Szegedi Tudományegyetem
Természettudományi és Informatikai Kar
Bolyai Intézet

Ortogonális tömbök létezése és egészértékű programozás

The existence of orthogonal arrays and integer programming

SZAKDOLGOZAT

Témavezető
DR. NAGY GÁBOR PÉTER
egyetemi tanár

Készítette
KISS REBEKA
matematika Bsc szakos hallgató

Szeged
2020

Tartalomjegyzék

Bevezetés

Az oroszországi Novoszibirszki Állami Egyetem 2014 óta évente megrendezi az NSUCRYPTO Nemzetközi Kriptográfiai Diákolimpiát (`https://nsucrypto.nsu.ru/`). A versenyt online bonyolítják le, regisztrációt követően bárki részt vehet rajta a neki megfelelő kategóriában. A 2018. évi verseny második fordulójában, októberben jelent meg egy feladat, amely a honlap szerint még ma is megoldatlan, megoldását a kitűzők sem ismerik.

Így szól a probléma: [9]

„Az ortogonális tömbök szorosan kapcsolódnak kriptográfiai Boole-függvényekhez. Nevezetesen, korreláció-immun függvények támaszai ortogonális tömböt eredményeznek, amennyiben azok elemeit a tömb sorainak tekintjük. Adott n, t és λ pozitív egészek esetén, ahol $t < n$, t-$(2, n, \lambda)$ ortogonális tömbnek nevezzük azt a $\lambda 2^t \times n$ bináris mátrixot, amelyben az oszlopok bármely t elemű részhalmazára igaz, hogy soraikban minden bináris t-es pontosan λ-szor fordul elő. Keressünk 4-$(2, 11, \lambda)$ *ortogonális tömböt minimális λ értékkel."*

Dolgozatom során erre a feladatra keresem a választ. Az ortogonális tömbök tulajdonságait, valamint a rendelkezésre álló szakirodalom tételeit segítségül hívva a probléma megoldásra kerül, röviden leírva a következőképpen: A Delsarte-féle LP-korlátból következik, hogy $\lambda \geq 6$. A $\lambda = 8$ értékre ilyen paraméterű tömböt Neil Sloane `http://neilsloane.com/oadir/` honlapján található 5-$(2, 16, 8)$ ortogonális tömb módosításával állítottam elő. A feladat nehéz része volt annak bizonyítása, hogy $\lambda \leq 7$ esetben az ortogonális tömbök nem léteznek a fenti paraméterekkel. A dolgozat ezen részében kiemelt szereppel bírt az egészértékű lineáris programozás (ILP). Egy, a számunkra legjelentősebb tétel szerint az, hogy egy ortogonális tömb adott paraméterekkel létezik, ekvivalens a hozzá tartozó ILP megoldásának létezésével.

Az ezen tételhez tartozó ILP-t a SageMath komputeralgebra rendszerben leprogramozva, Python szkripteket és a SCIP ILP megoldót felhasználva azt kaptuk, hogy a rögzített $t = 4$ és $n = 11$ értékekhez $N = \lambda 2^4 = 96$ és 112 sorszám esetén nem adható meg ennyi sorból álló ortogonális tömb. Mint már említettük, a fenti Sloane-konstrukció módosításával $\lambda = 8$-ra létezik ilyen ortogonális tömb, ezzel a feladatot megoldottuk.

1. FEJEZET

A téma ismertetése

1.1. Alapfogalmak

1.1. Definíció. *Legyen A egy $N \times k$-s mátrix, melynek elemei S-ből kerülnek ki: $S = \{0, 1, ..., s-1\}$. Ekkor A* ***ortogonális tömb*** *s szinttel, t erősséggel és λ indexszel, ($0 \leq t \leq k$), ha A-ból bárhogyan elhagyva $k - t$ darab oszlopot, az így kapott $N \times t-s$ mátrix sorai tartalmazzák az összes olyan t hosszú sorozatot, melynek elemei S-ből kerülnek ki, továbbá minden ilyen sorozat pontosan λ-szor szerepel. A fenti mátrix jelölése: $OA(N, k, s, t)$ vagy t-$(s, k, N/s^t)$.*

Példa az egyik legegyszerűbb ortogonális tömbre: $OA(4, 3, 2, 2)$ vagy 2 - $(2, 3, 1)$.

$$\begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{matrix}$$

Az alábbi tömb $OA(8, 4, 2, 3)$ vagy 3-$(2, 4, 1)$, ennek bármely 3 kiválasztott oszlopában minden 3 hosszú $0, 1$ sorozat pontosan egyszer szerepel.

$$\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$$

A mi esetünkben $k = 11, \quad S = \{0, 1\}, \quad t = 4$, továbbá keressük azt az N minimális értéket, amelyre $OA(N, 11, 2, 4)$ létezik, tehát olyan $N \times 11$-es mátrixot kell megadnunk, amelyben bármely 4 oszlopot kiválasztva, az így kapott $N \times 4$-es mátrixban minden lehetséges $0, 1$-esekből álló, 4-hosszú sor megjelenik, sőt, pontosan ugyanannyiszor jelenik meg. A definíció alapján tudjuk, hogy $N = \lambda \cdot s^t$, ez a mi esetünkben annyit jelent, hogy $N = \lambda \cdot 2^4$, így a keresett mátrix sorainak száma osztható 16-tal.

1.2. Ortogonális tömbök tulajdonságai

1.2. Tulajdonság. *Minden t erősségű ortogonális tömb t' erősségű is minden $0 \leq t' < t$-re. Ekkor a t' erősségű tömb indexe: $\lambda \cdot s^{t-t'}$, ahol λ a t erősséghez tartozó index.*

1.3. Tulajdonság. *Legyen A_i egy $OA(N_i, k, 2, t_i), i = 1, 2, ..., r$.*

A az A_i-kből képzett mátrix, azaz

$$A = \begin{bmatrix} A_1 \\ A_2 \\ . \\ . \\ . \\ A_r \end{bmatrix}$$

Ekkor A egy $OA(N, k, 2, t)$, amelyre $N = N_1 + N_2 + ... + N_r$, és $t \geq \min\{t_1, t_2, ..., t_r\}$.

Azt mondjuk, hogy az új A tömböt az $A_1, A_2, \cdots, A_r$ tömbök egymás alá helyezésével kaptuk.

