ΕΛΕΓΧΟΣ ΚΛΗΣΕΩΝ ΣΕ ΔΑΚΤΥΛΙΟ

U. Adamy, C. Ambuehl, R. Anand, T. Erlebach

Δομή παρουσίασης

- ✓ Γενικά
- ✓ Ορισμός προβλήματος
- ✓ Σχετιζόμενη δουλειά
- ✓ Εισαγωγικά
- ✓ Αλγόριθμος
- ✓ Παράδειγμα εκτέλεσης
- ✓ Υπολογισμός χρονικής πολυπλοκότητας

Γενικά:

- Το πρόβλημα ελέγχου κλήσεων είναι ένα σπουδαίο πρόβλημα βελτιστοποίησης το οποίο αντιμετωπίζεται στο σχεδιασμό και στη λειτουργία των τηλεπικοινωνιακών δικτύων.
- Στόχος του είναι: για δεδομένο δακτύλιο με συγκεκριμένες χωρητικότητες ακμών και σύνολο μονοπατιών να βρεθεί το μέγιστης πληθικότητας υποσύνολο μονοπατιών έτσι ώστε καμία χωρητικότητα ακμών να μήν παραβιάζεται.
- Εδώ θα δοθεί πολυωνυμικού χρόνου αλγόριθμος για μια optimal λύση.

- Ο αλγόριθμος βασίζεται σε διαδικασία απόφασης που ελέγχει αν υπάρχει λύση με τουλάχιστον κ μονοπάτια.

- Η λύση αυτή υλοποιείται με μια επαναληπτική greedy προσέγγιση.

- Θα δείξουμε ότι ο αλγόριθμος είναι αποδοτικός αφού θα χρησιμοποιήσουμε έναν γραμμικού χρόνου αλγόριθμο για να λύσουμε το πρόβλημα ελέγχου κλήσεων σε αλυσίδες optimally.

Ορισμός Προβλήματος

Ένα <u>στιγμιότυπο</u> του προβλήματος δίνεται:

- ✓ σε ένα μη κατευθυνόμενο γράφο (V,E)
- ✓ με χωρητικότητες ακμών $c : E \to N$
- ✓ και ένα σύνολο P από m μονοπάτια στο (V,E)

- Μονοπάτια:

  είναι οι αιτήσεις σύνδεσης των οποίων η αποδοχή απαιτεί την εξασφάλιση μιας συχνότητας σε όλες τις ακμές του μονοπατιού.

- Εφικτή λύση:

  είναι σύνολο $Q \supseteq P$ τέτοια ώστε για κάθε ακμή $e \in E$ ο αριθμός των μονοπατιών στο Q να περιέχει ακμές e που να είναι το πολύ c(e) .

- Ένα τέτοιο σετ μονοπατιών ονομάζεται εφικτό σετ μονοπατιών και τα μονοπάτια ονομάζονται αποδεκτά

- Ο αντικειμενικός σκοπός είναι να μεγιστοποιήσουμε τον αριθμό των αποδεκτών μονοπατιών.
- Εάν μιλούσαμε για O.W.D.M. δίκτυα δακτυλίων με w μήκη κύματος που έχουν μετατροπέα κύματος σε τουλάχιστον ένα κόμβο μπορεί να μοντελοποιηθεί ως έλεγχος κλήσεων με όλες τις χωρητικότητες ακμών ίσες με w.
- Επιπλέον πρέπει να σημειωθεί ότι αν μιλούσαμε για on-line επίλυση του αλγορίθμου θα σήμαινε ότι οι αιτήσεις εμφανίζονται μία προς μια και ο αλγόριθμος πρέπει να δεχθεί και να απορρίψει καθεμία αίτηση χωρίς να γνωρίζει μελλοντικές αιτήσεις

