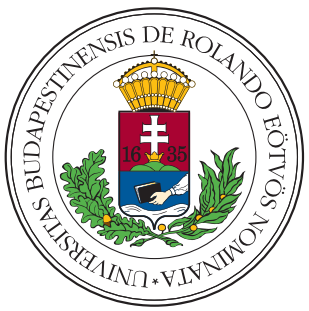
Közelíthetőségi problémák

Szakdolgozat

Készítette: Takács Kristóf
Matematika BSc, alkalmazott matematikus szakirány

Témavezető:
Dr. Grolmusz Vince, egyetemi tanár
Számítógéptudományi Tanszék
Eötvös Loránd Tudományegyetem Természettudományi Kar



Eötvös Loránd Tudományegyetem
Természettudományi Kar

2015

Tartalomjegyzék

Köszönetnyilvánítás

Ezúton is szeretném megköszönni témavezetőmnek, Dr. Grolmusz Vince tanár úrnak a számos konzultációt, valamint a sok segítséget és útmutatást, amelyek nélkül ez a szakdolgozat biztosan nem jöhetett volna létre. Külön szeretnék köszönetet mondani azért, hogy felhívta figyelmemet a többszörös szekvenciaillesztési probléma szakdolgozatom témájához kapcsolódó vonatkozásaira, s ezáltal el tudtam készíteni diplomamunkám egy fontos fejezetét. Köszönettel tartozom családomnak is, akik a szakdolgozat megírásának ideje alatt végig biztattak és biztosítottak támogatásukról, amely sokat segített a dolgozat elkészítése során.

1. fejezet

Bevezetés

Számos, a gyakorlati alkalmazások szempontjából is alapvető jelentőségűnek nevezhető problémáról közismert, hogy **NP**-teljes: például gondolhatunk egy gráf kromatikus számának meghatározására, az utazó ügynök problémára, a hátizsák-feladatra stb. Ugyanakkor éppen ezen problémák gyakorlati fontossága miatt szükséges, hogy optimalizálási feladat esetén legalább valamilyen becslést, közelítő eredményt lehessen alkalmazni, még annak tudatában is, hogy a pontos eredmény meghatározása általános esetben a jelenlegi technológiai korlátok mellett szinte lehetetlen, és ez várhatóan rövid és középtávon is így fog maradni.

Az utazó ügynök probléma (travelling salesman problem, TSP) természetesen merül fel minden olyan feladat esetén, amelyben egy hálózat pontjai között kell legrövidebb bejárási útvonalat keresni: gondolhatunk például egy telefonhálózat létesítésére vagy egy település csatornarendszerének kialakítására. Ide sorolható továbbá egy chipgyártásban felmerülő probléma is: az eszköz létrehozása során egy lézersugárnak végig kell járnia az integrált áramkör bizonyos pontjait úgy, hogy ez a lehető legrövidebb ideig tartson. Ezenkívül számos alkalmazás található a közlekedési hálózatok, a járműirányítás területén vagy éppen a genetikában is. A széles körű felhasználási igény következtében nagy számú közelítő algoritmus született erre a feladatra, amelyek a gyakorlati feladatok esetén általában az optimálishoz kimondottan közeli approximációt adnak (pl. Lin és Kernighan δ-path módszere). [19]

A csúcslefedési probléma gyakorlati alkalmazási köre szűkebbnek nevezhető a TSP-énél, ugyanakkor ennek a feladatnak a jelentőségét sem szabad alábecsülni. Ebben a vonatkozásban előtérbe kerülnek a biológiai alkalmazások: például fejlődéstörténeti fák egyes fehérjék genetikai információira épülő meghatározásánál fontos elemként jelenik

meg egy minimális elemszámú lefedő csúcshalmaz megtalálása. Megemlíthető továbbá az egyedi nukleotid polimorfizmus (single nucleotid polymorphism) illesztési feladat is, amelynek az egyik megoldási algoritmusa szintén magában foglal egy csúcslefedési problémát. [24]

Nem szabad megfeledkezni arról, hogy az **NP**-teljes problémák meglehetősen eltérő jellemzőkkel rendelkeznek azon a közös tulajdonságon kívül, hogy mindegyikük a legnehezebb matematikai feladatok közé tartozik. Ezért nem nevezhető meglepőnek, hogy ha közelíthetőségi szempontból vizsgáljuk ezeket a kérdéseket, akkor is kimondottan változatos eredmények születnek: egyes problémák approximációs viselkedése rendkívül jó, azaz akár az optimum tetszőlegesen jó közelítése elérhető, míg más esetekben egyáltalán nem vagy csak bizonyos speciális esetekben érhető el approximáció.

Fontos kiemelni, hogy az alkalmazott matematikában az utóbbi néhány évtizedben szintén fontos szereppel rendelkező véletlen (randomizált) algoritmusok nem képezik tárgyát szakdolgozatomnak. Természetesen bizonyos esetekben ezek az eljárások is használhatók approximációra, ugyanakkor ezekről sem jelenthető ki, hogy minden problémával kapcsolatban, általános érvényűen alkalmazhatóak lennének. A $\mathbf{P} = \mathbf{NP}$ kérdéshez hasonlóan nagy jelentőségű probléma, hogy vannak-e olyan feladatok, amelyeket randomizált módon nagy valószínűséggel meg tudunk oldani (ezeknek osztályát **RP**-vel jelölik), viszont hagyományos algoritmusokkal polinomiális időben nem oldhatók meg. A problémakörrel foglalkozó matematikusok többsége szerint nincsenek ilyen problémák, azaz feltételezésük szerint ha valamit sikerül a véletlen felhasználásával (nagy valószínűséggel) megoldanunk, akkor a véletlen eljárások alkalmazása nélkül is meg lehet oldani. Érdemes továbbá megjegyezni, hogy egy **NP**-teljes probléma csak akkor lehet **RP**-ben, ha $\mathbf{NP} = \mathbf{RP}$, így **NP**-teljes problémák közelítése valószínűleg egyáltalán nem lehetséges véletlen algoritmusokkal. [15]

Gyakran előfordul, hogy egy adott (akár **NP**-teljes) problémára sikerül olyan algoritmust találni, amely bizonyos speciális inputokon gyorsan lefut és pontos megoldást ad, sőt az sem nevezhető ritkának, hogy az eljárás a feladat legtöbb (esetleg tulajdonképpen az összes) gyakorlatban előforduló esetét meg tudja oldani. Ugyanakkor ezen algoritmusok esetében mindig konstruálható a problémának olyan példánya, amelyre a talált megoldási módszer nem képes polinomidőben lefutni. (Talán a legalapvetőbb példa ilyen eljárásra a szimplex módszer az operációkutatás területéről.) Bár felfedezhető bizonyos hasonlóság ezen algoritmusok és a jelen dolgozatban vizsgált közelítő eljárások között, fontosnak tartom egyértelműsíteni, hogy szakdolgozatomban olyan

approximációs algoritmusokat vizsgálok, amelyek egy probléma *minden* inputja esetén adott mértékű közelítést biztosítanak. Ebből következően az előzőekben említett algoritmusok vizsgálatára sem kerül sor diplomamunkám keretében.

Érdemes megemlíteni, hogy még az **NP**-teljes problémák között is vannak olyanok, amelyek statisztikailag könnyűnek nevezhetők: ez alatt azt értem, hogy előfordulhat, hogy bizonyos feltételek esetén egy adott közelítő algoritmus által szolgáltatott eredmény várható értékére meglehetősen pontos becslést lehet meghatározni. Például tekintsünk egy olyan n csúcsú véletlen G gráfot, amelynek minden élét egymástól függetlenül p valószínűséggel húzzuk be. Ekkor ismert, hogy ha $p < \frac{(1-\varepsilon)\log n}{n}$, akkor G majdnem biztosan nem összefüggő, azaz annak a valószínűsége, hogy G mégis összefüggő, 0-hoz tart, ha $n \to \infty$. [11] Így ebben az esetben kijelenthető, hogy G-ben majdnem biztosan nincs Hamilton-kör (hiszen G nagy valószínűséggel elég nagy n-re nem is összefüggő); ugyanakkor figyelembe véve, hogy egy így definiált véletlen gráfnak várhatóan $\binom{n}{2}p$ éle van, ha p-t az előbbi felső korlátnál kisebbnek, ugyanakkor $\frac{2}{n-1}$-nél nagyobbnak választjuk, akkor G-nek várhatóan legalább n éle lesz. Ebből következően bár a paraméterek előbbieknek megfelelő választása mellett majdnem biztosan nem lesz G-ben Hamilton-kör, mégis lesz olyan véletlen gráf, amely ezen paraméterek mellett is fog tartalmazni Hamilton-kört. Egy olyan Hamilton-kör kereső eljárás, amelynek csak várható értékben kell megbízható eredményt nyújtania, az ilyen gráfokra egyszerűen mint „rossz" inputokra tekinthet, és megteheti, hogy nem foglalkozik velük (hiszen várható értékben ilyen p-re egy véletlen gráfban valóban csak 0 valószínűséggel lesz Hamilton-kör). Mindazonáltal a szakdolgozatomban tárgyalt algoritmusokkal szemben az az alapvető elvárás, hogy az adott probléma minden példányára képesek legyenek meghatározott mértékű approximációt biztosítani, vagyis nem fordulhatnak elő az előbbihez hasonló rossz inputok, amelyekre az algoritmus nem az elvárásoknak megfelelő eredményt ad.

Szakdolgozatomban **NP**-teljes optimalizálási problémák közelíthetőségét vizsgálom meg: először néhány konkrét példa közelítő algoritmusain keresztül, majd áttekintem a fontosabb közelíthetetlenségi eredményeket, végül pedig az ezen a területen elért, viszonylag frissnek tekinthető eredményeket, továbbá az ide kapcsolódó főbb, aktuálisan megoldatlan problémákat is ismertetem. Diplomamunkám természetesen tartalmazza azon felhasznált fogalmak definícióját is, amelyek nem tekinthetők általánosan ismertnek. A dolgozatban kiemelt hangsúlyt kap a bioinformatikai problémák közül a többszörös szekvenciaillesztés közelíthetősége is.

2. fejezet

Közelítő algoritmusok

1. Definíció. *Legyen H egy optimalizálási probléma, amelynek minden y példányához tartozik egy $F(y)$ megoldáshalmaz. Minden $u \in F(y)$ lehetséges megoldás rendelkezik egy pozitív $c(u)$ költséggel, az optimális költség (minimalizálási feladat esetén): $opt(u) = \min\{c(s) : s \in F(u)\}$ (maximalizálási probléma esetén $opt(u) = \max\{c(s) : s \in F(u)\}$). Ha az M egy olyan algoritmus, amely minden y példány esetén egy $M(y) \in F(y)$ lehetséges megoldást eredményez, akkor M-et ε-közelítő algoritmusnak nevezzük ($\varepsilon \geq 0$), ha $\forall y$-ra*

$$\frac{|c(M(y)) - opt(y)|}{\max\{opt(y), c(M(y))\}} \leq \varepsilon. \quad [22]$$

A definíció jelentése szavakkal megfogalmazva: egy algoritmus ε-közelítő, ha bármely inputra az eredmény relatív hibája az optimálishoz viszonyítva legfeljebb ε. A definícióban szereplő tört nevezőjének választását az motiválja, hogy így a definíció maximalizálási és minimalizálási problémák esetén is érvényes. Előbbi esetben egy ε-közelítő algoritmus által adott eredmény mindig nagyobb vagy egyenlő az optimális megoldás $(1 - \varepsilon)$-szeresénél, utóbbi esetben pedig az optimumnak legfeljebb $(\frac{1}{1-\varepsilon})$-szerese.

