

XXXI. Országos Tudományos Diákköri Konferencia
——— Kémiai és Vegyipari Szekció ———

ÁRENDÁS PÉTER
MATEMATIKUS MESTERSZAK
II. ÉVFOLYAM

Gráfelméleti módszerek a molekulaspektroszkópiában

Témavezetők:

Dr. Császár Attila
tanszékvezető egyetemi tanár
Fizikai Kémiai Tanszék

Dr. Furtenbacher Tibor
tudományos munkatárs
Fizikai Kémiai Tanszék

——— Eötvös Loránd Tudományegyetem ———
——— Természettudományi Kar ———
—— Fizikai Kémiai Tanszék ——
—— Budapest, 2012. ——

Köszönetnyilvánítás

Ezúton is szeretném megköszönni azt a segítséget, amit dr. Császár Attila és dr. Furtenbacher Tibor nyújtott a kutatási téma tanulmányozása során, valamint azokat a hasznos tanácsokat, melyek nélkül jelen formájában nem jöhetett volna létre ez a dolgozat.

Kiemelt köszönettel tartozom továbbá támogatásukért és segítségükért családomnak, valamint Ivancsó Veronikának és Stark Andrásnak.

Köszönet illeti továbbá dr. Szabó Csabát, illetve Biszak Elődöt, Dosztály Katinkát, Kis-Benedek Ágneszt, Kiss Attilát és Zamostny Róbertet, hogy tanácsaikkal hozzájárultak a dolgozat elkészüléséhez.

Tartalomjegyzék

Köszönetnyilvánítás	2
Tartalomjegyzék	3
Ábrák jegyzéke	4
1. Bevezetés.....	5
2. A H_2^{16}O molekula spektroszkópiai hálózata.....	8
3. Gráfelméleti megalapozás	9
3.1 Alapfogalmak.....	9
3.2 Algoritmusok jellemzése	10
3.3 Használt módszerek	12
4. Kísérleti rész.....	16
4.1 A H_2^{16}O molekula mért spektroszkópiai hálózatának összefüggővé tétele.....	16
4.2 A mért hálózat bővítése új energiaszintekkel	17
5. Összefoglalás.....	22
5.1 Az eredmények értékelése	22
5.2 Kitekintés, kérdésfelvetések	23
Függelék	26
A mellékelt CD lemez tartalma	26
Perl programfájlok.....	27
mindketvege.pl	27
csakazegyikvege.pl	28
cimkezd.pl	29
visszacimkezd.pl.....	30
csakkettot.pl.....	30
Matlab programfájlok	31
kruskal.m	31
fysalida.m	32
iscircle.m	32
Irodalomjegyzék	33

Ábrák jegyzéke

1. ábra – A Depth First Search algoritmus szemléltetése	13
2. ábra – A Kruskal-algoritmus szemléltetése	15
3. ábra – Új átmenet két ismert, címkézett energiaszint között	16
4. ábra – Feszítőfa építése új energiaszintek elérésére	18
5. ábra – Új energiaszintek elérése	19
6. ábra – Iteráció szemléltetése, két egymást követő lépés	20
7. ábra – A mért hálózat bővítésére adott javaslat eredményei – energiaszintek száma	21
8. ábra – A mért hálózat bővítésére adott javaslat eredményei – átmenetek száma	21
9. ábra – Minimális súlyú feszítőfa és minimális átlagos távolságú feszítőfa közti különbség szemléltetése	24

1. Bevezetés

Köszönhetően a kvantumkémiai módszerfejlesztéseknek, valamint a rendelkezésre álló számítástechnikai eszközök exponenciális ütemű fejlődésének, ma már a kémia számos területén nyílik lehetőség az elméleti háttértudás segítségével új, kézzelfogható, gyakorlati eredmények elérésére, vagy azok lényegi kiegészítésére.

A nagyfelbontású molekulaspektroszkópia az egyik ilyen területe a kémiának. A tématerület egyik fontos kihívása, hogy a kísérleti módszerekkel meghatározható színekpek esetében gyakran megoldhatatlan feladatot jelent az összes rezgési-forgási átmenet asszignálása. További súlyos nehézség, hogy a gyakorlati, például termokémiai alkalmazások gyakran rendkívül nagyszámú spektroszkópiai adat meglétét igénylik, ezek száma már néhány atomos molekulák esetén is milliós nagyságrendű. A kvantumkémia eszköztárával azonban lehetséges olyan *ab initio* számítások végzése, melyekkel a molekuláris energiaszintek, a molekulaszínekpek, valamint a termokémiai tulajdonságok meghatározhatóak.

Dolgozatom központi témája, hogy a H_2^{16}O molekula, mint tesztrendszer esetén vizsgálom, hogy a kísérletekből meghatározható energiaszintek számát milyen módszerekkel tudjuk optimálisan megnövelni. A víz molekula fontosságát talán nem is kell hangsúlyozni, hiszen színekpe a mérnöki tudományoktól kezdve az asztrofizikai méréseken át a légkörmodellezésig szinte mindenhol kiemelkedő jelentőséggel rendelkezik. Ezért is fontos a H_2^{16}O molekula rezgési-forgási energiaszintjeinek minél pontosabb ismerete.

A jelenleg elérhető legnagyobb (energiaszint és átmenet) adatbázis a H_2^{16}O molekulára az ún. BT2 adatbázis [18]. Ez a variációs magmozgás számításoknak köszönhetően szinte teljes energiaszint listát tartalmaz, ám azok pontossága (főleg az alsóbb régiókban) messze elmarad a spektroszkópiai (kísérleti) pontosságtól. Ezért is merült fel az igény egy új alapokon nyugvó eljárás kifejlesztésére, ez az eljárás lett a MARVEL (Measured Active Rotational Vibrational Energy Levels) [5]. A MARVEL eljárás kapcsán került bevezetésre a spektroszkópiai hálózatok fogalma [1-3]. A spektroszkópiai hálózat (spectroscopic network, SN) fogalmát 2007-ben vezette be Császár és Furtenbacher [6]. A struktúra az energiaszintekből, mint csúcsokból, és a köztük megadott átmenetekből, mint élekből egy nagy hálózat mintájára épül fel. A csúcsok közötti élek súlyát a mért, illetve számolt (abszorpció vagy emisszió) intenzitás szolgáltatja.

A MARVEL egy olyan Hamilton-operátor mentes, inverziós eljárás, amely alkalmas a kísérletileg mért átmenetekből kísérleti pontosságú rezgési-forgási energiaszintek meghatározására. Az eljárás kulcsa tehát a kísérletileg észlelt, asszignált (címkével ellátott) átmenetek adatbázisa.

Az adatbázis, azaz a gráf növelésére három lehetőségünk van:

- (1) Már ismert, de új mérésből származó átmenet hozzáadása. Ez a lehetőség jár a legkevesebb hasznos eredménnyel, esetlegesen csak az átmenetben résztvevő energiaszintek bizonytalansága csökkenhet.
- (2) Új, eddig nem ismert átmenet hozzáadása, mely nem eredményez új energiaszintet. Az ilyen átmenetek fontossága abban rejlik, hogy ha ciklust (kört) zár be a hálózatban, az jelentősen csökkenti a körben található energiaszintek bizonytalanságát.
- (3) Új, eddig nem ismert átmenet hozzáadása, mely új energiaszintet határoz meg. Az ilyen átmeneteknek van a legnagyobb jelentősége, ugyanis új energiaszint meghatározására adnak lehetőséget.

Itt hívjuk fel a figyelmet a (3) eset egy különleges típusára. A mért átmenetek adatbázisa gyakran tartalmaz ún. lebegő hálózatokat (floating spectroscopic network, FSN) [1], melyeknek nincs kapcsolata a gyökereket tartalmazó komponensekkel, így a bennük résztvevő energiaszintek értékét nem tudjuk abszolút értékeben meghatározni. Ám amennyiben tudnánk olyan átmeneteket találni, melyek összekötik ezen FSN-ket az alaphálózattal, máris alkalmunk nyílna az energiaszintek meghatározására. Köszönhetően a nagy hálózatok és a gráfok megalapozott matematikai hátterének, lehetőségünk nyílik a mért és számított spektroszkópiai hálózatok gráfelméleti módszerekkel történő vizsgálatára.

TDK munkám során tehát két főbb problémával foglalkoztam. Vizsgáltam egyrészt, hogy a jelenlegi mért energiaszintek által meghatározott hálózatot milyen módon tudjuk optimalizálni, azaz komponenseit összefüggővé tenni, valamint jelentős intenzitású, eddig nem asszignált átmeneteket próbáltam keresni az energiaszintek között. A feladat megoldása során gráfelméleti algoritmusokat alkalmaztam, főként egy minimális súlyú feszítőfát előállító eljárást. Az algoritmikus vizsgálat megvalósítására saját programokat írtam.

Ezt követően új energiaszintek asszignálására kerestem módszereket. Megvizsgáltam, hogy az előző feladatra használt eljárás miért nem optimális ebben az esetben. A következtetések levonása után egy másik módszert alkalmaztam, ennek során adatbányászati

eszközökkel nyertem ki a hasznos információkat a jelentős méretű forrásfájlokból. Hasonlóan az előző feladathoz, ebben az esetben is saját adatbányász programokat készítettem.

