1η-2η Σειρά Ασκήσεων

Ειρήνη Χρυσικοπούλου-3180208

17-2-2022

1η Άσκηση

Πρέπει να αποδείξουμε ότι το πρόβλημα ανήκει στην κλάση $\mathcal{NP}$ και στην συνέχεια να αποδείξουμε ότι όλα τα υπόλοιπα προβλήματα της κλάσης $\mathcal{NP}$ ανάγονται σε αυτό.
Θεωρούμε την εκδοχή του προβλήματος απόφασης.Δεδομένου ενός συνόλου παικτών P,και ενός συνόλου αντικειμένων M,κάθε παίκτης πλειοδοτεί για την απόκτηση ενός υποσυνόλου αντικειμένων $M_p \subseteq M$ με μία αξία $u_p \geq 0$.Με δεδομένο επίσης έναν ακέραιο k μπορεί ο δημοπράτης να κατανείμει τα αντικείμενα σε ένα υποσύνολο παικτών $W \subseteq P$,έτσι ώστε τα πλειοδοτούμενα υποσύνολα αντικειμένων των παικτών του W να είναι ξένα μεταξύ τους και η συνολική αξία των παικτών του W να είναι τουλάχιστον k;

Αρχικά το πρόβλημα ανήκει στην κλάση $\mathcal{NP}$.Δεδομένης μια υποψίας λύσης,δηλαδή μιας κατανομής των αντικειμένων του συνόλου M σε ένα σύνολο παικτών$W \subseteq M$,μπορούμε σε πολυωνιμικό χρόνο $O(W)$ να ελέξουμε αν αυτή η συνολική αξία των παικτών $\geq k$

Για να δείξουμε ότι το πρόβλημα είναι $\mathcal{NP}$-complete αρκεί να δείξουμε ότι ένα οποιοδήποτε $\mathcal{NP}$-complete πρόβλημα ανάγεται σε αυτό σε πολυωνιμικό χρόνο.Θα δείξουμε ότι το Weighted Independent Set ανάγεται σε πολυωνιμικό χρόνο στο πρόβλημα μας.

Έστω ένα στιγμιότυπο του προβλήματος Weighted Independent Set,δηλαδή ένα γράφημα $G = (V, E)$,μια συνάρτηση $w : V- > \mathbf{R}$ και ένας ακέραιος B.Θα κατασκευάσουμε ένα στιγμιότυπο για το πρόβλημα μας,έστω A, έτσι ώστε το στιγμιότυπο του Weighted Independent Set να είναι ένα NAI στιγμιότυπο αν και μόνο αν το στιγμιότυπο του προβλήματος A που θα κατασκευάσουμε είναι NAI στιγμιότυπο.
Κατασκευάζουμε λοιπόν ένα στιγμιότυπο για το πρόβλημα A ως εξής:
Θέτουμε σαν σύνολο παικτών P το σύνολο V των κόμβων του γραφήματος G. Σαν σύνολο αντικειμένων M θέτουμε το σύνολο E των ακμών του γραφήματος G και σαν συνολική αξία το B. Λύνουμε επομένως το πρόβλημα A με παραμέτρους V, E, B.

Αν υπάρχει ανεξάρτητο σύνολο με βάρος B τότε υπάρχει κατανομή των αντικειμένων σε ένα υποσύνολο παικτών με αξία B:
Έστω ότι το S είναι ένα ανεξάρτητο σύνολο του G με βάρος $\geq B$.Τότε αν θεωρήσουμε κάθε κόμβο του S ως παίκτη βλέπουμε ότι κάθε αντικείμενο δίνεται σε έναν μόνο παίκτη(καθώς στο S δεν υπάρχουν κόμβοι που να είναι γειτονικοί,επομένως δεν υπάρχουν παίκτες που τους δίνεται ένα αντικείμενο από κοινού) και ότι η συνολική αξία qB.Επομένως βρήκαμε μια κατανομή των αντικειμένων σε υποσύνολο παικτών,στην οποία κάθε παίρνει το σύνολο των αντικειμένων που πλειοδοτεί.

Αν υπάρχει κατανομή των αντικειμένων σε ένα υποσύνολο παικτών με αξία B τότε υπάρχει ανεξάρτητο σύνολο με βάρος B: Εάν υπάρχει κατανομή αντικειμένων με ένα υποσύνολο παικτών W με αξία B τότε κάθε ένα από τα πλειοδοτούμενα αντικείμενα έχει ανατεθεί σε ένα μόνο παίκτη.Επομένως για το στιγμιότυπο του WIS,αν ένας κόμβος ανήκει στο σύνολο W τότε δεν ανήκει σε αυτό κάποιος

γειτονικός του κόμβος. Προφανώς το συνολικό βάρος των κόμβων που ανήκουν στο Independent Set $\geq B$.
Η αναγωγή έγινε σε πολυωνιμικό χρόνο.

