Eötvös Loránd Tudományegyetem
Természettudományi Kar

Hosszejni Darjus
Matematika BSc
Matematikus szakirány

Adaptív csoportos tesztelés

Szakdolgozat

Témavezető: Pálvölgyi Dömötör, adjunktus
Számítógéptudományi tanszék

Budapest, 2014.

Tartalomjegyzék

Köszönetnyilvánítás

Szeretném megköszönni témavezetőmnek, Pálvölgyi Dömötörnek a téma ajánlását, segítségét a konzultációk során és a dolgozat alapos átnézését. Köszönettel tartozom még a 409-es blokknak és Manz Ingridnek, amiért türelmesek voltak, és elősegítették a munkámat.

1. Bevezető

A csoportos tesztelés mint a matematika egyik ága alig 70 éve jött létre, Robert Dorfmant tartják a terület első kutatójának, aki egy, a második világháború alatt előjött problémára kereste a választ [9, 1. o.]. A feladat a katonák közül a szifiliszesek kiszűrése volt. A szifiliszre létezett egy vérteszt, aminek egyszerre több vérmintát is be lehetett adni, így katonák csoportjáról is megmondta, hogy van-e köztük beteg. A problémához többféleképp is hozzá lehet állni, két teljesen eltérő megközelítés a valószínűségszámítás, illetve a kombinatorika felőli.

A valószínűségszámításos megközelítés a gyakorlatban előforduló hibaforrásokat és hiányosságokat kezeli úgy, hogy ezek értékeire valószínűségi eloszlást definiál. Például hipergeometrikus eloszlás a beteg katonák számára, vagy a teszt helyességének ad valószínűséget. Ekkor a feladat az elvégzendő tesztek számának várható értékének a minimalizálása. Katona Gyula az elsők között volt, aki kiemelte ennek a területnek a fontosságát [18].

A kombinatorikus megközelítés ezzel szemben nem foglalkozik eloszlásokkal. Feltételezi, hogy a beteg katonák halmaza a katonák részhalmazainak egy családjában, a *mintatérben* van. A mintatér elemeit *mintapontoknak* nevezzük. Például ha tudjuk, hogy pontosan öt katona beteg, akkor a mintapontok a katonák ötelemű részhalmazai. A feladat a legrosszabb esetben a lehető legkevesebb teszt elvégzésével kideríteni, hogy melyik mintapont a megoldás. Ezzel a problémával a matematikának és más tudományterületeknek több ága is foglalkozik, többek között a bonyolultságelmélet, gráfelmélet, kommunikációs csatornák tudománya, illetve hibatűrő számításoknál is előjöhet.

Ezektől független csoportosítás, hogy adaptív vagy nem adaptív algoritmust szeretnénk-e használni. Az adaptív algoritmusok az adott teszt elvégzéséhez figyelembe veszik a korábbiak eredményét, míg a másik esetben az összes tesztelendő halmaz előre rögzített. Ebben a dolgozatban a kombinatorikus adaptív csoportos tesztelésről lesz szó.

A bevezetésben lerögzítjük azokat a jelöléseket, amikre végig szükségünk lesz, és bemutatunk néhány gyakorlati megfontolást és alkalmazást. A második fejezetben általános képet adunk arról, hogy a témakörben milyen jellegű kérdésekkel foglalkoznak, a végére két érdekes korlátot is mutatunk a problémára. Később két speciális esettel foglalkozunk: a harmadik fejezetben arra a kérdésre keressük a választ, hogy mikor nem is segít a csoportos tesztelés, a negyedik fejezetben pedig megmutatjuk, mit tudunk arról a könnyűnek tűnő esetről, amikor tudjuk, hogy pontosan két katona beteg.

1.1. A feladat megfogalmazása és jelölések

A problémának egy olyan egyszerűsített változatával foglalkozunk részletesen, amelyet *prototípus problémának* hívunk, és amelyben a katonák számát előre tudjuk, legyen ez d. Ily módon a bevezetőben beteg katonákkal leírt feladat egy átfogalmazása a következő: legyen I egy n elemű halmaz, és legyen $D \subset I$ egy d elemű részhalmaz, a *defektív* elemek halmaza, a többi elemet pedig *perfektnek* hívjuk. Ekkor a mintatér az I halmaz d elemű részhalmazaiból áll, a mintateret és a feladatot $S(d, n)$-nel jelöljük. Teszteket végezhetünk, amelyek során kiválasztunk egy $R \subset I$ részhalmazt, és megkérdezzük, hogy $R \cap D = \emptyset$ teljesül-e, más eszközünk nincs. Ha teljesül, az a negatív kimenet, ha nem, azt pozitívnak hívjuk. Ha már tudjuk egy m elemű halmazról, hogy defektív, akkor a feladatot $S(m; d, n)$-nel jelöljük. A cél, hogy a lehető legkevesebb teszt elvégzésével biztosan meghatározzuk D-t. Egy A algoritmussal a legrosszabb esetben elvégzett tesztek számát $M_A(d, n)$-nel jelöljük. Így definiálhatjuk a következőt:

$$M(d, n) = \min_A M_A(d, n),$$

ahol A az összes $S(d, n)$-t megoldó algoritmuson végigfut. $M(d, n)$-t *minimax* tesztszámnak hívjuk, ennyi tesztre van szükség $S(d, n)$ megoldásához. Az A algoritmus minimax az $S(d, n)$ problémára, ha $M(d, n) = M_A(d, n)$. Ha d értékét nem tudjuk vagy nem fontos, akkor $M(S)$-t vagy $M_A(S)$-t írunk. Egy másik feladat, ha nem pontosan d defektív van, hanem maximum d, ennek jele $S(\bar{d}, n)$.

A dolgozat során $\lfloor x \rfloor$ ($\lceil x \rceil$) jelöli az x-nél kisebb (nagyobb) legkisebb (legnagyobb) egészt, és $\log x$ a $\log_2 x$-et jelent.

1.2. Gyakorlati megfontolások

A gyakorlatban általában nem ismerjük d pontos értékét, sokszor csak egy becslés vagy egy felső korlát adott. A legtöbb, a feladatra adott algoritmus meghatározza az összes defektívet, ha legfeljebb d van, vagy d darabot talál meg, ha több van. Ez utóbbi esetről könnyen meggyőződhetünk, a tisztázatlan állapotú elemekre kell egy tesztet végrehajtani, és ha az pozitív, akkor a maradék problémát megoldani. Viszont ha tudjuk, hogy d csak egy felső korlát, akkor nem sokkal nehezebb a dolgunk, mintha pontos érték lenne, ugyanis Hwang, Song és Du [13] belátták, hogy $M(\bar{d}, n) \leq M(d, n) + 1$ (2.2.15 tétel). Ezzel a következő fejezetben foglalkozunk.

Ha egy teszt egy mintát nem tud akárhányszor használni, akkor előfordulhat, hogy korlátozva van az, hogy egy elemet hányszor lehet megvizsgálni. Ez probléma lehet, pl. ha csak egyszer lehet tesztelni mindenkit, akkor nincs jobb algoritmus az egyénenkénti tesztelésnél. Ha egy elem legfeljebb s teszten eshet át, akkor egy megoldás Li s szintű algoritmusa [9, 24. o.].

Tegyük fel, hogy van p darab feldolgozóegységünk, ekkor p diszjunkt csoportot párhuzamosan is vizsgálhatunk. Olyan esetekben, ahol az idő fontos lehet, vagy valami más okból a körök száma hangsúlyos, ott egy olyan algoritmus segíthet, ami ezt is kihasználja. De Bonis, Gasieniec és Vaccaro [6] mutattak olyan párhuzamos algoritmust az $S(\bar{d}, n)$ feladatra, ami összesen két kör alatt véget ér, és aszimptotikusan optimális.

Adott alkalmazásnál előfordulhat az is, hogy a körök száma helyett a tesztelhető csoport elemszáma felülről korlátozott [4], pl. egy kémiai anyag észleléséhez elegendően nagy koncentrációra van szükség, nem keverhetjük össze tetszőlegesen sok vízzel. Ezt ki lehet küszöbölni azzal, ha az algoritmusban a túl nagy csoportokat feldaraboljuk.

Egy gyártószalagon az elemek sorban jönnek egymás után, így lehet, hogy nincs lehetőség elmozdítani őket. Ekkor mindig csak a sorozatban egymás után lévőket lehet csoportba foglalni egy teszthez, erre egy megoldás pl. a Du és Hwang [9, 28. o.] által leírt *vonal* algoritmusosztály. Hasonlóan lehet olyat is vizsgálni, amikor körkörösen mennek az elemek, ekkor az elsőt és az utolsót is betehetjük egy csoportba, vagy bármilyen geometriai alakzatban elhelyezkedhetnek, ekkor az egymáshoz közel lévőket lehet egyszerre tesztelni.

Vannak olyan esetek is, amikor nem csak kétféle kimenete van egy tesztnek, hanem lehet $0,1,\ldots,k-1,k^+$, ahol i azt jelenti, hogy a csoportban pontosan i darab defektív elem van, k^+ pedig azt jelenti, hogy legalább k. Az adaptív fabejárás protokoll [19] is ilyen algoritmus, kihasználja azt, hogy egy több számítógép által elért hálózaton meg lehet különböztetni a zajos jelet, amit egyszerre több gép ad, illetve a tisztát, ami csak egy géphez tartozik. Ezzel a jelöléssel a prototípus feladatban a kimenetek a 0 és az 1^+.

A gyakorlatban az is előfordul, hogy a teszt bizonyos valószínűséggel téved [14; 10], pl. egy vérteszt nem biztos, hogy kiszűri a szifilisz baktériumokat. Itt kétfajta hibalehetőség is van: az egyik, amikor egy defektívről állítja, hogy perfekt (fals pozitív), illetve amikor egy perfektről állítja, hogy defektív (fals negatív).

Damaschke [5] a prototípus feladat egy általánosítását vizsgálta, amiben egy tesztre negatív a válasz, ha a tesztelt halmazban legfeljebb l defektív van, és pozitív, ha legalább u ($u \geq l$), egyéb esetben pedig nem definiált (a kettő közül bármelyiket adhatja). Ez a megközelítés alkalmazható olyan esetekben, ha a rossz elemeknek talán van valami hatása, ha elegen vannak, és biztosan van, ha egy nagyobb küszöböt is elérnek, pl. sok betegség tünetei is hasonló módon jelentkeznek.

Mindezekkel a hibalehetőségekkel és átfogalmazásokkal nem foglalkozunk a későbbi fejezetekben, csak a prototípus problémát vizsgáljuk.

