# A matematikai szigorúság fontossága a kiindulási struktúra meghatározásakor szétválasztási hálózatok szintézise esetén

HECKL ISTVÁN*
KOVÁCS ZOLTÁN**
ERCSEY ZSOLT*

*A kémiai technológiákban a szétválasztási műveletek nagy hangsúlyt kapnak, mert a gyakran többlépéses reakcióval előállított termékek kívánt tisztaságát csak helyesen megválasztott szeparációs művelettel, vagy szeparációs műveletek sorozatával lehet biztosítani. Ez egyúttal fontos környezetszennyezést csökkentő művelet. Mivel a szétválasztási műveletek fajlagos energiaigénye általában nagy, az alkalmazott szeparációs hálózat a termék tisztaságán túl az előállítási költséget is jelentősen befolyásolja. Ha a termék tisztasága iránti követelmény magasabb, akkor az előbbiekben leírtak fokozottabban érvényesülhetnek, tehát még nagyobb figyelmet kell fordítani az optimális szétválasztási hálózat meghatározására. A közlemény erre mutat be módszert és gyakorlati példákat, és így illik be a környezetbarát technológiák témakörébe.*

*Összetett rendszereknél csak szigorú matematikai alapokon nyugvó módszerekkel biztosítható az optimális szeparációs hálózat megtalálása. A módszernek tartalmaznia kell a szisztematikus modellgenerálás lépését is ahhoz, hogy olyan keresési teret hozzon létre, amely az optimális megoldás struktúráját garantáltan tartalmazza.*

## Bevezetés

Az általánosan használt algoritmikus módszerek sokszor nem képesek megoldani a globális optimalizálási feladatot még olyan viszonylag egyszerű feladatosztály esetén sem, ahol a költségfüggvény lineárisnak tekinthető. Ez a tény két dolognak a következménye: egyrészt a kiindulási struktúrák (szuperstruktúrák), amelyeken a megoldás alapszik, nem teljesek, vagyis nem tartalmaznak minden, az optimális megoldás szempontjából lényeges elemet, másrészt a probléma alapján felírt matematikai modellek szükségtelenül bonyolultak.

*Számítástudomány Alkalmazása Tanszék, Veszprémi Egyetem, Veszprém
**Számítástudományi Tanszék, Szegedi Tudományegyetem, Szeged

Egy olyan eljárás [1] alapján generáljuk a szuperstruktúrát és a hozzá kapcsolódó matematikai modellt, amely eljárás minden lépésének a helyessége bizonyított. Négy olyan publikált példát vizsgálunk meg, amelyeknél a szerzők azt állították, hogy megtalálták a globális optimumot. A problémák vizsgálata során azt tapasztaltuk, hogy *az általunk kifejlesztett módszer minden esetben legalább megegyező vagy akár 30 százalékkal jobb megoldást szolgáltat, mint az eddig ismertek.*

A szétválasztási hálózatok szintézise (Separation Network Synthesis, SNS) feladat optimális megoldásának megtalálásához szükségünk van egy megfelelő matematikai programozási modellre és egy globális optimalizálási módszerre. Az előbbit az utóbbival megoldva kapjuk meg az eredeti probléma optimális megoldását. Nyilvánvalóan nincs olyan módszer, amely általános esetben pontosan képes létrehozni a megfelelő programozási modellt, ellenben ismert néhány hatékony matematikai programozási modellt megoldó módszer.

Az, hogy néhány egyszerű SNS feladatosztály esetén is találtak meglepő tulajdonságú optimumot [2], azt mutatja, hogy a megfelelő matematikai programozási modell meghatározása általános esetben és az SNS feladatosztály esetében is egy igen nehéz feladat. Ebben az írásban megmutatjuk, hogyan lehet jobb eredményt kapni egy nem kellőképpen bizonyított eljárás által adott, optimálisnak gondolt megoldásnál úgy, hogy az új eredmény optimalitása már bizonyított. Egy viszonylag egyszerű SNS feladatosztály megoldására koncentrálunk, amelyben a műveleti egységek költségfüggvényei fix rész nélküli lineáris függvények. A jelenleg ismert algoritmikus módszerekkel ellentétben, amelyekben a modell előállítása nem algoritmikus, az általunk javasolt módszer minden egyes lépése algoritmikus a modell előállítástól kezdve az optimális megoldás meghatározásáig. A modell előállítása egy újszerű szisztematikus technika segítségével történik, sőt az eredményül kapott modell szigorúsága bizonyított.

