



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Számítástudományi és Információelméleti Tanszék

Kvantum gráfolyongások

MSC ÖNÁLLÓ LABORATÓRIUM 1.

Készítette
Nemkin Viktória

Konzulens
dr. Friedl Katalin

2021. május 9.

Tartalomjegyzék

1. Bevezetés	2
2. Klasszikus gráfolyongások	3
3. Kvantuminformatika	5
4. Kvantum séta	6
5. Architektúra	7
5.1. Gráfmodellek	7
5.2. Szimulátorok	8
5.3. Futtatás, konfiguráció, eredmények ábragenerátora	9
6. Szimulációk, eredmények	10
6.1. Súlyzó gráf	10
6.2. Ragasztott bináris gráf	12
7. Jövőbeli tervek	15
Irodalomjegyzék	16

1. fejezet

Bevezetés

A 20. századi fizika hatalmas változásokat hozott. A relativitáselmélet mellett megjelent a kvantummechanika, mely teljesen megváltoztatta a világnézetünket. Ugyanebben az időszakban kezdődött el a számítástechnika hajnala is. Megjelentek az első számítógépek és elkezdték megírni az első programokat, algoritmusokat.

Richard P. Feynman 1982-es cikkében fejtette ki, hogy a klasszikus számítógépekkel sajnos csak exponenciális időben lehet kvantumjelenségeket szimulálni, ez azonban túlságosan lassú a kísérletek elvégzéséhez. Ha viszont lenne egy kvantumjelenségek alapján működő számítógépünk, akkor azzal hatékonyan lehetne szimulációkat végezni, a fizikai kutatások elvégzéséhez. Így született meg a kvantumszámítógép gondolata.

Benioff, Deutsch, majd Bernstein és Vazirani munkássága nyomán megszületett a kvantum számítási modell, a kvantum Turing-gép a 80-as évek végére. Ettől kezdve az a kérdés foglalkoztatja a kvantum algoritmusok kutatóit, hogy vajon vannak-e használható kvantum algoritmusok, illetve vannak-e olyanok amik jobbak mint a klasszikus párjaik.

Shor 1990-es kvantumon alapuló prímfaktorizációs algoritmusa már használható algoritmus lett és az RSA alapú kódolás feltörésével fenyeget, ami komoly veszélyt, illetve komoly előnyt is jelent azoknak akiknek van kvantumszámítógépük. Erre már felfigyeltek a nagyhatalmak, multinacionális cégek és szép lassan elkezdtek a kvantumszámítógépek kifejlesztésével foglalkozni.

2019-ben a Google quantum supremacy bizonyítéka mutatott egy olyan kvantumalgoritmust amely bár nem túl hasznos, de a mai legjobb szuperszámítógépeket is megverte a Sycamore processzoruk teljesítménye.

Manapság egyre forróbb témává válik a kvantum és Magyarországon is egyre több támogatás jut kvantummal kapcsolatos kutatásokra. A BME-n a Kvantuminformatikai Nemzeti Labor tevékenykedik kvantum titkosításon alapuló internet kifejlesztésével és több tanszék, köztük a SZIT is bekapcsolódott a projektbe.

Ezen dolgozat a kvantumalgoritmusokon belül a kvantum bolyongásokkal foglalkozik. A kvantum bolyongáson alapuló algoritmusokat azért érdemes kutatni, mert több ismert kvantum algoritmusnak az alapját képezik. A félév során megismerkedtem a kvantuminformatika alapjaival, a gráf-bolyongások klasszikus illetve kvantum változatával, valamint elkészítettem egy Pythonos keretrendszert melyben könnyen lehet gráf-bolyongásokkal kapcsolatos kísérleteket, szimulációkat végezni.

A dokumentum hátralévő fejezeteiben bemutatom a kvantuminformatika alapjait, ismertetem a klasszikus gráf-bolyongást, majd a kvantum bolyongás egy speciális esetét, bemutatom az elkészített Pythonos keretrendszert, majd ismertetem a félév során kapott szimulációs eredményeket, végül a továbbfejlesztési lehetőségekről beszélek.

2. fejezet

Klasszikus gráfolyongások

A klasszikus gráfolyongások tulajdonképpen Markov-folyamatoknak felelnek meg.

A Markov-folyamat valószínűségi változóknak egy sorozata $(X_1, X_2, \dots, X_n, \dots)$, melyek ugyanabból a diszkrét eloszlásból származnak és melyekre teljesül a Markov-tulajdonság.

