u_i mennyiségekkel csökkentett mátrix j-edik oszlopának legkisebb értéke. Legyen a redukált költség az alábbi

$$\hat{c}_{ij} = c_{ij} - u_i - v_j \geqq 0.$$

Az u_i és a v_j definíciójából következik, hogy $\hat{c}_{ij}$ nemnegatív minden i és j indexre. Figyelembe véve a $\sum_{j=1}^{n} x_{ij} = 1$ és a $\sum_{i=1}^{n} x_{ij} = 1$ feltételeket a TSP feladat célfüggvénye az alábbiak szerint alakul:

$$z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} (\hat{c}_{ij} + u_i + v_j)x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} \hat{c}_{ij}x_{ij} + \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$$

Látható, hogy a célfüggvény értéke a $\hat{c}_{ij} \geqq 0$ miatt az utolsó két tag összegénél nem lehet kisebb, ez azt jelenti, hogy az

$$r = \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$$

érték a z célfüggvény értéknek egy alsó korlátja.

A módszerben tehát a korlátozás egyszerű művelettel (sor- és oszlopredukcióval) elvégezhető. A szétválasztás pedig azon alapszik, hogy lépésről lépésre építjük fel a körutat és minden lépésben ügyelünk arra, hogy ne jöhessen létre alkörút. A körútban szereplő éleken $x_{ij} = 1$. Egy x_{ij} döntési változót **szabad**nak nevezünk, ha a megoldási fában még nincs 0 vagy 1 hozzárendelve. Az elágazást mindig szabad változók alapján végezzük és kétirányú elágazást végzünk. Legyen a szabad változó az x_{ij}, ekkor az egyik ágon $x_{ij} = 0$, a másik ágon pedig $x_{ij} = 1$. Az ágak tehát azt jelentik, hogy az (i, j) él nincs benne, ill. benne van az épülő körútban. Az aktív részfeladatok közül a legkisebb alsó korláttal rendelkező részfeladatot választjuk és ott ágaztatunk el.

Példa:
Oldjuk meg az alábbi $\mathbf{C}$ költségmátrix-szal adott TSP feladatot a körút építő módszerrel!

$$\mathbf{C} = \begin{bmatrix} M & 8 & 3 & 7 & 4 \\ 13 & M & 4 & 5 & 1 \\ 2 & 6 & M & 5 & 3 \\ 8 & 5 & 1 & M & 7 \\ 7 & 4 & 8 & 2 & M \end{bmatrix}$$

0. részfeladat :

Sorredukció és oszlopredukció végrehajtása:

A mátrix sorai mögé írjuk fel az u_i sorminimum értékeket

$$\mathbf{C}_0 = \mathbf{C} = \begin{array}{cccccc} M & 8 & 3 & 7 & 4 & 3 \\ 13 & M & 4 & 5 & 1 & 1 \\ 2 & 6 & M & 5 & 3 & 2 \\ 8 & 5 & 1 & M & 7 & 1 \\ 7 & 4 & 8 & 2 & M & 2 \end{array}$$

Elvégezzük a mátrixon a sorok redukcióját és a keletkező mátrix oszlopai alá írjuk a v_j oszlopminimum értékeket

$$\begin{array}{c} \left[\begin{array}{ccccc} M & 5 & 0 & 4 & 1 \\ 12 & M & 3 & 4 & 0 \\ 0 & 4 & M & 3 & 1 \\ 7 & 4 & 0 & M & 6 \\ 5 & 2 & 6 & 0 & M \end{array}\right] \\ \begin{array}{ccccc} 0 & 2 & 0 & 0 & 0 \end{array} \end{array}$$

Elvégezzük a mátrixon az oszlopok redukcióját és ekkor kapjuk a $\hat{\mathbf{C}}$ mátrixot:

$$\hat{\mathbf{C}} = \left[\begin{array}{ccccc} M & 3 & 0 & 4 & 1 \\ 12 & M & 3 & 4 & 0 \\ 0 & 2 & M & 3 & 1 \\ 7 & 2 & 0 & M & 6 \\ 5 & 0 & 6 & 0 & M \end{array}\right]$$

A két redukció során használt reduláló mennyiségek összege:

$$r_0 = \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j = (3+1+2+1+2) + (0+2+0+0+0) = 11.$$

A célfüggvény alsó korlátja tehát:

$$\underline{z}_0 = r_0 = 11.$$

Induláskor minden döntési változó szabad. Válasszunk ki tetszőlegesen egyet, legyen ez az x_{31}, az elágaztatást e változó szerint végezzük el. A gyakorlatban azt a szabad döntési változót érdemes választani, amelyhez tartozó redukált költségérték a legkisebb.