Gyakorlati szempontból elsősorban a polinom idejű közelítő algoritmusok tekinthetők használhatónak, emiatt is került a következő definíció, amely egyes problémák közelíthetőségét egy paraméterrel jellemzi, kizárólag ilyen eljárások figyelembevételével meghatározásra.

2. Definíció. *Egy H optimalizálási probléma közelítési küszöbén azon pozitív ε-ok infimumát értjük, amelyekre H-nak létezik polinom idejű ε-közelítő algoritmusa.* [22]

Mivel a közelítő algoritmus definíciójában szereplő arány 0 és 1 közé eshet az adott probléma bármely példánya esetén, így egy optimalizálási probléma közelítési küszöbe szintén ezen intervallumba eső értékeket vehet fel. A legkedvezőbbnek az tekinthető, ha a közelítési küszöb 0, hiszen ekkor a probléma tulajdonképpen tetszőlegesen jól közelíthető, míg ha ez az érték 1-gyel egyenlő, akkor a feladatra lényegében nem létezik használható közelítő algoritmus.

2.1. Utazó ügynök probléma

Feladat: Legyen G egy teljes gráf, amelynek minden e éle egy nemnegatív egész $c(e)$ súllyal rendelkezik. Keressük azt a Hamilton-kört G-ben, amelynek súlya az adott súlyozás mellett minimális.

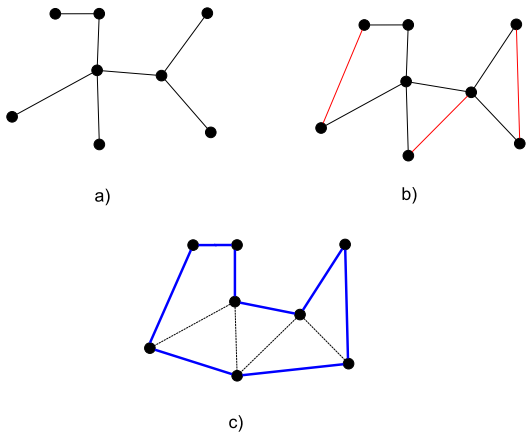
A TSP problémáról ismert, hogy **NP**-teljes, ám a következő tétel szerint közelíthetőségi szempontból is a legnehezebb algoritmuselméleti feladatok közé tartozik.

1. Tétel. *Ha* $\mathbf{P} \neq \mathbf{NP}$*, akkor az utazó ügynök probléma közelítési küszöbe 1.* [22]

Bizonyítás: Indirekt tegyük fel, hogy létezik a TSP-re olyan ε-közelítő algoritmus, ahol $\varepsilon < 1$. Legyen adott a $G = (V, E)$ gráf, és készítsük el hozzá azt a teljes gráfot, amelynek $|V|$ db csúcsa van, az e_{ij} él súlya pedig legyen 1, ha i és j között volt él G-ben, és $\frac{|V|}{1-\varepsilon}$ különben. Futtassuk le erre a gráfra a feltételezett ε-közelítő algoritmust.

Két eset lehetséges: ha egy $|V|$ összköltségű bejárást kaptunk vissza, akkor ez csak 1 súlyú, G-ben is szereplő éleket használ, azaz G-ben van Hamilton-kör. A másik esetben szerepel a bejárásban legalább egy $\frac{|V|}{1-\varepsilon}$ súlyú él, így a kör összköltsége biztosan nagyobb, mint $\frac{|V|}{1-\varepsilon}$. Az algoritmus ε-közelítő tulajdonsága miatt az optimális bejárás súlya legalább a visszadott bejárás súlyának $(1-\varepsilon)$-szorosa, azaz nagyobb, mint $|V|$, így G-ben nincs Hamilton-kör. Következésképpen az ε-közelítő algoritmus segítségével tetszőleges G gráfról polinomidőben eldönthető, hogy van-e benne Hamilton-kör, amely probléma **NP**-teljes, azaz $\mathbf{P} = \mathbf{NP}$ következik. □

Ez alapján azt lehetne gondolni, hogy a TSP-vel nem is érdemes közelíthetőségi szempontból foglalkozni, azonban pozitív eredménynek nevezhető, hogy bizonyos, az



2.1. ábra. Christofides algoritmusa: a) Tegyük fel, hogy T egy minimális súlyú feszítő fája G-nek. b) Tegyük fel, hogy M T páratlan fokú csúcsainak minimális súlyú párosítása (pirossal jelölve). c) $T + M$ Euler-bejárásának módosításával kapott közelítés a TSP-re (kékkel jelölve).

alkalmazások szempontjából gyakran logikusan teljesülő feltételek esetén léteznek 1-nél kisebb közelítési küszöbű algoritmusok. Például ha minden élsúly 1 vagy 2, akkor (amellett, hogy minden algoritmus $\frac{1}{2}$-közelítő) $\frac{1}{7}$-közelítő algoritmus is létezik [23], a következő tétel pedig a feladat egy másik speciális esetére vonatkozóan biztosít approximációt.

2. Tétel (Christofides algoritmusa). *Ha a c súlyfüggvény metrika, azaz teljesíti a háromszög-egyenlőtlenséget ($c(e_{ij}) \leq c(e_{ik}) + c(e_{kj})$ $\forall$ $1 \leq i < j \leq |V| = n, i \neq k, j \neq k$ esetén), akkor létezik a TSP-re $\frac{1}{3}$-közelítő algoritmus.* [7]

Bizonyítás: Legyen T egy minimális súlyú feszítő fa a G teljes gráfban, amelyre meg szeretnénk oldani a TSP-feladatot. Jelöljük Z-vel T páratlan fokú pontjait, ekkor $|Z|$ páros lesz. Vegyük Z-nek egy minimális költségű M párosítását, így a T és az M élei által alkotott $T + M$ gráf Euler-gráf lesz, mivel minden pontjának foka páros. Vegyük egy W Euler-bejárását $T + M$-nek, amelyből hagyjuk el azokat az éleket, amelyek azt eredményeznék, hogy egy-egy csúcson többször haladunk át (ezzel a bejárás összköltségén a háromszög-egyenlőtlenség miatt csak csökkenthetünk).

Legyen H az optimális Hamilton-kör G-ben. Ekkor egyrészt $c(T) \leq c(H)$, hiszen minden Hamilton-kör tartalmaz feszítő fát, másrészt pedig $c(M) \leq \frac{c(H)}{2}$ is teljesül: H meghatároz egy H' kört Z pontjain is, amelyre $c(H') \leq c(H)$. H' felbomlik két

diszjunkt párosítás (M_1 és M_2) uniójára, amelyekre M optimalitása miatt $c(M) \leq c(M_1)$ és $c(M) \leq c(M_2)$, így $2c(M) \leq c(H')$ teljesül. Összevetve tehát a $T+M$ gráf egy Euler-bejárásának költsége legfeljebb $\frac{3}{2}$-szerese az optimális Hamilton-kör költségének, azaz az algoritmus $\frac{1}{3}$-közelítő. □

2.2. Maximális vágás

Feladat: Adott a $G = (V, E)$ gráf. Particionáljuk a V csúcshalmazt két részre (S és T) úgy, hogy S és T között a lehető legtöbb él menjen.

Algoritmus: Vegyünk egy tetszőleges részhalmazt V-ben. Az algoritmus egy lépésében vizsgáljuk meg, hogy a vágás nagysága növelhető-e azáltal, hogy egy T-beli csúcsot áthelyezünk S-be, vagy egy S-beli csúcsot T-be. Ha van ilyen csúcs, akkor hajtsuk végre az áthelyezést és iteráljunk, ha nincs, akkor az algoritmus véget ér.

3. Tétel. *A fenti algoritmus $\frac{1}{2}$-közelítő.* [20]

Bizonyítás: Az algoritmus véges, sőt polinomiális, mivel minden lépésben legalább eggyel nő a vágás nagysága, így az eljárás legfeljebb $|E|$ iteráció után véget ér. Az $\frac{1}{2}$-közelítés bizonyításához tekintsük V diszjunkt felbontását $V = V_1 \cup V_2 \cup V_3 \cup V_4$-re úgy, hogy az optimális vágást $V_1 \cup V_2$ és $V_3 \cup V_4$ alkotja, míg az algoritmus által adott vágást $V_1 \cup V_3$ és $V_2 \cup V_4$. Jelölje e_{nm} $(1 \leq n \leq m \leq 4)$ a V_n és V_m közötti élek számát. A heurisztikus vágásról tudjuk, hogy nem lehet a nagyságát azzal növelni, hogy áthelyezünk egy csúcsot az egyik partícióból a másikba. Ez azt jelenti, hogy minden V_1-beli csúcsra igaz, hogy a belőle V_1-be és V_3-ba menő élek száma legfeljebb annyi, mint a V_2-be és a V_4-be menő élek száma. Ezt a gondolatmenetet V_1 összes csúcsára alkalmazva a $2e_{11} + e_{13} \leq e_{12} + e_{14}$ egyenlőtlenséget kapjuk, ebből pedig $e_{13} \leq e_{12} + e_{14}$ is teljesül. A másik három halmazt vizsgálva hasonló módon rendre a következő összefüggések teljesülnek:

$$e_{13} \leq e_{23} + e_{34}$$

$$e_{24} \leq e_{12} + e_{23}$$

$$e_{24} \leq e_{14} + e_{34}$$

.

Vegyük hozzá még a fenti egyenlőtlenségekhez a triviális módon teljesülő $e_{14}+e_{23} \leq e_{14}+e_{23}+e_{12}+e_{34}$ összefüggést. Ezeket összevetve látható, hogy az $e_{13}+e_{24}+e_{14}+e_{23} \leq 2(e_{12}+e_{34}+e_{14}+e_{23})$ egyenlőtlenség teljesül, azaz az algoritmus által adott vágás nagysága legalább a fele az optimálisnak, vagyis az algoritmus $\frac{1}{2}$-közelítő. □

2.3. Maximális kielégíthetőség

Feladat (k–MAX–$GSAT$): Adott n változós Boole-formulák egy $\Omega = \{\omega_1, \ldots, \omega_m\}$ halmaza, amelyek teljesen általánosak, kizárólag annyi feltételnek kell teljesülnie rájuk, hogy az n db változó közül pontosan k db-ot tartalmaznak. Keressük a változóknak azt a kiértékelését, amely Ω formulái közül a lehető legtöbbet elégíti ki.

Legyen s_i azon kiértékelések száma a 2^k db közül, amelyek ω_i-t kielégítik, így az n db változó egy véletlen kiértékelése ω_i-t $P(\omega_i) = \frac{s_i}{2^k}$ valószínűséggel elégíti ki. A kielégített formulák várható száma így egy véletlen kiértékelésnél

$$P(\Omega) = \sum_{i=1}^{m} P(\omega_i).$$

Jelölje $\Omega[x_1 = igaz]$ azt a formulahalmazt, amely az $x_2, \ldots, x_n$ változókat tartalmazza, és amelyet úgy kapunk, hogy Ω mindegyik formulájában az x_1 értékét igazra állítjuk be. Ekkor a $P(\Omega[x_1 = igaz])$ érték ismét kiszámolható, és $\Omega[x_1 = hamis]$ hasonló definícója mellett teljesül, hogy $P(\Omega) = \frac{1}{2}(P(\Omega[x_1 = igaz]) + P(\Omega[x_1 = hamis]))$. Ez azt jelenti, hogy x_1 értékét lehet úgy választani, hogy a keletkező formulahalmaz várható értéke legalább akkora lesz, mint az eredetié.