## Szigorú szuperstruktúra

Először körvonalazzuk, hogy melyek az alapvető információk, amelyek szükségesek az algoritmikus folyamatszintézis módszerhez és a hozzátartozó szuperstruktúrához. Egy SNS feladatbeli módszernek két lényeges lépést kell magába foglalnia: (i) a matematikai modell generálását, amely magában foglalja mind a szuperstruktúrát, mind a hozzá kapcsolódó matematikai programozási modellt; (ii) valamint a származtatott matematikai programozási modell megoldását, például az optimális struktúra meghatározását valamilyen optimalizálási módszerrel. Ha a matematikai programozási modellt egy nem teljes hálózati struktúrából nyerjük, akkor könnyen lehet, hogy nem az optimális megoldást kapjuk meg, függetlenül attól, hogy milyen optimalizálási módszert használtunk. Elméletileg a szuperstruktúrát csak akkor tudjuk előállítani, ha minden szóba jöhető optimális megoldást ismerünk explicit vagy implicit módon. Habár gyakorlatilag felesleges a szuperstruktúra meghatározása, ha minden lehetséges optimális megoldást expliciten ismerünk. Továbbá ezen megoldások előállítása különösen nehéz, hiszen óriási számú különböző struktúrát kell figyelembe venni a probléma megoldása során. Így ez az eljárás még a legegyszerűbb feladatosztályoknál sem vezet eredményre belátható időn belül [3]. Ez az oka annak, hogy különlegesen fontosak azok a módszerek, amelyek a szuperstruktúra generálásához az impliciten ismert megoldásokra támaszkodnak [4].

A szétválasztási hálózatok szintézise probléma megoldása során két különösen lényeges nézőpontot már hosszú ideje elhanyagoltak. Először is matematikailag nem definiálták a szuperstruktúra fogalmát, valamint nem dolgoztak ki olyan módszert, amely megmondaná, hogy egy adott szuperstruktúra tartalmazza-e az optimális struktúrát az adott SNS feladatosztály minden egyes feladata esetében. Azért, hogy legyőzzük ezeket a nehézségeket, bevezettük a szigorú szuperstruktúra fogalmát.

### A szigorú szuperstruktúra fogalma

*A definíció: Legyenek adottak a műveleti egységek halmazai a műveleti egységek matematikai modelljeivel együtt. Feltételezünk továbbá egy szisztematikus eljárást, mely az adott műveleti egységekből álló bármely hálózathoz olyan matematikai programozási modellt generál, amelynek megoldásával az optimális részhálózatát garantáltan meghatározza. Egy hálózatot szigorú szuperstruktúrának nevezünk, ha a feladatosztályba tartozó bármely feladatra ezen szigorú szuperstruktúra alapján generált matematikai modell megoldásánál kisebb költségű megoldás nem kapható más hálózat- és modellgenerálás alapján.*

A szigorú szuperstruktúra kifejezést azért vezettük be, hogy megkülönböztethető legyen a szuperstruktúrától, amelynek ez idáig nem született pontos definíciója. Az optimális megoldás teljes bizonyossággal megállapítható, amennyiben a matematikai programozási modellt a szigorú szuperstruktúrából állították elő és a megoldó módszerünk képes az optimum meghatározására. Vagyis a szigorú szuperstruktúra generálásának kell az SNS feladat első lépésének lennie.

A szigorú szuperstruktúra nem egyedi, mert két különböző szuperstruktúra is vezethet ugyanahhoz az optimális megoldáshoz az SNS feladatosztály minden egyes feladata esetén. Ennek az egyik oka a következő. Tételezzük fel, hogy egy szigorú szuperstruktúra tartalmaz minden lehetségesen optimális struktúrát az SNS feladatosztály minden feladata esetén. Ezáltal minden olyan struktúra, amely tartalmazza a szigorú szuperstruktúrát, egyúttal tartalmaz minden lehetségesen optimális struktúrát is. Még egy egyszerű SNS feladatosztály esetén is nehéz meghatározni azt a szigorú szuperstruktúrát, amely csak olyan műveleti egységeket és kapcsolataikat tartalmazza, amelyek egy lehetségesen optimális megoldáshoz tartoznak.