1. részfeladat : $x_{31} = 0$

Ebben a részfeladatban a $3 \to 1$ útszakaszt kizárjuk. Ezt a c_{31} költségadat nagy értékre állításával végezzük, azaz $c_{31} = M$.

A $\mathbf{C}_1$ mátrix és annak redukciója:

A továbbiakban csak a sorredukciót végezzük, az oszlopredukció v_j számait csupán leírjuk az oszlopok alá. Igaz, hogy ebben az esetben nem kapjuk meg a redukált költségmátrixot, sok esetben anélkül is megállapítható a legjobb szabad változó.

$$\mathbf{C}_1 = \begin{bmatrix} M & 8 & 3 & 7 & 4 \\ 13 & M & 4 & 5 & 1 \\ M & 6 & M & 5 & 3 \\ 8 & 5 & 1 & M & 7 \\ 7 & 4 & 8 & 2 & M \end{bmatrix} \begin{matrix} 3 \\ 1 \\ 3 \\ 1 \\ 2 \end{matrix} \qquad \begin{bmatrix} M & 5 & 0 & 4 & 1 \\ 12 & M & 3 & 4 & 0 \\ M & 3 & M & 2 & 0 \\ 7 & 4 & 0 & M & 6 \\ 5 & 2 & 6 & 0 & M \end{bmatrix}$$
$$\begin{matrix} 5 & 2 & 0 & 0 & 0 \end{matrix}$$

A redukció után nyert eredmények:
$r_1 = 17$
$\underline{z_1} = r_1 = 17$

Megjegyezzük, hogy elegendő, ha leírjuk a mátrix mellé a sorminimumokat és egyidőben leírjuk az alá az oszlop alá a 0-t, amely oszlopban a sorminimum eléretett, hiszen ebben az oszlopban a v_j oszlopminimum értéke zérus lesz. Ezután csak a szabadon maradt oszlopokban kell elvégezni a soronkénti redukciót és az oszlopminimum képzést. Ezt mutatja az alábbi táblázat

$$\mathbf{C}_1 = \begin{bmatrix} M & 8 & 3 & 7 & 4 \\ 13 & M & 4 & 5 & 1 \\ M & 6 & M & 5 & 3 \\ 8 & 5 & 1 & M & 7 \\ 7 & 4 & 8 & 2 & M \end{bmatrix} \begin{matrix} 3 \\ 1 \\ 3 \\ 1 \\ 2 \end{matrix}$$
$$\begin{matrix} 5 & 2 & 0 & 0 & 0 \end{matrix}$$

2. részfeladat : $x_{31} = 1$

Ebben a részfeladatban a $3 \to 1$ útszakaszt bevesszük a körútba. Ez azt jelenti, hogy a körút költsége legalább $c_{31} = 2$. Ebben az esetben két dologra kell figyelni, hogy ne keletkezzen alkörút:

a) Az $1 \to 3$ útszakaszt le kell tiltani ($x_{13} = 0$), mivel alkörút keletkezhet ($3 \to 1 \to 3$), ami nem megengedett. Ezt a $c_{13} = M$ értékre állítással végezzük.