Az algoritmus tehát minden lépésben meghatározza, hogy az adott változó *igaz* vagy *hamis* értékre történő beállítása esetén nem csökken-e az új formulahalmaz várható értéke, és ennek megfelelően teszi a változót *igaz*zá vagy *hamis*sá. Az algoritmus az eljárás végén a változóknak egy olyan kiértékelését adja meg, amelyben minden változó értéke be van állítva *igaz*ra vagy *hamis*ra, ugyanakkor mivel a várható érték nem csökkent, legalább $P(\Omega)$ formula ki van elégítve. Legyen N azon formulák száma, amelyek egyenként kielégíthetők, azaz $P(\omega_i) > 0$ teljesül. Ekkor $P(\Omega) = \sum_{i=1}^{N} P(\omega_i)$, ahol $\forall i$: $P(\omega_i) > 0$. A kielégíthető formulák maximális száma ($opt(\Omega)$) így legfeljebb N, azaz

$$1 - \frac{P(\Omega)}{opt(\Omega)} \leq 1 - \frac{P(\Omega)}{N} \leq 1 - \min\{P(\omega_i) : P(\omega_i) > 0\}.$$

Ebből adódóan a fenti algoritmus ε-közelítő, ahol ε értéke egy mínusz az Ω kielégíthető formulái közötti legkisebb kielégítési valószínűség. Ugyanakkor érvényes a $\min\{P(\omega_i) : P(\omega_i) > 0\} \geq \frac{1}{2^k}$ alsó becslés, hiszen ha ω_i kielégíthető, akkor változóinak 2^k db kielégítése közül legalább egy megfelelő lesz.

Így megfogalmazható a következő állítás:

4. Tétel. *A fenti algoritmus a $k-MAX-GSAT$ problémára ε-közelítést ad, ahol ε értéke $1 - \frac{1}{2^k}$.* [22]

Speciális esetben ez az érték viszonylag jó közelítést jelent: például a $MAX–SAT$ probléma esetén ha azon literálok diszjunkcióit tekintjük egy-egy formulának, amelyek két konjunkció között állnak, akkor ezen formulák kielégítési valószínűsége legalább $\frac{1}{2}$, így az algoritmus $\frac{1}{2}$-közelítő, vagyis legalább a formulák felét kielégíti. Ha pedig még az is teljesül, hogy minden formula legalább k db különböző literált tartalmaz, akkor a legkisebb kielégítési valószínűség $1 - \frac{1}{2^k}$-val becsülhető alulról, azaz az algoritmus $\frac{1}{2^k}$-közelítő lesz.

2.4. Csúcslefedés

Feladat: Adott egy $G = (V, E)$ gráf. Keressük a csúcsoknak azt a minimális elemszámú $F \subseteq V$ részhalmazát, amelyre teljesül, hogy minden E-beli élnek legalább az egyik végpontját tartalmazza.

Algoritmus: Induljunk ki az $F = \emptyset$ üres halmazból. Válasszunk egy élt E-ből, adjuk hozzá F-hez a végpontjait, és töröljük ki ezeket a csúcsokat (és a belőlük kiinduló éleket) G-ből. Ha van még él G-ben, akkor iteráljunk, ha nincs, akkor az algoritmus leáll.

5. Tétel. *A fenti algoritmus $\frac{1}{2}$-közelítő, azaz a minimálisnál legfeljebb kétszer nagyobb elemszámú lefogó csúcshalmazt ad vissza eredményként.* [14]

Bizonyítás: Az algoritmus futása során összesen $\frac{|F|}{2}$ db független élt választ ki, és ezek végpontjai alkotják F-et. Már ezeknek a független éleknek a lefogásához is szükség

van $\frac{|F|}{2}$ db csúcsra, így az optimális lefogó csúcshalmaz mérete is alulról becsülhető $\frac{|F|}{2}$-vel. Az algoritmus $|F|$ db csúcsot választott ki, azaz az optimális halmaz méretének legfeljebb kétszeresét, így az algoritmus $\frac{1}{2}$-közelítő. $\square$

Fontos megjegyezni, hogy a fentihez hasonlóan egyszerű és talán még természetesebben adódó algoritmus, amely minden lépésben egy maximális fokszámú csúcsot ad hozzá F-hez, majd a csúcs kitörlése után iteráció következik, semmilyen $\varepsilon > 0$ esetén nem lesz ε-közelítő algoritmus.

Mivel már az előző feladat is **NP**-teljes, így természetesen a következő, súlyokat is alkalmazó változata is az:

Feladat: Adott egy $G = (V, E)$ gráf egy a csúcsokon értelmezett, nemnegatív c költségfüggvénnyel. Keressük az F minimális súlyú lefogó csúcshalmazt a c költségfüggvény mellett.

A közelítő algoritmus felírásához érdemes a problémát IP-feladatként megfogalmazni: legyen A G él-pont incidenciamátrixa. Ekkor a feladat a következő alakban írható:

$$Ax \geq 1$$

$$x \in (0,1)^{|V|}$$

$$\min cx$$

x i-edik koordinátája 0, ha az i-edik csúcsot nem választottuk be F-be, és 1 különben. Az első sorban szereplő feltétel fejezi ki, hogy G minden éle le van fogva.

A duális LP-relaxáltja:

$$\forall e \in E\colon \sum_{e \in d(v)} y(e) \leq c(e)$$

$$y \geq 0$$

$$\max \sum_{e \in E} y(e),$$

ahol $d(v)$ a v csúcsból kiinduló élek halmazát jelöli. Jelöljük $\pi(v)$-vel a $\sum_{e \in d(v)} y(e)$ összeget.

Algoritmus: Kezdetben legyen $y \equiv 0$ és $F = \emptyset$. Az algoritmus általános lépésében válasszunk egy még le nem fogott $e = (u, v)$ élt, és módosítsuk $y(e)$ értékét a következőképpen: $y(e) := \min\{c(u) - \pi(u), c(v) - \pi(v)\}$. Ezután vizsgáljuk meg, hogy teljesül-e a $c(u) = \pi(u)$ és a $c(v) = \pi(v)$ egyenlőségek közül valamelyik vagy akár mindkettő. Ha igen, az adott ponto(ka)t adjuk hozzá F-hez és iteráljunk addig, amíg G-nek van olyan éle, amely még nincs lefogva. [14]

6. Tétel. *A fenti algoritmus $\frac{1}{2}$-közelítő.*

Bizonyítás:

$$\sum_{v \in F} c(v) = \sum_{v \in F} \pi(v) \leq \sum_{v \in V} \pi(v) = 2 \sum_{e \in E} y(e) \leq 2 \max \sum_{e \in E} y(e) =$$

$$= 2 \min c\widetilde{x} \leq 2 \min cx,$$

ahol $\widetilde{x}$ az LP-relaxált primál feladat optimumát jelöli, az utolsó egyenlőtlenség pedig az LP- és IP-feladatok optimumai között fennálló egyenlőtlenség miatt teljesül. Ebből következik, hogy a talált lefogó csúcshalmaz súlya a minimális súlyúnak legfeljebb a kétszerese, így az algoritmus valóban $\frac{1}{2}$-közelítő. □

3. fejezet

Közelíthetetlenségi eredmények

Az **NP**-teljes problémák általános vizsgálatában fontos szerepet tölt be a polinomiális visszavezetés fogalma, hiszen segítségével számos problémáról be lehet látni, hogy ebbe a bonyolultsági osztályba tartoznak. Ugyanakkor egy feladat polinomiális visszavezetése egy másikra általában nem őrzi meg az eredeti probléma közelíthetőségi tulajdonságait: előfordulhat, hogy az addig lényegében nem közelíthető probléma jól közelíthetővé válik, de ennek az ellenkezője is bekövetkezhet. Ezért szükséges egy a polinomiális visszavezetésnél erősebb fogalom bevezetése, amely megőrzi a közelíthetőséget. (A fejezet során alapvetően Christos H. Papadimitriou *Számítási bonyolultság* [22] című műve vonatkozó fejezetének áttekintését követem.)

3. Definíció. *Legyenek A és B optimalizálási problémák. A-nak B-re való L-visszavezetése egy F és G logaritmikus tárral kiszámítható függvényekből álló pár, amelyre teljesülnek a következők:*

i) Ha y az A egy $opt(y)$ optimális költségű példánya, akkor $F(y)$ a B egy olyan példánya, amelyre igaz, hogy $opt(F(y)) \leq \alpha opt(y)$, ahol $\alpha > 0$ konstans.

ii) Ha s az $F(y)$ egy tetszőleges megoldása, akkor $G(s)$ az y-nak egy olyan megoldása, amelyre $|opt(y) - c(G(s))| \leq \beta |opt(F(y)) - c(s)|$, ahol $\beta > 0$ konstans.

A definícióban tehát F A példányait képezi B példányaira, G pedig B és A megoldásai között hat. A *ii*) feltétel szavakkal megfogalmazva azt jelenti, hogy minden s-re G biztosan y egy lehetséges megoldását adja vissza eredményül, amelynek az optimumtól való távolsága felülről becsülhető $F(y)$ és s távolságának konstansszorosával.

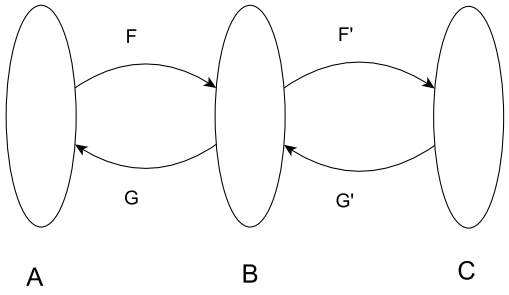
Példa: Tekintsük a maximális független csúcshalmaz és a minimális lefogó csúcshalmaz problémákat azokon a gráfokon, amelyek maximális fokszáma Δ (Δ-t $|V|$-től független konstansnak tekintjük). Legyen F az identitásfüggvény, G pedig helyettesítse a K lefogó halmazt $V-K$-val. Ekkor F és G L-visszavezetést alkot $\alpha=\Delta+1$ és $\beta=1$ konstansokkal, mivel ha Δ a maximális fokszám, akkor a maximális független csúcshalmaz mérete alulról becsülhető $\frac{|V|}{\Delta+1}$-gyel, a minimális lefogó ponthalmaz mérete pedig legfeljebb $|V|$, így $\alpha=\Delta+1$ valóban teljesíti i)-t. Továbbá Gallai tétele miatt egy tetszőleges K lefogó ponthalmaz és a minimális lefogás mérete közötti különbség megegyezik a $|V-K|$ és a maximális független csúcshalmaz közötti különbséggel, ezért $\beta=1$ megfelelő választás lesz. Belátható, hogy (F,G) ugyanezen α és β értékek mellett a másik irányban is L-visszavezetést alkot. (Tetszőleges gráfokra ($|V|\to\infty$ esetén) F és G nem L-visszavezetés, mivel ha V-nek egyetlen valódi részhalmaza sem lefogó (pl. teljes gráfok esetén), akkor a minimális lefogó csúcshalmaz mérete tetszőlegesen nagy lehet az optimális független csúcshalmaz méretéhez viszonyítva, azaz nem létezik olyan α, amely kielégítené i)-t.)

