Jelen jegyzet lényegében Vijay V. Vazirani: Approximation Algorithm című könyvének [11] egy kivonata.

1 Tematika

- alapfogalmak
- HALMAZLEFEDÉS, CSÚCSLEFEDÉS, ÁLTALÁNOSÍTOTT CSÚCSLEFEDÉS; mohó algoritmus, szintezés
- SHORTEST SUPERSTRING: visszavezetés a HALMAZLEFEDÉS problémára, továbbá egy 4-közelítő algoritmus
- MAX-SAT (MAX-k-SAT), random módszer és derandomizáció, ön-visszavezetési fa
- ellipszoid algoritmus
- MAX-VÁGÁS, szemidefinit-programozás, Goemans-Williamson algoritmus
- dualitás, (relaxált) primál/duál kölcsönös feszességi kritériumok, primál-duál séma
- MINIMÁLIS MULTIVÁGÁS és egy közelítő megoldása fák esetén primál-duál sémával
- VÁLLALATELHELYEZÉS és egy közelítő megoldása primál-duál sémával
- LEGKISEBB VEKTOR és egy közelítő megoldása az LLL algoritmus segítségével
- leszámlálási problémák és a #**P** osztály, FPRAS, DNF MEGOLDÁSAINAK LESZÁMLÁLÁSA
- PCP-tétel, KLIKK közelítésének nehézsége

1.1 Alapfogalmak

(α)	**eldöntési problémák**	(ζ)	**optimalizálási problémák**
(β)	nyelvet felismerő algoritmus	(η)	δ faktorú approximációs algoritmus (vagy δ-közelítő algoritmus)
(γ)	(polinomiális) tanúsítvány	(θ)	α faktorú (polinomiális) approximációs tanúsítvány
(δ)	polinomiális idejű visszavezetések	(κ)	approximációs faktort megőrző visszavezetések
(ϵ)	randomizált algoritmusok (**RP**, **ZPP**)	(λ)	δ faktorú randomizált approximációs algoritmus

(α) Eldöntési probléma: adott $L \subseteq \{0,1\}^*$ (általában valamilyen kompakt formában megadva) és $x \in \{0,1\}^*$ esetén válaszoljuk meg, hogy x eleme-e L-nek!

(β) Egy $L \subseteq \{0,1\}^*$ nyelv pontosan akkor eleme **P**-nek, ha van olyan M polinom időkorlátos (determinisztikus) Turing gép, hogy minden $x \in \{0,1\}^*$ esetén $M(x) = 1 \Leftrightarrow x \in L$.

(γ) Egy $L \subseteq \{0,1\}^*$ nyelv pontosan akkor eleme **NP**-nek, ha van olyan p polinom és olyan M polinom időkorlátos (determinisztikus) Turing gép, hogy

- minden $x \in L$-hez található olyan, legfeljebb $p(|x|)$ méretű y, melyre $M(x,y) = 1$, (ezen y-t hívjuk *polinomiális tanúsítvány*nak), és
- minden $x \in \bar{L}$ esetén *minden* legfeljebb $p(|x|)$ méretű y-ra $M(x,y) = 1$.

(δ) Az $L_1 \in \mathbf{NP}$ nyelv polinomiális időben visszavezethető az $L_2 \in \mathbf{NP}$ nyelvre (jelölésben $L_1 \preceq L_2$), ha létezik olyan polinom időkorlátos Turing gép, melynek outputja pontosan akkor L_2-beli, ha inputja L_1-beli.

(ϵ) Egy $L \subseteq \{0,1\}^*$ nyelv pontosan akkor eleme **RP**-nek, ha van olyan p polinom és olyan M polinom időkorlátos (determinisztikus) Turing gép, hogy

- minden $x \in L$ esetén a legfeljebb $p(|x|)$ méretű y-ok legalább felére $M(x,y) = 1$, és
- minden $x \in \bar{L}$ esetén minden legfeljebb $p(|x|)$ méretű y-ra $M(x,y) = 1$.

(ζ) Egy optimalizálási probléma egy $\Pi = (D_\Pi, S_\Pi, \mathrm{obj}_\Pi, m)$ rendezett négyes, ahol

- D_Π: a probléma érvényes példányai. Kell: $I \overset{?}{\in} D_\Pi$ hatékonyan (I méretében — azaz $|I|$-ben — polinomiális idő alatt) eldönthető.
- Az S_Π függvény egy $I \in D_\Pi$ problémapéldányhoz I lehetséges megoldásait rendeli hozzá. Kell: $S_\Pi(I) \neq \emptyset$, és minden $s \in S_\Pi(I)$ mérete I-ben legfeljebb polinomiális, illetve hogy legyen polinomiális algoritmus, mely tetszőleges (s, I) pár esetén eldönti, hogy (s, I) eleme-e $S_\Pi(I)$-nek.
- $\mathrm{obj}_\Pi : \{(s, I) : I \in D_\Pi, s \in S_\Pi(I)\} \to \mathbb{Q}^+$ polinom időben kiszámítható.
- m vagy "minimalizálási feladat" vagy "maximalizálási feladat" attól függően, hogy a legkisebb obj értékű megoldást keressük, vagy a legnagyobbat.

(η) $\mathcal{A}$ egy δ faktorú approximációs algoritmus egy Π minimalizálási problémára ($\delta : \mathbb{Z}^+ \to \mathbb{Q}^+$), ha bármely $I \in D_\Pi$ esetén $\mathcal{A}$ visszaad egy $s \in S_\Pi(I)$ megoldást, melyre $\mathrm{obj}_\Pi(I, s) \leq \delta(|I|)\mathrm{OPT}_\Pi(I)$. (Maximalizálási probléma esetén $\mathrm{obj}_\Pi(I, s) \geq \delta(|I|)\mathrm{OPT}_\Pi(I)$). $\mathcal{A}$-tól nyilván azt is elvárjuk, hogy időben hatékony legyen.

(θ) α faktorú (polinomiális) approximációs tanúsítvány

(κ) Approximációs faktort megőrző visszavezetés egy Π_1 minimalizálási problémáról egy Π_2 minimalizálási problémára egy olyan (f, g) polinomiális futásidejű algoritmus pár, ahol

- minden $I_1 \in \Pi_1$ esetén $I_2 = f(I_1) \in \Pi_2$, és $\mathrm{OPT}_{\Pi_2}(I_2) \leq \mathrm{OPT}_{\Pi_1}(I_1)$, és
- minden $t_2 \in S_{\Pi_2}(I_2)$ esetén $t_1 = g(I_1, t) \in S_{\Pi_1}(I_1)$, melyre $\mathrm{obj}_{\Pi_1}(I_1, t_1) \leq \mathrm{obj}_{\Pi_2}(I_2, t_2)$.

(λ) $\mathcal{A}$ egy δ faktorú randomizált approximációs algoritmus a Π minimalizálási problémához, ha bármely $I \in D_\Pi$ esetén $\mathcal{A}$ kimenete egy olyan $s \in S_\Pi(I)$, melyre

$$\mathbb{P}\Big(\mathrm{obj}_\Pi(I, S) \leq \delta(|I|) \cdot \mathrm{OPT}_\Pi(I)\Big) \geq 1/2$$

(és $\mathcal{A}$ futásideje polinomiális).

Megjegyzés: $L \in$ co-$\mathbf{NP} \Leftrightarrow \bar{L} \in \mathbf{NP}$, $L \in$ co-$\mathbf{RP} \Leftrightarrow \bar{L} \in \mathbf{RP}$, $\mathbf{ZPP} =$ co-$\mathbf{RP} \cap \mathbf{RP}$.

2 Approximációs garancia, és annak javíthatósága

Ha egy $\mathcal{A}$ algoritmus egy problémához annak egy v paramétere alapján állít elő megoldást, és minden $I \in D_\Pi$ esetén fennállnak a következő összefüggések:

(a) $f(v(I)) \leq OPT_\Pi(I)$

(b) $\mathcal{A}(I) \leq g(v(I))$

(c) $g = h(f)$

valamely $f, g, h : \mathbb{Q}^+ \to \mathbb{Q}^+$ függvényekre, akkor h nyilván felső becslés $\mathcal{A}$ approximációs faktorára. Amennyiben az

(a) $f(v(I)) \leq OPT_\Pi(I)$

(b') $OPT_\Pi(I) \leq g(v(I))$

egyenlőtlenségek élesek, h-nál jobb approximációs faktor eléréséhez v-n kívül egy-egy példány egyéb speciális tulajdonságait is ki kell használni.

3 Mohó algoritmus

3.1 Halmazfedés (set cover)

Egy U halmaznak *halmazfedés*e az U valamely részhalmazaiból álló $\{U'_1, \ldots, U'_\ell\}$ halmaz, ha $U = \cup_{i=1}^{\ell} U_i$.

HALMAZFEDÉS	
Input:	egy $(U, \mathcal{U}, c)$ rendezett hármas, ahol • U egy (véges) halmaz, • $\mathcal{U} = \{U_1, \ldots, U_k\}$ az U részhalmazainak egy halmaza, • és $c : \{1, \ldots, k\} \to \mathbb{Q}$ egy *költségfüggvény*.
Output:	egy $S \subseteq \{1, \ldots, k\}$, melyre $\{U_i : i \in S\}$ fedi U-t: $U = \cup_{i \in S} U_i$.
Cél:	$\sum_{i \in S} c(i) \to \min$.

3.2 Logaritmikus approximáció a HALMAZFEDÉS problémához

megjegyzés 3.1. *Tekintsük a* HALMAZFEDÉS *probléma egy $(U, \mathcal{U}, c)$ rendezett hármassal megadott példányát. Egy $u \in U$* elem gyakorisága *alatt az u-nak az $\mathcal{U}$-ban való előfordulásai számát értjük, azaz az $|\{U' \in \mathcal{U} : u \in U'\}|$ értéket. Ha minden elem gyakorisága legfeljebb kettő, akkor a* CSÚCSLEFEDÉS *probléma (súlyozott változatának) egy példányát kapjuk.*

Algorithm 1 MOHÓHALMAZFEDÉS$(U, \{U_1, \ldots, U_k\}, c)$

$i := 0$, $\tilde{U}_0 := \emptyset$ $\quad$ { i: iteráció számláló; $\tilde{U}_i$: az első i iterációban lefedett pontok halmaza}
while $\tilde{U}_i \neq U$ **do**
  $i := i + 1$
  Legyen j_i az $\text{atlagar}_i(j) := c(j)/(|U_j \setminus \tilde{U}_{i-1}|)$ *átlagár-függvény* egy minimumhelye
  {atlagar_i értelmezési tartománya $\{1, \ldots, k\} \setminus \{j_1, \ldots, j_{i-1}\}$ }
  $\tilde{U}_i := \tilde{U}_{i-1} \cup U_{j_j}$
end while
return $S = \{j_1, \ldots, j_i\}$

tétel 3.2. *A* HALMAZFEDÉS *probléma tetszőleges $(U, \mathcal{U}, c)$ példánya, annak bármely S_* optimális megoldása és a* MOHÓHALMAZFEDÉS$(U, \mathcal{U}, c)$ *által szolgáltatott S output esetén $\sum_{j \in S} c(j) \leq (1 + \log |U|) \sum_{j \in S_*} c(j)$.*