Definíció 1. Markov-tulajdonság A valószínűségi változók sorozatában az $i+1$. változó értéke csak az i . változó értékétől függ, azaz $P(X_{i+1} = x_{i+1} | X_i = x_i, \dots, X_1 = x_1) = P(X_{i+1} = x_{i+1} | X_i = x_i)$.

Definíció 2. Homogén Markov-lánc Olyan Markov-lánc melyben az átmeneti valószínűség nem függ az időtől, azaz $P(X_{n+1} = j | X_n = i) = p_{i,j}$.

Definíció 3. Átmeneti valószínűségi mátrix A Markov-lánc átmeneti valószínűségi mátrixa ezen $p_{i,j}$ elemekből áll.

Definíció 4. Stacionárius eloszlás Az X_i változó eloszlása ha i tart a végtelenbe?

Egy ilyen Markov folyamat megfigyelhető egy irányított súlyozott gráfnak, a következő megkötésekkel:

- A gráf csúcsai megfelelnek a valószínűségi változók (közös) értékkészletének az elemeinek.
- Az i . csúcsból a j . csúcsba mutató él súlya $p_{i,j}$.
- Amennyiben $p_{i,j} = 0$, úgy nem mutat él az i . csúcsból a j . csúcsba.

Ezek alapján a Markov-lánc átmeneti valószínűségi mátrixa a fent definiált gráf szomszédossági mátrixával egyezik.

A gráfon klasszikus értelemben vett bolyongást a következő módon végzünk:

- Kiindulunk egy előre megadott csúcsból.
- A csúcsból kifelé mutató élek valószínűségeivel súlyozva véletlenszerűen választunk egy következő csúcsot.
- Meghatározott lépésig ismétljük az előző pontot.

Egy gráfon egyszerre több ilyen bolyongást is végezhetünk, párhuzamosan. A k . lépésben hozzárendelhető a gráf csúcsaihoz, hogy aktuálisan hány bolyongó helyezkedik el bennük. Megfelelően sok bolyongóval így közelíthető az Markov-lánc k . változójának az eloszlása.

Bolyongásokkal kapcsolatban többféle tulajdonságot szoktak mérni adott gráfon.

Definíció 5. Keveredési idő A Markov lánc azon valószínűségi változójának az indexe, melynek az eloszlása a stacionárius eloszlás ϵ sugarú környezetében van.

Definíció 6. Hitting time Az $i \rightarrow j$ csúcsok közötti elérési idő a legkisebb szám, ahányadik hatványra emelve az átmeneti valószínűségi mátrixot az i . sor j . cellájában nem 0 az érték?

Nagyon sok gyakorlati probléma kapcsolódik a Markov-folyamatokhoz: közgazdaságtan, játékelmélet, statisztikus fizika. A klasszikus gráfolyongási algoritmusok ezek mellett jól használhatóak gráfelméleti problémák közelítő megoldására. Nagy méretű gráfok esetén egy akár a csúcsszám méretében lineáris algoritmus lépésszáma is túlságosan nagy lehet. Például az internetes keresőalgoritmusok (PageRank) is ilyenek.

3. fejezet

Kvantuminformatika

Mielőtt a kvantumalgoritmusokkal elkezdhetünk foglalkozni, először meg kell ismernünk a kvantuminformatika eszköztárát. A klasszikus számítógépek esetében az információátvitel alapegysége a bit, melynek értéke lehet 0 vagy 1. Kvantumszámítógépek esetében az alapegység a kvantum bit, vagy röviden qubit. Kvantum bitek esetében a fizikai tulajdonságaiknak köszönhetően az állapottér a komplex számokon értelmezett. A legkisebb, kvantum számításra használható állapottér 2 dimenziós, ezt nevezzük qubitnek.

Definíció 7. Qubit A 2 dimenziós Hilbert-tér (H_2) egy vektorát qubitnek nevezzük. A tér bázisvektorai a $|0\rangle$ és az $|1\rangle$ ket vektorok.

Egy általános qubit tehát $c_0|0\rangle$.

Definíció 8. Koordináta reprezentáció Egy általános qubit

Definíció 9. Kvantum regiszter

4. fejezet

Kvantum séta

Kvantum bolyongások tanulmányozása esetén először célszerű az egyenesen vett bolyongást tanulmányozni.

5. fejezet

Architektúra

Ebben a fejezetben bemutatom az elkészült keretrendszert.

