

# Egy kémiai feladattípus elemzése egészértékű programozás segítségével

Diplomamunka

Írta: Hubik Irén

Alkalmazott matematikus szak

Témavezető:

Vizvári Béla, egyetemi docens

Operációkutatási Tanszék

Eötvös Loránd Tudományegyetem, Természettudományi Kar



Eötvös Loránd Tudományegyetem

Természettudományi Kar

2004

# Tartalomjegyzék

<b>1. Bevezetés</b>	<b>1</b>
1.1. Általánosságban az egészértékű programozásról . . . . .	1
1.2. Kémiai alkalmazás . . . . .	2
<b>2. A csoportelméleti módszer</b>	<b>8</b>
2.1. A csoportfeladat megoldása dinamikus programozással . . . . .	11
2.2. A csoportfeladat optimális megoldásának néhány tulajdonsága . . . . .	15
2.3. A csoportelméleti módszer felhasználása a korlátozás és szétválasztás algoritmusában	16
2.4. A csoportelméleti feladat alkalmazása az (1.2)-ben adott feladatra . . . . .	20
<b>3. A Hilbert bázis meghatározása</b>	<b>25</b>
3.1. Contejean-Devie algoritmus . . . . .	27
3.2. A Contejean-Devie algoritmus módosítása a (3.0.2)-es tétel alapján . . . . .	28
3.3. A Hilbert bázis meghatározása az (1.2)-ben adott feladatra . . . . .	30
<b>Irodalomjegyzék</b>	<b>33</b>

# 1. fejezet

## Bevezetés

Az egészértékű programozás egy kémiai alkalmazásával fogunk foglalkozni. Adva van egy reakció, ami tulajdonképpen nem egy reakcióból áll, hanem több elemi reakció történik, de mi csak a kezdeti és végtermékeket ismerjük. A feladat, hogy meghatározzuk, hogy mik azok az elemi reakciók, amelyek az adott folyamatban lezajlanak. Erre adunk egy egészértékű programozási modellt, és megoldjuk a csoportelméleti módszerrel. Vizsgáljuk azt is, hogy mi az elemi reakciók Hilbert-bázisa, és ezekből a reakciókból előáll-e a teljes reakció.

### 1.1. Általánosságban az egészértékű programozásról

Az egészértékű programozás széles körben elterjedt, mind a tudomány, mind a gazdasági és ipari területeken. Egészértékű programozási feladatok szinte bárhol előfordulhatnak, és bár néhol egy alkalmas közelítő megoldás is jó, ez nem mindig elegendő. Mindig az adott probléma határozza meg, hogyan is kezdjük el megoldani a feladatot. Az egészértékű programozási feladat *NP* teljes, azaz nem tudjuk polinom időben megoldani. A matematika területéről ilyen problémák például a halmazfedési feladatok, míg a gazdaságban például az utazóügynök feladat, különböző raktározási, készletezési, és termelésütemezési feladatok. Elméletben, végig tudnánk nézni minden megoldást, de ezek száma exponenciálisan nő a változók számában.

Jelöljük  $\mathbb{Z}_+^n$ -szal az  $n$ -dimenziós, nemnegatív egész koordinátájú vektorokat.

Egy általános egészértékű feladat a következő alakú:

$$\begin{aligned} \min \quad & c_1^\top x + c_2^\top y \\ & Ax + By = b_1 \\ & Px + Qy \leq b_2 \\ & x \in \mathbb{Z}_+^n \\ & y \in \mathbb{Z}^m \end{aligned} \tag{1.1}$$

ahol  $A: (k \times n)$ ,  $B: (k \times m)$ ,  $P: (l \times n)$ ,  $Q: (l \times m)$ -es mátrixok,  $c^\top = (c_1^\top, c_2^\top)$ :  $(n + m)$ -es és  $b$ :  $(l + k)$ -as vektorok.

Többféle megoldási lehetőség és módszer létezik. Vannak heurisztikus algoritmusok, amelyek már egy meglévő megengedett megoldásból indulnak ki, és azt módosítják, amíg jobb megoldást tudnak így elérni. Ezekre a módszerekre létezik a *TSPLIB* mérőszám, ami mutatja, hogy az optimális megoldástól általában (tesztfeladatokra futatják őket, és így mérik) mennyire tér el az így kapott eredmény. Legtöbbször érdemes több ilyen heurisztikát is alkalmazni egymás után.

Egy másik megoldási lehetőség, hogy a feladat valamilyen relaxáltját oldjuk meg. Ez azt jelenti, hogy valamit elhagyunk a feltételek közül, ami nélkül egyszerűbb feladathoz jutunk. Az így kapott feladatot megoldva kapunk egy alsó, illetve felső korlátot a feladat minimum, illetve maximum értékére. Az egyik legelterjedtebb, mikor az egészértékűségi feltételt hagyják el, és a lineáris programozási relaxáltat oldják meg. Erre jól működő megoldó programok léteznek.

Erre aztán akár felépíthetünk egy korlátozás, és szétválasztás módszert is. Ez azt jelenti, hogy felbontjuk a megoldások halmazát, majd ezeken vizsgáljuk az optimumot, és azon a részhalmazon haladunk tovább, ahol a jobb optimumot tudjuk elérni. Léteznek vágás-típusú módszerek is, amik az egész pontok konvex burkát közelítik, és így állítanak elő megoldást.

Legtöbbször a feladat tulajdonságából adódóan, gyors megoldásra van szükség, ilyenek például a különböző ütemezési feladatok, ami újabb korlátot ad a feladatmegoldásnak.

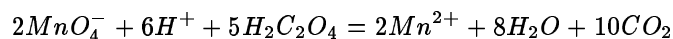
Ezeket a módszereket legtöbbször nem önmagukban használják, hanem kombinálják egymással, annak érdekében, hogy minél jobb optimumot tudjanak elérni.

## 1.2. Kémiai alkalmazás

Egy adott folyamat reakció útvonalainak meghatározása fontos elméleti kérdés, mindamelllett ipari fontossága is van. Szeretnénk pontosan ismerni mi zajlik egy folyamaton belül, és az ipar ezt szeretné szabályozni is, és azt a reakció útvonalat meghatározni, ami számára valamilyen szempontból a legjobb, például a legkisebb költségű.

A probléma tulajdonképpen kétszintű. Az egyik, hogy meghatározzuk, milyen elemi reakciók mehetnek egyáltalán végbe, és ezek után a második feladat annak leírása, hogy áll ezekből elő a folyamat. Ez természetesen több úton mehet végbe.

Az első problémával [5] foglalkozik részletesen. Hogy szemléltethessük, tekintsük a következő reakciót:



Elemi reakciónak azt nevezzük, amelyben legfeljebb két anyag reagál (molekula, ion, atom, illetve gyök). Tételezzük fel egy közvetítő oldat jelenlétét. Adva vannak azok az anyagok, amelyek a közvetítő oldatban benne vannak, ezeket a kémikusok mérésekkel állapítják meg. A mi példánkban összesen 19 anyag. Ezek 4 atomból és elektronokból állnak elő. Ezt mutatja a következő táblázat:

	<i>C</i>	<i>H</i>	<i>Mn</i>	<i>O</i>	<i>e</i>
$H_2C_2O_4$	2	2	0	4	0
$HC_2O_4^-$	2	1	0	4	1
$H^+$	0	1	0	0	-1
$C_2O_4^{2-}$	2	0	0	4	2
$Mn^{2+}$	0	0	1	0	-2
$MnC_2O_4$	2	0	1	4	0
$MnO_4^-$	0	0	1	4	1
$MnO_2$	0	0	1	2	0
$Mn^{3+}$	0	0	1	0	-3
$CO_2$	1	0	0	2	0
$H_2O$	0	2	0	1	0
$[MnO_2, H_2C_2O_4]$	2	2	1	6	0
$CO_2^-$	1	0	0	2	1
$[Mn(C_2O_4)]^+$	2	0	1	4	-1
$[Mn(C_2O_4)_2]^-$	4	0	1	8	1
$[MnC_2O_4, MnO_4^-, H^+]$	2	1	2	8	0
$[MnC_2O_4^{2+}, MnO_3^-]^+$	2	0	2	7	-1
$[MnC_2O_4^{2+}, MnO_3^-, H^+]^{2+}$	2	1	2	7	-2
$[H^+, MnO_2, H_2C_2O - 4]^+$	2	3	1	6	-1

A táblázat egy mátrixot ad, melynek transzponáltját  $D$ -vel jelöljük.  $D \in \mathbb{Z}^{5 \times 19}$ .  $D$  nevezhető atom mátrixnak, hiszen azt mondja meg, hogy az egyes jelenlévő anyagokban a különböző atomokból és az elektronból mennyi van. Az utóbbiakból csak a hiányt, vagy a többletet mutatja. Egy elemi reakció többféle lehet. Vagy egy létező molekula felbomlása, például ha az oldószer egy molekulája nekiütökzik és "felrobbantja", vagy két (azonos, vagy különböző) molekula ütközése és átalakulása nyomán jön létre. Jelölje  $d$  azt a vektort, ami vagy egy  $D$ -beli oszlop, ha egy elem felbomlását szeretnénk megkapni, vagy két  $D$ -beli oszlop összege, ha két anyag találkozásakor létrejövő reakciót akarjuk leírni. Ekkor egy  $x \in \mathbb{Z}_+^{19}$  vektor fogja megadni, hogy milyen anyagok jönnek létre:

$$Dx = d,$$

feltéve, hogy  $x$ -ben a  $d$ -t előállító oszlopokhoz tartozó komponensek 0-k. Például:  $H_2C_2O_4$  dekom-

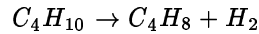
pozíciója. Ennek az anyagnak a  $d = \begin{pmatrix} 2 \\ 2 \\ 0 \\ 4 \\ 0 \end{pmatrix}$  vektor felel meg. Ekkor az

$$x = \left( 0 \ 0 \ 2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0 \ 0 \ 0 \ 0 \right)^T$$

megoldás a  $H_2C_2O_4 = 2H^+ + 2CO_2^-$  reakciót fogja meghatározni.

