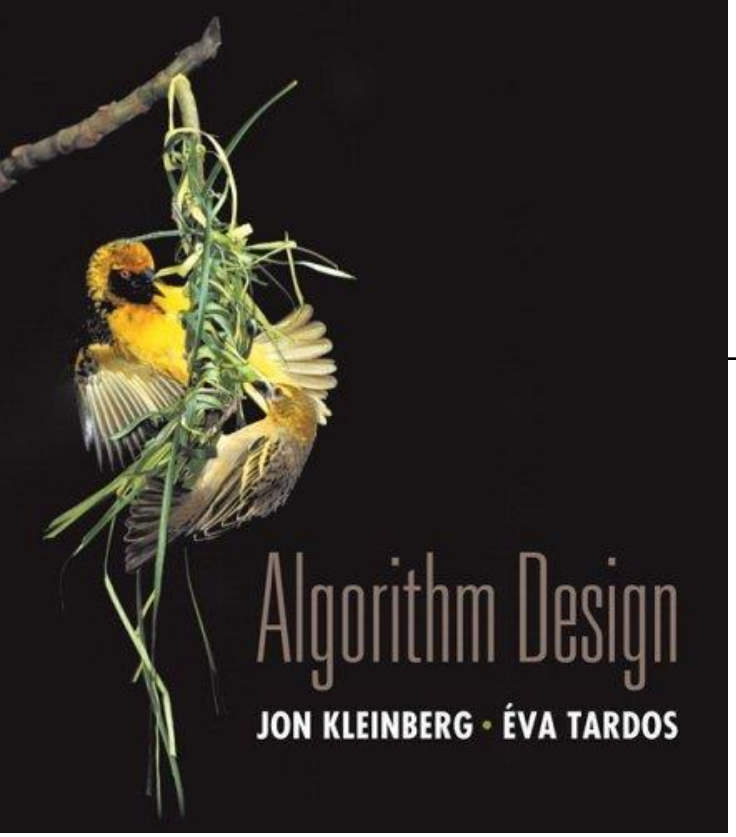


Ανάλυση Αλγορίθμων

Κεφ. 13: Τυχαίοι Αλγόριθμοι

Τυχαίοι Αλγόριθμοι

Ένας *τυχαίος αλγόριθμος* είναι ένας αλγόριθμος του οποίου η λειτουργία εξαρτάται από τυχαίους αριθμούς

Παράδειγμα τυχαίων αλγορίθμων:

- Quicksort
- Hashing

Άλλα παρόμοια θέματα:

- Η παραγωγή «τυχαίων» αριθμών
- Η παραγωγή τυχαίων δεδομένων για πειράματα ή άλλες χρήσεις
- Διάσπαση συμμετρίας

Ψευδοτυχαίοι Αλγορίθμοι

Ο Η/Υ *δεν μπορεί* να παράγει πραγματικά τυχαίους αριθμούς

- Ο Η/Υ μπορεί να παράγει ψευδοτυχαίους αριθμούς – αριθμούς που παράγονται από έναν τύπο
- Οι ψευδοτυχαίοι αριθμοί *φαίνονται* τυχαίοι αλλά είναι πλήρως προβλέψιμοι αν ξέρουμε τον τύπο
  - Οι ψευδοτυχαίοι αλγόριθμοι είναι επαρκείς για τις περισσότερες περιπτώσεις αλλά όχι για σοβαρές εφαρμογές (π.χ. ασφάλεια συστημάτων)
- Συσκευές παραγωγής πραγματικά τυχαίων αριθμών υπάρχουν
  - Για παράδειγμα μπορεί να βασίζονται σε ραδιενεργό διάσπαση

"Anyone who attempts to generate random numbers by deterministic means is, of course, living in a state of sin."