2η Άσκηση

(i)

Από το Division Theorem γνωρίζουμε ότι για κάθε ζεύγος ακεραίων x,n υπάρχουν μοναδικά k και r τέτοια ώστε: x=kn+r
Θέτοντας στην παραπάνω σχέση $x = b^c$ και $n = p - 1$ έχουμε:
$b^c = (p-1)k + r$,επομένως $a^{b^c} = a^{k(p-1)}a^r$
Επίσης $a^{k(p-1)} = a^{(p-1)^k}$.Επειδή το p είναι πρώτος αριθμός από το θεώρημα του Fermat γνωρίζουμε ότι $a^{(p-1)} \equiv 1 \bmod p \iff a^{(p-1)^k} \equiv 1^k \equiv 1 \bmod p$.
Συνεπώς $a^{(p-1)^k}a^r \equiv a^r \bmod p \iff a^{b^c} \equiv a^r \bmod p$.
Επομένως υπολογίζουμε πρώτα το $r = b^c \bmod (p-1)$ και στην συνέχεια το $a^r \bmod p$.Και οι δύο υπολογισμοί γίνονται σε πολυωνιμικό χρόνο από τον ακόλουθο αλγόριθμο.

Algorithm 1 $Exponentiation(x, y, N)$

Input: Integers x, N of n bits and integer exponent y
Output:$x^y \bmod N$

```
if y == 0 then
    return 1
end if
z ← Exponentiation(x, y/2, N)
if y is even then:
    return z^2 mod N
else
    return x * z^2 mod N
```

(ii)

Χωρίς βλάβη της γενικότητα υποθέτουμε ότι τα $N_{i,i=1\ldots3}$ είναι μεταξύ τους πρώτοι.
Η Αλίκη παράγει και στέλνει στον $i-$οστό φίλο της το ciphertext C_i
$C_1 = M^3 \bmod N_1 \iff M^3 \equiv C_1 \bmod N_1$
$C_2 = M^3 \bmod N_2 \iff M^3 \equiv C_2 \bmod N_2$
$C_3 = M^3 \bmod N_3 \iff M^3 \equiv C_3 \bmod N_3$
Το σύστημα έχει μοναδική λύση $\bmod N, N = N_1N_2N_3$
Έστω C αυτή η λύση.Τότε $C \equiv M^3 \bmod N_1N_2N_3 \iff M = \sqrt[3]{C}$ Η τελευταία ισοδυναμία ισχύει επειδή $M < N_i$ επομένως $M^3 < N_1N_2N_3$.

3η Άσκηση

Θα λύσουμε το πρόβλημα με δυναμικό προγραμματισμό.

Έστω η ακολουθία $S = S_j \dots S_l, S_i \in B$ που αντιστοιχίζεται με το χαμηλότερο δυνατό κόστος με την ακολουθία εισόδου Γ.Έστω το t τέτοιο ώστε το $\Gamma[t]$ ταιριάζει με κάποιο γράμμα μιας ακολουθίας S_l.Τότε η ακολουθία με το ελάχιστο κόστος αντιστοίχισης για την $\Gamma[t:n]$ είναι η S_l.

Έστω $c(x,y)$ το ελάχιστο κόστος αντιστοίχισης της $\Gamma[x\ y]$ με οποιοδήποτε ακολουθία $S_i \in B$

Κατασκευάζουμε πίνακα OPT με n θέσεις,τέτοιον ώστε $OPT[i] =$χαμηλότερο κόστος αντιστοίχησης της $\Gamma[1\ i]$ με κάποια ακολουθία της βάσης B.
Για να συμπληρώσουμε τον πίνακα OPT εξετάζουμε ολά τα $t < i$ αναζητώντας ένα t τέτοιο ώστε το κόστος αντιστοίχισης της $\Gamma[t\, i]$ με κάποια ακολουθία S_i στο B συν το κόστος αντιστοίχισης της ακολουθίας $\Gamma[1\ t-1]$ που απομένει να είναι το ελάχιστο δυνατόν.
Το κόστός αντιστοίχισης της $\Gamma[t\ i]$ με κάποια ακολουθία S_i στο B είναι $c(x,y)$ και το κόστος αντιστοίχισης της υπόλοιπης ακολουθίας $\Gamma[1\,t-1]$ είναι $OPT[i-1]$.