A két feladat megoldására kapott eredményeket tekintve törekedtem arra, hogy minél áttekinthetőbb, illetve jobban kezelhetőbb formátumban közöljem azokat, és *txt* fájlokban tároltam el az új információkat.

A dolgozat programozási háttere *Perl* nyelven, valamint a *MathWorks Matlab* szoftver segítségével készült. Az ábrák a *yEd Graph Editor* programban készültek.

A dolgozat során említett algoritmusok, illetve a mellékelt programkódok segítségével nagyobb hálózatok is optimálisan elemezhetőek, azonban a megnövekedett méret magasabb futásidőt eredményez, ezért a programok futtatása klaszterhálózaton javasolt.

Fontos megjegyezni, hogy míg a *Perl*-ben írt programok az itt alkalmazottaknál nagyobb bemeneti adatbázisokat is jól kezelnek, csak futásidejük nő, a *Matlab* esetén nagyobb méretű inputok megfelelő virtuális memória hiányában olyan hibát eredményezhetnek, amely meggátolja a program lefutását. Amennyiben nincs lehetőségünk, vagy nem szeretnénk klaszterre írni a programkódot, javasolt az algoritmust C++ vagy hasonló nyelven implementálni.

2. A H_2^{16}O molekula spektroszkópiai hálózata

A molekula spektroszkópiai hálózatának vizsgálata során felhasznált adatbázisokat a mellékelt CD lemez tartalmazza. A lemezen található adatok szerkezete és tartalmi ismertetése a Függelék első részében található. A dolgozathoz mellékelt CD lemezen találhatók meg a két hálózat adatbázisai. Formátumukat az alábbi táblázat tartalmazza.

<i>fájlnév</i>	<i>kiterjesztés</i>	<i>leírás</i>
transitions_vertex	txt	A mért spektroszkópiai hálózat átmenetei. <u>Formátum:</u> hullámszám, bizonytalanság, felső energiaszint BT2-indexe, alsó energiaszint BT2-indexe, referencia.
BT2_30000	txt	Az ab initio spektroszkópiai hálózat átmenetei 30.000 cm^{-1} -ig. <u>Formátum:</u> frekvencia, intenzitás, felső energiaszint BT2-indexe, alsó energiaszint BT2-indexe.

Áttekintésként elmondható, hogy a H_2^{16}O molekula spektroszkópiai hálózatát 99.335 mért átmenet alkotja, 183.437 különböző publikált átmenet eredményeként. Ezek az átmenetek 17.690 különböző energiaszint között teremtenek kapcsolatot.

Ab initio számítások segítségével ennél jóval nagyobb méretű adatbázist tudunk előállítani. A teljes adatbázis mintegy 505.806.202 átmenetet és 221.097 energiaszintet tartalmaz. Ezen hatalmas számú átmenetből azonban csupán elenyésző részt sikerült kísérletileg észlelni. Itt jegyezzük meg, hogy a dolgozat kizárólag az egyfotonos abszorpcióval foglalkozik, így megállapításai nem vonatkoznak a multifotonos, illetve emissziós spektroszkópiai technikákra.

Első lépésként a BT2 adatbázisból kiválogattam azon átmeneteket, amelyek kísérletileg esetlegesen észlelhetőek. A szobahőmérsékletű intenzitás értékre küszöbszámnak a 10^{-30} cm/molekula értéket választottam. Az így kapott, már kísérletileg észlelhetőnek tekintett hálózatban 397.146 különböző átmenetet határozhatunk meg, melyek segítségével 44.843 energiaszintet érhetünk el. A jelenleg rendelkezésre álló kísérleti adatbázis tehát az átmenetek és az energiaszintek mintegy felét tartalmazza.

3. Gráfelméleti megalapozás

3.1 Alapfogalmak

Gráf alatt egy csúcsokból, valamint csúcspárok között futó élekből álló struktúrát értünk. Amennyiben a csúcshalmazt V -vel, a rajtuk értelmezett élek halmazát pedig E -vel jelöljük, akkor egy G gráfot megadhatunk $G=(V, E)$ alakban. A csúcsok számát $|V|$, az élek számát pedig $|E|$ jelöli.

Az élhalmazt tekintve megkülönböztetünk egyszerű és összetett gráfokat. A gráf egyszerű, ha nem tartalmaz hurokért és többszörös élt, azaz minden élének mindkét végpontja különböző csúcs, valamint tetszőleges két csúcsa legfeljebb egy élt feszít.

Egy gráf éleit tekintve beszélhetünk irányított, illetve irányítatlan gráfokról. Utóbbi esetben nem teszünk különbséget aközött, hogy x és y csúcsok esetén egy e él x -ből y -ba vezet, vagy y -ból x -be, irányított gráf esetén azonban megkülönböztetjük a két esetet. A dolgozat során ezentúl gráf alatt minden esetben irányítatlan gráfot értünk.

A gráf egy útja alatt élek egy olyan sorozatát értjük, ahol két egymást követő él esetén az előző él végpontja megegyezik a következő él kezdőpontjával, és minden él legfeljebb egyszer szerepel a sorozatban. Ha megengedjük, hogy egy élt többször is felhasználjunk, sétáról beszélünk.

Összefüggőnek nevezünk egy irányítatlan gráfot, ha tetszőleges két csúcsa között találunk utat, azaz minden csúcsból minden csúcsba eljuthatunk. (Irányított élek esetén megkülönböztetünk erős, illetve gyenge összefüggőséget, ennek definiálására azonban itt nem térünk ki.)

Egy gráf komponensei alatt tartalmazásra nézve maximális méretű összefüggő részgráfjait értjük. Szemléletesen tehát egy komponensen belül tetszőleges két csúcs között találunk utat, és a komponens tetszőleges csúcsából elérhető összes csúcs benne van a komponensben. Amennyiben a teljes gráf összefüggő, úgy komponenseinek száma 1.

Fa alatt körmentes gráfot értünk, azaz olyan gráfokat, melyek tetszőleges két csúcsát pontosan egy út köti össze. Ha a gráf körmentes, de nem összefüggő, akkor erdőről beszélünk.

Súlyozott gráfok esetén az élekhez értékeket, általában valós számokat rendelünk. Szemléletesen ezek a súlyok legtöbbször az adott él hosszát vagy „költségét” reprezentálják.

A spektroszkópiai hálózatok gráfelméleti tulajdonságait [1-2] írja le, ezek tehát jellemzik a vizsgálatunk tárgyát adó H_2^{16}O molekula SN-jét is. Az áttekinthetőség kedvéért itt csak azokat a tulajdonságokat tüntetjük fel, melyek szükségesek a kitűzött feladatok, illetve az azokra adott megoldási módszerek és eredmények megértéséhez.

Az SN-ek irányítatlan, súlyozott, egyszerű gráfok. Csúcsaik az adott molekula energiaszintjeinek, élei pedig az energiaszintek közti átmeneteknek felelnek meg. A hálózat élein értelmezett súlyfüggvény az átmenet intenzitását reprezentálja.

További jellemző tulajdonságuk, hogy ún. skálafüggetlen hálózatok. Ez azt takarja, hogy az SN-ek fokszámoszlása a Yule-Simon-eloszlást követi, azaz annak a valószínűsége, hogy egy csúcs k -ad fokú, $P(k) \sim k^{-\gamma}$, ahol γ egy paraméter. Másképp úgy fogalmazhatunk, hogy a hálózat nagy foksámú csúcsai általában össze vannak kötve egymással.

A spektroszkópiai hálózatokkal kapcsolatos problémák vizsgálata esetén néhány esetben szükség lehet az előbbi modell részleges megváltoztatására. Az 5.2 részben többek közt olyan kérdést is felvetünk, mely megoldásához a súlyfüggvény olyan módosítását javasoljuk, hogy a függvényt az adott átmenet intenzitása helyett a mérési bizonytalanság adja.

3.2 Algoritmusok jellemzése

A gráfokon futó különféle algoritmusok tárgyalásánál több, gyakorlati szempontból is jelentős tulajdonságot kell szem előtt tartanunk. Ezek közül a legfontosabbak az algoritmus futási ideje, valamint a szükséges informatikai erőforrások mérete.

Algoritmusok futási ideje

Futásidő jellemzésére a nagy ordó („ O ”) függvényt használjuk, amely n méretű bemenet (input) függvényében adja meg az algoritmus lefutásához szükséges időt. Ez az idő nagyban függ attól, hogy milyen környezeten, milyen rendelkezésre álló erőforrások mellett fut egy program. Ebből a megfontolásból alakult ki, hogy algoritmusok összehasonlítása során az input függvényében szükséges lépések arányát tekintjük mérvadónak.