Ebben az esetben összesen  $19 + 19 + \binom{19}{2} = 209$  lineáris egyenletrendszert kell megoldani, az egész számok halmazán. A kapott reakciók számát csökkenteni lehet kémiai vizsgálatokkal.

[5] alapján, meg tudjuk adni a folyamatban lezajló elemi reakciókat. Tekintsük a [3] által adott egyszerűbb példát. A folyamat, azaz az összetett reakció a következő:



A bután dehidrogenizálása. Temkin (1971) határozta meg a folyamatban lévő elemi reakciókat:

- (1  $\rightarrow$ )  $C_4H_{10} + \lambda \rightarrow C_4H_8\lambda + H_2$
- (1  $\leftarrow$ )  $C_4H_8\lambda + H_2 \rightarrow C_4H_{10} + \lambda$
- (2  $\rightarrow$ )  $C_4H_8\lambda \rightarrow C_4H_8 + \lambda$
- (2  $\leftarrow$ )  $C_4H_8 + \lambda \rightarrow C_4H_8\lambda$
- (3  $\rightarrow$ )  $C_4H_8\lambda \rightarrow C_4H_6\lambda + H_2$
- (3  $\leftarrow$ )  $C_4H_6\lambda + H_2 \rightarrow C_4H_8\lambda$
- (4  $\rightarrow$ )  $C_4H_6\lambda \rightarrow C_4H_6 + \lambda$
- (4  $\leftarrow$ )  $C_4H_6 + \lambda \rightarrow C_4H_6\lambda$
- (5  $\rightarrow$ )  $C_4H_{10} + \lambda + C_4H_6\lambda \rightarrow 2C_4H_8\lambda$
- (5  $\leftarrow$ )  $2C_4H_8\lambda \rightarrow C_4H_{10} + \lambda + C_4H_6\lambda$

[3]-ban felteszik, hogy minden elemi reakció, mindkét irányban végbe tud menni. Bevezettek 6 axiómát, amelyek a folyamat lefolyásának szabályait adják meg, és amelyeknek teljesülniük kell minden egyes reakció útvonalra. (Azaz minden olyan reakció részhalmazra, amelyek előállítják az összetett reakciót.) Kiindulási anyagoknak fogjuk nevezni az összetett reakció kiindulási anyagait, és célanyagoknak a létrejövő anyagait. Ebben a példában, kiindulási anyag a  $C_4H_{10}$ , és célanyagok  $C_4H_8$ ,  $H_2$ . Közbülső anyagnak nevezünk minden olyan anyagot, amely a folyamatban létrejön, vagy keletkezik, a kiindulási, és célanyagokon kívül. Az axiómák:

- (R1) Minden célanyag előáll az útvonalban szereplő reakciókból.
- (R2) Minden kiindulási anyagot felhasználnak az útvonalban lévő reakciók.
- (R3) Minden közbülső anyag, ami előáll valamely reakcióból, az teljesen felhasználandó egy, vagy több reakcióban, amely az útvonalban szerepel. Illetve, minden közbülső anyag, amit felhasználunk valamely reakcióhoz, az előáll egy, vagy több reakcióból.
- (R4) Az útvonal csak előre definiált reakciókat tartalmaz.
- (R5) Az útvonal körmentes, azaz nincs olyan reakcióhalmaz (részútvonal) az útvonalban, amelyre a kiinduló és célanyagok megegyeznének.
- (R6) Legalább egy elemi reakció felhasználja a kiinduló anyagot.

Az elemi reakciókat, és az összetett reakciót le tudjuk írni vektorokkal aszerint, hogy melyek a kiindulási és célanyagaik. A vektor legyen a következő: Minden koordináta egy anyagnak felel meg

aszerint, hogy ha az az anyag kiindulási anyaga a reakciónak, akkor negatív szám, ha célanyag, akkor pozitív, egyébként 0. A számok az a mennyiséget jelölik, ahány egységnyi anyag felhasználódik, illetve létrejön a reakcióban. A példában összesen 7 anyag van, ezért most 7 dimenziós vektoroknak felelnek meg. Azaz a reakciókat leíró mátrix:

	1 →	1 ←	2 →	2 ←	3 →	3 ←	4 →	4 ←	5 →	5 ←	a folyamat
$C_4H_{10}$	-1	1	0	0	0	0	0	0	-1	1	-1
$\lambda$	-1	1	1	-1	0	0	1	-1	-1	1	0
$C_4H_8\lambda$	1	-1	-1	1	-1	1	0	0	2	-2	0
$H_2$	1	-1	0	0	1	-1	0	0	0	0	1
$C_4H_8$	0	0	1	-1	0	0	0	0	0	0	1
$C_4H_6\lambda$	0	0	0	0	1	-1	-1	1	-1	1	0
$C_4H_6$	0	0	0	0	0	0	1	-1	0	0	0

Ekkor az előző példához hasonlóan,  $A$  jelölje a táblázat első 10 oszlopa által adott mátrixot, míg  $b$  legyen az utolsó oszlop. Ekkor a következő egyenletrendszer és az egészértékűségi feltétel kielégíti az axiómákat, kivéve a körmentességet. De azt is biztosítani tudjuk a csupa 1-ből álló célfüggvényvektorral.

$$Ax = ab: \quad (A \in \mathbb{Z}^{7 \times 10}, b \in \mathbb{Z}^7) \quad x \in \mathbb{Z}_+^{10}, a \in \mathbb{Z}_+, a \neq 0.$$

A jelenlegi feladatunk olyan, hogy mindkét irányú reakció szerepel benne, ezért a feladatot átalakíthatjuk egy ekvivalens alakba, úgy, hogy a mátrixunk, csak az odairányú reakciónak megfelelő oszlopot tartalmazzák, és ha a neki megfelelő komponens pozitív, akkor a reakció az adott irányban fog végbemenni, míg ha negatív, akkor visszafelé. Ez következik abból, hogy az útvonal körmentes, azaz nem szerepelhet egyszerre az oda- és visszareakció is. A jobboldali  $b$  vektort is átvihetjük a másik oldalra, és akkor  $a$ -t is változóként tekintve, egy homogén rendszert kapunk. Azaz

$$\bar{A} = \begin{pmatrix} -1 & 0 & 0 & 0 & -1 & 1 \\ -1 & 1 & 0 & 1 & -1 & 0 \\ 1 & -1 & -1 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \bar{b} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Vizsgálunk kell az

$$\bar{A}x = \bar{b}, \quad x \in \mathbb{Z}^6$$

rendszert, az

$$x_6 > 0$$

feltétel mellett, ami ekvivalens az  $x_6 \geq 1$  korláttal. Az egyenletrendszert megoldva a következő

megoldásokat kapjuk:

$$x = \begin{pmatrix} a - c \\ a \\ c \\ 0 \\ c \\ a \end{pmatrix},$$

ahol  $a \geq 1$  és  $c$  tetszőleges egész számok. Ekkor  $a$  és  $c$  számok nagyságától függően a következő eredményeket kapjuk:

	a megoldás előjelei	reakciók
$c < 0 < a$	(+, +, -, 0, -, +)	1-es és 2-es oda-, 3-as és 5-ös visszafelé
$c = 0 < a$	(+, +, 0, 0, 0, +)	1-es, 2-es odafelé
$0 < c < a$	(+, +, +, 0, +, +)	1,2,3,és 5-ös odafelé
$0 < c = a$	(0, +, +, 0, +, +)	2,3,és 5-ös odafelé
$0 < a < c$	(-, +, +, 0, +, +)	1-es vissza-, 2,3,és 5-ös odafelé.

Ebből az eredményből látszik, hogy a 4-es reakció egyik útvonalban sem fut le, míg a 2-es reakció mindegyiknél odafelé minden esetben végbemegy. Az azonosan 1 célfüggvény, a feladat ezen alakjában a következő alakú:

$$|a - c| + 2|a| + 2|c|.$$

Ennek segítségével tudjuk meghatározni, hogy az egyes intervallumokon belül, melyik a körmentes megoldás. Ilyen módszerrel meghatározhatjuk az összes reakció útvonalat:

- Ha  $c < 0 < a$ , akkor a célfüggvény:  $3a - 3c$ . Ez akkor minimális, ha  $a = 1, c = -1$ . Ez azt jelenti, hogy

1→ reakció	2-szer
2→ reakció	1-szer
3← reakció	1-szer
5← reakció	1-szer
<hr/>	
teljes reakció	1-szer megy végbe

- Ha  $c = 0 < a$ , akkor a célfüggvény:  $3a$ , azaz  $a = 1, c = 0$  az optimális.

1→ reakció	1-szer
2→ reakció	1-szer
<hr/>	
teljes reakció	1-szer megy végbe

- Ha  $0 < c < a$ , akkor a célfüggvény:  $3a + c$ . Ha itt  $a = 1$  lenne, akkor nem lenne egész  $c$  érték, így a célfüggvény akkor minimális, ha  $a = 2, c = 1$ .



1→ reakció	1-szer
2→ reakció	2-szer
3→ reakció	1-szer
5→ reakció	1-szer
<hr/>	
teljes reakció	2-szer megy végbe

- Ha  $0 < c = a$ , akkor a célfüggvény:  $4a$ . Ekkor az  $a = c = 1$  az optimális.

2→ reakció	1-szer
3→ reakció	1-szer
5→ reakció	1-szer
<hr/>	
teljes reakció	1-szer megy végbe

- Ha  $0 < a < c$ , akkor a célfüggvény:  $a + 3c$ , azaz  $a = 1, c = 2$  minimális. Ekkor:

1← reakció	1-szer
2→ reakció	1-szer
3→ reakció	2-szer
5→ reakció	2-szer
<hr/>	
teljes reakció	1-szer megy végbe

Azt is észrevehetjük, hogy  $0 < c < a$  eset előáll a  $c = 0 < a$ , és a  $0 < c = a$  esetek összegeként.

Tehát a reakció útvonalakat az azonosan 1 célfüggvénnyel határozhatjuk meg, és így kaphatunk képet a folyamat hátteréről. Másrészt, ha célfüggvénynek például az adott elemi reakció előállításának költségét adjuk, akkor ipari haszna is lehet, hiszen meghatározhatjuk azt az útvonalat, ami a legköltségkímélőbb és optimálisan állíthatjuk elő az összetett reakciót. Feltehetjük, hogy a költségek pozitívak, így garantálni tudjuk ismét a körmentességet. (Hiszen így a kör súlya is pozitív és azt elhagyva jobbat kapnánk.)

