ÖT+EGY KIEMELT DOLGOZAT

ERDŐS PÉTER

A Matematikus Doktori Szakbizottság útmutatása szerint alább dióhéjban ismertetem a legfontosabbnak gondolt dolgozataimat. Mivel ezek közül egy tematikusan nem illik a disszertcáiómba, és az abban szereplő eredményeknél sokkal korábban született, ezért még egy dolgozatot csatoltam a listához, amely most nyomdában van, de szerintem érdeklődést fog kelteni.

P.L. Erdős - P. Frankl - G.O.H. Katona: Extremal hypergraphs problems and convex hulls, *Combinatorica* **5 (1985), 11–26.**

Az extremális halmazrendszerek elméletében a tipikus kérdés a következő alakú: adott egy véges alaphalmaz részhalmazainak rendszere (általában valamilyen kombinatorikus feltétellel definiálva), ahol maximalizálni kívánjuk a rendszer elemszámát, vagy a részhalmazok elemszámának összegét, esetleg - általánosabban - a részhalmazok elemszámától függő valamely súlyfüggvény összegét. Egyszóval a részhalmazok elemszámától függő lineáris optimalizálást szertnénk végrehajtani. A szokásos módszerek mellett minden egyes optimalizálást önállóan kell megoldani.

Az idézett cikkben (illetve iker-cikkében) megkezdtük halmazrendszerek *konvex burkának* vizsgálatát: valamely n-halmaz egy részhalmaz rendszerének a *profilja* egy $n+1$-hosszú vektor: az i-ik koordináta az i-elemű részhalmazok számát adja meg, és az $n+1$ dimenziós euklideszi tér egy (pozitív oktáns beli) pontjának tekinthető. A szóba jöhető összes halmazrendszer profiljai egy ponthalmazt alkotnak ugyanebben a térben. Ezután bármely, a részhalmazok elemszámában lineáris maximalizálási feladatot elegendő a kapott ponthalmaz csúcspontjain megoldani.

Az elárás előnye legalább kettős: ha egyszer sikerült a csúcspontokat leírni, akkor bármely, újonnan felmerülő maximalizálást is elegendő rajtuk megoldani. (Erre sok későbbi alkalmazás mutatott példát.) A másik nyilvánvaló előny - evvel összefüggésben - a figyelembe veendő csúcsok száma: míg elvben általában exponenciálisan sok részhalmazrendszer közül kell az optimálisat kiválasztani, a szóbajöhető csúcsok száma az esetek többségében csak polinomiális, továbbá még exponenciális méretű csúcshalmazzal rendelkező feladatok esetén is a csúcsokhoz tartozó rendszerek szerkezete egyszerű.

A hivatkozott cikkben ennek az eljárásnak elméleti alapjait fektettük le, bevezettük a szükséges definíciókat és módszereket adtunk a csúcsok meghatározásának egyszerűsítésére.

A dolgozat egy új területet indított az elméleten belül. Az elmélet de facto alapkönyve (Engel: *Sperner Theory,* Encyclopedia of Mathematics and Its Applications, Vol. 65 Cambridge University Press, 1997.) önálló fejezetetben tárgyalja.

P.L. Erdős - L. A. Székely: On weighted multiway cuts in trees, *Mathematical Programming* **65 (1994), 93–105.**

A *multiway cut (MC)* probléma, az él-Menger tétel kettőnél több színre történő esetleges általánosítása, fontos helyet tölt be a kombinatorikus optimalizálásban. A feladat polinom időben megoldható síkgráfokon, korlátos számú terminálpont esetén, egyébként NP-teljes feladat. (E. Dahlhaus - D.S. Johnson - C.H. Papadimitriou - P.D. Seymour - M. Yannakakis: The complexity of multiterminal cuts, *SIAM J. Computing* **23** (1994), 864–894.) Fenti cikkben (és előzményeiben) bevezettük az MC probléma egy általánosítását (néhényan *színezett MC* (szMC) problémának nevezik), amely természetes módon adódott egy bioinformatikai (evolúciós fák elmélete) problémából. Itt terminálpontok egy N halmaza adott, továbbá ennek egy k-színnel történő $\gamma : N \to [k]$ színezése. Egy szMC élek egy olyan halmaza, amely bármely két, eltérő színű terminálpontot szeparál. Cél: a lehető legkisebb élszámú (súlyú) szMC megtalálása. Mint Dahlhaus és társai kimutatták az szMC (amit hoszszabban elemeztek a cikkükben) bonyolultabb, mint az eredeti MC, már síkgráfokon és azonosan 1 élsúllyal is NP-teljes.

Cikkünkben megmutattuk, hogy a probléma polinomiális megoldható "fa szerű" objektumokon, és sikerült egy újtípusú minimax tételt is bebizonyítanunk, amelyet aztán (másoknak) sikerült is az eredeti bioinformatikai problémára alkalmazni. A cikk alkalmazásokat nyert továbbá a robot vision elméletben, klasszifikációs problémákban illetve szétosztott számítógéphálózatok esetén a kommunikációs költség minimalizálásában.

L.A. Székely - M.A. Steel - P.L. Erdős: Fourier calculus on evolutionary trees, *Advances in Appl. Math* **14 (1993), 200–216.**

Az 1990-es évek elején áttörést jelentett az evolúciós fák elméletében a Mike Hendy által bevezetett *Hadamard konjugáltak* módszere. A biológusok gyakran képzelik el az evolúció történetét, mint egy ismeretlen (gyakran györkeres) bináris fa mentén fejlődő két állapotú Markov modell. Ilyenkor az élek mentén jelentkező eloszlások illetve az észlelt levél-színezés eloszlások között egy Hadamard konjugált kapcsolat van: bármelyikből kiszámítható a másik. A módszer nagy számítás igényű, de megbízható.

Az új-zélandi iskola képviselőivel együttműködve kiterjesztettük a módszert négy állapotú (korábbi cikkek), illetve tetszőleges Abel csoport értékű (az idézett cikk) Markov modellekre is. Ilyenkor a jelzett eloszlások között Fourier inverz párkapcsolatok vannak. A leírt eljárásoknak egyfelől gyakorlati alkalmazásai vannak. Ezt jól illusztrálja, hogy a módszerből másfél éven belül tankönyv anyag lett. Másfelől már több elméleti következmény is kiderült: a módszer szoros kapcsolatot mutat a fizikai mezőelméletekben alkalmazott módszerekkel (P.D. Jarvis - J.D. Bashford), illetve modern algebrai geometriai eredmények is kapcsolódnak hozzá (trópikus geometriák illetve torikus ideálok - (E.S. Allman - J.A. Rhodes; L. Pachter - B. Sturmfels, stb).

P.L. Erdős - M.A. Steel - L.A. Székely - T.J. Warnow: A few logs suffice to build (almost) all trees (I), *Random Structures and Algorithms* **14 (1999), 153–184.**

Az evolúciós fák rekonstrukciójának egyik nagy osztálya az un. *supertree* módszerek: a címkézett leveleket tartalmazó keresett bináris fát topológikus részfái átlapoló rendszeréből kívánjuk helyreállítani. Ha a részfák ellentmondók, akkor ezt az ellentmondást valamilyen módon kezelni kell. Akkor is baj van, ha nem áll rendelkezésre elegendő részfa.

A supertree módszerek talán legtöbbet alkalmazott eljárása, amikor négy levelet tartalmazó részfákból, un. *quartet*-tekből végezzük a rekonstrukciót. Közkedveltségét legfőbbképpen annak köszönheti, hogy a négy levelet tartalmazó részfák helyreállítása egyszerűnek tekinthető, és sokféle bemenet (azaz biológiai adat) alkalmazható. Ismert, ha minden quartet helyes, akkor a rekonstrukció könnyű (és gyors). Azonban annak eldöntése, hogy egy adott quartet rendszer ellentmondás mentes-e egy NP-nehéz feladat. Az is közismert továbbá, hogy a gyakorlati alkalmazásokban mindig keletkeznek hibás (pontosabban ellentmondó) quartetek.

Az idézett cikkben először is felismertük azt a nem meglepő tényt, hogy minél messzebb vannak az eredeti fában egy adott quartet levelei, annál valószínűbb a quartet hibás rekonstruálása. Majd bebizonyítottuk azt a tényt, hogy elegendő csupa "rövid" (n levél esetén legfeljebb nagyjából $2 \log n$ hosszú) ágakat tartalmazó quarteteket tekinteni. Ez egy determinisztikus eredmény, ahol az eredeti fa dönti el, mik a rövid ágak. Ez az adat persze (sajnos) ismeretlen a konkrét alkalmazásokban: közvetett (például távolság) adatokból kell eldönteni, milyen quartetekben vannak rövid ágak.

A cikkben különféle Markov modellek mellett több ilyen eljárást is kifejlesztettünk, közülük a DCM módszer a legfontosabb. Az eljárások hatékonysága (gyorsasága és adatszükséglete) észszerű feltételek mellett kiszámítható volt. A kapott érték - nagyon meglepő módon - közel volt a szintén ebben a cikkben kifejlesztett alsó korláthoz, az eljárások majdnem optimálisak. Végül a cikkbe arra is javaslatot tettünk, miként lehet egy konkrét eljárás hatékonyságát értékelni.

P.L. Erdős - M.A. Steel - L.A. Székely - T.J. Warnow: A few logs suffice to build (almost) all trees (II), *Theoretical Computer Science*, **221** **(1-2) (1999), 77–118.**

Ebben a cikkben először különféle távolság alapú fa-rekonstrukciós algoritmusok hatékonyságának összehasonlítására fejlesztettünk ki egy módszert. Ez az elemzés sok elméleti munkában kerül felhasználásra – például a NeighborJoining algoritmust (a jeleneleg talán legnépszerűbb faépítő eljárást) elméletileg megalapozó Atteson cikkben. A cikk fő hozzájárulása a quartet módszerek témájához egy újonnan fejlesztett algoritmus, a *Witness-Antiwitness Módszer*, amely a DCM-nél csak kicsit hosszabb input sorozatokból lényegesen gyorsabban tudja 1 valószínűséggel rekonstruálni a fát.

Érdemes még megjegyezni, hogy az SQM módszerek inputként inhomogén adatokat is képesek elfogadni. Ez ott döntő jelentőségű, ahol a vizsgálandó élőlények diverzifikációja miatt homogén adatok nem elérhetők.

A két utóbbi cikkre rengeteg hivatkozás történt. A meghatározott hatékonyság korlátokhoz közel teljesítő eljárásokat elnevezték *fast converging* módszereknek. (Ezek szerint a cikkeinkben leirtak az első ilyen eljárások.) Az ott lefektetett elvek alapján azóta sok további ilyen eljárást fejlesztettek ki és elemeztek. Az eredményeket minden azóta megjelent evolúciós fákkal foglalkozó könyvben részletesen elemezték. A módszerek továbbfejlesztésében éppen napjainkban történt egy nagy ugrás E. Mossel és tanítványainak kutatásai nyomán.

PLUSSZ EGY DOLGOZAT

P.L. Erdős - L. Soukup: How to split antichains in infinite posets, *Combinatorica* **27** (2) **(2007), ?–??.**

Egy P részben rendezett halmazban (posetben) egy antilánc akkor maximális, ha az antilánc alatti és feletti pontok együttesen kimerítik az egész P-t. Ez a maximális antilánc akkor *splittel*, ha van egy olyan $< B, C >$ rendezett patíciója, amelyre már a B alatti és a C feletti pontok is kimerítik az egész P-t. (Persze kizárólag maximális antilánc splittelhet.) Végezetül egy $y \in P$ elem *elvágó-pont* ha vannak további

$x, z \in P$ pontok ($x < y < z$), hogy az $[x, z]$ zárt intervallum megegyezik a $[x, y]$ és a $[y, z]$ zárt intervallumok uniójával. 1995 óta ismeretes, hogy minden elvágópont mentes véges posetben minden maximális antilánc splittel, továbbá, hogy az a kérdés: "vajon egy tetszőleges véges poset minden maximális antilánca splittele" egy NP-nehéz probléma. Az eltelt tíz évben a splittelés sokféle kapcsolatára derült fény. Ezek egyike a véges relációs struktúrák homomorfizmus posetjében bevezett (*általánosított*) *dualitás*, amely lényegében egy splittelés. (Lásd J. Nešetril munkáit.)

A cikkben (főleg megszámlálhatóan) végtelen posetek splitting tulajdonságaival foglalkozunk. Sikerült splittelő antiláncokat találnunk jónéhány elvágó-pont mentes végtelen posetben. Kifejlesztettünk egy módszert, amely azt méri, mennyire "nem splittel" egy maximális antilánc. Ezután azonosíttunk egy *lazaságnak* (angolul *looseness*) nevezett tulajdonságot, amelynek segítségével véges, nem maximális antiláncok splittelő illetve nem-splittelő maximális antiláncokká terjeszthetők ki. Ennek segítségével megkonstruáltunk egy nem-splittelő maximális antiláncot a négyzetmentes számok elvágópont-mentes posetjében, amely egy korábbi bonyolult, Ahlswede és Khachatrian nevéhez fűződő konstrukció általánosítása. A módszer később alkalmasnak bizonyult irányított gráfok homomorphismus posetjében valamely véges antilánc általánosított dualitássá való kiterjesztéséhez. Végezetül megmutattuk, hogy a kiválasztási axióma a ZF axióma rendszer mellett ekvivalens egy alkalmasan választott poset egy maximális antiláncának splittelhetőségével.

COMBINATORICA

COMBINATORICA is an international journal of the J. Bolyai Mathematis cal Society (Anker köz 1—3, Budapest, Hungary H-1061), published jointly by Akadémiai Kiadó (Publishing House of the Hungarian Academy of Sciences), Budapest and North-Holland Publishing Company, Amsterdam. The journal is published in four issues making up a volume 400 pages per year.

COMBINATORICA publishes research papers in English from various branches of combinatorics and theory of computing. Applications-oriented paper- containing new mathematical results are also welcome.

Mailing address for editorial correspondence:

COMBINATORICA
Mathematical Institute of the
Hungarian Academy of Sciences
Reáltanoda u. 13—15
H-1053 Budapest, Hungary

or

COMBINATORICA
R. L. Graham
AT&T Bell Laboratories
600 Moutain Ave,
Murray Hill, NJ 07974 U.S.A.

Distributors for Albania, Bulgaria, China, Cuba, Czechoslovakia, German Democratic Republic, Korean People's Republic, Mongolia, Poland, Romania, USSR, Viet-Nam, Yugoslavia:

Kultura
Hungarian Foreign Trading Co.
P. O. Box 149, H-1389 Budapest 62, Hungary

For all other countries:

North-Holland Publishing Company
P. O. Box 211, 1000 AE Amsterdam
The Netherlands

Subscription Information

Combinatorica (ISSN 0209—9683) is published in one volume of 4 issues a year. The subscription price for 1985 (Volume 5) is Dfl. 200,— + Dfl. 22,— p.p.h. = Dfl. 222,—. Our p.p.h. (postage, pack-

COMBINATORICA 5 (1) (1985) 11—26

EXTREMAL HYPERGRAPH PROBLEMS AND CONVEX HULLS

Péter L. ERDŐS, P. FRANKL and G. O. H. KATONA

Received 16 February 1984
Revised 19 March 1984

The *profile* of a hypergraph on n vertices is $(f_0, f_1, \dots, f_n)$ where f_i denotes the number of i-element edges. The extreme points of the set of profiles is determined for certain hypergraph classes. The results contain many old theorems of extremal set theory as particular cases (Sperner, Erdős—Ko—Rado, Daykin—Frankl—Green—Hilton).

1. Introduction

Let X be a finite set of n elements and $\mathcal{F}$ be a family of its subsets ($\mathcal{F} \subset 2^X$). Then $\mathcal{F}_k$ denotes the subfamily of the k-element subsets in $\mathcal{F}$: $\mathcal{F}_k = \{A: A \in \mathcal{F}, |A| = k\}$. Its size $|\mathcal{F}_k|$ is denoted by f_k. The vector $(f_0, f_1, \dots, f_n)$ in the $(n+1)$-dimensional Enclidean space $\mathbf{R}^{n+1}$ is called the *profile* of $\mathcal{F}$.

If α is a finite set in $\mathbf{R}^{n+1}$, the *convex hull* $\langle\alpha\rangle$ of α is the set of all convex linear combinations of the elements of α. We say that $e \in \alpha$ is an *extreme point* of α iff e is not a convex linear combination of elements of α different from e. It is well-known that $\langle\alpha\rangle$ is equal to the set of all convex linear combinations of its extreme points. That is, the determination of the convex hull of a set is equivalent to finding its extreme points.

$\mathcal{F}$ is a *Sperner-family* iff it contains no members A, B with $A \subset B$. In the previous paper we determined the extreme points of the set of profiles of all Sperner-families. This was an easy consequence of a well-known inequality. A family is *intersecting* if $A, B \in \mathcal{F}$ implies $A \cap B \neq \emptyset$. The main result of [5] determines the extreme points of the set of profiles of the intersecting Sperner-families.

On the other hand, the present paper starts a systematic treatment of the area. It tries to determine the extreme points of the set of profiles of the simplest known classes of families, using the methods of the previous paper. The effort is successful for 3 classes:

1. intersecting families,

2. k-*Sperner-families* (there are no $k+1$ different members satisfying $F_1 \subset \dots \subset F_{k+1}$),

AMS subject classification (1980): 05 C 35; 05 C 65, 52 A 20

3. $\mathscr{F}_1, \dots, \mathscr{F}_t$ are not necessarily disjoint families, where $G\in\mathscr{F}_i$, $H\in\mathscr{F}_j$, $i\neq j$, $G\neq H$ imply $G\not\subset H$.

Moreover, the method of the previous paper is analyzed here. One of the ideas of the proofs is the following. A cyclic ordering $\mathscr{C}$ is taken of the underlying set X and consider only the sets containg consecutive elements in $\mathscr{C}$. Any problem of the above type can be realized for these consecutive sets, as well. Their solution is easier but in some cases (in all the cases solved in these 2 papers) is sufficient. Theorem 4 describes the connection between the sets of extreme points of the original problem and of the "consecutive" variant. An example will be given ($F_1, F_2\in\mathscr{F}$ implies $|F_1\cap F_2|\geqq l$) when the original problem is hopeless while the "consecutive" variant can be solved. Theorem 4 is, of course, too weak in this case.

We also list some known extremal theorems which are consequences of our results.

For instance in Case 3 our method gives a unified proof of 3 different statements of [1].

2. General results (=tools)

2.1. *Essential extreme points.* Let $\mathbf{A}$ be a class of families of subsets of the n-element set X, that is, $\mathbf{A}\subset 2^{2^X}$. $\mu(\mathbf{A})$ denotes the set of profiles of the families belonging to $\mathbf{A}$:

$$\mu(\mathbf{A}) = \{(f_0, \dots, f_n)\colon f_i = |\mathscr{F}_i|,\ \mathscr{F}\in\mathbf{A}\}. \tag{1}$$

The set of extreme points of $\mu(\mathbf{A})$ is denoted by $\varepsilon(\mathbf{A})$.

The $\mathbf{A}$'s considered in this paper are *hereditary,* that is, $\mathscr{G}\subset\mathscr{F}\in\mathbf{A}$ implies $\mathscr{G}\in\mathbf{A}$. For hereditary $\mathbf{A}$'s there is a way of reduction of the set of extreme points. Before stating the theorem we have to introduce some more notations. $\mu^*(\mathbf{A})$ is the set of *maximal profiles:* $\mu^*(\mathbf{A})$ contains those elements $(f_0, \dots, f_n)$ of $\mu(\mathbf{A})$ for which $(g_0, \dots, g_n)\in\mu(\mathbf{A})$ $(g_0, \dots, g_n)\geqq(f_0, \dots, f_n)$ (it denotes $g_0\geqq f_0, \dots, g_n\geqq f_n$) imply $(f_0, \dots, f_n)=(g_0, \dots, g_n)$. Furthermore let $\varepsilon^*(\mathbf{A})=\varepsilon(\mathbf{A})\cap\mu^*(\mathbf{A})$ be the set of the *essential extreme points.*

Theorem 1. *Suppose that* $\mathbf{A}$ *is hereditary. Then any element of* $\varepsilon(\mathbf{A})$ *can be obtained by changing some coordinates of an element of* $\varepsilon^*(\mathbf{A})$ *to zero.* ∎

This fact is obvious. The proof requires very simple technique, therefore it is omitted.

The significance of the theorem is that for a given $\mathbf{A}$ it is sufficient to determine the set $\varepsilon^*(\mathbf{A})$. Changing the components to zero we obtain a set of vectors, these should be individually checked if they are extreme points.

If we want to prove that a certain set of points is $\varepsilon(\mathbf{A})$ then we have to show that 1) any point of $\mu(\mathbf{A})$ can be expressed as a convex linear combination of the elements of $\varepsilon(\mathbf{A})$, and 2) the elements of $\varepsilon(\mathbf{A})$ are extreme points. To prove the first condition an equality should be proved. The next theorem reduces this equality for an inequality. If ε is a set of vectors, ε^0 denotes the set of vectors obtained by changing the components of the vectors of ε for zero in all possible ways.

Theorem 2. *Suppose that* $\mathbf{A}$ *is hereditary and a set* $\varepsilon=\{\underline{e}_1, \dots, \underline{e}_m\}\subseteqq\mu(\mathbf{A})$ *is given. If for any* $\underline{f}\in\mu(\mathbf{A})$ *there are constants* $\lambda_1, \dots, \lambda_m\geqq 0$, $\sum_{i=1}^{m}\lambda_i\leqq 1$ *satisfying*

$$\underline{f}\leqq\sum_{i=1}^{m}\lambda_i\underline{e}_i \tag{2}$$

then $\varepsilon^*(\mathbf{A})\subseteqq\varepsilon$. ∎

This claim is useful, but trivial. (If $\underline{g}\in\langle\mu(\mathbf{A})\rangle$ and $\underline{0}\leqq\underline{f}\leqq\underline{g}$ then $\underline{f}\in\langle\mu(\mathbf{A})\rangle$).

2.2. *Application of the duality theorem of linear programming.* Using the transposed forms $\underline{f}^T$ and $\underline{e}_i^T$ of the column vectors $\underline{f}$ and $\underline{e}_i$, resp., (2) can be written like

$$(\underline{e}_1^T \dots \underline{e}_m^T)\begin{pmatrix}\lambda_1\\ \lambda_2\\ \vdots\\ \lambda_m\end{pmatrix}\geqq\begin{pmatrix}f_0\\ f_1\\ \vdots\\ f_n\end{pmatrix} \tag{3}$$

where $(\underline{e}_1^T\dots\underline{e}_m^T)$ denotes the $(n+1)\times m$ matrix with columns $\underline{e}_1^T, \dots, \underline{e}_m^T$. Its constraints are

$$\lambda_i\geqq 0 \quad (1\leqq i\leqq m) \tag{4}$$

and

$$\sum_{i=1}^{m}\lambda_i\leqq 1.$$

Our aim is to find for $\underline{f}$ such λ_i's. This can be formulated in the way that

$$\min\sum_{i=1}^{m}\lambda_i \tag{5}$$

should be found under the conditions (3) and (4) and the solution (5) of this linear programming problem has to be $\leqq 1$. The dual of this problem is

$$\begin{pmatrix}\underline{e}_1\\ \underline{e}_2\\ \vdots\\ \underline{e}_m\end{pmatrix}\begin{pmatrix}y_0\\ y_1\\ \vdots\\ y_n\end{pmatrix}\leqq\begin{pmatrix}1\\ 1\\ \vdots\\ 1\end{pmatrix} \tag{6}$$

$$y_i\geqq 0 \quad (0\leqq i\leqq n) \tag{7}$$

$$\max\sum_{i=0}^{n}f_iy_i. \tag{8}$$

By the duality theorem of linear programming (8) is equal to (5). (5)$\leqq$1 iff (8)$\leqq$1. This latter inequality can easily be formulated as

$$\sum_{i=0}^{n}f_iy_i\leqq 1$$

for any y_i's satisfying (6) and (7). It is worthwhile formulating this statement as a theorem:

Theorem 3. *Suppose that* $\mathbf{A}$ *is hereditary, a set* $\varepsilon=\{\underline{e}_1, \dots, \underline{e}_m\}\subseteqq\mu(\mathbf{A})$ *is given and*

$$\sum_{i=0}^{n} f_i y_i \leqq 1$$

holds for any $y_0, \dots, y_n$ *satisfying* $y_i\geqq 0$ $(0\leqq i\leqq n)$ *and*

$$\begin{pmatrix}\underline{e}_1\\ \underline{e}_2\\ \vdots\\ \underline{e}_m\end{pmatrix}\begin{pmatrix}y_0\\ y_1\\ \vdots\\ y_n\end{pmatrix}\leqq\begin{pmatrix}1\\ 1\\ \vdots\\ 1\end{pmatrix}.$$

Then $\varepsilon^*(\mathbf{A})\subseteqq\varepsilon$. ▮

2.3. *Reduction to the circle.* Take a cyclic permutation $\mathscr{C}$ of the underlying set X and consider only such subsets of X whose elements are consecutive in $\mathscr{C}$. These sets are called *consecutive sets in* $\mathscr{C}$. If $\mathscr{F}$ is a family of subsets of X, then $\mathscr{F}(\mathscr{C})$ is defined by $\mathscr{F}(\mathscr{C})=\{F\colon F\in\mathscr{F},\ F$ is consecutive in $\mathscr{C}\}$. Similarly, let $\mathbf{A}(\mathscr{C})=\{\mathscr{F}(\mathscr{C})\colon \mathscr{F}\in\mathbf{A}\}$. It is well-known (see e.g. [7]) that for some classes $\mathbf{A}$ it is enough to determine $\max\{|\mathscr{F}|\colon \mathscr{F}\in\mathbf{A}(\mathscr{C})\}$ and $\max\{|\mathscr{F}|\colon \mathscr{F}\in\mathbf{A}\}$ can be obtained from it by a simple counting argument. Of course, this extremal problem for $\mathbf{A}(\mathscr{C})$ is easier than for $\mathbf{A}$. This method is sometimes called as the *permutation method.*

Before stating the result we have to introduce a notation. If $\underline{e}=(e_0, e_1, \dots, e_n)$ then let

$$T(\underline{e})=\left(e_0, e_1\binom{n}{1}\Big/n, e_2\binom{n}{2}\Big/n, \dots, e_{n-1}\binom{n}{n-1}\Big/n, e_n\right).$$

Theorem 4. *(Blowing up the circle.) If* $\underline{e}_1, \dots, \underline{e}_m$ *are the extreme points of* $\mu(\mathbf{A}(\mathscr{C}))$ *for any given cyclic permutation* $\mathscr{C}$ *then*

$$\mu(\mathbf{A})\subseteqq\langle\{T(\underline{e}_1), \dots, T(\underline{e}_m)\}\rangle.$$

Proof. Let $\mathscr{F}$ be an element of $\mathbf{A}$, with profile $(f_0, f_1, \dots, f_n)$. Define the weight-function

$$\underline{w}(F)=\Big(\underset{\widehat{0}}{0}, \underset{\widehat{1}}{0}, \dots, \underset{\widehat{|F|}}{\frac{1}{(n-1)!}}, \dots, \underset{\widehat{n}}{0}\Big)\quad (F\subset X).$$

Consider the sum $\sum\underline{w}(F)$ for all paits $(\mathscr{C}, F)$ where $\mathscr{C}$ is a cyclic permutation, $F\in\mathscr{F}$ and F is consecutive in $\mathscr{C}$.

For a fixed $\mathscr{C}$ we have

$$\sum_{F\in\mathscr{F}(\mathscr{C})}\underline{w}(F)=\frac{1}{(n-1)!}(\text{profile of } \mathscr{F}(\mathscr{C})).$$

Here the profile of $\mathscr{F}(\mathscr{C})$ is in $\mu(\mathbf{A}(\mathscr{C}))$, therefore it is a convex linear combination $\sum_{i=1}^{m}\lambda_i(\mathscr{C})\underline{e}_i$ of the extreme points $\underline{e}_1, \dots, \underline{e}_m$ of $\mu(\mathbf{A}(\mathscr{C}))$ $(\lambda_i(\mathscr{C})\geqq 0,\ \sum_{i=1}^{m}\lambda_i(\mathscr{C})=1)$.

Hence

$$\sum_{\mathscr{C},F}\underline{w}(F)=\sum_{\mathscr{C}}\sum_{F}\underline{w}(F)=\sum_{\mathscr{C}}\frac{1}{(n-1)!}\sum_{i=1}^{m}\lambda_i(\mathscr{C})\underline{e}_i=\sum_{i=1}^{m}\frac{1}{(n-1)!}\Big(\sum_{\mathscr{C}}\lambda_i(\mathscr{C})\Big)\underline{e}_i$$

follows where $\sum_{i=1}^{m}\frac{1}{(n-1)!}\sum_{\mathscr{C}}\lambda_i(\mathscr{C})=1$. We have proved that

(9) $\sum_{\mathscr{C},F}\underline{w}(F)$ *is a convex linear combination of* $\underline{e}_1, \dots, \underline{e}_m$.

On the other hand, summing in the other way around we obtain

$$\text{(10)}\qquad \sum_{\mathscr{C},F}\underline{w}(F)=\sum_{F}\sum_{\mathscr{C}}\underline{w}(F)=\sum_{F}{}^{*}\left(0, 0, \dots, \frac{|F|!(n-|F|)!}{(n-1)!}, \dots, 0\right)$$
$$=\left(f_0, f_1, f_2 n\Big/\binom{n}{2}, \dots, f_i n\Big/\binom{n}{i}, \dots, f_{n-1}, f_n\right);$$

where $\sum^*$ denotes that $(1, 0, \dots, 0)$ and $(0, 0, \dots, 0, 1)$ are taken for $F=\emptyset$ and $F=X$, resp., as the number of cyclic permutations in which F is consecutive is $|F|!(n-|F|)!$ for $0<|F|<n$ but it is $(n-1)!$ for $|F|=0, n$. It follows by (9) that (10) is a convex linear combination of $\underline{e}_1, \dots, \underline{e}_m$. This implies that $(f_0, f_1, \dots, f_n)$ is a convex linear combination of $T(\underline{e}_1), \dots, T(\underline{e}_m)$. ▮

This theorem is really useful if $T(\underline{e}_1), \dots, T(\underline{e}_m)\in\mu(\mathbf{A})$ holds. (This can easily be checked.) Then $\langle\{T(\underline{e}_1), \dots, T(\underline{e}_m)\}\rangle\subseteqq\mu(\mathbf{A})$ and $\mu(\mathbf{A})=\langle\{T(\underline{e}_1), \dots, T(\underline{e}_m)\}\rangle$ obviously follow. $T(\underline{e}_1), \dots, T(\underline{e}_m)$ are the extreme points of $\mathbf{A}$. Unfortunately, this is not true in general. An example will be given when $\langle\{T(\underline{e}_1), \dots, T(\underline{e}_m)\}\rangle$ is much larger than $\langle\mu(\mathbf{A})\rangle$.

3. k-Sperner-families

Let $\mathbf{S}_k$ denote the class of k-Sperner-families on an n-element set.

Theorem 5. *The extreme points of* $(\mathbf{S}_k)$ *are the vectors whose* ith *components are either* $\binom{n}{i}$ *or* 0 *but have at most* k *non-zero components.*

Proof. It is trivial that these vectors are in $\mu(\mathbf{S}_k)$. To the vector $\left(0, \dots, \binom{n}{i_1}, 0, \dots, 0, \binom{n}{i_l}, \dots, 0\right)$ $(l\leqq k)$ one can find a k-sperner-family $\mathscr{F}$ with this profile: take all $i_1, \dots, i_l$-element subsets of X.

Moreover, these points are extreme. Let $\mathscr{E}=\left(0, \dots, 0, \binom{n}{i_1}, 0, \dots, 0, \binom{n}{i_l}, 0, \dots, 0\right)$; $(l\leqq k)$. It is easy to check that no $\underline{u}\in\mathscr{E}$ is a convex linear combination of the other points of $\mu(\mathbf{S}_k)$.

On the other hand, we have to prove that any element of $\mu(\mathbf{S}_k)$ can be expressed as a convex linear combination of these vectors. Theorem 4 can be applied if we show that the extreme points of $\mu(\mathbf{S}_k(\mathscr{C}))$ are the vectors whose ith components are either n or 0 for $0<i<n$ and either 1 or 0 for $i=0, n$, but have at most k non-zero components. By Theorem 1 it is sufficient to prove that $\varepsilon^*(\mathbf{S}_k(\mathscr{C}))$ is the set of vec-

tors whose ith components are either n or 0 for $0<i<n$ and either 1 or 0 for $i=0, n$, but have exactly k non-zero components. To prove this we apply Theorem 3. The inequality

(11) $$\sum_{i=0}^{n} f_i y_i \leqq 1$$

has to be verified for any k-Sperner-family in $\mathscr{C}$ with profile $(f_0, \dots, f_n)$ and for any system of y's satisfying $y_i \geqq 0$ $(0 \leqq i \leqq n)$ and

(12) $$\sum_{j=1}^{k} \varrho(i_j) n y_{i_j} \leqq 1$$

for any choice $0 \leqq i_1 < \dots < i_k \leqq n$ where $\varrho(0)=\varrho(n)=1/n$ $\varrho(i)=1$ $(1 \leqq i \leqq n-1)$.

Let us first show that (11) holds for the following simple systems of values:

$$y_0 = \frac{1}{k}, \quad y_1 = \dots = y_{n-1} = \frac{1}{nk}, \quad y_n = \frac{1}{k},$$

$$y_0 = 1, \quad y_1 = \dots = y_n = 0$$

$$y_0 = 0, \dots, y_i = \frac{1}{n}, \dots, y_n = 0 \quad (1 \leqq i \leqq n-1)$$

and

$$y_0 = \dots = y_{n-1} = 0, \quad y_n = 1.$$

In other words we have to prove the inequalities

(13) $$\frac{f_0}{k} + \sum_{i=1}^{n-1} \frac{f_i}{nk} + \frac{f_n}{k} \leqq 1$$

(14) $$f_0 \leqq 1$$

(15) $$f_i \leqq n \quad (1 \leqq i \leqq n-1)$$

(16) $$f_n \leqq 1$$

for the profile $(f_0, \dots, f_n)$ of any k-Sperner-family. (14)—(16) are trivial. The real problem is (13). Suppose first that $f_0=f_n=0$ and consider a fixed $\mathscr{F}(\mathscr{C})$ with this profile. Any element of X can be the "starting" point of at most k members of $\mathscr{F}(\mathscr{C})$ because of the k-Sperner property. Thus $|\mathscr{F}(\mathscr{C})| = \sum_{i=1}^{n-1} f_i \leqq nk$. (13) follows. If exactly one of f_0 and f_n is 1 then the number of members $\mathscr{F}(\mathscr{C})$ "starting" with a fixed element is at most $k-1$. (13) follows like above. The case $f_0=f_n=1$ is analogous.

Let us prove now (11) under the general assumption (12). Consider a fixed system of y's and order $\varrho(i)y_i$: $\varrho(l_1)y_{l_1} \geqq \dots \geqq \varrho(l_{n+1})y_{l_{n+1}}$ where $l_1, \dots, l_{n+1}$ is a permutation of $0, 1, \dots, n$. It follows by (12) that $\sum_{j=1}^{k} \varrho(l_j) y_{l_j} \leqq 1/n$ If there is a strict inequality here, then multiply all the y's with a constant (>1) to achieve

(17) $$\sum_{j=1}^{k} \varrho(l_j) y_{l_j} = \frac{1}{n}.$$

It is easy to see that it is sufficient to prove (11) for such y's. (12) and (17) imply

$$\varrho(t) y_t \leqq \frac{1}{n} - \sum_{j=1}^{k-1} \varrho(l_j) y_{l_j} = \varrho(l_k) y_{l_k}$$

for any $t \neq l_1, \dots, l_{k-1}$. Hence we have

$$\sum_{i=0}^{n} f_i y_i \leqq \sum_{j=1}^{k-1} f_{l_j} y_{l_j} + \sum_{t \neq l_1, \dots, l_{k-1}} \frac{f_t}{\varrho(t)} \varrho(l_k) y_{l_k}$$

$$= \sum_{j=1}^{k-1} \frac{f_{l_j}}{\varrho(l_j)} \left(\varrho(l_j) y_{l_j} - \varrho(l_k) y_{l_k}\right) + \sum_{t=0}^{n} \frac{f_t}{\varrho(t)} \varrho(l_k) y_{l_k}.$$

For the latter row we obtain an upper estimate applying (13)—(16) and (17):

$$\leqq \sum_{j=1}^{k-1} n\left(\varrho(l_j) y_{l_j} - \varrho(l_k) y_{l_k}\right) + \varrho(l_k) y_{l_k} nk$$

$$= n\left(\frac{1}{n} - \varrho(l_k) y_{l_k}\right) - n(k-1)\varrho(l_k) y_{l_k} + nk\varrho(l_k) y_{l_k} = 1.$$

We have proved that (11) holds for y's satisfying $y_i \geqq 0$ $(0 \leqq i \leqq n)$ and (12). The application of Theorem 3 finishes the proof. ▮

The following theorem is an easy consequence of Theorem 5.

Theorem 5a. *The hyperplanes bordering* $\langle \mu(\mathbf{S}_k) \rangle$ *are*

$$f_i \geqq 0 \quad (0 \leqq i \leqq n)$$

$$f_i \Big/ \binom{n}{i} \leqq 1 \quad (0 \leqq i \leqq n)$$

$$\sum_{i=0}^{n} f_i \Big/ \binom{n}{i} \leqq k. \quad ▮$$

Theorem 5 makes it easy to maximize $|\mathscr{F}| = \sum_{i=0}^{n} f_i$ for families $\mathscr{F}$ belonging to $\mathbf{S}_k$. It is sufficient to look for this maximum among the extreme points of $\mu(\mathbf{S}_k)$.

Theorem (Erdős [3])

$$\max_{\mathscr{F} \in \mathbf{S}_k} |\mathscr{F}| = \sum_{i=\lfloor (n-k+1)/2 \rfloor}^{\lfloor (n+k-1)/2 \rfloor} \binom{n}{i}. \quad ▮$$

For $k=1$ this is the old Sperner theorem [8].

4. Intersecting families

A family $\mathscr{F}$ is called *t-intersecting* ($1 \leqq t \leqq n$) if $F_1, F_2 \in \mathscr{F}$ implies $|F_1 \cap F_2| \geqq t$. Let $\mathbf{I}_t$ denote the class of t-intersecting families on an n-element set. The 1-intersecting families are called simply *intersecting*. In case $t=1$, $\mathbf{I}$ is written rather than $\mathbf{I}_1$. It seems to be too hard to determine the extreme points of $\mu(\mathbf{I}_t)$. We are able to do this only for $t=1$. However, it can be done for $\mathbf{I}_t(\mathscr{C})$. Before formulating the result we prove some preliminary lemmas.

Lemma 1. *Suppose that $A_1, \dots, A_u$ are v-element consecutive sets along a cyclic permutation $\mathscr{C}$ of an n-element set such that $|A_i \cap A_j| \geqq t \geqq 1$ for any $1 \leqq i < j \leqq u$ where $t \leqq v \leqq 1/2(n+t-1)$. Then $u \leqq v-t+1$ holds.*

Proof. Let $A_1 = \{x_1, \dots, x_v\}$ and suppose that the elements are ordered in this way. Another A cannot meet A_1 in both ends by the conditions. Therefore the possible endpoints for A are $x_t, \dots, x_{v-1}$, while the possible starting points are $x_2, \dots, x_{v-t+1}$. However the set ending with $x_i (t \leqq i \leqq v-1)$ and the one starting with x_{i-t+2} meet in $t-1$ elements only. Hence at most one of them can be among the A's. Consequently there are at most $v-t$ such A's. ▮

Lemma 2. *If $A_1, \dots, A_u$ are v-element consecutive sets along a cyclic permutation of an n-element set then*

$$|\bigcup_{i=1}^{u} A_i| \geqq \min(n, u+v-1).$$

Proof. Suppose first that have is an A_i containing no starting point of another A. Then the number of starting points is u while the number of other points of A_i is $v-1$, that is, $|\bigcup_{i=1}^{u} A_i| \geqq u+v-1$. On the other hand, if any A_i contains the starting point of another one then the union of them is the whole underlying set X, that is, $|\bigcup_{i=1}^{u} A_i| = n$. ▮

Lemma 3. *Let $(f_0, \dots, f_n) \in \mu(\mathbf{I}_t(\mathscr{C}))$, $f_i \neq 0$ for some $i \left(t \leqq i \leqq \dfrac{n+t-1}{2}\right)$. Suppose that $t \leqq j \leqq n+t-1-i$ holds for some j. Then $f_j \leqq j+i-f_i-2(t-1)$ holds.*

Proof. Suppose that $\mathscr{F} \in \mathbf{I}_t(\mathscr{C})$ holds and its profile is $(f_0, \dots, f_n)$. Let $\mathscr{F}_i = \{F_1, \dots, F_{f_i}\}$. Consider the family $\mathscr{A} = \{A \colon |A| = n-j, |A \cap F_l| \geqq i-t+1$ for some $1 \leqq l \leqq i\}$. The starting points of the $(n-j)$-element consecutive sets satisfying $|A \cap F_l| \geqq i-t+1$ for a fixed l form a consecutive set of size $n-j-i+2t-1$. Applying Lemma 2 the total number of these starting points is at least $\min(n, n-j-i+2t-2+f_i)$. Therefore this is a lower bound for $|\mathscr{A}|$. $A \in \mathscr{A}$ implies that $|X-A| = j$ and $|(X-A) \cap F| \leqq t-1$. Hence we have at least $\min(n, n-j-i+2t-2+f_i)$ j-element consecutive subsets $X-A$ not belonging to $\mathscr{F}$. Therefore $f_j = |\mathscr{F}_j| \leqq \max(0, j+i-f_i-2(t-1))$. ▮

We remark that Lemma 1 implies $f_i \leqq i-t+1$ hence $j+i-f_i-2(t-1) > 0$.

Lemma 4. $(f_0, \dots, f_n) \in \mu(\mathbf{I}_t(\mathscr{C}))$ *iff the following conditions are fulfille.*

(18) $\qquad f_i = 0 \quad (0 \leqq i < t),$

(19) $\qquad f_i \leqq i-t+1 \quad (t \leqq i \leqq (n+t-1)/2),$

(20) $\qquad f_j \leqq \min\{j+i-f_i-2(t-1)\} ((n+t-1)/2 < j \leqq n)$

where the minimum is taken on all i satisfying

(21) $\qquad t \leqq i \leqq n+t-1-j, \; f_i \neq 0.$

If this set is empty then (20) *has the form $f_j \leqq n$ $(j<n)$, $f_n \leqq 1$.*

Proof. (18) trivially follows from $(f_0, \dots, f_n) \in \mu(\mathbf{I}_t(\mathscr{C}))$ by the definitions. (19) and (20) are consequenses of Lemmas 1 and 3, respectively.

Conversely we have to prove that if (18)—(20) hold than there is an $\mathscr{F} \in \mathbf{I}_t(\mathscr{C})$ with profile $(f_0, \dots, f_n)$. This will be done by a construction. Let $x_1, \dots, x_n$ be the elements of X according their order in $\mathscr{C}$. For $t \leqq i \leqq (n+t-1)/2$, choose the consecutive sets with endpoints $x_i, x_{i-1}, \dots, x_{i-f_i+1}$. On the other hand, if $(n+t-1)/2 < i$, take the sets with endpoints x_t; $x_{t+1}, \dots, x_{t+f_i-1}$. This family $\mathscr{F}$ is trivially t-intersecting. ▮

So we obtained a purely algebraic characterization of the polytope $\langle \mu(\mathbf{I}_t(\mathscr{C})) \rangle$. Now the description of its essential vertices (Lemma 5) requires only linear algebraic technique, so the proof of it will be sketched only.

Lemma 5. $\varepsilon^*(\mathbf{I}_t(\mathscr{C}))$ *consists of the following vectors*

(22)

$$\begin{array}{ccccccccccc}
(0, \dots, & 0, & k-t+1, & k-t+2, \dots, & n-k, & n, \dots, & n, & 1) & \\
\widehat{n} & \widehat{k-1} & \widehat{k} & \widehat{k+1} & \widehat{n+t-1-k} & \widehat{n+t-k} & \widehat{n-1} & \widehat{n} & \left(t \leqq k \leqq \dfrac{n+t-1}{2}\right)
\end{array}$$

$$\begin{array}{ccccc}
(0, & \dots, & 0, & n, \dots, n, & 1) \\
 & & \widehat{\tfrac{n+t}{2}} & & \widehat{n} \qquad (n+t \;\; \textit{is even}).
\end{array}$$

Proof. (Sketch). It is clear that $(22) \subseteqq \langle \mu(\mathbf{I}_t(\mathscr{C})) \rangle$ and they are convex linearly independent.

If $f \in \langle \mu(\mathbf{I}_t(\mathscr{C})) \rangle$ a vertex then it can be obtained as an intersection of $(n+1)$ hyperplanes of the form (18)—(21). It is easy to check that if $f \in \langle \mu(\mathbf{I}_t(\mathscr{C})) \rangle$ and f satisfies $(n+1)$ inequalities of form (18)—(21) by equality then f can be obtained from an element of (22) changing some components for zero. So (22) are the essential vertices of $\langle \mu(\mathbf{I}_t(\mathscr{C})) \rangle$. ▮

If $t=1$ we may apply Theorem 4, Lemma 6 and Theorem 1 to determine all the extreme points e of $\mu(\mathbf{I}(\mathscr{C}))$. The vectors $T(\underline{e})$ are

(23)

$$\Big(\underset{\widehat{0}}{0},\dots,0,\underset{\widehat{k}}{\binom{n-1}{k-1}},\underset{\widehat{k+1}}{\binom{n-1}{k}},\dots,\underset{\widehat{n-k}}{\binom{n-1}{n-k-1}},\underset{\widehat{n-k+1}}{\binom{n}{n-k+1}},\dots,\binom{n}{n-1},\underset{\widehat{n}}{1}\Big)\qquad\left(1\leqq k\leqq\frac{n}{2}\right)$$

$$\left(\underset{\widehat{0}}{0},\dots,\;,0,\underset{\widehat{\frac{n+1}{2}}}{\binom{n}{\frac{n+1}{2}}},\dots,\binom{n}{n-1},\;\underset{\widehat{n}}{1}\right)\quad(n\ \text{ is odd})$$

and the vectors obtained by substituting 0's into some components. The vectors listed in (23) are in $\mu(\mathbf{I})$ as the following construction shows. Fix an element x of the underlying set X and take all the k-element, $k+1$-element, ..., $(n-k)$-element subsets containing x and take all $(n-k+1)$-element, ..., n-element sets. It is easy to see that this is an intersecting family and its profile is the desired vector. The same construction works for the vectors with the zeros. This proves the following.

Theorem 6. *$\varepsilon^*(\mathbf{I})$ consists of the vectors listed under* (23). ∎

The number of extreme points is exponentially large. However, if $\sum_{i=0}^{n} C_i f_i$ should be maximized, where $C_i\geqq 0$ then it is sufficient to consider $\varepsilon^*(\mathbf{I})$. The size of this set is linear. The most known consequence of the above theorem is the

Erdős—Ko—Rado theorem [4]. *If $\mathscr{F}$ is an intersecting family of k-element subsets of an n-element set and $k\leqq n/2$ then* $\max|\mathscr{F}|=\binom{n-1}{k-1}$. ∎

This follows from Theorem 6 since no extreme point has a larger kth component.

To determine $\max|\mathscr{F}|$ over any intersecting family $\mathscr{F}\subset 2^n$ is trivial. However it can also be deduced from Theorem 6. $|\mathscr{F}|=\sum_{i=0}^{n} f_i$ implies that we have to consider the sum of the components in the extreme points. It is easy to see that $f_i+f_{n-i}=$ $=\binom{n}{n-i}$ for any extreme point and $0\leqq i\leqq(n-1)/2$. Moreover, $f_{n/2}=\frac{1}{2}\binom{n}{n/2}$ holds. Hence $\sum_{i=0}^{n} f_i=2^{n-1}$. In the same way, it is easy to deduce $\max|\mathscr{F}|$ for intersecting families with any size constraint. $\max\sum_{i=0}^{n} i\cdot f_i$ can also be determined. For a further application see [2].

If we try to combine Lemma 5 and Theorem 4 for t-intersecting families, then the vectors $T(e)$ will not belong to $\mu(\mathbf{I}_t)$, therefore they are not extreme points, either. To determine the extreme points of $\mu(\mathbf{I}_t)$ seems to be very hard. It would imply the solution of many open problems. Such an open problem, raised by Erdős, Ko and Rado, is to maximize the size of a 2-intersecting family of $2n$-element subsets of a $4n$-element set [4]. (Lemma 5 answers the same question for the circle.) Let us note that one extreme point of $\mu(\mathbf{I}_t)$ is known, the one maximizing $|\mathscr{F}|=\sum_{i=0}^{n} f_i$ [6].

Finally we give a variant of Theorem 6. It can be proved by the duality theorem.

Theorem 6a. *If $(f_0,\dots,f_n)\in\mu(\mathbf{I})$ and $y_0,y_1,\dots,y_n\geqq 0$ satisfy the inequalities*

$$\binom{n-1}{k-1}y_k+\binom{n-1}{k}y_{k-1}+\dots$$

$$\dots+\binom{n-1}{n-k-1}y_{n-k}+\binom{n}{n-k+1}y_{n-k+1}+\dots+\binom{n}{n}y_n\leqq 1\quad\left(1\leqq k\leqq\frac{n}{2}\right)$$

$$\binom{n}{\frac{n+1}{2}}y_{\frac{n+1}{2}}+\dots+\binom{n}{n}y_n\leqq 1\quad(\textit{if } n \textit{ is odd})$$

then

$$\sum_{i=0}^{n} f_i y_i\leqq 1.\quad ∎$$

5. More families without inclusion among them

Daykin, Frankl, Greene and Hilton [1] investigated the families with the following properties. Let $t\geqq 2$ be an integer and let $\mathscr{F}^i(1\leqq i\leqq t)$ be a family of distinct subsets of an n-element set X. The families are not necessarily disjoint but $A_i\in\mathscr{F}^i$, $A_j\in\mathscr{F}^j$, $i\neq j$, $A_i\neq A_j$ imply $A_i\not\subset A_j$. In notation: $(\mathscr{F}^1,\dots,\mathscr{F}^t)\in\mathbf{W}_t$. The profile of an element of $\mathbf{W}_t$ is $(f_0,\dots,f_n)$ where $f_i=\sum_{j=1}^{t}|\mathscr{F}_i^j|$. It can be considered as the profile of $\sum_{j=1}^{t}\mathscr{F}^j$ with multiplicities. The definitions and the results of Section 2 can be repeated for families with multiplicities. $\mathbf{W}_t$ is obviously hereditary, so it is enough to determine $\varepsilon^*(\mathbf{W}_t)$ instead of $\varepsilon(\mathbf{W}_t)$. Colour the sets occuring exactly ones or more times by green or red, resp. It is easy to see that a red set cannot be in inclusion with any other green or red set. Therefore a red set can be added to all $\mathscr{F}^j$ without violating the conditions. In this way we associated to any $(\mathscr{F}^1,\dots,\mathscr{F}^t)\in\mathbf{W}_t$ two families $\mathscr{R}$ and $\mathscr{G}$ where no member of $\mathscr{R}$ is in inclusion with any member of $\mathscr{R}\cup\mathscr{G}$ and the members of $\mathscr{G}$ have multiplicity 1 while the multiplicity of any member of $\mathscr{R}$ is between 1 and t. The set of such pairs $(\mathscr{R},\mathscr{G})$ is denoted by $\mathbf{B}_t$. It is easy to see that, conversely, the members of any $(\mathscr{R},\mathscr{G})\in\mathbf{B}_t$ can be distributed into sets $\mathscr{F}^1,\dots,\mathscr{F}^t$. (Put all green sets into $\mathscr{F}^1$, the copies of the red sets into different $\mathscr{F}$'s.) This shows $\mu(\mathbf{W}_t)=\mu(\mathbf{B}_t)$.

Theorem 7. $\varepsilon^*(\mathbf{W}_t)=\varepsilon^*(\mathbf{B}_t)$ $(t\geqq 2)$ *consists of the vectors*

$$\left(0, \ldots, 0,\ t\binom{n}{i},\ 0, \ldots, 0\right) \quad (0\leqq i\leqq n)$$

and additionally

$$\left(\binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n}\right) \quad \textit{if} \quad t<n+1.$$

The proof is based on the following lemmas.

