

A dark blue vertical bar runs down the left side of the slide. A blue arrow points to the right from this bar, containing the date.

22/9/2019

UNIT CAPACITY NETWORKS

ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ ΑΛΓΟΡΙΘΜΩΝ

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

ΠΑΝΑΓΙΩΤΗΣ ΤΖΟΛΑΣ 6230

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πίνακας περιεχομένων

Περιγραφή.....	3
Συναρτήσεις	4
Συναρτήσεις Γραφημάτων	4
Συναρτήσεις Αλγορίθμου	5
Υβριδικός Αλγόριθμος	6
Πρώτη Φάση	6
Δεύτερη Φάση.....	6
Πειραματική Αξιολόγηση	7
Μετρήσεις	7
Αιτιολόγηση αποτελέσματος	7
Στιγμιότυπα Εκτέλεσης.....	8

Περιγραφή

Η εργασία αναφέρεται στο πρόβλημα της μέγιστης ροής. Πιο συγκεκριμένα εξετάζει την περίπτωση δικτύων με χωρητικότητα ακμών διέλευσης ίση με την μονάδα ή αλλιώς “Unit capacity Networks”. Για την εύρεση της μέγιστης χωρητικότητας χρησιμοποιείται ένας υβριδικός αλγόριθμος αποτελούμενος από δύο φάσεις. Στη πρώτη φάση χρησιμοποιείται ο αλγόριθμος συντομότερων διαδρομών επαύξησης, ενώ στην δεύτερη φάση χρησιμοποιείται ένας αλγόριθμος ελάχιστων αποκοπών. Αυτό συμβαίνει γιατί οι δύο αυτοί αλγόριθμοι έχουν μειονέκτημα καθώς είναι αργοί στις επιμέρους φάσεις τους. Έτσι επιλέγεται ένα όριο με βάση το οποίο τερματίζει ο πρώτος πριν όμως ολοκληρωθεί και τον διαδέχεται ο δεύτερος επιτυγχάνοντας με αυτό το τρόπο καλύτερη χρονική πολυπλοκότητα.

Συναρτήσεις

Για την καλύτερη κατανόηση των συναρτήσεων και της λειτουργικότητάς τους τις χωρίζουμε σε 2 ομάδες. Η πρώτη ομάδα έχει να κάνει με συναρτήσεις που δημιουργούν, επεξεργάζονται και προβαίνουν σε αρχικοποιήσεις παραμέτρων των γραφημάτων. Στην δεύτερη ομάδα ανήκουν οι συναρτήσεις που αναφέρονται και υλοποιούν τον αλγόριθμο.

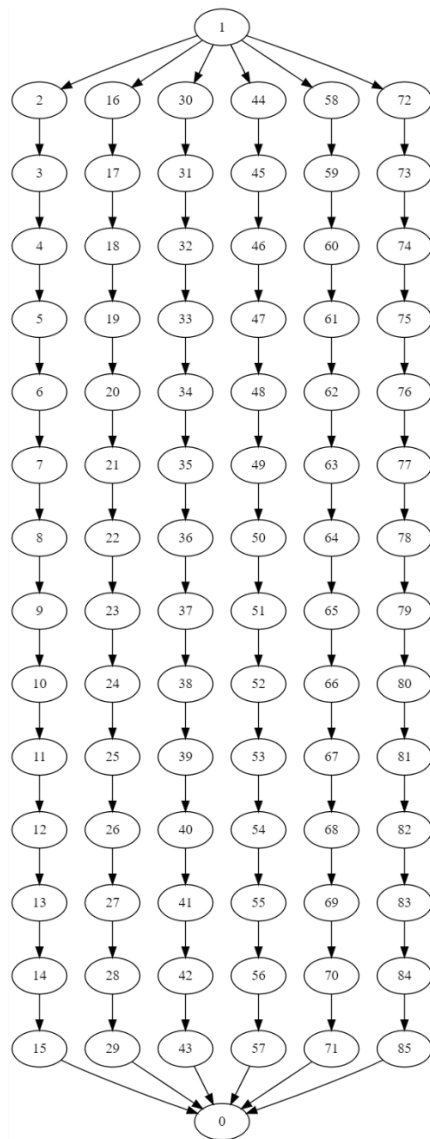
Συναρτήσεις Γραφημάτων

`create_graph(...)` : Η συνάρτηση αυτή δημιουργεί ένα απλό-δοκιμαστικό γράφημα.

