

Εργασία Κωδικοποίησης: Αναφορά Υλοποίησης & Ανάλυσης

Θέμα: Υλοποίηση Μηχανισμού Ομαδοποίησης Κόμβων Γραφήματος

Ονοματεπώνυμο: [Το Όνομά μου]

Ημερομηνία: 24/05/2024

1. Εισαγωγή & Στόχος Εργασίας

Η παρούσα αναφορά περιγράφει τον σχεδιασμό και την ανάλυση ενός αλγορίθμου για την ομαδοποίηση κόμβων (graph clustering) σε ένα δεδομένο μη κατευθυνόμενο και χωρίς βάρη γράφημα. Ο στόχος είναι ο αναδρομικός διαμερισμός του γράφου σε μικρότερες, συνεκτικές κοινότητες, βασιζόμενος στην αρχή της ελάχιστης τομής (minimum cut).

2. Θεμελίωση: Δομές Δεδομένων

Για την αναπαράσταση του γράφου, επιλέχθηκε η δομή της **Λίστας Γειτνίασης (Adjacency List)**.

- Αιτιολόγηση:** Για αραιούς (sparse) γράφους, όπως οι περισσότεροι πραγματικοί (συμπεριλαμβανομένου του `karateclub.txt`), η λίστα γειτνίασης προσφέρει βέλτιστη απόδοση σε χώρο ($O(|V| + |E|)$) και χρόνο για τις απαραίτητες λειτουργίες, όπως η εύρεση γειτόνων ενός κόμβου.

3. Αναλυτικό Σχέδιο Υλοποίησης & Ανάλυση Πολυπλοκότητας

Η υλοποίηση ακολουθεί τα βήματα που ορίζονται στην εκφώνηση.

Βήμα A: Έλεγχος Συνδεσιμότητας και Απομόνωση Μεγαλύτερης Συνιστώσας

- Διαδικασία:** Πριν την ανάλυση, ο αλγόριθμος εντοπίζει όλες τις συνεκτικές συνιστώσες του αρχικού γράφου. Αυτό επιτυγχάνεται με τη διαδοχική εκτέλεση του αλγορίθμου **Αναζήτησης κατά Πλάτος (BFS)**, ξεκινώντας από κόμβους που δεν έχουν ήδη εξερευνηθεί. Στη συνέχεια, επιλέγεται η συνιστώσα με το μεγαλύτερο πλήθος κόμβων για περαιτέρω επεξεργασία.
- Πολυπλοκότητα:** $O(|V| + |E|)$
- Ανάλυση:** Η διαδικασία απαιτεί την επίσκεψη κάθε κόμβου ($|V|$) και κάθε ακμής ($|E|$) του γράφου ακριβώς μία φορά για την εύρεση όλων των συνιστωσών. Επομένως, το κόστος είναι γραμμικό ως προς το μέγεθος του γράφου.

Βήμα B: Υπολογισμός Minimum-Cut & Διαμερισμός

Αυτό είναι το υπολογιστικά πιο απαιτητικό βήμα, το οποίο αποτελείται από δύο διακριτά μέρη.

1. Εύρεση Πηγής (s) και Καταβόθρας (t):

- Στρατηγική:** Ως s και t επιλέγονται τα δύο άκρα της **διαμέτρου** του γράφου, δηλαδή οι δύο πιο απομακρυσμένοι κόμβοι. Αυτό γίνεται με την ευριστική ότι η τομή μεταξύ τους είναι

πιθανότερο να αποκαλύψει μια φυσική διαίρεση του δικτύου.

- **Πολυπλοκότητα:** $O(|V| * (|V| + |E|))$
- **Ανάλυση:** Για τον υπολογισμό της διαμέτρου, εκτελείται ο αλγόριθμος BFS από κάθε κόμβο του γράφου ($|V|$ φορές) για να βρεθούν οι αποστάσεις του προς όλους τους άλλους. Δεδομένου ότι μια εκτέλεση BFS κοστίζει $O(|V| + |E|)$, το συνολικό κόστος είναι $|V| * O(|V| + |E|)$.