Lemma 6. *If* $(f_0, \ldots, f_n)\in\mu(\mathbf{B}_t(\mathscr{C}))$ *then*

$$\sum_{j=1}^{t} f_{i_j} \leqq tn$$

for any distinct $i_1, \ldots, i_t$.

Proof. Let $(\mathscr{R}, \mathscr{G})\in\mathbf{B}_t(\mathscr{C})$ and let $(f_0, \ldots, f_n)$ be its profile. Denote by r_j and g_j the number of i_j-element red and green members in $\mathscr{R}\cup\mathscr{G}$ resp. Hence

5((24) $$f_{i_j}\leqq tr_j+g_j$$

holds. The i_j-element green members and all the red ones in $\mathscr{R}\cup\mathscr{G}$ form a Sperner-family, therefore

$$g_j+\sum_{k=1}^{t} r_k \leqq n \quad (1\leqq j\leqq t)$$

follows. Summing these inequalities we obtain

$$\sum_{j=1}^{t}(g_j+tr_j)\leqq tn.$$

Hence (24) implies the validity of the lemma. ▮

Lemma 7. *Suppose that* $c_0, \ldots, c_n$ *are non-negative reals. Then, under the conditions*

(25) $$z_i\leqq\frac{1}{t} \quad (0\leqq i\leqq n,\ t \text{ is an integer}),$$

(26) $$\sum_{i=0}^{n} z_i\leqq 1,$$

$\max\sum_{i=0}^{n} c_iz_i$ *is attained for*

(27) $$z_0=\ldots=z_n=\frac{1}{t} \quad \textit{if} \quad n+1\leqq t,$$
$$(z_{i_1}=\ldots=z_{i_t}=\frac{1}{t}, \quad z_j=0 \quad (j\neq i_k) \quad \textit{if} \quad n+1>t.$$

Proof. It is trivial. ▮

Lemma 8. *Suppose that* $y_0, \ldots, y_n\geqq 0$ *satisfy the following inequalities:*

(28) 56 $$y_0\leqq\frac{1}{t}, \quad y_i\leqq\frac{1}{tn} \quad (1\leqq i<n), \quad y_n\leqq\frac{1}{t},$$

(29) 57 $$y_0+n\sum_{i=1}^{n-1} y_i+y_n\leqq 1.$$

Then $(f_0, \ldots, f_n)\in\mu(\mathbf{B}_t(\mathscr{C}))$ *implies*

(30) 58 $$\sum_{i=0}^{n} f_iy_i\leqq 1.$$

Proof. If $f_0\neq 0$ then the empty set is either a red or a green member of $\mathscr{R}\cup\mathscr{G}$. If $\emptyset\in\mathscr{R}$ then there is no other member: $f_i=0$ $(1\leqq i\leqq n)$. $f_0\leqq t$ and $y_0\leqq 1/t$ imply the statement. If $\emptyset\in\mathscr{G}$ then $\mathscr{R}$ is empty, therefore $f_0\leqq 1$, $f_i\leqq n$ $(1\leqq i<n)$, $f_n\leqq 1$. (29) implies (30). If $f_n\neq 0$, the situation is analogous. We may suppose that $f_0=f_n=0$.

Introduce the notations $z_i=ny_i$, $c_i=f_i/n$ $(1\leqq i<n)$. (28), (29) and $\sum_{i=0}^{n} f_iy_i$ give rise to $z_i\leqq 1/t(1\leqq i<n)$, $\sum_{i=1}^{n-1} z_i\leqq 1$ and $\sum_{i=1}^{n-1} c_iz_i$. We may apply Lemma 7:

$$\sum_{i=1}^{n-1} f_iy_i=\sum_{i=1}^{n-1} c_iz_i\leqq\begin{cases}\dfrac{1}{t}\displaystyle\sum_{i=1}^{n-1} c_i=\dfrac{1}{nt}\displaystyle\sum_{i=1}^{n-1} f_i & \text{if} \quad n+1\leqq t,\\[2ex] \dfrac{1}{t}\displaystyle\sum_{j=1}^{t} c_{i_j}=\dfrac{1}{nt}\displaystyle\sum_{j=1}^{t} f_{i_j} & \text{if} \quad n+1>t.\end{cases}$$

This is at most 1, in the first case trivially, in the second case by Lemma 6. (30) is proved. ▮

Proof of Theorem 7. The vectors $(t, 0, \ldots, 0), (0, \ldots, 0, tn, 0, \ldots, 0), (0, \ldots, 0, t)$ and $(1, n, \ldots, n, 1)$ are obviously in $\mu(\mathbf{B}_t(\mathscr{C}))$. Consequently, Lemma 8 and Theorem 3 imply that there vectors are the only candidates to be in $\varepsilon^*(\mathbf{B}_t(\mathscr{C}))$. Hence Theorem 1 gives the candidates for $\varepsilon(\mathbf{B}_t(\mathscr{C}))$.

If $t\geqq n+1$ then $(1, n, \ldots, n, 1)=t^{-1}(t, 0, \ldots, 0)+\sum t^{-1}(0, \ldots, 0, tn, 0, \ldots, 0)+$ $+t^{-1}(0, \ldots, 0, t)+(1-(n+1)t^{-1})\ (0, \ldots, 0)$ shows that $(1, n, \ldots, n, 1)$ is a convex linear combination of the other ones. The extreme points of $\mu(\mathbf{B}_t(\mathscr{C}))$ are $(0, \ldots, 0)$, $(t, 0, \ldots, 0)$, $(0, \ldots, 0, tn, 0, \ldots, 0)$ and $(0, \ldots, 0, t)$.

Suppose now that $t<n+1$. The set of possible extreme points of $\mu(\mathbf{B}_t(\mathscr{C}))$ is completed with $(1, n, \ldots, n, 1)$ and with the vectors obtained by writing zeros in the place of some components of $(1, n, \ldots, n, 1)$. However, if the number of non-zero components is $\leqq t$ then it is a convex linear combination of $(0, \ldots, 0)$, $(t, 0, \ldots, 0)$, $(0, \ldots, 0, tn, 0, \ldots, 0)$ and $(0, \ldots, 0, t)$. It is easy to see that the remaining ones are all extreme points of $\mu(\mathbf{B}_t(\mathscr{C}))$. Applying Theorem 4 the obtained vectors are all element of $\mu(\mathbf{B}_t)$. Moreover they are all extreme points. This proves the theorem. ▮

Theorem 7a. *The hyperplanes bordering* $\langle \mu(\mathbf{B}_t)\rangle$ *are*

$$f_i \geqq 0 \quad (0 \leqq i \leqq n)$$

and

59

$$\sum_{i=0}^{n} \frac{f_i}{\binom{n}{i} t} \leqq 1 \quad \textit{if} \quad t \geqq n+1, \tag{31}$$

60

$$\sum_{j=1}^{t} \frac{f_{i_j}}{\binom{n}{i_j} t} \leqq 1 \quad (0 \leqq i_1 < i_2 \ldots < i_t \leqq n) \quad \textit{if} \quad t < n+1. \tag{32}$$

Proof. Theorem 7 implies that for any $(f_0, \ldots, f_n) \in \mu(\mathbf{B}_t)$ there are $\lambda_0, \ldots, \lambda_n, \lambda_{n+1} \geqq 0$ satisfying $\sum_{i=0}^{n+1} \lambda_i \leqq 1$ and

$$\begin{pmatrix} f_0 \\ \vdots \\ \vdots \\ f_n \end{pmatrix} \leqq \lambda_0 \begin{pmatrix} t\binom{n}{0} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \lambda_1 \begin{pmatrix} 0 \\ t\binom{n}{1} \\ \vdots \\ 0 \end{pmatrix} + \ldots + \lambda_n \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ t\binom{n}{n} \end{pmatrix} + \lambda_{n+1} \begin{pmatrix} \binom{n}{0} \\ \binom{n}{1} \\ \vdots \\ \binom{n}{n} \end{pmatrix}$$

where $\lambda_{n+1}=0$ in the case $t \geqq n+1$. This can be considered as a linear programming problem with the result $\min \sum_{i=0}^{n+1} \lambda_i \leqq 1$. The dual problem maximizes $\sum_{i=0}^{n} f_i y_i$ under

61

$$y_i \leqq \frac{1}{t\binom{n}{i}} \quad (0 \leqq i \leqq n) \tag{33}$$

62

$$\sum_{i=0}^{n} \binom{n}{i} y_i \leqq 1 \quad \text{if} \quad t < n+1, \tag{34}$$

that is, $\sum_{i=0}^{n} f_i y_i \leqq 1$ holds under the conditions (33) an (34). Let us choose $y_i = \left(t\binom{n}{i}\right)^{-1}$ $(0 \leqq i \leqq n)$ if $t \geqq n+1$. $\sum_{i=0}^{n} f_i y_i \leqq 1$ becomes (31). Suppose now $t<n+1$ and choose $y_{i_1} = \ldots = y_{i_t} = \left(t\binom{n}{i}\right)^{-1}$ for some $0 \leqq i_1 < i_2 < \ldots < i_t \leqq n$. (33) and (34) are statisfied. This implies (32). Applying Lemma 8 with $z_i = y_i \binom{n}{i}$ and $c_i = f_i \binom{n}{i}^{-1}$ we obtain that if $\sum_{i=0}^{n} f_i y_i \leqq 1$ holds for the above special values of y's (that is, if (31) and (32) holds) then it holds for any system of non-negative y's statifying (33) and (34). The hyperplanes $\sum f_i y_i \leqq 1$ different from (31) and (32) are superfluous. ▮

Theorem 7 easily implies the first part of the theorem of [1]

$$\sum_{i=0}^{n} f_i \leqq \max\left(t \binom{n}{\left\lfloor \frac{n}{2} \right\rfloor}, 2^n \right) \quad \text{for} \quad (f_0, \ldots, f_n) \in \mathbf{B}_t. \tag{35}$$

The same theorem allows us to maximize $\sum_{i=0}^{n} f_i \binom{n}{i}^{-1}$ for $(f_0, \ldots, f_n) \in \mathbf{B}_t$:

$$\sum_{i=0}^{n} \frac{f_i}{\binom{n}{i}} \leqq \max(t, n+1). \tag{36}$$

This is the third part of the result in [1]. It is somewhat disturbing that (36) does not imply (35). The reason is that $\langle \mu(\mathbf{B}_t)\rangle$ cannot be well characterized by an arbitrarily chosen hyperplane.

To obtain the second part of the theorem of [1] the red and the green members of $(\mathcal{R}, \mathcal{G}) \in \mathbf{B}_t$ should be separated in the profile. The *colour profile* $(r_0, \ldots, r_n, g_0, \ldots, g_n)$ of $(\mathcal{R}, \mathcal{G})$ is defined by $r_i = |\mathcal{R}_i|$, $g_i = |\mathcal{G}_i|$ $(0 \leqq i \leqq n)$. $\chi(\mathbf{B}_t)$ denotes the set of colour profiles of all members of $\mathbf{B}_t$. The proof of the next theorem is left to the reader.

Theorem 8. *The essential extreme points of* $\chi(\mathbf{B}_t)$ *are*

$$\Bigl(\underset{\overset{\frown}{0}}{0}, \ldots, \underset{\overset{\frown}{i}}{\binom{n}{i}}, \ldots, \underset{\overset{\frown}{n}}{0}, \underset{\overset{\frown}{n+1}}{0}, \quad \ldots, \quad \underset{\overset{\frown}{2n+1}}{0}\Bigr) \quad (0 \leqq i \leqq n)$$

$$\Bigl(0, \ldots, \qquad 0, \binom{n}{0}, \binom{n}{1}, \ldots, \binom{n}{n}\Bigr). \quad ▮$$

In other words, for any profile $(r_0, \ldots, r_n, g_0, \ldots, g_n)$ there are $\lambda_0, \ldots, \lambda_n, \lambda_{n+1} \geqq 0$ satisfying $\sum_{i=0}^{n+1} \lambda_i \leqq 1$

63

$$\begin{pmatrix} r_0 \\ \vdots \\ \vdots \\ r_n \\ g_0 \\ \vdots \\ g_n \end{pmatrix} \leqq \lambda_0 \begin{pmatrix} \binom{n}{0} \\ 0 \\ \vdots \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} + \lambda_1 \begin{pmatrix} 0 \\ \binom{n}{1} \\ \vdots \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} + \ldots + \lambda_n \begin{pmatrix} 0 \\ \vdots \\ \binom{n}{n} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} + \lambda_{n+1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \binom{n}{0} \\ \binom{n}{1} \\ \vdots \\ \binom{n}{n} \end{pmatrix}. \tag{37}$$

Summing up the inequalities $r_i \leqq \lambda_i \binom{n}{i}$ $(0 \leqq i \leqq n)$; we obtain

$$r = \sum_{i=0}^{n} r_i \leqq \sum_{i=0}^{n} \lambda_i \binom{n}{i} \leqq \binom{n}{\left\lfloor \frac{n}{2} \right\rfloor} \sum_{i=0}^{n} \lambda_i .$$

Hence

$$\lambda_{n+1} \leqq 1 - \sum_{i=0}^{n} \lambda_i \leqq 1 - \frac{r}{\binom{n}{\left\lfloor \frac{n}{2} \right\rfloor}}$$

follows. Substituting this into (37), it is easy to see that

$$\sum_{i=0}^{n} g_i \leqq \lambda_{n+1} \sum_{i=0}^{n} \binom{n}{i} \leqq \left(1 - \frac{r}{\binom{n}{\left\lfloor \frac{n}{2} \right\rfloor}} \right) 2^n .$$

As the number of red sets with multiplicity is rt, the middle part of the theorem of [1] is proved: *If the number of* sets occuring at least twice in an $(\mathscr{F}^1, \ldots, \mathscr{F}^t) \in \mathbf{W}_t$ *is* r*, then*

$$\sum_{i=0}^{n} f_i \leqq rt + \left(1 - \frac{r}{\binom{n}{\left\lfloor \frac{n}{2} \right\rfloor}} \right) 2^n .$$

We are indebted to *Z.* Füredi for his many suggestions concerning the manuscript.

References

[1] D. E. Daykin, P. Frankl, C. Greene, and A. J. W. Hilton, A generalization of Sperner's theorem, *J. of the Austral. Math. Soc. A*, **31** (1981), 481—485.

[2] K. Engel, *submitted to Combinatorica.*

[3] P. Erdős, On a lemma of Littlewood and Offord, *Bull. of the Amer. Math. Soc.*, **51** (1945), 898—902.

[4] P. Erdős, Chao Ko, and R. Rado, Intersection theorems for systems of finite sets, *Quart. J. Math. Oxford (2)*, **12** (1961), 313—318.

[5] Péter L. Erdős, P. Frankl, and G. O. H. Katona, Intersecting Sperner families and their convex hulls, *Combinatorica*, **4** (1984), 21—34.

[6] G. O. H. Katona, Intersection theorems for system of finite sets, *Acta Math. Acad. Sci. Hungar.*, **15** (1964), 329—337.

[7] G. O. H. Katona, A simple proof of the Erdős—Chao Ko—Rado theorem, *J. Combinatorial Th. B*, **13** (1972), 183—184.

[8] E. Sperner, Ein Satz über Untermenge einer endlichen Menge, *Mat. Z.*, **27** (1928), 544—548.

Péter L. Erdős, G. O. H. Katona

Mathematical Institute of the
Hungarian Academy of Sciences
Budapest, P.O.B. 127
1364, Hungary

P. Frankl

C.N.R.S.
54 Bld. Raspail
75720 Parix, Cedex 06
France

Instructions to Authors

1. *Submission.* Manuscripts should be submitted (in triplicate) to the executive editor or to R. L. Graham (see inside front cover). Authors should retain a copy of everything submitted, since submitted material will usually not be returned along with the referees' reports and the galley proofs.

   In order to facilitate refereeing, copies of those papers (whether by the author or someone else) which are essentially referred to and have not appeared *by that time,* or are not conveniently accessible, should be enclosed.
2. *Form of Manuscript.* As a general rule, the length of the paper should not exceed 20 typewritten pages. The manuscript should be double-spaced, with about 25 lines to a page, about 50 characters to a line. Authors are requested to attach a separate sheet containing the title of the article, their names and affiliations, and a mailing address to which all correspondence will be sent. In addition, it should also contain a proposed running head (abbreviated form of the title) of at most 35 characters, a list of keywords, and primary and secondary AMS(1980) subject classification codes. On another separate sheet, the mathematical symbols and special characters (Greek, script, etc.) used should be listed.

   In a paper divided into sections, it is desirable to number theorems, lemmas, definitions, corollaries, examples, etc. *consecutively* using double arabic numerals. (E.g., Section 3 may start with Definition 3.1, followed by Remark 3.2 and Theorem 3.3).

   Mark the end of proof by ∎. The same symbol, placed at the end of a statement, indicates the absence of proof.
3. *Abstract.* The manuscript should be accompanied by a brief abstract. Bear in mind that this might be the only part of the paper read by most of the readers.
4. *Style.* Even first-rate results in combinatorics and in the theory of computing can be presented in a way acessible to mathematicians working in other fields. Authors are requested to give clear motivation, background, exact references, and all definitions which are not generally known.
5. *Reprints.* 50 reprints are provided free of charge. Additional reprints may be ordered.

Some of the papers to appear in forthcoming issues

J. Bierbrauer, A. Brandis: On generalized Ramsey numbers for trees

K. Deimer: A new upper bound for the length of snakes

M. El-Zahar, N. Sauer: The chromatic number of the product of two 4-chromatic graphs is 4

H. de Fraysseix, P. Rosenstiehl: A characterization of planar graphs by Trémaux orders

A. Hajnal: The chromatic number of the product of two $\aleph_1$-chromatic graphs can be countable

P. L. Hammer, N. V. R. Mahadev, D. de Werra: The struction of graphs; Application to *CN*-free graphs

S. C. Locke: A generalization of Dirac's theorem

W. Mader: Degree and local connectivity in digraphs

B. L. Rothschild, E. G. Straus: On triangulations of the convex hull of n points

Mathematical Programming 65 (1994) 93–105

On weighted multiway cuts in trees

Péter L. Erdős*[,a], László A. Székely**[,b]

[a]*Centrum voor Wiskunde en Informatica, 1098 SJ Amsterdam, Netherlands*
Mathematical Institute of the Hungarian Academy of Sciences, H-1055 Budapest, Hungary
[b]*Department of Computer Science, Eötvös University, H-1088 Budapest, Hungary*
Department of Mathematics, University of New Mexico, Albuquerque, NM 87131, USA

Received 11 September 1991; revised manuscript received 1 April 1993

Abstract

A min–max theorem is developed for the multiway cut problem of edge-weighted trees. We present a polynomial time algorithm to construct an optimal dual solution, if edge weights come in unary representation. Applications to biology also require some more complex edge weights. We describe a dynamic programming type algorithm for this more general problem from biology and show that our min–max theorem does not apply to it.

AMS 1991 Subject Classifications: 05C05, 05C70, 90C27

Keywords: Multiway cut; Menger's theorem; Tree; Duality in linear programming; Dynamic programming

1. Introduction

Let $G=(V,E)$ be a simple graph, $C=\{1,2,\ldots,r\}$ be a set of colours. For $N\subseteq V(G)$, a map $\chi:N\rightarrow C$ is a *partial colouration.* We usually think of a given partial colouration. A map $\bar{\chi}:V(G)\rightarrow C$ is a *colouration* if $\chi(v)=\bar{\chi}(v)$ holds for all $v\in N$.

A *colour dependent weight function* assigns to every edge (p,q) and colours i,j a natural number $w(p,q;i,j)$, which tells the weight of the edge (p,q) in a colouration $\bar{\chi}$, in which $\bar{\chi}(p)=i$, $\bar{\chi}(q)=j$. We assume that $w(p,q;i,i)=0$ and $w(p,q;i,j)=w(q,p;j,i)$. We say that w is *colour independent*, if for any (p,q), $i_1\neq j_1$, $i_2\neq j_2$, we have $w(p,q;i_1,j_1)=w(p,q;i_2,j_2)$. We say that w is *edge independent*, if for any $(p_1,q_1)\in E$ and $(p_2,q_2)\in E$, and

*Corresponding author.

**Research of the author was supported by the A. v. Humboldt-Stiftung and the U.S. Office of Naval Research under the contract N-0014-91-J-1385.

SSDI 0025-5610(93)E0073-N

$i, j \in C$, we have $w(p_1, q_1; i, j) = w(p_2, q_2; i, j)$. (Hence, any edge independent weight function satisfies $w(p, q; i, j) = w(p, q; j, i)$.) We say that w is *constant*, if it is colour and edge independent.

An edge (p, q) is *colour-changing* in the colouration $\bar{\chi}$, if $\bar{\chi}(p) \neq \bar{\chi}(q)$. The *changing number* of the colouration $\bar{\chi}$ is the sum of weights of the colour-changing edges in $\bar{\chi}$, i.e.:

$$\mathbf{change}(G, \bar{\chi}) = \sum_{(p,q) \in E(G)} w(p, q; \bar{\chi}(p), \bar{\chi}(q)) .$$

A partial colouration χ defines a partition of N *by* $N_i = \{v \in N: \chi(v) = i\}$. A set of edges that separates every N_i from all the other N_j's is termed a *multiway cut* [1]. Observe that the set of colour-changing edges of a colouration $\bar{\chi}$ forms a multiway cut and every multiway cut is represented in this way.

The *length* of the pair (G, χ) is the minimum weight of a multiway cut, in formula:

$$l(G, \chi) = \min\{\mathbf{change}(G, \bar{\chi}) : \bar{\chi} \text{ colouration}\} .$$

An *optimal colouration* is a colouration $\bar{\chi}$ such that $\mathbf{change}(G, \bar{\chi}) = l(G, \chi)$.

The multiway cut problem for colour independent weight functions has been extensively studied in combinatorial optimization (e.g. [1–3].). As Dahlhaus et al. pointed out [3], this problem is NP-hard, even for $|N| = 3$, $|N_i| = 1$ and constant weight.

On the other hand, if we restrict ourselves to planar graphs, a fixed number of colours, and constant weight, then the problem becomes solvable in polynomial time [3]. A well-known specialization of the multiway cut problem, which is solvable in polynomial time, is $r = 2$, which is considered in the undirected edge version of Menger's theorem [8].

Although it is less known in the operations research community, some instances of the multiway cut problem have great importance in biomathematics. In fact, the notions of the changing number and the length came from genetics and we follow the terminology used there. For the case of constant weight function, Fitch [6] and Hartigan [7] developed a polynomial time algorithm to determine the length of a given tree. Sankoff and Cedergren [13], and Williamson and Fitch [12] studied edge independent weight functions and made polynomial time algorithms to find the length. Some explanation of the significance of the multiway cut problem in biology is given in [4, 5].

The goal of the present paper is to study the multiway cut problem. In Section 2 we give a new lower bound for the length of a multiway cut. Section 3 provides a dynamic programming type algorithm to find the length of a tree with an arbitrary weight function. Section 4 uses the algorithm of Section 3 to establish a min–max theorem for the multiway cut problem of trees, in the case of colour independent weight functions. All the results can be extended to any graph G, in which N intersects every cycle. Section 5 describes our results in terms of linear programming.

A preliminary version of the present paper has already appeared [5]. We are indebted to the anonymous referees for their helpful observations that we use in this presentation.

2. Lower bound for the weight of a multiway cut

Let G be a simple graph, $N \subseteq V(G)$ and $\chi: N \to C$ be a partial colouration. Let w be a colour dependent weight function.

Definition. An oriented path P in G starting at $s(P) \in N$ and terminating at $t(P) \in N$ is a *colour-changing path*, if $\chi(s(P)) \neq \chi(t(P))$ and P has no internal vertex in N. (From now on path means oriented path, unless we explicitly say the opposite.) Let us fix a family $\mathscr{P}$ of colour-changing paths and let $e = (p, q) \in E(G)$. Define

$$n_i(e, \mathscr{P}) = \#\{P \in \mathscr{P}\colon\ (p, q) \in P \text{ and } \chi(t(P)) = i\} .$$

The notation $(p, q) \in P$ means that P enters the edge (p, q) at p and leaves at q.

Definition. Let $\chi: N \to C$ be a partial colouration and $\bar{\chi}$ be a colouration on G. A family $\mathscr{P}$ of colour-changing paths is a *path packing*, if all pairs of colours $i \neq j$ and all edges (p, q) satisfy

$$n_i((p, q), \mathscr{P}) + n_j((q, p), \mathscr{P}) \leqslant w(p, q; j, i) .$$

The maximum cardinality of a path packing is denoted by $p(G, \chi)$.

Theorem 1. *For any graph G and partial colouration χ, we have*

$$l(G, \chi) \geqslant p(G, \chi) .$$

Proof. Let $\mathscr{P}$ be a path packing and $\bar{\chi}: V(G) \to C$ be an optimal colouration. Define a map $f: \mathscr{P} \to E(G)$ as follows: let $f(P) = e$ if e is the last colour-changing edge in P in $\bar{\chi}$. For any colour changing edge $e = (p, q)$, $\bar{\chi}(p) = j$ and $\bar{\chi}(q) = i$ $(i \neq j$ since e is colour changing$)$, we have

$$\#\{P \in \mathscr{P}\colon f(P) = e\} \leqslant n_i((p, q), \mathscr{P}) + n_j((q, p), \mathscr{P}) \leqslant w(p, q; j, i) .$$

Therefore,

$$|\mathscr{P}| \leqslant \textbf{change}(G, \bar{\chi}) = l(G, \chi) . \qquad \square$$

3. An algorithm to find optimal colourations

Now we focus on the multiway cut problem of trees. Let T be a tree and $\chi: N \to C$ be a partial colouration, and let $L(T)$ denote the set of leaves, i.e. vertices of degree 1. We assume $N = L(T)$. (It is obvious that the solution of the multiway cut problem of trees with $N = L(T)$ easily generalizes to the solution of the multiway cut problem of trees with arbitrary N.) Let w be a colour dependent weight function. In this section we give a polynomial time algorithm to determine all optimal colouration of T for the weight w.

Let us fix an arbitrary non-leaf vertex, the **root** of T. Let (u, v) be an edge and let v be closer to the **root** than u, then we say $v = \mathbf{Father}(u)$. (**Father(root)** is **NIL.**) We denote the set of all u for which $v = \mathbf{Father}(u)$ by $\mathbf{Son}(v)$.

Our colouring algorithm has two phases. Starting from the leaves and approaching the **root** we determine a *penalty function* of every vertex v recursively, and subsequently we determine a suitable colouration $\bar{\chi}$ starting from the *root* and spreading to the leaves.

Definition. The vector-valued *penalty function* is a map

$$\mathbf{pen}: V(T) \to (\mathbb{N} \cup \{\infty\})^r,$$

such that $\mathbf{pen}_i(v)$ means the length of the subtree separated by v from the **root**, if the colour of v has to be i.

Phase I. For every leaf $v \in L(T)$ let

$$\mathbf{pen}_i(v) = \begin{cases} 0 & \text{if } v \in N_i, \\ \infty & \text{otherwise}, \end{cases}$$

where in an actual computation ∞ may be substituted by a sufficiently large number. Take a vertex v, such that $\mathbf{pen}(v)$ is not computed yet for the vertex v, but $\mathbf{pen}(u)$ is already known for every vertex $u \in \mathbf{Son}(v)$. Then compute

$$\mathbf{pen}_i(v) = \sum_{u \in \mathbf{Son}(v)} \min_{j=1,\ldots,r} \{w(u, v; j, i) + \mathbf{pen}_j(u)\}.$$

Phase II. Now we determine an optimal colouration $\bar{\chi}$ of T. First, let $\bar{\chi}(\mathbf{root})$ be a colour i, which minimizes the value $\mathbf{pen}_i(\mathbf{root})$. Furthermore, for a vertex v for which $\bar{\chi}(v)$ is not settled yet, but $\bar{\chi}(\mathbf{Father}(v))$ is already determined, let $\bar{\chi}(v)$ be a colour i, which minimizes the expression

$$w(v, \mathbf{Father}(v); i, \bar{\chi}(\mathbf{Father}(v))) + \mathbf{pen}_i(v).$$

It is easy to see, that every leaf $v \in N_i$ satisfies $\bar{\chi}(v) = i = \chi(v)$, for $i = 1, \ldots, r$.

The correctness of this algorithm is almost self-explanatory. Assume the positive integer edge weights are given in unary representation. Then, the time complexity is $O(n \cdot r^2 \cdot (\max \textit{weight}))$, since at each step we calculate r^2 sums, take the minimum, and roughly $2n$ steps are necessary because T has n vertices and $n-1$ edges. You may change max *weight* for log(max *weight*), if the edge weights come in binary representation.

In the rest of this section we focus on colour independent weight functions, since we can develop a slightly more efficient version of this algorithm, which also can determine all optimal colourations. Biologists may need all optimal colourations; the saving in running time comes from avoiding the second minimization in Phase II. Also, case (A2) in the proof of Theorem 2 will need the modified algorithm. For the sake of simplicity, for the rest of this section the weight function is a map $w: E(T) \to \mathbb{N}$ for colour changing edges and the weight of any edge not changing colour is 0. We use the usual *Kronecker delta* notation.

Phase I′. For every leaf v, set

$$M_1(v) = M_2(v) = \{i\colon \mathbf{pen}_i(v) = 0\} .$$

If $\mathbf{pen}(v)$ is not computed yet for the vertex v but $\mathbf{pen}(u)$ is already known for every vertex $u \in \mathbf{Son}(v)$, then set

$$\mathbf{pen}_i(v) = \sum_{u \in \mathbf{Son}(v)} \min_{j=1,\, i.,\, r} \{(1-\delta_{ij})w(u, v) + \mathbf{pen}_j(u)\} .$$

Let $p(v) = \min_i \mathbf{pen}_i(v)$, and

$$M_1(v) = \{i \in \{1, \ldots, r\}\colon \mathbf{pen}_i(v) = p(v)\} ,$$

$$M_2(v) = \{i \in \{1, \ldots, r\}\colon \mathbf{pen}_i(v) < p(v) + w(v, \mathbf{Father}(v))\} .$$

It is obvious that $M_1(v) \subseteq M_2(v)$.

Phase II′. For $\bar{\chi}(\mathbf{root})$, take an arbitrary element of $M_1(\mathbf{root})$. If $\bar{\chi}(v)$ is not settled yet for a vertex v, but $\bar{\chi}(\mathbf{Father}(v))$ is already determined, take

$$\bar{\chi}(v) = \begin{cases} \bar{\chi}(\mathbf{Father}(v)) & \text{if } \bar{\chi}(\mathbf{Father}(v)) \in M_2(v) , \\ \text{an arbitrary element of } M_1(v) & \text{otherwise} . \end{cases}$$

It is easy to see, that every vertex $v \in N_i$ satisfies $\bar{\chi}(v) = i = \chi(v)$, for $i = 1, \ldots, r$. This algorithm is obviously correct and permitting some extra freedom at certain steps, any optimal colouration can be obtained by the modified algorithm. For this purpose we introduce a third set of colours at Phase I′:

$$M_3(v) = \{i \in \{1, \ldots, r\}\colon \mathbf{pen}_i(v) = p(v) + w(v, \mathbf{Father}(v))\} .$$

If in Phase II′ we also allow to give the colour of $\bar{\chi}(\mathbf{Father}(v))$ to v, if $\bar{\chi}(\mathbf{Father}(v)) \in M_3(v)$, then the algorithm still yields an optimal colouration. Moreover, one can prove that running this algorithm in all possible ways yields all optimal colourations. (We leave the proof to the reader.) The complexity of this revised algorithm is better by a constant multiplicative factor than that of the original, but to get every optimal colouration may take exponential time, since M.A. Steel exhibited trees with exponentially many optimal colourations [11].

4. A min–max theorem

In this section we assume that the weight function is *colour-independent* and we prove that the lower bound of Theorem 1 is tight for leaf-coloured trees, and then even for a larger class of graphs.

Theorem 2. *Let T be an arbitrary tree with colour-independent weight function $w:E(T)\to\mathbb{N}$ and with leaf-colouration $\chi:L(T)\to C$. Then*

$$l(T,\chi)=p(T,\chi)\ .$$

We already know from Theorem 1 that the LHS is greater or equal than the RHS. We have to prove the other inequality. For this end we construct the desired optimal path packing in a recursive manner. At first, we explicitly construct optimal path packings for stars, i.e. for trees with 1 branching vertex. Then, for a tree T with at least 2 branching vertices and with

$$W(T)=\sum_{f\in E(T)} w(f)$$

sum of weights, we define a 'smaller' tree T' for which we can trace back the problem of the construction of an optimal path packing, such that we can 'lift up' the path packing from T' to T to get the solution. We may have at most $W(T)$ 'lift up' steps. Here we give the details.

For convenience, we want to use the functions **Son** and **Father**, therefore we fix, as in Section 3, a **root** of T. In the complexity issues we assume that our tree is represented by the vertices v and the sets **Son**(v) and **Father**(v), furthermore every element of **Son**(v) and **Father**(v) (which represents edges) also contains the weight of the edge. The paths under construction will be represented as double-linked lists, therefore, due to Theorem 1, the space complexity of the representation is $O(l(T,\chi)\cdot n)$.

Definition. We say that a vertex v is *of order* 1 if every element of **Son**(v) is a leaf.

Notice that every tree with at least 2 branching vertices has a non-root vertex of order 1. Before starting the main body of the proof we need the following lemma.

Lemma 1. *One can assume that no vertex of order* 1 *has two sons with the same colour.*

Let v be a vertex of order 1, such that **Son**(v) contains at least 2 leaves with identical colour. Let $\Sigma(T)$ denote the tree obtained from T by identification of the elements of **Son**(v) with identical colour and adding up their edge weights, respectively. Now one can easily construct an optimal path packing for T from an optimal path packing of $\Sigma(T)$. Anyhow, we give a formal proof, otherwise, the base case of our recursive algorithm would not be complete.

Proof. Define the tree $\Sigma(T)$ formally as follows: let the tree T' be a star with midpoint v and with leaves $\{l_i\colon \exists u\in\mathbf{Son}(v) \text{ with } \chi(u)=i\}$ and let $\Sigma(T)$ be the tree made of the trees $T\setminus\mathbf{Son}(v)$ and T' by identification of their common v. The leaf-colouration and weight function of $\Sigma(T)$ are as follows:

$$\chi'(u)=\begin{cases}\chi(u) & \text{if } u\in L\setminus\mathbf{Son}(v)\ ,\\ i & \text{if } u=l_i\ ,\end{cases}$$

$$w'(f)=\begin{cases}\sum\limits_{\substack{u\in\mathbf{Son}(v)\\ \chi(u)=i}} w((u,v)) & \text{if } f=(l_i,v)\,,\\ w(f) & \text{otherwise}\,.\end{cases}$$

Notice that $l(\Sigma(T),\chi')=l(T,\chi)$.

Claim. *If* $l(\Sigma(T),\chi')=p(\Sigma(T),\chi')$ *then* $l(T,\chi)=p(T,\chi)$.

Proof. Let $\mathbf{Son}(v)$ contain d different colours. We apply induction on $|\mathbf{Son}(v)|$.

Base case: if $|\mathbf{Son}(v)|=d$, then $\Sigma(T)=T$, $\chi=\chi'$, and we have nothing to prove.

Inductive step: Suppose that we know Lemma 1 for all $|\mathbf{Son}(v)|<k$. Assume now $|\mathbf{Son}(v)|=k$ and for some fixed $z_1, z_2\in\mathbf{Son}(v)$, let $\chi(z_1)=\chi(z_2)$. Join z_1 and z_2 into z. In the new tree T^* obtained by identification, define the leaf colouration and the weight function as follows:

$$\chi^*(u)=\begin{cases}\chi(u) & \text{if } u\neq z_1, z_2\,,\\ \chi(z_1) & \text{if } u=z\,,\end{cases}$$

$$w^*(f)=\begin{cases}w(f) & \text{if } f\neq(v,z_i)\,,\\ w(v,z_1)+w(v,z_2) & \text{if } f=(v,z)\,.\end{cases}$$

Now we have $\Sigma(T)=\Sigma(T^*)$, therefore $l(\Sigma(T))=l(\Sigma(T^*))$. By the hypothesis there exists a path packing $\mathscr{P}^*$ in the tree T^* satisfying $|\mathscr{P}^*|=l(T^*)$. It is easy to divide the paths of $\mathscr{P}^*$ adjacent to vertex z into two groups, such that the members of one group are adjacent to z_1 and the members of the other are adjacent to z_2 and both groups obey the weight restriction on the edge adjacent to z_i. In this way we obtain a path packing of $l(T)$ members in T. This proves the Claim as well as Lemma 1. □

The time complexity of this algorithm is $O(\Sigma_{u\in\mathbf{Son}(v)}\, w(u,v))$ so the time complexity of all applications of Lemma 1 altogether is $O(W(T))$.

We return to the main body of the proof; we assume that any two sons of an arbitrary vertex of order 1 have different colours. Our algorithm is given in a recursive form in the variables $b(T)$ and $W(T)$, where $b(T)$ is the number of branching (non-leaf) vertices of T.

Base case: let $b(T)=1$ and $W(T)$ be arbitrary. Then T is a star; let v denote the midpoint of it. Due to Lemma 1 we may assume that $|L(T)|=r$ (i.e. every colour occurs once). Assume that the edge (v,u) has maximum weight over all edges. Orient paths from u to every other leaf $z\in L(T)\setminus\{u\}$ with multiplicity $w(v,z)$. This path system is obviously a path packing and has $l(T)$ members. This case requires $O(W(T))$ steps.

Recursive step: For any tree T with at least 2 branching vertices we shall find 'smaller' tree T' with fewer branching vertices $(b(T')<b(T))$ or with smaller total weights

($b(T')=b(T)$ and $W(T')<W(T)$) such that an optimal path packing of T' can be lifted up to an optimal path packing of T. Define

$$s(v)=\max_{u\in\mathbf{Son}(v)} w(v,u)\ .$$

We distinguish two cases:

(A) There is a vertex v of order 1 such that $s(v)\neq w(v,\mathbf{Father}(v))$.

(B) $s(v)=w(v,\mathbf{Father}(v))$ for every vertex v of order 1.

Case (A). Let $\bar{\chi}$ be an optimal colouration of T such that v is the first branching vertex for which the colour sets M_i were determined. We have two subcases; in (A1) we have $s(v)>w(v,\mathbf{Father}(v))$, in (A2) we have $s(v)<w(v,\mathbf{Father}(v))$.

Case (A1). Let T'' be the tree with the same vertex set, edge set and leaf colouration as the tree T was, and let the new weight function $w':E(T)\to\mathbb{N}$ such that

$$w'(f)=\begin{cases} w(f)-1 & \text{if } f=(v,u) \text{ where } u\in\mathbf{Son}(v)\ ,\\ w(f) & \text{if otherwise}\ .\end{cases}$$

If $w'(f)=0$, then cancel this edge and its leaf endpoint from the tree T'' to obtain the tree T'. Due to our colouring algorithm, colouration $\bar{\chi}$ is also optimal for the tree T', therefore

$$l(T')+(|\mathbf{Son}(v)|-1)=l(T)\ .$$

The total weight of tree T' is less than of T. Assume now that we have an optimal path packing $\mathscr{P}'$ of $l(T',\chi)$ elements in T'. Denote by ΔT the star of $v\cup\mathbf{Son}(v)$ with weight function $w\equiv 1$ and with the original leaf colouration. Let $\Delta\mathscr{P}$ be optimal path packing in ΔT (use the base case). Now the path system $\mathscr{P}=\mathscr{P}'\cup\Delta\mathscr{P}$ is obviously optimal path packing in the tree T.

We can construct T' and the path packings $\Delta\mathscr{P}$ and $\mathscr{P}$ from the given tree T and path packing $\mathscr{P}'$ in $O(r\cdot\Sigma_{u\in\mathbf{Son}(v)}\, w(v,u))$ time, so that the total time complexity of the case (A1) is $O(rW(T))$.

Case (A2). Now we have $s(v)<w(v,\mathbf{Father}(v))$. Let the tree T' be identical with the tree T with the same leaf-colouration and with the weight function

$$w'(f)=\begin{cases} s(v) & \text{if } f=(v,\mathbf{Father}(v))\ ,\\ w(f) & \text{otherwise}\ .\end{cases}$$

Now it is easy to see that there exists an optimal colouration $\bar{\chi}$ of T' satisfying $\bar{\chi}(v)=\bar{\chi}(\mathbf{Father}(v))$ which is also optimal in T. (The only problem that can occur is that $\bar{\chi}(\mathbf{Father}(v))\in M_2(v)$ but $\bar{\chi}(\mathbf{Father}(v))\in M_3'(v)$. In that case we can apply the extended Phase II'.) Therefore, we have $l(T)=l(T')$ and $W(T')<W(T)$. Now we can easily 'lift up' any optimal path packing $\mathscr{P}$ of T' to the tree T, namely $\mathscr{P}$ itself is obviously path packing in T.

This operation takes $O(1)$ time, so the total time complexity of case (A2) is $O(n)$.

Case (B). From now on we assume that every vertex z of order 1 satisfies the condition $s(z)=w(z,\mathbf{Father}(z))$. For the rest of (B), we fix a vertex v; if the diameter of T is 3, then let v be the **root**, otherwise, let v be a non-root vertex such that $\mathbf{Son}(v) \not\subset L(T)$ and every non-leaf son is a vertex of order 1 (the existence of such a v is obvious). Let the non-leaf sons of v be the vertices $z_1, \ldots, z_k$.

By the definition of case (B) it is easy to see the existence of an optimal coloration $\bar{\chi}$ colouring v and every z_i to the same colour. Therefore if $\bar{T}$ is the tree derived from the tree T by contracting every edge of form (v, z_i) (leaving the name of the new vertex v), which is endowed with the original leaf-colouration and weight function on the existing edges, then the restriction of the same colouration $\bar{\chi}$ is also optimal for $\bar{T}$ and $l(\bar{T}) = l(T)$. On the other hand, the tree $\bar{T}$ has less branching vertices than T.

Now due to our hypothesis we have an optimal path packing $\bar{\mathscr{P}}$ in the tree $\bar{T}$. Therefore

$$|\bar{\mathscr{P}}| = l(T) .$$

Let us define the *lift up* $\mathscr{P} = \{\hat{P}\colon P \in \bar{\mathscr{P}}\}$ of the path packing $\bar{\mathscr{P}}$, where $\hat{P}$ is identical with P if no leaf u of $\mathbf{Son}(z_i)$ $(i = 1, \ldots, k)$ belongs to the path P, and $\hat{P}$ comes from P by subdivision of the edge (v, u) with vertex z_i if endvertex$(P) = u \in \mathbf{Son}(z_i)$ $(i = 1, \ldots, k)$. We have $l(T)$ many elements in $\mathscr{P}$.

Let $e_i = (v, z_i)$ (for every $i = 1, \ldots, k$). For an edge $f = (p, q)$, we write $-f = (q, p)$. Now, by the definition of $\mathscr{P}$, the condition

$$n_i(f, \mathscr{P}) + n_j(-f, \mathscr{P}) \leqslant w(f)$$

holds for every edge $f \neq e_i$ $(i = 1, \ldots, k)$, but unfortunately this is not necessarily the case for the edges e_i.

We solve this problem in a slightly more general setting (Lemma 2). For this we introduce the following notations: Let $[x]^+$ denote x, if x is non-negative, 0, if x is non-positive. Define the *badness* of the colour changing path system $\mathscr{P}$ by

$$\mathrm{bad}(\mathscr{P}) = \sum_{\substack{(i,j) \in C \times C \\ i \neq j}} \sum_{e \in E(G)} [n_i(e, \mathscr{P}) + n_j(-e, \mathscr{P}) - w(e)]^+ .$$

Call an edge *oversaturated* by the path system $\mathscr{P}$, if the contribution of the edge to the badness is positive. (We recall the definition $e_i = (v, z_i)$.)

Lemma 2. *Let $\mathscr{P}$ be a system of colour-changing paths on the tree T such that*

(i) *for all i, j, $n_j(\pm e_i, \mathscr{P}) \leqslant w(e_i)$,*

(ii) *$\mathscr{P}$ does not oversaturate any edge from $E(T) \setminus \{e_1, \ldots, e_k\}$.*

Then there exists a path packing $\mathscr{P}^*$ in T of the same size.

Proof. If $\mathrm{bad}(\mathscr{P}) = 0$ then $\mathscr{P}$ itself is a path packing. Suppose $\mathrm{bad}(\mathscr{P}) > 0$, and, say, the edge e_1 is oversaturated with colours 1 and 2, i.e.

$$n_1(e_1, \mathscr{P}) + n_2(-e_1, \mathscr{P}) > w(e_1) .$$

Take a path $P_1 \in \mathscr{P}$ such that $e_1 \in P_1$ and $\chi(t(P_1)) = 1$ (where, say, $t(P_1) \in \mathbf{Son}(z_1)$), and a path $P_2 \in \mathscr{P}$ such that $-e_1 \in P_2$ and $\chi(t(P_2)) = 2$ (where $t(P_2) \notin \mathbf{Son}(z_1)$ and $s(P_2) \in \mathbf{Son}(z_1)$). Now we distinguish the cases (BA) and (BB):

Case (BA). Suppose there is no $P_3 \in \mathscr{P}$ for which $-e_1 \in P_3$, $s(P_3) = s(P_2)$ and $\chi(t(P_3)) = 1$. In this case we define the following path system:

$$\mathscr{P}_1 = \mathscr{P} \cup \{P\} \setminus \{P_1\} ,$$

where the path P is $(s(P_2), z_1, t(P_1))$, oriented from left to right.

Claim A.

$$\operatorname{bad}(\mathscr{P}_1) \leqslant \operatorname{bad}(\mathscr{P}) - 1 .$$

Proof. It is easy to see that $n_i(\pm f, \mathscr{P}_1) \leqslant n_i(\pm f, \mathscr{P})$ for each $i = 1, \ldots, k$ and for each $f \in E(T) \setminus \{e_1, (z_1, s(P_2))\}$, furthermore

$$n_i(-e_1, \mathscr{P}_1) = n_i(-e_1, \mathscr{P}), \quad i = 1, \ldots, k ,$$

$$n_i(e_1, \mathscr{P}_1) = n_i(e_i, \mathscr{P}), \quad i = 2, \ldots, k ,$$

$$n_1(e_1, \mathscr{P}_1) = n_1(e_1, \mathscr{P}) - 1 .$$

Finally, for the edge $f_2 = (z_1, s(P_2))$ we have

$$n_i(f_2, \mathscr{P}_1) = n_i(f_2, \mathscr{P}), \quad i = 1, \ldots, k ,$$

$$n_i(-f_2, \mathscr{P}_1) = n_i(-f_2, \mathscr{P}), \quad i = 2, \ldots, k ,$$

$$n_1(-f_2, \mathscr{P}_1) + n_i(f_2, \mathscr{P}_1) \leqslant w(f_2), \quad i = 1, \ldots, k .$$

The last inequality is true, since otherwise $n_2(-f_2, \mathscr{P}) + n_i(f_2\ \mathscr{P}) > w(f_2)$ would hold, contradicting the assumptions of Lemma 2. □

Case (BB). Suppose there exists a path P_3 which was forbidden in (BA). Then let $\mathscr{P}_1$ be the following path system:

$$\mathscr{P}_1 = \mathscr{P} \cup \{P, P_3 \wedge P_1\} \setminus \{P_1, P_3\}$$

where $P_3 \wedge P_1$ denotes the (unique) path oriented from $s(P_3)$ to $t(P_1)$.

Claim B.

$$\operatorname{bad}(\mathscr{P}_1) \leqslant \operatorname{bad}(\mathscr{P}) - 1 .$$

Proof. Set

$$E_1 = \{e_1, (z_1, t(P_1)), (z_1, s(P_3))\} \quad \text{and} \quad E_2 = E(P_1) \cup E(P_2) \setminus E(P_3 \wedge P_1) .$$

Then for each edge $f \in E(T) \setminus (E_1 \cup E_2)$ the estimates of Claim A hold. Furthermore, for $f \in E_1$ we have

$$\begin{aligned}
&n_i(\pm f, \mathscr{P}_1) = n_i(\pm f, \mathscr{P}), \quad i=2, \dots, k, \\
&n_1(\pm f, \mathscr{P}_1) \leqslant n_1(\pm f, \mathscr{P}), \\
&n_i(\pm(z_1, t(P_1)), \mathscr{P}_1) = n_i(\pm(z_1, t(P_1)), \mathscr{P}), \quad i=1, \dots, k, \\
&n_i(\pm e_1, \mathscr{P}_1) = n_i(\pm e_1, \mathscr{P}), \quad i=2, \dots, k, \\
&n_1(\pm e_1, \mathscr{P}_1) = n_1(\pm e_1, \mathscr{P}) - 1, \\
&n_i(\pm(z_1, s(P_3))) = n_i(\pm(z_1, s(P_3)), \mathscr{P}) \quad i=1, \dots, k.
\end{aligned}$$

The equalities and inequalities above prove Claim B. □

The surgeries described in Case (BA) and Case (BB) obviously keep the conditions of Lemma 2, therefore they may be repeated until the badness drops to 0. Claims A and B guarantee, that we finally reach 0. Lemma 2 and Theorem 2 are proved. □

The determination of the tree $\bar{T}$ takes $O(n)$ steps, therefore the total time complexity of this procedure is $O(nb(T))$. To lift up the paths from $\bar{\mathscr{P}}$ to $\mathscr{P}$ takes

$$O\left(r \sum_{z \in \mathbf{Son}(v)} w(v, z)\right)$$

time, therefore the total time complexity of lift up operations is $O(rW(T))$. Finally, the badness at Lemma 2 is at most

$$\sum_{z \in \mathbf{Son}(v)} w(v, z)$$

and every edge can occur at most one application of Lemma 2 so the total time complexity of Lemma 2 is $O(\max\{rW(T), n^2\})$.

The bookkeeping of (edge, path) incidences is necessary. A possible execution of this task is to build up lists for every edge to store these incidences and to maintain these lists at every 'lift up' step. The total time complexity of our recursive procedure is $O(\max\{rW(T), n^2\})$, so it is unary polynomial.

The following theorem is an easy consequence of Theorem 2.

Theorem 3. *Let G be a graph with a weight function $w: E(T) \to \mathbb{N}$ and with a partial colouration $\chi: N \to C$. Assume that N intersects every cycle of G. Then*

$$l(G, \chi) = p(G, \chi)$$

Proof. Obtain a forest by eliminating the vertices of N and making leaves from the edges that were adjacent to them. Give the colour of n to the leaves that substitute a former $n \in N$. Apply Theorem 2 for each and every tree in the forest. □

5. The LP connection

One may consider the following linear programs related to the multiway cut problem with colour independent weight function. Note that this is something, which is different from the usual multiway cut polyhedron [1].

For every oriented edge (p, q) of G and every ordered pair of distinct colours ij define a variable $z_{pq,ij}$. If $q \in N$, then eliminate $z_{pq,ij}$ and $z_{qp,ji}$ for every $j \neq \chi(q)$. Introduce new quotient variables by identifying the surviving variables $z_{pq,ij}$ and $z_{qp,ji}$ in pairs. For convenience we use the same notation for the quotient variables. Then the primal linear program is:

$$z_{pq,ij} \geqslant 0 ;$$

for every colour-changing path P_{ab} $(a, b \in N)$, have

$$\sum_{(p,\ q) \in P_{ab}} \ \sum_{i : i \neq \chi(b)} z_{pq,i\chi(b)} \geqslant 1;$$

$$\min \ \sum z_{pq,ij}\, w(p, q) \ ,$$

where the last sum is for all quotient variables. To describe the dual linear program, for every colour-changing path P_{ab} introduce a variable λ_{ab}, such that

$$\lambda_{ab} \geqslant 0 ;$$

for every quotient variable $z_{pq,ij}$, have

$$\sum_{\substack{\chi(b) = j \\ (p,\ q) \in P_{ab}}} \lambda_{ab} + \sum_{\substack{\chi(v) = i \\ (q,\ p) \in P_{uv}}} \lambda_{uv} \leqslant w(p, q);$$

$$\max \ \sum \lambda_{ab} \, .$$

We claim that these linear programs have integer optimal solutions. It is easy to see, that

$$p(G, \chi) \leqslant \max \sum \lambda_{ab} : \lambda_{ab} \text{ integer} \leqslant \max \sum \lambda_{ab} = \min \sum z_{pq,ij}\, w(p, q)$$
$$\leqslant \min \sum z_{pq,ij}\, w(p, q) : z_{pq,ij} \text{ integer} \leqslant l(G, \chi) \ .$$

Only the first and last inequalities require proofs from the chain of inequalities above. The first one holds, since any path packing provides a feasible integer solution for the second linear program. The last one holds, since we have an optimal colouration $\bar{\chi}$ with total weight of the colour-changing edges of $l(G, \chi)$; define $z_{pq,ij}=1$, iff (p, q) is a colour-changing edge in the optimal colouration $\bar{\chi}$ and $\bar{\chi}(p)=i$, $\bar{\chi}(q)=j$ hold, and $z_{pq,ij}=0$ otherwise. If $l(G, \chi)=p(G, \chi)$. then equality holds everywhere in the chain.

