Kölcsönösen torzítatlan bázisok 6 dimenzióban

Diplomamunka

Írta: Simon Győző

Alkalmazott matematikus szak

Témavezető:

Matolcsi Máté, tudományos főmunkatárs
MTA Rényi Alfréd Matematikai Kutató Intézet

Belső konzulens:

Keleti Tamás, egyetemi docens
Analízis Tanszék
Eötvös Loránd Tudományegyetem, Természettudományi Kar

Eötvös Loránd Tudományegyetem
Természettudományi Kar
2010

Tartalomjegyzék

Bevezető

A kölcsönösen torzítatlan bázisok fontos szerepet játszanak a kvantummechanika különböző alkalmazásaiban. Néhány speciális dimenzióban ismert, hogy maximálisan hány kölcsönösen torzítatlan bázis létezik, a legkisebb olyan dimenzió, amelyben nyitott a kérdés, a hatdimenziós eset. Ezzel a kérdéssel foglalkozik a dolgozat. A probléma azért különösen nehéz, mert a kölcsönösen torzítatlan bázisok szoros kapcsolatban állnak a hatdimenziós komplex Hadamard mátrixokkal, és nem ismert ezen mátrixok teljes klasszifikációja 6 dimenzióban. Figyelemreméltó numerikus eredmények arra utalnak, hogy a kölcsönösen torzítatlan bázisok maximális száma 3 hat dimenzióban. A dolgozat során bemutatásra kerül egy diszkretizációs eljárás, amely a közeljövőben elvezethet annak bizonyításához, hogy 4 kölcsönösen torzítatlan bázist már semmiképpen nem lehet összeállítani 6 dimenzióban. Az eljárás lényege dióhéjban annyi, hogy feltesszük, hogy létezik 4 kölcsönösen torzítatlan bázis, amiből következik bizonyos komplex Hadamard mátrix hármasok léte. Ezen Hadamard mátrixhármasok diszkretizáltjait próbáljuk meg megkeresni számítógépes programot segítségül hívva, és reményeink szerint arra fogunk jutni, hogy a diszkretizáltak nem léteznek, ami maga után vonja azt, hogy a komplex Hadamard mátrixok, és így a kölcsönösen torzítatlan bázisok sem léteznek.

Az én feladatom a kutatás során az volt, hogy a probléma megértése után bizonyos speciális vektorokon értelmezett gráfokban 6-klikkeket keressek számítógépes kódot írva, illetve a 6-klikkek megtalálása után belássam, hogy az adott 6-klikkből származó Hadamard mátrix nem lehet egy kölcsönösen torzítatlan bázisnégyesnek megfelelő Hadamard mátrixhármas tagja. A munkám során sikerült egy saját ötlettel a töredékére csökkenteni a megvizsgálandó esetek számát. A dolgozatban a probléma és a diszkretizációs eljárás ismertetése után bemutatom ezt az ötletet és azt, hogy ez hogyan segíthet a célunk elérésében.

Jelölések

$\langle .,. \rangle$	komplex skaláris szorzat
$*$	vagy komplex konjugálás vagy mátrix adjungálás
$\approx$	ekvivalencia
$\{.,.\}$	rendezetlen pár, illetve halmaz
$(.,.)$	rendezett pár

1. fejezet

Elméleti alapok

1.1. Alapfogalmak

1.1.1. Kölcsönösen torzítatlan bázisok

Szakdolgozatomban témavezetőm, Matolcsi Máté kölcsönösen torzítatlan bázisokkal foglalkozó kutatása kapcsán 2009 őszén elvégzett munkámat mutatom be. A kölcsönösen torzítatlan bázisok fogalma először Schwinger 1960-as kvantummechanikával foglalkozó munkájában [15] jelent meg. Ma már a kvantuminformáció-elmélet egyik alapvető fogalma, amely fontos szerepet játszik a kvantumtomográfiában, kvantumkriptográfiában és teleportálási sémákban [20].

Definiáljuk akkor tehát, hogy mit is értünk kölcsönösen torzítatlan bázisokon:

1.1.1. Definíció. *Vegyünk két $\mathbb{C}^d$-beli ortonormált bázist, legyen $\mathcal{A} = \{\mathbf{e_1}, \ldots, \mathbf{e_d}\}$ és $\mathcal{B} = \{\mathbf{f_1}, \ldots, \mathbf{f_d}\}$. Ekkor $\mathcal{A}$ és $\mathcal{B}$ torzítatlan bázisok, ha minden $1 \leq j, k \leq d$ -re $|\langle \mathbf{e_j}, \mathbf{f_k} \rangle| = \frac{1}{\sqrt{d}}$*

A $\mathcal{B}_0, \ldots, \mathcal{B}_m$ bázisok halmaza kölcsönösen torzítatlan, ha közülük bármely két bázis torzítatlan.

Ismert minden d esetén, hogy $\mathbb{C}^d$-ben maximum $d+1$ db kölcsönösen torzítatlan bázis létezhet (pl. [21, 1]). Továbbá ismert az is, hogy konstruálható $d+1$ ilyen bázis, ha a d dimenzió prím, vagy prímhatvány (pl. [11]). Az előbbieken túl csak nagyon kevés ismerettel rendelkezünk ebben a témában, mindazonáltal tudjuk még, hogy ha d prímfelbontása $d = p_1^{k_1} \ldots p_r^{k_r}$, és $p_j^{k_j}$ a legkisebb prímhatvány osztó, akkor létezik $\mathbb{C}^d$-ben legalább $p_j^{k_j} + 1$ kölcsönösen torzítatlan bázis. Így a legkisebb olyan d, amire nem ismert a kölcsönösen torzítatlan bázisok maximális száma a 6. Ezzel elérkeztünk a munkám során vizsgált kérdéshez:

1.1.2. Kérdés. *Maximum hány kölcsönösen torzítatlan bázist lehet megadni $\mathbb{C}^6$-ban?*

Bár az utóbbi pár évben sok figyelmet kapott ez a probléma, a fenti kérdés mégis nyitott maradt. Az előző bekezdésben ismertetett eredmények alapján láthatjuk, hogy mivel $6 = 2 \times 3$, ezért 3 kölcsönösen torzítatlan bázis létezik, és biztos, hogy nincs több, mint 7. Kísérleti numerikus eredmények [4, 5, 6, 19] arra utalnak, hogy nincs 3-nál több kölcsönösen torzítatlan bázis $\mathbb{C}^6$-ban, ezt sejtésként Zauner [19] fogalmazta meg először:

1.1.3. Sejtés. *6 dimenzióban a kölcsönösen torzítatlan bázisok maximális száma 3.*

1.1.2. Komplex Hadamard mátrixok

Annak, hogy ilyen hosszú idő alatt ilyen kevés eredmény került napvilágra, az egyik oka az, hogy a kölcsönösen torzítatlan bázisok a komplex Hadamard mátrixokkal állnak kapcsolatban és hat

dimenzióban nagyon nehéz leírni ezeket a mátrixokat.

A kapcsolat a komplex Hadamard mátrixok és a kölcsönösen torzítatlan bázisok között a következő: Legyenek $\mathcal{B}_0, \ldots, \mathcal{B}_m$ kölcsönösen torzítatlan bázisok. Azonosítsunk minden $\mathcal{B}_l = \{\mathbf{e_0^{(l)}}, \ldots, \mathbf{e_d^{(l)}}\}$ bázist az unitér $H_l = [\langle \mathbf{e_k^{(l)}}, \mathbf{e_j^0} \rangle]_{1 \leq j,k \leq d}$ mátrixszal, azaz a k-adik oszlop álljon $\mathcal{B}_l$ k-adik vektorának $\mathcal{B}_0$ bázisbeli koordinátáiból (A $\langle ., . \rangle$ jellel az egész dolgozatban a komplex skaláris szorzást jelöljük, amely az első változójában lineáris, míg a másodikban konjugált lineáris). Ezzel a konvencióval $H_0 = Id$, azaz az egységmátrix, míg az összes többi mátrix unitér és minden elemük abszolútértéke $\frac{1}{\sqrt{d}}$. Ebből következik, hogy a $\sqrt{d}H_i$ mátrixok minden eleme 1 abszolútértékű. Ez utóbbi fajta mátrixokat hívjuk komplex Hadamard mátrixoknak. Látható, hogy $\mathcal{B}_0, \ldots, \mathcal{B}_m$ kölcsönösen torzítatlan bázisok léte ekvivalens olyan $\sqrt{d}H_1, \ldots, \sqrt{d}H_m$ Hadamard mátrixok létével, melyekre igaz, hogy minden $1 \leq j \neq k \leq m$-re $\sqrt{d}H_j^* H_k$ is Hadamard mátrix. Ilyen tulajdonságú Hadamard mátrixokra azt mondjuk, hogy kölcsönösen torzítatlanok.

A továbbiakban a fenti H_i és $\sqrt{d}H_i$ mátrixokat megfeleltetjük egymásnak. A két mátrix között csak egy skalárszorzó ($\frac{1}{\sqrt{6}}$) a különbség, és H_i azzal a kellemes tulajdonsággal rendelkezik, hogy ortonormált, míg $\sqrt{d}H_i$ minden elemének 1 az abszolútértéke. Mindkét alak elő fog fordulni a későbbiek során, annak függvényében, hogy melyik változattal kényelmesebb az adott környezetben dolgozni, mindazonáltal innentől, hacsak direkt mást nem mondunk, akkor H_i jelölje az 1 abszolútértékű elemekkel rendelkező komplex Hadamard mátrixokat, azaz az Hadamard elnevezés erre a változatra van fenntartva. Azért tüntetjük ki ezt a változatot nagyobb figyelemmel, mert a későbbiekben komplex Hadamard mátrixokat fogunk majd keresni, azaz olyanokat, amelyeknél az elemek abszolútértéke 1. Az Hadamard mátrixoknál a két különböző mátrixból vett vektorok skaláris szorzata nem $\frac{1}{\sqrt{d}}$, hanem $\sqrt{d}$. A későbbiekben az Hadamard mátrixokra megfogalmazott állítások többsége igaz marad a nekik megfelelő ortonormált mátrixokra is, hiszen ezek nagyon hasonlítanak egymásra, ahol kifejezetten csak az egyik változatra igaz állítást fogalmazunk meg, ott azt külön is jelezzük. Az Hadamard mátrixok közötti torzítatlanság fenti definíciója például csak az Hadamard mátrixokra igaz így, az ortonormált megfelelőkre a definícióban minden $1 \leq j \neq k \leq m$-re $H_j^* H_k$-nak kell ortonormált megfelelőnek lennie (itt H_i kivételesen az ortonormált megfelelőre utal).

1.2. A komplex Hadamard mátrixok tulajdonságai

1.2.1. Kölcsönös torzítatlanság mátrixokra

Az előző fejezetben már definiáltuk Hadamard mátrixokra a kölcsönös torzítatlanság fogalmát. Természetesen adódik a kérdés, hogy adott H Hadamard mátrixhoz mindig létezik-e olyan G Hadamard mátrix, ami torzítatlan rá? Ha a kérdést ilyen általános formában tesszük fel, akkor a válasz nem, ugyanis az 1.2.3. szakaszban leírt $\mathcal{S}_6$ mátrixhoz például nem található ilyen G Hadamard mátrix. A kérdést úgy is feltehetjük kevésbé általános formában:

1.2.1. Kérdés. *Adott H Hadamard mátrixhoz létezik-e rá torzítatlan vektor, és ha igen, akkor mennyi?*

Nem definiáltuk még, hogy mit jelent vektorok és mátrixok torzítatlansága, ezzel, és a kérdéssel magával az 1.2.2. szakaszban foglalkozunk részletesen.

Az Hadamard mátrixok torzítatlanságával kapcsolatban léteznek olyan műveletek, amelyek invariánsak a torzítatlanságra. Legyenek $H_1, \ldots, H_m$ kölcsönösen torzítatlan Hadamard mátrixok, $D, D_1, \ldots, D_m$ unitér diagonális mátrixok valamint $P, P_1, \ldots, P_m$ permutációmátrixok. Ekkor a

$DPH_iP_iD_i$ mátrixok is Hadamard mátrixok és kölcsönösen torzítatlanok is maradnak. Az Hadamard mátrixhalmazok között ekvivalenciarelációt is definiálhatunk: $DPH_iP_iD_i$ relációban van H_i-vel. Az előbbi ekvivalenciarelációnak köszönhetően az általánosság megszorítása nélkül feltehetjük, hogy a H_1 mátrix első sorának és oszlopának minden eleme 1 az Hadamard mátrixoknál és $\frac{1}{\sqrt{6}}$ az ortonormált megfelelőnél. Ezen túlmenően azt is feltehetjük, hogy a sorok és az oszlopok lexikografikusan rendezve vannak a fellépő fázisok szerint (ennek pontos definícióját, illetve annak bizonyítását, hogy ez valóban feltehető, később, a 2.3. szakaszban adjuk meg).

Az iménti, mátrixhalmazokon értelmezett ekvivalenciarelációt értelmezhetjük mátrixokra is, mégpedig úgy, hogy két mátrix akkor áll relációban egymással, ha mint egyelemű mátrixhalmazok relációban állnak egymással a fenti értelemben.

1.2.2. Torzítatlanság vektorokra

Definiáljuk a bázisok közti torzítatlanság mintájára két normált vektor, illetve normált vektorok és egy normált vektor torzítatlanságát. Két d dimenziós normált vektor torzítatlan egymásra, ha komplex skalárszorzatuk abszolútértéke $\frac{1}{\sqrt{d}}$. Egy normált vektor torzítatlan egy vektorhalmazra (ezt a halmazt alkothatják például egy ortonormált bázis vektorai, vagy azok közül csak néhány), ha minden halmazbeli vektorra torzítatlan. A fenti definíciókat értelmezhetjük a normált vektorok $\sqrt{d}$-szereseire is (ilyen vektorok között fogunk majd keresni a későbbiek során), ekkor csak annyi változik, hogy a skalárszorzatnak $\sqrt{d}$-nek kell lennie minden esetben. Előfordulhat „vegyes" eset is, azaz normált vektor és nem normált vektor illetve vektorhalmaz torzítatlanságáról is beszélhetünk, itt mindig azt értjük torzítatlanságon, hogy ha normáljuk a megfelelő összeszorzandó vektorokat, akkor azok a fenti értelemben torzítatlanok lesznek egymásra.

A későbbiekben speciális vektorok között fogunk keresni, mégpedig d dimenzióban a következő alakban felírható vektorok között:

$$\mathbf{u} = \frac{1}{\sqrt{d}} \left(1, e^{2i\pi\phi_1}, \ldots, e^{2i\pi\phi_{d-1}}\right) \tag{1.2.1}$$

Könnyű látni, hogy ha $\mathbf{u}$ torzítatlan egy Hadamard mátrix első $d-1$ oszlopára, akkor automatikusan torzítatlan az utolsóra is. Így ha egy Hadamard mátrixra torzítatlan 1.2.1 alakú vektorokat keresünk, akkor van $d-1$ torzítatlansági kritériumunk és $d-1$ szabad ϕ_j paraméterünk. Emiatt általánosságban véges sok megoldásra számítunk. Mindazonáltal ismert egy speciális négydimenziós eset, amikor végtelen sok vektor bizonyul torzítatlannak egy adott Hadamard mátrixra, de nincs olyan ismert eset, ahol ne lenne legalább egy torzítatlan vektor.

Speciálisan a hatdimenziós esetben vizsgálva az 1.2.1. kérdést fontos látni, hogy ha egy adott Hadamard mátrixra kevesebb, mint 30 torzítatlan vektort találunk, akkor az $\{Id, H\}$ párnak megfelelő torzítatlan bázispár biztosan nem egészíthető ki teljes hételemű kölcsönösen torzítatlan bázishalmazzá, mert a nekik megfelelő H-n felüli 5 Hadamard mátrixhoz kellene még $5 \times 6 = 30$ vektor.

Könnyű megmutatni, hogy egy 1.2.1 formára hozott $\mathbf{u}$ vektor akkor és csak akkor torzítatlan Id egység- és H Hadamard mátrixokra, ha a

$$\mathcal{H}(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5) = \sum_{j=1}^{6} |\langle \mathbf{u}, \mathbf{h_j} \rangle| \tag{1.2.2}$$

függvény (ahol $\mathbf{h_j}$ jelöli H oszlopait) felveszi a globális maximumát a $(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5) \in \mathbb{T}^5$ pontban.

Ebből következően természetesnek tűnik torzítatlan **u** vektorokat keresni úgy, hogy választunk egy véletlenszerű $(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5)$ pontot a $\mathbb{T}^5$ paramétertérben, majd gradiensmódszerrel az 1.2.2-ben definiált $\mathcal{H}$ függvény lokális maximumait keressük. Számíthatunk arra, hogy ha elég sokszor futtatjuk a keresést ezzel a módszerrel, akkor megtaláljuk a torzítatlan **u** vektorok többségét, esetleg az összeset (közben persze előfordulhat, hogy más, kisebb lokális maximumokat találunk, ezeket a pontokat egyszerűen eldobjuk). Természetesen ezek az eredmények numerikus eredmények, ebben a formában inkább csak tájékoztató jellegűek, szükséges az eredmények szilárd elméleti alapokon nyugvó igazolása.

1.2.3. Ismert hatdimenziós Hadamard mátrixok

Az 1.2.1. szakaszban definiált ekvivalenciarelációt értelmezhetjük egyes mátrixok között is. Az így keletkező ekvivalenciaosztályokat keressük akkor, amikor egy adott dimenzióban le akarjuk írni az Hadamard mátrixokat. $d = 2$, 3 és 5 esetén az 1.2.4. fejezetben leírt Fourier mátrixok alkotják az egyetlen osztályt (azaz ezekben a dimenziókban minden komplex Hadamard mátrix ekvivalens a Fourier mátrixszal). $d = 4$ esetén létezik egy egyparaméteres ekvivalenciaosztály-család, amely magában foglalja a négydimenziós Fourier mátrixot és a valós Hadamard mátrixot is. Ezen a családon kívül nem létezik más ekvivalenciaosztály 4 dimenzióban. Érdekességként megjegyezzük, hogy folytonos egyparaméteres ekvivalenciaosztályok megjelennek 7 dimenzióban is, azaz ilyen ekvivalenciaosztályok léte nem függ a dimenzió prím voltától.

6 dimenzióban sajnos nem ismert az Hadamard mátrixok teljes osztályozása. Néhány család azonban ismert, ezeket kövér nyomtatott betűvel jelölöm az alábbi felsorolásban:

- Egy kétparaméteres $\mathbf{F}(x_1, x_2)$ család [7], köztük a Fourier mátrix, ami az $\mathbf{F}(0, 0)$ reprezentánsa.
- Egy ciklikus **C** mátrix [2].
- Egy egyparaméteres $\mathbf{D}(x)$ család [7], benne a $\mathbf{D} = D(0)$ mátrixszal, ami a negyedik egységgyökökből áll.
- Egy egyparaméteres $\mathbf{B}(\theta)$ család [14], ami önadjungált mátrixokból áll, és összeköti a **C** és **D** mátrixokat.
- Tao **S** mátrixa [18], ami a harmadik egységgyökökből áll.
- Egy szimmetrikus mátrixokból álló $\mathbf{M}(t)$ család [13], amely összeköti az $\mathbf{F}(0, 0)$ és **D** mátrixokat.
- Egy blokk-mátrixokból álló $\mathbf{X}(a, b)$ család [17].
- Egy kétparaméteres $\mathbf{H}(x_1, x_2)$ család [10], amely kiterjesztése az $\mathbf{M}(x)$ családnak.

1.2.4. Fourier családba tartozó hatdimenziós Hadamard mátrixok

A Fourier családba tartozó mátrixokkal kicsit külön is foglalkozunk, mert erre a családra számítógép segítségével bizonyított [9], hogy nincs olyan négyelemű kölcsönösen torzítatlan bázishalmaz, amelynél az 1.1.1 fejezet jelöléseivel $H_0 = Id$ és H_1 a Fourier családba tartozna. Ezzel lényegében az 1.1.3. sejtésre részleges bizonyításunk van, és az itt alkalmazott módszer javított változata meglátásunk szerint alkalmas lehet a sejtés teljes bizonyítására.

Ismerkedjünk meg a „Fourier családba” tartozó 6 dimenziós $F(x_1, x_2)$ mátrixokkal:

$$F(x_1,x_2)=\frac{1}{\sqrt{6}}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -q^2z_1 & q & -z_1 & q^2 & -qz_1 \\ 1 & qz_2 & q^2 & z_2 & q & q^2z_2 \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & q^2z_1 & q & z_1 & q^2 & qz_1 \\ 1 & -qz_2 & q^2 & -z_2 & q & -q^2z_2 \end{bmatrix},$$

ahol $z_1 = e^{2\pi i x_1}$, $z_2 = e^{2\pi i x_2}$ és $q = e^{2\pi i/3}$.

Ahogy azt korábban az 1.2.3. szakaszban már említettük, $F(0,0) = \mathcal{F}_6$ a Fourier mátrix. Így azt a feladatot, hogy keressünk torzítatlan vektorokat a standard bázisra és valamilyen $F(a,b)$ általánosított Fourier bázisra, tekinthetjük az ún. Pauli feladat egy általánosításának. Pauli eredeti kérdése az volt, hogy egy $L^2(\mathbb{R}^d)$-beli f függvényt egyértelműen meghatároz-e abszolútértékfüggvénye $|f|$ és Fourier transzformáltjának abszolútértékfüggvénye $|\hat{f}|$. A probléma diszkrét változata úgy fogalmazható meg, hogy 1 abszolútértékű komplex számok olyan véges sorozatát keressük, hogy azok Fourier transzformáltja is 1 abszolútértékű legyen (az ilyen sorozatokat biunimoduláris sorozatoknak nevezzük). Így a torzítatlan vektorok keresése lényegében a 6 dimenziós diszkrét változat egyfajta általánosításának tekinthető.

A $H = \mathcal{F}_6$ konkrét esetben ismert az az analitikus eredmény [3, 8], hogy pontosan 48 torzítatlan vektor létezik az $\{Id, \mathcal{F}_6\}$ párra az 1.2.2. szakaszban leírt normalizált formában. Ismert az is, hogy ebből a 48 vektorból 16 ortonormált bázis $(C_1, \ldots, C_{16})$ állítható össze, mindazonáltal nincs olyan (C_i, C_j) pár, melyek torzítatlanok lennének egymásra, azaz nincs olyan $\{Id, \mathcal{F}_6, C\}$ mátrixhármas, ami kiegészíthető lenne kölcsönösen torzítatlan mátrixnégyessé.

Érdekes kérdés, hogy mi a helyzet az 1.2.1. kérdéssel az általános $H = F(a,b)$ esetben. Azt gondolnánk, hogy általában 48-nál lényegesen kevesebb torzítatlan vektort találunk, és azokból csak kivételes esetben lehet C torzítatlan bázist összerakni. Ezek a várakozások azonban nem bizonyulnak megalapozottnak, numerikus eredmények ugyanis teljesen mást mutatnak. Ha ugyanis torzítatlan vektorokat keresünk néhány konkrét általános a, b-t választva az 1.2.2. szakaszban leírt módon az 1.2.2. kifejezés maximumát keresve, akkor azt találjuk, hogy minden általános (a,b) párra 48 olyan vektort találunk, ami torzítatlan az $\{Id, \mathcal{F}(a,b)\}$ mátrixpárra, és általános (a,b) párra 8 különböző ortonormált bázis $(C_1, \ldots, C_8)$ állítható össze belőlük. Kiderül továbbá az is, hogy néhány speciális (a,b) párra sokkal több ortonormált bázist lehet összeállítani, a fentiekben láttuk, hogy például $(a,b) = (0,0)$ esetén 16 ortonormált bázist kapunk, míg $(a,b) = (\frac{1}{6}, 0)$ esetén 70-et lehet összeállítani. Mindazonáltal az is igaz, hogy nincs olyan $\{Id, F(a,b), C\}$ kölcsönösen torzítatlan mátrixhármas, amely kiegészíthető lenne kölcsönösen torzítatlan mátrixnégyessé.

Az iménti numerikus eredmények többsége szigorú analitikus eredményekkel is megerősíthető.