7. Tétel. *Ha (F,G) az A probléma B-re való L-visszavezetése, (F',G') pedig a B probléma C-re való L-visszavezetése, akkor az $(F'\circ F, G\circ G')$ visszavezetés A-nak C-re való L-visszavezetése (kompozíciós tulajdonság).*

Bizonyítás: Felhasználva, hogy $F'\circ F$ és $G\circ G'$ kiszámítható logaritmikus tárral, tekintsük az A probléma egy y példányát. Ekkor teljesül, hogy $opt(F(y))\leq\alpha_1 opt(y)$, illetve $opt(F'(F(y)))\leq\alpha_2 opt(F(y))$. Így $opt(F'(F(y)))\leq\alpha_1\alpha_2 opt(y)$, azaz az $F'\circ F$ függvényre az $\alpha_1\alpha_2$ konstans mellett teljesül az i) tulajdonság.

A ii) tulajdonság bizonyításához legyen s' $F'\circ F$ tetszőleges megoldása, és jelöljük s-sel B azon megoldását, amelyre $G'(s')=s$ teljesül. Ekkor a definícióból adódóan egyrészt $|opt(y)-c(G(s))|\leq\beta_1|opt(F(y))-c(s)|$, ugyanakkor mivel G' a B és a C közötti L-visszavezetés része, $|opt(F(y))-c(G'(s'))|\leq\beta_2|opt(F'(F(y)))-c(s')|$ is teljesül. Mindkét egyenlőtlenség bal oldalán felhasználva a $G'(s')=s$ összefüggést, látható, hogy a $G\circ G'$ függvény a $\beta_1\beta_2$ konstanssal kielégíti ii)-t, így $(F'\circ F, G\circ G')$ valóban A L-visszavezetése C-re. □

Az L-visszavezetés fogalmának bevezetése mögötti elsődleges motiváció a közelíthetőségi tulajdonság megőrzése volt, a következő állítás ezt a jellemzőt fogalmazza meg:



3.1. ábra. Az A és B, illetve a B és C közötti visszavezetések, amelyek kompozíciója A és C közötti visszavezetést alkot. F és F' az egyes problémák példányhalmazai között, G és G' pedig a megfelelő megoldáshalmazok között hat.

8. Tétel. *Ha (F, G) egy L-visszavezetés A-ról B-re α és β konstansokkal B-re létezik polinom idejű ε-közelítő algoritmus, akkor A-nak van $\frac{\alpha\beta\varepsilon}{1-\varepsilon}$-közelítő algoritmusa ($\varepsilon < 1$).*

Bizonyítás: Legyen y A adott példánya. Az algoritmus ehhez elkészíti B $F(y)$ példányát, majd a B ε-közelítő algoritmusát használva keletkezik $F(y)$-ra egy s közelítő megoldás. Ezt követően tekintjük A $G(s)$ megoldását, amely egy $\frac{\alpha\beta\varepsilon}{1-\varepsilon}$-közelítést ad.

Az $\frac{|opt(y)-c(G(s))|}{\max\{opt(y),c(G(s))\}}$ arányt vizsgálva megállapítható, hogy az L-visszavezetések ii) tulajdonsága miatt a számláló felülről becsülhető $\beta|opt(F(y)) - c(s)|$-sel. Az i) tulajdonság miatt viszont a nevező alulról becsülhető $\frac{opt(F(y))}{\alpha}$-val, ami tovább becsülhető $\frac{\max\{opt(F(y)),c(s)\}}{\alpha}(1-\varepsilon)$-nal. Így $\frac{|opt(y)-c(G(s))|}{\max\{opt(y),c(G(s))\}} \leq \frac{\alpha\beta}{1-\varepsilon}\frac{|opt(F(y))-c(s)|}{\max\{opt(F(y)),c(s)\}}$. A második tört B ε-közelítő algoritmusa miatt legfeljebb ε, így az egyenlőtlenség jobb oldala felülről becsülhető $\frac{\alpha\beta\varepsilon}{1-\varepsilon}$-val, azaz az algoritmus $\frac{\alpha\beta\varepsilon}{1-\varepsilon}$-közelítő. □

A következő definíció a fejezet hátralévő részében kiemelt fontosságú lesz:

4. Definíció. *Az A optimalizálási problémához egy polinom idejű közelítő séma egy olyan algoritmus, amely minden $\varepsilon > 0$ esetén A minden y példányára egy legfeljebb ε relatív hibájú megoldást ad vissza, és az algoritmus futási ideje $|y|$-nak egy ε-tól függő polinomjával korlátos. (Ha a futásidő $\frac{1}{\varepsilon}$-tól is polinomiálisan függ, akkor a közelítő sémát teljesen polinomiálisnak nevezzük.)*

Ha az előző tétel bizonyításában ε-nal 0-hoz tartunk jobbról, akkor $\frac{\alpha\beta\varepsilon}{1-\varepsilon}$ is 0-hoz tart, ebből következően: ha $\exists$ (F, G) L-visszavezetés A-ról B-re és B-re létezik polinom idejű közelítési séma, akkor A-ra is létezik.

A közelítő algoritmusok elméleti hátterének felépítéséhez szükség van az **NP** osztály megfelelőjére, azaz egy olyan problémahalmazra, amelyben számos fontos teljes probléma megtalálható. Ehhez előbb be kell vezetni egy másik, önmagában is érdekes problémaosztályt:

5. Definíció. *Az* **SNP** *(szigorú* **NP***) osztályba azon tulajdonságok tartoznak, amelyek kifejezhetők* $\exists S\ \forall\ x_1\ \forall\ x_2\ \ldots\ \forall\ x_n\ \phi(S, G, x_1, \ldots, x_n)$ *alakban, ahol* ϕ *olyan kvantormentes elsőrendű formula, amelyben az* x_i*-k változókat, a* G *egy struktúrát, a* S *pedig egy relációt jelöl a változók között.* [22]

Példa: A k-SAT probléma (azaz a konjunktív normálformákon értelmezett SAT feladatnak azon változata, amelyben minden klóz legfeljebb k db literált tartalmazhat) **SNP**-be tartozik. Jelölje ugyanis az előző definícióban x_i (i = 1, ..., k) a normálformában szereplő változókat, G a logikai műveletek halmazát, S pedig a változók egy kiértékelését. Így ϕ jelölheti az adott konjunktív normálformát, a definíció formulája pedig ebben az esetben azt fejezi ki, hogy létezik a változóknak egy olyan S kiértékelése, hogy minden klózban szerepel legalább egy igaz literál.

A fenti, eldöntési problémákat tartalmazó osztály viszonylag természetes módon optimalizálási problémák halmazává alakítható, ha nem azt követeljük meg, hogy a változók minden $(x_1, \ldots, x_n)$ alakú, n elemű sorozata elégítse ki ϕ-t S mellett, hanem csupán egy olyan S-t keresünk, hogy ϕ a lehető legnagyobb számú $(x_1, \ldots, x_n)$ n-esre teljesüljön. Ezt a célkitűzést figyelembe véve, illetve az **SNP** osztály definícióját általánosítva alakítható ki a következő definíció:

6. Definíció. *Az A probléma a* $\mathbf{MAXSNP_0}$ *osztályba tartozik, ha a következő alakban írható:*

$$\max_S |\{(x_1, \ldots, x_n) \in V^n : \phi(G_1, \ldots, G_m, S, x_1, \ldots, x_n)\}|,$$

ahol A inputja a V *véges alaphalmazon értelmezett* $G_1, \ldots, G_m$ *relációk halmaza.*

Tehát egy $\mathbf{MAXSNP_0}$-beli problémánál egy olyan S relációt keresünk, amely maximalizálja a ϕ-t teljesítő $(x_1, \ldots, x_n)$ n-eseket.

7. Definíció. *Egy optimalizálási probléma a* **MAXSNP** *osztályba tartozik, ha L-visszavezethető egy* $\mathbf{MAXSNP_0}$*-beli problémára.*

Példák: *i*) A maximális vágás probléma $\mathbf{MAXSNP_0}$-ban van, ugyanis a következő formában írható fel:

$$\max_{S \subset V} |\{(u,v) : ((G(u,v) \vee G(v,u)) \wedge S(u) \wedge \neg S(v)\}|$$

$G(u,v)$ azt jelöli, hogy G tartalmazza az (u,v) élt, $S(u)$ pedig azt jelenti, hogy az u csúcs eleme S-nek. Így a formula a csúcsoknak egy olyan S részhalmazát keresi, amely maximalizálja azon élek számát, amely benne vannak G-ben és pontosan az egyik végpontjuk van S-ben.

ii) A legfeljebb k-fokú gráfokon értelmezett független csúcshalmaz probléma szintén $\mathbf{MAXSNP_0}$-beli, mint ahogy azt a következő formalizálás mutatja:

$$\max_{S \subset V} |\{(u, v_1, \ldots, v_k) : ((u, v_1, \ldots, v_k) \in Z) \wedge (u \in S) \wedge (v_1 \notin S) \wedge \ldots \wedge (v_k \notin S)\}|$$

Itt Z azon V feletti $(u, v_1, \ldots, v_k)$ $(k+1)$-esek halmaza, amelyekben a v_i csúcsok u szomszédai (esetleg ismétlésekkel, ha u fokszáma kisebb, mint k). Azaz a formulát átfogalmazva: egy olyan S részhalmazát keressük a csúcsoknak, amelyre azon $u \in S$-ek száma maximális, hogy u szomszédai nem elemei S-nek.

iii) A legfeljebb k-fokú gráfokon értelmezett lefogó csúcshalmaz probléma $\mathbf{MAXSNP}$-ben van, mivel L-visszavezethető az előző problémára.

9. Tétel. *Ha A egy*

$$\max_{S} |\{(x_1, \ldots, x_n) : \phi\}|$$

alakú $\mathbf{MAXSNP_0}$*-beli probléma, akkor A-nak létezik $(1 - \frac{1}{2^{n_\phi}})$-közelítő algoritmusa, ahol n_ϕ a ϕ formula S-et tartalmazó atomi formuláinak száma.* [22]

Bizonyítás: Legyen y A egy példánya, V pedig az A-hoz tartozó véges univerzum. Helyettesítsünk be az összes lehetséges módon egy $a = (a_1, \ldots, a_n) \in V^n$ rendezett n-est az x_i változók helyébe, a keletkező formulákat jelölje ϕ_a. A ϕ_a formulákban előforduló atomi formulák három csoportba sorolhatók: azok, amelyekben az S reláció szerepel; valamely G_i bemeneti relációt tartalmazó atomi formulák; illetve az a_i változók közötti egyenlőséget kifejező formulák. Az utóbbi két típusba tartozó atomi formulák a G_i input relációk és az aktuális változók figyelembevételével rögtön kiértékelhetők, logikai értékük pedig behelyettesíthető ϕ_a-ba. Így ϕ_a $S(a_{j_1}, \ldots, a_{j_l})$ alakú atomi formulák kompozíciójára redukálható.

y ezek szerint az összes lehetséges n-eshez tartozó ϕ_a alakú formulák halmazából áll, a különböző (logikai változóként kezelhető) $S(a_{j_1}, \ldots, a_{j_l})$ atomi formulákat pedig úgy kell kiértékelnünk, hogy a kielégített ϕ_a formulák száma a lehető legmagasabb legyen. Ez a probléma viszont pontosan a $k - MAX - GSAT$ feladat egy példánya, és a 2.3 pontban szereplő bizonyítás alapján konstruálható $(1 - \frac{1}{2^{n_\phi}})$-közelítő algoritmus. □

A **MAXSNP** osztály előző tételben szereplő tulajdonsága az **NP** azon jellemzőjével állítható párhuzamba, hogy minden **NP**-beli probléma esetén létezik olyan nemdeterminisztikus algoritmus, amely polinom időben megoldást ad. Természetesen az **NP** esetében is determinisztikus polinomiális algoritmus találása a fő cél, a **MAXSNP**-t vizsgálva pedig egy polinom idejű közelítési séma tekinthető hasonlóan fontos eredménynek.