It is a natural question whether these linear programs are totally dual integral [10], i.e., whether they have integer optimal solutions for colour dependent weight functions $w(p, q; i, j)$. Unfortunately, this is not the case, take for example the 3-star with center c and leaves x, y, z with colours $\chi(x)=1$, $\chi(y)=2$ and $\chi(z)=3$; and the weight function $w(c, .; i, j)={}_iW_j$ defined by the matrix

$$W=\begin{pmatrix} 0 & 1 & 3 \\ 3 & 0 & 1 \\ 1 & 3 & 0 \end{pmatrix}.$$

References

[1] S. Chopra and M.R. Rao, ''On the multiway cut polyhedron,'' *Networks* 21 (1991) 51–89.

[2] W.H. Cunningham, ''The optimal multiterminal cut problem,'' *DIMACS Series in Discrete Math.* 5 (1991) 105–120.

[3] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P. Seymour and M. Yannakakis, ''The complexity of multiway cuts,'' extended abstract (1983).

[4] P.L. Erdős and L.A. Székely, ''Evolutionary trees: an integer multicommodity max–flow-min–cut theorem,'' *Advances in Applied Mathematics* 13 (1992) 375–389.

[5] P.L. Erdős and L.A. Székely, ''Algorithms and min–max theorems for certain multiway cut,'' in: E. Balas, G. Cornuéjols and R. Kannan, eds., *Integer Programming and Combinatorial Optimization*, Proceedings of the Conference held at Carnegie Mellon University, May 25–27, 1992, by the Mathematical Programming Society (CMU Press, Pittsburgh, 1992) 334–345.

[6] W.M. Fitch, ''Towards defining the course of evolution. Minimum change for specific tree topology,'' *Systematic Zoology* 20 (1971) 406–416.

[7] J.A. Hartigan, ''Minimum mutation fits to a given tree,'' *Biometrics* 29 (1973) 53–65.

[8] L. Lovász and M.D. Plummer, *Matching Theory* (North-Holland, Amsterdam, 1986).

[9] K. Menger, ''Zur allgemeinen Kurventheorie,'' *Fundamenta Mathematicae 10 (1926) 96–115.*

[10] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization* (John Wiley & Sons, New York, 1988).

[11] M. Steel, ''Decompositions of leaf-coloured binary trees,'' *Advances in Applied Mathematics* 14 (1993) 1–24.

[12] P.L. Williams and W.M. Fitch, ''Finding the minimal change in a given tree,'' in: A. Dress and A. v. Haeseler, eds., *Trees and Hierarchical Structures*, Lecture Notes in Biomathematics 84 (1989) 75–91.

[13] D. Sankoff and R.J. Cedergren, ''Simultaneous comparison of three or more sequences related by a tree,'' in: D. Sankoff and J.B. Kruskal, eds., *Time Wraps, String Edits and Macromoleculas: The Theory and Practice of Sequence Comparison* (Addison-Wesley, London, 1983) 253–263.

ADVANCES IN APPLIED MATHEMATICS 14, 200–216 (1993)

Fourier Calculus on Evolutionary Trees

L. A. SZÉKELY*

Department of Computer Science, Eötvös University, H-1088 Budapest and
Department of Mathematics, University of New Mexico,
Albuquerque, New Mexico 87131

M. A. STEEL

Department of Mathematics, University of Canterbury, Private Bag,
Christchurch, New Zealand

AND

P. L. ERDŐS

Hungarian Academy of Sciences, H-1055 Budapest, Hungary and
Centrum voor Wiskunde en Informatica, 1098SJ Amsterdam, The Netherlands

We describe a Fourier analysis approach to the reconstruction theory of evolutionary trees that is based on Kimura's model of molecular evolution. © 1993 Academic Press, Inc.

1. INTRODUCTION

The purpose of the present paper is to develop in full generality the mathematical tools that are being used in the spectral analysis/closest tree method [H, HP1, HP2, SESP, SHSE, HPS] for the reconstruction of evolutionary trees in Cavender's model [C1] and in Kimura's three-parameter model [K1, K2, K3]. All sections of this paper but the very last can be read with zero knowledge from biology. The last section explains the biological significance of the results from previous sections. An important tool of our work is the Fourier calculus over finite Abelian groups; we acknowledge the influence of Evans and Speed [ES]. We have already announced part of the results of the present paper without proofs in [SES]. The following lemma summarizes the basic facts that we need on

*Research supported by the A. v. Humboldt-Stiftung and the U.S. Office of Naval Research under the Contract N-0014-91-J-1385.

0196-8858/93 $9.00

characters and Fourier transform. We use the additive notation in Abelian groups.

Lemma 1. *Let G be a finite Abelian group, then*

(i) *the character group $\hat{G}$ is isomorphic to G.*

(ii) *if $f\colon G \to C$ is a complex-valued function and $\hat{f}\colon \hat{G} \to C$ is defined by*

$$\hat{f}(\chi) = \sum_{g \in G} \chi(g) f(g),$$

then for all $g \in G$

$$f(g) = \frac{1}{|G|} \sum_{\chi \in \hat{G}} \overline{\chi(g)} \hat{f}(\chi).$$

(iii) *The characters of a finite direct product of finite Abelian groups are exactly the sums of characters.*

Proof. See [Kö]. □

Assume $A = (a_{ij})$ is a $p \times q$ matrix with integer entries. Let us be given a finite Abelian group G and the elements of G^q written in a vector form $\mathbf{x} = (x_1, \ldots, x_q)^{\mathrm{T}}$, where $x_j \in G$. Define the vector $\mathbf{y} \in G^p$ by $\mathbf{y} = (y_1, \ldots, y_p)^{\mathrm{T}}$, such that

$$y_i = \sum_{j=1}^{q} a_{ij} x_j.$$

(We want to abbreviate this fact to $A\mathbf{x} = \mathbf{y}$ and do not abuse this formalism.) Let us be given $p_j\colon G \to C$ functions ($j = 1, \ldots, q$). Define for $\mathbf{x} = (x_1, \ldots, x_q)^{\mathrm{T}} \in G^q$,

$$F(\mathbf{x}) = \prod_{j=1}^{q} p_j(x_j).$$

For $\mathbf{y} = (y_1, \ldots, y_p)^{\mathrm{T}} \in G^p$, let

$$f(\mathbf{y}) = \sum_{\substack{\mathbf{x} \in G^q \\ A\mathbf{x} = \mathbf{y}}} F(\mathbf{x}).$$

THEOREM 2. *If* $\chi = (\chi_1, \dots, \chi_p)^{\mathrm{T}} \in \hat{G}^p$, *then*

$$\hat{f}(\chi) = \prod_{j=1}^{q} \sum_{x \in G} p_j(x) \left(\sum_{i=1}^{p} a_{ij} \chi_i \right)(x).$$

Proof. By definition,

$$\hat{f}(\chi) = \sum_{\mathbf{y} \in G^p} \chi(\mathbf{y}) f(\mathbf{y}) = \sum_{\mathbf{y} \in G^p} \chi(\mathbf{y}) \sum_{\substack{\mathbf{x} \in G^q: \\ A\mathbf{x} = \mathbf{y}}} F(\mathbf{x}) = \sum_{\mathbf{x} \in G^q} F(\mathbf{x}) \chi(A\mathbf{x}).$$

Now we have

$$\chi(A\mathbf{x}) = \prod_{i=1}^{p} \chi_i((A\mathbf{x})_i) = \prod_{i=1}^{p} \chi_i \left(\sum_{j=1}^{q} a_{ij} x_j \right) = \prod_{j=1}^{q} \prod_{i=1}^{p} \chi_i(a_{ij} x_j).$$

Hence,

$$\hat{f}(\chi) = \sum_{\mathbf{x} \in G^q} \prod_{j=1}^{q} p_j(x_j) \prod_{i=1}^{p} \chi_i(a_{ij} x_j) = \prod_{j=1}^{q} \sum_{x \in G} p_j(x_j) \prod_{i=1}^{p} \chi_i(a_{ij} x_j),$$

as claimed. □

Note that for $A = [1, 1]$, $\mathbf{x} = (f, g)^{\mathrm{T}}$, Theorem 2 gives back a special instance of the classical result for the Fourier transform of the convolution, $\widehat{f * g} = \hat{f} \cdot \hat{g}$.

2. OUR MODEL AND ITS BASIC IDENTITIES

First we describe the mathematical model, which we work with. Let us be given a tree T with leaf set L and one arbitrary leaf R, called a *root*. We assume that no vertex has degree two. Assume that we are given a finite Abelian group G and for the edges $e \in E(T)$ we have independent G-valued random variables ξ_e with distributions $p_e(g) := \mathrm{Prob}(\xi_e = g)$, such that $\sum_{g \in G} p_e(g) = 1$. We call the set of p_e distributions ($e \in E(T)$) a *transition mechanism* and denote it by p.

Take G^{n-1} = the set of leaf colourations σ: $L \setminus \{R\} \to G$ endowed with pointwise operation; we denote the value of σ at l by σ_l. Produce a random G-colouration of the leaves of the tree by evaluating ξ_e for every edge and giving as colour to the leaf l the sum of group elements along the unique Rl path. Let f_σ denote the probability that we obtain the leaf colouration σ: $L \setminus \{R\} \to G$ in this way. In case we want to emphasize the dependence from the tree T and the transition mechanism p, we will write $f_\sigma(T, p)$.

Let $\chi = (\chi_l \in \hat{G}\colon l \in L\setminus\{R\})$ be an ordered $(n-1)$-tuple of characters. Then $\chi \in \hat{G}^{n-1}$, and χ acts on G^{n-1} according to Lemma 1(iii). For $e \in E(T)$, set

$$L_e = \{l \in L\colon e \text{ separates } l \text{ from } R \text{ in } T\}.$$

For $e \in E(T)$ and $\chi \in \hat{G}^{n-1}$, set

$$\chi_e = \sum_{l\in L_e} \chi_l, \tag{1}$$

so $\chi_e \in \hat{G}$. For $h \in \hat{G}$, $e \in E(T)$ define

$$l_e(h) = \sum_{g\in G} h(g)p_e(g), \tag{2}$$

$$r_\chi = \prod_{e\in E(T)} l_e(\chi_e). \tag{3}$$

We have the following Fourier inverse pair:

THEOREM 3. *With* $\chi(\sigma) = \prod_{l\in L\setminus\{R\}}\chi_l(\sigma_l)$,

$$r_\chi = \sum_{\sigma\in G^{n-1}} \chi(\sigma)f_\sigma \tag{4}$$

$$f_\sigma = \frac{1}{|G|^{n-1}} \sum_{\chi\in\hat{G}^{n-1}} \overline{\chi(\sigma)}r_\chi. \tag{5}$$

Proof. Observe that (4) and (5) are equivalent by Lemma 1(ii) for *any* $f\colon G^{n-1} \to C$ and $r\colon \hat{G}^{n-1} \to C$. (We decided not to use the usual hat notation for this pair since their significance and frequent occurrence in this paper.) To prove (4) with *our* f_σ and r_χ, apply Theorem 2 in the following setting: $p = n - 1$, $q = |E(T)|$, $A = (a_{ie})$ with

$$a_{ie} = \begin{cases} 1 & \text{if edge } e \text{ lies on the } Ri \text{ path} \\ 0 & \text{otherwise.} \end{cases}$$

Take $\Xi = (\xi_e\colon e \in E(T))$ the vector of random group elements selected independently on the edges, $p_e(x) = \text{Prob}(\xi_e = x)$, Υ = the vector of the resulting random leaf colouration. Observe that the independence implies $F(\mathbf{x}) = \text{Prob}(\Xi = \mathbf{x})$, and $f(\mathbf{y}) = \text{Prob}(\Upsilon = \mathbf{y})$. □

For later use we define the polynomials $R_\chi = \sum_{\sigma \in G^{n-1}} \chi(\sigma) x_\sigma$, with independent variables x_σ. Observe that while R_χ is tree independent, $r_\chi = R_\chi|_{x_\sigma = f_\sigma}$ is tree dependent.

THEOREM 4. *For the transition mechanisms $p^{(i)}$, p^* on the tree T and $\sigma \in G^{n-1}$ we have*

$$\sum_{\substack{(\sigma_1, \sigma_2, \dots, \sigma_k): \\ \sigma_1 + \sigma_2 + \cdots + \sigma_k = \sigma \\ \sigma_i \in G^{n-1}}} \prod_{i=1}^{k} f_{\sigma_i}(T, p^{(i)}) = f_\sigma(T, p^*),$$

where for $g \in G$

$$p_e^*(g) = \sum_{\substack{(g_1, g_2, \dots, g_k): \\ g_1 + g_2 + \cdots + g_k = g \\ g_i \in G}} \prod_{i=1}^{k} p_e^{(i)}(g_i).$$

Proof. Define for $\sigma \in G^{n-1}$,

$$f(\sigma) = \sum_{\substack{(\sigma_1, \sigma_2, \dots, \sigma_k): \\ \sigma_1 + \sigma_2 + \cdots + \sigma_k = \sigma \\ \sigma_i \in G^{n-1}}} \prod_{i=1}^{k} f_{\sigma_i}(T, p^{(i)})$$

and $f_i(\sigma) = f_\sigma(T, p^{(i)})$. We are going to prove $f(\sigma) = f_\sigma(T, p^*)$. Applying Theorem 2 to the group G^{n-1} in the setting $p = k$, $q = 1$, $A = (1, 1, \dots, 1)$, $p_i(\sigma) = f_i(\sigma)$ yields

$$\hat{f}(\chi) = \prod_{i=1}^{k} \hat{f}_i(\chi);$$

and by Theorem 3 and (3)

$$\hat{f}_i(\chi) = \prod_{e \in E(T)} \sum_{g \in G} \chi_e(g) p_e^{(i)}(g).$$

Therefore,

$$\hat{f}(\chi) = \prod_{e \in E(T)} \sum_{g \in G} \chi_e(g) p_e^*(g).$$

Finally, by Theorem 3,

$$\frac{1}{|G|^{n-1}} \sum_{\chi \in G^{n-1}} \overline{\chi(\sigma)}\hat{f}(\sigma) = f_\sigma(T, p^*),$$

and by Lemma 1(ii),

$$f(\sigma) = \frac{1}{|G|^{n-1}} \sum_{\chi \in G^{n-1}} \overline{\chi(\sigma)}\hat{f}(\sigma),$$

yielding the wanted $f(\sigma) = f_\sigma(T, p^*)$. □

We note that a special case of Theorem 4 occurred in the Ph.D. thesis of the second author [S]. An algebra-oriented reader may be interested in the fact that Theorem 4 boils down to the commutative law in the group algebra $C[G^{n-1}]$.

3. Main Identities

For $e \in E(T)$, $0 \neq g \in G$, define $\rho^{e,g} \in G^{n-1}$ in the following way: $\rho_l^{e,g} = 0$ for $l \notin L_e$, $l \neq R$, and $\rho_l^{e,g} = g$ for $l \in L_e$. Define $\mathscr{C}(T) = \{\rho^{e,g}$: $e \in E(T)$, $0 \neq g \in G\}$. For the following theorem (and later on) we assume, that for every $e \in E(T)$, $p_e(0)$ is sufficiently close to 1, and hence r_χ is also sufficiently close to one; therefore "logarithm" (such that $\log 1 = 0$ and $\log (ab) = \log a + \log b$ sufficiently many times) can be given a satisfactory definition. Having the logarithm, complex exponentiation a^b will be $\exp(b \log a)$, as usual.

Theorem 5. *For* $0_{G^{n-1}} \neq \rho \in G^{n-1}$, $\rho \notin \mathscr{C}(T)$,

$$\prod_{\chi \in \hat{G}^{n-1}} r_\chi^{\chi(\rho)} = 1;$$

for $\rho = \rho^{e,g} \in \mathscr{C}(T)$,

$$\prod_{\chi \in \hat{G}^{n-1}} r_\chi^{\chi(\rho)} = \prod_{h \in \hat{G}} l_e(h)^{h(g)|G|^{n-2}};$$

and for $\rho = 0_{G^{n-1}}$,

$$\prod_{\chi \in \hat{G}^{n-1}} r_\chi^{\chi(\rho)} = \prod_{e \in E(T)} \prod_{h \in \hat{G}} l_e(h)^{|G|^{n-2}}.$$

The identities remain valid with all exponents conjugated.

Proof. By (3) we have

$$\prod_{\chi\in\hat{G}^{n-1}} r_\chi^{\chi(\rho)} = \prod_{e\in E(T)} \prod_{h\in\hat{G}} l_e(h)^{\Sigma\{\chi(\rho):\, \chi_e = h\}},$$

(1)–(2) altogether with $\chi(\rho) = \prod_{l\in L\setminus\{R\}}\chi_l(\rho_l)$ imply

$$\sum\{\chi(\rho): \chi_e = h\} = \sum_{\substack{\chi_l\in\hat{G}:\\ l\in L\setminus\{R\}}} \left\{\prod_{l\in L\setminus\{R\}} \chi_l(\rho_l): \sum_{l\in L_e} \chi_l = h\right\}. \tag{6}$$

Now it is obvious that for $\rho = 0_{G^{n-1}}$,

$$\sum_{\substack{\chi_l\in\hat{G}:\\ l\in L\setminus\{R\}}} \left\{1: \sum_{l\in L_e} \chi_l = h\right\} = |G|^{n-2},$$

since having fixed an arbitrary $j \in L_e$, we have $|G|$ choices for χ_l for any $l \in L\setminus\{R, j\}$, and finally a unique choice for χ_j. Similarly, for $\rho = \rho^{e,g} \in \mathscr{E}(T)$,

$$\sum_{\substack{\chi_l\in\hat{G}:\\ l\in L\setminus\{R\}}} \left\{\prod_{l\in L\setminus\{R\}} \chi_l(\rho_l): \sum_{l\in L_e} \chi_l = h\right\} = h(g)|G|^{n-2},$$

since for any $\chi = (\chi_l: l \in L\setminus\{R\})$, $\chi(\rho^{e,g}) = h(g)$ and having fixed an arbitrary $j \in L_e$, we have $|G|$ choices for χ_l for any $l \in L\setminus\{R, j\}$ and, finally, a unique choice for χ_j, like above.

The nontrivial part of the proof is the first identity. By the definition of $\mathscr{E}(T)$, for $0_{G^{n-1}} \neq \rho \notin \mathscr{E}(T)$, either there

(α) exists $l \notin L_e$, $l \neq R$ with $\rho_l \neq 0_G$, or

(β) exist $l, j \in L_e$, such that $\rho_l \neq \rho_j$.

In (α), take an $\eta \in \hat{G}$ such that $\eta(\rho_l) \neq 1$. Such an η exists, since by Lemma 1(ii) the matrix $[\chi(g)]$ is regular, and it already has a column full of ones, namely, for $\rho = 0$. In (6), assign to the character $\chi = (\chi_1, \ldots, \chi_l, \ldots, \chi_{n-1})$ the character $\chi = (\chi_1, \ldots, \eta + \chi_l, \ldots, \chi_{n-1})$. Observe that, on the one hand, we just permuted the terms in the sum (6) and therefore fixed the value of the sum; on the other hand, we multiplied the sum by $\eta(\rho_l) \neq 1$. Hence, the sum is 0.

In (β), take an $\eta \in \hat{G}$ such that $\eta(\rho_j - \rho_l) = \eta(\rho_j)\eta^{-1}(\rho_l) \neq 1$. Such an η exists, since like in (α), $\rho_j - \rho_l$ would yield a second column full of ones in $[\chi(g)]$, contradicting the regularity. In (6), assign to the character $\chi = (\chi_1, \dots, \chi_l, \dots, \chi_j, \dots, \chi_{n-1})$ the character $\chi = (\chi_1, \dots, \chi_l - \eta, \dots, \chi_j + \eta, \dots, \chi_{n-1})$. Observe that, on the one hand, we just permuted the terms in the sum (6) and therefore fixed the value of the sum; on the other hand, we multiplied the sum by $\eta(\rho_j - \rho_l) \neq 1$. Hence, the sum is 0.

The proof of the conjugated exponent version is virtually the same and we leave it to the reader. □

We give an alternative logarithmic formulation of Theorem 5, since this logarithmic formulation was discovered and published for $G = Z_2$ [H] and $G = Z_2 \times Z_2$ [SHSE]. Let $K = [h(g)]$ denote the matrix, in which rows correspond to $h \in \hat{G}$ and columns correspond to $g \in G$; let $H = [\chi(\sigma)]$ denote the matrix, in which rows correspond to $\chi \in \hat{G}^{n-1}$ and columns correspond to $\sigma \in G^{n-1}$. Let the logarithm of a vector denote the vector of logarithms of the components. Let $\mathbf{f}$ denote the vector of f_σ's ($\sigma \in G^{n-1}$), and let $\mathbf{p}_e$ denote the vector of $p_e(g)$'s ($g \in G$) for every $e \in E(T)$.

THEOREM 6.

$$\left[H^{-1} \log H\mathbf{f}\right]_\rho = \begin{cases} 0, & \text{if } 0 \neq \rho \notin \mathscr{E}(T), \\ \left[K^{-1} \log K\mathbf{p}_e\right]_h, & \text{if } \rho = \rho^{e,h} \in \mathscr{E}(T), \\ \Sigma_{e \in E(T)} \Sigma_{h \in G}\left[K^{-1} \log K\mathbf{p}_e\right]_h, & \text{if } \rho = 0. \end{cases} \tag{7}$$

Proof. Take the logarithm of the conjugated exponent versions of the identities in Theorem 5 and use the identities for the adjugates

$$\frac{1}{|G|}K^* = K^{-1}, \qquad \frac{1}{|G|^{n-1}}H^* = H^{-1}$$

to eliminate the powers of group orders. □

4. SERIES EXPANSION

We say that a vector $\mathbf{x}$ of x_σ's ($\sigma \in G^{n-1}$) is *regular*, if $\Sigma_\sigma x_\sigma = 1$, x_σ is non-negative real, $x_0 > \frac{1}{2}$. For the expansions in this section regularity is a convenient sufficient condition, although it is not necessary.

THEOREM 7. *For a regular* $\mathbf{x}$ *and* $\sigma \neq 0$,

$$\left[H^{-1}\log H\mathbf{x}\right]_{\sigma} = \sum_{r=1}^{\infty} \frac{(-1)^{r+1}}{r} \sum_{\substack{(\sigma_1,\dots,\sigma_r):\\ \sigma_1+\cdots+\sigma_r=\sigma\\ \sigma_i\neq 0}} \prod_{i=1}^{r} \frac{x_{\sigma_i}}{x_0}.$$

Proof. We use regularity to establish

$$\left|\sum_{\sigma:\,\sigma\neq 0} \chi(\sigma) x_0\right| < x_0. \tag{8}$$

Indeed,

$$\left|\sum_{\sigma:\,\sigma\neq 0} \chi(\sigma) x_{\sigma}\right| \leq \sum_{\sigma:\,\sigma\neq 0} |\chi(\sigma)|\,|x_{\sigma}| = \sum_{\sigma:\,\sigma\neq 0} x_{\sigma} = 1 - x_0 < x_0.$$

We start with

$$[H\mathbf{x}]_{\chi} = \sum_{\sigma} \chi(\sigma) x_{\sigma} = x_0\left(1 + \sum_{\sigma:\,\sigma\neq 0} \chi(\sigma)\frac{x_{\sigma}}{x_0}\right).$$

We combine (8) with the fact that radius of convergence of the Taylor series of $\log z$ at $z = 1$ is 1:

$$[\log H\mathbf{x}]_{\chi} = \log x_0 + \sum_{r=1}^{\infty} \frac{(-1)^{r+1}}{r}\left(\sum_{\sigma:\,\sigma\neq 0} \chi(\sigma)\frac{x_{\sigma}}{x_0}\right)^{r}.$$

Hence

$$\begin{aligned}\left[H^{-1}\log H\mathbf{x}\right]_{\rho} &= \frac{1}{|G|^{n-1}} \sum_{r=1}^{\infty} \frac{(-1)^{r+1}}{r} \sum_{\chi} \overline{\chi(\rho)} \left(\sum_{\sigma_1:\,\sigma_q\neq 0} \chi(\sigma_1)\frac{x_{\sigma_1}}{x_0}\right) \\ &\qquad \cdots \left(\sum_{\sigma_r:\,\sigma_r\neq 0} \chi(\sigma_r)\frac{x_{\sigma_r}}{x_0}\right) \\ &= \frac{1}{|G|^{n-1}} \sum_{r=1}^{\infty} \frac{(-1)^{r+1}}{r} \sum_{\substack{(\sigma_1,\dots,\sigma_r):\\ \sigma_i\neq 0}} \frac{x_{\sigma_1}x_{\sigma_2}\cdots x_{\sigma_r}}{x_0^{r}} \\ &\qquad \times \sum_{\chi} \chi(-\rho+\sigma_1+\cdots+\sigma_r).\end{aligned}$$

Now observe that $\Sigma_\chi \chi(-\rho + \sigma_1 + \cdots + \sigma_r)$ vanishes, except if $-\rho + \sigma_1 + \cdots + \sigma_r = 0$ according to the summation in the theorem; and in this case its value is $|G|^{n-1}$. □

Corollary 8. *For a regular* $\mathbf{x}$ *and* $\sigma \neq 0$, *we have the first- and second-order approximations*

$$[H^{-1} \log H\mathbf{x}]_\sigma \approx x_\sigma / x_0,$$

$$[H^{-1} \log H\mathbf{x}]_\sigma \approx \frac{x_\sigma}{x_0} - \frac{1}{2} \sum_{\substack{(\sigma_1, \sigma_2):\\ \sigma_1 + \sigma_2 = \sigma\\ \sigma_1, \sigma_2 \neq 0}} \frac{x_{\sigma_1} x_{\sigma_2}}{x_0^2},$$

respectively.

Let p^{*k} denote the k-order convolution of the transition mechanism with itself as defined in Theorem 4; now Theorems 4 and 7 and a standard inclusion–exclusion argument allows for the following expansion.

Corollary 9. *For regular* $\mathbf{x}$ *and* $\sigma \neq 0$,

$$[H^{-1} \log H\mathbf{f}]_\sigma = \sum_{r=1}^{\infty} \sum_{k=1}^{r} \frac{(-1)^{k+1} \binom{r}{k} f_\sigma(T, p^{*k})}{r f_0^k(T, p)}. \quad \square$$

5. Invariants

Let us be given a tree T and another tree T' on the same leaf set L and root R. Consider the indeterminates x_σ for $\sigma \in G^{n-1}$ again. A multivariate function $q_T(\ldots, x_\sigma, \ldots)$ is an *invariant* of the tree T, if q vanishes after the substitution of $f_\sigma(T, p)$'s into x_σ's, for any transition mechanism p of T. We expect that an invariant is non-zero for a typical substitution of $f_\sigma(T', p')$'s into the x_σ's; and hence searching for the tree T' and its transition mechanism p' that resulted in the observed f_σ, we may reject a wrong candidate T, using its invariant(s). Consider

$$\mathrm{Split}(T) = \{L_e(T) : e \in E(T)\}$$

and observe that every element of Split(T) is represented by a *unique* edge e, since T has no vertex of degree two. Call an edge $e \in E(T)$ *passive* for (T, p), if $p_e(0) = 1$. Consider the set of ordered pairs (tree, transition mechanism) on the same fixed leaf set L and root R; and define a relation $\sim$ by $(T, p) \sim (T', p')$ iff a (T'', p'') can be reached from both by contracting passive edges. It is easy to see that $\sim$ is an equivalence relation. For

$\rho \in G^{n-1}$, define the tree independent $C^n \to C$ functions

$$\delta_\rho = \prod_{\chi \in \hat{G}^{n-1}} R_\chi^{\overline{\chi(\rho)}} - 1$$

in a neighborhood of $x_0 = 1$, $x_\sigma = 0$. For $0 \neq \rho \notin \mathscr{C}(T)$, on the basis of Theorem 5, we term the δ_ρ's as the *canonical invariants* of the tree T.

Now we are ready to state the main results of this Section; writing $\mathbf{p}_e$ in vector form we put $p_e(0)$ into the first coordinate.

THEOREM 10. *Assume that for the transition mechanisms p and p', for any edge e the vectors $\mathbf{p}_e$ and $\mathbf{p}'_e$ are sufficiently close to* $(1, 0, \ldots, 0)^{\mathrm{T}}$.

(i) *If $f_\sigma(T, p)$ satisfies the canonical invariants of T', then the elements of* $\mathrm{Split}(T) \setminus \mathrm{Split}(T')$ *are represented by passive edges in T.*

(ii) *If $f_\sigma(T, p)$ satisfies the canonical invariants of T' and $f_\sigma(T', p')$ satisfies the canonical invariants of T, then $(T, p) \sim (T', p')$.*

(iii) *If a leaf colouration probability distribution f_σ comes from both (T, p) and (T', p'), then $(T, p) \sim (T', p')$.*

(iv) *The canonical invariants of the tree T are algebraically independent.*

Proof. (i) Take an $e \in E(T)$ such that $L_e \notin \mathrm{Split}(T')$. Then $\rho^{e,h} \notin \mathscr{C}(T')$ for $0 \neq h \in G$; and the hypothesis of (i) implies $[H^{-1} \log H\mathbf{f}]_{\rho^{e,h}} = 0$ for all $h \neq 0$. On the other hand, (7) implies $[H^{-1} \log H\mathbf{f}]_{\rho^{e,h}} = [K^{-1} \log K\mathbf{p}_e]_h$ for all $h \neq 0$. Hence, $[K^{-1} \log K\mathbf{p}_e]_h = 0$ for all $h \neq 0$. In other words, $K^{-1} \log K\mathbf{p}_e = (x, 0, \ldots, 0)^{\mathrm{T}}$ for some number x, and hence $\log K\mathbf{p}_e = (x, x, \ldots, x)^{\mathrm{T}}$, $K\mathbf{p}_e = (\exp(x), \exp(x), \ldots, \exp(x))^{\mathrm{T}}$, and finally $\mathbf{p}_e = (\exp(x), 0, \ldots, 0)$; i.e., the edge e must have been passive.

(ii) is a simple application of (i). Observe that the hypothesis of (iii) implies the hypothesis of (ii), and hence the conclusion of (ii) holds.

We finish the proof by (iv). We prove more: the δ_ρ's are algebraically independent for $\rho \in G^{n-1}$. By the multivariate Taylor formula the δ_ρ's are algebraically independent iff the $\delta_\rho + 1$'s are. Suppose that

$$\sum_s \lambda_s \prod_{\rho \in G^{n-1}} (\delta_\rho + 1)^{i_{\rho,s}} = \sum_s \lambda_s \prod_{\chi \in \hat{G}^{n-1}} R_\chi^{\Sigma_\rho i_{\rho,s} \overline{\chi(\rho)}} \tag{9}$$

is identically zero in a neighborhood of $x_0 = 1$, $x_\sigma = 0$ with a certain finite set of complex coefficients λ_s and non-negative integer exponents $i_{\rho,s}$. We may assume without loss of generality that $s \neq s'$ implies that for some ρ we have $i_{\rho,s} \neq i_{\rho,s'}$. Since the invertible linear transformation H turns the

x_σ's into the R_χ's, we may study the vanishing of (9) in the independent variables R_χ's, all in a neighborhood of one. Having independent variables, the only way of vanishing (9) is cancellation; i.e., for some $s \neq s'$ and all $\chi \in G^{n-1}$,

$$\sum_{\rho \in G^{n-1}} i_{\rho,s}\overline{\chi(\rho)} = \sum_{\rho \in G^{n-1}} i_{\rho,s'}\overline{\chi(\rho)}. \tag{10}$$

The matrix H and its conjugate $\overline{H}$ are regular; hence (10) implies $i_{\rho,s} = i_{\rho,s'}$ for all $\rho \in G^{n-1}$, a contradiction. □

The reader might ask if logarithms and all the resulting fuss about smallness of some quantities are necessary to obtain our results. Therefore we show a simple example to point out that Theorem 10(iii) turns false if we drop these conditions. Take an arbitrary tree T and define the transition mechanism by $p_e(g) = 1/|G|$ for all $e \in E(T)$, $g \in G$. Clearly, f_σ will follow the uniform distribution independently of the topology of the tree, contrary to Theorem 10(iii).

In the rest of this section we restrict ourselves to $G = Z_2^m$. For an arbitrary given $0 \neq \rho \in (Z_2^m)^{n-1}$, we define the polynomial δ'_ρ of all x_σ's:

$$\delta'_\rho = \prod_{\substack{\chi \in (\widehat{Z_2^m})^{n-1}: \\ \chi(\rho)=1}} R_\chi - \prod_{\substack{\chi \in (\widehat{Z_2^m})^{n-1}: \\ \chi(\rho)=-1}} R_\chi .$$

Clearly, we obtained polynomial invariants, of which most of Theorem 10 can be easily told, with the annoying exception of their algebraic independence. In fact, we conjecture that the polynomials δ'_ρ, together with the polynomial $R_0 - 1 = \Sigma_\sigma x_\sigma - 1$, are algebraically independent.

It is worth making the following comment here. Evans and Speed [ES] conjecture that "the number of algebraically independent invariants and the number of free parameters among the $p_e(g)$'s obtained by an informal parameter count add up to the number of variables x_σ." Their first problem seems to have been to set candidates for these independent invariants. We have the suggestion above. Assume that for $g \neq 0$, $p_e(g)$ is a variable and $p_e(0) = 1 - \Sigma_{g\neq 0} p_e(g)$; then the number of free parameters is $|E(T)|(2^m - 1)$, the number of variables x_σ is $2^{m(n-1)}$, the number of canonical invariants δ'_ρ is $2^{m(n-1)} - |\mathscr{E}(T)| - 1 = 2^{m(n-1)} - |E(T)|(2^m - 1) - 1$; and actually, we have one more invariant, $R_0 - 1 = \Sigma_\sigma x_\sigma - 1$. The numerology works, but a positive result here would seem to involve algebraic geometry. Our Theorem 10(i) is some support for the conjecture of Evans and Speed.

6. Kimura's Models of Molecular Evolution

One assumes that the process of evolution is described by a tree. In this tree the labelled leaves denote some existing species represented by corresponding segments of aligned DNA sequences; the unlabelled branching vertices may denote unknown extinct ancestors. Let r denote the immediate ancestor of the closest common ancestor of a given set of existing species. We define the *true tree* of this set of species by taking the subtree induced by them and r in the tree describing the process of evolution and undoing the vertices of degree two.

The very problem of reconstruction may be put in this way: given a set of species with corresponding segments of aligned DNA sequences, find the true tree.

For $G = Z_2$, the model described in Section 2 specializes to a model of Cavender [C], for which Hendy and Penny found the special case of the calculus above and applied it in their spectral analysis/closest tree method for tree reconstruction from sequences over a two-letter purine–pyrimidine alphabet [H, HP1, HP2]. Our part is the generalization for other groups; the practical importance of this generalization is mostly for $G = Z_2 \times Z_2$, i.e., for sequences over the four-letter alphabet A, G, C, T; see [SHSE]. However, it is theoretically possible to apply our calculus to either of the two Abelian groups of order 20 (if the transition mechanisms of amino acids follow either of these groups), and also to Z_4, in Kimura's two-parameter model and the Jukes–Cantor model (see below). We explain the $G = Z_2 \times Z_2$ case in detail, the explanation also applies, mutatis mutandis, to $G = Z_2$.

From now on we describe Kimura's three-parameter model [K2, K3] and some restricted versions of it, which are known as Kimura's two-parameter model [K1] and Jukes–Cantor model [JC] (the Jukes–Cantor model is more explicit in Neyman [N]). We assume that every bit of the aligned DNA sequence is one of the four nucleotides, A (adenine), G (guanine), C (cytosine), T (thymine); i.e., we neglect insertions and deletions. We follow the group-theoretical setting of the models from Evans and Speed [ES]. Identify the elements of $Z_2 \times Z_2$ with the four nucleotides, such that A is the unity. Take the true tree with a common ancestor r and assume that an element of $Z_2 \times Z_2$ is assigned under a certain (unknown) distribution to r. The random group element at r is regarded as the original nucleotide value there. To every edge of the tree a random element of $Z_2 \times Z_2$ is assigned independently; the distribution may vary from edge to edge. The random variable at an edge describes the nucleotide change on that edge. In terms of biology, adding A = 0 on an edge causes no change in the nucleotide, adding G causes *transition*, and adding C or T causes one of the two possible types of *transversions*. To every leaf l the sum of group elements along the unique path rl and in r itself is assigned. We have a random four-colouration of the leaves (in fact, of all vertices) of the tree. That is Kimura's three-parameter model of molecular evolution. Kimura's three-parameter model allows for every edge e of the tree four arbitrary probabilities which sum up to one; i.e., three free parameters, which may be different on different edges. Kimura's two-parameter model is similar, but further restricted by $p_e(\mathrm{G}) = p_e(\mathrm{T})$ for all edges, and finally, the Jukes–Canter model requires, in addition, $p_e(\mathrm{C}) = p_e(\mathrm{T})$ for all edges.

After the work of Kimura, the general assumption for the mechanism of molecular evolution is that changes in the DNA are *random*. It is assumed that changes at different sites are independent and of identical distribution. In case the data violates too much the condition on identical distribution, one may thin out the sequences by considering one site of each of the *codons* (the consecutive triplets of nucleotides encoding amino acids), particularly the third position, which is more redundant in the coding scheme than the other two positions, and therefore less influenced by natural selection. It is an interesting paradox of the theory of evolution, that evolution is random at the molecular level and follows natural selection at a high level. It is surprising enough, that the models above were equiped with substitution mechanisms for transitions and transversions that fit perfectly the group theoretical description, although this was not the motivation for their invention.

The model, in which we work, slightly differs from Kimura's models, namely, we do not have a root r for an unknown common ancestor. This is in no way a serious loss, since biologists easily recover it by a method called *outgroup comparison*. The root that we use, is, like in Section 2, *one arbitrary leaf R*, which represents an existing species. At every site of the sequence of R, we find a group element, and for standardization, in every leaf we multiply at the same site with the inverse of that group element. We refer to the sequences obtained as *standardized sequences*; note that the standardized sequence of R contains zeros only. From the standardized sequences we can read a leaf colouration at every bit; we count relative frequencies of leaf colourations and we treat these relative frequencies as if they were the f_σ leaf colouration probabilities from the model of Section 2. Observe that the propagation of group elements along the tree is direction dependent unless $p_e(g) = p_e(g^{-1})$ for all e and g; and without this condition the standardization would not make sense. However, for $G = Z_2^m$, the condition holds automatically. Standardization sets no restriction on the distribution at r, since we rather work with nucleotide changes than use the nucleotide values. Despite the small difference, our method will allow for reconstruction of the true tree that evolved according to Kimura's model, with the loss of r and with the possible loss of the vertex adjacent to r, if it has degree three.

We had a set of species with corresponding segments of aligned DNA sequences. We selected an arbitrary species for R and we standardized the sequence from R and obtained an f'_σ relative frequency of the colouration σ among the bits. Now we face the following problem: which tree T and transition mechanism p yield leaf colouration probabilities $f_\sigma = f'_\sigma$ for all σ? Working with real data, we must be satisfied with the best approximation in a reasonable norm. Having the transition mechanism of the true tree allows for estimating a time scale, i.e., how long ago the evolutionary events in question did happen. We note here, that the model of Section 2 does not imply the existence of the logarithms; however, for real data, there is no problem with them, due to the empirical fact that $f'_0 > \frac{1}{2}$. Working with $\mathbf{f}$ arising from the model of Section 2, Theorem 6 tells the edges of the tree, and one can obtain the transition mechanism, i.e., $\mathbf{p}_e$ for all edges as well. The message of Theorem 10(iii) is that we may expect a *unique* tree to yield the observed relative frequencies of leaf colourations.

Working with empirical $\mathbf{f}'$, the closest tree method [H], which is a branch-and-bound algorithm, determines then the evolutionary tree and its transition mechanism, which yields $\mathbf{f}$, such that $H^{-1} \log H\mathbf{f}$ approximates $H^{-1} \log H\mathbf{f}'$ best in the Euclidean norm.

The significance of the series expansion is that a second-order approximation of $H^{-1} \log H\mathbf{f}'$ can be computed $O(t^2)$ time, where t is the number of nonzero f'_σ's, which is subexponential by our experience for real data. The use of the second-order approximation is expected to be superior to computing of $H^{-1} \log H\mathbf{f}'$ by fast Fourier transform on real data; this is still to be tested.

The great advantage of using invariants is that one may discriminate against some trees without (strong) assumptions regarding the transition mechanism. Invariants were introduced by Cavender and Felsenstein [CF, C2, C3] and Lake [L]; and recently Evans and Speed [ES] gave an algebraic procedure based on Fourier analysis to decide if a polynomial is invariant or not for $G = Z_2^m$. The literature shows that all the efforts went for polynomial invariants. There is a good reason to look for linear invariants, namely, they are subject to reliable statistical methods. However, there are cases when linear invariants are known not to exist, including Kimura's three-parameter model [ES]. In lack of linear invariants, there is at most a theoretical reason to prefer polynomial invariants.

The advantage of our canonical invariants to other invariants is that they come from a predetermined list, and if you need the canonical invariants of a tree you just pick the right elements from the list. If it comes to application of our polynomial invariants, then values of the polynomial functions must be computed instead of the polynomials, since computer algebra in many variables is rather prohibitive.

We see the significance of the Fourier calculus on evolutionary trees in the fact that it puts the tree reconstruction to the basis of the generally accepted theory of molecular evolution by Kimura, while most tree reconstruction techniques lack any such mechanism in the background.

References

[C1] J. A. Cavender, Taxonomy with confidence, *Math. Biosci.* **40** (1978), 271–280.

[C2] J. A. Cavender, Mechanized derivations of linear invariants, *Molecular Biol. Evol.* **6** (1989), 301–316.

[C3] J. A. Cavender, Necessary conditions for the method of inferring phylogeny by linear invariants, *Math. Biosci.* **103** (1991), 69–75.

[CF] J. A. Cavender and J. Felsenstein, Invariants of phylogenies in a simple case with discrete states, *J. Class.* **4** (1987), 57–71.

[ES] S. N. Evans and T. P. Speed, Invariants of some probability models used in phylogenetic inference, *Ann. Statist.*, in press.

[H] M. D. Hendy, A combinatorial description of the closest tree algorithm for finding evolutionary trees, *Discrete Math.* **96** (1991), 51–58.

[HP1] M. D. Hendy and D. Penny, A framework for the quantitative study of evolutionary trees, *Systematic Zool.* **38**, No. 4 (1989), 297–309.

[HP2] M. D. Hendy and D. Penny, Spectral analysis of phylogenetic data, *J. Class.*, in press.

[HPS] M. D. Hendy, D. Penny, and M. A. Steel, Discrete Fourier spectral analysis for evolution, *Proc. Natl. Acad. Sci. U.S.A.* submitted.

[JC] T. H. Jukes and C. Cantor, Evolution in protein molecules, *in* "Mammalian Protein Metabolism" (H. N. Munro, Ed.), pp. 21–132, Academic Press, New York, 1969.

[K1] M. Kimura, A simple method for estimating evolutionary rates of base substitution through comparative studies of nucleotide sequences, *J. Mol. Evol.* **16** (1980), 111–120.

[K2] M. Kimura, Estimation of evolutionary sequences between homologous nucleotide sequences, *Proc. Nat. Acad. Sci. U.S.A.* **78** (1981), 454–458.

[K3] M. Kimura, "The Neutral Theory of Molecular Evolution," Cambridge Univ. Press, Cambridge, 1983.

[Kö] T. W. Körner, "Fourier Analysis," Cambridge Univ. Press, Cambridge, 1988.

[L] J. A. Lake, A rate-independent technique for analysis of nucleic acid sequences: Evolutionary parsimony, *Molecular Biol. Evol.* **4** (1987), 167–191.

[N] J. Neyman, Molecular studies of evolution: A source of novel statistical problems, *in* "Statistical Decision Theory and Related Topics" (S. S. Gupta and J. Yackel, Eds.), pp. 1–27, Academic Press, New York, 1971.

[S] M. A. Steel, "Distributions on Bicoloured Evolutionary Trees," Ph.D. thesis, Massey University, Palmerston North, 1989.

[SHSE] M. A. Steel, M. D. Hendy, L. A. Székely, and P. L. Erdős, Spectral analysis and a closet tree method for genetic sequences, *Appl. Math. Lett.*, **5**, No. 6 (1992), 63–67.

[SES] L. A. Székely, P. L. Erdős, M. A. Steel, The combinatorics of evolutionary trees —a survey, *in* Actes du Séminaire, Séminaire Lotharingien de Combinatoire, 28-ième session, 15–18 mars, 1992 (J. Zeng, Éd.), Publication de l'Institute de Recherche Mathématique Avancée, 129–143.

[SESP] L. A. Székely, P. L. Erdős, M. A. Steel, and D. Penny, A Fourier inversion formula for evolutionary trees, *Appl. Math. Lett.*, **6**, No. 2 (1993), 13–17.

A Few Logs Suffice to Build (Almost) All Trees (I)

Péter L. Erdős,[1] Michael A. Steel,[2] László A. Székely,[3] Tandy J. Warnow[4]

[1] *Mathematical Institute of the Hungarian Academy of Sciences, Budapest P.O. Box 127, Hungary-1364; e-mail: elp@math-inst.hu*

[2] *Biomathematics Research Centre, University of Canterbury, Christchurch, New Zealand; e-mail: m.steel@math.canterbury.ac.nz*

[3] *Department of Mathematics, University of South Carolina, Columbia, SC; e-mail: laszlo@math.sc.edu*

[4] *Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA; e-mail: tandy@central.cis.upenn.edu*

Received 26 September 1997; accepted 24 September 1998

ABSTRACT: A phylogenetic tree, also called an "evolutionary tree," is a leaf-labeled tree which represents the evolutionary history for a set of species, and the construction of such trees is a fundamental problem in biology. Here we address the issue of how many sequence sites are required in order to recover the tree with high probability when the sites evolve under standard Markov-style i.i.d. mutation models. We provide analytic upper and lower bounds for the required sequence length, by developing a new polynomial time algorithm. In particular, we show when the mutation probabilities are bounded the required sequence length can grow surprisingly slowly (a power of log n) in the number n of sequences, for almost all trees. © 1999 John Wiley & Sons, Inc. Random Struct. Alg., 14, 153–184, 1999

1. INTRODUCTION

Rooted leaf-labeled trees are a convenient way to represent historical relationships between extant objects, particularly in evolutionary biology, where such trees are

Correspondence to: László A. Székely

called *phylogenies*. Molecular techniques have recently provided large amounts of sequence data which are being used to reconstruct such trees. These methods exploit the variation in the sequences due to random mutations that have occurred at the sites, and statistically based approaches typically assume that sites mutate independently and identically according to a Markov model. Under mild assumptions, for sequences generated by such a model, one can recover, with high probability, the underlying *unrooted* tree provided the sequences are sufficiently long in terms of the number k of sites. How large this value of k needs to be depends on the reconstruction method, the details of the model, and the number n of species. Determining bounds on k and its growth with n has become more pressing since biologists have begun to reconstruct trees on increasingly large numbers of species, often up to several hundred, from such sequences.

With this motivation, we provide upper and lower bounds for the value of k required to reconstruct an underlying (unrooted) tree with high probability, and address, in particular, the question of how fast k must grow with n. We first show that under any model, and any reconstruction method, k must grow *at least* as fast as $\log n$, and that for a particular, simple reconstruction method, it must grow at least as fast as $n \log n$, for any i.i.d. model. We then construct a new tree reconstruction method (the dyadic closure method) which, for a simple Markov model, provides an upper bound on k which depends only on n, the range of the mutation probabilities across the edges of the tree, and a quantity called the "depth" of the tree. We show that the depth grows very slowly ($O(\log \log n)$) for almost all phylogenetic trees (under two distributions on trees). As a consequence, we show that the value of k required for accurate tree reconstruction by the dyadic closure method needs only to grow as a power of $\log n$ for almost all trees when the mutation probabilities lie in a fixed interval, thereby improving results by Farach and Kannan in [23].

The structure of the paper is as follows. In Section 2 we provide definitions, and in Section 3 we provide lower bounds for k. In Section 4 we describe a technique for reconstructing a tree from a partial collection of subtrees, each on four leaves. We use this technique in Section 5, as the basis for our "dyadic closure" method. Section 6 is the central part of the paper, here we analyze, using various probabilistic arguments, an upper bound on the value of k required for this method to correctly recover the underlying tree with high probability, when the sites evolve under a simple, symmetric 2-state model. As this upper bound depends critically upon the depth (a function of the shape of the tree) we show that the depth grows very slowly ($O(\log \log n)$) for a random tree selected under either of two distributions. This gives us the result that k need grow only sublinearly in n for nearly all trees.

Our follow-up paper [21] extends the analysis presented in this paper for more general, r-state stochastic models, and offers an alternative to dyadic closure, the "witness–antiwitness" method. The witness–antiwitness method is faster than the dyadic closure method on average, but does not yield a deterministic technique for reconstructing a tree from a partial collection of subtrees, as the dyadic closure method does; furthermore, the witness–antiwitness method may require somewhat longer (by a constant multiplicative factor) input sequences than the dyadic closure method.

2. DEFINITIONS

Notation. $\mathbb{P}[A]$ denotes the probability of event A; $\mathbb{E}[X]$ denotes the expectation of random variable X. We denote the natural logarithm by log. The set $[n]$ denotes $\{1, 2, \ldots, n\}$ and for any set S, $\binom{S}{k}$ denotes the collection of subsets of S of size k. $\mathbb{R}$ denotes the real numbers.

Definitions. (I) Trees. We will represent a phylogenetic tree T by a tree whose *leaves* (vertices of degree 1) are labeled (by extant species, numbered by $1, 2, \ldots, n$) and whose remaining internal vertices (representing ancestral species) are unlabeled. We will adopt the biological convention that phylogenetic trees are *binary*, so that all internal nodes have degree 3, and we will also assume that T is *unrooted*, for reasons described later in this section. There are $(2n-5)!! = (2n-5)(2n-7)\cdots 3 \cdot 1$ different binary trees on n distinctly labeled leaves.

The edge set of the tree is denoted by $E(T)$. Any edge adjacent to a leaf is called a *leaf edge*, any other edge is called an *internal edge*. The path between the vertices u and v in the tree is called the uv path, and is denoted $P(u, v)$. For a phylogenetic tree T and $S \subseteq [n]$, there is a unique minimal subtree of T, containing all elements of S. We call this tree the *subtree* of T induced by S, and denote it by $T_{|S}$. We obtain the *contracted subtree* induced by S, denoted by $T^*_{|S}$, if we substitute edges for all maximal paths of $T_{|S}$ in which every internal vertex has degree 2. Since all trees are assumed to be binary, all contracted subtrees, including, in particular, the subtrees on four leaves, are also binary. We use the notation $ij|kl$ for the contracted subtree on four leaves i, j, k, l in which the pair i, j is separated from the pair k, l by an internal edge, and we also call $ij|kl$ a *valid quartet split* of T. Clearly any four leaves i, j, k, l in a binary tree have exactly one valid quartet split out of $ij|kl, ik|jl, il|kj$.

The *topological distance* $d(u, v)$ between vertices u and v in a tree T is the number of edges in $P(u, v)$. A *cherry* in a binary tree is a pair of leaves at topological distance 2. The *diameter* of the tree T, diam(T), is the maximum topological distance in the tree. For an edge e of T, let T_1 and T_2 be the two rooted subtrees of T obtained by deleting edge e from T, and for $i = 1, 2$, let $d_i(e)$ be the topological distance from the root of T_i to its nearest leaf in T_i. The *depth* of T is $\max_e \max\{d_1(e), d_2(e)\}$, where e ranges over all internal edges in T. We say that a path P in the tree T is *short* if its topological length is at most depth$(T) + 1$, and say that a quartet i, j, k, l is a *short quartet* if it induces a subtree which contains a single edge connected to four disjoint short paths. The set of all short quartets of the tree T is denoted by $Q_{\text{short}}(T)$. We will denote the set of valid quartet splits for the short quartets by $Q^*_{\text{short}}(T)$.