A függvény szemléletesen lineáris idejű algoritmusok esetén például $O(n)$. Kétszer nagyobb bemenet esetén kétszer hosszabb lesz a lefutáshoz szükséges idő. Négyzetes algoritmusok esetén $O(n^2)$ a futásidő, tehát kétszer akkora input mellett az eljárás négyszer

több idő alatt ér véget. Azokra az esetekre, ahol a szükséges idő független a bemenet méretétől, $O(1)$ (konstans futásidő) jelölést használunk.

Gyakorlati szempontból a legfeljebb $O(x \log x)$ nagyságrendű eljárások számítanak gyorsnak, míg az $O(n^2)$, $O(n^3)$, ..., $O(2^n)$ esetekben az eljárások csak nagyon kis inputméretek esetén lehetnek hatékonyak.

Memória

A gyorsaság mellett szintén sarkalatos kérdés az algoritmustervezés során, hogy milyen erőforrásokra van szükség a futáshoz. Ezek közül az egyik legfontosabb a számítógép memóriakapacitása, amely meghatározza, hogy az utasítássorozat közben mennyi adatot vagyunk képesek „fejben tartani”.

Egy gráfot, hálózatot többféle struktúrában is megadhatunk. Egyik leginkább használt mód, hogy ún. incidenciamátrix formát használunk. A mátrix sorai és oszlopai a gráf csúcsainak felelnek meg, így $|V| = n$ csúcs esetén $n \times n$ méretű lesz. Jelöljük a mátrix i . oszlopának j . elemét m_{ij} -vel. Ennek értéke 0, ha az i és j csúcsok közt nem fut él, egyébként pedig az adott él súlyával egyenlő.

Megadhatunk emellett egy hálózatot éllistas alakban is. Amennyiben az adott gráf $|E|=m$ éllel rendelkezik, az éllistas ábrázolás egy $3 \times m$ méretű mátrixot jelent. Ez soronként felsorolja az éleket, első két oszlopában az él két végpontját, harmadik oszlopában pedig az adott élhez tartozó súlyt tartalmazza.

Mindkét említett reprezentáció más-más előnyökkel bír. Arányaiban kevés éllel rendelkező gráfok ritka incidenciamátrixokat jelentenek, azaz a mátrix elemeinek szignifikáns része 0 lesz. Ez sok esetben indokolatlanul igényelhet nagy tárhelyet. Szintén számításba kell venni, hogy a gráfelméleti algoritmusok a legtöbb esetben preferálnak egy bizonyos adattípust, és más inputra nem optimális gyorsasággal futnak.

A dolgozatban szereplő módszerek esetében főként az éllistas reprezentációt választottuk, mivel ez az ábrázolásmód amellet, hogy az itt használt algoritmusok esetén hatékony inputot ad, sokkal kevesebb memóriát is igényel a tárolásra, így csökkenthetjük az adott program futási idejét. A dolgozatban felvetett kérdésekre adott ilyen típusú eredményeket szintén éllisták formájában közöljük, a minél könnyebb áttekinthetőség érdekében.

A szükséges memóriával kapcsolatban meg kell jegyeznünk, hogy bár esetünkben az előbbi, gráfok tárolására vonatkozó tárhelyprobléma a jellemző, a feladat változtatása más

korlátokat helyez előtérbe. Számításaink során ugyanis kizárólag maximum százezres nagyságrendű egész számokkal dolgoztunk, mivel az átmenetek esetén csak az intenzitás nagyságrendjével számoltunk. Ha olyan módosítást szeretnénk, amely sok helyiértékes számot kezel, különösen a *MathWorks Matlab* szoftver használata közben problémaként merülhet fel ezen értékek eltárolása a memóriában. Javasolt ebben az esetben a program klaszterre optimalizálása.

3.3 Használt módszerek

A dolgozatban említett és vizsgálataim során használt fontosabb eljárásokat ebben a szakaszban ismertetjük.

Felmerülhet a kérdés, miért van szükség algoritmikus vizsgálatra olyan problémák vizsgálatakor, melyek látszólag egyszerű próbálgatással is megoldhatóak. Az itt tárgyalt megoldási módszereknek több előnye is van.

Egyrészt nem szabad elfelejtenünk, hogy az itt felhasznált adatbázisok a legtöbb esetben legalább százezres nagyságrendű sort tartalmaznak. Ezeken kézi rendezéssel és számításokkal legfeljebb csak már megadott megoldások tesztelése, ellenőrzése javasolt. A Függelékben, valamint a CD mellékleten található programok segítségével azonban rövid idő alatt, gyorsan kaphatunk teljes körű megoldást a dolgozatban tárgyalt kérdésekre.

További előnye az algoritmusokkal történő megközelítésnek, hogy könnyen fejleszthetőek tovább, így ha változtatjuk a felvetett problémát, a hozzá tartozó megoldás kereséséhez sok esetben nem lesz szükséges új eljárás kifejlesztése, csak a meglevő módszer módosítása. Ugyanez igaz abban az esetben is, ha egy másik spektroszkópiai hálózat esetén szeretnénk vizsgálni az itt tárgyalt kérdéseket.

Gráfok bejárása

Gráfok, hálózatok bejárása során gyakran merülnek fel olyan kérdések, melyekhez a gráf szerkezetét kell „felderítenünk”, csúcsait, éleit valamilyen rendszer szerint végignéznünk, esetleg egy előre megadott tulajdonságot keresve. Ilyen attribútum lehet például bizonyos fokszerű csúcsok megtalálása, vagy a hálózatban körök keresése.

Spektroszkópiai vizsgálatok során gráfbejárési eljárások segítségével azt szeretnénk eldönteni, hogy egy adott mért hálózat összefüggő-e, illetve amennyiben nem, hány

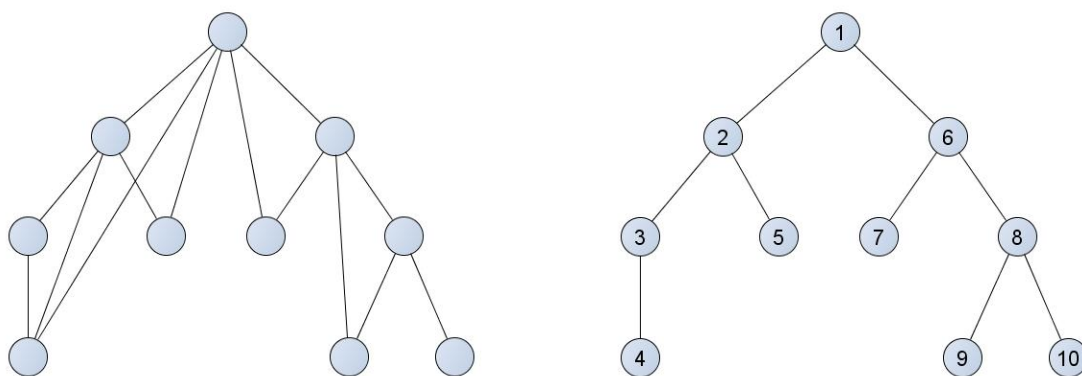
komponens alkotja. A feladat megoldására két algoritmust javasol a szakirodalom: a Depth First Search (DFS, mélységi keresés) [11] és a Breadth First Search (BFS, szélességi keresés) eljárást. A DFS algoritmus máshol is alkalmazott a szakirodalomban, [3]-ban hasonló célokra szintén szerepel ez a bejárési módszer.

A DFS eljárás lényege, hogy a gráf egy tetszőleges csúcsából indulva addig lépkedünk a még nem bejárt szomszédos csúcsok között visszalépés nélkül, amíg vagy megtaláltuk a keresett csúcsot, vagy az aktuális csúcsnak már nincs nem bejárt szomszédja. Ha nincs bejáratlan szomszédos csúcs, akkor egyet visszalépve ismét új szomszédos csúcsot keresünk. Ezt addig iteráljuk, míg a gráf (vagy nem összefüggő gráf esetén az adott komponens) összes csúcsát be nem jártuk, vagy nem találtuk meg a keresett csúcsot.

A BFS algoritmus elve hasonló, azzal a különbséggel, hogy egy adott csúcsból először bejárja annak összes még nem érintett szomszédját, majd ezen csúcsok egyikére lép tovább, és járja be annak szomszédait. Ha nincs bejáratlan szomszéd, az előzőleg bejárt csúccsal folytatódik az eljárás.

A spektroszkópiában alapvető feladat annak vizsgálata, hogy a mért spektroszkópai hálózat összefüggő-e. Gyakran azt szeretnénk elérni, hogy tetszőleges csúcsból bejárható legyen a gráf minden csúcsa. Szerencsére mindkét módszer, a DFS és a BFS lépésgénye is $O(|V|+|E|)$, azaz kicsi. A vizsgálatunk során a mélységi keresést fogjuk használni.

Az 1. ábrán bemutatott gráf kicsiben szemlélteti a DFS algoritmus működését:



1. ábra – A Depth First Search algoritmus szemléltetése

Minimális súlyú feszítőfa előállítása

Számos olyan gyakorlati probléma létezik, melyek megoldásához minél kevesebb éllel szeretnénk lefedni vagy elérni a feladatot modellező gráf csúcsait úgy, hogy ezek összszúlya a

lehető legkisebb legyen. Ilyen típusú kérdések gyakran fordulnak elő többek közt a logisztikai tervezés, vagy a térinformatikai rendszerek fejlesztése során.

