



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ



Μαρία Κεραμά

A RANDOMISED ALGORITHM FOR MINIMIZING
 k IN k -PLANE GRID DRAWINGS OF GRAPHS

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων Καθηγητής: Μιχαήλ Μπέκος

Ιωάννινα, 2026

ΚΕΦΑΛΑΙΟ 1

Περιγραφή του Προβλήματος

Είσοδος: Ένα γράφημα $G = (V, E)$.

Έξοδος: Μία k -επίπεδη απεικόνιση του G που ελαχιστοποιεί το k , δηλαδή, μια απεικόνιση στην οποία κάθε ακμή τέμνεται το πολύ k φορές.

Ιδιότητες της παραγόμενης απεικόνισης:

- Οι κόμβοι του γραφήματος τοποθετούνται σε *ακέραιο πλέγμα*.
- Οι ακμές του γραφήματος είναι, αποκλειστικά, *ευθύγραμμα τμήματα* μεταξύ των κόμβων.

Πολυπλοκότητα του Προβλήματος

Το πρόβλημα της εύρεσης μιας k -επίπεδης απεικόνισης ενός γραφήματος είναι ένα NP -πλήρες πρόβλημα, ακόμη και αν $k = 1$.

Κεφάλαιο 1

ΚΕΦΑΛΑΙΟ 2

Έστω Γ μια απεικόνιση του γραφήματος G . Για τον υπολογισμό της k -επιπεδότητας και του συνολικού πλήθους των διασταυρώσεων της απεικόνισης Γ ακολουθούμε τα εξής βήματα.

1. Χαρτογράφηση των ακμών του γραφήματος

Μέσω ενός χάρτη (map), συσχετίζουμε αρχικά κάθε ακμή του γραφήματος με ένα μοναδικό ακέραιο που αντιστοιχεί στο αναγνωριστικό της ακμής αυτής.

Το αναγνωριστικό παρέχει πρόσβαση στο κελί ενός πίνακα, το οποίο περιέχει τις γεωμετρικές πληροφορίες της ακμής, δηλαδή τις συντεταγμένες των άκρων του αντίστοιχου ευθύγραμμου τμήματος.

2. Τα ζεύγη των ακμών που πιθανώς τέμνονται

Έστω e_1 και e_2 δύο ακμές του γραφήματος G .

Έστω b_1 και b_2 δύο ορθογώνια με πλευρές παράλληλες προς τους άξονες που έχουν τις ακμές e_1 και e_2 ως διαγωνίους τους στην απεικόνιση Γ .

Αν $b_1 \cap b_2 \neq \emptyset$, τότε το $\{e_1, e_2\}$ είναι ζεύγος πιθανώς τεμνόμενων ακμών στην απεικόνιση Γ .

Διαφορετικά, οι ακμές e_1 και e_2 δεν μπορεί να τέμνονται στην απεικόνιση Γ .

3. Τα ζεύγη των ακμών που τέμνονται

Έστω (a, b) και (c, d) δύο ακμές, που πιθανώς τέμνονται στην απεικόνιση Γ .

Προφανώς, αν οι (a, b) και (c, d) έχουν κοινό άκρο, τότε δεν τέμνονται στην απεικόνιση Γ .

Οι (a, b) και (c, d) μπορεί να τέμνονται σε ακριβώς ένα εσωτερικό σημείο των δύο ακμών (τομή X) ή μπορεί να δημιουργούν επικάλυψη.