Proof. Jelölje t a while ciklus iterációinak számát. A ciklus minden i-edik iterációja után, a j_i kiválasztásánál alkalmazott stratégia illetve $\cup_{j \in S_*} U_j = U$ miatt

$$\sum_{j \in S_*} c(j) \geq \sum_j \left(|U_j \setminus \tilde{U}_{i-1}| \cdot \frac{c(j)}{|U_j \setminus \tilde{U}_{i-1}|} \right) \geq \sum_j |U_j \setminus \tilde{U}_{i-1}| \cdot \text{atlagar}_i(j_i) \geq \text{atlagar}_i(j_i) \cdot |U \setminus \tilde{U}_{i-1}| \tag{1}$$

(ahol nincs másként jelölve, ott a j index mind a szummákban mind a minimalizálás során a $\{j \in S_* : U_j \setminus \tilde{U}_i \neq \emptyset\}$ halmazon fut végig). Felhasználva az átlagárra az előbbi egyenlőtlenségből kapott egyenlőtlenséget kapjuk, hogy

$$\sum_{i \in S}^{t} c(i) = \sum_{i=1}^{t} \mathrm{atlagar}_i(j_i) \cdot |U_{j_i} \setminus \tilde{U}_{i-1}| \overset{(1)}{\leq} \left(\sum_{j \in S_*} c(j) \right) \cdot \sum_{i=1}^{t} \frac{|U_{j_i} \setminus \tilde{U}_{i-1}|}{|U \setminus \tilde{U}_{i-1}|} \leq \left(\sum_{j \in S_*} c(j) \right) \cdot \left(\sum_{t=1}^{|U|} \frac{1}{\ell} \right) ,$$

ahol az utolsó egyenlőtlenség abból következik, hogy minden $1 \leq i \leq t$ egészre

$$\frac{|U_{j_i} \setminus \tilde{U}_{i-1}|}{|U \setminus \tilde{U}_{i-1}|} = \frac{1}{|U \setminus \tilde{U}_{i-1}|} + \cdots + \frac{1}{|U \setminus \tilde{U}_{i-1}|} \leq \frac{1}{|U \setminus \tilde{U}_{i-1}|} + \frac{1}{|U \setminus \tilde{U}_{i-1}| - 1} + \cdots + \frac{1}{|U \setminus \tilde{U}_i| + 1} .$$

□

példa 3.3 (a 3.2 tétel élessége)**.** *Rögzítsünk egy tetszőleges $n \in \mathbb{N}$ és $\epsilon \in (0,1)$ konstanst. Legyen $U = \{1, 2, \ldots, n\}$, álljon $\mathcal{U}$ az $U_1 = \{1\}$, …, $U_n = \{n\}$, $U_{n+1} = \{1, 2, \ldots, n\}$ halmazokból, és legyen $c(n+1) = 1+\epsilon$ illetve $c(i) = 1/i$, $i = 1, \ldots, n$. A* MohóHalmazfedés *az $(U, \mathcal{U}, c)$* Halmazfedés *példányhoz az $1 + 1/2 + 1/3 + \cdots + 1/n > \log n$ költségű $S = \{1, \ldots, n\}$ outputot konstruálja meg, holott az optimális megoldás az $(1+\epsilon)$ költségű $S_* = \{n+1\}$ fedés.*

megjegyzés 3.4. *A fenti mohó algpritmus optimalitását Feige [2] igazolta. Pontosabban azt mutatta meg teszőleges $\delta \in (0,1)$ konstans esetére, hogy ha van olyan algoritmus, mely minden $(U, \mathcal{U}, c)$* Halmazfedés *példányhoz hatékonyan előállít egy olyan fedést, melynek számossága legfeljebb $(1-\delta) \cdot (\log |U|)$-szorosa az optimálisénak, akkor minden* NP*-beli problémához létezik $n^{O(\log \log n)}$ futásidejű determinisztikus algoritmus (ahol n az adott* NP*-beli probléma méretét jelöli)—ami ellentmond az úgynevezett Exponenciális Idő sejtésnek [6].*

megjegyzés 3.5. *Gyakoriságon alapuló approximáció a* Halmazfedés *problémához: lásd a Feladatsort!*

4 Szintezés

4.1 Csúcslefedés (vertex cover)

Legyen $G = (V, E)$ egy tetszőleges gráf. Egy $v \in V$ csúcsra illeszkedő élek számát jelölje $\deg(v)$. Egy $V' \subseteq V$ halmaz *csúcslefedés*, ha az minden E-beli élnek legalább egyik végpontját tartalmazza. Egy $M \subseteq E$ halmaz *párosítás*, ha semelyik két különböző M-beli élnek nincs közös végpontja.

Csúcslefedés

Input: egy (G, w) pár, ahol
- $G = (V, E)$ egy gráf,
- $w : V \to \mathbb{Q}$ pedig egy *súlyfüggvény*.

Output: egy V' fedés.

Cél: $\sum_{v \in V'} w(v) \to \min$. Jelölje e minimumot MinCsL(G,w)

definíció 4.1. *Egy* Csúcslefedés *példány* súlyozatlan, *ha a súlyfüggvénye konstans* 1.

4.2 2-közelítő algoritmus a Csúcslefedés problémához

definíció 4.2. *Legyen $G = (V, E)$ egy tetszőleges gráf. $w : V \to \mathbb{Q}$* fokszámarányos súlyozás *G-hez, ha van olyan $c \in \mathbb{R}$ konstans, hogy minden $v \in V$ csúcs esetén* $\deg(v) = c \cdot w(v)$.

lemma 4.3. *Legyen $G = (V, E)$ tetszőleges gráf, $w : V \to \mathbb{Q}$ pedig egy fokszámarányos súlyozás G-hez. Ekkor G bármely csúcslefedésének értéke legalább* $(1/2) \cdot \sum_{v \in V} w(v)$.

Bizonyítás: Legyen c az a konstans, melyre $c \cdot w(v) = \deg(v)$ minden $v \in V$ csúcsra. Ekkor egyrészt $\sum_{v \in V} c \cdot w(v) = 2|E|$, hiszen a V-beli csúcsok fokszámösszege pontosan $2|E|$. Másrészt viszont bármely V' fedésre $\sum_{v \in V'} c \cdot w(v) \geq |E|$, hiszen a V'-beli csúcsok fokszámösszege legalább $|E|$. Tehát $\sum_{v \in V} c \cdot w(v) \leq 2 \sum_{v \in V'} c \cdot w(v)$, amiből $2c$-vel való osztás után kapjuk a lemma állítását. □

Algorithm 2 SzintezésesCsúcslefedés$(G = (V, E), w : V \to \mathbb{Q})$

$i := 0$, $w_0 := w$ {i: iteráció számláló; w_i: aktuális súlyozás;}
$A_0 := \emptyset$, $B_0 := \emptyset$ {A_i: beválasztott pontok; B_i: kihagyandó pontok}
repeat
  $i := i + 1$
  $\hat{w}_i : V \to \mathbb{Q}$ súlyfüggvényt válasszuk a 4.4 tételben leírtak szerint {ezáltal $0 \leq \hat{w}_i \leq w_{i-1}$}
  $w_i := w_{i-1} - \hat{w}_i$ {ezáltal $w_i = w - \sum_{j=1}^{i} w_j$}
  $A_i := \{v \in V : w_i(v) = 0\}$ {ezáltal $A_i \supseteq A_{i-1} \supseteq \cdots \supseteq A_1$}
  $B_i := \{v \in V : v$ szomszédai mind A_i-beliek$\}$ {ezáltal $B_i \supseteq B_{i-1} \supseteq \cdots \supseteq B_1$}
until $V = A_i \cup B_i$
return A_i

tétel 4.4. *Legyen $(G = (V, E), w : V \to \mathbb{Q})$ egy tetszőleges példánya a* Csúcslefedés *problémának. Legyen továbbá A a* SzintezésesCsúcslefedés *algoritmus outputja a (G, w) inputon, az i-edik iterációban a $\hat{w}_i$ függvényt úgy választva, hogy az* *(a) az $(A_{i-1} \cup B_{i-1})$ halmazon azonosan 0,*

(b) sehol sem nagyobb w_{i-1}-nél, de van olyan $v \in V_i = V \setminus (A_{i-1} \cup B_{i-1})$, hogy $\hat{w}_i(v) = w_{i-1}(v)$,

(c) a V_i halmazra megszorítva fokszámarányos súlyozás a G gráf V_i által feszített részgráján.

Ekkor az algoritmus repeat ciklusa legfeljebb $|V|$ alkalommal fut le, és bármely V' fedés esetén $\sum_{v\in A} w(v) \leq 2\sum_{v\in V'} w(v)$ teljesül. Speciálisan $\sum_{v\in A} w(v) \leq 2 \cdot \textsc{MinCsL}(G, w)$.

Bizonyítás: Jelölje i_0 az iterációk számát. $i_0 \leq |V|$ igazolásához először is vegyük észre, hogy A_i mindig bővebb lesz A_{i-1}-nél (pl. a (b) pontban megadott feltételt teljesítő v benne lesz A_i-ben, de nem lesz eleme A_{i-1}-nek), és így $|A_i| \geq i$, $i = 1, \ldots, i_0$. Másfelől viszont $A_i \subseteq V$, és így $|A_i| \leq |V|$. Következésképpen $i_0 \leq |A_{i_0}| \leq |V|$.

A tétel másik állításának igazolásához először vegyük észre, hogy

$$w(v) = \left(\overset{(A_i\text{konstr.})}{=} w_i(v) \overset{(w_i\text{konstr.})}{=} \sum_{j=1}^{i} \hat{w}_i(v) \overset{(a)}{=} \right) = \sum_{j=1}^{i_0} \hat{w}_i(v), \qquad v \in (A_i \setminus A_{i-1}),\ \ i = 1, \ldots, i_0 \ , \tag{2}$$

$$w(v) = \left(\overset{(B_i\text{konstr.})}{\geq} w_i(v) \overset{(w_i\text{konstr.})}{=} \sum_{j=1}^{i} \hat{w}_i(v) \overset{(a)}{=} \right) = \sum_{j=1}^{i_0} \hat{w}_i(v), \qquad v \in (B_i \setminus B_{i-1}),\ \ i = 1, \ldots, i_0 \ . \tag{3}$$

Az algoritmus outputjának összköltsége (bevezetve az $A_0 = \emptyset$ jelölést)

$$\sum_{v\in A} w(v) = \left(\overset{(2)}{=} \sum_{v\in A_{i_0}} \sum_{i=1}^{i_0} \hat{w}_i(v) = \sum_{i=1}^{i_0} \sum_{v\in A_{i_0}} \hat{w}_i(v) \overset{(a)}{=} \sum_{i=1}^{i_0} \sum_{v\in (A_{i_0}\setminus A_{i-1})} \hat{w}_i(v) \overset{(V_i\text{ def.})}{\leq} \right) \leq \sum_{i=1}^{i_0} \sum_{v\in V_i} \hat{w}_i(v) \ .$$

Ha V' csúcslefedés G-hez, akkor $V' \cap V_i$ is csúcslefedés G $(V' \cap V_i)$ által feszített részgráfjához, így

$$\sum_{v\in V_i} \hat{w}_i(v) \overset{(4.3\text{ lemma})}{\leq} 2 \sum_{v\in (V'\cap V_i)} \hat{w}_i(v) \ .$$