Az elemi reakciók mátrixa, többnyire alacsony rangú, ezért jól alkalmazható rájuk például a csoportelméleti módszer.

Egy másik lehetőség, hogy az elemi reakciók által kifeszített kúp Hilbert-bázisát vizsgáljuk. Feleltessünk meg minden reakciónak egy vektort, ugyanúgy mint az előbb. Ezek a vektorok egy kúpot feszítenek ki. Ha ismerjük ezen kúp Hilbert-bázisát, akkor minden reakció a kútból előáll ezen Hilbert-bázis elemeinek egész együtthatós lineáris kombinációjaként. Azt kell ezután megvizsgálunk, hogy az összetett folyamatunk benne van-e ebben a kúpban, azaz előáll-e Hilbert-bázis-beli elemek összegeként.

## 2. fejezet

# A csoportelméleti módszer

Ebben a fejezetben az egészértékű programozási feladat megoldását relaxáció segítségével fogjuk megadni, és így egy olyan feladathoz jutunk, amelyben a feltételek együtthatói egy csoport elemei.

A megoldani kívánt egészértékű feladat:

$$\begin{aligned} \min c^\top x \\ Ax = b \\ x \in \mathbb{Z}_+^n \end{aligned} \tag{2.1}$$

ahol  $A \in \mathbb{Z}^{m \times n}$  teljes rangú, ( $m \leq n$ ),  $b \in \mathbb{Z}^m$ ,  $c \in \mathbb{Z}^n$ . Megoldjuk a lineáris programozási relaxáltat és felbontjuk az  $A$  mátrixot, az optimális bázis szerint:

$$A = (B, N)$$

( $B$  a lineáris programozás szerinti optimális bázis), így a megfelelő vektorok is felbomlanak két részre:

$$c = \begin{pmatrix} c_B \\ c_N \end{pmatrix}, x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$$

Ekkor az  $Ax = b$  átírható

$$Bx_B + Nx_N = b$$

alakba. Mivel feltettük, hogy  $A$  teljes rangú,  $B$  optimális bázisa, ezért létezik a  $B$  mátrix inverze. Így  $x_B$  a következő alakba írható:

$$x_B = B^{-1}b - B^{-1}Nx_N.$$

Innen a feladat következő ekvivalens alakját kapjuk:

$$\begin{aligned}
 \min & \quad (c_N^\top - c_B^\top B^{-1}N) x_N + c_B^\top B^{-1}b \\
 & \quad x_B = B^{-1}b - B^{-1}N x_N \\
 & \quad x_B \in \mathbb{Z}_+^m \\
 & \quad x_N \in \mathbb{Z}_+^{n-m}
 \end{aligned} \tag{2.2}$$

A feladatnak ebből az alakjából készítünk relaxációt. Többnyire az egészértékűségi feltételt szokták elhagyni, és így lineáris programozási feladathoz jutnak, amire vannak megoldási lehetőségek. Most ehelyett  $x_B$  pozitivitási megkötését hagyjuk el.  $x_B$  egészértékűsége miatt, az  $x_B = B^{-1}b - B^{-1}N x_N$ -nek is egésznek kell lennie. Így a feladat következő alakját kapjuk:

$$\begin{aligned}
 \min & \quad (c_N^\top - c_B^\top B^{-1}N) x_N + c_B^\top B^{-1}b \\
 & \quad B^{-1}N x_N \equiv B^{-1}b \pmod{1} \\
 & \quad x_N \in \mathbb{Z}_+^{n-m}
 \end{aligned} \tag{2.3}$$

A kongruencia minden sorra vonatkozik.

Az  $A$  mátrix elemeinek egészértékűségéből következik, hogy a feltétel együtthatói, és a jobboldal egy véges csoport elemei. Ezt a csoportot a  $B^{-1}$  mátrix oszlopai generálják, a komponensenkénti mod 1 összeadással. Ennek a csoportnak a  $B^{-1}$  mátrix elemeit explicit módon megadó algebrai tétel következtében legfeljebb  $|\det(B)|^m$  eleme lehet.

Hogy a csoport elemeit meg tudjuk adni és a feladatot megoldjuk, szükségünk van a mátrixok Smith-féle normálalakjára.

**2.0.1. Tétel.** *Legyen  $B \in \mathbb{Z}^{m \times m}$  nem szinguláris mátrix. Ekkor léteznek olyan, szintén  $(m \times m)$ -es  $P$  és  $Q$  mátrixok, hogy*

i.  $|\det(P)| = |\det(Q)| = 1$ ,

ii. *léteznek olyan  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m$  egészek, hogy*

$$PBQ = \begin{pmatrix} \varepsilon_1 & 0 & \dots & 0 \\ 0 & \varepsilon_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \varepsilon_m \end{pmatrix}$$

iii.  $\varepsilon_i \mid \varepsilon_{i+1}$ ,  $i = 1, \dots, m-1$ .

Azaz a mátrix olyan alakra transzformálható, hogy a fődiagonálisban lévő elemek, az előzőek többszörösei. Ennek egy következményét használjuk majd fel a csoportfeladatnál.

**2.0.2. Következmény.** *Legyen  $d = |\det(B)|$ . Legyen  $p^k$  a  $d$  szám legnagyobb kitevős prím faktora. Ekkor az előző tétel ii-pontjában, a  $PBQ$  transzformált alakban legfeljebb  $k$  darab  $\varepsilon_i$  lehet különböző az 1 vagy  $-1$  számoktól.*

Mivel a  $PBQ$  mátrix diagonális, ezért könnyen megadhatjuk az inverzét. Jelöljük  $\varepsilon$ -nal a  $PBQ$  mátrixot, ekkor

$$\varepsilon^{-1} = \begin{pmatrix} \frac{1}{\varepsilon_1} & 0 & \dots & 0 \\ 0 & \frac{1}{\varepsilon_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\varepsilon_m} \end{pmatrix}$$

Ekkor világos, hogy

$$PBQ\varepsilon^{-1} = E,$$

ahol  $E$  az  $m \times m$ -es egységmátrixot jelöli. Ezt az egyenletet beszorozva balról  $P^{-1}$ -el, jobbról  $P$ -vel a következőt kapjuk:

$$BQ\varepsilon^{-1}P = E.$$

Ebből látszik, hogy a  $B$  mátrix inverze:

$$B^{-1} = Q\varepsilon^{-1}P.$$

Most visszatérhetünk a feladathoz, de nem  $B^{-1}$ -el fogunk beszorozni, mint (2.2)-ben, hanem  $\varepsilon^{-1}P$ -vel. Ez szintén nem szinguláris mátrix, így továbbra is ekvivalens feladatot fogunk kapni. Az előbbieket szerint  $\varepsilon^{-1}P = Q^{-1}B^{-1}$ . A felbontás tulajdonságai miatt

$$|\det(Q)| = |\det(Q^{-1})| = 1,$$

ezért  $Q^{-1}$  mátrixnak is egészek az elemei. A következő állítás szerint ezért a  $Q^{-1}$  mátrixszal való szorzás nem változtat a feladat kongruenciáján.

**2.0.3. Állítás.** *Legyen  $r$  és  $s$  két rögzített pozitív egész,  $Y$  tetszőleges  $(r \times s)$  méretű mátrix,  $w$   $r$ -dimenziós vektor. Legyen továbbá  $V \in \mathbb{Z}^{r \times r}$  mátrix, melyre  $|\det(V)| = 1$ . Ekkor tetszőleges  $x \in \mathbb{R}^s$  esetén az*

$$Yx \equiv w \pmod{1}$$

*kongruencia akkor és csak akkor áll fenn, ha*

$$VYx \equiv Vw \pmod{1},$$

*ahol a kongruencia soronként értendő.*

**Bizonyítás:** Tegyük fel, hogy  $Yx \equiv w \pmod{1}$  igaz, akkor  $\exists z \in \mathbb{Z}^r$  vektor, hogy

$$Yx - w = z.$$

Ezt megszorozva balról a  $V$  mátrixszal, kapjuk:

$$VYx - Vw = Vz.$$

Mivel  $V$  egészekből álló mátrix, így  $Vz$  is egész, amiből  $VYx \equiv Vw \pmod{1}$  következik. Visszafelé ugyanígy bizonyítható, csak most  $V^{-1}$  szintén egész mátrixszal kell szorozni.  $\square$

Az állítás szerint a feladatunk ekvivalens alakja:

$$\begin{aligned} \min \quad & (c_N^\top - c_B^\top B^{-1}N) x_N + c_B^\top B^{-1}b \\ & \varepsilon^{-1}P^{-1}N x_N \equiv \varepsilon^{-1}P^{-1}b \pmod{1} \\ & x_N \in \mathbb{Z}_+^{n-m} \end{aligned} \quad (2.4)$$

Az  $\varepsilon^{-1}$  mátrix diagonális, ezért a vele való szorzás azt jelenti, hogy minden sort beszorunk a hozzá tartozó  $\frac{1}{\varepsilon_i}$ -vel ( $i = 1, \dots, m$ ).  $P$  és  $N$  is egész együtthatós mátrixok, ezért ha  $\varepsilon_i = \pm 1$ , akkor a kongruencia tetszőleges egész  $x_N$  vektor esetén fenn fog állni. Most tudjuk felhasználni a (2.0.2.)-es következményt, hogy  $\varepsilon$  diagonálisában, csak legfeljebb annyi különbözik  $\pm 1$ -től, mint  $d = |\det(B)|$  prímtényező felbontásában szereplő legnagyobb kitevő. Általában ez jelentősen csökkenti a feltételek számát.