Súlyozott gráfokon értelmezhetünk minimális súlyú feszítőfát, mely az eredeti gráf éleinek olyan részhalmazát adja meg, melyek összsúlya minimális, tartalmazásra nézve maximális, és tetszőleges két csúcsa között legfeljebb egy éldiszjunkt út fut, azaz nem tartalmaz kört.

Minimális súlyú feszítőfa keresésére több gyors algoritmus létezik, közülük a két legismertebb Joseph Kruskal amerikai matematikus 1956-ban publikált módszere [13], valamint Robert C. Prim 1957-ben kifejlesztett eljárása [14]. Mivel nagyságrendileg a vizsgált spektroszkópiai hálózat élsűrűsége nem nagy, a Kruskal-algoritmust implementáltuk a feszítőfa kiszámítására.

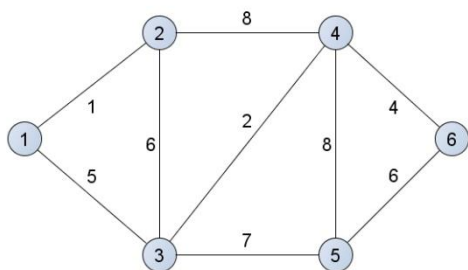
Az eljárás súlyozott élekkel ellátott irányítatlan gráfokon fut, mohón. Mohó algoritmus alatt olyan utasítássorozatot értünk, mely lokális optimalizálás segítségével keresi a globális megoldást, azaz lépéseiben mindig az aktuálisan legjobbnak tűnő lehetőséget választja. A Kruskal-algoritmus hasznos tulajdonsága, hogy amennyiben nem összefüggő hálózatra futtatjuk, nem jelez hibát, hanem feszítő erdőt hoz létre.

Az eljárás során kezdetben tekintsük azt a gráfot, melynek csúcsai az eredeti gráf csúcsai, élei viszont nincsenek. Új élek hozzávételével fogjuk felépíteni a minimális súlyú feszítőfát. Az algoritmus lépései:

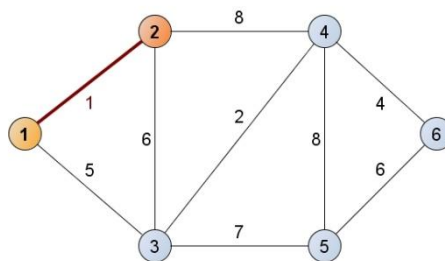
1. Tekintsük a legkisebb súlyú olyan élt, melyet még nem vizsgáltunk.
2. Adjuk hozzá az élt az épülő feszítőfához, amennyiben hozzáadásával nem keletkezik kör.

Ezt a két lépést iteráljuk az eredeti gráf minden élére. Az eredményként kapott feszítőfa összefüggő, $|V|=m$ csúcsú gráf esetén $m-1$ élből fog állni. Nem összefüggő gráf esetén a feszítőfa $m - k$ élű, ahol k a komponensek számát jelöli. Az algoritmus futási ideje $O(|E| \log |V|)$, tehát gyors.

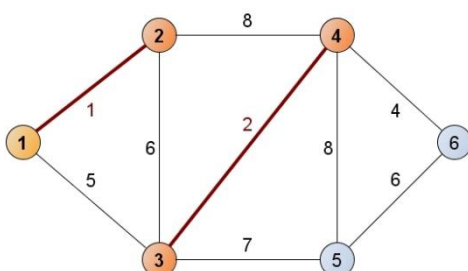
A Kruskal-algoritmus futását a 2. ábra segítségével szemléltetjük.



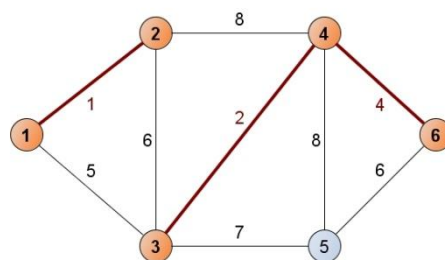
Kiindulási gráf, a feszítőfa élhalmaza üres.



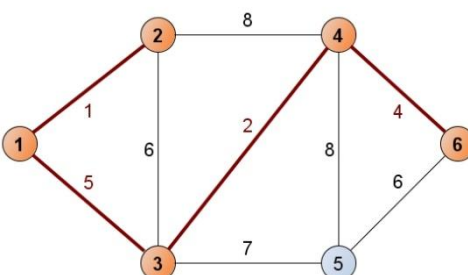
1. lépés: a vastagon jelölt él a legkisebb súlyú, ezért hozzávesszük a feszítőfához, így az 1-es és 2-es csúcsok már elérték lesznek.



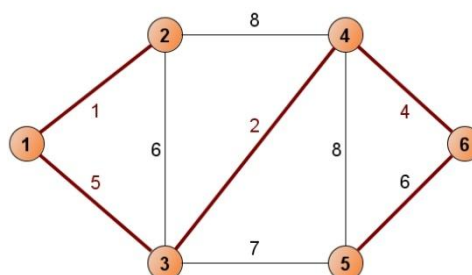
2. lépés: a következő legolcsóbb él nem feszít kört, ezért a feszítőfába kerül, így már elértük a 3-as és 4-es csúcsokat.



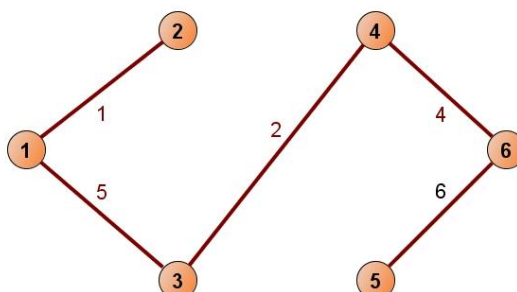
3. lépés: a következő él sem feszít kört, beválasztásával elérjük a 6-os csúcsot.



4. lépés: az 1-3 él feszítőfába emelése. Új csúcsot nem értünk el.



5. lépés: az utolsó él beválasztásával elértük az 5-ös csúcsot is. Mivel a 2-3 él hozzávételével kört kaptunk volna, így ez nem került a fa élei közé.



A végeredményként kapott minimális súlyú feszítőfa.

2. ábra – A Kruskal-algoritmus szemléltetése

4. Kísérleti rész

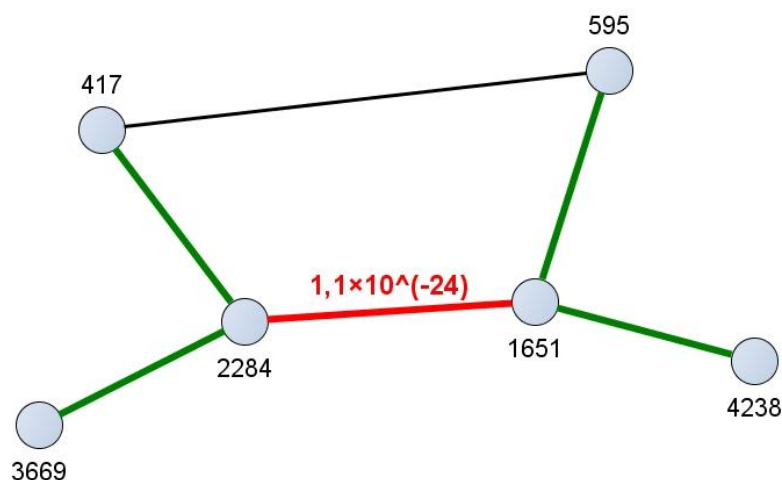
4.1 A $H_2^{16}O$ molekula mért spektroszkópiai hálózatának összefüggővé tétele

Az első részfeladat kiindulási alapja a $H_2^{16}O$ molekula mért spektroszkópiai hálózata, mely nem összefüggő, több komponensből áll. Ezt a hálózatot szeretnénk összefüggővé tenni, az *ab initio* SN-ben meglevő, de a mért hálózatban nem szereplő átmenetei segítségével. A tesztfeladat megoldása során a mért, illetve a számított spektroszkópiai hálózatok élein 1000 cm^{-1} -es vágást vettünk, azaz nem vizsgáltuk az ennél nagyobb hullámszámú átmeneteket.

A cél eléréséhez először építsük fel azt a gráfot, mely a mért hálózat csúcaiból áll, élhalmazát pedig a számított hálózat átmenetei közül azok adják, melyek két mért hálózatbeli csúcs között futnak. Az így kapott gráf éleire azt a súlyfüggvényt vesszük fel, mely az adott intenzitás nagyságrendjének (-1) -szerese. Erre a gráfra egy Kruskal-féle minimális súlyú feszítőfa algoritmust futtatunk, melynek működését részletesebben a 3.3 részben mutattuk be.

A Kruskal-algoritmust a *MathWorks Matlab* szoftver segítségével implementáltuk, mely program kódja a Függelékben, valamint a mellékelt CD lemezen található. Ezt követően néhány, a CD-n szintén megtalálható Perl program futtatása szükséges.