—John von Neumann

Παραγωγή Ψευδοτυχαίων Αριθμών

Ο καλύτερος τρόπος παραγωγής είναι από τη γραμμική βαθμιδωτή μέθοδο:

- $r = (a * r + b) \% m;$
  Όπου a και b είναι μεγάλοι πρώτοι αριθμοί και το m είναι 2^{32} ή 2^{64}
- Η αρχική τιμή του r καλείται *σπόρος* «seed»
- Αν ξεκινήσουμε την διαδικασία με τον ίδιο σπόρο θα πάρουμε την ίδια ακολουθία ψευδοτυχαίων αριθμών

Ένα πλεονέκτημα της μεθόδου είναι ότι τελικά θα παραχθούν όλοι οι δυνατοί αριθμοί

Σχεδόν όλες οι «βελτιώσεις» της μεθόδου οδηγούν σε χειρότερα αποτελέσματα

- Συνήθως έχουν αρκετά μικρότερους κύκλους

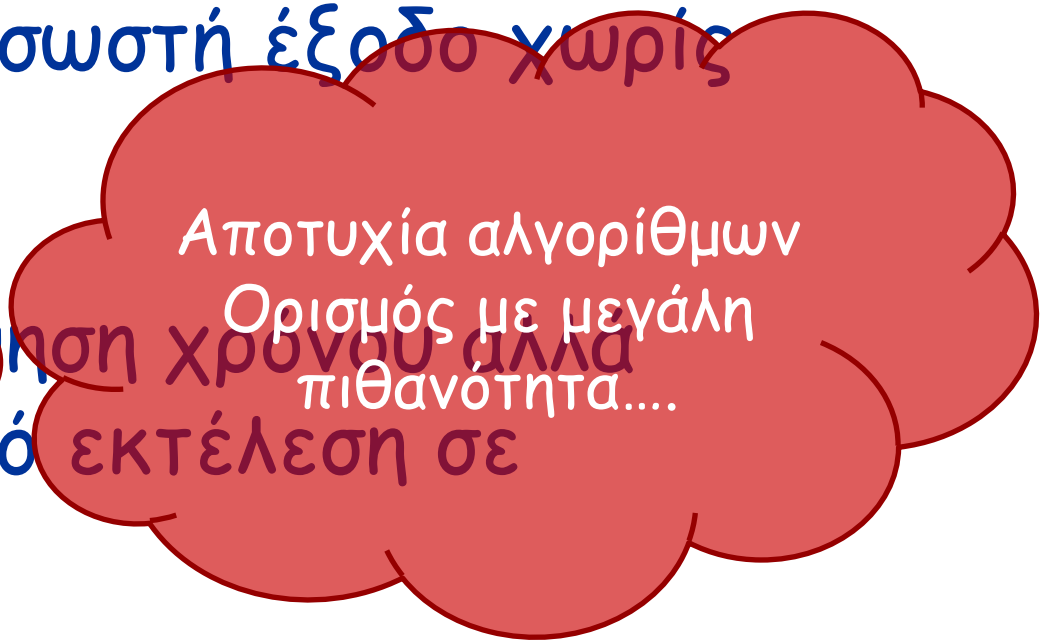
Υπάρχουν πολύπλοκοι μαθηματικοί αλγόριθμοι που ελέγχουν την «τυχαιότητα» μίας ακολουθίας αριθμών

Τύποι Τυχαίων Αλγορίθμων

Las Vegas (πάντα σωστή έξοδο χωρίς εγγύηση χρόνου)

Monte Carlo (εγγύηση χρόνου αλλά έξοδος αλλάζει από εκτέλεση σε εκτέλεση)

Sherwood (πάντα σωστή έξοδο και εγγύηση χρόνου αλλά ταυτόχρονα δεν έχουν τόσο ενδιαφέρον)



Επαναλαμβανόμενο Στοιχείο

Πίνακας με *n* στοιχεία, από τα οποία ένα επαναλαμβάνεται *n*/2 φορές (όλα τα άλλα είναι ανά δύο διαφορετικά).

Αλγόριθμος: Πάρε τυχαία ζευγάρια. Αν βρεις ίδια στοιχεία τότε τελειώνουμε.

Las Vegas

I was posed the following question in a technical interview for a Software Engineering position by a major multinational NASDAQ company: [Paraphrasing] "You are given an array of 1,000,000 32-bit integers. One int value x occurs 500,001 times or more in the array. Specify an algorithm to determine x." comp.theory 2009

Πλειοψηφικό Στοιχείο

Να βρεθεί αν υπάρχει πλειοψηφικό στοιχείο σε πίνακα *n* στοιχείων (>*n*/2 εμφανίσεις)

Επιλέγουμε τυχαία ένα στοιχείο και ελέγχουμε αν είναι πλειοψηφικό

Monte Carlo

```
     function maj
1.   i <-- random(1,n); x <-- A[i]; k <-- 0;
2.   for j <-- 1 to n do
3.      if A[j]=x then k <-- k+1
4.   return (k>n/2)
```