Programozási nyelvnek a Python 3-at választottam. Ennek oka az, hogy nagyon sok data science-el kapcsolatos modulja van, mely nagyban megkönnyíti a különböző matematikai, algoritmuselméleti problémák feltárását, könnyen iterálhatunk a különböző prototípusokon. Emellett a szintaxisa rövid, tömör, lényegretörő programkódok megírását teszi lehetővé.

A forráskód három nagy részre bomlik:

- Gráfmodellek
- Szimulátorok
- Futtatás, konfiguráció, eredmények ábragenerátora

5.1. Gráfmodellek

A félév során sokféle gráfon futtattam szimulációs kísérleteket, melyek során több problémába ütköztem. Kezdetben úgy oldottam meg a szimulációkat, hogy a cél gráfok szomszédossági mátrixait generáltam le, egyben a memóriában tartva azokat és a lépések során a megfelelő csúcshoz tartozó sorokat lekérdezve.

Ezzel a módszerrel több probléma is jelentkezett. Az első gondot az okozta, hogy a szomszédossági mátrix mérete a csúcsszám négyzetével arányos, ezért pár ezer csúcsú gráfot már nem tudtam a memóriában tartva szimulálni. A második probléma pedig az volt, hogy a szomszédossági mátrixos ábrázolás nagyon távol esett az emberi szempontból természetes ábrázolástól. A kvantum bolyongásos szimulációkat tipikusan nem véletlenszerű gráfokon szokták kipróbálni, hanem jól ismert struktúrával rendelkező gráfokon. Ilyen gráfok például a súlyzó vagy a ragasztott bináris fa gráfok.

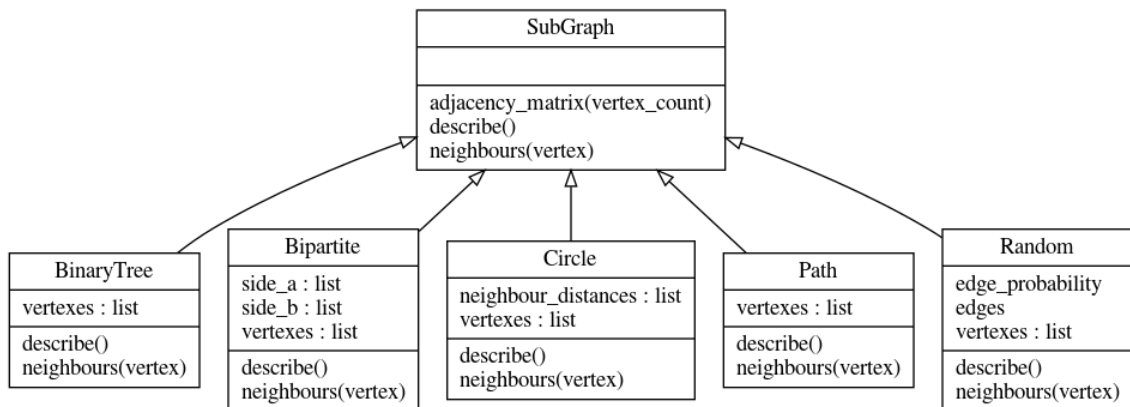
A súlyzó gráf két egyforma méretű kört tartalmaz, mindkét körből kiválasztva k darab csúcsot, melyek teljes páros gráfot alkotnak (a súlyzó középső rúdját). A kör gráfokban pedig nem csak az egymás melletti csúcsok között fut él, hanem futhat él minden i . csúcs között is. A ragasztott bináris fa gráf pedig olyan, hogy két teljes bináris gráf leveleit szembefordítjuk és a két oldali levelek közé egy teljes páros gráfot készítünk.

A fenti leírásból látható, hogy az ember számára természetes leírás a gráfokat ismert részgráfok kompozitjaként adja meg. A félév során olyan architektúrát alakítottam ki a szimulációkhoz, mely ezt a szemléletet támogatja. A szomszédossági mátrixos tárolási mód helyett pedig a szomszédossági orákulum megközelítést használva nagyban csökkent a memóriaigénye az alkalmazásnak. Ennek a megközelítésnek a lényege, hogy az ismert struktúrájú gráfokra nem tárolok a memóriában szomszédossági információt, helyette biz-

tosítok egy függvényt, amely a bemeneti paraméterként kapott csúcsindexre kiszámolja a vele szomszédos csúcsok indexeit.