Jelölje  $p_i$  a  $PN$  mátrix  $i$ -dik sorát, és  $q_i$  a  $Pb$  vektor  $i$ -dik elemét. Tegyük fel, hogy az utolsó  $r$   $\varepsilon_i$  különbözik  $\pm 1$ -től, ahol  $r : 0 \leq r \leq m$ . Ekkor a feladat ekvivalens alakja:

$$\begin{aligned} \min \quad & (c_N^\top - c_B^\top B^{-1}N) x_N + c_B^\top B^{-1}b \\ & p_i x_N \equiv q_i \pmod{\varepsilon_i}, \quad i = m - r + 1, \dots, m \\ & x_N \in \mathbb{Z}_+^{n-m} \end{aligned} \quad (2.5)$$

Ebből látható, hogy a kongruenciarendszer bal oldalán legfeljebb  $d$  különböző vektor állhat, melyek a komponensenkénti  $(\text{mod } \varepsilon_i)$  vett összeadásra nézve egy legfeljebb  $d$ -edrendű csoportot alkotnak. Tehát így javítani tudtuk korábbi  $d^m$ -es becslésünket.

## 2.1. A csoportfeladat megoldása dinamikus programozással

Az egyszerűbb jelölés kedvéért vezessük be a következőket:  $t := n - m$ ,  $x := x_N$ . Legyenek a csoport generáló elemei, azaz a  $PN$  mátrix oszlopai  $\alpha_1, \alpha_2, \dots, \alpha_t$ , és a jobb oldalon álló elem  $Pb = \beta$ . Jelölje a csoportbeli összeadást  $\oplus$ , a csoport nullemeit  $0$ , és a célfüggvény vektort  $\gamma = c_N^\top - c_B^\top B^{-1}N$ . Az általánosság megszorítása nélkül feltehetjük, hogy

$$\alpha_i \neq \alpha_j, \quad i \neq j.$$

Így a csoportfeladat:

$$\begin{aligned} \min \quad & \sum_{j=1}^t \gamma_j x_j \\ & \sum_{j=1}^t \alpha_j x_j = \beta \\ & x \in \mathbb{Z}_+^t \end{aligned} \quad (2.6)$$

Jelölje  $G$  az  $\alpha_i, i = 1, \dots, t$  elemek által generált részcsoport elemeinek halmazát, azaz

$$G = \left\{ \sum_{j=1}^t \oplus \alpha_j x_j : x \in \mathbb{Z}_+^t \right\}.$$

$G$  véges halmaz, elemeinek száma legfeljebb  $d$ .  $G$  nem az összes  $r$ -dimenziós,  $(\text{mod } \varepsilon_i)$  komponensenként vett vektor additív csoportja, hanem ennek csak a részcsoportja, amit az  $\alpha_i$ -k  $(i = 1, \dots, t)$  feszítenek ki. Így előfordulhat, hogy  $\beta \notin G$ . Ez azt jelentené, hogy a csoportfeladatnak nincs megengedett megoldása. Ebből az is következne, hogy az eredeti egészértékű programozási feladatnak sincs megengedett megoldása, hiszen a csoportfeladat ennek relaxációja.

Ha

$$\beta \in G,$$

akkor a csoportfeladat ekvivalens egy gráfban legrövidebb utat kereső feladattal. Legyen ez a gráf a következő: a csúcsok halmaza  $G$ , az élek:

$$E = \{(\mu, \nu) : \mu, \nu \in G; \exists j : \mu \oplus \alpha_j = \nu\},$$

a  $(\mu, \mu \oplus \alpha_j)$  él költsége pedig  $\gamma_j$ . Ebben a gráfban keressük a 0-ból  $\beta$ -ba vezető legrövidebb utat. Ilyen optimális út akkor és csak akkor létezik, ha a gráfban nincs negatív kör. Erre szükségünk van ahhoz, hogy a dinamikus programozási algoritmus megtalálja egy legrövidebb utat. Negatív kör akkor és csak akkor van a gráfban, ha

$$\inf \left\{ \gamma^\top x : x \in \mathbb{Z}_+^t; \sum_{j=1}^t \oplus \alpha_j x_j = 0 \right\} < 0. \quad (2.7)$$

Most fog nagyon fontos szerepet játszani, hogy miért a lineáris programozás szerinti optimális bázis szerint osztottuk fel az eredeti mátrixunkat. Lineáris programozásban optimális bázis esetén a  $\gamma$  célfüggvény vektor nemnegatív. Ha nem tettük volna ezt fel, és létezne egy  $j$  index, melyre

$$\gamma_j < 0.$$

Mivel  $G$  véges csoport, ezért elemeinek rendje véges, azaz  $\exists k \in \mathbb{Z}_+$ , hogy  $k\alpha_j = 0$ , így

$$k\gamma_j < 0,$$

és teljesül (2.7).

Ezek alapján már fel lehet építeni, egy dinamikus programozáson alapuló legrövidebb út kereső algoritmust, amely megoldja a feladatot. Ehhez persze fel kell tenni, hogy a számítógépnek van elegendően nagy memóriája, a csoportelemekhez tartozó összes adat tárolásához. Szükségünk van arra is, hogy egyszerűen tudjunk hivatkozni a csoportelemekre. Felhasználva a  $p_i x \equiv q_i \pmod{\varepsilon_i}$  kongruenciát, minden csoportelem egyértelműen felírható egy  $u$ ,  $r$ -dimenziós vektorként:

$$u \in \mathbb{Z}^r; \quad 0 \leq u_i \leq \varepsilon_i - 1, \quad i = 1, 2, \dots, r.$$

Ez pedig egyértelműen megfeleltethető egy 0 és  $d - 1$  közé eső egésznek, az  $\varepsilon_i$  ( $i = 1, \dots, r$ ) alapszámokkal meghatározott általánosított számrendszerben a következő módon:

$$u \mapsto \sum_{i=1}^r u_i \prod_{k=1}^{i-1} \varepsilon_k.$$

Így sorba rendeztük a csoportelemeket.

Vezessük be a következő jelöléseket:

$\delta(q)$ : a  $q$  indexű csoportelem,

$c(\mu), c(q)$ : a  $\mu$ , illetve a  $\delta(q)$  csoportelemhez vezető eddig megtalált legrövidebb út,

$m(\nu)$ : a  $\nu$ -höz vezető, eddig megtalált legrövidebb út utolsó élének típusa, azaz ha az út utolsó éle  $(\mu, \mu \oplus \alpha_j = \nu)$ , akkor  $j$ ,

$w(q)$ : a  $\delta(q)$  csoportelem állapotváltozója, értéke true, ha  $\delta(q)$ -ból, mint a gráf csúcsából még nem léptünk tovább, különben false,

$z$ : egy valós értékű segédváltozó.

A dinamikus programozás a Bellman egyenletek kielégítésével adja meg a gráf minden pontjához a legrövidebb utat, az egyenletek mostani alakja, a következő:

$$c(0) = 0,$$

$$c(\nu) = \min \{c(\mu) + \gamma_j : \mu \oplus \alpha_j = \nu\}$$

Az algoritmusban mindig abból a legkisebb  $c(q)$  értékkel rendelkező  $\delta(q)$  csúcsból / csoportelemből lépünk tovább, amelyből ezt még nem tettük meg, mert itt már nem érhető el javulás a nemnegatív költségek miatt.

Az algoritmus:

```

1  Begin
2       $c(0) := 0;$ 
3       $w(0) := \text{true};$ 
4      for  $k := 1$  to  $d - 1$  do
5          begin
6               $c(k) := +\infty;$ 
7               $w(k) := \text{true};$ 
8          end;
9       $q := 0;$ 

```

```

10   for  $k := 0$  to  $d - 2$  do
11   begin
12        $z := +\infty$ ;
13   for  $j := 1$  to  $d - 1$  do
14       if  $c(j) < z$  and  $w(j)$  then
15           begin
16                $q := j$ ;
17                $z := c(j)$ ;
18           end;
19        $w(q) := \text{false}$ ;
20   for  $j = 1$  to  $t$  do
21       begin
22            $\phi := \delta(q) \oplus \alpha_j$ ;
23       if  $c(\phi) > c(q) + \gamma_j$  then
24           begin
25                $c(\phi) := c(q) + \gamma_j$ ;
26                $m(\phi) := j$ ;
27           end;
28       end;
29   end;
30 end;

```

Az első 9 sor a változók beállítása, és a kezdeti értékek meghatározása. Utána következik a fő ciklus, ami két másik ciklust tartalmaz. Az első, a 13-diktól a 18-dik sorig kiválasztja azt a csúcsot, ahonnét továbblépünk. A második, a 20-dik sortól, a továbblépés. A fő ciklus  $d - 1$ -szer fut le, minden lépésben továbblép egy csúcsból, és aztán abból többször nem lép, ezt azzal tudjuk biztosítani, hogy mindig a legkisebb potenciálértékű csúcsból lépünk tovább. A fő ciklus  $d - 1$ -szer, a két belső ciklus  $d - 1$ -szer és  $t$ -szer fut le. Így az algoritmus lépésszáma:  $O(d^2 + dt)$ .

Az algoritmus meghatározta a  $\beta$ -ba vezető legrövidebb utat, ebből ezután meg kell adnunk a csoportfeladat optimális  $x$  megoldását. Minden csoportelemhez az algoritmus eltárolta, az oda vezető legrövidebb út utolsó élének típusát, így a következő algoritmussal meghatározhatjuk az  $x$ -et.



```

1   $\delta := \beta;$ 
2  while  $\delta \neq 0$  do
3  begin
4       $x_{m(\delta)} := x_{m(\delta)} + 1;$ 
5       $\delta := \delta \ominus \alpha_{m(\delta)};$ 
6  end;

```

## 2.2. A csoportfeladat optimális megoldásának néhány tulajdonsága

Tekintsük újra a csoportfeladatot.

$$\begin{aligned}
 \min \quad & \sum_{j=1}^t \gamma_j x_j \\
 \sum_{j=1}^t \alpha_j x_j &= \beta \\
 x &\in \mathbb{Z}_+^t
 \end{aligned}$$

Jelöljük  $v_1, v_2, \dots, v_t$ -vel az  $\alpha_1, \alpha_2, \dots, \alpha_t$  elemek rendjét a csoportban.