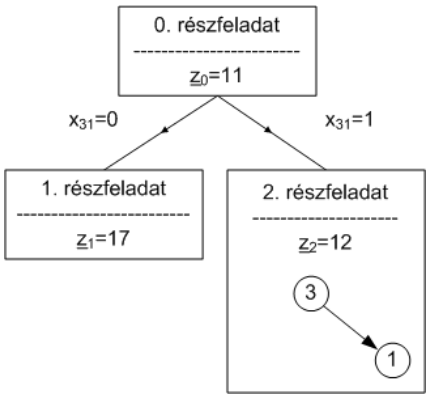
b) Ha tehát a $3 \to 1$ útszakasz a körútban van, akkor ez egyrészt azt jelenti, hogy a 3 jelű városból már egyik városba sem mehetünk, másrészt pedig azt, hogy az 1 jelű városba már egyik városból sem érkezhetünk. Le kell tiltani tehát az összes ilyen útszakaszt. Ezt a költségadatok nagy értékre állításával végezhetjük, azaz legyen $c_{3j} = M$ és $c_{i1} = M$. Célszerű azonban az az eljárás, hogy elhagyjuk a 3-adik sort és az 1. oszlopot, így tehát a 2. részfeladatunk 4×4-es méretűre redukálódik. A táblázatunkat azonban 5×5-ös méretűre rajzoljuk, így az indexeket egyszerűbben tudjuk nyomon követni a megoldás során. Az elhagyott sort és oszlopot $-$ jellel jelöljük a táblázatban.

A $\mathbf{C}_2$ mátrix és annak redukciója:

$$\mathbf{C}_2 = \begin{bmatrix} - & 8 & M & 7 & 4 \\ - & M & 4 & 5 & 1 \\ - & - & - & - & - \\ - & 5 & 1 & M & 7 \\ - & 4 & 8 & 2 & M \end{bmatrix} \begin{matrix} 4 \\ 1 \\ - \\ 1 \\ 2 \end{matrix} \qquad \begin{bmatrix} - & 4 & M & 3 & 0 \\ - & M & 3 & 4 & 0 \\ - & - & - & - & - \\ - & 4 & 0 & M & 6 \\ - & 2 & 6 & 0 & M \end{bmatrix}$$
$$\begin{matrix} - & 2 & 0 & 0 & 0 \end{matrix}$$

A redukció után nyert eredmények:
$r_2 = 10$

$\underline{z}_2 = c_{31} + r_2 = 12$
A feladat eddigi megoldása során kapott fagráf (bináris fa) az alábbi:



Az 1. és a 2. részfeladat célfüggvénykorlátja közül a 2. részfeladaté a kisebb, ezért a 2. részfeladatnál ágaztatunk el. Válasszuk a szabad változók közül az x_{43} döntési változót.

3. részfeladat : $x_{43} = 0$
A 2. részfeladat $\mathbf{C}_2$ költségmátrixából úgy kapjuk a 3. részfeladat költségmátrixát, hogy $c_{43} = M$.

A $\mathbf{C}_3$ mátrix és annak redukciója:

$$\mathbf{C}_3 = \begin{bmatrix} - & 8 & M & 7 & 4 \\ - & M & 4 & 5 & 1 \\ - & - & - & - & - \\ - & 5 & M & M & 7 \\ - & 4 & 8 & 2 & M \end{bmatrix} \quad \begin{matrix} 4 \\ 1 \\ - \\ 5 \\ 2 \end{matrix} \qquad \begin{bmatrix} - & 4 & M & 3 & 0 \\ - & M & 3 & 4 & 0 \\ - & - & - & - & - \\ - & 0 & M & M & 2 \\ - & 2 & 6 & 0 & M \end{bmatrix}$$
$$\begin{matrix} - & 0 & 3 & 0 & 0 \end{matrix}$$

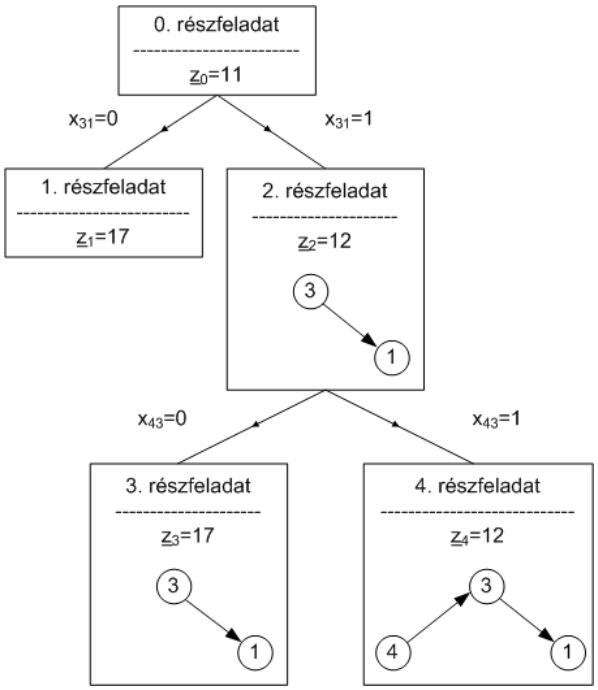
A redukció után nyert eredmények:
$r_3 = 15$
$\underline{z}_3 = c_{31} + r_3 = 17$