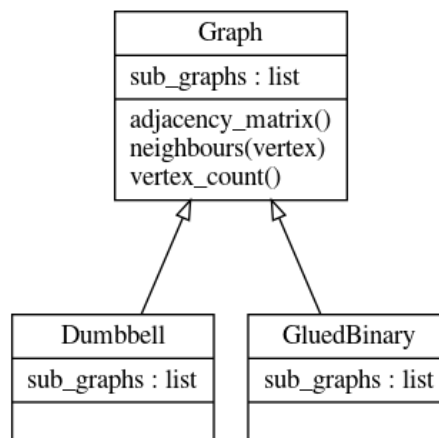
A félév során a következő nevesített részgráfok szomszédossági orákulumját implementáltam:

- BinaryTree
- Bipartite
- Circle
- Path
- Random



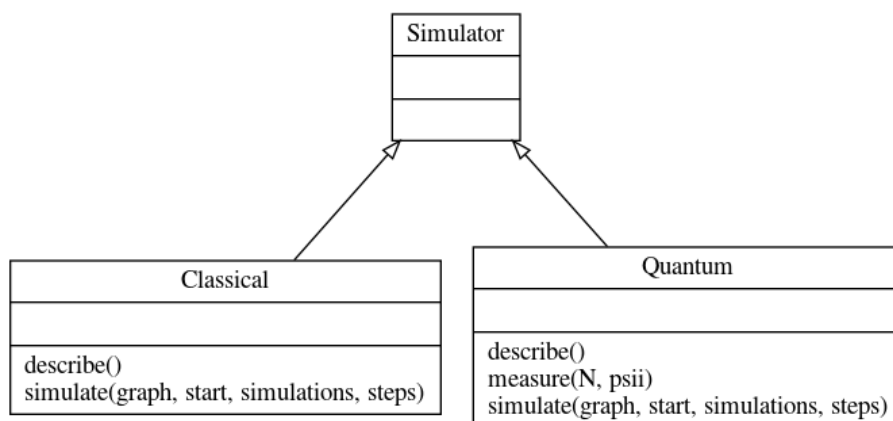
Ezen részgráfokból épülnek fel az alábbi kompozit gráfok:

- Dumbbell
- GluedBinary



5.2. Szimulátorok

A szimulátor osztályok közül a klasszikus tetszőleges kompozit gráfot tud fogadni, a kvantum szimulátor jelenleg a kvantum bolyongás egy speciális esetét, az egyenesen való bolyongást képes kezelni, mely a 2-regularitása miatt egyszerűbben implementálható. Hosszú távú cél a k -reguláris, illetve az általános gráfokra kiterjeszteni ezt a szimulátort.



5.3. Futtatás, konfiguráció, eredmények ábragenerátora

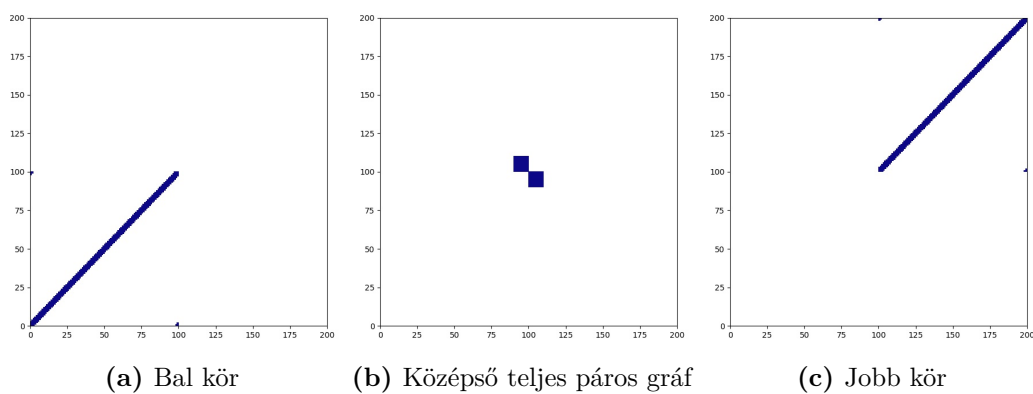
A fenti osztályok segítségével egy olyan keretrendszert alakítottam ki, melyben nagyon gyorsan fel lehet 1-1 futtatást konfigurálni. A futtatás eredményeit egy összesített Latex dokumentumba gyűjti a program. Ez tartalmazza a beadott gráf részgráfjainak nevesített típusát, szomszédossági mátrixait, illetve a teljes gráf szomszédossági mátrixát, valamint a szimulációk eloszlási eredményeit. A következő fejezetben több ilyen ábrát is bemutatok.