2. Υπολογισμός Ελάχιστης Τομής (Min-Cut):

- **Αλγόριθμος:** Χρησιμοποιείται ο αλγόριθμος **Edmonds-Karp** για την εύρεση της μέγιστης ροής (max-flow) από το s στο t . Βάσει του θεωρήματος max-flow min-cut, η τιμή αυτή είναι ίση με τη χωρητικότητα της ελάχιστης τομής. Ο διαμερισμός προκύπτει από το σύνολο των κόμβων που είναι προσβάσιμοι από το s στο τελικό υπολειμματικό γράφημα.
- **Πολυπλοκότητα:** $O(|V| * |E|^2)$
- **Ανάλυση:** Ο αλγόριθμος Edmonds-Karp βρίσκει επαναληπτικά μονοπάτια αύξησης με BFS (κόστος $O(|E|)$) στο υπολειμματικό γράφημα. Ο αριθμός των επαναλήψεων είναι φραγμένος από $O(|V| |E|)$. Το γινόμενο αυτών των δύο όρων δίνει τη συνολική πολυπλοκότητα.

Συνολική Πολυπλοκότητα για έναν Διαμερισμό: Η πολυπλοκότητα του Βήματος B κυριαρχείται από τον υπολογισμό του Edmonds-Karp, άρα είναι $O(|V| |E|^2)$.

Βήμα Γ: Αναδρομική Διαδικασία και Συνθήκη Τερματισμού

- **Διαδικασία:** Το Βήμα B εφαρμόζεται αναδρομικά στα υπογραφήματα που προκύπτουν από κάθε διαμερισμό.
- **Συνθήκη Τερματισμού:** Η αναδρομή σταματά όταν μια συνιστώσα:
 1. Έχει μέγεθος μικρότερο από ένα κατώφλι N_{min} (π.χ., 3).
 2. Αποτελεί έναν πλήρη γράφο (κλίκα), δηλαδή μια απόλυτα συνεκτική κοινότητα.
- **Συνολική Πολυπλοκότητα Αλγορίθμου:** Η πολυπλοκότητα ολόκληρης της διαδικασίας καθορίζεται σχεδόν εξ ολοκλήρου από τον **πρώτο διαμερισμό** που εκτελείται στον αρχικό, μεγάλο γράφο. Οι επόμενες αναδρομικές κλήσεις σε πολύ μικρότερα υπογραφήματα έχουν αμελητέο κόστος σε σύγκριση. Επομένως, η πολυπλοκότητα του συνολικού αλγορίθμου εκτιμάται ως $O(|V| * |E|^2)$, όπου $|V|$ και $|E|$ αφορούν το μέγεθος της μεγαλύτερης συνεκτικής συνιστώσας του αρχικού γράφου.

4. Ψευδοκώδικας Αλγορίθμων

Κύριος Αναδρομικός Αλγόριθμος

```

ΣΥΝΑΡΤΗΣΗ Αναδρομική_Ομαδοποίηση(Λίστα_Κόμβων C, Γράφος G_αρχικός):
  Υπογράφημα_C = Δημιουργήσε_Υπογράφημα(G_αρχικός, C)

  // Συνθήκες Τερματισμού
  ΑΝ |C| < N_min Η' το Υπογράφημα_C είναι κλίκα:
    ΕΠΙΣΤΡΕΨΕ [C] // Επιστρέφει μια λίστα που περιέχει αυτό το τελικό cluster

  // Αναδρομικό Βήμα
  (Partition_1, Partition_2) = Διαμέρισε_τον_Γράφο(Υπογράφημα_C)

```

```

Clusters_1 = Αναδρομική_Ομαδοποίηση(Partition_1, G_αρχικός)
Clusters_2 = Αναδρομική_Ομαδοποίηση(Partition_2, G_αρχικός)

ΕΠΙΣΤΡΕΨΕ Clusters_1 + Clusters_2