A 3. ábra a használt módszer által talált egyik új átmenet mért spektroszkópiai hálózathoz történő hozzávételét szemlélteti. A 2284-es és az 1651-es BT2-indexű csúcsok közti, a mért hálózatban nem szereplő átmenet intenzitása $1,1 \times 10^{-24}\text{ cm/molekula}$, hullámszáma $52,15962\text{ cm}^{-1}$ értékű. A feszítőfa-algoritmus által megtalált, mért átmeneteket zöld színnel jelöltük.



3. ábra – Új átmenet két ismert, címkézett energiaszint között

Az eljárás során az 1000 cm^{-1} -ig vágott spektroszkópiai hálózatokon a feszítőfa éleire eredményül 3.587 átmenetet kaptunk, melyekkel a hálózat csúcsai, azaz az energiaszintek közül 3.612-t fedtünk le. Az átmenetek adatbázisa éllistás formátumban, BT2 indexekkel a mellékelt CD lemezen *feszitofaBT2.txt* néven található.

A kiindulási hálózatunk 6.360 energiaszintet tartalmazott. Annak oka, hogy nem sikerült minden csúcsot lefednünk a fenti algoritmussal, abban keresendő, hogy az 1.000 cm^{-1} értékű vágás miatt sok hasznos átmenet még nem szerepel a teljes SN itt szereplő kezdőszeletében.

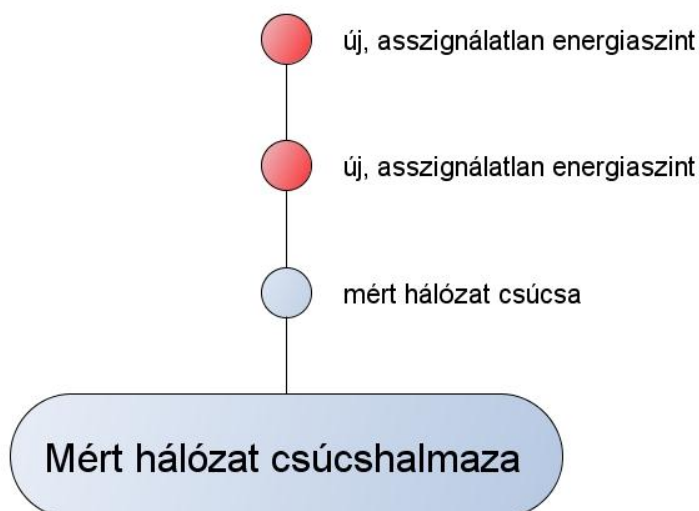
Várakozásaink szerint az algoritmust nagyobb kezdőszeletre futtatva olyan éllistát kapunk, mely lehetővé teszi a mért hálózat összefüggővé tételét. Ez többek közt összefügg [1]-ben közölt eredményekkel, mely szerint a molekula spektroszkópiai hálózata nagy ütemben esik szét, ha az átmenetek legalább 85%-át véletlenszerűen töröljük, előtte viszont nagymértékben összefüggő marad.

4.2 A mért hálózat bővítése új energiaszintekkel

Az *ab initio* számítások a 30.000 cm^{-1} -es vágás mellett 29.545-tel több energiaszintről tartalmaznak adatokat a mért hálózat csúcsaihoz képest. Ebben a részben azt elemezzük, milyen eljárás segítségével adhatunk mérési javaslato(ka)t arra, miként emelhetünk át új, a mért hálózatban még nem szereplő energiaszinteket az *ab initio* hálózat átmeneteit felhasználva.

Felmerülhet az előző részben tárgyalt feszítőfaépítő eljárás alkalmazása erre a problémára is, a teljes *ab initio* hálózatra futtatva. Ez biztosítaná, hogy amennyiben összefüggő hálózatból indulunk ki, a lehető legnagyobb intenzitású átmenetek segítségével lehetünk képesek összekötni az *ab initio* energiaszinteket. A kísérleti spektroszkópikus szempontból azonban nem ideális egy olyan eljárás, amely az új energiaszinteket több más, szintén új energiaszinten keresztül éri el.

A 4. ábra szemlélteti a feszítőfa használatának hátrányát: ez az eljárás olyan eseteket is megenged, hogy asszignálatlan energiaszintet másik, asszignálatlan energiaszinten át vizsgáljunk.

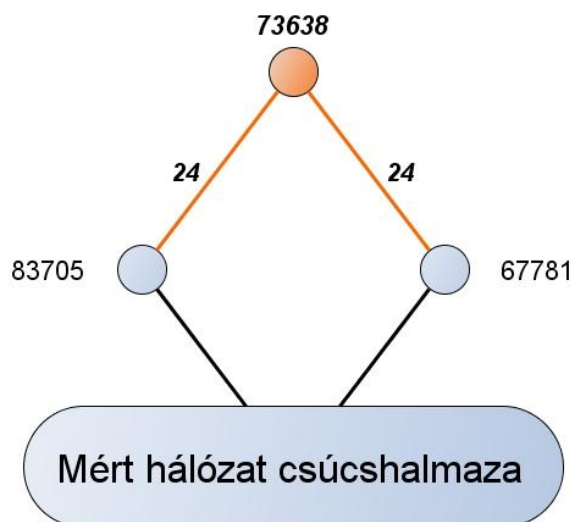


4. ábra – Feszítőfa építése új energiaszintek elérésére

Ebből a megfontolásból a probléma megoldásához ezúttal egy iteratív algoritmust adunk meg. Ennek lényege, hogy tekintjük az *ab initio* energiaszintek közül a mért hálózat csúcsainak megfelelő részgráfot, és egy lépésben az egyetlen élen elérhető szomszédos csúcsokat vesszük hozzá a lehető legnagyobb intenzitású átmenetekkel elérve a hálózathoz. Egy általános lépésben az aktuálisan elértnek tekintett csúcsok szomszédait, és az optimális új átmeneteket fogjuk a rendszerbe emelni.

Szintén gyakorlati előny az asszignáció során, ha egy új energiaszintet két különböző átmeneten keresztül is el tudunk érni, hiszen ez az energiaszint meghatározásának bizonytalanságát csökkenti. A megoldás során ezért minden új csúcs hozzávételekor, ha lehetőség nyílik rá, két nagy intenzitású élt adunk meg.

Az eljárást az 5. ábrán szereplő hálózati példa szemlélteti, az egyes csúcsok címkéi a H_2^{16}O molekula spektroszkópai hálózat energiaszintjeinek BT2 indexei. A példa a 73638-as kódú energiaszint hozzávételét mutatja. Az éleken található súlyok az átmenetek intenzitását jelentik, azaz itt mind a két új átmenet esetén 10^{-24} -es nagyságrendű élekről van szó.

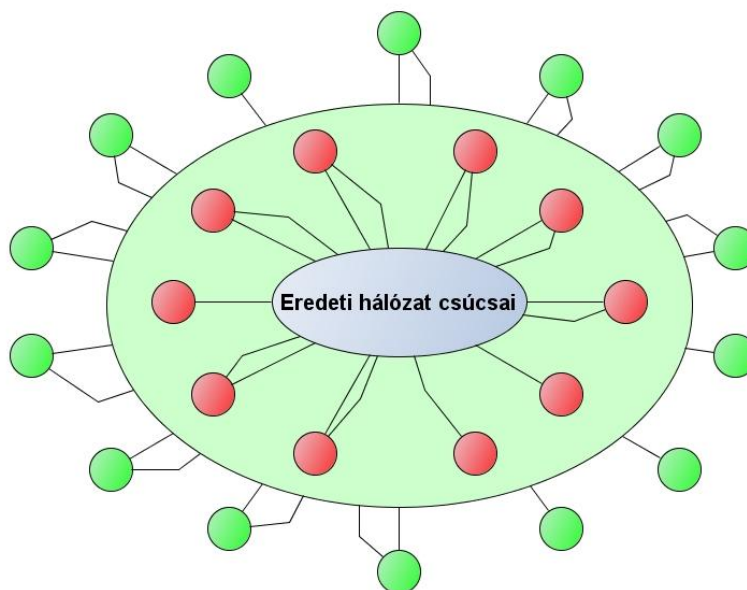


5. ábra – Új energiaszintek elérése

Az algoritmust *Perl* nyelven, illetve a *MathWorks Matlab* szoftverben modelleztük. A programok a Függelékben és a CD lemezen *csakazegyik.pl* és *maxkettot.pl*, illetve *kruskal.m*, *fysaida.m* és *iscircle.m* [15] néven találhatók. Az első alkalmazás inputként a mért hálózat csúcsainak listáját, valamint a BT2 átmeneteket kéri éllistas formátumban, eredményként pedig szintén ebben a formában választja ki és sorolja fel azokat az éleket, melyek a BT2 átmenetek közül két mért csúcs között futnak. Amennyiben a kapott listát tetszőleges módszerrel az élek súlya szerinti növekvő sorrendbe rendezzük, a *maxkettot.pl* program kiválasztja minden új, még nem mért energiaszinthez a legfeljebb két legnagyobb intenzitású átmenetet.