(II) Sites. Let us be given a set C of character states (such as $C = \{A, C, G, T\}$ for DNA sequences; $C = \{$the 20 amino acids$\}$ for protein sequences; $C = \{R, Y\}$ or $\{0, 1\}$ for purine-pyrimidine sequences). A *sequence of length* k is an ordered k-tuple from C—that is, an element of C^k. A collection of n such sequences—one for each species labeled from $[n]$—is called a *collection of aligned sequences*.

Aligned sequences have a convenient alternative description as follows. Place the aligned sequences as rows of an $n \times k$ matrix, and call *site i* the ith column of this matrix. A *pattern* is one of the $|C|^n$ possible columns.

(III) Site substitution models. Many models have been proposed to describe, stochastically, the evolution of sites. Usually these models assume that the sites evolve identically and independently under a distribution that depends on the model tree. Most models are more specific and also assume that each site evolves on a rooted tree from a nondegenerate distribution π of the r possible states at the root, according to a Markov assumption (namely, that the state at each vertex is dependent only on its immediate parent). Each edge e oriented out from the root has an associated $r \times r$ stochastic transition matrix $M(e)$. Although these models are usually defined on a rooted binary tree T where the orientation is provided by a time scale and the root has degree 2, these models can equally well be described on an unrooted binary tree by (i) suppressing the degree 2 vertex in T, (ii) selecting an arbitrary vertex (leaves not excluded), assigning to it an appropriate distribution of states π', possibly different from π, and (iii) assigning an appropriate transition matrix $M'(e)$ [possibly different from $M(e)$] for each edge e. If we regard the tree as now rooted at the selected vertex, and the "appropriate" choices in (ii) and (iii) are made, then the resulting models give *exactly* the same distribution on patterns as the original model (see [46]) and as the rerooting is arbitrary we see why it is impossible to hope for the reconstruction of more than the *unrooted* underlying tree that generated the sequences under some time-induced, edge-bisection rooting. The assumption that the underlying tree is binary is also in keeping with the assumption in systematic biology, that speciation events are almost always binary.

(IV) The Neyman model. The simplest stochastic model is a symmetric model for binary characters due to Neyman [37], and also developed independently by Cavender [12] and Farris [25]. Let $\{0, 1\}$ denote the two states. The root is a fixed leaf, the distribution π at the root is uniform. For each edge e of T we have an associated *mutation probability*, which lies strictly between 0 and 0.5. Let p: $E(T) \to (0, 0.5)$ denote the associated map. We have an instance of the general Markov model with $M(e)_{01} = M(e)_{10} = p(e)$. We will call this the *Neyman 2-state model*, but note that it has also been called the Cavender–Farris model. Neyman's original paper allows more than 2 states.

The Neyman 2-state model is hereditary on the subsets of the leaves—that is, if we select a subset S of $[n]$, and form the subtree $T_{|S}$, then eliminate vertices of degree 2, we can define mutation probabilities on the edges of $T^*_{|S}$ so that the probability distribution on the patterns on S is the same as the marginal of the distribution on patterns provided by the original tree T. Furthermore, the mutation probabilities that we assign to an edge of $T^*_{|S}$ is just the probability p that the endpoints of the associated path in the original tree T are in different states. The probability that the endpoints of a path p are in different states is nicely related to the mutation probabilities $p_1, p_2, \ldots, p_k$ of edges of the k-path,

$$p = \frac{1}{2}\left(1 - \prod_{i=1}^{k} (1 - 2p_i)\right). \tag{1}$$

Formula (1) is well known, and is easy to prove by induction.

(V) Distances. Any symmetric matrix, which is zero-diagonal and positive off-diagonal, will be called a *distance matrix*. An $n \times n$ distance matrix D_{ij} is called *additive*, if there exists an n-leaf (not necessarily binary) with positive edge weights on the internal edges and nonnegative edge weights on the leaf edges, so that D_{ij} equals the sum of edge weights in the tree along the $P(i,j)$ path connecting i and j. In [10], Buneman showed that the following Four-Point Condition characterizes additive matrices (see also [42] and [53]):

Theorem 1 (Four-Point Condition). A matrix D is additive if and only if for all i, j, k, l (not necessarily distinct), the maximum of $\mathrm{D_{ij}} + \mathrm{D_{kl}}, \mathrm{D_{ik}} + \mathrm{D_{jl}}, \mathrm{D_{il}} + \mathrm{D_{jk}}$ is not unique. The edge-weighted tree with positive weights on internal edges and nonnegative weights on leaf edges representing the additive distance matrix is unique among the trees without vertices of degree 2.

Given a pair of parameters (T, p) for the Neyman 2-state model, and sequences of length k generated by the model, let $H(i,j)$ denote the *Hamming distance* of sequences i and j and

$$h^{ij} = \frac{H(i,j)}{k} \tag{2}$$

denote the *dissimilarity score* of sequences i and j. The *empirical corrected distance* between i and j is denoted by

$$d_{ij} = -\tfrac{1}{2}\log(1 - 2h^{ij}). \tag{3}$$

The probability of a change in the state of any fixed character between the sequences i and j is denoted by $E^{ij} = \mathbb{E}(h^{ij})$, and we let

$$D_{ij} = -\tfrac{1}{2}\log(1 - 2E^{ij}) \tag{4}$$

denote the *corrected model distance* between i and j. We assign to any edge e a positive weight,

$$w(e) = -\tfrac{1}{2}\log(1 - 2p(e)). \tag{5}$$

By Eq. (1), D_{ij} is the sum of the weights (see previous equation) along the path $P(i,j)$ between i and j. Therefore, d_{ij} converges in probability to D_{ij} as $k \to \infty$. Corrected distances were introduced to handle the problem that Hamming distances underestimate the "true evolutionary distances." In certain continuous time Markov models the edge weight means the expected number of back-and-forth state changes along the edge, and defines an additive distance matrix.

(VI) Tree reconstruction. A *phylogenetic tree reconstruction method* is a function Φ that associates either a tree or the statement `fail` to every collection of aligned sequences, the latter indicating that the method is unable to make such a selection for the data given. Some methods are based upon sequences, while others are based upon distances.

According to the practice in systematic biology (see, for example, [29, 30, 49]), a method is considered to be *accurate* if it recovers the unrooted binary tree T, even if it does not provide any estimate of the mutation probabilities. A necessary condition for accuracy, under the models discussed above, is that two distinct trees, T, T', do not produce the same distribution of patterns no matter how the trees are rooted, and no matter what their underlying Markov parameters are. This "identifiability" condition is violated under an extension of the i.i.d. Markov model when there is an unknown distribution of rates across sites as described by Steel, Székely, and Hendy [46]. However, it is shown in Steel [44] (see also Chang and Hartigan [13]) that the identifiability condition holds for the i.i.d. model under the weak conditions that the components of π are not zero and the determinant $\det(M(e)) \neq 0, 1, -1$, and in fact we can recover the underlying tree from the expected frequencies of patterns on just *pairs* of species.

Theorem 1 and the discussion that follows it suggest that appropriate methods applied to corrected distances will recover the correct tree topology from sufficiently long sequences. Consequently, one approach to reconstructing trees from distances is to seek an additive distance matrix of minimum distance (with respect to some metric on distance matrices) from the input distance matrix. Many metrics have been considered, but all resultant optimization problems have been shown or are assumed to be NP-hard; see [1, 15, 24].

We will use a particular simple distance method, which we call the (*Extended Four-Point Method* (FPM), to reconstruct trees on four leaves from a matrix of interleaf distances.

Four-Point Method (*FPM*). *Given a* 4×4 *distance matrix* d, *return the set of splits* $ij|kl$ *which satisfy* $d_{ij} + d_{kl} \leq \min\{d_{ik} + d_{jl}, d_{il} + d_{jk}\}$.

Note that the Four-Point Method can return one, two, or three splits for a given quartet. One split is returned if the minimum is unique, two are returned if the two smallest values are identical but smaller than the largest, and three are returned if all three values are equal.

In [26], Felsenstein showed that two popular methods—*maximum parsimony* and *maximum compatibility*—can be statistically inconsistent, namely, for some parameters of the model, the probability of recovering the correct tree topology tends to 0 as the sequence length grows. This region of the parameter space has been subsequently named the "Felsenstein zone." This result, and other more recent embellishments (see Hendy [28], Zharkikh and Li [54], Takezaki and Nei [50], Steel, Székely, and Hendy [46]), are asymptotic results—that is, they are concerned with outcomes as the sequence length, k, tends to infinity.

We consider the question of how many sites k must be generated independently and identically, according to a substitution model M, in order to reconstruct the underlying binary tree on n species with prespecified probability at least ϵ by a particular method Φ. Clearly, the answer will depend on Φ, ϵ, and n, and also on the fine details of M—in particular the unknown values of its parameters. It is clear that for all models that have been proposed, if no restrictions are placed on the parameters associated with edges of the tree then the sequence length might need to be astronomically large, even for four sequences, since the "edge length" of the internal edge(s) of the tree can be made arbitrarily short (as was pointed out by Philippe and Douzery [38]). A similar problem arises for four sequences when one or more of the four noninternal edges is "long"—that is, when site saturation has occurred on the line of descent represented by the edge(s). Unfortunately, it is difficult to analyze how well methods perform for sequences of a given length, k. There has been some empirical work done on this subject, in which simulations of sequences are made on different trees and different methods compared according to the sequence length needed (see [31] for an example of a particularly interesting study of sequence length needed to infer trees of size 4), but little analytical work (see, however, [38]).

In this paper we consider only the Neyman 2-state model as our choice for M. However, our results extend to the general i.i.d. Markov model, and the interested reader is referred to the companion paper [21] for details.

3. LOWER BOUNDS

Since the number of binary trees on n leaves is $(2n-5)!!$, encoding deterministically all such trees by binary sequences at the leaves requires that the sequence length, k, satisfy $(2n-5)!! \leq 2^{nk}$, i.e., $k = \Omega(\log n)$. We now show that this information-theoretic argument can be extended for *arbitrary* models of site evolution and *arbitrary* deterministic or even randomized algorithms for tree reconstruction. For each tree, T, and for each algorithm A, whether deterministic or randomized, we will assume that T is equipped with a mechanism for generating sequences, which allows the algorithm A to reconstruct the topology of the underlying tree T from the sequences with probability bounded from below.

Theorem 2. *Let A be an arbitrary algorithm, deterministic or randomized, which is used to reconstruct binary trees from* 0-1 *sequences of length k associated with the leaves, under an arbitrary model of substitutions. If A reconstructs the topology of any binary tree T from the sequences at the leaves with probability greater than ϵ (respectively, greater than $\frac{1}{2}$), then $(2n-5)!!\epsilon < 2^{nk}$ (respectively, $(2n-5)!! \leq 2^{nk}$, under the assumption of (stochastic) independence of the substitution model and the reconstruction) and so $k = \Omega(\log n)$.*

We prove this theorem in a more abstract setting:

Theorem 3. *We have finite sets X and S and random functions f: $S \rightarrow X$ and g: $X \rightarrow S$.*

- **(i)** *If $\mathbb{P}[fg(x)=x] > \epsilon$ for all $x \in X$ then $|S| > \epsilon|X|$.*
- **(ii)** *If f, g are independent and $\mathbb{P}[fg(x)=x] > \frac{1}{2}$ for all $x \in X$ then $|S| \geq |X|$.*

Proof. Proof of (i). By hypothesis $\epsilon|X| < \Sigma_x \mathbb{P}[fg(x)=x] = \Sigma_x \Sigma_s \mathbb{P}[g(x)=s$ and $f(s)=x] \leq \Sigma_s(\Sigma_x \mathbb{P}[f(s)=x]) = \Sigma_s 1 = |S|$.

Proof of (ii). First note that $\mathbb{P}[fg(x)=y] = \Sigma_s \mathbb{P}[f(s)=y]\mathbb{P}[g(x)=s]$ by independence. Observe that for each x, there exists an $s=s_x$ for which $\mathbb{P}[f(s_x)=x] > \frac{1}{2}$, since otherwise we have $\mathbb{P}[fg(x)=x] \leq \frac{1}{2}$. Now, the map sending x to s_x is one-to-one from X into S (and so $|X| \leq |S|$ as required) since otherwise, if two elements get mapped to s, then $1 = \Sigma_x \mathbb{P}[f(s)=x] > \frac{1}{2} + \frac{1}{2}$. ■

The following example shows that our theorem is tight for $\epsilon < \frac{1}{2}$: Let $X = \{x_{11}, x_{12}, x_{21}, x_{22}, \dots, x_{n1}, x_{n2}\}$ and $S = \{1, 2, \dots, n\}$, and let $g(x_{ij}) = i$ (with probability 1); and let $f(i) = x_{i1}$ with probability $\frac{1}{2}$; x_{i2} with probability $\frac{1}{2}$. Then $\mathbb{P}[fg(x) = x] = \frac{1}{2}$, so $\mathbb{P}[fg(x) = x] > \epsilon$, for *any* epsilon less than $\frac{1}{2}$. However, notice that $|X|/2 = |S|$.

Curiously, once ϵ exceeds $\frac{1}{2}$ we must have $|X| \le |S|$, under the assumption of independence. Examples [52] show that the assumption of independence is necessary. Independence is a reasonable assumption if we try to apply this result for evolutionary tree reconstruction, and holds automatically if the tree reconstruction method is deterministic.

This lower bound applied to an *arbitrary* algorithm, but *particular* algorithms may admit much larger lower bounds. Consider, for example, the *Maximum Compatibility Method* (MC), which we now define. Given a set of binary sequences, each site defines a partition of the sequences into two sets, those containing a 0 in that position, and those containing a 1 in that position. The site is said to be *compatible* on a tree T if the tree T contains an edge whose removal would define the same partition. The objective of the maximum compatibility method is a tree T which has the largest number of sites compatible with it. Maximum compatibility is an NP-hard optimization problem [16], although the MC method can clearly be implemented as a nonpolynomial time algorithm. We now show that the sequence length needed by MC to obtain the correct topology with constant probability must grow *at least* as fast as $n \log n$.

Theorem 4. *Assume that 2-state sites on n species evolve on a binary tree T according to any stochastic model in which the sites evolve identically and independently. Let $k(n)$ denote the smallest number of sites for which the Maximum Compatibility Method is guaranteed to reconstruct the topology of T with probability greater than $\frac{1}{2}$. Then, for n large enough,*

$$k(n) > (n-3)\log(n-3) - (n-3). \tag{6}$$

Proof. We say that a site is *trivial* if it defines a partition of the sequences into one class or into two classes so that one of the classes is a singleton. Now, fix x and assume that we are given $k^* = \lceil (n-3)\log(n-3) + x(n-3) \rceil$ nontrivial sites independently selected from the same distribution. We show that the probability of obtaining the correct tree under MC is at most $e^{-e^{-x}}$ for n large enough. This proves the theorem by setting $x = -1$, since $k(n) \ge k^*|_{x=-1}$ is needed.

Let $\sigma(T)$ denote the set of internal splits of T. Since T is binary, $|\sigma(T)| = n - 3$ [10]. For $\sigma \in \sigma(T)$, let the random variable X_σ be the number of nontrivial sites which induce split σ. Define $X = \sum_{\sigma \in \sigma(T)} X_\sigma$. A necessary (though not sufficient) condition for maximum compatibility to select T is that all the internal splits of T are present among the k^* nontrivial sites. Thus, we have the inequality,

$$\begin{aligned}
\mathbb{P}[MC(\mathbf{S}) = T] &\le \mathbb{P}\left[\cap_{\sigma \in \sigma(T)} \{X_\sigma > 0\}\right] \\
&= \sum_{i=1}^{k^*} \mathbb{P}\left[\cap_{\sigma \in \sigma(T)} \{X_\sigma > 0\} | X = i\right] \times \mathbb{P}[X = i] \\
&\le \max_{1 \le i \le k^*} \mathbb{P}\left[\cap_{\sigma \in \sigma(T)} \{X_\sigma > 0\} | X = i\right] \\
&= \mathbb{P}\left[\cap_{\sigma \in \sigma(T)} \{X_\sigma > 0\} | X = k^*\right].
\end{aligned} \tag{7}$$

Let $p(\sigma)$ denote the probability of generating split σ at a particular site. Due to the model, $p(\sigma)$ does not depend on the site. It is not difficult to show that (7) is maximized when the $p(\sigma)$s are all equal $(\sigma \in \sigma(T))$ and sum to 1.

Indeed, by compactness arguments, there exists a probability distribution maximizing (7). We show that it cannot be nonuniform, and therefore the uniform distribution maximizes (7). Assume that the maximizing distribution p is nonuniform, say, $p(\sigma) \neq p(\rho)$. We introduce a new distribution p' with $p'(\sigma) = p'(\rho) = \frac{1}{2}(p(\sigma) + p(\rho))$, and $p'(\alpha) = p(\alpha)$ for $\alpha \neq \sigma, \rho$. The probability of having exactly i sites supporting σ or ρ is the same for p and p'. Conditioning on the number of sites supporting σ or ρ, it is easy to see that any distribution of sites supporting all nontrivial splits has strictly higher probability in p' than in p.

Knowing that the $p(\sigma)$s are all equal $(\sigma \in \sigma(T))$ and sum to 1, determining (7) is just the classical occupancy problem where k^* balls are randomly assigned to $n-3$ boxes with uniform distribution, and one asks for the probability that each box has at least one ball in it. Equation (6) now follows from a result on the asymptotics of this problem (Erdős and Rényi [18]): for $x \in \mathbb{R}$, k^* balls (k^* as defined above), and $n-3$ boxes, the limit of probability of filling each boxes is $e^{-e^{-x}}$. ■

This theorem shows that the sequence length that suffices for the MC method to be accurate is in $\Omega(n \log n)$, but does not provide us with any *upper bound* on that sequence length. This upper bound remains an open problem.

In Section 5, we will present a new method [the *Dyadic Closure Method* (DCM)] for reconstructing trees. DCM has the property that for almost all trees, with a wide range allowed for the mutation probabilities, the sequence length that *suffices* for correct topology reconstruction grows no more than polynomially in the lower bound of $\log n$ (see Theorem 2) required for any method. In fact the same holds for *all trees* with a *narrow range* allowed for the mutation probabilities. First, however, we set up a combinatorial technique for reconstructing trees from selected subtrees of size 4.

4. DYADIC INFERENCE OF TREES

Certain classical tree reconstruction methods [6,14,47,48,55] are based upon reconstructing trees on quartets of leaves, them combining these trees into one tree on the entire set of leaves. Here we describe a method which requires only certain quartet splits be reconstructed (the “representative quartet splits”), and then infers the remaining quartet splits using “inference rules.” Once we have splits for all the possible quartets of leaves, we can then reconstruct the tree (if one exists) that is uniquely consistent with all the quartet splits.

In this section, we prove a stronger result than was provided in [19], that the *representative quartet splits* suffice to define the tree. We also present a tree reconstruction algorithm, DCTC (for *Dyadic Closure Tree Construction*) based upon dyadic closure. The input to DCTC is a set Q of quartet splits and we show that DCTC is guaranteed to reconstruct the tree properly if the set Q contains only valid quartet splits and contains all the representative quartet splits of T. We also show that if Q contains all representative quartet splits but also contains invalid quartet splits, then DCTC discovers incompatibility. In the remaining case, where Q does not contain all the representative quartet splits of any T, DCTC returns *Inconsistent* (and then the input was inconsistent indeed), or a tree (which is then the only tree consistent with the input), or *Insufficient*.

4.1. Inference Rules

Recall that, for a binary tree T on n leaves, and a quartet of leaves,

$$q = \{a, b, c, d\} \in \binom{[n]}{4}, \qquad t_q = ab|cd$$

is a *valid quartet split* of T if $T^*_{|q} = ab|cd$ (i.e., there is at least one edge in T whose removal separates the pair a, b from the pair c, d). It is easy to see that

$$\text{if } ab|cd \text{ is a valid quartet split of T, then so are } ba|cd \text{ and } cd|ab, \tag{8}$$

and we identify these three splits; and if $ab|cd$ holds, then $ac|bd$ and $ad|bc$ are not valid quartet splits of T, and we say that any of them *contradicts* $ab|cd$. Let

$$Q(T) = \left\{ t_q\colon q \in \binom{[n]}{4} \right\}$$

denote the set of valid quartet splits of T. It is a classical result that $Q(T)$ determines T (Colonius and Schulze [14], Bandelt and Dress [6]); indeed for each $i \in [n]$, $\{t_q\colon\ i \in q\}$ determines T, and T can be computed from $\{t_q\colon\ i \in q\}$ in polynomial time.

It would be nice to determine for a set of quartet splits whether there is a tree for which they are valid quartet splits. Unfortunately, this problem is NP-complete (Steel [43]). It also would be useful to know which subsets of $Q(T)$ determine T, and for which subsets a polynomial time procedure would exist to reconstruct T. A natural step in this direction is to define *inference*: we can infer from a set of quartet splits A a quartet split t, if whenever $A \subseteq Q(T)$ for a binary tree T, then $t \in Q(T)$ as well.

Instead, Dekker [17] introduced a restricted concept, *dyadic* and higher order inference. Following Dekker, we say that a set of quartet splits A *dyadically implies* a quartet split t, if t can be derived from A by repeated applications of rules (8)–(10):

$$\begin{gathered}\text{if } ab|cd \text{ and } ac|de \text{ are valid quartet splits of } T,\\ \text{then so are } ab|ce,\ ab|de, \text{ and } bc|de,\end{gathered} \tag{9}$$

and,

$$\text{if } ab|cd \text{ and } ab|ce \text{ are valid quartet splits of } T, \text{ then so is } ab|de. \tag{10}$$

It is easy to check that these rules infer valid quartet splits from valid quartet splits, and the set of quartet splits dyadically inferred from an input set of quartet splits can be computed in polynomial time. Setting a complete list of inference rules seems hopeless (Bryant and Steel [9]): for any r, there are r-ary inference rules, which infer a valid quartet split from some r valid quartet splits, such that their action cannot be expressed through lower order inference rules.

4.2. Tree Inference Using Dyadic Rules

In this section we define the *dyadic closure* of a set of quartet splits, and describe conditions on the set of quartet splits under which the dyadic closure defines all valid quartet splits of a binary tree. This section extends and strengthens results from earlier work [19, 45].

Definition 1. Given a finite set of quartet splits Q, we define the *dyadic closure* $\mathrm{cl}(Q)$ of Q as the set of quartet splits than can be inferred from Q by the repeated use of the rules (8–10). We say that Q is *inconsistent*, if Q is not contained in the set of valid quartet splits of any tree, otherwise Q is *consistent*. For each of the $n-3$ internal edges of the n-leaf binary tree T we assign a *representative quartet* $\{s_1, s_2, s_3, s_4\}$ as follows. The deletion of the internal edge and its endpoints defines four rooted subtrees t_1, t_2, t_3, t_4. Within each subtree t_i, select from among the leaves which are closest topologically to the root the one, s_i, which is the smallest natural number (recall that the leaves of our trees are natural numbers). This procedure associates to each edge a set of four leaves, i, j, k, l. (By construction, it is clear that the quartet i, j, k, l induces a short quartet in T—see Section 2 for the definition of "short quartet.") We call the quartet split of a representative quartet a *representative quartet split* of T, and we denote the set of representative quartet splits of T by R_T.

The aim of this section is to show that the dyadic closure suffices to compute the tree T from any set of valid quartet splits of T which contain R_T. We begin with:

Lemma 1. *Suppose S is a set of $n-3$ quartet splits which is consistent with a unique binary tree T on n leaves. Furthermore, suppose that S can be ordered $q_1, \ldots, q_{n-3}$ in such a way that q_i contains at least one label which does not appear in $\{q_1, \ldots, q_{i-1}\}$ for $i = 2, \ldots, n-3$. Then, the dyadic closure of S is $Q(T)$.*

Proof. First, observe that it is sufficient to show the lemma for the case when q_i contains *exactly* one label which does not appear in $\{q_1, \ldots, q_{i-1}\}$ for $i = 2, \ldots, n-3$, since $n-4$ quartets have to add $n-4$ new vertices. Let $S_i = \{q_1, \ldots, q_i\}$, and let L_i be the union of the leaves of the quartet splits in S_i, and let $T_i = T^*_{|L_i}$ be the binary subtree of T induced by L_i. We first make

Claim 1. *The only tree on L_i consistent with S_i is T_i, for* $1, \ldots, n-3$.

Proof of Claim 1. The claim is true by the hypothesis of Lemma 1 for $i = n-3$; suppose for some $i < n-3$ it is false. Then there exist (at least) two trees that realize S_i, one of which is T_i, the other we will call $T^\#$. Now each quartet $q_{i+1}, \ldots, q_{n-3}$ adds a new leaf to the tree so far constructed from T_i and $T^\#$. Now for each quartet *we can always* attach that new leaf in at least one position in the tree so far constructed so as to satisfy the corresponding quartet split (and all earlier ones, since they don't involve that leaf). Thus we end up with two trees consistent with S, and these are different trees since when we restrict them to L_i, they differ. But this contradicts our hypothesis. ■

Next we make

Claim 2. *If x is the new leaf introduced by $q_{n-3} = xa|bc$ then x and a form a cherry of T.*

Proof of Claim 2. First assume that x belongs to the cherry xy but $a \neq y$. Since this quartet is the only occurrence of x we do not have any information about this cherry, therefore the reconstruction of the tree T cannot be correct, a contradiction.

Now assume that x is not in a cherry at all. Then the neighbor of x has two other neighbors, and those are not leaves. In turn they have two other neighbors each. Hence, we can describe x's place in T in the following representation in Fig. 1: take a binary tree with five leaves, label the middle leaf x, and replace the other four leaves by corresponding subtrees of T.

Now suppose $q_{n-3} = ax|bc$. Regardless of where a, b, c come from (among the four subtrees in the representation), we can always move x onto at least two of the other four edges in T, and so obtain a different tree consistent with S (recall that q_{n-3} is the only quartet containing x, and thereby the only obstruction to us moving x!). Since the theorem assumes that the quartets are consistent with a unique tree, this contradicts our assumptions. ■

Finally, it is easy to show the following:

Claim 3. *Suppose xy is a cherry of T. Select leaves a, b from each of the two subtrees adjacent to the cherry. Let T' be the binary tree obtained by deleting leaf x. Then* $\mathrm{cl}(Q(T') \cup \{xy|ab\}) = Q(T)$.

Now, we can apply induction on n to establish the lemma. It is clearly (vacuously) true for $n = 4$, so suppose $n > 4$. Let x be the new leaf introduced by q_{n-3}, and let the binary tree T' be T with x deleted.

In view of Claim 1, S_{n-4} is a set of $n - 4$ quartets that define $T_{n-4} = T'$, a tree on $n - 1$ leaves and which satisfy the hypothesis that q_i introduces exactly one new leaf. Thus, applying the induction hypothesis, the dyadic closure of S_{n-4} is $Q(T')$. Since $S = S_{n-3}$ contains S_{n-4}, the dyadic closure of S also contains $Q(T')$, which is the set of all quartet splits of T that do not include x.

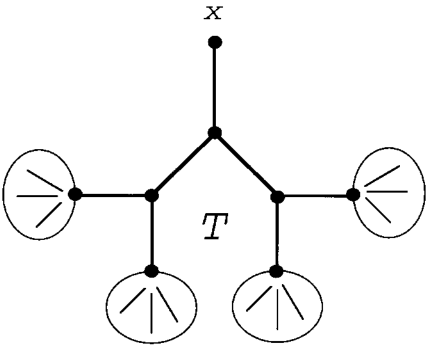


Fig. 1. Position of a leaf x, which is not a cherry, in a binary tree.

Now, by Claim 2, x is in a cherry; let its sibling in the cherry be y, so $q_{n-3} = ab|xy$, say, where a and b must lie in each of the two subtrees adjacent to the cherry. (It is easy to see that if a, b both lie in just one of these subtrees, then S would not define T.)

Now, as we just said, the dyadic closure of S contains $Q(T')$ and it also contains $ab|xy$ (where a, b are as specified in the preceding paragraph) and so by the idempotent nature of dyadic closure [i.e., $\mathrm{cl}(B) = \mathrm{cl}(\mathrm{cl}(B))$] it follows from Claim 3 that the dyadic closure of S equals $Q(T)$. ■ ■ ■

Lemma 2. *The set of representative quartet splits R_T of a binary tree T satisfies the conditions of Lemma* 1. *Hence, the dyadic closure of R_T is $Q(T)$.*

Proof. In order to make an induction proof possible, we make a more general statement. Given a binary tree T with a positive edge weighting w, we define the *representative quartet* of an edge e to be the quartet tree defined by taking the lowest indiced closest leaf in each of the four subtrees, where we define "closest" in terms of the weight of the path (rather than the topological distance) to the root of the subtree. We also define the *representative quartet splits of the weighted tree*, $R_{T,w}$ as in the definition of representative quartets of unweighted trees, with the only change being that each $s_i \in t_i$ is selected to minimize the *weighted* path length rather than topological path length (i.e., the edge weights on the path are summed together, to compute the weighted path length). Observe that if all weights are equal to 1, then we get back the original definitions. When turning to binary subtrees of a given weighted tree, we assign the sum of weights of the original edges to any newly created edge which is composed of them, and denote the new weighting by w^*. Now we can easily prove by induction the following generalization of the statement of Lemma 2:

Claim 4. *Take the set of representative quartet splits $R_{T,w}$ of a weighted n-leaf binary tree T. Then for every other n-leaf binary tree F, we have that $R_{T,w} \subseteq Q(F)$ implies $T = F$ as unweighted trees. Furthermore, $R_{T,w}$ can be ordered $q_1, \ldots, q_{n-3}$ in such a way that q_i contains exactly one label that does not appear in $\{q_1, \ldots, q_{i-1}\}$ for $i = 2, \ldots, n-3$.*

Proof of Claim 4. First we show that the only tree consistent with the set of representative splits $R_{T,w}$ of a binary tree T is T itself. Look for the smallest (in n) counterexample T, such that $R_{T,w} \subseteq Q(F)$ for a tree $F \neq T$. Clearly n has to be at least 5. Therefore T has at least two different cherries, say xy and uv, such that $d(u, x) \geq 4$. Let us denote by $w(l)$ the weight of the leaf edge corresponding to the leaf l. If $w(x) < w(y)$ or $[w(x) = w(y)$ and $x < y]$, then due to the construction of $R_{T,w}$, vertex y occurs in exactly one elements of $R_{T,w}$, say p, which is the representative of the edge that separates xy from the rest of the tree. A similar argument would show that one of u, v, say v, occurs in exactly one element of $R_{T,w}$, say q. It also follows that $p \neq q$. It is not difficult to check that

$$R_{T^*_{|[n]\setminus\{y\}}, w^*} = R_T \setminus \{p\} \quad \text{and} \quad R_{T^*_{|[n]\setminus\{v\}} w^*} = R_T \setminus \{q\} \tag{11}$$

according to the definition of weight after contracting edges, where $T^*_{|K}$ is the binary tree obtained by contracting paths into edges in the subtree of T spanned by the vertex set K. Hence, by the minimality of the counterexample, $T^*_{|[n]\setminus\{y\}} = F^*_{|[n]\setminus\{y\}}$ and $T^*_{|[n]\setminus\{v\}} = F^*_{|[n]\setminus\{v\}}$. We know that any edge of F defines a bipartition of $[n]$, and traces of these bipartitions on $[n]\setminus\{y\}$ and $[n]\setminus\{v\}$ are exactly the bipartitions produced by the edges of $F^*_{|[n]\setminus\{y\}}$ on $[n]\setminus\{y\}$ and the bipartitions produced by the edges of $F^*_{|[n]\setminus\{v\}}$ on $[n]\setminus\{v\}$. Therefore also in F both xy and uv make cherries, and hence $T = F$, a contradiction.

For the other part of the claim, it immediately follows by induction from formula (11) that $R_{T,w}$ can be ordered so that every quartet in the order contains *at least one* (and therefore *exactly one*) new leaf. [Eliminate quartet splits recursively using (11), and put $R_{T,w}$ in the reverse order.] ■

Note that the generalization for weighted trees was necessary, since without weights formula (11) would fail. ■ ■ ■

We note here that representative quartets cannot be defined by selecting *any* nearest leaf in the four subtrees associated with an internal edge. For example, consider the tree T on six leaves labeled 1 through 6, with a central vertex and cherries $(1,2)$, $(3,4)$, and $(5,6)$, hanging from the central vertex. If we selected the quartet splits by arbitrarily picking closest leaves in each of the four subtrees around each internal edge, we could possibly select splits 12|36, 34|15, and 56|24; however, these splits do not uniquely identify the tree T, since the tree with cherries 15, 24, and 36, is also consistent with these quartets.

4.3. Dyadic Closure Tree Construction Algorithm

We now present the Dyadic Closure Tree Construction method (DCTC) for computing the dyadic closure of a set Q of quartet splits, and which returns the tree T when $\mathrm{cl}(Q) = Q(T)$.

Before we present the algorithm, we note the following interesting lemma:

Lemma 3. *If* $\mathrm{cl}(Q)$ *contains exactly one split for each possible quartet then* $\mathrm{cl}(Q) = Q(T)$ *for a unique binary tree* T.

Proof. By Proposition (2) of [6], a set Q^* of noncontradictory quartet splits equals $Q(T)$ for some tree T precisely if it satisfies the substitution property: If $ab|cd \in Q^*$, then for all $e \notin \{a,b,c,d\}$, $ab|ce \in Q^*$, or $ae|cd \in Q^*$. Furthermore, in that case, T is unique.

Applying this characterization to $Q^* = \mathrm{cl}(Q)$, suppose $ab|cd \in \mathrm{cl}(Q)$ but $ab|ce \notin \mathrm{cl}(Q)$. Thus, either $ae|bc \in \mathrm{cl}(Q)$ or $ac|be \in \mathrm{cl}(Q)$. In the either case, the dyadic inference rule applied to the pair $\{ab|cd, ae|bc\}$ or to $\{ab|cd, ac|be\}$ implies $ae|cd \in \mathrm{cl}(Q)$, and so $\mathrm{cl}(Q)$ satisfies the substitution property. Thus $\mathrm{cl}(Q) = Q(T)$ for a unique tree T. Finally, since $\mathrm{cl}(Q)$ contains a split for each possible quartet, it follows that T must be binary. ■

We now continue with the description of the DCTC algorithm.

Algorithm DCTC.

Step 1. We compute the dyadic closure, cl(Q), of Q.

Step 2.

- **Case 1.** cl(Q) contains a pair of contradictory splits for some quartet: return *Inconsistent*.
- **Case 2.** cl(Q) has no contradictory splits, but fails to have a split for every quartet: Return *Insufficient*.
- **Case 3.** cl(Q) has exactly one split for each quartet: apply standard algorithms [6,51] to cl(Q) to reconstruct the tree T such that $Q(T) = \mathrm{cl}(Q)$. Return T.

(Case 3 depends upon Lemma 3 above.)

To completely describe the DCTC method we need to specify how we compute the dyadic closure of a set Q of quartet splits.

Efficient computation of dyadic closure. The description we now give of an efficient method for computing the dyadic closure will only actually completely compute the dyadic closure of Q if $\mathrm{cl}(Q) = Q(T)$ for some tree T. Otherwise, cl(Q) will either contain a contradictory pair of splits for some quartet, or cl(Q) will not contain a split for every quartet. In the first of these two cases, the method will return *Inconsistent*, and in the second of these two cases, the method will return *Insufficient*. However, the method can be easily modified to compute cl(Q) for all sets Q.

We will maintain a four-dimensional array `Splits` and constrain $\texttt{Splits}_{\texttt{i,"j,"k,"l}}$ to either be empty, or to contain exactly one split that has been inferred so far for the quartet i, j, k, l. In the event that two conflicting splits are inferred for the same quartet, the algorithm will immediately return *Inconsistent*, and halt. We will also maintain a queue Q_{new} of new splits that must be processed. We initialize `Splits` to contain the splits in the input Q, and we initialize Q_{new} to be Q, ordered arbitrarily.

The dyadic inference rules in equations (8)–(10) show that we infer new splits by combining two splits at a time, where the underlying quartets for the two splits share three leaves. Consequently, each split $ij|kl$ can only be combined with splits on quartets $\{a, i, j, k\}$, $\{a, i, j, l\}$, $\{a, i, k, l\}$, and $\{a, j, k, l\}$, where $a \notin \{i, j, k, l\}$. Consequently, there are only $4(n - 4)$ other splits with which any split can be combined using these dyadic rules to generate new splits.

Pop a split $ij|kl$ off the queue Q_{new}, and examine each of the appropriate $4(n - 4)$ entries in `Splits`. For each nonempty entry in `Splits` that is examined in this process, compute the $O(1)$ splits that arise from the combination of the two splits. Suppose the combination generates a split $ab|cd$. If $\texttt{Splits}_{a,b,c,d}$ contains a different split from $ab|cd$, then Return *Inconsistent*. If $\texttt{Splits}_{a,b,c,d}$ is empty, then set $\texttt{Splits}_{a,b,c,d} = ab|cd$, and add $ab|cd$ to the queue Q_{new}. Otherwise $\texttt{Splits}_{a,b,c,d}$ already contains the split $ab|cd$, and we do not modify the data structures.

Continue until the queue Q_{new} is empty, or Inconsistency has been observed. If the Q_{new} empties before Inconsistency is observed, then check if every entry of `Splits` is nonempty. If so, then $\text{cl}(Q) = Q(T)$ for some tree; Return `Splits`. If some entry in `Splits` is empty, then return *Insufficient*.

Theorem 5. *The efficient computation of the dyadic closure uses* $O(n^5)$ *time, and at the termination of the algorithm the* `Splits` *matrix is either identically equal to* $\text{cl}(Q)$, *or the algorithm has returned* Inconsistent. *Furthermore, if the algorithm returns* Inconsistent, *then* $\text{cl}(Q)$ *contains a pair of contradictory splits.*

Proof. It is clear that if the algorithm only computes splits using dyadic closure, so that at any point in the application of the algorithm, `Splits` $\subseteq \text{cl}(Q)$. Consequently, if the algorithm returns *Inconsistent*, then $\text{cl}(Q)$ does contain a pair of contradictory splits. If the algorithm does not return *Inconsistent*, then it is clear from the design that every split which could be inferred using these dyadic rules would be in the `Splits` matrix when the algorithm terminates.

The running time analysis is easy. Every combination of quartet splits takes $O(1)$ time to process. Processing a quartet split involves examining $4(n-4)$ entries in the `Splits` matrix, and hence costs $O(n)$. If a split $ij|kl$ is generated by the combination of two splits, then it is only added to the queue if $\texttt{Splits}_{i,j,k,l}$ is empty when $ij|kl$ is generated. Consequently, at most $O(n^4)$ splits ever enter the queue. ■

We now prove our main theorem of this section:

Theorem 6. *Let* Q *be a set of quartet splits.*

1. *If* $\text{DCTC}(Q) = T$, $\text{DCTC}(Q') = T'$, *and* $Q \subseteq Q'$, *then* $T = T'$.
2. *If* $\text{DCTC}(Q) =$ Inconsistent *and* $Q \subseteq Q'$, *then* $\text{DCTC}(Q') =$ Inconsistent.
3. *If* $\text{DCTC}(Q) =$ Insufficient *and* $Q' \subseteq Q$, *then* $\text{DCTC}(Q') =$ Insufficient.
4. *If* $R_T \subseteq Q \subseteq Q(T)$, *then* $\text{DCTC}(Q) = T$.

Proof. Assertion (1) follows from the fact that if $\text{DCTC}(Q) = T$, then the dyadic closure phase of the DCTC algorithm computes exactly one split for every quartet, so that $\text{cl}(Q) = Q(T)$ by Lemma 3. Therefore, if $Q \subseteq Q'$, then $\text{cl}(Q) \subseteq \text{cl}(Q')$, so that $Q(T) \subseteq \text{cl}(Q') = Q(T')$. Since T and T' are binary trees, it follows that $Q(T) = Q(T')$ and $T = T'$.

Assertion (2) follows from the fact that if $\text{DCTC}(Q) = \textit{Inconsistent}$, then $\text{cl}(Q)$ contains two contradictory splits for the same quartet. If $Q \subseteq Q'$, then $\text{cl}(Q')$ also contains the same two contradictory splits, and so $\text{DCTC}(Q') = \textit{Inconsistent}$.

Assertion (3) follows from the fact that if $\text{DCTC}(Q) = \textit{Insufficient}$, then $\text{cl}(Q)$ does not contain contradictory pairs of splits, and also lacks a split for at least one quartet. If $Q' \subseteq Q$, then $\text{cl}(Q')$ also does not contain contradictory pairs of splits and also lacks a split for some quartet. Consequently, $\text{DCTC}(Q') = \textit{Insufficient}$.

Assertion (4) follows from Lemma 2 and Assertion (1). ■

Note that $\text{DCTC}(Q) = \textit{Insufficient}$ does not actually imply that $Q \subset Q(T)$ for any tree; that is, it may be that $Q \not\subseteq Q(T)$ for any tree, but $\text{cl}(Q)$ may not contain any contradictory splits!

5. DYADIC CLOSURE METHOD

We now describe a new method for tree reconstruction, which we call the *Dyadic Closure Method*, or DCM.

Suppose T is a fixed binary tree. From the previous section, we know that if we can find a set Q of quartet splits such that $R_T \subseteq Q \subseteq Q(T)$, then DCTC($Q$) will reconstruct T.

One approach to find such a set Q would be to let Q be the set of splits (computed using the Four-Point Method) on all possible quartets. However, it is possible that the sequence length needed to ensure that *every* quartet is accurately analyzed might be too large to obtain accurate reconstructions of large trees, or of trees containing short edges.

The approach we take in the Dyadic Closure Method is to use sets of quartet splits based upon the quartets whose topologies should be easy to infer from short sequences, rather than upon all possible quartets. (By contrast, other quartet based methods, such as Quartet Puzzling [47, 48], the Buneman tree construction [7], etc. infer quartet splits for all the possible quartets in the tree.) Basing the tree reconstruction upon properly selected sets of quartets makes it possible to expect, even from short sequences, that all the quartet splits inferred for the selected subset of quartets will be valid.

Since what we need is a set Q such that $R_T \subseteq Q \subseteq Q(T)$, we need to ensure that we pick a *large enough* set of quartets so that it contains all of R_T, and yet not too large that it contains any invalid quartet splits. Surprisingly, obtaining such a set Q is quite easy (once the sequences are long enough), and we describe a greedy approach which accomplishes this task. We will also show that the greedy approach can be implemented very efficiently, so that not too many calls to the DCTC algorithm need to be made in order to reconstruct the tree, and analyze the sequence length needed for the greedy approach to succeed with $1 - o(1)$ probability.

We now describe how this is accomplished.

Definition 2. [Q_w, and the *width* of a quartet]. The *width* of a quartet i, j, k, l is defined to be the maximum of $h^{ij}, h^{ik}, h^{il}, h^{jk}, h^{jl}, h^{kl}$, where h^{ij} denotes the dissimilarity score between sequences i and j (see Section 2). For each quartet whose width is at most w, compute all feasible splits on that quartet using the four-point method. Q_w is defined to be the set of all such reconstructed splits.

(We note that we could also compute the split for a given quartet of sequences in any number of ways, including maximum likelihood estimation, parsimony, etc., but we will not explore these options in this paper.)

For large enough values of w, Q_w will with high probability contain invalid quartet splits (unless the sequences are very long), while for very small values of w, Q_w will with high probability only contain valid quartet splits (unless the sequences are very short). Since our objective is a set of quartet splits Q such that $R_T \subseteq Q \subset Q(T)$, what we need is a set Q_w such that Q_w contains only valid quartet splits, and yet w is large enough so that all representative quartets are contained in Q_w as well.

We define sets

$$\mathscr{A} = \{w \in \{h^{ij}: 1 \le i, j \le n\}: R_T \subseteq Q_w\}, \tag{12}$$

and

$$\mathscr{B} = \{w \in \{h^{ij}: 1 \le i, j \le n\}: Q_w \subseteq Q(T)\}. \tag{13}$$

In other words, $\mathscr{A}$ is the set of widths w (drawn from the set of dissimilarity scores) which equal to exceed the largest width of any representative quartet, and $\mathscr{B}$ is the set of widths (drawn from the same set) such that all quartet splits of that dissimilarity score are correctly analyzed by the Four-Point Method.

It is clear that $\mathscr{B}$ is an initial segment in the list of widths, and that $\mathscr{A}$ is a final segment (these segments can be empty). It is easy to see that if $w \in \mathscr{A} \cap \mathscr{B}$, then DCTC($Q_w$) $= T$. Thus, if the sequences are long enough, we can apply DCTC to each of the $O(n^2)$ sets Q_w of splits, and hence reconstruct the tree properly. However, the sequences may not be long enough to ensure that such a w exists; i.e., $\mathscr{A} \cap \mathscr{B} = \emptyset$ is possible! Consequently, we will require that $\mathscr{A} \cap \mathscr{B} \ne \emptyset$, and state this requirement as an hypothesis (later, we will show in Theorem 9 that this hypothesis holds with high probability for sufficiently long sequences),

$$\mathscr{A} \cap \mathscr{B} \ne \emptyset. \tag{14}$$

When this hypothesis holds, we clearly have a polynomial time algorithm, but we can also show that the DCTC algorithm enables a binary search approach over the realized widths values, so that instead of $O(n^2)$ calls to the DCTC algorithm, we will have only $O(\log n)$ such calls.

Recall that DCTC(Q_w) is either a tree T, Inconsistent, or Insufficient.

- Insufficient. This indicates that w is too small, because not all representative quartet splits are present, and we should increase w.
- Tree output. If this happens, the quartets are consistent with a unique tree, and that tree is returned.
- Inconsistent. This indicates that the quartet splits are incompatible, so that no tree exists which is consistent with each of the constraints. In this case, we have computed the split of at least one quartet incorrectly. This indicates that w is too large, and we should decrease w.

If not all representative quartets are inferred correctly, then every set Q_w will be either insufficient or inconsistent with T, perhaps consistent with a different tree. In this case the sequences are too short for the DCM to reconstruct a tree accurately.

We summarize our discussion as follows:

Dyadic Closure Method.

Step 1. Compute the distance matrices d and h (recall that d is the matrix of corrected empirical distances, and h is the matrix of normalized Hamming distances, i.e., the *dissimilarity* score).

Step 2. Do a *binary search* as follows: for $w \in \{h^{ij}\}$, determine Q_w. If DCTC(Q_w) $= T$, for some tree T, then Return T. If DCTC returns *Inconsistent*, then w is too large; decrease w. If DCTC returns *Insufficient*, then w is too small; increase w.

Step 3. If for all w, DCTC applied to Q_w returns *Insufficient* or *Inconsistent*, then Return *Fail*.

We now show that this method accurately reconstructs the tree T if $\mathscr{A} \cap \mathscr{B} \neq \emptyset$ [i.e., if hypothesis (14) holds].

Theorem 7. *Let T be a fixed binary tree. The Dyadic Closure Method returns T if hypothesis* (14) *holds, and runs in $O(n^5 \log n)$ time on any input.*

Proof. If $w \in \mathscr{A} \cap \mathscr{B}$, then DCTC applied to Q_w returns the correct tree T by Theorem 6. Hypothesis (14) implies that $\mathscr{A} \cap \mathscr{B} \neq \emptyset$, hence the Dyadic Closure Method returns a tree if it examines any width in that intersection; hence, we need only prove that DCM either examines a width in that intersection, or else reconstructs the correct tree for some other width. This follows directly from Theorem 6.

The running time analysis is easy. Since we do a binary search, the DCTC algorithm is called at most $O(\log n)$ times. The dyadic closure phase of the DCTC algorithm costs $O(n^5)$ time, by Lemma 5, and reconstructing the tree T from $\mathrm{cl}(Q)$ uses at most $O(n^5)$ time using standard techniques. ■

Note that we have only guaranteed performance for DCM when $\mathscr{A} \cap \mathscr{B} \neq \emptyset$; indeed, when $\mathscr{A} \cap \mathscr{B} = \emptyset$, we have no guarantee that DCM will return the correct tree. In the following section, we discuss the ramifications of this requirement for accuracy, and show that the sequence length needed to guarantee that $\mathscr{A} \cap \mathscr{B} \neq \emptyset$ with high probability is actually not very large.

6. PERFORMANCE OF DYADIC CLOSURE METHOD FOR TREE RECONSTRUCTION UNDER THE NEYMAN 2-STATE MODEL

In this section we analyze the performance of a distance-based application of DCM to reconstruct trees under the Neyman 2-state model under two standard distributions.

6.1. Analysis of the Dyadic Closure Method

Our analysis of the Dyadic Closure Method has two parts. In the first part, we establish the probability that the estimation (using the Four-Point Method) of the split induced by a given quartet is correct. In the second part, we establish the probability that the greedy method we use contains all short quartets but no incorrectly analyzed quartet.

Our analysis of the performance of the DCM method depends heavily on the following two lemmas:

Lemma 4 [Azuma–Hoeffding inequality, see [3]]. *Suppose $X = (X_1, X_2, \ldots, X_k)$ are independent random variables taking values in any set S, and L: $S^k \to \mathbb{R}$ is any function that satisfies the condition: $|L(\mathbf{u}) - L(\mathbf{v})| \leq t$ whenever $\mathbf{u}$ and $\mathbf{v}$ differ at just* *one coordinate. Then,*

$$\mathbb{P}[L(\mathbf{X}) - \mathbb{E}[L(\mathbf{X})] \geq \lambda] \leq \exp\left(-\frac{\lambda^2}{2t^2k}\right),$$

$$\mathbb{P}[L(\mathbf{X}) - \mathbb{E}[L(\mathbf{X})] \leq -\lambda] \leq \exp\left(-\frac{\lambda^2}{2t^2k}\right). \qquad \blacksquare$$

We define the (standard) L_∞ metric on distance matrices, $L_\infty(d, d') = \max_{ij}|d_{ij} - d'_{ij}|$. The following discussion relies upon definitions and notations from Section 2.

Lemma 5. *Let T be an edge weighted binary tree with four leaves i, j, k, l, let D be the additive distance matrix on these four leaves defined by T, and let x be the weight on the single internal edge in T. Let d be an arbitrary distance matrix on the four leaves. Then the Four-Point Method infers the split induced by T from d if $L_\infty(d, D) < x/2$.*

Proof. Suppose that $L_\infty(d, D) < x/2$, and assume that T has the valid split $ij|kl$. Note that the four-point method will return a single quartet, split $ij|kl$ if and only if $d_{ij} + d_{kl} < \min\{d_{ik} + d_{jl}, d_{il} + d_{jk}\}$. Note that since $ij|kl$ is a valid quartet split in T, $D_{ij} + D_{kl} + 2x = D_{ik} + D_{jl} = D_{il} + D_{jk}$. Since $L_\infty(d, D) < x/2$, it follows that

$$d_{ij} + d_{kl} < D_{ij} + D_{kl} + x,$$
$$d_{ik} + d_{jl} > D_{ik} + D_{jl} - x,$$

and

$$d_{il} + d_{jk} > D_{il} + D_{jk} - x,$$

with the consequence that $d_{ij} + d_{kl}$ is the (unique) smallest of the three pairwise sums. $\blacksquare$

Recall that DCM applied to the Neyman 2-state model computes quartet splits using the four-point method (FPM).

