

ΜΕΛΕΤΗ ΑΛΓΟΡΙΘΜΩΝ ΕΛΑΧΙΣΤΗΣ ΤΟΜΗΣ ΚΑΙ  
ΕΦΑΡΜΟΓΗ ΤΟΥΣ ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΚΑΤΑΤΜΗΣΗΣ  
ΕΙΚΟΝΑΣ

Η ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

υποβάλλεται στην

ορισθείσα από την Γενική Συνέλευση Ειδικής Σύθεσης  
του Τμήματος Πληροφορικής Εξεταστική Επιτροπή

από την

Αυγούστα Θεοδούλου

ως μέρος των Υποχρεώσεων για τη λήψη του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ  
ΜΕ ΕΞΕΙΔΙΚΕΥΣΗ  
ΣΤΗ ΘΕΩΡΙΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ιούνιος 2008

# ΑΦΙΕΡΩΣΗ

---

Στην αγαπημένη μου γιαγιά που “έφυγε”.

# ΕΥΧΑΡΙΣΤΙΕΣ

---

Πρωτίστως θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, κ. Σταύρο Νικολόπουλο, για τη πολύτιμη βοήθεια του.

Επίσης ένα μεγάλο ευχαριστώ οφείλω στους κ. Λεωνίδα Παληό και κ. Χάρη Παπαδόπουλο γιατί ήταν πάντα πρόθυμοι να προσφέρουν βοήθεια σε οποιαδήποτε δυσκολία.

Ιδιαίτερος να ευχαριστήσω τον Αργύρη Καλογεράτο για τις ενδιαφέρουσες συζητήσεις που είχαμε οι οποίες με βοήθησαν σε διάφορα ζητήματα που με απασχόλησαν στα πλαίσια της εργασίας. Θα ήθελα να ευχαριστήσω επίσης τους Γιώργο Μαργαρίτη, Δημήτρη Γερογιάννη και Κωνσταντίνα Λουκοπούλου για την ενθάρρυνση, τη συμπαράσταση και την ηθική υποστήριξη που μου παρείχαν.

Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου για την ηθική, ψυχολογική και οικονομική υποστήριξη που μου παρείχαν όλα αυτά τα χρόνια.

# ΠΕΡΙΕΧΟΜΕΝΑ

---

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Εισαγωγή</b>                                  | <b>1</b>  |
| 1.1      | Το πρόβλημα                                      | 1         |
| 1.2      | Υπόβαθρο και Βασικές έννοιες                     | 2         |
| 1.2.1    | Γραφήματα και Τομές                              | 3         |
| 1.2.2    | Συρρίκνωση Ακμής $uw$ - Συρρίκνωση Κόμβων $u, w$ | 3         |
| 1.2.3    | Δίκτυο   | 4         |
| 1.2.4    | Δίκτυα Ροής                                      | 4         |
| 1.2.5    | Εναπομένον Γράφημα (Residual Network)            | 5         |
| 1.2.6    | Αλγόριθμος Prim Εύρεσης Maximal Spanning Tree    | 7         |
| 1.3      | Πρόβλημα Ελάχιστης Τομής                         | 7         |
| 1.4      | Ομαδοποίηση με Διαμέριση Γραφήματος              | 8         |
| 1.5      | Συμβολή  | 11        |
| 1.6      | Διάρθρωση Μεταπτυχιακής Εργασίας                 | 13        |
| <b>2</b> | <b>Προβλήματα Μέγιστης Ροής</b>                  | <b>14</b> |
| 2.1      | Ορισμός του προβλήματος                          | 14        |
| 2.2      | Μέθοδοι Μέγιστης Ροής                            | 17        |
| 2.2.1    | Αλγόριθμος Ford-Fulkerson                        | 17        |
| 2.2.2    | Αλγόριθμος Edmonds-Karp                          | 20        |
| 2.2.3    | Αλγόριθμος Push-Relabel                          | 21        |
| 2.3      | Μέθοδοι Δημιουργίας Δέντρων Τομής                | 24        |
| 2.3.1    | Αλγόριθμος Gomory-Hu                             | 25        |
| <b>3</b> | <b>Αλγόριθμοι Εύρεσης Ελάχιστης Τομής</b>        | <b>30</b> |
| 3.1      | Αλγόριθμος Nagamochi-Ibaraki                     | 31        |
| 3.2      | Αλγόριθμος Karger-Stein                          | 34        |
| 3.3      | Αλγόριθμος Padberg-Rinaldi                       | 35        |

|          |  |           |
|----------|--|-----------|
| 3.4      | Αλγόριθμος Stoer-Wagner . . . . .                                    | 38        |
| 3.5      | Αλγόριθμος Brinkmeier . . . . .                                      | 41        |
| <b>4</b> | <b>Ομαδοποίηση-Κατάτμηση Εικόνας</b>                                 | <b>44</b> |
| 4.1      | Μέθοδοι ομαδοποίησης . . . . .                                       | 44        |
| 4.1.1    | Η διαδικασία ομαδοποίησης . . . . .                                  | 44        |
| 4.1.2    | Τεχνικές ομαδοποίησης . . . . .                                      | 45        |
| 4.1.3    | Επιλογή κατάλληλης μεθόδου ομαδοποίησης . . . . .                    | 45        |
| 4.1.4    | Μέτρα ομοιότητας . . . . .   | 46        |
| 4.1.5    | Αλγόριθμος K-μέσων . . . . .   | 47        |
| 4.1.6    | Συσσωρευτικός Αλγόριθμος Ομαδοποίησης . . . . .                      | 47        |
| 4.2      | Κατάτμηση Εικόνας . . . . .  | 49        |
| 4.2.1    | Μέθοδος Wu-Leahy . . . . .   | 50        |
| 4.2.2    | Μέθοδος Shi-Malic . . . . .  | 51        |
| 4.2.3    | Μέθοδος Scanlon-Deo . . . . .  | 52        |
| 4.2.4    | Κατάτμηση Εικόνας με μεθόδους Ομαδοποίησης . . . . .                 | 53        |
| <b>5</b> | <b>Εφαρμογή του Αλγορίθμου Stoer-Wagner στην Κατάτμηση Εικόνας</b>   | <b>55</b> |
| 5.1      | Αλγόριθμος των Stoer-Wagner . . . . .                                | 55        |
| 5.2      | Εφαρμογή Αλγορίθμου SW στην Κατάτμηση Εικόνας . . . . .              | 59        |
| 5.2.1    | Αναπαράσταση της Εικόνας με Γράφημα . . . . .                        | 60        |
| 5.2.2    | Καθορισμός Συνάρτησης βαρών . . . . .                                | 61        |
| 5.2.3    | Επίλυση του Προβλήματος του Κριτηρίου Ελάχιστης τομής . . . . .      | 63        |
| 5.2.4    | Επιλογή Κατάλληλου Υπογραφήματος προς Διαμέριση . . . . .            | 64        |
| 5.2.5    | Μερική Συνένωση των Ομάδων με Αλγόριθμο MCT (MIN CUT TREE) . . . . . | 66        |
| <b>6</b> | <b>Αλγόριθμος Ομαδοποίησης με Δέντρα Ελάχιστης Αποκοπής</b>          | <b>68</b> |
| 6.1      | Βασική Ιδέα . . . . .  | 68        |
| 6.2      | Επιλογή $\alpha$ . . . . .   | 73        |
| 6.3      | Ευριστική Μέθοδος για τον Αλγόριθμο Cut-clustering . . . . .         | 73        |
| 6.4      | Ιεραρχικός Αλγόριθμος Cut-clustering . . . . .                       | 75        |
| 6.4.1    | Παράδειγμα Ιεραρχικού Αλγορίθμου Cut-Clustering . . . . .            | 76        |
| <b>7</b> | <b>Πειραματικά αποτελέσματα</b>                                      | <b>80</b> |
| 7.1      | Περιγραφή πειραμάτων . . . . .                                       | 80        |

|          |  |            |
|----------|--|------------|
| 7.2      | Πειράματα . . . . .                      | 82         |
| <b>8</b> | <b>Συμπεράσματα - Μελλοντική εργασία</b> | <b>103</b> |
| 8.1      | Συμπεράσματα . . . . .                   | 103        |
| 8.2      | Μελλοντική εργασία . . . . .             | 104        |

# ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

---

|      |   |    |
|------|---|----|
| 1.1  | Αριστερά φαίνεται το γράφημα $G$ . Έστω $ab$ ακμή που πρόκειται να συρρικνωθεί. Δεξιά φαίνεται το γράφημα $G/\{a, b\}$ που προκύπτει μετά τη συρρίκνωση.  | 4  |
| 1.2  | Η ακμή $ij$ του γραφήματος $N$ . . . . .  | 6  |
| 1.3  | Πίνακας: Μια ακμή $a$ ανήκει στο γράφημα $N_r$ αν ικανοποιούνται οι αντίστοιχες συνθήκες. Κάθε ακμή $ij \in A(N_r)$ χαρακτηρίζεται με την ποσότητα $r(i, j)$ ανάλογα με τη συνθήκη που ικανοποιείται. . . . . | 6  |
| 1.4  | Αριστερά φαίνεται το αρχικό γράφημα ροής $N$ και δεξιά το αντίστοιχο εναπομένον $N_r$ . . . . .   | 6  |
| 1.5  | Ο κόμβος $u$ με επιγραφή $d(u) = 2$ αποτελεί έναν ενεργό κόμβο . . . . .  | 7  |
| 1.6  | (a) Παράδειγμα ομαδοποίησης με κριτήριο ομαδοποίησης την απόσταση των δεδομένων της αρχικής συλλογής. . . . .   | 9  |
| 2.1  | Νόμιμη ροή τιμής 3. . . . .   | 15 |
| 2.2  | Παράδειγμα δικτύου $N$ . . . . .  | 19 |
| 2.3  | Στο ακόρεστο μονοπάτι $P_1 = \{s, v_3, v_4, v_2, t\}$ περνά ροή ίση με $\Delta = 2$ . . . . .   | 19 |
| 2.4  | Στο ακόρεστο μονοπάτι $P_2 = \{s, v_1, v_2, v_4, t\}$ περνά ροή ίση με $\Delta = 2$ . . . . .   | 19 |
| 2.5  | Στο ακόρεστο μονοπάτι $P_3 = \{s, v_1, v_2, t\}$ περνά ροή ίση με $\Delta = 1$ . . . . .  | 19 |
| 2.6  | Στο ακόρεστο μονοπάτι $P_4 = \{s, v_3, v_4, t\}$ περνά ροή ίση με $\Delta = 1$ . . . . .  | 20 |
| 2.7  | Αντιπαράδειγμα εφαρμογής μεθόδου Ford-Fulkerson . . . . .   | 20 |
| 2.8  | Ανάθεση επιγραφών τους κόμβους του $N_r$ . . . . .  | 21 |
| 2.9  | Αρχικοποίηση των επιγραφών των κόμβων του $N_r$ , και προώθηση ροής από τον κόμβο πηγή $s$ στους γείτονες του. . . . .  | 23 |
| 2.10 | Ο σκιαγραφημένος ενεργός κόμβος, προωθεί ροή $f=2$ προς τον κόμβο προορισμό $t$ . . . . .   | 23 |
| 2.11 | Ο σκιαγραφημένος ενεργός κόμβος, προωθεί ροή $f=3$ προς τον κόμβο προορισμό $t$ . . . . .   | 24 |
| 2.12 | Ο σκιαγραφημένος ενεργός κόμβος επαναπροσδιορίζει την επιγραφή του. . . . .   | 24 |

|      |  |    |
|------|--|----|
| 2.13 | Ο σκιαγραφημένος ενεργός κόμβος, προωθεί ροή $f=3$ προς την πηγή. . . . .  | 24 |
| 2.14 | Πρώτο βήμα του Gomory-Hu . . . . .   | 26 |
| 2.15 | Δεύτερο βήμα του Gomory-Hu . . . . .   | 27 |
| 2.16 | Γενικό στάδιο του Gomory-Hu . . . . .  | 27 |
| 2.17 | Γενικό βήμα του Gomory-Hu . . . . .  | 28 |
| 2.18 | Αρχικό γράφημα και το αντίστοιχο Gomory-Hu δέντρο ελάχιστης τομής . .  | 28 |
| 3.1  | Τεχνικές αλγορίθμων που επιλύουν το πρόβλημα ελάχιστης τομής. . . . .  | 31 |
| 4.1  | Παράδειγμα στο οποίο η ελαχιστοποίηση του κριτηρίου $SCut$ οδηγεί σε μια κακή διαμέριση του γραφήματος. . . . .  | 51 |
| 5.1  | Το γράφημα $G$ στο οποίο πρόκειται να εφαρμόσουμε τον αλγόριθμο SW. . .  | 56 |
| 5.2  | Το γράφημα $G\{s, t\}$ το οποίο προκύπτει μετά τη συρρίκνωση των κόμβων $s, t$ .   | 57 |
| 5.3  | Το γράφημα που προκύπτει μετά τη δεύτερη εκτέλεση της MinCutPhase. . .   | 57 |
| 5.4  | Το γράφημα που προκύπτει μετά την τρίτη εκτέλεση της MinCutPhase. . . .  | 58 |
| 5.5  | Το γράφημα που προκύπτει μετά την τέταρτη εκτέλεση της MinCutPhase. . .  | 58 |
| 5.6  | Το γράφημα που προκύπτει μετά την πέμπτη εκτέλεση της MinCutPhase. . .   | 58 |
| 5.7  | Το γράφημα που προκύπτει μετά την έκτη εκτέλεση της MinCutPhase. . . .   | 58 |
| 5.8  | Η γειτονιά των εικοστοιχείων $(i, j)$ και $(i, j + 1)$ που συμμετέχει για τον καθορισμό του βάρους της ακμής που συνδέει τους αντίστοιχους κόμβους τους. . . . . | 62 |
| 5.9  | Η γειτονιά των εικοστοιχείων $(i, j)$ και $(i + 1, j)$ που συμμετέχει για τον καθορισμό του βάρους της ακμής που συνδέει τους αντίστοιχους κόμβους τους. . . . . | 62 |
| 6.1  | Αρχικό γράφημα $G$ και το αντίστοιχο $G_\alpha$ μετά την προσθήκη του τεχνητού κόμβου $t$ . . . . .  | 76 |
| 6.2  | Δέντρο Ελάχιστης Τομής $T_{G_\alpha}$ για τιμές του $\alpha$ στο διάστημα $3 \leq \alpha \leq \infty$ . . .  | 76 |
| 6.3  | Δέντρο Ελάχιστης Τομής $T_{G_\alpha}$ για $\alpha = 2$ . . . . .   | 77 |
| 6.4  | Δέντρο Ελάχιστης Τομής $T_{G_\alpha}$ για $\alpha = 1$ . . . . .   | 77 |
| 6.5  | Δέντρο Ελάχιστης Τομής $T_{G_\alpha}$ για $\alpha = 0$ . . . . .   | 77 |
| 7.1  | Αρχική εικόνα. . . . .   | 84 |
| 7.2  | a) ModHSW-MyCap και $M=30$ . b) ModHSW-MyCap και $M=60$ . c) HSW-SCap και $M=60$ , d) HSW-MyCap και $M=60$ . . . . .   | 84 |