2. Néhány általános megfigyelés

A gyakorlati alkalmazásokban d általában jóval kisebb, mint n. A csoportos tesztelés ezt használja ki úgy, hogy több minta együttes tesztelésével a sok perfektet egyszerre kiszűri. A tesztelt csoport mérete azonban nehéz kérdés. Egyrészt, ha nagy mintaszám együttes tesztelésének eredménye negatív, akkor azt a sok elemet ki tudjuk zárni, viszont ha pozitív, akkor meg nem nyertünk sok információt. Másrészt, ha kevés mintát veszünk egy-egy teszthez, akkor a negatív kimenet hordoz esetleg kevés adatot, míg egy pozitív eredménnyel le van szűkítve a lehetséges defektívek száma. E kettő közötti egyensúlyt keresi a legtöbb algoritmus.

2.1. Ésszerű algoritmusok

Szűkítsük le a vizsgálandó algoritmusok számát a következő észrevétellel: ha a tesztek során van olyan lépés, aminek a kimenete megjósolható korábbi eredményekből, akkor az kihagyható, ezzel csökkentettük a lépések számát, viszont információt nem vesztettünk. Pl. ha A teszteli $\{a\}$-t és $\{b\}$-t is, akkor $\{a, b\}$-t felesleges lenne. Az ilyen algoritmusok neve *ésszerű.*

Ezzel a definícióval igaz az alábbi:

$$M(d, n) = \min_{A \text{ ésszerű}} M_A(d, n),$$

hisz minden nem ésszerű algoritmusnál van jobb, ha kitörlünk egy felesleges tesztet.

2.1.1. Tétel (Információelméleti Alsó Korlát)**.** $M(S) \geq \lceil \log|S| \rceil$.

Bizonyítás. Minden teszt két diszjunkt részhalmazra bontja a mintateret, $|S|$ elemmel (mintaponttal) kezdünk, és egyet szeretnénk megtalálni, tehát a tétel a jól ismert *Információelméleti Alsó Korlátot* adja. □

2.1.2. Állítás. $M(1, n) = \lceil \log|n| \rceil$.

Bizonyítás. Az elemek számát minden teszttel meg tudjuk felezni (páratlan számnál felfele kerekítve), így kapjuk az állítást. □

2.1.3. Állítás. $S \subset S' \implies M(S) \leq M(S')$.

Bizonyítás. Minden algoritmus S'-re egyben algoritmus S-re is, így S' feladatát biztosan nem lehet gyorsabban megoldani. □

2.1.4. Következmény. *$M(d, n)$ n-ben monoton növő.*

2.2. Adaptív algoritmus bináris fa reprezentációja

A bináris fa olyan irányított gyökeres fa, amelynek a gyökerén kívül minden csúcsának egy a befoka, és minden csúcsának pontosan nulla vagy kettő kimenő éle van. Előbbiek a *levelek*, utóbbiak a *belső pontok*. Ha u csúcsból megy él v-be, akkor u a v-nek a *szülője*, és v az u-nak a *gyereke*. Ha két csúcsnak közös a szülője, akkor ők *testvérek*. Egy u csúcsból v-be vezető út csúcsok és élek olyan alternáló sorozata, amelyben bármely csúcsból az őt követő él a következő csúcsba vezet. Ennek az útnak a *hossza* a sorozatban lévő csúcsok száma v kivételével, jele az út abszolút értéke. A gyökértől a v-hez vezető utat röviden *v-hez vezető útnak* hívjuk. Azok a csúcsok, amelyekből vezet út v-be, *v ősei*. Egy v csúcs az s-edik szinten van, ha a hozzá vezető út hossza s, a fa *mélysége* pedig a legnagyobb szint.

Legyen S egy mintatér, és A egy algoritmus S-re. Ekkor A-nak megfeleltethető egy döntési fa, melyben a megengedett teszt-függvények I részhalmazaihoz rendelnek pozitív vagy negatív értéket. Ez a döntési fa egy bináris fa, és a megfeleltetés az alábbi:

– Egy u belső csúcshoz egy $t(u)$ teszt tartozik, a két kimenő éléhez pedig a teszt két lehetséges kimenete: a bal él a negatívhoz, a jobb a pozitívhoz.

– Az u belső csúcshoz tartozó tesztelőzményeket $H(u)$-val jelöljük. Ez az u-hoz vezető úton lévő csúcsokhoz és élekhez tartozó tesztek és kimenetek halmaza.

– Minden u csúcshoz a mintatér azon $S(u)$ részhalmazát is hozzárendeljük, amelyben a $H(u)$-val konzisztens mintapontok vannak.

Ekkor minden v levélre $|S(v)| \leq 1$, és mivel minden $t(u)$ diszjunkt részhalmazokra bont, ezért S minden eleme pontosan egy ilyen $S(v)$-ben fordul elő. Mivel pontosan azok az ésszerű algoritmusok, amelyeknél $t(u)$ diszjunkt nem üres halmazokra bont, ezért azoknál $|S(v)| = 1$, tehát a levelek és a mintapontok közt kölcsönösen egyértelmű megfeleltetés van.

Legyen A ésszerű, $s \in S$ és v az s-hez rendelt levél A fájában, amit szintén A-val jelölünk, továbbá jelölje $p(s)$ a v-hez vezető utat a fában. Ekkor

$$M_A(S) = \max_{s\in S}|p(s)| = \text{A mélysége.}$$

2.2.1. Lemma. *Minden ésszerű algoritmusra* $\sum_{s\in S} 2^{-|p(s)|} = 1$.

Bizonyítás. Bináris fákra $\sum_{l\in L} 2^{-s(l)} = 1$, ahol L a levelek halmaza, $s(l)$ pedig az l levél szintje. Innen triviálisan következik az állítás. □

Egy S mintatérhez tartozó A csoportos tesztelő algoritmus majorálható, ha létezik S-hez A' algoritmus úgy, hogy $\forall s \in S \;\; |p_A(s)| \geq |p_{A'}(s)|$, és $\exists s \in S \;\; |p_A(s)| > |p_{A'}(s)|$, ahol $p_X(s)$ a fent definiált $p(s)$ az X algoritmusra. Ekkor A' algoritmus A majoránsa.

2.2.2. Tétel (Du, Hwang [9]). *Egy csoportos tesztelő algoritmus pontosan akkor ésszerű, ha nem majorálható.*

Bizonyítás. Tegyük fel, hogy A nem ésszerű. Ekkor létezik olyan csúcs, aminek a kimenetét előre meg lehet jósolni. Egy ilyen csúcsot távolítsunk el, a helyére az előre megjósolt kimenetet (az egyik gyerekét) tegyük, így a részfájában minden levélre csökken a hozzá vezető út hossza, tehát kaptunk egy majoráns algoritmust.

A másik irányhoz indirekt tegyük fel, hogy A egy ésszerű majorálható algoritmus. Ekkor létezik A' majoránsa, tehát minden $s \in S$-re

$$|p_A(s)| \geq |p_{A'}(s)|$$

legalább egy szigorú egyenlőtlenséggel. Ekkor viszont

$$1 = \sum_{s \in S} 2^{-|p_A(s)|} < \sum_{s \in S} 2^{-|p_{A'}(s)|},$$

vagyis A' nem ésszerű, tehát majorálható. Így készíthetünk egy végtelen $A, A', A'', \ldots$ majorálható sorozatot, amiben minden tag az előző majoránsa, és a szumma szigorúan nő, tehát kaptunk végtelen sok különböző algoritmust. A fájában véges sok csúcs van, és a sorozat minden tagja mint fa szigorúan kisebb, mint az előző, ez pedig ellentmond a végtelen sok algoritmusnak.

Tehát a két fogalom egymást kizáró. □

Ezentúl a dolgozatban algoritmus alatt ésszerű algoritmust értünk. Az alábbi eredmények Hu, Hwang és Wangtól [11] származnak:

2.2.3. Tétel (Hu, Hwang, Wang [11]). *Ha $m \geq 2$, és $0 < d < n$, akkor*

$$M(m; d, n) \geq 1 + M(d-1, n-1).$$

Bizonyítás. 2.1.3 szerint $M(m; d, n) \geq M(2; d, n)$, ha $m \geq 2$, hisz $S(m; d, n)$ azokból az $S(d, n)$-beli elemekből áll, amelyek belemetszenek az m elemű részhalmazba, így nyilván $S(2; d, n) \subset S(m; d, n)$.

Legyen A egy algoritmus az $S(2; d, n)$ feladatra, és legyen $M_A(2; d, n) = k$. Jelölje a korábban tesztelt kételemű csoport elemeit E_1 és E_2.

A következő állítás azt mondja ki, hogy a legrosszabb esetekben mindig bele kell kérdezni a kételemű csoportba.

Állítás. *A-ban minden k-adik szintű levél útjában van olyan teszt, aminek E_1 vagy E_2 eleme.*

Bizonyítás. Indirekt tegyük fel, hogy van olyan k-adik szintű v levél A-ban, hogy $H(v)$ nem tartalmaz olyan tesztet, aminek E_1 vagy E_2 eleme. Mivel $\{E_1, E_2\}$ halmaznak van defektív eleme, és nem lehet az elemeit megkülönböztetni, ezért ebben az esetben E_1 és E_2 is defektív. Jelölje v testvérét u, ekkor u is levél. Legyen az u-hoz tartozó defektív elemek halmaza s_u, a v-hez tartozó s_v. Mivel u-nak és v-nek ugyanazok voltak a tesztjei, az előbbi érvek u-ra is igazak, tehát s_u-ban is defektív mindkét elem. Viszont a két mintapont nem egyezik meg, ezért lennie kell két elemnek, E_u és E_v, hogy E_u defektív s_u-ban, s_v-ben pedig nem, és ugyanez E_v-vel fordítva.

Legyen u és v szülője w. Ekkor $H(w)$-n nem lehet olyan teszt, ami tetszőleges, s_u-tól diszjunkt G-re $G \cup \{E_v\}$ alakú, hisz egy ilyen teszt s_u-ra negatív lenne, s_v-re pedig pozitív, így u és v már w előtt elváltak volna. Legyen s egy olyan mintapont, ami megegyezik s_u-val azzal a kivétellel, hogy s-ben E_2 helyett E_v van. Úgy lehet csak s_u-t és s-t megkülönböztetni w előtt, ha van $H(w)$-n olyan teszt, aminek E_2 eleme, vagy $G \cup \{E_v\}$ alakú. Mind a kettőt kizártuk, tehát w alatt kell lennie az s-hez tartozó levélnek, és s nem egyezik meg sem s_u-val, sem s_v-vel, tehát u vagy v nem levél, ami ellentmondás. □

Mivel A ésszerű, ezért $\{E_1, E_2\}$-t tartalmazó teszt nincs már. Szimmetriai okokból feltehetjük, hogy minden k hosszú úton E_1-et tartalmazó teszt is van, mert különben az adott úton E_2 első előfordulásától kezdve az alatta lévő részfában mindenhol felcserélhetjük E_2-t és E_1-et.