Konstruálható továbbá analitikus alakban olyan kontinuum számosságú kölcsönösen torzítatlan mátrixhármas halmaz [9], amely hármasok egyik tagja az Id mátrix, egy másik pedig egy Fourier család-beli $F(a,b)$ mátrix, mégpedig olyan, hogy a=0, valamint b-re is teljesülnie kell néhány feltételnek. (Természetesen adódik a kérdés, hogy a mátrixhármas harmadik mátrixai milyen Hadamard mátrixok, nem alkotnak-e esetleg egy, az eddigiekben nem említett mátrixcsaládot. A válasz nem, mert megmutatható, hogy a harmadik mátrixok is az $F(a,b)$ családból kerülnek ki.)

A fentiek alapján (a numerikus eredményekről egy pillanatra elfeledkezve) azt gondolhatnánk, hogy ezen kontinuum számosságú kölcsönösen torzítatlan mátrixhármas halmaz létének bizonyítása egy nagy előrelépés egy kölcsönösen torzítatlan mátrixnégyes felé, de ez nem így van, mert

bebizonyítható analitikusan is a 4 bekezdéssel korábban említett numerikus eredmény [9]:

1.2.2. Tétel. *Az $(Id, F(a,b))$ kölcsönösen torzítatlan ortonormált mátrixpárok egyike sem egészíthető ki kölcsönösen torzítatlan mátrixnégyessé.*

Az így megfogalmazott eredmény és az ehhez vezető út jogos reményekkel tölthet el minket abban a tekintetben, hogy az alkalmazott módszer megfelelő lehet az 1.1.3. sejtés bizonyítására is.

1.3. A kölcsönösen torzítatlan bázisok tulajdonságai

1.3.1. Ekvivalenciareláció kölcsönösen torzítatlan bázisokon

Az 1.1. szakaszban már láttuk, hogy d dimenzióban maximálisan $d+1$ darab kölcsönösen torzítatlan bázis létezhet. Amennyiben ennyi létezik, akkor azt teljes kölcsönösen torzítatlan bázishalmaznak hívjuk. Azt is láttuk korábban, hogy a kölcsönösen torzítatlan bázisok léte hogyan függ kölcsönösen torzítatlan Hadamard mátrixok lététől, így kimondhatjuk, hogy akkor létezik teljes kölcsönösen torzítatlan bázishalmaz, ha létezik d darab kölcsönösen torzítatlan Hadamard mátrix.

Az 1.2.1. szakaszban már definiáltunk egy ekvivalenciarelációt Hadamard mátrixokon. Emlékeztetőül, H_1 és H_2 ekvivalensek, ha $H_2 = DPH_1P'D'$, ahol P, P' permutációmátrixok, és D, D' unitér diagonális mátrixok. Ezt a szokásos definíciót most kissé kiterjesztjük, annak segítségével, hogy bevezetünk egy ekvivalencia relációt a kölcsönösen torzítatlan bázispárok halmazán.

Jelölje (M_0, M_1) két torzítatlan bázis rendezett párját, míg $\{M_0, M_1\}$ a rendezetlen párjukat. A bázisokat alkotó vektorok sorrendje nem érdekes számunkra, mint ahogy két olyan bázist sem tekintünk különbözőnek, amelyben a vektorok csak fázisszorzókban térnek el. Ez a kétfajta azonosítás felel meg annak, hogy ha M_0 szerint koordinátázunk, akkor M_1 mátrixa valamely H_1 Hadamard mátrix teljes ekvivalencia osztályát végigfuthatja. Természetes azonban egy további azonosítás is: az $\{M_0, M_1\}$ párt azonosnak tekintjük minden $\{UM_0, UM_1\}$ párral, ahol U unitér transzformáció. Ez a mátrixok nyelvén azt jelenti, hogy az $\{\mathbb{1}, H\}$ pár (ahol H egy Hadamard mátrix) ekvivalens a $\{H^*, \mathbb{1}\}$ párral (a speciális $U = H^*$ választás miatt).

További nyilvánvaló észrevétel, hogy vektorok egy tetszőleges rendszerében az összes fellépő elem konjugálása nem változtatja meg a vektorok skalárszorzatának abszolút értékét, így speciálisan az ortogonalitást és torzítatlanságot sem. Ezért természetes H-t és konjugáltját $\overline{H}$-t is azonosítani.

Összefoglalva, a Hadamarad mátrixok szokásos (és az 1.2.1. szakaszban már definiált) ekvivalenciarelációját kiterjeszthetjük a következőképpen: H_1 és H_2 ekvivalensek, ha $H_2 = DPH_1^{(+)}P'D'$, ahol P, P' permutációmátrixok, és D, D' unitér diagonális mátrixok, és $(+)$ jelenthet konjugálást, adjungálást, transzponálást, vagy semmit.

1.4. Lemmák az ortogonalitás és a torzítatlanság vizsgálatára

A későbbi fejezetekben 1 abszolútértékű koordinátájú komplex vektorok diszkretizáltjainak ortogonalitását és torzítatlanságát fogjuk vizsgálni. Ebben nagy segítséget fog nyújtani az alábbi két lemma (lásd [9]), amely mindkettő arról szól, hogy ha két 1 abszolútértékű koordinátákkal rendelkező komplex vektor koordinátáit nem ismerjük pontosan, csak azt tudjuk róluk, hogy az egyes koordináták fázisa adott intervallumokba esik, akkor milyen esetekben tudjuk kizárni a vektorok ortogonalitását illetve torzítatlanságát.

1.4.1. Egyszerű hibabecslés

1.4.1. Lemma. *Vegyünk I_k és J_k $(1 \le k \le 5)$ $[0,1]$-beli zárt, akár elfajuló intervallumokat. Jelölje rendre L_k és T_k I_k és J_k hosszúságát, valamint m_k és s_k a középpontjukat.*

Legyen $S = \frac{1}{6}(1 + \sum_{k=1}^{5} e^{2i\pi(m_k - s_k)})$ a középpontok összege. Ekkor a két következő állítást fogalmazhatjuk meg:

- *ha kiválaszthatóak olyan $\phi_k \in I_k$ és $\psi_k \in J_k$ pontok, melyekre $|\frac{1}{6}(1+\sum_{k=1}^{5} e^{2i\pi(\phi_k - \psi_k)})| = \frac{1}{\sqrt{6}}$, akkor*

$$\frac{\pi}{6}\sum_{k=1}^{5}(L_k + T_k) \ge \left||S| - \frac{1}{\sqrt{6}}\right|, \tag{1.4.1}$$

- *ha kiválaszthatóak olyan $\phi_k \in I_k$ és $\psi_k \in J_k$ pontok, melyekre $|\frac{1}{6}(1 + \sum_{k=1}^{5} e^{2i\pi(\phi_k - \psi_k)})| = 0$, akkor*

$$\frac{\pi}{6}\sum_{k=1}^{5}(L_k + T_k) \ge |S|\,, \tag{1.4.2}$$

Bizonyítás Vezessük be az $E(\underline{x}, \underline{y}) = S - \frac{1}{6}(1 + \sum_{k=1}^{5} e^{2i\pi(x_k - y_k)})$ „hibafüggvényt", ahol x_k és y_k rendre I_k és J_k intervallumokban van. Könnyen látható, hogy minden $1 \le k \le 5$-re $|(m_k - s_k) - (x_k - y_k)| \le \frac{1}{2}(L_k + T_k)$. A triviális $|e^{2i\pi(m_k - s_k)} - e^{2i\pi(x_k - y_k)}| \le \pi(L_k + T_k)$ becslésből következik, hogy $|E(\underline{x}, \underline{y})| \le \frac{\pi}{6}\sum_{k=1}^{5}(L_k + T_k)$. Azaz az $\frac{1}{6}(1 + \sum_{k=1}^{5} e^{2i\pi(\phi_k - \psi_k)})$ függvény értékei egy $\frac{\pi}{6}\sum_{k=1}^{5}(L_k + T_k)$ sugarú, S középpontú körön belül találhatóak. Így az első állítás feltételeiből azt a következtetést vonhatjuk le, hogy S távolsága az origó középpontú, $\frac{1}{\sqrt{6}}$ sugarú körvonaltól nem nagyobb, mint $\frac{\pi}{6}\sum_{k=1}^{5}(L_k + T_k)$, míg a második állításnál az következik, hogy S távolsága az origótól nem nagyobb, mint $\frac{\pi}{6}\sum_{k=1}^{5}(L_k + T_k)$. Ezt kellett bizonyítanunk. □

1.4.2. Pontosabb hibabecslés

Az iménti lemma nem túl erős olyan értelemben, hogy nem foglalkozik azzal, hogy az x_k és y_k értékek a nekik megfelelő intervallumok melyik részén helyezkednek el. Az alábbi lemma lényegében azt az állítást fogalmazza meg, hogy bizonyos feltételek teljesülése esetén a skaláris szorzat a szélsőértékét akkor veszi fel, amikor az egyes koordináták felveszik szélsőértéküket.

1.4.2. Lemma. *Vegyünk I_k és J_k $(1 \le k \le 5)$ $[0,1]$-beli zárt, akár elfajuló intervallumokat. Jelölje rendre L_k és T_k I_k és J_k hosszúságát, m_k és s_k a középpontjukat, valamint rendre i_k^-, i_k^+ és j_k^-, j_k^+ az alsó és felső végpontjukat. Tegyük fel továbbá, hogy $\frac{1}{6}\sum_{k=1}^{5}(L_k + T_k) < \frac{1}{\pi}$.*

Legyen $S = \frac{1}{6}(1 + \sum_{k=1}^{5} e^{2i\pi(m_k - s_k)})$ a középpontok összege, továbbá vegyük mind a 32 végponti összeget: $S_{\underline{\epsilon}} = \frac{1}{6}(1 + \sum_{k=1}^{5} e^{2i\pi(i_k^{\epsilon_k} - j_k^{-\epsilon_k})})$ (itt $\underline{\epsilon}$ bármilyen $\pm$ elemekből álló vektor jelenthet; vegyük észre, hogy $-\epsilon_k$ található j_k felsőindexében). Ekkor a két következő állítást fogalmazhatjuk meg:

- *ha kiválaszthatóak olyan $\phi_k \in I_k$ és $\psi_k \in J_k$ pontok, melyekre $|\frac{1}{6}(1+\sum_{k=1}^{5} e^{2i\pi(\phi_k - \psi_k)})| = \frac{1}{\sqrt{6}}$, akkor*

$$\max\{|S - S_{\underline{\epsilon}}|\} \ge ||S| - \frac{1}{\sqrt{6}}|, \tag{1.4.3}$$

- *ha kiválaszthatóak olyan $\phi_k \in I_k$ és $\psi_k \in J_k$ pontok, melyekre $|\frac{1}{6}(1 + \sum_{k=1}^{5} e^{2i\pi(\phi_k - \psi_k)})| = 0$, akkor*

$$\max\{|S - S_{\underline{\epsilon}}|\} \ge |S|. \tag{1.4.4}$$

Bizonyítás Legyen $r = \max\{|S-S_\varepsilon|\}$, és vezessük be újra az $E(\underline{x},\underline{y}) = S-\frac{1}{6}(1+\sum_{k=1}^{5} e^{2i\pi(x_k-y_k)})$ „hibafüggvényt", ahol x_k és y_k rendre I_k és J_k intervallumokban van. Az előző lemma bizonyításánál látottakkal egyező módon könnyen látható, hogy minden $1 \le k \le 5$-re $|(m_k-s_k)-(x_k-y_k)| \le \frac{1}{2}(L_k+T_k)$. A triviális $|e^{2i\pi(m_k-s_k)}-e^{2i\pi(x_k-y_k)}| \le \pi(L_k+T_k)$ becslésből következik, hogy $|E(\underline{x},\underline{y})| \le \frac{\pi}{6}\sum_{k=1}^{5}(L_k+T_k)$. Ebből a $\frac{1}{6}\sum_{k=1}^{5}(L_k+T_k) < \frac{1}{\pi}$ feltételünket felhasználva következik, hogy $|E(\underline{x},\underline{y})| < \frac{1}{6}$. Mivel $|E(\underline{x},\underline{y})|$ egy kompakt tartományon értelmezett folytonos függvény, igaz rá, hogy fel is veszi a szuprémumát. Azt állítjuk, hogy ez a maximum akkor vétetik fel, amikor az összes x_k és y_k I_k és J_k ellentétes végpontjai (például ha x_k I_k alsó végpontja, akkor y_k J_k felső végpontja).

Indirekt bizonyítást alkalmazunk, tegyük fel, hogy az előbbi állítás nem teljesül, mondjuk x_1-re és y_1-re. Ekkor $x_1 - y_1$ az $I_1 - J_1$ intervallum belsejében van. Ez azt jelenti, hogy elég kicsi t-re mozgathatjuk úgy x_1-et és/vagy y_1-et x_1'-be, illetve y_1'-be, hogy egyrészt továbbra is rendre I_1 és J_1 intervallumok belsejében maradjanak, másrészt $x_1' - y_1' = x_1 - y_1 + t$. Ebből következik, hogy

$$|E(\underline{x}',\underline{y}')| = |S - \frac{1}{6} - \frac{1}{6}e^{2i\pi(x_1-y_1+t)} - \frac{1}{6}\sum_{k=2}^{5} e^{2i\pi(x_k-y_k)}| = \tag{1.4.5}$$

$$= |E(\underline{x},\underline{y}) + \frac{1}{6}(1-e^{2ti\pi})e^{2i\pi(x_1-y_1)}|. \tag{1.4.6}$$

Ha t-t a nulla egy kis környezetében változtatjuk, akkor az $E(\underline{x},\underline{y})+\frac{1}{6}(e^{2ti\pi}-1)e^{2i\pi(x_1-y_1)}$ függvény pontjai egy kis köríven mozognak, amelynek a sugara $\frac{1}{6}$, a középpontja pedig $E(\underline{x},\underline{y})-\frac{1}{6}e^{2i\pi(x_1-y_1)}$. Ez az ív $t = 0$-nál átmegy $|E(\underline{x},\underline{y})|$-en. Ebből, és abból, hogy $|E(\underline{x},\underline{y})| < \frac{1}{6}$ egyszerű síkgeometriai megfontolásokkal kijön, hogy ezen a kis íven létezik olyan pont, hogy $|E(\underline{x}',\underline{y}')|$ nagyobb, mint $|E(\underline{x},\underline{y})|$. Ezt az érvelést el lehet mondani minden x_k, y_k-ra, így következik, hogy $|E(\underline{x},\underline{y})|$ tényleg akkor veszi fel a maximumát, amikor minden x_k, y_k az I_k és J_k intervallumok átellenes végpontjaiban vannak. Ez azt jelenti, hogy r $|E(\underline{x},\underline{y})|$ maximuma.

Az iméntiekből következik, hogy a $\frac{\pi}{6}(1+\sum_{k=1}^{5} e^{2i\pi(\phi_k-\psi_k)})$ függvény értékei egy r sugarú, S középpontú körön belül találhatóak. Így az első állítás feltételeiből azt a következtetést vonhatjuk le, hogy S távolsága az origó középpontú, $\frac{1}{\sqrt{6}}$ sugarú körvonaltól nem nagyobb, mint r, míg a második állításnál az következik, hogy S távolsága az origótól nem nagyobb, mint r. Ezt kellett bizonyítanunk. □

1.5. A dolgozat felépítése

Ebben a fejezetben összefoglaltam a problémához kapcsolódó elméleti és numerikus eredményeket, valamint közöltem két a későbbiekhez fontos lemmát a bizonyításukkal.

A második fejezetben részletesen ismertetek egy olyan diszkretizációs eljárást, amely a közeljövőben elvezethet a probléma megoldásához, illetve bemutatom, hogy egy speciális esetben, a Fourier családba tartozó Hadamard mátrixoknál hogyan vezetett eredményre egy hasonló gondolatmenet.

A harmadik fejezetben a saját munkám eredményeit ismertetem, ami egyrészt a diszkretizációs eljárás egyes elemeinek számítógépes implementációjából, azok sebességének optimalizációjából állt, másrészt egy saját ötlettel sikerült elérni, hogy a teljes futási idő akár nagyságrendekkel is kisebb lehet.

A negyedik fejezetben áttekintést adok a kutatás jelenlegi állásáról, valamint arról, hogy az ötletem alkalmazása hol gyorsíthat még az algoritmuson, illetve milyen egyéb lehetőségeket látunk a módszerünk javítására, amelyek elvezethetnek a probléma megoldásához.

2. fejezet

A probléma diszkretizált változata és egy lehetséges bizonyítási mód

2.1. Alapelvek

Az előző fejezetben megismerhettünk néhány elméleti és numerikus eredményt az 1.1.3. sejtéssel kapcsolatban. A numerikus eredmények sajátja, hogy csak tájékoztató jellegűek lehetnek, bármennyire sok eredményt gyűjtenénk is össze, azok önmagukban így nem alkalmasak a sejtés bizonyítására. Az elméleti eredmények tekintetében viszont még nagyon távol állunk attól, hogy a sejtést tisztán elméleti úton is belássuk. Így egy „hibrid" megoldással próbálkozunk, amely magában hordozza az elméleti megoldás szigorú precizitását és kihasználja a numerikus eredményekben rejlő perspektívákat.

Az alapgondolat az, hogy diszkretizáljuk a problémát, ezáltal véges sok (mindazonáltal nagyon sok) eset vizsgálatát kell végrehajtanunk, viszont ehhez a vizsgálathoz már igénybe tudjuk venni a számítógépek segítségét. Ez a fajta megközelítés nem példa nélküli matematikai tételek bizonyításánál, gondoljunk csak a Négyszíntétel bizonyítására, illetve például azt, hogy nem létezik 10 rendű véges projektív sík szintén kimerítő számítógépes kereséssel bizonyították be [12].

Az egyes esetek vizsgálata azt jelenti, hogy eldöntjük, hogy a vizsgált esetben előfordulhat-e, hogy létezik kölcsönösen torzítatlan mátrixnégyes. A módszert úgy konstruáltuk meg, hogy ha a keresés eredménye negatív, akkor nem létezhet ilyen mátrixnégyes, míg a keresés pozitív kimeneteléből még nem következik a mátrixnégyes léte, mivel nem a sejtés cáfolatát, hanem a bizonyítását szeretnénk megtalálni. A módszer egyes elemeinél is mindig figyelembe vesszük az iménti gondolatot, azaz amikor kénytelenek vagyunk valami miatt a pontosságból engedni, akkor mindig olyan irányba „tévedünk", hogy inkább azt engedjük meg, hogy olyan esetek is téves pozitív jelzést adjanak, amelyek valójában mégsem pozitívak, mint hogy olyan esetet is negatív jelzésnek vegyünk, ami valójában tartalmaz kölcsönösen torzítatlan mátrixnégyest.

A diszkretizációt úgy konstruáltuk meg, hogy lehessen változtatni a „finomságán". Ez fontos, mert a kísérletek mutatták, hogy nem minden finomságú diszkretizáció kecsegtet a sikerrel. Túl „finom" diszkretizációval valószínűleg be lehet ugyan bizonyítani a sejtést, de a futáshoz szükséges idő még klasztereken is olyan hosszúra nyúlhat (akár 100 év), hogy az elméletileg tökéletesnek tűnő megoldás a gyakorlatban nem lesz megvalósítható. A másik oldalról viszont ha túl „durva" diszkretizációt választunk, akkor ugyan lefuthat a keresés „emberi időben", de előfordulhat, hogy „talál" ilyen négyest. Az előző bekezdés értelmében ez nem jelenti a sejtés cáfolatát, csak a bizonyításra

irányuló kísérlet adott paraméterek melletti kudarcát.

Egy eset vizsgálata lényegében abból áll, hogy az 1.1. szakaszban látottak alapján a kölcsönösen torzítatlan bázisok azonosíthatóak kölcsönösen torzítatlan komplex Hadamard mátrixokkal, és ezeket a mátrixokat próbáljuk meg megkeresni. Az Id mátrixhoz megkeresünk egy lehetséges H_1 mátrixot, ez tekinthető egy esetnek az előző bekezdések értelmében. Az eset vizsgálata egyszerűen azt jelenti, hogy megpróbáljuk összeállítani H_2 és H_3 mátrixokat. Abban bízunk, hogy minden esetben arra jutunk, hogy ez nem lehetséges. A bizonyítás elméleti háttere az, hogy ha létezik kölcsönösen torzítatlan bázisnégyes, akkor létezik hozzá olyan $\{Id, H_1, H_2, H_3\}$ mátrixnégyes, hogy a H_i-k a bázisnégyesnek megfelelő kölcsönösen torzítatlan komplex Hadamard mátrixok, ebből következően létezik diszkretizált megfelelőjük is. Ha nem találunk olyan diszkretizált esetet, amelyben létezhetne ilyen mátrixnégyes, azaz semelyik H_1-hez nem tudunk összeállítani H_2-t és H_3-at egyszerre, akkor ellentmondásra jutunk az eredeti feltevésünkkel, miszerint létezik kölcsönösen torzítatlan bázisnégyes, így az az állítás hamis.

2.2. A diszkretizáció leírása

A számítógépes keresés során programozástechnikai okokból kényelmesebb, ha úgy tekintjük, hogy a mátrix sorait alkotják egy bázis vektorai, amikor egy bázisból mátrixot képezünk, így ezen túl már a mátrix sorait tekintjük bázisvektoroknak. A diszkretizáció során kényelmesebb a komplex Hadamard mátrixok között keresni, így innentől a mátrixok és vektorok elemei 1 abszolútértékűek (nem pedig $\frac{1}{\sqrt{6}}$), és a torzítatlanság teljesítéséhez a skaláris szorzat abszolútértékének $\sqrt{6}$-nak kell lennie (erről a kérdésről bővebben az 1.2.2. szakaszban olvashattunk).

A fentiek alapján tehát olyan Hadamard mátrixokat keresünk, amelyeknek minden eleme 1 abszolútértékű, és bárhogyan is választunk egy-egy sorvektort két különböző mátrixból a skaláris szorzatuk abszolútértéke mindig $\sqrt{6}$ kell, hogy legyen. Az 1.2.1. szakaszban már láttuk, hogy az általánosság megszorítása nélkül feltehetjük, hogy az első sorban és oszlopban csupa egyes szerepel, amikor az első mátrixot, H_1-et akarjuk rögzíteni, míg H_2 és H_3 esetében a permutációkkal és skalárral való szorzásokkal azt tudjuk elérni, hogy minden első oszlopbeli elem pontosan 1. Azaz a H_1-et alkotó bázis első vektora csupa 1-es, és mindhárom mátrixban az összes bázisvektor 1-essel kezdődik. Mivel mindhárom mátrixban az összes elem 1 abszolútértékű komplex szám, felírhatóak $e^{2\pi i\rho}$ alakban, ahol $\rho \in [0, 1)$. Diszkretizáljunk a következőféleképpen: válasszunk egy N pozitív egészet (N az első diszkretizációs paraméter), és osszuk fel a $[0, 1)$ intervallumot N db egyforma hosszú $I_0^{(N)}, I_1^{(N)}, \ldots, I_{N-1}^{(N)}$ kisebb intervallumra, azaz legyen $I_j^{(N)} = [j/N, (j+1)/N)$ (Más felosztás is elképzelhető, de programozási szempontból ez tűnik a legkényelmesebbnek). Egy $e^{2\pi i\rho}$ alakú H_i-beli mátrixelemet reprezentáljunk azzal a j egésszel, amelyre $\rho \in I_j^{(N)}$ $(0 \leq j \leq N-1)$. Ez azt jelenti, hogy akármelyik diszkretizált mátrixban ha látunk egy j elemet, akkor annyit tudunk arról az elemről, hogy az eredeti ρ fázisnak az $I_j^{(N)}$ intervallumba kell esnie. Az elemről pontosan ennyi információnk van, nem tudunk róla se többet, se kevesebbet. A mátrixok első oszlopában szereplő pontosan 1 értékeket reprezentáljuk 0-val, de tartsuk észben, hogy ezek a nullák nem ugyanazt jelentik, mintha a mátrix belsejében fordulnának elő, hiszen itt azt jelentik, hogy a fázis pontosan nulla, míg a mátrix belsejében azt, hogy az adott elem $I_0^{(N)}$-be esik (természetesen H_1 első sorában is „pontos” nullák szerepelnek).

A fentiek másképp megfogalmazva azt jelentik, hogy a következő alakú sorvektorokkal foglalkozunk:

$$u = (0, j_1, j_2, j_3, j_4, j_5) \tag{2.2.1}$$

ahol $0 \leq j_k \leq N-1$ és az első koordinátában található 0 azt jelenti, hogy ott pontosan 1 szerepel, míg j_k jelentése az, hogy az adott ρ_k elem az $I_{j_k}^{(N)}$-be esik. Míg az eredeti, nem diszkretizált mátrixot H_i-vel jelöljük, addig a diszkretizált változatot jelölje $\hat{H}_i$. H_i elemeit jelölje $\rho_{m,k}$, $\hat{H}_i$ elemeit pedig $j_{m,k}$.