1.4. Tulajdonság. *A fenti jelölésekkel $r = 2 = s$, továbbá, A_i-k mindegyike $OA(N, k, 2, t)$, $i = \{1, 2\}$. Ekkor hozzáfűzve az A_1 minden sorához egy 0-át, A_2 minden sorához egy 1-est, a kapott A mátrix $OA(2N, k + 1, 2, t)$. Ekkor azt mondjuk, hogy az A tömböt az A_1 és A_2 tömbök bővített egymás alá helyezésével kaptuk.*

Ortogonális tömb sorainak, oszlopainak permutálásával kapott mátrix is ortogonális tömb ugyanazokkal a paraméterekkel, továbbá igaz ez akkor is, ha tetszőleges oszlopokban a szinteket (S elemeit) permutáljuk. Az ehhez tartozó definíció:

1.5. Definíció. *Két ortogonális tömb izomorf egymással, ha az egyik megkapható a másikból sorok, oszlopok vagy tetszőleges oszlopokban az $S = \{0, 1, ..., s - 1\}$-beli elemek permutálásával.*

1.6. Tulajdonság. *Az $N \times k$-s $OA(N, k, 2, t)$ minden $N \times k'$-s részmátrixa $OA(N, k', 2, t')$, ahol $t' = min\{k', t\}$. Tehát egy ortogonális tömbből oszlopokat elhagyva a kapott mátrix is ortogonális tömb, melynek erőssége a fenti módon kapható meg.*

Láthatjuk, hogy bizonyos tulajdonságokkal rendelkező OA-hoz megfelelően csatolva egy újabb oszlopot, nagyobb oszlopszámú OA-t kapunk azonos erősséggel, (1.4. Tulajdonság), továbbá OA-ból oszlopokat elhagyva kisebb oszlopszámú OA-t konstruálhatunk (1.6. Tulajdonság).

1.7. Tulajdonság. *Legyen A egy $OA(N,k,2,t)$ mátrix. Permutáljuk a mátrix sorait úgy, hogy az első N/2 sorba azon sorok kerüljenek, melyeknek első eleme 0 (később A_1), a maradék N/2 sorba pedig azok, amelyek 1-essel kezdődnek (később A_2). Ekkor a kapott A^* mátrix csak annyiban különbözik A-tól, hogy sorai más sorrendben vannak, így ugyanúgy $OA(N,k,2,t)$, és előáll a 1.3. Tulajdonság szerint $A_1 = OA(N/2,k,2,t)$ és $A_2 = OA(N/2,k,2,t)$ egymás alá helyezésével. Ezek után ha A_1-ből, majd A_2-ből elhagyjuk az első oszlopot, akkor kettő (nem feltétlenül különböző) $OA(N/2,k-1,2,t-1)$-et kapunk. Tehát adott ortogonális tömbből tudunk készíteni kettő új, feleakkora sorszámú ortogonális tömböt, de ezek nem feltétlenül különböznek.*

1.3. Alsó határok OA sorainak számára

Ha egyelőre pontos értéket nem is tudunk adni $OA(N,11,2,4)$-ban az N minimális értékére, legalább ismerünk rá alsó és felső határokat. Általánosan, ortogonális tömbök esetén kereshetjük azt a minimális N értéket, amelyre adott k,s,t értékek mellett az $OA(N,k,s,t)$ létezik. Ezen minimális sorszám jelölése: $F(k,s,t)$.

A másik keresendő érték az oszlopszám, a többi paraméter ismerete mellett. A 1.6 Tulajdonságban láttuk, hogy egy $OA(N,k,s,t)$-ból oszlopok törlésével $OA(N,k',s,t)$-t kapunk bármely $t \leq k' \leq k$-ra. Így ha meghatározott N,s,t esetén olyan k-kat keresünk, amelyre az adott ortogonális tömb létezik, elég csak azt a maximális k^* értéket — amit $f(N,s,t)$-vel jelölünk — megadnunk, amelyre $OA(N,k^*,s,t)$ még létezik. A legkorábbi felső határ az oszlopok maximális számára C. R. Rao-tól való (1947) . A mi esetünkben nagyobb szerepe lesz az elsőként tárgyalt $F(k,s,t)$ értéknek. Erről szól Rao alábbi eredménye [4, Theorem 2.1].

1.8. Tétel. *Egy $OA(N,k,s,t)$ paraméterei, ahol $t = 2u$ valamely $u \geq 0$ esetén, kielégítik az alábbi egyenlőtlenséget:*

$$N \geq \sum_{i=0}^{u} \binom{k}{i} \cdot (s-1)^i.$$

Ortogonális tömböknél általában $s = 2$-vel dolgozunk, azaz a mátrixokban 0, 1-esek állnak. Ezt feltételezzük a tétel bizonyításakor is. Továbbá a mi esetünkben $k = 11$, $t = 4$, és így $u = 2$, ezen értékek esetén így szól a tétel:

Egy $OA(N,11,2,4)$-t tekintve az ismeretlen N paraméter kielégíti az alábbi egyenlőtlenséget:

$$N \geq \sum_{i=0}^{2} \binom{11}{i} = 1 + 11 + \binom{11}{2}.$$

Bizonyítás. Legyen $A = (a_{ij})$ egy $OA(N, 11, 2, 4)$, ahol $a_{ij} \in \{0, 1\}$, $i = 1, ..., N, \quad j = 1, ..., 11$. Jelölje M a tételbeli, jobb oldali kifejezést. A bizonyítás során az A mátrix segítségével konstruálunk egy olyan H mátrixot, amely $N \times M$-es, és rangja M. Ezek után egyből következik, hogy $M \leq N$, ami pont az állítás.

A H mátrix, amit konstruálunk, nemcsak M rangú, hanem rendelkezik azzal a tulajdonsággal is, hogy $H^T H$ egy $M \times M$-es diagonális mátrix. H-t az A mátrix elemeit felhasználva állítjuk elő:

- H( ) egy $N \times 1$-es mátrix, minden eleme 1-es, a későbbiekben j-vel jelöljük:

$$H() = 1_N = j = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

- $H(1)$ egy $N \times 11$-es mátrix, amelynek elemeit az alábbi módon kapjuk: a $H(1)$ mátrix i-edik sorában és l_1-edik oszlopában ($l_1 = 1, ..., 11$) 1-es elem áll, ha A ugyanezen pozíciójában 0 van, különben pedig -1. Másképpen kifejezve: $h^1_{i,l_1} = (-1)^{a_{i,l_1}}$.

- $H(1, 1)$ egy $N \times \binom{11}{2} = N \times 55$-ös mátrix, oszlopait az A ortogonális tömb oszlopainak kételemű részhalmazaival indexeljük, így $H(1, 1)$ elemeit így jelöljük: $h^{(1,1)}_{i,(l_1,l_2)}$, ahol $l_1, l_2 \in \{1, ..., 11\}$ és $l_1 < l_2$, továbbá $h^{(1,1)}_{i,(l_1,l_2)} = (-1)^{a_{i,l_1}} \dot{(-1)}^{a_{i,l_2}}$, vagyis elemei szintén a $\{-1, 1\}$ halmazból kerülnek ki. Az i-edik sorban és (l_1, l_2)-edik oszlopban álló elem 1-es, ha az A mátrix i-edik sorában az l_1 és l_2 oszlopokhoz tartozó elemek megegyeznek, -1 pedig különben.