Έχουμε επομένως την παρακάτω αναδρομική σχέση:
$OPT[i] = \min_{t<i}\{c(t,i)\} + OPT[t-1] \quad j \geq 1$ με $OPT[0] = 0$

Για να βρούμε το ελάχιστο $c(x,y)$ για κάθε δυνατό ζεύγος τιμών υπολογίζουμε το κόστος αντιστοιχισης κάθε ακολουθίας της βάσης B με την ακολουθία $\Gamma[x:y]$ και ως κρατάμε το ελάχιστο από αυτά.
Αποθηκεύουμε τα κόστη αυτά καθώς και την ακολουθία που κάθε φορά αντιστιχοίζεται βέλτιστη με την την ακολουθία $\Gamma[x:y]$ σε έναν πίνακα $C[i,j]$,όπου στην θέση $[i,j]$ αποθηκεύεται το ελάχιστο κόστος αντιστοίχισης της $\Gamma[x:y]$ με κάποια ακολουθία S_iτης B,καθώς και η ίδια η ακολουθία S_i

Για να βρούμε την ακολουθία $S = S_j \dots S_l, S_i \in B$ ξεκινάμε από το τέλος του πίνακα OPT,και βρίσκουμε με πια ακολουθία έχει αντιστοιχηθεί βέλτιστα με ποιο τμήμα της λέξης μέχρι να φτάσουμε στην αρχή της.

Algorithm 2

Input: String Γ and strings $S_i \in B$
Output:least cost alignment of Γ with string $S = Si_, \ldots, Sl, S_j \in B$

$OPT[0] = 0$
Compute C
for $i = 1$ *to* n **do**
  $min = \infty$
  for $t = 1$ *to* i **do**
    if $OPT[t-1] + C[i,t] < min$ **then**
      $min = OPT[t-1] + C[i,t]$
    end if
  end for
  $OPT[i] = min$
end for
$t = n$
while $t! = 1$ **do**
  $i = i$ *such that* $OPT[t-1] + C[i,t]$ *is minimum*
  Print $C[i,t]$
  $t = i - 1$
end while
return $OPT[n]$ =0

4η Άσκηση

Έστω OPT το Minimum Steiner Tree, με βάρος $c(OPT)$.
Διπλασιάζουμε τις ακμές πάνω στο OPT και βρίσκουμε μια Euler διαδρομή πάνω σε αυτές τις ακμές(προφανώς υπάρχει τέτοια διαδρομή γιατί οι κορυφές μετά τον διπλασισμό των ακμών είναι άρτιου βαθμού).
Η διαδρομή αυτή, έστω W, έχει κόστος $c(W) = 2c(OPT)$,καθώς περνάμε από κάθε κόμβο ακριβώς δύο φορές.
Επειδή ισχύει η τριγωνική ανισότητα μπορούμε να κάνουμε shortcuts πάνω στην διαδρομή W,χωρίς να χειροτερέψει η λύση μας.
Έτσι λαμβάνουμε έναν Hamiltonian κύκλο πάνω σε όλους του κόμβους του OPT (δηλαδή όλους του κόμβους του V' και ίσως κάποιους από τους υπόλοιπους κόμβους του V)
Διαγράφοντας μια ακμή από τον κύκλο έχουμε ένα δέντρο T',με κόστος $c(T') < c(W)$.Προφανώς το T' είναι ένα Spanning Tree πάνω στους κόμβους που ανήκουν στο OPT.
Έστω T το Minimum Spanning Tree πάνω στους κόμβους του V',με κόστος $c(T)$.
Προφανώς $c(T) \leq c(T')$ καθώς το T' περιέχει τους κόμβους του V' που περιέχει και το T και ίσως κάποιους παραπάνω.Επομένως:
$c(T) \leq c(T') < c(W) = 2c(OPT)$.
Άρα $\frac{c(T)}{c(OPT)} \leq 2$.

5η Άσκηση

Έστω ότι η βέλτιστη λύση έχει συνολικό βάρος OPT.Αυτό σημαίνει ότι χρησιμοποιώντας k σύνολα,το μέγιστο συνολικό άθροισμα βαρών που μπορούμε να έχουμε είναι OPT.Επομένως υπάρχει ένα σύνολο στο S,έστω S_i που τα στοιχεία του έχουν άθροισμα $\geq OPT/k$
Έστω ότι ο άπληστος αλγόριθμος μας δίνει λύση SOL.Όπως και στην βέλτιστη λύση έχουν χρησιμοποιηθεί και εδώ k σύνολα από το S,επομένως υπάρχει σύνολο στο S,έστω S_j που τα στοιχεία του έχουν άθροισμα $\geq SOL/k$
Αφού η OPT είναι η βέλτιστη λύση θα ισχύει ότι $OPT \geq SOL \iff$
$OPT/k \geq SOL/k (1)$
Ο άπληστος αλγόριθμος επιλέγει το "καλύτερο σύνολο",δηλαδή το σύνολο με τον μεγαλύτερο λόγο αθροίσματος βαρών ακάλυπτων στοιχείων /πλήθος ακάλυπτων στοιχείων. Λόγω της (1) θα επιλέξει το σύνολο S_i με άθροισμα στοιχείων $\geq OPT/k$.
Επομένως μετά την πρώτη επανάληψη θα έχει καλυφθεί βάρος $\geq OPT/k$ και θα μένει να καλυφθεί βάρος $OPT - \frac{OPT}{k} = \frac{OPT(k-1)}{k}$
Επειδή το βάρος αυτό καλύπτεται από την βέλτιστη λύση υπάρχει σύνολο στο S που καλύπτει τουλάχιστον $\frac{\frac{OPT(k-1)}{k}}{k} = \frac{OPT(k-1)}{k^2}$ στοιχεία.Επομένως ο αλγόριθμος θα προσθέσει στο επόμενο βήμα το σύνολο αυτό στην λύση αφήνοντας ακάλυπτο βάρος $\frac{OPT(k-1)}{k} - \frac{OPT(k-1)}{k^2} = \frac{OPT(k-1)^2}{k^2}$