## Példák

Négy [5]-beli irodalmi példát dolgoztunk ki részletesen, hogy szemléltessük az általunk használt módszert. Két példa esetén lényegesen jobb megoldást kaptunk, mint az eddig ismertek, a másik kettőnél pedig kicsit jobb, illetve ugyanazon megoldáshoz jutottunk.

### *Az 1. példa*

Az 1. példa adatait az *1. táblázat* mutatja be.

*1. táblázat*

**Az 1. példa adatai**

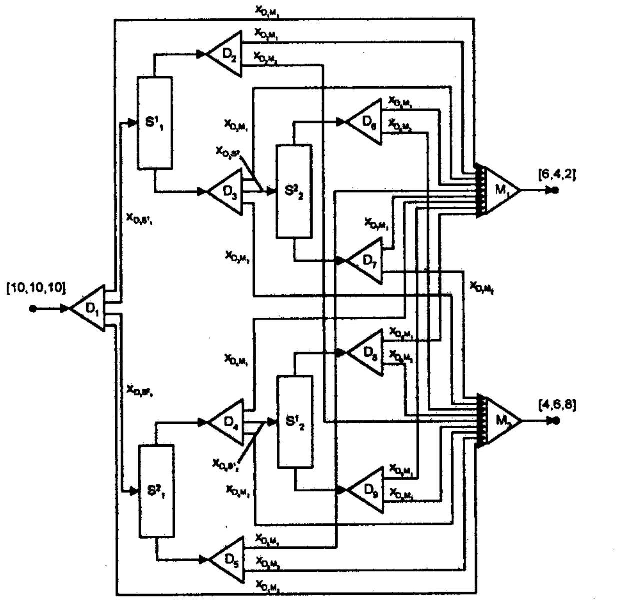
| Komponens | A | B | C |
|---|---|---|---|
| Betáplálás | 10 | 10 | 10 |
| Termék 1 | 6 | 4 | 2 |
| Termék 2 | 4 | 6 | 8 |

Tegyük fel, hogy egy háromkomponensű, 1:1:1 komponens arányú betáplálást szeretnénk szétválasztani két többkomponensű termékké. A termékek komponens arányai az [5] cikkben definiáltak. Cél, hogy a szétválasztókra érkező anyagáramok összege minimális legyen. Ez azt jelenti, hogy a szétválasztók nehézségi foka megegyezik. A szükséges adatok az *1. táblázatban* vannak összefoglalva.

Az [1] cikkben leírt SNS-LMSG algoritmus által generált szigorú szuperstruktúra az *1. ábrán* látható. Ábráinkon D jelöli a szétválasztót (divider), M a keverőt (mixer), S pedig a szétválasztót (separator). A szétválasztó felső indexe jelöli a szétválasztó típusát, az alsó indexszel pedig több ugyanolyan típusú szétválasztót különböztetünk meg. A háromkomponensű betáplálást A, B és C komponenssel, négy folyamra bontottuk: közülük kettő két különböző típusú szétválasztó bemeneteként szolgál, a másik kettőt pedig egyenesen a végtermékekhez vezettük. Mind a négy folyamhoz egy-egy változót rendeltünk, mert a folyamok egy megosztóból erednek és a megosztási arányokat a változók adják majd meg. Konkrétan az $x_{D_1M_1}$ és az $x_{D_1M_2}$ azokhoz kapcsolódnak, amelyek közvetlenül az 1-es és 2-es termékekhez vezetnek, az $x_{D_1S_1^1}$ és $x_{D_1S_1^2}$ változók pedig azokhoz a folyamokhoz, amelyek az $S_1^1$ és $S_1^2$ szétválasztóknak szolgálnak bemenetként. Az $S_1^1$ szétválasztó fejterméke csak az A komponenst tartalmazza, ezt a folyamot kettéosztjuk és a termékek előtti keverőbe vezetjük őket. Ehhez a két folyamhoz az $x_{D_2M_1}$ és az $x_{D_2M_2}$ változókat rendeljük. Az $S_1^1$ szétválasztó fenékterméke a B és C komponenst tartalmazza, a folyamot háromfelé osztjuk: ebből két anyagáramot vezetünk a termékekhez, a harmadik pedig az $S_2^2$ szétválasztó bemenete lesz. A folyamokhoz rendelt változókat az *1. ábráról* olvashatjuk le. Az $S_1^2$ szétválasztó fejterméke az A és B komponenst tartalmazza, az anyagáramot háromfelé osztjuk: közülük két folyamot vezetünk a termékekhez, a harmadik pedig az $S_2^1$ szétválasztó bemenete lesz. Az $S_1^2$ szétválasztónak a fenékterméke csak a C komponenst tartalmazza. Ezt a folyamot megosztás után a két termékhez vezetjük. A megosztás miatt két új változót kell bevezetni a két folyamhoz.