Végül pedig

$$\sum_{i=1}^{i_0} \sum_{v\in (V'\cap V_i)} \hat{w}_i(v) \leq \left(\leq \sum_{i=1}^{i_0} \sum_{v\in V'} \hat{w}_i(v) = \sum_{v\in V'} \sum_{i=1}^{i_0} \hat{w}_i(v) \overset{(2)\&(3)}{\leq} \right) \leq \sum_{v\in V'} w(v) \ .$$

□

megjegyzés 4.5. *Korábbi eredményeket megjavítva Håstad [5] megmutatta, hogy amennyiben* $\mathbf{P} \neq \mathbf{NP}$*, akkor bármely $\epsilon \in (0,1)$ esetén egy hatékony algoritmus nem képes tetszőleges gráfhoz mindig olyan fedést konstruálni, melynek súlya az otpimálisénak legfeljebb $(7/6-\epsilon)$-szorosa. Khot és Regev [9] pedig azt igazolta, hogy ha még egy erősebb sejtés, az úgynevezett Unique Games sejtés [7] is teljesül, akkor nemcsak az optimum $(7/6-\epsilon)$-szorosát, hanem a $(2-\epsilon)$-szorosát sem lehet elérni hatékony algoritmussal — a fenti algoritmus tehát lényegében optimális.*

5 Visszavezetés

5.1 Legrövidebb szófedés (shortest superstring)

Egy Σ *ábécé feletti szó* (vagy egyszerűen *szó*) alatt a $\Sigma^* = \cup_{n=0}^{\infty}\Sigma^n$ halmaz egy tetszőleges elemét értjük. Egy $a = (a_1, \ldots, a_n) \in \Sigma^n$ *résszava* egy $b = (b_1, \ldots, b_m) \in \Sigma^m$ szónak, ha létezik olyan $\ell \leq m - n$ nemnegatív egész, melyre $a_1 = b_{\ell+1}, \ldots, a_n = b_{\ell+n}$. Az $a \in \Sigma^n$ szó *hossza* n (jelölésben: $|a| = n$). Egy $s \in \Sigma$ szó *fedése* szavak egy $\mathcal{S} \subseteq \Sigma^*$ halmazának, ha minden $\mathcal{S}$-beli szó részszava s-nek. Végül pedig egy s szó esetén jelölje $\text{reszszo}(s)$ az s összes részszavát tartalmazó halmazt.

LegrövidebbSzófedés	
Input:	szavak egy véges $\mathcal{S} = \{s_1, \ldots, s_n\}$ halmaza.
Output:	$\mathcal{S}$ egy s fedése
Cél:	$\lvert s \rvert \to \min.$

Szavak egy tetszőleges $\mathcal{S} = \{s_1, \ldots, s_n\}$ halmaza esetén legyen

$$\mathcal{S}^\circ = \left\{\sigma \;:\; s', s'' \in \mathcal{S},\; |\sigma| = |s''| + k,\; \sigma_i = s'_i,\; \sigma_{k+j} = s''_j,\; i = 1, \ldots, |s'|,\; j = 1 \ldots, |s''|\right\} .$$

5.2 Logaritmikus approximáció a LegrövidebbSzófedés problémához

tétel 5.1. *Legyen* $\mathcal{S} = \{s_1, \ldots, s_n\}$ *a* LegrövidebbSzófedés *egy példánya. Futtassuk a* MohóHalmazfedés *algoritmust az* $(\mathcal{S}, \{\text{reszszo}(\sigma) : \sigma \in \mathcal{S}^\circ\}, c)$ HalmazFedés *példányon, ahol* $c(\text{reszszo}(\sigma)) = |\sigma|$. *Az algoritmus outputja* $\{\text{reszszo}(\sigma) : \sigma \in \mathcal{S}'\}$ *valamely* $\mathcal{S}' \subseteq \mathcal{S}^\circ$ *részhalmazra. Ezen felül teljesül, hogy az* $\mathcal{S}'$*-beli szavak (tetszőleges sorrendben történő) összefűzésével előállított* s *szó fedi* $\mathcal{S}$*-et, hossza pedig legfeljebb* $(2 + 2 \cdot \log n)$*-szerese az* $\mathcal{S}$ *bármely másik fedésének.*

Bizonyítás: Az általánosság megszorítása nélkül feltehető, hogy $\mathcal{S}$ egyik szava sem részszava a másiknak. Az alábbiak során az s hosszára adott korlátot igazoljuk; a tétel többi állítása triviális.

1. Legyen $s' \in \Sigma^*$ egy fedése $\mathcal{S}$-nek.
2. Helyezzük el az $\mathcal{S}$-beli szavakat úgy, hogy mind megfelelően illeszkedjen s'-re.
3. Alakítsunk ki blokkokat, az $i+1$-edikbe mindig belevéve az első i blokkból kimaradtak közül egyrészt a legbaloldalibbat, másrészt pedig az ezzel fedésben levőket — mást semmit. Jelölje t a blokkok számát.
4. Legyen π_i az s' szó azon része, amelyet az i-edik blokkban levő szavak fednek. Ekkor $\pi_i \in \mathcal{S}^\circ$.

```
s   | ABAAACDABECDABCAADEABCEEEB
----+---------------------------
s7  | ABAAACD
s5  |   AAACDABEC
s1  |     ACDABECDAB
s3  |           CDABC
s9  |            DABCAAD
s6  |                AADEAB
s4  |                 ADEABCEE
s8  |                  DEABCEEE
s2  |                         EB
----+---------------------------
π1  | ABAAACDABECDAB
π2  |           CDABCAAD
π3  |                AADEABCEEE
π4  |                         EB
```

A $\{\text{reszszo}(\pi_i) : i = 1, \ldots, t\}$ halmaz nyilvánvalóan fedése az állításban megadott $(\mathcal{S}, \{\text{reszszo}(\sigma) : \sigma \in \mathcal{S}^\circ\}, c)$ Halmazfedés problémának, a blokkok konstrukciójából adódóan pedig $|s'| \geq \sum_{i=1}^{\lceil t/2 \rceil} |\pi_{2i-1}|$ továbbá $|s'| \geq \sum_{i=1}^{\lfloor t/2 \rfloor} |\pi_{2i}|$, azaz ezen fedés költsége legfeljebb $2|s'|$. Következésképpen (a 3.2 tételből adódóan) $\sum_{\sigma \in \mathcal{S}'} |\sigma| \leq (\log n + 1) \cdot 2 \cdot |s'|$. □

megjegyzés 5.2. *A* LegrövidebbSzófedés *problémához adható olyan algoritmus, mely outputjának hossza mindig legfeljebb háromszorosa az optimálisénak (lásd pl. Vazirani könyvét [11]).*

6 Visszavezetés 2: súlyozott és súlyozatlan problémák

A jelen fejezetben példát látunk arra, hogyan vezethetjük vissza a súlyozott optimalizálási problémák közelítését súlyozatlanokéra. Az alábbiakban föltesszük, hogy a súlyok mind egészek; ez mégsem jelent különösebb megszorítást, hiszen bármely w súlyfüggvényt felszorozva az értékkészletében lévő törtek nevezőinek legkisebb közös többszörösével, az eredetivel ekvivalens példányokat kapunk (legalábbis az általunk vizsgált problémák esetében).

6.1 MinSAT

MinSAT	
Input:	egy $(\mathcal{C}, w)$ pár, ahol
	• $\mathcal{C}$ klózok halmaza,
	• $w : \mathcal{C} \to \mathbb{N}$ pedig egy súlyfüggvény.
Output:	φ változóinak x értékadása.
Cél:	$\sum_{C \in \mathcal{C}: C(x)=1} w(C) \to \min$. Jelölje e minimumot MinSAT$(\mathcal{C}, w)$.

A MinSAT probléma egy $(\mathcal{C}, w)$ példánya *súlyozatlan*, ha a w súlyfüggvény azonosan 1.

6.2 A MinSAT, a Csúcslefedés, a súlyozatlan MinSAT és a súlyozatlan Csúcslefedés egyformán jól közelíthető

A következő tételt négy lépésben látjuk be, a négy problémát láncszerűen egymásra visszavezetve (és felhasználva, hogy mivel minden súlyozatlan MinSAT példány egyben súlyozott MinSAT példány is, a súlyozatlan változat legalább olyan jól közelíthető, mint a súlyozott).

tétel 6.1. *A* MinSAT*, a súlyozatlan* MinSAT*, a* Csúcslefedés *és a súlyozatlan* Csúcslefedés *problémák lényegében ugyanolyan jól közelíthetők: ha bármelyikhez van olyan hatékony algoritmus, mely az adott probléma tetszőleges példányához mindig olyan megoldást generál, melynek súlya a minimálisénak legfeljebb α-szorosa, akkor tetszőleges $\epsilon > 0$ konstans esetén létezik a másik háromhoz is olyan hatékony algoritmus, mely minden példányhoz olyan megoldást konstruál, melynek súlya a minimálisénak legfeljebb $(\alpha + \epsilon)$-szorosa.*

Első lépésként azt látjuk be, hogy a súlyozatlan Csúcslefedés legalább olyan jól közelíthető, mint a súlyozatlan MinSAT.

lemma 6.2. *Tegyük fel, hogy van olyan $\alpha \geq 1$ konstans és A hatékony algoritmus, melynek tetszőleges $\mathcal{C}$ klózhalmaz melletti $A(\mathcal{C})$ outputja egy olyan értékadás, mely legfeljebb $\alpha \cdot$ MinSAT$(\mathcal{C}, \mathbb{1})$ darabot elégít ki $\mathcal{C}$ klózai közül. (Itt $\mathbb{1}$ az azonosan 1 függvényt jelöli.) Ekkor létezik olyan hatékony B algoritmus, mely tetszőleges G gráfhoz előállítja annak egy olyan $B(G)$ csúcsfedését, melyben legfeljebb $\alpha \cdot$ MinCsL$(G, \mathbb{1})$ csúcs szerepel.*

Bizonyítás: Legyen G tetszőleges gráf E élhalmazzal és n elemű $V = \{v_1, \ldots, v_n\}$ csúcshalmazzal. Konstruáljuk meg a $C_i = \left(\bigvee_{j<i:(v_i,v_j)\in E} x_{i,j}\right) \vee \left(\bigvee_{j>i:(v_i,v_j)\in E} \overline{x}_{i,j}\right)$, $i = 1, \ldots, n$ klózokat, és legyen $\mathcal{C} = \{C_1, \ldots, C_n\}$.

Ekkor egyrészt teljesül $\text{MinSAT}(\mathcal{C}, \mathbb{1}) \leq \text{MinCsL}(G, \mathbb{1})$, hiszen G tetszőleges $V' \subseteq V$ csúcslefedéséhez létezik egy olyan x értékadás, mely bármely $v_i \notin V'$ és $(v_i, v_j) \in E$ él esetén $x_{i,j}$-hez pontosan akkor rendel igazat, ha $i > j$, és így C_i-t hamissá teszi.

Másrészt $B(G) = \{v_i : A(\mathcal{C}) \text{ mellett igaz } C_i\}$ a G-nek csúcsfedése lesz, és pontosan annyi csúcsot tartalmaz, amennyi $\mathcal{C}$-beli klózt $A(\mathcal{C})$ kielégít.