Ο off-line αλγόριθμος ελέγχου κλήσεων χρειάζεται:

a) στο δίκτυο για τη φάση του σχεδιασμού όπου θεωρούμε γνωστή την υποψήφια τοπολογία δικτύου με τις δυνατότητες σύνδεσης επίσης γνωστές και κάποιος θέλει να ξέρει πόσες από τις προβλεπόμενες αιτήσεις δρομολόγησης θα ικανοποιηθούν από το δίκτυο
b) είναι χρήσιμος στο σενάριο που υποστηρίζει προκαταβολικά κρατήσεις συνδέσεων γιατί έτσι είναι δυνατόν να συλλέξει έναν αριθμό από αιτήσεις κράτησης πριν διεξαχθεί ο έλεγχος εισόδου για ολόκληρη την παρτίδα αιτήσεων και

c) τέλος ένας optimal off-line αλγόριθμος είναι χρήσιμος ως σημείο αναφοράς για αξιολόγηση άλλων off-line ή on-line ελέγχου κλήσεων στρατηγικών.

Σχετιζόμενη δουλειά

a) Γράφοι συγκρούσεων μονοπατιών σε δακτύλιο μπορούν να αναγνωριστούν αποδοτικά.

b) Για δεδομένο δακτύλιο το πρόβλημα του max independent set και το πρόβλημα της max clique μπορεί να υπολογιστεί σε πολυωνυμικό χρόνο.

c) Ο χρωματισμός δακτυλίου με minimum αριθμό χρωμάτων είναι NP-hard.

d) Ένας χρωματισμός με το πολύ w χρώματα υπάρχει και μπορεί να υπολογιστεί αποδοτικά με w τη maximum κλίκα του γράφου.

Εισαγωγικά

Έστω $P = \{ p_1 \ldots\ldots p_m \}$ το σετ των m μονοπατιών καθένας από τους οποίους συνδέει 2 κόμβους στο δακτύλιο.

Κάθε $p_i \in P$ είναι ένα διατεταγμένο ζεύγος κόμβων $p_i = (s_i , t_i) \in V^2$ με $s_i \neq t_i$.

Το p_i περιλαμβάνει όλες τις ακμές από τον κόμβο αφετηρίας s_i στον κόμβο στόχο t_i με ωρολογιακή φορά.

Για $Q \subseteq P$ ο φόρτος δακτυλίου $L(Q,e_i)$ ακμής e_i είναι ο αριθμός των μονοπατιών στο Q που χρησιμοποιούν την ακμή e_i .

Δηλ. $L(Q,e_i) = |\{ p \in Q : e_i \in p\}|$

Ένα $Q \subseteq P$ ονομάζεται εφικτό εαν ο φόρτος του δακτυλίου δεν υπερβαίνει τη χωρητικότητα καμίας ακμής δηλ. $L(Q,e_i) \leq c(e_i) \quad \forall e_i \in E$

Αν θεωρήσουμε ως αφετηρία τον κόμβο *O* χωρίσουμε το σύνολο P των μονοπατίων σε δύο υποσύνολα P_1 και P_2 όπου το P_1 είναι το σύνολο των μονοπατίων που δεν περιέχει τον κόμβο *O* ως εσωτερικό κόμβο και P_2 τα υπόλοιπα μονοπάτια.

Κάθε μονοπάτι στο P_2 αποτελείται από δύο τμήματα: Η κεφαλή του μονοπατιού που ξεκινά από τον κόμβο αφετηρίας έως τον κόμβο O. Η ουρά του μονοπατιού που ξεκινά από τον *O* έως τον κόμβο στόχο.

Δηλ. $P_1 = \{ p_i \in P : s_i < t_i \}$

$P_2 = \{ p_i \in P : s_i > t_i \}$

Και φυσικά $P = P_1 \cup P_2$

Θεωρούμε γραμμική ταξινόμηση των μονοπατιών στο P ως εξής:

Όλα τα μονοπάτια στο P_1 είναι αυστηρά μικρότερα από όλα τα μονοπάτια του P_2.

Μέσα και στα δύο υποσύνολα διατάσσουμε τα μονοπάτια αυξανομένου του κόμβου στόχου.