A szétválasztási hálózat szintézis probléma modellje a következő fő részekből áll:

$$\min\left(30\cdot x_{D_1S_1^1}+30\cdot x_{D_1S_1^2}+20\cdot x_{D_4S_2^1}\right) \qquad \text{célfüggvény (1)}$$




*1. ábra.* Az 1. példa szigorú szuperstruktúrája

Feltéve, hogy

$$0\le x_{ij} \qquad (i,j)\in\{D_1M_1, D_1M_2, D_1S_1^1, D_1S_1^2, D_2M_1, D_2M_2, D_3M_1, D_1S_2^2, D_3M_2, D_4M_1, D_4S_2^1, D_4M_2, D_5M_1, D_5M_2, D_6M_1, D_6M_2, D_7M_1, D_7M_2, D_8M_1, D_8M_2, D_9M_1, D_9M_2\} \qquad \text{nem negatívitási feltételek (2)}$$

$$x_{D_1M_1}+x_{D_1S_1^1}+x_{D_1S_1^1}+x_{D_1M_2}=1 \qquad \text{az első megosztó egyenlete (3)}$$

$$\left.\begin{aligned}
x_{D_2M_1}+x_{D_1M_1}&=x_{D_1S_1^1}\\
x_{D_3M_1}+x_{D_3S_2^2}+x_{D_1M_1}&=x_{D_1S_1^1}\\
x_{D_4M_1}+x_{D_4S_2^1}+x_{D_4M_1}&=x_{D_1S_1^2}\\
x_{D_5M_1}+x_{D_5M_2}&=x_{D_1S_1^2}\\
x_{D_6M_1}+x_{D_6M_2}&=x_{D_3S_2^2}\\
x_{D_7M_1}+x_{D_7M_2}&=x_{D_3S_2^2}\\
x_{D_8M_1}+x_{D_8M_2}&=x_{D_4S_2^1}\\
x_{D_9M_1}+x_{D_9M_2}&=x_{D_4S_2^1}
\end{aligned}\right\}\ \text{a többi megosztó egyenlete (4)}$$

$$\left.\begin{aligned}
6&=10\cdot(x_{D_1M_1}+x_{D_2M_1}+x_{D_4M_1}+x_{D_8M_1}\\
4&=10\cdot(x_{D_1M_1}+x_{D_3M_1}+x_{D_4M_1}+x_{D_6M_1}+x_{D_9M_1})\\
2&=10\cdot(x_{D_1M_1}+x_{D_3M_1}+x_{D_5M_1}+x_{D_7M_1})\\
4&=10\cdot(x_{D_1M_2}+x_{D_2M_2}+x_{D_4M_2}+x_{D_8M_2})\\
6&=10\cdot(x_{D_1M_2}+x_{D_3M_2}+x_{D_4M_2}+x_{D_6M_2}+x_{D_9M_2})\\
8&=10\cdot(x_{D_1M_2}+x_{D_3M_2}+x_{D_5M_2}+x_{D_7M_2})
\end{aligned}\right\}\ \text{termékekre vonatkozó feltevések (5)}$$

Az eredményül kapott matematikai programozási modellt megvizsgálva, láthatjuk, hogy a programozási modell LP (linear programming). Figyeljük meg, hogy az [5] által javasolt modell nemlineáris. Az LP modell megoldásával kapjuk meg a célfüggvény minimális értékét, amely 12,00 lesz. Ennél az értéknél a változók értéke a következő:

$$x_{D_1M_1}=0.2 \quad x_{D_1S_1^1}=0.2 \quad x_{D_1S_1^2}=0.2 \quad x_{D_1M_2}=0.4$$
$$x_{D_2M_1}=0.2 \quad x_{D_3M_2}=0.2 \quad x_{D_4M_1}=0.2 \quad x_{D_5M_2}=0.2$$

A *2. ábrán* láthatjuk a megoldáshoz tartozó optimális hálózatot, amely megegyezik az [5] által találttal.