|      |  |    |
|------|--|----|
| 7.3  | Αριστερά MCT-SCap και M=60. Δεξιά MCT-MyCap και M=60. . . . .  | 85 |
| 7.4  | Αριστερά MCT- $\alpha$ , $\alpha = 20$ και M=64. Δεξιά MCT- $\alpha$ $\alpha = 46$ και M=161. . .  | 85 |
| 7.5  | ModHSW-MyCap, M=15. . . . .  | 86 |
| 7.6  | ModHSW-MyCap, M=15. Ορίζοντας κατώφλι $t = 0$ κατά τη διάρκεια της κατασκευής της διάταξης συνδεσιμότητας. . . . .                           | 86 |
| 7.7  | Αρχική εικόνα. . . . .   | 87 |
| 7.8  | Αριστερά MCT-SCap και M=4. Δεξιά MCT-MyCap και M=7. . . . .  | 87 |
| 7.9  | Αριστερά ModHSW-SCap και M=4. Δεξιά MoHdSW-MyCap και M=4. . .  | 87 |
| 7.10 | Αριστερά MCT- $\alpha$ -SCap, $\alpha = 20$ και M=4. Δεξιά MCT- $\alpha$ -SCap, $\alpha = 1$ και M=3. . . . .                                | 88 |
| 7.11 | Αρχική εικόνα. . . . .   | 88 |
| 7.12 | Αριστερά MCT-SCap και M=400. Δεξιά ModHSW-MyCap και M=115. . .   | 88 |
| 7.13 | Αριστερά MCT-SCap, και M=1500. Δεξιά ModHSW-MyCap και M=115. .   | 89 |
| 7.14 | Αριστερά MCT- $\alpha$ -SCap, $\alpha = 30$ και M=1400. Δεξιά ModHSW-MyCap και M=115. . . . .  | 89 |
| 7.15 | Αριστερά MCT- $\alpha$ -SCap, $\alpha = 20$ και M=171. Δεξιά ModHSW-MyCap και M=115. . . . .   | 90 |
| 7.16 | Αρχική εικόνα. . . . .   | 90 |
| 7.17 | Αριστερά MCT-SCap, και M=100. Δεξιά ModHSW-MyCap και M=100 στο αποτέλεσμα του οποίου εφαρμόσαμε τη διαδικασία MCT-SCap και M=10.             | 91 |
| 7.18 | Αριστερά ModHSW-MyCap, και M=100. Δεξιά ModHSW-MyCap και M=100 στο αποτέλεσμα του οποίου εφαρμόσαμε τη διαδικασία MCT-SCap και M=10. . . . . | 91 |
| 7.19 | Αριστερά ModHSW-MyCap, και M=10. Δεξιά ModHSW-MyCap και M=100 στο αποτέλεσμα του οποίου εφαρμόσαμε τη διαδικασία MCT-SCap και M=10.          | 91 |
| 7.20 | Αρχική εικόνα. Εκτέλεση ModHSW-MyCap, και M=4. Εκτέλεση ModHSW-MyCap και M=2. . . . .  | 93 |
| 7.21 | Αρχική εικόνα. . . . .   | 93 |
| 7.22 | Εκτέλεση ModHSW-MyCap με M=2. Εκτέλεση ModHSW-MyCap με M=3. Εκτέλεση ModHSW-MyCap με M=4. . . . .  | 93 |
| 7.23 | Εκτέλεση HSW-MyCap με M=2. Εκτέλεση HSW-MyCap με M=3. Εκτέλεση HSW-MyCap με M=4. . . . .   | 94 |
| 7.24 | Αρχική εικόνα. . . . .   | 94 |

|      |  |     |
|------|--|-----|
| 7.25 | a) ModHSW-MyCap για $M = 50$ . b) Εκτέλεση ModHSW-MyCap για $M = 100$ . c) HSW-MyCap για $M = 100$ . d) MCT- $\alpha$ -SCap, $\alpha = 40$ προκύπτουν ομάδες $M = 209$ . e)MCT-MyCap για $M = 200$ . f)MCT-SCap για $M = 200$ . . . . .                              | 95  |
| 7.26 | Αρχική εικόνα. . . . .   | 96  |
| 7.27 | a) K-μέσων για $M = 4$ . b) Εκτέλεση ModHSW-MyCap για $M = 20$ . c) MCT- $\alpha$ -SCap, $\alpha = 15$ προκύπτουν ομάδες $M = 90$ . d) MCT- $\alpha$ -SCap, $\alpha = 10$ προκύπτουν ομάδες $M = 84$ . e)MCT-SCap για $M = 300$ . f)MCT-SCap για $M = 500$ . . . . . | 97  |
| 7.28 | a) Αρχική εικόνα. b) MCT- $\alpha$ -SCap, $\alpha = 10$ προκύπτουν ομάδες $M = 81$ . c) MCT- $\alpha$ -SCap, $\alpha = 20$ προκύπτουν ομάδες $M = 340$ . d) MCT- $\alpha$ -SCap, $\alpha = 30$ προκύπτουν ομάδες $M = 352$ . . . . .                                 | 99  |
| 7.29 | a) Αρχική εικόνα. b)ModHSW-MyCap για $M = 40$ . c) ModHSW-MyCap για $M = 80$ . d) ModHSW-MyCap για $M = 120$ . . . . .   | 99  |
| 7.30 | a) Αρχική εικόνα. b)MCT-SCap και $M = 100$ . c) MCT-SCap και $M = 400$ .   | 100 |
| 7.31 | a) Αρχική εικόνα με μείωση στο 40% . . . . .   | 100 |
| 7.32 | a) MCT- $\alpha$ -SCap, $\alpha = 5$ προκύπτουν ομάδες $M = 233$ . b) MCT- $\alpha$ -SCap, $\alpha = 10$ προκύπτουν ομάδες $M = 256$ . c) ModHSW-MyCap για $M = 115$ . .   | 101 |
| 7.33 | a) Αρχική εικόνα . . . . .   | 101 |
| 7.34 | Εκτέλεση του ModHSW-MyCap για $M = 115$ στην αρχική εικόνα. . . . .  | 102 |

## LIST OF TABLES

---

|     |  |    |
|-----|--|----|
| 2.1 | Αλγόριθμοι Εύρεσης Μέγιστης Ροής . . . . .   | 25 |
| 3.1 | Αλγόριθμοι Εύρεσης Ελάχιστης Τομής . . . . . | 43 |
| 7.1 | Περιγραφή Συμβόλων . . . . .                 | 82 |

# ΕΥΡΕΤΗΡΙΟ ΑΛΓΟΡΙΘΜΩΝ

---

|    |   |    |
|----|---|----|
| 1  | Αλγόριθμος των Ford-Fulkerson με ακόρεστα μονοπάτια . . . . . | 18 |
| 2  | Multiterminal Cut Problem . . . . .                           | 26 |
| 3  | Αλγόριθμος Nagamochi-Ibaraki . . . . .                        | 32 |
| 4  | Ρουτίνα ContractSafe . . . . .                                | 33 |
| 5  | Επαναληπτικός αλγόριθμος συρρίκνωσης . . . . .                | 35 |
| 6  | MinimumCutPhase . . . . .                                     | 39 |
| 7  | Minimum Cut . . . . .   | 40 |
| 8  | Αλγόριθμος K-μέσων . . . . .                                  | 48 |
| 9  | Συσσωρευτικός Αλγόριθμος Ομαδοποίησης . . . . .               | 48 |
| 10 | Αλγόριθμος των ModHSW-MCT. . . . .                            | 66 |
| 11 | Αλγόριθμος Cut-clustering . . . . .                           | 73 |
| 12 | Ιεραρχικός Αλγόριθμος Cut-clustering . . . . .                | 76 |

# ΠΕΡΙΛΗΨΗ

---

Αυγούστα Θεοδούλου του Ιωάννη και της Χριστίνας.

ΜΕ, Τμήμα Πληροφορικής, Πανεπιστήμιο Ιωαννίνων, Ιούνιος 2008.

Τίτλος: Μελέτη Αλγορίθμων Ελάχιστης Τομής και Εφαρμογή τους σε Προβλήματα Κατά-  
τμησης Εικόνας.

Επιβλέπωντας: Σταύρος Νικολόπουλος.

Στην παρούσα εργασία μελετούμε το πρόβλημα ελάχιστης τομής το οποίο διαμερίζει ένα γράφημα  $G$ , με ακμικά βάρη, σε δύο ανεξάρτητα υπογραφήματα έτσι ώστε το άθροισμα των βαρών των ακμών που συνδέουν τα δύο υπογραφήματα να ελαχιστοποιείται (η διαμέριση αυτή λέγεται ελάχιστη τομή). Στη βιβλιογραφία υπάρχουν πολλές τεχνικές για το πρόβλημα της εύρεσης ελάχιστης τομής γραφήματος (πχ. τεχνικές ροής, συρρικνώσεις ακμών).

Το πρόβλημα αυτό βρίσκει πολλές εφαρμογές σε προβλήματα κατάτμησης εικόνας, το οποίο μπορεί να θεωρηθεί ως πρόβλημα διαμέρισης των στοιχείων της εικόνας σε διαφορετικές ομάδες. Επομένως, μέσω της μοντελοποίησης μιας εικόνας με ένα γράφημα το πρόβλημα κατάτμησης της μπορεί να λυθεί με εφαρμογή γραφοθεωρητικών τεχνικών διαμέρισης. Στη βιβλιογραφία υπάρχουν πολλοί τέτοιοι αλγόριθμοι οι οποίοι χρησιμοποιούν κυρίως τεχνικές ροής.

Αναπτύξαμε μια ιεραρχική μέθοδο διαμέρισης γραφήματος  $HSW$  η οποία βασίζεται στον αλγόριθμο εύρεσης ελάχιστης τομής Stoer-Wagner (SW) (χρησιμοποιεί τεχνικές συρρικνώσης ακμών). Στη συνέχεια εφαρμόσαμε τη μέθοδο αυτή στο πρόβλημα κατάτμησης εικόνας αφού χρησιμοποιήσαμε ένα στάδιο προεπεξεργασίας. Στο στάδιο της προεπεξεργασίας, μια εικόνα μοντελοποιείται με ένα μη κατευθυνόμενο γράφημα πλέγμα. Κάθε κόμβος του γραφήματος αντιστοιχεί σε ένα στοιχείο της εικόνας (pixel) και υπάρχει ακμή  $e$  με άκρα τους κόμβους  $v_1, v_2$  αν και μόνο αν τα στοιχεία  $p_1$  και  $p_2$  που αντιστοιχούν στους κόμβους  $v_1$  και  $v_2$  είναι γειτονικά (είτε σε οριζόντια είτε σε κατακόρυφη διεύθυνση). Έπειτα αντιστοιχεί σε κάθε ακμή ένα βάρος το οποίο εκφράζει την ομοιότητα των εικονοστοιχείων  $p_1$  και  $p_2$ .

Στο πρώτο βήμα του *HSW*, το γράφημα διαμερίζεται σε δύο υπογράφημα με βάση τον αλγόριθμο *SW* και σε κάθε υπογράφημα που προκύπτει επαναπροσδιορίζονται τα βάρη των ακμών του και υπολογίζεται μια ποσότητα ανομοιογένειας στους κόμβους που το επάγουν ως προς τη μέση τιμή φωτεινότητας των κόμβων του υπογραφήματος. Στη συνέχεια, η διαδικασία αυτή επαναλαμβάνεται στο υπογράφημα που παρουσιάζει τη μεγαλύτερη ανομοιογένεια, έως ότου προκύψουν  $K$  υπογράφημα. Το σύνολο των κόμβων που επάγουν κάθε υπογράφημα αποτελεί και μια ομάδα-τμήμα στην εικόνα.

Παρατηρήσαμε ότι ο αλγόριθμος *HSW*, όπως και πολλοί άλλοι που χρησιμοποιούν ως κριτήριο διαμέρισης την ελάχιστη τομή και εφαρμόζονται στο πρόβλημα κατάτμησης εικόνας, παρουσιάζει το πρόβλημα ότι δημιουργεί πολλές και μικρές ομάδες στο αποτέλεσμα. Για το λόγο αυτό, προτείνουμε τον αλγόριθμο *ModHSW*, ο οποίος τροποποιεί το κριτήριο διαμέρισης του *SW* και επιτυγχάνει καλύτερα αποτελέσματα στο πρόβλημα κατάτμησης εικόνας.

Μελετήσαμε επιπλέον δύο αλγόριθμους ομαδοποίησης που υπάρχουν στη βιβλιογραφία και βασίζονται σε διαμερίσεις γραφήματος τους οποίους χρησιμοποιήσαμε στην πειραματική διαδικασία για να συγκρίνουμε τα αποτελέσματα τους με τον αλγόριθμο κατάτμησης τον οποίο προτείνουμε.

Παρατηρούμε ότι ο αλγόριθμος *ModHSW* δίνει ικανοποιητικά αποτελέσματα παράγοντας πολύ λιγότερες ομάδες στο αποτέλεσμα σε σύγκριση με τους άλλους δύο αλγόριθμους. Παρόλα αυτά, υπάρχουν προβλήματα στα οποία χρειάζεται μεγάλο πλήθος ομάδων έως ότου ληφθούν όλα τα βασικά χαρακτηριστικά της εικόνας.

# EXTENDED ABSTRACT IN ENGLISH

---

Theodoulou Augousta, J., Ch.

MSc, Department of Computer Science, University of Ioannina, Greece. June 2008.

Thesis Title: Min-Cut Algorithms and Applications on Image Segmentation.

Thesis Supervisor: St. Nokolopoulos.

In this thesis we study the minimum cut problem that is to partition a graph  $G$ , containing edge weights, in two disjoint subgraphs such that the sum of the edge weights that connect the two subgraphs is minimized. The optimal partition is called min-cut. In literature there are many techniques that are related to this problem (e.g. flow-based techniques, edge contraction-based techniques).

The min-cut problem is widely applied to the image segmentation field. The problem of image segmentation can be considered as the partitioning of image pixels into different pixel groups. Thus, by modeling an image using a graph, the partitioning problem can be solved by applying graph theoretic partitioning techniques. Algorithms of this type are presented in literature and the majority of them are flow-based techniques.

We developed an hierarchical graph partitioning method *HSW* that is based on the Stoer-Wagner (*SW*) min-cut algorithm (it is a edge contraction-based approach). We applied this method on the image segmentation problem assuming that a preprocessing phase takes place beforehand. At that phase, an image is modeled using a undirected grid graph. Each graph node corresponds to a pixel of the image and an edge  $e$  is formed connecting the nodes  $u_1, u_2$  if and only if the pixels  $p_1$  and  $p_2$  that correspond to these nodes are neighbors in the image (either in horizontal, or in vertical order). Finally, a weight is considered on each edge that implies the similarity between the connected pixels  $p_1$  and  $p_2$ .

At the first step of *HSW*, the graph is partitioned in two subgraphs using the *SW* min-cut algorithm and the edge weights are being recomputed for each of these subgraphs. Moreover, a quantity is computed based on the intensity difference between each pixel

and the mean intensity of the pixels in the subgraph. This quantity describes how homogeneous are the pixel subgraphs formed so far and enables us to choose to continue this procedure on the least homogeneous of them until  $K$  subgraphs are formed. The nodes of a subgraph of the solution is also an image segment.

We observed that the *HSW* algorithm, similarly to many others that use the min-cut criterion and are applied on the image segmentation problem, tends to produce many and very small image segments. For this reason, we propose the *ModHSW* algorithm that modifies the partitioning criterion of SW and achieves better results on the problem under consideration.

Two other clustering algorithms that are presented in literature and are based to graph partitioning were also studied and tested in our comparative experimental evaluation of our proposed method.

The main observation is that *ModHSW* produces satisfactory results in much smaller number of image segments compared to the results of the two other tested approaches. Nevertheless, there are cases for which a great number of segments is needed to capture all the important image characteristics in the formed partitioning result.



# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ

- 
- 1.1 Το Πρόβλημα
  - 1.2 Υπόβαθρο και Βασικές Έννοιες
  - 1.3 Ελάχιστη Διαμέριση Γραφήματος
  - 1.4 Μέθοδοι Ομαδοποίησης-Κριτήρια Ομαδοποίησης
  - 1.5 Συμβολή
  - 1.6 Διάρθρωση Μεταπτυχιακής Εργασίας
- 

### 1.1 Το πρόβλημα

Πολλά πρακτικά προβλήματα της καθημερινής ζωής ή ακόμα και πιο σύνθετα επιστημονικά προβλήματα μπορούν να περιγραφούν με τη βοήθεια ενός γραφήματος που αποτελείται από ένα σύνολο κόμβων και ένα σύνολο ακμών, οι οποίες ενώνουν συγκεκριμένα ζεύγη κόμβων. Στη γενική περίπτωση μπορεί να ορίζονται και βάρη στα διάφορα στοιχεία του γραφήματος. Κατά συνέπεια η λύση σε αυτά τα προβλήματα μπορεί να αναζητηθεί βάσει γραφοθεωρητικών αλγορίθμων.