Ezek után H az alábbi módon definiált:

$$H = \begin{bmatrix} 1_N & H(1) & H(1,1) \end{bmatrix}$$

H-nak N darab sora, valamint $\binom{11}{0} + \binom{11}{1} + \binom{11}{2} = 1 + 11 + 55 = M$ darab oszlopa van. $H^T H$ egy $M \times M$-es mátrix. Belátjuk, hogy $H^T H = N I_M$.

1. Először is tisztázzuk, hogy $H^T H$ főátlójának minden eleme N: vesszük egy-egy oszlop skaláris szorzatát önmagával, ekkor az összegben $1 \cdot 1$ és $(-1) \cdot (-1)$-ek fognak szerepelni, összesen N-szer, azaz összeadunk N darab 1-est.

2. j ortogonális $H(1)$ oszlopaira, ha minden $l_1 \in \{1, ...11\}$ esetén, ahol l_1 jelöli $H(1)$ oszlopait, $< j, l_1 > = 0$, vagyis az l_1 oszlopban az elemek összege 0.

Ez csak akkor teljesülhet, ha az l_1 oszlopban ugyanannyi 1-es és -1-es szerepel, ami azt jelentené — $H(1)$ definíciója miatt — , hogy A l_1 indexű oszlopában ugyanannyi 0 van, ahány 1-es. Ez utóbbi igaz, hiszen A ortogonális tömb 4 erősséggel, és ekkor kihasználva a 1.2-es tulajdonságot, tudjuk hogy 1-erősséggel is, azaz bármely oszlopát nézve, abban ugyanannyi — méghozzá $N/2$ darab — 0 szerepel, ahány 1-es.

3. j ortogonális $H(1,1)$ oszlopaira $\Leftrightarrow$ $H(1,1)$ oszlopaiban az elemek összege 0, azaz az 1-esek és -1-esek száma egyenlő.

   A $h^{(1,1)}_{i,(l_1,l_2)}$ pozíciójú elem akkor 1, ha a meghatározásában résztvevő A-beli, a_{i,l_1}, a_{i,l_2} elemek megegyeznek, -1 pedig, ha különböznek. A-nál ismét kihasználva a 1.2. Tulajdonságot, tudjuk, hogy 2 erősségű is, így bármely két oszlopát kiválasztva: $\{l_1, l_2\}$, az így kapott $N \times 2$-es részmátrixban minden 2-hosszú 0,1-es ugyanannyiszor ($N/4$-szer) szerepel $\Rightarrow$ ebben a részmátrixban a két oszlop elemei $N/2$ helyen különböznek, hiszen $N/4$ darab $(0,1)$-es és $N/4$ $(1,0)$-es van, és $N/2$ helyen egyeznek meg: a $(0,0)$ és $(1,1)$-esek száma $\Rightarrow$ $H(1,1)$-ben az 1-esek és -1-esek száma egyenlő, egyaránt $N/2, N/2$, ezek összege 0.

4. $H(1)$ ortogonális $H(1,1)$-re, ha tetszőleges $H(1)$-beli és $H(1,1)$-beli oszlop skaláris szorzata 0. Felfoghatjuk ezt úgy, hogy A-ból kiválasztunk 3 - speciális esetben 2 - oszlopot, és az ezek közötti összefüggéseket figyeljük.

   Legyen a $H(1)$-ből kiválasztott oszlop indexe l_1, a $H(1,1)$-ből kiválasztotté pedig (l_2, l_3), $l_1, l_2, l_3 \in \{1, ..., 11\}$, valamint feltehető, hogy $l_1 \leq l_2 < l_3$.

   A továbbiakban két esetet vizsgálunk:

   (i) Ha $l_1 = l_2$:

   A speciális esethez jutottunk, amikor A-ból csak 2 oszlopot kell kiválasztanunk. A $H(1)$-ből kiválasztott oszlop: $h^1_{l_1}$, a $H(1,1)$-ből kiválasztott pedig $h^{(1,1)}_{(l_1,l_3)}$.
   $h^{(1,1)}_{i,(l_1,l_3)} = 1$, ha az a_{i,l_1} és a_{i,l_3} elemek azonosak A-ban, különben -1.
   Gondoljuk meg, hogy a $h^1_{l_1}$ és $h^{(1,1)}_{(l_1,l_3)}$ oszlopok skaláris szorzat vételéhez az alábbi eseteket kell vizsgálnunk:

   - $a_{i,l_1} = 0$ és $a_{i,l_1} = a_{i,l_3}$ $\Rightarrow$ $(0,0)$ áll az $N \times 2$-es részmátrix i-edik sorában, amelyet A-ból az $\{l_1, l_3\}$ oszlopok kiválasztásával kapunk. Ekkor $h^1_{i,l_1} = 1$, $h^{(1,1)}_{i,(l_1,l_3)} = 1 \Rightarrow$ a skalárszorzat i-edik komponense 1.

- $a_{i,l_1} = 0$ és $a_{i,l_1} \neq a_{i,l_3} \Rightarrow$ a részmátrix i-edik sorában $(0,1)$ áll. Ekkor $h^1_{i,l_1} = 1$, $h^{(1,1)}_{i,(l_1,l_3)} = -1 \Rightarrow$ a skalárszorzat i-edik komponense -1.
- $a_{i,l_1} = 1$ és $a_{i,l_1} \neq a_{i,l_3} \Rightarrow$ a részmátrix i-edik sorában $(1,0)$ áll. Ekkor $h^1_{i,l_1} = -1$, $h^{(1,1)}_{i,(l_1,l_3)} = -1 \Rightarrow$ a skalárszorzat i-edik komponense 1.
- $a_{i,l_1} = 1$ és $a_{i,l_1} = a_{i,l_3} \Rightarrow$ a részmátrix i-edik sorában $(1,1)$ áll. Ekkor $h^1_{i,l_1} = -1$, $h^{(1,1)}_{i,(l_1,l_3)} = 1 \Rightarrow$ a skalárszorzat i-edik komponense -1.

Az A ortogonális tömb 2 erőséggel, így mind a 4 fenti, 2-hosszú 0, 1-es kombináció $N/4$-szer tűnik fel A-nak bármely $N \times 2$-es részmátrixában. Ezek alapján a skalárszorzatban $2 \cdot N/4$, azaz $N/2$ darab 1-es és $N/2$ darab -1-es van, ezeket összeadva valóban 0-hoz jutunk.

(ii) Ha $l_1 < l_2$:

Ekkor A-nak három különböző oszlopával dolgozunk.

Mivel az A ortogonális tömb 3 erősségű is, így teljesülnie kell annak, hogy tetszőlegesen kiválasztott 3 oszlopban a 3-hosszú 0, 1-esek ugyanannyiszor szerepelnek: mindegyik $N/2^3$-szor.