Μονοπάτια με ίδιο κόμβο στόχο ταξινομούνται αυθαίρετα.

Ονομάζουμε την ταξινόμηση greedy.

Ο αλγόριθμος θεωρεί αλυσίδα 2n κόμβων αποτελούμενη από δύο αντίγραφα του δακτυλίου κολλημένα. Οι ουρές των P_2 μονοπατιών είναι στο δεύτερο αντίγραφο ενώ τα P_1 μονοπάτια και οι κεφαλές των P_2 μονοπατιών είναι στο πρώτο αντίγραφο.

Για δοσμένο σύνολο Q μονοπατιών ορίζουμε:

$L_1(Q,e_i)$ και $L_2(Q,e_2)$ το φορτίο των μονοπατιών στο Q στο πρώτο αντίγραφο του e_i και το φορτίο των μονοπατιών στο Q στο δεύτερο αντίγραφο του e_i αντίστοιχα

Έτσι τα μονοπάτια του P_1 και οι κεφαλές των μονοπατίων του P_2 συμβάλλουν στο φορτίο $L_1(Q,e_i)$ ενώ οι ουρές των P_2 μονοπατιών καθορίζουν την τιμή φορτίου για $L_2(Q,e_i)$.

Το φορτίο του δακτυλίου είναι απλά $L_1 + L_2$.

Ορισμός προφίλ

Έστω Q σύνολο μονοπατιών. Το προφίλ π του Q είναι μια μειούμενη ακολουθία n τιμών φόρτου L_2 για τις ακμές e_0 , e_1……..e_{n-1} στο δεύτερο αντίγραφο του δακτυλίου.

$$\pi_Q := L_2(Q,e_0)\ldots\ldots L_2(Q,e_{n-1})$$

Με $\pi_Q(e_i)$ ορίζουμε τις τιμές προφίλ $L_2(Q,e_i)$ για όλες τις ακμές $e_i \in E$.

Το άδειο προφίλ είναι μηδέν παντού.

Για π,π΄ έχουμε π≤π΄ ανν $\pi(e_i) \le \pi'(e_i)$ για κάθε $e_i \in E$

Ένα σύνολο Q μονοπατιών είναι chain-feasible εαν δεν υπερβαίνει την χωρητικότητα καμίας ακμής στην αλυσίδα μήκους 2n.

Με άλλα λόγια Q είναι chain-feasible εαν δεν υπερβαίνει τις χωρητικότητες και στα δυο αντίγραφα του δακτυλίου

$L_1(Q,e_i) \le c(e_i)$ και $L_2(Q,e_i) \le c(e_i)$ για κάθε $e_i \in E$

Αλγόριθμος

Σκοπός του αλγορίθμου να βρει το maximum μέγεθος εφικτού υποσυνόλου μονοπατιών στο P.

- ✓ Ο αλγόριθμος φτιάχνει την αλυσίδα 2n ακμών
- ✓ Ταξινομεί τα μονοπάτια στο P σύμφωνα με τη greedy διάταξη
- ✓ Η καρδιά του αλγορίθμου είναι μια διαδικασία απόφασης ώστε με δοσμένη παράμετρο κ να αποφασίζει αν υπάρχει εφικτή λύση Q μεγέθους κ ή όχι.

- ✓ Προφανώς το maximum κ μπορεί να βρεθεί με μερικές επαναλήψεις αυτής της διαδικασίας

- ✓ Η διαδικασία απόφασης κάνει χρήση του greedy αλγορίθμου.

- ✓ Εάν προσθέτοντας το υπόψην μονοπάτι δεν υπερβαίνει κανένα περιορισμό χωρητικότητας ακμών το μονοπάτι γίνεται δεκτό και οι χωρητικότητες των ακμών μειώνονται ανάλογα αλλιώς απορρίπτεται.

