

## ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2<sup>η</sup>

ΓΙΑΝΝΑΚΟΥΡΟΣ ΠΕΤΡΟΣ ΑΜ:1084555 ΕΤΟΣ:5ο

### ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ BELLMAN FORD ΣΕ BOOST

Ο βασικός κορμός του αλγορίθμου (εκτέλεση σε φάσεις, χρήση ουράς, έλεγχος αρνητικών κύκλων) παρέμεινε ίδιος. Αυτό που αλλάζει είναι η χρήση των εργαλείων που παρέχει η Boost.

Στη Boost:

- Οι δομές `node_array/edge_array` της LEDA αντικαθίστανται με `std::vector`.
- Οι κόμβοι/ακμές προσπελούνται με `vertex_descriptor`, `edge_descriptor` και `iterators` (`out_edges`, `edges` κ.λπ.).
- Το "nil" node αναπαρίσταται με `boost::graph_traits<Graph>::null_vertex()`.
- Οι τιμές προσπελάσιμων κορυφών κρατούνται με `property maps` ή `vector<bool>`.

Στο τέλος, αφαιρούνται ακμές που δεν ανήκουν στο γράφημα προκατόχων, και γίνεται DFS ώστε να εντοπιστούν κορυφές που επηρεάζονται από αρνητικούς κύκλους. Αν υπάρχουν τέτοιες, επιστρέφεται `false`.

### ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ EVALUATE\_RANDOMGRAPH\_EXPERIMENT

Σε κάθε εκτέλεση, δημιουργείται ένας τυχαίος συνεκτικός κατευθυνόμενος γράφος LEDA με  $n$  κορυφές και  $m=20n\log n$  ακμές. Σε κάθε ακμή αντιστοιχίζεται ένα τυχαίο βάρος από το διάστημα  $[-100, 10000]$ . Ο γράφος αυτός μετατρέπεται σε Boost γράφο μέσω της συνάρτησης LEDA\_Graph\_To\_Boost, η οποία αντιστοιχίζει κάθε κόμβο και ακμή του LEDA γράφου σε αντίστοιχο κόμβο και ακμή στον Boost γράφο, διατηρώντας τα βάρη. Στη συνέχεια, επιλέγεται τυχαία μία αρχική κορυφή  $s$  και εκτελείται πρώτα η δική μου υλοποίηση του Bellman-Ford με Boost, μετρώντας τον χρόνο εκτέλεσης και ελέγχοντας αν υπάρχει αρνητικός κύκλος. Έπειτα, εκτελείται ο Bellman-Ford της LEDA (BELLMAN\_FORD\_B\_T) με τα ίδια δεδομένα και αντίστοιχη μέτρηση χρόνου και ελέγχου για αρνητικό κύκλο. Τέλος, εφόσον δεν έχει ανιχνευθεί αρνητικός κύκλος στη δική μου υλοποίηση, εκτελείται και ο έτοιμος αλγόριθμος `bellman_ford_shortest_paths()` της Boost. Με τον τρόπο αυτό επιτυγχάνεται άμεση σύγκριση ως προς την απόδοση και την ορθότητα των τριών διαφορετικών εκτελέσεων του αλγορίθμου.

### ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ EVALUATE\_GRID\_EXPERIMENT

Αυτός ο κώδικας υλοποιεί την πειραματική αξιολόγηση του αλγορίθμου Bellman-Ford σε ειδικά κατασκευασμένα grid γραφήματα. Αρχικά κατασκευάζεται ένα τετραγωνικό πλέγμα  $n \times n$  με κόμβους(`create_grid_graph`), οι οποίοι αποθηκεύονται σε έναν πίνακα `grid`. Στην περιοχή πάνω αριστερά και στην κάτω δεξιά, οι ακμές έχουν προκαθορισμένη κατεύθυνση από αριστερά προς τα δεξιά και από πάνω προς τα κάτω. Στην κάτω αριστερή περιοχή, οι ακμές αποκτούν τυχαία κατεύθυνση. Επιπλέον, προστίθενται δύο ειδικές ακμές με κόστος  $-100000$ .