Για να ελέγξουμε αν οι ακμές (a, b) και (c, d) τελικά τέμνονται χρησιμοποιούμε την έννοια του προσανατολισμού ενός σημείου p_k ως προς τα άκρα p_w και p_z ενός

Κεφάλαιο 2

ευθύγραμμου τμήματος:

$$\text{orient}(p_w, p_z, p_k) = (x(p_z) - x(p_w))(y(p_k) - y(p_w)) - (y(p_z) - y(p_w))(x(p_k) - x(p_w))$$

- $\text{orient}(p_w, p_z, p_k) > 0$: το σημείο p_k βρίσκεται αριστερά της ευθείας που διέρχεται από τα p_w και p_z καθώς κινούμαστε από το p_w προς το p_z .
- $\text{orient}(p_w, p_z, p_k) = 0$: το p_k είναι συγγραμμικό με τα p_w και p_z
- $\text{orient}(p_w, p_z, p_k) < 0$: το σημείο p_k βρίσκεται δεξιά της ευθείας που διέρχεται από τα p_w και p_z καθώς κινούμαστε από το p_w προς το p_z .

Με βάση τα παραπάνω μπορούμε να ελέγξουμε αν η ακμή (a, b) τέμνεται από την ακμή (c, d) ελέγχοντας τους προσανατολισμούς $\text{orient}(p_a, p_b, p_c)$, $\text{orient}(p_a, p_b, p_d)$, $\text{orient}(p_c, p_d, p_a)$, $\text{orient}(p_c, p_d, p_b)$, όπου p_a , p_b , p_c και p_d είναι τα σημεία στα οποία βρίσκονται οι κορυφές a , b , c και d , αντίστοιχα, στην απεικόνιση Γ .

4. Οργάνωση των τομών σε δύο πίνακες

Διατηρούμε δύο πίνακες A και B , έτσι ώστε: Ο πίνακας A στη θέση j περιέχει το σύνολο των αναγνωριστικών των ακμών που τέμνουν την ακμή με αναγνωριστικό j , ενώ ο πίνακας B στη θέση j περιέχει το πλήθος των ακμών που τέμνουν την ακμή με αναγνωριστικό j .

Συνεπώς για τον υπολογισμό της k -επιπεδότητας μιας απεικόνισης αρκεί να υπολογίσουμε το παρακάτω μέγιστο:

$$\max_j B[j]$$

Αντίστοιχα, για τον υπολογισμό του συνολικού πλήθους τομών μιας απεικόνισης αρκεί να υπολογίσουμε το παρακάτω άθροισμα:

$$\sum_{j=0}^{|E|-1} B[j]$$

5. Ενημερώσεις μετά την μετακίνηση ενός κόμβου

Ας υποθέσουμε ότι μετακινούμε ένα κόμβο του γραφήματος στην απεικόνιση Γ και λαμβάνουμε έτσι μια νέα απεικόνιση Γ' . Σε αυτή την περίπτωση στόχος μας να μην επανυπολογίσουμε από την αρχή την k -επιπεδότητα και το συνολικό πλήθος τομών της Γ' . Συνεπώς, εστιάζουμε μονάχα σε εκείνες τις ακμές που προσπίπτουν στον κόμβο που μετακινήθηκε και ενημερώνουμε τις αντίστοιχες θέσεις των πινάκων A και B .

ΚΕΦΑΛΑΙΟ 3

Υπολογισμός Αρχικής Απεικόνισης

Για τον υπολογισμό μιας αρχικής απεικόνισης του γραφήματος G χρησιμοποιούμε έναν spring embedder αλγόριθμο που ουσιαστικά είναι ένας αλγόριθμος εύρεσης μιας απεικόνισης ενός γραφήματος βασισμένος σε δυνάμεις (ή προσομοίωσης δυνάμενων). Στον αλγόριθμο αυτό, υπάρχουν δύο σύνολα δυνάμενων.

1. *Απωστικές Δυνάμεις*: Κάθε κόμβος απωθεί κάθε άλλον για την αποφυγή επικαλύψεων κορυφών ή ακμών. Το μέτρο της δύναμης είναι αντιστρόφως ανάλογο της απόστασής τους.
2. *Ελκτικές Δυνάμεις*: Γειτονικοί κόμβοι στο γράφημα έλκονται μεταξύ τους, ώστε να διατηρηθούν κοντά στην απεικόνιση. Το μέτρο της δύναμης είναι ανάλογο της απόστασής τους.