Παρατηρούμε ότι στην i-οστή επανάληψη μένει ακάλυπτο βάρος $\frac{OPT(k-1)^i}{k^i} = OPT(\frac{k-1}{k})^i = OPT(1-\frac{1}{k})^i$.
Επομένως μετα από k επαναλήψεις ο αλγόριθμος επιστρέφει την λύση SOL,δηλαδή το βάρος που κάλυψε χρησιμοποιώντας k σύνολα απο το S
SOL=$OPT - OPT(\frac{k-1}{k})^k = OPT - OPT(1-\frac{1}{k})^k$.

Όμως $OPT(1-\frac{1}{k})^k < OPT(e^{-1/k})^k = OPTe^{-1}$
$\iff -OPT(1-\frac{1}{k})^k > -OPT\frac{1}{e}$
$\iff OPT - OPT(1-\frac{1}{k})^k > OPT - OPT\frac{1}{e} \iff 1-(1-\frac{1}{k})^k > 1-\frac{1}{e}$

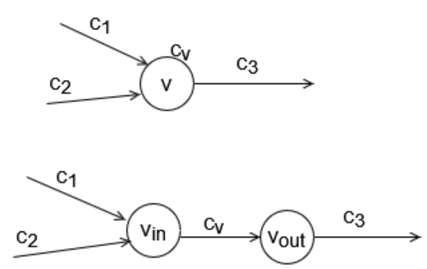
6η Άσκηση

(i)

Έστω ότι θέλουμε να μεγιστοποιήσουμε την ροή από ένα σύνολο κόμβων $V_1 \subseteq V$ προς ένα άλλο σύνολο κόμβων $V_2 \subseteq V$. Θα προσθέσουμε έναν επιπλέον κόμβο s που θα συνδέεται με κάθε κόμβο του V_1 με μια κατευθυνόμενη ακμή $e_i = (s, v_i), v_i \in V_1$ άπειρης χωρητικότητας.
Επιπλέον θα προσθέσουμε έναν κόμβο t που θα συνδέεται με κάθε κόμβο του V_2 με μια κατευθυνόμενη ακμή $e_j = (v_j, t), v_j \in V_2$ άπειρης χωρητικότητας.
Τώρα μπορούμε σε πολυωνιμικό χρόνο να βρούμε την μέγιστη $s - t$ ροή,με τον γνωστό αλγόριθμο.

(ii)

Αντικαθιστούμε κάθε κόμβο $v \in V$ με 2 νέους κόμβους v_{in} και v_{out}.Κάθε ακμή που ήταν εισερχόμενη στον v τώρα είναι εισερχόμενη στον v_{in} και κάθε ακμή που ήταν εξερχόμενη από τον v θα είνα εξερχόμενη από τον v_{out}.Δημιουργούμε μια κατευθυνόμενη ακμή από τον v_{in} προς τον v_{out} με χωριτικότητα όση η χωριτικότητα του κόμβου v.



Τώρα μπορούμε σε πολυωνιμικό χρόνο να βρούμε την μέγιστη ροή με τον γνωστό αλγόριθμο για εύρεση μέγιστης ροής.

(iii)

Η μέγιστη ροή σε ένα δίκτυο είναι η μέγιστη ροή που μπορεί να περάσει από τον κόμβο s προς του γείτονες του.
Επομένως η ποσότητα που θέλουμε να μεγιστοποίησουμε είναι η $\sum_{u:(s,u)\in V} f(s,u)$,με f συμβολίζουμε την ροή από τον αρχικό κόμβο $s \in V$ προς κάποιον γείτονα του $u \in V$.
Η ροή που εισέρεχεται σε κάθε κόμβο $u \in V - \{s, t\}$ πρέπει να ισόυται με την ροή που εξέρχεται από αυτόν.Επομένως έχουμε τον περιορισμό:
$\sum_{u:(u,v)\in V} f(u,v) = \sum_{v:(v,w)\in V} f(v,w), \forall v \in V - \{s,t\}$.
Τέλος πρέπει η ροή σε κάθε ακμή να μην υπερβαίνει την μέγιστη χωριτικότητα της,έστω c_e και να μην είναι μικρότερη από το κάτω φράγμα της,έστω l_e το