Vegyük az $S(d-1, n-1)$ problémát kiegészítve egy képzeletbeli defektív elemmel, amit E_1-nek nevezünk el, és ezt az új feladatot jelölje S'. Legyen S' alaphalmaza I', $S(2; d, n)$ alaphalmaza pedig I. Ekkor $I' \setminus \{E_1\}$-et rendeljük kölcsönösen egyértelműen $I \setminus \{E_1\}$-hez, a kételemű defektív halmaz másik elemének az ősképe nem fontos, hisz E_1 miatt $\{E_1, E_2\}$ mindenféleképp defektív lesz. Ekkor A megoldja S'-t is, de mivel ebben az esetben az E_1-et tartalmazó tesztek mind előre jósolhatók, ezért azokat kihagyjuk. Minden leghosszabb úton van E_1-et tartalmazó kérdés, így A eggyel kevesebb lépésből, $k-1$ lépés alatt is megoldja S'-t.

Legyen A most egy minimax algoritmus $S(2; d, n)$-re, így

$$M(2; d, n) = M_A(2; d, n) \geq 1 + M_A(d-1, n-1) \geq 1 + M(d-1, n-1).$$

Ezzel a 2.2.3 tétel bizonyítását befejeztük. □

2.2.4. Következmény. *$M(2; d, n) = 1 + M(d-1, n-1)$, ha $0 < d < n$.*

Bizonyítás. Legyen A egy olyan algoritmus az $S(2; d, n)$ problémára, ami először tesztel egy elemet a kételemű defektes halmazból, aztán egy minimax algoritmust használ a maradék feladatra. Ekkor

$$\begin{aligned} M_A(2; d, n) &= 1 + \max\{M(d-1, n-2), M(d-1, n-1)\} \\ &= 1 + M(d-1, n-1) \qquad \text{2.1.3 szerint.} \end{aligned}$$

Ezzel, és az előző tétellel az állítást beláttuk. □

2.2.5. Lemma. $M(d,n) \leq n-1$.

Bizonyítás. Egyenként tesztelve $n-1$ elemet és d-t ismerve az utolsó elem állapotát ki tudjuk következtetni, ezzel megadtunk egy $n-1$ lépéses algoritmust. $\square$

2.2.6. Következmény. $M(n-1,n) = n-1$

Bizonyítás. $n = 1$-re triviális. Indukcióval $n \geq 2$-re:

$$M(n-1,n) = M(n; n-1,n) \geq 1 + M(n-2,n-1) = n-1.$$

$\square$

2.2.7. Tétel (Hu, Hwang, Wang [11])**.** *Ha* $0 < d < n$*, akkor*

$$M(d,n) \geq 1 + M(d-1,n-1) \geq M(d-1,n).$$

Bizonyítás. A 2.2.3 tételt alkalmazva $M(d,n) = M(n;d,n) \geq 1 + M(d-1,n-1)$.

A második egyenlőtlenséget d-re való indukcióval látjuk be. $d = 1$-re triviálisan igaz. Az indukciós lépésben $(d+1)$-re az $M(d,n) \geq M(d-1,n)$ egyenlőtlenséget használjuk. Legyen A egy olyan algoritmus, amely először letesztel egy elemet, majd egy minimax algoritmust használ a maradék problémára.

Ekkor

$$\begin{aligned} M(d-1,n) &\leq M_A(d-1,n) \\ &= 1 + \max\{M(d-1,n-1), M(d-2,n-1)\} \\ &= 1 + M(d-1,n-1). \end{aligned}$$

$\square$

2.2.8. Lemma (Hu, Hwang, Wang [11])**.** *Ha* $0 < d < n-1$ *és* $M(d,n) = n-1$*, akkor*

$$M(d,n-1) = n-2.$$

Bizonyítás. Indirekt tegyük fel, hogy $M(d,n-1) \neq n-2$, azaz $M(d,n-1) < n-2$ a 2.2.5 lemma szerint. Legyen A egy olyan algoritmus $S(d,n)$-re, ami először tesztel egy elemet, majd a maradék problémát egy minimax algoritmussal oldja meg.

Ekkor

$$\begin{aligned} M(d,n) &\leq M_A(d,n) \\ &= 1 + \max\{M(d,n-1), M(d-1,n-1)\} \\ &= 1 + M(d,n-1) && \text{2.2.7 szerint} \\ &< n-1, \end{aligned}$$

ami ellentmond a feltevésnek. □

2.2.9. Tétel (Hu, Hwang, Wang [11]). *Tegyük fel, hogy* $0 < d < n$*, ekkor*

$$M(d,n) = M(d-1,n) \implies M(d,n) = n-1$$

.

Bizonyítás. A tételt $(n-d)$-re vonatkozó indukcióval bizonyítjuk. $n-d = 1$-re igaz 2.2.6 szerint.

Az általános esetben először vegyük észre, hogy

$$M(d,n) = 1 + M(d-1,n-1) \tag{1}$$

is teljesül 2.2.7 szerint. Legyen A egy minimax algoritmus az $S(d,n)$ problémára. Azt állítjuk, hogy A először egy egyelemű halmazra kérdez. Indirekt legyen az A által először tesztelt halmaz mérete $m > 1$.

Ekkor

$$\begin{aligned} M_A(d,n) &= 1 + \max\{M(m;d,n), M(d,n-m)\} \\ &\geq 1 + M(m;d,n) \\ &\geq 2 + M(d-1,n-1) && \text{2.2.3 szerint,} \end{aligned}$$

ami ellentmond a feltevésnek. Tehát A első tesztje egyelemű, és

$$\begin{aligned} M(d,n) &= 1 + \max\{M(d,n-1), M(d-1,n-1)\} \\ &= 1 + M(d,n-1) && \text{a 2.2.7 tétel szerint.} \end{aligned} \tag{2}$$

(1) és (2) alapján $M(d-1,n-1) = M(d,n-1)$, amiből indukcióval $M(d,n-1) = n-2$. Végül (2) egyenlettel

$$M(d,n) = n-1.$$

□

2.2.10. Lemma (Hu, Hwang, Wang [11]). *Tegyük fel, hogy* $M(d,n) < n-1$*. Ekkor minden* $0 < l \leq d < n$*-re*

$$M(d,n) \geq 2l + M(d-l,n-l).$$

Bizonyítás. 2.2.6 szerint $d \neq n-1$, és mivel $d < n$, így $d < n-1$ is teljesül. Az állítást l-re vonatkozó indukcióval bizonyítjuk be.

$l = 1$ esetén legyen A egy minimax algoritmus $S(d,n)$-re. Ha A először egy $m > 1$ elemű halmazt tesztel, akkor az állítás következik 2.2.3 tételből, tehát tegyük fel, hogy A először egyelemű halmazra kérdez. Indirekt módon tegyük fel, hogy az állítás nem teljesül, ekkor

$$\begin{aligned}
1 + M(d-1, n-1) &\geq M_A(d,n) \\
&= 1 + \max\{M(d, n-1), M(d-1, n-1)\} \\
&= 1 + M(d, n-1) && \text{2.2.7 szerint.} \qquad (3)
\end{aligned}$$

Ismét a 2.2.7 tételt használva (3) egyenlőtlenségben egyenlőség teljesül, amiből a 2.2.9 tétel szerint $M(d, n-1) = n-2$, vagyis

$$M(d,n) = M_A(d,n) = 1 + M(d, n-1) = n-1,$$

ami ellentmond a lemma feltételének. Tehát a lemma igaz $l = 1$-re.

Az általános esetben nézzük $l+1$-et, $l \leq d-1$. Most is legyen A minimax, ami először egy m elemű halmazt tesztel. Ha $m \geq 2$, akkor

$$M(d,n) = 1 + M(m; d, n) \geq 2 + M(d-1, n-1) \geq 2 + 2l + M(d-1-l, n-1-l)$$

az indukciós feltevést használva. Ha $m = 1$, akkor az $l = 1$ esethez hasonlóan indirekt módon bizonyítjuk az állítást. Ekkor

$$\begin{aligned}
2l + 1 + M(d-l-1, n-l-1) &\geq M(d,n) \\
&= M_A(d,n) \\
&= 1 + M(d, n-1) && \text{2.2.7 szerint} \\
&\geq 1 + 2l + M(d-l, n-1-l)
\end{aligned}$$

2.2.7 tételt használva az egyenlőtlenségláncban minden tag egyenlő, amiből 2.2.9 szerint $M(d-l, n-l-1) = n-l-2$, így $M(d,n) = 2l+1+n-l-2 = n+l-1 \geq n$, ami ellentmond a feltételnek. Ezzel a bizonyítás végére értünk. □

2.2.11. Következmény (Hu, Hwang, Wang [11]). *Legyen* $0 < d < n$, *ekkor*

$$M(d,n) \geq \min\left\{ n-1, 2l + \left\lceil \log \binom{n-l}{d-l} \right\rceil \right\} \text{ minden } 0 < l \leq d\text{-re.}$$

Ebből az alábbi állítás triviálisan következik.

2.2.12. Következmény (Hwang [12]). *Ha* $0 < d < n \leq 2d$, *akkor*

$$M(d,n) = n-1.$$

A 3.1. fejezetben foglalkozunk hasonló állításokkal. A fejezet maradék részében az $S(\bar{d}, n)$ feladatra adunk alsó és felső korlátot, és ezzel egy időben újabb egyenlőtlenséget kapunk az $S(d,n)$ problémára is. A 2.2.14 tétel Hwang, Song és Du eredménye, de annak belátásához szükség van először egy lemmára.

Legyen A egy algoritmus az $S(d,n)$ feladatra, legyen v egy levél A-ban, és $S(v) = \{s\}$. Particionáljuk s halmazt: nevezzük $x \in s$ elemet *fixnek*, ha létezik olyan teszt $H(v)$-ben, amiben s-nek más eleme nincsen benne. A többi elemet *szabadnak* nevezzük.

2.2.13. Lemma. *Legyen v egy levél A-ban $f \geq 1$ szabad elemmel, és jelölje v testvérét u. Ekkor u egy legalább f szintű fa gyökere.*

Bizonyítás. Mivel v levél, jelölje $S(v)$-t $\{s\}$, továbbá legyen v szülője w, a w-nél tesztelt halmaz pedig X. Vegyük észre, hogy a szabad elemeket $n-d$ perfekt azonosításával találja meg A, emiatt a levelekben a szabad elemeket csak az utolsó teszt után ismerjük meg. Emiatt X negatív kimenetéhez tartozik v, és $X \cap s = \emptyset$.