Tehát $B(G)/\text{MinCsL}(G, \mathbb{1}) \leq A(\mathcal{C})/\text{MinSAT}(\mathcal{C}, \mathbb{1}) \leq \alpha$. □

A második lépésben azt igazoljuk, hogy a súlyozott Csúcslefedés ugyanolyan jól közelíthető, mint a súlyozatlan, amennyiben a súlyok “nem túl nagyok”.

lemma 6.3. *Tegyük fel, hogy van olyan $\alpha \geq 1$ konstans és A hatékony algoritmus, mely tetszőleges G' gráfhoz előállítja annak egy olyan $A(G')$ csúcsfedését, melyben legfeljebb $\alpha \cdot \text{MinCsL}(G', \mathbb{1})$ csúcs szerepel. (Itt $\mathbb{1}$ az azonosan 1 függvényt jelöli.) Ekkor létezik olyan hatékony B hatékony algoritmus, mely tetszőleges G gráf és tetszőleges $\tilde{w} : V(G) \to \{0, 1, \ldots, n^2\}$ súlyfüggvény esetén (ahol n a G csúcsszámát jelöli) előállítja G egy olyan $V_0 = B(G, \tilde{w})$ csúcslefedését, melynek $\tilde{w}$ melletti összsúlya $\sum_{v \in V_0} \tilde{w}(v) \leq \alpha \cdot \text{MinCsL}(G, \tilde{w})$.*

Bizonyítás: Legyen G tetszőleges gráf E élhalmazzal és n elemű $V = \{v_1, \ldots, v_n\}$ csúcshalmazzal. Konstruáljuk meg a $G' = (V', E')$ gráfot, ahol $V' = \{v_{i,\ell} : 1 \leq i \leq n,\ 1 \leq \ell \leq \tilde{w}(v_i)\}$ és $E' = \{(v_{i,\ell}, v_{j,k}) : (v_i, v_j) \in E, 1 \leq \ell \leq \tilde{w}(v_k), 1 \leq \ell \leq \tilde{w}(v_k)\}$.

Ekkor egyrészt teljesül $\text{MinCsL}(G', \mathbb{1}) \leq \text{MinCsL}(G, \tilde{w})$, hiszen G' tetszőleges $V_1 \subseteq V'$ csúcslefedése esetén $V_0 = \{v_{i,\ell} : v_i \in V_1, 1 \leq \ell \leq \tilde{w}(i)\}$ csúcslefedése lesz G-nek, ami ráadásul legfeljebb annyi csúcsot tartalmaz, amekkora V_1 $\tilde{w}$ melletti $\sum_{v \in V_0} \tilde{w}(v)$ súlya.

Másrészt könnyen látható, hogy $B(G) = \{v_i : v_{i,\ell} \in A(G'), \ell = 1, \ldots \tilde{w}(i)\}$ csúcsfedése G-nek (ugyanis minden $(v_i, v_j) \in E$ él esetén, ha $v_{i,\ell} \notin A(G)$ valamely $1 \leq \ell \leq \tilde{w}(v_i)$ indexre, akkor a $\{(v_{i,\ell}, v_{j,k}) : 1 \leq \ell \leq \tilde{w}(v_k), 1 \leq \ell \leq \tilde{w}(v_k)\}\} \subseteq E'$ élek lefogásához $A(G)$-nek tartalmaznia kell minden $v_{j,k}$, $1 \leq k \leq \tilde{w}(v_j)$, csúcsot), ráadásul $\tilde{w}$ melletti $\sum_{v \in B(G)} \tilde{w}(v)$ súlya nem haladja meg $A(G')$ számosságát. Továbbá az is teljesül, hogy G' megkonstruálásának időigénye n-ben polinomiális.

Tehát $B(G)/\text{MinCsL}(G, \tilde{w}) \leq A(G')/\text{MinSAT}(G', \mathbb{1}) \leq \alpha$. □

A következő lépés annak belátása, hogy az előző lépésnél megfogalmazott súlykorlát nem túl nagy megkötés: a Csúcslefedés tetszőleges példányához lényegében ugyanolyan jó közelítő megoldás adható, mint azokhoz, amelyeknél a súlyok “nem túl nagyok”.

lemma 6.4. *Tegyük fel, hogy van olyan $\alpha \geq 1$ konstans és A hatékony algoritmus, hogy A minden $G = (V, E)$ gráf és minden $\tilde{w} : V(G) \to \{0, 1, \ldots, n^2\}$ súlyfüggvény esetén (ahol n a G csúcsszámát jelöli) a G egy olyan V_0 csúcslefedését adja vissza, melyre $\sum_{v \in V_0} \tilde{w}(v) \leq \alpha \cdot \text{MinCsL}(G, \tilde{w})$. Ekkor bármely $\epsilon > 0$ konstanshoz létezik olyan hatékony algoritmus, mely minden $G = (V, E)$ gráf és minden $w : V(G) \to \mathbb{N}$ súlyfüggvény esetén G egy olyan V_2 csúcslefedését adja vissza, melyre $\sum_{v \in V_2} \tilde{w}(v) \leq (\alpha + \epsilon) \cdot \text{MinCsL}(G, w)$.*

Bizonyítás: Legyen G tetszőleges gráf, $w : V(G) \to \mathbb{N}$ pedig tetszőleges súlyfüggvény. Legyen továbbá $V_1 = \text{SzintezésesCsúcslefedés}(G, w)$ és $s_1 = s_1(G, w) = \sum_{v \in V_1} w(v)$. Jelölje végül n a G csúcsainak számát, $\tilde{w} : V(G) \to \{0, 1, \ldots, n^2\}$ pedig azt a súlyfüggvényt, melyre

$$\tilde{w}(v) = \begin{cases} \left\lfloor \frac{n^2 \cdot w(v)}{s_1} \right\rfloor, & \text{ha } w(v) \leq s_1, \\ n^2 & \text{egyébként} \end{cases},$$

$v \in V(G)$. Ekkor tehát G bármely V' csúcsfedése esetén

$$\sum_{v \in V'} w(v) \geq \frac{s_1}{n^2} \cdot \sum_{v \in V'} \tilde{w}(v) \geq \frac{s_1}{n^2} \cdot \textsc{MinCsL}(G, \tilde{w}(v)) \ ,$$

ami miatt

$$\textsc{MinCsL}(G, w) \geq \frac{s_1}{n^2} \cdot \textsc{MinCsL}(G, \tilde{w}(v)) \ . \tag{4}$$

Legyen most $V_0 = A(G, \tilde{w})$, és legyen a V_2 csúcslefedés V_0, amennyiben $w(v) \leq s_1$ minden $v \in V_0$ csúcs esetén, illetve legyen $V_2 = V_1$ egyébként. Bármelyik eset is adódik elő,

$$\sum_{v \in V_0} \tilde{w}(v) \geq \sum_{v \in V_2} \tilde{w}(v) \tag{5}$$

mindenképpen teljesül, ugyanis ha $V_2 \neq V_0$, akkor $w(v_0) > s_1$ valamely $v_0 \in V_0$ csúcs esetén, és így

$$\sum_{v \in V_0} \tilde{w}(v) \geq \tilde{w}(v_0) = n^2 = \frac{n^2}{s_1} \cdot s_1 = \frac{n^2}{s_1} \cdot \sum_{v \in V_1} w(v) \geq \sum_{v \in V_1} \tilde{w}(v) = \sum_{v \in V_2} \tilde{w}(v) \ .$$

Ezen felül V_2 minden v csúcsára $w(v) \leq s_1$, ebből adódóan pedig

$$\sum_{v \in V_2} w(v) \leq \frac{s_1}{n^2} \cdot \sum_{v \in V_2} (1 + \tilde{w}(v)) \leq \frac{s_1}{n^2} \cdot \left(n + \sum_{v \in V_2} \tilde{w}(v) \right) \ . \tag{6}$$

Mindezekből következően

$$\begin{aligned}
\frac{\sum_{v \in V_2} w(v)}{\textsc{MinCsL}(G, w)} &\overset{(6)}{\leq} \frac{(s_1/n^2) \cdot (n + \sum_{v \in V_2} \tilde{w}(v))}{\textsc{MinCsL}(G, w)} = \\
&= \frac{s_1}{n \cdot \textsc{MinCsL}(G, w)} + \frac{(s_1/n^2) \cdot \sum_{v \in V_2} \tilde{w}(v)}{\textsc{MinCsL}(G, w)} \leq \\
&\overset{(4)}{\leq} \frac{1}{n} + \frac{\sum_{v \in V_2} \tilde{w}(v)}{\textsc{MinCsL}(G, \tilde{w})} \leq \\
&\overset{(5)}{\leq} \frac{1}{n} + \frac{\sum_{v \in V_0} w(v)}{\textsc{MinCsL}(G, w)} \leq \\
&\leq \frac{1}{n} + \alpha \ .
\end{aligned}$$

Azaz $n < 1/\epsilon$ esetén az összes lehetséges csúcslefedést végigpróbálva, $n \geq 1/\epsilon$ esetén pedig V_2-t visszaadva, egy polinomiális futásidejű algoritmussal megkonstruáltuk G egy olyan csúcslefedését, melynek w melletti összsúlya a minimálisan elérhetőnek legfeljebb $(\alpha + \epsilon)$-szorosa. $\square$

Végül megmutatjuk, hogy a MinSAT legalább olyan jól közelíthető, mint a Csúcslefedés.

lemma 6.5. *Tegyük fel, hogy van olyan $\alpha \geq 1$ konstans és A hatékony algoritmus, mely minden $G = (V, E)$ gráf és minden $w : V(G) \to \mathbb{N}$ súlyfüggvény esetén G egy olyan V_0 csúcslefedését adja vissza, melyre $\sum_{v \in V_0} w(v) \leq \alpha \cdot \textsc{MinCsL}(G, w)$. Ekkor létezik olyan hatékony B algoritmus, mely tetszőleges $\mathcal{C}$ klózhalmazt és $w : \mathcal{C} \to \mathbb{N}$ súlyfüggvényt kapva inputként előállít egy olyan értékadást, mely által kielégített $\mathcal{C}$-beli klózok w melletti összsúlya legfeljebb $\alpha \cdot \textsc{MinSAT}(\mathcal{C}, \mathbb{1})$.*

Bizonyítás: Legyen $\mathcal{C} = \{C_1, \dots, C_n\}$ klózok egy tetszőleges halmaza, $w : \mathcal{C} \to \mathbb{N}$ pedig egy tetszőleges súlyfüggvény. Konstruáljuk meg a $G = (V, E)$ gráfot, ahol $V = \{1, \dots, n\}$ és $E = \{(i, j) :$ van olyan változó, mely C_i és C_j egyikében negálva, másikában negálatlanul szerepel$\}$.

Ekkor egyrészt teljesül $\textsc{MinCsL}(G, w) \leq \textsc{MinSAT}(\mathcal{C}, w)$, hiszen tetszőleges b értékadás esetén $V_0 = \{i : b$ kielégíti a $C_i$klózt$\}$ csúcslefedés (hiszen bármely $(i, j) \in E$ él esetén C_i és C_j egyike mindenképpen igaz kell legyen).

Másrészt a $\{C_i \in \mathcal{C} : i \notin A(G, w)$ klózhalmazban egyik változó sem fordul elő negálva és negálatlanul is, ezért definiálható olyan $B(\mathcal{C}, w)$ értékadás, mely az előbbiekhez igaz, az utóbbiakhoz pedig hamis értéket rendel. Ráadásul emellett minden olyan $\mathcal{C}$-beli klóz hamis lesz, aminek indexe nem szerepel V_0-ban — így a kielégített klózok w melletti összsúlya sem haladja meg V_0-ét.