### *A 2. példa*

A példa adatait a *2. táblázat* mutatja be.

*2. táblázat*

**A 2. példa adatai**

| Komponens | A | B | C | D |
|---|---|---|---|---|
| Betáplálás 1 | 6 | 4 | 0 | 0 |
| Betáplálás 2 | 8 | 6 | 10 | 6 |
| Betáplálás 3 | 0 | 0 | 5 | 5 |
| Nehézségi fok | 4 | 1,5 | 4 | |

| Termék | Komponensek összege | Komponens információ |
|---|---|---|
| Termék 1 | 15 | A≥9 B≤3 C≤3 D=0 |
| Termék 2 | 20 | B≥7 C≥7 B=C |
| Termék 3 | 15 | D≥9 A=0 |

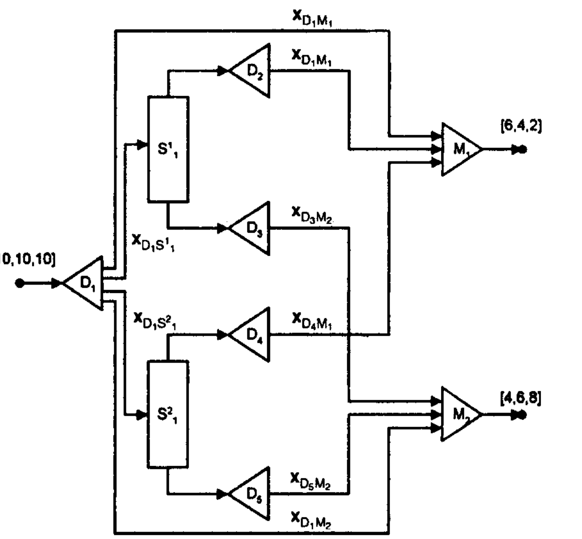
Három több komponensű terméket szeretnénk megkapni három betáplálásból. A minimalizálandó célfüggvény az egyes szétválasztók költségének az összegéből áll. Egy szétválasztó költségét úgy kapjuk, hogy a bemenetére érkezett anyagáram mennyiségét megszorozzuk a szétválasztás nehézségi fokával.

Szuperstruktúrának [5] a *3. ábrán* látható struktúrát tekintette és ez alapján generálták a *4. ábrán* levő optimális hálózatot. A célfüggvény érték, amelyet egy NLP modellből, 113 változó felhasználásával kaptak 138,18 volt. A feladat megoldási ideje 0,74 másodperc (a szerzők által adott érték), egy IBM RS600/530-as számítógépen.

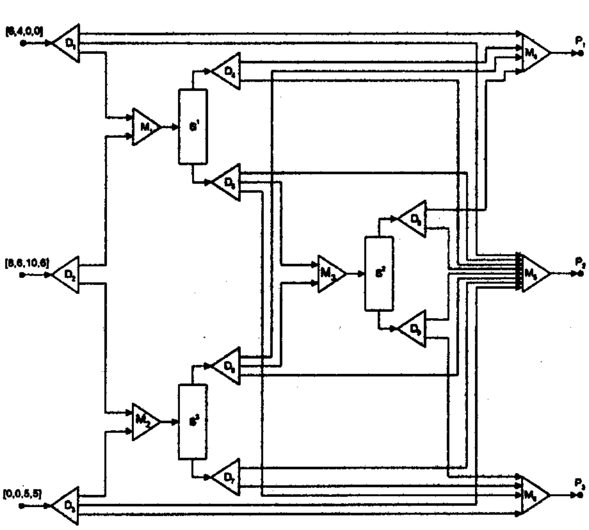
Generáltuk a szigorú szuperstruktúrát az SNS-LMSG algoritmikus segítségével. A belőle származtatott matematikai programozási modell lineáris, amely 90 változót tartalmaz, amelyből 80 a hálózathoz kapcsolódik, például a megosztókhoz. A többi 10 változó pedig a termékekhez tartozik, ugyanis a termékek nem konkrétan adottak, hanem az egyes komponensekhez rendelt korlátozási feltételekkel. Főként az A komponensnek legalább 9-nek kell lennie, a B, C komponens legfeljebb 3 lehet és D komponens egyáltalán nem vagy csak elhanyagolható mennyiségben lehet az 1-es termékben.