4. részfeladat : $x_{43} = 1$
Ebben a részfeladatban a $3 \to 1$ útszakasz mellett az újonnan bevett $4 \to 3$ útszakasz is benne van az épülő körútban. A $\mathbf{C}_2$ mátrixból törölhető a 4-edik sor és a 3-adik oszlop. A $4 \to 3 \to 4$ alkörút keletkezésének megakadályozása miatt a $3 \to 4$ útszakaszt le kell tiltani, azaz legyen $c_{34} = M$. Ezt most példánkban nem tudjuk érvényesíteni a korábbi sortörlés miatt. A 4. részfeladatban két útszakasz van és ezek a $4 \to 3 \to 1$ útvonalat alkotják. Ahhoz, hogy ne keletkezzen alkörút, le kell tiltani az $1 \to 4$ útszakaszt, ezért legyen $c_{14} = M$.

A $\mathbf{C}_4$ mátrix és annak redukciója:

$$\mathbf{C}_4 = \begin{bmatrix} - & 8 & - & M & 4 \\ - & M & - & 5 & 1 \\ - & - & - & - & - \\ - & - & - & - & - \\ - & 4 & - & 2 & M \end{bmatrix} \quad \begin{matrix} 4 \\ 1 \\ - \\ - \\ 2 \end{matrix} \qquad \begin{bmatrix} - & 4 & - & M & 0 \\ - & M & - & 4 & 0 \\ - & - & - & - & - \\ - & - & - & - & - \\ - & 2 & - & 0 & M \end{bmatrix}$$
$$\begin{matrix} - & 2 & - & 0 & 0 \end{matrix}$$

A redukció után nyert eredmények:
$r_4 = 9$
$\underline{z}_4 = c_{31} + c_{43} + r_4 = 12$
A feladat megoldása során kapott fagráf (bináris fa) az alábbi:



Az eddigiek alapján három aktív részfeladatunk van, nevezetesen az 1., a 3. és a 4. részfeladat, közülük a 4. részfeladat alsó korlátja a legkisebb, ezért a 4. részfeladatnál ágaztatunk el. A szabad döntési változók közül válasszuk az x_{25} változót.

5. részfeladat : $x_{25} = 0$
A 4. részfeladat $\mathbf{C}_4$ költségmátrixából úgy kapjuk az 5. részfeladat mátrixát, hogy $c_{25} = M$.

A $\mathbf{C}_5$ mátrix és annak redukciója:

$$\mathbf{C}_5 = \begin{bmatrix} - & 8 & - & M & 4 \\ - & M & - & 5 & M \\ - & - & - & - & - \\ - & - & - & - & - \\ - & 4 & - & 2 & M \end{bmatrix} \quad \begin{matrix} 4 \\ 5 \\ - \\ - \\ 2 \end{matrix} \qquad \begin{bmatrix} - & 4 & - & M & 1 \\ - & M & - & 0 & M \\ - & - & - & - & - \\ - & - & - & - & - \\ - & 2 & - & 0 & M \end{bmatrix}$$
$$- \quad 2 \quad - \quad 0 \quad 0$$

A redukció után nyert eredmények:
$r_5 = 13$
$\underline{z}_5 = c_{31} + c_{43} + r_5 = 16$

6. részfeladat : $x_{25} = 1$
A $2 \to 5$ útszakasz miatt a $\mathbf{C}_4$ mátrixból törölhető a 2. sor és az 5. oszlop. Az alkörút keletkezésének megakadályozása miatt legyen $c_{52} = M$. A már meglévő $4 \to 3 \to 1$ útvonal és a $2 \to 5$ új útszakasz nem alkothat alkörutat, így további megkötés nincs.