Το σημείο ισορροπίας των δυνάμεων αντιστοιχεί στην αρχική απεικόνιση.

Στην υλοποίησή μας χρησιμοποιήθηκε ο spring embedder αλγόριθμος organic της βιβλιοθήκης yFiles με τις εξής παραμετροποιήσεις:

Παράμετρος	Τιμή
Preferred Edge Length	$\approx \sqrt{\frac{grid\ width \times grid\ height}{ V }}$
Minimum Node Distance	$\approx grid\ scale$
Compactness	$\approx density / (e^{density})$
Gravity	$\approx density / (e^{density})$
Quality Time Ratio	$\approx 1.0 / (e^{density})$
Maximum Duration	$\approx (graph\ size) \times (e^{-density})$
Number of Executions	$\sim \frac{1}{density}, \frac{1}{graph\ size}$

Κεφάλαιο 3

ΚΕΦΑΛΑΙΟ 4

Υπολογισμός Τελικής Απεικόνισης

Μετά τον υπολογισμό της αρχικής απεικόνισης, εφαρμόζουμε ένα τυχαιοκρατικό αλγόριθμο ο οποίος επαναληπτικά βελτιώνει την k -επιπεδότητα της απεικόνισης. Σε κάθε βήμα του, ο αλγόριθμος αυτός με βάση μια υπολογισθείσα απεικόνιση Γ :

1. Υπολογίζει ένα σύνολο υποψήφιων κορυφών προς μετακίνηση.
2. Υπολογίζει με τυχαιοκρατικό τρόπο για κάθε υποψήφια κορυφή προς μετακίνηση μια νέα θέση στην απεικόνιση.
3. Υπολογίζει την επόμενη απεικόνιση Γ' , η οποία ορίζεται μετακινώντας τελικά (το πολύ) μία από τις υποψήφιες κορυφές.
4. Διαχειρίζεται περιπτώσεις εγκλωβισμού σε τοπικά βέλτιστες λύσεις.
5. Αποφαινεται τον τερματισμό της επαναληπτικής διαδικασίας.

Παρακάτω παρουσιάζουμε λεπτομέρειες για καθένα από αυτά τα βήματα του αλγόριθμου.

Βήμα 1

Για τον υπολογισμό του συνόλου των υποψήφιων κόμβων προς μετακίνηση αρχικά χρωματίζουμε ένα υποσύνολο των ακμών του γραφήματος με δύο χρώματα:

- **κόκκινες ακμές:** Οι ακμές που φέρουν το μέγιστο πλήθος διασταυρώσεων στην απεικόνιση Γ .
- **μαύρες ακμές:** Οι ακμές που τέμνουν τις κόκκινες ακμές στην απεικόνιση Γ .

Από το σύνολο των χρωματισμένων ακμών του γραφήματος τελικά επιλέγονται τυχαία 5 κορυφές, οι οποίοι είναι και οι υποψήφιοι κόμβοι προς μετακίνηση.

Κεφάλαιο 4

Βήμα 2

Για κάθε ένα από τους υποψήφιους κόμβους προς μετακίνηση του βήματος 1, υπολογίζονται συνολικά το πολύ 13 θέσεις στις οποίες δύναται να μετακινηθεί ο κόμβος. Πιο συγκεκριμένα, έστω v ένας τέτοιος κόμβος.