Theorem 8. *Assume that z is a lower bound for the transition probability of any edge of a tree T in the Neyman 2-state model, $y \geq \max E^{ij}$ is an upper bound on the compound changing probability over all ij paths in a quartet q of T. The probability that FPM fails to return the correct quartet split on q from k sites is at most*

$$18 \exp \frac{-(1 - \sqrt{1 - 2z})^2 (1 - 2y)^2 k}{8}. \tag{15}$$

Proof. First observe from formula (1) that z is also a lower bound for the compound changing probability for the path connecting *any* two vertices of T. We know that FPM returns the appropriate subtree given the additive distances D_{ij}; furthermore, if $|d_{ij} - D_{ij}| \leq -\frac{1}{4}\log(1 - 2z)$ for all i, j, then FPM also returns the appropriate subtree on all *ijkl*, by Lemma 5. Consequently,

$$\mathbb{P}[\text{FPM errs}] \leq \mathbb{P}\left[\exists i, j\colon |D_{ij} - d_{ij}| > -\tfrac{1}{4}\log(1-2z)\right]. \tag{16}$$

Hence by (16), we have

$$\mathbb{P}[\text{FPM errs}] \leq \sum_{ij} \mathbb{P}\left[|D_{ij} - d_{ij}| > -\tfrac{1}{4}\log(1-2z)\right]. \tag{17}$$

For convenience, we drop the subscripts when we analyze the events in (17) and just write D and d; we write p for the corresponding transition probability E^{ij} and $\hat{p}$ for the relative frequency h^{ij}. By simple algebra,

$$|D - d| = \frac{1}{2}\log\frac{1-2p}{1-2\hat{p}}, \quad \text{if } p < \hat{p}, \tag{18}$$

$$|D - d| = \frac{1}{2}\log\frac{1-2\hat{p}}{1-2p}, \quad \text{if } p \geq \hat{p}. \tag{19}$$

Now we consider the probability that the Four-Point Method fails, i.e., the event estimated in (17). If $p \geq \hat{p}$, then formula (19) applies, so that $\mathbb{P}[\text{FPM errs}]$ is algebraically equivalent to

$$p - \hat{p} \geq \tfrac{1}{2}\left[(1-2z)^{-1/2} - 1\right](1-2p). \tag{20}$$

This can then be analyzed using Lemma 4. The other case is where $p < \hat{p}$. In this case, formula (18) applies, and $\mathbb{P}[\text{FPM errs}]$ is algebraically equivalent to

$$\frac{\hat{p}-p}{1-2\hat{p}} \geq \frac{1}{2}\left[(1-2z)^{-1/2} - 1\right]. \tag{21}$$

Select an arbitrary positive number ϵ. Then $\hat{p} - p \geq (1-2p)\epsilon$ with probability

$$\exp\frac{-\epsilon^2(1-2p)^2 k}{2}, \tag{22}$$

by Lemma 4. If $\hat{p} - p < (1-2p)\epsilon$, then

$$\frac{1}{1-2\hat{p}} < \frac{1}{(1-2p) - 2\epsilon(1-2p)} = \frac{1}{(1-2p)}\frac{1}{(1-2\epsilon)}.$$

Hence

$$\mathbb{P}\left[\frac{\hat{p}-p}{1-2\hat{p}} \geq \frac{1}{2}\left[(1-2z)^{-1/2} - 1\right]\right]$$

$$\leq \mathbb{P}\left[\frac{\hat{p}-p}{(1-2p)(1-2\epsilon)} \geq \frac{1}{2}\left[(1-2z)^{-1/2} - 1\right]\right] + \exp\frac{-\epsilon^2(1-2p)^2 k}{2}$$

$$\leq \exp\frac{-\epsilon^2(1-2p)^2 k}{2} \tag{23}$$

$$+ \exp\frac{-(1-2p)^2(1-2\epsilon)^2\left[(1-2z)^{-1/2} - 1\right]^2 k}{8}. \tag{24}$$

Note that $\epsilon = (\frac{1}{2})[1-(1-2z)^{1/2}]$ is the optimal choice. Formulae (22–24) contribute each the same exponential expression to the error, and (16) or (17) multiplies it by 6, due to the six pairs in the summation. ■

This allows us to state our main result. First, recall the definition of *depth* from Section 2.

Theorem 9. *Suppose k sites evolve under the Neyman 2-state model on a binary tree T, so that for all edges e, $p(e) \in [f, g]$, where we allow f, g to be functions of n. Then the dyadic closure method reconstructs T with probability* $1 - o(1)$, *if*

$$k > \frac{c \cdot \log n}{\left(1-\sqrt{1-2f}\right)^2 (1-2g)^{4\,\mathrm{depth}(T)+6}}, \tag{25}$$

where c is a fixed constant.

Proof. It suffices to show that hypothesis (14) holds. For k evolving sites (i.e., sequences of length k), and $\tau > 0$, let us define the following two sets, $S_\tau = \{\{i,j\}: h^{ij} < 0.5 - \tau\}$ and

$$Z_\tau = \left\{ q \in \binom{[n]}{4} : \text{for all } i,j \in q,\ \{i,j\} \in S_{2\tau} \right\},$$

and the following four events,

$$A = Q_{\mathrm{short}}(T) \subseteq Z_\tau, \tag{26}$$

$$B_q = \text{FPM correctly returns the split of the quartet } q \in \binom{[n]}{4}, \tag{27}$$

$$B = \bigcap_{q \in Z_\tau} B_q, \tag{28}$$

$$C = S_{2\tau} \text{ contains all pairs } \{i,j\} \text{ with } E^{ij} < 0.5 - 3\tau \text{ and no pair } \{i,j\} \text{ with } E^{ij} \geq 0.5 - \tau. \tag{29}$$

Thus, $\mathbb{P}[\mathscr{A} \cap \mathscr{B} \neq \varnothing] \geq \mathbb{P}[A \cap B]$. Define

$$\lambda = (1-2g)^{2\,\mathrm{depth}(T)+3}. \tag{30}$$

We claim that

$$\mathbb{P}[C] \geq 1 - (n^2 - n)e^{-\tau^2 k/2}, \tag{31}$$

and

$$\mathbb{P}[A|C] = 1, \quad \text{if } \tau \leq \frac{\lambda}{6}. \tag{32}$$

To establish (31), first note that h^{ij} satisfies the hypothesis of the Azuma–Hoeffding inequality (Lemma 4 with X_i the sequence of states for site i and $t = 1/k$).

Suppose $E^{ij} \geq .5 - \tau$. Then,

$$\begin{aligned}\mathbb{P}[\{i,j\} \in S_{2\tau}] &= \mathbb{P}[h^{ij} < 0.5 - 2\tau] \\ &\leq \mathbb{P}[h^{ij} - E^{ij} \leq 0.5 - 2\tau - E^{ij}] \leq \mathbb{P}\big[h^{ij} - \mathbb{E}[h^{ij}] \leq -\tau\big] \leq e^{-\tau^2 k/2}.\end{aligned}$$

Since there are at most $\binom{n}{2}$ pairs $\{i,j\}$, the probability that at least one pair $\{i,j\}$ with $E^{ij} \geq 0.5 - \tau$ lies in $S_{2\tau}$ is at most $\binom{n}{2}e^{-\tau^2 k/2}$. By a similar argument, the probability that $S_{2\tau}$ fails to contain a pair $\{i,j\}$ with $E^{ij} < 0.5 - 3\tau$ is also at most $\binom{n}{2}e^{-\tau^2 k/2}$. These two bounds establish (31).

We now establish (32). For $q \in R(T)$ and $i,j \in q$, if a path $e_1 e_2 \cdots e_t$ joins leaves i and j, then $t \leq 2\,\mathrm{depth}(T) + 3$ by the definition of $R(T)$. Using these facts, (1), and the bound $p(e) \leq g$, we obtain $E^{ij} = 0.5[1 - (1-2p_1)\cdots(1-2p_t)] \leq 0.5(1-\lambda)$. Consequently, $E^{ij} < 0.5 - 3\tau$ (by assumption that $\tau \leq \lambda/6$) and so $\{i,j\} \in S_{2\tau}$ once we condition on the occurrence of event C. This holds for all $i,j \in q$, so by definition of Z_τ we have $q \in Z_\tau$. This establishes (32).

Define a set,

$$X = \left\{ q \in \binom{[n]}{4} : \max\{E^{ij} : i,j \in q\} < 0.5 - \tau \right\},$$

(note that X is not a random variable, while Z_τ, S_τ are). Now, for $q \in X$, the induced subtree in T has mutation probability at least $f(n)$ on its central edge, and mutation probability of no more than $\max\{E^{ij}: i,j \in q\} < 0.5 - \tau$ on any pendant edge. Then, by Theorem 8 we have

$$\mathbb{P}[B_q] \geq 1 - 18\exp\frac{-\left(1 - \sqrt{1-2f}\right)^2 \tau^2 k}{8}. \tag{33}$$

whenever $q \in X$. Also, the occurrence of event C implies that

$$Z_\tau \subseteq X, \tag{34}$$

since if $q \in Z_\tau$, and $i,j \in q$, then $i,j \in S_{2\tau}$, and then (by event C), $E^{ij} < 0.5 - \tau$, hence $q \in X$. Thus, since $B = \bigcap_{q \in Z_\tau} B_q$, we have

$$\mathbb{P}[B \cap C] = \mathbb{P}\left[\left(\bigcap_{q \in Z_\tau} B_q\right) \cap C\right] \geq \mathbb{P}\left[\left(\bigcap_{q \in X} B_q\right) \cap C\right],$$

where the second inequality follows from (34), as this shows that when C occurs, $\bigcap_{q \in Z_\tau} B_q \supseteq \bigcap_{q \in X} B_q$. Invoking the Bonferonni inequality, we deduce that

$$\mathbb{P}[B \cap C] \geq 1 - \sum_{q \in X} \mathbb{P}[\overline{B_q}] - \mathbb{P}[\overline{C}]. \tag{35}$$

Thus, from above,

$$\mathbb{P}[A \cap B] \geq \mathbb{P}[A \cap B \cap C] = P[B \cap C],$$

(since $\mathbb{P}[A|C] = 1$), and so, by (33) and (35),

$$\mathbb{P}[A \cap B] \geq 1 - 18\binom{n}{4}\exp\frac{-\left(1-\sqrt{1-2f}\right)^2 \tau^2 k}{8} - (n^2 - n)e^{-\tau^2 k/2}.$$

Formula (25) follows by an easy calculation. ∎

6.2. Distributions on Trees

In the previous section we provided an upper bound on the sequence length that suffices for the Dyadic Closure Method to achieve an accurate estimation with high probability, and this upper bound depends critically upon the *depth* of the tree. In this section, we determine the depth of a random tree under two simple models of random binary trees.

These models are the *uniform* model, in which each tree has the same probability, and the *Yule–Harding* model, studied in [2, 8, 27] (the definition of this model is given later in this section). This distribution is based upon a simple model of speciation, and results in "bushier" trees than the uniform model. The following results are needed to analyze the performance of our method on random binary trees.

Theorem 10.

- **(i)** *For a random semilabeled binary tree T with n leaves under the uniform model,* $\operatorname{depth}(T) \leq (2 + o(1))\log_2 \log_2(2n)$ *with probability* $1 - o(1)$.
- **(ii)** *For a random semilabeled binary tree T with n leaves under the Yule–Harding distribution, after suppressing the root,* $\operatorname{depth}(T) = (1 + o(1))\log_2 \log_2 n$ *with probability* $1 - o(1)$.

Proof. This proof relies upon the definition of an *edi-subtree*, which we now define. If (a, b) is an edge of a tree T, and we delete the edge (a, b) but not the endpoints a or b, then we create two subtrees, one containing the node a and one containing the node b. By rooting each of these subtrees at a (or b), we obtain an edge-deletion induced subtree, or "edi-subtree."

We now establish (i). Recall that the number of all semilabeled binary trees is $(2n-5)!!$ Now there is a unique (unlabeled) binary tree F on $2^t + 1$ leaves with the following description: one endpoint of an edge is identified with the degree 2 root of a complete binary tree with 2^t leaves. The number of semilabeled binary trees whose underlying topology is F is $(2^t + 1)!/2^{2^t - 1}$. This is fairly easy to check and this also follows from Burnside's lemma as applied to the action of the symmetric group on trees, as was first observed by [32] in this context. A rooted semilabeled binary forest is a forest on n labeled leaves, m trees, such that every tree is either a single leaf or a binary tree which is rooted at a vertex of degree 2. It was proved by Carter et al. [11] that the number of rooted semilabeled binary forests is

$$N(n, m) = \binom{2n - m - 1}{m - 1}(2n - 2m - 1)!!.$$

Now we apply the probabilistic method. We want to set a number t large enough, such that the total number of edi-subtrees of depth at least t in the set of all semilabeled binary trees on n vertices is $o((2n-5)!!)$. The theorem then follows for this number t. We show that some $t=(2+o(1))\log_2\log_2(2n)$ suffices. We count ordered pairs in two ways, as usual: Let E_t denote the number of edi-subtrees of depth at least t (edi-subtrees induced by internal edges and leaf edges combined) counted over of all semilabeled trees. Then E_t is equal to the number of ways to construct a rooted semilabeled binary forest on n leaves of 2^t+1 trees, then use the 2^t+1 trees as leaf set to create all F-shaped semilabeled trees (as described above), with finally attaching the leaves of F to the roots of the elements of the forest. Then $E_t=((2^t+1)!/2^{2^t-1})N(n,2^t+1)$. Hence everything boils down to finding a t for which

$$\frac{(2^t+1)!}{2^{2^t-1}}\binom{2n-2^t-2}{2^t}(2n-2^{t+1}-3)!!=o((2n-5)!!).$$

Clearly $t=(2+\delta)\log_2\log_2(2n)$ suffices.

We now consider (ii). First we describe the proof for the usual rooted Yule−Harding trees. These trees are defined by the following construction procedure. Make a random permutation $\pi_1,\pi_2,\ldots,\pi_n$ of the n leaves, and join π_1 and π_2 by edges t a root R of degree 2. Add each of the remaining leaves sequentially, by randomly (with the uniform probability) selecting an edge incident to a leaf in the tree already constructed, subdividing the edge, and make π_i adjacent to the newly introduced node. For the depth of a Yule−Harding tree, consider the following recursive labeling of the edges of the tree. Call the edge π_iR (for $i=1,2$) "i new." When π_i is added ($i\geq 3$) by insertion into an edge with label "j new," we given label "i new" to the leaf edge added, give label "j new" to the leaf part of the subdivided edge, and turn the label "j new" into "j old" on the other part of the subdivided edge. Clearly, after l insertions, all numbers $1,2,\ldots,l$ occur exactly once with label new, in each occasion labeling leaf edges. The following which may help in understanding the labeling: edges with "old" label are exactly the internal edges and j is the smallest label in the subtree separated by an edge labeled "j old" from the root R, any time during the labeling procedure.

We now derive an upper bound for the probability that an edi-subtree of depth d develops. If it happens, then a leaf edge inserted at some point has to grow a deep edi-subtree on one side. Let us denote by T_i^R the rooted random tree that we already obtained with i leaves. Consider the probability that the most recently inserted edge i new ever defines an edi-subtree with depth d. Such an event can happen in two ways: this edi-subtree may emerge on the leaf side of the edge or on the tree side of the edge (these sides are defined when the edge is created). Let us denote these probabilities by $\mathbb{P}[i,\mathrm{OUT}|T_i^R]$ and $\mathbb{P}[i,\mathrm{IN}|T_i^R]$, since these probabilities may depend on the shape of the tree already obtained (and, in fact, the second probability does so depend on the shape of T_i^R). We estimate these quantities with tree-independent quantities.

For the moment, take for granted the following inequalities,

$$\mathbb{P}\left[i,\mathrm{OUT}|T_i^R\right]\leq\mathbb{P}\left[i,\mathrm{IN}|T_i^R\right],\tag{36}$$

$$\mathbb{P}\left[i,\mathrm{IN}|T_i^R\right]\leq\epsilon(d,n),\tag{37}$$

for some function $\epsilon(d, n)$ defined below. Clearly,

$$\mathbb{P}[\exists \text{ depth } d \text{ edi-subtree}] \leq \sum_{i=1}^{n} \sum_{T_i^R} \mathbb{P}\left[i, \mathrm{OUT}|T_i^R\right]\mathbb{P}\left[T_i^R\right] + \mathbb{P}\left[i, \mathrm{IN}|T_i^R\right]\mathbb{P}\left[T_i^R\right], \tag{38}$$

and using (36) and (37), (38) simplifies to

$$\mathbb{P}[\exists \text{ depth } d \text{ edi-subtree}] \leq 2n\epsilon(d, n). \tag{39}$$

We now find an appropriate $\epsilon(d, n)$.

For convenience we assume that $2^s = n - 2$, since it simplifies the calculations. Set $k = 2^{d-1} - 1$, it is clear that at least k properly placed insertions are needed to make the current edge "i new" have depth d on its tree side. Indeed, π_i was inserted into a leaf edge labeled "j new" and one side of this leaf edge is still a leaf, which has to develop into depth $d - 1$, and this development requires at least k new leaf insertions.

Focus now entirely on the k insertions that change "j new" into an edi-subtree of depth $d - 1$. Rank these insertions by $1, 2, \dots, k$ in order, and denote by 0 the original "j new" leaf edge. Then any insertion ranked $i \geq 1$ may go into one of those ranked $0, 1, \dots, i - 1$. Call the function which tells for $i = 1, 2, \dots, k$, which depth i is inserted into, a *core*. Clearly, the number of cores is at most k^k.

We now estimate the probability that a fixed core emerges. For any fixed $i_1 < i_2 < \cdots < i_k$, the probability that inserting π_{i_j} will make the insertion enumerated under depth j, for all $j = 1, 2, \dots, k$, is at most

$$\frac{1}{i_1 - 1} \cdot \frac{1}{i_2 - 1} \cdots \frac{1}{i_k - 1},$$

by independence. Summarizing our observations,

$$\begin{aligned} \mathbb{P}\left[i, \mathrm{IN}|T_i^R\right] &\leq k^k \sigma_{n-i}^k\left(\frac{1}{i}, \frac{1}{i+1}, \dots, \frac{1}{n-1}\right) \\ &\leq k^k \sigma_{n-2}^k\left(\frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{n-1}\right), \end{aligned} \tag{40}$$

where σ_m^k is the symmetric polynomial of m variables of degree k. We set $\epsilon(n, d) = \sigma_{n-2}^k(\frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{n-1})$. To estimate (40), observe that any term in $\sigma_{n-2}^k(\frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{n-1})$ can be described as having exactly a_i reciprocals of integers substituted from the interval $(2^{-(i+1)}, 2^{-i}]$. The point is that those reciprocals differ little in each of those intervals, and hence a close estimate is possible. A generic term of σ_{n-2}^k as described above is estimated from above by

$$2^{-(1 \cdot a_1 + 2 \cdot a_2 + \cdots + (s-1)a_{s-1})}. \tag{41}$$

Hence $\epsilon(n,d)$ is at most

$$\sum_{\substack{a_1+a_2+\cdots+a_{s-1}=k\\ a_i\le 2^i}} \binom{2}{a_1}\binom{4}{a_2}\binom{8}{a_3}\cdots\binom{2^{s-1}}{a_{s-1}} 2^{-(1\cdot a_1+2\cdot a_2+\cdots+(s-1)a_{s-1})}, \tag{42}$$

by (41). Since

$$\binom{2^i}{a_i}2^{-ia_i}\le \frac{1}{a_i!},$$

(42) is less than or equal

$$\sum_{\substack{a_1+a_2+\cdots+a_{s-1}=k\\ a_i\le 2^i}} \frac{1}{a_1!a_2!\cdots a_{s-1}!}. \tag{43}$$

Observe that the number of terms in (43) is at most the number of compositions of k into $s-1$ terms,

$$\binom{k+s-2}{s-2}.$$

The product of factorials is minimized (irrespective of $a_i\le 2^i$) if all a_is are taken equal. Hence, setting $k=s^{1+\delta}$ for any fixed $\delta>0$, (43) is at most

$$\left(\frac{(k+s-2)^{s-2}}{(s-2)!}\right)\Big/\left(\left(\frac{k}{s-1}\right)!^k\right),$$

and hence

$$\epsilon(n,d)\le k^k\left(\frac{(k+s-2)^{s-2}}{(s-2)!}\right)\Big/\left(\left(\frac{k}{s-1}\right)!^k\right)\le n^{-c\log n\log\log n},$$

and (39) goes to zero. For the depth d, our calculation yields $(1+\delta+o(1))\log_2\log_2 n$ with probability $1-o(1)$.

We leave the establishment of (36) to the reader. Now, to obtain a similar result for unrooted Yule–Harding trees, just repeat the argument above, but use the unrooted T_i instead of the rooted T_i^R. The probability of any T_i is the sum of probabilities of $2i-3$ rooted T_i^Rs, since the root could have been on every edge of T_i. Hence formula (37) has to be changed for $\mathbb{P}[i,\mathrm{IN}|T_i]\le(2n-3)\epsilon(d,n)$. With this change the same proof goes through, and the threshold does not change. ■

6.3. The Performance of Dyadic Closure Method and Two Other Distance Methods for Inferring Trees in the Neyman 2-State Model

In this section we describe the convergence rate for the DCM method, and compare it briefly to the rates for two other distance-based methods, the Agarwala et al. 3-approximation algorithm [1] for the L_∞ nearest tree, and neighbor-joining

[40]. We make the natural assumption that all methods use the same corrected empirical distances from Neyman 2-state model trees.

The neighbor-joining method is perhaps the most popular distance-based method used in phylogenetic reconstruction, and in many simulation studies (see [33, 34, 41] for an entry into this literature) it seems to outperform other popular distance based methods. The Agarwala et al. algorithm [1] is a distance-based method which provides a 3-approximation to the L_∞ nearest tree problem, so that it is one of the few methods which provide a provable performance guarantee with respect to any relevant optimization criterion. Thus, these two methods are two of the most promising distance-based methods against which to compare our method. Both these methods use polynomial time.

In [23], Farach and Kannan analyzed the performance of the 3-approximation algorithm with respect to tree reconstruction in the Neyman 2-state model, and proved that the Agarwala et al. algorithm converged quickly for the *variational distance* (a related but different concern). Recently, Kannan [35] extended the analysis and obtained the following counterpart to (25): If T is a Neyman 2-state model tree with mutation rates in the range $[f, g]$, and if sequences of length k' are generated on this tree, where

$$k' > \frac{c' \cdot \log n}{f^2(1-2g)^{2\,\mathrm{diam}(T)}}, \tag{44}$$

for an appropriate constant c', and were diam(T) denotes the "diameter" of T, then with probability $1 - o(1)$ the result of applying Agarwala et al. to corrected distances will be a tree with the same topology as the model tree. In [5], Atteson proved an identical statement for neighbor-joining, though with a different constant (the proved constant for neighbor-joining is smaller than the proved constant for the Agarwala et al. algorithm).

Comparing this formula to (25), we note that the comparison of depth and diameter is the issue, since $(1-\sqrt{1-2f})^2 = \Theta(f^2)$ for small f. It is easy to see that diam(T) $\geq 2\,$depth(T) for binary trees T, but the diameter of a tree can in fact be quite large (up to $n - 1$), while the depth is never more than $\log n$. Thus, for every fixed range of mutation probabilities, the sequence length that suffices to guarantee accuracy for the neighbor-joining or Agarwala et al. algorithms can be quite large (i.e., it can grow exponentially in the number of leaves), while the sequence length that suffices for the Dyadic Closure Method will never grow more than polynomially. See also [20, 21, 39] for further studies on the sequence length requirements of these methods.

The following table summarizes the worst case analysis of the sequence length that suffices for the dyadic closure method to obtain an accurate estimation of the tree, for a fixed and a variable range of mutation probabilities. We express these sequence lengths as functions of the number n of leaves, and use results from (25) and Section 6.2 on the depth of random binary trees. "Best case" (respectively, "worst case") trees refers to best case (respectively worst case) *shape* with respect to the sequence length needed to recover the tree as a function of the number n of leaves. Best case trees for DCM are those whose depth is small with respect to the number of leaves; these are the *caterpillar* trees, i.e., trees which are formed by

TABLE 1 Sequence Length Needed by Dyadic Closure Method to Return Trees under the Neyman 2-State Model

	Range of Mutation Probabilities on Edges:	
	$[f, g]$ f, g are constants	$\left[\frac{1}{\log n}, \frac{\log\log n}{\log n}\right]$
Worst case trees	polynomial	polylog
Best case trees	logarithmic	polylog
Random (uniform) trees	polylog	polylog
Random (Yule–Harding) trees	polylog	polylog

attaching n leaves to a long path. Worst case trees for DCM are those trees whose depth is large with respect to the number of leaves; these are the *complete binary trees*. All trees are assumed to be binary.

One has to keep in mind that comparison of performance guarantees for algorithms do not substitute for comparison of performances. Unfortunately, no analysis is available yet on the performance of the Agarwala et al. and neighbor-joining algorithms on random trees, therefore we had to use their worst case estimates also for the case of random leaves.

7. SUMMARY

We have provided upper and lower bounds on the sequence length k for accurate tree reconstruction, and have shown that in certain cases these two bounds are surprisingly close in their order of growth with n. It is quite possible that even better upper bounds could be obtained by a tighter analysis of our DCM approach, or perhaps by analyzing other methods.

Our results may provide a nice analytical explanation for some of the surprising results of recent simulation studies (see, for example, [30]) which found that trees on hundreds of species could be accurately reconstructed from sequences of only a few thousand sites long. For molecular biology the results of this paper may be viewed, optimistically, as suggesting that large trees can be reconstructed accurately from realistic length sequences. Nevertheless, some caution is required, since the evolution of real sequences will only be approximately described by these models, and the presence of very short and/or very long edges will call for longer sequence lengths.

ACKNOWLEDGMENTS

Thanks are due to Sampath Kannan for extending the analysis of [23] to consider the topology estimation, and to David Bryant and Éva Czabarka for proofreading the manuscript.

Tandy Warnow was supported by an NSF Young Investigator Award CCR-9457800, a David and Lucille Packard Foundation fellowship, and generous research support from the Penn Research Foundation and Paul Angello. Michael Steel was supported by the New Zealand Marsden Fund and the New Zealand Ministry of Research, Science and Technology. Péter L. Erdős was supported in part by the Hungarian National Science Fund contracts T 016 358. László Székely was supported by the National Science Foundation grant DMS 9701211, the Hungarian National Science Fund contracts T 016 358 and T 019 367, and European Communities (Cooperation in Science and Technology with Central and Eastern European Countries) contract ERBCIPACT 930 113. This research started in 1995 when the authors enjoyed the hospitality of DIMACS during the Special Year for Mathematical Support to Molecular Biology, and was completed in 1997 while enjoying the hospitality of Andreas Dress, at Universität Bielefeld, in Germany.

REFERENCES

[1] R. Agarwala, V. Bafna, M. Farach, B. Narayanan, M. Paterson, and M. Thorup, On the approximability of numerical taxonomy: fitting distances by tree metrics, Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms, 1996, pp. 365–372.

[2] D.J. Aldous, “Probability distributions on cladograms,” Discrete random structures, IMA Vol. in Mathematics and its Applications, Vol. 76, D.J. Aldous and R. Permantle (Editors), Springer-Verlag, Berlin/New York, 1995, pp. 1–18.

[3] N. Alon and J.H. Spencer, The probabilistic method, Wiley, New York, 1992.

[4] A. Ambainis, R. Desper, M. Farach, and S. Kannan, Nearly tight bounds on the learnability of evolution, Proc of the 1998 Foundations of Comp Sci, to appear.

[5] K. Atteson, The performance of neighbor-joining algorithms of phylogeny reconstruction, Proc COCOON 1997, Computing and Combinatorics, Third Annual International Conference, Shanghai, China, Aug. 1997, Lecture Notes in Computer Science, Vol. 1276, Springer-Verlag, Berlin/New York, pp. 101–110.

[6] H.-J. Bandelt and A. Dress, Reconstructing the shape of a tree from observed dissimilarity data, Adv Appl Math 7 (1986), 309–343.

[7] V. Berry and O. Gascuel, Inferring evolutionary trees with strong combinatorial evidence, Proc COCOON 1997, Computing and Combinatorics, Third Annual International Conference, Shanghai, China, Aug. 1997, Lecture Notes in Computer Science, Vol. 1276, Springer-Verlag, Berlin/New York, pp. 111–123.

[8] J.K.M. Brown, Probabilities of evolutionary trees, Syst Biol 43 (1994), 78–91.

[9] D.J. Bryant and M.A. Steel, Extension operations on sets of leaf-labelled trees, Adv Appl Math 16 (1995), 425–453.

[10] P. Buneman, “The recovery of trees from measures of dissimilarity,” Mathematics in the archaeological and historical sciences, F.R. Hodson, D.G. Kendall, P. Tatu (Editors), Edinburgh Univ. Press, Edinburgh, 1971, pp. 387–395.

[11] M. Carter, M. Hendy, D. Penny, L.A. Székely, and N.C. Wormald, On the distribution of lengths of evolutionary trees, SIAM J Disc Math 3 (1990), 38–47.

[12] J.A. Cavender, Taxonomy with confidence, Math Biosci 40 (1978), 271–280.

[13] J.T. Chang and J.A. Hartigan, Reconstruction of evolutionary trees from pairwise distributions on current species, Computing Science and Statistics: Proc 23rd Symp on the Interface, 1991, pp. 254–257.

[14] H. Colonius and H.H. Schultze, Tree structure for proximity data, British J Math Stat Psychol 34 (1981), 167–180.

[15] W.H.E. Day, Computational complexity of inferring phylogenies from dissimilarities matrices, Inform Process Lett 30 (1989), 215–220.

[16] W.H.E. Day and D. Sankoff, Computational complexity of inferring phylogenies by compatibility, Syst Zoology 35 (1986), 224–229.

[17] M.C.H. Dekker, Reconstruction methods for derivation trees, Master's Thesis, Vrije Universiteit, Amsterdam, 1986.

[18] P. Erdős and A. Rényi, On a classical problem in probability theory, Magy Tud Akad Mat Kutató Int Közl 6 (1961), 215–220.

[19] P.L. Erdős, M.A. Steel, L.A. Székely, and T. Warnow, Local quartet splits of a binary tree infer all quartet splits via one dyadic inference rule, Comput Artif Intell 16(2) (1997), 217–227.

[20] P.L. Erdős, M.A. Steel, L.A. Székely, and T. Warnow, "Inferring big trees from short quartets," ICALP'97, 24th International Colloquium on Automata, Languages, and Programming (Silver Jubilee of EATCS), Bologna, Italy, July 7–11, 1997, Lecture Notes in Computer Science, Vol. 1256, Springer-Verlag, Berlin/New York, 1997, 1–11.

[21] P.L. Erdős, M.A. Steel, L.A. Székely, and T. Warnow, A few logs suffice to build (almost) all trees-II, Theoret Comput Sci special issue on selected papers from ICALP 1997, to appear.

[22] P.L. Erdős, K. Rice, M. Steel, L. Szekely, and T. Warnow, The short quartet method, Mathematical Modeling and Scientific Computing, to appear.

[23] M. Farach and S. Kannan, Efficient algorithms for inverting evolution, Proc ACM Symp on the Foundations of Computer Science, 1996, pp. 230–236.

[24] M. Farach, S. Kannan, and T. Warnow, A robust model for inferring optimal evolutionary trees, Algorithmica 13 (1995), 155–179.

[25] J.S. Farris, A probability model for inferring evolutionary trees, Syst Zoology 22 (1973), 250–256.

[26] J. Felsenstein, Cases in which parsimony or compatibility methods will be positively misleading, Syst Zoology 27 (1978), 401–410.

[27] E.F. Harding, The probabilities of rooted tree shapes generated by random bifurcation, Adv Appl Probab 3 (1971), 44–77.

[28] M.D. Hendy, The relationship between simple evolutionary tree models and observable sequence data, Syst Zoology 38(4) (1989), 310–321.

[29] D. Hillis, Approaches for assessing phylogenetic accuracy, Syst Biol 44 (1995), 3–16.

[30] D. Hillis, Inferring complex phylogenies, Nature 383(12) (Sept. 1996), 130–131.

[31] D. Hillis, J. Huelsenbeck, and D. Swofford, Hobgoblin of phylogenetics? Nature 369 (1994), 363–364.

[32] M. Hendy, C. Little, and D. Penny, Comparing trees with pendant vertices labelled, SIAM J Appl Math 44 (1984), 1054–1065.

[33] J. Huelsenbeck, Performance of phylogenetic methods in simulation, Syst Biol 44 (1995), 17–48.

[34] J.P. Huelsenbeck and D. Hillis, Success of phylogenetic methods in the four-taxon case, Syst Biol 42 (1993), 247–264.

[35] S. Kannan, personal communication.

[36] M. Kimura, Estimation of evolutionary distances between homologous nucleotide sequences, Proc Nat Acad Sci USA 78 (1981), 454–458.

[37] J. Neyman, "Molecular studies of evolution: a source of novel statistical problems," Statistical decision theory and related topics, S.S. Gupta and J. Yackel (Editors), Academic Press, New York, 1971, pp. 1–27.

[38] H. Philippe and E. Douzery, The pitfalls of molecular phylogeny based on four species, as illustrated by the cetacea/artiodactyla relationships, J Mammal Evol 2 (1994), 133–152.

[39] K. Rice and T. Warnow, "Parsimony is hard to beat!," Proc COCOON 1997, Computing and combinatorics, Third Annual International Conference, Shanghai, China, Aug. 1997, Lecture Notes in Computer Science, Vol. 1276, Springer-Verlag, Berlin/New York, pp. 124–133.

[40] N. Saitou and M. Nei, The neighbor-joining method: A new method for reconstructing phylogenetic trees, Mol Biol Evol 4 (1987), 406–425.

[41] N. Saitou and T. Imanishi, Relative efficiencies of the Fitch–Mzargoliash, maximum parsimony, maximum likelihood, minimum evolution, and neighbor-joining methods of phylogenetic tree construction in obtaining the correct tree, Mol Biol Evol 6 (1989), 514–525.

[42] Y.S. Smolensky, A method for linear recording of graphs, USSR Comput Math Phys 2 (1969), 396–397.

[43] M.A. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, J Classification 9 (1992), 91–116.

[44] M.A. Steel, Recovering a tree from the leaf colourations it generates under a Markov model, Appl Math Lett 7 (1994), 19–24.

[45] M.A. Steel, L.A. Székely, and P.L. Erdős, The number of nucleotide sites needed to accurately reconstruct large evolutionary trees, DIMACS Technical Report No. 96-19.

[46] M.A. Steel, L.A. Székely, and M.D. Hendy, Reconstructing trees when sequence sites evolve at variable rates, J Comput Biol 1 (1994), 153–163.

[47] K. Strimmer and A. von Haeseler, Quartet puzzling: a quartet maximum likelihood method for reconstructing tree topologies, Mol Biol Evol 13 (1996), 964–969.

[48] K. Strimmer, N. Goldman, and A. von Haeseler, Bayesian probabilities and quartet puzzling, Mol Biol Evol 14 (1997), 210–211.

[49] D.L. Swofford, G.J. Olsen, P.J. Waddell, and D.M. Hillis, "Phylogenetic inference," Molecular systematics, D.M. Hillis, C. Moritz, and B.K. Mable (Editors), Chap. 11, 2nd ed., Sinauer Associates, Inc., Sunderland, 1996, pp. 407–514.

[50] N. Takezaki and M. Nei, Inconsistency of the maximum parsimony method when the rate of nucleotide substitution is constant, J Mol Evol 39 (1994), 210–218.

[51] T. Warnow, Combinatorial algorithms for constructing phylogenetic trees, Ph.D. thesis, University of California-Berkeley, 1991.

[52] P. Winkler, personal communication.

[53] K.A. Zaretsky, Reconstruction of a tree from the distances between its pendant vertices, Uspekhi Math Nauk (Russian Math Surveys), 20 (1965), 90–92 (in Russian).

[54] A. Zharkikh and W.H. Li, Inconsistency of the maximum-parsimony method: The case of five taxa with a molecular clock, Syst Biol 42 (1993), 113–125.

[55] S.J. Wilson, Measuring inconsistency in phylogenetic trees, J Theoret Biol 190 (1998), 15–36.

ELSEVIER

Theoretical Computer Science 221 (1999) 77–118

Theoretical Computer Science

www.elsevier.com/locate/tcs

A few logs suffice to build (almost) all trees: Part II

Péter L. Erdős[a,*], Michael A. Steel[b], László A. Székely[c], Tandy J. Warnow[d]

[a] *Mathematical Institute of the Hungarian Academy of Sciences, P.O.Box 127,1364 Budapest, Hungary*
[b] *Biomathematics Research Centre, University of Canterbury, Christchurch, New Zealand*
[c] *Department of Mathematics, University of South Carolina, Columbia, SC, USA*
[d] *Department of Computer and Information Science University of Pennsylvania, Philadelphia, PA, USA*

Abstract

Inferring evolutionary trees is an interesting and important problem in biology, but one that is computationally difficult as most associated optimization problems are NP-hard. Although many methods are provably statistically consistent (i.e. the probability of recovering the correct tree converges to 1 as the sequence length increases), the actual rate of convergence for different methods has not been well understood. In a recent paper we introduced a new method for reconstructing evolutionary trees called the dyadic closure method (DCM), and we showed that DCM has a very fast convergence rate. DCM runs in $O(n^5 \log n)$ time, where n is the number of sequences, and so, although polynomial, the computational requirements are potentially too large to be of use in practice. In this paper we present another tree reconstruction method, the witness–antiwitness method (WAM). WAM is faster than DCM, especially on random trees, and converges to the true tree topology at the same rate as DCM. We also compare WAM to other methods used to reconstruct trees, including Neighbor Joining (possibly the most popular method among molecular biologists), and new methods introduced in the computer science literature. © 1999 Published by Elsevier Science B.V. All rights reserved.

Keywords: Phylogeny; Evolutionary tree reconstruction; Distance-based methods; Quartet methods; Short quartet methods; Dyadic closure method; Witness–antiwitness method

1. Introduction

Rooted leaf-labelled trees are a convenient way to represent historical relationships between extant objects, particularly in evolutionary biology (where such trees are called

[*] Corresponding author.

E-mail addresses: elp@math-inst.hu (P.L. Erdős), m.steel@math.canterbury.ac.nz. (M.A. Steel), laszlo@math.sc.edu. (L.A. Székely), tandy@central.cis.upenn.edu. (T.J. Warnow)

PII: S0304-3975(99)00028-6

"phylogenies"). Molecular techniques have recently provided large amounts of sequence (DNA, RNA, or amino-acid) data that are being used to reconstruct such trees. Statistically based methods construct trees from sequence data, by exploiting the variation in the sequences due to random mutations that have occurred. A typical assumption made by these tree construction methods is that the evolutionary process operates through "point mutations", where the positions, or "sites", within the sequences mutate down the tree. Thus, by modelling how the different sites evolve down the tree, the entire mutational process on the sequences can be described. A further assumption that is typically made is that the evolutionary processes governing each site are identical, and independent (i.i.d.). For such models of evolution, some tree construction methods are guaranteed to recover the underlying *unrooted* tree from adequately long sequences generated by the tree, with arbitrarily high probability.

There are two basic types of tree reconstruction methods: *sequence-based methods* and *distance-based methods*. Distance-based methods for tree reconstruction have two steps. In the first step, the input sequences are represented by an $n \times n$ matrix d of pairwise dissimilarities (these may or may not observe the triangle inequality, and hence may not be truly "distances"). In the second step, the method M computes an additive matrix $M(d)$ (that is, an $n \times n$ distance matrix which exactly fits an edge-weighted tree) from the pairwise dissimilarity matrix, d. Distance methods are typically polynomial time. Sequence-based methods, on the other hand, do not represent the relationship between the sequences as a distance matrix; instead, these methods typically attempt to solve NP-hard optimization problems based upon the original sequence data, and are computationally intensive. See [26] for further information on phylogenetic methods in general.

A tree reconstruction method, whether sequence-based or distance-based, is considered to be accurate with respect to the topology prediction if the tree associated (uniquely) with the computed additive matrix has the same unrooted topology as the tree used to generate the observed sequences. A method is said to be *statistically consistent* for a model tree T if the probability of recovering the topology of T from sequences generated randomly on T converges to 1 as the sequence length increases to infinity. It has long been understood that most distance-based methods are statistically consistent methods for inferring trees under models of evolution in which the sites evolve i.i.d., but that some sequence-based methods (notably, the optimization problem *maximum parsimony* [25]) are not statistically consistent on all trees under these models. For this reason, some biologists prefer to use distance-based methods. However, not much is known, even experimentally, about the sequence length a given distance-based method needs for exact topological accuracy with high probability. How long the sequences have to be to guarantee high probability of recovering the tree depends on the reconstruction method, the details of the model, and the number n of species. Determining bounds on that length and its growth with n has become more pressing since biologists have begun to reconstruct trees on increasingly larger numbers of species (often up to several hundred) from such sequences.

In a previous paper [20], we addressed this question for trees under the Neyman 2-state model of site evolution, and obtained the following results:

1. We established a lower bound of $\log n$ on the sequence length that every method, randomized or deterministic, requires in order to reconstruct any given n-leaf tree in any 2-state model of sequence evolution,
2. We showed that the maximum compatibility method of phylogenetic tree construction requires sequences of length *at least* $n \log n$ to obtain the tree with high probability, and
3. We presented a new polynomial time method (the *dyadic closure method* (DCM)) for reconstructing trees in the Neyman 2-state model, and showed that polylogarithmic length sequences suffice for accurate tree reconstruction with probability near one on almost all trees, and polynomial length sequence length always suffices for any tree under reasonable assumptions on mutation probabilities.

Thus, the DCM [20] has a very fast convergence rate, which on almost all trees is within a polynomial of our established lower bound of $\log n$ for any method. However, although DCM uses only polynomial time, it has large computational requirements (it has $\Omega(n^2k + n^5 \log n)$ running time, and uses $O(n^4)$ space), where k is the sequence length. This may make it infeasible for reconstructing large trees.

In this paper, we present the *witness–antiwitness method* (WAM), a new and faster quartet-based method for tree reconstruction, which has the same asymptotic convergence rate as the DCM. The running time of WAM has a worst-case bound $O(n^2k + n^4 \log n \log k)$ where k is the sequence length, and is even faster under some reasonable restrictions on the model (see Theorem 12 for details). Thus, WAM is a faster algorithm than DCM, and has essentially the same convergence rate to the true tree topology as DCM. The *provable* bounds on the running time of WAM depend heavily on the depth of the model tree. We introduced the "depth" in [20] and showed that $depth(T)$ is bounded from above by $\log n$ for all binary trees T, and that random trees have depths bounded by $O(\log \log n)$.

In addition to presenting the new method, we present a framework for a comparative analysis of the convergence rates of different distance based methods. We apply this technique to several different methods, *neighbor joining* [43], the Agarwala et al. [1] "single-pivot" algorithm and its variant [21], the "double-pivot" algorithm, and the naive quartet method (a method we describe in this paper). We obtain *upper bounds* on the sequence lengths that suffice for accuracy for these distance-based methods, and show that these upper bounds grow exponentially in the *weighted diameter* of the tree, which is the maximum number of expected mutations for a random site on any leaf-to-leaf path in the tree. We analyze the weighted diameter of random trees under two distributions. We show that the diameter of random trees is $\Omega(\sqrt{n})$ under the uniform distribution, and $\Omega(\log n)$ under the Yule–Harding distribution. Consequently, these upper bounds on the sequence lengths that suffice for accuracy for these other distance-based methods are significantly larger than the upper bounds obtained for DCM and WAM. We note that our upper bounds for the algorithms in [1, 21] match those given by Sampath Kannan (personal communication). Finally, we generalize our methods and results to more general Markov models, and find the same relative performance (these results should be compared to those of Ambainis et al. in [4]). (While this framework provides a comparison between the convergence rates of these methods, it is limited by the fact that these are *upper bounds* on the sequence lengths that suffice for accuracy for these distance methods. These upper bounds may be loose, but no better upper bounds on these methods are yet known, to our knowledge. Obtaining better bounds on the convergence rates of these and other methods is an important open question.)

The structure of the paper is as follows. In Section 2 we provide definitions and discuss tree reconstruction methods in general. In Section 3, we describe the analytical framework for deriving upper bounds on the sequence lengths needed by different methods for exact accuracy in tree reconstruction, and we use this framework to provide an initial comparison between various distance-based methods. In Section 4, we describe the witness–antiwitness tree construction algorithm (WATC), and in Section 5, we describe the witness–antiwitness method (WAM) in full. In Section 6, we analyze the performance of WAM for reconstructing trees under the Neyman model of site evolution, and compare its performance to other promising distance-based methods. We extend the analysis of WAM to reconstructing trees under the general r-state Markov model in Section 7. Finally, in Section 8, we disucss the applicability of our results to biological data.

2. Definitions

Notation. $\mathbb{P}[A]$ denotes the probability of event A; $\mathbb{E}[X]$ denotes the expectation of random variable X. We denote the natural logarithm by log. The set $[n]$ denotes $\{1,2,\ldots,n\}$ and for any set S, $\binom{S}{k}$ denotes the collection of subsets of S of size k. $\mathbb{R}$ denotes the real numbers.

Definition. (I) *Trees.* We will represent a phylogenetic tree T by a *semi-labelled* tree whose *leaves* (vertices of degree one) are labelled by extant species, numbered by $1,2,\ldots,n$, and whose remaining internal vertices (representing ancestral species) are unlabelled. We will adopt the biological convention that phylogenetic trees are *binary*, meaning that all internal nodes have degree three, and we will also assume that T is *unrooted* (this is due to scientific and technical reasons which indicate that the location of the root can be either difficult or impossible to determine from data). We let $B(n)$ denote the set of all $(2n-5)!!=(2n-5)(2n-7)\cdots 3\cdot 1$ semi-labelled binary trees on the leaf set $[n]$.

The path between vertices u and v in the tree is called the *uv path*, and is denoted $P(u,v)$. The *topological distance* $L(u,v)$ between vertices u and v in a tree T is the number of edges in $P(u,v)$. The edge set of the tree is denoted by $E(T)$. Any edge adjacent to a leaf is called a *leaf edge*, any other edge is called an *internal edge*. For a phylogenetic tree T and $S\subseteq[n]$, there is a unique minimal subtree of T, containing all elements of S. We call this tree the *subtree* of T induced by S, and denote it by

$T_{|S}$. We obtain the *contracted subtree induced by S*, denoted by $T^*_{|S}$, if we substitute edges for all maximal paths of $T_{|S}$ in which every internal vertex has degree two. We denote by $ij|kl$ the tree on four leaves i,j,k,l in which the pair i,j is separated from the pair k,l by an internal edge. When the contracted subtree of T induced by leaves i,j,k,l is the tree $ij|kl$, we call $ij|kl$ a *valid quartet split* of T on the quartet of leaves $\{i,j,k,l\}$. Since all trees are assumed to be binary, all contracted subtrees (including, in particular, the quartet subtrees) are also binary. Consequently, the set $Q(T)$ of valid quartet splits for a binary tree T has cardinality $\binom{n}{4}$.

(II) *Sites.* Consider a set C of character states (such as $C=\{A,C,G,T\}$ for DNA sequences; $C=\{\text{the 20 amino acids}\}$ for protein sequences; $C=\{R,Y\}$ or $\{0,1\}$ for purine–pyrimidine sequences). A *sequence of length k* is an ordered k-tuple from C – that is, an element of C^k. A collection of n such sequences – one for each species labelled from $[n]$ – is called a *collection of aligned sequences.*

Aligned sequences have a convenient alternative description as follows. Place the aligned sequences as rows of an $n\times k$ matrix, and call *site i* the ith column of this matrix. A *pattern* is one of the $|C|^n$ possible columns.

(III) *Site substitution models.* Many models have been proposed to describe the evolution of sites as a stochastic process. Such models depend on the underlying phylogenetic tree T and some randomness. Most models assume that the sites are independently and identically distributed (i.i.d.).

The models on which we test our algorithm also assume the Markov property that the random assignment of a character state to a vertex v is determined by the character state of its immediate ancestor, and a random substitution on the connecting edge. In the most general stochastic model that we study, the sequence sites evolve i.i.d. according to the general Markov model from the root [47]. We now briefly discuss this general Markov model. Since the i.i.d. condition is assumed, it is enough to consider the evolution of a single site in the sequences. Substitutions (point mutations) at a site are generally modelled by a probability distribution π on a set of $r>1$ character states at the root ρ of the tree (an arbitrary vertex or a subdividing point on an edge), and each edge e oriented out from the root has an associated $r\times r$ stochastic transition matrix $M(e)$. The random character state at the root "evolves" down the tree – thereby assigning characters randomly to the vertices, from the root down to the leaves. For each edge $e=(u,v)$, with u between v and the root, $(M(e))_{\alpha\beta}$ is the probability that v has character state β given that u has character state α.

(IV) *The Neyman model.* The simplest stochastic model is a symmetric model for binary characters due to Neyman [40], and was also developed independently by Cavender [12] and Farris [24]. Let $\{0,1\}$ denote the two states. The root is a fixed leaf, the distribution π at the root is uniform. For each edge e of T we have an associated *mutation probability*, which lies strictly between 0 and 0.5. Let $p:E(T)\to(0,0.5)$ denote the associated map. We have an instance of the general Markov model with $M(e)_{01}=M(e)_{10}=p(e)$. We will call this the *Neyman 2-state model*, but note that it has also been called the Cavender–Farris model, and is equivalent to the Jukes–Cantor model when restricted to two states.

The Neyman 2-state model is hereditary on subsets of the leaves – that is, if we select a subset S of $[n]$, and form the subtree $T_{|S}$, then eliminate vertices of degree two, we can define mutation probabilities on the edges of $T^*_{|S}$ so that the probability distribution on the patterns on S is the same as the marginal of the distribution on patterns provided by the original tree T. Furthermore, the mutation probabilities that we assign to an edge of $T^*_{|S}$ is just the probability p that the endpoints of the associated path in the original tree T are in different states.

Lemma 1. *The probability p that the endpoints of a path P of topological length k are in different states is related to the mutation probabilities $p_1, p_2, \ldots, p_k$ of edges of P as follows*:

$$p = \frac{1}{2}\left(1 - \prod_{i=1}^{k}(1 - 2p_i)\right).$$

Lemma 1 is folklore and is easy to prove by induction.

(V) *Distances.* Any symmetric matrix, which is zero-diagonal and positive off-diagonal, will be called a *distance matrix*. (These "distances", however, may not satisfy the triangle inequality, because the distance corrections used in phylogenetics, and described below, do not always satisfy the triangle inequality. Since it is nevertheless the practice in systematics to refer to these quantities as "distances", we will do so here as well.) An $n \times n$ distance matrix D_{ij} is called *additive*, if there exists an n-leaf tree (not necessarily binary) with positive edge lengths on the internal edges and non-negative edge lengths on the leaf edges, so that D_{ij} equals the sum of edge lengths in the tree along the $P(i,j)$ path connecting leaves i and j. In [10], Buneman showed that the following four-point condition characterizes additive matrices (see also [45, 64]):

Theorem 1 (Four-point condition). *A matrix D is additive if and only if for all i, j, k, l (not necessarily distinct), the maximum of $D_{ij}+D_{kl}$, $D_{ik}+D_{jl}$, $D_{il}+D_{jk}$ is not unique. The tree with positive lengths on internal edges and non-negative lengths on leaf edges representing the additive distance matrix is unique among the trees without vertices of degree two.*

Given a pair of parameters (T, p) for the Neyman 2-state model, and sequences of length k generated by the model, let $H(i,j)$ denote the *Hamming distance* of sequences i and j and $h^{ij} = H(i,j)/k$ denote the *dissimilarity score* of sequences i and j. The *empirical corrected distance* between i and j is denoted by

$$d_{ij} = -\tfrac{1}{2}\log(1 - 2h^{ij}). \qquad (1)$$

The probability of a change in the state of any fixed character between the sequences i and j is denoted by $E^{ij} = \mathbb{E}(h^{ij})$, and we let

$$D_{ij} = -\tfrac{1}{2}\log(1 - 2E^{ij}) \qquad (2)$$

denote the *corrected model distance* between i and j. We assign to any edge e a positive length

$$l(e) = -\tfrac{1}{2}\log(1-2p(e)). \tag{3}$$

By Lemma 1, D_{ij} is the sum of the lengths (see previous equation) along the path $P(i,j)$ between i and j, and hence D_{ij} is an additive distance matrix. Furthermore, d_{ij} converges in probability to D_{ij} as the sequence length tends to infinity. These mathematical facts also have significance in biology, since under certain continuous time Markov models [48], which may be used to justify our models, $l(e)$ and D_{ij} are the expected number of back-and-forth state changes along edges and paths, respectively. A similar phenomenon and hence a similar distance correction exists for the general stochastic model [47], and is discussed in detail in Section 7.

(VI) *Tree reconstruction.* A *phylogenetic tree reconstruction method* is a function Φ that associates either a tree or the statement *Fail* to every collection of aligned sequences, the latter indicating that the method is unable to make such a selection for the data given.

According to the practice in systematic biology (see, for example, [31, 32, 52]), a method is considered to be *accurate* if it recovers the unrooted binary tree T, even if it does not provide any estimate of the mutation probabilities. A necessary condition for accuracy, under the models discussed above, is that two distinct trees, T, T', do not produce the same distribution of patterns no matter how the trees are rooted, and no matter what their underlying Markov parameters are. This "identifiability" condition is violated under an extension of the i.i.d. Markov model when there is an unknown distribution of rates across sites as described by Steel et al. [49]. However, it is shown in [47] (see also [13]) that the identifiability condition holds for the i.i.d model under the weak conditions that the components of π are not zero and, for each edge e, the determinant $\det(M(e)) \neq 0, 1, -1$, and in fact we can recover the underlying tree from the expected frequencies of patterns on just *pairs* of species.