**2.2.1. Tétel.** *A csoportfeladatnak van olyan  $x^*$  optimális megoldása, amelyre*

$$x_j^* \leq v_j - 1, \quad j = 1, 2, \dots, t.$$

**Bizonyítás:** Jelölje  $z^*$  az  $x^*$  értékhez tartozó optimális célfüggvényértéket. Indirekt tegyük fel, hogy  $x^*$  valamely  $k$  komponensére:  $x_k^* > v_k - 1$ . Legyen  $\bar{x}$  az a vektor, melynek elemei:

$$\bar{x}_j = \begin{cases} x_j^* & j \neq k \\ x_k^* - v_k & j = k \end{cases}$$

Legyen  $\bar{z}$  az ehhez az elemhez tartozó célfüggvényérték. Ekkor  $\bar{x} \geq 0$ , és

$$\sum_{j=1}^t \alpha_j \bar{x}_j = \sum_{j=1}^t \alpha_j x_j^* \ominus \alpha_k v_k = \sum_{j=1}^t \alpha_j x_j^* = \beta,$$

azaz  $\bar{x}$  kielégíti a csoportfeladat feltételeit, és

$$z^* = \bar{z} + v_k \gamma_k \geq \bar{z}.$$

Mivel  $x^*$  optimális, ezért itt csak egyenlőség lehet, azaz  $\bar{x}$  is optimális megoldása a csoportfeladatnak.  $\square$

**2.2.2. Tétel.** *A csoportfeladatnak van olyan  $x^*$  optimális megoldása, amelyre*

$$\sum_{j=1}^t x_j^* \leq d - 1.$$

**Bizonyítás:** Tegyük fel indirekt, hogy a csoportfeladat egy tetszőleges megoldására:

$$\sum_{j=1}^t x_j \geq d.$$

Ekkor a gráfban  $x$ -nek egy legalább  $d+1$  csúcst érintő út felel meg, ami azt jelenti, hogy tartalmaz kört. Ha kört elhagyjuk, akkor csökkenhet az út hossza, és így a célfüggvényérték.  $\square$

A csoportfeladat az egészértékű programozási feladatnak egy relaxációja, így az optimális megoldás nem biztos, hogy az egészértékű programozási feladatnak is optimális megoldása lesz. Ahhoz, hogy optimális megoldást kapjunk az egészértékű programozási feladatra is, teljesülnie kell a következőknek:

$$x_B = B^{-1}b - B^{-1}Nx_N \geq 0.$$

**2.2.3. Tétel.** *Jelölje  $v = (v_1, v_2, \dots, v_t)$ ,  $\bar{N}$  azt a mátrixot, amelyre  $\bar{N}_{ij} = |(B^{-1}N)_{ij}|$ . Legyen  $x^*$  a csoportfeladat optimális megoldása, amelyre  $x_j^* \leq v_j - 1$ , ( $j = 1, 2, \dots, t$ ) teljesül. Jelölje  $e$  a csupa 1-esből álló  $t$ -dimenziós vektort. Ha*

$$B^{-1}b \geq \bar{N}(v - e),$$

akkor

$$B^{-1}b - B^{-1}Nx^* \geq 0.$$

**Bizonyítás:** Legyen  $y_i = |(B^{-1}Nx^*)_{ij}|$ , ( $i = 1, 2, \dots, m$ ), ekkor

$$B^{-1}Nx^* \leq y \leq \bar{N}x^* \leq \bar{N}(v - e) \leq B^{-1}b. \quad \square$$

**2.2.4. Következmény.** *Az előző tétel jelölései mellett, ha*

$$B^{-1}b \geq \bar{N}(d - 1)e,$$

akkor

$$B^{-1}b - B^{-1}Nx^* \geq 0.$$

**Bizonyítás:** Az előző tételből adódik  $v - e \leq (d - 1)e$  miatt.  $\square$

## 2.3. A csoportelméleti módszer felhasználása a korlátozás és szétválasztás algoritmusában

A korlátozás és szétválasztás algoritmusok azon alapulnak, hogy adva van egy relaxáció, amelyet könnyen, gyorsan meg tudunk oldani, és ennek segítségével szűkítjük a megengedett megoldások

halmazát. Pontosabban felbontjuk a megengedett halmazt diszjunkt részekre és a relaxáció segítségével alsó korlátot adunk a célfüggvény értékére az egyes halmazokon. Ennek segítségével lépésről lépésre kizárjuk a megengedett halmaz egy-egy részét és eljutunk az optimális megoldásig.

Jelölje  $v$ , mint eddig, a csoportfeladatot meghatározó elemek rendjét.

**2.3.1. Definíció.** Legyen  $y \in \mathbb{Z}_+^t$ ,  $y < v$  egy tetszőleges vektor. Ekkor  $z \in \mathbb{Z}_+^t$  az  $y$  vektor folytatása, ha

$$\begin{aligned} y &\not\leq z, \\ z &< v. \end{aligned}$$

Tehát  $y$  folytatásai nem egyenlők vele.

Legyen  $0 \leq y < v$  egy tetszőleges egész vektor, és tekintsük a (2.6)-os csoportfeladatot az  $x \geq y$  plusz feltétel mellett. Legyen ennek optimális megoldása  $\bar{x}$ . Ekkor a következő feladatot kell megoldanunk:

$$\begin{aligned} \min \sum_{j=1}^t \gamma_j u_j \\ \sum_{j=1}^t \alpha_j u_j = \beta \ominus \sum_{j=1}^t \alpha_j y_j \\ u \in \mathbb{Z}_+^t \end{aligned} \tag{2.8}$$

Ennek  $u^*$  optimális megoldása megadja az  $\bar{x}$  vektort is,  $\bar{x} = u^* + y$ . Ezután legyen  $y \leq w \leq \bar{x}$  egy tetszőleges egész vektor, ekkor ha megoldjuk a csoportfeladatot ((2.6)-ot) az  $x \geq w$  feltétel mellett, akkor az optimális megoldás megint  $\bar{x}$  lesz. Ezért vezessük be a következőt:

**2.3.2. Definíció.** Legyen egy rögzített  $y$  vektor mellett a (2.8)-as feladat optimális megoldása  $u^*$ . A  $0 \not\leq z < v$ ,  $z \in \mathbb{Z}_+^t$  az  $y$  vektor közvetlen folytatása, ha létezik olyan  $1 \leq k \leq t$  index, hogy

$$z_j = \begin{cases} y_k + u_k^* + 1 & \text{ha } j = k \\ y_j & \text{ha } j \neq k. \end{cases}$$

Tehát az  $y$  vektor közvetlen folytatásai, azok a parciális rendezésben minimális folytatásai, amelyek a csoportfeladat garantáltan más megoldását adják meg, mint  $y$ . A definícióban megadott  $z$  vektort  $y$   $k$ -adik közvetlen folytatásának fogjuk nevezni. Ha

$$y_k + u_k^* \geq v_k - 1,$$

akkor a  $k$ -adik közvetlen folytatás nem létezik.

A közvetlen folytatások meghatározásához a (2.8)-as feladatot kell megoldani. Ehhez nincs szükség a csoportfeladatot megoldó algoritmust újra és újra lefuttatni, mert az minden jobboldalra meghatározza a megoldást. Ez még inkább felgyorsítja a megoldást.

Az algoritmusban a megoldások szerinti részhalmazokat az  $y$  folytatásai adják. Tehát a halmazt legfeljebb  $t$  részre bontjuk fel, az egyes részekben  $y$   $k$ -adik közvetlen folytatása, (ha létezik) és annak folytatásai vannak.

Legyen  $S$  az a halmaz, amelybe  $y$  és folytatásai tartoznak.  $y$ -t az  $S$  halmaz gyökérpontjának fogjuk nevezni.

Az algoritmus jelölései:

$L$ : a felbontandó halmazok listája, amelyeket a gyökérpontjukkal adunk meg,

$h(y)$ : az  $y$  gyökérpontú halmazhoz tartozó alsó becslés, azaz a (2.8)-as feladat optimális célfüggvényértéke,

$Opt(y)$ : az  $y$  vektorral definiált (2.8)-as feladat optimális megoldásainak halmaza,

$f$ : az eddig megtalált legjobb megengedett megoldás célfüggvényértéke,

$g$ : segédváltozó.

Az eljárásban feltételezzük, hogy a csoportfeladatot már megoldottuk, és volt optimális megoldása, de az nem volt megoldása az egészértékű programozási feladatnak. Ezért a  $h(y)$  értékeket és a hozzájuk tartozó optimális megoldásokat ki tudjuk olvasni a dinamikus programozás által készített táblázatból.

Az algoritmus:

```

1  Begin
2       $L := \{0\};$ 
3       $f := \infty;$ 
4      while  $L \neq \emptyset$  do
5          begin
6              kiválasztunk egy  $y$  elemet  $L$ -ből;
7               $L := L - y;$ 
8              megoldjuk a (2.8)-as feladatot  $y$ -nal;
9              while  $Opt(y) \neq \emptyset$  do
10                 begin
11                     kiválasztunk egy  $u^*$  elemet  $Opt(y)$ -ből;
12                      $Opt(y) := Opt(y) - u^*;$ 
13                     if  $B^{-1}b \geq B^{-1}N(y + u^*)$  then

```

```

14         begin
15              $g := c_B^\top B^{-1}b + (c_N^\top - c_B^\top B^{-1}N)(y + u^*);$ 
16             if  $g < f$  then  $f := g;$ 
17         end
18         else
19         begin
20             for  $j := 1$  to  $t$  do
21                 begin
22                     if  $(\exists y$  k-dik közvetlen folytatása) then  $z := y$  k-dik közvetlen folytatása;
23                     if  $h(z) < f$  then  $L := L + z;$ 
24                 end;
25             end;
26         end;
27     end;
28 end;

```

**2.3.3. Tétel.** *Ez az algoritmus véges.*

**Bizonyítás:** Tegyük fel indirekt, hogy az algoritmus nem véges. Ez csak úgy fordulhat elő, ha az  $L$  lista nem ürül ki. Minden  $L$ -beli elem egy előzőleg már  $L$ -be kerülő elem folytatása, és kezdetben csak a 0 vektor van  $L$ -ben. Ezért létezik egy egész vektorokból álló  $\{y_i\}$  végtelen sorozat úgy, hogy mind a 0 folytatása, és kisebb, mint a  $v$  vektor, azaz

$$0 = y_1 \leq y_2 \leq \dots$$

$$y_i < v \quad i = 1, 2, \dots$$

Ez viszont ellentmondás, mert csak véges sok olyan  $\mathbb{Z}_+^t$ -beli vektor van, ami a  $v$  vektornál kisebb.  $\square$

A korlátozás és szétválasztás módszer valamilyen módon végignézi az összes megoldást, ezért be kell látnunk, hogy az algoritmus nem hagy ki egy szóba jöhető vektort sem. Azt kell bizonyítanunk, hogy egy tetszőleges  $w \in \mathbb{Z}_+^t, w < v$  vektor minden (2.8)-as feladat által adott megoldását megvizsgálja.

