



3η Εργασία για το μάθημα Οπτικά Δίκτυα Επικοινωνιών

Υλοποίηση Αλγορίθμου “*k shortest path routing*”

Το πρόβλημα *k-shortest path routing* είναι μια γενίκευση του συντομότερου προβλήματος δρομολόγησης διαδρομής σε ένα δεδομένο δίκτυο. Στο πρόβλημα αυτό, ζητείται η εύρεση, όχι μόνο της μιας συντομότερης διαδρομής αλλά και οι επόμενες *k-1* συντομότερες διαδρομές (που μπορεί να είναι μεγαλύτερες από τη μικρότερη διαδρομή). Η εύρεση *k* συντομότερων διαδρομών είναι δυνατή με την επέκταση του αλγορίθμου Dijkstra ή του αλγόριθμου Bellman-Ford ώστε να βρεθούν περισσότερες από μία διαδρομές.

A) Υλοποιήστε τον αλγόριθμό εύρεσης των *k* πιο σύντομων διαδρομών είτε σε περιβάλλον Matlab είτε σε python χρησιμοποιώντας τον βασικό αλγόριθμο Dijkstra.

Η τοπολογία δικτύου θα είναι μεταβλητή και θα ορίζεται στην αρχή του προγράμματος σας, με έναν πίνακα. Οι στήλες -γραμμές του πίνακα θα ορίζουν τόσο την ύπαρξη ζεύξης αλλά και το κόστος αυτής. Για παράδειγμα το στοιχείο $\alpha_{ij} = 5$ θα ορίζει ότι το κόστος από τον κόμβο *i* στον *j* είναι 5. Όλες οι ζεύξεις θα είναι αμφίδρομες με το ίδιο κόστος (δλδ $\alpha_{ij} = \alpha_{ji} = 5$).

Για παράδειγμα:

```
A = [ 0 0 5 0 0  
      0 0 0 3 7  
      5 0 0 1 0  
      0 3 1 0 1  
      0 7 0 1 0 ]
```

Το πρόγραμμα στην έξοδο του πρέπει να τυπώνει: α) την τοπολογία που χρησιμοποιήθηκε (τα κόστη θα ορίζονται ως σταθερές στο πρόγραμμα), β) τα *k=3* μονοπάτια που υπολογίστηκαν ανά ζευγάρι κόμβων. Ο αριθμός *k* θα είναι μεταβλητή του προγράμματος.

B) Στα αποτελέσματα που υπολογίσατε να υλοποιήσετε αλγόριθμο επιλογής (εκ των *k=3*) για κάθε ζεύγος κόμβων ώστε να **ελαχιστοποιείται** η χρήση κάθε ζεύξης.

Εάν μια ζεύξη εμφανίζεται σε *N* μονοπάτια, τότε η χρήση της είναι *N*, δλδ ζητείτε η ελαχιστοποίηση των μηκών κύματος που περνάνε από εκείνη την ζεύξη (άρα και των διαφορετικών μηκών κύματος που χρησιμοποιούνται στο δίκτυο).

Το πρόγραμμα στην έξοδο του να τυπώνει:

- α) τον μέγιστο αριθμό *N* χρήσης μιας ζεύξης,
- β) τον ελάχιστο αριθμό χρήσης μιας ζεύξης και
- γ) το μ.ο. χρήσης όλων των ζεύξεων.

Γ) Έστω ότι στο δίκτυο που έχετε σχεδιάσει, υπάρχουν διαθέσιμα μόνο $\Lambda=5$ μήκη κύματος, η κίνηση είναι δυναμική και καθορίζεται από ένα πίνακα (traffic matrix) όπως παρακάτω:

```
B = [ 0 0 2 0 0  
      0 0 0 1 2  
      1 0 0 1 0
```



Εργαστήριο Δικτύων Τμήμα Μηχανικών Η/Υ & Πληροφορικής Πανεπιστήμιο Πατρών

0 1 1 0 1
0 2 0 1 0]

όπου κάθε στοιχείο προσδιορίζει τον αριθμό των μηκών κύματος που ζητάει το ζευγάρι κόμβων ij . Δλδ το στοιχείο $a_{13}=2$ δηλώνει ότι ο κόμβος $i=1$ (πηγή) ζητάει την εγκατάσταση σύνδεσης 2 μηκών κύματος (lightpaths) με τον κόμβο $j=2$ (προορισμός). Η παράμετρος Λ θα οριστεί ως μεταβλητή στο πρόγραμμά σας.