Tehát $B(\mathcal{C}, w)/\textsc{MinCsL}(\mathcal{C}, w) \leq A(G, w)/\textsc{MinSAT}(G, w) \leq \alpha$. $\square$

megjegyzés 6.6. *Crescenzi, Silvestri és Trevisan ezen felül a* $\textsc{MaxCut}$ *súlyozott változatáról is megmutatta, hogy ugyanolyan jól közelíthető, mint a súlyozott (lásd [1]).*

7 Derandomizálás a feltételes várható értékek módszerével

7.1 MaxSAT

Egy x értékadás és egy C klóz esetén legyen $C(x) = 1$ (másként: x kielégíti a C klózt), ha x mellett legalább egy C-beli literál igaz értéket vesz fel; egyébként legyen $C(x) = 0$.

MaxSAT	
Input:	egy $(\mathcal{C}, w)$ pár, ahol
	• $\mathcal{C}$ klózok halmaza,
	• $w : \mathcal{C} \to \mathbb{Q}^+$ pedig egy súlyfüggvény.
Output:	φ változóinak x értékadása.
Cél:	$w(x, \mathcal{C}) = \sum_{C \in \mathcal{C} : C(x)=1} w(C) \to \max.$

7.2 (1/2)-közelítés a MaxSAT problémához

Legyen $(\mathcal{C}, w)$ egy tetszőleges példánya a MaxSAT problémának. Egy $C \in \mathcal{C}$ *klóz hossza* alatt a literáljainak számát értjük (jelölésben $\mathrm{hossz}(C)$). Vezessük még be az $\alpha_k = 1 - 2^{-k}$ jelölést.

Az egyszerűség kedvéért tegyük fel, hogy $\mathcal{C}$ a $v_1, \ldots, v_n$ változók felett van értelmezve, és legyen $X = (X_1, \ldots, X_n)$ az a véletlen értékadás, mely a változóknak egymástól függetlenül 1/2 valószínűséggel igaz, illetve hamis értéket ad. Ekkor tetszőleges $C \in \mathcal{C}$ klózra illetve $x_1, \ldots, x_i \in \{0, 1\}$, $1 \le i \le n$, értékekre

$$\mathbb{P}\left[C(X) = 1 \middle| X_1 = x_1, \ldots, X_i = x_i\right] = \mathbb{E}\left[C(X) \middle| X_1 = x_1, \ldots, X_i = x_i\right] =$$
$$= \begin{cases} 1, & \text{ha valamely } j \le i \text{ esetén a } j\text{-edik változó negálva szerepel } C\text{-ben és } x_j = 0, \\ & \text{vagy pedig negálatlanul szerepel } C\text{-ben és } x_j = 1, \\ \alpha_{\ell(C,i)} & \text{egyébként, ahol } \ell(C,i) \text{ az } i\text{-nél nagyobb indexű } C\text{-beli változók száma ,} \end{cases} \tag{7}$$

továbbá a várható érték linearitása miatt

$$\begin{aligned} w_{x_1,\ldots,x_i}(\mathcal{C}) &= \mathbb{E}\left[w(X, C) \middle| X_1 = x_1, \ldots, X_i = x_i\right] = \\ &= \sum_{C \in \mathcal{C}} \left(w(C) \cdot \mathbb{E}\left[C(X) \middle| X_1 = x_1, \ldots, X_i = x_i\right] \right) . \end{aligned} \tag{8}$$

Speciálisan

$$\mathbb{E}[w(X, \mathcal{C})] \overset{(8)\&(7)}{=} \sum_{C \in \mathcal{C}} \left(w(C) \cdot \left(1 - 2^{\mathrm{hossz}(C)}\right) \right) \ge (1/2) \sum_{C \in \mathcal{C}} w(C) \ . \tag{9}$$

állítás 7.1. *Legyen X az a véletlen értékadás, mely a változókhoz egymástól függetlenül 1/2 valószínűséggel igaz illetve hamis értéket rendel. Ekkor az (1/2)-MaxSAT algoritmus a MaxSAT probléma tetszőleges $(\mathcal{C}, w)$ példányához polinom időben megkonstruál egy olyan x értékadást, mely a $\mathcal{C}$-beli klózok legalább $\mathbb{E}[w(X, \mathcal{C})] \ge (1/2) \cdot \sum_{C \in \mathcal{C}} w(C)$ összsúlyú részhalmazát elégíti ki, azaz legalább feleakkorát, amekkora maximálisan lehetséges.*

Algorithm 3 (1/2)-MaxSAT$(\mathcal{C}, w)$

Require: $(\mathcal{C}, w)$ a MaxSAT probléma egy példánya

for $i = 1, \ldots, n$ **do** { Feltesszük: $\mathcal{C}$ a $\{v_1, \ldots, v_n\}$ változók halmaza felett értelmezett}
  if $w_{x_1,\ldots,x_{i-1},0}(\mathcal{C}) \geq w_{x_1,\ldots,x_{i-1},1}(\mathcal{C})$ **then**
    $x_i = 0$
  else
    $x_i = 1$
  end if
end for
return $x = (x_1, \ldots, x_n)$

Bizonyítás: (7) illetve (8) miatt az algoritmus futásideje polinomiális az input méretében. Ezen felül a

$$w_{x_1,\ldots,x_i}(\mathcal{C}) = (1/2) \cdot w_{x_1,\ldots,x_i,0}(X, \mathcal{C}) + (1/2) \cdot w_{x_1,\ldots,x_i,1}(\mathcal{C})$$

összefüggés illetve az algoritmus konstrukciója következtében

$$\frac{1}{2} \sum_{C \in \mathcal{C}} w(C) \overset{(9)}{\leq} \mathbb{E}[w(X, \mathcal{C})] \leq w_{x_1}(\mathcal{C}) \leq \cdots \leq w_{x_1,\ldots,x_n}(\mathcal{C}) = w(x, \mathcal{C}) \ .$$

$\square$

megjegyzés 7.2. *Az az n mélységű teljes bináris fa, melynek gyökerének címkéje $\mathbb{E}[w(X, C)]$, és melyben egy $w_{x_1,\ldots,x_i}(\mathcal{C})$ címkéjű belső csúcs két fiának a címkéje $w_{x_1,\ldots,x_i,0}(\mathcal{C})$ illetve $w_{x_1,\ldots,x_i,1}(\mathcal{C})$ a $(\mathcal{C}, w)$* önreducibilitási fája. *Az* (1/2)-MaxSAT *algoritmus egy ezen alapuló* derandomizációs *módszer, a* feltételes várható érték módszere, *mely az X véletlen értékadás alapján konstruál egy x (determinisztikus) értékadást, mely legalább olyan jó, mint amilyen X átlagosan.*

8 Egészértékű programozási feladat és LP-relaxáció

Legyen $(\mathcal{C}, w)$ a MaxSAT probléma egy tetszőleges példánya. Jelölje $C \in \mathcal{C}$ klóz esetén S_C^+ a klózban negálatlanul szereplő szereplő változók indexét, S_C^- pedig a negálva szereplőkét. A $(\mathcal{C}, w)$ példányhoz tartozó *egészértékű programozási feladat*, illetve *LP-relaxáció* az 1. ábrán látható módon definiálható. (Az előbbire az angol elnevezésből adódóan szokás röviden *ILP*-ként hivatkozni.) Könnyen látható, hogy az ILP lehetséges megoldásai megegyeznek az eredeti feladat lehetséges megoldásaival, és hogy mindkét feladatnak ugyanazon pontokban van az optimuma. A két probléma tehát teljesen ekvivalens, és így a nehézségük is ugyanolyan. Ezzel szemben az LP-relaxáció hatékonyan megoldható, viszont nincs meg az előbb említett szoros kapcsolat a másik két problémával. A kérdés tehát, hogy milyen módon tudunk az utóbbi egy optimális megoldásából az előbbi kettő problémához egy lehetséges megoldást konstruálni, és hogy az mennyivel lesz rosszabb, mint ezeknek az optimuma.

ILP

$$\begin{aligned} \sum_{i\in S_C^+} y_i + \sum_{i\in S_C^-}(1-y_i) &\geq z_C, && C\in\mathcal{C} \\ z_C &\in \{0,1\}, && C\in\mathcal{C} \\ y_i &\in \{0,1\}, && i=1,\dots,n \end{aligned}$$

$$\sum_{C\in\mathcal{C}} w(C) \to \max$$

LP relaxáció

$$\begin{aligned} \sum_{i\in S_C^+} y_i + \sum_{i\in S_C^-}(1-y_i) &\geq z_C, && C\in\mathcal{C} \\ z_C &\in [0,1], && C\in\mathcal{C} \\ y_i &\in [0,1], && i=1,\dots,n \end{aligned}$$

$$\sum_{C\in\mathcal{C}} w(C) \to \max$$

Figure 1: ILP és LP-relaxáció a $(\mathcal{C}, w)$ MaxSAT példányhoz

8.1 $(1-(1/e))$-közelítés MaxSAT problémára

lemma 8.1. *Legyen $(\mathcal{C}, w)$ a MaxSAT probléma egy tetszőleges példánya, legyen (y^*, z^*) a hozzá tartozó LP-relaxáció egy optimális megoldása, és legyen $Y = (Y_1, \dots, Y_n)$ az a véletlen értékadás, ahol*

$$\mathbb{P}[Y_i = b] = \begin{cases} y_i^*, & \text{ha } b = 1 \\ 1 - y_i^* & \text{ha } b = 0 \end{cases}$$

minden $i = 1, \dots, n$ esetén, egymástól függetlenül. Ekkor $\mathcal{C}$ tetszőleges C klózára $\mathbb{P}[C(Y) = 1] \geq \beta_k \cdot z_C^*$, ahol k a C hossza és $\beta_k = 1 - (1-(1/k))^k$.