A 2.2.1. alakú vektorokból N^5 különböző van. Ezen vektorok között adódik egy természetes rendezés is, mégpedig úgy, hogy $u \leq v$ akkor és csak akkor, ha a lexikografikus sorrendjük is ugyanez. Az így definiált rendezést fogjuk használni.

2.3. Diszkretizált $\hat{H}_1$ Hadamard mátrix keresése

2.3.1. Definíció. *Legyen adott egy olyan egész elemű $\hat{H}_1$, amelynek első sorának és oszlopának minden eleme 0, a többi elem pedig mind 0 és $N-1$ közé esik egy adott N diszkretizációs paraméterre, valamint jelölje a mátrix elemeit $j_{m,k}$. Azt mondjuk, hogy $\hat{H}_1$ egy komplex Hadamard mátrix N-diszkretizált reprezentánsa, ha létezik olyan H_1 komplex Hadamard mátrix, hogy annak $\rho_{m,k}$ elemeire teljesül, hogy $\rho_{m,k} \in I_{j_{m,k}}^{(N)}$. Jelöléssel: $\hat{H}_1 \in HAD_N$, ahol HAD_N jelöli az N-diszkretizált komplex Hadamard mátrix-reprezentánsok halmazát.*

Ebben a szakaszban ismertetünk egy olyan algoritmust, amellyel hatékonyan meg lehet találni az összes HAD_N-beli $\hat{H}_1$ mátrixot. Jelentős numerikus eredmények [16] támasztják alá azt a sejtést, miszerint a 6×6-os komplex Hadamard mátrixok tartománya 4 dimenziós. Emiatt arra számíthatunk, hogy a HAD_N halmaz számossága megközelítőleg cN^4 lesz valamely alkalmas c konstanssal. Mindenesetre az összes HAD_N-beli elem megtalálása ijesztőnek tűnhet első pillantásra, hiszen a 25 szabad paraméter miatt N^{25} N-diszkretizált mátrixból kellene kiválasztani a HAD_N-belieket, és N^{25} már viszonylag kis számokra is csillagászati nagyságú. Mindazonáltal látni fogjuk, hogy megfelelően ügyes algoritmussal megoldható ez a feladat.

Van egy-két tulajdonság, amit feltehetünk $\hat{H}_1$-ról. Azt már korábban láttuk, hogy az első sor és oszlop minden eleme 0, és az 1.2.1. szakaszban már utaltunk rá, hogy az is elérhető sorok és oszlopok cserélgetésével, hogy mind a sorok, mind az oszlopok lexikografikusan rendezve legyenek. (Ez egyébként nem teljesen triviális, mert a sorok átrendezése elronthatja az oszlopok rendezettségét és viszont. Mindazonáltal, ha kiírjuk egymás után sorfolytonosan a mátrix elemeit, mind a 36-ot, akkor ez a vektor minden ilyen átrendezésnél lexikografikusan kisebb vektorrá válik, függetlenül attól, hogy éppen a sorokat vagy az oszlopokat rendeztük át. Emiatt nem lehet olyan végtelen átrendezés sorozatot találni, amelynek minden eleme olyan átrendezés, amely lexikografikus sorrendbe teszi a sorokat vagy az oszlopokat, tehát minden ilyen sorozat véges, azaz folyamatosan ilyen átrendezéseket végrehajtva egyszer elérünk egy olyan állapotba, amelyben már nem lehet végrehajtani több ilyen átrendezést, azaz mind a sorok, mind az oszlopok lexikografikusan rendezve lesznek.) Az első sor és oszlop csupa 0 volta miatt ez a rendezettség automatikusan biztosítja, hogy a második sor és oszlop elemei monoton növőek legyenek. Ez egy nagyon hasznos tulajdonság, mert nagyban lecsökkenti a szóbajöhető mátrixok számát. Feltehetjük továbbá, hogy a második sor lexikografikusan kisebb vagy egyenlő, mint a második oszlop (ezt $\hat{H}_1$ transzponálásával érhetjük el, amennyiben szükséges).

A szóbajöhető mátrixokban továbbá a sor- és oszlopvektorok valamilyen diszkretizált értelemben ortogonálisak egymásra. Ehhez meg kell fogalmaznunk egy olyan feltételt két diszkretizált vektorra, hogy amennyiben léteznek olyan vektorok, amelyek „benne vannak” a diszkretizált vektorok jelentette tartományokban (egy-egy vektor mindkét diszkretizált vektorban), és azok ortogonálisak egymásra, akkor biztosan teljesítsék a feltételünket. A későbbiekben mind a feltételt, mind a tőle

elvárt tulajdonságot formalizáljuk. Fontos látni, hogy a $\hat{H}_1$ mátrix első sora és oszlopa csupa 0, ráadásul ez pontosan 1-et jelent a megfelelő koordinátákban. Ebből következően csak 5 ismeretlen oszlopunk és sorunk van, ráadásul azok első koordinátája is ismert, továbbá a 2.2.1. alakban felírhatóak. Egy fontos, sok mátrixjelöltet elvető szűrőfeltétel lehet az, hogy minden sor- és oszlopvektornak ortogonálisnak kell lennie az $(1,1,1,1,1,1)$ vektorra (itt most a komplex 1-es szerepel, nem a diszkretizált értelemben vett.) Definiáljunk egy a fentieknek megfelelő feltételt:

2.3.2. Definíció. *Egy 2.2.1. alakú vektor ORT_N-beli, ha léteznek olyan $\phi_k \in I_{j_k}$ számok, melyekre $1 + \sum_{k=1}^{5} e^{2i\pi\phi_k} = 0$.*

Ez az ORT_N halmaz az összes 2.2.1 alakú vektor egy „kis" részhalmazát alkotja, azokat, amelyek mint tartomány, tartalmaznak olyan komplex vektort, amely ortogonális az $(1,1,1,1,1,1)$ vektorra. Nyilván minden sor- és oszlopvektor ORT_N-beli, így nagyon fontos, hogy minél pontosabban meg tudjuk határozni az ORT_N halmazt, azaz minél kevesebb vektor legyen benne.

Egy 2.2.1 alakú vektor ORT_N-beliségének eldöntésében az 1.4. szakaszban szereplő lemmák segíthetnek, mégpedig az a részük, ami a vektorok origótól vett távolságára fogalmaz meg állítást. Legyen $u = (0, j_1, j_2, j_3, j_4, j_5)$ és jelölje r_{j_k} az I_{j_k} intervallumok középpontját (az egyszerűbb jelölés kedvéért az N felsőindexet töröltük). Az 1.4.1. lemmából a következő állítás adódik: ha $u \in ORT_N$, akkor

$$|1 + \sum_{k=1}^{5} e^{2i\pi r_{j_k}}| \leq \pi \sum_{k=1} 5\frac{1}{N} = \frac{5\pi}{N}. \tag{2.3.1}$$

Ez a feltétel még nem elég erős olyan értelemben, hogy ha ezzel vizsgálnánk egy vektor ORT_N-beliségét, akkor sok olyan vektor átmenne ezen a szűrőn, ami valójában nem is ORT_N-beli. Ezt kivédendő megvizsgáljuk a vektor „leszármazottait"- Ez precízen a következőt jelenti: ha egy u vektorhoz léteznek a 2.3.2. definícióban meghatározott tulajdonságú ϕ_k számok, akkor minden I_{j_k} intervallumra a ϕ_k számok vagy az I_{j_k} bal, vagy a jobb felében vannak. Összesen 32 különböző lehetőség van aszerint, hogy az összes I_{j_k} intervallumnál melyik félintervallumban van ϕ_k. Ezt a 32 különböző lehetőséget, pontosabban az általuk generált 32 olyan vektort, ahol a koordinátákról már azt tudjuk, hogy egy $\frac{1}{2N}$ hosszúságú intervallumba esnek, nevezzük u „gyerekeinek". Nyilvánvaló, hogy ahhoz, hogy u ORT_N-beli legyen, legalább egy gyerekének teljesítenie kell a 2.3.1.feltételt, csak immár $\frac{5\pi}{2N}$-nél kell kisebbnek lennie az új, kisebb intervallumok középpontösszegének. Ha egyetlen „gyerek" sem teljesíti ezt a jóval szigorúbb feltételt, akkor u biztosan nem ORT_N-beli, további vizsgálata szükségtelen. Ezt a vizsgálatot ráadásul lehet folytatni tovább a feltételt teljesítő („életben maradó") „gyerekek" „gyerekeivel", akár 8-10 „generáción" át, amiknek egyre szigorúbb feltételnek kell megfelelnie. Értelemszerűen egy u vektor akkor „marad életben", ha legalább egy leszármazottja életben marad minden generációban.

Megjegyzések:

2.3.3. Megjegyzés. *Az ORT_N halmaz nyilván invariáns a vektorok utolsó 5 koordinátájának $(j_1, j_2, j_3, j_4, j_5)$ permutálására nézve. Így érdemes bevezetni az $ORT_{N,mon}$ halmazt, amely olyan ORT_N-beli vektorokból áll, amelyek koordinátái monoton növő sorrendben vannak. Ezzel aztán számítási időt is spórolhatunk, mégpedig úgy, hogy úgy keressük ORT_N elemeit, hogy csak a monoton növő koordinátájú vektorokat vizsgáljuk meg az előbbi módszerrel, és ezzel előállítjuk az $ORT_{N,mon}$ halmazt, majd ennek vektorai utolsó 5 koordinátájának permutálásával kaphatjuk meg ORT_N halmazt.*

2.3.4. Megjegyzés. *A tényleges megvalósítás során kitűnt, hogy míg az első gyerekek vizsgálata még felezi a potenciális ORT_N-beli elemek számát, majd később a generációk számának növekedé-*

sével a kiszűrt vektorok száma exponenciálisan csökken, addig a futtatáshoz szükséges idő, főleg az „életben maradó" vektoroknál, exponenciálisan nő. Megítélésünk szerint valahol 5 és 10 generáció között lehet az optimális mélység. Az 1.4.2. lemmát is kipróbáltuk az ORT_N-beli elemek szűréséhez, és azt tapasztaltuk, hogy kicsit lassítja futást (azaz az adott algoritmus tovább fut), de érdemesnek tűnik mindkét lemmából következő feltételt ellenőrizni.

2.3.5. Megjegyzés. *Különböző N-ekre futtattuk a megírt számítógépes kódot, és azt találtuk, hogy például N=17-re $|ORT_N|$=58450, N=19-re $|ORT_N|$=82630 és N=53-ra $|ORT_N|$=1875110. Mivel az összes lehetséges vektorból N^5 db van, kiszámítható, hogy N növekedésével az ORT_N-beli vektorok aránya csökken az összes lehetséges vektorhoz viszonyítva. Ez megfelel annak a várakozásnak, hogy N növelése javítja a pontosságot. További tapasztalat, hogy ORT_N mérete a fenti trendből kiugróan nagy is lehet, ha N osztható 2-vel vagy 3-mal, mivel a -1 illetve a 3. egységgyökök így intervallumhatárra esnek, és mivel ezek sok valódi ortogonális vektorban előfordulnak, két vagy még több diszkretizált vektort ORT_N-belivé tesznek. Az előbbiek miatt csak olyan N számokat vizsgáltunk, amelyek nem oszthatóak sem 2-vel, sem 3-mal (ezek ilyen kis számoknál majdnem megegyeznek a prímekkel).*

2.3.6. Megjegyzés. *N optimális megválasztása nagyon fontos a kutatás sikere szempontjából. Egyrészt nyilván, ha N túl kicsi, akkor túl gyengék a feltételeink. Túl sok helyen nem szűrünk ki olyan eseteket, amik a valóságban kiszűrhetőek lennének. Nem sikerül ellentmondásra jutnunk, mert sikerül összeállítani megfelelő mennyiségű komplex Hadamard mátrixot. Másrészt, ha N túl nagy, akkor mind ORT_N, mind a megfelelő HAD_N mérete olyan nagy lehet, ami már nem kezelhető számítógéppel sem. A kutatás jelenlegi állása szerint $N \approx 80$ egy jó választás lehet.*

Térjünk vissza $\hat{H}_1$ mátrix előállításához. Láttuk, hogy minden sor és oszlop ORT_N-beli, és páronként N-ortogonálisnak kell lenniük a következő értelemben:

2.3.7. Definíció. *Azt mondjuk, hogy az $u = (0, j_1, j_2, j_3, j_4, j_5)$ és $v = (0, m_1, m_2, m_3, m_4, m_5)$ vektorok N-ortogonálisak egymásra, ha léteznek olyan $\phi_k \in I_{j_k}$ és $\psi_k \in I_{m_k}$ számok, hogy $1 + \sum_{k=1}^{5} e^{2i\pi(\phi_k - \psi_k)} = 0$.*

Ez a tulajdonság nyilván eltolásinvariáns abban az értelemben, hogy csak a $(j_1 - m_1, \ldots, j_5 - m_5)$ modulo N értékektől függ. Emiatt megtehetjük, hogy $m_1 = \cdots = m_5 = 0$, ezzel $v_0 = (0, 0, 0, 0, 0, 0)$ esetet választjuk (itt természetesen az első 0 a pontos nullát jelenti, míg az utolsó öt koordináta 0-ja az I_0 intervallumot jelöli), és ezzel definiáljuk az $ORT_{N,eps}$ halmazt. Azaz egy 2.2.1. alakú vektor $ORT_{N,eps}$-beli, ha az iménti v_0 vektorra N-ortogonális. (Az $ORT_{N,eps}$ jelölésben az *eps* azt jelenti, hogy v_0 vektort nem ismerjük pontosan, csak „epszilonnyi" hibával, mert az utolsó 5 koordinátáról csak azt tudjuk, hogy I_0-beliek, nem pontosan nullák.) Az előbbi jelöléssel úgy is megfogalmazhatjuk az eltolásinvarianciát, hogy u és v akkor és csak akkor N-ortogonálisak, ha a $(0, j_1 - m_1, \ldots, j_5 - m_5)$ modulo N vektor $ORT_{N,eps}$-beli.

Miután az ORT_N halmazt már megkonstruáltuk az imént, egyszerű előállítani az $ORT_{N,eps}$ halmazt is. Ugyanis definíció szerint egy $u = (0, j_1, j_2, j_3, j_4, j_5)$ vektor csak akkor lehet N-ortogonális v_0-ra, ha léteznek $\phi_k \in I_k$ és $\psi_k \in [0, \frac{1}{N})$ számok, melyekre $1 + \sum_{k=1}^{5} e^{2i\pi(\phi_k - \psi_k)} = 0$. $\psi_k \in [0, \frac{1}{N})$ miatt $\phi_k - \psi_k$ $I_{j_k - \epsilon_k}$ intervallumba esik, ahol ϵ_k vagy 0 vagy 1, és emiatt az $u_{\underline{\epsilon}} = (0, j_1 - \epsilon_1, \ldots, j_5 - \epsilon_5)$ ORT_N-beli.

A fentiek miatt $ORT_{N,eps}$ mindazon $u^{\underline{\epsilon}} = (0, j_1 + \epsilon_1, \ldots, j_5 + \epsilon_5)$ vektorokból fog állni, ahol ϵ_k 0 vagy 1 és a $(0, j_1, j_2, j_3, j_4, j_5)$ vektor ORT_N-beli.

2.3.8. Megjegyzés. *Minden $u \in ORT_N$ vektorból 32 különböző fenti alakú u^{ϵ} vektor származik, így arra számítanánk, hogy $ORT_{N,eps}$ mérete ORT_N méretének mintegy 32-szerese lesz. Szerencsére ennél sokkal jobb a helyzet, mivel sok különböző u vektorból származó u^{ϵ} vektor egybeesik. A tapasztalat azt mutatja, hogy N-től függetlenül $ORT_{N,eps}$ mérete ORT_N méretének csak mintegy 4-szerese.*

Most már készen állunk arra, hogy elkezdjük megkeresni a lehetséges $\hat{H}_1$ mátrixokat. Emlékeztetőül az első sor és oszlop csupa nullából áll, majd utána felváltva próbálunk meg meghatározni egy-egy sort és oszlopot, azaz megpróbálunk találni egy megfelelő második sort, aztán egy ehhez passzoló második oszlopot, majd egy harmadik sort, aztán egy harmadik oszlopot, és így tovább. Minden egyes lépésnél figyelnünk kell a következőkre:

- Minden sor és oszlop ORT_N-beli.
- Minden sor (oszlop) lexikografikusan nagyobb kell, hogy legyen, mint bármely előző sor (oszlop). Emiatt a második sor és oszlop koordinátáinak monoton nővőknek kell lennie, azaz $ORT_{N,mon}$-belinek.
- A második sornak lexikografikusan kisebbnek vagy egyenlőnek kell lennie a második oszlopnál.
- Minden sornak (oszlopnak) N-ortogonálisnak kell lennie bármely korábbi sorra (oszlopra). Ez ekvivalens azzal, hogy minden oszlop- és sorpárra a modulo N vett különbségvektornak $ORT_{N,eps}$-belinek kell lennie.
- Minden sornak (oszlopnak) kompatibilisnek kell lennie az addigra előállt mátrixrésszel, azaz amikor például a negyedik sort keressük, akkor már meghatározott az első három oszlop, és így a negyedik sor első három koordinátája is.

Elkészült egy számítógépes kód, amely az előbbiekben leírt keresést hajtja végre. A futási idő „tűrhető", 1-4 nap alatt lefut N-től függően. Mindazonáltal a megtalált mátrixok száma váratlanul nagy, 10^9 és $5 \cdot 10^{10}$ közé esik, ha N 17 és 53 között van. Jelölje $PREHAD_N$ az ezzel a kereséssel előálló mátrixhalmazt. Nyilván $HAD_N \subset PREHAD_N$.

Kérdés, hogy tényleg csak azon $\hat{H}_1$ mátrixokat találtuk-e meg így, amelyek HAD_N-beliek, azaz igaz-e, hogy $HAD_N = PREHAD_N$? Vajon minden lehetséges szűrőfeltételt alkalmaztunk ezen $PREHAD_N$ halmaz keresésekor? Úgy néz ki, hogy nem ez a helyzet, és még léteznek további lehetőségek a halmaz szűkítésére. Vegyünk ehhez egy $\hat{H}_1 \in PREHAD_N$ mátrixot. 25 db „hibával terhelt" koordinátája van (az első sor és oszlop koordinátái ugye pontosan ismertek). Erről a 25 koordinátáról tudjuk, hogy az $\frac{1}{N}$ hosszúságú $I_{j_{m,k}}$ intervallumba esnek. Itt még egyszer megvizsgálhatjuk a „leszármazottakat", ahogy azt tettük az ORT_N halmaz meghatározásánál. Ez azt jelenti, hogy vehetjük mind a 25 nem pontosan ismert koordináta esetében az $I_{j_{m,k}}$ intervallumok bal vagy jobb felét. Így egy $\hat{H}_1$ mátrixhoz 2^{25} darab „gyereket" definiáltunk. Hasonlóan az ORT_N halmaz keresésekor meggondoltakhoz itt is igaz, hogy egy $\hat{H}_1$ mátrix csak akkor lehet HAD_N-beli, ha legalább egy gyerekére teljesülnek a szigorúbb páronkénti sor- és oszloportogonalitási feltételek. Ha egy gyerekre sem teljesülnek, akkor az adott $\hat{H}_1$ mátrixot nem kell tovább vizsgálnunk, biztosan nem HAD_N-beli. Nyilván mind a 2^{25} darab „gyerek" vizsgálata rendkívül lassú, de ha minden sor beillesztésekor elvégezzük ezt a vizsgálatot, akkor csak néhány ezer ilyen próbát kell elvégeznünk. A keresési algoritmus ezen részét még nem valósítottuk meg megfelelő szigorúsággal, de előzetes eredmények arra utalnak, hogy a $PREHAD_N$-beli mátrixoknak csak egy kis része állja ki ezt a

próbát, azaz HAD_N mérete sokkal kisebb az eddig megkonstruált $PREHAD_N$ méreténél. Ez nagyon fontos a teljes algoritmus futási idejének szempontjából, mert HAD_N méretét a $10^8 - 10^9$ tartományban kell tartanunk, ha „emberi" időben lefutó algoritmust szeretnénk.

2.4. $\hat{H}_1$ mátrixra torzítatlan vektorok és ellentmondás elérése

Míg az előző fejezetben leírtuk, hogy hogyan lehet meghatározni a lehetséges $\hat{H}_1$ mátrixokat, addig ebben a fejezetben egy rögzített $\hat{H}_1 \in HAD_N$ mátrixról próbáljuk meg belátni, hogy az $\{Id, \hat{H}_1\}$ párhoz nem lehet olyan $\hat{H}_2$ és $\hat{H}_3$ mátrixokat találni, hogy amelyek kielégítenék az összes olyan ortogonalitási és torzítatlansági kritériumot, amely abból ered, hogy a $\hat{H}_i$ mátrixok egy kölcsönösen torzítatlan Hadamard mátrixhármas diszkretizáltjai. $\hat{H}_2$ és $\hat{H}_3$ mátrixok sorainak például „torzítatlannak" kell lennie $\hat{H}_1$ mátrix minden sorára. Emiatt első feladatunk az ilyen vektorok listájának meghatározása.

2.4.1. Megjegyzés. *Teljes szabadságunk van a $\hat{H}_2$ és $\hat{H}_3$ mátrixoknál használandó N' diszkretizációs paraméter megválasztásának tekintetében, olyan az algoritmusunk, hogy ez különbözhet a $\hat{H}_1$ mátrixnál használt N-től. A megfelelő N és N' kiválasztásával csökkenthetjük a futáshoz szükséges időt. A Fourier család mátrixainál szerzett tapasztalataink szerint N'-t érdemes N-nél jóval kisebbre választani, mindenesetre az egyszerűbb jelölés érdekében N' helyett is N-et alkalmazunk, amíg nem akarjuk direkt megkülönböztetni őket valamiért.*

Mivel $\hat{H}_1$ mátrix első sora konstans $(0,0,0,0,0,0)$ (amik az eredeti H_1 mátrix első sorában található valós 1 értékeket reprezentálják, hiba nélkül), hasznos definiálni a következő halmazokat:

2.4.2. Definíció. *Azt mondjuk, hogy egy $u = (0, j_1, j_2, j_3, j_4, j_5)$ vektor az UB_N halmazhoz tartozik, ha léteznek olyan $\phi_k \in I_{j_k}$ számok, melyekre $|1 + \sum_{k=1}^{5} e^{2i\pi\phi_k}| = \sqrt{6}$. Ha még az is igaz, hogy u koordinátái monoton növőek, akkor u az $UB_{N,mon}$ halmazhoz is tartozik.*

Ezt az UB_N halmazt az ORT_N-hez hasonló módon konstruálhatjuk meg az 1.4. szakaszban található lemmák torzítatlanságra vonatkozó részeit segítségül hívva. Jelölje r_{j_k} az I_{j_k} intervallumok középpontját. Ekkor az iméntiek alapján a következő adódik: ha $u \in UB_N$, akkor

$$\left| |1 + \sum_{k=1}^{5} e^{2i\pi r_{j_k}}| - \sqrt{6} \right| \leq \frac{5\pi}{N}. \tag{2.4.1}$$

Hasonlóan az ORT_N esetéhez, önmagában ez a feltétel még túl gyenge lenne, így teljesen analóg módon alkalmazható itt is a „leszármazottak" 5-10 generációig lemenő vizsgálata.