A javasolt módszeren alapuló optimális struktúra az *5. ábrán* látható, amelyet 0,55 másodperc alatt kaptunk meg egy 100 MHz-es Pentium processzoros számítógép segítségével. Az LP modell megoldására a GAMS 2.25 BDMLP [6] megoldóját használtuk. Az eredményül kapott célfüggvény érték 104,26.

A *6. ábra* mutatja azt az optimális hálózatot, amelyet úgy kaptunk, hogy összevontuk azokat a szétválasztókat, amelyek ugyanazt a szétválasztási feladatot végzik. Habár ez a hálózat különbözik a *5. ábrán* láthatótól, a költsége ugyanaz, mint a korábbi esetben, mivel a költségfüggvény lineáris. Megjegyezzük, hogy az egyes termékek komponens értékei nem konkrét számokkal, hanem egyenlőtlenségekkel adottak, a termékek mégis megegyeznek az [5] optimális megoldásában kapottakkal.




*2. ábra.* Az 1. példa optimális struktúrája (költség = 12)



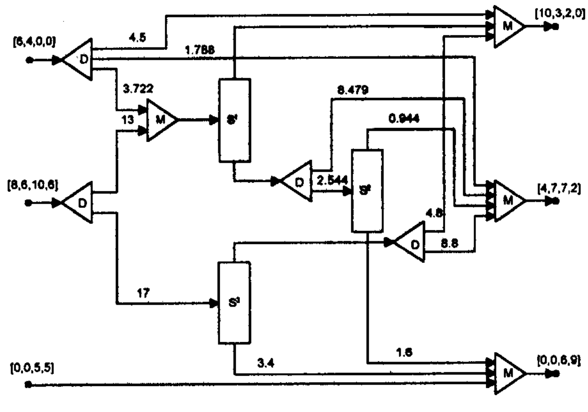

*3. ábra.* A 2. példa [5] szerinti szuperstruktúrája

### *A 3. példa*

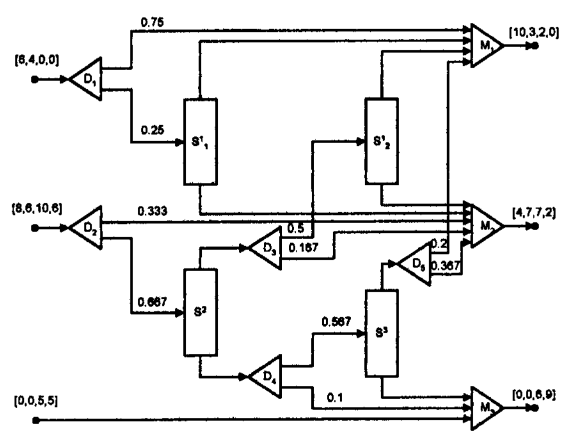
*3. táblázat*

**A 3. példa adatai**

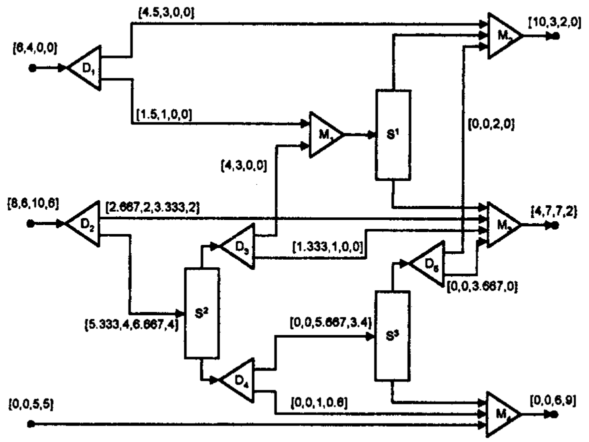
| Komponens | A | B | C | D |
|---|---|---|---|---|
| Betáplálás | 15 | 20 | 10 | 15 |
| Termék 1 | 5 | 10 | 4 | 10 |
| Termék 2 | 10 | 10 | 6 | 5 |
| Nehézségi fok | 2,5 | 3,0 | 1,5 | |