Περιγραφή διαδικασίας απόφασης

a) Ξεκινάμε με άδειο προφίλ

b) Η διαδικασία απόφασης δουλεύει με επαναλήψεις

Σε κάθε επανάληψη υπολογίζει μια greedy λύση κ μονοπατιών για δοσμένο προφίλ ως εξής:

Ξεκινά με χωρητικότητες ακμών $c(e_i)$ και αφαιρεί τις τιμές προφίλ από τις αρχικές χωρητικότητες από τις ακμές στο πρώτο αντίγραφο αφού αυτές είναι κατειλλημένες από το προφίλ.

c) Μετά ξεκινά να βάζει κ μονοπάτια χρησιμοποιώντας το greedy αλγόριθμο.

d) Εάν της διαδικασίας τελειώσουν τα μονοπάτια πριν επιλέξει κ από αυτά τότε δεν υπάρχει εφικτή λύση μεγέθους κ

Απαντά όχι και σταματά

Αλλιώς έστω Q_i το υποψήφιο σύνολο κ επιλεγμένων μονοπατιών στην i-οστή επανάληψη.

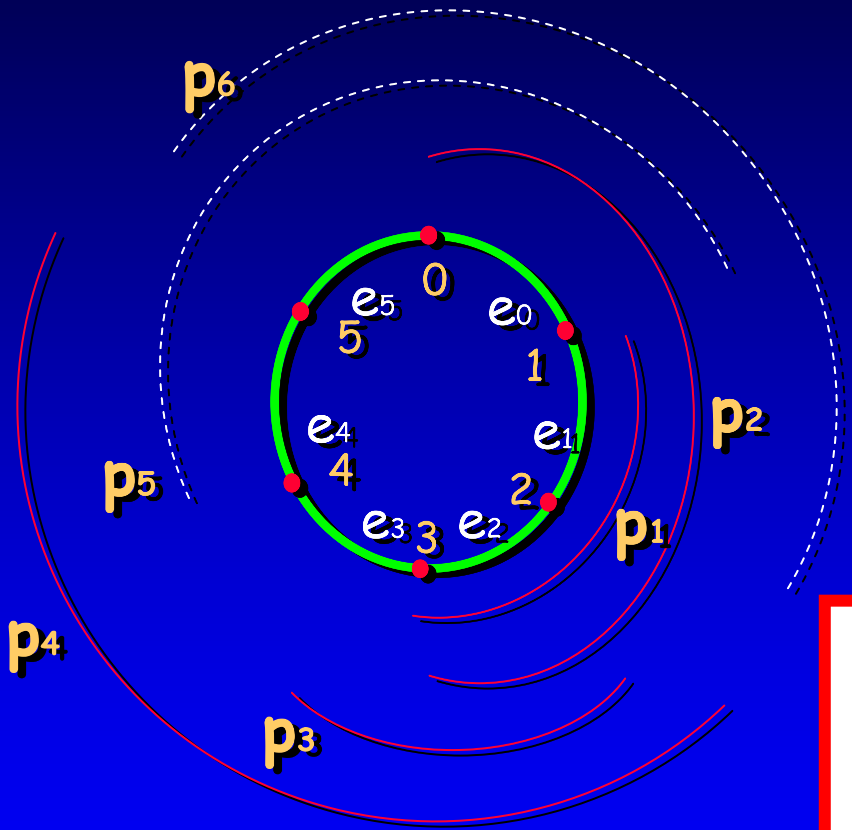
Από κατασκευής το σύνολο Q_i είναι chain-feasible για το αρχικό προφίλ αλλά όχι απαραίτητα επιλύσιμο για δακτύλιο αφού οι ουρές των επιλεγμένων P_2 μονοπατιών μαζί με τα επιλεγμένα μονοπάτια του P_1 μπορούν να παραβιάσουν περιορισμούς χωρητικότητας.

e) Στο τέλος της i-οστής επανάληψης η διαδικασία συγκρίνει το προφίλ Q_i με το προφίλ της προηγούμενης επανάληψης. Εάν είναι ίσα τότε: τα μονοπάτια του Q_i είναι αποδεκτή λύση μεγέθος κ.

f) Η διαδικασία τυπώνει Q_i απαντά ναι και σταματά αλλιώς η διαδικασία χρησιμοποιεί το Q_i ως αφετηρία της $i + 1$ επανάληψης.