Αρχικά, εντοπίζεται το βαρύκεντρο (b_x, b_y) των γειτόνων του κόμβου v στην απεικόνιση Γ . Το βαρύκεντρο και τα παρακάτω τέσσερα σημεία ορίζουν τις πέντε από τις (το πολύ) 13 συνολικά θέσεις στις οποίες δύναται να μετακινηθεί ο κόμβος v :

$$(b_x + 1, b_y), (b_x, b_y + 1), (b_x - 1, b_y), (b_x, b_y - 1).$$

Οι εναπομείναντες (το πολύ) οκτώ θέσεις υπολογίζονται εφαρμόζοντας μια τυχαιοκρατική προσέγγιση. Αρχικά, υπολογίζεται μια ορθογώνια περιοχή με κέντρο τον κόμβο v . Το μήκος (πλάτος) της περιοχής αυτής ισούται με το μισό του μήκους (πλάτους) του πλέγματος της απεικόνισης Γ . Οι οκτώ θέσεις επιλέγονται με τυχαίο τρόπο από την περιοχή αυτή. Αποδεκτές γίνονται μονάχα εκείνες που είτε δεν αυξάνουν το k περισσότερο από μια μονάδα, είτε διατηρούν το k αλλά δεν αυξάνουν το συνολικό πλήθος των διασταυρώσεων.

Από τα δύο παραπάνω βήματα, συλλέγονται το πολύ 13 θέσεις συνολικά, για τον υποψήφιο κόμβο v . Οι θέσεις αυτές ταξινομούνται σε *buckets* ως εξής:

- **Bucket 0:** Περιλαμβάνει τις θέσεις που μειώνουν το k .
- **Bucket 1:** Περιλαμβάνει τις θέσεις που διατηρούν το k και μειώνουν το συνολικό πλήθος των διασταυρώσεων.
- **Bucket 2:** Περιλαμβάνει τις θέσεις που διατηρούν το k και δεν μειώνουν το συνολικό πλήθος των διασταυρώσεων.
- **Bucket 3:** Περιλαμβάνει τις θέσεις που αυξάνουν το k .

Με βάση τα παρακάτω κριτήρια μπορούμε να ιεραρχήσουμε την ποιότητα των 13 αυτών θέσεων. Έστω p και q δύο υποψήφιες θέσεις του κόμβου v .

1. Αν η θέση p βρίσκεται στο bucket i και η θέση q βρίσκεται στο bucket j έτσι ώστε $i < j$, τότε η θέση p προτιμάται της θέσης q .
2. Αν μετακινώντας τον κόμβο v στη θέση p η παραγόμενη απεικόνιση είναι k -επίπεδη, ενώ μετακινώντας τον κόμβο v στη θέση q η παραγόμενη απεικόνιση ℓ -επίπεδη με $k < \ell$, τότε η θέση p προτιμάται της θέσης q .

Κεφάλαιο 4

3. Αν μετακινώντας τον κόμβο v στη θέση p η παραγόμενη απεικόνιση έχει c διασταυρώσεις, ενώ μετακινώντας τον κόμβο v στη θέση q η παραγόμενη απεικόνιση έχει d διασταυρώσεις με $c < d$, τότε η θέση p προτιμάται της θέσης q .
4. Αν μετακινώντας τον κόμβο v στη θέση p η παραγόμενη απεικόνιση έχει συνολικό μήκος ακμών h , ενώ μετακινώντας τον κόμβο v στη θέση q η παραγόμενη απεικόνιση έχει συνολικό μήκος ακμών g με $h < g$, τότε η θέση p προτιμάται της θέσης q .
5. Διαφορετικά, επιλέγεται με ομοιόμορφο τυχαίο τρόπο μια από τις δύο θέσεις ως ποιοτικότερη.

Βήμα 3

Για καθέναν από τους πέντε υποψήφιους κόμβους προς μετακίνηση που όρισε ο αλγόριθμος στο Βήμα 1, επιλέγεται και μετακινείται ο κόμβος εκείνος που οδηγεί στην μεγαλύτερη μείωση της k -επιπεδότητας της απεικόνισης Γ και με τον τρόπο αυτό ορίζεται η επόμενη απεικόνιση Γ' . Σημειώνεται ότι μια τέτοια μετακίνηση μπορεί να μην είναι εφικτή. Σε αυτό το σενάριο, $\Gamma' = \Gamma$. Επίσης, προσμετράμε και πόσες διαδοχικές επαναλήψεις του αλγορίθμου μπορεί να μην οδηγήσουν σε μετακίνηση, καθώς αυτές αποτελούν ένδειξη ότι ίσως ο αλγόριθμος να έχει εγκλωβιστεί σε μια τοπικά βέλτιστη λύση.