Legyen $x \in X$, y egy szabad elem s-ből és $Y = (s \setminus \{y\}) \cup \{x\}$. Azt állítjuk, hogy $Y \in S(u)$. Ehhez egyrészt vegyük észre, hogy $H(v)$-n bármely teszt, ami y-t tartalmazza, tartalmaz más s-beli elemet is, hisz y szabad elem, emiatt y perfektre állítása $H(v)$ egyik tesztjének kimenetén sem változtatna, így $Y \in S(w)$. Itt megjegyezzük, hogy következésképp s fix elemei már defektívek $S(w)$ összes elemében. Másrészt Y konzisztens X pozitív kimenetével, így $Y \in S(u)$.

Jelölje Y_x az $\{(s \setminus \{y\}) \cup \{x\} \mid y \in s,\, y \text{ szabad}\}$ családot és a hozzátartozó csoportos tesztelési feladatot. Ekkor $Y_x = (f-1, f)$, hiszen s fix elemeiről és x-ről tudjuk, hogy defektívek. Emiatt 2.1.3 és 2.2.6 szerint $M(S(u)) \geq M(Y_x) = |Y_x| - 1 = f - 1$, ezzel az állítást beláttuk. □

2.2.14. Tétel (Hwang, Song, Du [13])**.** *Legyen A egy algoritmus az $S(d,n)$ problémához, $0 < d < n$, ekkor létezik olyan A' algoritmus $S(\bar{d},n)$-hez, hogy*

$$M_A(d, n+1) \geq M_{A'}(\bar{d}, n).$$

Bizonyítás. Legyen A' az az algoritmus, amit A-ból kapunk úgy, hogy minden olyan v levelet, aminek van szabad eleme helyettesítünk egy A_v fával, ahol A_v a szabad elemeket teszteli egyesével, így $f+1$ szintű. A' egy algoritmus $S(\bar{d},n)$-re, hisz egy adott levélnél a fix elemek már biztosan defektívek. 2.2.13 lemmából v testvére A-ban (és A'-ben is) egy legalább f szintű fa gyökere, így az A'-re kapott fa legfeljebb eggyel magasabb A-nál. □

2.2.15. Következmény (Du, Hwang [9])**.** *Legyen $0 < d < n$. Ekkor*

$$M(d,n) + 1 \geq M(\bar{d}, n) \geq M(d, n+1).$$

Bizonyítás. Az előző tételből közvetlenül következik az első egyenlőtlenség. A második egyenlőtlenséghez vegyük észre, hogy egy elemet félretéve $S(d,n+1)$-ből olyan problémát kapunk, amit $S(\bar{d},n)$ minden algoritmusa meg tud oldani, hisz a félrerakott elem állapotát ki tudjuk következtetni a többi n eleméből. □

3. Mikor minimax az egyénenkénti tesztelés?

3.1. Eddigi eredmények

Az $S(d, n)$ feladatban, ahol ismerjük a defektívek számát, az egyénenkénti tesztelés $n-1$ tesztet jelent, az utolsó elem állapotát a többiből ki lehet következtetni. A problémával régóta foglalkoznak, Hwang [12] 1971-ben belátta, hogy ha legalább az elemek fele defektív, akkor már nem éri meg csoportosan tesztelni őket. Később Hu, Hwang és Wang [11] megjavították ezt $d < n < \lfloor 2{,}5d + 0{,}5 \rfloor$-re, amit Du és Hwang [8] múlt felül, ők megmutatták, hogy

$$M(d, n) = n - 1, \text{ ha } 0 < d < n \leq \lfloor 2{,}625d \rfloor,$$

majd ezt Leu, Lin és Weng [16] megjavították, az ő eredményük

$$M(d, n) = n - 1, \text{ ha } 193 \leq d < n \leq \lfloor 2{,}6875d \rfloor.$$

Az eddigi legjobb eredmény Riccio és Colbourn nevéhez fűződik:

3.1.1. Tétel (Riccio, Colbourn [17]). *Legyen* $\alpha < \log_{\frac{3}{2}} 3 \approx 2{,}7095$. *Ekkor* $M(d, n) = n - 1$ *elég nagy* d *és* $n \leq \alpha d$ *esetén.*

Hu, Hwang és Wang [11] azt sejtették, hogy a végső válasz $d < n < 3d$ lesz. Vannak, akik a témakörben ezt tartják a legnagyobb nyitott kérdésnek, lásd Leu [15]. Ha igaz a sejtés, akkor a korlát éles, ugyanis

3.1.2. Tétel (Hu, Hwang, Wang [11]). $n > 3d > 0$ *esetén*

$$M(d, n) < n - 1.$$

Bizonyítás. Mivel $M(d, n)$ monoton növő n-ben, elég belátni, hogy $M(d, 3d + 1) < 3d$ minden d-re, és az alább bevezetett A algoritmus ezt bizonyítja is.

Rendezzük sorba az elemeket. A álljon blokkokból, egy blokk egy vagy két teszt. Az aktuális blokk elején legyen d' elemről ismert, hogy defektív, p'-ről ismert, hogy perfekt, és n'-ről nincs információnk, ezek jelölik a nekik megfelelő halmazokat is. Kezdetben tehát $d' = p' = 0$, és $n' = 3d + 1$. A minden blokkban meghatároz vagy egy defektívet, vagy egy defektívet és egy perfektet, vagy két perfektet. Egy blokkban A először tesztelje n' utolsó két elemét, és ha ez negatív, akkor menjen a következő blokkra. Ellenkező esetben n' utolsó elemét ismét teszteli, ezzel már egy defektívet biztosan talál, azonban ha a teszt negatív, akkor egy perfektet is talált ingyen, ezzel vége a blokknak. Addig ismétli, amíg $d' < d$ és $p' < 2d + 1$, ugyanis ha valamelyiknél egyenlőség van, akkor már ki tudjuk következtetni a többi elem állapotát.

Vegyük észre, hogy ez az algoritmus legfeljebb $3d$ teszt után véget ér, ugyanis legrosszabb esetben $2d$ teszt kell d defektívhez, és d teszt kell $2d$ elemhez. $3d$ teszt pedig csak úgy fordulhat elő, ha az utolsó blokkra egy defektív és egy perfekt elem marad, ekkor viszont felesleges a blokk első tesztjével a két elemre rákérdezni. Módosítsuk A-t úgy, hogy ebben az esetben hagyja ki azt a tesztet, ezzel az állítást beláttuk. □

A fejezet hátralévő részében megmutatunk három feltételt, amik meglepő módon egyenként ekvivalensek a Hu-Hwang-Wang sejtéssel.

3.2. A Hu-Hwang-Wang sejtésről

Legyen I egy $m+n$ elemű halmaz, és legyenek $G, H \in I$ diszjunktak, $|G| = n_1$, $|H| = n_2$. Legyen továbbá G-ben pontosan d_1, H-ban pontosan d_2 darab defektív elem. Az ilyen feladatok jele $(d_1, n_1) \times (d_2, n_2)$.

Ezután már ki tudjuk mondani a sejtéseket. Az utolsó három sejtés Leutól [15] származik, ő is mutatta meg, hogy ezek ekvivalensek.

3.2.1. Sejtés (Hu, Hwang, Wang [11])**.** *Ha $0 < d < n \leq 3d$, akkor*

$$M(d,n) = n - 1.$$

3.2.2. Sejtés (Leu [15])**.** *Ha $3(d_1 + d_2) \geq n_1 + n_2 > d_1 + d_2 > 0$, $n_1 > d_1 \geq 0$ és $n_2 > d_2 \geq 0$, akkor*

$$M(d_1, n_1) + M(d_2, n_2) < M(d_1 + d_1, n_1 + n_2).$$

3.2.3. Sejtés (Leu [15])**.** *Ha $d > 0$, akkor*

$$M(d, 3d+2) = 3d.$$

3.2.4. Sejtés (Leu [15])**.** *Ha $d > 1$, akkor*

$$M((1,3) \times (d-1, 3d-1)) < M(3; d, 3d+2).$$

3.2.5. Tétel (Leu [15])**.** *3.2.1 és 3.2.2 ekvivalensek.*

Bizonyítás. Először 3.2.1 elégségességét látjuk be. $n = n_1 + n_2$, $d = d_1 + d_2$ jelölésekkel, ahol $n_1 > d_1 \geq 0$, $n_2 > d_2 \geq 0$ és $3d \geq n > d > 0$. 2.2.5 lemma szerint

$$M(d_1, n_1) + M(d_2, n_2) \leq n_1 - 1 + n_2 - 1 = n - 2 < n - 1 = M(d, n).$$

A szükségességhez 2.1.4 szerint elég megmutatni, hogy $M(d, 3d) = 3d - 1$. Ez könnyen látszik $d = 1$-re, nagyobb d-re pedig 3.2.2 sejtésből

$$M(d, 3d) > M(1,3) + M(d-1, 3d-3) = 2 + 3d - 4 = 3d - 2,$$

így 2.2.5 lemmával az állítást beláttuk. □

A következő cél, hogy belássuk, hogy az első sejtés ekvivalens a harmadikkal, ahhoz kell a 3.2.7 tétel, amihez pedig szükség van az alábbi lemmára.

3.2.6. Lemma. *Ha* $M(d,n) = n-1$ *és* $n-2 \leq M(d-1,n) \leq n-1$*, akkor* $M(d,n+1) = n$.

Bizonyítás. Legyen m az első teszt mérete egy minimax algoritmusban az $S(d, n+1)$ problémára, ekkor

$$M(d,n+1) = 1 + \max_m \{M(m;d,n+1), M(d,n+1-m)\}.$$

$m = 1$-re $M(d,n+1) = 1 + M(1;d,n+1) = 1 + M(d-1,n) = n$.

Legyen $m \geq 2$. Ekkor a 2.2.5 lemma segítségével $1+M(d,n+1-m) \leq 1+n-2 = n-1$. Továbbá a 2.2.3-at felhasználva $1 + M(m;d,n+1) \geq 1+1+M(d-1,n) \geq n$. Ezzel az állítást beláttuk. □

3.2.7. Tétel (Leu [15]). *Ha* $M(d,3d) = 3d-1$*, akkor*

$$M(d,3d+2) = 3d \iff M(d+1,3(d+1)) = 3(d+1)-1.$$

Bizonyítás. Először az elégségességet bizonyítjuk. A 2.2.7 lemma következményeke, hogy $3d-1 = M(d,3d) \leq M(d+1,3d)$, amiből 2.2.5 felhasználásával $M(d+1,3d) = 3d-1$, így a 3.2.6 lemmával $M(d+1,3d+1) = 3d$. Könnyen látható 2.1.3 és 3.2.7 segítségével, hogy $M(d,3d+1) = 3d-1$. A 3.2.6 lemmát $n = 3d+1$-re alkalmazva $M(d+1,3d+2) = 3d+1$-et, majd $n = 3d+2$-re alkalmazva pedig $M(d+1,3(d+1)) = 3d+2$-t kapunk.