Τα προφίλ των επιλεγμένων με τη μέθοδο greedy Q_i λειτουργούν ως lower bound για εφικτή λύση με την έννοια ότι δεν υπάρχει εφικτή λύση με μικρότερο προφιλ.

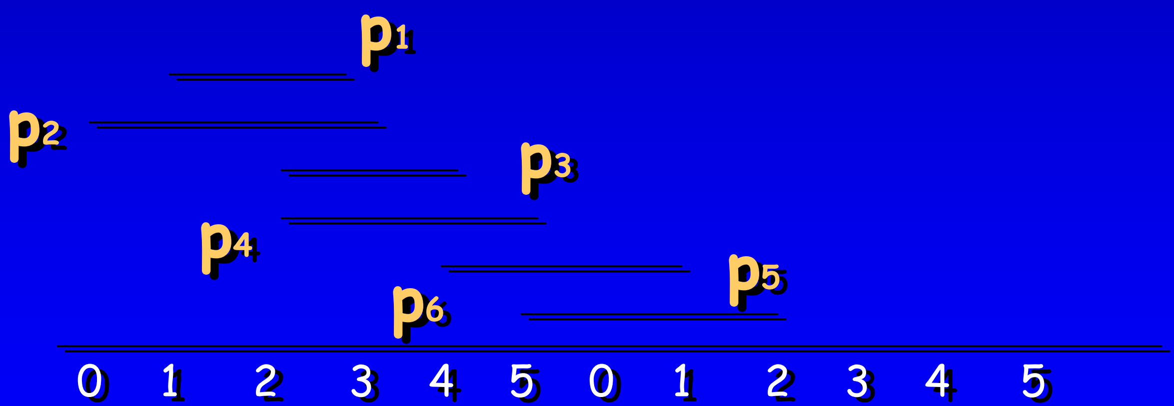
Παράδειγμα εφαρμογής



Σχήμα 1.

- ✓ Έστω χωρητικότητες $c(e_i) = 2$
- ✓ Ρωτάμε για $κ = 4$

Τα μονοπάτια θα εξεταστούν με τη σειρά του σχήματος 2. (greedy διάταξη)



Σχήμα 2.

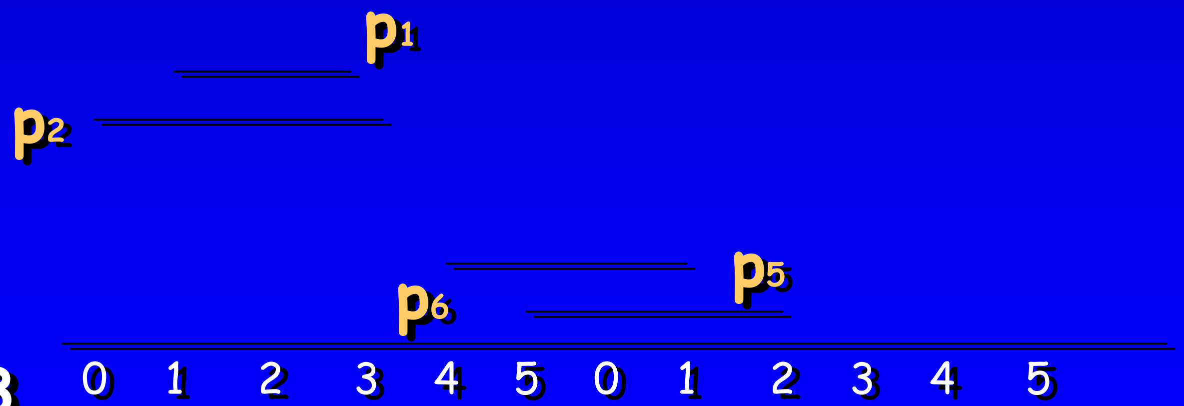
Αρχικά τα p_1 και p_2 γίνονται δεκτά ενώ τα p_3 και p_4 απορρίπτονται γιατί προφανώς παραβιάζουν τα όρια χωρητικότητας. Τα $\{p_5,p_6\}$ γίνονται και τα δύο δεκτά