κάτω φράγμα της ροής που μπορεί να διέλυθει από την ακμή $e \in E$,δηλαδή $f(u,v) \leq c(u,v)$ και $f(u,v) \geq l(u,v)$,$\forall (u,v) \in E$. Με βάση αυτούς τους περιορισμούς έχουμε το ακόλουθο γραμμικό πρόγραμμα:

$$\begin{aligned}
\text{maximize} \sum_{(s,u)\in V} f(s,u) \\
\text{subject to} \\
\sum_{(u,v)\in V} f(u,v) = \sum_{(v,w)\in V} f(v,w), \quad \forall v \in V - \{s,t\} \\
f(u,v) \leq c(u,v), \qquad \forall (u,v) \in E \\
f(u,v) \geq l(u,v), \qquad \forall (u,v) \in E
\end{aligned} \tag{1}$$

(iv)

Εδώ θέλουμε να μεγιστοποίησουμε πάλι την ροή από τον αρχικό κόμβο προς τους γείτονες του,επομένως η αντικειμεική συνάρτηση είναι ίδια.Έχουμε όμως απώλεια ροής ϵ_u σε κάθε κόμβο $u \in V = \{s,t\}$.Άρα η εξέρχόμενη ροή στον κόμβο $u \in V$ ισούται με $\sum_{v:(v,u)\in V} f(u,v) = \sum_{w:(v,w)\in V} f(v,w) * (1 - \epsilon_u)$.
Επομένως έχουμε το ακόλουθο γραμμικό πρόγραμμα:

$$\begin{aligned}
\text{maximize} \sum_{(s,u)\in V} f(s,u) \\
\text{subject to} \\
\sum_{(v,u)\in V} f(v,u) = \sum_{(u,w)\in V} f(u,w) * (1 - \epsilon_v), \quad \forall v \in V - \{s,t\} \\
f(u,v) \leq c(u,v), \quad \forall (u,v) \in E
\end{aligned} \tag{2}$$

7η Άσκηση

i)

Ορίζουμε μια μεταβλητή x_i, ως εξής:

$$x_i = \begin{cases} 1 \text{ αν ο κόμβος } i \text{ ανήκει στο ανεξάρτητο σύνολο} \\ 0 \text{ αλλιώς} \end{cases}$$

Για κάθε ακμή $(i,j) \in E$ στο ανεξάρτητο σύνολο μπορεί να ανήκει το πολύ ένα από τα δύο άκρα της,δηλαδή είτε ο κόμβος i,είτε ο κόμβος j,είτε κανένας από αυτούς.
Επομένως για κάθε ακμή ακμή $(i,j) \in E$ ισχύει ότι $x_i + x_j \leq 1$.
Με βάση αυτά διαμορφώνουμε το ακόλουθο γραμμικό πρόγραμμα:

$$\begin{aligned} &\text{maximize} \sum_{i \in V} x_i \\ &\text{subject to} \\ &\qquad x_i + x_j \leq 1, \quad \forall (i,j) \in E \\ &\qquad x_i = \{0,1\} \quad i = 1, \dots, n \end{aligned} \tag{3}$$

Χαλάρωση του γραμμικού προβλήματος, οι μεταβλητές x_i,μπορούν να λάβουν πραγματικές τιμές,επομένως το πρόγραμμα γίνεται:

$$\begin{aligned} &\text{maximize} \sum_{i \in V} x_i \\ &\text{subject to} \\ &\qquad x_i + x_j \leq 1, \quad \forall (i,j) \in E \\ &\qquad x_i \in [0,1] \quad i = 1, \dots, n \end{aligned} \tag{4}$$

ia)

Έστω ότι έχουμε μια 3-κλίκα.Η βέλτιστη ακέραια λύση είναι η $OPT_{ILP} = 1$(το ανεξάρτητο σύνολο σε κλίκα αποτελείται από έναν μόνο κόμβο).
Η βέλτιστη λύση της γραμμικής χαλάρωσης είναι η $OPT_{LP} = 3/2$(αφού μπορούμε να πάρουμε μέρος ένος κόμβου στη βέλτιστη λύση θα ισχύει $x_i = 1/2$).
Έχουμε:$\frac{OPT_{ILP}}{OPT_{LP}} = \frac{2}{3} \leq \frac{2}{n}$,

ii)

(a)

Έστω ότι το $S(\pi)$ δεν είναι ανεξάρτητο σύνολο.Τότε υπάρχουν κόμβοι $i,j \in S(\pi)$ τέτοιοι ώστε $(i,j) \in E$,(δηλαδή τα i,j είναι γείτονες). Ισχύει ότι:
$i \in S(\pi) \iff$ κανένα γείτονας του i(άρα και ο j) δεν βρίσκεται πριν από τον i στην μετάθεση $\pi \iff$ ο j βρίσκεται μετά τον i.