A szükségesség indirekt módon bizonyítjuk. Tegyük fel, hogy $M(d,3d+2) \neq 3d$, ekkor 2.1.3 és 3.2.7 segítségével $M(d,3d+2) = 3d+1$. Az $S(d+1,3d+3)$ feladatra legyen A egy olyan algoritmus, ami először egy kételemű K halmazt tesztel, pozitív kimenet esetén rákérdez K egyik elemére, végül a maradék problémát minimax számú teszttel oldja meg. Erre az algoritmusra

$$M_A(d+1,3d+3) = 1 + \max\{M(d+1,3d+1), 1+M(d,3d+1), 1+M(d,3d+2)\}.$$

2.1.3, 2.2.8 és 3.1.2-nek köszönhetően tudjuk, hogy a maximum zárójelében lévő értékek egyenlőek és az értékük $3d$, tehát $M_A(d+1,3d+3) = 3d+1$. Ebből

$$3d+2 = M(d+1,3d+3) \leq M_A(d+1,3d+3) = 3d+1,$$

ami ellentmondás. □

3.2.8. Tétel (Leu [15]). *A 3.2.1 sejtés és a 3.2.3 sejtés ekvivalensek.*

A tétel mindkét iránya könnyen belátható indukcióval, ha felhasználjuk az előző tételt és azt, hogy $M(1,3) = 2$. A fejezet maradék részében megmutatjuk, hogy 3.2.3 ekvivalens 3.2.4-gyel, ehhez azonban szükségünk lesz néhány lemmára és a 3.2.11 tételre.

3.2.9. Lemma. *$M((1{,}2)\times(d,n)) = 1 + M(d,n)$.*

Bizonyítás. Vegyük észre, hogy $M((1{,}2)\times(d,n)) \leq M(1{,}2)+M(d,n) = 1+M(d,n)$, tehát elég megmutatni, hogy $M(d,n) < M((1{,}2)\times(d,n))$.

Legyen az $(1{,}2)\times(d,n)$ két halmaza $X = \{x_1, x_2\}$ és Y. Legyen A egy minimax algoritmus $S((1{,}2)\times(d,n))$-re, és v egy a legalsó szinten lévő levél A-ban. Jelöljük v szülőjét w-vel. $H(w)$-t vizsgálva két esetet különböztetünk meg.

1. eset: $H(w)$-ben minden teszt diszjunkt X-től. Ekkor $S(w)$-ben már eldőlt, hogy Y-ból kik a defektívek, ugyanis az utolsó teszttel tudhatunk csak meg X-ről információt.

2. eset: $H(w)$-ben van olyan teszt, ami belemetsz X-be, legyen $H(w)$-ben az első ilyen teszt az u csúcsnál. Jelöljük S_u-val az $S(d,n)$ mintatér azon elemeit, amelyek konzisztensek $H(u)$-val. Ekkor $S(u) = \{H \cup \{x_i\} | H \in S_u, i \in \{1{,}2\}\}$. Valamely $G \in Y$-ra $t(u)$ teszt $G \cup \{x_i\}$ alakú, szimmetriai okokból feltehetjük, hogy $i = 1$. Két esetet vizsgálunk.

2.1. eset: $G \neq \emptyset$. A $t(u)$ teszt után $S(u)$ két diszjunkt nem üres halmazra bomlik, pozitív kimenetnél

$$S_p = \{H \cup \{x_1\} | H \in S_u\} \cup \{H \cup \{x_2\} | H \in S_u, H \cap G \neq \emptyset\},$$

negatív kimenetnél

$$S_n = \{H \cup \{x_1\} | H \in S_u, H \cap G = \emptyset\}.$$

Jegyezzük meg, hogy $\{H \cup \{x_1\} | H \in S_u\}$ esetén a mintatér már csak S_u.

2.2. eset: $G = \emptyset$. Ekkor

$$S_p = \{H \cup \{x_1\} | H \in S_u\},$$

és

$$S_n = \{I \cup \{x_2\} | H \in S_u\}.$$

Vegyük észre, hogy ha X nem létezne, akkor u-nál a mintatér $S(u)$ helyett S_u lenne, ezért $t(u)$ elhagyható lenne.

Így akár az 1., akár a 2. esetet nézzük, X kidobásával olyan algoritmust kapunk, ami legfeljebb eggyel kevesebb tesztet hajt végre, mint A. Ezzel beláttuk az állítást. □

Ez után a tétel után az alábbi állítás már egyenesen következik a 2.1.3 lemmából.

3.2.10. Lemma. *$M(d,n) \leq 3d - 1 + \lceil \frac{n-3d-1}{2} \rceil$, ha $n \geq 3d + 1 \geq 4$.*

Bizonyítás. Az egyenlőtlenséget indukcióval bizonyítjuk. Ha $d = 1$, következik az állítás minden n-re az $M(1, n) = \lceil \log n \rceil$ egyenletből, nagyobb d-re pedig az $n = 3d + 1$ és az $n = 3d + 2$ esetek vezethetők le könnyen a 3.1.2 tételből. A továbbiakhoz feltesszük $n - 1 \leq 3d + 2$-re, hogy

$$M(d, n-1) \leq 3d - 1 + \left\lceil \frac{n - 1 - 3d - 1}{2} \right\rceil.$$

Legyen A egy olyan algoritmus az $S(d, n)$ feladatra, amely először egy kételemű halmazt tesztel, és utána a maradékra egy minimax algoritmust használ. Ekkor

$$\begin{aligned} M(d, n) &\leq M_A(d, n) \\ &= 1 + \max\{M(d, n-2), M(2; d, n)\} \\ &= 1 + \max\{M(d, n-2), 1 + M(d-1, n-1)\} && \text{2.2.4 szerint} \\ &\leq 3d - 1 + \left\lceil \frac{n - 3d - 1}{2} \right\rceil && \text{az indukció szerint.} \end{aligned}$$

□

3.2.11. Tétel (Leu [15])**.** *$M((1,3) \times (d-1, 3d-1)) \leq 3d - 2$, ha $d \geq 2$.*

Bizonyítás. Legyen $X = \{x_1, x_2, x_3\}$ és $Y = \{y_1, \ldots, y_{3d-1}\}$ a két diszjunkt halmaz. A 4.2.2 lemmából tudjuk, hogy $M((1,3) \times (1,5)) = 4$, tehát a tétel $d = 2$-re igaz. Nagyobb d esetén az alábbi algoritmus bizonyítja az állítást.

1. lépés: $i := 1$.

2. lépés: Teszteljük az $\{x_1, y_i\}$ halmazt. Ha pozitív, akkor menjünk a 3. lépésre. Ellenkező esetben a maradék problémát, ami az $(1,2) \times (d - i, 3d - i - 1)$, oldjuk meg egy minimax algoritmussal. A 3.2.9 és a 3.2.10 lemmákból következik, hogy

$$\begin{aligned} M((1,2) \times (d - i, 3d - i - 1) &= 1 + M(d - i, 3d - i - 1) \\ &\leq 1 + 3d - 2i - 2 \\ &= 3d - 2i - 1. \end{aligned}$$

Ebben az esetben maximum $2(i - 1) + 1 + 3d - 2i - 1 = 3d - 2$ tesztre van szükség.

3. lépés: Teszteljük az $\{x_2, x_3, y_i\}$ halmazt. Ha pozitív, akkor tudjuk, hogy y_i defektív. Ha $i < d - 1$, akkor eggyel növeljük i-t, majd a 2. lépéssel folytatjuk. Ha $i = d - 1$, akkor Y minden defektív elemét megtaláltuk $2d - 2$ teszttel, és az $(1, 3)$ probléma maradt hátra, amihez két teszt elég.

Ha a teszt negatív lett, akkor egy minimax algoritmussal oldjuk meg a megmaradt $(d-i, 3d-i-1)$ feladatot. Vegyük észre, hogy ennél a lépésnél az azonosított defektívek az $\{x_1, y_1, y_2, \ldots, y_{i-1}\}$, a perfektek pedig az $\{x_2, x_3, y_i\}$ elemek. Eddig $2i$ teszten vagyunk túl, hátravan $M(d-i, 3d-i-1)$, így a 3.2.10 lemma szerint összesen maximum $3d-2$ kell.

Látható, hogy az algoritmus legfeljebb $3d-2$ lépésben véget ér. $\square$

3.2.12. Következmény. *Ha* $d \geq 2$*, és* $M(d-1, 3d-1) = 3d-3$*, akkor*

$$M((1,3) \times (d-1, 3d-1)) = 1 + M(d-1, 3d-1).$$

Bizonyítás. A 2.1.3 állításból következik, hogy $M((1,a) \times (d,n)) \leq M((1,b) \times (d,n))$, ha $1 \leq a \leq b$ és $0 < d < n$. Ezt és az előző tételt felhasználva kapjuk, hogy $d \geq 2$-re

$$1 + M(d-1, 3d-1) = M((1{,}2) \times (d-1, 3d-1)) \leq M((1,3) \times (d-1, 3d-1)) \leq 3d-2.$$

$\square$

3.2.13. Következmény (Leu [15])**.** *Ha* $d \geq 2$*, és minden* $0 < d' \leq d$*-re* $M(d', 3d') = 3d' - 1$*, akkor*

$$M(d, 3d+2) = 3d \iff M((1,3) \times (d-1, 3d-1)) < M(3; d, 3d+2).$$

Bizonyítás. Először az elégségességet bizonyítjuk.

$$\begin{aligned} 3d &= M(d, 3d+2) \\ &\leq 1 + \max\{M(3; d, 3d+2), M(d, 3d-1)\} && \text{először három elemet tesztelve} \\ &= 1 + M(3; d, 3d+2) && \text{a feltétel és 2.2.8 szerint.} \end{aligned}$$

Majd

$$\begin{aligned} M(3; d, 3d+2) &> 3d-2 \\ &= 1 + M(d-1, 3d-1) && \text{3.2.7 szerint} \\ &= M((1,3) \times (d-1, 3d-1)) && \text{3.2.12 szerint.} \end{aligned}$$

A szükségesség bizonyításánál feltesszük, hogy $M(3; d, 3d+2) > 3d-2$. A 2.1.3 és a 3.1.2 állításokból tudjuk, hogy $3d-1 = M(d, 3d) \leq M(d, 3d+2) \leq 3d$, tehát elég belátni, hogy $M(d, 3d+2) \geq 3d$. Vegyünk egy minimax algoritmust, ami először m elemet tesztel, ekkor $M(d,n) \geq 1 + \max_m\{M(m; d, n), M(d, n-m)\}$. Ha $m = 1$ vagy $m = 2$, akkor

$$\begin{aligned}
M(d,3d+2) &\geq 1 + M(d,3d+2-m) \\
&\geq 1 + M(d,3d) && \text{2.1.4 szerint} \\
&= 3d && \text{a feltétel szerint.}
\end{aligned}$$

Ha $m \geq 3$, akkor

$$\begin{aligned}
M(d,3d+2) &\geq 1 + M(m;d,3d+2) \\
&\geq 1 + M(3;d,3d+2) && \text{2.1.3 szerint} \\
&\geq 1 + 3d - 1 = 3d.
\end{aligned}$$

Ezzel az állítást beláttuk. □

Most már következhet a sejtések ekvivalenciája.