A $H(1)$-beli l_1 és a $H(1,1)$-beli (l_2, l_3) oszlop skaláris szorzatának i-edik komponense, azaz $h^1_{i,l_1} h^{(1,1)}_{i,(l_2,l_3)} = 1$,
ha $a_{i,l_1} = 0$ ($\Rightarrow h^1_{i,l_1} = 1$), továbbá az a_{i,l_2} és a_{i,l_3}-beli elemek megegyeznek ($\Rightarrow h^{(1,1)}_{i,(l_2,l_3)} = 1$),
vagy ha $a_{i,l_1} = 1$ ($\Rightarrow h^1_{i,l_1} = -1$), továbbá az a_{i,l_2} és a_{i,l_3}-beli elemek eltérőek ($\Rightarrow h^{(1,1)}_{i,(l_2,l_3)} = -1$).
Ez négy lehetőség: $\{(0,0,0), (0,1,1), (1,1,0), (1,0,1)\}$.

Az $\{l_1, l_2, l_3\}$ oszlopok kiválasztásával kapott $N \times 3$-as részmátrixban ezek összesen $4 \cdot N/2^3$-szor szerepelnek a 3-as erősség miatt, azaz $N/2$-ször van a skaláris szorzatban 1-es, és mivel a maradék 4 darab 3-hosszú kombináció a skaláris szorzatban -1-est eredményez, ezért szintén $N/2$-ször jelenik meg -1-es, így az elemek összege a skalárszorzatban 0.

5. Láttuk, hogy $H(1)$ ortogonális $H(1,1)$-re, így az is teljesül, hogy $H(1,1)$ ortogonális $H(1)$-re.

   A fentiekhez hasonlóan működik a maradék kettő eset:

6. Ahhoz, hogy belássuk, hogy $H(1)$ önortogonális, az A ortogonális tömb 2-es erősségét kell használnunk, míg $H(1,1)$ önortogonális tulajdonsága abból következik,

hogy A 3-as, és 4-es erősségű is.

Így megmutattuk, hogy $H^T H$ egy nem-elfajuló diagonális mátrix, rangja M.

□

Esetünkben a Rao-határt alkalmazva $k = 11, s = 2, t = 4, u = 2$ paraméterekkel azt kapjuk, hogy:

$$N \geq \binom{11}{0} \cdot (2-1)^0 + \binom{11}{1} \cdot (2-1)^1 + \binom{11}{2} \cdot (2-1)^2,$$

így $OA(N, 11, 2, 4)$ esetén $N \geq 67$. Az ortogonális tömbök sorszámainak alsó határára ezt a fenti Rao-határt, valamint az úgynevezett LP-határt ismertetjük.

1.9. Definíció. *Egy lineáris programozási feladat (LP- Linear Programming) a következőképpen formalizálható. Legyen $A \in \mathbb{R}^{k \times n}$ mátrix, $b \in \mathbb{R}^k$ k-komponensű, $c \in \mathbb{R}^n$ n-komponensű vektor, mindhárom adott. Keresünk olyan n-komponensű x vektort, amely minimalizálja a $c^T x$ lineáris függvényt (skaláris szorzatot) az $Ax \leq b$, $x \geq 0$ lineáris feltételek mellett.*

Amennyiben szerepel további feltételként, hogy az x vektor minden komponense egészértékű, azaz $x \in \mathbb{Z}^n$, akkor egészértékű lineáris programozásról, ILP-ről beszélünk.

A [4]-beli 4.15. Tétel, a Delsarte-féle LP-korlát így szól:

1.10. Tétel. *Legyenek adottak a k, s és t értékek.*

$A_0, A_1, ..., A_k$ a lenti lineáris programozási feladat változói.

Definiáljuk a B_i kifejezéseket az alábbi módon:

$$B_i = \sum_{j=0}^{k} A_j \cdot P_i(j), 0 \leq i \leq k$$

Ahol $P_i(j)$ az úgynevezett Krawtchouk-polinom:

$$P_i(j) = \sum_{r=0}^{i} (-1)^r (s-1)^{i-r} \binom{j}{r} \binom{k-j}{i-r}, 0 \leq i \leq k$$

Vegyük a következő lineáris programozási feladatot:

$$min\ A_0 + A_1 + ... + A_k$$

feltéve, hogy:

$$A_0 \geq 1, A_i \geq 0, 1 \leq i \leq k$$

$$B_0 \geq 1, B_i \geq 0, 1 \leq i \leq k$$

$$B_1 = B_2 = ... = B_t = 0$$

Legyen $N_{LP}(k, t+1) = min(\sum_{i=0}^{k} A_i)$ *optimális megoldás.*
Ekkor $OA(N, k, s, t)$*-re* $N \geq N_{LP}(k, t+1)$.

A fenti tétel $s = 2$ mellett leprogramozva így szól:

```
def lp(k,t):
  def pol(k,i,j):
    return sum((-1)**(r)*binomial(j,r)*binomial(k-j,i-r) for r in range(i
   +1))
  m=matrix(k+1,k+1)
  for i in range(k+1):
    for j in range(k+1):
      m[i,j]=pol(k,i,j)

  p = MixedIntegerLinearProgram(maximization=False,solver="GLPK")
  a = p.new_variable()
  p.add_constraint(a[0] >= 1)
  for i in range(k+1):
    p.add_constraint(a[i] >= 0)
  p.add_constraint(sum(a[j]*m[0,j] for j in range(k+1)) >= 1)
  for i in range(k+1):
    p.add_constraint(sum(a[j]*m[i,j] for j in range(k+1)) >= 0)
  for i in range(1,t+1):
    p.add_constraint(sum(a[j]*m[i,j] for j in range(k+1)) == 0)
  p.set_objective(sum(a[i] for i in range(k+1)))
  print(sol = p.solve())
```

A programot futtatva a $k = 11, t = 4$ értékekkel:

```
(sol = 85.33333333333314)
```

Az LP határ szerint $N \geq 85.3$, továbbá mivel N osztható 16-tal, így az első lehetséges érték az adott probléma megoldására 96.

Érdemes megjegyezni, hogy a $t = 4$, $s = 2$ esetben a $k = 13, 14, 15$ értékek esetén az LP-határ pontos, vagyis léteznek olyan konstrukciók a felsorolt ortogonális tömbökre, amelyek sorainak száma éppen annyi, amennyi az LP-határ által megkövetelt minimum ($N = 128$).

2. FEJEZET

Konstrukció keresés

2.1. Ismert konstrukciók

Imént említettük, hogy $k = 13, 14, 15$ esetén az LP-határ éles az ortogonális tömb sorainak számára, vagyis ezen k-kra ismertek $OA(128, k, 2, 4)$ konstrukciók. A dolgozat elején bemutatott 1.6. Tulajdonság miatt nekünk az utolsó, 15 oszlopos tömb fontos, hiszen a másik kettő OA-t megkapjuk ebből oszlopok elhagyásával.