Για τα μονοπάτια που έχετε υπολογίσει στο β) ερώτημα, κάντε ανάθεση μηκών κύματος με τους αλγόριθμους 1) Random 2) First Fit και 3) Least used.

Στην έξοδο του προγράμματος να τυπώνεται :

1. η λίστα των συνδέσεων στη μορφή $\lambda_i = i-j, \dots$ δλδ το λ_i ποιες συνδέσεις εξυπηρετεί.
2. Το ποσοστό των αιτήσεων που δεν εξυπηρετείται δλδ το ποσοστό των αιτήσεων που έγινε "blocked" από τον κάθε αλγόριθμο, γιατί εξαντλήθηκαν τα μήκη κύματος. *Να βελτιστοποιήσετε το πρόγραμμα σας ώστε να επιτυγχάνεται το μικρότερο ποσοστό μη-εξυπηρέτησης (blocking).*
3. Ο χρόνος εκτέλεσης (εφόσον είναι δυνατόν) του κάθε αλγορίθμου.
4. Να επαναλάβετε για διαφορετικό $\Lambda=6,7,++$ μέχρι να μηδενιστεί το ποσοστό μη-εξυπηρέτησης. Σε ποιο Λ επιτεύχθηκε?
5. Ποιος αλγόριθμος είναι πιο αποδοτικός?

Περιγραφή αλγορίθμων:

Random: Αναζητά όλα τα διαθέσιμα μήκη κύματος σε κάθε σύνδεσμο της διαδρομής και στη συνέχεια επιλέγει ένα διαθέσιμο μήκος κύματος τυχαία με ομοιόμορφη πιθανότητα. Σε αυτόν τον αλγόριθμο, καθορίζονται πρώτα όλες οι πιθανές διαδρομές μεταξύ ενός ζεύγους κόμβου προέλευσης-προορισμού (έχετε ήδη βρει τα μονοπάτια στο ερώτημα Β). Στη συνέχεια, ανακαλύπτονται όλα τα ελεύθερα μήκη κύματος (τα οποία δεν χρησιμοποιούνται αυτήν τη στιγμή). Στη συνέχεια, εκχωρείται τυχαία ένα μήκος κύματος για τη μετάδοση δεδομένων. Ο αλγόριθμος είναι κεντρικοποιημένος, αφού απαιτείται η γνώση πόσο χρησιμοποιείτε το κάθε μήκος κύματος.

First-Fit: Σε όλα τα μήκη κύματος ανατίθεται ένας δείκτης (πχ $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$). Όταν γίνεται μια αίτηση σύνδεσης, το μήκος κύματος που έχει τον μικρότερο εκχωρημένο αριθμό και είναι ελεύθερο στη πρώτη μόνο ζεύξη, επιλέγεται από το διαθέσιμο σύνολο μήκους κύματος. Ο αλγόριθμος αυτός δεν είναι κεντρικοποιημένος σαν τον random και δεν είναι γνωστή χρήση του επιλεχθέντος μήκους κύματος στις επόμενες ζεύξεις.

Least used: Αυτός ο αλγόριθμος, επιλέγει τα λιγότερο χρησιμοποιούμενα μήκη κύματος διατηρώντας έτσι ίσο το φορτίο σε όλα τα μήκη κύματος. Αυτό επιτρέπει να διατίθεται περισσότερος αριθμός μηκών κύματος για τα νέα αιτήματα. Ο αλγόριθμος είναι κεντρικοποιημένος, αφού απαιτείται η γνώση πόσο χρησιμοποιείτε το κάθε μήκος κύματος.