`create_figure_graph(...)`: Η συνάρτηση αυτή δημιουργεί γραφήματα αντίστοιχα με αυτά του παρατίθενται στο άρθρο εργασίας.

`create_figure_2(...)`: Το ίδιο με την παραπάνω συνάρτηση.

`create_figure_3(...)`: Δημιουργεί γράφημα με προκαθορισμένο αριθμών γραμμών και στηλών που δηλώνεται στην αρχή του προγράμματος. Και έχει την παρακάτω μορφή.



`generate_random_graph()`: Αυτή η συνάρτηση παίρνει ένα από τα παραπάνω γραφήματα και με την χρήση συνάρτησης `rand()` τα μετατρέπει και δημιουργεί τυχαία γραφήματα. Δεν δημιουργεί τα ίδια γραφήματα αλλά επιλέγει κόμβους με πιθανότητα και τους ενώνει καταφέροντας έτσι να δίνει συνεχώς διαφορετικά γραφήματα.

`set_vertices(...)`: Αυτή η συνάρτηση αρχικοποιεί παραμέτρους των κόμβων του γραφήματος όπως το `property_map color` που θα χρειαστεί αργότερα για την Αναζήτηση Πρώτα κατά Πλάτος.

`set_edges()`: Η συνάρτηση αυτή αρχικοποιεί παραμέτρους των ακμών του γραφήματος, π.χ. τα `property_map` για την χωρητικότητα και την ροή.

`print_vertices()`: Αυτή η συνάρτηση εκτυπώνει τις κορυφές/κόμβους του γραφήματος.

Συναρτήσεις Αλγορίθμου

`my_leda_alg(...)` : Η συνάρτηση αυτή μετατρέπει το γράφημα της `boost` σε ένα πανομοιότυπο της `LEDA` και στην συνέχεια εκτελεί τον αλγόριθμο `MAX_FLOW` της `LEDA`.

`my_BFS(...)`: Η συνάρτηση αυτή εκτελεί Αναζήτηση Πρώτα κατά Πλάτος. Πιο συγκεκριμένα χρησιμοποιείται στην δεύτερη φάση του αλγορίθμου για να ορίσει τα σύνολα αποκοπής που σχετίζονται με την ετικέτα απόστασης.

`reverse_BFS(...)`: Αυτή η συνάρτηση αποτελεί μέρος της πρώτης φάσης του υβριδικού αλγορίθμου και χρησιμοποιείται για να ορίσει τις αποστάσεις των ακμών από τον κόμβο απόληξης, ή `sink`, `t`. Ξεκινάει από το `t` και εκτελεί ανάποδα `BFS`.

`advance(...)`: Συνάρτηση που χρησιμοποιείται για την εύρεση του μονοπατιού.

`retreat(...)` : Συνάρτηση που χρησιμοποιείται κατά την αναζήτηση του μονοπατιού και δηλώνει την υποχώρησή του κατά μία ακμή. Έχει μια επιπλέον «λειτουργία» με την οποία αυξομειώνει τις ετικέτες απόστασης.

`retreat_b(...)` : Η συνάρτηση αυτή χρησιμοποιείται στην δεύτερη φάση του αλγορίθμου και εκτελεί σχεδόν ότι και η προηγούμενη κάνοντας επιπλέον και τις προσαυξήσεις στα σύνολα αποκοπής όπως αναφέρεται στο επιστημονικό άρθρο ενασχόλησής μας.

`augment2(...)`: Αφού καταλήξουμε στον κόμβο απόληξης `t`, έχουμε εντοπίσει μια διαδρομή επαύξησης. Αυτή η συνάρτηση προωθεί ροή στην διαδρομή `s-t` που έχει εντοπιστεί.