Theorem 1 and the discussion that follows it suggest that appropriate methods applied to corrected distances will recover the correct tree topology from sufficiently long sequences. Consequently, one approach (which is guaranteed to yield a *statistically consistent* estimate) to reconstructing trees from distances is to seek an additive distance matrix of minimum distance (with respect to some metric on distance matrices) from the input distance matrix. Many metrics have been considered, but all resultant optimization problems have been shown or are assumed to be NP-hard (see [1, 17, 23] for results on such problems).

(VII) *Specific tree construction algorithms.* In this paper, we will be particularly interested in certain distance methods, the *four-point method* (FPM), the *naive method, neighbor joining*, and the *Agarwala* et al. algorithm. We now describe these methods.

Four-Point Method (FPM). Given a 4×4 distance matrix d, return the split $ij|kl$ which satisfies $d_{ij} + d_{kl} < \min\{d_{ik} + d_{jl}, d_{il} + d_{jk}\}$. If there is no such split, return *Fail*.

FPM is a not truly a tree reconstruction method, because it can only be applied to datsets of size four. We include it here, because it is a subroutine in the Naive Method, which we now describe.

The *Naive Method* uses the four-point method to infer a split for every quartet i,j,k,l. Thus, if the matrix is additive, the four-point method can be used to detect the valid quartet split on every quartet of vertices, and then standard algorithms [6, 14] can be used to reconstruct the tree from the set of splits. Note that the naive method is guaranteed to be accurate when the input distance matrix is *additive*, but it will also be accurate even for non-additive distance matrices under conditions which we will describe later (see Section 3). Most quartet-based methods (see, for example, [7, 50, 51]) begin in the same way, constructing a split for every quartet, and then accommodate possible inconsistencies using some technique specific to the method; the naive method, by contrast, only returns a tree if all inferred splits are consistent with that tree. The obvious optimization problem (find a maximum number of quartets which are simultaneously realizable) is of unknown computational complexity.

The *Agarwala et al.* algorithm [1] is a 3-approximation algorithm for the nearest tree with respect to the L_∞-metric, where $L_\infty(A,B) = \max_{ij} |A_{ij} - B_{ij}|$. Given input d, the result of applying the Agarwala et al. algorithm to d is an additive distance matrix D such that $L_\infty(d,D) \leqslant 3L_\infty(d,D^{\text{opt}})$, where D^{opt} is an optimal solution.

The use of the Agarwala et al. algorithm for inferring trees has been studied in two papers (see [22] for a study of its use for inferring trees under the Neyman model, and [4] for a study of its use for inferring trees under the general Markov model). However, both [22, 4] consider the performance of the Agarwala et al. algorithm with respect to the *variational distance* metric. Optimizing with respect to this metric is related to – but distinct from – estimating the tree T, since it is concerned as well with the mutational parameters p.

The *neighbor joining* method [43] is a method for reconstructing trees from distance matrices, which is based upon agglomerative clustering. It is possibly the most popular method among molecular biologists for reconstructing trees, and does surprisingly well in some experimental studies; see, for example, [34, 35].

All these methods are known to be statistically consistent for inferring trees both under the Neyman 2-state model and under the general r-state Markov model of site evolution.

3. A framework for the comparison of distance-based methods

Although it is understood that all reasonable distance-based methods will converge on the true tree given sequences of adequate length, understanding the rate of convergence (as a function of sequence length) to the true topology is more complicated. However, it is possible sometimes to compare different distance-based methods, without reference to the underlying model. The purpose of this section is to provide a framework for an explicit comparison among different distance-based methods. We will use this technique to compare the 3-approximation algorithm of Agarwala et al. to the *Naive method.* Our analysis of these two algorithms shows that on any distance matrix for which the first algorithm is guaranteed to reconstruct the true tree, so is the naive method. Since our new method, WAM, is guaranteed to reconstruct the true tree on any dataset for which the naive method is also guaranteed to reconstruct the true tree, this analysis also establishes a comparison between the Agarwala et al. algorithm and WAM.

By the four-point condition (Theorem 1) every additive distance matrix corresponds to a unique tree without vertices of degree 2, and with positive internal edge lengths, and non-negative lengths on edges incident with leaves.

Suppose we have a binary model tree T with positively weighted internal edges. Let x be the minimum edge-weight among internal edges, and let D be the associated additive distance matrix. Let d be an observed distance matrix, and let $\Delta = L_\infty(d,D)$.

For every distance-based reconstruction method Φ, we seek a constant $c(\Phi)$ such that

$$c(\Phi) = \sup\{c\colon\ \Delta < cx \Rightarrow \Phi(d) \text{ yields } T\}.$$

Lemma 2. (i) *Two additive distance matrices D and D' define the same topology if and only if for all quartets the relative orders of the pairwise sums of distances for that quartet are identical in the two matrices.*

(ii) *For every edge-weighted binary tree T with minimum internal edge weight x, and any $\vartheta > 0$, there is a different binary tree T' such that $L_\infty(D,D') = x/2 + \vartheta$, where D' is the additive distance matrix for T'.*

(iii) *Given any $n \times n$ distance matrix d, four indices i,j,k,l in $[n]$, let p_{ijkl} denote the difference between the maximum and the median of the three pairwise sums, $d_{ij} + d_{kl}$, $d_{ik} + d_{jl}$, $d_{il} + d_{jk}$. Let P be the maximum of the p_{ijkl} over all quartets i,j,k,l. Then there is no additive distance matrix D such that $L_\infty(d,D) < P/4$.*

Proof. Claim (i) is a direct consequence of the four-point condition (Theorem 1).

To prove (ii), for a given T, contract an internal edge e having minimum edge weight x, obtaining a non-binary tree T'. T' has exactly one vertex adjacent to four edges. Add $x/4$ to the weight of each of the four edges. Insert a new edge of weight ϑ to resolve the vertex of degree four, so that we obtain a binary tree T'', different from T. Let D be the additive distance matrix for T and let D'' be the additive distance matrix for T''. It is easy to see that then $L_\infty(D,D') = x/2 + \vartheta$.

For the proof of (iii), let D be an additive distance matrix with $L_\infty(d,D) = \varepsilon < t/4$. For all quartets i,j,k,l, the median and the maximum of the three pairwise sums induced by i,j,k,l are identical in D. Now consider the quartet i,j,k,l for which $p_{ijkl} = t$. The maximum and the median of the three pairwise sums in d differ by p_{ijkl}. In order for the maximum and median of the three pairwise sums to be equal in D, at least one pairwise distance must change by at least $p_{ijkl}/4$. However $\varepsilon < p_{ijkl}/4$, contradicting the assumption. □

Theorem 2. *Let D be an additive $n \times n$ distance matrix defining a binary tree T, d be a fixed distance matrix, and let $\delta = L_\infty(d,D)$. Assume that x is the minimum weight of internal edges of T in the edge weighting corresponding to D.*

(i) *A hypothetical exact algorithm for the L_∞-nearest tree is guaranteed to return the topology of T from d if $\delta < x/4$.*

(ii) (a) *The 3-approximation algorithm for the L_∞-nearest tree is guaranteed to return the topology of T from d if $\delta < x/8$.* (b) *For all n there exists at least one d with $\delta = x/6$ for which the method can err.* (c) *If $\delta \geqslant x/4$, the algorithm can err for every such d.*

(iii) *The naive method is guaranteed to return the topology of T from d if $\delta < x/2$, and there exists a d for any $\delta > x/2$ for which the method can err.*

Proof. To prove (i), assume that D^* is an additive distance matrix with $L_\infty(d,D^*) \leqslant \delta$, and let T^* denote the tree topology corresponding to D^*. According to Lemma 2, Part (i), D^* and D define the same tree iff the relative order of pairwise sums of distances agree for all quartets in the two matrices. We will prove that D^* and D define the same tree topology by contradiction.

So suppose D^* and D do not define the same tree topology. Then there is a quartet, i,j,k,l, of leaves, where (without loss of generality) the topology induced by matrix D is $ij|kl$ and the topology induced by matrix D^* is $ik|jl$. Thus, there exist positive constants P and ε so that $2P + D_{ij} + D_{kl} = D_{ik} + D_{jl}$ and $D^*_{ij} + D^*_{kl} = D^*_{ik} + D^*_{jl} + 2\varepsilon$. Now $P \geqslant x$, since P is an internal path length in T. By the triangle inequality we have

$$L_\infty(D,D^*) \leqslant 2\delta. \tag{4}$$

We have

$$2P + 2\varepsilon = D_{ik} + D_{jl} - D_{ij} - D_{kl} + D^*_{ij} + D^*_{kl} - D^*_{ik} - D^*_{jl} \tag{5}$$

and hence by the triangle inequality

$$2x < 2P + 2\varepsilon \leqslant 8\delta. \tag{6}$$

Since $\delta < x/4$, this implies that such a quartet i,j,k,l does not exist, and so D and D^* define the same tree topology.

To prove (ii)(a), let D^* denote the output of the 3-approximation algorithm and T^* denote the corresponding tree. Following similar arguments, $L_\infty(d,D^*) \leqslant 3\delta$, so that corresponding to formula (4) we have $L_\infty(D,D^*) \leqslant 4\delta$, and corresponding to formula (6) we have $2x < 16\delta$. To prove (ii)(b), we now give an example where the 3-approximation algorithm can fail in which $L_\infty(D,d) = x/6$. Let d be distance matrix defined by $d_{uv} = d_{wx} = 7/3$, $d_{uw} = d_{vx} = 3$ and $d_{ux} = d_{vw} = 10/3$. By item (iii) of Lemma 2, it follows that there is no additive distance matrix D with $L_\infty(d,D) < 1/6$. Now let D be the additive distance matrix induced by the binary tree T on leaves u,v,w,x with topology $uv|wx$ and with edge length as follows: the central edge in T has weight 1 and all other edges have weight 13/12. Then, $L_\infty(D,d) = 1/6$ so that D

is a closest additive distance matrix to d. Furthermore, $L_\infty(d,D)=x/6$, since $x=1$ is the lowest edge weight in T. However there is another additive distance matrix induced by a different tree which lies within 3 times this minimal distance. Namely, let D'' be the additive distance matrix induced by the binary tree with topology $uw|vx$ with interior edge weighted 1/3 and other edges weighted 5/4. Then, $L_\infty(D'',d)=1/2=3L_\infty(D,d)=3\min_D\{L_\infty(D,d)\}$, as claimed. It is easy to see that this example can be embedded in any size distance matrix so that for all n such examples exist. For (ii)(c), suppose d is a distance matrix, D is its closest additive distance matrix, and x is the smallest weight of any edge in D. Then contract the edge e of weight x in T, the edge-weighted realization of D, and add $x/4$ to every edge originally incident to e. Let D' be the distance matrix of the new edge-weighted tree, T'. It follows that $L_\infty(D,D')=x/2$ and so that $L_\infty(d,D')\leqslant L_\infty(d,D)+L_\infty(D,D')$. If $L_\infty(d,D)=x/4$, then $L_\infty(d,D')\leqslant 3x/4$, by the triangle inequality. Hence the 3-approximation algorithm could return the topology of T or of T', and since they are different there is a possibility of making the wrong choice.

To prove (iii), arguments similar to the ones above obtain

$$2P+2\varepsilon=D_{ik}+D_{jl}-D_{ij}-D_{kl}+d_{ij}+d_{kl}-d_{ik}-d_{jl}$$

and $2x<2P+2\varepsilon\leqslant 4\delta$. The required example is in Lemma 2, Part (ii). □

In other words, *given any matrix d of corrected distances, if an exact algorithm for the L_∞-nearest tree can be guaranteed – by this analysis – to correctly reconstruct the topology of the model tree, then so can the Naive method.* This may suggest that there is an inherent limitation of the L_∞-nearest tree approach to reconstructing phylogenetic tree topologies. However, note that the analytical results are pessimistic; that is, they guarantee a high probability of an accurate performance once sequence lengths exceed some threshold, but do not guarantee a low probability of accurate performance for sequences below those lengths. Even so, these techniques are essentially the same ones that have been used in other studies to obtain analytical results regarding convergence to the true tree (see also [4, 22]).

4. The witness–antiwitness tree construction (WATC)

4.1. Introduction

In this section we describe the witness–antiwitness tree construction algorithm (WATC). This procedure, which is the heart of our witness–antiwitness method (WAM), solves certain restricted instances of the NP-complete quartet consistency problem [46], and solves them faster than the dyadic closure tree construction algorithm (DCTC) that we used as a procedure previously in our dyadic closure method (DCM) [20]. We therefore achieve an improvement with respect to computational requirements over DCM, and pay for it by requiring somewhat longer sequences.

Let e be an edge in T. Deleting e but not its endpoints creates two rooted subtrees, T_1 and T_2; these are called *edi-subtrees*, where "edi" stands for "*e*dge-*d*eletion-*i*nduced". Each edi-subtree having at least two leaves can be seen as being composed of two smaller edi-subtrees. The algorithm we will describe, the witness–antiwitness tree construction algorithm, or WATC, constructs the tree "from the outside in", by inferring larger and larger edi-subtrees, until the entire tree is defined. Thus, the algorithm has to decide at each iteration at least one pair of edi-subtrees to "join" into a new edi-subtree. In the tree, such pairs can be recognized by the constraints (a) that they are disjoint, and (b) that their roots are at distance two from each other. These pairs of edi-subtrees are then said to be "siblings". The algorithm determines whether a pair of edi-subtrees are siblings by using the quartet splits. We will show that if the set Q satisfies certain conditions then WATC is guaranteed to reconstruct the tree T from Q.

The conditions that Q must satisfy in order for WATC to be guaranteed to reconstruct the tree T are slightly more restrictive than those we required in the DCTC method, but do not require significantly longer sequences. Sets Q which satisfy these conditions are said to be *T-forcing*. The first stage of WATC assumes that Q is *T-forcing*, and on that basis attempts to reconstruct the tree T. If during the course of the algorithm it can be determined that Q is *not* T-forcing, then the algorithm returns *Fail*. Otherwise, a tree T' is constructed. At this point, the second stage of WATC begins, in which we determine whether T is the unique tree that is consistent with Q. If Q fails this test, then the algorithm returns *Fail*, and otherwise it returns T.

Just as in the dyadic closure method (DCM) we will need a search technique to find an appropriate set Q. Whereas binary search was a feasible technique for the DCM, it is no longer feasible in this case. Search techniques for an appropriate set Q are discussed in Section 5.

4.2. Definitions and preliminary material

Within each *edi*-subtree t, select that unique leaf which is the lowest valued leaf among those closest topologically to the root (recall that leaves are identified with positive integers). This is called the *representative* of t, and is denoted $rep(t)$. If the edi-subtree consists of a single leaf, then the representative leaf is identical with this single leaf, which also happens to be the root of the edi-subtree at the same time.

The *diameter* of the tree T, $diam(T)$, is the maximum topological distance in the tree between any pair of leaves. We define the depth of an edi-subtree t to be $L(root(t), rep(t))$, and denote this quantity by $depth(T)$. The *depth* of T is then $\max_t\{depth(t)\}$, as t ranges over all edi-subtrees yielded by internal edges of T. We say that a path P in the tree T is *short* if its topological length is at most $depth(T)+1$, and say that a quartet i,j,k,l is a *short quartet* if it induces a subtree which contains a single edge connected to four disjoint short paths. The set of all short quartets of the tree T is denoted by $Q_{\text{short}}(T)$. We will denote the set of valid quartet splits for the short quartets by $Q^*_{\text{short}}(T)$.

For each of the $n-3$ internal edges of the n-leaf binary tree T we assign a *representative quartet* $\{i,j,k,l\}$ as follows. The deletion of the internal edge *and* its endpoints defines four rooted subtrees. Pick the representative from each of these subtrees to obtain i,j,k,l; by definition, the quartet i,j,k,l is a *short quartet* in the tree. We call the split of this quartet a *representative quartet split* of T, and we denote the set of representative quartet splits of T by R_T. Note that by definition

$$R_T \subseteq Q^*_{\text{short}}(T) \subseteq Q(T). \tag{7}$$

We will say that a set Q of quartet splits is *consistent* with a tree T if $Q \subseteq Q(T)$. We will say that Q is *consistent* if there exists a tree T with which Q is consistent, and otherwise Q is said to be *inconsistent*. In [20], we proved:

Theorem 3. *Let T be a binary tree on $[n]$. If R_T is consistent with a binary tree T' on $[n]$, then $T = T'$. Therefore, if $R_T \subseteq Q$, then either Q is inconsistent, or Q is consistent with T. Furthermore, Q cannot be consistent with two distinct trees if $R_T \subseteq Q$.*

Let S be a set of n sequences generated under the Neyman model of evolution, and let d be the matrix of corrected empirical distances. Given any four sequences i,j,k,l from S, we define the *width* of the quartet on i,j,k,l to be $\max(d_{ij}, d_{ik}, d_{il}, d_{jk}, d_{jl}, d_{kl})$. For any $w \in \mathbb{R}^+$, let Q_w denote the set of quartet splits of width at most w, inferred using the four-point method.

4.3. The dyadic closure method

The dyadic closure method is based on the dyadic closure tree construction (DCTC) algorithm, which uses dyadic closure (see [20, 18]) to reconstruct a tree T consistent with an input set Q of quartet splits. Recall that $Q(T)$ denotes the set of all valid quartet splits in a tree T, and that given $Q(T)$, the tree T is uniquely defined. The *dyadic closure* of a set Q is denoted by $cl(Q)$, and consists of all splits that can be inferred by combining two splits at a time from Q, and from previously inferred quartet splits. In [20], we showed that the dyadic closure $cl(Q)$ could be computed in $O(n^5)$ time, and that if Q contained all the representative quartet splits of a tree, and contained only valid quartet splits, (i.e. if $R_T \subseteq Q \subseteq Q(T)$), then $cl(Q) = Q(T)$. Consequently, the DCTC algorithm reconstructs the tree T if $R_T \subseteq Q \subseteq Q(T)$. It is also easy to see that no set Q can simultaneously satisfy this condition for two distinct binary trees T, T', by Theorem 3, and furthermore, if Q satisfies this condition for T, it can be quickly verified that T is the unique solution to the reconstruction problem. Thus, when Q is such that for some binary tree T, $R_T \subseteq Q \subseteq Q(T)$, then the DCTC algorithm properly reconstructs T. The problem cases are when Q does not satisfy this condition for any T.

We handle the problem cases by specifying the output $DCTC(Q)$ to be as follows:

- *binary tree* T such that $cl(Q) = Q(T)$ (this type of output is guaranteed when $R_T \subseteq Q \subseteq Q(T)$),

- *inconsistent* when $cl(Q)$ contains two contradictory splits for the same quartet, or
- *insufficient* otherwise.

Note that this specification does not prohibit the algorithm from reconstructing a binary tree T, even if Q does not contain all of R_T. In such a case, the tree T will nevertheless satisfy $cl(Q)=Q(T)$; therefore, no other binary tree T' will satisfy $Q \subseteq Q(T')$). Note that if $DCTC(Q)=Inconsistent$, then $Q \not\subseteq Q(T)$ for any binary tree T, so that if $Q \subseteq Q'$ then $DCTC(Q')=Inconsistent$ as well. On the other hand, if $DCTC(Q)=Insufficient$ and $Q' \subseteq Q$, then $DCTC(Q')=Insufficient$ also. Thus, if $DCTC(Q)$ is Inconsistent, then there is no tree T consistent with Q, but if $DCTC(Q)$ is Insufficient, then it is still possible that some tree exists consistent with Q, but the set Q is *insufficient* with respect to the requirements of the DCTC method.

Now consider what happens if we let Q be Q_w the set of quartet splits based upon quartets of width at most w. The output of the DCTC algorithm will indicate whether w is too big (i.e. when $DCTC(Q_w)=Inconsistent$), or too small (i.e. when $DCTC(Q_w)=Insufficient$). Consequently, DCTC can be used as part of a tree construction method, where splits of quartets (of some specified width w) are estimated using some specified method, and we search through the possible widths w using binary search.

In [20], we studied a specific variant of this approach, called the Dyadic Closure Method (DCM), in which quartet trees are estimated using the four-point method (see Definition VII in Section 2). We analyzed the sequence length that suffices for accurate tree construction by DCM and showed that it grows very slowly; for almost all trees under two distributions on binary trees the sequence length that suffices for tree reconstruction under DCM is only polylogarithmic in n, once $0<f \leqslant g<.5$ are fixed and $p(e) \in [f,g]$ is assumed. Thus, DCM has a very fast convergence rate. DCM uses $O(n^2k + n^5 \log n)$ time and $O(n^4)$ space; therefore it is a statistically consistent polynomial time method for inferring trees under the Neyman model of evolution. For practical purposes, however, the computational requirements of the DCM method are excessive for inferring large trees, where n can be on the order of hundreds.

4.4. Witnesses, antiwitnesses, and T-forcing sets

Recall that the witness–antiwitness tree construction algorithm constructs T from the outside in, by determining in each iteration which pairs of edi-subtrees are siblings. This is accomplished by using the quartet splits to guide the inference of edi-subtrees. We now describe precisely how this is accomplished.

Definition 1. Recall that an *edi-subtree* is a subtree of T induced by the deletion of an edge in the tree. Two edi-subtrees are *siblings* if they are disjoint, the path between their roots contains exactly two edges, and there are at least two leaves not in either of these two edi-subtrees. (The last condition – that there are at least two leaves not in either of the two edi-subtrees – is nonstandard, but is assumed because it simplifies our discussion.) Let t_1 and t_2 be two vertex disjoint *edi*-subtrees. A *witness* *to the siblinghood of* t_1 *and* t_2 is a quartet split $uv|wx$ such that $u \in t_1$, $v \in t_2$, and $\{w,x\} \cap (t_1 \cup t_2) = \emptyset$. We call such quartets *witnesses*. An *anti-witness to the siblinghood of* t_1 *and* t_2 is a quartet split $pq|rs$, such that $p \in t_1$, $r \in t_2$, and $\{q,s\} \cap (t_1 \cup t_2) = \emptyset$. We will call these *anti-witnesses*.

Definition 2. Let T be a binary tree and Q a set of quartet splits defined on the leaves of T.

- Q has the *witness property for* T: Whenever t_1 and t_2 are sibling edi-subtrees of T and $T - t_1 - t_2$ has at least two leaves, then there is a quartet split of Q which is a witness to the siblinghood of t_1 and t_2.
- Q has the *antiwitness property for* T: Whenever there is a witness in Q to the siblinghood of two edi-subtrees t_1 and t_2 which are not siblings in T, then there is a quartet split in Q which is an antiwitness to the siblinghood of t_1 and t_2.

Theorem 4. *If* $R_T \subseteq Q$, *then* Q *has the witness property for* T. *Furthermore, if* $R_T \subseteq Q \subseteq Q(T)$, *and* t_1 *and* t_2 *are sibling edi-subtrees, then* Q *contains at least one witness, but no antiwitness, to the siblinghood of* t_1 *and* t_2.

The proof is straightforward, and is omitted.

Suppose T is a fixed binary tree, and Q is a set of quartet splits defined on the leaves of T. The problem of reconstructing T from Q is in general NP-hard [46], but in [20] we showed that if $R_T \subseteq Q \subseteq Q(T)$ we can reconstruct T in $O(n^5)$ time, and validate that T is the unique tree consistent with Q. Now we define a stronger property for Q which, when it holds, will allow us to reconstruct T from Q (and validate that T is the unique tree consistent with Q) in $O(n^2 + |Q| \log |Q|)$ time. Thus, this is a faster algorithm than the DCTC algorithm that we presented in [20].

Definition 3 (*T-forcing sets of quartet splits*). A set Q of quartet splits is said to be *T-forcing* if there exists a binary tree T such that

1. $R_T \subseteq Q \subseteq Q(T)$, and
2. Q has the antiwitness property for T.

Two points should be made about this definition. Since $R_T \subseteq Q$, Q has the *witness* property for T, and it is impossible for Q to be both T-forcing and T'-forcing for distinct T and T', since by Theorem 3, R_T is consistent with a unique tree. Finally, note that the first condition $R_T \subseteq Q \subseteq Q(T)$ was the requirement we made for the dyadic closure tree construction (DCTC) algorithm in [20], and so T-forcing sets of quartet splits have to satisfy the assumptions of the DCTC algorithm, plus one additional assumption: having the *antiwitness* property.

4.5. WATC

The algorithm we will now describe operates by constructing the tree from the outside in, via a sequence of iterations. Each iteration involves determining a new set of edi-subtrees, where each edi-subtree is either an edi-subtree in the previous iteration or is the result of making two edi-subtrees from the previous iteration siblings. Thus, each iteration involves determining which pairs of edi-subtrees from the previous iteration are siblings, and hence should be joined into one edi-subtree in this iteration.

We make the determination of siblinghood of edi-subtrees by applying the witness and antiwitness properties, but we note that only certain splits are considered to be relevant to this determination. In other words, we will require that any split used either as a witness or an anti-witness have leaves in four distinct edi-subtrees that exist at the time of the determination of siblinghood for this particular pair. Such splits are considered to be *active*, and other splits are considered to be *inactive*. All splits begin as active, but become inactive during the course of the algorithm (and once inactive, they remain inactive). We will use the terms "active witness" and "active antiwitness" to refer to active splits which are used as witnesses and antiwitesses. We will infer that two edi-subtrees are siblings if and only if there is an active witness to their siblinghood and no active anti-witness. (Note that this inference will be accurate if Q has the witness and antiwitness properties, but otherwise the algorithm may make a false inference, or fail to make any inference.)

We represent our determination of siblinghood as a graph on the edi-subtrees we have currently found. Thus, suppose at the beginning of the current iteration there are p edi-subtrees, $t_1, t_2, \ldots, t_p$. The graph for this iteration has p nodes, one for each edi-subtree, and we put an edge between every pair of edi-subtrees which have at least one witness and no anti-witness in the set of quartet topologies. The algorithm proceeds by then merging pairs of sibling edi-subtrees (recognized by edges in the graph) into a single (new) edi-subtree. The next iteration of the algorithm then requires that the graph is reconstructed, since witnesses and antiwitnesses must consist of four leaves, each drawn from distinct edi-subtrees (these are the *active* witnesses and antiwitnesses – thus, quartet splits begin as active, but can become inactive as edi-subtrees are merged).

The last iteration of the algorithm occurs when the number of edi-subtrees left is four, *or* there are no pairs of edi-subtrees which satisfy the conditions for siblinghood. If no pair of edi-subtrees satisfy the criteria for being siblings, then the algorithm returns *Fail*. On the other hand, if there are exactly four edi-subtrees, and if there are two disjoint pairs of sibling edi-subtrees, then we return the tree formed by merging each of the two pairs of sibling edi-subtrees into a single edi-subtree, and then joining the roots of these two (new) edi-subtrees by an edge.

If a tree T' is reconstructed by the algorithm, we will not return T' until we verify that

$$R_{T'} \subseteq Q \subseteq Q(T').$$

If the tree T' passes this test, then we return T', and in all other cases we return *Fail*.

We summarize this discussion in the following:

The WATC algorithm

Stage I:

- Start with every leaf of T defining an *edi*-subtree.

- While there are at least four *edi*-subtrees do:
  - ○ Form the graph G on vertex set given by the edi-subtrees, and with edge set defined by siblinghood; i.e., $(x,y) \in E(G)$ if and only if there is at least one witness and no antiwitness to the siblinghood of edi-subtrees x and y. All witnesses and antiwitnesses must be splits on four leaves in which each leaf lies in a distinct edi-subtree; these are the *active* witnesses and antiwitnesses.
    - – *Case: there are exactly four edi-subtrees*: Let the four subtrees be x,y,z,w. If the edge set of the graph G is $\{(x,y),(z,w)\}$, then construct the tree T formed by making the edi-subtrees x and y siblings, the edi-subtrees z and w siblings, and adding an edge between the roots of the two new edi-subtrees; else, return *Fail*.
    - – *Case: there are more than four edi-subtrees*: If the graph has at least one edge, then select one, say (x,y), and make the roots of the *edi*-subtrees x and y children of a common root r, and replace the pair x and y by one *edi*-subtree. If no component edge exists, then Return *Fail*.

Stage II

- Verify that T satisfies the constraints $R_T \subseteq Q \subseteq Q(T)$. If so, return T, and else return *Fail*.

The runtime of this algorithm depends upon how the two *edi*-subtrees are found that can be siblings.

4.6. Implementation of WATC

We describe here a fast implementation of the WATC algorithm.

We begin by constructing a multigraph on n nodes, bijectively labelled by the species. Edges in this multigraph will be colored either green or red, with one green edge between i and j for each witness to the siblinghood of i and j, and one red edge between i and j for each antiwitness. Thus, each quartet split $ij|kl$ defines six edges in the multigraph, with two green edges $((ij)$ and $(kl))$ and four red edges $((ik),(il),(jk),(jl))$. Each green edge is annotated with the quartet that defined it and the topology on that quartet, so that the other edges associated to that quartet can be identified. Constructing this multi-graph takes $O(|Q|)$ time. Note that *edi*-subtrees x and y are determined to be siblings if there exists a green edge (x,y) but no red edge (x,y).

We will maintain several data structures:

- $Red(i,j)$, the number of red edges between nodes i and j, so that accesses, increments, and decrements to $Red(i,j)$ take $O(1)$ time,
- $Green(i,j)$, the set of green edges between nodes i and j, maintained in such a way that we can enumerate the elements in $|Green(i,j)|$ time, and so that we can union two such sets in $O(1)$ time,
- T_i, the ith edi-subtree (i.e. the edi-subtree corresponding to node i), maintained as a directed graph with edges directed away from the root,
- $Tree$, an array such that $Tree[i]=j$ indicates that leaf i is in tree T_j. This is initialized by $Tree[i]=i$ for all i, and

- *Candidates*, the set of pairs of edi-subtrees which have at least one green edge and no red edges between them (and hence are candidates for siblinghood). We maintain this set using doubly-linked lists, and we also have pointers *into* the list from other datastructures ($Green(i,j)$) so that we can access, add, and delete elements from the set in O(1) time.

Finding a sibling pair: A pair of edi-subtrees are inferred to be siblings if and only if they have at least one green edges and no red edges between them. We maintain a list of possible sibling pairs of edi-subtrees in the set *Candidates*, and the members of *Candidates* are pairs of the form i,j where both i and j are edi-subtrees. (Testing whether i is a current edi-subtree is easy; just check that $Tree[i]=i$.) We take an element (i,j) from the set *Candidates* and verify that the pair is valid. This requires verifying that both i and j are current names for *edi*-subtrees, which can be accomplished by checking that $Tree[i]=i$ and $Tree[j]=j$. If (i,j) fails this test, we delete (i,j) from the set of *Candidates*, and examine instead a different pair. However, if (i,j) passes this test, we then verify that the pair i,j have at least one green edge and no red edges between them. For technical reasons (which we describe below), it is possible that $Green(i,j)$ will contain a *ghost* green edge. We now define what ghost green edges are, and how we can recognize them in O(1) time.

Definition 4. A *ghost* green edge is a green edge (a,b) which was defined by a quartet split $ab|cd$, but which was not deleted after the edi-subtrees containing c and d were merged into a single edi-subtree.

Detecting whether a green edge is a ghost is done as follows. Recall that every green edge (a,b) is annotated with the quartet (a,b,c,d) that gave rise to it. Therefore, given a green edge (a,b), we look up the edi-subtrees for the members of the *other* green edge (c,d) (using the *Tree* array), and see if c and d still belong to distinct edi-subtrees. If $Tree[c]=Tree[d]$ then (a,b) is a ghost green edge (since c and d were already placed in the same edi-subtree) and otherwise it is a true green edge.

Every ghost we find in $Green(i,j)$ we simply delete, and if $Green(i,j)$ contains only ghost edges, we remove (i,j) from the set *Candidates* (the edi-subtrees i and j are not actually siblings). If we find any non-ghost green edge in $Green(i,j)$, then (i,j) are inferred to be sibling edi-subtrees, and we enter the next phase.

Processing a sibling pair: Having found a pair i and j of edi-subtrees which are siblings, we need to update all the data-structures appropriately. We now describe how we do this.

First, we process every green edge e in $Green(i,j)$ by deleting the four red edges associated to e (this is accomplished by decrementing appropriate entries in the matrix *Red*). Note that we do not explicitly (or implicitly) delete the other green edge associated with edge e, and rather leave that green edge to be handled later; this is how *ghost* green edges arise.

After we finish processing every green edge, we merge the two edi-subtrees into one edi-subtree. We will use one index, say i, to indicate the number of the new *edi*-subtree created. We update T_i so that it has a new root, and the children of the new root are the roots of the previous edi-subtrees T_i and T_j, and we update the *Tree* array so that all entries which previously held a j now hold i.

We also have to reset $Red(i,k)$ and $Green(i,k)$ for every other edi-subtree k, since the edi-subtree labelled i has changed. We set $Red(i,k) = Red(i,k) + Red(j,k)$, and $Green(i,k) = Green(i,k) \cup Green(j,k)$ for all k. We then set $Red(j,k) = 0$ and $Green(j,k) = \emptyset$, if we wish (this is for safety, but is not really needed).

We also have to update the *Candidates* set. This involves deletions of some pairs, and insertions of others. The only pairs which need to be deleted are those i,k for which there is now a red edge between edi-subtrees i and k, but for which previously there was none. This can be observed during the course of updating the $Red(i,k)$ entries, since every pair (i,k) which should be deleted has $Red(i,k) = 0$ before the update, and $Red(i,k) > 0$ after the update. Pairs (i,k) which must be inserted in the *Candidates* set are those (i,k) which previously had $Green(i,k) = \emptyset$ and which now have $Green(i,k) \neq \emptyset$. Accessing, inserting, and deleting the elements of *Candidates* takes O(1) time each, so this takes O(1) additional time.

We now discuss the runtime analysis of the first stage of WATC:

Theorem 5. *The first stage of WATC uses* $O(n^2 + |Q|)$ *time.*

Proof. Creating the multi-graph clearly costs only $O(|Q|)$ time. Initializing all the datastructures takes $O(n^2)$ time. There are at most $O(|Q|)$ green edges in the multigraph we create, and each green edge is processed at most once, after which it is deleted. Processing a green edge costs O(1) time, since *Tree* can be accessed in O(1) time. There are at most $n - 1$ siblinghood detections, and updating the datastructures after detecting siblinghood only costs $O(n)$ time (beyond the cost of processing green edges). Implementing the datastructures $Green(i,j)$ and *Candidates* so that updates are efficient is easy through the use of pointers and records. Hence, the total cost of the first stage is $O(n^2 + |Q|)$. □

So suppose the result of the first phase constructs a tree T from the set Q of splits. The second stage of the WATC algorithm needs to verify that $R_T \subseteq Q \subseteq Q(T)$; we now describe how this is accomplished efficiently.

Given T, we can compute R_T in $O(n^2)$ time in a straightforward way: for each of the $O(n)$ edi-subtrees t, we compute the representative $rep(t)$ in $O(n)$ time. We then use the representatives to compute R_T, which has size $O(n)$, in $O(n)$ additional time. Verifying that $R_T \subseteq Q$ then takes at most $O(n \log n + |Q| \log |Q|)$ time. First we make sorted list of quartet splits by the lexicographic order of the 4 vertices involved. Sorting is in $O(|Q| \log |Q|)$ time. Then we use a binary search to determine membership, which costs $O(\log n)$ time for each element of R_T, since $|Q| = O(n^4)$. Verifying that $Q \subseteq Q(T)$ then can be done by verifying that $q \in Q(T)$ for each $q \in Q$. This is easily done in O(1) time per q using O(1) *lca* queries (to determine the valid split for each quartet which has a split in Q). Preprocessing T so that we can do *lca* queries in O(1) time per query can be done in $O(n)$ time, using the algorithm of Harel and Tarjan [53]. Consequently, we have proven:

Theorem 6. *The second stage of WATC takes* $O(n^2 + |Q| \log |Q|)$ *time. Therefore, WATC takes* $O(n^2 + |Q| \log |Q|)$ *time.*

4.7. *Proof of correctness of WATC*

We begin by proving that the WATC algorithm correctly reconstructs the tree T provided that Q is T-forcing.

Theorem 7. *If Q is T-forcing, then $WATC(Q) = T$.*

Proof. We first prove that all decisions made by the algorithm are correct, and then prove that the algorithm never fails to make a correct decision.

We use induction on the number of iterations to prove that no incorrect decisions are made by the algorithm. At the first iteration, every edi-subtree is a leaf, and these are correct. Now assume that so far the WATC algorithm applied to Q has constructed only correct edi-subtrees, and the next step merges two edi-subtrees, t_1 and t_2, into one, but that these are not actually siblings.

Since Q has the antiwitness property, there is a valid quartet split $ab|cd \in Q$ with $a \in t_1, c \in t_2$ and $\{b,d\} \cap (t_1 \cup t_2) = \emptyset$. We need only show that this antiwitness is still active at the time that we merged t_1 and t_2 into one edi-subtree.

Suppose that the split $ab|cd$ is not active at the time we merged t_1 and t_2. In this case, then the four leaves a,b,c,d are in fewer than four distinct edi-subtrees. The assumption $\{b,d\} \cap (t_1 \cup t_2) = \emptyset$ then implies that we have already created an edi-subtree t containing both b and d. This edi-subtree is true, since we have assumed all edi-subtrees constructed so far are accurate. Now, consider the edge e' whose deletion creates the subtree t. This edge cannot exist if $ab|cd$ is a valid quartet split and neither b nor d are in $t_1 \cup t_2$. Consequently, the antiwitness $ab|cd$ is *still* active at the time we merged t_1 and t_2, contradicting that we made that merger, and hence all inferred edi-subtrees are correct.

We now show that the algorithm never fails to be able to make a correct decision. If Q is T-forcing, then $R_T \subseteq Q$. Now if t and t' are sibling edi-subtrees, then let e be the edge in T whose deletion disconnects $t \cup t'$ from the rest of the tree T. Let q be the representative quartet split associated to e. This quartet split is a witness to the siblinghood of t and t', which will remain active throughout the iterations of the algorithm until the entire tree is constructed (otherwise there are only three edi-subtrees present at some point, and this is contradicted by the structure of the algorithm). Furthermore, since $Q \subseteq Q(T)$, there is no invalid quartet split, and consequently no antiwitness to the siblinghood of t and t'. Therefore, the algorithm will never fail to have opportunities to merge pairs of sibling edi-subtrees. □

Theorem 8. *If the WATC algorithm returns a tree T given a set Q of quartet splits, then Q is consistent with T and with no other tree T'. If WATC does not return a tree T, then Q is not T-forcing.*

Proof. The proof is not difficult. If T is returned by WATC, then Q satisfies $R_T \subseteq Q \subseteq Q(T)$. Under this condition Q is consistent with T and with no other tree, by Theorem 3. Hence the first assertion holds. For the second assertion, if Q is T-forcing, then by the previous theorem WATC returns T after the first stage. The conditions for being T-forcing include that $R_T \subseteq Q \subseteq Q(T)$, so that the verification step is successful, and Q is returned. □

5. The witness–antiwitness method (WAM)

In the previous section we described the WATC algorithm which reconstructs T given a T-forcing set of quartet splits, Q. In this section we describe a set of search strategies for finding such a set Q. These strategies vary in their number of queries on quartet split sets (ranging from $O(\log\log n)$ to $O(n^2)$), but also vary in the sequence length needed in order for the search strategy to be successful with high probability. All have the same asymptotic sequence length requirement as the dyadic closure method [20], but differ in terms of the multiplicative constant.

Before we describe and analyze these search strategies, we begin with some results on the four-point Method, and on random trees.

5.1. Previous results

Lemma 3 (Azuma–Hoeffding inequality, see [3]). *Suppose $X = (X_1, X_2, \dots, X_k)$ are independent random variables taking values in any set S, and $L : S^k \to \mathbb{R}$ is any function that satisfies the condition: $|L(\boldsymbol{u}) - L(\boldsymbol{v})| \leqslant t$ whenever $\boldsymbol{u}$ and $\boldsymbol{v}$ differ at just one coordinate. Then, for any $\lambda > 0$, we have*

$$\mathbb{P}[L(\boldsymbol{X}) - \mathbb{E}[L(\boldsymbol{X})] \geqslant \lambda] \leqslant \exp\left(-\frac{\lambda^2}{2t^2k}\right),$$

$$\mathbb{P}[L(\boldsymbol{X}) - \mathbb{E}[L(\boldsymbol{X}] \leqslant -\lambda] \leqslant \exp\left(-\frac{\lambda^2}{2t^2k}\right).$$

In [20], we proved:

Theorem 9. *Assume that z is a lower bound for the transition probability of any edge of a tree T in the Neyman 2-state model, $y \geqslant \max E^{ij}$ is an upper bound on the compound changing probability over all ij paths in a quartet q of T. The probability that FPM fails to return the correct quartet split on q is at most*

$$18\exp(-(1-\sqrt{1-2z})^2(1-2y)^2k/8)). \tag{8}$$

In [20] we also provided an upper bound on the growth of the depth of random trees under two distributions:

Theorem 10. (i) *For a random semilabelled binary tree T with n leaves under the uniform model, depth*$(T) \leqslant (2+\mathrm{o}(1))\log_2 \log_2(2n)$ *with probability* $1-\mathrm{o}(1)$.

(ii) *For a random semilabelled binary tree T with n leaves under the Yule-Harding distribution, after suppressing the root, depth*$(T) = (1+\mathrm{o}(1))\log_2 \log_2 n$ *with probability* $1-\mathrm{o}(1)$.

5.2. Search strategies

Let Q_w denote the set of splits inferred using the four-point method on quartets whose width is at most w; recall that the width of a quartet i,j,k,l is the maximum of $d_{ij}, d_{ik}, d_{il}, d_{jk}, d_{jl}, d_{kl}$. The objective is to find a set Q_w such that Q_w is T-forcing.

Definition 5.

$$\mathscr{A} = \{w \in \mathbb{R}^+ \colon R_T \subseteq Q_w\},$$

$$\mathscr{B} = \{w \in \mathbb{R}^+ \colon Q_w \subseteq Q(T)\}.$$

We now state without proof the following observation which is straightforward.

Observation 1. *$\mathscr{A}$ is either $\emptyset$, or is (w_A, ∞) for some positive real number w_A. $\mathscr{B}$ is either $\emptyset$, or is $(0, w_B)$, for some positive real number w_B.*

Sequential search for T-forcing Q_w: A sequential search through the sets Q_w, testing each Q_w for being T-forcing by a simple application of WATC algorithm, is an obvious solution to the problem of finding a T-forcing set which will find a T-forcing set from shorter sequences than any other search strategy through the sets Q_w. However, in the worst case, it examines $\mathrm{O}(n^2)$ sets Q_w, since w can be any of the values in $\{d_{ij}\colon 1 \leqslant i < j \leqslant n\}$, and hence it has high computational requirements.

Sparse-high search for a T-forcing Q_w: We describe here a sparse search that examines at most $\mathrm{O}(\log k)$ sets Q_w and hence has lower computational requirements, but may require longer sequences. Even so, we prove that the sequence length requirement has the same order of magnitude as the sequential search. This sparse search examines the high end of the values of w, and so we call it the *Sparse-high* search strategy.

Let $\tau < 1/4$ be given. We define Z_τ to be the set of quartets i,j,k,l such that $\max\{h^{ij}, h^{ik}, h^{il}, h^{jk}, h^{jl}, h^{kl}\} < 1/2 - 2\tau$. Note then that the set of splits (inferred using the four-point method) on quartets in Z_τ is $Q_{w(\tau)}$, where $w(\tau) = -\frac{1}{2}(\log(4\tau))$.

The sparse-high search examines $\tau = 1/8, 1/16, \ldots$, until it finds a τ such that $Z_\tau = Q_{w(\tau)}$ is T-forcing, or until $w(\tau)$ exceeds every d_{ij}.

We now define conditions under which each of these search strategies are guaranteed to find a T-forcing set Q_w. Recall the sets $\mathscr{A} = \{w\colon R_T \subseteq Q_w\}$,and $\mathscr{B} = \{w\colon Q_w \subseteq Q(T)\}$.

We now define the following *assumptions*:

$$\mathscr{A} \cap \mathscr{B} \neq \emptyset, \tag{9}$$

$$\exists w^* \in \mathscr{A} \cap \mathscr{B}, \text{ s.t. } Q_{w^*} \text{ has the antiwitness property}, \tag{10}$$

$$\exists \tau^*, \text{ s.t. } \forall \tau \in [\tau^*/2, \tau^*], w(\tau) \in \mathscr{A} \cap \mathscr{B}, \text{ and } Q_{w(\tau)} \text{ has the antiwitness property}. \tag{11}$$

It is clear that if assumptions (16) and (17) hold, then the sequential search strategy will be guaranteed to succeed in reconstructing the tree, and that the Sparse-high search strategy requires that assumption (11) hold as well.

We now analyze the sequence length needed to get each of these assumptions to hold with constant probability.

6. How WAM performs under the Neyman 2-state model

In this section we analyze the performance of the witness–antiwitness method (WAM), with respect to computational and sequence-length requirements. The analysis of the sequence length requirement follows a similar analysis for DCM in [20], but turns out to be more complicated, and results in constant times longer sequences. The analysis of the computational complexity of WAM is both in the worst case, and under the assumption that the tree topology is drawn from a random distribution. Finally, we compare the performance of WAM to other methods, with respect to both these issues.

6.1. Sequence length needed by WAM

Theorem 11. *Suppose k sites evolve under the Cavender–Farris model on a binary tree T, so that for all edges e, $p_e \in [f,g]$, where we allow $f = f(n)$ and $g = g(n)$ to be functions of n. We assume that* $\limsup_n g(n) < 1/2$. *Then both the sparse-high and sequential search based on the WATC algorithm returns the true tree T with probability* $1 - o(1)$, *if*

$$k > \frac{c \cdot \log n}{(1-\sqrt{1-2f})^2(1-2g)^{4\mathrm{depth}(T)}}, \tag{12}$$

where c is a fixed constant.

Proof. Note that the sparse-high search requires assumptions (16)–(18), while the sequential search only requires assumptions (16) and (17). We will show that the given sequence length suffices for all three assumptions to hold with probability $1 - o(1)$.

We begin by showing that assumption (9) holds, i.e. that $R_T \subseteq Q_w \subseteq Q(T)$ for some w.

For k evolving sites (i.e. sequences of length k), and fixed $\tau > 0$, let us define the following two sets:

$$S_\tau = \{\{i,j\}:\ h^{ij} < 0.5 - \tau\},$$

and

$$Z_\tau = \left\{q \in \binom{[n]}{4}: \text{ for all } i,j \in q, \{i,j\} \in S_{2\tau}\right\},$$

and the following four events:

$$A = Q_{\text{short}}(T) \subseteq Z_\tau, \tag{13}$$

$$B_q = \text{FPM correctly returns the split of the quartet } q \in \binom{[n]}{4}, \tag{14}$$

$$B = \bigcap_{q \in Z_\tau} B_q, \tag{15}$$

$$C = S_{2\tau} \text{ contains all } \{i,j\} \text{ with } E^{ij} < 0.5 - 3\tau \text{ and no } \{i,j\} \text{ with } E^{ij} \geqslant 0.5 - \tau. \tag{16}$$

Note that B is the event that $Q_{w(\tau)} \subseteq Q(T)$, so that $A \cap B$ is the event that $Q^*_{\text{short}} \subseteq Q_{w(\tau)} \subseteq Q(T)$, or $w(\tau) \in \mathscr{A} \cap \mathscr{B}$. Thus, $\mathbb{P}[\mathscr{A} \cap \mathscr{B} \neq \emptyset] \geqslant \mathbb{P}[A \cap B]$. Define

$$\lambda = (1 - 2g)^{2\text{depth}(T)+3}. \tag{17}$$

We claim that

$$\mathbb{P}[C] \geqslant 1 - (n^2 - n)\mathrm{e}^{-\tau^2 k/2} \tag{18}$$

and

$$\mathbb{P}[A|C] = 1 \quad \text{if } \tau \leqslant \lambda/6. \tag{19}$$

To establish (18), first note that h^{ij} satisfies the hypothesis of the Azuma–Hoeffding inequality (Lemma 3 with $X_l = 1$ if the lth bits of the sequences of leaves i and j differ, and $X_l = 0$ otherwise, and $t = 1/k$). Suppose $E^{ij} \geqslant 0.5 - \tau$. Then,

$$\begin{aligned}\mathbb{P}[\{i,j\} \in S_{2\tau}] &= \mathbb{P}[h^{ij} < 0.5 - 2\tau]\\ &\leqslant \mathbb{P}[h^{ij} - E^{ij} \leqslant 0.5 - 2\tau - E^{ij}] \leqslant \mathbb{P}[h^{ij} - \mathbb{E}[h^{ij}] \leqslant -\tau] \leqslant \mathrm{e}^{-\tau^2 k/2}.\end{aligned}$$

Since there are at most $\binom{n}{2}$ pairs $\{i,j\}$, the probability that at least one pair $\{i,j\}$ with $E^{ij} \geqslant 0.5 - \tau$ lies in $S_{2\tau}$ is at most $\binom{n}{2}\mathrm{e}^{-\tau^2 k/2}$. By a similar argument, the probability that $S_{2\tau}$ fails to contain a pair $\{i,j\}$ with $E^{ij} < 0.5 - 3\tau$ is also at most $\binom{n}{2}\mathrm{e}^{-\tau^2 k/2}$. These two bounds establish (18).

We now establish (19). For $q \in Q_{\text{short}}(T)$ and $i,j \in q$, if a path $e_1 e_2 \cdots e_t$ joins leaves i and j, then $t \leqslant 2\mathit{depth}(T) + 3$ by the definixtion of $Q_{\text{short}}(T)$. Using these facts, Lemma 1, and the bound $p_e \leqslant g$, we obtain $E^{ij} = 0.5\,[1-(1-2p_1)\cdots(1-2p_t)] \leqslant 0.5(1-\lambda)$. Consequently, $E^{ij} < 0.5 - 3\tau$ (by assumption that $\tau \leqslant \lambda/6$) and so $\{i,j\} \in S_{2\tau}$ once we condition on the occurrence of event C. This holds for all $i,j \in q$, so by definition of Z_τ we have $q \in Z_\tau$. This establishes (19).

Define a set

$$X = \left\{ q \in \binom{[n]}{4} \colon \max\{E^{ij}\colon i,j \in q\} < 0.5 - \tau \right\}$$

(note that X is not a random variable, while Z_τ, S_τ are). Now, for $q \in X$, the induced subtree in T has mutation probability at least $f(n)$ on its central edge, and mutation probability of no more than $\max\{E^{ij}\colon i,j \in q\} < 0.5 - \tau$ on any pendant edge. Then, by Theorem 9 we have

$$\mathbb{P}[B_q] \geqslant 1 - 18\exp(-(1-\sqrt{1-2f})^2\tau^2 k/8) \tag{20}$$

whenever $q \in X$. Also, the occurrence of event C implies that

$$Z_\tau \subseteq X \tag{21}$$

since if $q \in Z_\tau$, and $i,j \in q$, then $i,j \in S_{2\tau}$, and then (by event C), $E^{ij} < 0.5 - \tau$, hence $q \in X$. Thus,

$$\mathbb{P}[B \cap C] = \mathbb{P}\left[\left(\bigcap_{q \in Z_\tau} B_q\right) \cap C\right] \geqslant \mathbb{P}\left[\left(\bigcap_{q \in X} B_q\right) \cap C\right],$$

where the second inequality follows from (21), as this shows that when C occurs, $\bigcap_{q \in Z_\tau} B_q \supseteq \bigcap_{q \in X} B_q$. Invoking the Bonferonni inequality, we deduce that

$$\mathbb{P}[B \cap C] \geqslant 1 - \sum_{q \in X} \mathbb{P}[\overline{B_q}] - \mathbb{P}[\overline{C}]. \tag{22}$$

Thus, from above,

$$\mathbb{P}[A \cap B] \geqslant \mathbb{P}[A \cap B \cap C] = P[B \cap C]$$

(since $\mathbb{P}[A|C] = 1$), and so, by (20) and (22),

$$\mathbb{P}[A \cap B] \geqslant 1 - 18\binom{n}{4}\exp(-(1-\sqrt{1-2f})^2\tau^2 k/8) - (n^2-n)e^{-\tau^2 k/2}.$$

Formula (12) follows by an easy calculation for $\tau = c \cdot \lambda$, for any $0 < c \leqslant 1/6$.