Alapvető kérdés, hogy minden **MAXSNP**-beli problémának van-e polinomiális közelítési sémája (ez a kérdés tekinthető **P** = **NP** állítás közelítő algoritmusokra vonatkozó analógiájának). Ennek a kérdéskörnek a vizsgálatához szükséges a teljesség fogalmának definiálása a **MAXSNP** osztályban:

8. Definíció. *Egy* **MAXSNP***-beli probléma* **MAXSNP***-teljes, ha* **MAXSNP** *minden eleme L-visszavezethető rá.*

A 4. Definíció utáni megjegyzésből következik, hogy ha egy **MAXSNP**-teljes problémára létezik polinom idejű közelítési séma, akkor minden **MAXSNP**-beli problémára is létezik ilyen.

10. Tétel. *A MAX-3SAT probléma (azaz a SAT azon speciális esete, amelyben minden klózban legfeljebb három különböző literál szerepel)* **MAXSNP***-teljes.*

Ezen tételt, valamint a fejezet hátralévő részében szereplő eredményeket igazolásuk összetettségére való tekintettel bizonyítás nélkül közlöm, ugyanakkor az előző tétel segítségével bizonyíthatók a következő állítások:

11. Tétel. *A következő problémák* **MAXSNP***-teljesek:*

i) 4-fokú független csúcshalmaz (a maximális független csúcshalmaz feladat azon gráfokra megszorítva, amelyek maximális fokszáma legfeljebb 4)

ii) 4-fokú lefogó csúcshalmaz (a minimális lefogó csúcshalmaz feladat azon gráfokra megszorítva, amelyek maximális fokszáma legfeljebb 4)

iii) maximális vágás.

A közelíthetetlenségi eredmények közül mindenképpen az egyik legfontosabb a következő:

12. Tétel. *Egy A* **MAXSNP***-teljes problémára pontosan akkor létezik polinom idejű közelítési séma, ha* $\mathbf{P} = \mathbf{NP}$.

Az előző tétel következménye, hogy a 11. Tételben felsorolt problémák egyikére, továbbá a csúcslefedési feladatra sem létezik polinom idejű közelítési séma, illetve a maximális független csúcshalmaz és a maximális klikk problémák közelítési küszöbe 1, ha $\mathbf{P} \neq \mathbf{NP}$.

4. fejezet

A közelítő algoritmusok egy fontos gyakorlati alkalmazása: többszörös szekvenciaillesztés

Az elmúlt évtizedekben a bioinformatika egyik legfontosabb problémájává vált a multiple sequence alignment (többszörös szekvenciaillesztés, MSA) kérdésköre. A probléma biológiai–genetikai szempontú megközelítése a következő: adott néhány fehérje-, DNS- vagy RNS-szekvencia, ezek halmazát jelölje S. A feladat egy olyan S' szekvenciahalmaz keresése, amely a kiindulási szekvenciákhoz a legközelebb esik abban az értelemben, hogy az összes szekvenciahalmaz közül, amely bizonyos feltételek betartása mellett kialakítható S-ből, S'-nek a legkisebb a létrehozási költsége.

A probléma gyakorlati súlyát az adja, hogy segítségével kimondottan nagy pontossággal meghatározhatóak egy-egy protein- vagy nukleinsav-család konzervatív régiói, azaz a szekvenciák azon elemei, pozíciói, amelyek az adott család alapvető jellemzőit, funkcióit kialakítják. Az utóbbi néhány évtizedben egyre nagyobb hangsúlyt szerző genetikai kutatásokat tekintve nem nevezhető meglepőnek, hogy az MSA probléma a bioinformatikával foglalkozó szakemberek érdeklődésének középpontjában áll. Fontos negatív eredmény volt az MSA probléma **NP**-teljességének bizonyítása $|S| \geq 3$ esetén, különösen figyelembe véve azt, hogy az $|S| = 2$ esetben Needleman és Wunsch polinomiális, $O(n^2)$ idejű, dinamikus programozáson alapuló módszert talált a feladat megoldására. [21] Ugyanakkor az MSA kiemelt jelentőségére tekintettel számos, az általános esetre vonatkozó közelítő algoritmus került kidolgozásra, amelyek a gyakorlatban

előforduló esetekben többnyire meglehetősen jól használható eredményt szolgáltatnak.

Szakdolgozatom ezen fejezetének témája a multiple sequence alignment probléma bemutatása, kiemelt tekintettel az ezzel kapcsolatban kidolgozott approximációs algoritmusokra. A dolgozat a témakörhöz kapcsolódó definíciók, elnevezések meghatározása után tartalmazza az MSA feladat **NP**-teljességének bizonyítását, illetve néhány hasonlóan jelentős eredményt a problémakörhöz kötődően. Ezt követően részletezem a többszörös szekvenciaillesztési probléma közelítésére az egyik legszélesebb körben használt algoritmus, a Clustal működési elvét.

4.1. Definíciók

Egy véges Σ ábécé feletti stringet *szekvenciá*nak nevezünk. Az s'_1 és s'_2 stringek az s_1 és s_2 szekvenciák *illesztés*ét alkotják (alignment), ha s'_i úgy kapható meg s_i-ből, hogy s_i-be, illetve s_i elejére vagy végére ún. üres jeleket (space vagy gap, jelölése: $-$) szúrunk be, és az így keletkező s'_1 és s'_2 azonos hosszúságú. Következésképpen s'_1 minden karaktere egyértelműen megfeleltethető s'_2 egy karakterének. (Feltehető, hogy $-$ eredetileg nem eleme Σ-nak.) Két azonos pozícióban lévő megegyező karakter *match*-et alkot, míg két eltérő karakter egy *mismatch*-et, amely az adott pozícióban bekövetkező cseréként is értelmezhető. Ha az egyik szekvencia egy $x \in \Sigma$ karakterének a másikban egy gap felel meg, akkor ez tekinthető x törlésének a második szekvenciából, vagy x beszúrásának az első szekvenciába.

Jelöljük s'_1 és s'_2 hosszát l-lel. Egy illesztés *költség*e a következő módon definiálható: $\sum_{i=1}^{l} d(s'_1(i), s'_2(i))$, ahol $s'_1(i)$ és $s'_2(i)$ az alignment i-edik pozíciójában szereplő két karaktert, $d(s'_1(i), s'_2(i))$ pedig két azonos pozíciójú karakter (átalakítási) költségét jelöli egy adott s *költségfüggvény* mellett. (Számos széles körben használt költségfüggvény került kidolgozásra aminosavakra és nukleotidokra vonatkozóan.) Egy s költségfüggvénnyel szemben általánosan elfogadott elvárás, hogy elégítse ki a háromszög-egyenlőtlenséget, azaz $\forall x, y, z \in \Sigma \cup \{-\} : s(x, y) \leq s(x, z) + s(z, y)$ teljesüljön. Két szekvencia *optimális illesztés*ének azt az illesztést nevezzük, amely az összes lehetséges alignmentet tekintve minimalizálja az illesztési költséget. Két szekvencia *szerkesztési távolság*a a két string minimális illesztési költségeként definiálható.

Az alignment fogalma természetes módon általánosítható kettőnél több szekvenciára. Egy *A többszörös illesztés* (multiple alignment) a következőképpen értelmezhető $k \geq 2$ szekvencia esetén: minden szekvenciába gapeket illesztünk úgy, hogy a kelet-

kező stringek azonos l hosszúságúak legyenek, és a szekvenciákat egy kxl-es mátrixban helyezzük el. A szekvenciák minden mezőjére kiterjedően definiálva van egy költségfüggvény, és A költségén az egyes mezők költségeinek összegét értjük. Az egyik leggyakrabban használt költségfüggvény az SP (sum of pairs) költség: ezt alkalmazva egy mező értéke a szekvenciák adott pozícióban álló karaktereiből alkotott párok páronkénti költségeinek összegeként áll elő. Így A költsége nem más, mint az A által meghatározott $\frac{k(k-1)}{2}$ db páronkénti illesztés költségének összege. [29]

Példa: Legyen az illesztendő szekvenciák halmaza $S = \{GCATA, CTAG, GATAC\}$, az SP költségfüggvényt leíró táblázat pedig legyen a következő:

S	G	C	A	T	$-$
G	0	2	1	1	1
C	2	0	2	1	1
A	1	2	0	1	1
T	1	1	1	0	1
$-$	1	1	1	1	0

(Ellenőrizhető, hogy a költségfüggvény teljesíti a háromszög-egyenlőtlenséget.)

Ekkor egy S-re vonatkozó többszörös szekvenciaillesztés lehet az $A = \{GCATA-$, $-C-TAG, G-ATAC\}$ halmaz, amelynek költsége (az egyes mezőkre lebontott költségeket összeadva): $2+2+2+0+0+4 = 10$.

4.2. A többszörös szekvenciaillesztés NP-teljessége

A következő alfejezetben az MSA probléma **NP**-teljességére vonatkozó bizonyítást ismertetem. (Érdekességként érdemes megjegyezni, hogy a téma szakirodalmának áttekintése során olyan, ennek a tételnek egy speciális esetére vonatkozó bizonyítással is találkoztam, amely nagy valószínűséggel hibás volt.)

Legyenek A és B stringhalmazok ugyanazon Σ ábécé felett. Jelölje a továbbiakban:

$$d(A,B) = \sum_{a \in A} \sum_{b \in B} d(a,b)$$

A bizonyítás során szükség lesz a következő fontos lemmára:

1. Lemma. *Legyen U az illesztendő S szekvenciahalmaznak egy olyan részhalmaza, amely kizárólag ugyanannak az u stringnek néhány példányát tartalmazza. Ekkor S egy optimális illesztésében $d(U, U) = 0$.* [4]

Bizonyítás: Indirekt tegyük fel, hogy $\mathcal{A}$ egy optimális illesztése S-nek és ebben U költsége $\alpha > 0$. Legyen $|U| = k$; az előbbi feltevésből következően az $\{u_1, \ldots, u_k\}$ halmaz legalább kételemű, ahol u_i jelöli u i-edik U-beli példányának $\mathcal{A}$-ban szereplő illesztett változatát. Legyen $\hat{u}$ ezek közül az, amelyik minimalizálja a $\{d(u_i, S \setminus U) : i = 1, \ldots, k\}$ értéket. Definiáljuk a következő $\mathcal{A}'$ illesztést: U minden elemét illesszük úgy, ahogy $\mathcal{A}$-ban $\hat{u}$ volt illesztve, $S \setminus U$-n pedig $\mathcal{A}'$ egyezzen meg $\mathcal{A}$-val. Ekkor $\mathcal{A}'$-ben $d(U, U) = 0$, azaz szigorúan kisebb, mint $\mathcal{A}$-ban, $d(U, S \setminus U)$ költsége $\hat{u}$ választása miatt biztosan nem nőtt, $d(S \setminus U, S \setminus U)$ költsége pedig nem változott. Így $\mathcal{A}'$ költsége kisebb $\mathcal{A}$ költségénél, azaz ellentmondásra jutottunk. □

Az MSA probléma formálisan a következőképpen fogalmazható meg:

Input: S stringhalmaz Σ felett, $d : \Sigma^2 \to \mathbf{R}_0^+$ költségfüggvény, $K \geq 0$ valós szám.
Kérdés: Létezik-e S-nek legfeljebb K költségű többszörös szekvenciaillesztése SP-költségszámítás és d költségfüggvény mellett?