Βήμα 4

Αν ο αλγόριθμος για ένα δεδομένο αριθμό διαδοχικών επαναλήψεων (που εξαρτάται από το πόσο μεγάλο ή/και πυκνό είναι το γράφημα) δεν μετακινήσει κάποιο κόμβο, τότε η κάθε επανάληψη του αλγορίθμου τροποποιείται ως εξής:

- αυξάνεται ο αριθμός των υποψήφιων κόμβων προς μετακίνηση,
- αυξάνεται ο αριθμός των πιθανών θέσεων μετακίνησης καθενός υποψηφίου,
- διευρύνονται τα όρια της αναζήτησης υποψήφιων θέσεων μετακίνησης σε όλο το υποκείμενο αχέραιο πλέγμα,
- ο αλγόριθμος δύναται να αποδεχθεί επόμενες απεικονίσεις οι οποίες αυξάνουν την k -επιπεδότητά της επόμενης απεικόνισης περισσότερο από μια μονάδα.

Κεφάλαιο 4

Βήμα 5

Ο αλγόριθμος τερματίζει είτε όταν το αποφανθεί ο χρήστης είτε όταν ο συνολικός χρόνος εκτέλεσης του αλγορίθμου φτάσει μια προκαθορισμένη σταθερά είτε όταν μετά από ένα μεγάλο αριθμό επαναλήψεων δε βελτιωθεί η k -επιπεδότητα της απεικόνισης. Σε αυτή την περίπτωση επιστρέφεται η καλύτερη απεικόνιση που έχει υπολογιστεί μέχρι αυτό το σημείο και λειτουργεί ως αφετηρία για (πιθανές) επόμενες εκτελέσεις του τυχαιοκρατικού αλγορίθμου.

ΚΕΦΑΛΑΙΟ 5

Αξιολόγηση του Αλγορίθμου

Το έναυσμα για να δουλέψουμε στο πρόβλημα της ελαχιστοποίησης της k -επιπεδότητας ενός γραφήματος αποτέλεσε ο διαγωνισμός Live Challenge: Automatic category του συνεδρίου Graph Drawing and Network Visualization 2026.

Σημειώνεται ότι το 2025 ο διαγωνισμός είχε το ίδιο θέμα στην συγκεκριμένη κατηγορία, στην οποία έλαβαν μέρος συνολικό οκτώ ομάδες. Καθεμία από τις ομάδες κλήθηκε να εφαρμόσει τον αλγόριθμό της σε ένα σύνολο εννέα γραφημάτων, έχοντας συνολικά μια ώρα στη διάθεση της. Τα γραφήματα ήταν κοινά για όλες τις ομάδες. Για καθένα από αυτά, παρατίθεται η καλύτερη λύση που προτάθηκε στο διαγωνισμό και η αντίστοιχη δική μας που προέκυψε με εφαρμογή του αλγορίθμου μας στο σύνολο των γραφημάτων αυτών με τον ίδιο χρονικό περιορισμό.

Γράφημα	# κόμβων	# ακμών	GD2025	Ο αλγόριθμός μας
1	50	202	9	9
2	144	314	3	4
3	100	420	34	48
4	196	571	6	8
5	500	2556	76	83
6	200	3000	560	749
7	500	1740	31	32
8	10466	20288	15	10
9	2519	4938	12	15

Όπως φαίνεται από τον παραπάνω πίνακα ο αλγόριθμος που παρουσιάστηκε σε αυτή την εργασία ήταν ιδιαίτερα ανταγωνιστικός.