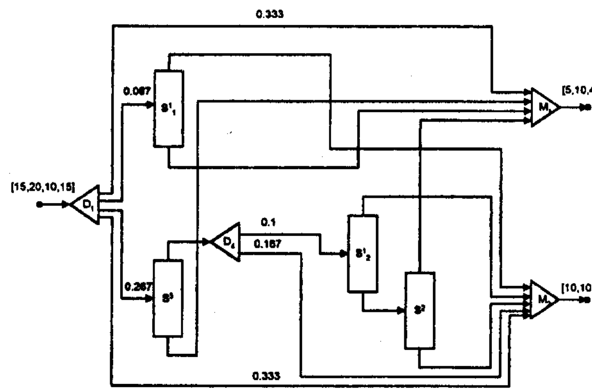

*4. ábra.* A 2. példa [5] szerinti optimális struktúrája (költség = 138,18)




*5. ábra.* A 2. példa optimális struktúrája (költség = 104,26)




*6. ábra.* A 2. példa optimális struktúrája összevont szétválasztókkal (költség = 104,26)




*7. ábra.* A 3. példa optimális struktúrája (költség = 54,25)

Egy négykomponensű betáplálást kell szétválasztani két négykomponensű termékké. A feladat adatai a *3. táblázatban* láthatóak. Tehát a matematikai programozási modellt a szigorú szuperstruktúrából kapjuk az SNS-LMSG algoritmus segítségével. Az eredményül kapott négy szétválasztót tartalmazó optimális struktúra a *7. ábrán* látható, amelynek célfüggvény értéke 54,25. Ez különbözik [5] által kapott három szétválasztós struktúrától, amelynek célfüggvény értéke 55,50.

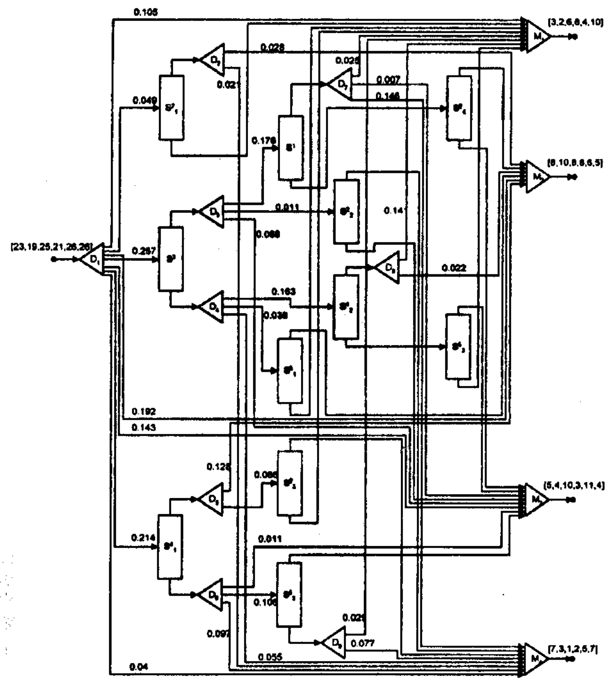
*A 4. példa*

*4. táblázat*

**A 4. példa adatai**

| Komponens | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| Betáplálás | 23 | 19 | 25 | 21 | 26 | 26 |
| Termék 1 | 3 | 2 | 6 | 8 | 4 | 10 |
| Termék 2 | 8 | 10 | 8 | 8 | 6 | 5 |
| Termék 3 | 5 | 4 | 10 | 3 | 11 | 4 |
| Termék 4 | 7 | 3 | 1 | 2 | 5 | 7 |
| Nehézségi fok | 1,5 | 3,0 | 2,0 | 2,5 | 4,0 | |

Egy hatkomponensű betáplálásból szeretnénk kapni négy hatkomponensű terméket. A példa adatait a *4. táblázat* mutatja be. Az eredményül kapott optimális struktúra 11 szétválasztót tartalmaz, a hálózat a *8. ábrán* látható. Az ehhez tartozó célfüggvény érték 330,76, ami 15 %-kal kevesebb, mint a [5] által kapott 388,00. Az általuk kapott struktúra 5 szétválasztóból épült fel. A számolási idő a mi esetünkben 1,65 másodperc volt egy 100 MHz-es Pentium PC-n, szemben a másik eljárás 33 másodpercével, amelyet egy IBM RS600/530-as számítógépen mértek.



*8. ábra.* A 4. példa optimális struktúrája (költség = 330,76)

Az *5. táblázat* összegzi az általunk kapott eredményeket. Ezeket összehasonlítottuk a [5] eredményeivel. A megoldó és a példák elérhetőek a http://www.dcs.vein.hu/capo/demo internet címen.