6. fejezet

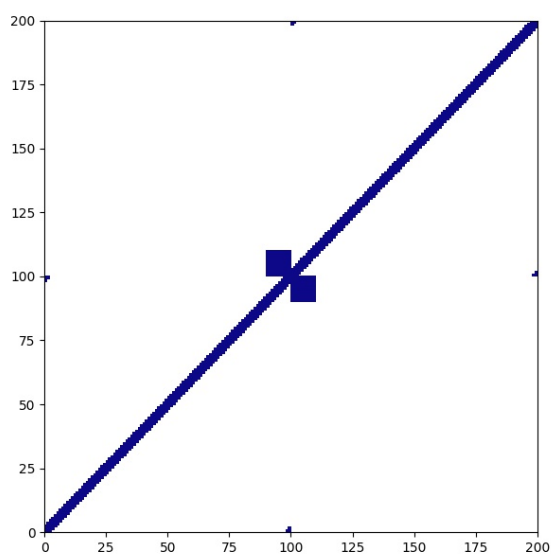
Szimulációk, eredmények

6.1. Súlyzó gráf

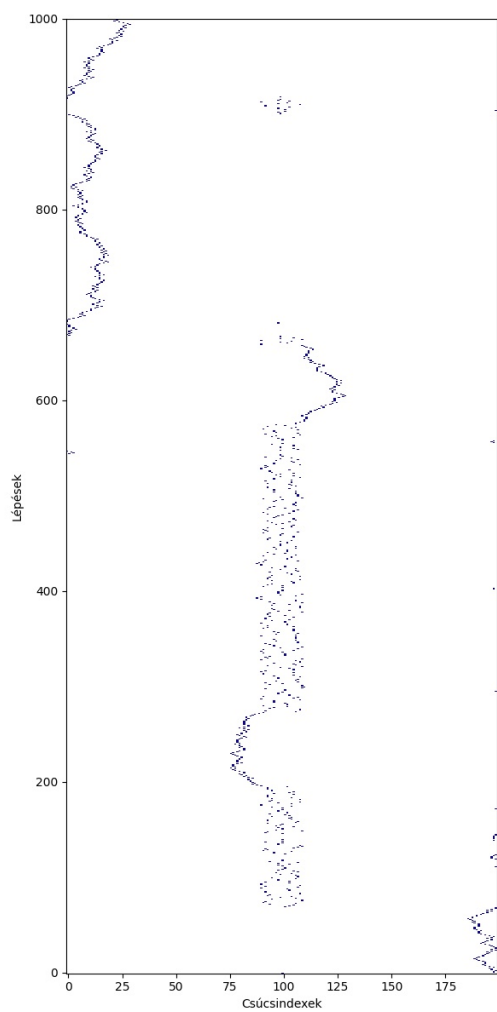
Például egy súlyzó gráfról a következő képek készültek:



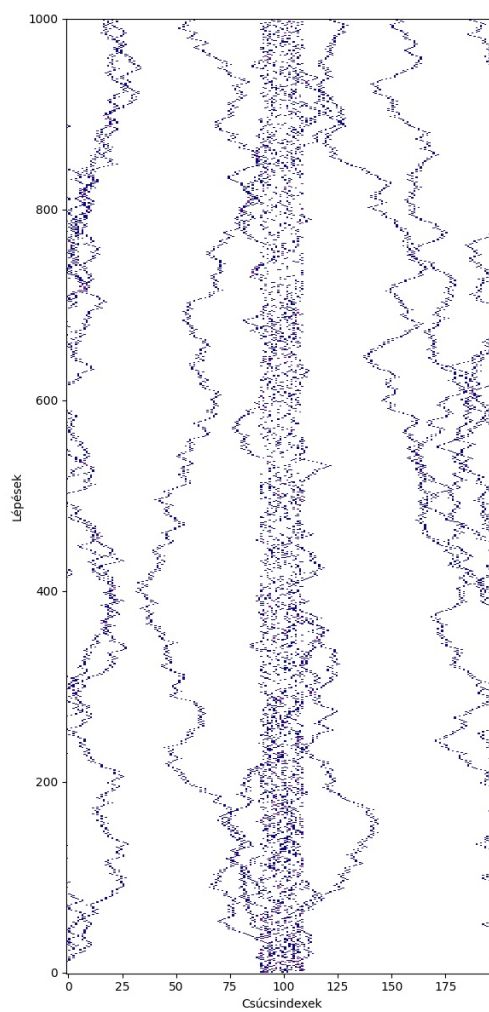
6.1. ábra. Súlyzó gráf részgráfjai



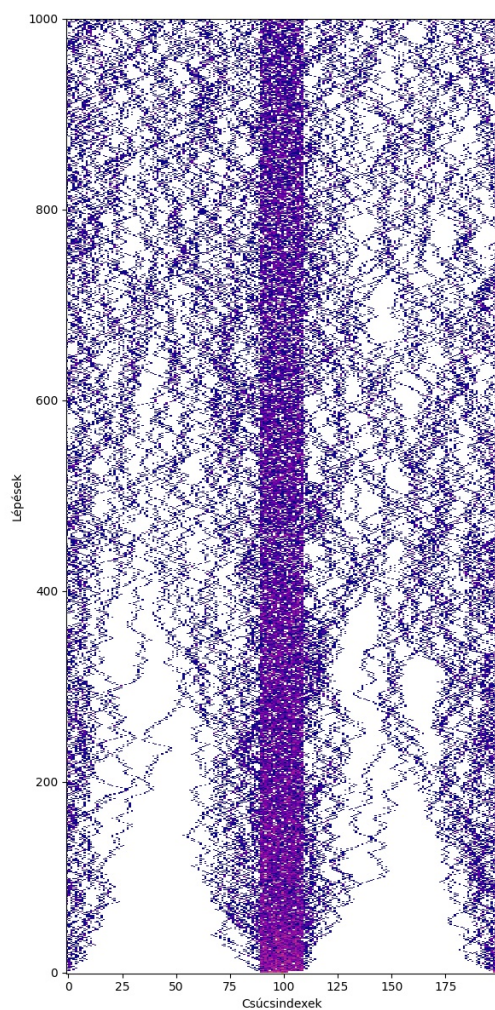
6.2. ábra. Súlyzó gráf



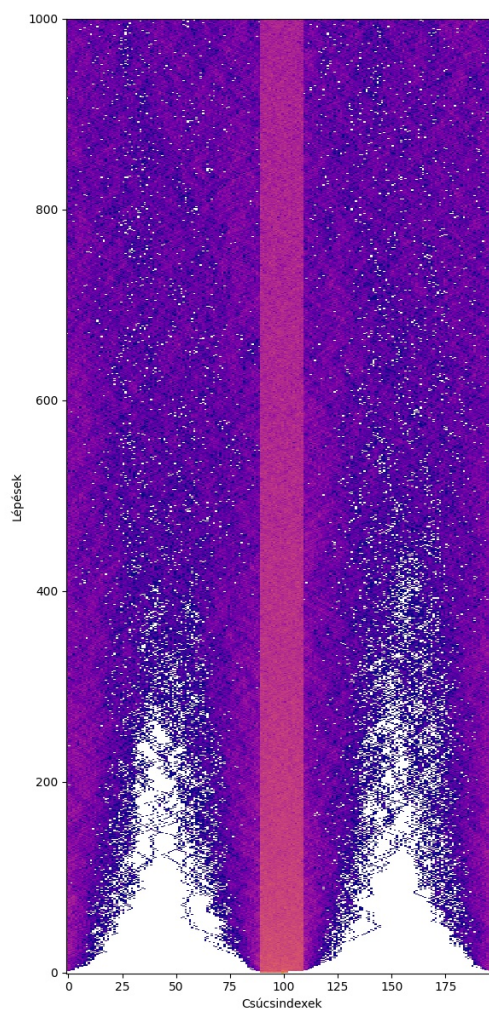
(a) 1 bolyongó



(b) 10 bolyongó

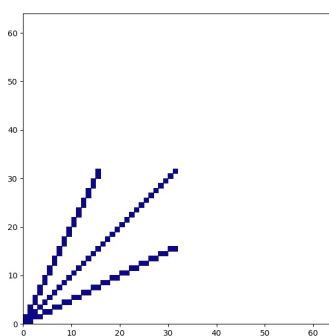


(a) 100 bolyongó

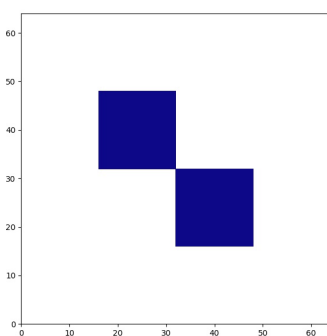