Bizonyítás: Az általánosság megszorítása nélkül feltehető, hogy $S_C^- = \emptyset$. A számtani-mértani közepek közötti összefüggés alapján, és mert (y^*, z^*) lehetséges megoldása az LP-relaxációnak,

$$\mathbb{P}[C(Y)=1] = 1 - \prod_{i\in S_C^+}(1-y_i^*) \geq 1 - \left(\frac{1}{k}\sum_{i\in S_C^+}(1-y_i^*)\right)^k = 1 - \left(1 - \sum_{i\in S_C^+}\frac{y_i^*}{k}\right)^k \geq g(z_C^*) \ ,$$

ahol $g(z) = 1 - (1 - z/k)^k$. A g függvény második deriváltja nempozitív a $[0, 1]$ intervallumon, azaz ezen az intervallumon g konkáv, és így $g(z) \geq (1 - z) \cdot g(0) + z \cdot g(1) = z \cdot \beta_k$. Következésképpen a fenti egyenlőtlenség jobb oldala legalább $z_C^* \cdot \beta_k$. □

állítás 8.2. *Az 8.1 lemma állításában használt Y véletlen értékadás szintén derandomizálható a feltételes várható érték módszer segítségével. Az ezzel a módszerrel nyert y értékadásra teljesül, hogy $w(y, \mathcal{C}) \geq \mathbb{E}[w(Y, \mathcal{C}]$.*

8.2 (3/4)-es közelítés a MaxSAT problémához

lemma 8.3. *Legyen $(\mathcal{C}, w)$ a MaxSAT probléma egy tetszőleges példánya, (y^*, z^*) pedig a hozzá tartozó LP-relaxáció egy optimális megoldása. Legyen ezen felül X a 7.1 állításban szereplő, Y pedig a 8.1 lemmában megadott véletlen értékadás. Ekkor bármely $C \in \mathcal{C}$ klóz esetén $(1/2) \cdot \big(\mathbb{E}[w(X, \mathcal{C})] + \mathbb{E}[w(Y, \mathcal{C})]\big) \geq (3/4) \cdot w(C) \cdot z_C^*$.*

Bizonyítás: Legyen B egy szabályos érmedobás, és legyen $T = X$, ha B fej volt, illetve $T = Y$, ha írás. Ekkor nyilván $(1/2) \cdot \mathbb{E}[w(X, C)] + (1/2) \cdot \mathbb{E}[w(Y, C)] = \mathbb{P}[C(T) = 1]$. Ezen felül bármely $C \in \mathcal{C}$ klóz esetén a 8.1 lemma és a (7) egyenlőtlenség következtében

$$\mathbb{P}[C(T) = 1] = \frac{\mathbb{P}[C(X) = 1] + \mathbb{P}[C(Y) = 1]}{2} \geq \frac{\alpha_{\text{hossz}(C)} + z_C^* \cdot \beta_{\text{hossz}(C)}}{2} \geq z_C^* \cdot \frac{\alpha_{\text{hossz}(C)} + \beta_{\text{hossz}(C)}}{2} ,$$

hiszen $z_C^* \leq 1$. Márpedig $\alpha_1 + \beta_1 = \alpha_2 + \beta_2 = 3/4$, és minden $k \geq 3$ esetén $\alpha_k + \beta_k \geq (7/8) + (1 - (1/e)) \geq 3/2$. □

következmény 8.4. *Legyen $(\mathcal{C}, w)$ a MaxSAT probléma egy tetszőleges példánya, és legyen x illetve y a 7.1 illetve 8.2 állításokban szereplő polinomiális futásidejű algoritmusok outputja a $(\mathcal{C}, w)$ inputon. Ekkor x és y közül legalább az egyik a $\mathcal{C}$-beli klózoknak legalább $(3/4)$-szer akkora összsúlyú részhalmazát elégíti ki, mint amekkora maximálisan lehetséges.*

Bizonyítás: Az állítás könnyen adódik a

$$\begin{aligned} \max\big\{w(x, \mathcal{C}),\ w(y, \mathcal{C})\big\} &\geq \max\big\{\, \mathbb{E}[w(X, \mathcal{C})],\ \mathbb{E}[w(Y, \mathcal{C})]\big\} && \text{(lásd a 7.1 és 8.2 állításokat)} \\ &\geq (1/2) \cdot \big(\mathbb{E}[w(X, \mathcal{C})] + \mathbb{E}[w(Y, \mathcal{C})]\big) \\ &\geq (3/4) \cdot \sum_{C \in \mathcal{C}} w(C) \cdot z_C^* && \text{(lásd a 8.3 lemmát)} \end{aligned}$$

összefüggésből, ahol (y^*, z^*) a $(\mathcal{C}, w)$ LP-relaxációjának egy optimális megoldása. □

megjegyzés 8.5. *Korábbi eredményeket megjavítva Håstad [5] megmutatta, hogy amennyiben* $\mathbf{P} \neq \mathbf{NP}$*, akkor bármely $\epsilon \in (0, 1)$ esetén egy hatékony algoritmus nem képes tetszőleges φ 3-CNF formulához mindig olyan értékadást konstruálni, mely által kilégített φ-beli klózok száma az otpimálisénak legalább $(7/8 - \epsilon)$-szorosa.*

9 Vektoriális programozás

9.1 MaxCut

Egy $G = (V, E)$ gráfban $V_1, V_2 \subseteq V$ esetén (V_1, V_2) egy *vágás*, ha V a V_1 és V_2 diszkrét uniója. Az ezen *vágáshoz tartalmazó élek* alatt az $E \cap (V_1 \times V_2)$ élhalmazt értjük, azaz azon E-beli élek halmazát, melyek V_1-beli csúcsokat kötnek össze V_2-beliekkel. A (V_1, V_2) *vágás súlya* az E-n értelmezett $w : E \to \mathbb{Q}^+$ *súlyfüggvény* mellett a hozzá tartozó élek összsúlya, $\sum_{e \in (E \cap (V_1 \times V_2))} w(e)$.

MaxCut	
Input:	egy (G, w) pár, ahol • $G = (V, E)$ egy gráf, • $w : E \to \mathbb{Q}^+$ pedig egy *súlyfüggvény*.
Output:	Egy (V_1, V_2) vágás G-ben.
Cél:	$\sum_{e \in (E \cap (V_1 \times V_2))} w(e) \to \max$.

megjegyzés 9.1. *A* MaxCut *problémához természetes módon konstruálható egy mohó algoritmus, mely egy tetszőleges* (G, w) *input esetén a* G *gráf csúcsait úgy osztja két részre, hogy a köztük menő élek összsúlya legalább* $(1/2) \cdot \sum_{e \in E(G)} w(e)$ *— és így legalább fele a maximálisan elérhetőnek is.*

9.2 Goemans-Williamson algoritmus: $0,87856$-közelítés a MaxCut problémához

A 2. ábrán látható egy egészértékű programozási feladat a problémához. Fontos, hogy ennek minden egyenlete másodfokú, lineáris tagok nélkül. Az ilyen feladatokat hívjuk *szigorúan kvadratikus programozási feladatnak*. Ezek fontosságát adja, hogy automatikusan ún. *vektoriális programozási feladat* formába írhatók át, melyek olyan optimalizálási feladatok, ahol az n változó mind n-dimenziós valós vektor, a célfüggvény ezek belsőszorzatainak valamilyen lineáris függvénye, a feltételek pedig szintén ezen belsőszorzatokon értelmezett lineáris egyenlőtlenségek. Egy vektoriális programozási feladat pedig automatikusan átírható szemidefinit programozási feladattá, ami pedig már hatékonyan megoldható az ellipszoid algoritmus segítségével (erről bővebben a 13 illetve a 11 szekcióban) A MaxCut problémához tartozó vektoriális programozási feladat szintén a 2. ábrán látható.

egészértékű programozási feladat	vektoriális programozási relaxáció
$x_v^2 = 1, \quad v \in V(G)$ $x_v \in \{-1, 1\}, \quad v \in V(G)$ $\frac{1}{2} \cdot \sum_{e=(u,v) \in E(G)} w(e) \cdot (1 - x_u x_v) \quad \to \max$	$\langle y_v, y_v \rangle = 1, \quad v \in V(G)$ $y_v \in \mathbb{R}^n, \quad v \in V(G)$ $\frac{1}{2} \cdot \sum_{e=(u,v) \in E(G)} w(e) \cdot (1 - \langle y_u, y_v \rangle) \quad \to \max$

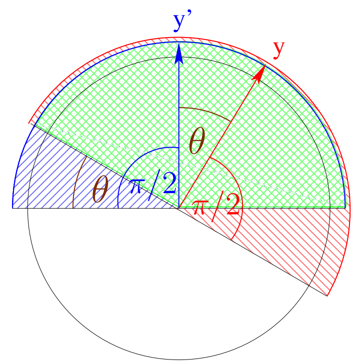
Figure 2: ILP és vektoriális programozási relaxáció a (G, w) MaxCut példányhoz. $(n = |V(G)|.)$

lemma 9.2. *Jelölje $\mathcal{D}$ az $\{x \in \mathbb{R}^n : \|x\| = 1\}$ halmaz feletti az egyenletes eloszlást, és legyen $X \sim \mathcal{D}$. Ekkor tetszőleges $y, y' \in \mathbb{R}^n$ vektor valamint az általuk bezárt $\theta = \arccos(\langle y, y'\rangle/(\|y\| \cdot \|y'\|))$ szög esetén $\mathbb{P}\left[\langle y, X\rangle \cdot \langle y', X\rangle < 0\right] = \theta/\pi$ — azaz az y és y' által bezárt szög $(1/\pi)$-szerese megegyezik annak a valószínűségével, hogy $\langle y, X\rangle$ és $\langle y', X\rangle$ azonos előjelű.*

Bizonyítás: A bizonyítás alapötlete a 3. ábrán látható.

Amennyiben az X vektornak az y és y' vektorok által meghatározott síkra vett merőleges vetülete a pirossal jelölt ív valamelyik pontjának irányába mutat, akkor $\langle y, X\rangle \geq 0$, egyébként $\langle y, X\rangle < 0$. Hasonlóképpen, ha a kékkel jelölt ív valamelyik pontjának irányába mutat, akkor $\langle y', X\rangle \geq 0$, egyébként $\langle y, X\rangle < 0$. Következésképpen $\langle y, X\rangle \cdot \langle y, X\rangle$ pontosan akkor negatív, ha X fenti vetülete vagy olyan pont irányába mutat, mely rajta van a piros íven, de nincs rajta a kéken, vagy pedig olyanéba, mely rajta van a kék íven, de nincs rajta a piroson. Márpedig ezen ívek összhossza 2θ, és így $\mathbb{P}\left[\langle y, X\rangle \cdot \langle y', X\rangle < 0\right] = 2\theta/(2\pi)$. □

Figure 3: A 9.2 lemma bizonyításának alapötlete



Legyen (G, w) egy tetszőleges példánya a MaxCut problémának, és jelölje y^* a hozzá tartozó vektoriális programozási relaxáció egy optimális megoldását. Ekkor a $V_x^+ = \{v \in V(G) : 0 \leq \langle y_v^*, x\rangle\}$ és $V_x^- = \{v \in V(G) : 0 > \langle y_v^*, x\rangle\}$ halmazok nyilvánvalóan a G csúcshalmazának egy partícionálását adják, azaz (V_x^+, V_x^-) vágás lesz a G gráfban. Jelölje $W(x)$ az ezen vágáshoz tartozó élek összsúlyát. Mint azt alább megmutatjuk, az x vektort a fenti lemmában megadott véletlen eloszlás szerint választva teljesülni fog, hogy az így kapott véletlen vágáshoz tartozó élek összsúlya várható értékben az maximálisnak legalább $\alpha_{\text{G-W}}$-szerese, ahol

$$\alpha_{\text{G-W}} = \frac{2}{\pi} \cdot \min_{\theta \in [0,\pi]} \frac{\theta}{1 - \cos\theta} \geq 0,87856 \ .$$

lemma 9.3. *Legyen (G, w) a MaxCut probléma egy tetszőleges példánya, és válasszunk egy X véletlen vektort a $\{x \in \mathbb{R}^n : \|x\| = 1\}$ halmazból egyenletes valószínűséggel. Ekkor a $W(X)$ vágás súlyának várható értéke legalább $\alpha_{\text{G-W}}$-szer akkora, mint a G-beli vágások súlyának maximuma.*

Bizonyítás: $\alpha_{\text{G-W}}$ definíciója miatt minden $\theta \in [0, \pi]$ értékre

$$\alpha_{\text{G-W}} \leq \frac{2}{\pi} \cdot \frac{\theta}{1 - \cos\theta} \ . \tag{10}$$