Ezen algoritmus segítségével bővíthetjük a mért energiaszintek halmazát új, mért csúcsokkal szomszédos energiaszintekkel. Ha az eljárást megismétljük a bővített mért hálózatra, akkor a következő szomszédok asszignálására kaphatunk ehhez szükséges átmeneteket. Ilyen módon egy bővülő csúcshalmaz segítségével előbb-utóbb az összes BT2-beli energiaszint asszignálására lehetőségünk nyílna.

Az algoritmus iteratív működését a hálózaton a 6. ábra szemlélteti. A piros csúcsok reprezentálják az i . lépésben beemelt csúcsokat, az $i+1$. lépésben pedig a zölddel jelölt csúcsokkal bővítettük a gráfot.

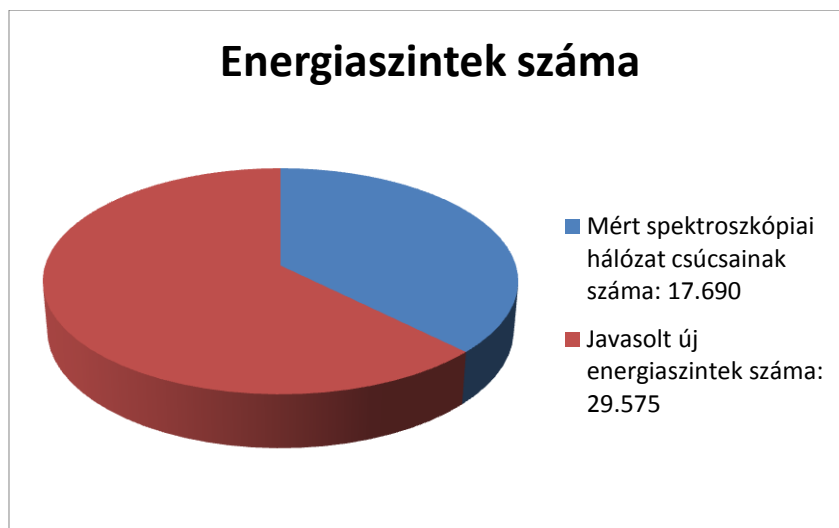


6. ábra – Iteráció szemléltetése, két egymást követő lépés

Megjegyezzük, hogy nagyban javíthatja az algoritmus hatékonyságát, ha két lépése között optimalizáljuk az új hálózatot: az egy vagy két energiaszinten elért új csúcsokhoz hasznos további, jelentős nagyságú átmeneteket keresnünk, például a 4.1 fejezetben megadott eljárás segítségével.

A H_2^{16}O molekula spektroszkópai hálózatára futtatva (tekintettel a 30.000 cm^{-1} -es vágásra az *ab initio* hálózaton) az algoritmus 50.253 új átmenet asszignálására ad javaslatot, melyek felsorolása éllistas alakban az *ujenergiaszintek.txt* fájlban szerepel. Az első oszlopban található átmenetek a már asszignált csúcsokat kódolják, a második oszlopban pedig az új energiaszintek BT2-indexe található. A harmadik oszlop az adott átmenet intenzitásának nagyságrendjét mutatja. Ezzel az eljárással 29.575 új energiaszintet tudtunk a mért hálózat csúcsai közé emelni, ezek felsorolása az *ujelertszomszedok.txt* fájlban található.

A mért spektroszkópai hálózat bővítésére adott javaslat eredményeit a 7. és 8. ábra két kördiagramjával szemléltetjük.



7. ábra – A mért hálózat bővítésére adott javaslat eredményei – energiaszintek száma



8. ábra – A mért hálózat bővítésére adott javaslat eredményei – átmenetek száma

5. Összefoglalás

5.1 Az eredmények értékelése

A dolgozat során a 2007-ben bevezetett spektroszkópiai hálózatok matematikai, gráfelméleti módszerekkel történő elemzésével foglalkoztunk. Munkánk során a H_2^{16}O spektroszkópiai hálózatát, mint tesztrendszert vizsgáltuk. Választásunk azért esett erre a molekulára, mivel korábbi publikációk [1-3] jó kiindulási alapot adnak a kutatás folytatására, valamint a víz spektroszkópiája kiemelten fontos kutatási terület, hiszen eredményeit számos tudományterületen és mérnöki alkalmazásban felhasználják (pl. a földi üvegházhatás magyarázatakor).

Vizsgálatunk során két kérdést fogalmaztunk meg. Először arra kerestük a választ, hogy a H_2^{16}O molekula mért spektroszkópiai hálózata mely új, a mért hálózatban nem szereplő, de az *ab initio* kvantumkémiái számítások segítségével előállított átmenetekkel tehető összefüggővé. Ezután célul új, még nem asszignált, így a mért SN-en kívül eső energiaszintek elérését tűztük ki, melyekkel lehetőség nyílik a jelenlegi mért spektroszkópiai hálózat bővítése.

A dolgozatban összefoglaltuk a vizsgálatához szükséges gráfelméleti alapfogalmakat, és áttekintettük, miként tudjuk matematikai környezetbe ágyazni és a gráfok nyelvén megfogalmazni a felmerülő spektroszkópiai problémákat. Ezt követően elemeztük és példákkal szemléltettük a használt algoritmusok működését. A gráfok bejárására a mélységi keresés módszerét, a minimális súlyú feszítőfa létrehozására pedig a Kruskal-algoritmust találtuk optimálisnak.

A bemutatott algoritmusokat Perl nyelven, valamint a *MathWorks Matlab* szoftver segítségével implementáltuk, ezeket a Függelékben, valamint a mellékelt CD lemezen közzétettük. Az elkészült programok lehetővé teszik, hogy az itt tárgyalt vizsgálatokat a későbbiekben könnyen elvégezhessük más molekulák vagy hálózatok esetén is, emellett könnyen módosíthatóak, ha egy „nagyjából hasonló” kérdésre szeretnénk választ kapni.

A H_2^{16}O molekula vizsgálata során egyrészt eljárást mutattunk arra, miként lehet a mért spektroszkópiai hálózat komponenseit új, még nem mért átmenetek segítségével összefüggővé tenni. Törekedtünk arra, hogy a kifejlesztett eljárás minél könnyebben legyen alkalmazható a gyakorlatban, a gyakorlati mérések során, ezért minél nagyobb intenzitású új átmeneteket

kerestünk. Módszerünk lényege, hogy a mért hálózat csúcsai, valamint a köztük futó, az *ab initio* számítások által meghatározott átmenetek alkotta gráfra egy minimális súlyú feszítőfa algoritmust futtatunk. Az e célt szolgáló algoritmusok közül a spektroszkópai hálózatok jellegzetességeit figyelembe véve a Joseph Kruskal által publikált módszert választottuk. Vizsgálatunkat ezután a H_2^{16}O molekulán folytattuk, és példát mutattunk általunk talált új, két ismert, címkézett energiaszint közti átmenetre.

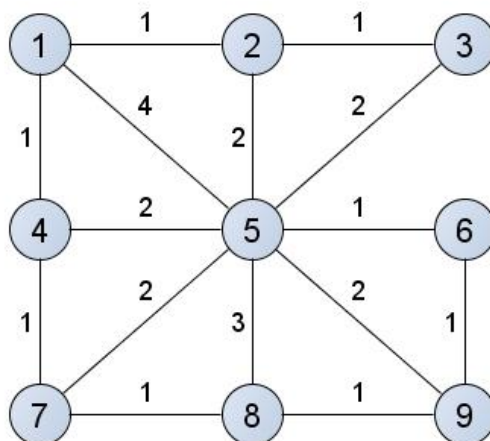
Második feladatként arra kerestük a választ, hogy a még nem asszignált energiaszinteket milyen átmenetek megméréssel emelhetjük be a mért spektroszkópai hálózat csúcsai közé. Eszközeink főleg adatbányászati típusú algoritmusok voltak, hiszen ebben a problémában a legnagyobb nehézséget – hasonlóan az előző feladathoz – az SN-t tartalmazó adatbázisok nagy, többszázezer soros mérete okozza. Megoldásként iteratív algoritmust adtunk meg, melyekre két programot írtunk. Az algoritmus egy bővülő csúcshalmazt hoz létre, és minden lépésben az aktuális mért hálózat csúcsainak szomszédjai egy részének elérésére ad javaslatot, a mért csúcsokból az ilyen szomszédos csúcsokba futó átmenetek formájában. Munkánk eredményeképp 50.253 olyan, még nem mért átmenetet találtunk, melyek asszignálásával 29.575 új energiaszintet emelhetünk a mért hálózatba.

Végezetül megemlítenénk néhány olyan további kérdést a spektroszkópai hálózatok területéről, melyekre az itt tárgyaltakhoz hasonlóan szintén matematikai módszerek alkalmazásával kereshetünk hatékonyan megoldást, legyen az gráfelmélet, adatbányászat, valószínűségszámítás vagy statisztika.