**2.3.4. Tétel.** *Az algoritmus minden  $w \in \mathbb{Z}_+^t, w < v$  vektorhoz tartozó (2.8)-as feladat egy optimális megoldásából származó  $w + u^*$  vektort megvizsgál, feltéve, hogy a csoportfeladatnak van optimális megoldása.*

**Bizonyítás:** Legyen a csoportfeladat egy optimális megoldása  $x^*$ . Ekkor az

$$u_j = \begin{cases} x_j^* - w_j + v_j & \text{ha } x_j^* < w_j \\ x_j^* - w_j & \text{ha } x_j^* \geq w_j \end{cases}$$

választással a (2.8)-as feladatnak egy megengedett megoldását kapjuk, így annak létezik optimális megoldása is. Indirekt tegyük fel, hogy egy,  $w$  vektorhoz tartozó  $w + u^*$  vektort nem vizsgált meg az algoritmus. Ez csak úgy lehet, hogy  $w$  nem került be az  $L$  listába. Kezdetben csak a  $0$  vektor van a listában, ennek  $w$  folytatása, a választása miatt. Legyen  $\bar{y}$  az  $L$  listából utoljára kikerülő vektort, amelynek  $w$  a folytatása. Ha  $w$  nem közvetlen folytatása az  $\bar{y}$  vektornak, és nem folytatása  $\bar{y}$  valamely közvetlen folytatásának, akkor  $w$  vektort az algoritmus 13-dik sorában vizsgálta, amikor  $y$  az  $\bar{y}$  értéket vette fel. Ezek után tegyük fel, hogy  $\bar{y}$ -nak van olyan  $\bar{z}$  közvetlen folytatása, amelyre  $\bar{z} \leq w$ . Mivel  $\bar{z}$  nem került be az  $L$  listába, ezért az algoritmus 23-dik sorában a feltétel nem teljesült, azaz az  $\bar{y}$ -nal felírt (2.8)-as feladat megoldása nem adott jobb megoldását az egészértékű programozási feladatnak, mint az algoritmus által előbb megvizsgált vektorok.  $\square$

Az egészértékű programozási feladat, így a következőképpen oldható meg. Először megoldjuk a feladat lineáris programozási relaxáltját, ha ekkor egész megoldást kapunk, akkor kész. Ha nem, akkor ez ad nekünk egy optimális bázist, amelynek segítségével fel tudjuk írni a csoportfeladatot. Ha a csoportfeladatnak létezik megoldása, és az optimális, akkor megint kész vagyunk, ha nem akkor az előző algoritmust lefuttatva kapunk megoldást. Azaz az egészértékű programozási feladat megoldásának menete a következő:

- 1 Begin
- 2     Az egészértékű programozási feladat LP-relaxációjának megoldása;
- 3     a csoportfeladat felállítása;
- 4     if ( $\exists$  a csoportfeladatnak optimális megoldása) then  $x :=$  egy optimális megoldás;
- 5     if  $B^{-1}b \not\leq B^{-1}Nx$  then az előző algoritmus lefuttatása;
- 6 end;

## 2.4. A csoportelméleti feladat alkalmazása az (1.2)-ben adott feladatra

A feladat a következő:

$$Ax = ab : \quad x \in \mathbb{Z}_+^{10}, a \in \mathbb{Z}_+, a \neq 0$$

ahol

$$A = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ -1 & 1 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 2 & -2 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{pmatrix}, b = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

és a célfüggvény:

$$c = \left( 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \right)^\top.$$

Ez azonban nagyon sok feladat, így nehéz lenne megoldani, de átírható egy ekvivalens alakba, ahol csak egy egyenlőségrendszerünk lesz. Vigyük át a  $b$  jobboldali vektorunkat a bal oldalra. Így az  $a$  tekinthető mint a  $-b$ -hez tartozó változó. Ekkor viszont fel kell még tennünk, hogy ez a változó pozitív, ami az egészértékűség miatt ekvivalens azzal, hogy  $a \geq 1$ .

Így a következő feladathoz jutunk:

$$\min (c, 1)^\top x$$

$$\begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 \\ 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 2 & -2 & 0 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{pmatrix} x = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$x_{11} \geq 1$$

$$x \in \mathbb{Z}_+^{11}$$

Itt  $(c, 1)$  a csupa egyesből álló 11 dimenziós vektor.

A csoportelméleti módszert úgy építettük fel, hogy a korlátozások egyenlőségek, ezért vezessünk be egy kiegészítő változót, legyen ez  $x_{12}$ , olyan, hogy

$$x_{11} \geq 1 \Leftrightarrow -x_{11} + x_{12} = -1, \quad x_{12} \geq 0$$

Ekkor a feladatunk a következő alakba írható:

$$\min \hat{c}^\top x$$

$$\hat{A} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 2 & -2 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} x = \hat{b} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}$$

$$x \in \mathbb{Z}_+^{12}$$

Itt  $\hat{c}$  a csupa egyesből álló 12 dimenziós vektor.

A csoportelméleti módszer alkalmazásához kell még, hogy a mátrixunk teljes sorrangú. Az  $\hat{A}$  mátrix rangja 5, azaz el kell hagynunk 3 sorát, amely lineárisan függ a megmaradó 5-től. Jelöljük az  $\hat{A}$  mátrix sorait  $a^i$ -vel  $i = 1, \dots, 8$ . Ekkor

$$\begin{aligned} a^1 &= -a^4 + a^6 + a^7 \\ a^2 &= -a^4 + a^5 + a^6 + 2a^7 \\ a^3 &= a^4 - a^5 - 2a^6 - 2a^7. \end{aligned}$$

Így a következő mátrixhoz és hozzá tartozó jobboldali vektorhoz jutunk:

$$\bar{A} := \begin{pmatrix} 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}, \bar{b} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \end{pmatrix}.$$

Ekkor a feladatunk:

$$\begin{aligned} \min \hat{c}^\top x \\ \bar{A}x &= \bar{b} \\ x &\in \mathbb{Z}_+^{12} \end{aligned}$$

A csoportfeladat felállításához szükségünk lesz egy optimális bázisra, az első, a harmadik, a hetedik, a kilencedik, és a tizenegyedik, (azaz a  $b$ -nek megfelelő,) oszlopokból álló vektorok megfelelnek, azaz

$$B := \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix},$$



ekkor

$$B^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}, N = \begin{pmatrix} -1 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

A bázis optimális, mert

$$c_N^\top - c_B^\top B^{-1}N = \begin{pmatrix} 2 & 2 & 1 & 1 & 2 & 2 & 4 \end{pmatrix} \geq \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Meg kell határozni a B bázis Smith-féle normálalakját is.

$$P = B^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

Ez megfelel, mert  $P$ -vel balról megszorozva egységmátrixot, azaz diagonális mátrixot kapunk, és  $P$  determinánsa  $-1$ . Ezért a  $\xi = \xi^{-1} = E$ , ahol  $E$  a  $(5 \times 5)$ -ös egységmátrix. Így a csoportfeladat:

$$\min \begin{pmatrix} 2 & 2 & 1 & 1 & 2 & 2 & 4 \end{pmatrix} x + 3$$

$$\xi^{-1}PNx = \begin{pmatrix} -1 & 0 & 1 & -1 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} x \equiv \xi^{-1}P\bar{b} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \pmod{1}$$

$$x \in \mathbb{Z}_+^7$$

Mivel a  $B$  mátrix determinánsa  $-1$  volt, ezért a kongruencia  $\pmod{1}$ , és így minden egész  $x$  vektorra teljesülni fog. Ezek közül a minimális a

$$x_N = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^\top.$$

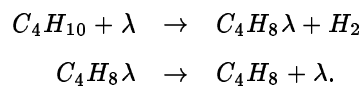
Ekkor

$$x_B = B^{-1}b - B^{-1}Nx_N = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \geq 0$$

és így az eredeti feladat optimális megoldása

$$x = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Ez azt jelenti, hogy a következő két elemi reakcióból áll elő a kívánt reakció optimálisan:



(A megoldásnál a számolásokat MATLAB-bal végeztem.) A feladat lineáris programozási relaxáltját a MAPLEV simplex csomagjának minimum programjával megoldva is ugyanezt az egész megoldást kapjuk.

## 3. fejezet

# A Hilbert bázis meghatározása

Ebben a fejezetben a következő alakú egészértékű feladattal fogunk foglalkozni:

$$\left. \begin{array}{l} Ax = 0 \\ x \geq 0 \\ x \in \mathbb{Z}^n \end{array} \right\} M$$

ahol  $A \in \mathbb{Z}^{m \times n}$ ,  $b \in \mathbb{Z}^m$ . Ennek a rendszernek a megoldásai egy kúpot alkotnak, az egész megoldásait pedig generálják a nem nulla minimális elemek.

**3.0.1. Definíció.** Az  $M$  rendszer egész megoldásainak a következő rendezés szerinti minimális elemeinek véges halmazát,  $\mathcal{H}(M)$ -t Hilbert bázisnak nevezzük. A rendezés:

$$(x_1, \dots, x_n) \leq (y_1, \dots, y_n) \Leftrightarrow \forall i \quad 1 \leq i \leq n, \quad x_i \leq y_i.$$

Ekkor az  $M$  rendszer minden megoldása előáll a  $\mathcal{H}(M)$  Hilbert bázis elemeinek valamilyen pozitív, egész együtthatós lineáris kombinációjaként.

Vezessük be a következő jelöléseket:

$$\|M\|_1 = \sup_{x \in \mathcal{H}(M)} \|x\|_1, \quad \|A\|_{1,\infty} = \sup_{i=1,\dots,m} \left( \sum_{j=1}^n |a_{ij}| \right),$$

ahol  $\|x\|_1 = \sum_{i=1}^n |x_i|$ , továbbá legyen  $r := \text{rang}(A)$ .