3.2.14. Tétel (Leu [15])**.** *3.2.3 és 3.2.4 ekvivalensek.*

Bizonyítás. Ha feltesszük, hogy $M(d, 3d+2) = 3d$, akkor a 3.2.8 tételből $d < n \leq 3d$-re $M(d,n) = n-1$. Alkalmazhatjuk 3.2.13-at, amiből 3.2.4 azonnal következik.

Most tegyük fel, hogy $M((1,3)\times(d-1,3d-1)) < M(3;d,3d+2)$. Indukcióval bizonyítjuk ezt az irányt. Igaz a tétel $d=1$-re, hiszen $M(1,5) = \lceil \log 5 \rceil = 3$. Az általános esetben feltesszük, hogy $M(d,3d+2) = 3d$ minden $0 < d \leq k$-ra. A 3.2.7 tétel ismételt alkalmazásával kapjuk, hogy $M(d,3d) = 3d-1$ minden $0 < d \leq k+1$-re. Megint alkalmazhatjuk 3.2.13-at, amiből

$$M(k+1, 3(k+1)+2) = 3(k+1).$$

Ezzel az állítást beláttuk. □

4. A kétdefektes eset

Annak ellenére, hogy az $M(1,n) = \lceil \log n \rceil$ könnyű eredmény, a $d = 2$ eset meglepően máig megoldatlan. Legyen $n_t(d)$ a legnagyobb olyan n, amire $M(d,n) \leq t$. A 2.1.4 állítás értelmében $M(d,n)$ monoton növő n-ben, ezért az $n_t(d)$ értékek ismeretében $M(d,n)$-t is ismernénk. Ez a fejezet $n_t(2)$-vel foglalkozik.

4.1. Felső korlát

Jelölje $n_t(2)$-t az egyszerűség kedvéért n_t. Legyen $t \geq 1$-re i_t az az egész, amire

$$\binom{i_t}{2} < 2^t < \binom{i_t+1}{2}.$$

Nem létezik olyan i egész és $t \geq 1$, amire $\binom{i}{2} = 2^t$, tehát i_t jól definiált. Jegyezzük meg, hogy 2.1.1 és 2.1.4 szerint $n_t \leq i_t < 2^t = n_t(1)$. A következő lemmák i_t-ről Chang, Hwang és Lin eredményéhez szükségesek, miszerint $t \geq 4$ esetén már $i_t - 1$ is felső korlát n_t-re.

4.1.1. Lemma. $i_t = \left\lfloor 2^{\frac{t+1}{2}} - \frac{1}{2} \right\rfloor + 1$.

Bizonyítás. Legyen $j_t = \left\lfloor 2^{\frac{t+1}{2}} - \frac{1}{2} \right\rfloor + 1$. Elég belátni, hogy

$$\binom{j_t}{2} < 2^t < \binom{j_t+1}{2}.$$

Mivel $j_t < 2^{\frac{t+1}{2}} + \frac{1}{2} < j_t + 1$, ezért

$$\binom{j_t}{2} < \frac{\left(2^{\frac{t+1}{2}} + \frac{1}{2}\right)\left(2^{\frac{t+1}{2}} - \frac{1}{2}\right)}{2} = 2^t - \frac{1}{8} < \binom{j_t+1}{2},$$

és észrevéve, hogy nem létezik olyan i egész és $t \geq 1$, amire $\binom{i}{2} = 2^t$, következik az állítás. □

4.1.2. Lemma. $\binom{i_t+1}{2} - \binom{i_t-1}{2} > 2^t$.

Bizonyítás. Páros t-re $i_{t+1} = 2^{\frac{t+2}{2}}$, így

$$\begin{aligned}
\binom{i_{t+1}}{2} - \binom{i_t-1}{2} &= \frac{1}{2} \cdot \left[2^{\frac{t+2}{2}}\left(2^{\frac{t+2}{2}} - 1\right) - (i_t-1)(i_t-2)\right] \\
&= \frac{1}{2} \cdot \left[2^{t+2} - 2^{\frac{t+2}{2}} - (i_t-1)i_t + 2(i_t-1)\right] \\
&= 2^{t+1} - \binom{i_t}{2} + (i_t-1) - 2^{\frac{t}{2}} \\
&> 2^t,
\end{aligned}$$

hiszen

$$2^{t+1} - \binom{i_t}{2} > 2^{t+1} - 2^t = 2^t,$$

és

$$\binom{2^{\frac{t}{2}}+1}{2} < 2^t \qquad \Longrightarrow \qquad i_t - 1 \geq 2^{\frac{t}{2}}.$$

Páratlan t-re $i_t = 2^{\frac{t+1}{2}}$. Könnyen ellenőrizhetjük $t = 1$-re és 3-ra. Tegyük fel, hogy $t \geq 5$, ezt csak az utolsó gondolatnál fogjuk felhasználni. Ekkor

$$\begin{aligned}
\binom{i_{t+1}}{2} - \binom{i_t - 1}{2} &= \frac{1}{2} \cdot \left[(i_{t+1} - 1)i_{t+1} - \left(2^{\frac{t+1}{2}} - 1\right)\left(2^{\frac{t+1}{2}} - 2\right)\right] \\
&= \frac{1}{2} \cdot \left[(i_{t+1} + 1)i_{t+1} - 2(i_{t+1} + 1) - 2^{t+1} + 3 \cdot 2^{\frac{t+1}{2}}\right] \\
&= \binom{i_{t+1} + 1}{2} - 2^t + 3 \cdot 2^{\frac{t-1}{2}} - (i_{t+1} + 1) \\
&> 2^t,
\end{aligned}$$

mivel

$$\binom{i_{t+1} + 1}{2} - 2^t > 2^{t+1} - 2^t = 2^t,$$

és

$$\binom{3 \cdot 2^{\frac{t-1}{2}}}{2} - 2^t \qquad \Longrightarrow \qquad 3 \cdot 2^{\frac{t-1}{2}} \geq i_{t+1} + 1.$$

□

4.1.3. Tétel (Chang, Hwang, Lin [2])**.** *Ha $t \geq 4$, akkor*

$$n_t \leq i_t - 1.$$

Bizonyítás. Az állítást t-re vonatkozó indukcióval bizonyítjuk be. Könnyű belátni, hogy $n_4 = 5 = i_4 - 1$.

Az általános esetben tekintsünk egy tetszőleges A algoritmust, erről fogjuk megmutatni, hogy $M_A(2, i_t) > t$. A hajtson végre először egy m elemű tesztet. Ha $m < i_t - (i_{t-1} - 1)$, akkor tekintsük a negatív kimenetet. Ekkor visszavezettük a problémát $S(2, i_t - m)$-re, és mivel $i_t - m > i_{t-1} - 1$ és az indukciós feltevés miatt legalább t tesztre van még szükség. Ha $m \geq i_t - (i_{t-1} - 1)$, akkor a pozitív kimenet után kell még legalább t teszt, hiszen a mintatér elemszáma az első teszt után

$$\binom{i_t}{2} - \binom{i_t - m}{2} \geq \binom{i_t}{2} - \binom{i_{t-1} - 1}{2} > 2^t \qquad \text{4.1.2 szerint.}$$

□

A következő részben látni fogjuk, hogy ez a korlát aszimptotikusan éles.

4.2. Alsó korlát

Legyen az A betűvel jelölt algoritmushoz a kisbetűs $a_t(d)$ a legnagyobb egész, amire A az $S(d, a_t(d))$-t t lépésben megoldja, más szóval a legnagyobb olyan egész, amire A *bizonyítja*, hogy $n_t(d)$-re alsó korlát. A későbbiekben a rövidség kedvéért $a_t(2)$ helyett a_t-t írunk.

Chang, Hwang és Lin [2] alsó korlátot is adtak n_t-re, ezt később Chang, Hwang és Weng [3] megjavították egy C algoritmussal, amire $c_t/n_t > 0.983$. Itt egy olyan U algoritmust is megadtak, mely elég nagy t-re az $u_t/n_t > 0.995$ korlátot adja. Deppe és Lebedev [7] tavaly még tovább javították az alsó korlátot egy W algoritmussal, ami már aszimptotikusan optimális.

4.2.1. Tétel (Deppe, Lebedev [7])**.**

$$w_t \geq \left\lfloor 2^{\frac{t+1}{2}} - t \cdot 2^{\frac{t}{4}} \right\rfloor.$$

Ebben a fejezetben be is bizonyítjuk a tételt, ehhez először is szükség van egy lemmára, aminek a bizonyítását alatta vázoljuk. Egyszerűsítsük a 3.2 szekcióban bevezetett jelölést. Legyen I egy $n+m$ elemű halmaz, és $A, B \subset I$ diszjunktak, melyekre $|A| = m$, és $|B| = n$. A-ban is, B-ben is pontosan egy defektív elem van. Legyen az $(1, m) \times (1, n)$ feladat másik jele $m \times n$ feladat vagy $A \times B$ feladat.

4.2.2. Lemma (Chang, Hwang [1])**.**

$$M(m \times n) = \lceil \log mn \rceil.$$

Egy mintateret A-*diszjunktnak* nevezünk, ha $A \times B$ alakú, és mindegyik A-beli elem egy mintapontban van csak benne.

Legyen S egy mintatér, és P egy algoritmus S-re. Jelölje $v(i)$ az i-edik csúcsot P-ben azon az úton, amelyiken minden kiment pozitív, és legyen $v(i)$ bal (negatív) gyereke $v'(i)$. S elemszámától függően két esetben nevezzük P-t A-*éles* algoritmusnak.