Nézzük meg, hogy milyen ismert konstrukciók vannak:

- N. J. A. Sloane honlapján szerepel egy $OA(256, 16, 2, 5)$, lásd: [8].

  Az 1.7. Tulajdonságot használjuk: Permutáljuk a sorait úgy, hogy az első 128 sorba azok kerüljenek, amiknek első eleme 0, (így a maradék 128 sor 1-essel kezdődik). Legyen ezek után az első 128 sorból álló mátrix A_1, a második 128 sorból álló pedig A_2. Hagyjuk el mindkét mátrixból az első oszlopot. Ezzel két, nem feltétlenül különböző $OA(128, 15, 2, 4)$-et kapunk.

- Pieter Eendebak közöl a honlapján egy $OA(128, 15, 2, 4)$-t, ezentúl azt is megjegyzi, hogy az ilyen paraméterekkel rendelkező OA izomorfia erejéig egyértelmű, lásd: [2].

A fentiek után könnyű dolgunk van: az ismert, 15 oszlopos konsrukciók valamelyikéből elhagyunk (tetszőleges) 4 oszlopot, és így egy $OA(128, 11, 2, 4)$-hez jutunk.

Ennek következményeként $F(11, 2, 4) \leq 128$, vagyis a későbbiekben a feladatunk az, hogy $N = 96, 112$ esetén megkeressük $OA(N, 11, 2, 4)$-t vagy megmutassuk, hogy ilyen ortogonális tömbök nem léteznek.

2.2. Az egészértékű programozás használata

A hivatkozásban említett szerzők közül sokan foglalkoztak a konstrukció kereséssel, közülük egy D.A. Bulutoglu [1], aki a lineáris programozást hívta segítségül adott paraméterű OA-k keresésére. Ebben a fejezetben az általa kidolgozott eljárás kerül bemutatásra.

Ahhoz, hogy ezt ismertessük, vegyük át a használandó jelöléseket:

- D_{2^k}: 2^k sorból, k oszlopból álló mátrix, melyben 0-nak és 1-nek minden k hosszú kombinációja pontosan egyszer szerepel. Ezen tömb sorait — azaz a k-hosszú 0,1-eseket — lexikografikusan növekvő sorrendbe rendezzük: egy d_i sor lexikografikusan kisebb a d_j sornál, ha az első olyan k indexre, melyre $d_{ik} \neq d_{jk}$, teljesül, hogy $d_{ik} < d_{jk}$.

- D: k oszlopos mátrix, melynek sorai D_{2^k} soraiból kerülnek ki, úgy, hogy egy-egy sor többször is előfordulhat.

- x: $x \in (\mathbb{N}_0)^{2^k}$, ebből meghatározható D: x i-edik komponense megmondja, hogy hányszor szerepel D-ben a D_{2^k} i-edik sora.

- Adott t és k esetén:

  T: $T = \left\{A_1, A_2, ..., A_{\binom{k}{t}}\right\}$, ahol A_j -k mindegyike $2^t \times 2^k$ méretű mátrix. Ezeket úgy kapjuk, hogy vesszük t darab $I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ és $k-t$ darab $1_2^T = \begin{pmatrix} 1 & 1 \end{pmatrix}$ mátrix Kronecker-szorzatát a tényezők sorrendjét tekintve az összes lehetséges módon. Azt, hogy hány ilyen szorzat van, meg tudjuk határozni: kiválasztjuk, hogy a t darab I_2 tényező hány helyre kerülhet a k-tényezős szorzatban: így $\binom{k}{t}$ darab A_j-vel lesz dolgunk.

  A Kronecker-szorzást az $m \times n$-es U és $p \times q$-s V mátrixok esetén így definiáljuk:

$$U \otimes V = (u_{ij} \cdot V) = \begin{pmatrix} u_{11}V & \dots & u_{1n}V \\ \vdots & \ddots & \vdots \\ u_{m1}V & \dots & u_{mn}V \end{pmatrix},$$

  ahol $U \otimes V$ mérete $mp \times nq$.

  Így tehát adott A_j-re: $A_j = M_{1j} \otimes M_{2j} \otimes ... \otimes M_{kj}$, ahol

$$M_{ij} \in \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \end{pmatrix} \right\}.$$

  Ezentúl legyen $L_j = \left\{i_{1j}, i_{2j}, ..., i_{tj}\right\}$ azon i-k halmaza, amelyekre $M_{ij} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ az A_j-t létrehozó szorzatban.

- p_{max}: felső határ arra, hogy adott N, k, t paraméterek esetén tetszőleges $OA(N, k, 2, t)$-ben legfeljebb hányszor szerepelhet egy D_{2^k}-beli sor.

Ahogyan azt fent említettük, a keresendő OA-t lineáris programozás segítségével szeretnénk megtalálni.

A konstrukció megadásában az alábbit használjuk: A fenti jelölések használatával $A_j x = \lambda \cdot 1_{2^t}$ akkor és csak akkor, ha az összes 2^t bináris kombináció λ-szor szerepel azon részmátrixban, amelyet D-ből kapunk úgy, hogy vesszük azon indexű oszlopait, amelyek L_j-ben szerepelnek (azaz pont t darabot).

Ahhoz, hogy D ortogonális tömb legyen, ezt a tulajdonságot bármely t darab, tetszőlegesen kiválasztott oszlopából kapott részmátrixának teljesítenie kell, ezt mondja a következő tétel [1, Theorem 2].

2.1. Tétel. *Legyen $s \geq 2, k \geq t, t \geq 1$, λ, p_{max} adott egész számok. A_i-k a fent definiáltak szerintiek. $A(s, k, t)$ jelölje azt a $\binom{k}{t} \cdot s^t \times s^k$-s mátrixot, melyre*

$$A(s,k,t) = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_{\binom{k}{t}} \end{pmatrix}$$

Ekkor a következő ILP(1) akkor és csak akkor megoldható, ha létezik $OA(\lambda s^t, k, s, t)$, és ebben bármely D_{s^k}-beli sor legfeljebb p_{max}-szor fordul elő.

$$min\ 1_{s^k}^T x$$

$$A(s,k,t) \cdot x = \lambda \cdot 1_{\binom{k}{t}s^t}$$

$$x \in \{0, 1, \ldots, p_{max}\}^{s^k}.$$

Ezen ILP(1) feladat minden egész x megoldása egyértelműen meghatároz egy $D = OA(\lambda s^t, k, s, t)$-t, és ez fordítva is teljesül: minden $OA(\lambda s^t, k, s, t)$ meghatároz egy x vektort, amely optimális megoldása ennek az $ILP(1)$-nek.