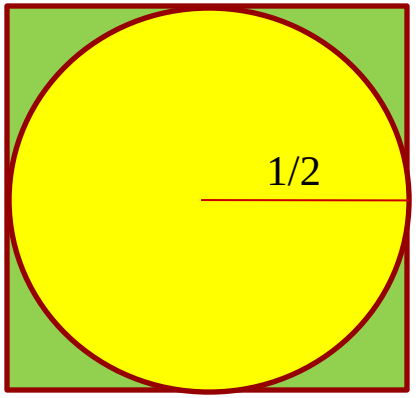
```
     function maj2
1.   if maj then return true
2.   else return maj
```

```
     function majMC
1.   k <-- log(1/e)
2.   for j <-- 1 to k do
3.      if maj then return true
4.   return false
```

Monte Carlo vs Las Vegas

- Ένας αλγόριθμος Las Vegas μπορεί πάντα να μετατραπεί σε Monte Carlo. Αντίστροφα;
- Μία διαδικασία για να βρούμε το π είναι η εξής:

1. Σε ένα μοναδιαίο τετράγωνο ζωγραφίζουμε ένα κύκλο ακτίνας 1/2.
2. Ρίχνουμε ομοιόμορφα αντικείμενα ίδιου μεγέθους στο τετράγωνο.
3. Μετράμε το πλήθος των αντικειμένων μέσα στον κύκλο, πολλαπλασιάζουμε με 4, και διαιρούμε με το πλήθος των αντικειμένων στο τετράγωνο.
4. Δίνει προσέγγιση στο π με βάση τις επιφάνειες του κύκλου και του τετραγώνου.



Monte Carlo vs Las Vegas

Δεν υπάρχει Las Vegas για αυτή τη διαδικασία (γιατί;)

Οι Monte Carlo αλγόριθμοι είναι εξαιρετικά πιο χρήσιμοι για δύσκολα προβλήματα στα οποία δεν «υπάρχει» Las Vegas αλγόριθμος

Θεωρία Υπολογισμού 6ο Εξάμηνο

Θα ήθελα αλλά ποτέ δεν προλαβαίνω

RP και ZPP

RP. [Monte Carlo] Προβλήματα απόφασης που επιλύονται με σφάλμα μίας πλευράς σε πολυωνυμικό χρόνο.

Μπορούμε να μειώσουμε τη πιθανότητα λάθους σε 2^{-100} κάνοντας 100 ανεξάρτητες επαναλήψεις

Σφάλμα μίας πλευράς.

- Αν η σωστή απάντηση είναι `OXI`, τότε πάντα επιστρέφει `OXI`.
- Αν η σωστή είναι `NAI`, επέστρεψε `NAI` με πιθανότητα $\geq \frac{1}{2}$.

ZPP. [Las Vegas] Προβλήματα απόφασης επιλύσιμα σε αναμενόμενο πολυωνυμικό χρόνο.

Ο χρόνος μπορεί να είναι άπειρος αλλά αναμένεται ότι θα είναι μικρός

Θεώρημα: $P \subseteq ZPP \subseteq RP \subseteq NP$.

Ανοικτά Προβλήματα: Σε τι βαθμό βοηθά η τυχαιότητα; Ισχύει P = ZPP; Ισχύει ZPP = RP; Ισχύει RP = NP;

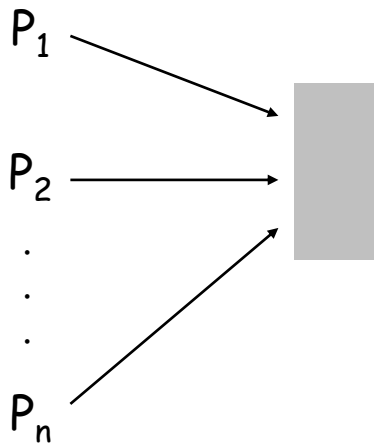
13.1 Επίλυση Ανταγωνισμού

Επίλυση Ανταγωνισμού σε Κατανεμημένο Σύστημα

Επίλυση Ανταγωνισμού: Δοθέντων n διεργασιών P_1, …, P_n, όπου κάθε μία ανταγωνίζεται με τις άλλες για έναν κοινό πόρο. Αν δύο διεργασίες προσπελαύνουν τον πόρο ταυτόχρονα τότε όλες οι διεργασίες πετάγονται έξω. Βρείτε πρωτόκολλο που να εγγυάται ότι όλες οι διεργασίες θα προχωρήσουν.