– Ha $|S| = 2^r$, és

  - P legfeljebb r lépés alatt véget ér.

– Illetve ha $|S| = 2^r + 2^{r-1} + \cdots + 2^{r-p} + q$ valamely $0 < p$-re és $0 < q \leq 2^{r-p-1}$-ra, és

  - P legfeljebb $r+1$ lépés alatt véget ér, és
  - $|S(v'(i))| = 2^{r-i}$ minden $0 \leq i \leq p$-re, és
  - $|S(v(p+1))| = q$ és $S(v(p+1))$ A-diszjunkt.

4.2.3. Lemma. *Minden A-diszjunkt mintatérhez létezik A-éles algoritmus.*

Bizonyítás. B elemeit hagyjuk figyelmen kívül, egyértelmű megfeleltetés van A elemei és a mintapontok közt. Így könnyen választhatunk csak A elemeiből is megfelelő méretű teszthalmazokat, és egyszerűen konstruálhatunk egy A-éles algoritmust. □

A következő tétel kulcsfontosságú a 4.2.2 lemma bizonyításában.

4.2.4. Tétel (Chang, Hwang [1])**.** *Legyen m fix, és legyen n_k a legnagyobb egész úgy, hogy $mn_k \leq 2^k$. Ekkor $M(m \times n_k) = k$ minden $n_k \geq 1$-re. Ha m és n_k páratlan, akkor létezik A-éles algoritmus $m \times n_k$-ra.*

A bizonyítás m-re és n_k-ra vonatkozó indukcióval történik. Van olyan k, amire $n_k = 1$, így megtehetjük, hogy ha m vagy n_k páros, a megfelelő halmaz felét tesztelve visszavezetjük kisebb elemszámra, tehát csak páratlan m-et és n_k-t nézünk. Megkeressük a legnagyobb $l < k$-t úgy, hogy n_l páratlan, és annak az esetnek az A-éles algoritmusát felhasználva készítünk n_k-ra is algoritmust. Ennek a menetét itt most nem részletezzük.

A 4.2.2 lemma következik a tételből és abból, hogy 2.1.3 szerint $M(m \times n)$ monoton növő n-ben.

A 4.2.1 tétel bizonyítása. Itt leírjuk a korábban már említett W algoritmust, és belátjuk, hogy teljesíti a tétel állítását. W-t t szerint rekurzívan adjuk meg, először arra az esetre, ha $0 < t \leq 44$, a bizonyítás második felében pedig a nagyobb értékekre.

Legyen $\overline{w}_t = \left\lfloor 2^{\frac{t+1}{2}} - t \cdot 2^{\frac{t}{4}} \right\rfloor$, és I az $S(d, \overline{w}_t)$ alaphalmaza, tehát $|I| = \overline{w}_t$. Az algoritmus során legyen $X, Y \subset I$ úgy, hogy X-ben biztosan van defektív, Y-ban pedig a bizonytalan állapotú elemek vannak.

Bizonyítás. $\mathbf{0 < t \leq 44}$.

Arra az esetre, ha $t \leq 11$, Chang, Hwang és Weng [3] mutattak optimális algoritmust, ezt használjuk, de ennek a leírását itt nem részletezzük.

Ha $12 \leq t \leq 44$, akkor az itt megadott H algoritmust használjuk. Az első teszt mérete legyen $h_t - h_{t-1}$, jelölje ezt x_t^1. Ha ez negatív, akkor kisebb t-re visszavezettük, egyébként a tesztelt elemek mennek X-be, a többi Y-ba. Következő két teszt legyen Y-ból, jelölje a méretüket y_t^2 és y_t^3, és ezekre teljesüljenek az alábbiak:

$$x_t^1 y_t^2 \leq 2^{t-2} \tag{4}$$

$$x_t^1 y_t^3 \leq 2^{t-3} \tag{5}$$

$$h_t - y_t^2 - y_t^3 \leq h_{t-3}. \tag{6}$$

A (4) és az (5) feltételek következménye, hogy ha a második vagy a harmadik teszt pozitív, akkor a 4.2.2 lemma szerint be tudjuk fejezni megfelelő lépésszámmal. A (6) feltételből pedig következik, hogy ha a második és a harmadik teszt is negatív, akkor visszavezettük kisebb t-re az algoritmust, és ismét csak be tudjuk fejezni a megadott lépésszám alatt az algoritmust.

Ha $12 \leq t \leq 20$, akkor az 1. táblázatban jelölt értékeket használjuk, ezekről könnyen ellenőrizhető, hogy megfelelnek a feltételeknek. Ha $t \geq 21$, akkor a tesztek méretét $x^1_{t+2} = 2x^1_t$, $y^2_{t+2} = 2y^2_t$ és $y^3_{t+2} = 2y^3_t$ definiálják, és $h_{t+1} = h_t + x^1_{t+1}$. Így (4) és (5) automatikusan teljesülnek a nagyobb t-kre is. Vegyük észre, hogy $t = 19$-re és $t = 20$-ra (6) feltételben egyenlőség van, ekkor $x^1_t + x^1_{t-1} + x^1_{t-2} = h_t - h_{t-3} = y^2_t + y^3_t$. Legyen most $t \geq 21$, így indukcióval

$$\begin{aligned} h_t - h_{t-3} &= \left(h_{t-2} + x^1_{t-1} + x^1_t\right) - \left(h_{t-5} + x^1_{t-4} + x^1_{t-3}\right) \\ &= (h_{t-2} - h_{t-5}) + \left(3x^1_{t-4} + x^1_{t-3}\right) \\ &= y^2_{t-2} + y^3_{t-2} + x^1_{t-4} + x^1_{t-3} + x^1_{t-2} \\ &= y^2_{t-2} + y^3_{t-2} + y^2_{t-2} + y^3_{t-2} \\ &= y^2_t + y^3_t, \end{aligned}$$

tehát $t \geq 21$-re is egyenlőség van.

A kapott H algoritmusra már $h_t > \overline{w}_t$ teljesül, ha $0 < t \leq 44$ (sőt, $0 < t \leq 50$-re is). H-t használjuk W-nek $0 < t \leq 44$-re, így ekkor $w_t \geq \left\lfloor 2^{\frac{t+1}{2}} - t \cdot 2^{\frac{t}{4}} \right\rfloor$.

t	h_t	x^1_t	y^2_t	y^3_t	$h_t - y^2_t - y^3_t$
9	31				
10	44				
11	63				
12	89	26	39	19	31
13	126	37	55	27	44
14	178	52	78	39	61
15	252	74	110	55	87
16	357	105	156	78	123
17	506	149	219	109	178
18	717	211	310	155	252
19	1015	298	439	219	357
20	1437	422	621	310	506

1. táblázat. A H és W algoritmus első három lépése $12 \leq t \leq 20$ esetén x^1_t, y^2_t, majd y^3_t.

***t* > 44.**

Jelölje a k-adik teszt után X, illetve Y elemszámát rendre x_k és y_k, a mintatér elemszámát A_k. Az adott teszteset részletezésénél a pozitív kimenethez tartozó mintatér elemszám A^+_k, a negatívhoz tartozó pedig A^-_k.

Tegyük fel, hogy $t-1$ tesztre igaz a tétel, és nézzük t esetét. Az algoritmus során a második és $(k+1)$. teszt között X elemszámának csökkentése a cél, és a következő k teszttel az Y halmazt szeretnénk eltüntetni, ezzel kisebb t-re visszavezetve az algoritmust. Kezdetben $X = I$, és $Y = \emptyset$.

1. teszt. Teszteljünk $\overline{x}_1$ elemet X-ből, ahol $\overline{x}_1 = \left\lfloor \left(\sqrt{2} - 1\right) 2^{\frac{t}{2}} \right\rfloor$. Ha a kimenet negatív, akkor a feladatot visszavezettük kisebb t-re, és a maradék feladatra elég $t-1$ lépés, mivel $t \geq 3$-ra $\overline{w}_t - \overline{x}_1 < \overline{w}_{t-1}$.

Ha a kimenet pozitív lett, akkor $|X| = x_1 = \overline{x}_1$, és $|Y| = y_2 = \overline{w}_t - x_1$. Legyen $g(x,y) = \binom{x}{2} + x(y-x)$, a két defektív elem lehetséges helyeinek a száma egy y elemű alaphalmazban egy x elemű pozitív teszt után. Ekkor, mivel a másik esetet már lezártuk, $A_1^+ = A_1 = g(x_1, \overline{w}_t)$. Így

$$\begin{aligned} A_1 &= \binom{x_1}{2} + x_1(\overline{w}_t - x_1) \\ &\leq \frac{\left(\left(\sqrt{2}-1\right)2^{\frac{t}{2}}\right)^2}{2} + \left(\sqrt{2}-1\right)2^{\frac{t}{2}}\left(2^{\frac{t+1}{2}} - t\cdot 2^{\frac{t}{4}} - \left(\sqrt{2}-1\right)2^{\frac{t}{2}}\right) \\ &= \frac{\left(2-2\sqrt{2}+1\right)2^t}{2} + \left(\sqrt{2}-1\right)2^t - t\left(\sqrt{2}-1\right)2^{\frac{3t}{4}} \\ &= 2^{t-1} - t\left(\sqrt{2}-1\right)2^{\frac{3t}{4}}. \end{aligned}$$

Jegyezzük meg, hogy Y elemszámára

$$\begin{aligned} y_1 &\geq \left\lfloor 2^{\frac{t+1}{2}} - t\cdot 2^{\frac{t}{4}} \right\rfloor - \left\lfloor \left(\sqrt{2}-1\right)2^{\frac{t}{2}} \right\rfloor \\ &\geq 2^{\frac{t+1}{2}} - t\cdot 2^{\frac{t}{4}} - 2^{\frac{t+1}{2}} + 2^{\frac{t}{2}} - 1 \\ &\geq 2^{\frac{t}{2}} - t\cdot 2^{\frac{t}{4}} - 1. \end{aligned} \tag{7}$$

2. teszt. Teszteljünk $\overline{x}_2$ elemet X-ből úgy, hogy $\overline{x}_2$-vel $A_2^+ = g(\overline{x}_2, \overline{w}_t)$ a lehető legközelebb legyen $A_1/2$-höz. Nézzük meg a $g(i+1, \overline{w}_t) - g(i, \overline{w}_t)$ különbséget, ezzel becsülhetjük A_2^+ eltérését $A_1/2$-től:

$$\left[\frac{(i+1)i}{2} + (i+1)(\overline{w}_t - i - 1)\right] - \left[\frac{i(i-1)}{2} + i(\overline{w}_t - i)\right] = \overline{w}_t - i - 1 \leq \overline{w}_t.$$

Tehát $\left|\frac{A_1}{2} - A_2^+\right| \leq \overline{w}_t$. Vegyük észre, hogy emiatt $\left|\frac{A_1}{2} - A_2^-\right| \leq \overline{w}_t$ is teljesül, ugyanis A_2^- és A_2^+ particionálják A_1-et. Nekünk a felső becslésre van szükségünk:

$$A_2 \leq 2^{t-2} - t\left(\sqrt{2}-1\right)2^{\frac{3t}{4}-1} + \overline{w}_t.$$

Könnyen látszik g tulajdonságaiból, hogy $\overline{x}_2 < x_1/2$, tehát ha a kimenet negatív volt, akkor $x_2 = x_1 - \overline{x}_2$, $y_2 = y_1$, illetve $x_2 = \overline{x}_2$, $y_2 = y_1 + x_1 - x_2$ a másik esetben.