$j \in S(\pi) \iff$ κανένας γείτονας του j(άρα και ο i) δεν βρίσκεται πριν από τον i στην μετάθεση $\pi \iff$ ο i βρίσκεται μετά τον j.
Καταλήξαμε σε άτοπο καθώς οι δύο παραπάνω προτάσεις δεν μπορούν να ισχύουν ταυτόχρονα.Επομένως το $S(\pi)$ είναι ανεξάρτητο σύνολο.

(b)

Ορίζουμε μια τυχαία μεταβλητή Q_i ως εξής:

$$X_i = \begin{cases} 1 \text{ αν η κορυφή } i \text{ ανήκει στο ανεξάρτητο σύνολο} \\ 0 \text{ αλλιώς} \end{cases}$$

Τότε: $E[X_i] = 1 * Pr[i \in Independent\ Set] + 0 * Pr[i\ not\ inIndependent\ Set]$
Η πιθανότητα να ανήκει μια κορυφή i στο ανεξάρτητο σύνολο είναι η πιθανότητα να εμφανίζεται πριν από τους γείτονες του στην μετάθεση π.Έστω d_i ο βαθμός της $i-$οστής κορυφής.Η πιθανότητα αυτή η κορυφή να εμφανιστεί πριν απο τους γείτονές της στην μετάθεση είναι $\frac{1}{d_i+1}$
Επομένως:

$E[|S(\pi)|] = E[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} \frac{1}{d_i+1}$

8η Άσκηση

1

Έστω οι μεταβλητές x_i, y_j:

$$x_i = \begin{cases} 1 \text{ αν ο ηθοποιός } i \text{ συμμετέχει στην ταινία} \\ 0 \text{ αλλιώς} \end{cases}$$

$$y_j = \begin{cases} 1 \text{ αν ο επενδυτής } j \text{ επενδύσει στην ταινία} \\ 0 \text{ αλλιώς} \end{cases}$$

Κάποιος ηθοποιός μπορεί να συμπεριλαμβάνεται στα σύνολα L_k, L_j δύο διαφορετικών επενδυτών k, j.Επομένως θα ισχύει $y_j \leq x_i \forall i \in L_j$.Η ισότητα ισχύει μόνο αν στην ταινία επενδύσει μόνο ένας επενδυτής που θέλει στην διανομή της ταινίας έναν συγκεκριμένο ηθοποιό.Επομένως έχουμε το ακόλουθο ακέραιο πρόγραμμα:

$$\begin{aligned} &\text{maximize} \sum_{j=1}^{n} y_j p_j - \sum_{i=1}^{m} x_i s_i \\ &\text{subject to} \\ &\qquad y_j \leq x_i, \quad \forall i \in L_j, j = 1, \ldots, n \\ &\qquad x_i \in \{0, 1\} \quad i = 1, \ldots, m \\ &\qquad y_j \in \{0, 1\} \quad j = 1, \ldots, n \end{aligned} \tag{5}$$

2

Στην χαλάρωση του γραμμικού προγράμματος οι μεταβλητές x_i, y_j μπορούν να λάβουν πραγματικές τιμές,επομένως το πρόγραμμα είναι το ακόλουθο:

$$\begin{aligned} &\text{maximize} \sum_{j=1}^{n} y_j p_j - \sum_{i=1}^{m} x_i s_i \\ &\text{subject to} \\ &\qquad y_j \leq x_i, \quad \forall i \in L_j, j = 1, \ldots, n \\ &\qquad x_i \in [0, 1] \quad i = 1, \ldots, m \\ &\qquad y_j \in [0, 1] \quad j = 1, \ldots, n \end{aligned} \tag{6}$$

Για να δείξουμε ότι η βέλτιστη λύση που επιστρέφεται από την λύση της χαλάρωσης του γραμμικού προβλήματος είναι ακέραια,αρκεί να ανάγουμε το πρόβλημα στο πρόβλημα εύρεσης μέγιστης ροής.