Στη διατριβή αυτή, το γενικό πρόβλημα που μας απασχολεί είναι αυτό της διαμέρισης γραφήματος, η οποία ορίζεται ως η διαίρεση του γραφήματος σε  $k$  ανεξάρτητα υπογραφήματα ( $k$ -διαμέριση) βάσει ενός κατάλληλου για το πρόβλημα κριτηρίου διαμέρισης. Το

πρόβλημα  $k$ -διαμέρισης χαρακτηρίζεται από την  $NP$ -πληρότητα, το οποίο σημαίνει πως η εύρεση της βέλτιστης λύσης είναι υπολογιστικά αδύνατη (ιδίως για γραφήματα με μεγάλο πλήθος κόμβων). Για το λόγο αυτό οι σχετικοί αλγόριθμοι επιχειρούν να προσεγγίσουν μια “καλή λύση” του προβλήματος - αποφεύγοντας την αναλυτική επίλυση - προσπαθώντας να βελτιστοποιήσουν το κριτήριο διαμέρισης (ελαχιστοποίηση ή μεγιστοποίηση).

Μια διαδεδομένη προσέγγιση στο πρόβλημα της  $k$ -διαμέρισης είναι η ιεραρχική διαμέριση η οποία επαναληπτικά χωρίζει ένα από τα ανεξάρτητα υπογραφήματα (αρχικά υπάρχει ένα ενιαίο γράφημα) σε δύο τμήματα. Συνεπώς, το πρόβλημα ανάγεται σε πολλά απλούστερα προβλήματα 2-διαμέρισης τα οποία επιλύονται σε πολυωνυμικό χρόνο.

Στην εργασία αυτή, λοιπόν, μελετήθηκαν αλγόριθμοι διαμέρισης με βάση το κριτήριο (διαμέρισης) ελάχιστης τομής (minimum cut). Στόχος τους είναι η 2-διαμέριση του γραφήματος σε δύο υπογραφήματα  $A$ ,  $B$  έτσι ώστε το άθροισμα των βαρών των ακμών που έχουν το ένα άκρο τους στο υπογράφημα  $A$  και το άλλο άκρο τους στο υπογράφημα  $B$  να είναι το ελάχιστο.

Τα τελευταία χρόνια οι αλγόριθμοι αυτοί χρησιμοποιούνται σε ένα σημαντικό πρόβλημα της επιστήμης των υπολογιστών, την ομαδοποίηση δεδομένων. Δοθέντος ενός συνόλου δεδομένων που αποτελείται από αντικείμενα, μια διαδικασία ομαδοποίησης στοχεύει στο διαχωρισμό των αντικειμένων αυτών σε ομάδες με συναφή χαρακτηριστικά. Η συνάφεια μεταξύ ζευγών αντικειμένων ορίζεται μέσω μιας κατάλληλης για τα αντικείμενα συνάρτησης ομοιότητας.

Η διαδικασία εφαρμογής των αλγορίθμων διαμέρισης σε προβλήματα ομαδοποίησης απαιτεί μια προεπεξεργασία κατά την οποία το πρόβλημα που καλούμαστε να επιλύσουμε μοντελοποιείται με ένα γράφημα: τα δεδομένα προς ομαδοποίηση αναπαρίστανται με κόμβους και η ομοιότητα μεταξύ ζευγών δεδομένων ορίζει τα βάρη των ακμών που συνδέουν τους αντίστοιχους κόμβους.

Στην εργασία αυτή εφαρμόσαμε γραφοθεωρητικούς αλγόριθμους διαμέρισης γραφημάτων για την επίλυση του προβλήματος κατάτμησης εικόνας που επιλύεται μέσω μεθόδων ομαδοποίησης.

## 1.2 Υπόβαθρο και Βασικές έννοιες

Σε αυτή την ενότητα, ορίζουμε και περιγράφουμε μερικές από τις βασικές τεχνικές που χρησιμοποιούνται στους αλγόριθμους που μελετήσαμε.

### 1.2.1 Γραφήματα και Τομές

Έστω  $G = (V, E, c)$ , ένα μη κατευθυνόμενο γράφημα με σύνολο κόμβων  $V$ , σύνολο ακμών  $E$  και  $c : E \rightarrow \mathbb{R}^+$  μια μη-αρνητική συνάρτηση πραγματικών τιμών που αντιστοιχεί στις χωρητικότητες των ακμών. Στην παρουσίαση αυτή θα ασχοληθούμε μόνο με γραφήματα με βάρη και γι'αυτό θα αναφερόμαστε σε ένα τέτοιο γράφημα με το συμβολισμό  $G(V, E)$ . Έστω  $n = |V|$  και  $m = |E|$ . Συμβολίζουμε μια μη-κατευθυνόμενη ακμή με άκρα τους κόμβους  $u$  και  $w$  ως  $uw$ , και την χωρητικότητα της αντίστοιχης ακμής ως  $c(u, w)$ .

**Τομή:** είναι μια διαμέριση των κόμβων σε δύο μη κενά υποσύνολα  $A$  και  $\bar{A}$ , όπου  $\bar{A} = V - A$ .

**Χωρητικότητα μιας τομής:** είναι μια ποσότητα που χαρακτηρίζει μια ακμή (ορίζεται με μια συνάρτηση) με  $c(A, \bar{A})$  και ορίζεται ως

$$c(A, \bar{A}) = \sum_{u \in A, v \in \bar{A}, \{u, v\} \in E} c(u, v) \quad (1.1)$$

Εν συντομία μπορούμε να αναφερόμαστε στην τομή χρησιμοποιώντας τον συμβολισμό  $c(A)$  αντί του  $c(A, \bar{A})$ . Επίσης αν το σύνολο  $A$  αποτελείται από ένα και μοναδικό κόμβο  $u$  (singleton) δηλαδή  $A = \{u\}$ , τότε αναφερόμαστε στην τομή και ως  $c(u)$ . Θα αναφερόμαστε στις ακμές που περιλαμβάνονται στο άθροισμα 1.1 ως ακμές τομής.

**Ελάχιστη τομή:** είναι η τομή που ελαχιστοποιεί την ποσότητα  $c(A)$ . Συμβολίζουμε με  $\lambda(G)$  την τιμή της ελάχιστης τομής του γραφήματος  $G$ .

**Ελάχιστη  $(u, v)$ -τομή:** είναι η ελάχιστη τομή  $(A, B)$  του  $G$  κατά την οποία ο κόμβος  $u$  ανήκει στο σύνολο  $A$  και ο  $v$  στο σύνολο  $B$  και τη συμβολίζουμε με  $\lambda_{u,v}(G)$ .

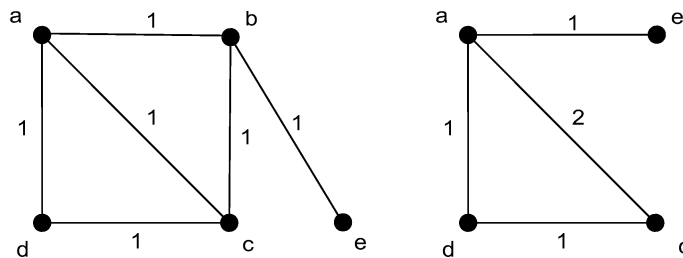
### 1.2.2 Συρρίκνωση Ακμής $uw$ - Συρρίκνωση Κόμβων $u, w$

Δοθέντος ενός γραφήματος  $G$  και ακμής  $uw$ , ορίζουμε ως  $G \setminus \{u, w\}$ , τη συρρίκνωση της ακμής  $uw$ , κατά την οποία η ακμή  $uw$  διαγράφεται, ο κόμβος  $w$  διαγράφεται και κάθε ακμή της μορφής  $wx$ , αντικαθίσταται με την ακμή  $ux$ . Αν προκύψουν παράλληλες ακμές, τότε αυτές συγχωνεύονται σε μία με χωρητικότητα το άθροισμα των χωρητικότητων τους. Αν

προκύψουν βρόγχοι κατά τη διαδικασία, διαγράφονται. Ακριβώς η ίδια διαδικασία χρησιμοποιείται και για τη συρρίκνωση δύο κόμβων  $u, w$  με τη διαφορά ότι σε αυτή τη περίπτωση μπορεί να μην υπάρχει ακμή  $uw$  και επομένως δε χρειάζεται η διαγραφή της.

Η βασική ιδέα των αλγορίθμων που χρησιμοποιούν επαναληπτικές συρρίκνώσεις ακμών είναι η εξής: αν αναγνωριστεί ακμή  $η$  οποία δεν συμμετέχει στην ελάχιστη τομή, τότε η ακμή αυτή μπορεί με ασφάλεια να συρρικνωθεί (ο αλγόριθμος KS μπορεί να συρρικνώσει και ακμές οι οποίες συμμετέχουν στην ελάχιστη τομή δεδομένου του ότι είναι πιθανοτικός αλγόριθμος).

Το πιο κάτω σχήμα δείχνει την συρρίκνωση της ακμής  $ab$ .



Σχήμα 1.1: Αριστερά φαίνεται το γράφημα  $G$ . Έστω  $ab$  ακμή που πρόκειται να συρρικνωθεί. Δεξιά φαίνεται το γράφημα  $G/\{a, b\}$  που προκύπτει μετά τη συρρίκνωση.

### 1.2.3 Δίκτυο

Ένα δίκτυο (network),  $N(V, A, c)$ , είναι ένα κατευθυνόμενο γράφημα με σύνολο κόμβων  $V$ , σύνολο κατευθυνόμενων ακμών  $A$  και επιπλέον:

1. έχει μία κορυφή-αφετηρία  $s$  με έσω-βαθμό  $d^-(s) = 0$  που ονομάζεται πηγή.
2. έχει μία κορυφή-προορισμό  $t$  με έξω-βαθμό  $d^+(t) = 0$  (συνήθως), που ονομάζεται προορισμός (sink), και
3. για κάθε τόξο  $a \in A$  δίδεται μια μη-αρνητική πραγματική τιμή  $c(a)$ , που ονομάζεται χωρητικότητα τόξου (arc capacity).

### 1.2.4 Δίκτυα Ροής

Οι τεχνικές ροής χρησιμοποιούνται σε κατευθυνόμενα γραφήματα αλλά ωστόσο μπορούν να επεκταθούν και σε μη-κατευθυνόμενα γραφήματα. Αυτό συμβαίνει γιατί ένα μη-κατευθυνόμενο γράφημα μπορεί να μετασχηματιστεί σε ένα κατευθυνόμενο γράφημα με την εξής

διαδικασία: αντικατέστησε κάθε ακμή  $uw$  χωρητικότητας  $c(u, w) = x$  με δύο τόξα  $\overrightarrow{uw}$  και  $\overleftarrow{wu}$  χωρητικότητας  $x$  το κάθε ένα. Για κάθε ζεύγος κόμβων  $s, t$ , μια τομή είναι η ελάχιστη  $(s, t)$ -τομή στο αρχικό γράφημα αν και μόνο αν είναι η ελάχιστη  $(s, t)$ -τομή στο μετασχηματισμένο γράφημα.

Έστω  $N = (V, A, c)$  το κατευθυνόμενο γράφημα με δύο ξεχωριστούς κόμβους  $s$  (πηγή) και  $t$  (προορισμό). Ορίζεται μια συνάρτηση ροής, η οποία καθορίζει την ποσότητα ροής για κάθε  $\overrightarrow{uw}$  και ικανοποιεί τις σχέσεις:

1.  $f(u, w) \leq c(u, w), \quad \forall uw \in A$
2.  $f(u, w) = -f(w, u), \quad uw \in A$
3.  $\sum_{u \in V} f(u, w) = 0, \quad \forall w \in V - \{s, t\}$

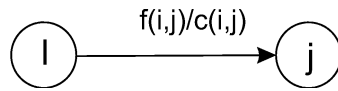
Η Εξίσωση 1, δηλώνει ότι η ροή που μπορεί να περάσει από μια ακμή φράσσεται πάνω από τη χωρητικότητα της ακμής. Η Εξίσωση 2 και 3 δηλώνουν ότι η ροή που εισέρχεται ενός κόμβου  $u$  είναι ίση με τη ροή που εξέρχεται του κόμβου  $u$ . Η τιμή μιας ροής  $|f|$  είναι η ροή δικτύου που εισέρχεται στον κόμβο προορισμό  $t$  και η τιμή της ορίζεται από την ακόλουθη σχέση:

$$|f| = \sum_{u \in V} f(u, t)$$

Το πρόβλημα μέγιστης ροής (maximum flow problem) είναι ο καθορισμός μιας ροής  $f$  της οποίας η τιμή  $|f|$  είναι η μέγιστη. Το πολύ γνωστό θεώρημα μέγιστης ροής-ελάχιστης τομής (maxflow-mincut theorem) ([22]) αποδεικνύει ότι η τιμή της μέγιστης  $(s, t)$ -ροής είναι ίση με τη τιμή της ελάχιστης  $(s, t)$ -τομής, δηλαδή  $|f| = \lambda_{s,t}(G)$ . Επομένως οι αλγόριθμοι εύρεσης μέγιστης  $(s, t)$ -ροής, μπορούν να χρησιμοποιηθούν στην επίλυση του προβλήματος εύρεσης ελάχιστης  $(s, t)$ -τομής. Συνεπώς η ελάχιστη  $(s, t)$ -τομή από όλες τις δυνατές επιλογές  $\binom{n}{2}$  των κόμβων  $s$  και  $t$ , δίνει και την τιμή της ελάχιστης τομής.

### 1.2.5 Εναπομένον Γράφημα (Residual Network)

Δοθέντος ενός δικτύου  $N(V, A)$  κάθε ακμή του οποίου χαρακτηρίζεται από μια ροή και μια χωρητικότητα, όπως φαίνεται στο Σχήμα 1.2, το εναπομένον γράφημα  $N_r$  του  $N$  έχει σύνολο κόμβων, το σύνολο  $V$ , ενώ το σύνολο των ακμών  $A(N_r)$ , αποτελείται από τα “χρήσιμα τόξα”,  $\overrightarrow{ij}$ , του  $N$ . Τα “χρήσιμα τόξα” του  $N$  είναι αυτά που ικανοποιούν τις συνθήκες του πίνακα στο Σχήμα 1.3 και κάθε ακμή  $ij \in A(N_r)$  χαρακτηρίζεται από μια ποσότητα  $r(i, j)$ . Κάθε τόξο του  $N_r$  με μη-αρνητική τιμή  $r(i, j)$ , υποδεικνύει την μέγιστη ροή που μπορεί να προστεθεί στην ακμή  $\overrightarrow{ij}$  του  $N$ .

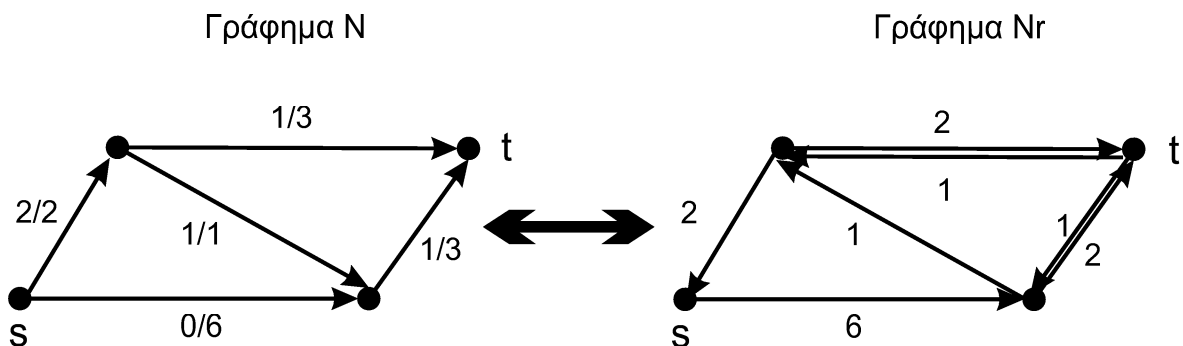


Σχήμα 1.2: Η ακμή  $ij$  του γραφήματος  $N$

| residual arc | conditions          | residual capacity             |
|--------------|---------------------|-------------------------------|
| $(i, j)$     | $c(i, j) > f(i, j)$ | $r(i, j) = c(i, j) - f(i, j)$ |
| $(j, i)$     | $f(i, j) > 0$       | $r(i, j) = f(i, j)$           |

Σχήμα 1.3: Πίνακας: Μια ακμή  $a$  ανήκει στο γράφημα  $N_r$  αν ικανοποιούνται οι αντίστοιχες συνθήκες. Κάθε ακμή  $ij \in A(N_r)$  χαρακτηρίζεται με την ποσότητα  $r(i, j)$  ανάλογα με τη συνθήκη που ικανοποιείται.