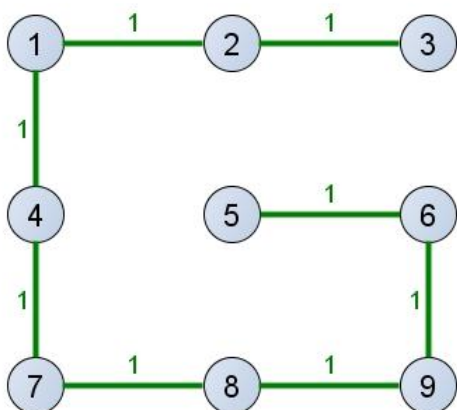
5.2 Kitekintés, kérdésfelvetések

A H_2^{16}O molekula spektroszkópai hálózata a fenti problémakörök mellett számos más kutatási lehetőséget rejt magában. Kitekintésként olyan kérdéseket vetünk fel, melyek megválaszolása meghatározott matematikai módszerek segítségével (is) lehetséges.

A 3.1 fejezetben Kruskal-algoritmus segítségével a hálózat olyan feszítőfáját hoztuk létre, mely a fa éleinek összsúlyát tekintve minimális. Gyakorlati szempontból is érdekes lehet azonban annak vizsgálata, minként néz ki a hálózat azon feszítőfája, melyre a csúcsok közötti átlagos távolság minimális. A probléma megoldására több eljárás is ismert a gráfelméletben, például [16]. A két feszítőfa közti különbséget a 9. ábra szemlélteti:

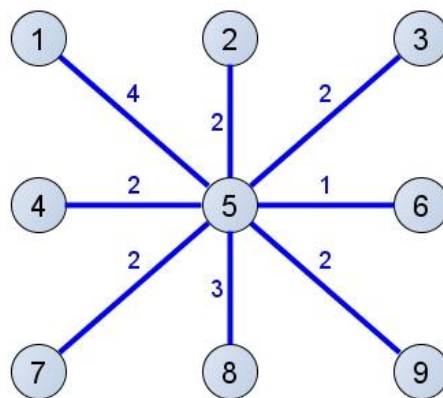


Kiinduló gráfunk, melyen a két feszítőfa különbségét szemléltetjük.



Minimális súlyú feszítőfa.

Sok esetben nem célravezető, ha nagy a csúcspárok közti átlagos távolság a fa élein.



A hálózat útjai közt az átlagos távolság. Két tetszőleges csúcs távolsága a fa élein legfeljebb 2.

9. ábra – Minimális súlyú feszítőfa és minimális átlagos távolságú feszítőfa közti különbség szemléltetése

Valamelyest rokon probléma az az eset, amikor nem az átmenetek intenzitása adja a hálózat élein értelmezett súlyfüggvényt, hanem a két energiaszint közti átmenet mérési bizonytalansága. A mérési bizonytalanság értéke azt reprezentálja, hogy mennyire pontosan adja meg az adott átmenet hullámszámát, és ezáltal az átmenet energiaszintjeinek bizonytalanságát. Hasznos lehet ezért gyakorlati szempontból egy olyan feszítőfa létrehozása, illetve olyan átmenetek megmutatása, melyekre a mérési hiba esélye minél alacsonyabb.

A spektroszkópiai hálózatok gráfelméleti módszerekkel történő modellezése során érdekes eredményekhez vezethet a Szemerédi-féle regularitási lemma különféle alkalmazásainak interpretálása [17]. Az említett tétel az extrémális gráfelmélet egyik legnagyobb hatású eredménye a XX. században, Szemerédi Endre magyar-amerikai

matematikust 2012-ben részben ezért az eredményéért tüntették ki a „matematikai Nobel-díjnak” is nevezett Abel-díjjal.

A gráfelméleti megközelítés mellett a matematika más tudományterületeivel történő ötvözés is érdekes eredményekhez vezethet. Vizsgálhatjuk például a gráf csúcsainak fokszámát, ennek függvényében pedig valószínűségi eloszlásokat illeszthetünk a hálózat energiaszintjeire.

A hálózat átmeneteit ritkíthatjuk sztochasztikus vágások segítségével is. Érdekes kérdés lehet annak vizsgálata, hogy a spektroszkópiai hálózat átmeneteit érintő vágások milyen hatással vannak a gráf szerkezetére. Az ilyen típusú elemzésekhez jó kiindulási alapot nyújthat [1-2].

Végül megemlítyük, hogy szintén érdekes eredményekhez vezethet, ha az itt tárgyalt vagy felvetett kérdéseket más molekulák spektroszkópiai hálózata esetén járjuk körül. Ezen kutatáshoz reményeink szerint hatékony eszközöket sikerült készítenünk.

Függelék

A mellékelt CD lemez tartalma

mappa	fájlnév	kiterjesztés	fejezet
	arendas_peter_tdk2012	pdf	TDK dolgozat
	feszitofaBT2	txt	a 4.1 fejezetben mutatott új átmenetek adatbázisa
	ujenergiaszintek	txt	a 4.2 fejezetben mutatott új átmenetek és energiaszintek adatbázisa
adatbázisok	BT2_1000	txt	BT2 adatbázis az 1000 cm ⁻¹ -es vágás mellett
adatbázisok	BT2_30000	txt	BT2 adatbázis 30000 cm ⁻¹ -ig
adatbázisok	mert_csucslista_szurt_1000	txt	mért csúcsok a BT2_1000 adatbázishoz
adatbázisok	transitions_vertex	txt	a mért spektroszkópiai hálózat adatbázisa
adatbázisok	transitions_vertex_1000	txt	a mért spektroszkópiai hálózat adatbázisa 1000 cm ⁻¹ -ig
Matlab	fysaida	m	Kruskal-algoritmus implementáció részfeladata
Matlab	iscycle	m	Kruskal-algoritmus implementáció részfeladata
Matlab	kruskal	m	Kruskal-algoritmus implementációja
Perl	cimkezd	pl	éllista átalakítása a Matlab számára is megfelelő inputhoz
Perl	csakazegyikvege	pl	egy bemeneti éllistára azokat az éleket választja ki, melyek csak az egyik végpontja egy bemeneti csúcslista valamely csúcsa
Perl	csakkettot	pl	egy bemeneti éllista sorai közül töröl úgy, hogy a csúcspárok közül a 2. koordinátájú legfeljebb kétszer szerepeljen az adatbázisban
Perl	mindketvege	pl	egy bemeneti éllistára azokat az éleket választja ki, melyek mindkét végpontja egy bemeneti csúcslista valamely csúcsa
Perl	visszacimkezd	pl	éllista visszaalakítása BT2 indexű formátumra a Matlab outputjáról (mely a cimkezd.pl után kapott input alapján ad eredményt)

Perl programfájlok

mindketvege.pl

```
@ARGV < 3 and die "USAGE: perl.exe BT2conv.pl [mert csucsok] [BT2 fajlja] [outfile]";
open(IN1,$ARGV[0]) or die "Unable to open infile...";
open(IN2,$ARGV[1]) or die "Unable to open infile...";
open(OUT,">$ARGV[2]") or die "Unable to open outfile...";

@mertcsucsok;
@elek;
$i = 0;
$j = 0;
$k = 0;

for( <IN1> ){
    @fields=split(/\s+/);
    $mertcsucsok[$i]=$fields[0];
    $i=$i+1;
    print "Csucsszam: $i\n";
}

for( <IN2> ){
    @fields=split(/\s+/);
    $hossz=substr($fields[1], -2);
    $node1=$fields[2];
    $node2=$fields[3];

    $j=0;
    $k=0;

    while($j < $#mertcsucsok) {
        if ($node1 eq $mertcsucsok[$j] ) {
            while($k < $#mertcsucsok) {
                if ($node2 eq $mertcsucsok[$k]) {
                    push(@elek, "$node1 $node2 $hossz");
                    $k=$#mertcsucsok;
                }
                $k=$k+1;
            }
            $j=$#mertcsucsok;
        }
        $j=$j+1;
    }

    print "Elszam: $#elek\n";
}

for $h ( 0 .. $#elek ){
```

```
    print OUT "$elek[$h]\n";  
}
```

```
close IN1;  
close IN2;  
close OUT;
```

csakazegyikvege.pl

```
@ARGV < 3 and die "USAGE: perl.exe BT2conv.pl [mert csucsok] [BT2 fajlja] [outfile]";  
open(IN1,$ARGV[0]) or die "Unable to open infile...";  
open(IN2,$ARGV[1]) or die "Unable to open infile...";  
open(OUT,">$ARGV[2]") or die "Unable to open outfile...";
```

```
@mertcsucsok;  
@elek;  
$i = 0;  
$j = 0;  
$k = 0;  
$m = 0;
```

```
for( <IN1> ){  
    @fields=split(/\s+/);  
    $mertcsucsok[$i]=$fields[0];  
    $i=$i+1;  
    print "Csucsszam: $i\n";  
}
```

```
for( <IN2> ){  
    @fields=split(/\s+/);  
    $hossz=substr($fields[1], -2);  
    $node1=$fields[2];  
    $node2=$fields[3];  
  
    $j=0;  
    $k=0;  
    $m=0;  
  
    while($j < $#mertcsucsok+1) {  
        $m=0;  
        if ($node1 eq $mertcsucsok[$j] ) {  
            while($k < $#mertcsucsok+1) {  
                if ($node2 eq $mertcsucsok[$k]) {  
                    $m=1;  
                }  
                $k=$k+1;  
            }  
            if ($m == 0) {
```

```
                push(@elek, "$node1 $node2 $hossz");
            }
        }
        $j=$j+1;
    }

    print "Elszam: $#elek\n";

}

for $h ( 0 .. $#elek ){
    print OUT "$elek[$h]\n";
}

close IN1;
close IN2;
close OUT;
```

cimkezd.pl

@ARGV < 2 and die "USAGE: perl.exe cimkezd.pl [infile] [outfile]";

open(IN,\$ARGV[0]) or die "Unable to open infile...";

open(OUT,">\$ARGV[1]") or die "Unable to open outfile...";