2.4.3. Megjegyzés. *Itt is igaz az, hogy az UB_N halmaz invariáns az utolsó 5 koordináta $(j_1, j_2, j_3, j_4, j_5)$ permutációjára. Így a gyakorlatban csak a monoton növő vektorokat keressük, azaz $UB_{N,mon}$ halmazt határozzuk meg, majd ezen halmaz vektorai koordinátáinak permutálásával készítjük el az UB_N halmazt.*

2.4.4. Megjegyzés. *Elkészítettünk egy olyan kódot is, amely kilistázza adott N-re UB_N elemeit. Például N=17-re $|UB_N| = 479340$, míg N=19-re $|UB_N| = 764060$.*

2.4.5. Megjegyzés. *A tapasztalatok szerint, ahogy az látható az előbbi megjegyzésben, az UB_N halmaz sokkal nagyobb, mint az ORT_N halmaz. Ez elméleti érvekkel is alátámasztható, mert a komplex vektorok ortogonalitásának megkövetelése a skalárszorzat eredményéül kapott komplex számnak mind a valós, mint a képzetes részével szemben külön-külön támaszt egy feltételt, míg a torzítatlanságból csak egy feltétel következik.*

Az $ORT_{N,eps}$ halmaz mintájára szükségünk van az $UB_{N,eps}$ halmazra is:

2.4.6. Definíció. *Azt mondjuk, hogy az* $u = (0, j_1, j_2, j_3, j_4, j_5)$ *és* $v = (0, m_1, m_2, m_3, m_4, m_5)$ *vektorok* N*-torzítatlanok egymásra, ha léteznek olyan* $\phi_k \in I_{j_k}$ *és* $\psi_k \in I_{m_k}$ *számok, hogy* $|1 + \sum_{k=1}^{5} e^{2i\pi(\phi_k - \psi_k)}| = \sqrt{6}$.

Ez a tulajdonság is eltolásinvariáns abban az értelemben, hogy csak a $(j_1 - m_1, \ldots, j_5 - m_5)$ modulo N értékektől függ. Emiatt megtehetjük, hogy $m_1 = \cdots = m_5 = 0$, ezzel $v_0 = (0, 0, 0, 0, 0, 0)$ esetet választjuk (itt természetesen az első 0 a pontos nullát jelenti, míg az utolsó öt koordináta 0-ja az I_0 intervallumot jelöli), és ezzel definiáljuk az $UB_{N,eps}$ halmazt. Azaz egy 2.2.1. alakú vektor $UB_{N,eps}$-beli, ha az iménti v_0 vektorra N-torzítatlan. (Az $UB_{N,eps}$ jelölésben az *eps* azt jelenti, hogy v_0 vektort nem ismerjük pontosan, csak „epszilonnyi" hibával, mert az utolsó 5 koordinátáról csak azt tudjuk, hogy I_0-beliek, nem pontosan nullák.) Az előbbi jelöléssel úgy is megfogalmazhatjuk az eltolásinvarianciát, hogy u és v akkor és csak akkor N-torzítatlanok, ha a $(0, j_1 - m_1, \ldots, j_5 - m_5)$ modulo N vektor $UB_{N,eps}$-beli.

Legyen egy 2.2.1. alakú v vektor $\hat{H}_2$ vagy $\hat{H}_3$ egyik sora. Mivel $\hat{H}_1$ első sora konstans $(0, 0, 0, 0, 0, 0)$ (ezek a nullák valós 1-eseket reprezentálnak, hiba nélkül), következik, hogy $v \in UB_N$. Jelölje $h_2, \ldots, h_6$ $\hat{H}_1$ többi 5 sorát. Ezzel a jelöléssel a $v - h_j$ modulo N különbségek mind $UB_{N,eps}$-beliek kell, hogy legyenek minden $j = 2, \ldots, 6$-ra. Jelölje $UB_{\hat{H}_1}$ azon vektorok halmazát, amelyek $\hat{H}_1$ minden sorára torzítatlanok. A korábbiak értelmében $\hat{H}_2$ és $\hat{H}_3$ minden sora ebből a halmazból kerül ki.

Az $UB_{\hat{H}_1}$ elemeinek meghatározásakor első megközelítésben a fentiek értelmében úgy járunk el, hogy vesszük azon vektorokat, amelyek minden sorra torzítatlanok. A leszármazottak vizsgálata miatt azonban egy ennél erősebb szűrést is alkalmazhatunk. Nevezzük ezt a vizsgálatot „univerzális" torzítatlanság vizsgálatnak. Univerzális a következő értelemben: vegyünk l db (esetünkben első körben 6) 2.2.1. alakú vektort: $v_0, \ldots, v_{l-1}$. Jelölje minden $0 \leq i \leq l-1$-re $m_{i,k}$ v_i koordinátáit, és $J_{i,k}$ a nekik megfelelő intervallumot. Nekünk azon u vektorok kellenek, amelyek az összes v_i $0 \leq i \leq l-1$ vektorra „egyszerre" torzítatlanok. Ezt a következőképp fogalmazhatjuk meg formálisan: $u = (0, j_1, j_2, j_3, j_4, j_5)$ univerzálisan N-torzítatlan a $v_0, \ldots, v_{l-1}$ vektorokra, ha léteznek olyan $\phi_k \in I_{j_k}$ és $\psi_{i,k} \in J_{i,k}$ számok, melyekre minden $0 \leq i \leq l-1$-re $|1 + \sum_{k=1}^{5} e^{2i\pi(\phi_k - \psi_k)}| = \sqrt{6}$. Itt fontos látni, hogy míg a $\psi_{i,k} \in J_{i,k}$ számok függenek i-től, addig a ϕ_k számok nem. Ez azt jelenti, hogy olyan ϕ_k számokra van szükségünk, amelyek mind az l db v_i vektorhoz megfelelőek. Az iméntiek és az 1.4. szakaszban található lemmák alapján kidolgozható egy hatékonyabb módszer, mellyel a v_i vektorhalmazra N-torzítatlan vektorok előbbi módszerrel kapottnál jóval szűkebb halmaza határozható meg. A módszer szerint közvetlenül alkalmazzuk az 1.4. szakaszban található lemmákat, azaz amikor vizsgáljuk mondjuk v_i és u vektorok N-torzítatlanságát, akkor például az 1.4.1. lemmát a következő szereposztással alkalmazzuk: I_k egyértelmű, az u koordinátáinak megfelelő intervallumokat jelöl, J_k-t pedig $J_{i,k}$-val azonosítjuk. Ekkor ahhoz, hogy u minden v_i-re egyszerre torzítatlan legyen, fenn kell állnia az 1.4.1. egyenlőtlenségnek. Ez önmagában még semmilyen többletet nem hordozna az egyenkénti vizsgálathoz, ám amikor a „gyerekeket" vizsgáljuk, akkor az kell u vektor „túléléséhez", hogy legyen olyan „gyerek", amelyik mindegyik v_i-re torzítatlan, míg ha vektoronként vizsgálnánk a torzítatlanságot, akkor „életben" hagynánk olyan u vektort, amelyhez ugyan létezik olyan leszármazott minden v_i vektorhoz, amellyel torzítatlan, de ezek a „gyerekvektorok" különböznek v_i vektoronként, viszont az „univerzális" vizsgálat kiszűri az ilyen u vektorokat. A tapasztalatok azt mutatják, hogy ez a fajta szűrés jelentős többletet (azaz jelentősen kisebb méretű $UB_{\hat{H}_1}$ halmazt) eredményez, mint egyszerűen a soronként megfelelő

halmazok.

2.4.7. Megjegyzés. *Az 1.4. szakaszban található lemmák alakjából látszik, hogy nem kell, hogy a két diszkretizációs paraméter (a $\hat{H}_1$ mátrixhoz illetve a $\hat{H}_2$ és $\hat{H}_3$ mátrixokhoz használt) megeggyezzen (ezt már implicit ki is használtuk az ORT_N halmaz előállításánál), így itt is megjegyezzük, hogy lehetséges különböző diszkretizációs paraméterek használata.*

2.4.8. Megjegyzés. *Megvalósítottunk egy „univerzális" torzítatlan vektorokat kereső programkódot, és az eredmények azt mutatják, hogy az $UB_{\hat{H}_1}$ halmaz mérete lényegében független $\hat{H}_1$ választásától, míg $\hat{H}_1$ mátrix N diszkretizációs paraméterét növelve csökken a mérete, azonban $\hat{H}_2$ N' diszkretizációs paraméterét növelve nő $UB_{\hat{H}_1}$ mérete. Mindazonáltal nagyságrendileg minden esetben az $10^3 - 10^4$ tartományban mozog ez az érték, ha N-et és N'-t a 17-53 tartományból választjuk.*

2.4.9. Megjegyzés. *Ha feltesszük, hogy valamilyen módon meghatároztunk egy megfelelő $\hat{H}_1$ és $\hat{H}_2$ mátrixpárt, akkor a $\hat{H}_3$ mátrix soraira igaz az, hogy mind az $UB_{\hat{H}_1} \bigcap UB_{\hat{H}_2}$ halmaz elemei, sőt a fenti értelemben „univerzálisan" mind a 12 vektorra torzítatlannak kell lenniük. Ez már egy nagyon erős feltétel, a tapasztalatok szerint csak néhány vektor „éli túl". Megjegyezzük, hogy a 12 vektor között lehet különböző diszkretizációs paraméterrel diszkretizált, ez nem zavarja az algoritmust.*

Így, hogy elkészítettük az $UB_{\hat{H}_1}$ halmazt, be kell látnunk, hogy nem lehetséges belőle $\hat{H}_2$ és $\hat{H}_3$ mátrixok összeállítása.

Vegyük $UB_{\hat{H}_1}$ vektorait, és próbáljuk meg összeállítani belőlük $\hat{H}_2$ mátrixot. Ehhez szükségünk van 6 vektorra (legyenek ezek $b_1, b_2, b_3, b_4, b_5, b_6$), amelyekre igaz, hogy a $b_k - b_m$ modulo N különbségvektorok mind $ORT_{N,eps}$-beliek. Ezeket a feltételeket és paramétereket figyelembe véve arra számíthatnánk, hogy csak kevés $\{Id, \hat{H}_1, \hat{H}_2\}$ kölcsönösen torzítatlan komplex Hadamard mátrixhármas létezik, azaz egy $\{Id, \hat{H}_1\}$ torzítatlan Hadamard mátrixpár általában nem egészíthető ki $\{Id, \hat{H}_1, \hat{H}_2\}$ torzítatlan mátrixhármassá. Ez azt jelentené, hogy reménykedhetünk abban, hogy az ellentmondást már ennél a lépésnél elérjük, azaz nem találunk 6, ezen feltételeknek megfelelő diszkretizált vektort. A helyzet azonban sajnos nem ez. Újkeletű eredmények[5][9] azt mutatják ugyanis, hogy kölcsönösen torzítatlan komplex Hadamard mátrixhármasok végtelen számosságú családjai léteznek. Numerikus eredmények is azt mutatják, hogy $\hat{H}_2$ mátrixot minden $\hat{H}_1$ esetén tényleg össze lehet állítani $UB_{\hat{H}_1}$ elemeiből. Emiatt ezen a ponton még nem kapunk ellentmondást, tovább kell vizsgálódnunk, azaz minden összeállítható $\hat{H}_2$-re elő kell állítanunk az $UB_{\hat{H}_1,\hat{H}_2}$ halmaz elemeit, amely halmaz azon vektorokból áll, amelyek az előbbiekben definiáltaknak megfelelően univerzálisan torzítatlanok $\hat{H}_1$ és $\hat{H}_2$ minden sorára, azaz 12 vektorra. A következő lépés az, hogy most $UB_{\hat{H}_1,\hat{H}_2}$ halmaz elemeiből kell összeállítanunk $\hat{H}_3$ mátrixot, itt is figyelnünk kell arra a feltételre, hogy a sorok modulo N vett különbségvektorainak $ORT_{N,eps}$-belinek kell lenniük. Az ellentmondást ezen a ponton érhetjük el, azaz nem tudunk kiválasztani 6, minden ortogonalitási feltételnek eleget tevő vektort $UB_{\hat{H}_1,\hat{H}_2}$ halmazból, ha N elegendően nagy. A tapasztalat azt mutatja, hogy ha $\hat{H}_1$ esetében az $N = 53$, míg $\hat{H}_3$ és $\hat{H}_3$ esetében az $N = 19$ diszkretizációs paramétert választjuk, akkor már elérjük az ellentmondást. A kutatás ezen részébe, azaz $\hat{H}_2$ és $\hat{H}_3$ mátrixok keresésébe kapcsolódtam be én. A végső cél az, hogy minden adott $\hat{H}_1$ mátrixra ez az egész keresés lefusson néhány másodperc alatt.

2.5. Eredmények a Fourier családdal kapcsolatban

Az előző szakaszban ismertetett diszkretizációs eljárás teljes egészében megvalósításra került a Fourier családba tartozó mátrixok esetében [9]. A számítógépes kódok és az eredmények teljes

dokumentációja megtalálható a [22] weboldalon. A teljes futásidő mintegy 6 órát tett ki egy 3,2 GHz-es processzorral rendelkező számítógépen.

Az 1.2.4. szakaszban már láttuk, hogy a Fourier család egy kétparaméteres család, azaz $F(a,b)$ alakban írhatunk le egy elemet, ahol (a,b) a $[0,1)^2$ négyzet pontjain vesz fel értékeket. Az Hadamard mátrixok között megfogalmazott ekvivalenciarelációkon túl megfogalmazhatóak további ekvivalenciarelációk[1], így keresésünket az általánosság megszorítása nélkül leszűkíthetjük a $[0,1)^2$ négyzet egy olyan háromszögalakú részére, amelynek csúcsai $(0,0),(\frac{1}{6},0)$ és $(\frac{1}{6},\frac{1}{12})$. Tehát csak olyan (a,b) párok jöhetnek szóba, melyek az imént meghatározott háromszögbe esnek, ez nagyban lecsökkenti a megvizsgálandó $\hat{H}_1$ mátrixok számát.

Az 1.2.4. szakaszban azt is láthattuk, hogy a Fourier családba tartozó mátrixoknál csak a 2., a 4. és a 6. oszlopvektorban van az (a,b) paraméterektől függő elem, a többi oszlopot pontosan ismerjük. Emiatt, amikor az ORT és UB halmazokat próbáljuk meghatározni, és az 1.4. szakaszban ismertetett lemmákat próbáljuk alkalmazni, akkor az intervallumok hosszánál (L_k és T_k) az olyan oszlopvektoroknál, amelyekben nincs (a,b) paraméterektől függő elem, minden intervallum elfajuló, azaz 0 hosszúságú, míg a második, negyedik és hatodik oszlopvektornál csak 4 intervallum nemelfajuló, emiatt a halmazokat meghatározó távolságok is kisebbek lesznek, mint az általános esetben.

A keresésnél a $\hat{H}_1$ mátrix N diszkretizációs paraméterét 180-nak választottuk, míg a $\hat{H}_2$ és $\hat{H}_3$ mátrixok N' diszkretizációs paramétere 19 volt. Ezekkel a paraméterekkel általában 110-140 vektor kerül az $UB_{\hat{H}_1}$ halmazba. Ez egy elég jó eredmény, hiszen az 1.2.4. szakaszban leírt numerikus eredmények alapján azt sejthetjük, hogy ha pontosan ismerjük a-t és b-t, akkor 48 vektort tartozik ebbe a halmazba. Az előző szakaszban ismertetett 10^3-10^4 darabos eredménynél sokkal jobb eredményt egyrészt a jóval nagyobb diszkretizációs paraméter, valamint az előző bekezdésben ismertetett tények okozhatják, vagyis az, hogy sokkal „jobban”, azaz kisebb bizonytalansággal ismerjük a $\hat{H}_1$ mátrixot, mint az általános esetben.

Az előző bekezdésben említett 110-140 vektorból kell összeállítanunk $\hat{H}_2$ és $\hat{H}_3$ mátrixokat. A tapasztalat azt mutatja, hogy adott $\hat{H}_1$ mátrix esetén általában 1000-5000 db az ortogonalitási feltételeknek eleget tevő $\hat{H}_2$ mátrixot tudunk összeállítani, mindazonáltal némely speciális $\hat{H}_1$ mátrixra több millió $\hat{H}_2$ mátrix állítható elő (ez összhangban van azzal a tapasztalattal, hogy $F(1/6,0)$ mátrixnál 70 db ilyen Hadamard mátrix van, míg általában csak 8).

A keresés befejezéseként minden előállt $(\hat{H}_1,\ \hat{H}_2)$ mátrixpárra meg kell próbálnunk az előző szakaszban leírt módon összeállítani egy $\hat{H}_3$ mátrixot. Ezt számítógépes kód végigpróbálta az összes esetre, és egyik esetben sem sikerült összeállítani a $\hat{H}_3$ mátrixot, ezzel az 1.2.2. tételt bizonyítottuk. □

2.5.1. Megjegyzés. *Elviekben az ebben a fejezetben leírt, és a Fourier családba tartozó mátrixokra végrehajtott számítógépes keresés eredményre vezethetne az 1.1.3. sejtés bizonyításához is, de a gyakorlatban súlyos problémák merülnek fel. Az általános esetben nyilván nem tehetjük fel, hogy H_1 mátrix $F(a,b)$ alakú, csak azt tudjuk, hogy H_1 komplex Hadamard mátrix. Ha ismert lenne a 6 dimenziós komplex Hadamard mátrixok teljes klasszifikációja néhány paraméteres formában leírható család formájában, akkor a Fourier családra alkalmazott módszer végrehajtható lenne „emberi” időben. Viszont sajnos ilyen klasszifikáció jelenleg nem ismert, így marad az a módszer, hogy mind a 25 nem rögzített koordinátában $1/N$ hibát engedünk meg. Ebben az esetben viszont a Fourier családnál tapasztalt 270 különböző $\hat{H}_1$ mátrixszal szemben többmilliárd $\hat{H}_1$ mátrixot kell megvizsgálnunk még jóval kisebb N diszkretizációs paraméter mellett is, ezen túlmenően az $UB_{\hat{H}_1}$ halmaz mérete is jóval nagyobb lesz, egyrészt amiatt, hogy N kisebb, másrészt amiatt, hogy mind*

a 25 koordináta hibával terhelt. A nagyobb $UB_{\hat{H}_1}$ halmaz pedig, ahogy azt a következő fejezetben majd látni fogjuk, azt eredményezi, hogy az ellentmondást akár nagyságrendekkel lassabban érjük el, ha elérjük egyáltalán. Emiatt újabb ötleteket kell majd bevetnünk.

3. fejezet

$\hat{H}_2$ és $\hat{H}_3$ mátrixok keresése az $UB_{\hat{H}_1}$ halmazban

A kutatásba én ezen a ponton kapcsolódtam be, első feladatom az volt, hogy minél hatékonyabban, gyorsabban találjak $\hat{H}_2$ mátrixokat az $UB_{\hat{H}_1}$ halmazban.

A programozási munkát javarészt a Microsoft által kifejlesztett .NET technológiát alkalmazva, C# nyelvet használva végeztem el, bár a témavezetőmtől kapott kódok C++ nyelven íródtak, és nagy valószínűséggel a végső kód is C++ nyelven fog megvalósulni. Annak az oka, hogy mégis így döntöttem, az az, hogy C#-ban jóval nagyobb tapasztalattal rendelkezem, valamint az a tény, hogy a .NET keretrendszer beépített szolgáltatásai (a C++ STL-jénél jóval bővebb class library, valamint a memóriamenedzsment) jóval gyorsabbá teszik a fejlesztést, és a kutatás ezen fázisára jellemző az útkeresés, ami miatt a megírt kódoknak csak egy kis része lenne felhasználható a végső verzióban. A kódok futtatáshoz kétmagos 3,2 GHz-es processzorral és 2 GB RAM-mal rendelkező gépet használtam, a memóriaigényes kódokat egy négymagos, 2,5 GHZ-es processzorral, valamint 4 GB RAM-mal szerelt szerveren futtattam.

3.1. $\hat{H}_2$ mátrix keresése az $UB_{\hat{H}_1}$ halmazban

Az előző fejezetben ismertettünk egy diszkretizációs eljárást. Az alábbiakban használt jelölések mind az ott ismertetett jelentéssel bírnak ebben a fejezetben is.

Ismételjük át akkor, hogy milyen tulajdonságoknak kell $\hat{H}_2$ mátrixnak megfelelnie:

- Minden sorvektora $UB_{\hat{H}_1}$-beli
- Sorai N-ortogonálisak egymásra

Ezen két feltétel teljesülésének vizsgálatához gráfelméleti eszközöket hívunk segítségül. Definiáljunk egy $G_{\hat{H}_1}$ gráfot egy adott $\hat{H}_1$-hoz a következőképp: legyenek a gráf csúcsai az összes $UB_{\hat{H}_1}$-beli vektor, két csúcs pedig legyen összekötve akkor, ha a nekik megfelelő $UB_{\hat{H}_1}$-beli vektorok N-ortogonálisak. Ekkor $\hat{H}_2$ keresése a gráfelmélet nyelvén azt jelenti, hogy 6-klikkeket kell keresnünk a $G_{\hat{H}_1}$ gráfban. Erre a problémára sajnos nem ismert a „nyers erő” alkalmazásánál lényegesen gyorsabb megoldás, azaz azt kell tennünk, hogy vesszük az összes csúcshatost és megnézzük, hogy minden él be van-e húzva. Ennél a megoldásnál némileg jobb az, ha rendezzük a csúcsokat valahogy, és rendezett csúcshatosokat keresünk, majd a közöttük lévő éleket vizsgáljuk. A csú-

csok rendezésére jó megoldásnak bizonyult a nekik megfelelő vektorok lexikografikus rendezésének átvétele.

Kardinális kérdés az élek vizsgálata. Az előző fejezetben láttuk, hogy hogyan lehet eldönteni két 2.2.1. alakú vektorról, hogy N-ortogonálisak-e: N-ortogonálisak, ha modulo N vett különbségvektoruk $ORT_{N,eps}$-beli. Mivel a keletkező $G_{\hat{H}_1}$ gráf mérete nem túl nagy (N=53 és N'=19 esetén $\approx 10^3$ csúcs és 10^6 él), érdemesenek tűnik először az egész gráfot az összes csúccsal és éllel együtt előállítani a memóriában, így a különbségvektorokat csak egyszer kell előállítani, ezzel futási időt lehet spórolni. A gráf leírását egyrészt a csúcsai adják, másrészt a rendezettség miatt minden csúcshoz elég letárolni a nála nagyobb csúcsokba vezető éleket.

Az élek meghatározásakor a következő problémával állunk szemben: adott két csúcs, előállítjuk a különbségvektort és el kell döntenünk, hogy az $ORT_{N,eps}$ halmazba tartozik-e. Ilyen jellegű problémával a későbbiek során is többször találkozunk, így érdemes programozástechnikai szempontból is megvizsgálni a kérdést. Ebből a szempontból egy adott elemről akarjuk eldönteni, hogy egy adott halmazban benne van-e. A naiv megoldás az lehet, hogy a halmaz elemeit egy listába tesszük, és az adott elemet minden halmazelemmel összehasonlítjuk, ám ez $O(n)$ idejű megoldást ad, amennyiben n-nel jelöljük a halmaz méretét. Ehhez a megoldáshoz kell folyamodnunk akkor, ha semmilyen plusz információnk nincs a halmaz elemeiről, pontosabban semmilyen plusz struktúra nincs definiálva a halmaz elemein és a lehetséges vizsgálandó elemeken. Szerencsére esetünkben nem ez a helyzet, hiszen például egy rendezést definiáltunk az összes a halmazban vagy vizsgálandó elemként előfordulható vektoron, így ha a halmaz elemeit rendezett listában tároljuk, akkor megvalósítható egy $O(logn)$ idejű megoldás, mégpedig a bináris keresés. Ha a szóbajöhető elemeken definiálható egy jó hasítófüggvény (azaz egy olyan függvény, ami minden elemhez hozzárendel egy természetes számot, és viszonylag kevés elempárra igaz az, hogy ugyanazt a számot rendeli hozzájuk), akkor egy asszociatív tömbben is tárolhatjuk a listát (C#-ban például HashSetnek hívják a megfelelő nyelvi elemet). Ennek az az előnye, hogy az ELEME-E művelet $O(1)$ idejű műveletté válik. Ennek a módszernek a hátránya, hogy memóriaproblémák miatt csak maximum 10^5-10^6 méretű halmazok kezelhetőek vele a rendelkezésre álló memória méretétől függően, és az eleme-e művelet sebessége is csökken a méret növekedésével. N=19-re $ORT_{N,eps}$ mérete, mint láttuk $\approx 4 \cdot 10^5$, így kezelhetőség határán vagyunk már ilyen kis N-re is. Nagyobb N-ekre már biztosan nem használható a módszer.