Οι ποσότητες  $r(i, j)$  ορίζονται και αυτές με βάση τον πίνακα στο Σχήμα 1.3. Στη συνέχεια παρατίθεται ένα παράδειγμα στο οποίο φαίνεται πως από ένα δίκτυο  $N$  προκύπτει το αντίστοιχο εναπομένον γράφημα  $N_r$ .



Σχήμα 1.4: Αριστερά φαίνεται το αρχικό γράφημα ροής  $N$  και δεξιά το αντίστοιχο εναπομένον  $N_r$ .

**Excess ενός κόμβου:** Η ποσότητα excess ενός κόμβου  $i$  συμβολίζεται με  $e(i)$  και ισούται με:

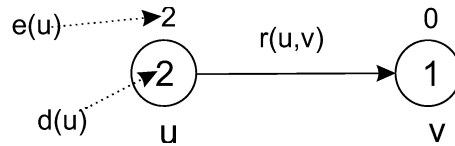
$$e(i) = \sum_{k \in A} f(k, i) - \sum_{j \in A} f(i, j)$$

**Admissible ακμή:** είναι μια ακμή  $uv$  στο  $N_r$  η οποία ικανοποιεί την ιδιότητα  $d(u) = d(v) + 1$ , όπου  $d$  είναι μια συνάρτηση καθορισμού αποστάσεων των κόμβων του  $N_r$  από τον κόμβο προορισμό του αντίστοιχου δικτύου.

**Ενεργός κόμβος (Active node):** Ενεργός κόμβος είναι ο κόμβος για τον οποίο ισχύει:

- $e(u) > 0$

Ο κόμβος  $u$  που φαίνεται στο Σχήμα 1.5 αποτελεί έναν ενεργό κόμβο.



Σχήμα 1.5: Ο κόμβος  $u$  με επιγραφή  $d(u) = 2$  αποτελεί έναν ενεργό κόμβο

### 1.2.6 Αλγόριθμος Prim Εύρεσης Maximal Spanning Tree

Έστω γράφημα  $G$  με  $m$  ακμές όπου κάθε ακμή  $e$  χαρακτηρίζεται από ένα βάρος  $w(e)$ . Αρχικά, ο αλγόριθμος Prim παίρνει έναν αρχικό κόμβο  $s$ , τον προσθέτει στο σύνολο  $U = u$  και θεωρεί ένα συνδετικό δέντρο (spanning tree)  $F$  με μοναδικό κόμβο τον κόμβο  $u$ . Στην συνέχεια ο αλγόριθμος προσθέτει την ακμή  $\{u, v\}$  με το μεγαλύτερο βάρος στο συνδετικό δέντρο καθώς και τον κόμβο  $v$  στο σύνολο  $U$ . Η διαδικασία επαναλαμβάνεται έως ότου όλοι οι κόμβοι του γραφήματος  $G$  συμπεριληφθούν στο σύνολο  $U$ . Η έξοδος του αλγορίθμου είναι το maximal συνδετικό δέντρο του  $G$ .

### 1.3 Πρόβλημα Ελάχιστης Τομής

Το πρόβλημα ελάχιστης τομής (minimum cut) είναι ένα πρόβλημα διαμέρισης των κόμβων ενός μη κατευθυνόμενου γραφήματος  $G$  με βάρη, το οποίο έχει  $n$  κόμβους και  $m$  ακμές, σε δύο υποσύνολα έτσι ώστε το συνολικό βάρος των ακμών με το ένα άκρο τους στο ένα υποσύνολο και το άλλο άκρο τους στο άλλο να ελαχιστοποιείται. Το πρόβλημα αυτό έχει πολλές εφαρμογές στην ομαδοποίηση δεδομένων, θεωρία αξιοπιστίας δικτύων ([21], [24]), εξόρυξη δεδομένων και ως υπορουτίνα σε αλγορίθμους που επιλύουν το πρόβλημα του περιοδεύοντος πωλητή (Traveling Salesman problem) ([25]). (Το πρόβλημα του περιοδεύοντος πωλητή διατυπώνεται ως εξής, λαμβάνοντας υπόψη διάφορες πόλεις και το κόστος μετάβασης από τη μια πόλη στην άλλη, αναζητείται η ελάχιστου κόστους μετ' επιστροφής διαδρομή κατά την οποία ο πωλητής επισκέπτεται κάθε πόλη ακριβώς μια φορά και επιστρέφει έπειτα στην αρχική πόλη.)

Το πρόβλημα εύρεσης της ελάχιστης χωρητικότητας τομής (minimum capacity cut), ανάμεσα σε δύο συγκεκριμένους κόμβους,  $s$  και  $t$ , ονομάζεται πρόβλημα ελάχιστης  $(s, t)$ -τομής. Το πρόβλημα αυτό είναι στενά συνδεδεμένο με το πρόβλημα ελάχιστης τομής. Ο κλασικός αλγόριθμος Gomory-Hu [6], επιλύει το πρόβλημα ελάχιστης τομής χρησιμοποιώντας  $n - 1$  υπολογισμούς της ελάχιστης  $(s, t)$ -τομής. Κάποιοι γρήγοροι αλγόριθμοι εύρεσης ελάχιστης  $(s, t)$ -τομής, που χρησιμοποιούν τεχνικές ροής, αναφέρονται στην βιβλιογραφία [12], [13], [14] και εκτελούνται σε χρόνο  $\omega(nm)$ . Για το πρόβλημα ελάχιστης τομής οι Hao και Orlin ([17], [18]) έδωσαν έναν αλγόριθμο (HO), ο οποίος δείχνει με ποιο τρόπο μπορεί να προσαρμοστούν όλες οι  $n - 1$  ελάχιστες  $(s, t)$ -τομές σε χρόνο ασυμπτωτικά ίσο με αυτόν που χρειάζεται για να εφαρμοστεί ένας μόνο υπολογισμός ελάχιστης  $(s, t)$ -τομής. Αυτός ο αλγόριθμος εκτελείται σε χρόνο  $O(nm \log(n^2/m))$ .

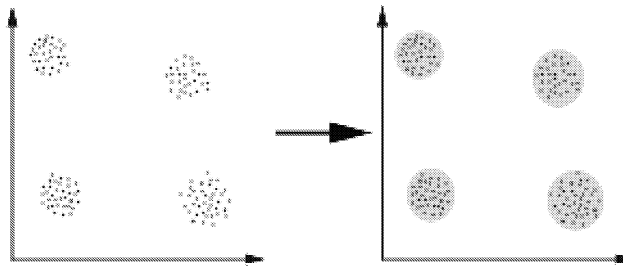
Πρόσφατα, νέοι αλγόριθμοι έχουν αναπτυχθεί για την επίλυση του προβλήματος ελάχιστης τομής, οι οποίοι θεωρητικά είναι πιο αποδοτικοί και συγκρίσιμοι ως αναφορά το χρόνο εκτέλεσης τους με τους καλύτερους σε χρόνο αλγόριθμους εύρεσης ελάχιστης  $(s, t)$ -τομής. Ο αλγόριθμος των Nagamochi και Ibaraki [10](NI), εκτελείται σε χρόνο  $O(n(m + n \log n))$ . Ο αλγόριθμος των Karger και Stein [20] (KS) εκτελείται σε χρόνο  $O(n^2 \log^3 n)$ . Δύο άλλες παραλλαγές του αλγορίθμου του Karger [19] (K) εκτελούνται σε χρόνους  $O(m \log^3 n)$  και  $O(n^2 \log n)$ . Ο αλγόριθμος των Stoer και Wagner [26], ο οποίος εκτελείται σε χρόνο  $O(|V||E| + |V|^2 \log |V|)$ . Αυτοί οι αλγόριθμοι βασίζονται σε μια νέα τεχνική η οποία δε χρησιμοποιεί θεωρία ροής και δεν επεκτείνονται σε κατευθυνόμενα γραφήματα όπως οι αλγόριθμοι που βασίζονται στη θεωρία ροής (συμπεριλαμβανομένου και του αλγορίθμου HO). Οι αλγόριθμοι KS και K έχουν κάποια πλεονεκτήματα, με μεγάλη πιθανότητα βρίσκουν όλες τις ελάχιστες τομές. Αυτό είναι πολύ σημαντικό σε διάφορες εφαρμογές και ιδιαίτερα στο πρόβλημα του περιοδεύοντος πωλητή (TSP). Σε κάποιες κλάσεις προβλημάτων, μπορούν να δώσουν σωστές απαντήσεις εφαρμόζοντας λιγότερες δοκιμές. Επομένως στις κλάσεις αυτές μπορούν να εκτελεστούν ταχύτερα.

## 1.4 Ομαδοποίηση με Διαμέριση Γραφήματος

Το πρόβλημα της ομαδοποίησης μπορεί να θεωρηθεί ως το πιο σημαντικό πρόβλημα μάθησης χωρίς επίβλεψη. Κάθε τέτοιου τύπου πρόβλημα προϋποθέτει την εύρεση μιας δομής που θα αντιστοιχεί στη συλλογή των δεδομένων. Μια ομάδα επομένως, είναι μια συλλογή δεδομένων τα οποία είναι “όμοια” μεταξύ τους και “διαφορετικά” με τα δεδομένα που ανήκουν



σε διαφορετικές ομάδες. Χρησιμοποιείται ένα κριτήριο ομαδοποίησης το οποίο εξαρτάται από το πρόβλημα, με το οποίο χωρίζουμε τα δεδομένα σε δύο ή περισσότερες ομάδες. Στο Σχήμα 1.6 φαίνεται ένα γραφικό παράδειγμα ομαδοποίησης με κριτήριο ομαδοποίησης την απόσταση των δεδομένων της αρχικής συλλογής.



Σχήμα 1.6: (a) Παράδειγμα ομαδοποίησης με κριτήριο ομαδοποίησης την απόσταση των δεδομένων της αρχικής συλλογής.

Τον τελευταίο καιρό προτείνεται μια νέα μέθοδος ομαδοποίησης η οποία βασίζεται σε γραφοθεωρητικούς αλγόριθμους (graph clustering), και επιτυγχάνεται μέσω της μοντελοποίησης του προβλήματος με γράφημα. Μπορεί κανείς να αντιληφθεί τη μέθοδο αυτή ως μια επαναληπτική διαμέριση, όπου σε κάθε βήμα το γράφημα διαμερίζεται σε δύο ανεξάρτητα υπογραφήματα κάθε ένα από τα οποία αποτελείται από τους κόμβους  $A \subset V$  και  $B \subset V$  αντίστοιχα, έτσι ώστε  $A + B = V$ , με βάση κάποιο κριτήριο διαμέρισης. Τα κριτήρια διαμέρισης ποικίλουν και η επιλογή του κατάλληλου κριτηρίου εξαρτάται από το υπό εξέταση πρόβλημα.

Στο σημείο αυτό αναφέρονται τρία από τα κριτήρια αυτά:

1. ελάχιστη διαμέριση (minimum cut),
2. μέση διαμέριση (average cut),
3. κανονικοποιημένη διαμέριση (normalized cut).

Το πρώτο κριτήριο διαμέρισης ελαχιστοποιεί το συνολικό βάρος των ακμών που συνδέουν

τα τμήματα  $A, B$  της διαμέρισης.

$$Cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

Το δεύτερο κριτήριο ελαχιστοποιεί την ποσότητα:

$$Cut(A, B) = \frac{Cut(A, B)}{|A|} + \frac{Cut(A, B)}{|B|}$$

πριμοδοτώντας έτσι διαμερίσεις οι οποίες αποτελούνται από σχετικά μεγάλα τμήματα.

Το τρίτο κριτήριο ελαχιστοποιεί την ποσότητα:

$$Cut(A, B) = \frac{Cut(A, B)}{assoc(A, V)} + \frac{Cut(A, B)}{assoc(B, V)}, \text{ όπου } assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$$

Η ποσότητα  $assoc(A, V)$ , εκφράζει το άθροισμα των βαρών των ακμών, των οποίων το ένα άκρο βρίσκεται στο σύνολο  $A$  και το άλλο άκρο τους στο σύνολο  $V = A + B$ . Αντίστοιχα, ορίζεται και η ποσότητα  $assoc(B, V)$ .

Προκύπτουν δηλαδή οι πιο κάτω σχέσεις οι οποίες συσχετίζουν τις ποσότητες  $assoc(A, V)$ ,  $assoc(B, V)$ ,  $assoc(A, A)$  και  $assoc(B, B)$  με τη τιμή της διαμέρισης του συνόλου  $V$  στα υποσύνολα  $A$  και  $B$ ,  $Cut(A, B)$ :

$$Cut(A, B) = assoc(A, V) - Assoc(A, A) = assoc(B, V) - assoc(B, B)$$

Το κριτήριο αυτό πριμοδοτεί ομάδες με μεγάλη εσωτερική ομοιότητα (intracluster similarity). Το συνολικό βάρος της διαμέρισης διαιρείται με το συνολικό βάρος των συνδέσεων στο εσωτερικό κάθε συνόλου  $assoc(A, A)$  και  $assoc(B, B)$  των συνόλων  $A$  και  $B$  αντίστοιχα. Το κριτήριο διαμέρισης κανονικοποιείται στο  $[0, 1]$  διαιρώντας το  $Cut(A, B)$  με τις ποσότητες  $assoc(A, V) = assoc(A, A) + Cut(A, B)$  και  $assoc(B, V) = assoc(B, B) + Cut(A, B)$ .

Η στρατηγική ομαδοποίησης που βασίζεται σε γραφήματα αποτελείται από τρία βασικά μέρη:

1. αντλούνται χαρακτηριστικά χαμηλού επιπέδου (πχ. αποστάσεις μεταξύ των στοιχείων),
2. κατασκευή του γραφήματος αναπαράστασης,
3. διαμέριση του γραφήματος αναπαράστασης χρησιμοποιώντας το κατάλληλο κριτήριο διαμέρισης.

Όπως παρατηρήθηκε και από πειράματα που έγιναν στην παρούσα εργασία αλλά και από άλλες αναφορές στο πρόβλημα, το κριτήριο ελάχιστης διαμέρισης δεν οδηγεί πάντα σε σωστή διαμέριση. Το αποτέλεσμα της χρήσης αυτού του κριτηρίου θα είναι σωστό μόνο αν οι ομάδες είναι ισχυρά συνδεδεμένες στο εσωτερικό τους και οι συνδέσεις μεταξύ των ομάδων είναι πολύ μικρές.

Η ελαχιστοποίηση των κριτηρίων μέσης διαμέρισης και κανονικοποιημένης διαμέρισης αντίθετα θα οδηγήσουν σε σωστή τμηματοποίηση. Μια παραλλαγή του κριτηρίου κανονικοποιημένης διαμέρισης στο οποίο η κανονικοποίηση γίνεται με βάση τα  $assoc(A, A)$  και  $assoc(B, B)$  στα σύνολα  $A$  και  $B$  αντίστοιχα, οδηγεί σε λάθος ομάδες σε κάποιες εικόνες.