Περιορισμός: Οι διεργασίες δεν επικοινωνούν.

Πρόκληση: Διάσπαση συμμετρίας.



Τυχαιοποιημένο Πρωτόκολλο

Κάθε διεργασία ζητά προσπέλαση στον πόρο τη χρονική στιγμή t με πιθανότητα p = 1/n.

Ισχυρισμός. Έστω S[i, t] = το γεγονός ότι η διεργασία *i* προσπελαύνει τον πόρο στο χρόνο t. Τότε $1/(e \cdot n) \leq \Pr[S(i, t)] \leq 1/(2n)$.

Απόδειξη. Από ανεξαρτησία $\Pr[S(i, t)] = p\,(1-p)^{n-1}$.

η διεργασία i απαιτεί προσπέλαση

καμία από τις υπόλοιπες n-1 διεργασίες δεν απαιτεί προσπέλαση

- Θέτοντας p = 1/n, έχουμε $\Pr[S(i, t)] = 1/n\,(1 - 1/n)^{n-1}$. ▪

Τιμή που μεγιστοποιεί Pr[S(i, t)]

μεταξύ 1/e και 1/2

Deathbed Formulas. Καθώς το *n* αυξάνει η συνάρτηση:

- $(1 - 1/n)^n$ συγκλίνει μονότονα από 1/4 μέχρι 1/e
- $(1 - 1/n)^{n-1}$ συγκλίνει μονότονα από το 1/2 προς το 1/e.

Τυχαιοποιημένο Πρωτόκολλο

Ισχυρισμός. Η πιθανότητα ότι η διεργασία i αποτυγχάνει να προσπελάσει τον πόρο σε en γύρους είναι το πολύ 1/e. Έπειτα από e·n(c ln n) γύρους, η πιθανότητα είναι το πολύ n^{-c}.

Άπόδειξη. Έστω F[i, t] = το γεγονός ότι η διεργασία i αποτυγχάνει να προσπελάσει τον πόρο στους γύρους από 1 μέχρι και t. Λόγω ανεξαρτησίας:

$$\Pr[F(i, t)] \leq (1 - 1/(en))^t$$

- Έστω $t = \lceil e \cdot n \rceil$: $\Pr[F(i,t)] \leq \left(1-\frac{1}{en}\right)^{\lceil en \rceil} \leq \left(1-\frac{1}{en}\right)^{en} \leq \frac{1}{e}$
- Έστω $t = \lceil e \cdot n \rceil \lceil c \ln n \rceil$: $\Pr[F(i,t)] \leq \left(\frac{1}{e}\right)^{c \ln n} = n^{-c}$

Τυχαιοποιημένο Πρωτόκολλο

Ισχυρισμός. Η πιθανότητα ότι όλες οι διεργασίες επιτυγχάνουν μέσα σε $2e \cdot n \ln n$ γύρους είναι τουλάχιστον $1 - 1/n$.

Απόδειξη: Έστω F[t] = το γεγονός ότι τουλάχιστον μία διεργασία αποτυγχάνει στους γύρους 1 έως t.

$$\Pr[F[t]] = \Pr\left[\bigcup_{i=1}^{n} F[i,t]\right] \le \sum_{i=1}^{n} \Pr[F[i,t]] \le n\left(1-\frac{1}{en}\right)^{t}$$

όριο ένωσης — από πριν

- Επιλέγοντας $t = 2\lceil en\rceil\lceil c \ln n\rceil$ έχουμε $\Pr[F[t]] \le n \cdot n^{-2} = 1/n$. ▪

Όριο Ένωσης. Δοθέντων γεγονότων $E_1, \ldots, E_n$, $\Pr\left[\bigcup_{i=1}^{n} E_i\right] \le \sum_{i=1}^{n} \Pr[E_i]$

13.2 Καθολική Ελάχιστη Αποκοπή

Καθολική Ελάχιστη Αποκοπή

Δοθέντος ενός συνεκτικού μη κατευθυνόμενου γραφήματος $G = (V, E)$ βρείτε μία αποκοπή (A, B) ελάχιστου μεγέθους (πλήθος ακμών μεταξύ A και B).