13. Tétel. *A többszörös szekvenciaillesztés probléma* **NP***-teljes.*

Bizonyítás: A tétel igazolása során Isaac Elias kételemű ábécére ($\Sigma = \{0, 1\}$) és az úgynevezett egységmetrikára ($\forall i \in \Sigma \cup \{-\} : d(i, i) = 0$ és $d(i, j) = 1$ különben) vonatkozó bizonyítását követem. (A bizonyításban szereplő gondolatmenet néhány teljesen technikai jellegű módosításával az **NP**-teljesség minden olyan Σ-n értelmezett költségfüggvényre igazolható, amely rendelkezik egy metrika tulajdonságaival.) [10]

Az MSA feladat nyilván **NP**-beli (megfelelő tanú lehet egy legfeljebb K költségű illesztés), így az **NP**-teljesség bizonyításához már csak egy **NP**-teljes problémát kell polinomiálisan visszavezetni MSA-ra. Az ismertetésre kerülő bizonyításban ez a probléma a maximális független csúcshalmaz feladat 3-reguláris gráfokra történő megszorítása ($IND3$) lesz, amely ebben a speciális esetben is **NP**-teljes marad. [3]

$IND3$ inputjában tehát adott egy $G = (V, E)$ 3-reguláris gráf, valamint egy $c > 0$ egész szám, a megválaszolandó kérdés pedig az, hogy van-e G-ben legalább c elemű

független csúcshalmaz. A bizonyítás során az adott G-hez és c-hez megkonstruálunk egy S Σ feletti stringhalmazt és egy K pozitív számot, amelyekre az fog teljesülni, hogy pontosan akkor található G-ben c darab független csúcs, ha S-nek létezik K költségű illesztése. Jelölje G csúcsait $\{v_1, \ldots, v_n\}$, a v_i és a v_j csúcs közötti élt pedig e_{ij} $(1 \leq i < j \leq n)$, továbbá a szokásos módon legyen $|V| = n, |E| = m$, valamint $b = 6nm^2$.

S-et három partíciója, T, P és C leírásával adjuk meg, azaz $S = T \cup P \cup C$, ahol T, P, illetve C a következőkben kerülnek meghatározásra. T ugyanannak a t stringnek b darab példányát tartalmazza, amelynek felépítése a következő: $t = (10^b)^{n-1}1$, így $|t| = n + b(n-1)$. A bizonyítás során a $(b+1)(i-1)+1$-edik $(i = 1, \ldots, n)$ helyen álló 1-es fogja v_i szerepét játszani, ezért ezt a mezőt az i-edik csúcsmezőnek is lehet nevezni. P szintén egy p string b darab példányát tartalmazza, ahol $p = 1^c$. Ezek a stringek fogják meghatározni, hogy melyik c darab csúcsot szeretnénk beválasztani a független csúcshalmazba.

S továbbá tartalmazza még C-t is, amelyben m darab string található. Ezek mindegyike G egy-egy élének felel meg, felépítésük pedig a következő: ha c_{ij} az e_{ij} élt reprezentálja S-ben, akkor

$$c_{ij} = 0^{4n}(00^b)^{i-1}10^{b-4n}(00^b)^{j-i-1}10^b(00^b)^{n-j-1}00^{4n}.$$

A gyakorlatban c_{ij} a következőképpen állítható elő t-ből: az i-edik és a j-edik csúcsmezőben álló 1-es kivételével t összes mezőjébe 0-t írunk, a string elejére és a végére is illesztünk $4n - 4n$ db 0-t, valamint az i-edik csúcsmező után következő b db 0-ból törlünk $4n$ db-ot. (Ha $j = n$, akkor c_{in} második 1-ese után a fenti leírás a következőt adja: $c_{in} = \ldots 10^b(00^b)^{-1}00^{4n}$, ahol a negatív kitevőt úgy kell értelmezni, hogy a második 1-es utáni b db 0-t és a string végén lévő $4n+1$ db 0-ból 1-et törölni kell.) A konstrukcióból látszik, hogy $|c_{ij}| = 5n + b(n-1)$, illetve egy további fontos tulajdonság, hogy ha nem szúrunk be gapeket egyik stringbe sem, akkor t-t és c_{ij}-t csak úgy lehet egymáshoz illeszteni, hogy a c_{ij}-ben lévő két 1-es közül legfeljebb az egyik lehet a neki megfelelő csúcsmezőben lévő 1-essel párosítva. A konstrukció célja az lesz, hogy ha t-ben a v_i-nek megfelelő csúcsmező 1-ese bármely c_{ij}-ben vagy c_{ki}-ben nem a neki megfelelő C-beli stringben található 1-essel van párosítva, akkor v_i nincs benne a független csúcshalmazban.

Definiáljuk S illesztéseinek egy viszonylag természetesen adódó osztályát, amely fontos szerepet fog játszani a bizonyítás további részében:

9. Definíció. *Kanonikus illesztésnek nevezzük S egy többszörös szekvenciaillesztését, ha a következők teljesülnek: a T-beli stringekbe nincsenek gapek beszúrva, azaz minden T-hez tartozó sorban csak t előtt vagy után szerepelhetnek gapek, valamint a stringek egymás alatt helyezkednek el; a P-beli illesztett stringek szintén egymás alatt helyezkednek el és 1-eseik a T-beli stringek csúcsmezőinek 1-eseivel vannak párosítva; valamint minden $c_{ij} \in C$ string úgy van a T-beli stringekhez illesztve, hogy c_{ij} első vagy utolsó $4n$ db 0-ja gapekkel van párosítva, c_{ij} fennmaradó része pedig a T-beli stringek 0-ival és 1-eseivel (így c_{ij}-be sem történhet gapek beszúrása, csak előtte vagy utána szerepelhetnek gapek).*

Példa: Legyen G egy három csúcsból és két élből álló út (így persze G nem 3-reguláris). Ekkor $b = 72$, így $|T| = |P| = 72$, valamint $|C| = 2$. G legnagyobb független csúcshalmazának mérete 2, így legyen $c = 2$. Ekkor $S = T \cup P \cup C$ egy lehetséges kanonikus illesztése:

				v_1								v_2								v_3			
	–	…	–	1	0	…	…	…	…	…	0	1	0	…	…	…	…	…	0	1	–	…	–
	.		.	.	.						.	.	.						.	.	.		.
T	.		.	.	.						.	.	.						.	.	.		.
	.		.	.	.						.	.	.						.	.	.		.
	–	…	–	1	0	…	…	…	…	…	0	1	0	…	…	…	…	…	0	1	–	…	–
	–	…	–	1	–	…	…	…	…	…	–	–	–	…	…	…	…	…	–	1	–	…	–
	.		.	.	.						.	.	.						.	.	.		.
P	.		.	.	.						.	.	.						.	.	.		.
	.		.	.	.						.	.	.						.	.	.		.
	–	…	–	1	–	…	…	…	…	…	–	–	–	…	…	…	…	…	–	1	–	…	–
c_{12}	0	…	0	1	0	…	0	1	0	…	0	0	0	…	…	…	…	…	0	0	–	…	–
c_{23}	–	…	–	0	0	…	…	…	…	…	0	0	0	…	0	1	0	…	0	1	0	…	0

A táblázatban felülről lefelé haladva, egymástól vízszintes vonalakkal elválasztva T, P és C illesztése látható. v_1 és v_2, illetve v_2 és v_3 mezője között $b - b$ db mező található, v_1 mezője előtt és v_3 mezője után pedig $4n - 4n$ db. c_{12} és c_{23} illesztéséből látható, hogy ez a kanonikus illesztés azt fejezi, hogy a független csúcshalmazba v_1-et és v_3-at választottuk be, v_2-t (az út középső csúcsát) pedig nem, mivel így a v_1-ből kiinduló éleknek megfelelő minden C-beli stringnél a v_1-nek megfelelő 1-es van a hozzá tartozó csúcsmezőhöz párosítva, és ugyanez teljesül a v_3-ból kiinduló élekre is.

A d költségfüggvény definíciójából látható, hogy S bármely illesztésére annak költségét $\frac{1}{2}d(S,S)$ fogja adni, továbbá S particionálását kihasználva

$$\frac{1}{2}d(S,S) = \frac{1}{2}d(T,T) + \frac{1}{2}d(P,P) + \frac{1}{2}d(C,C) + d(T,P) + d(T,C) + d(P,C).$$

Vizsgáljuk meg, az összeg egyes tagjaira milyen összefüggések teljesülnek:

(1) $d(T,T) = d(P,P) = 0$ minden kanonikus illesztésben a definícióból adódóan, illetve minden optimális illesztésben is a korábbi lemmából következően.

(2) $d(T,P) = (n-c+b(n-1))b^2$ bármely kanonikus illesztést tekintve: ugyanis egy T-beli és egy P-beli stringet vizsgálva egy kanonikus illesztésben t összes ($n+b(n-1)$ db) mezőjében eltérés van, kivéve azt a c db pozíciót, amelyben p 1-esei t csúcsmezőihez vannak illesztve, azaz egy ilyen pár költsége $n-c+b(n-1)$. Mivel $|T| = |P| = b$, így $d(T,P)$-t b^2 db ilyen pár költségének összege fogja alkotni. Az is látható, hogy egy optimális illesztésre is érvényes ez a becslés, figyelembe véve, hogy egy $t-p$ párt nem lehet $n-c+b(n-1)$-nél kisebb költséggel egymáshoz illeszteni, valamint hogy a T-ben, illetve a P-ben szereplő stringeknek a korábbi lemma miatt egymás alatt kell elhelyezkedniük, így $d(T,P)$ elérhető minimális költsége egy $t-p$ pár optimális költségének b^2-szerese.

(3) $d(T,C) = 5nbm$ minden kanonikus illesztésben, mivel egy $t-c_{ij}$ párt vizsgálva a definíció szerint c_{ij}-ben $4n$ db 0 gapekkel van párosítva, valamint t-ben $n-1$, c_{ij}-ben pedig 1 db 1-es 0-kkal van illesztve, ezzel további n költséget okozva, a többi mezőben viszont a két stringnél ugyanolyan karakterek szerepelnek. Ezenkívül az is teljesül, hogy $5n$-nél kisebb költséggel nem lehet egymáshoz illeszteni egy $t-c_{ij}$ párt, így egy tetszőleges illesztésre $d(T,C) \geq 5nbm$.