Jelölje most G élhalmazát E, illetve $e = (u, v) \in E$ esetén $\theta_e = \arccos\langle y_u^*, y_v^*\rangle$ az y_u^* és y_v^* által bezárt szöget (mind y_u^* mind y_v^* egységvektor, hiszen y^* megoldása a vektoriális programozási relaxációnak). Ekkor

$$\begin{aligned}
\mathbb{E}[W(X)] &= \sum_{e=(u,v)\in E} \Big(w(e) \cdot \mathbb{P}[\langle y_u^*, X\rangle \cdot \langle y_v^*, X\rangle \geq 0] \Big) && \text{(lásd ($W(X)$ definícióját)} \\
&= \sum_{e\in E} \left(w(e) \cdot \frac{\theta_e}{\pi} \right) && \text{(lásd a 9.2 lemmát)} \\
&\geq \alpha_{\text{G-W}} \cdot \frac{1}{2} \cdot \sum_{e\in E} \big(w(e) \cdot (1 - \cos\theta_e) \big) && \text{(lásd a (10) egyenlőtlenséget)} \\
&= \alpha_{\text{G-W}} \cdot \frac{1}{2} \cdot \sum_{e=(u,v)\in E} \big(w(e) \cdot (1 - \langle y_u^*, y_v^* \rangle) \big) \ .
\end{aligned}$$

A fenti egyenlőtlenség jobb oldala pedig a vektoriális programozási relaxáció optimumával egyenlő, ami nyilván legalább akkora, mint az eredeti feladat optimuma. □

Algorithm 4 GOEMANS-WILLIAMSON-ALGORITMUS(G, w, ϵ)

Require: (G, w) a MAXCUT probléma egy példánya

1: $c = \lceil ((1/\alpha_{\text{G-W}}) - 1 + \epsilon)/\epsilon \rceil$
2: **for** $i = 1, \ldots, c$ **do**
3: Generáljuk az X^i vektort a 9.2 lemmában leírtaknak megfelelően
4: {az $X^1, \ldots, X^{i-1}$ vektoroktól függetlenül}
5: **end for**
6: Válasszunk egy $\hat{X}$ vektort az $\{X^1, \ldots, X^c\}$ halmazból úgy, hogy $W(\hat{X}) = \max_{i=1,\ldots,c} W(X^i)$
7: **return** $\left(V_{\hat{X}}^+, V_{\hat{X}}^-\right)$

tétel 9.4. *Tetszőleges $\epsilon > 0$, továbbá a* MAXCUT *probléma egy tetszőleges (G, w) példánya esetén* GOEMANS-WILLIAMSON-ALGORITMUS(G, w, ϵ) *legalább $(1-(1/e))$ valószínűséggel G-nek egy olyan vágását állítja elő, melynek súlya a maximálisan elérhetőnek legalább $0,87856 \cdot (1-\epsilon)$-szorosa. Ezen felül az algoritmus futásideje (az input méretében és $1/\epsilon$-ban) polinomiális.*

Bizonyítás: A vektoriális programozási feladatok — ahogy azt a bevezetésük után említettük — hatékonyan megoldhatók, így a futásidővel kapcsolatos állítás igazolásához elég megmutatni, hogy a 9.2 lemmában megadott véletlen változó generálása hatékonyan megoldható. Ehhez először is azt kell észrevenni, hogy (standard) normális eloszlású változók független véletlen bitek segítségével hatékonyan közelíthetők (a centrális határeloszlás tétele következtében). Márpedig ha $Y_1, \ldots, Y_n$ független standard normálisok, akkor $(Y_1, \ldots, Y_n)$ n-dimenziós standard normális, aminek az eloszlása gömbszimmetrikus, így azt egységnyi hosszúságúra normálva a kívánt eloszláshoz jutunk.

Az approximációra vonatkozó állítás igazolásához először is legyen X az $X^1, \ldots, X^c$ vektorokkal azonos eloszlású változó, jelölje $\hat{W} = \sum_{e\in E(G)} w(e)$ és legyen $b = (1-\epsilon)\cdot\mathbb{E}[W(X)]$. Ekkor $\mathbb{E}[W(X)] \leq \mathbb{P}[W(X) < b] \cdot b + \mathbb{P}[W(X) \geq b] \cdot \hat{W}$, azaz, bevezetve a $p = \mathbb{P}[W(X) \geq b]$ jelölést, $\mathbb{E}[W(X)] \leq (1-p) \cdot b + p \cdot \hat{W}$. Ebből pedig egyszerű átalakítással kapjuk, hogy

$$p \geq \frac{\mathbb{E}[W(X)] - b}{\hat{W} - b} = \frac{\mathbb{E}[W(X)] - (1-\epsilon)\cdot\mathbb{E}[W(X)]}{\hat{W} - (1-\epsilon)\cdot\mathbb{E}[W(X)]} = \frac{\epsilon}{\hat{W}/\,\mathbb{E}[W(X)] - (1-\epsilon)}.$$

Következésképpen, felhasználva a 9.3 lemma eredményét, $p \geq \epsilon/((1/\alpha_{\text{G-W}}) - 1 + \epsilon)$. Így pedig legalább $1-(1-p)^c \geq 1-(1-(1/c))^c \geq 1-(1/e)$ valószínűséggel az algoritmusban előállított c darab vágás lesz olyan, aminek a súlya legalább b. Ez, megint csak a 9.3 lemma szerint, igazolja az állítást. □

megjegyzés 9.5. *A fenti módszert Goemans és Williamson publikálta [4], majd később Mahajan és Ramesh derandomizálta [10].*

Korábbi eredményeket megjavítva Håstad [5] megmutatta, hogy amennyiben $\mathbf{P} \neq \mathbf{NP}$*, akkor bármely* $\epsilon \in (0,1)$ *esetén egy hatékony algoritmus nem képes tetszőleges gráfhoz mindig olyan vágást konstruálni, melynek súlya az otpimálisénak legalább* $(16/17 - \epsilon)$*-szorosa. Végül pedig Khot és szerzőtársai igazolták, hogy a Goemans-Williamson algoritmus optimális (azaz hogy semmilyen* $\epsilon > 0$ *konstanshoz nem létezik olyan algoritmus, amely tetszőleges inputhoz hatékonyan előállít egy olyan vágást, melynek súlya a maximálisan elérhetőnek legalább* $(\alpha_{G\text{-}W} + \epsilon)$*-szorosa), feltéve, hogy teljesül a Unique Games sejtés [8].*

10 Primál-duál séma

komplementaritási tétel

10.1 Egy üzemelhelyezési probléma

10.2 Minimális multivágás

10.3 2-közelítés a Minimális multivágás problémára fák esetén

Algorithm 5 MinMultiVág-Fa$(G = (V, E),\ w : V \to \mathbb{Q},\ P \subseteq V^2\})$

Require: G gyökereztetett fa; egy $v \in V$ *csúcs mélysége* a gyökértől való (súlyozatlan!) távolsága; egy $e \in E$ *él mélysége* a két végpontja mélységének a minimuma

$y_p := 0$ minden $p \in P$ párra;
legyen $v_1, v_2, \ldots, v_n$ a G csúcsai mélység szerint *nem* növekvő sorrendben
for $i = 1, 2, \ldots, n$ **do**
  for minden olyan $p = (s, t) \in P$ párra, melynek legalacsonyabb közös őse v_i **do**
    $y_p :=$ az aktuálisan választható legmagasabb érték
    $E_i :=$ aktuálisan telítetté vált élek halmaza
  end for
end for
$E^+ := \bigcup_{i=1}^n E_i$ $\{E^+$ megoldása a primál feladatnak!$\}$
$E^- := \emptyset$ $\{E^+$-ből kidobandó élek gyűjtőhalmaza$\}$
for $i = n, (n-1), \ldots, 1$ **do**
  for $e \in E_i$ mélység szerint nem csökkenő sorrendben **do**
    if $E^+ \setminus (E^- \cup \{e\})$ lehetséges megoldása a primál feladatnak **then**
      $E^- := E^- \cup \{e\}$
    end if
  end for
end for
return $(E' := (E^+ \setminus E^-))$

lemma 10.1. *Legyen $p = (s, t) \in P$ tetszőleges, és jelölje j az s és t legalacsonyabb közös ősének indexét. Ekkor $y_p > 0$ esetén mind a v-ből s-be menő útnak, mind a v_j-ből t-be menő útnak csak legfeljebb egy éle szerepelhet az E' halmazban.*

Bizonyítás: s és t szimmetriája miatt az állítást elég csak a v_j-ből az s-be menő útra bebizonyítani.

Tekintsük a v_j és s között vezető út azon éleit, melyek elemei E'-nek (ha ilyen nincs, akkor a lemma állítása automatikusan teljesül), és legyen e ezek közül a legmagasabban fekvő. Megmutatjuk, hogy minden *nála alacsonyabban* fekvő olyan E^+-beli él, mely szintén rajta van az s és v_j között vezető úton, nem lehet benne E'-ben.

Ha $e' \in E^+$ a p egy e-nél alacsonyabban fekvő éle, akkor van olyan $k' \in \{1, 2, \ldots, n\}$ index, hogy $e' \in E_{k'}$. Sőt, $y_p > 0$ miatt az is teljesül, hogy $j \leq k'$. Azt állítjuk, hogy az algoritmus második for ciklusának $i = k'$ iterációjában e' bekerül E^--ba. (Ebből pedig a lemma állítása már automatikusan következik.)

Valóban. Legyen ugyanis $p' = (s', t') \in P$ egy olyan út, amin rajta fekszik e', és jelölje j' az s' és t' legalacsonyabb közös ősének indexét. Ekkor $j' \geq j$ esetén e szükségszerűen rajta van a p' úton — márpedig a kiinduló feltevésünk szerint e sosem kerülhet E^--be. $j' < j$ esetén viszont lennie kell egy olyan $0 \leq j'' \leq j'$ (és így $j'' < k'$) indexnek, hogy $E_{j''}$ tartalmaz p'-beli élet.Bárhogy is, a második for ciklus $i = k'$ iterációjában p'-nek van e'-től különböző éle az $E^+ \setminus (E^- \cup E_{k'})$ halmazban, és így e' bekerül E^--ba. $\square$

Legyen most $x_e = 1$ minden E'-beli e élre, és $x_e = 0$ minden egyéb e él esetén. Ekkor

$$\sum_{p\in P} y_p \overset{10.1\text{lemma}}{\geq} \sum_{p\in P}\left(\left(\frac{1}{2}\sum_{e\in p} x_e\right)\cdot y_p\right) \geq \frac{1}{2}\sum_{e\in E}\left(x_e\cdot\left(\sum_{p\in P:\ e\in p} y_p\right)\right) \geq \frac{1}{2}\sum_{e\in E}(x_e\cdot w(e)) \quad .$$

megjegyzés 10.2. *Tekintve, hogy (amint azt Garg, Yannakakis és Vazirani is kifejtette [3]) a* Csúcslefedés *probléma bármely példánya hatékonyan visszavezethető a* MinMultiVág *olyan példányára, melynek gráfja fa, az alfejezetben tekintett probléma valóban NP-teljes.*

11 Ellipszoid algoritmus

Vizsgálatainkat megszorítjuk arra az esetre, amikor $\hat{P}$, a lehetséges megoldások halmaza, valamely $S \subseteq \{0,1\}^n$ halmaz konvex burka (például valamely megfelelő $\{0,1\}^n$ feletti LP feladat relaxációjával van dolgunk). **Feltesszük továbbá, hogy $\hat{P}$ nem elfajuló**, azaz hogy térfogata pozitív.