Εφαρμογές: Διαμέριση αντικειμένων σε μία βάση δεδομένων, συσταδοποίηση εγγράφων, ανθεκτικότητα δικτύων, σχεδίαση δικτύων, κτλ.

Λύση με Ροές σε Δίκτυα.

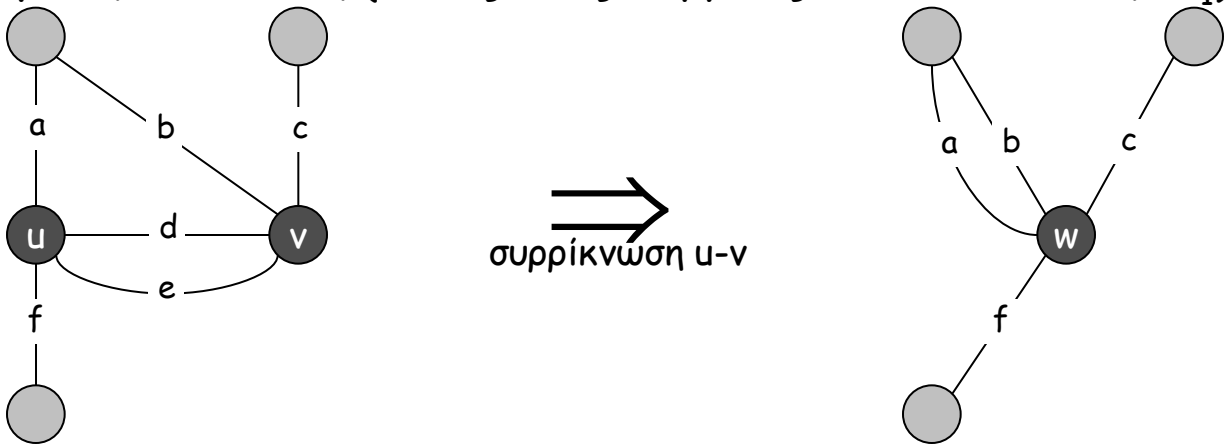
- Αντικατάσταση κάθε ακμής (u, v) με δύο αντίθετα τόξα (u, v) και (v, u).
- Επέλεξε μία κορυφή s και υπολόγισε την ελάχιστη s-v αποκοπή μεταξύ s και κάθε κορυφής $v \in V$.

Λανθασμένη διαίσθηση: Η καθολική ελάχιστη αποκοπή είναι πιο δύσκολη από την ελάχιστη s-t αποκοπή.

Αλγόριθμος Συρρίκνωσης

[Karger 1995]

- Επέλεξε μία ακμή e = (u, v) ομοιόμορφα με τυχαίο τρόπο.
- Συρρίκνωσε την e.
  - Αντικατέστησε την u και v με μία νέα κορυφή w
  - Διατήρηση ακμών, με ενημέρωση των άκρων των ακμών στο u και v προς το w
  - Κρατάμε παράλληλες ακμές, αλλά διαγράφουμε τις ακμές από κορυφή στον εαυτό της
- Επανάληψη μέχρι το γράφημα να έχει δύο κόμβους v_1 και v_2.
- Επέστρεψε την αποκοπή (όλους τους κόμβους που έδωσαν την v_1).

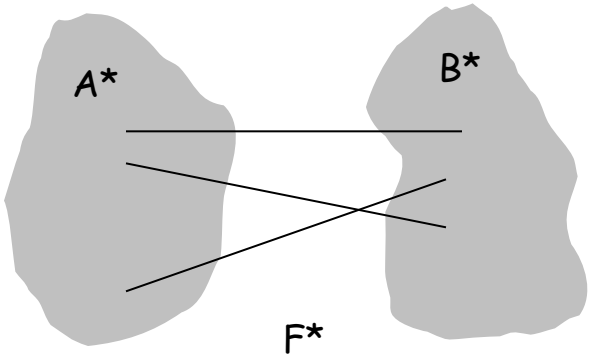


Αλγόριθμος Συρρίκνωσης

Ισχυρισμός. Ο αλγόριθμος επιστρέφει την ελάχιστη αποκοπή με πιθανότητα $\geq 2/n^2$.

Απόδειξη. Έστω μία καθολική αποκοπή (A^*, B^*) του G. Έστω F^* οι ακμές με ένα άκρο στο A^* και το άλλο στο B^*. Έστω $k = |F^*|$ = μέγεθος ελάχιστης αποκοπής.