## 1.5 Συμβολή

Στη διατριβή αυτή μελετήσαμε το πρόβλημα της εύρεσης ελάχιστης τομής ενός γραφήματος το οποίο δείχνει να αποκτά ιδιαίτερο ενδιαφέρον σε πολλά προβλήματα, όπως ομαδοποίησης δεδομένων, τεχνητής όρασης (κατάτμησης εικόνας), προβλήματα σχεδιασμού κυκλωμάτων VLSI, δικτύων, κ.τ.λ.

Οι αλγόριθμοι εύρεσης ελάχιστης τομής σε ένα γράφημα  $G$  χωρίζεται σε δύο μεγάλες κατηγορίες:

- A) Η πρώτη κατηγορία αλγορίθμων επιλύει το πρόβλημα εύρεσης ελάχιστης τομής βασισμένη στο θεώρημα των Ford και Fulkerson το οποίο αποδεικνύει ότι η τιμή της μέγιστης  $(s, t)$ -ροής (maximum  $(s, t)$ -flow) είναι ίση με την τιμή της ελάχιστης  $(s, t)$ -τομής (minimum  $(s, t)$ -cut), για οποιουδήποτε κόμβους  $s$  και  $t$ . Το θεώρημα αυτό αποτελεί τη βάση όλων των αλγορίθμων που εντάσσονται στην κατηγορία αυτή.
- B) Η δεύτερη κατηγορία αλγορίθμων εύρεσης ελάχιστης τομής ενός γραφήματος  $G$ , βασίζεται στις επαναληπτικές συρρικνώσεις ακμών μεγάλου βάρους οι οποίες δεν συμμετέχουν στην ελάχιστη τομή.

Οι αλγόριθμοι της πρώτης κατηγορίας μπορούν να υπολογίσουν ελάχιστες τομές τόσο σε κατευθυνόμενα όσο και σε μη κατευθυνόμενα γραφήματα, ενώ οι αλγόριθμοι της δεύτερης κατηγορίας μπορούν να υπολογίσουν τομές μόνο σε μη κατευθυνόμενα γραφήματα.

Μελετήθηκαν διάφορες εφαρμογές των γραφοθεωρητικών αυτών αλγορίθμων σε προβλήματα ομαδοποίησης ([29], [5]), κατάτμησης εικόνας ([27], [2], [3]), και δικτύων [24].

Αλγόριθμοι εύρεσης ελάχιστης τομής που ανήκουν στην πρώτη κατηγορία έχουν προσαρμοστεί και εφαρμοστεί σε προβλήματα ομαδοποίησης ([29], [5]) και κατάτμησης εικόνας ([27], [2], [3]), δίδοντας ικανοποιητικά αποτελέσματα.

Μελετήσαμε έναν αλγόριθμο ομαδοποίησης ο οποίος βασίζεται σε τεχνικές μέγιστης ροής και συγκεκριμένα σε δέντρα ελάχιστης τομής (minimum cut trees) [29], ο οποίος χρησιμοποιεί τον αλγόριθμο των Gomory και Hu της κατηγορίας (A). Επίσης μελετήσαμε και έναν αλγόριθμο της κατηγορίας (B), τον αλγόριθμο των Stoer και Wagner [26] που αποτελεί βελτίωση του αλγορίθμου των Nagamochi και Ibaraki [9], τον οποίο χρησιμοποιήσαμε επαναληπτικά για την επίλυση του προβλήματος ομαδοποίησης. Έπειτα από ένα στάδιο προεπεξεργασίας, συγκρίναμε τα αποτελέσματα των δύο αυτών αλγορίθμων σε προβλήματα κατάτμησης εικόνας.

Στο στάδιο της προεπεξεργασίας, διαβάζουμε μέσω της Matlab αρχεία εικόνων οποιασδήποτε μορφής (bmp, jpg κ.α), από τα οποία εξάγουμε στοιχειώδη χαρακτηριστικά της εικόνας (φωτεινότητες των στοιχείων της). Έπειτα, δημιουργείται ένα αρχείο περιγραφής των στοιχείων της εικόνας με χαρακτήρες ASCII, το οποίο αποτελεί και την είσοδο του αλγορίθμου ομαδοποίησης που χρησιμοποιείται για την επίλυση του προβλήματος κατάτμησης εικόνας. Στη συνέχεια, ο αλγόριθμος ομαδοποίησης διαβάζει το αρχείο εισόδου και αναπαριστά την εικόνα με ένα μη κατευθυνόμενο γράφημα  $G$ . Κάθε κόμβος του γραφήματος αντιστοιχεί σε ένα στοιχείο της εικόνας (pixel) και υπάρχει ακμή  $e$  με άκρα τους κόμβους  $v_1$ ,  $v_2$  αν και μόνο αν τα στοιχεία  $p_1$  και  $p_2$  που αντιστοιχούν στους κόμβους  $v_1$  και  $v_2$  είναι γειτονικά (είτε σε οριζόντια είτε σε κατακόρυφη κατεύθυνση). Έπειτα αντιστοιχεί σε κάθε ακμή ένα βάρος το οποίο εκφράζει την ομοιότητα των εικονοστοιχείων  $p_1$  και  $p_2$ .

Στην επαναληπτική διαδικασία ομαδοποίησης με βάση τον αλγόριθμο των Stoer και Wagner που παρουσιάζουμε, παρατηρήσαμε ότι το αποτέλεσμα εξαρτάται σε μεγάλο βαθμό από τον τρόπο καθορισμού των βαρών του γραφήματος. Ορίζοντας τα βάρη με τρόπο ώστε να τείνουν γρήγορα στο μηδέν, παρατηρήσαμε ότι το αποτέλεσμα της κατάτμησης ήταν ικανοποιητικό. Παρ' όλα αυτά, όταν τα βάρη τείνουν γρήγορα στο μηδέν είναι πολύ πιθανόν να χαθεί και σημαντική πληροφορία. Τροποποιώντας τα βάρη έτσι ώστε να μην τείνουν γρήγορα στο μηδέν, ο αλγόριθμος αποτυγχάνει δίδοντας σχεδόν τετριμμένες ομάδες, γεγονός που αποδίδεται στο κριτήριο αποκοπής του αλγορίθμου. Μια επιπλέον τροποποίηση που εφαρμόσαμε είναι η κανονικοποίηση του κριτηρίου αποκοπής του αλγορίθμου. Ο επαναληπτικός (τροποποιημένος) αλγόριθμος που υλοποιήσαμε δίδει καλύτερα αποτελέσματα (χωρίς να χάνει χρήσιμη πληροφορία) σε σύγκριση με τον επαναληπτικό μη τροποποιημένο

αλγόριθμο ο οποίος εφαρμόζεται σε γράφημα του οποίου τα βάρη τείνουν γρήγορα στο μηδέν.

## 1.6 Διάρθρωση Μεταπτυχιακής Εργασίας

Στο Κεφάλαιο 1 γίνεται μια σύντομη αναφορά στο πρόβλημα διαμέρισης γραφήματος και στα διάφορα κριτήρια διαμέρισης που χρησιμοποιούνται για την επίλυση του. Επίσης γίνεται μια ιστορική αναδρομή σε μεθόδους έχουν χρησιμοποιηθεί για την επίλυση προβλημάτων διαμέρισης με κριτήριο διαμέρισης την ελάχιστη τομή (minimum cut) του γραφήματος. Στο Κεφάλαιο 2 ορίζεται το γραφοθεωρητικό πρόβλημα μέγιστης ροής (maximum flow) και τη σχέση αυτού του προβλήματος με το πρόβλημα ελάχιστης τομής. Επίσης αναφέρονται διάφορες μέθοδοι εύρεσης μέγιστης ροής καθώς και μέθοδοι δημιουργίας δέντρων τομής. Στο Κεφάλαιο 3 αναφέρονται μέθοδοι διαμέρισης γραφημάτων με κριτήριο διαμέρισης την ελάχιστη τομή. Στο Κεφάλαιο 4 γίνεται αναφορά σε προβλήματα ομαδοποίησης και κατάτμησης εικόνας, στο κεφάλαιο αυτό επίσης αναφέρεται και ο αλγόριθμος των Wu και Leahy [27] (MCT). Στο Κεφάλαιο 5 παρουσιάζεται η ιεραρχική μέθοδος που αναπτύξαμε (ModHSW) για διαμέριση γραφήματος σε  $K$  ανεξάρτητα υπογραφήματα η οποία χρησιμοποιεί τον αλγόριθμο εύρεσης ελάχιστης τομής των Stoer και Wagner [26]. Ο αλγόριθμος αυτός βασίζεται στη θεωρία συρρίκνωσης ακμών. Έπειτα περιγράφονται τα ζητήματα που επιλύσαμε για την εφαρμογή του στο πρόβλημα κατάτμησης εικόνας. Στο Κεφάλαιο 6 παρουσιάζεται ο αλγόριθμος ομαδοποίησης (graph clustering) με δέντρα ελάχιστης τομής (minimum cut trees) των Flake, Tarjan και Tsioutsoulouklis [29] (MCT-α). Στο Κεφάλαιο 7 παρουσιάζεται η πειραματική μελέτη που κάναμε χρησιμοποιώντας τους αλγόριθμους ModHSW, MCT και MCT-α ενώ παράλληλα αναφέρονται και τα συμπεράσματα που εξήχθησαν από την μελέτη και τα πειράματα της εργασίας.

## ΚΕΦΑΛΑΙΟ 2

# ΠΡΟΒΛΗΜΑΤΑ ΜΕΓΙΣΤΗΣ ΡΟΗΣ

---

2.1 Ορισμός του προβλήματος

2.2 Μέθοδοι Μέγιστης Ροής

2.3 Μέθοδοι Δημιουργίας Δέντρων Τομής

---

### 2.1 Ορισμός του προβλήματος

Το πρόβλημα μέγιστης ροής (maximum flow problem) έχει πολλές εφαρμογές. Μια σημαντική κλάση αυτών των προβλημάτων είναι τα προβλήματα εύρεσης ελάχιστου μονοπατιού. Η σχέση αυτών με την πιο γενική κλάση είναι ότι: Μπορεί να εξεταστεί κατά πόσο μια εφικτή λύση είναι βέλτιστη, επιλύοντας το πρόβλημα ελαχίστου μονοπατιού.

Έστω ότι πρόκειται να σταλούν όσο το δυνατό περισσότερα φορτηγά από ένα σημείο  $s$  σε ένα δίκτυο δρόμου σε ένα άλλο σημείο  $t$ . Ο περιορισμός που υπάρχει, είναι ότι σε ένα κομμάτι δρόμου  $e$ , υπάρχει ένα άνω όριο  $c_e$  στο πλήθος των φορτηγών που επιτρέπεται να χρησιμοποιήσουν το  $e$ . Μοντελοποιώντας το πρόβλημα αυτό με ένα δίκτυο  $N$  τότε ανάγεται σε πρόβλημα εύρεσης μιας οικογένειας μονοπατιών  $(P_1, \dots, P_k)$  από (όχι απαραίτητα ξεχωριστά)  $(s, t)$ -κατευθυνόμενα μονοπάτια στο  $G$ , τέτοια ώστε κάθε ακμή  $e$  να συμμετέχει το πολύ σε  $c_e$  κατευθυνόμενα μονοπάτια και η τιμή  $k$  να μεγιστοποιείται. Χωρίς βλάβη της γενικότητας τα μονοπάτια  $P_i$  μπορούν να θεωρηθούν απλά (simple). Επίσης το γράφημα  $G$  είναι επίσης απλό γράφημα. Έστω  $f_e$  το πλήθος των μονοπατιών που χρησιμοποιούν την ακμή χωρητικότητας  $c_e$ . Μια άλλη σημαντική παρατήρηση είναι ότι για κάθε κόμβο  $v \neq s, t$ ,

κάθε μονοπάτι  $P_i$  πρέπει να εισέρχεται και να εξέρχεται του  $v$  τον ίδιο αριθμό φορών. Γι' αυτό, το  $f$  ικανοποιεί τα εξής:

### Διατήρηση Ροής

$$\sum (x_{vw} : w \in V, vw \in E) - \sum (x_{vw} : w \in V, vw \in E) = 0, \quad \forall v \in V \setminus \{s, t\} \quad (2.1)$$

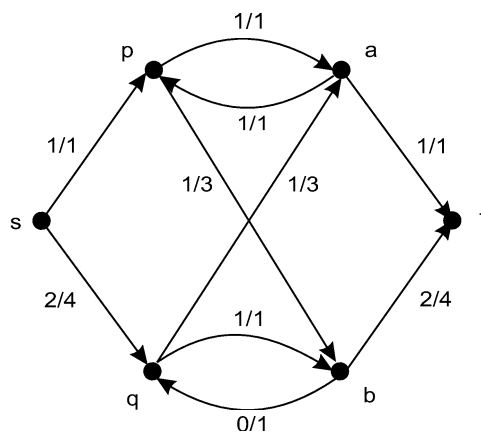
### Περιορισμός Χωρητικότητας

$$0 \leq f_{vw} \leq c_{vw}, \quad \forall vw \in E \quad (2.2)$$

$$f_{vw} \text{ ακέραιος}, \quad \forall vw \in E \quad (2.3)$$

Επιπλέον το πλήθος  $k$  των κατευθυνόμενων μονοπατιών ικανοποιεί τη σχέση  $k = \sum (f_{wt} : w \in V, wt \in E) - \sum (f_{tw} : w \in V, tw \in E)$ . Για παράδειγμα, στο Σχήμα 2.1 τα κατευθυνόμενα μονοπάτια  $\{s, p, a, t\}$ ,  $\{s, q, b, t\}$  και  $\{s, q, a, p, b, t\}$  ικανοποιούν τους περιορισμούς που δίνονται από το  $c_e$ . Κάθε ακμή του γραφήματος σημειώνεται με  $f_e/c_e$ . Έστω  $f$  ένα διάνυσμα  $(s, t)$ -ροής, το  $f$  ονομάζεται νόμιμη ροή (legal flow) αν ικανοποιεί τις συνθήκες 2.1 και 2.2. Το αριστερό μέρος της συνθήκης 2.1 δηλώνει τη ροή δικτύου στον κόμβο  $v$ , η διαφορετικά την ποσότητα excess του  $f$  στο  $v$  και συμβολίζεται με  $f(v)$ . Η συνθήκη  $f(v) = 0$  αναγκάζει τη διατήρηση της ροής στον κόμβο  $v$ . Για τους ειδικούς κόμβους  $s, t$  που αποτελούν αντίστοιχα τον κόμβο πηγή και τον κόμβο προορισμό δεν απαιτείται διατήρηση ροής. Η ποσότητα  $f(t)$  αποτελεί και τη σύνολο ροής του δικτύου.

Υπάρχει μια οικογένεια  $(P_1, \dots, P_k)$  από  $(s, t)$ -κατευθυνόμενα μονοπάτια τέτοια ώστε  $|\{i : P_i \text{ χρησιμοποιεί την ακμή } e\}| \leq c_e$  για κάθε ακμή  $e \in E$  αν και μόνο αν υπάρχει ένα νόμιμη ακέραια  $(s, t)$ -ροή τιμής  $k$ .



Σχήμα 2.1: Νόμιμη ροή τιμής 3.

Υπάρχει ένας φυσικός τρόπος εύρεσης των άνω ορίων της μέγιστης τιμής μιας ροής. Έστω το σύνολο  $\delta(R) = \{uv : uv \in E, v \in R, w \notin R\}$  για κάποιο  $R \subseteq V$ , μια τομή. Εύκολα παρατηρεί κανείς ότι η ποσότητα  $\delta$  και "cut" διαφέρουν από το αντίστοιχο μη κατευθυνόμενο γράφημα. Μια  $(s, t)$ -τομή είναι μια τομή κατά την οποία  $s \in R$  και  $t \notin R$ . Για κάθε σύνολο  $A \subseteq V$  ο συμβολισμός  $\bar{A}$  χρησιμοποιείται για να δηλώσει το σύνολο  $V \setminus A$ . Τέλος, για κάθε  $v \in V$ , χρησιμοποιείται  $\delta(v)$  ως συντομογραφία του  $\delta(\{v\})$ , και  $\delta(\bar{v})$  ως συντομογραφία του  $\delta(\{\bar{v}\})$ .