Térjünk vissza azonban a hasítófüggvény kérdéséhez. Jó hasítófüggvény lesz a következő:

$$f(\underline{u}) = f((0, j_1, j_2, j_3, j_4, j_5)) = (((j_1 N + j_2)N + j_3)N + j_4)N + j_5 \tag{3.1.1}$$

ahol $0 \leq j_k \leq N-1$, így $0 \leq f(\underline{u}) \leq N^5 - 1$. Kellemes tulajdonsága az így definiált hasítófüggvénynek, hogy nincs kulcsütközés, azaz ha $\underline{u} \neq \underline{v}$ akkor $f(\underline{u}) \neq f(\underline{v})$, azaz az $\underline{u}$ vektorok és az $f(\underline{u})$ értékek között egy egyértelmű megfeleltetés van. Fontos még megjegyezni, hogy a lehetséges $\underline{u} = (0, j_1, j_2, j_3, j_4, j_5)$ sorvektorokon értelmezett lexikografikus rendezés és az $f(\underline{u})$-val definiált ($\underline{u} < \underline{v}$ akkor és csak akkor, ha $f(\underline{u}) < f(\underline{v})$) rendezés ugyanaz. Emiatt a továbbiakban a lexikografikus rendezés helyett az így kapott rendezést használjuk, sőt, mivel az eredeti vektoroknak minden, a 6-klikkek kereséséhez szükséges információját tárolja $f(\underline{u})$, nem is a vektorokkal magukkal, hanem a nekik megfelelő $f(\underline{u})$ értékekkel dolgozunk a továbbiakban, azaz ha azt mondjuk, hogy egy csúcs nagyobb, mint egy másik, akkor azt értjük alatta, hogy az első csúcshoz tartozó függvényérték nagyobb, mint a másodikhoz tartozó, és ez magával vonja azt is, hogy a megfelelő vektorok közül az első lexikografikusan nagyobb, mint a második.

3.1.1. Megjegyzés. *A 3.1.1. függvény sajnos nem alkalmas közvetlenül arra, hogy két vektorról meghatározzuk, hogy mi a modulo N különbségük, mivel általában* $f(\underline{u}) - f(\underline{v}) \neq f((\underline{u} - \underline{v})modulo$

N).

Láttuk, hogy a 3.1.1. hasítófüggvény alkalmas az asszociatív tömbbel megvalósított halmaz-eleme-e vizsgálathoz, ám maga a módszer csak kis N esetén használható. Az alábbiakban bemutatásra kerülő módszer jóval nagyobb N esetén is használható. A módszer a bool-tömbös módszer. Akkor alkalmazható, ha az összes szóbajöhető elem M száma véges, pontosabban a memória bitekben mért nagyságánál kisebb M. Szükséges egy függvény, ami minden elemet különböző 0 és $M-1$ közé eső (a határokat is beleértve) természetes számra képez le úgy, hogy f minden értéket felvesz ebben a képhalmazban (a 3.1.1. hasítófüggvény pontosan ilyen). Definiáljunk egy olyan tömböt, melynek minden eleme bool értékű. A tömb i-edik eleme igaz, ha $f^{-1}(i)$ eleme a halmaznak, és hamis, ha nem. Ekkor minden lehetséges elemhez egy bitnyi memória tartozik, a halmazba tartozás $o(1)$ időben eldönthető, és a tapasztalatok azt mutatják, hogy egy ilyen bool tömb egy elemének elérése egy nagyságrenddel gyorsabb, mint az asszociatív tömb elemeinek elérése, tehát a konstansban egy nagyságrendnyi eltérés van. Mind a C#, mind a C++ esetén létezik bool típus, ám mindkét nyelvben egy egész bájtot, azaz 8 bitet lefoglal egy ilyen típusú változónak a program. Szerencsére mindkét nyelvben létezik olyan típus, ami egy biten tárol egy bool értéket, amennyiben sok bool értéket akarunk tárolni, a C# esetén a BitArray, míg C++ esetén a vector<bool> típus, közös bennük, hogy a megvalósítás miatt némileg lassabb megoldást nyújtanak, mint a tisztán bool tömbös megoldás, ám a 8-szoros memóriahasználatbeli különbség miatt mindenképpen érdemes használni őket.

3.1.2. Megjegyzés. *A bool-tömbös módszer memóriaigényéről elmondható, hogy $N^5/8$ bájtnyi, azaz mondjuk 4 GB memóriában maximum $N = \sqrt[5]{2^{32} \cdot 8} = 128$ esetén fér el egy ilyen bool tömb.*

3.1.3. Megjegyzés. *Az oszlopok ortogonalitásvizsgálatánál szükség lesz majd olyan vektorok adott halmazba tartozásának vizsgálatára, amelyeknek mind a 6 koordinátája csak hibával terhelten, 1/N pontossággal ismert, ilyen vektorokból N^6 db van, és teljesen analóg módon definiálható hasítófüggvény hozzájuk, valamint a hasítófüggvénnyel a bool-tömbös módszer is teljesen hasonló módon működik. A memóriaigény viszont a fenti esetnek pont az N-szerese.*

Ezután visszatérhetünk a 6-klikkek kereséséhez. A minél gyorsabb keresés érdekében először megkeressük minden élhez azon csúcsokat, amelyekkel az él mindkét végpontja össze van kötve, valamint az adott csúcs nagyobb, mint az él bármely végpontja, ezeket eltároljuk egy speciális T adatstruktúrában: ez az adatstruktúra legfelső szinten egy rendezett lista, mely azokat az a csúcsokat tartalmazza, melyek egy vagy több háromszög legkisebb csúcsai, emellett minden ilyen a csúcshoz tartalmazza azon b csúcsok rendezett listáját, melyekre létezik olyan háromszög, melynek két kisebb csúcsa a és b, és végül minden ilyen a, b párhoz eltároljuk az ilyen háromszögek harmadik c csúcsát. Ezzel lényegében eltároltuk rendezve az összes háromszöget. Minden 6-klikk két háromszögből áll elő. Sőt, fel lehet bontani minden 6-klikket egyértelműen két olyan háromszögre, A-ra és B-re, hogy minden A-beli csúcs kisebb legyen bármely B-beli csúcsnál. Nyilván, hiszen akárhogy is osztjuk két hármasra a 6-klikk csúcsait, a kapott hármasok háromszöget alkotnak, hiszen a 6-klikkben definíció szerint minden él be van húzva. Így speciálisan akkor is két háromszöget kapunk, ha a 3 legkisebb csúcs kerül az egyik csoportba, és a 3 nagyobb pedig a másik csoportba.

Az előbbiek alapján, ha a háromszögek már rendelkezésünkre állnak, akkor a 6-klikkek keresése lényegében abból áll, hogy a háromszögek között keresünk egymáshoz „passzolóakat", azaz olyanokat, amelyekre igaz, hogy bárhogyan is vegyünk egy-egy csúcsot a két háromszögből, mindig fut él a két kiválasztott csúcs között. Az előbbi bekezdésben megfogalmazottak miatt feltehetjük,

hogy a két háromszög csúcsai olyanok, hogy az egyik A háromszög minden csúcsánál nagyobb a B háromszög minden csúcsa. A keresést a következőképp hajtjuk végre: a T adatstruktúrában lexikografikusan rendezve vannak a háromszögek a csúcsok alapján az előző bekezdésben leírt módon, majd egy A háromszöget vizsgálunk: vesszük azon csúcsokat, melyek legkisebb csúcsai egy vagy több háromszögnek, és emellett nagyobbak, mint a vizsgálandó háromszög legnagyobb csúcsa, - ez a T adatstruktúra legkülső szintje, tehát ezen csúcsok listája készen van -, és ezeken végigmenve kiválasztjuk azokat, amelyekbe A minden csúcsából vezet él, jelölje ezen csúcshalmazt $T_{A,4}$. A 3 csúcsát $T_{A,4}$ bármely elemével 4 csúcsra kiegészítve olyan csúcsnégyest kapunk, ami lehet, hogy kiegészíthető 6-klikké, tehát nem kapjuk meg az összes 4-klikket. Most rögzítjük $T_{A,4}$ egy e elemét. Vesszük az összes olyan f csúcsot, amelyek nagyobbak, mint e és létezik olyan háromszög, amely két kisebb csúcsa e és f - ez a T adatstruktúrában szintén rendelkezésünkre áll - ezen f csúcsok közül kiválasztjuk azokat, amelyekbe A minden csúcsából megy él (e csúcsból nyilván megy él, mert T adatstruktúra felépítése ezt biztosítja) jelölje ezen pontok halmazát $T_{A,5}$. Itt is megjegyezzük, hogy nem kaptuk meg az összes 5-klikket, csak egy részüket, azokat, amelyekből lehet 6-klikk. Végül T adatstruktúrából véve azon g csúcsok listáját, amik e és f csúcsokkal háromszöget alkotnak már csak azt kell megvizsgálnunk, hogy g-be A minden csúcsából megy-e él, hiszen e és f csúcsokból T adatstruktúra felépítése miatt ez biztosított. Az ilyen e, f és g csúcshármasok az A háromszög csúcsaival kiegészülve alkotják a gráfban található 6-klikkeket.

3.1.4. Megjegyzés. *A 6-klikk keresésének megvalósítása során azt tapasztaltam, hogy a futási időre rendkívül nagy hatással van az, hogy mennyi műveletet végzünk a 4. 5. és 6. csúcsok kiválasztásakor, sőt minél nagyobb sorszámú csúcsot keresünk, annál fontosabb, hogy lehetőleg minél kevesebb műveletet hajtsunk végre. A magyarázat az, hogy minél nagyobb sorszámú csúcsot akarunk kiválasztani, annál többször lefut a megfelelő rész, így nagyon hangsúlyos a magas sorszámú csúcsok kiválasztásának módja. A T adatstruktúra szerepe az, hogy ezen magas sorszámú csúcsok kiválasztásakor lecsökkentsük a szóbajöhető csúcsok számát, hogy minél kevesebb csúccsal kelljen - akár csak egy nagyon egyszerű élvizsgálat erejéig - foglalkoznunk.*

Az eredmények mérsékelten biztatóak voltak. N=53 elsődleges és N'=19 másodlagos diszkretizációs paramétert választva ≈2000 elemű $UB_{\hat{H}_1}$ halmaz keletkezik, az ebből létrejövő $G_{\hat{H}_1}$ gráfon $\approx 5 \cdot 10^5$ élet definiálnak az ortogonalitási feltételek. Az algoritmus $\approx 10^{10}$ db 6-klikket talált. A fenti kereséssel $\approx$ 10 percet tölt el a program.

Az $UB_{\hat{H}_1}$ halmaz méretét próbaképp önkényesen (egyszerűen az első néhány vektort kiválasztva), illetve N diszkretizációs paramétert növelve csökkentettem. Kiderült, hogy a 6-klikkek száma rendkívüli mértékben érzékeny $UB_{\hat{H}_1}$ halmaz méretére. A halmaz méretének felezése a 6-klikkek számát egyszázadára csökkentette. Ezzel párhuzamosan a futási idő is csökkent egy nagyságrendet. Világos, hogy ilyen nagy átlagos fokszámú ($\approx$ 200) gráfnál néhány újabb csúcs megsokszorozhatja a 6-klikkek számát, mert rengeteg újabb kapcsolat keletkezik.

N növelésével az $UB_{\hat{H}_1}$ halmaz mérete szépen lassan csökken, N=73-nál $\approx$ 1100, N=101-nél $\approx$ 700, míg N=149-nél már $\approx$ 400 a mérete, ezzel párhuzamosan azonban az előállításhoz szükséges idő kb. 1-2 perc, szintén csökkenő tendenciát mutat. Mindazonáltal a 100 feletti N-ek esetében a 6-klikkek száma már kezelhetővé válik.

3.2. $\hat{H}_2$ mátrix oszlopai ortogonalitásának vizsgálata

Nem vizsáltuk eddig az oszlopokra vonatkozó ortogonalitási feltételeket. Ebben a szakaszban rátérünk erre a témára. Emlékeztetőül, ebben a szakaszban végig - az általános konvencióval ellentétesen - úgy tekintünk a mátrixokra, hogy annak sorvektorai alkotják a bázist, a sorvektorokat normáljuk úgy, hogy az első koordinátában 1 legyen, és a torzítatlansági feltételt is a mátrixok soraira fogalmazzuk meg. Az előző szakaszban is a sorvektorok ortogonalitását vizsgáltuk. Mindazonáltal tudjuk $\hat{H}_2$ mátrixról, hogy egy komplex Hadamard mátrix N'-diszkretizáltja, és a komplex Hadamard mátrixok tulajdonsága, hogy mind a sor-, mind az oszlopvektorai ortogonálisak egymásra. Emiatt minden, nem diszkretizált Hadamard mátrixra igaz, hogy mind a sor-, mind az oszlopvektorai ortogonálisak egymásra, és emiatt minden diszkretizáltjára is megfogalmazhatunk valamilyen oszloportogonalitási feltételeket is. Fontos megjegyezni, hogy abból, hogy a diszkretizált mátrix sorai teljesítik a fenti sorortogonalitási feltételeket, még nem következik, hogy a mátrix oszlopai is automatikusan teljesítenének minden megfogalmazható ortogonalitási feltételt, hiszen a diszkretizált sorok ortogonalitási feltételei csak annyit jelentenek, hogy léteznek az egész számoknak megfelelő, az adott kicsi ívrészre eső komplex számok, amelyekkel a sorok páronként ortogonálisak. Vegyünk 3 tetszőleges sort, h_1, h_2 és h_3 sorvektorokat, ekkor nem következik még az sem, hogy léteznének olyan komplex számok ehhez a három vektorhoz, amelyek beleesnének az egész számoknak megfelelő kicsi ívrészekbe, és egyszerre teljesítenék a 3 páronkénti ortogonalitási feltételt, hiszen előfordulhat, hogy más és más komplex számokkal teljesül ez a három feltétel, és így a páronkénti feltételeket teljesítik a sorok, de egyszerre már nem. Az oszlopok vizsgálata mellett szól az a tény, hogy a módszerünk, mellyel az ortogonalitási feltételeket ellenőrizzük becslésre épül, azaz olyan diszkretizált vektorokat is ortogonálisnak tekintünk, amelyek valójában nem azok.

A diszkretizált oszlopvektorok közül az első csupa 0, méghozzá ezek a nullák azt jelentik, hogy abban a koordinátában pontosan a valós 1 van. A második diszkretizált oszlop az elkészítés módja miatt, azaz amiatt, hogy a diszkretizált sorvektorokat lexikografikusan növekvő sorrendben próbáljuk meg beilleszteni a mátrixba, monoton növő koordinátákkal rendelkezik, viszont az eddigiekkel ellentétben nem tehetjük fel, hogy az első koordináta 0, illetve a nemdiszkretizált esetben 1 lenne. A többi diszkretizált oszlopvektorról még ennyit sem tudunk, tehát ott a 6 koordináta mindegyike egy [0,N'-1] intervallumból kikerülő szám. A nemdiszkretizált oszlopvektorokról tehát azt tudjuk, hogy a következő, az 1.2.1. alakhoz hasonló alakúak:

$$\mathbf{u} = \frac{1}{\sqrt{6}}(e^{2i\pi\phi_0}, e^{2i\pi\phi_1}, e^{2i\pi\phi_2}, e^{2i\pi\phi_3}, e^{2i\pi\phi_4}, e^{2i\pi\phi_5}) \tag{3.2.1}$$

Ez a diszkretizált vektorokra formálisan a következőt jelenti:

$$\hat{u} = (j_0, j_1, j_2, j_3, j_4, j_5) \tag{3.2.2}$$

ahol $0 \leq j_k \leq N' - 1$.

3.2.1. Megjegyzés. *Emlékeztetőül: ez az $\hat{u}$ vektor reprezentálja a lehetséges* $\mathbf{u}$ *vektorok egy részhalmazát, mégpedig azokat, melyekre minden* $0 \leq k \leq N' - 1$*-re* $\phi_k \in I_{j_k}^{N'}$.

Az alábbiakban bizonyítás nélkül közöljük az 1.4. szakaszban ismertetett lemmák egy olyan általánosítását, amely lehetővé teszi, hogy a mind a hat koordinátájában ismeretlen egész számokat tartalmazó vektorokra is megfogalmazhassunk a 2.3.2. és a 2.3.1. feltételhez hasonló ortogonalitási feltételeket.

3.2.2. Lemma. *Vegyünk I_k és J_k $(0 \le k \le 5)$ $[0,1]$-beli zárt, akár elfajuló intervallumokat. Jelölje rendre L_k és T_k I_k és J_k hosszúságát, valamint m_k és s_k a középpontjukat.*

Legyen $S = \frac{1}{6}(\sum_{k=0}^{5} e^{2i\pi(m_k - s_k)})$ a középpontok összege. Ekkor a következő állítást fogalmazhatjuk meg: ha kiválaszthatóak olyan $\phi_k \in I_k$ és $\psi_k \in J_k$ pontok, melyekre $|\frac{1}{6}(\sum_{k=0}^{5} e^{2i\pi(\phi_k - \psi_k)})| = 0$, akkor

$$\frac{\pi}{6}\sum_{k=0}^{5}(L_k + T_k) \ge |S|\,, \tag{3.2.3}$$

3.2.3. Lemma. *Vegyünk I_k és J_k $(0 \le k \le 5)$ $[0,1]$-beli zárt, akár elfajuló intervallumokat. Jelölje rendre L_k és T_k I_k és J_k hosszúságát, m_k és s_k a középpontjukat, valamint rendre i_k^-, i_k^+ és j_k^-, j_k^+ az alsó és felső végpontjukat. Tegyük fel továbbá, hogy $\frac{1}{6}\sum_{k=0}^{5}(L_k + T_k) < \frac{1}{\pi}$.*

Legyen $S = \frac{1}{6}(\sum_{k=0}^{5} e^{2i\pi(m_k - s_k)})$ a középpontok összege, továbbá vegyük mind a 64 végponti összeget: $S_{\underline{\epsilon}} = \frac{1}{6}(\sum_{k=0}^{5} e^{2i\pi(i_k^{\epsilon_k} - j_k^{-\epsilon_k})})$ (itt $\underline{\epsilon}$ bármilyen $\pm$ elemekből álló vektor jelenthet; vegyük észre, hogy $-\epsilon_k$ található j_k felsőindexében). Ekkor a következő állítást fogalmazhatjuk meg: ha kiválaszthatóak olyan $\phi_k \in I_k$ és $\psi_k \in J_k$ pontok, melyekre $|\frac{1}{6}(\sum_{k=0}^{5} e^{2i\pi(\phi_k - \psi_k)})| = 0$, akkor

$$\max\{|S - S_{\underline{\epsilon}}|\} \ge |S|. \tag{3.2.4}$$

A bizonyítás szinte szóról-szóra ugyanúgy megy, mint az 1.4. szakaszban, csak a középponti összegben az első konstans 1 helyett a szumma kibővült az első koordináta intervallumának középpontjával.

3.2.4. Megjegyzés. *Az imént megfogalmazott lemmák természetéből adódik, hogy teljesen analóg módon megfogalmazható és bizonyítható a torzítatlanságról szóló változatuk, mindazonáltal mivel az oszlopok torzítatlanságára a komplex Hadamard mátrixok esetén nincs semmilyen megkötésünk, ezek közlésétől eltekintünk.*

A továbbiakban először definiáljunk egy a 2.3.2. definícióval analóg egzakt feltételt arra nézve, hogy mikor tekintjük N-ortogonálisnak az oszlopvektorokat a csupa 0 első oszlopra.

3.2.5. Definíció. *Egy 3.2.2 alakú vektor $ORT'_{N'}$-beli, ha léteznek olyan $\phi_k \in I_{j_k}$ számok, melyekre $\sum_{k=0}^{5} e^{2i\pi\phi_k} = 0$.*

Ez az $ORT'_{N'}$ halmaz nem tévesztendő össze az ORT_N halmazzal, bár ahhoz nagyon hasonló, a különbség mindössze annyi, hogy az összes 3.2.2 alakú vektor egy „kis" részhalmazát alkotja (nem csak 2.2.1 alakúakat tartalmaz, mint ORT_N), mégpedig azokból áll, amelyek, mint tartomány, tartalmaznak olyan komplex vektort, amely ortogonális az $(1,1,1,1,1,1)$ komplex vektorra.

Az ORT_N előállításánál elmondottakhoz teljesen hasonlóan következik a 3.2.2. lemmából, hogy ha $\hat{u} \in ORT'_{N'}$ akkor

$$\left|\sum_{k=0}^{5} e^{2i\pi r_{j_k}}\right| \le \pi \sum_{k=0} 6\frac{1}{N'} = \frac{6\pi}{N'}. \tag{3.2.5}$$

3.2.6. Megjegyzés. *Az ORT_N halmaz előállításánál leírt módszert, mellyel a vektor „leszármazottait" is vizsgáljuk, itt is érdemes használnunk, mert nagymértékben csökkenti az $ORT'_{N'}$ halmaz méretét. A különbség annyi, hogy mivel ebben az esetben az első koordináta sem ismert pontosan, az első koordinátának megfelelő intervallumot is osztogatnunk kell, így 32 helyett 64 „gyerekvektor" keletkezik minden vektorból.*

3.2.7. Megjegyzés. *Az $ORT_{N,mon}$ halmazhoz hasonlóan definiálhatjuk az $ORT'_{N',mon}$ halmazt. Itt is igaz, hogy a 3.2.5. feltétel invariáns a koordináták permutálására, itt azonban mind a 6 koordinátát permutálhatjuk. Az $ORT'_{N',mon}$ halmazba így azon $ORT'_{N'}$-beli vektorok tartoznak, melyek*

koordinátái monoton növőek. Az általánosság megszorítása nélkül feltehetjük, hogy a $\hat{H}_2$ mátrix második oszlopa ebből a halmazból kerül ki, mert szabadon átrendezhetjük a sorokat, hogy ez teljesüljön.

3.2.8. Megjegyzés. *$ORT'_{N'}$ halmaz előállítása ezek után úgy megy, hogy a 3.2.5. feltételt vizsgáljuk az összes lehetséges monoton növő diszkretizált vektorra, illetve azok „leszármazottaira" 5-10 generáción keresztül, és ha minden generációban találunk olyan „leszármazottat, amelyik teljesíti a feltételt, akkor az adott vektor $ORT'_{N',mon}$-beli. Ezután e halmaz elemeinek permutálásával kapjuk $ORT'_{N'}$ halmazt. Ez utóbbi halmaz mérete várakozásaink szerint kb. N'-szöröse lesz ORT_N halmazénak, konkrétan N'=19 esetén $\approx 2 \cdot 10^6$ lesz az általunk megírt kód alapján.*

Láttuk, hogy $\hat{H}_2$ második oszlopa $ORT'_{N',mon}$-ből kerül ki, míg a többi oszlop, (az első csupa 0 oszlopot leszámítva) $ORT'_{N'}$-ből. Ezek a feltételek biztosítják azt, hogy az oszlopvektorok ortogonálisak legyenek az első oszlopra. További feltétel, hogy ortogonálisak kell, hogy legyenek egymásra is, ezt szintén a korábban már látott módon vizsgáljuk, azaz két N'-diszkretizált vektor N'-ortogonális egymásra, ha „tartalmaznak" olyan vektort, amik ortogonálisak egymásra. Ezt a tulajdonságot formálisan így fogalmazhatjuk meg:

3.2.9. Definíció. *Azt mondjuk, hogy az $u = (j_0, j_1, j_2, j_3, j_4, j_5)$ és $v = (m_0, m_1, m_2, m_3, m_4, m_5)$ vektorok N'-ortogonálisak egymásra, ha léteznek olyan $\phi_k \in I_{j_k}$ és $\psi_k \in I_{m_k}$ számok, hogy $\sum_{k=0}^{5} e^{2i\pi(\phi_k - \psi_k)} = 0$.*

Az $ORT_{N,eps}$ halmaz definiálásánál elmondottakkal teljesen analóg módon itt is igaz, hogy az eltolásinvariancia miatt u és v vektorok modulo N' vett különbségvektorát kell vizsgálnunk, ha azt akarjuk eldönteni, hogy u és v vektorok N'-ortogonálisak-e egymásra. Itt is igaz továbbá, hogy ez a vizsgálat azt jelenti, hogy az imént meghatározott különbségvektornak a 3.2.9. definíció szerinti értelemben N'-ortogonálisnak kell lennie a csupa 0 vektorra, és itt fontos különbség, hogy mind a hat 0 azt jelenti, hogy az adott koordináta az I_0 intervallumba esik. Az $ORT'_{N',eps}$ halmazt is a korábbiakkal összhangban úgy definiálhatjuk, hogy $ORT'_{N'}$ halmazból indulunk ki, és minden vektorához definiálunk itt 32 helyett 64 vektort, melyek mindegyikének koordinátái vagy megegyeznek az eredeti vektor koordinátáival, vagy azoknál 1-gyel nagyobbak modulo N', ezután $ORT'_{N',eps}$ ezen utóbbi vektorokból áll. Ez a definíció formálisan:

3.2.10. Definíció. *A fentiek miatt $ORT'_{N',eps}$ halmaz mindazon $u^{\underline{\epsilon}} = (j_0 + \epsilon_0, j_1 + \epsilon_1, \ldots, j_5 + \epsilon_5)$ vektorokból áll, ahol ϵ_k 0 vagy 1 és a $(j_0, j_1, j_2, j_3, j_4, j_5)$ vektor $ORT'_{N'}$-beli.*

Az előállítás módja alapján azt várnánk itt is, hogy $ORT'_{N',eps}$ halmaz mérete $ORT'_{N'}$ halmazénak mintegy 64-szerese lesz, de itt is azt mutatja a tapasztalat, hogy csak mintegy 4-szeres szorzót kapunk, ugyanis $ORT'_{N',eps}$ mérete $\approx 9 \cdot 10^6$-nek adódik N'=19-nél.