- Ο αλγόριθμος συρρικνώνει μία ακμή στο F^* με πιθανότητα $k / |E|$.
- Κάθε κορυφή έχει βαθμό $\geq k$ αλλιώς διαφορετικά το (A^*, B^*) δεν θα ήταν ελάχιστη αποκοπή. $\Rightarrow$ $|E| \geq \frac{1}{2}kn$.
- Άρα ο αλγόριθμος συρρικνώνει μία ακμή στο F^* με πιθανότητα $\leq 2/n$.



Αλγόριθμος Συρρίκνωσης

Ισχυρισμός. Ο αλγόριθμος επιστρέφει την ελάχιστη αποκοπή με πιθανότητα $\geq 2/n^2$.

Απόδειξη. Έστω μία καθολική αποκοπή (A*, B*) του G. Έστω F* οι ακμές με ένα άκρο στο A* και το άλλο στο B*. Έστω k = |F*| = μέγεθος ελάχιστης αποκοπής.

- Έστω G' το γράφημα μετά από j επαναλήψεις. Υπάρχουν n' = n-j κόμβοι.
- Έστω ότι καμία ακμή στο F* δεν έχει συρρικνωθεί. Η ελάχιστη αποκοπή στο G' είναι ακόμα k.
- Αφού η ελάχιστη αποκοπή είναι k, $|E'| \geq \frac{1}{2}kn'$.
- Άρα, ο αλγόριθμος συρρικνώνει μία ακμή στο F* με πιθανότητα $\leq 2/n'$.
- Έστω E_j = το γεγονός ότι μία ακμή στο F* δεν συρρικνώνεται στην επανάληψη j.

$$
\begin{aligned}
\Pr[E_1 \cap E_2 \cdots \cap E_{n-2}] &= \Pr[E_1] \times \Pr[E_2 \mid E_1] \times \cdots \times \Pr[E_{n-2} \mid E_1 \cap E_2 \cdots \cap E_{n-3}] \\
&\geq \left(1-\tfrac{2}{n}\right)\left(1-\tfrac{2}{n-1}\right)\cdots\left(1-\tfrac{2}{4}\right)\left(1-\tfrac{2}{3}\right) \\
&= \left(\tfrac{n-2}{n}\right)\left(\tfrac{n-3}{n-1}\right)\cdots\left(\tfrac{2}{4}\right)\left(\tfrac{1}{3}\right) \\
&= \frac{2}{n(n-1)} \\
&\geq \frac{2}{n^2}
\end{aligned}
$$

Αλγόριθμος Συρρίκνωσης

Ενίσχυση. Για να ενισχύσουμε την πιθανότητα επιτυχίας εκτελούμε τον αλγόριθμο πολλές φορές.

Ισχυρισμός. Αν εκτελέσουμε τον αλγόριθμο $n^2 \ln n$ φορές με ανεξάρτητες τυχαίες επιλογές, η πιθανότητα αποτυχίας είναι το πολύ $1/n^2$.

Απόδειξη. Λόγω ανεξαρτησίας, η πιθανότητα αποτυχίας είναι το πολύ

$$\left(1-\frac{2}{n^2}\right)^{n^2 \ln n} = \left[\left(1-\frac{2}{n^2}\right)^{\frac{1}{2}n^2}\right]^{2\ln n} \leq \left(e^{-1}\right)^{2\ln n} = \frac{1}{n^2}$$

$(1 - 1/x)^x \leq 1/e$

Αποτελέσματα

Πολυπλοκότητα. Αργός αλγόριθμος μιας και απαιτούνται $\Theta(n^2 \log n)$ επαναλήψεις όπου κάθε μια απαιτεί $\Omega(m)$ χρόνο.

Βελτίωση. [Karger-Stein 1996] $O(n^2 \log^3 n)$.

- Οι αρχικές επαναλήψεις είναι λιγότερο επικίνδυνες σε σχέση με τις επόμενες: η πιθανότητα συρρίκνωσης ακμής της ελάχιστης αποκοπής είναι 50% όταν απομένουν $n / \sqrt{2}$ κορυφές.
- Εκτελείται ο αλγόριθμος συρρίκνωσης έως ότου να απομείνουν $n / \sqrt{2}$ κορυφές.
- Εκτελούμε δύο φορές τον αλγόριθμο συρρίκνωσης και επιστρέφουμε την καλύτερη αποκοπή.