(4) $d(P,C) \geq (5n+b(n-1))mb - 3cb$ teljesül minden kanonikus illesztésre: a $d(p,C)$ értéket vizsgálva látható, hogy ha minden $p-c_{ij}$ pár minden pozíciójában költség keletkezne, akkor $d(p,C) = (5n+b(n-1))m$ lenne. Viszont tudjuk, hogy p 1-esei egy kanonikus illesztésben c db csúcsmezőhöz vannak illesztve, így ezek ezt az értéket csökkenthetik, hiszen minden c_{ij}-ben az általa tartalmazott két 1-es közül az egyik szintén egy csúcsmezőhöz van illesztve. Egy csúcsmező legfeljebb 3-mal csökkentheti ilyen módon ezt a költséget, mivel G 3-reguláris, így az összes C-beli stringet tekintve és egy adott csúcsmezőt vizsgálva legfeljebb 3 db 1-es szerepelhet ezekben a

pozíciókban. Így a $d(p, C) \geq (5n + b(n-1))m - 3c$ alsó becslést kaptuk, amelyből következően $d(P, C) \geq (5n+b(n-1))mb-3cb$. Az előbbi levezetésből az is látható, hogy ha G tartalmaz c elemű független csúcshalmazt, akkor létezik S-nek olyan kanonikus illesztése, amelyre a becslésben egyenlőség teljesül.

(5) $\frac{1}{2}d(C, C) \leq (8n+4)\binom{m}{2} < 5nm^2$ bármely kanonikus illesztésben, mivel a definícióból következően két C-beli stringet tekintve az egyik string elején, a másiknak pedig a végén $4n - 4n$ db 0 lehet gapekkel párosítva, továbbá összesen legfeljebb 4 db 1-es lehet 0-kkal illesztve, a többi mezőben biztosan egyezés van. Így egy ilyen pár költsége legfeljebb $8n + 4$, amelyből az állítás már következik.

A fenti becsléseket figyelembe véve a következőképpen határozzuk meg az MSA probléma K felső korlátjának értékét:

$$K = (n - c + b(n-1))b^2 + 5nbm + (5n + b(n-1))mb - 3cb + 5nm^2$$

Az előbbi becslésekkel összevetve nyilvánvaló, hogy ha G-ben van c elemű független csúcshalmaz, akkor S-nek létezik legfeljebb K költségű többszörös szekvenciaillesztése (ráadásul ilyen kanonikus illesztés is létezik). Így már csak azt kell belátni, hogy amennyiben G-ben nincs c db független csúcs, akkor S minden illesztésének költsége nagyobb, mint K.

Először vizsgáljuk meg a kanonikus illesztések költségét ebben esetben. A (4) becslést tekintve megállapítható, hogy mivel nincs c elemű független csúcshalmaz G-ben, ezért minden P-beli string 1-esei közül legalább 1 db egy olyan csúcsmezőhöz van illesztve, amely pozícióban a C-beli stringeket vizsgálva legfeljebb 2 db 1-es található, ez pedig legalább b-vel növeli $d(P, C)$ költségét. Így egy kanonikus illesztés költsége ekkor legalább $K - 5nm^2 + b > K$, ahol kihasználásra került, hogy $\frac{1}{2}d(C, C) \geq 0$. Ebből következően ha igazoljuk, hogy ilyenkor egy optimális illesztés mindenképpen kanonikus, akkor a tétel bizonyítása befejezettnek tekinthető.

Tekintsünk ezért egy optimális illesztést, amelyről azt gondoljuk, hogy nem kanonikus. Figyelembe véve, hogy az (1) és (2) becslések minden optimális illesztésre is teljesülnek, valamint hogy $d(T, C)$ egy optimális illesztésben sem lehet kisebb költségű, mint a (3) becslésben, látható, hogy az egyetlen lehetőség arra, hogy egy kanonikus illesztésnél kisebb költséget érjünk el, az az, ha $d(P, C)$ értékét próbáljuk csökkenteni. Az az eset, hogy a $d(P, C)$ költség egy kanonikus illesztéshez viszonyítva kisebb értéket ér el, csak úgy fordulhat elő, hogy néhány C-beli stringben mindkét 1-es a P-beli stringek által kiválasztott csúcsmezők valamelyikéhez van illesztve. Legyen az ilyen nem

kanonikus módon illesztett C-beli stringek száma r; egy kanonikus illesztéshez képest így a $d(P,C)+\frac{1}{2}d(C,C)$ összeg nagysága legfeljebb $2br+5nm^2$-tel csökkenhet. Ahhoz viszont, hogy egy c_{ij}-ben mindkét 1-es egy csúcsmező pozíciójába kerüljön, legalább $4n$ db gapet be kell szúrni, ezáltal a $d(T,C)$ érték legalább $(8n-2)br$-rel nagyobb lesz, mint egy kanonikus illesztésben. Mivel $(8n-2)br > 2br+5nm^2$, így valójában nem sikerült egy kanonikus illesztéshez képest javítást elérnünk, azaz ha G-ben nincs c elemű független csúcshalmaz, akkor az optimális illesztés biztosan kanonikus, így ilyenkor minden illesztés költsége nagyobb, mint K. □

4.3. A többszörös szekvenciaillesztés probléma egy lehetséges approximációja: a Clustal program működése

Mint ahogy az előző alfejezetben is szerepelt, a többszörös szekvenciaillesztés feladat **NP**-teljes, így a megoldására vonatkozó teljesen általános és gyors algoritmus megtalálása nem valószínűsíthető. Mindazonáltal a probléma fontossága miatt számos közelítő algoritmust dolgoztak ki, amelyek közül a következőkben az egyik leggyakrabban alkalmazottat, a Clustal programcsalád approximációs eljárását mutatom be.

A Clustal algoritmusa az úgynevezett progresszív illesztési konstrukció elvét követi. A módszer lényegét tekintve páronkénti illesztésekkel dolgozik, a két egymásra leginkább hasonlító szekvenciától kezdődően a legkevésbé egyezők felé haladva.

Az algoritmus a következő lépésekre bontható:

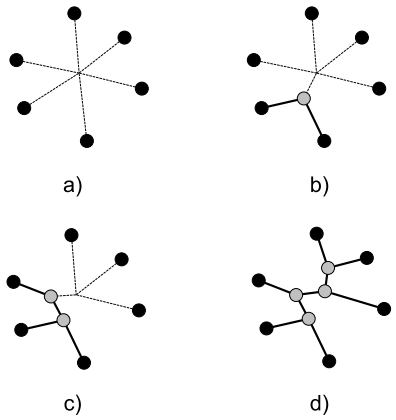
(1) Az összes páronkénti illesztés költségének megállapítása (dinamikus programozással), az egyes stringpárok optimális illesztéseinek költségét tartalmazó D távolságmátrix létrehozása;

(2) Ún. vezérfa (guide tree) konstruálása neighbor joining módszerrel (részletezés később);

(3) A vezérfa által kijelölt szekvenciák optimális páronkénti illesztésének végrehajtása;

(4) A vezérfa felépítését követve az előző lépésben nem illesztett szekvenciák hozzáillesztése a páronként illesztett szekvenciákhoz, illetve utóbbiak egymáshoz illesztése.

Az algoritmus addig fut, ameddig van olyan szekvencia, amely nem került illesztésre.



4.1. ábra. Neighbor joining algoritmus: az illesztendő szekvenicákat egy gráf csúcsai reprezentálják (az ábrán fekete színnel jelölve), az algoritmus kezdetén ezeket egy fiktív központi csúccsal rendelkező vezérfa (ld. a) ábra) köti össze. A b) és a c) ábrán az algoritmus első, illetve második lépése látható: mindkét lépésben egy-egy új, a kiindulási szekvenciák csúcsai között nem szereplő csúcs került felvételre (szürke színnel jelölve). A d) ábrán az algoritmus által felépített vezérfa látható, amelynek létrehozásához két további kiegészítő csúcs volt szükséges.

Az algoritmus (2) lépésében szereplő neighbour joining („szomszédok egyesítése") módszer $n \geq 3$ számú szekvencia esetén a következő fázisokból áll:

(1) Az előző algoritmus D mátrixának felhasználásával szükséges egy Q mátrix definiálása, amelynek $q(i,j)$ elemeire a következő összefüggés teljesül:

$$q(i,j) = (n-2)d(i,j) - \sum_{k=1}^{n} d(i,k) - \sum_{k=1}^{n} d(j,k),$$

ahol $d(i,j)$ s_i és s_j minimális távolságát jelöli.

(2) A kiinduló vezérfa egy csillag, amelynek levelei az illesztendő szekvenciák. Ezt módosítjuk úgy, hogy a Q mátrix minimális eleméhez tartozó két szekvenciát egy új u csúccsal kötjük össze, u-t pedig a kezdeti csillag központi csúcsával.

Legyen s_i és s_j a Q mátrix minimális eleméhez tartozó két szekvencia. Ekkor a vezérfában az s_i-t, illetve s_j-t u-val összekötő él súlya:

$$\delta(s_i,u) = \frac{1}{2}d(i,j) + \frac{1}{2(n-2)}\Big(\sum_{k=1}^{n} d(i,k) - \sum_{k=1}^{n} d(j,k)\Big)$$

$$\delta(s_j, u) = d(i, j) - \delta(s_i, u)$$

(Ezek az élsúlyok az algoritmus hátralévő részében változatlanok maradnak.)

(3) Ezt követően szükséges a D és a Q mátrixok újradefiniálása, amelyekben ezután s_i és s_j helyett u szerepel, a D-ben lévő megfelelő súlyok pedig a következőképpen számíthatók ki:

$$d(u, k) = \frac{1}{2}\Big(d(i, k) + d(j, k) - d(i, j)\Big)$$

Ezután a Q mátrix (új D-beli értékek felhasználásával történő) ismételt kiszámításával iteráció kezdődik, amely addig tart, ameddig meghatározásra kerül a vezárfa minden élének súlya ($n - 3$ db iteráció). Az algoritmus minden lépésben egy legfeljebb n x n méretű mátrixot hoz létre, így az eljárás polinomiális, maximális műveletigénye $O(n^3)$. [26]

5. fejezet

Néhány fontos újabb eredmény, illetve nagy jelentőségű, jelenleg is nyitott probléma a közelítő algoritmusok területéről

Az **NP**-teljes problémák közelíthetőségének fontosságát mutatja többek között az a tény is, hogy kidolgozásuk óta számos matematikus dolgozott ezeknek a feladatoknak az approximálhatóságán, illetve hogy ez a munka világszerte jelentős számú kutatócsoport részvételével jelenleg is zajlik. Szakdolgozatom ezen részében megpróbálom röviden áttekinteni a közelítő algoritmusok kapcsán elért legfontosabb friss eredményeket, elsősorban azokra a problémákra koncentrálva, amelyeket a dolgozat korábbi fejezeteiben is érintettem.