Az előbbiek alapján tehát azt állapítottuk meg, hogy szükséges feltétel még, hogy $\hat{H}_2$ mátrix oszlopainak modulo N' képzett különbségvektorai (10 db összesen) $ORT'_{N',eps}$-beliek legyen. Összefoglalva az oszlopok ortogonalitásából az alábbi három feltétel adódik $\hat{H}_2$ mátrixra:

- A második oszlopvektor $ORT'_{N',mon}$-beli.
- A 3.-6. oszlopvektorok $ORT'_{N'}$-beliek.
- A 2.-6. oszlopvektorok modulo N' képzett különbségvektorai $ORT'_{N',eps}$-beliek.

Az iménti feltételek vizsgálata az $ORT'_{N',mon}$, $ORT'_{N'}$ és $ORT'_{N',eps}$ halmazok ismeretében egyszerű, egy adott $\hat{H}_2$ mátrix oszlopaira kell ellenőrizni, hogy maguk, illetve a különbségvektorok

a megfelelő halmazba esnek-e. Arról, hogy egy ilyen vizsgálatot hogyan lehet hatékonyan elvégezni, már a 3.1. szakaszban szót ejtettünk, és láthatjuk, hogy ezek a halmazok, nagyobb méretük miatt, a bool-tömbös módszerrel keresésnél az ORT_N-énál egy nagyságrenddel nagyobb memóriaigénnyel bírnak.

A $\hat{H}_2$ mátrix tényleges összeállítása során hasznos lehet, ha nem csak a már összeállított mátrixot vizsgáljuk, hanem amint kiválasztottunk néhány sort, megvizsgáljuk, hogy kiegészíthető-e még az a néhány sor teljes mátrixszá úgy, hogy az megfeleljen az ortogonalitási feltételeknek. Ez a következőt jelenti: tegyük fel, hogy lerögzítettünk 3 sorvektort. Ekkor ismert az összes oszlopvektor első három koordinátája. Most megvizsgálhatjuk például azt, hogy az $ORT'_{N',mon}$ halmazban létezik-e olyan vektor, amelynek első három koordinátája megegyezik a második oszlopvektor immár rögzített első három koordinátájával. Ha nem, akkor biztos, hogy az adott három sor nem egészíthető ki megfelelő $\hat{H}_2$ mátrixszá, így további vizsgálata szükségtelen. Ugyanezt megtehetjük a többi oszlop első három koordinátájával és az $ORT'_{N'}$ halmazzal, valamint a különbségvektorok első három koordinátájával és az $ORT'_{N',eps}$ halmazzal, akár minden sor kiválasztása után is, mindazonáltal a tapasztalat azt mutatja, hogy az első sor, és így az első koordináták rögzítése még biztosan nem zárja ki az ortogonalitási feltételek teljesítését, mert sok vektor van ezen halmazokban.

3.2.11. Megjegyzés. *Azt, hogy például létezik-e olyan vektor $ORT'_{N',mon}$ halmazban, melynek első három koordinátája egy adott számhármas, úgy ellenőrizhetjük hatékonyan, hogy előre meghatározzuk ezen 3 dimenziós vektorokat, azaz azon vektorokat, melyek, mint prefix előfordulnak $ORT'_{N',mon}$ halmazban, és ezen halmazon alkalmazzuk a 3.1. szakaszban ismertetett bool-tömbös módszert. Ez nyilván alkalmazható több koordináta ismeretében és a többi halmazra is, és a memóriaigény a jóval kisebb alaphalmazok miatt elenyésző a hatdimenziós esetekhez képest.*

Az előbbiekben taglalt vizsgálatot is megvalósítottuk a $\hat{H}_2$ mátrixokat előállító kódban. A tapasztalatok szerint a második oszlopvektorok szűrése 10-15%-nyi mátrixot szűr ki, a többi oszlop vizsgálata tovább felezi a megmaradó mátrixok számát, míg a különbségvektorok vizsgálatával végül összességében az ortogonalitási feltételek vizsgálatának mellőzésével előálló $\hat{H}_2$ mátrixoknak mintegy egytizede marad meg, azaz N=53 és N'=19 esetén $\approx 10^9$ db $\hat{H}_2$ mátrixot találunk. A futási idő közben jelentősen megnőtt, kb. kétszeresére, mivel a sok ellenőrzés elvégzése sok időt vesz igénybe.

Kipróbáltunk más N' értékeket is, de azt láttuk, hogy N' növelésével növekszik $UB_{\hat{H}_1}$ mérete, és erre nagyon érzékeny a $\hat{H}_2$ mátrix-előállító algoritmus, mind az előálló mátrixok száma, mind az előállításhoz szükséges idő $UB_{\hat{H}_1}$ méretének kb. 5.-6. hatványával arányos.

3.3. $\hat{H}_3$ mátrix keresése és ellentmondás elérése

Az előző szakaszban megismerhettük, hogy hogyan állítunk elő $\hat{H}_2$ mátrixokat. Ebben a fejezetben $\hat{H}_3$ mátrixokat próbálunk előállítani, illetve mivel hisszük, hogy az 1.1.3. sejtés igaz, azt szeretnénk látni, hogy egy négyelemű kölcsönösen torzítatlan bázishalmaz léte ellentmondásra vezet, azaz nem sikerül a feltételeknek megfelelő $\hat{H}_3$ mátrixot előállítani.

A 2.4. szakaszban láttuk, hogy $\hat{H}_3$ mátrix sorai az $UB_{\hat{H}_1,\hat{H}_2}$ halmazból kerülnek ki, ami azokat a vektorokat tartalmazza, melyek mind $\hat{H}_1$, mind $\hat{H}_2$ mátrix soraira torzítatlanok, ráadásul „univerzálisan”, azaz léteznek olyan „leszármazottaik”, melyek mind a 12 sorvektorra egyszerre torzítatlanok. A következő feladat tehát $UB_{\hat{H}_1,\hat{H}_2}$ halmaz előállítása. Ez azonban egyszerű, hiszen csak arról van szó, hogy az $UB_{\hat{H}_1}$ halmazt előállító algoritmust kicsit módosítanunk kell, mégpedig

annyiban, hogy a $\hat{H}_2$ soraival vett torzítatlanságot is vizsgálja, valamint futtatási időt spórolandó nem kell mind az N'^5 szóbajöhető vektor között keresnünk, elég csak $UB_{\hat{H}_1}$ elemeit megvizsgálni, hiszen nyilván $UB_{\hat{H}_1,\hat{H}_2} \subset UB_{\hat{H}_1}$. Itt is megjegyezzük, hogy az $UB_{\hat{H}_1,\hat{H}_2}$ halmazt előállító algoritmusban nem számít, hogy $\hat{H}_1$ és $\hat{H}_2$ esetleg más N illetve N' paraméterrel van diszkretizálva.

Amennyiben az $UB_{\hat{H}_1,\hat{H}_2}$ halmazt meghatároztuk, akkor onnantól teljesen ugyanaz a dolgunk, mint amikor a $\hat{H}_2$ mátrixot próbáltuk előállítani $UB_{\hat{H}_1}$ elemeiből, azaz teljesen ugyanazon feltételeknek kell eleget tennie a soroknak illetve az oszlopoknak. Az ellentmondást akkor érjük el, ha nem sikerül ezen feltételeknek megfelelő $\hat{H}_3$ mátrixot előállítanunk.

A megvalósított $UB_{\hat{H}_1,\hat{H}_2}$ halmazt előállító kódot kísérletképp jóval nagyobb, N=101, illetve N=149 értékek, valamint N'=19 paraméterekkel vizsgáltuk. Ekkor számíthattunk arra, hogy „emberi” idő alatt lefut a kód. Konkrétan N=149 esetén $UB_{\hat{H}_1}$ mérete ≈ 400 volt, és ekkor $\approx 4 \cdot 10^5$ db $\hat{H}_2$ mátrixot találtunk. Fontos látni, hogy lehetőségünk nyílik arra, hogy az $UB_{\hat{H}_1,\hat{H}_2}$ halmazt előállító kód paramétereit (ezen paraméterekre később visszatérünk még) úgy állítsuk be, hogy az esetek túlnyomó többségében $UB_{\hat{H}_1,\hat{H}_2}$ mérete kisebb legyen, mint 6, ebben az esetben ugyanis rögtön ellentmondásra jutunk, hiszen legalább 6 vektorra szükségünk van ahhoz, hogy $\hat{H}_3$ mátrixot össze tudjuk állítani.

Az $UB_{\hat{H}_1,\hat{H}_2}$ halmaz elemeit előállító kódnak három paramétere is van. Az első paraméter az, hogy milyen N' diszkretizációs paramétert akarunk használni az előállítandó vektoroknál. A második paraméter az az, hogy a „leszármazottakat” milyen mélységig, azaz hány generációig akarjuk megvizsgálni. Láttuk, hogy ezen paraméter növelésével először drasztikusan csökken a megmaradó vektorok száma, majd 5-10-es mélységet elérve már alig változik, miközben a futtatáshoz szükséges idő viszont drasztikusan megnő. Míg $UB_{\hat{H}_1}$ előállításakor az 5-10-es mélység tűnt ideálisnak, addig $UB_{\hat{H}_1,\hat{H}_2}$ előállításakor azt tapasztaltuk, hogy már 2-3-as mélységet beállítva elérhetjük, hogy az esetek túlnyomó többségében 6-nál kevesebb vektor maradjon, azaz elérjük az ellentmondást. Azért fontos ez, mert ilyen kis mélységnél rendkívül gyorsan lefut a keresés. A fent említett $\approx 4 \cdot 10^5$ esetből 3-as mélységet beállítva csak ≈ 100 esetben lesz $UB_{\hat{H}_1,\hat{H}_2}$ mérete legalább 6, és ezt az eredményt 2 percnyi futási idő alatt megkapjuk. Ebben a megmaradó ≈ 100 esetben többféleképp is megpróbálhatjuk elérni az ellentmondást, kézenfekvőnek tűnik az, hogy nagyobb mélységgel is megpróbáljuk előállítani $UB_{\hat{H}_1,\hat{H}_2}$ halmazt, bízva abban, hogy ilyen beállításokkal már 6-nál kevesebb vektor marad csak, míg egy másik megoldás lehet az, hogy $UB_{\hat{H}_1,\hat{H}_2}$ halmazban az $UB_{\hat{H}_1}$ halmazhoz hasonlóan 6-klikkeket keresünk, mindenesetre a vektorok kis száma miatt mindkét módszer, esetleg ezek kombinációja biztató eredményekkel kecsegtet.

Az algoritmus harmadik paramétere lehet az, hogy hány vektort veszünk $\hat{H}_2$ mátrixból, mert előfordulhat, hogy 6-klikkek egy része tartalmaz például azonos 4-klikket, és ha igaz az, hogy $\hat{H}_1$ mátrix sorai mellé csak ezen 4-klikk sorait illesztve, és ezekhez „univerzálisan” torzítatlan vektorokat keresve már nem kapunk 6-nál több vektort, akkor az összes, ezt a 4 vektort tartalmazó $\hat{H}_2$ mátrixot elvethetjük. A tapasztalat azt mutatja, hogy a $\hat{H}_1$ mátrix soraihoz illesztendő vektorok számát növelve csökken a megmaradó vektorok száma (nyilván, hiszen egyre erősebb feltételnek kell megfelelniük), és első pillantásra meglepő módon csökken a megtalálásukhoz szükséges idő. Ez utóbbinak az a magyarázata, hogy az algoritmus futási idejének legnagyobb részét a végül megmaradó vektorokban tölti, mert ezesetben egy olyan leszármazottat talál, ami minden generációban rendelkezik olyan „leszármazottal”, amelyik teljesíti az N'-ortogonalitási feltételeket, ám ha ez a „leszármazott”, illetve az N'-ortogonalitást biztosító komplex értékek a kicsi ívrészek második felében találhatóak, akkor sok vizsgálatot kell végrehajtani, mielőtt megtaláljuk a megfelelő, az N'-ortogonalitási feltételt teljesítő vektort. Az algoritmus olyan, hogy minden „leszármazottat”

minden $\hat{H}_1$ és $\hat{H}_2$-beli vektorral tesztel, így amennyiben egy adott „leszármazott” legalább egy vektorral nem N'-ortogonális, akkor az adott „leszármazottat” rögtön nem vizsgáljuk tovább, eldobjuk, így az ő leszármazottainak vizsgálata nem igényel több futásidőt. Az előbbiek miatt minél több vektort veszünk $\hat{H}_2$-ből, annál kevesebb lesz mind a megmaradó vektorok száma, mind a futáshoz szükséges idő.

3.3.1. Megjegyzés. *A fenti megállapításokat a következő mért adatok támasztják alá: N'-t 35-nek választva $UB_{\hat{H}_1}$ halmaz mérete $\approx 6 \cdot 10^2$ lesz ebben $\approx 6 \cdot 10^6$ db 6-klikket találunk. A 6-klikkek keresése közben vizsgáltam a 3-, 4-, 5-klikkek számát is, belőlük rendre $6 \cdot 10^4$, $7 \cdot 10^5$ és $2,5 \cdot 10^6$ darabot találtunk. (Itt mégegyszer megjegyzem, hogy ezek nem az összes 3-, 4- és 5-klikkek számát jelenti, hanem azokét, amelyek a háromszögek alapján potenciálisan kiegészíthetőek 6-klikké.) Megvizsgáltuk azt, hogy $\hat{H}_1$ mátrix sorai mellé az adott 3-, 4-, 5- illetve 6-klikk vektorait beillesztve és az univerzális $UB_{\hat{H}_1,\hat{H}_2}$ keresőt kettes mélységgel futtatva mennyi a futásidő, illetve hány esetben találunk legalább 6 torzítatlan vektort. Az eredmények: a futásidő rendre kb. 10 másodperc, 1 perc, 5 perc illetve 10 perc, a megmaradó esetek pedig rendre $3 \cdot 10^4$, $3 \cdot 10^4$, $4 \cdot 10^3$ illetve 10^2. Tehát látható, hogy a szóbajöhető 3-klikkek felénél már maga a 3-klikk kizárja, hogy össze tudjuk állítani $\hat{H}_3$ mátrixot, és ez ráadásul viszonylag gyorsan ki is derül, látszik az is továbbá, hogyha növeljük a tesztelendő klikkek méretét, akkor rohamosan csökken a megmaradó esetek száma. Ez alapján kirajzolódhat egy adaptív algoritmus, amelynél megvizsgálunk minden 3-klikket, hogy a $\hat{H}_1$ mátrix soraival együtt vizsgálva találunk-e legalább 6 rájuk torzítatlan vektort, majd csak azokat az eseteket vizsgáljuk tovább, melyeknél igen a válasz, itt minden 4-klikket megvizsgálunk, akár csak az imént meghatározott torzítatlan vektorokat vizsgálva, majd ugyanúgy csak azokat az eseteket vizsgáljuk, melyeknél marad elegendő torzítatlan vektor, ugyanezt tesszük az 5-klikkekkel, majd végül a megmaradó 6-klikkekkel is. Az utolsó szűrőn is átmenő rendkívül kevés 6-klikket szűrhetjük úgy, hogy a torzítatlan vektorokat nagyobb mélységgel keressük, illetve az előző szakaszban ismertetett módon kereshetünk 6-klikkeket az előálló kisméretű $UB_{\hat{H}_1,\hat{H}_2}$ halmazban. A fenti adatok alapján egy ilyen kereső algoritmus egy $\hat{H}_1$ mátrixnál $UB_{\hat{H}_1}$ meghatározása után kb. 1 percnyi futásidő alatt éri el az ellentmondást. Mivel azonban az $N=149$-es érték mellett rendkívül sok $\approx 10^{12}$ db lehetséges $\hat{H}_1$ mátrix van, ez még mindig nagyon sok futásidő összességében.*

A teljes algoritmus legkényesebb pontja az $UB_{\hat{H}_1}$ halmaz mérete, mivel a 6-klikkek száma ezen halmaz méretének növekedésével ugrásszerűen (kb. 5. hatványával arányosan) megnő. Pont ezen halmaz nagy mérete miatt kell N-et nagyon nagyra választanunk, ami viszont a lehetséges $\hat{H}_1$ mátrixok számát növeli meg nagymértékben. Nagyon fontos cél lenne tehát az $UB_{\hat{H}_1}$ halmaz méretének csökkentése, mert itt egy kisarányú csökkentés is nagy futásidő-csökkenéssel jár a fentiek miatt.

3.4. A vektorok „csoportosítása”

Az eddigiek során még nem tértünk ki az $UB_{\hat{H}_1}$ illetve az $UB_{\hat{H}_1,\hat{H}_2}$ halmazok belső struktúrájára. A kódok ellenőrzése során az 1.2.3. szakaszban ismertetett néhány Hadamard mátrixot diszkretizáltunk különböző N diszkretizációs paraméterekkel, majd előállítottuk $UB_{\hat{H}_1}$ halmazt különféle N' diszkretizációs paraméterek mellett. Néhány esetben numerikusan ismertek az adott Hadamard mátrixra torzítatlan vektorok, így egy jó teszt volt az, hogy vajon az ismert torzítatlan vektorok diszkretizált megfelelői mind megtalálhatóak-e a megfelelő $UB_{\hat{H}_1}$ halmazban. Ezen túlmenően felmerül a kérdés, hogy vajon a többi, a torzítatlanság nem pontos meghatározása miatt bekerü-

lő vektor (hívjuk ezen vektorokat téves vektoroknak) miért is kerül be pontosan, valamint ezen vektorok hogyan viszonyulnak a tudottan tényleg torzítatlan vektorokhoz (hívjuk őket ezentúl helyes vektoroknak). A tapasztalat azt mutatja, hogy a téves vektorok egyrészt a helyes vektorok „közelében” helyezkednek el, másrészt kisebb számban előfordulnak olyan téves vektorok is, amelyek „közelében” nincs helyes vektor. Mindazonáltal jellemző a halmazra, hogy a vektorok eloszlása „csomósodást” mutat, azaz szemmel elkülöníthetőek olyan csoportok, amelyeken belül a vektorok „közel” vannak egymáshoz. Pontosítsuk egy kicsit a vektorok távolságának fogalmát.

3.4.1. Definíció. *Legyen u és v két 2.2.1. alakú vektor: $u = (0, j_1, j_2, j_3, j_4, j_5)$ és $v = (0, m_1, m_2, m_3, m_4, m_5)$. A következő összeget nevezzük u és v összkoordináta-távolságának, és jelöljük $d(u, v)$-vel:*

$$d(u,v) = \sum_{k=1}^{5} min\{j_k - m_k \ mod \ N', m_k - j_k \ mod \ N'\} \tag{3.4.1}$$

3.4.2. Megjegyzés. *A fenti definíció kevésbé formálisan azt jelenti, hogy vesszük koordinátánként a koordináták távolságát, és ezeket összeadjuk, ügyelve arra, hogy ezek a koordinátaértékek az egységkörön akkor is közel lehetnek egymáshoz, ha a komplex 1 érték közöttük helyezkedik el, azaz mondjuk a 0 és az $N' - 1$ koordinátaértékek távolsága 1.*

3.4.3. Megjegyzés. *Ez a fajta távolság összhangban van a komplex vektorterekben szokásos euklideszi távolságfogalommal, azaz ha két komplex vektor euklideszi távolsága kicsi, akkor diszkretizáltjaik 3.4.1. definíció szerinti távolsága is kicsi lesz.*

A további vizsgálatokhoz vezessünk be néhány jelölést. Korábban már $G_{\hat{H}_1}$ jelölte a csoportosítatlan $UB_{\hat{H}_1}$-beli vektorokon definiált gráfot (emlékeztetőül: akkor ment két csúcs között él, ha a nekik megfelelő vektorok N'-ortogonálisak voltak egymásra). Jelölje $'$ a csoportosítást. Ekkor ezzel a jelöléssel $UB'_{\hat{H}_1}$ jelölje a csoportosított vektorok csoportjainak halmazát. Legyenek $UB'_{\hat{H}_1}$ elemei a $V'_1, \ldots, V'_l$ csoportok - amik lényegében $UB_{\hat{H}_1}$ részhalmazai - tehát l db csoportunk van. Hivatkozzunk az i-edik csoport elemeire a $v'_{i,1}, \ldots, v'_{i,m}$ jelölésekkel, amik mind egy-egy $UB_{\hat{H}_1}$-beli vektort jelentenek. Ekkor egy adott $UB_{\hat{H}_1}$ halmazon egy adott csoportosítást definiál a $V'_1, \ldots, V'_l$ halmazok összessége, melyekre igaz, hogy diszjunktak és uniójuk kiadja a teljes $UB_{\hat{H}_1}$ halmazt. A csoportosítás definíciója ezek után:

3.4.4. Definíció. *A $V'_1, \ldots, V'_l$ halmazokat $UB_{\hat{H}_1}$ halmaz egy csoportosításának mondjuk, ha:*

- *minden $1 \leq i, j \leq l$-re $V'_i \bigcap V'_j = \varnothing$*
- $\bigcup V'_i = UB_{\hat{H}_1}$

A csoportosítás definiálása után - ahhoz hogy 6 klikkeket tudjunk keresni a csoportokon - éleket kell definiálnunk a csoportokon értelmezett gráfon. Jelölje $G'_{\hat{H}_1}$ az $UB'_{\hat{H}_1}$ halmaz elemein (vektorhalmazok az elemek) definiált gráfot, tehát a gráf csúcsai a $V'_1, \ldots, V'_l$ halmazok. Most mindenképpen az a célunk (az éleket így akarjuk behúzni), hogy minden $G_{\hat{H}_1}$-ben található 6-klikkre igaz legyen az, hogy ha a 6-klikk csúcsai $v_1, \ldots, v_6$ vektorok, akkor létezzen $G'_{\hat{H}_1}$-ben olyan $V'_{j_1}, \ldots, V'_{j_6}$ 6-klikk, melyre minden i-re $v_i \in V'_{j_i}$. Ehhez elengedhetetlen, hogy minden v_i más V'_{j_i} halmazba essen, ezt úgy biztosíthatjuk, ha minden olyan v_i és v_k pár, melyek között él megy $G_{\hat{H}_1}$-ben, külön V'_{j_i} és V'_{j_k} halmazokba kerül. Ha az előbbi feltétel teljesül, és az éleket az alábbi módon húzzuk be, akkor nyilvánvalóan elérhetjük a fenti célt:

3.4.5. Definíció. *V'_i és V'_j csoportok pontosan akkor ortogonálisak (azaz megy közöttük él $G'_{\hat{H}_1}$-ben), ha léteznek $v'_{i,k}$ és $v'_{j,m}$ vektorok (értelemszerűen rendre V'_i-ben és V'_j-ben), melyek N-ortogonálisak, (azaz közöttük megy él $G_{\hat{H}_1}$-ben).*