Az $A(s, k, t)$ mátrixot megadó parancs így szól:

```
1 def boldA(k, t):
2   id = Matrix([[1, 0], [0, 1]])
3   jj = Matrix([[1, 1]])
```

```
  li = []
  for c in Combinations(k, t):
    Aj = Matrix([[1]])
    for j in range(k):
      if (j in c):
        m = id
      else:
        m = jj
      Aj = Aj.tensor_product(m)
    li.append(Aj)
  return block_matrix(len(li), 1, li)
```

A tétel elején szerepel, hogy $s, k, t, \lambda, p_{max}$ adottak. Abból, hogy milyen paraméterekkel keresünk ortogonális tömböt, az első négy érték közvetlenül adódik. A p_{max}-ot a lehető legkisebbre szeretnénk választani, hiszen így könnyebben megoldható a lineáris programozási feladat. Könnyen meggondolható, hogy $p_{max} \leq \lambda$. Tudunk adni azonban szigorúbb felső korlátot, erről szól az [1, 5. Lemma], amely szintén egy lineáris programozási feladatot használ. Mielőtt ismertetnék ezt a felső korlátot, tekintsük a [4, 2.7. Lemma]-t.

2.2. Lemma. *Legyen $u = (u_1, u_2, ..., u_k)$ egy rögzített sor egy $OA(\lambda s^t, k, s, t)$-ban, $A_i(u)$, $(0 \leq i \leq k)$ jelölje azon $v = (v_1, v_2, ..., v_k)$ sorok számát az OA-ban, amelyek pontosan i komponensben különböznek a rögzített u sortól, azaz pontosan i darab a index esetén teljesül, hogy $u_a \neq v_a$. Ekkor a következő egyenlőség fennáll:*

$$\sum_{i=h}^{k} \binom{i}{h} A_{k-i}(u) = \lambda s^{t-h} \binom{k}{h}, \ 0 \leq h \leq t,$$

méghozzá u választásától függetlenül.

Bizonyítás. Miután egyes faktorok szintjeit megváltoztatjuk a tömbben, megkapjuk, hogy a rögzített sor az $u = (0, 0, ..., 0)$. Ha h=0 , akkor a fenti formula azt állítja, hogy

$$\sum_{i=0}^{k} A_i(u) = \lambda s^t.$$

Ez igaz, hiszen az egyenlőség mindkét oldala N-nel, azaz a sorok számával egyenlő: a jobb oldalon ez egyből látható: így definiáltuk az OA első paraméterét, amivel dolgozunk, a bal oldal pedig könnyen meggondolható: kiírva az $A_0 + A_1 + ... + A_k$ összegről van szó, ami megszámolja, hogy hány sor van, amely $0, 1, ..., k$ helyen tér el a k-hosszú, csupa 0-át tartalmazó sortól.

Ha $1 \leq h \leq t$, az egyenlőség érvényességét úgy látjuk be, hogy kétféleképpen számoljuk meg azon különböző, h hosszú sorozatokat OA-ban, amelyeknek minden eleme 0.

Mivel az adott, t erősségű OA-nk h erősséggel is ortogonális tömb ($0 \leq h \leq t$), a hozzátartozó λs^{t-h} indexszel, így tetszőleges $N \times h$ méretű részmátrixában λs^{t-h} darab sor van, amely h hosszú, és csak 0-kat tartalmaz.

Az eredeti ortogonális tömbnek k oszlopa van, ebből $\binom{k}{h}$-féleképpen tudjuk kiválasztani az $N \times h$-s részmátrixokat, tehét az $N \times k$-s ortogonális tömbben összesen $\lambda s^{t-h}\binom{k}{h}$ darab különböző, h-hosszú 0-ás sorozat van.

Másrészről az eredeti OA minden sorában, amely i darab 0-át tartalmaz, ($i \geq h$), $\binom{i}{h}$ darab h hosszú 0-ás sorozatot tudunk kiválasztani. Továbbá A_{k-i} azon sorok száma, amelyek pontosan i darab 0-át tartalmaznak, ennélfogva a különböző, h-hosszú, 0-ákat tartalmazó sorozatok száma az $N \times k$-s ortogonális tömbben ezen összeszámlálási módszer alapján $\sum_{i=h}^{k} \binom{i}{h} A_{k-i}(u)$.

Ebből következik a fenti egyenlőség fennállása. □

Ekkor a p_{max}-ra vonatkozó becslés:

2.3. Lemma. *Legyen $Y_0^* = max\ Y_0$ olyan, hogy teljesül az alábbi:*

$$\sum_{i=h}^{k} \binom{i}{h} Y_{k-i} = \lambda s^{t-h} \binom{k}{h},\ 0 \leq h \leq t,$$

$$Y_i \in \mathbb{Z}_+,\ 0 \leq i \leq k.$$

Ekkor Y_0^* felső határ arra, hogy egy sor legfeljebb hányszor fordulhat elő egy $OA(\lambda s^t, k, s, t)$-ban.

Bizonyítás. Legyen $u = (u_1, u_2, ..., u_k)$ egy tetszőleges sor $OA(\lambda s^t, k, s, t)$-ban és $Y_i(u)$, ($0 \leq i \leq k$) jelölje a számát azon r soroknak $OA(\lambda s^t, k, s, t)$-ban, amelyekre $r - u$ pontosan i darab nemnullát tartalmaz. (Azaz megadja, hogy rögzített u vektor esetén hány vektor különbözik pontosan i komponensben tőle.) Vegyük a következő $ILP(2)$-et:

$$Y_0^*(u) = max\ Y_0(u)$$

$$\sum_{i=h}^{k} \binom{i}{h} Y_{k-i}(u) = \lambda s^{t-h} \binom{k}{h},\ 0 \leq h \leq t$$

$$Y_i(u) \in \mathbb{Z}_+,\ 0 \leq i \leq k.$$

Ekkor az [4, 2.7. Lemma], vagyis az itteni 2.2. Lemma azt állítja, hogy $(Y_0(u), Y_1(u), ..., Y_k(u))^T$ kielégíti a fenti $ILP(2)$ feltételeit u választásától függetlenül.