Επεκτάσεις. Δουλεύει και με θετικά βάρη στις ακμές.

Καλύτερο αποτέλεσμα. [Karger 2000] $O(m \log^3 n)$.

↖ Πιο γρήγορο από τον καλύτερο αλγόριθμο μέγιστης ροής ή από αιτιοκρατικό καθολικής αποκοπής αλγόριθμο.

13.3 Γραμμικότητα Μέσης Τιμής

Αναμενόμενη Τιμή

Δοθείσας μία διακριτής τυχαίας μεταβλητής X, η αναμενόμενη τιμή της E[X] είναι:

$$E[X] = \sum_{j=0}^{\infty} j \Pr[X = j]$$

Παράδειγμα (περιμένοντας την πρώτη επιτυχία). Ένα νόμισμα είναι κορώνα με πιθανότητα p και γράμματα με 1-p. Πόσες ανεξάρτητες ρίψεις X θα γίνουν μέχρι την πρώτη φορά που έρχεται κορώνα?

$$E[X] = \sum_{j=0}^{\infty} j \cdot \Pr[X = j] = \sum_{j=0}^{\infty} j\,(1-p)^{j-1} p = \frac{p}{1-p} \sum_{j=0}^{\infty} j\,(1-p)^{j} = \frac{p}{1-p} \cdot \frac{1-p}{p^2} = \frac{1}{p}$$

Αναμενόμενη Τιμή: 2 Ιδιότητες

Αν η X είναι 0/1 τυχαία μεταβλητή τότε, E[X] = Pr[X = 1].

Απόδειξη.

$$E[X] = \sum_{j=0}^{\infty} j \cdot \Pr[X = j] = \sum_{j=0}^{1} j \cdot \Pr[X = j] = \Pr[X = 1]$$

Γραμμικότητα Μέσης Τιμής. Δοθέντων δύο τυχαίων μεταβλητών X και Y (όχι απαραίτητα ανεξάρτητες και ορισμένες στον ίδιο χώρο πιθανότητας,

E[X + Y] = E[X] + E[Y]

Coupon Collector

Κάθε κουτί δημητριακών περιέχει ένα κουπόνι. Υπάρχουν n διαφορετικοί τύποι κουπονιών. Υποθέτοντας ότι όλα τα κουτιά έχουν την ίδια πιθανότητα να περιέχουν ένα κουπόνι, πόσα κουτιά πρέπει να μαζέψουμε μέχρι να έχουμε ≥ 1 κουπόνι κάθε τύπου;

Ισχυρισμός. Το αναμενόμενο πλήθος βημάτων είναι $\Theta(n \log n)$.

Απόδειξη.

- Φάση j = χρόνος μεταξύ του να έχουμε j καιj+1 διαφορετικά κουπόνια.
- Έστω X_j = πλήθος βημάτων για τη φάση j.
- Έστω X = πλήθος συνολικών βημάτων = $X_0 + X_1 + \ldots + X_{n-1}$.

$$E[X] = \sum_{j=0}^{n-1} E[X_j] = \sum_{j=0}^{n-1} \frac{n}{n-j} = n \sum_{i=1}^{n} \frac{1}{i} = n\,H(n)$$

Πιθανότητα επιτυχίας = (n-j)/n

$\Rightarrow$ αναμενόμενος χρόνος αναμονής = n/(n-j)

13.4 MAX 3-SAT

Μέγιστη 3-Ικανοποιησιμότητα

Ακριβώς 3 διαφορετικά λεξιγράμματα ανά πρόταση

Δοθέντος ενός 3-SAT λογικού τύπου, βρείτε μία απόδοση τιμών αληθείας που ικανοποιεί όσες περισσότερες προτάσεις γίνεται.

$$
\begin{aligned}
C_1 &= x_2 \vee \overline{x_3} \vee \overline{x_4} \\
C_2 &= x_2 \vee x_3 \vee \overline{x_4} \\
C_3 &= \overline{x_1} \vee x_2 \vee x_4 \\
C_4 &= \overline{x_1} \vee \overline{x_2} \vee x_3 \\
C_5 &= x_1 \vee \overline{x_2} \vee \overline{x_4}
\end{aligned}
$$

NP-δύσκολο πρόβλημα βελτιστοποίησης.