- Kielégíthetőség és optimalizálás:
  Ha a $\mathbf{c}^\top \mathbf{x}$ célfüggvény minimumát keressük a $\hat{P}$ konvex halmazon (feltehető, hogy $\mathbf{c} \in \mathbb{Z}$: szorozzunk fel a nevezők legkisebb közos többszörösével), vizsgáljuk meg a $P_d := \hat{P} \cap \{\mathbf{x} : \mathbf{c}^\top \mathbf{x} \leq d + 1/2\}$ halmazokat. P_d nyilván pontosan akkor nem-üres, ha S-nek van olyan eleme, melynek célfüggvényértéke legfeljebb d. Tekintve, hogy $S \subseteq \{0,1\}^n$, elég $-\sum_{i=1}^n |c_i|$ és $\sum_i = 1^n |c_i|$ közötti értékeket vizsgálni, azaz bináris keresést alkalmazva $O(\log(\sum_{i=1}^n |c_i|))$ d érték megvizsgálásával megtalaláljuk a keresett optimum értékét.

- Kezdő ellipszoid:
  Az $(1/2, \ldots, 1/2)^\top$ középpontú, $\sqrt{n}/2$ sugarú n-dimenziós gömb tartalmazza az egész $\{0,1\}^n$ halmazt, így jó választás E_0 kezdő ellipszoidnak. Ezen gömb térfogata az egység sugarú n-dimenziós gömb (jelölésben B_n) térfogatának $(\sqrt{n}/2)^n$-szerese, azaz

  $$\mathrm{Vol}(E_0) = (\sqrt{n}/2)^n \mathrm{Vol}(B_n) \leq (\sqrt{n}/2)^n \pi^{n/2}.$$

  Tehát $\log \mathrm{Vol}(E_0) = O(n \log n)$.

- Korlát a szükséges iterációk számára:
  Tegyük fel, hogy $\{\mathbf{x} : \mathbf{c}^\top \mathbf{x} \leq d\} \cap S \neq \emptyset$, mondjuk $\mathbf{v}_0 \in \{\mathbf{x} : \mathbf{c}^\top \mathbf{x} \leq d\} \cap S \subseteq P_d \cap S$; megmutatjuk, hogy ekkor P_d térfogata sem lehet túl kicsi. Azon feltételünkből, hogy $\hat{P}$ nem elfajuló, következik, hogy van olyan $\mathbf{v}_1, \ldots, \mathbf{v}_n \in S$, hogy $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_n$ konvex burka nem elfajuló (azaz pozitív térfogata van). Az persze elképzelhető, hogy $\mathbf{v}_1, \ldots, \mathbf{v}_n$ a P_d-nek már nem lesz eleme, de — bevezetve az $\alpha = 1/(2nc)$ konstanst, ahol c a $\mathbf{c}$ legnagyobb komponense — a

  $$\mathbf{w}_i := \begin{cases} \mathbf{v}_i, & \text{ha } \mathbf{c}^\top \mathbf{v}_i \leq d + 1/2 \\ \mathbf{v}_0 + \alpha(\mathbf{v}_i - \mathbf{v}_0) & \text{egyébként} \end{cases} \qquad i = 1, \ldots, n$$

  vektorok már P_d-ben lesznek, hiszen $\mathbf{c}^\top \mathbf{v}_i > d + 1/2$ esetén

  $$\mathbf{c}^\top \mathbf{w}_i = \mathbf{c}^\top \mathbf{v}_0 + \alpha \mathbf{c}^\top (\mathbf{v}_i - \mathbf{v}_0) \leq d + \alpha n c = d + 1/2.$$

  Tehát P_d tartalmazza a $\mathbf{v}_0$ és a $\mathbf{w}_1, \ldots, \mathbf{w}_n$ vektorokat, és így ezek C konvex burkát is, melynek térfogata, $\mathrm{Vol}(C)$ a $\mathbf{w}_i - \mathbf{v}_0$ vektorok által meghatározott parallelepipedon térfogatának $1/(n!)$-szorosa. Így

  $$\mathrm{Vol}(P_d) \geq \mathrm{Vol}(C) = \frac{1}{n!} |\det(U)|,$$

  ahol U az a mátrix, melynek i-edik oszlopa $\mathbf{w}_i - \mathbf{v}_0$, és így $|\det(U)|$ pont a $\mathbf{w}_i - \mathbf{v}_0$ vektorok által meghatározott parallelepipedon térfogatát adja. Azt is tudjuk azonban, hogy a $\mathbf{v}_i$ vektorok komponensei egészek, és így a $\mathbf{w}_i$ és a $\mathbf{w}_i - \mathbf{v}_0$ vektorok komponensei is vagy egészek, vagy olyan törtek, melynek nevezője osztója $2nc$-nek. Így ha U minden sorát $2nc$-vel beszorozzuk (mellyel a determináns értékét pontosan a $(2nc)^n$-szeresére növeltük), egy $\mathbb{Z}^{n \times n}$-beli mátrixot kapunk, melynek determinánsa is ily módon egész lesz, méghozzá (mivel sorai lineárisan függetlenek), nem nulla egész — azaz ezen mátrix determinánsának abszolútértéke legalább egy. Következésképpen $|\det(U)| \geq 1/(2nc)^n$, és így[1] $|\log \mathrm{Vol}(P_d)| = O(n \log(cn))$.

Az algoritmus felépítése és a fentiek alapján a következőket tudjuk:

- $a_k \in P \Rightarrow$ az algoritmus terminál,
- $a_k \notin P \Rightarrow \mathrm{Vol}(E_{k+1}) \leq e^{-1/(2n-2)}\mathrm{Vol}(E_k)$,
- $a_k \notin P \Rightarrow P \subseteq E_{k+1}$,
- $\log \mathrm{Vol}(E_0) = O(n \log n)$,
- $|\log \mathrm{Vol}(P)| = O(n \log n)$.

Legyen $k_0 := \lceil 2(n+1) \ln \big(\mathrm{Vol}(E_0)/\mathrm{Vol}(P)\big) \rceil + 1$. A fentiek alapján, ha az ellipszoid algoritmus legalább k iteráción át fut, akkor $a_i \notin P$, $i = 1, \dots, k$, ami miatt $P \subseteq E_i$, $i = 1, \dots, k$, következésképpen $\mathrm{Vol}(P) \leq \mathrm{Vol}(E_k)$ — akkor pedig $k < k_0$. Az ellipszoid algoritmus tehát legfeljebb $k_0 = O(n^2 \log n)$ lépés után megáll.

[1] Felhasználva, hogy a Stirling formula szerint $n! \sim (n/e)^n \sqrt{2\pi n}$.

12 LP alapú approximációs módszerek

A kurzuson a következő módszerekkel ismerkedtünk meg a címben említettek közül:

- A relaxált LP feladat megoldásán alapuló valószínűségelméleti meggondolások derandomizálása:
  1. írjuk fel az adott feladathoz tartozó egész értékű programozási feladatot
  2. oldjuk meg a relaxált feladatot $\to \mathbf{x}^*$
  3. (esetleg $\mathbf{x}^*$-ot felhasználó) valószínűségelméleti megfontolásokkal vizsgáljuk meg, milyen összefüggés adható az eredeti egész értékű programozási feladat optimuma, illetve a relaxált feladat optimuma között
  4. a fenti eredmény igazolásához alkalmazott okfejtéseket és meggondolásokat alapul véve (esetleg viszonylag kevés elem végigpróbálgatásával) keressünk egy olyan *konkrét* egész értékű megoldást, melynek létezését az előző pontban igazoltuk (derandomizálás)

- Primál-duál séma (Boole-változók esetén[2]):
  Írjuk fel a duális kölcsönös feszességi kritériumok valamely megfelelő α-relaxált változatát, valamint a primális kölcsönös feszességi kritériumokat. (Esetleg fordítva.) A duális feladat egy tetszőleges $\mathbf{y}$ lehetséges megoldásához a primális feltételek alapján a a következő $\mathbf{x}$ tartozna:

  $$x_i := \begin{cases} 1, & \text{ha az } i\text{-edik primális kritériumot figyelembe véve egengedett } x_i \neq 0, \\ 0, & \text{egyébként} \end{cases}$$

  Ez a duális feladatnak az algoritmus elején választott $\mathbf{y}$ lehetséges megoldása esetén (mely gyakran maga a $\mathbf{0}$ vektor) várhatólag nem lesz lehetséges megoldása a primális feladatnak. A következő fázis feladata, hogy iteratíven egyre újabb $\mathbf{y}$ vektorokat generálva azok optimalitása (a duális feladat tekintetében) egyre jobb legyen, míg a hozzájuk tartozó $\mathbf{x}$ vektorok egyre inkább teljesítsék a primál feladat feltételeit. Az iteráció akkor ér véget, amikor találunk egy olyan $\mathbf{y}$ vektort, melyhez tartozó $\mathbf{x}$ is már lehetséges megoldás. Amennyiben ezen $\mathbf{x}$ (esetleg egy-két komponensének lenullázása után) az aktuális $\mathbf{y}$ vektorral teljesíti a relaxált duális kölcsönös feszességi kritériumokat is (a primálisokat $\mathbf{x}$ konstrukciója miatt mindig automatikusan teljesítik), akkor (a relaxált kölcsönös feszességi kritériumokra vonatkozó tétel alapján) $\mathbf{x}$ a primális, $\mathbf{y}$ pedig a duális feladat egy α-közelítő megoldása lesz. Ráadásul — szintén a konstrukcióból adódóan — $\mathbf{x}$ egész értékű, azaz nemcsak a relaxált primál feladatnak lesz lehetséges megoldása, hanem az eredeti egész értékű programozási feladatnak is.

[2]Azaz az x_i értékek vagy 0-k vagy 1-ek

13 Szemidefinit programozás

13.1 Szemidefinit és vektoriális programozás kapcsolata

13.2 Szemidefinit programozási feladat megoldása az ellipszoid algoritmussal

14 Belső pontos módszerek

14.1 Ye algoritmusa

References

[1] P. Crescenzi, R. Silvestri, and L. Trevisan. On weighted vs unweighted versions of combinatorial optimization problems. *Information and Computation*, 167(1):10–26, 2001.

[2] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[3] N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997.

[4] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.

[5] J. Håstad. Some optimal inapproximability results. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, pages 1–10, New York, NY, USA, 1997. ACM.

[6] R. Impagliazzo and R. Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[7] S. Khot. On the power of unique 2-prover 1-round games. In *IEEE Conference on Computational Complexity*, page 25, 2002.

[8] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, 2007.

[9] S. Khot and O. Regev. Vertex cover might be hard to approximate to within $2 - \varepsilon$. *Computational Complexity, Annual IEEE Conference on*, 0:379, 2003.

[10] S. Mahajan and H. Ramesh. Derandomizing semidefinite programming based approximation algorithms. In *FOCS '95: Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, page 162, Washington, DC, USA, 1995. IEEE Computer Society.

[11] V. V. Vazirani. *Approximation algorithms.* Springer-Verlag New York, Inc., New York, NY, USA, 2001.