Κατασκευάζουμε διμερές γράφημα ως εξής: Για κάθε επενδυτή j δημιουργούμε έναν κόμβο,και για κάθε ηθοποιό i δημιουργούμε επίσης έναν κόμβο.
Δημιουργούμε μια ακμή j, i ανάμεσα σε στον επενδυτή j και τον ηθοποιό i αν $i \in L_j$ και θέτουμε το κόστος της ίσο με άπειρο.
Προσθέτουμε και δύο κόμβους s, t,και προσθέτουμε τις ακμές (s, j) με κόστος

p_j,για κάθε κόμβο j και (i,t) ,για κάθε κόμβο i με κόστος u_i.
Βρίσκοντας την μέγιστη ροή στο παραπάνω γράφημα,βρίσκουμε την ελάχιστη τομή.Κάθε τομή μας δίνει μια διαμέριση των κόμβων σε δύο σύνολα A, B τέτοια ώστε $s \in A$ και $t \in B$ και αντιπροσωπεύει μια ανάθεση(ανάθεση προέρχεται από τους κόμβους που βρίσκονται στην ίδια συνεκτική συνιστώσα με το s) επομένως η ελάχιστη τομή είναι η ανάθεση ελάχιστου κόστους.

9η Άσκηση

i)

Έστω x_i το πλήθος νομισμάτων αξίας V_i και έστω E το συνολικό ποσό που πρέπει να καλύψουμε με αυτά τα νομίσματα.Έστω ότι έχουμε n διαφορετικές αξίες στα νομίσματα.Διαμορφώνουμε το ακόλουθο ακέραιο πρόγραμμα:

$$\begin{aligned} \text{minimize} & \sum_{i=1}^{n} x_i \\ \text{subject to} & \\ & \sum_{i=1}^{n} x_i V_i = E \\ & x_i \in \mathbb{Z}^+, \quad i = 1, \dots, n \end{aligned} \tag{7}$$

ii)

Θεωρούμε την χαλάρωση του γραμμικού προβλήματος:

$$\begin{aligned} \text{minimize} & \sum_{i=1}^{n} x_i \\ \text{subject to} & \\ & \sum_{i=1}^{n} x_i V_i = E \\ & x_i \in \mathbb{R}^+, \quad i = 1, \dots, n \end{aligned} \tag{8}$$

Έστω το ακόλουθο στιγμιότυπο του προβλήματος:
$E = 28$,$V = \{20, 10, 5, 1\}$
Η λύση του ακέραιου γραμμικού προβλήματος είναι$\{1, 0, 1, 3\}$,δηλαδή: ένα νόμισμα των 20 λεπτών,ένα νόμισμα των 5 λεπτών και τρία νομίσματα του ενός λεπτού.Συνολικά χρησιμοποιήθηκαν 5 νομίσματα.

Η λύση της χαλάρωσης του γραμμικού προγράμματος είναι:$1, 8/10, 0, 0$ δηλαδή: ένα νόμισμα των 20 λεπτών και 8/10 από το νόμισμα των 10 λεπτών.Συνολικά χρησιμοποιήθηκαν 1,8 κέρματα.Επομένως η βέλτιστη λύση της χαλάρωσης του γραμμικού προβλήματος δεν είναι πάντα ακέραια.

10η Άσκηση

i)

Έστω w_c το βάρος της clause c του τύπου C.Επίσης για κάθε clause c ορίζουμε ως P_c το σύνολο των literals που εμφανίζονται χωρίς άρνηση στην c και αντίστοιχα το σύνολο N_c ως το σύνολο των literals που εμφανίζονται με άρνηση στην c.Θεωρούμε μια απότιμηση y που αποδίδει τιμές αλήθειας στα literals κάθε clause (πχ $y_i = 1$ δηλώνει ότι $x_i = 1$.
Ορίζουμε την μεταβλητή z_c ως εξής:

$$z_c = \begin{cases} 1 \text{ αν η clause } c \text{ ικανοποιείται} \\ 0 \text{ αλλιώς} \end{cases}$$

Για να ικανοποιηθεί μια clause αρκεί ένα από τα literals να έχει την τιμή 1.
Θέλουμε να μεγιστοποιήσουμε το συνολικό βάρος των clauses που ικανοποιούνται,επομένως έχουμε το ακόλουθο γραμμικό πρόγραμμα:

$$\begin{aligned} &\text{maximize} \sum_{c \in C} w_c z_c \\ &\text{subject to} \\ &\qquad \sum_{i \in P_c} y_i + \sum_{i \in N_c} (1 - y_i) \geq z_c, \quad \forall c \in C \\ &\qquad z_c \in \{0, 1\}, \quad \forall c \in C \\ &\qquad y_i \in \{0, 1\}, \quad \forall i \in C \end{aligned} \tag{9}$$

Θεωρούμε την χαλάρωση του παραπάνω προβλήματος:

$$\begin{aligned} &\text{maximize} \sum_{c \in C} w_c z_c \\ &\text{subject to} \\ &\qquad \sum_{i \in P_c} y_i + \sum_{i \in N_c} (1 - y_i) \geq z_c, \quad \forall c \\ &\qquad z_c \in [0, 1], \quad \forall c \\ &\qquad y_i \in [0, 1], \quad \forall i \end{aligned} \tag{10}$$

Λύνουμε την χαλάρωση του γραμμικού προγράμματος και λαμβάνουμε μια βέλτιστη λύση $(y*, z*)$.Σε κάθε literal x_i αναθέτουμε την τιμή True με πιθανότητα y_i^* και επιστρέφουμε την αποτίμηση που προκύπτει,έστω τ.
Ορίζουμε την τυχαία μεταβλητή W ως το συνολικό βάρος των clauses που ικανοποιούνται και ως την W_c την τυχαία μεταβλητή που δηλώνει το βάρος που συνεισφέρει στην λύση η clause c,δηλαδή $W = \sum_{c \in C} W_c$.