We proceed to prove that assumption (10) holds. Recall the definition of $Q_{w(\tau)} = \{FPM(q)\colon q \in Z_\tau\}$. Now let D be the event that whenever t and t' are two edi-subtrees which are *not* siblings, but there is a witness in $Q_w(\tau)$ to the siblinghood of T, then there is also an antiwitness in $Q_w(\tau)$.

Recalling Theorem 4, it is obvious that event $A \cap B \cap D$ implies Assumptions (9) and (10). We are going to show that $\mathbb{P}[A \cap B \cap D] = 1 - o(1)$ under the conditions of

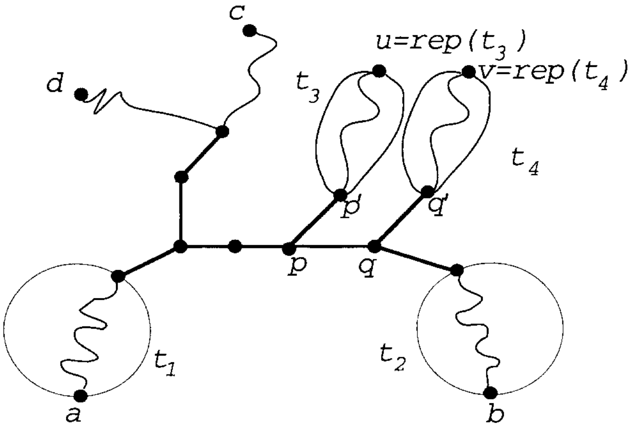


Fig. 1. Finding an antiwitness.

the theorem for a certain choice of τ, which is just slightly smaller than the τ that sufficed for the assumption (9). Technically, we are going to show

$$\mathbb{P}[D|A\cap B\cap C]=1. \tag{23}$$

proof of (23): $\overline{D}=\bigcup_{t_1,t_2}H_{t_1,t_2}$, where t_1,t_2 denote two disjoint edi-subtrees of T, and H_{t_1,t_2} denotes the event that there is a witness but no antiwitness for the siblinghood of t_1,t_2 in $Q_{w(\tau)}$. Therefore, in order to prove (23), it suffices to prove

$$\mathbb{P}[H_{t_1,t_2}|A\cap B\cap C]=0. \tag{24}$$

Assume that there is a witness for the siblinghood of t_1,t_2 where t_1 and t_2 are *not* siblings. We will show that $Q_{w(\tau)}$ contains an antiwitness to the siblinghood of t_1 and t_2. Let the witness to the siblinghood of t_1 and t_2 be $ab|cd$, where $a\in t_1$, $b\in t_2$, and c,d not in $t_1\cup t_2$. Let pq be an internal edge of the unique ab path in T containing the midpoint of the path $P(a,b)$ measured using the lengths defined by the corrected model distances D, and with p closer to a and q closer to b, i.e. the edge (p,q) maximizes the following quantity:

$$\min_{pq\ \textit{internal edge}}(1-2E^{ap},1-2E^{qb}). \tag{25}$$

Let p' and q' be neighbors of p and q respectively that are not on the path between nodes a and b. Consider the *edi*-subtrees t_3 and t_4 rooted at p' and q' respectively, formed by deleting (p,p') and (q,q'), respectively. Set $u=rep(t_3)$, $v=rep(t_4)$ (Fig. 1).

We are going to show that

$$\{a,b,u,v\}\in Z_\tau, \tag{26}$$

and $au|bv\in Q_{w(\tau)}$. The proof of (26) is the only issue, since by (15) the split of $\{a,b,u,v\}$ is correctly reconstructed, and is $au|bv$ by construction. Clearly

$$\mathbb{P}[H_{t_1,t_2}|A\cap B\cap C]\leqslant\mathbb{P}[\{a,b,u,v\}\notin Z_\tau|A\cap B\cap C]. \tag{27}$$

The RHS of (27) can be further estimated by

$$\begin{aligned}&\mathbb{P}[h^{au}\geqslant 0.5-2\tau|A\cap B\cap C]+\mathbb{P}[h^{av}\geqslant 0.5-2\tau|A\cap B\cap C]\\&\mathbb{P}[h^{bu}\geqslant 0.5-2\tau|A\cap B\cap C]+\mathbb{P}[h^{bv}\geqslant 0.5-2\tau|A\cap B\cap C]\\&\qquad\qquad\qquad\qquad+\mathbb{P}[h^{uv}\geqslant 0.5-2\tau|A\cap B\cap C].\end{aligned}\tag{28}$$

The fifth term $\mathbb{P}[h^{uv}\geqslant 0.5-2\tau|A\cap B\cap C]=0$, since it is easy to find a short quartet which contains u,v; and therefore by (13), $h^{uv}<0.5-2\tau$. Here is how to find a short quartet containing u and v. Let a' denote the neighbor of p on the ab path towards a, and let q denote the neighbor of q on the ab path towards b. Consider the edi-subtree t_5 defined by pa', which contains the leaf a, and the edi-subtree t_6 defined by qb', which contains the leaf b. It is easy to check that $\{u,v,rep(t_5),rep(t_6)\}$ is a short quartet.

In order to finish the proof of (24), and hence the proof of (23), it suffices to show that the other four terms in (28) are zero as well. The third and fourth terms are symmetric to the first and second, and in fact the second has a worse bound than the first. Therefore it suffices to prove that

$$\mathbb{P}[h^{av}\geqslant 0.5-2\tau|A\cap B\cap C]=0.\tag{29}$$

We assume that $\{a,v\}\notin S_{2\tau}$, and show that consequently τ is large. Hence, for a properly small τ, Formula (29), and hence (23) holds. From $\{a,v\}\notin S_{2\tau}$, conditioning on C,

$$E^{av}>0.5-3\tau,\tag{30}$$

and $\{a,b\}\in S_{2\tau}$, and hence, conditioning on C,

$$E^{ab}<0.5-\tau.\tag{31}$$

There is no difficulty to extend the definition of E^{ij} to cases when at least one of i,j is an internal vertex of the tree. Simple algebra yields from formula (30) and Lemma 1, that

$$6\tau\geqslant 1-2E^{av}=(1-2E^{pv})(1-2E^{pa}).\tag{32}$$

We have

$$1-2E^{pv}\geqslant(1-2g)^{depth(T)+2}=\sqrt{\lambda(1-2g)}\tag{33}$$

by the definition of λ (see formula (17)) and the choice of v as representative. By formula (25), it is easy to see that

$$1-2E^{pa}\geqslant q(1-2g)^2\sqrt{1-2E^{ab}}.\tag{34}$$

Combining (31)–(34), we obtain $6\tau>\sqrt{\lambda(1-2g)}(1-2g)^2\sqrt{2\tau}$. This formula fails, if we select

$$\tau=c_2\cdot(1-2g)^5\lambda\tag{35}$$

with a sufficiently small positive constant c_2.

Case 1: *$p \notin t_1$ and $q \notin t_2$ (as in Fig.* 1). Then $au|bv \in Q_{w(\tau)}$ is an anti-witness, as desired.

When Case 1 does not hold, the only problem that can arise is if the valid split $au|bv$ does not satisfies the condition $\{u,v\} \cap (t_1 \cup t_2) = \emptyset$, and hence is not an antiwitness.

Case 2: *$p \in t_1$ or $q \in t_2$*. Without loss of generality we may assume $p \in t_1$. Now we *redefine* the location of the edge pq on the ab path as follows. Let p denote the first vertex after $root(t_1)$ on the ab path and let q denote the second. Clearly $q \notin t_2$, since t_1 and t_2 are not siblings. We also redefine p', q', t_3, u, t_4, v according to the new p and q. Redefine a to be $rep(t_1)$ and call the old a as a^*. Now we are going to show (26) and that $au|bv \in Q_{w(\tau)}$ is the sought-for antiwitness (note a, u, v have been redefined, but b has not). Again, we have to see (27) and prove that (28) is termwise zero.

For pairs u, v where $\{u,v\} \in S_{2\tau}$, we proceed exactly as in Case 1. Observe that E^{bu} and E^{bv} *decreased* during the redefinition, so a calculation like (29)–(35) still goes through. Observe that $L(a,u) \leqslant 2depth(T)+2$, $L(a,v) \leqslant 2depth(T)+3$, and hence $\{a,u\} \in S_{2\tau}$ and $\{a,v\} \in S_{2\tau}$, exactly as in the proof of (19). The only thing left to prove is $\{a,b\} \in S_{2\tau}$.

In order to prove $\mathbb{P}[h^{ab} \geqslant 0.5 - 2\tau | A \cap B \cap C] = 0$, since under the condition C, it suffices to prove $1 - 2E^{ab} > 6\tau$. However,

$$1 - 2E^{ab} = (1 - 2E^{a,root(t_1)})(1 - 2E^{root(t_1),b}) \geqslant (1-2g)^{depth(T)}(1-2g)^2\sqrt{1 - E^{a^*b}},$$

and we still have $\sqrt{1 - E^{a^*b}} > \sqrt{2\tau}$ according to (31). A calculation like the one resulting in (35) gives the result wanted, and we are finished with the proof of (23).

Using these statements, $\mathbb{P}[A \cap B \cap D] \geqslant \mathbb{P}[A \cap B \cap D | A \cap B \cap C] \times \mathbb{P}[A \cap B \cap C] = \mathbb{P}[A \cap B \cap C] = \mathbb{P}[B \cap C]$, and we are back to the same estimates that proved assumption (9), but we need a slightly smaller τ and consequently slightly larger k.

Note that the proof above applies to all $c_3 \in [c_2/2, c_2]$, if it applies to $c_3 = c_2$ and $c_3 = c_2/2$, so that assumption (11) holds. □

Note that the proof also handled the problem that arises if some of the dissimilarity scores exceed 1/2, and so we cannot even compute corrected distances. The moral is that those pairs are not needed according to the proof. Therefore there is no need for additional conditioning for the shape of the observed data.

6.2. Runtime analysis of the search strategies

Theorem 12. (i) *The running time of WAM based on sequential search is* $O(n^2k + n^6 \log n)$

(ii) *The running time of WAM based on sparse-high search is* $O(n^2k + n^4 \log n \log k)$. *Assume now that our model tree is a random binary tree, under the uniform or Yule–Harding distribution, and all mutation probabilities are taken from an interval* $(p - \varepsilon_n, p + \varepsilon_n)$, *for a sufficiently small sequence* ε_n. *If* k *is as large as in* (12), *then with probability* $1 - o(1)$

(iii) *The running time of WAM based on sequential search is* $O(n^2k + n^3 poly \log n)$.

(iv) *The running time of WAM based on sparse-high search is* $O(n^2k + n^2 poly \log n)$.

Proof. Computing the matrices h and d takes $O(n^2k)$ time. (All distance methods begin by computing these distance matrices, but this "overhead cost" is usually always mentioned in the running time analysis of a given method.) Let w_0 be defined to be the smallest $w \in h^{ij}$ such that Q_w is T-forcing. Let $i(w)$ be the *order* of w within the sorted h^{ij} values. Then, since each call of the WATC algorithm uses $O(n^2 + |Q| \log |Q|)$ time, the running time of the sequential search is $O(i(w_0)(n^2 + |Q_{w_0}| \log |Q_{w_0}|))$, after the preprocessing.

For (i), the sequential search application of the WATC algorithm is $O(n^6 \log n)$, since we need never do more than examine all sets Q_w, and the largest such set has cardinality $O(n^4)$.

Claim (ii) follows form the observations that the sparse-high search calls the WATC algorithm at most $O(\log k)$ times, and each call costs at most $O(n^4 \log n)$ time.

We now prove (iii). The depth of a random tree (under either the uniform or Yule–Harding distributions) is with high probability $O(\log\log n)$ by Theorem 10, and so there are at most $O(poly \log n)$ leaves which are no more than about $O(\log\log n)$ distance (measured topologically) from any fixed leaf. This is the only fact that we exploit from the assumption of randomness of the tree. For two leaves i,j, recall that $L(i,j)$ denotes the topological distance between i and j. We are going to show that if τ is the value at which the search reconstructs the tree in the proof of Theorem 11, then with probability $1 - o(1)$ we have $L(i,j) = O(\log\log n)$, whenever $i,j \in q \in Q_\tau$. This yields $|Q_{w(\tau)}| = n \cdot poly \log(n)$. In the proof of Theorem 11, according to formula (18), event C holds with probability $1 - o(1)$. In that proof $Q_{w(\tau)}$ is denoted by $Z_{\tau/4}$. Now

$$(1-2g)^{L(i,j)} = 1 - 2E^{ij} \geqslant \tau/2, \tag{36}$$

where the equality follows from Lemma 1, and the inequality follows from the conditioning on the event C. Plugging in (35) for τ immediately yields $L(i,j) = O(\log\log n)$. Since the sequential search makes $O(npoly \log(n))$ calls to the WATC algorithm, (iii) follows.

To obtain (iv), observe that Formulae (35), (17), and $depth(T) = O(\log\log n)$ imply that the number of iterations in the sparse-high search is

$$-\log_2 \tau = O(-\log(1-2g) \cdot depth(T)) = O(\log\log n). \qquad \square$$

6.3. The performance of other distance methods under the Neyman 2-state model

In this section we describe the convergence rate for the WAM and DCM method, and compare it briefly to the rates for two other distance-based methods, the Agarwala et al. 3-approximation algorithm [1] for the L_∞-nearest tree, and neighbor joining [43]. We make the natural assumption that all methods use the same corrected empirical distances from Neyman 2-state model trees. The comparison we provide in this section will establish that our method requires *exponentially shorter* sequences in order to ensure accuracy of the topology estimation than the algorithm of Agarwala et al., for almost all trees under uniform or Yule–Harding probability distributions. The trees for which the two methods need comparable sequence lengths are those in which the diameter and the depth are as close as possible – such as complete binary trees. Even in these cases, WAM and DCM will nevertheless need shorter sequences than Agarwala et al. to obtain the topology with high probability, as we showed it in Section 3. (Again, note that this analysis is inherently pessimistic, and it is possible that the methods may obtain accurate reconstructions from shorter sequences than suffice by this analysis.)

The neighbor joining method is perhaps the most popular distance-based method used in phylogenetic reconstruction, and in many simulation studies (see [34, 35, 44] for an entry into this literature) it seems to outperform other popular distance based methods. The Agarwala et al. algorithm [1] is a distance-based method which provides a 3-approximation to the L_∞ nearest tree problem, so that it is one of the few methods which provide a provable performance guarantee with respect to any relevant optimization criterion. Thus, these two methods are two of the most promising distance-based methods against which to compare our method. All these methods use polynomial time.

In [22], Farach and Kannan analyzed the performance of the Agarwala et al. algorithm with respect to tree reconstruction in the Neyman 2-state model, and proved that the Agarwala et al. algorithm converged quickly for the variational distance. Personal communication from S. Kannan gave a counterpart to (12): if T is a Neyman 2-state model tree with mutation rates in the range $[f, g]$, and if sequences of length k' are generated on this tree, where

$$k' > \frac{c' \cdot \log n}{f^2(1-2g)^{2diam(T)}} \tag{37}$$

for an appropriate constant c', and where $diam(T)$ denotes the "diameter" of T, then with probability $1 - o(1)$ the result of applying Agarwala et al. to corrected distances will return the topology of the model tree. In [5], Atteson proved the same result for Neighbor Joining though with a different constant. (The constant for neighbor joining is smaller than the constant for the Agarwala et al. algorithm, suggesting that neighbor joining can be guaranteed to be accurate from shorter sequences than Agarwala et al., on any tree in the Neyman 2-state model. However, remember that this analysis is pessimistic, and it may be that correct reconstruction is possible from shorter sequences than this analysis suggests.)

Comparing this formula to (12), we note that the comparison of depth and diameter is the issue, since $(1-\sqrt{1-2f})^2 = \Theta(f^2)$ for small f. It is easy to see that $diam(T) \geqslant 2depth(T)$ for binary trees T, but the diameter of a tree can in fact be quite large (up to $n-1$), while the depth is never more than $\log n$. Thus, for every fixed range of mutation probabilities, the sequence length that suffices to guarantee accuracy for the Neighbor Joining or Agarwala et al. algorithms can be quite large (i.e. it can grow exponentially in the number of leaves), while the sequence length that suffices for the witness-antiwitness method will never grow more than polynomially.

In order to understand the bound on the sequence length needed by these methods, we now turn to an analysis of the diameter of random trees. The models for random trees are the *uniform* model, in which each tree has the same probability, and the

Yule–Harding model, studied in [2, 8, 29]. This distribution is based upon a simple model of speciation, and results in "bushier" trees than the uniform model.

Theorem 13. (i) *For a random semilabelled binary tree T with n leaves under the uniform model, diam$(T) > \varepsilon\sqrt{n}$ with probability* $1 - \mathrm{O}(\varepsilon^2)$.

(ii) *For a random semilabelled binary tree T with n leaves under the Yule–Harding distribution, after suppressing the root, diam$(T) = \Theta(\log n)$, with probability* $1 - \mathrm{o}(1)$.

Proof. We begin by establishing (i). The result of Carter et al. [11] immediately implies that leaves a, b have distance $m+1$ with probability *exactly* $m!N(n-2,m)/(2n-5)!!$ under the uniform model. For small enough ε, $m \leqslant \varepsilon\sqrt{n}$, this probability is $\Theta(m/n)$. Summing up the probabilities from $m = 1$ to $m = \varepsilon\sqrt{n}$, we see that $diam(T) > \varepsilon\sqrt{n}$ with probability at least $1 - \mathrm{O}(\varepsilon^2)$.

We now consider (ii). First we describe rooted Yule–Harding trees. These trees are defined by the following constructive procedure. Make a random permutation $\pi_1, \pi_2, \ldots, \pi_n$ of the n leaves, and join π_1 and π_2 by edges to a root R of degree 2. Add each of the remaining leaves sequentially, by randomly (with the uniform probability) selecting an edge incident to a leaf in the tree already constructed, subdividing the edge, and make π_i adjacent to the newly introduced node. For a rooted Yule–Harding tree T^{R}, let $h(T^{\mathrm{R}})$ denote the maximum distance of any leaf from the root. Let T be the unrooted Yule–Harding tree obtained from T^{R} by suppressing the root, and identifying the two edges incident with the root. Let $diam(T)$ denote the diameter of T. Then, we always have

$$h(T^{\mathrm{R}}) \leqslant diam(T) \leqslant 2h(T^{\mathrm{R}}) - 1.$$

Now Aldous [2] shows that $h(T^{\mathrm{R}})/\log n$ converges in distribution to a (nonzero) constant c. Then, with probability tending to 1, $diam(T)/\log n$ will lie between c and $2c$. □

In Table 1, we summarize sequence length that *suffice* for accurate reconstruction with high probability of WAM and DCM, and compare these to the sequence lengths that suffice for the Agarwala et al. algorithm, according to the analyses that we have given above (thus, our summary is based upon (12), (37), and Theorems 10 and 13). Sequence lengths are given in terms of growth as a function of n, and assume that mutation probabilities on edges lie within the specified ranges.

7. Extension to general stochastic models

In this section we consider the generalization of the WAM and DCM for inferring trees in the general stochastic model. Just as in the case of the Neyman 2-state model, we find that WAM and DCM obtains accurate estimations of the tree from sequences whose length is never more than polynomial in the number of leaves (for every fixed

Table 1

		Range of mutation probabilities on edges	
		$[f,g]$ f,g are constants	$\left[\frac{1}{\log n}, \frac{\log\log n}{\log n}\right]$
Binary trees	DCM/WAM	Polynomial	Polylog
Worst-case	Agarwala et al.	Superpolynomial	Superpolynomial
Random binary trees	DCM/WAM	Polylog	Polylog
(uniform model)	Agarwala et al.	Superpolynomial	Superpolynomial
Random binary trees	DCM/WAM	Polylog	Polylog
(Yule–Harding)	Agarwala et al.	Polynomial	Polylog

range for the mutation probabilities), and in general only polylogarithmic in the number of leaves. This should be contrasted to the study of Ambainis et al. [4].

Suppose the sequence sites evolve *i.i.d.* according to the "general" Markov model – that is, there is some distribution of states π at the root of the tree, and each edge e has an associated stochastic transition matrix $M(e)$, and the (random) state at the root evolves down the tree under a natural Markov assumption, as in the general stochastic model of Definition (III).

Let $f_{ij}(\alpha,\beta)$ denote the probability that leaf i is in state α and leaf j is in state β. By indexing the states, $f_{ij}(\alpha,\beta)$ forms a square matrix, $F_{ij}=[f_{ij}(\alpha,\beta)]$. Then

$$\phi_{ij} = -\log\det(F_{ij}) \tag{38}$$

denotes the *corrected model distance* between i and j. (There will be a guarantee for $\det(F_{ij})>0$.)

The *corrected empirical distance* $\hat{\phi}_{ij}$ of two species is computed as in (38), but uses the matrix $\hat{F}_{ij}$ composed of the relative frequencies $\hat{f}_{ij}(\alpha,\beta)$ of i being in state α and j being in state β, instead of the probability $f_{ij}(\alpha,\beta)$:

$$\hat{\phi}_{ij} = -\log\det(\hat{F}_{ij}). \tag{39}$$

Then, ϕ_{ij} can be derived from a positive edge weighting of the model tree, provided that the identifiability condition described in Section 2 (Tree Reconstruction) holds. These mild conditions only require that $\det(M(e))$ not take on the values $0,1,-1$, and that the components of π are nonzero (i.e. every state has a positive probability of occurrence at the root).

Note that $\det(M(e))$ takes the values 1 or -1 precisely if $M(e)$ is a permutation matrix. Also, for the Neyman 2-state model $\det(M(e)) = 1-2p(e)$, where $p(e)$ is the mutation probability on edge e; thus, $\det(M(e))>0$ and $\det(M(e))$ tend to 0 as p approaches 0.5, and tend to 1 as p approaches 0. In general, $(1/2)[1-\det(M(e)]$ plays the role of $p(e)$ in the general model. Thus, a natural extension of our restriction $f \leqslant p(e) \leqslant g$ and from the Neyman 2-state model corresponds to

$$0<1-2x' \leqslant \det(M(e)) \leqslant 1-2x<1, \tag{40}$$

for suitable x, x', and we will henceforth impose this restriction for all edges of the tree. For technical reasons, we also impose the mildly restrictive condition that every vertex can be in each state μ with at least a certain fixed positive probability:

$$\pi(v)_\mu > \varepsilon. \tag{41}$$

This condition (41) certainly holds under the Neyman 2-state model, the Kimura 3-state model [39], and much more general models (providing each state has positive probability of occurring at the root). Indeed this last weaker condition might be enough, but it would seem to complicate the analysis quite a lot.

Now, let $\lambda(e)$ be the weight of edge e in the realization of ϕ on the (unrooted version) of the true underlying tree T.

Lemma 4. *Set $\delta(x) = -0.5 \log(1-2x)$. Then*

$$\lambda(e) \geqslant -0.5 \log(\det(M(e))) \geqslant \delta(x) \tag{42}$$

for every edge e of T.

Proof. The second inequality follows from the restriction we imposed above on $\det(M(e))$. The first inequality in (42) follows from similar arguments to those appearing in [47]; for the sake of completeness we give a proof.

Let T be the unrooted version of T^ρ. Now the edges of T correspond bijectively to the edges of T^ρ, except perhaps for one *troublesome* edge of T which arises whenever the root of T^ρ has degree two – in that case, two edges e_1, e_2 of T^ρ adjacent to ρ are identified to form e. For convenience, we assume in this proof that ρ is not a leaf.

We now prove that $\lambda(e) \geqslant -0.5 \log \det(M(e))$ for all (non-troublesome) edges e of T, and if T has a troublesome edge e corresponding to edges e_1 and e_2 in T^ρ, then $\lambda(e) \geqslant -0.5 \log(\det(M(e_1)) \det(M(e_2)))$.

For any edge $e = (v, w)$ of T^ρ where w is a leaf, let

$$h(e) = -\log \det(M(e)) - 0.5 \log \left[\prod_\mu \pi(v)_\mu \right]$$

while, for any edge $e = (v, w)$ of T^ρ for which neither of v, w are leaves, let

$$h(e) = -\log \det(M(e)) - 0.5 \log \left[\prod_\mu \pi(v)_\mu \right] + 0.5 \log \left[\prod_\mu \pi(w)_\mu \right].$$

Thus, h describes a weighting of the edges of T^ρ and thereby a weighting h^* of the edges of T by setting h^* equal to h on the non-troublesome edges, and the convention that if T has a troublesome edge e arising from the identification of a pair e_1, e_2 of edges of T^ρ then $h^*(e) = h(e_1) + h(e_2)$. Now, h realizes the ϕ_{ij} values on T^ρ. Thus, h^* also realizes the ϕ_{ij} values, on T and since (as we show) the edge weighting is strictly positive, it follows, by classical results [10], that this is the unique such edge weighting of T. Thus $\lambda = h^*$.

Now for an edge $e=(v,w)$ of T^ρ where w is a leaf,

$$h(e) \geqslant -\log\det(M(e)) \geqslant -0.5\log\det(M(e))$$

as claimed. Alternatively, for an edge $e=(v,w)$ of T for which neither of v, w are leaves, we have

$$h(e) = -\log\det(M(e)) - 0.5\log\left[\prod_\mu \pi(v)_\mu\right] + 0.5\log\left[\prod_\mu \pi(w)_\mu\right].$$

In order to derive our desired inequality we establish a further result. Let us suppose $M=[M_{\mu\nu}]$ is any $r\times r$ matrix with non-negative entries and $\boldsymbol{x}$ is a row vector of length r with non-negative entries. We claim that

$$\prod_\mu (\boldsymbol{x}M)_\mu \geqslant |\det(M)|\prod_\mu x_\mu.$$

To obtain this, note that the left-hand side is just

$$\prod_\mu\left(\sum_\nu x_\nu M_{\nu\mu}\right) \geqslant \left(\sum_\sigma M_{\sigma(1)1}M_{\sigma(2)2}\ldots M_{\sigma(r)r}\right)\prod_\mu x_\mu,$$

where the second summation is over all permutations σ of $(1,2,\ldots,r)$, and so this sum is at least $|\det(M)|$, since the permanent of a nonnegative matrix is never smaller than the absolute value of its determinant. Now, $[\pi(w)_1,\ldots,\pi(w)_r]=[\pi(v)_1,\ldots,\pi(v)_r]M(e)$, and so, applying the above inequality to the case $M=M(e)$ and $x=[\pi(v)_1,\ldots,\pi(v)_r]$, we obtain

$$\prod_\mu \pi(w)_\mu \geqslant \det(M(e))\prod_\mu \pi(v)_\mu.$$

Thus,

$$0.5\log\det(M(e)) \leqslant 0.5\log\left[\prod_\nu \pi(w)_\nu\right] - 0.5\log\left[\prod_\mu \pi(v)_\mu\right]$$

and so

$$\begin{aligned} h(e) &= -0.5\log\det(M(e)) \\ &\quad -0.5\left(\log\det(M(e)) + \log\left[\prod_\mu \pi(v)_\mu\right] - \log\left[\prod_\mu \pi(w)_\mu\right]\right) \\ &\geqslant -0.5\log\det(M(e)), \end{aligned}$$

as claimed.

The inequalities for h now extend to $h^*=\lambda$ for all (non-troublesome) edges of T. If T^ρ has a troublesome edge e then $\lambda(e)=h^*(e)=h(e_1)+h(e_2)$, and from the above we have $h(e_i)\geqslant -0.5\log\det(M(e_i))$ for $i=1,2$. □

Theorem 14. *Let* $x = x(n)$ *and* $x' = x'(n)$ *be such that for all edges in the tree* T, $0 < 1 - 2x' \leqslant \det(M(e)) \leqslant 1 - 2x < 1$. *Assume* x' *has an upper bound strictly less than* 1/2. *Mutatis mutandis, algorithms FPM, DCM and WAM, Theorems* 9, 11, *and* 12 *generalize to the general stochastic model under* (40) *and* (41). *WAM and DCM returns the binary model tree* T *with probability* $1 - o(1)$ *if*

$$k > \frac{c \cdot \log n}{x^2(1 - 2x')^{4depth(T)}} \tag{43}$$

with a certain constant c.

Proof. Recall the definition of the corrected empirical distance, $\hat{\phi}_{ij}$, and $\delta(x)$ $(= -0.5\log(1 - 2x))$. We first establish the following

Claim: *If*

$$|\phi_{ij} - \hat{\phi}_{ij}| > \delta(x)/2 \tag{44}$$

then

$$|\det(F^{ij}) - \det(\hat{F}^{ij})| > x\det(F^{ij})/4. \tag{45}$$

Proof of Claim: By inequality (44),

$$|\log(\det(F^{ij})) - \log(\det(\hat{F}^{ij}))| = \left|\log\left(\frac{\det(\hat{F}^{ij})}{\det(F^{ij})}\right)\right| > -\frac{1}{4}\log(1 - 2x)$$

and so $\det(\hat{F}^{ij})/\det(F^{ij})$ is either greater than $(1-2x)^{-1/4}$, or less than $(1-2x)^{1/4}$. Thus, $|\det(F^{ij}) - \det(\hat{F}^{ij})| > \min\{\alpha^-(x), \alpha^+(x)\}\det(F^{ij})$ where $\alpha^+(x) := 1 - (1 - 2x)^{1/4}; \alpha^- = (1 - 2x)^{-1/4} - 1$. Now, it can be checked that, for x strictly between 0 and 1/2, $\alpha^-(x), \alpha^+(x) > x/4$ which establishes the Claim.

To apply Lemma 3, we need to know how $\det(\hat{F}^{ij})$ responds to the replacement at one site of a pattern by a different pattern. If $\hat{F}_1^{ij}$ is the resulting F-matrix for this perturbed data set, then

$$\hat{F}_1^{ij} = \hat{F}^{ij} + (1/k)D^{ij}$$

where D^{ij} has one entry of +1, one entry of −1, and all other entries 0. Consequently,

$$|\det(\hat{F}_1^{ij}) - \det(\hat{F}^{ij})| \leqslant c_1/k \tag{46}$$

for some constant c_1.

Next, for any real analytic function f defined on a vector x having a normalized multinomial distribution with parameters k and μ, we have (by Taylor expansion of f about μ to the second derivative term, followed by application of the expectation operator):

$$|\mathbb{E}[f(x)] - f(\mu)| \leqslant \frac{1}{2}M\sum_{i,j}|cov(x_i, x_j)|,$$

where $cov(x_i,x_j)$ is the covariance of x_i,x_j (equal to $\mu_i(1-\mu_i)/k$, when $i=j$, and $-\mu_i\mu_j/k$ otherwise); and where M is the maximal value of any of the second derivatives of f over the unit simplex. Thus, since $\det(F^{ij})$ is a polynomial in the entries of F^{ij}, we have:

$$|\mathbb{E}[\det(\hat{F}^{ij})]-\det(F^{ij})|\leqslant c_2/k \tag{47}$$

for some constant c_2. Combining (47) with the triangle inequality gives

$$|\det(F^{ij})-\det(\hat{F}^{ij})|\leqslant|\det(\hat{F}^{ij})-\mathbb{E}[\det(\hat{F}^{ij})]|+c_2/k$$

and so

$$\mathbb{P}[|\det(F^{ij})-\det(\hat{F}^{ij})|>t]\leqslant\mathbb{P}[|\det(\hat{F}^{ij})-\mathbb{E}[\det(\hat{F}^{ij})]|>(t-c_2/k)] \tag{48}$$

for any $t>0$. Hence by Lemma 3, applied with (46), we have

$$\mathbb{P}[|\det(F^{ij})-\det(\hat{F}^{ij})|>x\det(F^{ij})/4]\leqslant 2\exp\left(-d\left(\frac{x\det(F^{ij})}{4}-\frac{c_2}{k}\right)^2 k\right) \tag{49}$$

for a constant d. For the validity of the latter argument we need that

$$\frac{x\det(F^{ij})}{4}-\frac{c_2}{k}>0. \tag{50}$$

Now, how can we set a lower bound for $\det(F^{ij})$? Note that $\det(F^{ij})$ is just the product of $\det(M(e))$ over all edges on the path from i to j, times the product of $\pi(v_{ij})_\mu$ over all states μ, where $\pi(v)$ is the vector of probabilities of states at vertex v, and v_{ij} is the most recent common ancestor of i and j in the tree. Due to our hypotheses (41), we have

$$\det F^{ij}>c_3(1-2x')^{d(i,j)} \tag{51}$$

with a positive constant c_3. However, the conditions of the Theorem required $k>cx^{-1}(1-2x')^{-d(i,j)}$, and therefore taking a sufficiently large c guarantees (50).

Putting the pieces (44), (45) and (49) together we see that

$$\mathbb{P}[|\phi_{ij}-\hat{\phi}_{ij}|>\delta(x)/2]\leqslant 2\exp\left(-d\left(\frac{x\det(F^{ij})}{4}-\frac{c_2}{k}\right)^2 k\right). \tag{52}$$

Combining (51) and (42), we have

$$\mathbb{P}\left[|\phi_{ij}-\hat{\phi}_{ij}|>(1/2)\min_e\{\lambda(e)\}\right]\leqslant 2\exp\left(-d\left(c_4\frac{x(1-2x')^{d(i,j)}}{4}-\frac{c_2}{k}\right)^2 k\right),$$

where c_4 is a positive constant, and $d(i,j)$ is the number of edges in T separating leaves i and j. Hence, for any fixed quartet q of diameter $diam(q)$,

$$\mathbb{P}[\text{FPM errs on } q]\leqslant K\exp(-D'x^2(1-2x')^{2diam(q)}k) \tag{53}$$

for constants D',K. Thus we have an analogue of Theorem 9.

Now we show how to generalize the proof of Theorem 11. To avoid needless repetitions, we give details for the proof of assumption (9) only, and leave the proofs of assumptions (10) and (11) to the Reader. Note that the proof of correctness of DCM hinges exactly on assumption (9). Having a distance function in the general model, the width and algorithmic operations based on width generalize in a straightforward way.

For k evolving sites (i.e. sequences of length k), and $\tau > 0$, let us define the following two sets, $S_\tau = \{\{i,j\}\colon \det(\hat{F}^{ij}) > 2\tau\}$ and $Z_\tau = \{q \in \binom{[n]}{4}\colon$ for all $i,j \in q, \{i,j\} \in S_{2\tau}\}$ (note the similarity between the definition for the set Z_τ, and that for the set Q_w of quartet splits of quartets of width at most w). We also define the following two events, $A = \{Q_{\text{short}}(T) \subseteq Z_\tau\}$ and $B =$ FPM correctly reconstructs the tree for all $q \in Z_\tau$. Thus, $\mathbb{P}[\mathscr{A} \cap \mathscr{B} \neq \emptyset] \geqslant \mathbb{P}[A \cap B]$. Let C be the event: "$S_{2\tau}$ contains all pairs $\{i,j\}$ with $\det(F^{ij}) > 6\tau$, and no pair $\{i,j\}$ with $\det(F^{ij}) \leqslant 2\tau$". Define $\lambda = \varepsilon^r(1-2x')^{2depth(T)+3}$. We claim that

$$\mathbb{P}[C] \geqslant 1 - (n^2 - n)\mathrm{e}^{-c\tau^2 k} \tag{54}$$

for a constant $c > 0$ and

$$\mathbb{P}[A|C] = 1 \quad \text{if } \tau \leqslant \lambda/6. \tag{55}$$

Suppose $\det(F^{ij}) \leqslant 2\tau$. To establish (54), using arguments similar to those between (45) and (49) one easily sees that Lemma 3 applies and

$$\begin{aligned}\mathbb{P}[\{i,j\} \in S_{2\tau}] &= \mathbb{P}[\det(\hat{F}^{ij}) > 4\tau]\\ &\leqslant \mathbb{P}[\det(\hat{F}^{ij}) - \det(F^{ij}) \geqslant 2\tau] \leqslant \mathrm{e}^{-c\tau^2 k}\end{aligned}$$

for a constant $c > 0$.

Since there are at most $\binom{n}{2}$ such pairs $\{i,j\}$ such that $\det(F^{ij}) \leqslant 2\tau$, the probability that at least one such pair lies in $S_{2\tau}$ is at most $\binom{n}{2}\mathrm{e}^{-c\tau^2 k}$. By a similar argument, the probability that $S_{2\tau}$ fails to contain a pair $\{i,j\}$ with $\det(F^{ij}) > 6\tau$ is also at most $\binom{n}{2}\mathrm{e}^{-c\tau^2 k}$. These two bounds establish (54).

We now establish (55). For $q \in Q_{\text{short}}(T)$ and $i,j \in q$, if a path $e_1 e_2 \ldots e_t$ joins leaves i and j, then $t \leqslant 2depth(T)+3$ by the definition of $Q_{\text{short}}(T)$. Using these facts, and the bound $\det(M(e)) \geqslant 1-2x'$, we obtain $\det(F^{ij}) \geqslant \varepsilon^r(1-2x')^t$. Consequently, $\det(F^{ij}) > 6\tau$ (by assumption that $\tau \leqslant \lambda/6$) and so $\{i,j\} \in S_{2\tau}$ once we condition on the occurrence of event C. This holds for all $i,j \in q$, so by definition of Z_τ we have $q \in Z_\tau$. This establishes (55).

Then for any quartet $q \in Q_{\text{short}}(T)$, if e is the central edge of the contracted subtree induced by q in T, then $\det(M(e)) \leqslant 1 - 2x$. Furthermore, conditional on C, for any pendant edge $e, \det(M(e)) > \min\{\det(F^{ij})\colon i,j \in q\} > 2\tau$. Thus, by (53), which is the analogue of Theorem 9, and the Bonferroni inequality, we can follow the corresponding proof from Theorem 11, to obtain (using (54) and (55))

$$\mathbb{P}[A \cap B] \geqslant 1 - K \binom{n}{4} \exp(-D'x^2(1-2x')^{2depth(T)+3}k) - (n^2 - n)\mathrm{e}^{-d\lambda^2 k}$$

for constants $K, D' > 0$ Formula (43) now follows by an easy calculation.

Note that the proof also handled the problem that arises if some logarithms are to be taken of negative numbers and so we cannot even compute corrected distances. The morale is that those pairs are not needed according to the proof. Therefore there is no need for additional conditioning for the shape of the observed data.

8. Considerations for biological data analysis

The focus of this paper has been to establish analytically that every evolutionary tree is accurately reconstructable from quartets of closely related taxa, and, furthermore, this requires just very short sequences, given certain assumptions about the model tree. This is a significant theoretical result, especially since the bounds that we obtain indicate that the sequence lengths that suffices for accuracy with high probability using our new methods are very much shorter than those that suffice for accuracy using other very promising distance-based methods. However, are these observations significant for biological datasets? And if they are, are these methods likely to be practically useful (or merely indications of what might be achieved in future)?

The answer to the first question, concerning the significant for biological datasets, depends upon whether there are biologically realistic evolutionary trees that have smaller "weighted depth" than "weighted diameter", a concept that we now define.

Let T be an edge-weighted tree with positive weights on the internal edges and non-negative weights on the edges incident with leaves. Let e be an internal edge of the tree. The *weighted depth* around edge e is the minimum value of q so that there exists a set of four leaves, i, j, k, l, with one leaf in each of the four subtrees induced by the removal of e and its endpoints, where $q = \max\{d^T_{ij}, d^T_{ik}, d^T_{il}, d^T_{jk}, d^T_{jl}, d^T_{kl}\}$. The *weighted depth* of the tree T is then the maximum weighted depth of any edge in T. The *weighted diameter* of a tree T is simply the maximum d^T_{xy}, taken over all pairs of leaves x, y. We will denote the weighted depth of a tree T by $w\,depth(T)$ and its weighted diameter by $w\,diam(T)$.

The analysis given in the previous sections of the sequence length that suffices for accuracy for various methods can be restated as follows:

Corollary 1. *DCM and WAM will be accurate with probability* $1-\delta$ *if the sequence length exceeds*

$$c \log n \mathrm{e}^{\mathrm{O}(w\,depth(T))},$$

where c is a constant that depends upon only $f = \min_e p(e)$ *and* δ. *The other distance based methods* (*Agarwala et al.'s single-pivot algorithm and its variant, the double-*

pivot algorithm, the naive method, and neighbor-joining) are accurate with the same probability if the sequence length exceeds

$$c' \log n e^{O(w\,diam(T))},$$

where c' is a constant that also depends only upon $f = \min_e\{p(e)\}$ and δ.

These are only upper bounds (i.e. these may be loose, and exact accuracy may be possible from shorter sequences), but these are also currently the best upper bounds that are known for these methods, to our knowledge.

Thus, to compare the sequence lengths that suffice for exact topological accuracy, we need to compare the weighted depth to the weighted diameter. A reasonable comparison between these two quantities for biologically realistic trees is difficult, as there are very few well established evolutionary trees, especially of large divergent datasets. On the other hand, for some data sets, evolution may proceed in a more-or-less clock-like fashion (i.e. the number of mutations that occurs along an evolutionary lineage is roughly proportional to time). For such data sets, it can be seen that the weighted depth and the weighted diameter are exactly the same. Under these circumstances, there is no benefit to using DCM or WAM instead of one of the better other distance methods, such as neighbor joining, although this analysis also does not suggest that neighbor joining will outperform DCM or WAM (to be precise, the conditions that guarantee accuracy for neighbor-joining will also guarantee accuracy for DCM and WAM, and vice versa). Thus, for clock-like evolutionary conditions, these techniques do not provide any advantage from a theoretical standpoint.

On the other hand, there *are* important biological data sets for which evolution proceeds in a very non-clock like fashion, according to various analyses by biologists and statisticians (see, for example, [55, 56]). For these data sets, there *could* be significant advantage obtained by using techniques such as DCM and WAM, which examine only closely related taxa in order to reconstruct the tree. The degree to which DCM and WAM could provide an advantage would theoretically depend upon the magnitude of the difference between the weighted depth and weighted diameter. This magnitude is likely to be largest for sets of highly divergent taxa, rather than for closely related taxa.

As a practical tool, DCM and WAM are not entirely satisfactory, in part because DCM and WAM only return trees when the conditions hold for exact accuracy. Although some biologists would rather get no tree than get an incorrect tree [41], not all biologists share this view, and so partially correct trees are often desirable. Thus, the answer to the second question is basically negative.

However, DCM and WAM were not designed to be practical tools, but rather to indicate theoretical possibilities, and to suggest how better methods might be invented which could have the theoretical guarantees that DCM and WAM provide, while having better performance in practice. Furthermore, such methods *have* recently been developed. The *disk-covering method* of Huson et al. [36] the *harmonic greedy triples method* of Csuros and Kao [16], and the method of Cryan et al. [15] have each used the observations in this paper and obtained methods with convergence rates that are never worse than polynomial by using only small distances to (re)construct the tree.

Acknowledgements

We thank Scott Nettles for fruitful discussions of efficient implementations, and David Bryant and Éva Czabarka for proof reading the manuscript. We also thank A. Ambainis and another (anonymous) referee, for their careful reading of the manuscript and helpful suggestions for improving the exposition.

Péter L. Erdős was supported in part by the Hungarian National Science Fund contract T 016 358. László A. Székely was supported by the National Science Foundation grant DMS 9701211, the Hungarian National Science Fund contracts T 016 358 and T 019 367, and European Communities (Cooperation in Science and Technology with Central and Eastern European Countries) contract ERBCIPACT 930 113. Michael A. Steel was supported by the New Zealand Marsden Fund and the New Zealand Ministry of Research, Science and Technology. Tandy J. Warnow was supported by an NSF Young Investigator Award CCR-9457800, a David and Lucille Packard Foundation fellowship, and generous research support from the Penn Research Foundation and Paul Angello.

This research started in 1995 when the authors enjoyed the hospitality of DIMACS during the Special Year for Mathematical Support to Molecular Biology, and was completed in 1997 while enjoying the hospitality of Andreas Dress, at Universität Bielefeld, in Germany.

References

[1] R. Agarwala, V. Bafna, M. Farach, B. Narayanan, M. Paterson, M. Thorup, On the approximability of numerical taxonomy: fitting distances by tree metrics, in: Proc. 7th Annual ACM–SIAM Symp. on Discrete Algorithms, 1996, pp. 365–372.

[2] D.J. Aldous, Probability distributions on cladograms, in: D.J. Aldous, R. Permantle (Eds.), Discrete Random Structures, IMA Volume in Mathematics and its Applications, vol. 76, Springer, Berlin, 1995, pp. 1–18.

[3] N. Alon, J.H. Spencer, The Probabilistic Method, Wiley, New York, 1992.

[4] A. Ambainis, R. Desper, M. Farach, S. Kannan, Nearly tight bounds on the learnability of evolution, in: 38th Annual Symp. on Foundations of Computer Science, Miami Beach, FL, 20–22 October 1997. IEEE Science, New York, pp. 524–533.

[5] K. Atteson, The performance of neighbor-joining algorithms of phylogeny reconstruction, in: Computing and Combinatorics, 3rd Annual Internat. Conf., COCOON'97, Shanghai, China, August 1997, COCOON'97, Lecture Notes in Computer Science, vol. 1276, Springer, Berlin, pp. 101–110.

[6] H.-J. Bandelt, A. Dress, Reconstructing the shape of a tree from observed dissimilarity data, Adv. Appl. Math. 7 (1986) 309–343.

[7] V. Berry, O. Gascuel, Inferring evolutionary trees with strong combinatorial evidence, in: Computing and Combinatorics, 3rd Annual Internat. Conf., COCOON'97, Shanghai, China, August 1997, COCOON'97, Lecture Notes in Computer Science, vol. 1276, Springer, Berlin, pp. 111–123.

[8] J.K.M. Brown, Probabilities of evolutionary trees, Syst. Biol. 43 (1994) 78–91.

[9] D.J. Bryant, M.A. Steel, Extension operations on sets of leaf-labelled trees, Adv. Appl. Math. 16 (1995) 425–453.

[10] P. Buneman, The recovery of trees from measures of dissimilarity, in: F.R. Hodson, D.G. Kendall, P. Tautu (Eds.), Mathematics in the Archaeological and Historical Sciences, Edinburgh University Press, Edinburgh, 1971, pp. 387–395.

[11] M. Carter, M. Hendy, D. Penny, L.A. Székely, N.C. Wormald, On the distribution of lengths of evolutionary trees, SIAM J. Discrete Math. 3 (1990) 38–47.

[12] J.A. Cavender, Taxonomy with confidence, Math. Biosci. 40 (1978) 271–280.

[13] J.T. Chang, J.A. Hartigan, Reconstruction of evolutionary trees from pairwise distributions on current species, in: Computing Science and Statistics: Proc. 23rd Symp. on the Interface, 1991, pp. 254–257.

[14] H. Colonius, H.H. Schultze, Tree structure for proximity data, Br. J. Math. Statist. Psychol. 34 (1981) 167–180.

[15] M. Cryan, L.A. Goldberg, P.W. Goldberg, Evolutionary trees can be learned in polynomial time in the two-state general Markov-model, in: Proc. 39th Annual IEEE Symp. on Foundations of Computer Science, 1998, pp. 436–445.

[16] M. Csuros, M.-Y. Kao, Fast reconstruction of evolutionary trees through Harmonic Greedy Triplets, in: Proc. ACM–SIAM Symp. on Discrete Algorithms, 1999, pp. 261–268.

[17] W.H.E. Day, Computational complexity of inferring phylogenies from dissimilarities matrices, Inform. Process. Lett. 30 (1989) 215–220.

[18] P.L. Erdős, M.A. Steel, L.A. Székely, T. Warnow, Local quartet splits of a binary tree infer all quartet splits via one dyadic inference rule, Comput. Artif. Intell. 16 (2) (1997) 217–227.

[19] P.L. Erdős, M.A. Steel, L.A. Székely, T. Warnow, Inferring big trees from short quartets, in: ICALP'97, 24th Internat. Colloquium on Automata, Languages, and Programming, Silver Jubilee of EATCS, Bologna, Italy, July 7–11, 1997, Lecture Notes in Computer Science, vol. 1256, Springer, Berlin, 1997, pp. 1–11.

[20] P.L. Erdős, M.A. Steel, L.A. Székely, T. Warnow, A few logs suffice to build (almost) all trees (I), Random Struct. Algorithms 14 (1999) 153–184.

[21] M. Farach, J. Cohen, Numerical taxonomy on data: experimental results, in: Proc. ACM–SIAM Symp. on Discrete Algorithms, 1997.

[22] M. Farach, S. Kannan, Efficient algorithms for inverting evolution, in: Proc. ACM Symp. on the Foundations of Computer Science, 1996, pp. 230–236.

[23] M. Farach, S. Kannan, T. Warnow, A robust model for inferring optimal evolutionary trees, Algorithmica 13 (1995) 155–179.

[24] J.S. Farris, A probability model for inferring evolutionary trees, Syst. Zool. 22 (1973) 250–256.

[25] J. Felsenstein, Cases in which parsimony or compatibility methods will be positively misleading, Syst. Zool. 27 (1978) 401–410.

[26] J. Felsenstein, Numerical methods for inferring evolutionary trees, Quart. Rev. Biol. 57 (1982) 379–404.

[27] D. Feng, R. Doolittle, Progressive sequence alignment as a prerequisite to correct phylogenetic trees, J. Mol. Evol. 25 (1987) 351–360.

[28] Green Plant Phylogeny Research Coordination Group, Summary Report of Workshop #1: Current Status of the Phylogeny of the Charophyte Green Algae and the Embryophytes (University and Jepson Herbaria, University of California, Berkeley, June 24–28, 1995), 1996.

[29] E.F. Harding, The probabilities of rooted tree shapes generated by random bifurcation, Adv. Appl. Probab. 3 (1971) 44–77.

[30] J. Hein, A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given, Mol. Biol. Evol. 6 (1989) 649–668.

[31] D. Hillis, Approaches for assessing phylogenetic accuracy, Syst. Biol. 44 (1995) 3–16.

[32] D. Hillis, Inferring complex phylogenies, Nature 383 (1996) 130–131.

[33] P. Hogeweg, B. Hesper, The alignment of sets of sequences and the construction of phylogenetic trees, an integrated method, J. Mol. Evol. 20 (1984) 175–186.

[34] J. Huelsenbeck, Performance of phylogenetic methods in simulation, Syst. Biol. 44 (1995) 17–48.

[35] J.P. Huelsenbeck, D. Hillis, Success of phylogenetic methods in the four-taxon case, Syst. Biol. 42 (1993) 247–264.

[36] D. Huson, S. Nettles, T. Warnow, Inferring very large evolutionary trees from very short sequences, Proc., RECOMB, 1999.

[37] T. Jiang, E. Lawler, L. Wang, Aligning sequences via an evolutionary tree: complexity and approximation ACM STOC'94.

[38] S. Kannan, personal communication.

[39] M. Kimura, Estimation of evolutionary distances between homologous nucleotide sequences, Proc. Natl. Acad. Sci. USA 78 (1981) 454–458.

[40] J. Neyman, Molecular studies of evolution: a source of novel statistical problems, in: S.S. Gupta, J. Yackel (Eds.), Statistical Decision Theory and Related Topics, Academic Press, New York, 1971, pp. 1–27.

[41] K. Rice, personal communication.

[42] K. Rice, T. Warnow, Parsimony is hard to beat, in: T. Jiang, D.T. Lee (Eds.), COCOON'97, Computing and Combinatorics, 3rd Annual Internat. Conf., Shanghai, August 20–22, 1997, Lecture Notes in Computer Science, vol. 1276, Springer, Berlin, pp. 124–133.

[43] N. Saitou, M. Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees, Mol. Biol. Evol. 4 (1987) 406–425.

[44] N. Saitou, T. Imanishi, Relative efficiencies of the Fitch-Margoliash, maximum parsimony, maximum likelihood, minimum evolution, and neighbor-joining methods of phylogenetic tree construction in obtaining the correct tree, Mol. Biol. Evol. 6 (1987) 514–525.

[45] Y.A. Smolensky, A method for linear recording of graphs, USSR Comput. Math. Phys. 2 (1969) 396–397.

[46] M.A. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, J. Classification 9 (1992) 91–116.

[47] M.A. Steel, Recovering a tree from the leaf colourations it generates under a Markov model, Appl. Math. Lett. 7 (1994) 19–24.

[48] M. Steel, M.D. Hendy, D. Penny, Reconstructing phylogenies from nucleotide pattern probabilities: a survey and some new results, Discrete Appl. Math. 89 (1999) 367–396.

[49] M.A. Steel, L.A. Székely, M.D. Hendy, Reconstructing trees when sequence sites evolve at variable rates, J. Comput. Biol. 1 (1994) 153–163.

[50] K. Strimmer, A. von Haeseler, Quartet Puzzling: a quartet Maximum Likelihood method for reconstructing tree topologies, Mol. Biol. Evol. 13 (1996) 964–969.