```

Συνάρτηση Διαμερισμού (με ενσωματωμένη εύρεση διαμέτρου)

```

ΣΥΝΑΡΤΗΣΗ Διαμέρισε_τον_Γράφο(Γράφος G):
// -- Φάση 1: Εύρεση των άκρων της διαμέτρου (επιλογή s και t) --
μέγιστη_απόσταση = -1
πηγή_s, καταβόθρας_t = ΚΑΝΕΝΑΣ, ΚΑΝΕΝΑΣ
ΓΙΑ ΚΑΘΕ κόμβο u ΣΤΟΝ G:
    αποστάσεις_από_u = BFS_για_αποστάσεις(G, u)[0] // [0] για το λεξικό αποστάσεων
    ΓΙΑ ΚΑΘΕ (κόμβο v, απόσταση d) ΣΤΟ αποστάσεις_από_u:
        ΑΝ d > μέγιστη_απόσταση:
            μέγιστη_απόσταση = d
            πηγή_s = u
            καταβόθρας_t = v

// -- Φάση 2: Υπολογισμός τομής με βάση τα s και t --
G_flow = Μετέτρεψε_σε_Γράφο_Ροής(G)
(τιμή_ροής, Υπολειμματικό_Γράφημα) = Edmonds_Karp(G_flow, πηγή_s, καταβόθρας_t)
Ομάδα_1 = οι κόμβοι προσβάσιμοι από την `πηγή_s` στο `Υπολειμματικό_Γράφημα`
(μέσω BFS)
Ομάδα_2 = όλοι οι υπόλοιποι κόμβοι του G
ΕΠΙΣΤΡΕΨΕ (Ομάδα_1, Ομάδα_2)

```

Αλγόριθμος BFS

```

ΣΥΝΑΡΤΗΣΗ BFS_για_αποστάσεις(Γράφος G, Κόμβος αρχής):
    Ουρά Q = νέα ουρά
    αποστάσεις = νέο λεξικό
    γονείς = νέο λεξικό

    πρόσθεσε τον 'αρχής' στην Q
    αποστάσεις[αρχής] = 0
    γονείς[αρχής] = ΚΑΝΕΝΑΣ

    ΟΣΟ η Q ΔΕΝ είναι κενή:
        τρέχων_κόμβος = αφαίρεσε από την Q
        ΓΙΑ ΚΑΘΕ γείτονα v ΤΟΥ τρέχων_κόμβος:
            ΑΝ ο v ΔΕΝ είναι στο 'αποστάσεις':
                πρόσθεσε τον v στην Q
                αποστάσεις[v] = αποστάσεις[τρέχων_κόμβος] + 1
                γονείς[v] = τρέχων_κόμβος
    ΕΠΙΣΤΡΕΨΕ (αποστάσεις, γονείς)

```

Αλγόριθμος Edmonds-Karp

```
ΣΥΝΑΡΤΗΣΗ Edmonds_Karp(Γράφος G, Κόμβος s, Κόμβος t):
  Υπολειμματικό_Γράφημα = αντίγραφο του G
  μέγιστη_ροή = 0
  ΠΑΝΤΑ:
    (αποστάσεις, γονείς) = BFS_για_αποστάσεις(Υπολειμματικό_Γράφημα, s)
    ΑΝ το t ΔΕΝ είναι στο 'γονείς':
      ΔΙΑΚΟΨΕ

    ροή_μονοπατιού = ΑΠΕΙΡΟ
    κόμβος = t
    ΟΣΟ ο κόμβος != s:
      προηγούμενος = γονείς[κόμβος]
      ροή_μονοπατιού = MIN(ροή_μονοπατιού,
        Υπολειμματικό_Γράφημα.χωρητικότητα(προηγούμενος, κόμβος))
      κόμβος = προηγούμενος

    κόμβος = s
    ΟΣΟ ο κόμβος != s:
      προηγούμενος = γονείς[κόμβος]
      Υπολειμματικό_Γράφημα.μείωσε_χωρητικότητα(προηγούμενος, κόμβος,
        ροή_μονοπατιού)
      Υπολειμματικό_Γράφημα.αύξησε_χωρητικότητα(κόμβος, προηγούμενος,
        ροή_μονοπατιού)
      κόμβος = προηγούμενος

    μέγιστη_ροή += ροή_μονοπατιού

  ΕΠΙΣΤΡΕΨΕ (μέγιστη_ροή, Υπολειμματικό_Γράφημα)
```