\$nodecount=1;

@edges;

\$i=0;

for(<IN>){

\$i=\$i+1;

print "\$i\n";

@fields = split(/\s+/);

\$node1 = (\$fields[0]);

\$node2 = (\$fields[1]);

\$hossz = \$fields[2];

if (!exists \$nodes{\$node1}) {

\$nodes{\$node1} = \$nodecount;

\$nodecount=\$nodecount+1;

}

if (!exists \$nodes{\$node2}){

\$nodes{\$node2} = \$nodecount;

\$nodecount=\$nodecount+1;

}

push(@edges, "\$nodes{\$node1}\t\$nodes{\$node2}\t\$hossz");

}

visszacimkezd.pl

```
@ARGV < 3 and die "USAGE: perl.exe visszacimkezd.pl [cimkefajl] [feszitofa] [outfile]";
open(IN1,$ARGV[0]) or die "Unable to open infile...";
open(IN2,$ARGV[1]) or die "Unable to open infile...";
open(OUT,">$ARGV[2]") or die "Unable to open infile...";

$i = 0;
$j=0;
%cimkek;
@feszitofa;

for( <IN1> ){
    $i=$i+1;
    print "$i\n";

    @fields=split(/\s+/);
    $eredeti=$fields[0];
    $matlabos=$fields[1];
    $cimkek{$matlabos}=$eredeti;
}

for( <IN2> ){
    $j=$j+1;
    print "$j\n";

    @fields=split(/\s+/);
    $node1=$fields[0];
    $node2=$fields[1];

    push(@feszitofa, "$cimkek{$node1}\t$cimkek{$node2}");
}

for $h ( 0 .. $#feszitofa ){
    print OUT "$feszitofa[$h]\n";
}

close IN1;
close IN2;
close OUT;
```

csakkettot.pl

```
@ARGV < 2 and die "USAGE: perl.exe BT2conv.pl [infile] [outfile]";
open(IN,$ARGV[0]) or die "Unable to open infile...";
open(OUT,">$ARGV[1]") or die "Unable to open infile...";
```

```
%mibolmennyi;
$i=0;

for( <IN> ){
    @fields=split(/\s+/);
    $node1=$fields[0];
    $node2=$fields[1];
    $hossz=$fields[2];

    $mibolmennyi{$node2}=$mibolmennyi{$node2}+1;

    if ($mibolmennyi{$node2} < 3) {
        push (@elek, "$node1 $node2 $hossz");
    }
    $i=$i+1;
    print "$i\n";
}

for $h ( 0 .. $#elek ){
    print OUT "$elek[$h]\n";
}

close IN;
close OUT;
```

Matlab programfájlok

kruskal.m

```
function [w,PV] = kruskal(PV)
row = size(PV,1);
X = [];
for i = 1 : row
    X(PV(i,1),PV(i,2)) = 1;
    X(PV(i,2),PV(i,1)) = 1;
end
n = size(X,1);
PV = fysalida(PV,3);
korif = zeros(1,n);
for i = 1 : row
    akmi = PV(i,[1 2]);
    [korif,c] = iscycle(korif,akmi);
    if c == 1
        PV(i,:) = [0 0 0];
    end
end
w = sum(PV(:,3)');
PV( ~any(PV,2), : ) = [];
```

```
PV=PV(:,[1,2]);
```

fysalida.m

```
function A = fysalida(A,col)
[r c] = size(A);
if col < 1 | col > c | fix(col) ~= col
    uiwait(msgbox([' error input value second argumment takes only integer values
between 1 & ' num2str(c)],'ERROR','error'));
    error;
end
for i = 1 : r - 1
    d = r + 1 - i;
    for j = 1 : d - 1
        if A(j,col) > A(j + 1,col)
            % row swap j <--> j + 1
            A([j j + 1],:) = A([j + 1 j],:);
        end
    end
end
end
```

iscircle.m

```
function [korif,c]=iscircle(korif,akmi)
g=max(korif)+1;
c=0;
n=length(korif);
if korif(akmi(1))==0 & korif(akmi(2))==0
    korif(akmi(1))=g;
    korif(akmi(2))=g;
elseif korif(akmi(1))==0
    korif(akmi(1))=korif(akmi(2));
elseif korif(akmi(2))==0
    korif(akmi(2))=korif(akmi(1));
elseif korif(akmi(1))==korif(akmi(2))
    c=1;
    return
else
    m=max(korif(akmi(1)),korif(akmi(2)));
    for i=1:n
        if korif(i)==m
            korif(i)=min(korif(akmi(1)),korif(akmi(2)));
        end
    end
end
end
```

Irodalomjegyzék

- [1] A. G. Császár and T. Furtenbacher, Spectroscopic Networks, *J. Mol. Spectrosc.* **2011**, 266, 99-103.
- [2] T. Furtenbacher and A. G. Császár, The role of intensities in determining characteristics of spectroscopic networks, *J. Mol. Struct. (Boris Galabov Special Issue)* **2012**, 1009, 123-129.
- [3] T. Furtenbacher and A. G. Császár, MARVEL: Measured Active Rotational-Vibrational Energy Levels. II. Algorithmic Improvements, *J. Quant. Spectr. Rad. Transfer (Flaud, Camy-Peyret, Barbe Special Issue)* **2012**, 113, 929-935.
- [4] A. G. Császár and T. Furtenbacher, From a Network of Computed Reaction Enthalpies to Atom-Based Thermochemistry (NEAT), *Chem. Eur. J.*, **2010**, 16(16), 4826-4835.
- [5] T. Furtenbacher, A. G. Császár, and J. Tennyson, MARVEL: Measured Active Rotational-Vibrational Energy Levels, *J. Mol. Spectry.* **2007**, 245, 115-125.
- [6] A. G. Császár, G. Czakó, T. Furtenbacher, and E. Mátyus, An Active Database Approach to Complete Spectra of Small Molecules, *Ann. Rep. Comp. Chem.* **2007**, 3, 155-176.
- [7] T. Furtenbacher, Adatbázis alapú molekulaspektroszkópia, *doktori értekezés, ELTE TTK Kémiai Doktori Iskola*, **2008**
- [8] L. Li, D. Alderson, R. Tanaka, J. C. Doyle, W. Willinger, Towards a Theory of Scale-Free Graphs: Definition, Properties, and Implications, **2005**
- [9] P. Erdős, A. Rényi, On random graphs, I., *Publ. Math. Debrecen* 6, **1959**, 290-297.
- [10] A.-L. Barabási, E. Ravasz, T. Vicsek, Deterministic scale-free networks, *Physica A*, **2001**, 299., 559–564.
- [11] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, Second Edition, *MIT Press and McGraw-Hill*, **2001**, Section 22.3: Depth-first search, 540–549.
- [12] D. Knuth, The Art of Computer Programming, Volume 3: Sorting and Searching, Third Edition. *Addison-Wesley*, **1997**, 106–110.
- [13] J. B. Kruskal, On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem, *Proceedings of the American Mathematical Society*, Vol 7, No. 1, **1956**, 48–50.
- [14] R. C. Prim, Shortest connection networks and some generalizations, *Bell System Technical Journal*, 36, **1957**, pp. 1389–1401.
- [15] N. Cheilakos, a Kruskal's algorithm implementation code in Matlab, **2006**, <http://www.mathworks.com/matlabcentral/fileexchange/13457-kruskal-algorithm>

- [16]E. Dahlhaus, P. Dankelmann, W. Goddard, H. C. Swart, MAD Trees and Distance-Hereditary Graphs, *Discrete Applied Mathematics*, Volume 131 Issue 1, **2003**, 151-167.
- [17]J. Komlós, A. Shokoufandeh, M. Simonovits and E. Szemerédi, The Regularity Lemma and its applications in Graph Theory, *Combinatorics, Paul Erdős is eighty, Vol. 2*, , *Bolyai Soc. Math. Stud.*, 2, *János Bolyai Math. Soc.*, Budapest, **1996**, 295-352.
- [18]R. J. Barber, J. Tennyson, G. J. Harris, and R. N. Tolchenov, A high-accuracy computed water line list, *Mon. Not. R. Astron. Soc.* 368, **2006**, 1087-1094.