**3.0.2. Tétel.**

$$\|M\|_1 \leq \left( 1 + \|A\|_{1,\infty} \right)^r$$

**Bizonyítás:** Az  $Ax = b$  egyenletrendszerből kiválaszthatunk  $r$  lineárisan független sort, ezért feltehető, hogy az  $A$  mátrixunk  $r$  sorból áll. Legyen  $x = (x_1, \dots, x_n) \in M$ , az  $M$  egy nem nulla eleme,  $p := \|x\|_1$ , és  $e_i, (i = 1, \dots, n)$  az a vektor, amelynek az  $i$ -edik koordinátája 1, a többi 0. Minden  $y \in \mathbb{R}^n$  vektorra, jelölje  $C_y$  azt az egységkockát, amelyre

$$z \in C_y \Leftrightarrow z = y + \sum_{i=1}^n \lambda_i e_i, \quad \lambda_i \in [0, 1] \quad \forall i = 1, \dots, n.$$

Definiálunk  $y^0, \dots, y^p \in \mathbb{Z}_+^n$  és  $z^0, \dots, z^p \in \mathbb{R}^n$  vektorokat úgy, hogy

$$y^0 = 0 < y^1 < \dots < y^p = x$$

$$\forall k \in [0, p-1], \quad \exists j \quad y^{k+1} = y^k + e_j$$

$$\forall k \in [0, p] \quad \exists z^k \in C_{y^k} \cap [0, x]$$

Azaz lépésenként valamelyik koordinátát eggyel növelve haladunk 0-ból  $x$ -be úgy, hogy közben a megfelelő egységkocka messe a  $[0, x]$  szakaszt. (Ezekből a metszetekből választjuk a  $z^k$ -kat) Tudunk ilyen vektorokat választani, mert tegyük fel, hogy  $y^k$ -kat és  $z^k$ -kat ismerjük valamilyen  $k$ -ig ( $0 \leq k \leq p-1$ ). ( $x^0 = z^0 = 0$ ) Ekkor, mivel  $z^k \in [0, x] \cap C_{y^k}$ , ezért  $\exists \lambda : 0 \leq \lambda \leq 1$ , hogy  $z^k$  felírható

$$z^k = \lambda \sum_{i=1}^n x_i e_i$$

alakban, úgy

$$\forall i = 1, \dots, n \quad y_i^k \leq \lambda x_i \leq y_i^k + 1.$$

Legyen

$$\lambda_k = \min_{i: x_i \neq 0} \frac{y_i^k + 1}{x_i}$$

Legyen  $j$  az az index, amelyre  $\lambda_k = \frac{y_j^k + 1}{x_j}$ . Ekkor legyen

$$z^{k+1} = \lambda_k \sum_{i=1}^n x_i e_i, \quad y^{k+1} = y^k + e_j.$$

Már csak azt kell belátnunk, hogy  $z^{k+1} \in C_{y^{k+1}}$ .  $\lambda_k$  választása szerint

$$\forall i = 1, \dots, n : \quad y_i^{k+1} \leq \lambda_k x_i \leq y_i^{k+1} + 1,$$

azaz  $z^{k+1} \in C_{y^{k+1}}$  teljesül. Ha  $k = p-1$  akkor  $y^p = x$ , mert  $y^p \leq x$  és  $\|y^p\|_1 = p = \|x\|_1$ . Ebből az is következik, hogy  $z^p = x$ .

Legyen most  $\bar{y}^k = z^k - y^k$ . Ekkor

$$\forall i = 1, \dots, n \quad 0 \leq \bar{y}_i^k \leq 1.$$

Jelölje  $(Ay^k)_i$  az  $Ay^k$   $i$ -edik koordinátáját. Most  $Az^k = 0$ , mert  $z^k$  a  $[0, x]$  szakaszon van, ezért

$$|(Ay^k)_i| = |(Az^k)_i - (A\bar{y}^k)_i| = |(A\bar{y}^k)_i| = \left| \sum_{j=1}^n a_{ij} \bar{y}_j^k \right| \leq \sum_{j=1}^n |a_{ij}|.$$

Mivel  $|(Ay^k)_i|$  nemnegatív egész, ezért összesen  $\sum_j |a_{ij}| + 1$  különböző értéket vehet fel. ( $0, 1, \dots, \sum_j |a_{ij}|$  értékeket veheti fel.)  $Ay^k$  egy  $r$ -dimenziós vektor, így legfeljebb

$$\left( 1 + \sup_i \sum_j |a_{ij}| \right)^r$$

különböző értéket vehet fel.

Tegyük fel, hogy  $p > \left(1 + \sup_i \sum_j |a_{ij}|\right)^r$ . Ekkor léteznek  $i, j : i > j > 0$  indexek, hogy  $Ay^i = Ay^j$ .  
Legyen

$$z = y^i - y^j$$

Ekkor  $Az = 0$ , és  $0 < z < x$  miatt  $z \in M$ , amiből következik, az ellentmondás, hogy  $x \notin \mathcal{H}(M)$ .  $\square$

### 3.1. Contejean-Devie algoritmus

E.Contejean és H.Devie algoritmusához nincs szükség  $\mathcal{H}(M)$  korlátjára. Az algoritmus elve a következő: rendezzük  $\mathbb{Z}_+^n$ -t a már definiált rendezés szerint, így kapunk egy körmentes, irányított gráfot, amelynek a gyökere 0. Nevezzük el a gráfot DAG-nak (directed acyclic graph). Az algoritmus ebben a gráfban fog keresni, mégpedig a következők szerint: Induljunk ki a nullából.

Csak olyan  $x$  csúcsra lépünk tovább, amely teljesíti, hogy  $x$  minden  $y$  szülőjére  $A(x - y) \neq 0$ , és  $x$  csak olyan  $x + e_j$  gyerekeire lépünk tovább, amelyekre teljesül, hogy  $\langle Ax, Ae_j \rangle \leq 0$ .

Vizsgáljuk meg, miért így választjuk a csúcokat.

**3.1.1. Lemma.** *Elég azokat az  $x \geq 0$  csúcokat vizsgálni, amelyekre:*

$$A(x - y) \neq 0$$

az  $x$  minden  $y$  őse.

**Bizonyítás:** Tegyük fel indirekt, hogy létezik  $x$ -nek  $y$  őse, hogy  $A(x - y) = 0$ . Azaz

$$\exists y : y \leq x, \quad Ax = Ay$$

Ha  $x$  megoldás, akkor  $y$  is az és  $y \leq x$ , ami ellentmond annak, hogy  $x$  minimális megoldás. Így  $x$  nem eleme a Hilbert bázisnak.

Ha  $x$  nem megoldás, de létezik  $z \geq 0, z \neq 0$  vektor, hogy  $x + z$  megoldás, akkor  $A(x + z) = A(y + z)$  miatt  $y + z$  is megoldás.  $y + z$  őse  $x + z$ -nek, ezért csak  $y + z$  lehet a Hilbert bázis eleme. Azaz  $x$  egyik leszármazottja sem lehet tagja a Hilbert bázisnak.  $\square$

**3.1.2. Lemma.** *Legyen  $x \geq 0, x \neq 0$ . Elég  $x$  azon  $x + e_j$  ( $j \in 1, 2, \dots, n$ ) gyermekeit vizsgálni, amelyekre*

$$\langle Ax, Ae_j \rangle < 0.$$

**Bizonyítás:** Azt kell belátnunk, hogy minden, a rendezés szerint, minimális megoldásba el lehet jutni ilyen úton.

Tegyük fel, hogy  $y$  minimális megoldás, azaz a Hilbert-bázis eleme. Ekkor  $\forall 0 \leq x \leq y \quad \exists z \geq 0 : x + z = y$ . Mivel  $Ay = 0$ , ezért

$$\begin{aligned} 0 &= \langle Ay, Ax \rangle = \langle A(x + z), Ax \rangle = \\ &= \|Ax\|^2 + \langle Az, Ax \rangle = \\ &= \|Ax\|^2 + \sum_{j=0}^n z_j \langle Ae_j, Ax \rangle \end{aligned}$$

Mivel  $y$  minimális megoldás, ezért  $Ax \neq 0$ , azaz

$$\|Ax\| > 0.$$

A  $z$ -ről feltettük, hogy minden koordinátája nem negatív,  $\forall z_j \geq 0$ , ezért

$$\sum_{j=1}^n z_j \langle Ae_j, Ax \rangle < 0.$$

Ebből következik, hogy létezik egy  $j \in \{1, \dots, n\}$  index, hogy

$$\langle Ae_j, Ax \rangle < 0.$$

Így az  $e_i$  ( $i = 1, \dots, n$ ) vektorokból kiindulva, minden megoldáshoz meg tudunk konstruálni egy ilyen utat. Ezt az utat az algoritmus vizsgálni fogja, és így eljut a megoldáshoz.  $\square$

Ha nem vizsgálunk többször egy csúcsot a DAG-ban, és csak a rendezés szerinti legkisebb megoldásokat keressük, akkor megkapjuk  $\mathcal{H}(M)$ -et. Ez az algoritmus jól működik a gyakorlatban, de időigényes, ha  $\mathcal{H}(M)$  elemeinek nagy a normájuk.