(b) 1000 bolyongó

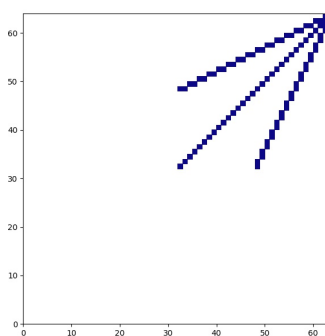
6.2. Ragasztott bináris gráf



(a) Bal bináris fa

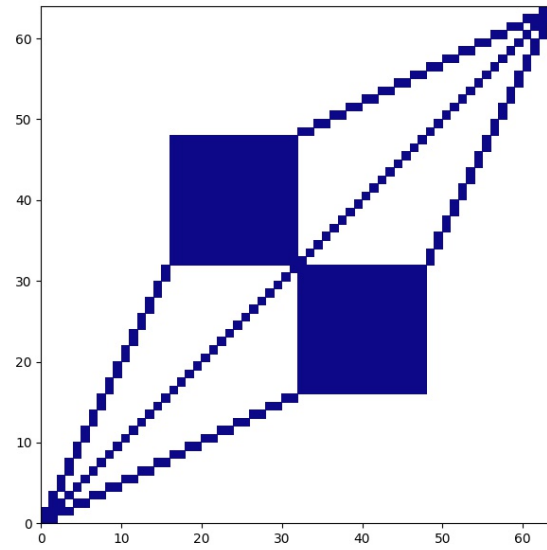


(b) Középső teljes páros gráf

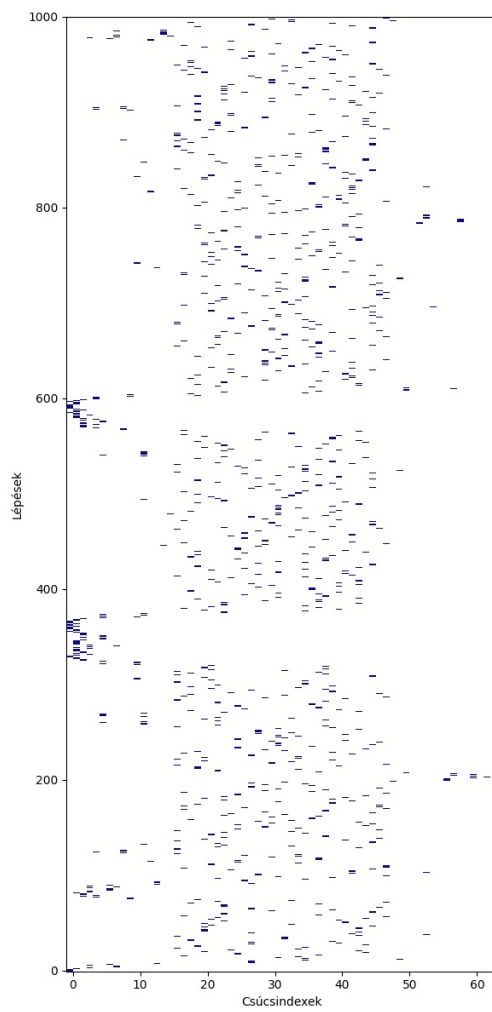


(c) Jobb bináris fa

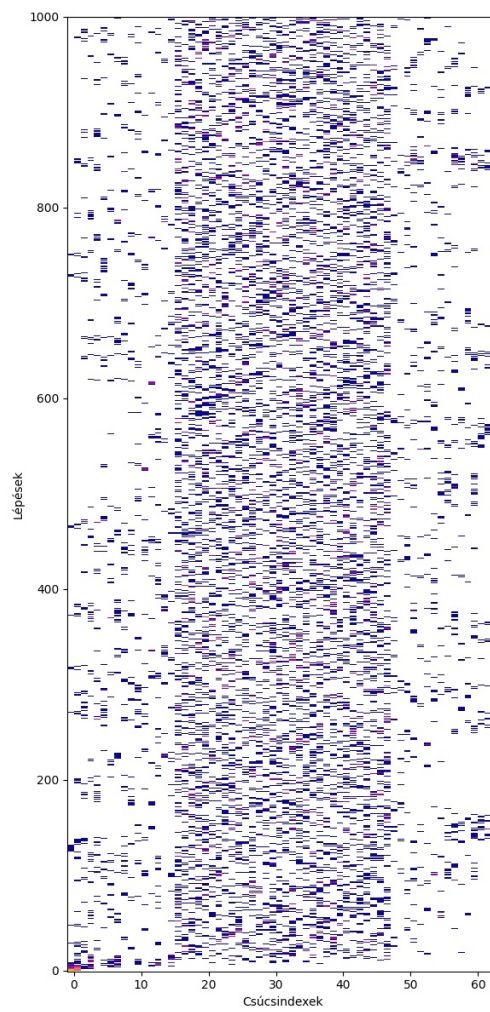
6.5. ábra. Ragasztott bináris gráf részgráfjai