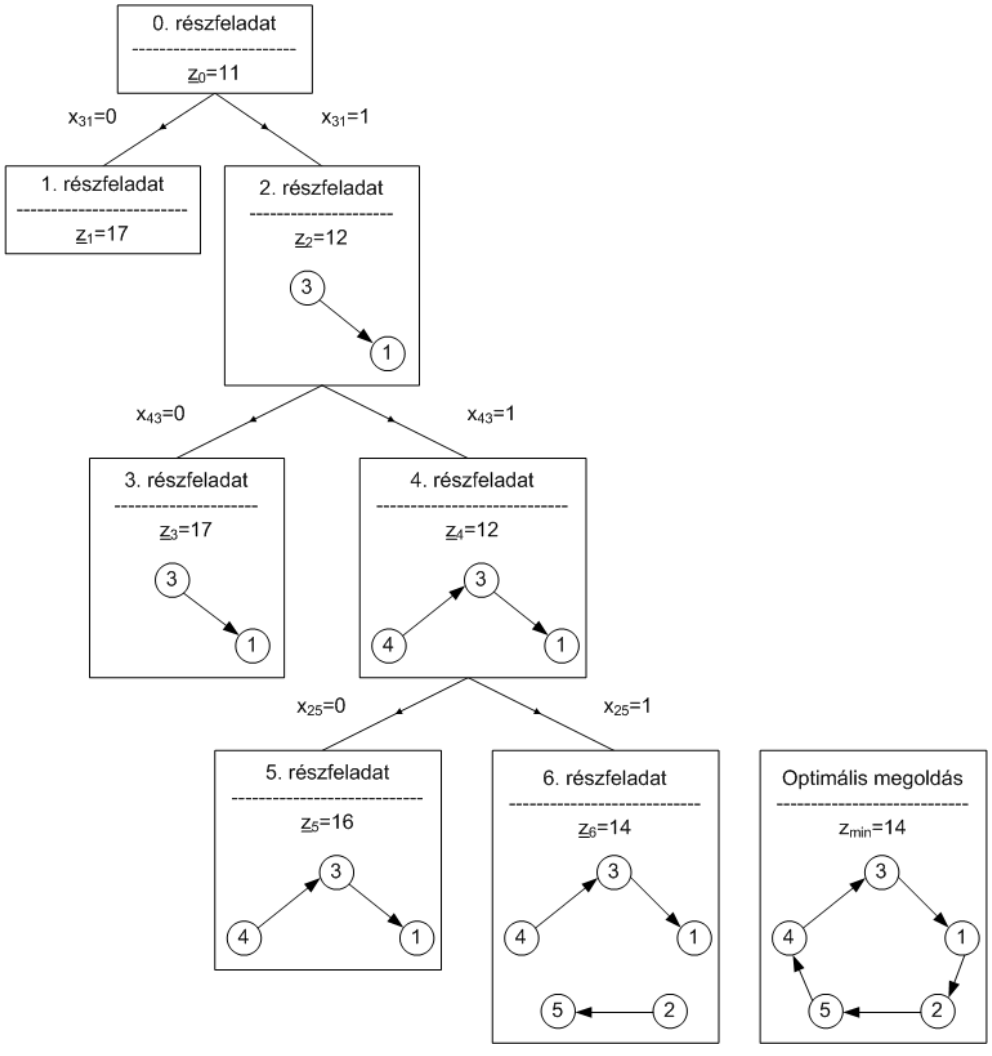
A $\mathbf{C}_6$ mátrix és annak redukciója:

$$\mathbf{C}_6 = \begin{bmatrix} - & 8 & - & M & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & M & - & 2 & - \end{bmatrix} \begin{matrix} 8 \\ - \\ - \\ - \\ 2 \end{matrix} \qquad \begin{bmatrix} - & 0 & - & M & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & M & - & 0 & - \\ - & 0 & - & 0 & - \end{bmatrix}$$

A redukció után nyert eredmények:
$r_6 = 10$
$\underline{z}_6 = c_{31} + c_{43} + c_{25} + r_6 = 14$

Mivel ebben a részfeladatban a $\mathbf{C}_6$ mátrix 2×2-es, így csak két szabad változó maradt, nevezetesen az x_{12} és az x_{54}. Az elágaztatásnak már nincs értelme, így a 6. részfeladat lezárható. Legyen $x_{12} = x_{25} = 1$, ezzel egy körutat kaptunk, a körút költsége $z_6 = 14$.