*5. táblázat*

**Az eredmények összefoglalása és összehasonlítása**

| Példa szám | Komponensek száma | Termékek száma | Változók száma | | Idő | | Az optimális megoldás költségfüggvény értéke | |
|---|---|---|---|---|---|---|---|---|
| | | | [5] | Jelen módszer | [5]* | Jelen módszer** | [5] | Jelen módszer |
| 1 | 3 | 2 | 65 | 22 | 0,13 | 0,22 | 12,00 | 12,00 |
| 2 | 4 | 3 | 113 | 90 | 0,74 | 0,55 | 138,18 | 104,26 |
| 3 | 4 | 2 | 107 | 67 | 1,37 | 0,26 | 55,50 | 54,25 |
| 4 | 6 | 4 | 430 | 1094 | 33,00 | 1,65 | 388,00 | 330,76 |

* IBM RS600/530

** 100MHz Pentium PC

## IRODALOM

[1] *Kovacs, Z. – Z. Ercsey – F. Friedler –L. T. Fan:* Separation-Network Synthesis: Global Optimum through Rigorous Super-Structure, Computers Chem. Engng, *24*, 1881-1900 (2000).

[2] *Kovács, Z. – F. Friedler – L. T. Fan:* Recycling in a Separation Process Structure. AIChE J. *39(6)*, 1087-1089 (1993).

[3] *Thompson, R. W. – C. J. King:* Systematic Synthesis of Separation Schemes. AIChE J. *18(5)*, 941-948 (1972).

[4] *Friedler, F. – K. Tarján – Y. W. Huang – L. T. Fan:* Graph-Theoretic Approach to Process Synthesis: Polynomial Algorithm for Maximal Structure Generation. Computers Chem. Engng *17(9)*, 929-942 (1993).

[5] *Quesada, I. – I. E. Grossmann:* Global Optimization of Bilinear Process Networks with MultiComponens Flows. Computers Chem. Engng 19(12), 1219-1242 (1995).

[6] *Brooke, A.– D. Kenderick – A. Meeraus:* GAMS: A User's Guide, Release 2.25. *GAMS Development Corporation*, 1996.


## ÖSSZEFOGLALÁS


*Heckl István – Kovács Zoltán – Ercsey Zsolt:* **A matematikai szigorúság fontossága a kiindulási struktúra meghatározásakor szétválasztási hálózatok szintézise esetén**

Eddig már sokan és igen alaposan vizsgálták az SNS feladatosztályba tartozó algoritmikus módszereket, hiszen ezen módszerek segítségével lehet megkapni egy SNS probléma optimális megoldását. Azonban úgy tűnik, hogy eddig nem volt ismert olyan módszer, amely algoritmikusan oldja meg az SNS probléma minden lépését, ugyanakkor képes bizonyítani a megoldás optimalitását. Ez annak tulajdonítható, hogy nehéz feladat szisztematikusan előállítani egy megfelelő matematikai programozási modellt. Összehasonlítottunk egy ilyen szigorú szuperstruktúrán alapuló, minden lépésében bizonyított eljárást egy rangos nemzetközi folyóiratban megjelent módszerrel. A tény, hogy számos gyakorlati tényezőt (keverők és megosztók költsége, a csővezeték rendszer bonyolultsága, szivattyúzási energia, karbantarthatóság, irányíthatóság) figyelmen kívül hagytunk, megkérdőjelezheti a kapott struktúra létjogosultságát.



[Magy Kém. Lapja, *56*, 372 (2001)]


## SUMMARY


*I. Heckl – Z. Kovács – Zs. Ercsey:* **Separation-Network Synthesis: Global Optimum through Rigorous Super-Structure**

Algorithmic methods have been extensively explored because of the importance of attaining the optimality of the solutions in separation-network synthesis (SNS). It appears, however, that hitherto no method has been available to algorithmically and rigorously solve every step of an SNS problem and at the same time is capable of ensuring the optimality of the solution. This is attributable to the difficulty in systematically generating appropriate mathematical programming models. We compared such a method based on rigorous superstructure with a method published by a prestigious international journal. It should be cautioned, however, that a variety of practical considerations, such as the costs of mixers and dividers, piping complexity, pumping energy, maintainability, and controllability, may overshadow the desirability of a separation-network.