Αποδεικνύεται εύκολα ότι για κάθε  $(s, t)$ -τομή  $\delta(R)$  και κάθε  $(s, t)$ -ροή  $f$ , ισχύει ότι  $f(\delta(R)) - f(\delta(\bar{R})) = f(t)$ . Για παράδειγμα, έχοντας τη ροή  $f$  που φαίνεται στο Σχήμα 2.1 και την τομή  $R = \{s, p, q\}$  η πιο πάνω σχέση ικανοποιείται αφού  $f(\delta(R)) - f(\delta(\bar{R})) = 4 - 1 = 3$ .

Για κάθε  $(s, t)$ -ροή  $f$  και κάθε  $(s, t)$ -τομή  $\delta(R)$ , ισχύει  $f(t) \leq c(\delta(R))$ . Η απόδειξη της σχέσης αυτής προκύπτει αμέσως δεδομένου ότι  $f(\delta(R)) \leq c(\delta(R))$  και  $f(\delta(R)) \geq 0$ . Για παράδειγμα για τομή  $R = \{s, p, q\}$  στο Σχήμα 2.1, το  $f_t \leq 8$ . Συνεπώς η τιμή της μέγιστης ροής είναι φραγμένη από το άθροισμα των χωρητικοτήτων των ακμών που συμμετέχουν στην ελάχιστη τομή (minimum cut capacity). Επομένως η εύρεση μιας ροής και μιας τομής ώστε η τιμή της ροής να ισούται με το συνολικό βάρος της τομής τότε η ροή αυτή είναι και η μέγιστη. Αυτό συμβαίνει πάντοτε σύμφωνα με το θεώρημα *Max-flow Min-Cut* το οποίο διατυπώθηκε από τους Ford και Fulkerson [22].

**Θεώρημα 2.1.** *Αν υπάρχει μέγιστη  $(s, t)$ -ροή, σε ένα δίκτυο  $N$ , τότε ισχύει η σχέση:*  
 $\max\{f(t) : \text{μια νόμιμη } (s, t)\text{-ροή}\} = \min\{c(\delta(R)) : \delta(R) \text{ μια } (s, t)\text{-τομή}\}$

Η απόδειξη του θεωρήματος 2.1 δίνεται με μια βασική ιδέα ενός αλγορίθμου επίλυσης των δύο προβλημάτων βελτιστοποίησης.

Ορίζουμε ένα αυξανόμενο μονοπάτι (incrementing path) το στο οποίο όλες οι ακμές  $e$ , με κατεύθυνση προς τον κόμβο  $t$ , έχουν ροή  $f_e < c_e$  και κάθε ακμή  $e$ , με κατεύθυνση προς τον κόμβο  $s$ , έχει ροή  $f_e > 0$ . Ένα ακόρεστο μονοπάτι (augmenting path) είναι ένα  $(s, t)$ -μονοπάτι το οποίο είναι αυξανόμενο. Έχοντας ένα μονοπάτι το οποίο είναι ακόρεστο τότε όλες οι ακμές με κατεύθυνση προς τον κόμβο  $t$  αυξάνονται κατά μια ποσότητα  $\varepsilon$  και κάθε ακμή με κατεύθυνση προς τον κόμβο  $s$  μειώνεται αντίστοιχα κατά μια ποσότητα  $\varepsilon$ . Η διαδικασία αυτή θα οδηγήσει σε μια ροή μεγαλύτερης τιμής. Για παράδειγμα στο Σχήμα 2.1 το μονοπάτι  $\{s, q, a, p, b, t\}$  είναι ακόρεστο. Αν σε αυτό το μονοπάτι εφαρμοστεί η πιο πάνω διαδικασία, ενισχύονται όλες τις ακμές, με κατεύθυνση προς τον κόμβο  $t$ , κατά τη ποσότητα  $\varepsilon = 1$  τότε αυτό θα είχε ως αποτέλεσμα μια ροή με μεγαλύτερη τιμή ίση με 4



(και όχι 3). Επιπλέον επειδή υπάρχει τομή  $\delta(\{s, q, a\})$  με συνολικό βάρος ίσο με 4 τότε με βάση τη σχέση  $f_t \leq c(\delta(R))$  η μέγιστη τιμή ροής έχει τιμή 4.

Σύμφωνα με τη σχέση  $f_t \leq c(\delta(R))$ , αυτό που χρειάζεται μόνο για την απόδειξη του θεωρήματος είναι η εύρεση μιας ροής  $f$  και μιας τομής  $\delta(R)$  ώστε να ισχύει  $f_t = c(\delta(R))$ . Έστω  $f$  μια ροή με μέγιστη τιμή και ένα σύνολο  $R$  για το οποίο ισχύει η ακόλουθη σχέση:  $R = \{v \in V : \text{υπάρχει αυξανόμενο μονοπάτι από τον } s \text{ στο } v\}$ . Δηλαδή το σύνολο  $R$  περιέχει κόμβους του συνόλου  $V$  για τους οποίους υπάρχει αυξανόμενο μονοπάτι από τον κόμβο  $s$  προς αυτούς. Είναι προφανές ότι, ο  $s \in R$ , και επιπλέον  $t \notin R$ . Για κάθε ακμή  $uv$  που ανήκει στο  $\delta(R)$  θα πρέπει να ισχύει  $f_{uv} = c_{uv}$  σε διαφορετική περίπτωση η προσθήκη της ακμής  $uv$  στο αυξανόμενο μονοπάτι από τον κόμβο  $s$  στον  $v$  θα σήμαινε ότι υπάρχει αυξανόμενο μονοπάτι από τον κόμβο  $s$  στον  $w$  το οποίο είναι άτοπο αφού ο  $w \notin R$ . Ομοίως, για κάθε ακμή  $vw \in \delta(\bar{R})$  ισχύει  $f_{vw} = 0$ . Επομένως, η απόδειξη ολοκληρώθηκε αφού  $f(t) = f(\delta(R)) - f(\delta(\bar{R})) = c(\delta(R))$ .

**Θεώρημα 2.2.** *Μια ροή  $f$  είναι μέγιστη αν και μόνο αν δεν υπάρχουν άλλα ακόρεστα μονοπάτια.*

Προφανώς, αν  $f$  η μέγιστη ροή τότε δεν υπάρχουν άλλα ακόρεστα μονοπάτια. Αν δεν υπάρχουν άλλα ακόρεστα μονοπάτια, τότε με βάση τα όσα αναφέρθηκαν κατά την απόδειξη του Θεωρήματος 2.2, αυτό οδηγεί σε μια τομή  $\delta(R)$  με  $f(t) = c(\delta(R))$ , επομένως η ροή  $f$  είναι μέγιστη.

## 2.2 Μέθοδοι Μέγιστης Ροής

Στην ενότητα αυτή παρουσιάζονται κάποιοι αλγόριθμοι υπολογισμού μέγιστης ροής.

### 2.2.1 Αλγόριθμος Ford-Fulkerson

Οι Ford και Fulkerson ήταν οι πρώτοι που διατύπωσαν έναν αλγόριθμο για τον υπολογισμό των μέγιστων ροών, που είναι πλέον κλασσικός στην θεωρία γραφημάτων. Η μέθοδος των Ford-Fulkerson, βρίσκει μονοπάτια από την πηγή  $s$  στο προορισμό  $t$ , τα οποία δεν χρησιμοποιούνται κατά τον καλύτερο τρόπο. Δηλαδή, για κάθε τόξο του μονοπατιού ισχύει  $f(a) < c(a)$ . Έτσι, στη συνέχεια, προσπαθεί επαναληπτικά να αυξήσει τη ροή δια μέσου των μονοπατιών και για το λόγο αυτό τα μονοπάτια αυτά ονομάζονται ακόρεστα (augmenting paths).

Κάθε ακόρεστο μονοπάτι χρησιμοποιείται για την αύξηση της συνολικής ροής στο δίκτυο. Η διαδικασία επαναλαμβάνεται με την εύρεση νέων ακόρεστων μονοπατιών και αντίστοιχα την αύξηση της συνολικής ροής σε κάθε ένα από αυτά. Η συνολική ροή είναι μέγιστη αν και μόνο αν δεν υπάρχει κανένα ακόρεστο μονοπάτι από τον κόμβο πηγή στον κόμβο προορισμό σύμφωνα με το Θεώρημα 2.2.

Ο αλγόριθμος που περιγράφηκε είναι κατανοητός και η υλοποίησή του είναι σχετικά εύκολη. Με απλά λόγια μπορεί να συνοψισθεί ως εξής: “ξεκίνα με μηδενική αρχική ροή σε κάθε μονοπάτι χωρίς κορεσμένα τόξα με κατεύθυνση προς τον κόμβο προορισμό και χωρίς τόξα μηδενικής ροής με κατεύθυνση προς τον κόμβο πηγή μέχρι να μην υπάρχουν τέτοια μονοπάτια στο δίκτυο”. Ο αλγόριθμος παρουσιάζεται πιο κάτω.

FORDFULKERSON  $N(V, A, s, t, c), f(a) = 0 \forall a \in A$

1. Εντοπίζεται ένα ακόρεστο μη κατευθυνόμενο μονοπάτι  $P$  από τον κόμβο  $s$  προς τον κόμβο  $t$ .

2. **Αν** δεν υπάρχει τέτοιο ακόρεστο μονοπάτι, τότε  $f^* = f$  και ο αλγόριθμος τερματίζει (και επιστρέφει  $f^*$ ) επειδή η συνολική ροή είναι η μέγιστη

**αλλιώς**

υπολογίζεται το μικρότερο περιθώριο  $\Delta$  των τόξων του μονοπατιού  $P$

**Για κάθε** τόξο  $(u, v)$  του  $P$  εκτελείται η εντολή:

**Αν** τόξο  $(u, v)$  έχει κατεύθυνση  $(s, t)$ , τότε:

$$f(u, v) \leftarrow f(u, v) + \Delta$$

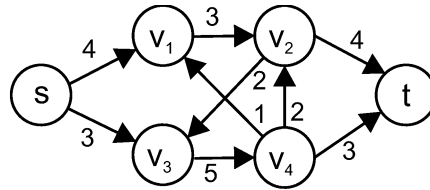
**αλλιώς**  $f(u, v) \leftarrow f(u, v) - \Delta$

3. Τίθεται  $f^* = f$  επαναλαμβάνεται το Βήμα 1

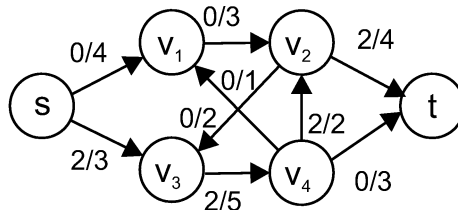
Αλγόριθμος 1: Αλγόριθμος των Ford-Fulkerson με ακόρεστα μονοπάτια.

**Παράδειγμα Ford-Fulkerson** Έστω το δίκτυο του Σχήματος 2.2, με πηγή τον κόμβο  $s$  και προορισμό τον κόμβο  $t$ . Λαμβάνεται αρχικά το μονοπάτι  $P_1 = (s, v_3, v_4, v_2, t)$ . Όλα τα τόξα έχουν κατεύθυνση προς τον κόμβο  $t$ . Το περιθώριο αύξησης ροής είναι η μικρότερη διαφορά  $\Delta$  όλων των ακμών στο μονοπατιού  $P_1$ , δηλαδή 2. Η ποσότητα αυτή είναι η αρχική ροή. Στο Σχήμα 2.3 παρουσιάζεται το δίκτυο όπου κάθε τόξο χαρακτηρίζεται από ένα ζεύγος τιμών  $f(a)/c(a)$ . Κατόπιν, λαμβάνεται το δεύτερο μονοπάτι  $P_2 = (s, v_1, v_2, v_4, t)$  μια ακμή που περιέχει μία ακμή  $(v_4, v_2)$  με κατεύθυνση προς την πηγή. Το ελάχιστο περιθώριο  $\Delta$  είναι και πάλι 2, οπότε οι ροές ενημερώνονται και λαμβάνεται το δίκτυο του Σχήματος 2.4. Το τρίτο μονοπάτι είναι το  $P_3 = (s, v_1, v_2, t)$ , όπου το περιθώριο  $\Delta$  είναι 1. Έτσι,

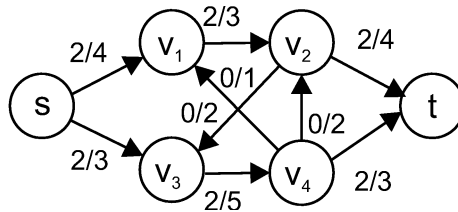
ενημερώνονται οι ροές και λαμβάνεται το δίκτυο του Σχήματος 2.5. Τέλος, ένα τέταρτο μονοπάτι  $P_4 = (s, v_3, v_4, t)$ , όπου το περιθώριο είναι και πάλι 1. Έτσι, ενημερώνονται οι ροές και λαμβάνεται το δίκτυο του Σχήματος 2.6. Στο δίκτυο αυτό, το τόξο  $(s, v_3)$  είναι κορεσμένο, όχι όμως το τόξο  $(s, v_1)$ . Βέβαια, δε μπορεί να αυξηθεί η συνολική ροή γιατί τα άλλα προσπίπτοντα τόξα στην κορυφή  $v_1$  είναι είτε κορεσμένα, όπως για παράδειγμα το  $(v_1, v_2)$ , είτε έχουν κατεύθυνση προς την κορυφή  $v_1$  και έχουν μηδενική ροή, όπως δηλαδή το  $(v_4, v_1)$ .



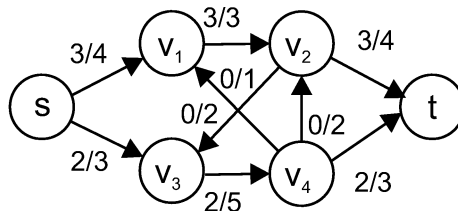
Σχήμα 2.2: Παράδειγμα δικτύου  $N$



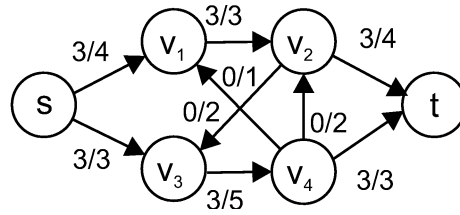
Σχήμα 2.3: Στο ακόρεστο μονοπάτι  $P_1 = \{s, v_3, v_4, v_2, t\}$  περνά ροή ίση με  $\Delta = 2$ .



Σχήμα 2.4: Στο ακόρεστο μονοπάτι  $P_2 = \{s, v_1, v_2, v_4, t\}$  περνά ροή ίση με  $\Delta = 2$ .

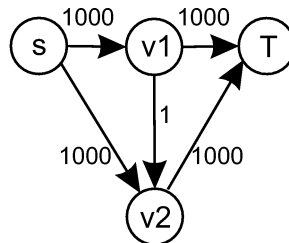


Σχήμα 2.5: Στο ακόρεστο μονοπάτι  $P_3 = \{s, v_1, v_2, t\}$  περνά ροή ίση με  $\Delta = 1$ .



Σχήμα 2.6: Στο ακόρεστο μονοπάτι  $P_4 = \{s, v_3, v_4, t\}$  περνά ροή ίση με  $\Delta = 1$ .