Απλή ιδέα: Ρίψη νομίσματος και θέσε την τιμή κάθε μεταβλητής ΑΛΗΘΕΣ με πιθανότητα $\frac{1}{2}$, ανεξάρτητα για κάθε μεταβλητή.

Ανάλυση

Ισχυρισμός. Δοθέντος ενός 3-SAT λογικού τύπου με k προτάσεις, το αναμενόμενο πλήθος προτάσεων που ικανοποιούνται από την τυχαία απόδοση τιμών είναι 7k/8.

Απόδειξη. Έστω η τυχαία μεταβλητή:

$$Z_j = \begin{cases} 1 & \text{αν } C_j \text{ ικανοποιείται} \\ 0 & \text{διαφορετικά} \end{cases}$$

Έστω Z = πλήθος προτάσεων που ικανοποιούνται

$$\begin{aligned} E[Z] &= \sum_{j=1}^{k} E[Z_j] \\ &= \sum_{j=1}^{k} \Pr[\text{η ππρότασ } C_j \text{ ικανοποιείται}] \\ &= \tfrac{7}{8} k \end{aligned}$$

Πιθανοτική Μέθοδος

Συνέπεια: Για κάθε στιγμιότυπο του 3-SAT, υπάρχει τιμή αληθείας που ικανοποιεί τουλάχιστον ένα κλάσμα 7/8 από όλες τις προτάσεις.

Έστω ένα αυθαίρετο στιγμιότυπο του 3-SAT με 7 προτάσεις. Μπορεί να ικανοποιηθεί;

Απόδειξη: Η τυχαία μεταβλητή θα πρέπει να παίρνει τουλάχιστον μία τιμή τουλάχιστον ίση με την μέση της τιμή. ▪

Πιθανοτική Μέθοδος: Δείξαμε την ύπαρξη μίας μη-τετριμμένης ιδιότητας του 3-SAT δείχνοντας ότι μία τυχαιοποιημένη κατασκευή την παράγει με θετική πιθανότητα.!

Προσεγγιστικός Αλγόριθμος

Μπορούμε να φτιάξουμε έναν 7/8-προσεγγιστικό αλγόριθμο? Γενικά, μία τυχαία μεταβλητή μπορεί σχεδόν πάντα να είναι κάτω από την αναμενόμενη τιμή της.

Λήμμα: Η πιθανότητα ότι μία τυχαία απόδοση ικανοποιεί ≥ 7k/8 προτάσεις είναι τουλάχιστον 1/(8k).

Απόδειξη: Έστω p_j η πιθανότητα ότι ακριβώς j προτάσεις ικανοποιούνται. Έστω p η πιθανότητα ότι ≥ 7k/8 προτάσεις ικανοποιούνται.

$$\begin{aligned} \tfrac{7}{8}k = E[Z] &= \sum_{j \geq 0} j\, p_j \\ &= \sum_{j < 7k/8} j\, p_j + \sum_{j \geq 7k/8} j\, p_j \\ &\leq (\tfrac{7k}{8} - \tfrac{1}{8}) \sum_{j < 7k/8} p_j + k \sum_{j \geq 7k/8} p_j \\ &\leq (\tfrac{7}{8}k - \tfrac{1}{8}) \cdot 1 + k\, p \end{aligned}$$

Παίρνουμε p ≥ 1 / (8k). ▪

Μέγιστη Ικανοποιησιμότητα

Επεκτάσεις.

- Επιτρέπουμε 1, 2 ή περισσότερα λεξιγράμματα ανά πρόταση.
- Εύρεση μέγιστου ζυγισμένου συνόλου ικανοποιήσιμων προτάσεων.

Θεώρημα. [Asano-Williamson 2000] Υπάρχει ένας 0.784-προσεγγιστικός αλγόριθμος για το MAX-SAT.

Θεώρημα. [Karloff-Zwick 1997, Zwick+computer 2002] Υπάρχει ένας 7/8-προσεγγιστικός αλγόριθμος για το MAX-3SAT όπου κάθε πρόταση έχει το πολύ 3 λεξιγράμματα.

Θεώρημα. [Håstad 1997] Εκτός και αν P = NP, δεν υπάρχει ρ-προσεγγιστικός αλγόριθμος για το MAX-3SAT (και άρα και για το MAX-SAT) για κάθε $\rho > 7/8$.