Mielőtt a tényleges csoportosítási módszerekkel megismerkednénk, vizsgáljuk meg, hogy mit is várunk a csoportoktól, mi a célunk a csoportosítással. Az egész csoportosítási ötlet onnan indul ki, hogy $UB_{\hat{H}_1}$ vektorai „csomósodnak”, azaz elkülöníthetőek szemmel olyan csoportok, amelyeken belül egymáshoz közeli vektorok vannak. Megfigyeltük továbbá azt is, hogy a 6-klikkek is „csomósodnak”, mégpedig olyan értelemben, hogy azért van rendkívül sok belőlük, mert rengeteg olyan 6-klikk párt lehet találni, melyek csak néhány koordinátában térnek el, és az eltérő koordinátákban is kicsi a különbség. A cél az, hogy ezeket a kicsit különböző 6-klikkeket ne kelljen egyesével vizsgálnunk, hanem valamilyen módon egyszerre tudjuk őket kezelni, mint ahogy a csoportosított vektorokat egyszerre kezeljük, amikor 6-klikkeket keresünk $G_{\hat{H}_1}$ gráfban. Ezt azért tudjuk megtenni, mert az egymáshoz közel lévő vektorok „ugyanúgy viselkednek”, azaz nagyjából ugyanazon vektorokra lesznek torzítatlanok és ortogonálisak. Az, hogy két vektor ortogonális vagy torzítatlan egymásra, csak a skaláris szorzatuktól függ, ami egy folytonos függvény, és mint ilyen, igaz rá, hogy egy rögzített v vektorra, ha u' adott u érték közelében van, akkor az $\langle u', v\rangle$ értékek is $\langle u, v\rangle$ közelében lesznek, így várható, hogy nagyjából ugyanazon vektorokra lesznek torzítatlanok és ortogonálisak az egy csoporton belüli vektorok. Így amikor 6-klikket keresünk $G'_{\hat{H}_1}$-ben, akkor egy megtalált 6-klikk várhatóan azonosít rengeteg $G_{\hat{H}_1}$-beli 6-klikket, mégpedig azokat, amelyek úgy keletkeznek, hogy veszünk minden csoportból egy-egy vektort, és ezen vektorok alkotják a $G_{\hat{H}_1}$-beli 6-klikket. A skalárszorzás folytonossága és a 6-klikkek „csomósodása” miatt várható az, hogy a $G'_{\hat{H}_1}$-beli 6-klikkek a fenti értelemben leírják a $G_{\hat{H}_1}$-beli 6-klikkeket, mindazonáltal nem zárható ki, hogy néhány olyan $G_{\hat{H}_1}$-beli vektorhatost is 6-klikként azonosítunk így, amelyek nem is alkotnak 6-klikket, ám a ez a fajta pontatlanság nekünk „jó” irányú pontatlanság, azaz ha még így is fennáll az ellentmondás, akkor bizonyíthatjuk az 1.1.3. sejtést. Láttuk az imént, hogy fontos, hogy a csoportokon belül ne legyenek N-ortogonális vektorpárok. Később látni fogjuk, hogy az is szükséges lesz, hogy torzítatlan párok se legyenek. Látni fogjuk, hogy az, hogy egy csoportba egymáshoz közeli vektorokat teszünk, egyszerre fogja biztosítani azt, hogy a csoporton belül ne legyenek se ortogonális, se torzítatlan párok, valamint azt is, hogy az egyes csoportok elemei „ugyanúgy viselkedjenek” ortogonalitás és torzítatlanság szempontjából. Nem feledkezhetünk el arról a szempontról, hogy a csoportosításnak gyorsan megvalósíthatónak kell lennie. Az alábbiakban összefoglalom (informálisan „definiálom”) a „jó” csoportosítás ismérveit:

3.4.6. Definíció. *Egy 3.4.4. definíció szerinti csoportosítás akkor felel meg a céljainknak, ha az alábbiak teljesülnek rá (ez nem egy igazi definíció, inkább csak iránymutatás):*

- *Az „ugyanúgy viselkedés” követelménye : az egyes csoportokba olyan vektorok kerüljenek, melyek nagyjából ugyanazon a vektorokra lesznek ortogonálisak illetve torzítatlanok*
- *A „speciális pár-mentesség” követelménye: a csoportokon belül ne legyenek sem ortogonális, sem torzítatlan párok*
- *A csoportosítás műveletigénye minél kisebb legyen - a gyorsaság követelménye*

A 3.4.1. definíció szerinti távolság fogalmával már csoportosíthatjuk a vektorokat. A csoportosítás a következőképp mehet: választunk egy maximális távolságértéket, jelölje ezt t, vesszük sorban a vektorokat a lexikografikusan rendezett halmazukból. Az első vektornak nyitunk egy első csoportot. Minden utána következő u vektorral a következőt tesszük: vesszük sorban most a csoportokat, és minden csoportban vesszük a már csoporttag vektorokat. Ha találunk olyant, aminek távolsága u vektortól nem nagyobb, mint t, akkor u vektort az adott csoportba tesszük, és nem vizsgálódunk tovább. Ha nem találunk ilyen csoportot, akkor u vektornak nyitunk egy új csoportot. Az

egész processzus lezárásaként összevonunk minden olyan csoportpárt egy csoporttá, amelyekben van olyan vektorpár, hogy a vektorpár egyik tagja az egyik, a másik tagja a másik csoportban van, és távolságuk nem nagyobb, mint t. Mivel a lexikografikusan rendezett sorban az egymáshoz közeli sorszámú vektorok sok esetben közel vannak egymáshoz a 3.4.1. definíció szerinti távolságfüggvény értelmében, ez a fajta csoportosítás jól megválasztott t esetén egy elég jó csoportosítást fog biztosítani, ráadásul a csoportosítás időigénye eléggé minimális. Az így kapott csoportosítás mindenesetre eléggé esetleges, sem a csoportok létszámát, sem a csoportok számát nem tudjuk kontrollálni, és lényegében csak annyit mondhatunk a csoportokban lévő vektorokról, hogy ha két vektor (a és b) egy csoportban van, akkor létezik olyan csoportbeli vektorsorozat, amelynek első eleme a, utolsó eleme b, és a sorozat bármely két szomszédos vektorára igaz, hogy távolságuk nem nagyobb, mint t. A csoportosítás végén végrehajtott összevonások miatt igaz az előző állítás megfordítottja is, azaz ha a és b olyan $UB_{\hat{H}_1}$-beli vektorok, melyekhez létezik olyan $UB_{\hat{H}_1}$-beli $c_1, \ldots, c_n$ vektorsorozat, melyre $a{=}c_1$ $b{=}c_n$ és minden $1 \leq i \leq n-1$-re $\mathbf{d}(c_i, c_{i+1}) \leq t$, akkor az összes c_i, és így speciálisan a és b is egy csoportba esnek. Vizsgáljuk meg, hogy mennyiben felel meg ez a csoportosítás a 3.4.6. definícióban leírtaknak: a tapasztalat azt mutatja, hogy amiatt, hogy az „ugyanúgy viselkedés” követelményét jól teljesíti ez a fajta csoportosítás a skalárszorzás folytonossága miatt, míg a „speciális pár-mentesség” követelményének teljesítését a csoportosítás esetlegessége miatt semmilyen matematikailag megfogható körülmény nem segíti elő. Mindazonáltal egyrészt a tapasztalat azt mutatja, hogy t-t nagyon nagyra kell ahhoz választani, hogy megjelenjenek ilyen párok, valamint a módszer kis módosításával egyszerűen biztosítható, hogy a csoportok teljesítsék ezt a követelményt: amikor egy adott vektort egy csoporthoz akarunk hozzáadni, akkor a távolságok mellett megvizsgáljuk azt is, hogy az éppen beteendő vektor nem okozza-e „speciális párok” megjelenését, és amennyiben igen, akkor új csoportot nyitunk neki. A gyorsasági feltételt nyilván teljesíti a módszer, hiszen a csoportosításhoz kevés műveletet kell elvégezni.

3.4.7. Megjegyzés. *Érdemesnek tűnik megvizsgálni, hogy egy adott konkrét esetben, amikor numerikusan ismertek $UB_{\hat{H}_1}$ elemei, illetve a belőlük előállítható H_2 mátrixok, akkor milyen viszonyban vannak egymással a csoportosítással előálló csoportok és a tényleges $UB_{\hat{H}_1}$-beli vektorok, valamint ami még érdekesebb, hogy a numerikusan ismert H_2 mátrixok milyen viszonyban vannak a csoportokon definiált gráfban megtalált 6-klikkekkel. Egy konkrét, numerikusan ismert mátrixhoz ismerjük numerikusan a torzítatlan vektorokat, összesen 72 van belőlük. Ugyanezen mátrix N-diszkretizáltjához $N{=}149$-nél 452 különböző $N'{=}19$ másodlagos diszkretizációs paraméterű $UB_{\hat{H}_1}$-beli vektort kapunk. $t{=}1$-re pont 72 csoportot kapunk a fent leírt algoritmussal. Felmerül a gondolat, hogy ez a 72 csoport pont megfelelne a 72 igazi, diszkretizálatlan torzítatlan vektornak abban az értelemben, hogy minden csoporthoz pontosan egy olyan diszkretizálatlan vektort találunk, amelynek diszkretizáltja az adott csoportban található. Ez azonban nincs így, ugyanis léteznek az adott esetben olyan csoportok, amelyekhez nem található ilyen diszkretizálatlan vektor, és olyan is, amelyekhez több is található. Fontosabb kérdés azonban, hogy minden olyan 6-klikket megtalálunk-e, amik a diszkretizálatlan vektorokból előállíthatóak, hiszen ebben semmiképp nem szabad tévednünk. Ebben a konkrét esetben azt tapasztaljuk, hogy míg a diszkretizálatlan vektorokból 36 különböző H_2 mátrixot lehet összeállítani, addig a diszkretizált vektorok csoportjain csak 28-at. Ez nem jelenti azonban azt, hogy ne kapnánk meg valamilyen formában mind a 36 eredeti H_2 mátrixot, ugyanis egyszerűen arról van szó, hogy a 72 numerikusan ismert vektor között vannak olyanok, amelyek egyrészt egy csoportba kerülnek, másrészt a 36 H_2 mátrix némelyikében elő is fordulnak, így a 28 diszkretizált, csoportosított vektorokon értelmezett mátrix némelyike többet is „lefed” a 36 H_2 mátrix közül.*

Az iménti csoportosítás javítható úgy, hogy a csoportba tartozás feltételeként nem azt szabjuk, hogy létezzen t-nél közelebbi vektor, hanem azt, hogy egy csoporton belül minden vektorpár közelebb legyen egymáshoz, mint t. Azaz minden V_i' csoportra és minden $j \neq k$-ra $d(v'_{i,j}, v'_{i,k}) \leq t$. Ez a csoportosítás annyiban jobb, hogy kizárja annak a lehetőségét, hogy egymástól viszonylag távol lévő vektorok egy csoportba kerüljenek csak azért, mert létezik olyan, őket összekötő (akármilyen hosszú) vektorsorozat, amelynek szomszédos elemei ugyan közel vannak egymáshoz, de a két végpontról már nem mondható ez el.

Vizsgáljuk meg erre a csoportosításra a 3.4.6. követelmények teljesülését. Az „ugyanúgy viselkedés" követelményét sokkal inkább teljesítik az így előálló csoportok, mint az előző módszernél, mert kizárjuk az egy csoporton belüli nagy távolságokat, és a módszer sem annyira esetleges már. A „speciális pár-mentesség" követelményeire továbbra sem tudunk elméleti megfontolásokat mutatni, így itt is marad az előző módszernél alkalmazott, a pár-mentességet expliciten biztosító módosítás. A gyorsaságból vesztünk egy keveset, mert itt már a jelölt csoportok minden tagjával össze kell hasonlítanunk a megfelelő vektort, hogy az adott csoportba tartozónak tekinthessük, de ez a tapasztalatok szerint nem nagy áldozat. A tapasztalat azt mutatja, hogy t=3 vagy 4 esetén konkrétan teljesülnek a feltételek, és a csoportok száma töredéke lesz a csoportosítandó vektorok számának.

3.5. Egy ígéretes csoportosítás

Az előző fejezetben már bemutattam két csoportosítást, ám mindkettőnek akadtak hiányosságai. Az egyik hiányosság az volt, hogy t változtatása nem adott elegendő lehetőséget arra, hogy az egyes csoportok méretét csökkentsük (a csoportok száma ekkor növekszik értelemszerűen). Ez azért lesz majd fontos, mert a csoportok mérete fontos szerepet játszik abban, hogy a $G'_{\hat{H}_1}$ gráfbeli 6-klikkek további vizsgálatánál ellentmondásra jussunk, ugyanis kisebb csoportméretnél nagyobb valószínűséggel jutunk ellentmondásra az összes többi paramétert változatlanul hagyva. A másik hiányosság az volt, hogy a „speciális pár-mentesség" követelményét mindig csak explicit tudtuk biztosítani, azaz a csoportok összeállításakor külön vizsgálnunk kellett ezen feltételek teljesülését. A tapasztalat ugyan azt mutatta, hogy csak rendkívül nagy csoportoknál fordul elő ilyen pár (ez arra utal, hogy valami elméleti ok miatt nem fordulnak elő kisméretű csoportokban), ám mivel nem áll rendelkezésünkre ezt alátámasztó elméleti ok, kénytelenek vagyunk ennek a vizsgálatával időt tölteni. A harmadik hiányosság az, hogy nem volt eddig szó arról, hogy milyen módon juthatunk ellentmondásra a megtalált $G'_{\hat{H}_1}$ gráfbeli 6-klikkek esetén, és a csoportosítás esetlegessége miatt nem is tűnik egyszerűnek az ellentmondás elérése. Az ebben a fejezetben ismertetésre kerülő módszer kiküszöböli ezeket a hiányosságokat.

Ismerkedjünk meg pár definícióval, melyek a csoportosítási módszernél fontos szerepet játszanak. Tekintsünk vissza először, hogy mit is jelöl pontosan egy 2.2.1 alakú vektor. Legyen a 2.2.1. definíció szerint $u = (0, j_1, j_2, j_3, j_4, j_5)$, ahol j_i-k egészek. Ekkor azt tudjuk, hogy $\rho_i \in [\frac{j_i}{N}, \frac{j_i+1}{N})$. Figyeljük meg, hogy ekkor azt tudjuk, hogy minden ρ_i egy $\frac{1}{N}$ hosszúságú intervallumba esik. Ennek természetes általánosítása az gondolat, hogy megengedjük, hogy ρ_i egy olyan intervallumba essen, melynek hosszúsága $\frac{1}{N}$ valamely többszöröse. Egy ilyen jelölést definiál a szupervektor fogalma:

3.5.1. Definíció. *Legyen egy szupervektor jelölése a következő:* $\tilde{u} = (0, j_1 : j'_1, j_2 : j'_2, j_3 : j'_3, j_4 : j'_4, j_5 : j'_5)$ *A szupervektor jelentése pedig legyen a következő: minden* $1 \leq i \leq 5$*-re* $\rho_i \in [\frac{j_i}{N}, \frac{j'_i+1}{N})$. *Fontos megjegyezni, hogy előfordulhat, hogy* $j_i > j'_i$. *Ekkor a szupervektor adott koordinátájában lévő* $j_i : j'_i$ *jelölés értelme az, hogy vagy* $\rho_i \in [\frac{j_i}{N}, 1)$ *vagy* $\rho_i \in [0, \frac{j'_i+1}{N})$.

3.5.2. Megjegyzés. *A szupervektor jelentése kevésbé formálisan az, hogy olyan 6 dimenziós komplex vektorokat reprezentál, amelyek elemei 1 abszolútértékűek, első koordinátájuk 1, a többiről pedig csak annyit tudunk, hogy a nekik megfelelő ρ_i értékek valamely $(j_i : j_i')$-nek megfelelő olyan intervallumba esnek, melynek hossza $\frac{1}{N}$ valamely egészszámszorosa. Ez azt is jelenti továbbá, hogy ezen komplex koordináta a teljes komplex egységkör egy olyan egybefüggő ívdarabján helyezkedik el, mely a teljes egységkör hosszúságának $\frac{k_i}{N}$-ed részét teszi ki, ahol $k_i = j_i' - j_i + 1 \mod N$. Vegyük észre, hogy ez az ívdarab akkor is egybefüggő lesz, ha $j_i > j_i'$, hiszen ekkor a definíció szerint a két intervallumból alkotott ívdarabok pont az 1 pontban érintkeznek.*

3.5.3. Megjegyzés. *Könnyen látszik, hogy miért általánosítás a szupervektor fogalma: ha ugyanis minden i-re $j_i = j_i'$, akkor visszakapjuk a 2.2.1 alakú vektorokat, hiszen akkor minden intervallum hossza $\frac{1}{N}$ lesz.*

3.5.4. Megjegyzés. *A szupervektorok felfoghatóak a 2.2.1 alakú vektorok egy halmazának is a következő értelemben: vegyük a $j_i : j_i'$ jelölést, illetve azt az intervallumot (intervallumokat), amelyek a 3.5.1. definíció alapján keletkeznek. Vegyük továbbá minden i-re azon $I_{j_i}, I_{j_i+1}, \ldots, I_{j_i'}$ (itt az alsóindexekben az összeadásokat modulo N végezzük, ennek $j_i > j_i'$ esetén van jelentősége) intervallumokat, melyek diszjunkt uniója kiadja a $j_i : j_i'$-nek megfelelő nagy intervallumot (intervallumokat). Ekkor az $\tilde{u} = (0, j_1 : j_1', j_2 : j_2', j_3 : j_3', j_4 : j_4', j_5 : j_5')$ alakú szupervektornak az a V' vektorcsoport felel meg, mely azon $(0, m_1, m_2, m_3, m_4, m_5)$ vektorokból áll, melyekre minden i-re $m_i \in \{j_i, j_i + 1, \ldots, j_i'\}$ (az összeadást itt is modulo N végezzük, így ez a halmaz véges). Vegyük észre továbbá, hogy V' halmaz mérete $\prod_{i=1}^{5} k_i$.*

3.5.5. Definíció. *A korábban már előkerült $k_i = j_i' - j_i + 1 \mod N$ értékeket nevezzük az $\tilde{u} = (0, j_1 : j_1', j_2 : j_2', j_3 : j_3', j_4 : j_4', j_5 : j_5')$ alakú szupervektor koordinátánkénti szélességének.*

A 3.5.4. megjegyzésben a szupervektorhoz kerestünk neki megfelelő vektorcsoportot. A fentiek alapján nyilván minden szupervektornak megfelel pontosan egy vektorcsoport, a fordított állítás viszont nem igaz, így itt csak „burkoló” szupervektor(okat) tudunk definiálni.

3.5.6. Definíció. *Legyen V' egy vektorcsoport. Ekkor V' burkoló szupervektorának nevezzük azokat az $\tilde{u}$ szupervektorokat, melyekre igaz, hogy az $\tilde{u}$-nak megfelelő vektorcsoportnak részhalmaza V' és $\tilde{u}$ mérete minimális.*

3.5.7. Megjegyzés. *Az előző definíció jó, mert az a halmaz, amelyen a minimumot keressük, nemüres, mert mondjuk a $(0, 0 : N, 0 : N, 0 : N, 0 : N, 0 : N)$ szupervektor mindig eleme. Véges, mert csak véges sok szupervektort lehet definiálni. A minimumot a következőképp előálló szupervektorokon veszi fel: minden koordinátában vegyük a halmazbeli vektorokban előforduló koordináták unióját. Ehhez az egész számokból álló halmazhoz határozzuk meg a legkisebb olyan modulo N értelemben egymás melletti számokból álló halmazokat, melyek kivétel nélkül tartalmazzák a koordinátaként előforduló számokat. Ezen intervallumok jelentik a burkoló szupervektor koordinátáit. Könnyen látható, hogy ha a vektorhalmaz vektorainak koordinátájából képzett unióhalmazok modulo N egybefüggőek, vagy létezik olyan, maximum $\lfloor \frac{N}{2} \rfloor$ hosszúságú modulo N értelemben egybefüggő számhalmaz, mely tartalmazza őket, akkor a minimum pontosan egy szupervektoron vétetik fel (mégpedig azon, amely ezekből az egybefüggő intervallumokból, mint koordinátákból épül fel). Mivel a későbbiek során mindig ilyen vektorhalmazokkal lesz dolgunk, az ilyen V' csoportok burkoló szupervektora alatt az előbb meghatározott szupervektort értjük ezentúl.*

A szupervektorok bevezetése azért különösen hasznos, mert ha V_i' csoportokat szupervektorokkal helyettesítjük, akkor könnyebben kezelhető egységeket kapunk, hiszen a szupervektoroknál a

definíciójukból kifolyólag egészen pontos, egyszerű képünk van arról, hogy milyen komplex vektorokat reprezentálnak. Emiatt megfogalmazható rájuk egyszerű feltétel, amelynek teljesülése esetén biztosítható, hogy mint vektorcsoportok speciális pár-mentesek legyenek, ezen feltételeket később meg is fogalmazzuk és be is bizonyítjuk majd.

Definiáljuk akkor a szupervektoros csoportosítást. Az előző szakasz végén ismertetett csoportosításból indulunk ki, azaz minden csoporttól megköveteljük, hogy egy adott t-re és bármely két u és v csoportbeli vektorra $d(u,v) \leq t$. Válasszunk továbbá k_i szupervektor koordinátánkénti szélességeket. A csoportoktól megköveteljük az eddigieken túl, hogy a 3.5.7. megjegyzés értelmében egyértelmű burkoló szupervektoruk koordinátánkénti szélessége ne legyen nagyobb, mint a k_i értékek. Az így kapott csoportok nem feltétlenül fogják teljesen „kitölteni" a burkoló szupervektorukat, ám a vektorok csomósodása miatt számíthatunk arra, hogy eléggé jól kitöltik azt, főleg ha a koordinátánkénti szélességet, és így az egész szupervektor méretét kicsire vesszük.

Vizsgáljuk meg ezek után a 3.4.6. követelményeket. Az „ugyanúgy viselkedés" követelményét méginkább teljesítik a csoportjaink, hiszen az egyes csoportokba kerülő vektorok még kevésbé térhetnek el egymástól. A „speciális pár-mentesség" kritériumát könnyen ellenőrizhető, t és k_i értékekre vonatkozó, nem túl szigorú, elméleti jellegű feltételek teljesítésével elégíthetjük ki. A sebesség tovább lassult, de a tapasztalatok szerint még mindig nem jelentős a műveletigény, főleg, ha a t-ből adódó korlátot vizsgáljuk először.

Az ezen szakasz elején említett hiányosságokból a másodikkal, amely a „speciális pár-mentesség" kritériumának elméleti megfontolásokon alapuló teljesítését rótta fel hibául, hamarost foglalkozunk. Az első, mely arról szólt, hogy nem lehet elég finom felbontást definiálni, ezzel a módszerrel kiküszöbölhetjük úgy, ha csak néhány koordinátában engedünk meg eltérést, és a többiben nem, ezzel rendkívül finoman vezérelhetjük a csoportosítást. A harmadik hiányossággal, az ellentmondás elérésével a következő szakaszban foglalkozunk.

Azt, hogy két diszkretizált vektor ortogonális-e egymásra, illetve, hogy torzítatlanok-e egymásra, úgy néztük meg, hogy a modulo N vett különbségvektoruk $ORT_{N,eps}$ illetve $UB_{N,eps}$-beli-e. Mivel a szupervektoros csoportosításnál pont az egy csoporton belüli vektorok különbségvektoraira teszünk megszorításokat, érdemesnek tűnik ezt felhasználva elméleti megfontolásokat tenni arra, hogy milyen t és k_i értékek mellett lesz automatikusan biztosított az, hogy ne legyen speciális pár a halmazban. Nézzük a k_i értékek maximumát, legyen ez M. Könnyen látható, hogy a halmaz különbségvektorainak koordinátái között csak a következő számok fordulhatnak elő: M, $M-1, \ldots, 0, N, \ldots, N-M$.