Könnyen látható, hogy $Y_0^*(u)$ egy felső határ arra, hogy adott sor legfeljebb hányszor tűnhet fel egy tetszőleges $OA(\lambda s^t, k, s, t)$-ban: $Y_0(u)$ összeszámolja azon sorokat egy adott $OA(\lambda s^t, k, s, t)$-ban, amelyek u-val megegyeznek, vagyis megadja, hogy az u sor hányszor szerepel az OA-ban. Vesszük $Y_0(u)$-t az összes, paramétereket teljesítő tömbben, majd legyen $Y_0^*(u)$ ezek maximuma. Ebből következik a tétel. □

A 2.3. Lemma egy egészértékű lineáris programozási feladatot határoz meg. Ez a SAGE-ben így néz ki:

```
def p_max(N, k, t):
  lam = N/(2 ^ t)
  q = MixedIntegerLinearProgram(maximization=True, solver="glpk")
  y = q.new_variable(nonnegative=True, integer=True, name='y')
  q.set_objective(y[0])
  for h in range(t+1):
    q.add_constraint(sum(binomial(i, h)*y[k-i] for i in range(h, k+1)) ==
   lam*2 ^ (t-h)*binomial(k, h))
  return q.solve()
```

Látjuk, hogy ahhoz, hogy a 2.1 Tétel leprogramozható legyen, ismernünk kell p_{max}-ot és $A(s, k, t)$-t is. Ezeket az előbbi két paranccsal definiáltuk, most már csak az a dolgunk, hogy használatukkal megadjuk az ILP(1)-hez szükséges feltételeket, és megoldjuk a feladatot. Ha a megoldandó feladatban szereplő paraméterekkel dolgozunk, továbbá N-et 128-nak választjuk:

```
N,k,t = 128,11,4
lam = N/(2 ^ t)
pm = p_max(N, k, t)
bbA = boldA(k, t)
p = MixedIntegerLinearProgram(maximization=False, solver="glpk")
x = p.new_variable(nonnegative=True, integer=True, name='x')
p.set_objective(sum(x[i] for i in range(2 ^ k)))
for i in range((binomial(k, t)*2 ^ t)):
  l = []
  for j in range((2 ^ k)):
    if bbA[i,j] == 1:
      l.append(j)
  p.add_constraint((sum(x[l[k]] for k in range(len(l)))) == lam)
for i in range(2 ^ k):
```

```
15   p.add_constraint(0 <= x[i] <= pm)
16 p.solve(log=2)
17 for i in range(2 ^ k):
18   print(p.get_values(x[i]))
```

Az $ILP(1)$ megoldásának kapott x vektor izomorfia erejéig egyértelműen meghatározza a $D = OA(\lambda s^t, k, s, t)$ ortogonális tömböt, hiszen megmondja, hogy D-ben D_{s^k} melyik sora hányszor szerepel. Éppen ezért hívjuk D-t az x-nek megfelelő $OA(\lambda s^t, k, s, t)$-nek, x-et pedig indikátorvektornak.

A tételben szereplő $1_{s^k}^T \cdot x$ a célfüggvény, ami — amennyiben létezik x megoldás — megmondja, hogy a kapott megoldás által meghatározott D-nek hány sora van. A feltételeket a második és a harmadik sor adja, itt $A(k, s, t)$-t indikátormátrixnak hívjuk. Észrevehetjük, hogy amennyiben létezik megoldása az ILP(1) feladatnak, akkor az abból meghatározásra kerülő D ortogonális tömb t erősséggel, hiszen a második sorban szereplő $A(s, k, t) \cdot x = \lambda \cdot 1_{\binom{k}{t}2^t}$ feltétel teljesül, ami pont azt jelenti, hogy minden lehetséges, 2^t hosszú, bináris sorozat pontosan λ-szor szerepel abban a mátrixban, melyet D-ből tetszőleges t darab oszlop kiválasztásával kapunk.

3. FEJEZET

A feladat megoldása a bemutatott tételek, lemmák segítségével

3.1. Az eddigi eredmények összefoglalása

Számunkra az alábbi két ortogonális tömb jelentős:

(1) $OA(96, 11, 2, 4)$

(2) $OA(112, 11, 2, 4)$

Ahhoz, hogy belássuk, hogy egy adott $OA(N, k, s, t)$ ortogonális tömb nem létezik, elég megmutatnunk, hogy $OA(N, k^*, s, t)$ nem létezik valamilyen $t \leq k^* < k$ esetén.

Vegyük észre, hogy az adott paraméterekhez tartozó lineáris programozási feladatban a feltételek száma függ az oszlopszámtól: az ILP(1) második sorát vizsgálva $\binom{k}{t}2^t$ feltételt kell teljesítenie a keresendő x vektornak.

Úgy, ahogy törekedtünk p_{max} minimalizására, célunk az is, hogy minél kevesebb feltétel szerepeljen az optimalizálási feladatban, hisz így dolgozunk a lehető legegyszerűbb lineáris programozási feladattal.

Az könnyen látszik, hogy ha tudjuk, hogy egy adott k^* oszlopszámra (rögzített N, s, t esetén), hogy $OA(N, k^*, s, t)$ nem létezik, akkor nem létezik konstrukció $k > k^*$ esetén sem. Ha már k* oszlopra nem teljesült, hogy bármely t darabot kiválasztva közülük az összes t-hosszú sorozat szerepel a sorokban, méghozzá ugyanannyiszor, akkor a tömböt további oszlopokkal bővítve a kiválasztás során biztosan találunk az oszlopoknak olyan t-elemű részhalmazát, amely nem felel meg a feltételeknek: például ha azokat az oszlopokat választjuk, amelyek az $N \times k^*$ méretű tömbben nem teljesítették az erősséget.

Így tehát amint megtaláljuk a legkisebb k-t, amelyre adott N, s, t esetén $OA(N, k, s, t)$ nem létezik, nem is szükséges további vizsgálatot tennünk a k értékének növelésével.

3.2. A probléma megoldása

A 2.1-es Tétel-beli ILP(1) optimalizálási feladattal dolgozunk tovább. Az ehhez szükséges programokat a fent látott módon a SageMath komputeralgebra rendszerben leprogramoztuk, majd ezeket exportálva a SCIP ILP megoldó segítségével vizsgáljuk, hogy a keresett paraméterekre milyen megoldások adódnak.

Az $N = 96$-os esetben $k = 7$-re létezik ortogonális tömb, hiszen a programozási feladatnak van megoldása, míg $OA(96, 8, 2, 4)$ esetén az ILP(1) feladatnak nincs megoldása, amiből az következik, hogy ilyen ortogonális tömb nem létezik. $N = 112$-t vizsgálva $k = 6$-ra az ILP(1)-nek van megoldása, $k = 7$-re azonban nincsen olyan x, amely teljesítené az adott feltételeket, tehát nem tudunk konstruálni $OA(112, 7, 2, 4)$-et.

Az alábbi táblázat tartalmazza a 4 számunkra fontos ortogonális tömbhöz tartozó lineáris programozási feladatok eredményeit, futási időket és a feltételek számát. A táblázatban rövidítések szerepelnek, ezek jelentései:

- OSF = Optimal solution found, vagyis a program talált optimális megoldást az ILP feladatra.
- INF = Infeasible, vagyis a program lefutott, de nem létezik megoldása a feladatnak.

A harmadik oszlopban az egyes futási időket mutatjuk, másodpercben mérve. Azon két tömbnél, amikor az optimális megoldás létezik, gyorsan megkaptuk az eredményeket. A másik esetben, amikor a feladatnak nincs megoldása, a futás több ideig tartott, különösen $OA(96, 8, 2, 4)$-nél, itt több, mint 14 óra kellett. Ennek oka az, hogy itt szerepel a legtöbb feltétel: összesen $\binom{8}{4} \cdot 2^4 + 2 \cdot 2^8 = 1632$.