[51] K. Strimmer, N. Goldman, A. von Haeseler, Bayesian probabilities and Quartet Puzzling, Mol. Biol. Evol. 14 (1997) 210–211.

[52] D.L. Swofford, G.J. Olsen, P.J. Waddell, D.M. Hillis, Phylogenetic inference, Ch. 11, in: D.M. Hillis, C. Moritz, B.K. Mable (Eds.), Molecular Systematics, 2nd ed., Sinauer Associates, Inc., Sunderland, 1996, pp. 407–514.

[53] D. Harel, R.E. Tarjan, Fast algorithms for finding nearest common ancestors, SIAM J. Comput. 13 (1984) 338–355.

[54] N. Takezaki, M. Nei, Inconsistency of the maximum parsimony method when the rate of nucleotide substitution is constant, J. Mol. Evol. 39 (1994) 210–218.

[55] L. Vawter, W. Brown, Nuclear and mitochondrial DNA comparisons reveal extreme rate variation in the molecular clock, Science 234 (1986) 194–196.

[56] L. Vawter, W. Brown, Rates and patterns of base change in the small subunit ribosomal RNA gene, Genetics 134 (1993) 597–608.

[57] L. Wang, D. Gusfield, Improved approximation algorithms for tree alignment, J. Algorithms 25 (1997) 255–273.

[58] L. Wang, T. Jiang, On the complexity of multiple sequence alignment, J. Comput. Biol. 1 (1994) 337–348.

[59] L. Wang, T. Jiang, E. Lawler, Approximation algorithms for tree alignment with a given phylogeny, Algorithmica 16 (1996) 302–315.

[60] L. Wang, T. Jiang, D. Gusfield, A more efficient approximation scheme for tree alignment, in: Proc. 1st Annual Internat. Conf. on Computational Molecular Biology, January 1997, Santa Fe, NM, USA.

[61] T. Warnow, Combinatorial algorithms for constructing phylogenetic trees, Ph.D. Thesis, University of California-Berkeley, 1991.

[62] M.S. Waterman, T.F. Smith, M. Singh, W.A. Beyer, Additive evolutionary trees, J. Theoret. Biol. 64 (1977) 199–213.

[63] A.C. Wilson, R.L. Cann, The recent African genesis of humans, Scientific Amer. 266 (1992) 68–73.

[64] K.A. Zaretsky, Reconstruction of a tree from the distances between its pendant vertices, Uspekhi Math. Nauk (Russian Mathematical Surveys) 20 (1965) 90–92 (in Russian).

COMBINATORICA
Bolyai Society – Springer-Verlag

HOW TO SPLIT ANTICHAINS IN INFINITE POSETS*

PÉTER L. ERDŐS, LAJOS SOUKUP

Received May 17, 2005

A maximal antichain A of poset P *splits* if and only if there is a set $B \subset A$ such that for each $p \in P$ either $b \leq p$ for some $b \in B$ or $p \leq c$ for some $c \in A \setminus B$. The poset P is *cut-free* if and only if there are no $x < y < z$ in P such that $[x,z]_P = [x,y]_P \cup [y,z]_P$. By [1] every maximal antichain in a finite cut-free poset splits. Although this statement for infinite posets fails (see [2])) we prove here that if a maximal antichain in a cut-free poset "resembles" to a finite set then it splits. We also show that a version of this theorem is just equivalent to Axiom of Choice.

We also investigate possible strengthening of the statements that "A does not split" and we could find a maximal strengthening.

1. Introduction

Given a poset $\mathcal{P} = (P, <)$ and subset $A \subset P$ we define the *upset* $A^{\uparrow}$ and the *downset* $A^{\downarrow}$ of A as follows:

$$A^{\uparrow} = \{p \in P : \exists a \in A \ a \leq_P p\}$$

and

$$A^{\downarrow} = \{p \in P : \exists a \in A \ p \leq_P a\}.$$

An *antichain* in P is a set of pairwise incomparable elements. If A is a maximal antichain in P then clearly $P = A^{\uparrow} \cup A^{\downarrow}$. We say that A *splits* if

Mathematics Subject Classification (2000): ....(please fill in!)

* This work was supported, in part, by Hungarian NSF, under contract Nos. T37846, T34702, T37758, AT 048 826, NK 62321. The second author was also supported by Bolyai Grant.

there is $B \subset A$ such that $P = B^{\uparrow} \cup (A \setminus B)^{\downarrow}$. Some maximal antichain may split in a trivial way: e.g. $P = A^{\uparrow}$. Some antichains can not split for the following trivial reason: there are $x, y, z \in P$ such that $x <_P y <_P z$ and y is the only element in the antichain which is comparable to x or z.

Let us remark that the splitting property can be considered as a generalization of property-B, for an explanation see [7].

You can not expect an "easy" characterization of the maximal antichains in finite posets which splits because this question is NP-complete, see [1]. However in the same paper it was also shown that if a finite poset P has a property which is just a bit stronger than the lack of above type obstacle points y then every maximal antichain of P splits. To recall that result we should introduce some new notions.

An element $y \in P$ is called *cutting point* if and only if there are $x, z \in P$ such that $x <_P y <_P z$ and $[x, z] = [x, y] \cup [y, z]$. (The interval $[x, z] = \{\forall y \in P : x \leq y \leq z\}$.) We say that P is *cut-free* if there is no cutting point in it. (This property was called *dense*, see e.g. [1], but the current wording seems to be more adequate.)

Theorem 1.1 ([1, Theorem 3.1]). *Let $\mathcal{P}$ be a finite cut-free poset. Then every maximal antichain A splits.*

This result yields immediately following question: what about infinite posets?

Ahlswede and Khachatrian showed ([2]) that the plain generalization of Theorem 1.1 for infinite posets fails: the finite-subset-lattice $\left\langle [\omega]^{<\omega}, \subset \right\rangle$, which is cut-free, contains an infinite antichain which does not split.

In Section 2 we prove Theorem 2.7 saying that if a maximal antichain of an infinite poset satisfies some extra assumptions than it splits. This result yield that if a maximal antichain of a cut-free poset "resembles" a finite antichain then it splits (see Theorem 2.10).

On the other hand, in Section 3 we show that that the non-splitting behavior of the poset $\left\langle [\omega]^{<\omega}, \subset \right\rangle$ is not exceptional: if an infinite poset is rich enough in elements then it should contain non-splitting antichains, see Theorem 3.6. Let us recall that Ahlswede and Khachatrian use number theory in [2] to construct a non-splitting antichain; our proof is purely combinatorial. Besides this result in Section 3 we also investigate possible strengthening of the statements that "A does not split". To formulate these results we introduce the following notation. If P is a poset and $A \subset P$ is a maximal antichain put

$$\mathfrak{S}(A) = \{\langle B, C \rangle : B \subset A, C \subset A, P = B^{\uparrow} \cup C^{\downarrow}\}.$$

Clearly A splits if and only if there is $\langle B,C\rangle \in \mathfrak{S}(A)$ with $B\cap C=\emptyset$. The maximal strengthening of the above mention result of Ahlswede and Khachatrian would be a cut-free poset P and a maximal antichain $A\subset P$ with $\mathfrak{S}(P)=\{\langle A,A\rangle\}$, but Corollary 3.3 says that this is not possible. In Theorem 3.8 we show that Theorem 3.6 is the maximal possible strengthening.

Quite surprisingly, the technique we developed to construct non-splitting antichain can be used to build splitting antichains as well, see Theorem 3.9.

Our notation is standard. Put $A^{\updownarrow}=A^{\downarrow}\cup A^{\uparrow}$. If $x\in P$ write $x^{\uparrow}$ for $\{x\}^{\uparrow}$, $x^{\downarrow}$ for $\{x\}^{\downarrow}$ and $x^{\updownarrow}$ for $\{x\}^{\updownarrow}$. If $A\subset P$ and $\mathcal{P}$ is not clear form the context we write $A^{\uparrow_{\mathcal{P}}}$ for $A^{\uparrow}$, and $A^{\downarrow_{\mathcal{P}}}$ for $A^{\downarrow}$. On the poset $\mathcal{P}$ we always think the poset $\mathcal{P}=(P,<)$.

2. Positive theorems

Definition 2.1. Let $\mathcal{P}$ be a poset and $A\subset P$ be a maximal antichain. An element $x\in A^{\downarrow}\setminus A$ is *high* if and only if there is no $y\in x^{\uparrow}\cap A^{\downarrow}$ with $y^{\uparrow}\cap A \underset{\neq}{\subseteq} x^{\uparrow}\cap A$. An element $z\in A^{\uparrow}\setminus A$ is *low* if and only if there is no $v\in z^{\downarrow}\cap A^{\uparrow}$ with $v^{\downarrow}\cap A \underset{\neq}{\subseteq} z^{\downarrow}\cap A$.

Lemma 2.2. *If $\mathcal{P}$ is a poset, $A\subset P$ is a maximal antichain which does not contain cutting points, $x\in A^{\downarrow}\setminus A$ is high and $z\in A^{\uparrow}\setminus A$ is low then* $|[x,z]\cap A|\neq 1$.

Proof. Assume on the contrary that $[x,z]\cap A=\{y\}$. Since y is not a cutting point there is $u\in[x,z]$ such that y and u are incomparable. By the indirect assumption we have $u\notin A$. If $u\in A^{\uparrow}$ then $u^{\downarrow}\cap A\subset(z^{\downarrow}\cap A)\setminus\{y\}$, i.e. z is not low. Hence $u\in A^{\downarrow}$. But then $u^{\uparrow}\cap A\subset(x^{\uparrow}\cap A)\setminus\{y\}$, i.e. x is not high. Contradiction. ∎

Definition 2.3. Given a family $\mathcal{A}\subset\mathcal{P}(X)$ a well-ordering $\prec$ of X is called *maximizing well-ordering for* $\mathcal{A}$ if and only if $\max_{\prec}A$ exists for each $A\in\mathcal{A}$. The family $\mathcal{A}$ is said to be *maximizing* if and only if there is a maximizing well-ordering for $\mathcal{A}$.

For example, the family $[X]^{<\omega}$ is clearly maximizing because any well-ordering of X is maximizing for this family.

If $\mathcal{A}\subset\mathcal{P}(X)$ and $\prec$ is a well-ordering of X let $\mathrm{MIN}(\mathcal{A},\prec)=\{\min_{\prec}A:A\in\mathcal{A}\}$ and $\mathrm{MAX}(\mathcal{A},\prec)=\{\max_{\prec}A:A\in\mathcal{A}$ and $\max_{\prec}A$ exists$\}$.

In [9] Klimó gave a characterization of maximizing families. Although he used a different terminology we can formulate his result as follows:

Theorem 2.4 **([9, Theorem 7]).** *$\mathcal{A}\subset\mathcal{P}(X)$ is a maximizing family if and only if there is a function $f:\mathcal{A}\to X$ such that $f(A)\in A$ for each $A\in\mathcal{A}$ and there is no sequence $\langle A_i:i<\omega\rangle$ in $\mathcal{A}$ such that $f(A_i)\neq f(A_{i+1})\in A_i$ for each $i<\omega$ and the set $\{A_i:i<\omega\}$ is infinite.*

Definition 2.5. Given a family $\mathcal{A}\subset\mathcal{P}(X)$ a set $Y\subset X$ is called a *point cover* if and only if $A\cap Y\neq\emptyset$ for each $A\in A$. Y is a *minimal point cover* if and only if it is a point cover but no proper subset of Y is a point cover.

The following lemma gives us a method to construct splits of certain antichains in certain posets.

Lemma 2.6. *Let $\mathcal{P}$ be a poset and $A\subset P$ be a maximal antichain. Assume that there are two functions $\underline{\mathrm{B}}$ and $\overline{\mathrm{B}}$ such that*

(i) $\underline{\mathrm{B}}:A^{\uparrow}\setminus A\to\mathcal{P}(A)$ *and* $\emptyset\neq\underline{\mathrm{B}}(y)\subset A\cap y^{\downarrow}$ *for each* $y\in A^{\uparrow}\setminus A$,
(ii) $\overline{\mathrm{B}}:A^{\downarrow}\setminus A\to\mathcal{P}(A)$ *and* $\emptyset\neq\overline{\mathrm{B}}(x)\subset A\cap x^{\uparrow}$ *for each* $x\in A^{\downarrow}\setminus A$,
(iii) $|\underline{\mathrm{B}}(y)\cap\overline{\mathrm{B}}(x)|\neq 1$ *for each* $x\in A^{\downarrow}\setminus A$ *and* $y\in A^{\uparrow}\setminus A$

Write $\overline{\mathcal{B}}=\{\overline{\mathrm{B}}(x):x\in A^{\downarrow}\setminus A\}$ *and* $\underline{\mathcal{B}}=\{\underline{\mathrm{B}}(x):x\in A^{\uparrow}\setminus A\}$.

(1) *If $\prec$ is a maximizing well-ordering of $\overline{\mathcal{B}}$ then* $\mathrm{MIN}(\underline{\mathcal{B}},\prec)\cap\mathrm{MAX}(\overline{\mathcal{B}},\prec)=\emptyset$, *and so $\mathcal{A}$ splits.*
(2) *If $C\subset A$ is a minimal point cover of $\overline{\mathcal{B}}$ then $\langle A\setminus C,C\rangle\in\mathfrak{S}(A)$ and so A splits.*

Proof. (1) Indeed, $\max_{\prec}\overline{\mathrm{B}}(x)=\min_{\prec}\underline{\mathrm{B}}(y)$ would imply that $\overline{\mathrm{B}}(x)\cap\underline{\mathrm{B}}(y)=\{\max_{\prec}\overline{\mathrm{B}}(x)\}$ which contradicts to property (iii) in the choice of $\underline{\mathrm{B}}$ and $\overline{\mathrm{B}}$.

Since clearly $A^{\downarrow}\setminus A\subset\mathrm{MIN}(\overline{\mathcal{B}},\prec)^{\downarrow}$ and $A^{\uparrow}\setminus A\subset\mathrm{MAX}(\underline{\mathcal{B}},\prec)^{\uparrow}$ we have that A splits.

(2) Since C is a point cover we have $A^{\downarrow}\setminus A\subset C^{\downarrow}$. To prove the other property assume on the contrary that $A^{\uparrow}\setminus A\not\subset(A\setminus C)^{\uparrow}$, i.e. there is $y\in A^{\uparrow}\setminus A$ such that $\underline{\mathrm{B}}(y)\subset C$. Pick an arbitrary $z\in\underline{\mathrm{B}}(y)$. Since $C\setminus\{z\}$ is not a point cover of $\overline{\mathcal{B}}$ there is $x\in A^{\downarrow}\setminus A$ such that $\overline{\mathrm{B}}(x)\cap C=\{z\}$. But then $\{z\}\subset\overline{\mathrm{B}}(x)\cap\overline{\mathrm{B}}(y)\subset\overline{\mathrm{B}}(x)\cap C=\{z\}$ which contradicts (iii). ∎

Theorem 2.7. *Let $\mathcal{P}$ be a poset and $A\subset P$ be a maximal antichain which does not contain cutting points. Assume that*

(i) *for each $y\in A^{\uparrow}\setminus A$ there is a low $z\in A^{\uparrow}\setminus A$ with $z\leq y$,*
(ii) *for each $x\in A^{\downarrow}\setminus A$ there is a high $t\in A^{\downarrow}\setminus A$ such that $x\leq t$,*

If either

(1) *the family $\{x^{\uparrow}\cap A:x$ is high $\}$ is maximizing or*
(2) *the family $\{x^{\uparrow}\cap A:x$ is high $\}$ has a minimal point cover*

then A splits.

Proof. Let $L=\{y\in A^{\uparrow}\setminus A: y \text{ is low}\}$ and $H=\{x\in A^{\downarrow}\setminus A: x \text{ is high}\}$. Let $M=A\cup H\cup L$ and let Q be the subordering of P with the underlining set M. Since $[x,y]\subset M$ for each $\{x,y\}\in[M]^2$, (i.e. M is "convex" in P) the antichain A does not contain cutting points in Q.

Since A is clearly a maximal antichain in Q, every element of H is high in Q and every element of L is low. Thus, by Lemma 2.2, we have

$$|[x,y]\cap A|\neq 1 \text{ for each } x\in H \text{ and } y\in L. \tag{1}$$

Let $\overline{\mathrm{B}}(x)=x^{\uparrow}\cap A$ and $\underline{\mathrm{B}}(y)=y^{\downarrow}\cap A$. We want to apply Lemma 2.6. Properties (i)–(ii) are clear. Since $\overline{\mathrm{B}}(x)\cap\underline{\mathrm{B}}(y)=[x,y]\cap A$, property (1) implies that the functions $\underline{\mathrm{B}}$ and $\overline{\mathrm{B}}$ satisfies Lemma 2.6.(iii).

Since (1) implies Lemma 2.6.(1), and (2) implies Lemma 2.6.(2) hence we have that A splits in Q: there is $B\subset A$ such that $B^{\uparrow}=L$ and $(A\backslash B)^{\downarrow}=H$ in Q. Since $L^{\uparrow}=A^{\uparrow}\setminus A$ in P and $H^{\downarrow}=L^{\downarrow}\setminus A$ in P we have that $B^{\uparrow}=A^{\uparrow}\setminus A$ and $(A\setminus B)^{\downarrow}=A^{\downarrow}\setminus A$ in P. Thus B witnesses that A splits. ∎

Let us remark the nontrivial fact that condition (1) is stronger than (2): as Klimó proved in [9] a maximizing family $\mathcal{A}$ has a minimal point cover. However we included the statement with proof here because you can get two different splits for A when $\{x^{\uparrow}\cap A: x \text{ is high }\}$ is maximizing: one applying Lemma 2.6.(1) directly and the other by finding a minimal point cover for $\{x^{\uparrow}\cap A: x \text{ is high }\}$ and then applying Lemma 2.6.(2).

A poset $\mathcal{P}=\langle P,<\rangle$ is called *well-founded* (or satisfies the *Descending Chain Condition*), if there exists no infinite descending chain: if $x_1\geq x_2\geq x_n\geq\dots$ then there exists an integer i such that $x_i=x_j$ for all $j>i$.

Theorem 2.8. *Let $\mathcal{P}$ be a well-founded poset and let A be a maximal, cutting point free antichain, such that for every $p\in A^{\downarrow}\setminus A$ there exists element $x(p)\in A^{\downarrow}\setminus A$ with $p\leq a(p)$ such that $a(p)^{\uparrow}\cap A$ is finite. Then A splits.*

Proof. We want to apply Theorem 2.7. Property (ii) holds by assumptions. Moreover $x^{\uparrow}\cap A$ is finite for each high elements and so Property (1) holds. The minimal elements of $A^{\uparrow}\setminus A$ are all low, hence Property (i) also holds. ∎

The next observation provides a very useful tool to manipulate the antichain pairs in $\mathfrak{S}(A)$ of maximal antichains in cut-free posets.

Lemma 2.9. *Assume that $\mathcal{P}$ is a poset, $A\subset P$ is a maximal antichain, and $\langle B,C\rangle\in\mathfrak{S}(A)$. Then for each $y\in B\cap C$ if y is not a cutting point then either $\langle B\setminus\{y\},C\rangle\in\mathfrak{S}(A)$ or $\langle B,C\setminus\{y\}\rangle\in\mathfrak{S}(A)$.*

Proof. Assume on the contrary that this is not true, so there are $x, z \in P$ such that $x < y < z$, $x \notin (C \setminus \{y\})^{\downarrow}$ and $z \notin (B \setminus \{y\})^{\uparrow}$. Since y is not a cutting point, there is $t \in [x, z]$ such that y and t are incomparable. Then $t \in (B \setminus \{y\})^{\uparrow} \cup (C \setminus \{y\})^{\downarrow}$. If $y' < t$ for some $y' \in B \setminus \{y\}$ then $y' < z$, contradiction. If $t < y'$ for some $y' \in C \setminus \{y\}$ then $x < y'$, contradiction. ∎

Theorem 2.10. *Let A be a maximal antichain in the poset $\mathcal{P}$ such that A does not contain cutting points and*

$$|(x^{\updownarrow}) \cap A| < \omega \text{ for all } x \in P,$$

then A splits.

This result is a direct generalization of Theorem 1.1 ([1]). We give here two different proofs. However it is not clear yet the complexity of these methods to find a splitting (at least of the second one) in the case of finite cut-free posets. It is also a question whether all possible splitting arise along the second method.

First proof. Consider the poset $Q(P) = \langle \mathfrak{S}(A), \prec \rangle$ where $\langle B, C \rangle \prec \langle B', C' \rangle$ if and only if $B \supset B'$ and $C \supset C'$.

We want to apply the Zorn lemma to find a maximal elements of $Q(P)$. So let $\langle \langle B_\xi, C_\xi \rangle : \xi < \eta \rangle$ be an increasing chain in $Q(P)$. Put $B = \cap \{B_\xi : \xi < \eta\}$ and $C = \cap \{C_\xi : \xi < \eta\}$. Let $x \in P$ be arbitrary. Since $(x^{\updownarrow}) \cap A$ is finite there is $\zeta < \eta$ such that $(x^{\updownarrow}) \cap B = (x^{\updownarrow}) \cap B_\zeta$ and $(x^{\updownarrow}) \cap C = (x^{\updownarrow}) \cap C_\zeta$. Since $x \in B_\zeta^{\uparrow} \cup C_\zeta^{\downarrow}$ we have $x \in B^{\uparrow} \cup C^{\downarrow}$. Since x was arbitrary we have $\langle B, C \rangle \in \mathfrak{S}(A)$, and so $\langle B, C \rangle$ is the required upper bound of $\langle \langle B_\xi, C_\xi \rangle : \xi < \eta \rangle$.

Thus the Zorn lemma implies that $Q(P)$ has a maximal element $\langle B, C \rangle$. But then $B \cap C = \emptyset$ by Lemma 2.9. ∎

Second proof. Apply Theorem 2.7. Since (1) and (2) clearly holds we can apply that result to get that A splits. ∎

Finally we give one more application of Theorem 2.7: we prove a theorem on the subset lattice of the natural numbers.

Let A be a maximal antichain in $\mathcal{P}(\omega)$ and let $x \in (A^{\downarrow} \setminus A)$. Denote $\text{Card}(A)$ the set of the cardinalities present in A, and denote $\text{Card}_x(A)$ the set of cardinalities of those elements in A which are comparable to x. We say that this x *behaves well* if $|\text{Card}_x(A)| = \omega$ then $\omega \in \text{Card}_x(A)$ as well. If, for example, $|\text{Card}(A)|$ is finite, then every element behaves well.

Theorem 2.11. *Let A be a maximal antichain in $\mathcal{P}(\omega)$. Assume that*

$$\forall y \in (A^{\uparrow} \setminus A) \quad : \quad [y^{\downarrow} \cap A \cap [\omega]^{<\omega}] \neq \emptyset, \tag{2}$$

furthermore every element $x \in A^{\downarrow} \setminus A$ behaves well. Then A splits.

Proof. Let $I = A\cap[\omega]^\omega$, $F = A\cap[\omega]^{<\omega}$ and $Q = \mathcal{P}(\omega)\setminus I^\downarrow$. Clearly F is a maximal antichain in Q. Next we show that:

Claim 2.12. *For each $c\in(F^\downarrow\setminus F)\cap Q$ there is a high $h\in Q$ with $c\subset h$.*

$\mathrm{Card}_c(F)$ is finite, because c behaves well. Write $n=\max\mathrm{Card}_x(F)$. Fix $f\in F\cap c^\uparrow\cap[\omega]^n$ and pick $h\in[\omega]^{n-1}$ with $c\subset h\subset f$. Then h is high in Q because it is maximal in $(F^\downarrow\setminus F)\cap Q$.

Claim 2.13. *For each $b\in F^\uparrow\setminus F$ there is a low $\ell\in Q$ such that $\ell\subset b$ and $\ell^\downarrow\cap F$ is finite.*

Indeed, let $j=\min\{|f|:f\in F\cap b^\downarrow\}$, pick $f\in b^\downarrow\cap F\cap[\omega]^j$ and let $\ell\in[\omega]^{j+1}$ with $f\subset\ell\subset b$. Then ℓ is minimal in $F^\downarrow\setminus F$ hence it is low in Q. Moreover $\ell^\downarrow\cap F$ is clearly finite.

Hence we can apply Theorem 2.7 for Q^{-1} (the dual of poset Q) and F to yield that F splits in Q: there is $G\subset F$ such that $G^\uparrow\setminus G = F^\uparrow\setminus G$ and $F\setminus G^\downarrow = F^\downarrow$.

Then G shows that A splits in P. Indeed, $F^\uparrow = A^\uparrow$ because of assumption (2). Hence $G^\uparrow=A^\uparrow$ in P. On the other hand, if $c\in A^\downarrow$ then either $c\in Q$ and so $c\in(F\setminus G)^\downarrow$, or $c^\uparrow\cap A\cap[\omega]^\omega\neq\emptyset$ and so $c\in(A\setminus F)^\downarrow\subset(A\setminus G)^\downarrow$. ∎

3. Negative theorems

In this Section we study maximal antichains of countable posets, together the possible structures of non-splitting maximal antichains.

To start we give some consequences of Lemma 2.9. At first we have:

Corollary 3.1. *If a maximal antichain A does not split in a cut-free poset $\mathcal{P}$ then $|B\cap C|=\omega$ for each $\langle B,C\rangle\in\mathfrak{S}(A)$.*

Which in turns gives a direct generalization of Theorem 1.1:

Corollary 3.2. *Every finite maximal antichain splits in every cut-free poset.*

We think that in the future Lemma 2.9 will provide the standard proof of Theorem 1.1. Lemma 2.9 also shows that in cut-free posets there are no maximal antichains A with maximally degenerated $\mathfrak{S}(A)$:

Corollary 3.3. *There exits no cut-free poset $\mathcal{P}$ such that $\mathfrak{S}(A)=\{\langle A,A\rangle\}$ for some maximal antichain $A\subset P$.*

On the other hand, in Theorem 3.6 below we show that the structure of $\mathfrak{S}(A)$ can be quite degenerated: it might happen that every pair in $\mathfrak{S}(A)$ contains A itself. To formulate this result we need one more definition.

Definition 3.4. A poset $\mathcal{P}$ is *loose* if and only if for each $x \in P$ and $F \in [P]^{<\omega}$ if $x \notin F^{\uparrow}$ then there is $y \in x^{\uparrow} \setminus \{x\}$ such that $y \notin F^{\downarrow} \cup F^{\uparrow}$.

Assume that $\mathcal{P}$ is loose and $p \in P$. Let $F = \emptyset$. Then $p \notin F^{\uparrow}$ hence by sleaziness there is $y \in P$ with $y \in x^{\uparrow} \setminus \{x\}$, i.e. $y > x$. Thus we have:

Remark. A loose poset does not have maximal elements. Especially, it is infinite.

Claim 3.5. *$\left\langle [\omega]^{<\omega}, \subset \right\rangle$ is loose.*

Proof. Indeed, if $x \in [\omega]^{<\omega}$ and F is a finite subset of $[\omega]^{<\omega}$ with $x \notin F^{\uparrow}$ then let n be a natural number not belonging to x or any set in F, and put $y = x \cup \{n\}$. Let $f \in F$. Then $\emptyset \neq f \setminus x = f \setminus y$ hence $y \notin F^{\uparrow}$. Moreover, $n \in y \setminus f$ and so $y \notin F^{\downarrow}$. ∎

Theorem 3.6. *Assume that $\mathcal{P} = \langle P, \leq \rangle$ is a countable, loose poset. Then $\mathcal{P}$ contains a maximal antichain A such that*

(i) *if $\langle B, C \rangle \in \mathfrak{S}(A)$ then $B = A$,*
(ii) *if A is finite then $\cap\{C : \langle B, C \rangle \in \mathfrak{S}(A)\} \neq \emptyset$,*
(iii) *if A is infinite then so is C for each $\langle B, C \rangle \in \mathfrak{S}(A)$,*
(iv) *if $\mathcal{P}$ is cut-free then A is infinite.*

Proof. Let $\langle p_n : n < \omega \rangle$ be an enumeration of the elements of P. By induction on $n \in \omega$ we choose elements $x_n, y_n, z_n \in P$ with $x_n < y_n < z_n$ as follows.

Let $m_n = \min\{m : p_m \notin \{y_i : i < n\}^{\uparrow} \cup \{y_i : i < n\}^{\downarrow}\}$. If m_n is not defined then the we stop the construction. Assume that m_n is defined. Since $y_i < z_i$ we have $p_{m_n} \notin \{y_i, z_i : i < n\}^{\uparrow}$. Furthermore since $\mathcal{P}$ is loose there is $x_n \in P$ with $p_{m_n} < x_n$ such that

$$x_n \notin \{y_i, z_i : i < n\}^{\uparrow} \cup \{y_i, z_i : i < n\}^{\downarrow}.$$

Applying the sleaziness of $\mathcal{P}$ once more there is $y_n \in P$ with $x_n < y_n$ such that

$$y_n \notin \{y_i, z_i : i < n\}^{\uparrow} \cup \{y_i, z_i : i < n\}^{\downarrow}. \tag{3}$$

Applying the sleaziness of $\mathcal{P}$ a third time there is $z_n \in P$ with $y_n < z_n$ such that

$$z_n \notin \{y_i, z_i : i < n\}^{\uparrow} \cup \{y_i, z_i : i < n\}^{\downarrow}. \tag{4}$$

We claim that $A=\{y_n:n<\omega, y_n \text{ is defined}\}$ has the required properties. First observe, that A is an antichain by Property (3).

By induction on n we can see that $m_n\ge n$ and so $p_n\in\{y_i:i<n\}^{\uparrow}\cup\{y_i:i<n\}^{\downarrow}\cup y_n{}^{\downarrow}$, hence the antichain A is maximal.

Assume that $\langle B,C\rangle\in\mathfrak{S}(A)$. Let n be arbitrary such that m_n is defined. By Property (4) we have $z_n\notin\{y_i:i<n\}^{\uparrow}\cup\{y_i:i<n\}^{\downarrow}$. By Property (3) we have $z_n\notin\{y_i:i>n\}^{\uparrow}\cup\{y_i:i>n\}^{\downarrow}$. Since $y_n<z_n$ we have $z_n\notin A^{\downarrow}\cup(A\setminus\{y_n\})^{\uparrow}$. Thus $z_n\in B^{\uparrow}\cup C^{\downarrow}$ implies then $y_n\in B$. Hence $B=A$. (That is (i) holds.)

Since $p_{m_n}<y_n$ we have $p_{m_n}\notin A^{\uparrow}$. By the choice of m_n we have $p_{m_n}\notin\{y_i,:i<n\}^{\uparrow}\cup\{y_i:i<n\}^{\downarrow}$. Thus $p_{n_m}\in\{y_k\in C:k\ge n\}^{\downarrow}$. Hence $\{m:x_m\in C\}$ is cofinal in $\{m:x_m \text{ is defined}\}$. Therefore $y_n\in C$ provided that A is finite and $n=\max\{n':m_{n'} \text{ is defined}\}$ and so (ii) holds, and C is infinite provided that A is infinite. (That is (iii) holds.) Let's remark that one can prove (iii) by observing that if C would be finite then Lemma 2.9 and Property (i) together would prove that $\langle B,\emptyset\rangle\in\mathfrak{S}(A)$, a clear contradiction.

Properties (ii) and (iii) imply that A does not split. Since, according to Corollary 3.2, finite antichains split in a cut-free posets we have that A is infinite provided that P is cut-free. (That is (iv) holds.) ∎

Since $\langle[\omega]^{<\omega},\subset\rangle$ is loose and cut-free, we can apply Theorem 3.6 to get the following corollary.

Corollary 3.7. *$\langle[\omega]^{<\omega},\subset\rangle$ contains a maximal antichain A such that if $\langle B,C\rangle\in\mathfrak{S}(A)$ then $A=B$ and C is infinite, and so A does not split.*

This result is a farfetched generalization of the construction given by Ahlswede and Khachatrian in [2].

The following result shows that even more can be said about maximal antichains A in cut-free posets, where every pair in $\mathfrak{S}(A)$ contains A itself, showing also that Theorem 3.6 is sharp in a certain sense.

Theorem 3.8. *Assume that $\mathcal{P}=\langle P,\le\rangle$ is a countable, cut-free poset, $A\subset P$ is a maximal antichain such that $A=B$ for each $\langle B,C\rangle\in\mathfrak{S}(A)$. Then there is $\langle A,C\rangle\in\mathfrak{S}(A)$ with $|A\setminus C|=\omega$.*

Proof. Since $\langle A\setminus\{a\},A\rangle\notin\mathfrak{S}(A)$ we can pick $z_a\in P$ such that $a<z_a$ and $z_a\notin(A\setminus\{a\})^{\uparrow}$ for each $a\in A$.

We claim that $x^{\uparrow}\cap A$ is infinite for each $x\in A^{\downarrow}\setminus A$ and this statement finishes the proof: Indeed, in this case there is $C\in[A]^{\omega}$ such that $|(x^{\uparrow}\cap A)\cap C|=|(x^{\uparrow}\cap A)\setminus C|=\omega$ for each $x\in A^{\downarrow}\setminus A$, and so $\langle A,C\rangle\in\mathfrak{S}(A)$ with $|A\setminus C|=\omega$. (This is the well-known Bernstein's Lemma [3].)

To prove our claim assume on the contrary that $B=x^{\uparrow}\cap A$ is finite for some $x\in A^{\downarrow}\setminus A$. Choose x such that $|B|$ is minimal. Clearly $|B|>0$. Let

$y \in B$ be arbitrary. Then $x < y < z_y$ and $\mathcal{P}$ is cut-free so there is $t \in [x, z_y]$ which is incomparable with y.

Now $t \in A^{\downarrow}$ because $a \leq t$ would imply $a < z_y$ and so $a = y$ for any $a \in A$, but t and y were incomparable. Moreover $t^{\uparrow} \cap A \subset (x^{\uparrow} \cap A) \setminus \{y\}$, which contradicts the minimality of the cardinality of $x^{\uparrow} \cap A$. ∎

Till now we used the sleaziness to show that certain antichain can not split, or to restrict the structure of $\mathfrak{S}(A)$. The next theorem shows that the sleaziness can be used even in the other direction: to guarantee the existence of splitting antichains.

Theorem 3.9. *Assume that $\mathcal{P} = \langle P, \leq \rangle$ is a countable poset such that $\mathcal{P}$ and $\mathcal{P}^{-1}$ are loose. Then $\mathcal{P}$ contains a maximal antichain A which splits.*

Proof. Write $\mathcal{P} = \{p_n : n < \omega\}$. By induction on n we will construct finite disjoint subsets B_n and C_n of P such that

(i) $B_n \cup C_n$ is an antichain,
(ii) $B_{n-1} \subset B_n$ and $C_{n-1} \subset C_n$,
(iii) $p_{n-1} \in B_n^{\uparrow} \cup C_n^{\downarrow}$.

It is enough to show that we can carry out the induction because taking $B = \cup\{B_n : n \in \omega\}$ and $C = \cup\{C_n : n \in \omega\}$ we have that $A := B \cup C$ is a maximal antichain having the splitting $\langle B, C \rangle$.

Let $B_0 = C_0 = \emptyset$. Assume that B_{n-1} and C_{n-1} are constructed. Write $p = p_{n-1}$. If $p \in B_{n-1}^{\uparrow} \cup C_{n-1}^{\downarrow}$ then let $C_n = C_{n-1}$ and $B_n = B_{n-1}$. So we can assume that $p \notin B_{n-1}^{\uparrow} \cup C_{n-1}^{\downarrow}$.

Case 1. $p \notin C_{n-1}^{\uparrow}$.

Then $p \notin (C_{n-1} \cup B_{n-1})^{\uparrow}$. Since $\mathcal{P}$ is loose there is $p \leq q$ such that $q \notin (C_{n-1} \cup B_{n-1})^{\uparrow} \cup (C_{n-1} \cup B_{n-1})^{\downarrow}$, i.e. $B_{n-1} \cup C_{n-1} \cup \{q\}$ is an antichain. Let $C_n = C_{n-1} \cup \{q\}$ and $B_n = B_{n-1}$. Then $p \in q^{\downarrow} \subset C_n^{\downarrow}$, B_n and C_n are disjoint and $B_n \cup C_n$ is an antichain.

Case 2. $p \notin B_{n-1}^{\downarrow}$.

Then $p \notin (B_{n-1} \cup C_{n-1})^{\downarrow}$. Since $\mathcal{P}^{-1}$ is loose there is $q \leq p$ such that $q \notin (B_{n-1} \cup C_{n-1})^{\downarrow} \cup (B_{n-1} \cup C_{n-1})^{\uparrow}$, i.e. $C_{n-1} \cup B_{n-1} \cup \{q\}$ is an antichain. Let $B_n = B_{n-1} \cup \{q\}$ and $C_n = C_{n-1}$. Then $p \in q^{\uparrow} \subset B_n^{\uparrow}$, C_n and B_n are disjoint and $C_n \cup B_n$ is an antichain.

Case 3. $p \in B_{n-1}^{\downarrow} \cap C_{n-1}^{\uparrow}$.

Then there is $b\in B_{n-1}$ and $c\in C_{n-1}$ such that $c\le p\le b$, i.e. $B_{n-1}\cup C_{n-1}$ is not an antichain. Contradiction, this case is not possible which finishes the proof. ∎

The maximal antichains in the poset $\mathbb{Z}$ of the integer are the singletons and they clear don't split.

Problem 3.10. Is there a countable cut-free poset P which does not contain splitting maximal antichains?

Consider the following countable, well-founded, cut-free poset. Let the underlying set of $\mathcal{P}$ be $\omega\times\omega$. Put $\langle n,m\rangle <_P \langle n',m'\rangle$ if and only if $n<n'$. Then the antichains in $\mathcal{P}$ are the sets $\{n\}\times\omega$ for $n<\omega$, and $\{n\}\times\omega$ splits because

$$P=\{\langle n,i\rangle\}^{\downarrow}\cup\{\langle n,j\rangle\}^{\uparrow}$$

whenever $i\neq j$. We do not have any characterization of posets having only splitting maximal antichains.

Till now we were interested the existence of splitting of maximal antichains. One can ask, however, how many different splits can be found.

Problem 3.11. Fix a cardinal κ. Is there a countable cut-free poset P having a maximal antichain A such that

$$\kappa_P \overset{\text{def}}{=} |\{B : \langle B, A\setminus B\rangle \in \mathfrak{S}(A)\}| = \kappa?$$

In general, we do not know the answer. Since $|A|$ is countable 2^A can be considered as a topological space homeomorphic to the ω^{th} power of the two element discrete topological space $\mathbf{2}=\{0,1\}$, i.e. to the Cantor set. Hence we have the Borel hierarchy on 2^A. Since $\mathfrak{S}(A)$ is a G_δ-subset of $2^A\times 2^A$ hence either $\mathfrak{S}(A)$ is at most countable or has cardinality 2^ω by [8, Theorem 11.18(iii)]. The case $\kappa=2^\omega$ is trivial. The case $\kappa=1$ is also trivial: let P be well-founded and A be the minimal points of P. However the κ_P can be 1 in a less trivial way.

Claim 3.12. *There is a countable, cut-free poset P and an infinite maximal antichain $A\subset P$ such that*

(i) $\forall a\in A\ \exists x,y\in P\ x<a<y$,
(ii) $|\{B\subset A:\langle B,A\setminus B\rangle\in\mathfrak{S}(A)\}|=1$.

Proof. Consider the poset Q on Figure 1. The poset Q is cut-free. The set $A=\{b,c\}$ is a maximal antichain in Q and $\mathfrak{S}(A)=\{\langle b,c\rangle\}$. Let P be the disjoint union of countable many copies of Q. ∎

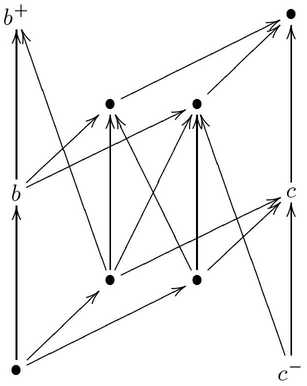


Figure 1. Poset Q

4. Some set-theory

In this section we will use the standard set-theoretical notation throughout, see e.g. [8].

The answers to the questions which we investigated in connection with countable posets in Section 3 does not depend on the actual set-theoretical universe in which we work. The reason is that all the statements can be formulated as a $\Sigma^1_2(a)$ or $\Pi^1_2(a)$ formula with some parameter $a \in \omega^\omega$, and so they are absolute by Schoenfield's absoluteness theorem, [8, Theorem 25.20]. For example, given a countable poset $\mathcal{P}$ and maximal antichain $A \subset \mathcal{P}$ statements like "A splits", or "no maximal antichains of $\mathcal{P}$ splits", or "every maximal antichain of $\mathcal{P}$ splits" are all absolute: their truth value depends on only $\mathcal{P}$ and A and independent of the set-theoretical universe. Same argument gives that although we do not know the answer to the problem 3.10 we can expect a yes or no answer in ZFC.

The situation changes dramatically if we consider uncountable partially ordered sets. We will give an example after Proposition 4.3 that given a poset $\mathcal{P}$ of size ω_1 and maximal antichain $A \subset \mathcal{P}$ the statement "A splits" can depend on the set-theoretical universe in which we live. We will also show that axiom $\clubsuit$ can be reformulated as a statement on splitting property of certain antichains in certain posets, see proposition 4.3.

Definition 4.1. Let $\mathcal{L}$ be the set of the countable limit ordinals. We say that $\langle T_\alpha : \alpha \in \mathcal{L} \rangle$ is a $\clubsuit$-*sequence* if and only if $T_\alpha \subset \alpha$ is cofinal for each $\alpha \in \mathcal{L}$ and for each $X \in [\omega_1]^{\omega_1}$ there is $\alpha \in \mathcal{L}$ with $T_\alpha \subset X$. Axiom $\clubsuit$ holds if and only if there is a $\clubsuit$-sequence.

It is well-known that axiom ♣ is independent from ZFC: there is a ♣ sequence in the constructible universe L of Gödel but Martin's Axiom excludes the existence of such a sequence.

Definition 4.2. Given a sequence $\mathcal{T}=\{T_\beta:\beta\in\mathcal{L}\}$, where $T_\alpha\subset\alpha$ is cofinal, we define the poset $Q(\mathcal{T})$ as follows. The underlying set of $Q(\mathcal{T})$ is $(\{2\}\times\mathcal{L})\cup(2\times\omega_1)$. Let $\langle 0,\eta\rangle\prec\langle 1,\xi\rangle$ if and only if $\eta<\xi$. Let $\langle 1,\zeta\rangle\prec\langle 2,\beta\rangle$ if and only if $\zeta\in T_\beta$. Let $\le_{Q(\mathcal{T})}$ be the partial ordering generated by $\prec$.

The poset $Q(\mathcal{T})$ is clearly cut-free.

Proposition 4.3. *Let $\mathcal{T}=\{T_\beta:\beta\in\mathcal{L}\}$, where $T_\alpha\subset\alpha$ is cofinal. The maximal antichain $A=\{1\}\times\omega_1$ splits in $Q(\mathcal{T})$ if and only if $\mathcal{T}$ is not a ♣-sequence.*

Proof. If $\mathcal{T}$ is not a ♣ sequence then there is $X\in[\omega_1]^{\omega_1}$ such that $T_\alpha\setminus X\neq\emptyset$ for each $\alpha\in\mathcal{L}$. Let $B=\{1\}\times(\omega_1\setminus X)$ and $C=\{1\}\times X$. Then for each $\alpha\in\mathcal{L}$ there is $\xi\in\omega_1\setminus X$ with $\xi\in T_\alpha$ and so $\langle 1,\xi\rangle\prec\langle 2,\alpha\rangle$, i.e. $B^\uparrow\supset\{2\}\times\mathcal{L}$. Moreover for each $\eta<\omega_1$ there is $\xi\in X$ with $\eta<\xi$ and so $\langle 0,\xi\prec\langle 1,\eta\rangle\rangle$. Thus $\{0\}\times\omega_1\subset C^\downarrow$. Hence $B^\uparrow\cup C^\downarrow=Q(\mathcal{T})$.

Assume now that $\mathcal{T}$ is a ♣-sequence and let $\langle B,C\rangle\in\mathfrak{S}(A)$. We show that $A\setminus B$ is countable and C is uncountable. If $C\subset A$ is countable then $\langle 0,\sup\{\alpha:\langle 1,\alpha\rangle\in C\}+1\rangle\notin C^\downarrow$. Assume on the contrary that e.g. $A\setminus B$ is uncountable. Then $X=\{\xi:\langle 1,\xi\rangle\notin B\}\in[\omega_1]^{\omega_1}$ and so there is $\alpha\in\mathcal{L}$ with $T_\alpha\subset X$. Let $x=\langle 2,\alpha\rangle$. Then $A\cap x^\downarrow=\{1\}\times T_\alpha$ and so $B\cap x^\downarrow=\emptyset$, i.e. $x\notin B^\uparrow$. Since $C^\downarrow$ is disjoint to $\{2\}\times\mathcal{L}$ we obtain that $x\notin B^\uparrow\cup C^\downarrow$, a contradiction. Hence the set $A\setminus B$ is countable. ∎

Example. Fix a ♣-sequence $\mathcal{T}=\langle T_\beta:\beta\in\mathcal{L}\rangle$ in L. Then, by proposition 4.3, the antichain $A=\{1\}\times\omega_1$ does not split in $Q(\mathcal{T})$. It is well-known that there is a c.c.c generic extension of L in which Martin's Axiom holds, and so axiom ♣ fails, especially $\mathcal{T}$ is not a ♣-sequence. Hence, applying proposition 4.3 again we obtain that A splits in this generic extension. Hence the statement "A splits" is not absolute.

As we have seen splitting property can be used to formulate an equivalent of axiom ♣. The next proposition shows that even the Axiom of Choice can be reformulated in a similar way.

Theorem 4.4. *(ZF) The Axiom of Choice is equivalent to the statement of Theorem 2.8.*

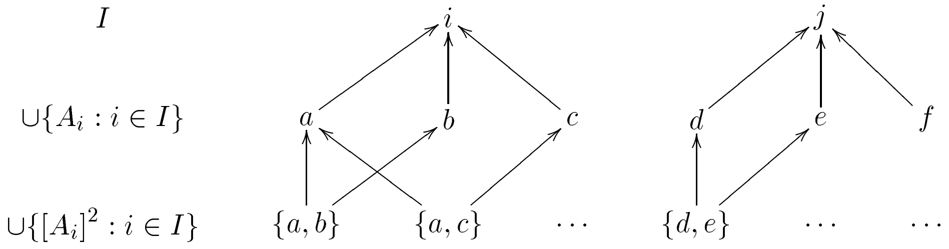
Proof. Assume that the statement of Theorem 2.8 holds. Let $\mathcal{A}=\{A_i:i\in I\}$ be a family of pairwise disjoint nonempty sets. Without loss of generality

$|A_i| \neq 1$ for each $i \in I$. We need to show that there is a choice function on $\mathcal{A}$. To do so define the poset $R(\mathcal{A}) = \langle R, \leq_R \rangle$ as follows:

$$R = I \cup \left(\cup\{A_i : i \in I\}\right) \cup \left(\cup\{[A_i]^2 : i \in I\}\right), \tag{5}$$

$$\prec = \{\langle a, i\rangle : i \in I, a \in A_i\} \cup \{\langle\{a,b\}, a\rangle : a \in A_i, b \in A_i \setminus \{a\}, i \in I\}, \tag{6}$$

and let $\leq_R$ be the partial order generated by $\prec$.



The poset $R(\mathcal{A})$ is well-founded and cut-free. The set $A = \cup\{A_i : i \in I\}$ is a maximal antichain in it and $|x^{\uparrow} \cap A| = 2$ for each $x \in A^{\downarrow} \setminus A = \cup\{[A_i]^2 : i \in I\}$. Hence A splits by theorem 2.8, $R = B^{\uparrow} \cup (A \setminus B)^{\downarrow}$ for some $B \subset A$. Since $I \subset B^{\uparrow}$ we have $B \cap A_i \neq \emptyset$ for each $i \in I$. On the other hand $|A_i \cap B| \leq 1$. Indeed $\{b,c\} \in [B]^2 \cap [A_i]^2$ would imply that $\{b,c\} \notin (A \setminus B)^{\downarrow}$. Hence $|B \cap A_i| = 1$ for each $i \in I$ and so we have a choice function f on $\mathcal{A}$: let $f(i) = \cup(A_i \cap B)$ for $i \in I$. ∎

Let us conclude this Section with a generalization of property "loose" to bigger cardinals. The proofs of the results are very similar to those in the first part of Section 3, therefore we leave them to the diligent reader.

Definition 4.5. Given a cardinal κ, a poset $\mathcal{P}$ is *κ-loose* if and only if for each $x \in P$ and $F \in [P]^{<\kappa}$ if $x \notin F^{\uparrow}$ then there is $y \in x^{\uparrow} \setminus \{x\}$ such that $y \notin F^{\downarrow} \cup F^{\uparrow}$.

Claim 4.6. *If κ and λ cardinal such that $\lambda < \kappa$ or $\lambda = \kappa = \operatorname{cf}(\kappa)$ then $\left\langle [\kappa]^{<\lambda}, \subseteq \right\rangle$ is κ-loose.*

Theorem 4.7. *Assume that $\mathcal{P} = \langle P, \leq \rangle$ is a κ-loose poset of cardinality κ. Then $\mathcal{P}$ contains a maximal antichain A such that*

(i) *if $\langle B, C \rangle \in \mathfrak{S}(A)$ then $B = A$,*
(ii) $\operatorname{cf}(|C|) = \operatorname{cf}(|A|)$ *for each $\langle B, C \rangle \in \mathfrak{S}(A)$.*
(iii) *if $\mathcal{P}$ is cut-free then A is infinite.*

Corollary 4.8. *If* $\kappa^{<\lambda} = \lambda = \kappa$ *then* $\left\langle [\kappa]^{<\lambda}, \subseteq \right\rangle$ *contains a maximal antichain which does not split. In particular,*

(i) *for each infinite cardinal* κ *the poset* $\left\langle [\kappa]^{<\omega}, \subseteq \right\rangle$ *contains maximal antichain which does not split,*

(ii) *if the continuum hypothesis holds then* $\langle [\omega_1]^{\omega}, \subseteq \rangle$ *contains maximal antichain which does not split.*

Proof. Since $\left|[\kappa]^{<\lambda}\right| = \kappa^{<\lambda} = \kappa$, and $\left\langle [\kappa]^{<\lambda}, \subseteq \right\rangle$ is cut-free and κ-loose we can apply Theorem 4.7 to get the required maximal antichain. ∎

Corollary 4.9. *If* $2^{\omega} = \omega_1$ *then the cut-free poset* $\mathcal{P}(\omega)/[\omega]^{<\omega}$ *contains an antichain which does not split.*

References

[1] R. Ahlswede, P. L. Erdős and N. Graham: A splitting property of maximal antichains, *Combinatorica* **15** (1995), 475–480.

[2] R. Ahlswede and L. H. Khachatrian: Splitting properties in partially ordered sets and set systems, in *Numbers, Information and Complexity* (Althöfer et. al. editors) Kluvier Academic Publisher, (2000), 29–44.

[3] F. Bernstein: Zur Theorie der triginomischen Reihen, *Leipz. Ber. (Berichte über die Verhandlungen der Königl. Sächsischen Gesellschaft der Wissenschaften zu Leipzig. Math.-Phys. Klasse)* **60** (1908), 325–338.

[4] D. Duffus and B. Sands: Finite distributive lattices and the splitting property, *Algebra Universalis* **49** (2003), 13–33.

[5] Mirna Džamonja: Note on splitting property in strongly dense posets of size $\aleph_0$, *Radovi Matematički* **8** (1992), 321–326.

[6] P. L. Erdős: Splitting property in infinite posets, *Discrete Mathematics* **163** (1997), 251–256.

[7] P. L. Erdős: Some generalizations of property B and the splitting property, *Annals of Combinatorics* **3** (1999), 53–59.

[8] T. Jech: *Set Theory*, Springer-Verlag, Berlin–Heilderberg–New York, 2003.

[9] J. Klimó: On the minimal coverint of infinite sets, *Discrete Applied Mathematics* **45** (1993), 161–168.

Péter L. Erdős

A. Rényi Institute of Mathematics
Hungarian Academy of Sciences
P.O. Box 127
H-1364 Budapest
Hungary
elp@renyi.hu

Lajos Soukup

A. Rényi Institute of Mathematics
Hungarian Academy of Sciences
P.O. Box 127
H-1364 Budapest
Hungary
soukup@renyi.hu