Προφανώς $E[W_c] = w_c Pr[c \quad is \quad satisfied]$

Για $k \geq 1$,ορίζουμε: $\beta_k = 1 - (1 - \frac{1}{k})^k$.
Έστω $size(c)$ το μέγεθος της clause c. Αν $size(c) = k$ τότε χωρίς βλάβη της γενικότητας η c είναι της μορφής:$c = x_1 \vee \cdots \vee x_k$.
Η clause c δεν ικανοποιείται μόνο στην περίπτωση που όλα τα literals έχουν τιμή 0.Σε όλες τις άλλες περιπτώσεις ικανοποιείται,επομένως η πιθανότητα να ικανοποιείται η clause c είναι:
$1 - \prod_{i=1}^{k}(1 - y_i) \geq 1 - (\frac{\sum_{i=1}^{k}(1-y_i)}{k})^k = 1 - (1 - \frac{\sum_{i=1}^{k} y_i}{k})^k \geq 1 - (1 - \frac{z_c^*}{k})^k$

Ορίζουμε την συνάρτηση $g(z) = 1 - (1 - \frac{z}{k})^k$.
Η g είναι κοίλη και αύξουσα συνάρτηση με $g(0) = 0$ και $g(1) = \beta_k$.
Επομένως για $z \in [0, 1]$, $g(z) \geq \beta_k z$.
Άρα η πιθανότητα να ικανοποιηθεί η c είναι $Pr[c \quad is \quad satisfied] \geq \beta_k z_c^*$.
Αν όλες οι clauses έχουν μέγεθος το πολυ k,τότε:
$E[W] = \sum_{c \in C} E[W_c] \geq \beta_k \sum_{c \in C} w_c z_c^* = \beta_k OPT_f \geq \beta_k OPT$,όπου OPT_f η βέλτιστη λύση της χαλάρωσης του γραμμικού προγράμματος .
Επομένως για στιγμιότυπα του MAX_SAT με το πολύ k literals ανά clause ο αλγόριθμος είναι β_k προσεγγιστικός.
$\forall k \in \mathbb{Z}^+$ ισχύει $(1 - \frac{1}{k})^k < \frac{1}{e}$ Άρα ο αλγόριθμος έχει λόγο προσέγγισης $1 - \frac{1}{e}$.

ii)

Αν τρέξουμε τον αλγόριθμο των διαφανειών η αναμενόμενη τιμή του βάρους που συνεισφέρει μια clause ειναι $E[W_c] = w_c(1 - \frac{1}{2^k})$,$k$ το πλήθος literals στην clause c.
Για τον αλγόριθμο που περιγράψαμε πιο πάνω προκύπτει ότι η αναμενόμενη τιμή του βάρους που συνεισφέρει στο συνολικό βάρος η clause c μεγέθους k είναι $E[W_c] \geq \beta_\kappa z_c^* w_c$.
Για τον αλγόριθμο των διαφανειών ισχύει ότι $E[W_c] = (1 - \frac{1}{2^k})w_c \geq (1 - \frac{1}{2^k})w_c z_c^*$, επειδή $z_c^* \leq 1$. Επομένως αν τρέχουμε καθέναν από τους αλγορίθους με πιθανότητα $1/2$,η αναμενόμενη τιμή της W_c είναι:
$E[W_c] = \frac{1}{2}(E[W_c|b = 0]) + \frac{1}{2}(E[W_c|b = 1]) \geq \frac{(1-2^{-k}+\beta_k)w_c z_c^*}{2}$
Για $k = 1, 2$ ισχύει ότι $1 - 2^{-k} + \beta_\kappa = 3/2$,
και για $k \geq 3$ ισχύει: $1 - 2^{-k} + \beta_k = 7/8 + (1 - 1/e) \geq 3/2$.
Επομένως η αναμενόμενη τιμή του συνολικού βάρους W ισούται με:
$E[W] = \sum_{c \in C} E[W_c] \geq \frac{3}{4} \sum_{c \in C} w_c z_c^* \geq \frac{3}{4} OPT_f \geq \frac{3}{4} OPT$,όπου OPT_f η βέλτιστη λύση της γραμμικής χαλάρωσης και OPT η βέλτιστη λύση του ακέραιου προγράμματος.