## 3.2. A Contejean-Devie algoritmus módosítása a (3.0.2)-es tétel alapján

Az előbbi algoritmus futását lerövidíthetjük, ha figyeljük azt is, mekkora a választandó csúcs normája. A  $\mathcal{H}(M)$  normájára adott egy felső becslést a (3.0.2)-es tétel. Ez alapján adódik a következő tétel:

**3.2.1. Tétel.** *Elég  $x$  azon  $x + e_l$  gyermekeit vizsgálnunk, amelyre az  $A(x + e_l)$   $i$ -edik koordinátája  $-\sum_{j=1}^n a_{ij}^-$  és  $\sum_{j=1}^n a_{ij}^+$  közé esik.*

**Bizonyítás:** Legyen egy  $a \in \mathbb{R}$  szám pozitív része:  $a^+ := \max\{0, a\}$ , illetve negatív része:  $a^- := \max\{0, -a\}$ . Azaz mind a pozitív, mind a negatív rész pozitív szám, emellett igaz a következő:

$$a = a^+ - a^-,$$

$$|a| = a^+ + a^-.$$

A (3.0.2)-es tétel bizonyításának alapján megbecsülhetjük  $(Ay^k)_i$ -t:

$$(Ay^k)_i = (Az^k)_i - (A\bar{y}^k)_i = (A\bar{y}^k)_i = \sum_{j=1}^n a_{ij} \bar{y}_j^k = \sum_{j=1}^n a_{ij}^+ \bar{y}_j^k - \sum_{j=1}^n a_{ij}^- \bar{y}_j^k.$$

Mivel  $0 \leq \bar{y}_j^k \leq 1$ , és  $a_{ij}^-, a_{ij}^+ \geq 0$  ( $\forall i, j$ -re), ezért:

$$\sum_{j=1}^n a_{ij}^+ \bar{y}_j^k - \sum_{j=1}^n a_{ij}^- \bar{y}_j^k \leq \sum_{j=1}^n a_{ij}^+ \bar{y}_j^k \leq \sum_{j=1}^n a_{ij}^+$$

és

$$\sum_{j=1}^n a_{ij}^+ \bar{y}_j^k - \sum_{j=1}^n a_{ij}^- \bar{y}_j^k \geq - \sum_{j=1}^n a_{ij}^- \bar{y}_j^k \geq - \sum_{j=1}^n a_{ij}^-,$$

így a következő becslésből következik a tétel:

$$- \sum_{j=1}^n a_{ij}^- \leq (Ay^k)_i \leq \sum_{j=1}^n a_{ij}^+.$$

□

Ezek alapján felírhatunk egy elméleti, szélességi keresésen alapuló algoritmust. Legyen az  $\mathcal{X}$  a vizsgálandó csúcsok halmaza, a DAG valamelyik szintjén.

Az algoritmus:

```
1 Begin
2    $\mathcal{X} := \{e_k : k = 1, \dots, n\};$ 
3    $\mathcal{H}(M) := \{e_k : k = 1, \dots, n; Ae_k = 0\};$ 
4    $\mathcal{X} := \mathcal{X} \setminus \mathcal{H}(M);$ 
5   while  $\mathcal{X} \neq \emptyset$  do
6     begin
7        $\mathcal{X} := \{x + e_k : x \in \mathcal{X}, k = 1, \dots, n$ 
8          $\langle Ax, Ae_k \rangle < 0$ 
9          $Ax \neq Ay : 0 \neq y \leq x$ 
10         $-\sum_{j=1}^n a_{ij}^- \leq A(x + e_k) \leq \sum_{j=1}^n a_{ij}^+\};$ 
11       $\mathcal{H}(M) := \mathcal{H}(M) \cup \{x \in \mathcal{X} : Ax = 0\};$ 
12       $\mathcal{X} := \mathcal{X} \setminus \{x \in \mathcal{X} : Ax = 0\};$ 
13    end;
14  end;
```

Az algoritmus minden lépésében egy szintjét vizsgáljuk a DAG-nak, azokat a vektorokat, amelyek 1-es normája  $k$ . (ha a  $k$ -adik szinten vagyunk.) Az adott szinten viszont, csak azokat a vektorokat vizsgálja, amelyek megfelelnek a választási szabályoknak, és így sokat kizár, ami nem lehet tagja a Hilbert bázisnak.

### 3.3. A Hilbert bázis meghatározása az (1.2)-ben adott feladatra

A feladatunk a következő:

$$\begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 0 & 0 & 1 & -1 & -1 & 1 & 0 \\ 1 & -1 & -1 & 1 & -1 & 1 & 0 & 0 & 2 & -2 & 0 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 & 1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{pmatrix} x = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$x_{11} \geq 1$$

$$x \in \mathbb{Z}_+^{11}$$

Tekintsünk el egy kicsit az  $x_{11} \geq 1$  feltételtől. Ekkor  $x_{11} = 0$ , ami azt mutatja, hogy az adott reakciók vektoraiból a nullát állítjuk elő, azaz az így kapott megoldások okoznak köröket a reakció útvonalban. Így elég az algoritmussal azokat a vektorokat vizsgálni, ahol  $x_{11} \geq 1$ .

Az 1.fejezetben megadott megoldásból a feladat általános megoldása a következő:

$$(|a - c|_+ + b_1, |a - c|_- + b_1, a + b_2, b_2, |c|_+ + b_3, |c|_- + b_3, b_4, b_4, |c|_+ + b_5, |c|_- + b_5, a)^T.$$

( $a$  nemnegatív egész,  $c$  egész.) A  $b_i$ , ( $i = 1, \dots, 5$ ) azoknak a triviális köröknek felelnek meg, amikor az egyes reakciók oda, és vissza irányban is végbemennek. Ezért ezeket el is hagyhatjuk, hiszen a parciális rendezés szerinti legkisebb megoldásokat keressük. Ha  $a = 0$ , akkor megkapjuk a nem triviális köröket:

	$  -c _+$	$  -c _-$	0	0	$ c _+$	$ c _-$	0	0	$ c _+$	$ c _-$	0
$c > 0$	0	1	0	0	1	0	0	0	1	0	0
$c < 0$	1	0	0	0	0	1	0	0	0	1	0

azaz az  $(1 \leftarrow, 3 \rightarrow, 5 \rightarrow)$  és az  $(1 \rightarrow, 3 \leftarrow, 5 \leftarrow)$  reakciók köröket alkotnak.

Ezek után tegyük fel, hogy  $a \geq 1$ . Ekkor két eset lehet:  $a - c > 0$  és  $0 < a \leq c$ , a parciális rendezés szerinti legkisebb megoldások ekkor:

	$ a - c _+$	$ a - c _-$	$a$	0	$ c _+$	$ c _-$	0	0	$ c _+$	$ c _-$	$a$
$a - c > 0$	1	0	1	0	0	0	0	0	0	0	1
$0 < a \leq c$	0	0	1	0	1	0	0	0	1	0	1

Minden más megoldás előáll ezeknek a vektoroknak nemnegatív, egész lineáris kombinációjaként. Nézzük meg az 1.fejezetben kapott megoldásokat. Ezekből kettőt megkaptunk most is, így ezeknél triviális.



Az  $(1 \rightarrow, 2 \rightarrow, 3 \rightarrow; 5 \rightarrow)$  a kapott két megoldás összege, azaz:

$$\begin{array}{r|cccccccccc}
 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 + & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
 \hline
 & 1 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 2
 \end{array}$$

Ebből a felírásból kiderül viszont, hogy a másik két megoldás tartalmaz kört, hiszen egyik sem állítható elő az előzőleg megkapott megoldások lineáris kombinációjaként, csak ha olyat veszünk hozzá, ami kört okoz.

$$\begin{array}{r|cccccccccc}
 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 + & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
 \hline
 & 2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1
 \end{array}$$

$$\begin{array}{r|cccccccccc}
 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
 + & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
 \hline
 & 0 & 1 & 1 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 1
 \end{array}$$

Ez azt jelenti, hogy összesen 3 reakció útvonal van,  $(1 \rightarrow, 2 \rightarrow)$ ,  $(2 \rightarrow, 3 \rightarrow, 5 \rightarrow)$ , és ennek a kettőnek az összege.

Futtassuk le az algoritmust úgy a feladatra, hogy elhagyjuk  $x_{11} \geq 1$  feltételt, és megengedjük, hogy nulla legyen. Ekkor megkapjuk azokat a vektorokat amikhez az előbb eljutottunk, és azokhoz amik kört okoznak. Azaz a Hilbert bázis:

$$\begin{aligned}
 & \left( 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \right)^T \\
 & \left( 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \right)^T \\
 & \left( 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \right)^T \\
 & \left( 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \right)^T \\
 & \left( 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \right)^T \\
 & \left( 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \right)^T \\
 & \left( 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \right)^T \\
 & \left( 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \right)^T
 \end{aligned}$$

$$\left( 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \right)^\top$$

Összehasonlításképpen az algoritmus a második szinten  $11^2$  vektor helyett, csak 26-ot vizsgál meg, a harmadik szinten  $11^3$  vektor helyett, csak 34-et, a negyedik szinten  $11^4$  vektor helyett 36-ot.

(Ha csak  $e_{11}$ -ből indítjuk, akkor is eljut az algoritmus a Hilbert bázis 2 eleméhez, és így még kevesebb vektort vizsgál szintenként, például: a harmadik szinten 9-et, a negyediken 13-at, az ötödiken 9-et.)

(Az elméleti algoritmus lépéseinél a számolásokat MATLAB-bal végeztem.)

# Irodalomjegyzék

- [1] Vizvári Béla: *Egészértékű programozás*, Tankönyvkiadó, Budapest, 1989
- [2] A.Schrijver: *Theory of Linear and Integer Programming*, Wiley, 1986
- [3] L.T. Fan, B.Bertók, F.Friedler: *A graph-theoretic method to identify candidate mechanisms for deriving the rate law of a catalytic reaction*, Computers & Chemistry, 2001
- [4] L.Pottier: *Minimal solution of linear diophantine systems: bounds and algorithms*, S.A.F.I.R Project, Institut National de Recherche en Informatique et Automatique, jan. 1991  
<http://www-sop.inria.fr/lemme/Loic.Pottier/rta91.ps>
- [5] Kovács Krisztián, Vizvári Béla, Riedel Miklós, Tóth János: *Decomposition of the permanganate/oxalic acid overall reaction to elementary steps based on integer programming theory*, Physical Chemistry Chemical Physics, (megjelenés alatt)
- [6] E.Contejean and H.Devie: *Solving Systems of linear diophantine equations UNIF'89*, proc. of the third international Workshop on unification,Lambrecht, RFA 89.
- [7] Frank András: *Operációkutatás előadásjegyzet*,  
<http://www.cs.elte.hu/~frank/jegyzet/ulinp.ps>
- [8] Papp Dávid: *Bruttó reakciók felbontása diszkrét matematikai eszközökkel*, TDK dolgozat, konzulens: Dr. Tóth János egyetemi docens, 2003