`phase_b(...)`: Αυτή η συνάρτηση εκτελεί την δεύτερη φάση του αλγορίθμου ψάχνοντας-δημιουργώντας ελάχιστη αποκοπή μεταξύ των συνόλων.

`shortest_augmenting_path_alg(...)` : Αυτή η συνάρτηση εκτελεί τον αλγόριθμο της πρώτης φάσης όπως παρατίθεται στο επιστημονικό άρθρο και στην συνέχεια καλεί την συνάρτηση για την δεύτερη φάση.

Υβριδικός Αλγόριθμος

Όπως έχουμε αναφέρει το πρόβλημα της μέγιστης ροής στη συγκεκριμένη εργασία επιλύεται με έναν υβριδικό αλγόριθμο αποτελούμενο από δύο φάσεις.

Πρώτη Φάση

Εκτελούμε τον Αλγόριθμο Συντομότερων Διαδρομών Επαύξεσης όπως αναφέρεται στο επιστημονικό άρθρο. Καθώς ο αλγόριθμος εκτελείται αλλάζει τις ετικέτες απόστασης των κόμβων. Τερματίζει όταν η ετικέτα απόστασης του κόμβου έναρξης ξεπεράσει το όριο $\min(\lfloor n^{2/3}m \rfloor, \lfloor m^{1/2} \rfloor)$ όπου n το πλήθος των κόμβων και όπου m το πλήθος των ακμών.

Συνθήκη τερματισμού: $d(s) > \min(\lfloor n^{2/3}m \rfloor, \lfloor m^{1/2} \rfloor)$

Δεύτερη Φάση

Καθώς έχουμε ορίσει τις ετικέτες με την Αναζήτηση Πρώτα κατά Πλάτος χρησιμοποιούμε έναν επιπλέον πίνακα του οποίου οι δείκτες ποικίλουν ανάλογα με τις ετικέτες αποστάσεις. Δηλαδή η τιμή του $numb(k)$ είναι ο αριθμός των κόμβων όπου η ετικέτα τους ισούται με k .

Συνθήκη τερματισμού: Καθώς γίνεται το relabel στις ετικέτες απόστασης, κάθε φορά που αυξάνεται μία ετικέτα ενός κόμβου π.χ. από $k1$ σε $k2$ τότε μειώνεται κατά 1 το $numb(k1)$ και αυξάνεται το $numb(k2)$ κατά 1. Όταν το $numb(k1) = 0$, τότε τερματίζει δημιουργώντας μια ελάχιστη αποκοπή στο γράφημα.

Πειραματική Αξιολόγηση

Μετρήσεις

EDGES	LEDA	MY_FUNCTION
27613	0.03	0.03
27560	0.03	0.06
52654	0.06	0.16
52727	0.04	0.07

Αιτιολόγηση αποτελέσματος

Το 75% του χρόνου του αλγορίθμου καταναλώνεται στην συνάρτηση `phase_b` που εκτελεί την δεύτερη φάση του αλγορίθμου. Πιο συγκεκριμένα αυτό που καθυστερεί την υβριδική συνάρτηση κατασκευής μας είναι η χρησιμοποίηση του `map` που κάνει πιο αργό το πρόγραμμα σε σχέση με το αντίστοιχο της LEDA.

Στιγμιότυπα Εκτέλεσης

```
[ptzolas@diogenis Project_Final]$ ./program
Num of edges = 27613
Flow = 11
TIME: 0.03
flow = 11
duration leda = 0.03
```

```
[ptzolas@diogenis Project_Final]$ ./program
Num of edges = 27525
Flow = 11
TIME: 0.03
flow = 11
duration leda = 0.02
```

```
[ptzolas@diogenis Project_Final]$ ./program
Num of edges = 52654
Flow = 21
TIME: 0.16
flow = 21
duration leda = 0.06
```

```
[ptzolas@diogenis Project_Final]$ ./program
Num of edges = 52727
Flow = 21
TIME: 0.07
flow = 21
duration leda = 0.04
```