A feladat megoldása során kapott fagráf (bináris fa) az alábbi:



A megoldásnak ebben a szakaszában három aktív részfeladatunk van, nevezetesen a 2., a 3. és az 5. részfeladat. Mivel ezekben a célfüggvény alsó korlátja (17, 17, 16) nagyobb, mint a körút célfüggvényértéke (14), így befejezhetjük az algoritmust.

Az utazó ügynök feladat optimális megoldása:

$$x_{12} = x_{25} = x_{54} = x_{43} = x_{31} = 1, \text{ a többi } x_{ij} = 0, \ z_{\min} = 14,$$

tehát az optimális körút:

$$1 \to 2 \to 5 \to 4 \to 3 \to 1.$$

Megjegyzés:
A körút építő algoritmus gráfpontonkénti számításigénye sokkal kevesebb, mint az alkörút elimináló algoritmusé. Azonban ennek az egyszerűségnek az az ára, hogy a megoldási fagráf megnagyobbodik. Körút legalább $n-1$ ág létrehozása után alakulhat csak ki.

Feladat:
Oldjuk meg a fenti példákat a másik módszerrel!

Ütemezési probléma:
Legyen adott egy univerzális gép, amelyet át lehet állítani különböző termékek megmunkálására. Legyen n féle termék, amelyet meg lehet munkálni a géppel. Adottak az átállítási idők, a t_{ij} jelentse, hogy az i-edik termék megmunkálásáról a j-edik termék megmunkálásához mennyi idő alatt lehet a gépet átállítani. Az egyes termékekből egy sorozatot legyártva állítjuk át a gépet. Amikor mindegyik termékfajtából egy-egy sorozatot elkészítettünk, folytatjuk a gyártást az előzőeknek megfelelő sorrendben. Milyen sorrendben vegyük a terméksorozatokat gyártásba, hogy a szükséges átállítási idők összege minimális legyen?

A feladat TSP feladatra vezet. Jelentse az x_{ij} döntési változó azt, hogy melyik termék gyártásáról melyik termék gyártására állunk át. Ha $x_{ij} = 1$, akkor az i-edikről a j-edikre átálluk, ha $x_{ij} = 0$, akkor nem.

Ahhoz, hogy a minden terméktípusról valamelyikre, de csak egyikre átálljunk, fenn kell állnia, hogy

$$x_{i1} + x_{i2} + \ldots x_{in} = 1, \quad i = 1, \ldots, n$$

Annak szükséges feltétele, hogy minden terméktípusra valamelyik termékről, de csak egyről átálljunk a következő:

$$x_{1j} + x_{2j} + \ldots + x_{nj} = 1, \quad j = 1, \ldots, n$$

Nem nehéz észrevenni, hogy ezek a feltételek a ciklikus gyártáshoz nem elegendőek, itt is szükség van az ún. **indexlánc** feltételre is. Az is egyszerűen látható, hogy az összes átállási idő is a TSP feladat célfüggvénye szerint alakul, azaz

$$t_{11}x_{11} + t_{12}x_{12} + \ldots + t_{ij}x_{ij} + \ldots + t_{nn}x_{nn}.$$

Feladat:
Oldja meg a fentebb megfogalmazott **ütemezési problémát**, ha az átállítások időszükségletét tartalmazó $\mathbf{T}$ mátrix az alábbi:

$$\mathbf{T} = \begin{bmatrix} M & 8 & 5 & 3 & 2 & 4 \\ 4 & M & 4 & 3 & 5 & 7 \\ 3 & 4 & M & 5 & 6 & 4 \\ 8 & 2 & 6 & M & 5 & 5 \\ 5 & 5 & 4 & 6 & M & 4 \\ 5 & 4 & 4 & 7 & 3 & M \end{bmatrix}$$