Αφού δημιουργηθεί το γράφημα, κάθε ακμή παίρνει βάρος σύμφωνα με τη θέση της(assign\_random\_weights): οι ακμές στις περιοχές 1 και 2 έχουν μη αρνητικά βάρη στο [0, 10000], ενώ στην περιοχή 3 μπορεί να έχουν αρνητικό βάρος. Οι ειδικές ακμές αναγνωρίζονται μέσω συντεταγμένων και λαμβάνουν βάρος -100000. Η αρχική κορυφή είναι η πάνω αριστερή (0,0).

Το γράφημα μετατρέπεται σε Boost και εκτελείται η δική μας υλοποίηση Bellman-Ford με μέτρηση χρόνου και ανίχνευση αρνητικού κύκλου. Στη συνέχεια εκτελείται ο LEDA Bellman-Ford με τα ίδια δεδομένα. Αν δεν εντοπιστεί αρνητικός κύκλος, εκτελείται και η έτοιμη Boost υλοποίηση. Τέλος, εκτυπώνονται οι πρώτες δέκα αποστάσεις από την αρχική κορυφή.

#### ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ(random graph n=1000)

n=1000	ΠΕΙΡ1	ΠΕΙΡ2	ΠΕΙΡ3	ΠΕΙΡ4	ΠΕΙΡ5	<b>M.O</b>
My_BellFord	2.33	1.8	2.03	1.93	1.67	<b>1,95</b>
LEDA	1.3	1.15	1.25	1.29	1.07	<b>1,21</b>
NEG_CYCLE	yes	yes	yes	yes	yes	<b>yes</b>
BOOST	-	-	-	-	-	-

#### ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ(random graph n=4000)

n=4000	ΠΕΙΡ1	ΠΕΙΡ2	ΠΕΙΡ3	ΠΕΙΡ4	ΠΕΙΡ5	<b>M.O</b>
My_BellFord	77.65	60.10	109.33	79.77	64.96	<b>78,36</b>
LEDA	40.51	64.03	50.72	53.31	41.23	<b>49.96</b>
NEG_CYCLE	yes	yes	yes	yes	yes	<b>yes</b>
BOOST	-	-	-	-	-	-

ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ(random graph n=8000)

n=8000	ΠΕΙΡ1	ΠΕΙΡ2	ΠΕΙΡ3	ΠΕΙΡ4	ΠΕΙΡ5	<b>M.O</b>
My_BellFord	320.16	287.15	282.24	259.92	267.77	<b>283.44</b>
LEDA	230.96	246.70	194.23	236.19	212.96	<b>224.20</b>
NEG_CYCLE	yes	yes	yes	yes	yes	<b>yes</b>
BOOST	-	-	-	-	-	-

ΣΧΟΛΙΑΣΜΟΣ ΠΕΙΡΑΜΑΤΙΚΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ RANDOM GRAPH

Στα πειραματικά αποτελέσματα που προέκυψαν από τη χρήση τυχαίων γραφημάτων για τρεις διαφορετικές τιμές πλήθους κορυφών ( $n = 1000, 4000, 8000$ ), παρατηρούμε συνεπή παρουσία αρνητικών κύκλων και στις πέντε επαναλήψεις κάθε περίπτωσης. Γι' αυτόν τον λόγο, η υλοποίηση της Boost δεν εκτελείται. Έτσι, η σύγκριση γίνεται μόνο μεταξύ της δικής μας Boost-based υλοποίησης (My\_BellFord) και της LEDA.

Η LEDA σε όλες τις περιπτώσεις εμφανίζεται σταθερά πιο αποδοτική, με σημαντικά μικρότερους χρόνους εκτέλεσης σε σύγκριση με τη δική μας υλοποίηση. Η διαφορά είναι μικρότερη για  $n = 1000$  (μέσος όρος 1.21 sec έναντι 1,95 sec), αλλά γίνεται εντονότερη όσο αυξάνεται το μέγεθος του γράφου. Για παράδειγμα, στην περίπτωση  $n = 8000$ , οι χρόνοι για τη δική μας υλοποίηση κυμαίνονται μεταξύ 259 και 320 sec(μέσος όρος 283.44), ενώ για τη LEDA μεταξύ 194 και 246 sec(μέσος όρος 224.20).

ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ(grid graph n=100)

n=100	ΠΕΙΡ1	ΠΕΙΡ2	ΠΕΙΡ3	ΠΕΙΡ4	ΠΕΙΡ5	<b>M.O</b>
My_BellFord	0.41	0.47	0.66	0.49	0.42	<b>0.49</b>
LEDA	0.94	1.13	1.49	1.04	0.94	<b>1,10</b>
NEG_CYCLE	yes	yes	yes	yes	yes	<b>yes</b>
BOOST	-	-	-	-	-	-

ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ(grid graph n=200)

n=100	ΠΕΙΡ1	ΠΕΙΡ2	ΠΕΙΡ3	ΠΕΙΡ4	ΠΕΙΡ5	<b>M.O</b>
My_BellFord	10.74	11.38	8.87	9.8	6.97	<b>9.55</b>
LEDA	23.27	25.04	19.16	22.3	16.24	<b>21,2</b>
NEG_CYCLE	yes	yes	yes	yes	yes	<b>yes</b>
BOOST	-	-	-	-	-	-

ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ(grid graph n=300)

n=100	ΠΕΙΡ1	ΠΕΙΡ2	ΠΕΙΡ3	ΠΕΙΡ4	ΠΕΙΡ5	<b>M.O</b>
My_BellFord	88.38	41.19	88.61	51.61	86.68	<b>71.29</b>
LEDA	53.84	33.79	62.09	39.00	55.75	<b>48.89</b>
NEG_CYCLE	yes	yes	yes	yes	yes	<b>yes</b>
BOOST	-	-	-	-	-	-

## ΣΧΟΛΙΑΣΜΟΣ ΠΕΙΡΑΜΑΤΙΚΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ RANDOM GRAPH

Στα πειραματικά αποτελέσματα για grid γράφους μεγέθους  $n=100$ ,  $200$  και  $300$ , παρατηρούμε σταθερή παρουσία αρνητικού κύκλου σε όλες τις εκτελέσεις, γεγονός που οδηγεί σε παράκαμψη της Boost υλοποίησης `bellman_ford_shortest_paths()`. Στο σύνολο των δοκιμών, η δική μας υλοποίηση `My_BellFord` εντοπίζει με συνέπεια τον αρνητικό κύκλο, ενώ παρουσιάζει σημαντικά βελτιωμένες επιδόσεις για το μικρό μέγεθος  $n=100$ , με μέσο χρόνο  $0.49$  δευτερόλεπτα έναντι  $1.10$  της LEDA. Ωστόσο, καθώς αυξάνεται το μέγεθος του grid, ο χρόνος εκτέλεσης της δικής μας υλοποίησης αυξάνεται πιο έντονα. Για  $n=200$ , η `My_BellFord` ολοκληρώνει κατά μέσο όρο σε  $9.55$  δευτερόλεπτα, σχεδόν διπλάσια ταχύτερα από την LEDA που χρειάζεται  $21.2$ . Αντίθετα, στην περίπτωση  $n=300$ , το πλεονέκτημα αντιστρέφεται: η LEDA εμφανίζει καλύτερους χρόνους με μέσο όρο  $48.89$  δευτερόλεπτα, ενώ η δική μας υλοποίηση ανεβαίνει στα  $71.29$ . Η αλλαγή αυτή πιθανότατα οφείλεται στη σταδιακή επιβάρυνση που προκαλεί η μεταγενέστερη φάση του αλγορίθμου μας, η οποία περιλαμβάνει αφαίρεση ακμών και DFS για ανίχνευση του συνόλου  $R$ . Παρ' όλα αυτά, η ακρίβεια της ανίχνευσης είναι υψηλότερη στη δική μας υλοποίηση, καθώς η LEDA δεν εντοπίζει τον αρνητικό κύκλο για  $n=300$ .