A $(k+1)$. lépésig hasonlóan járunk el, így mindig X elemszámának kicsit kevesebb, mint a felét teszteljük.

(k+1). teszt. Teszteljünk $\overline{x}_{k+1}$ elemet X-ből úgy, hogy $A_{k+1}^+ = g(\overline{x}_{k+1}, x_k + y_k)$ a lehető legközelebb legyen $A_k/2$-höz. Az előző eset gondolatmenetét alkalmazva

$$A_{k+1} \leq 2^{t-k-1} - t\left(\sqrt{2}-1\right)2^{\frac{3t}{4}-k} + k\overline{w}_t. \tag{8}$$

Mivel definíció szerint $A_{k+1} = g(x_{k+1}, x_{k+1} + y_{k+1})$, könnyen látszik, hogy

$$x_{k+1} \leq \frac{A_{k+1}}{y_{k+1}}. \tag{9}$$

(k+2). teszt. Legyen a tesztelt csoport Y-ból $\left\lfloor \frac{2^{t-k-2}}{x_{k+1}} \right\rfloor$ elem, illetve az egész Y, ha kevesebb eleme van.

Pozitív kimenet esetén a 4.2.2 lemmát használva $t-k-2$ teszttel megtalálható a két defektív. Ellenkező esetben megyünk a következő lépésre.

$(k+3)$. teszttől. A $(k+i)$. tesztben legyen a tesztelt csoport Y-ból $\left\lfloor \frac{2^{t-k-i}}{x_{k+1}} \right\rfloor$ elem, vagy, ha Y kisebb, akkor az egész Y. A stratégia végig ez marad, pozitív kimenet esetén a lemma algoritmusát használjuk a befejezésre, negatív kimenettel pedig tovább megyünk, amíg még $|Y| > 0$. Nézzük azt az esetet, amikor mindegyik kimenet negatív a $(k+1)$. teszt után. Szeretnénk megmutatni, hogy $k = \left\lfloor \frac{t}{4} \right\rfloor$ választással $y_{2k+1} = 0$, tehát több elemet veszünk el Y-ból k teszt alatt, mint amennyit a $(k+1)$. teszt után tartalmazott:

$$R = \sum_{i=k+2}^{2k+1} \left\lfloor \frac{2^{t-i}}{x_{k+1}} \right\rfloor \leq y_{k+1}.$$

(9)-ből tetszőleges a pozitív számra

$$\left\lfloor \frac{a}{x_{k+1}} \right\rfloor \geq \frac{a y_{k+1}}{A_{k+1}} - 1,$$

amiből

$$R \leq \frac{y_{k+1}\left(2^{t-k-1} - 2^{t-2k-1}\right)}{A_{k+1}} - k.$$

Így elég lenne megmutatni, hogy $y_{k+1}\left(2^{t-k-1} - 2^{t-2k-1} - A_{k+1}\right) \leq kA_{k+1}$. Y elemszáma monoton nő a $(k+1)$. tesztig (utána pedig monoton csökken), ezért (7)-et felhasználva

$$y_{k+1} \geq y_1 \geq 2^{\frac{t}{2}} - t \cdot 2^{\frac{t}{4}} - 1.$$

(8)-ból kapjuk, hogy

$$2^{t-k-1} - 2^{t-2k-1} - A_{k+1} \geq t\left(\sqrt{2}-1\right)2^{\frac{3t}{4}-k} - 2^{t-2k-1} - k2^{\frac{t+1}{2}},$$

tehát elég, ha $k = \left\lfloor \frac{t}{4} \right\rfloor$-re teljesül

$$\left(2^{\frac{t}{2}} - t \cdot 2^{\frac{t}{4}} - 1\right)\left(t\left(\sqrt{2}-1\right)2^{\frac{3t}{4}-k} - 2^{t-2k-1} - k2^{\frac{t+1}{2}}\right) \geq$$

$$\geq k\left(2^{t-k-1} - t\left(\sqrt{2}-1\right)2^{\frac{3t}{4}-k} + k2^{\frac{t+1}{2}}\right),$$

ami $t \geq 20$-ra könnyen le is vezethető.

Még hátravan, hogy $k = \left\lfloor \frac{t}{4} \right\rfloor$-re $x_{k+1} \leq w_{t-2k-1}$. Ehhez (9)-et felhasználva

$$x_{k+1} \leq \frac{A_{k+1}}{y_{k+1}} \leq \frac{2^{t-k-1} - t\left(\sqrt{2}-1\right)2^{\frac{3t}{4}-k} + k2^{\frac{t+1}{2}}}{2^{\frac{t}{2}} - t \cdot 2^{\frac{t}{4}} - 1},$$

továbbá

$$w_{t-2k-1} \geq 2^{\frac{t}{2}-k} - (t-2k-1) \cdot 2^{\frac{t-2k-1}{4}} - 1$$

becsléssel az alábbi egyenlőtlenséget kapjuk:

$$2^{t-k-1} - t\left(\sqrt{2}-1\right)2^{\frac{3t}{4}-k} + k2^{\frac{t+1}{2}} \leq$$

$$\leq \left(2^{\frac{t}{2}-k} - (t-2k-1) \cdot 2^{\frac{t-2k-1}{4}} - 1\right)\left(2^{\frac{t}{2}} - t \cdot 2^{\frac{t}{4}} - 1\right).$$

Könnyű ellenőrizni, hogy ez $t > 44$-re valóban teljesül.

□

4.2.5. Következmény. $\frac{w_t}{n_t} \to 1$, *ahogy* $t \to \infty$.

Bizonyítás. Két oldalról becsülve a hányadost $t \geq 4$-re

$$\frac{2^{\frac{t+1}{2}} - t \cdot 2^{\frac{t}{4}} - 1}{2^{\frac{t+1}{2}}} \leq \frac{w_t}{i_t - 1} - \frac{1}{2} \leq \frac{w_t}{n_t} \leq 1,$$

és mivel

$$\frac{2^{\frac{t+1}{2}} - t \cdot 2^{\frac{t}{4}} - 1}{2^{\frac{t+1}{2}}} \to 1,$$

a rendőr-elvet alkalmazhatjuk, és ezzel beláttuk az állítást. □

Hivatkozások

[1] G. J. Chang – F. K. Hwang: A group testing problem on two disjoint sets. In *SIAM Journal on Algebraic Discrete Methods*, 2. évf. (1981) 1. sz., 35–38. p.

[2] G. J. Chang – F. K. Hwang – S. Lin: Group testing with two defectives. In *Discrete Applied Mathematics*, 4. évf. (1982) 2. sz., 97–102. p.

[3] X. M. Chang – F. K. Hwang – J. F. Weng: Group testing with two and three defectives. In *Annals of the New York Academy of Sciences*, 576. évf. (1989), 86–96. p.

[4] P. Damaschke: The algorithmic complexity of chemical threshold testing. In *Algorithms and Complexity*. Lecture Notes in Computer Science sorozat, 1203. köt. Springer Berlin Heidelberg, 1997, 205–216. p.

[5] P. Damaschke: Threshold group testing. In *General Theory of Information Transfer and Combinatorics*. Lecture Notes in Computer Science sorozat, 4123. köt. Springer Berlin Heidelberg, 2006, 707–718. p.

[6] A. DeBonis – L. Gasieniec – U. Vaccaro: Optimal two-stage algorithms for group testing problems. In *SIAM Journal on Computing*, 34. évf. (2005).

[7] C. Deppe – V. S. Lebedev: Group testing problem with two defectives. In *Problems of Information Transmission*, 49. évf. (2013) 4. sz., 375–381. p.

[8] D.-Z. Du – F. K. Hwang: Minimizing a combinatorial function. In *SIAM Journal on Algebraic Discrete Methods*, 3. évf. (1982) 4. sz., 523–528. p.

[9] D.-Z. Du – F. K. Hwang: *Combinatorial Group Testing and its Applications*. 2. kiad. World Scientific, 2000.

[10] David Eppstein – Michael T. Goodrich – Daniel S. Hirschberg: Improved combinatorial group testing algorithms for real world problem sizes. In *SIAM Journal on Computing*, 36. évf. (2007).

[11] M. C. Hu – F. K. Hwang – J. K. Wang: A boundary problem for group testing. In *SIAM Journal on Algebraic Discrete Methods*, 2. évf. (1981) 2. sz., 81–87. p.

[12] F. K. Hwang: A minimax procedure on group testing problems. In *Tamkang Journal of Mathematics*, 2. évf. (1971), 39–44. p.

[13] F. K. Hwang – T. Song – D.-Z. Du: Hypergeometric and generalized hypergeometric group testing. In *SIAM Journal on Algebraic Discrete Methods*, 2. évf. (1981) 4. sz., 426–428. p.

[14] R. M. Kainkaryam – P. J. Woolf: Pooling in high-throughput drug screening. In *Curr. Opin. Drug Discov. Devel.*, 12. évf. (2009).

[15] M.-G. Leu: A note on the hu-hwang-wang conjecture for group testing. In *The ANZIAM Journal*, 49. évf. (2008) 4. sz., 561–571. p.

[16] M.-G. Leu – C.-Y. Lin – S.-Y. Weng: Note on a conjecture for group testing. In *Ars Combinatoria*, 64. évf. (2002), 29–32. p.

[17] L. Riccio – C. J. Colbourn: Sharper bounds in adaptive group testing. In *Taiwanese Journal of Mathematics*, 4. évf. (2000), 669–673. p.

[18] J. N. Srivastava – F. Harary – C. R. Rao – G. C. Rota – S. S. Shrikhande: *A survey of combinatorial theory*. North-Holland, 1973.

[19] A. S. Tanenbaum: *Computer Networks*. 5. kiad. Prentice Hall, 2003.