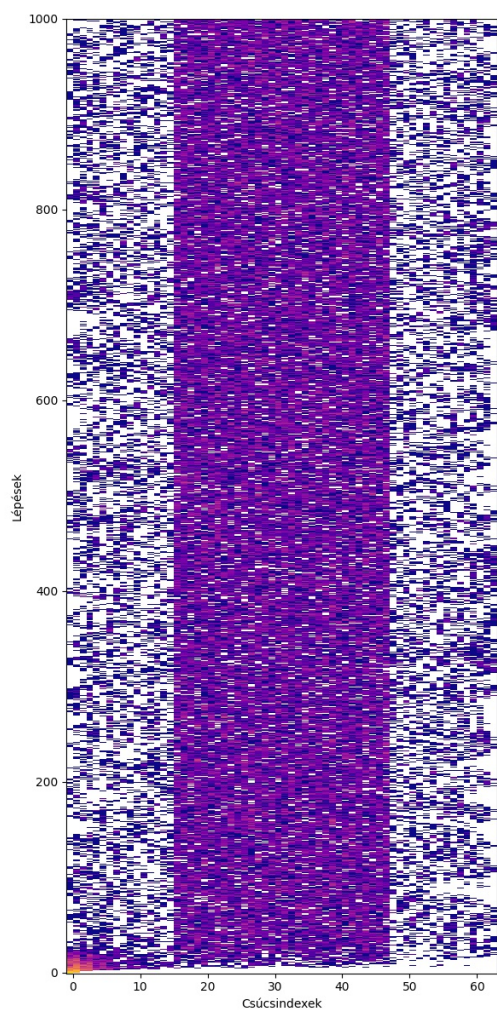
6.6. ábra. Ragasztott bináris gráf



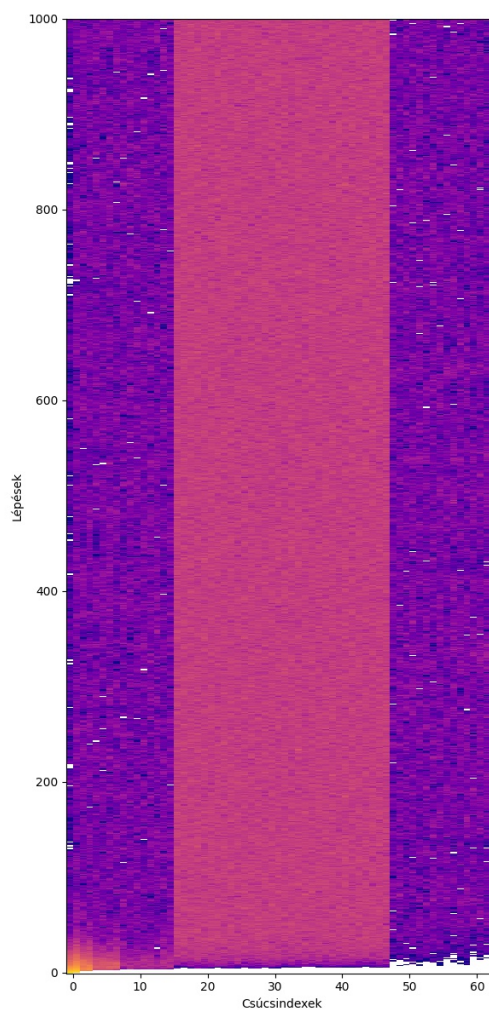
(a) 1 bolyongó



(b) 10 bolyongó



(a) 100 bolyongó



(b) 1000 bolyongó

7. fejezet

Jövőbeli tervek

- Elolvasni a Springer sorozatos kvantum bolyongós könyveket.
- Implementálni k -reguláris gráfra a bolyongást.
- Implementálni általános gráfra a bolyongást.
- Számolni hitting time-ot és mixing rate-t.
- Számolni a szomszédossági mátrixból sajátértékeket és azokat elemezni.
- Számolni határeloszlást a Markov-láncokhoz.

Források (rendesen hivatkozva majd...):

- Hirvensalo könyv