- ✓ Και $Q_1 = \{p_1,p_2,p_5,p_6\}$
- ✓ Το προφίλ του Q_1 είναι 2 για e_0

  1 για e_1

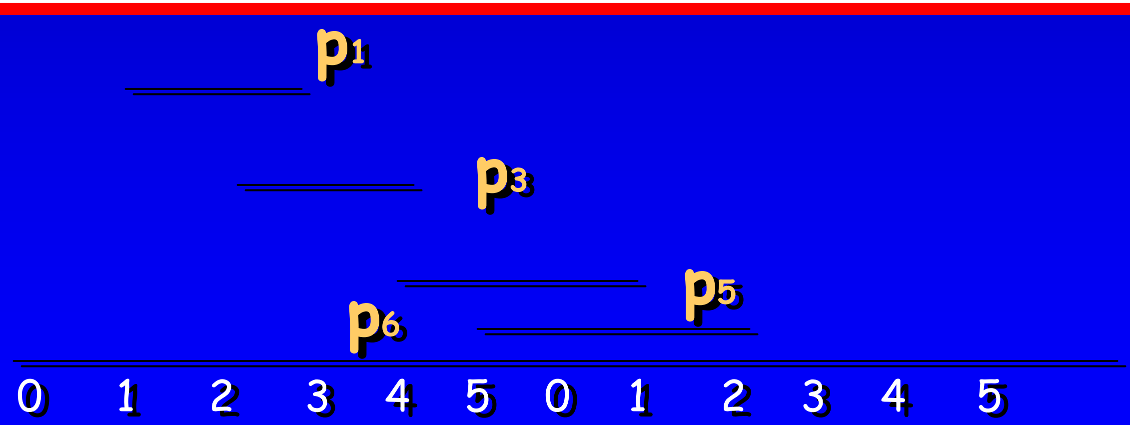
  0 αλλού.

Η Q1 δεν είναι εφικτή λύση γιατί $L(Q_1,e_0) = 3 > c(e_0)$



Σχήμα 3.

- Η διαδικασία ξεκινά δεύτερη επανάληψη με προφιλ Q_1 ως αρχικό προφίλ.Σε αυτήν την επανάληψη η διαδικασία αποδέχεται τα μονοπάτια $Q_2 = \{ p_1, p_3, p_5, p_6 \}$
- Το p_2 απορρίπτεται γιατί οι ακμές e_0 , e_1 γεμίζουν με το προφίλ Q_1 και το μονοπάτι p_1.Το p_4 απορρίπτεται.
- Το προφίλ Q_2 είναι 2 για e_0
1 για e_1
0 αλλού
- αφού $\pi_{Q1} = \pi_{Q2}$ η Q_2 είναι εφικτή λύση μεγέθους 4.
- Η διαδικασία σταματά.



Σχήμα 4.

Υπολογισμός Χρονικής Πολυπλοκότητας

Καλούμε τη διαδικασία απόφασης με binary search άρα logm φορές.

Σε κάθε εφαρμογή της διαδικασίας απόφασης ο αριθμός των επαναλήψεων είναι το πολύ nc_{min}

Γιατί: ισχύει $\sum_{j=0}^{n-1} \pi_{Q}(e_j) \leq nc_{min}$

και επειδή προφανώς το προφίλ μεγαλώνει σε κάθε επανάληψη

άρα οι επαναλήψεις είναι το πολύ nc_{min}

Αλλά κάθε επανάληψη απαιτεί την εκτέλεση του greedy αλγορίθμου που θέλει χρόνο $O(mn)$.

Άρα συνολικά $O(mn^2 c_{min} \log m)$.

Βελτίωση χρόνου

Ο χρόνος μπορεί να μειωθεί σε $O(mnc_{min} \log m)$.