Έστω το δίκτυο στο Σχήμα 2.7, με πηγή τον κόμβο  $S$  και προορισμό τον κόμβο  $t$ . Αρχικά λαμβάνονται τα ακόρεστα μονοπάτια  $P_1 = (s, v_1, v_2, t)$  και  $P_2 = (s, v_2, v_1, t)$ . Κάθε φορά το ελάχιστο περιθώριο είναι  $\Delta = 1$ . Συνεπώς, σε κάθε επανάληψη τα τόξα με χωρητικότητα  $c = 1000$  δέχονται επιπλέον ροή σε κάθε επανάληψη ίση με  $\Delta = 1$ , οπότε η λύση θα επιτευχθεί μετά από 2000 επαναλήψεις. Αντίθετα, αν επιλέγονταν ως ακόρεστα μονοπάτια τα  $P_1 = (s, v_1, t)$  και  $P_2 = (s, v_2, t)$ , τότε η λύση θα επιτυγχανόταν μετά από δύο βήματα. Ο αλγόριθμος αυτός λοιπόν δεν είναι πολυωνυμικός αλλά η επίδοση του εξαρτάται από τα μεγέθη των χωρητικοτήτων. Εύκολα ο αναγνώστης μπορεί να συμπεράνει ότι η πολυπλοκότητα αυτής της μεθόδου είναι  $O(|A|F)$ . Σημειώνεται επίσης ότι αν οι τιμές των χωρητικοτήτων είναι μη πραγματικοί αριθμοί, τότε η μέθοδος μπορεί να εκτελείται επ' άπειρον.



Σχήμα 2.7: Αντιπαράδειγμα εφαρμογής μεθόδου Ford-Fulkerson

### 2.2.2 Αλγόριθμος Edmonds-Karp

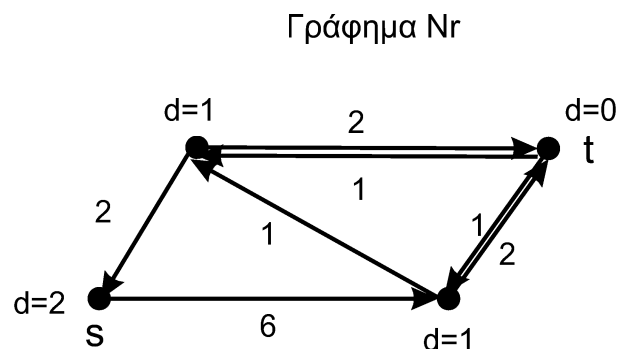
Οι Edmonds και Karp [8] τελειοποίησαν τον αλγόριθμο των Ford-Fulkerson, προτείνοντας μια μέθοδο εύρεσης των ακόρεστων μονοπατιών που βασίζεται στην αναζήτηση με προτεραιότητα πλάτους. Ο τρόπος δηλαδή με τον οποίο υπολογίζεται μια μέγιστη ροή είναι ίδιος με τον αλγόριθμο Ford-Fulkerson, απλά αλλάζει ο τρόπος με τον οποίο επιλέγουν ένα ακόρεστο μονοπάτι.

Η αναζήτηση ακόρεστων μονοπατιών γίνεται εφαρμόζοντας τον αλγόριθμο BFS στο γράφημα  $N_r$  που απεικονίζει το εναπομένον (residual) δίκτυο  $N$  (παραπέμπουμε τον ανα-

γνώστη στην Ενότητα 1.2). Με τον τρόπο αυτό θέτονται επιγραφές αποστάσεων (distance labels)  $d$  σε όλους τους κόμβους του γραφήματος. Πιο συγκεκριμένα, σε κάθε κόμβο  $i$ , τίθεται μια τιμή  $d(i)$  με μια συνάρτηση ανάθεσης αποστάσεων (distance function). Η συνάρτηση είναι έγκυρη, αν  $d(s) = 0$  και  $d(i) \leq d(j) + 1$  για κάθε ακμή  $ij$  στο εναπομένον γράφημα. Για παράδειγμα, εφαρμόζοντας την συνάρτηση ανάθεσης αποστάσεων στο εναπομένον γράφημα  $N_r$  που φαίνεται στο Σχήμα 1.4 παίρνουμε για κάθε κόμβο την επιγραφή που φαίνεται στο Σχήμα 2.8.

Στη συνέχεια, η αναζήτηση του ελάχιστου (το μονοπάτι με το μικρότερο δυνατόν πλήθος ακμών) ακόρεστου μονοπατίου (shortest augmenting path), γίνεται επιλέγοντας κάθε φορά μια ακμή  $ij$  στο  $N_r$  η οποία ικανοποιεί την ιδιότητα  $d(i) = d(j) + 1$  (admissible ακμή). (Το μονοπάτι που αποτελείται μόνο από admissible ακμές είναι και το ελάχιστο ακόρεστο μονοπάτι από τον κόμβο πηγή στον κόμβο προορισμό). Έπειτα, αυξάνεται η ροή κατά μήκος των ελάχιστων κατευθυνόμενων μονοπατιών από τον κόμβο πηγή στον κόμβο προορισμό.

Ο αλγόριθμος των Edmonds-Karp είναι πολυωνυμικός και έχει πολυπλοκότητα  $O(|V||A|^2)$ .



Σχήμα 2.8: Ανάθεση επιγραφών τους κόμβους του  $N_r$

### 2.2.3 Αλγόριθμος Push-Relabel

Πριν το 1988, όλοι γνωστοί αλγόριθμοι που είχαν προταθεί για την επίλυση του προβλήματος μέγιστης ροής βασίζονταν στην εύρεση ακόρεστων μονοπατιών. Μια εναλλακτική μέθοδος βασίζεται σε στην τεχνική preflow, η οποία προτάθηκε αρχικά από τον Karzanov. Μια preflow είναι μια ροή (flow), με τη διαφορά ότι το συνολικό ποσό ροής που εισέρχεται σε έναν κόμβο επιτρέπεται να είναι μεγαλύτερο από το συνολικό ποσό ροής που εξέρχεται του κόμβου αυτού. Άλλοι αλγόριθμοι που χρησιμοποίησαν την preflow τεχνική είναι ([14],[15]). Η πολυπλοκότητα και των τριών αυτών αλγορίθμων είναι  $O(n^3)$ . Στο σημείο αυτό δίνουμε

δύο βασικούς ορισμούς.

**Συνάρτηση Preflow** Η συνάρτηση αυτή αποτελεί χαλάρωση της συνάρτησης ροής και ικανοποιεί τους πιο κάτω περιορισμούς:

- $f(i, j) \leq c(i, j)$
- $\sum_{\{i,j\} \in E} f(i, j) - \sum_{\{k,j\} \in E} f(k, j) \geq 0, i \in V - \{s, t\}$

**Excess ενός κόμβου**

- $e(i) = \sum_{\{k,i\} \in E} f(k, i) - \sum_{\{i,j\} \in E} f(i, j)$

**Ενεργός κόμβος είναι ο κόμβος για τον οποίο ισχύει η σχέση:**

- $e(i) > 0$

Ο αλγόριθμος δέχεται ως είσοδο ένα δίκτυο  $N(V, A)$  και αρχικά θέτει τιμές σε κάθε ακμή  $a \in A$  με μια συνάρτηση preflow. Πιο συγκεκριμένα, δίνει σε κάθε ακμή που εξέρχεται του κόμβου πηγή, τιμή ροής ίση με την χωρητικότητα της ακμής. Αντίθετα σε κάθε άλλη ακμή, δίδεται η τιμή μηδέν. Επίσης ο αλγόριθμος αρχικά, δίδει επιγραφές απόστασης  $d$  σε όλους τους κόμβους του δικτύου. Ένας τρόπος να δοθούν επιγραφές απόστασης στους κόμβους είναι:

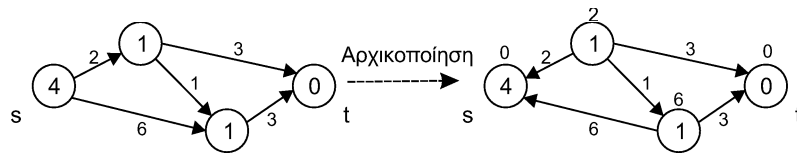
1. Αρχικοποιούνται όλοι οι κόμβοι με την επιγραφή  $n = |V|$ , εκτός από τον κόμβο προορισμό που αρχικοποιείται με την τιμή μηδέν.
2. Εφαρμόζεται η διαδικασία backward BFS η οποία αρχίζει την αναζήτηση από τον κόμβο προορισμό  $t$ .
3. Αναθέτει σε ένα κόμβο  $i$  επιγραφή  $d(i) = \min(d(j) + 1)$  αν  $ij \in E(N_r), r(i, j) > 0$ . Όπου  $E(N_r)$  είναι οι ακμές στο εναπομένον γράφημα προκύπτουν από το δίκτυο ροής  $N$ .
3. Αναθέτει στον κόμβο πηγή την επιγραφή  $d(s) = n$ .

Στη συνέχεια, ο αλγόριθμος επαναληπτικά εφαρμόζει τις διαδικασίες προώθησης (push) και επαναπροσδιορισμό επιγραφής (relabel), μέχρις ότου δεν υπάρχει κόμβος ενεργός στο  $N_r$ .

Κάθε φορά δηλαδή που επιλέγεται ένας ενεργός κόμβος  $i$ , εξετάζεται αν προσπίπτει σε αυτόν μια admissible ακμή (βλ. Ενότητα 1.2) και προωθείται στην ακμή αυτή ροή ίση με  $f(i, j) = \min\{e(i), r(i, j)\}$ . Διαφορετικά, επαναπροσδιορίζεται η επιγραφή του κόμβου με τιμή ίση με  $d(i) = \min\{d(j) + 1\}$ , όπου  $\{i, j\} \in E(N_r)$  και  $r(i, j) > 0$ .

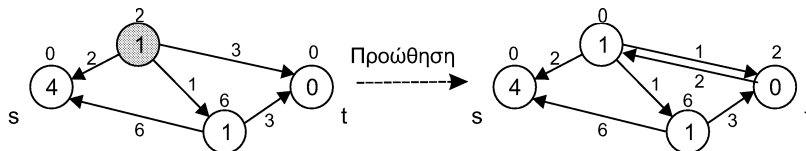
Οι διαδικασίες προώθησης και επαναπροσδιορισμού επιγραφής κόμβου, περιγράφεται με τη βοήθεια του παραδείγματος που ακολουθεί.

**Παράδειγμα** Έστω το δίκτυο που φαίνεται αριστερά στο Σχήμα 2.9. Στο πρώτο βήμα του αλγόριθμου αρχικοποιούνται οι επιγραφές των κόμβων του δικτύου με τον τρόπο που περιγράφηκε πιο πάνω. Επίσης στο βήμα αυτό προωθείται ροή από τον κόμβο πηγή  $s$  προς τους γείτονες του. Το αποτέλεσμα του πρώτου βήματος φαίνεται δεξιά στο Σχήμα 2.9.



Σχήμα 2.9: Αρχικοποίηση των επιγραφών των κόμβων του  $N_r$ , και προώθηση ροής από τον κόμβο πηγή  $s$  στους γείτονες του.

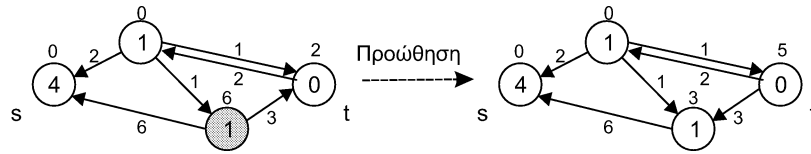
Στην συνέχεια, επιλέγεται ένας από τους ενεργούς κόμβους (υπενθυμίζουμε ότι ένας κόμβος  $i$  είναι ενεργός όταν ισχύει  $e(i) > 0$ ), εξετάζουμε αν υπάρχει admissible ακμή που να εξέρχεται του κόμβου αυτού. Στο παράδειγμα επιλέγεται ο σκιαγραφημένος κόμβος που φαίνεται στο Σχήμα 2.10. Ο κόμβος αυτός στη συνέχεια προωθεί στην admissible ακμή που προσπίπτει σε αυτόν, ροή μεγέθους  $f(i, j) = \min\{e(i), r(i, j)\} = 2$ . Ο κόμβος αυτός γίνεται ανενεργός. Σε κάθε βήμα βλέπουμε το εναπομένον (residual network) δίκτυο που προκύπτει μετά την προώθηση ροής (βλ. Ενότητα 1.2).



Σχήμα 2.10: Ο σκιαγραφημένος ενεργός κόμβος, προωθεί ροή  $f=2$  προς τον κόμβο προορισμό  $t$ .

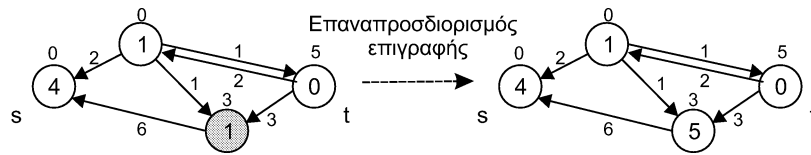
Έπειτα, επιλέγεται ο επόμενος ενεργός κόμβος, ο οποίος προωθεί ροή  $f = 3$  στην admissible ακμή που προσπίπτει σε αυτόν (Σχήμα 2.11).

Στη συνέχεια, επιλέγεται ένας ενεργός κόμβος, ο οποίος δεν έχει admissible ακμή που να προσπίπτει σε αυτόν (Σχήμα 2.12). Στην περίπτωση αυτή ο ενεργός κόμβος που



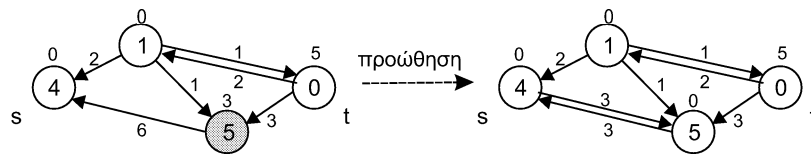
Σχήμα 2.11: Ο σκιαγραφημένος ενεργός κόμβος, προωθεί ροή  $f=3$  προς τον κόμβο προορισμό  $t$ .

επιλέγηκε δεν προωθεί ροή αλλά επαναπροσδιορίζει την επιγραφή του με βάση τον τύπο  $d(i) = \min(d(j) + 1)$ .



Σχήμα 2.12: Ο σκιαγραφημένος ενεργός κόμβος επαναπροσδιορίζει την επιγραφή του.

Ο κόμβος, που επιλέχθηκε στο προηγούμενο βήμα, αφού αλλάξει την επιγραφή του, παραμένει ενεργός, και επιλέγεται στο τελευταίο βήμα κάνοντας οπισθοδρόμηση της ροής που περίσσεψε πίσω στον κόμβο πηγή (Σχήμα 2.13).



Σχήμα 2.13: Ο σκιαγραφημένος ενεργός κόμβος, προωθεί ροή  $f=3$  προς την πηγή.

Στο Πίνακα 2.1, φαίνονται κάποιοι αλγόριθμοι εύρεσης μέγιστης ροής. Με  $C$  συμβολίζουμε το άνω φράγμα στη χωρητικότητα των ακμών του δικτύου ροής  $N$  με  $|V(N)| = n$  και  $|E(G)| = m$ .

### 2.3 Μέθοδοι Δημιουργίας Δέντρων Τομής

Τα δέντρα ελάχιστης τομής παρουσιάστηκαν αρχικά από τους Gomory και Hu [6] και είναι επίσης γνωστά ως Gomory-Hu δέντρα. Ένα τέτοιο δέντρο είναι μια συμπαγής αναπαράσταση όλων των  $(s, t)$ τομών ενός μη κατευθυνόμενου γραφήματος  $G$ . Τα δέντρα τομής έχουν πολλές εφαρμογές.