Vizsgáljuk most meg a skalárszorzást. Vegyünk két 1.2.1. alakú vektort: legyen $u = \frac{1}{\sqrt{6}}(1, e^{2i\pi\phi_1}, e^{2i\pi\phi_2}, e^{2i\pi\phi_3}, e^{2i\pi\phi_4}, e^{2i\pi\phi_5})$ és $v = \frac{1}{\sqrt{6}}(1, e^{2i\pi\rho_1}, e^{2i\pi\rho_2}, e^{2i\pi\rho_3}, e^{2i\pi\rho_4}, e^{2i\pi\rho_5})$ A skalárszorzat ekkor:

$$\langle u, v \rangle = \frac{1}{6}\left(1 + \sum_{i=1}^{5} e^{2i\pi(\phi_i - \rho_i)}\right). \tag{3.5.1}$$

A korábban már látott eltolásinvarianciából következik, hogy bármely két vektor skaláris szorzata megegyezik egy olyan vektorpár skaláris szorzatával, amelyek közül az egyik diszkretizáltjában minden koordinátában 0 van. Speciálisan az ilyen vektorpárok skaláris szorzata is 3.5.1 alakú. Ez a skalárszorzat ebben az esetben felfogható úgy is, hogy a komplex 1 értékhez hozzáadunk öt $e^{2i\pi(\phi_i-\rho_i)}$ alakú komplex számot, majd egy skalárral szorzunk. Az $e^{2i\pi(\phi_i-\rho_i)}$ alakú komplex számok abszolútértéke 1. Ha tudjuk két diszkretizált vektorról, hogy a koordinátáik modulo N vett különbségének abszolútértéke nem nagyobb, mint M, mint ahogy azt láttuk fent, akkor adhatunk egy alsó becslést az általuk reprezentált komplex elemű vektorok skaláris szorzatának valós részére,

mégpedig úgy, hogy alsó becslést adunk a $\sum_{i=1}^{5} e^{2i\pi(\phi_i-\rho_i)}$ összeg valós részére. A korábbiak miatt az általánosság megszorítása nélkül feltehetjük, hogy $\rho_i \in I_0$ minden i-re. Az előbbiekből következik, hogy az $e^{2i\pi(\phi_i-\rho_i)}$ tagokban szereplő $\phi_i-\rho_i$ értékek az $M, M-1, \ldots, 0, N, \ldots, N-M, N-M-1$ egészeknek megfelelő intervallumokba esnek, ha ϕ_i a $M, M-1, \ldots, 0, N, \ldots, N-M$ egészeknek megfelelő intervallumba esik. Ez azt jelenti, hogy $e^{2i\pi(\phi_i-\rho_i)}$ valós részére alsó becslés $\cos\frac{2\pi(M+1)}{N}$, és ezt az értéket akkor veszi fel ha $\phi_i-\rho_i = \frac{M+1}{N}$ vagy $\phi_i-\rho_i = \frac{N-M-1}{N}$, azaz a koordináták különbségeként előálló komplex szám a lehető legjobban balra mutat. Ebből az alsó becslésből a $Re\langle u,v\rangle > \frac{1}{6}(1+5\cos\frac{2\pi(M+1)}{N})$ alsó becslés következik. A torzítatlanság kizárásához elégséges feltétel, ha $Re\langle u,v\rangle > \frac{1}{\sqrt{6}}$ fennáll, míg az ortogonalitás kizárásához $Re\langle u,v\rangle > 0$ elégséges, látható, hogy az előbbi jóval szigorúbb feltételt jelent, így csak ezzel foglalkozunk a továbbiakban, azaz M-re teljesülnie kell a következőnek:

$$M < \arccos\left(\frac{\sqrt{6}-1}{5}\right)\frac{N}{2\pi} - 1 \approx 0.2031N - 1 \tag{3.5.2}$$

N=19 esetén M=2-vel teljesül a feltétel, így tehát azt mondhatjuk, hogy amíg egy csoporton belül nem engedünk meg 2-nél nagyobb távolságot a koordinátákban, azaz $k_i \leq 2$ esetén biztosan teljesítik a csoportok a „speciális pár-mentesség" feltételét. A 3.5.2. feltételből M-re egy 3-hoz nagyon közeli szám jön ki, így érdemes lehet megpróbálni valahogy gyengíteni a kritériumot. Ehhez ad segítséget az $UB_{N,eps}$ halmaz vizsgálata. Az ugyanis, ha semelyik két vektor különbségvektora nem szerepel $UB_{N,eps}$ halmazban, nyilván egy elégséges feltétel. Ténylegesen azt tapasztaljuk, hogy $UB_{N,eps}$ halmazban N=19 esetén 20 db olyan vektor van, amelyek elemei mind a $\{3,2,1,0,18,17,16\}$ halmazból kerülnek ki, mégpedig a (0,3,3,3,16,16) és a (0,3,3,16,16,16) vektorok és permutáltjaik adják ezt a 20 vektort. Azt, hogy a különbségvektor ezen vektorok valamelyike legyen, kizárhatjuk, ha t=14-et választunk, és emellett k_i=3 minden i-re, mert t=15 ezen 20 vektoron.

Összefoglalva: a szupervektoros csoportosítást N=19 esetén t=14 és $k_i = 3$, vagy ennél kisebb értékekkel alkalmazva olyan csoportokat kapunk, amelyekben biztos, hogy nincs sem torzítatlan, sem ortogonális vektorpár.

3.5.8. Megjegyzés. *Más N-t választva M-re más korlát adódhat a 3.5.2. kritérium alapján, ez a korlát lényegében az $\frac{M+1}{N}$ hányadosra nézve jelent feltételt, emiatt nagyobb N-ekre még jobban megközelíthetjük a korlátot, illetve az $UB_{N,eps}$ halmaz vizsgálatával és kiegészítő feltétel megfogalmazásával akár át is léphetjük, ahogy azt láttuk N=19-nél.*

3.6. Ellentmondás elérése a csoportosított vektorokkal

Most már csak egy lépés maradt hátra, mégpedig az ellentmondás elérése. Ebben a fejezetben ezt a témát járjuk körül.

Az előző szakaszban bemutatott csoportosítás tehát rendelkezik azzal a tulajdonsággal, hogy pontosan meg tudjuk adni az egyes csoportok burkoló szupervektorának koordinátánkénti szélességét, illetve az egy csoportba kerülő vektorok maximális távolságát, illetve a fenti korlátokat alkalmasan megválasztva biztosíthatjuk, hogy a csoportokon belül ne legyen se ortogonális, se torzítatlan pár. Emlékezzünk arra, hogy a csoportok közötti ortogonalitást úgy definiáltuk, hogy két csoport akkor ortogonális egymásra, ha létezik egy-egy olyan vektor a két csoportból, hogy azok ortogonálisak egymásra. Amennyiben sikerül a csoportok között torzítatlansági kritériumot definiálnunk, akkor lényegében készen vagyunk, hiszen akkor a csoportokon keresünk 6-klikkeket, majd

meghatározzuk a 6-klikk minden csoportjára torzítatlan csoportokat, és ezen csoportokon próbálunk újra 6-klikkeket keresni, immár a $\hat{H}_3$ mátrixok összeállítása céljából. Abban bízunk, hogy ez nem fog menni, és ezzel elérjük az ellentmondást.

Definiáljuk akkor tehát a csoportok torzítatlanságát, először páronként, teljesen hasonlóan a csoportok ortogonalitásának 3.4.5. definíciójához:

3.6.1. Definíció. *V'_i és V'_j csoportok pontosan akkor torzítatlanok, ha léteznek $v'_{i,k}$ és $v'_{j,m}$ vektorok (értelemszerűen rendre V'_i-ben és V'_j-ben), melyek N-torzítatlanok, (azaz különbségvektoruk $UB_{N,eps}$-beli).*

A módszer sikere azon múlik, hogy ennél a lépésnél jön-e az ellentmondás, ez pedig nagyban függ attól, hogy milyen gyakorisággal lesznek torzítatlanok egymásra a csoportok. Sajnos az $UB_{N,eps}$ halmaz mérete eléggé nagy $N = 19$-re, kb. $1.4 \cdot 10^6$, ami azt jelenti, hogy nagyjából $\frac{2}{3}$ az esélye annak, hogy két véletlenszerűen választott vektor torzítatlan legyen egymásra, mivel $19^5 \approx 2.4 \cdot 10^6$. Emiatt a csoportpárok nagyon nagy része torzítatlan lesz egymásra.

Érdemes lehet újra felidézni, hogy milyen paramétereink vannak és azok változtatása milyen hatással van a futási időre, illetve az ellentmondás elérésére.

- N: H_1 diszkretizációs paramétere, növelésével csökken $UB_{\hat{H}_1}$ mérete, de növekszik a lehetséges $\hat{H}_1$ mátrixok száma, optimális értéke 70-100 között lehet a tapasztalat szerint
- N': H_2 és H_3 mátrixok diszkretizációs paramétere, növelésével nő $UB_{\hat{H}_1}$ mérete, illetve a benne található vektorok pontossága is, így a belőlük képzett csoportok között arányaiban kevesebb él megy, optimális értéke 13-35 között lehet
- m: $UB_{\hat{H}_1}$ illetve $UB_{\hat{H}_1,\hat{H}_2}$ halmazok előállításánál a a vizsgálandó generációk száma, növelése először jelentősen, majd egyre kevésbé csökkenti az UB halmazok méretét, illetve jelentősen növeli a futásidőt, optimális értéke valahol 5-10 között lehet.
- k_i: csoportosítási paraméterek, a csoportok burkoló szupervektorának koordinátánkénti szélességét lehet szabályozni
- t: csoportosítási paraméter, a csoportokon belül előforduló legnagyobb távolságot korlátozza, az előző paraméterrel együtt a csoportok méretét, illetve mennyiségét, azaz a csoportosítás finomságát szabályozza, finomabb csoportosításnál sokkal több 6-klikk keletkezik (a csoportok számának 4-5. hatványával arányosan), de nagyobb valószínűséggel jön az ellentmondás. Ezt a nagyobb valószínűséget megpróbáltuk megfogni valami olyan mérőszámmal, ami már kevés számú (akár néhány) $\hat{H}_1$ vizsgálatával képet ad az egyes paraméterek ellentmondás-elérésre gyakorolt hatásáról. Jó mérőszámnak tűnik az, ha egy adott $\hat{H}_1$-re összegezzük minden $\hat{H}_2$-re, hogy amikor a $\hat{H}_3$ mátrixokat próbáljuk összeállítani, azaz másodlagos 6-klikkeket keresünk, akkor ha nem keletkezik ugyan 6-klikk, akkor keletkezik-e 5-, 4- vagy 3-klikk, illetve, hogy mennyi keletkezik belőlük. Ezeket összegezve minél kevesebb keletkezik belőlük, illetve minél kisebb méretű a legnagyobb keletkező klikk, annál nagyobb valószínűséggel érjük el az ellentmondást.

A fenti paramétereket vizsgálva sikerült olyan értékeket meghatározni, mellyel a $\hat{H}_1$ mátrixok túlnyomó többségére elérjük az ellentmondást. A tapasztalatok azt mutatták, hogy ha a k_i értékek között 3 db 1-es és 2 db 2-es szerepel, akkor a nekik megfelelő szupervektorméret 4, és N=101-et N'=19-et, m=5-öt választva $UB_{\hat{H}_1}$ mérete kb. 500 lesz, a csoportok száma így kb. 130-150 lesz, és

ebből az algoritmus $\hat{H}_2$ és $\hat{H}_3$ mátrixokat előállító része (mostantól hívjuk második résznek) kb. 0,1 mp alatt lefut, míg az $UB_{\hat{H}_1}$ halmazt előállító rész futása eltart kb. 10 mp-ig, tehát ez a rész jóval lassabb. Szerencsénkre a $\hat{H}_1$ mátrixok is „csomósodnak" olyan értelemben, hogy beoszthatóak olyan többezres méretű csoportokba, amelyekben található $\hat{H}_1$ mátrixok első 4 sora megegyezik. Ekkor az $UB_{\hat{H}_1}$ halmazokat úgy is elő lehet állítani, hogy először meghatározzuk azon vektorokat, melyek a közös első négy sorra univerzálisan torzítatlanok, ez viszonylag sokáig tart, viszont csak ritkán kell végrehajtani (csoportonként egyszer), és megmarad néhány ezer vektor. Az $UB_{\hat{H}_1}$ halmazokba kerülő vektorokat már csak ezen néhány ezer vektor között kell keresnünk, így jelentősen lerövidül a keresésre fordítandó idő. A konkrét vizsgált esetben kb. 0,3 mp alatt határozzuk meg $UB_{\hat{H}_1}$ halmazokat. Tesztelési célokból több ilyen $\hat{H}_1$-csoportot is megvizsgáltunk, és azt tapasztaltuk, hogy a fenti paraméterekkel a $\hat{H}_1$ mátrixok kb. 1/20-ánál nem jutunk ellentmondásra.

Kérdés, hogy ezekkel az esetekkel hogyan bánunk el. Általános elv lehet, hogy mivel az esetek túlnyomó többségénél ezen paraméterek mellett jön az ellentmondás, először minden esetet így próbálunk meg megoldani, és amennyiben nem jön az ellentmondás, akkor változtatunk a paramétereken, vagy nagyobb mélységben vizsgálódunk $UB_{\hat{H}_1}$ előállításakor, vagy a csoportosításon finomítunk. Arra számíthatunk, hogy a csoportosításon finomítva előbb-utóbb ellentmondásra jutunk, mert kipróbáltuk, hogy mi történik, ha a csoportosítás szélsőségesen finom, azaz minden vektor külön csoportba kerül. Ekkor természetesen jön az ellentmondás, olyannyira, hogy a $\hat{H}_3$ mátrixok előállításakor nem hogy 6-klikket, de már 3-klikket sem találunk.

További egyszerű szűrést biztosíthatnak az oszlopokra vonatkozó feltételek, amikkel korábban a csoportosítatlan esetben a 3.2. szakaszban foglalkoztunk. Az ott ismertetett módszert nem tudjuk közvetlenül alkalmazni, hiszen a 6-klikkek most nem vektorokból, hanem szupervektorokból állnak, és így az oszlopvektorok koordinátái nem egész számok, hanem intervallumok. Definiálhatóak azonban nekünk megfelelő feltételek. Először is az oszlopok felfoghatók általánosított szupervektoroknak abban az értelemben, hogy itt a 3.5.1. definícióval szemben az első koordinátában is intervallumot engedünk meg, a többi koordinátához teljesen hasonló módon. Ebben az esetben is a 3.5.4. megjegyzésben leírtaknak megfelelően beszélhetünk a szupervektornak megfelelő vektorhalmazról, amelyben 3.2.2 alakú vektorok találhatóak. Amiatt, hogy intervallumok alkotják a koordinátákat, már nem építhetünk arra, hogy a 2. oszlop szupervektorának megfelelő halmazban csak monoton növő koordinátájú vektorok lennének, ám azt megkövetelhetjük minden oszloptól, hogy az adott oszlop szupervektorához tartozó halmazban létezzen olyan vektor, ami ortogonális az első oszlopra, ami még mindig a csupa 0 vektor, azaz ennek a vektornak ORT_N-belinek kell lennie. Ezen túlmenően további követelmény még, hogy minden oszlopvektorpárra a hozzájuk tartozó szupervektornak megfelelő halmazban legyen egy-egy olyan vektor, melyek ortogonálisak, azaz modulo N vett különbségvektoruk $ORT_{N,eps}$-beli. Ezeket a feltételeket is vizsgálva már azt tapasztaltuk, hogy minden $\hat{H}_1$ mátrixra elérjük az ellentmondást, bár az is igaz, hogy az elérési valószínűségre bevezetett mérőszámaink azt mutatják, hogy pont a határon vagyunk, mert jópár $\hat{H}_1$ mátrix esetén találunk $\hat{H}_3$ mátrix összeállításakor 5-klikket.

Felmerülhet a kérdés, hogy vajon miben különbözik a csoportosítás attól, mintha egyszerűen csökkentenénk N'-t, hiszen mindkét esetben arról van szó, hogy az egyes vektorok koordinátánkénti szélességét növeljük, ezzel csökkentve a vektorok számát. Egyrészt a tapasztalat azt mutatja, hogy különbségek vannak az algoritmus teljesítőképességében, egyrészt nem csökken olyan nagy mértékben a vektorok száma N' csökkentésekor, mint a csoportosítást alkalmazva, másrészt az ellentmondás is nehezebben jön. Ennek a magyarázata az lehet, hogy a csoportosítás sokkal jobban fogja meg azon tartományokat, amelyekben a torzítatlan vektorok vannak, mert előfordulhat, hogy

csak néhány koordinátában növeljük a tartományt, a többiben ugyanolyan kicsi marad, míg N' csökkentésekor minden koordináta pontosságát rontjuk.

4. fejezet

A kutatás jelenlegi állása

Összefoglalva azt mondhatjuk, hogy az előző fejezetbeli beállításokkal és módszerekkel egy adott $\hat{H}_1$ mátrix esetén átlagosan 0,4 mp alatt elérjük az ellentmondást, és mivel az algoritmus futási idejének legnagyobb részét ez teszi ki, ezt tekinthetjük az egy $\hat{H}_1$ mátrixra jutó átlagos futási időnek. Mivel becsléseink szerint kb. $10^{11} - 10^{12}$ ilyen $\hat{H}_1$ mátrix van N=101 esetén, kiszámolható, hogy a teljes futási idő $10^{10} - 10^{11}$ mp, ami egy gépen $\approx$ 1000 év lenne. Egy megfelelően nagy klaszteren (gépenként több magot feltételezve) futtatva ez már emberi időben, akár néhány hónap, esetleg egy év alatt lefuthat.

A kutatás közben felmerültek azonban még ötletek, amelyek még ezt a futási időt nagyságrendekkel javíthatják akár. Az egyik ötlet azzal van összefüggésben, hogy a $\hat{H}_3$ mátrixok összeállításakor, amikor a szupervektorokból álló $\hat{H}_2$ mátrixokra torzítatlan vektorokat vizsgáljuk, akkor nagyon gyenge feltételt alkalmazunk. Ezt a feltételt le lehet cserélni egy a 2.4. szakaszban leírthoz hasonló „univerzális" torzítatlan vektor kereső algoritmusra. Az ott leírt algoritmust nem nehéz általánosítani a szupervektorok esetére, hiszen az 1.4. szakaszban ismertetett lemmák csak az egyes koordináták intervallumának kezdő-, vég- illetve középpontját használják, és ezek a szupervektoroknál is rendelkezésünkre állnak. Az elképzelt algoritmust megvalósítottuk arra a speciális esetre, amikor a $\hat{H}_1$ mátrix sorai mellé illesztett 6-klikk vektorai 2.2.1 alakúak, azaz a csoportosítást figyelmen kívül hagyjuk. Ekkor azt tapasztaltuk, hogy ez az univerzális vizsgálat töredékére csökkenti a torzítatlan vektorok számát. Az időigény sem túl magas, hiszen csak $UB_{\hat{H}_1}$ vektorain kell végigmennünk. Reményeink szerint ez a fajta módosítás lehetőséget fog adni arra, hogy jelentősen csökkentsük N paraméter értékét, és ezzel a futásidőt.

A vektorok csoportosításának ötletét alkalmazhatónak látjuk $\hat{H}_1$ mátrixok csoportosításánál is. Itt valami olyasmire gondolunk, hogy a mostani algoritmusban, amint beillesztettünk két sort és oszlopot, és meghatároztuk azon sorokat, amelyek a többi sorok lehetnek, akkor csoportosítjuk őket a 3.5. szakaszban leírt módon, és a továbbiakban a csoportokat próbáljuk meg beilleszteni sorként. Az így kapott immár némileg pontatlanabb $\hat{H}_1$ mátrixokhoz az előző bekezdésben vázolt univerzális torzítatlan vektor keresővel meghatározhatjuk $UB_{\hat{H}_1}$ halmaz vektorait. Azokat pedig már a szokásos módon, csoportosítva vizsgálhatjuk. Arra számíthatunk, hogy az így keletkezett $UB_{\hat{H}_1}$ halmaz mérete nagyobb lesz, de bízunk abban, hogy ezt ellensúlyozza az, hogy ezzel több $\hat{H}_1$ mátrixot vizsgálunk egyszerre.

A fenti módosítások reményeink szerint nagyságrendekkel csökkenthetik a futási időt, és így akár már a közeljövőben is bizonyíthatjuk az 1.1.3. sejtést.

Köszönetnyilvánítás

A diplomamunka megszületésében nagyon fontos szerepet játszott témavezetőm, Matolcsi Máté, akihez bármikor fordulhattam kérdéseimmel, köszönet illeti továbbá Móra Pétert és Juhász Pétert, akik a kutatásban részt véve segítették munkámat.

Párom, Vértesi Ágnes végtelen türelemmel viselte, míg én a dolgozat megírásával töltöttem éjszakáimat, sok lemondással járt ez a munkám az ő részéről is.

Szintén köszönet illeti a családomat, hogy nélkülözni tudták segítségemet és jelenlétemet a munkával töltött hónapok alatt, és amiatt is, hogy végig támogattak tanulmányaim során.

Irodalomjegyzék

[1] I. Bengtsson, W. Bruzda, şA. Ericsson, J.-A. Larsson, W. Tadej & K. Zyczkowski: *Mutually unbiased bases and Hadamard matrices of order six.* J. Math. Phys. 48 (2007), no. 5, 052106, 21 pp.

[2] G. Björck: *Functions of modulus* 1 *on* $\mathbb{Z}_n$*, whose Fourier transform have constant modulus, and „cyclic n-roots".* Recent Advances in Fourier Analysis and its applications. NATO *Adv. Sci. Int. Ser. C: Math. Phys. Sci.*, Kluwer **315**., 131–140. (1990)

[3] G. Björck & B. Saffari: *New classes of finite unimodular sequences with unimodular Fourier transforms. Circulant Hadamard matrices with complex entries.* C. R. Acad. Sci. Paris, Serie 1 **320** (1995), 319–324.

[4] S. Brierley & S. Weigert: *Maximal sets of mutually unbiased quantum states in dimension six.* arXiv:0808.1614 (quant-ph).

[5] S.Brierley & S. Weigert: *Constructing Mutually Unbiased Bases in Dimension Six.* arXiv:0901.4051 (2009)

[6] P. Butterley & W. Hall: *Numerical evidence for the maximum number of mutually unbiased bases in dimension six.* Physics Letters A 369 (2007) 5-8.

[7] P. Diţă: *Some results on the parametrization of complex Hadamard matrices.* J. Phys. A, **37** no. 20, 5355–5374. (2004)

[8] M. Grassl: *On SIC-POVMs and MUBs in Dimension 6.* in: Proc. ERATO Conference on Quantum Information Science (EQUIS 2004), J. Gruska (ed.)

[9] Ph. Jaming, M. Matolcsi, P. Móra, F. Szöllősi, M. Weiner: *A generalized Pauli problem and an infinite family of MUB-triplets in dimension 6.* J. Physics A: Mathematical and Theoretical, Vol. 42, Number 24, 245305, 2009.

[10] Bengt R. Karlsson: *Two-parameter complex Hadamard matrices for N=6.* J. Math. Phys. 50, 082104; doi:10.1063/1.3198230, August (2009).

[11] A. Klappenecker & M. Rötteler: *Constructions of Mutually Unbiased Bases.* Finite fields and applications, 137–144, Lecture Notes in Comput. Sci., **2948**, Springer, Berlin, 2004.

[12] C.W.H. Lam, L. H. Thiel & S. Swiercz: *The non-existence of finite projective planes of order 10.* Can. J. Math., Vol: XLI, (1989) 1117-1123.

[13] M. Matolcsi, F. Szöllősi: *Towards the classification of* 6×6 *complex Hadamard matrices.* Open Sys. & Inf. Dyn. **15:2**, 93–108. (2008)

[14] K. Beauchamp & R. Nicoara: *Orthogonal maximal Abelian $*$-subalgebras of the 6×6 matrices.* Linear Algebra Appl. **428** (2008), 1833–1853.

[15] J. Schwinger, *Unitary Operator Bases.* Proc Nat. Acad. Sci. U.S.A. 46, (1960) 560.

[16] A. J. Skinner, V. A. Newell, R. Sanchez: *Unbiased bases (Hadamards) for 6-level systems: Four ways from Fourier.* arXiv:0810.1761 (2008)

[17] F. Szöllősi: *A two-parameter family of Hadamard matrices of order* 6 *induced by hypocycloids.* Proc. AMS, megjelenés alatt.

[18] T. Tao: *Fuglede's Conjecture Is False in* 5 *and Higher Dimensions.* Math Res. Letters, **11**, 251–258. (2004)

[19] G. Zauner: *Quantendesigns Grundzüge einer nichtkommutativen Designtheorie.* PhD thesis, Universität Wien, 1999. (available at http://www.mat.univie.ac.at/~neum/ms/zauner.pdf)

[20] R. F. Werner, *All teleportation and dense coding schemes. Quantum information and computation.* J. Phys. A, **34** (2001), 7081–7094.

[21] W. K. Wootters & B. D. Fields: *Optimal state-determination by mutually unbiased measurements.* Ann. Physics **191** (1989), 363–381.

[22] A [9] cikk eredményeinek dokumentációja: http://www.math.bme.hu/~matolcsi/angpubl.html

[23] Az elkészült és felhasznált kódok, illetve segédfájlok megtalálhatóak a http://www.cs.elte.hu/~gyozo12/MUB/ címen