	Eredmény	Futási idő	Feltételek száma
OA(96,7,2,4)	OSF	2.00	816
OA(96,8,2,4)	INF	51630.00	1632
OA(112,6,2,4)	OSF	0.00	368
OA(112,7,2,4)	INF	481.00	816

Eredményeink megerősítik a [1] cikk 661. oldalán szereplő táblázatban foglaltakat. Az $N = 96$ esetén két ortogonális tömb szerepel: $OA(96, 7, 2, 4)$ és $OA(96, 8, 2, 4)$. Az előbbihez tartozó ILP(1)-re 4 egymással nem izomorf, míg az utóbbira 0 megoldás adódott, $N = 112$-re vonatkozóan pedig a szereplő tömbök: $OA(112, 6, 2, 4)$ és $OA(112, 7, 2, 4)$, az első esetben 3 egymással nem izomorf, a második esetben 0 megoldást kapott a szerző.

Összefoglalva a fentieket $k = 11, s = 2, t = 4$ esetén:

- A Rao határ: 67.
- Az LP határ: 85.3.
- A keresendő ortogonális tömb sorainak számának 16-tal oszthatónak kell lennie, hogy az OA-kra vonatkozó feltétel teljesüljön: minden 2^4 hosszú 0,1-esekből álló kombináció pontosan ugyanannyiszor szerepel OA egy tetszőleges $N \times 4$ méretű részmátrixában, amelyet OA-ból oszlopok elhagyásával kapunk. Így a legkisebb lehetséges sorszám a fenti k, s, t paraméterek mellett 96.
- Adott paraméterek generálnak egy lineáris programozási feladatot, amelynek megoldhatósága egyértelműen megmondja, hogy konstruálható-e a rögzített paraméterekre ortogonális tömb, és ha igen, akkor a megoldás meghatároz egy ilyen konstrukciót.
- $OA(96, 8, 2, 4)$ esetén az ILP(1)-nek nincs megoldása, azaz ilyen ortogonális tömb nem létezik, így tehát nem konstruálható $OA(96, k', 2, 4)$, tetszőleges $k' > 8$-at választva sem.
- $OA(112, 7, 2, 4)$ esetén az ILP(1)-nek nincs megoldása, azaz ilyen ortogonális tömb nem létezik, így tehát nem konstruálható $OA(112, k'', 2, 4)$, tetszőleges $k'' > 7$-et választva sem.
- $OA(128, 11, 2, 4)$-ra ismert konstrukció.

Abból, hogy $OA(96, 8, 2, 4)$ és $OA(112, 7, 2, 4)$ nem létezik, az következik, hogy $OA(96, 11, 2, 4)$ és $OA(112, 11, 2, 4)$ sem megkonstruálható.

3.3. Válaszadás a kiinduló problémára

A dolgozat elején leírt feladatban az adott k, s, t értékekre kellett meghatároznunk azt a minimális λ értéket, melyre $OA(\lambda s^t, k, s, t)$ létezik. Ez ekvivalens azzal, hogy ezekre a k, s, t értékekre meghatározzuk a minimális N értéket, hiszen $\lambda = N/s^t$.

Az imént leírtak alapján arra engedünk következtetni, hogy ez az érték az $N = 128$, vagyis a feladat megoldása: $\lambda = 8$, ez a minimális érték, amelyre $OA(\lambda 2^4, 11, 2, 4)$ létezik.

Irodalomjegyzék

[1] D. A. Bulutoglu and F. Margot, *Classification of orthogonal arrays by integer programming*, J. Statist. Plann. Inference **138** (2008), no. 3, 654–666, DOI 10.1016/j.jspi.2006.12.003. MR2382560

[2] Pieter Eendebak, *Complete series of non-isomorphic orthogonal arrays* (2020), `http://www.pietereendebak.nl/oapackage/series.html`. Accessed 2020.

[3] Gerald Gamrath and Daniel Anderson and Ksenia Bestuzheva and Wei-Kun Chen and Leon Eifler and Maxime Gasse and Patrick Gemander and Ambros Gleixner and Leona Gottwald and Katrin Halbig and Gregor Hendel and Christopher Hojny and Thorsten Koch and Pierre Le Bodic and Stephen J. Maher and Frederic Matter and Matthias Miltenberger and Erik Mühmer and Benjamin Müller and Marc Pfetsch and Franziska Schlösser and Felipe Serrano and Yuji Shinano and Christine Tawfik and Stefan Vigerske and Fabian Wegscheider and Dieter Weninger and Jakob Witzig, *The SCIP Optimization Suite 7.0*, Optimization Online, 2020.

[4] A. S. Hedayat, N. J. A. Sloane, and John Stufken, *Orthogonal arrays*, Springer Series in Statistics, Springer-Verlag, New York, 1999. Theory and applications; With a foreword by C. R. Rao. MR1693498

[5] Andrew Makhorin, *GNU Linear Programming Kit*, April 13, 2020.

[6] Eric D. Schoen, Pieter T. Eendebak, and Man V. M. Nguyen, *Complete enumeration of pure-level and mixed-level orthogonal arrays*, J. Combin. Des. **18** (2010), no. 2, 123–140, DOI 10.1002/jcd.20236. MR2604638

[7] ———, *Correction to: Complete enumeration of pure-level and mixed-level orthogonal arrays [MR2604638]*, J. Combin. Des. **18** (2010), no. 6, 488, DOI 10.1002/jcd.20270. MR2743138

[8] N. J. A. Sloane, *A Library of Orthogonal Arrays* (2020), `http://neilsloane.com/oadir/`. Accessed 2020.

[9] NSUCrypto2018, *Unsolved problems*: *Problem 2: Orthogonal Arrays*, 2018.

[10] The Sage Developers, *Sagemath, the Sage Mathematics Software System (Version 9.0)*, 2020. `https://www.sagemath.org`.

Nyilatkozat

Alulírott Kiss Rebeka kijelentem, hogy a szakdolgozatban foglaltak saját munkám eredményei, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel. A szakdolgozat a 2018-1.2.1-NKP-2018-00004 számú "IoT rendszerek biztonságát növelõ technológiák (SETIT)" projekt keretében, a Nemzeti Kutatási és Innovációs Alapból biztosított támogatással, a "Nemzeti Kiválósági Program: 2018-1.2.1-NKP" pályázati program finanszírozásában valósult meg. Tudomásul veszem, hogy szakdolgozatomat a Szegedi Tudományegyetem könyvtárában a kölcsönözhető könyvek között helyezik el, és az interneten is nyilvánosságra hozhatják.

Szeged, 2020. május 14.

.....

aláírás