Az utazó ügynök problémával foglalkozó szakértők az utóbbi években az egyik legintenzívebben a következő speciális feladatot tanulmányozták: legyenek $X_1, \ldots, X_n$ egyenletes eloszlású valószínűségi változók $[0,1]^2$-en, és jelölje L_n^* az összes X_i-n áthaladó legrövidebb út hosszát az euklideszi távolság mellett. Már az 1950-es évektől ismert volt, hogy $\frac{L_n^*}{\sqrt{n}} \to \beta$ 1 valószínűséggel, ha $n \to \infty$, az azóta eltelt évtizedekben pedig β-ra számos közelítés született (alsó és felső becslés egyaránt). Ezek közül a legfrissebb és jelenleg a legpontosabb Stefan Steinerberger approximációja, amely szerint $\frac{5}{8} + \frac{19}{5184} \approx 0,63 \leq \beta \leq 0,92$. [27]

A maximális vágás problémára **MAXSNP**-teljességéből következően nem létezik polinomiális idejű közelítési séma, kivéve ha $\mathbf{P} = \mathbf{NP}$, ugyanakkor ennél az általános eredménynél konkrétabb megállapítások is megfogalmazhatók ezzel a feladattal kapcsolatban. A 2.2. pontban bemutatottnál lényegesen pontosabb közelítő algoritmust fejlesztett ki Goemans és Williamson szemidefinit programozás és véletlenített kerekítés (egy IP-feladat LP-relaxáltjára kapott tört értékű megoldás komponenseinek randomizált módon történő kerekítése az eredeti feladat megoldására) segítségével: eljárásuk egy $(1-\alpha)$-approximáxiót biztosít tetszőleges kiindulási gráf esetén, ahol $\alpha = \frac{2}{\pi} \min_{0 \leq \theta \leq \pi} \frac{\theta}{1-\cos\theta} \approx 0,878$. [12] Fontos megjegyezni, hogy amennyiben az ún. "egyedi játék-sejtés" (unique games conjecture [17]) igaz, akkor nem érhető el ennél jobb közelítés az általános problémára, ugyanakkor ha ezt a sejtést nem tesszük fel, még akkor is **NP**-nehéz feladat $\frac{1}{17} \approx 0,059$-nél jobb közelítő algoritmust adni a feladatra, mint ahogy azt Johan Hastad bebizonyította. [13]

Bár a $MAX-SAT$ probléma **MAXSNP**-teljes, ez a tény nem jelenti azt, hogy (a maximális vágáshoz hasonlóan) a témával foglalkozó matematikusok ne próbálnának egyre jobb approximációs algoritmusokat megalkotni. A szakdolgozatom 2.3. pontjában szereplő, az általánosabb $k-MAX-GSAT$ probléma közelítéséből származó $\frac{1}{2}$-approximációt megjavítva az elmúlt években kidolgozásra került többféle $\frac{1}{4}$-közelítő algoritmus is. (Érdekesség, hogy egy, az előző eredményt biztosító randomizált algoritmus megalkotása után, a konstrukció mögött álló ötleteket felhasználva sikerült létrehozni egy olyan eljárást is, amely már determinisztikusan, a korábban már említett LP-IP átmenet segítségével képes elérni ugyanezt a közelítést.) [25]

Az előbbiekben felsorolt problémákkal szemben a (súlyozás nélküli) minimális lefogó csúcshalmaz feladattal kapcsolatban az elmúlt évtizedekben sem sikerült a dolgozatomban is szereplő egyszerű $\frac{1}{2}$-közelítő algoritmusnál hatékonyabbat találni, így a témával foglalkozó szakemberek körében elfogadott sejtésnek számít, hogy nem is lehetséges olyan ε-approximációt kidolgozni erre a problémára, amelyre $\varepsilon < \frac{1}{2}$. A PCP-tétel bizonyítására épülő technikák segítségével Dinur és Safra igazolták, hogy a minimális lefogó csúcshalmaz problémára nem létezhet 0,26-approximáció vagy ennél jobb közelítés, kivéve ha $\mathbf{P} = \mathbf{NP}$. [9] Ha pedig igaznak bizonyul a maximális vágással kapcsolatban már említett egyedi játék-sejtés, abból következik, hogy valóban nem lehet $\frac{1}{2}$-nél jobb közelítő algoritmust előállítani. [18]

Természetesen az előbbiekben ismertetett eredmények ellenére továbbra is számos fontos nyitott probléma határozható meg az approximációs algoritmusokkal összefüg-

gésben. Példaként említhető, hogy a maximális vágás probléma közelítéséhez kapcsolódóan hivatkozott $(1-\alpha)$-approximáció elérése jelenleg csak szemidefinit programozás alkalmazásával valósítható meg, ez azonban meglehetősen számításigényes eljárás. Logikusan felmerülő kérdés, hogy létezik-e esetleg egy olyan algoritmus, amely eléri vagy akár csak jól közelíti ennek az α-approximációnak az eredményét, ugyanakkor könnyebben kiszámítható (pl. egy primál-duál módszer). Ide kapcsolódó ígéretes eredménynek nevezhető, hogy Luca Trevisan egy sajátérték-meghatározásra alapuló 0,469-közelítő algoritmust tudott kidolgozni, amelynek számításigénye messze elmarad a Goemans-Williamson-algoritmusétól. [28]

Szintén számos fontos, approximációval kapcsolatos, jelenleg még megoldatlan kérdés kapcsolódik a gráfok színezhetőségének problémaköréhez, hiszen mint ismert, egy adott gráf kromatikus számának meghatározása szintén **NP**-teljes feladat. Ezen problémák közül az egyiknek az inputja egy irányítatlan, 3-színezhető gráf ($G = (V, E)$), a feladat pedig egy megfelelő k-színezés találása a lehető legkisebb k mellett. Erre a problémára sikerült kidolgozni egy szemidefinit programozást használó algoritmust, amely megfelelően nagy gráfokra nagyságrendileg legfeljebb $\sqrt[5]{n}$ színt használ, ahol $|V| = n$. [2] Emellett viszont az is ismert eredmény, hogy **NP**-nehéz probléma egy 3-színezhető gráf egy jó 4-színezésének megtalálása, illetve ha a már többször említett egyedi játéksejtés teljesül, akkor adott (nem feltétlenül 3-színezhető) G és k mellett hasonlóan **NP**-nehéz annak eldöntése, hogy $\chi(G)$ eleme-e a következő halmaznak: $\{3, k, k+1, \dots\}$. [16], [8] Ebből adódóan kézenfekvő célkitűzés lehet egy olyan algoritmus kidolgozása, amely elég nagy gráfok esetén egy legfeljebb $\log n$-nel arányos számú színt használó színezést ad meg.

A gyakorlati alkalmazások szempontjából is kiemelt jelentőséggel rendelkezik az általánosított Steiner-fa probléma, amelynek inputjában adott egy irányítatlan $G = (V, E)$ gráf $c \geq 0$ élsúlyozással, valamint V-ben ki van jelölve k db $s_i - t_i$ forrás-nyelő pár. A feladat az élek egy olyan minimális súlyú F részhalmazának meghatározása, hogy $\forall i$-re s_i és t_i legyenek összekötve (V, F)-ben. 1995-ben sikerült a problémára egy primál-duál módszert alkalmazó $\frac{1}{2}$-közelítő algoritmust találni [1], valamint ha $\forall i : s_i = s$, akkor visszakapjuk az eredeti Steiner-fa feladatot, amelynek megoldására egy 0,28-approximációs algoritmus is ismert [5]. Továbbá azt a negatív eredményt is sikerült meghatározni, hogy ha $\mathbf{P} \neq \mathbf{NP}$, akkor nem létezhet $\frac{1}{96}$-nál jobb közelítő algoritmus az eredeti Steiner-fa problémára. [6] Ezen eredményeket figyelembe véve célszerű lenne az általánosított Steiner-fa feladatra egy olyan ε-approximációt találni, amelyre $\varepsilon < \frac{1}{2}$.

Irodalomjegyzék

[1] Ajit Agrawal, Philip Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *Society for Industrial and Applied Mathematics*, 24 (3):440–456, 1995.

[2] Sanjeev Arora, Eden Chlamtac, and Moses Charikar. New approximation guarantee for chromatic number. *STOC '06*, 2006.

[3] Piotr Berman and Toshihiro Fujito. On approximation properties of the independent set problem for degree 3 graphs. In *In Proc. of Workshop on Algorithms and Data Structures*, pages 449–460. Springer, 1995.

[4] Paola Bonizzoni and Gianluca Della Vedova. The complexity of multiple sequence alignment with SP-score that is a metric. *Theoretical Computer Science*, 259:63–79, 2001.

[5] Jaroslaw Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. An improved LP-based approximation for Steiner tree. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 583–592, 2010.

[6] Miroslav Chlebík and Janka Chlebíková. The Steiner tree problem on graphs: inapproximability results. *Theoretical Computer Science*, 406:207–214, 2008.

[7] Louis N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. *Technical report GSIA*.

[8] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. *SIAM Journal on Computing*, 39 (3):843–873, 2009.

[9] Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162 (1):439–485, 2005.

[10] Isaac Elias. Settling the intractability of multiple alignment. In *Proc. of the 14th Ann. Int. Symp. on Algorithms and Computation (ISAAC*, pages 352–363. Springer, 2003.

[11] Pál Erdős and Alfréd Rényi. On the evolution of random graphs. *MTA Matematikai Kutatóintézet Közleményei*, 5:17–61, 1960.

[12] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the Association for Computing Machinery*, 42 (6):1115–1145, 1995.

[13] Johan Hastad. Some optimal inapproximability results. *Journal of the ACM*, 48 (4):798–859, 2001.

[14] Dorit S. Hochbaum. Approximating covering and packing problems: set cover, vertex cover, independent set and related problems. In Dorit S. Hochbaum, editor, *Approximation Algorithms for* **NP***-hard problems*, pages 94–143. PWS Publishing Company, 1997.

[15] David S. Johnson. The **NP**-completeness column: an ongoing guide. *Journal of Algorithms*, 6 (6):291–305, 1985.

[16] Sanjeev Khanna, Nathan Linial, and Shmuel Safra. On the hardness of approximating the chromatic number. *Combinatorica*, 20 (3):393–415, 2000.

[17] Subhash Khot, Guy Kindler, Elchanan Mossel, Andrzej Lingas, and Eike Seidel. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37 (1):319–357, 2007.

[18] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. *Journal of Computer and System Sciences*, 74 (3):335–349, 2008.

[19] Shen Lin and Brian W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operation Research*, 21 (2):498–516, 1973.

[20] Michael Mitzenmacher and Eli Upfal. *Probability and computing: randomized algorithms and probabilistic analysis*. Cambridge University, Cambridge, 2005.

[21] Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48 (3):443–453, 1970.

[22] Christos H. Papadimitriou. Számítási bonyolultság. 1999.

[23] Christos H. Papadimitriou and Mihalis Yannakakis. The travelling salesman problem with distances one and two. *Mathematics of Operation Research*, 18 (1):1–12, 1993.

[24] Shariefuddin Pirzada and Ashay Dharwadker. Applications of graph theory. *Journal of the Korean Society for Industrial and Applied Mathematics*, 11 (4):19–38, 2007.

[25] Matthias Poloczek, David P. Williamson, and Anke van Zuylen. On some recent MAX SAT approximation algorithms. *CoRR*, abs/1308.3405, 2013.

[26] Naruya Saitou and Masatoshi Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4 (4):406–425, 1987.

[27] Stefan Steinerberger. New bounds for the travelling salesman constant. *Advances in Applied Probability*, 47 (1):27–36, 2015.

[28] Luca Trevisan. Max cut and the smallest eigenvalue. *CoRR*, abs/0806.1978, 2008.

[29] Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1 (4):337–348, 1994.