Όλοι οι γνωστοί αλγόριθμοι κατασκευής δέντρων τομής, χρησιμοποιούν μια ρουτίνα



Πίνακας 2.1: Αλγόριθμοι Εύρεσης Μέγιστης Ροής

| Αναφορά  | Πολυπλοκότητα                       |
|--|-------------------------------------|
| Dinic (1970)                                   | $O(n^2m)$                           |
| Edmonds and Karp (1972)                        | $O(nm^2)$                           |
| Karzanov (1974)                                | $O(n^3)$                            |
| Cherkasky (1977)                               | $O(n^2m^{\frac{1}{2}})$             |
| Malhotra, Pramodh Kumar and Matheshwari (1978) | $O(n^3)$                            |
| Galil (1978)                                   | $O(n^{\frac{5}{3}}m^{\frac{2}{3}})$ |
| Galil and Naamad (1978)                        | $O(nm(\log n)^2)$                   |
| Sleator and Tarjan (1980)                      | $O(nm \log n)$                      |
| Shiloach and Vishkin (1982)                    | $O(n^3)$                            |
| Gabow (1983)                                   | $O(nm \log C)$                      |
| Tarjan (1984)                                  | $O(n^3)$                            |
| Goldberg (1985)                                | $O(n^3)$                            |
| Goldberg and Tarjan (1986)                     | $O(nm \log(n^2/m))$                 |
| Ahuja and Orlin (1986)                         | $O(nm + n^2 \log C)$                |

υπολογισμού ελάχιστης  $(s, t)$ -τομής. Ένας αποτελεσματικός τρόπος εύρεσης της ελάχιστης  $(s, t)$ -τομής είναι χρησιμοποιώντας για παράδειγμα κάποιον από τους αλγόριθμους μέγιστης ροής του Πίνακα 2.1. Οι Gomory και Hu έδειξαν πως μπορεί να επιλυθεί το πρόβλημα του δέντρου τομής, χρησιμοποιώντας μόνο  $|V(G)| - 1$  υπολογισμούς ελάχιστης τομής και συρρικνώσεις γραφήματος. Είναι σημαντικό η υλοποίηση του αλγόριθμου αυτού να είναι αποδοτική. Ο Gusfield [30] πρότεινε έναν αλγόριθμο ο οποίος δε χρησιμοποιεί συρρικνώσεις γραφήματος αλλά οι  $|V(G)| - 1$  υπολογισμοί ελαχίστων  $(s, t)$ -τομών οι οποίοι εφαρμόζονται στο γράφημα εισόδου (και όχι σε συρρικνωμένο υπογράφημα). Ο αλγόριθμος του Gusfield είναι πολύ απλός και μπορεί να υλοποιηθεί μόνο με την προσθήκη μερικών γραμμών στον κώδικα μέγιστης ροής.

### 2.3.1 Αλγόριθμος Gomory-Hu

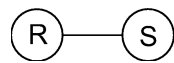
Στην ενότητα αυτή ορίζεται η "πολυτερματικό" (multiterminal) περίπτωση, όπου έχοντας ένα σύνολο από  $K$  τερματικούς κόμβους, επιθυμούμε να βρούμε ελάχιστες  $(r, s)$ -τομές για κάθε ζεύγος  $r, s \in K$ . Ένα τέτοιο πρόβλημα για παράδειγμα προκύπτει κατά τον σχεδιασμό δικτύων επικοινωνίας.

**MULTITERMINAL CUT PROBLEM**

Δοθέντος ενός μη κατευθυνόμενου γραφήματος  $G = (V, E)$ ,  $u_e \geq 0$  για κάθε ακμή  $e \in E$  και ένα σύνολο τερματικών κόμβων  $K \in V$ , βρες τις ελάχιστες  $(r, s)$ -τομές για κάθε ζεύγος κόμβων  $r, s \in K$

**Αλγόριθμος 2: Multiterminal Cut Problem**

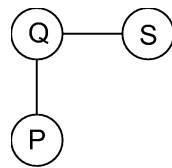
Αν το  $|K|$  είναι πολύ μεγάλο τότε είναι πολύ δύσκολο να βρεθεί η συλλογή όλων των τομών. Ωστόσο οι Gomory και Hu ανέπτυξαν μια "κομψή" λύση στο πρόβλημα. Η μέθοδος τους περιλαμβάνει την λύση μόνο  $|K| - 1$  προβλημάτων μέγιστης ροής, σε γραφήματα τα οποία δεν είναι μεγαλύτερα του  $G$  (και τις περισσότερες φορές κατά πολύ μικρότερα). Επιπλέον, η έξοδος της διαδικασίας αυτής είναι μια δενδρική δομή η οποία διευκολύνει την επιλογή μιας  $(r, s)$ -τομής η οποία αντιστοιχεί σε οποιοδήποτε ζεύγος τερματικών κόμβων. Η διαδικασία ξεκινάει με την επιλογή ενός ζεύγους τερματικών κόμβων  $r, s \in K$  και εύρεση της ελάχιστης  $(r, s)$ -τομής η οποία διαχωρίζει τους κόμβους σε δύο σύνολα  $R$  και  $S$ . Αναπαριστούμε την συγκεκριμένη τομή ως ακμή ενός δέντρου  $T$  της οποίας το ένα άκρο ανήκει στο σύνολο  $R$  και το άλλο άκρο στο σύνολο  $S$ . Το αρχικό δέντρο φαίνεται στο Σχήμα 2.14.



Σχήμα 2.14: Πρώτο βήμα του Gomory-Hu

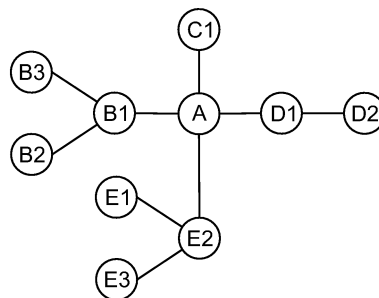
Αν τα σύνολα  $R$  και  $S$  περιέχουν ακριβώς έναν κόμβο από το σύνολο  $K$ , τότε η διαδικασία σταματά. Διαφορετικά, επιλέγεται ένας κόμβος του δέντρου, έστω ο κόμβος  $R$ , ο οποίος περιλαμβάνει δύο τερματικούς κόμβους,  $p$  και  $q$ . Τότε, κατασκευάζεται ένα γράφημα  $G_R$  από το  $G$  κάνοντας συρρίκνωση όλων των κόμβων του συνόλου  $S$  (αντικαθιστώντας δηλαδή όλους τους κόμβους του  $S$  με ένα μοναδικό κόμβο). Ακολουθεί η εύρεση της ελάχιστης  $(p, q)$ -τομής  $d(X)$  στο γράφημα  $G_R$ . Η τομή χωρίζει το σύνολο  $R$  σε δύο υποσύνολα  $Q$  και  $P$ . Στη συνέχεια δημιουργείται ένα καινούριο δέντρο με την αντικατάσταση (στο δέντρο του προηγούμενου βήματος) του κόμβου  $R$  από τους κόμβους  $Q$  και  $P$ , ενώνοντας τους δύο αυτούς κόμβους με μία ακμή βάρους ίση με την ποσότητα της τομής  $d(X)$ . Στη συνέχεια ο κόμβος  $S$  συνδέεται με τον ένα από τους δύο καινούριους κόμβους που προκύπτουν με βάση τον ακόλουθο κανόνα: Αν το σύνολο  $P$  που προκύπτει από τη τομή περιλαμβάνει τον συρρικνωμένο κόμβο που αντιστοιχεί στο σύνολο  $S$  τότε ο κόμβος  $S$  συνδέεται με τον κόμβο  $P$ , διαφορετικά ο κόμβος  $S$  συνδέεται με τον κόμβο  $Q$ . Το βάρος της ακμής αυτής δε μεταβάλλεται κατά τη διάρκεια της λειτουργίας αυτής. Το νέο δέντρο

που προκύπτει μπορεί να είναι αυτό που φαίνεται στο Σχήμα 2.15.



Σχήμα 2.15: Δεύτερο βήμα του Gomory-Hu

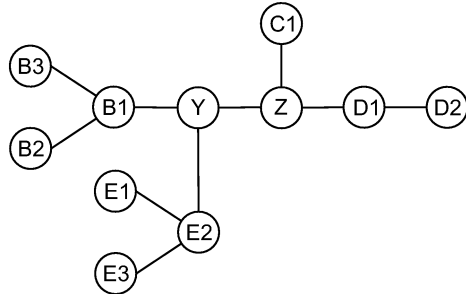
Έστω ότι μετά από μερικά βήματα (σε ένα γενικό στάδιο) του αλγορίθμου προκύπτει το δέντρο που φαίνεται στο Σχήμα 2.16.



Σχήμα 2.16: Γενικό στάδιο του Gomory-Hu

Τα σύνολα που αντιστοιχούν στους κόμβους του δέντρου αποτελούν μια διαμέριση του  $V$ . Έστω ότι κάποιο σύνολο  $A$  περιέχει τους τερματικούς κόμβους  $y, z \in K$ . Δημιουργείται το γράφημα  $G_A$  από το γράφημα  $G$ , αφού συρρικνώσουμε τους κόμβους που αντιστοιχούν στις συνιστώσες που προκύπτουν μετά την αφαίρεση του κόμβου του κόμβου  $A$  από το δέντρο. Δηλαδή, κάθε σύνολο από τα  $B1 \cup B2 \cup B3$ ,  $D1 \cup D2$  και  $E1 \cup E2 \cup E3$  συρρικνώνεται σε έναν κόμβο αντιπρόσωπο. Έπειτα ο κόμβος  $A$  διαμερίζεται στα υποσύνολα  $Y$  και  $Z$ , με βάση την ελάχιστη  $(y, z)$ -τομή στο γράφημα  $G_A$ . Έπειτα δημιουργείται ένα καινούριο δέντρο μετά την αντικατάσταση του κόμβου  $A$  στο τρέχον δέντρο με τους κόμβους  $Y$  και  $Z$ , συνδέοντας τους καινούριους κόμβους με μία ακμή βάρους ίσου με το συνολικό βάρος της τομής. Κάθε ένας από τους γείτονες  $(B1, C1, D1, E1)$  του κόμβου  $A$ , ενώνεται είτε με τον νέο κόμβο  $Y$  ή  $Z$ , σύμφωνα με το ποιο κομμάτι της τομής περιέχει τον αντίστοιχο συρρικνωμένο κόμβο. Πάλι, τα βάρη των ακμών στο δέντρο δε μεταβάλλονται κατά τη διάρκεια της λειτουργίας αυτής. Το νέο δέντρο που προκύπτει μπορεί να είναι αυτό που προκύπτει φαίνεται στο Σχήμα 2.17.

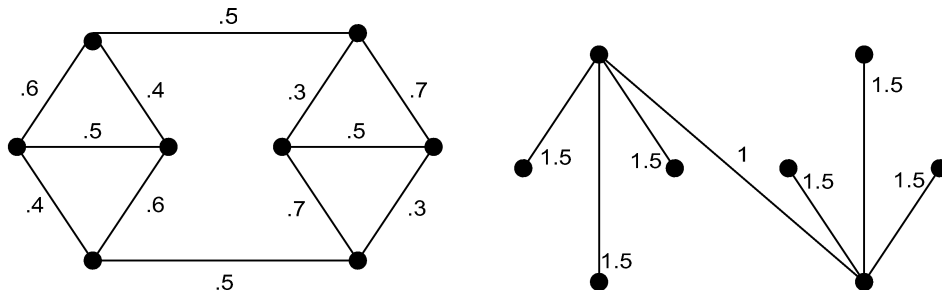
Η διαδικασία αυτή τερματίζει όταν όλοι οι κόμβοι του δέντρου  $T$  αντιστοιχούν σε σύνολα που περιέχουν μόνο ένα κόμβο. Έστω  $w_e$  το βάρος κάθε ακμής στο δέντρο.



Σχήμα 2.17: Γενικό βήμα του Gomory-Hu

**Θεώρημα 2.3.** Για κάθε ζεύγος τερματικών κόμβων  $r, s \in K$ , το συνολικό βάρος της ελάχιστης  $(r, s)$ -τομής στο γράφημα  $G$  είναι ίσο με το ελάχιστο βάρος  $w_e$  όλων των ακμών που βρίσκονται στο μονοπάτι από τον  $r$  στον  $s$  στο δέντρο  $T$ . Έστω  $e^*$  η ακμή με το ελάχιστο βάρος στο μονοπάτι. Τότε η ελάχιστη  $(r, s)$ -τομή δίνεται με τη διαμέριση του συνόλου  $V$  που αντιστοιχεί στα δύο υποδέντρα που προκύπτουν με την αφαίρεση της  $e^*$  από το δέντρο  $T$ .

Το δέντρο που προκύπτει ακολουθώντας την πιο πάνω διαδικασία αυτό ονομάζεται *Gomory-Hu  $K$ -cut-tree*. Αν  $K$  αντιστοιχεί σε όλο το σύνολο κόμβων  $V$ , τότε το " $K$ " παραλείπεται από τους συμβολισμούς. Ένα παράδειγμα ενός γραφήματος  $G$  και του αντίστοιχου του δέντρου-τομής *cut-tree*  $T$  φαίνεται στο Σχήμα 2.18.



Σχήμα 2.18: Αρχικό γράφημα και το αντίστοιχο Gomory-Hu δέντρο ελάχιστης τομής

Ο αλγόριθμος των Gomory-Hu δουλεύει καλά στην πράξη, παρόλο που αρχίζει να ξεπερνά την εκτιμώμενη πολυπλοκότητα του σε παραδείγματα προβλημάτων με περισσότερους από μερικές χιλιάδες κόμβους. Παρ' όλα αυτά η επιβάρυνση των λειτουργιών συρρίκνωσης αντισταθμίζει μερικώς τα οφέλη από το ότι ο αλγόριθμος εργάζεται σε γραφήματα μικρότερου μεγέθους από το αρχικό γράφημα  $G$ . Αυτό, σε συνδυασμό με την προγραμματιστική δυσκολία μιας καλής υλοποίησης του αλγόριθμου των Gomory-Hu, οδήγησε τον Gusfield το 1990 να αναπτύξει μια έκδοση του αλγόριθμου που αποφεύγει εντελώς τις συρρικνώσεις

κόμβων. Η μέθοδος του είναι εκπληκτικά απλή, παρόλο που το να αποδείξει κανείς την ορθότητα του είναι πιο περίπλοκο σε σύγκριση με τον αρχικό Gomory-Hu. Ένα ενδιαφέρον χαρακτηριστικό της προσέγγισης του Gusfield είναι το ότι μπορεί να λειτουργήσει χωρίς ειδική γνώση για το γράφημα  $G$ , βασιζόμενος σε μια σειρά από  $|V(G)| - 1$  κλήσεις σε μια ρουτίνα η οποία επιστρέφει μια ελάχιστη  $(r, s)$ -τομή δίνοντας της ως είσοδο τους κόμβους  $r, s$ .

## ΚΕΦΑΛΑΙΟ 3

# ΑΛΓΟΡΙΘΜΟΙ ΕΥΡΕΣΗΣ ΕΛΑΧΙΣΤΗΣ ΤΟΜΗΣ

---

3.1 Αλγόριθμος Nagamochi-Ibaraki

3.2 Αλγόριθμος Karger-Stein

3.3 Αλγόριθμος Padberg-Rinaldi

3.4 Αλγόριθμος Stoer-Wagner

3.5 Αλγόριθμος Brinkmeier

---

Το πρόβλημα ελάχιστης τομής (minimum cut) είναι ένα πρόβλημα διαμέρισης των κόμβων ενός μη κατευθυνόμενου γραφήματος  $G(V, E)$  με ακμικά βάρη, σε δύο υποσύνολα έτσι ώστε το άθροισμα των βαρών των ακμών που έχουν το ένα άκρο τους στο ένα υποσύνολο και το άλλο άκρο τους στο άλλο, να ελαχιστοποιείται.

Το πρόβλημα εύρεσης της ελάχιστης χωρητικότητας τομής (minimum capacity cut), ανάμεσα σε δύο συγκεκριμένους κόμβους,  $s$  και  $t$ , ονομάζεται πρόβλημα ελάχιστης  $(s, t)$ -τομής. Το πρόβλημα αυτό είναι στενά συνδεδεμένο με το πρόβλημα ελάχιστης τομής (minimum cut). Όπως περιγράφηκε αναλυτικά στο Κεφάλαιο 2 το πρόβλημα εύρεσης ελάχιστης  $(s, t)$ -τομής είναι ισοδύναμο με το πρόβλημα εύρεσης μέγιστης ροής. Για το λόγο αυτό, ένας αλγόριθμος υπολογισμού μέγιστης ροής μπορεί να χρησιμοποιηθεί ως βασική υπορουτίνα για την επίλυση του προβλήματος ελάχιστης τομής.

Το πρόβλημα εύρεσης ελάχιστης τομής σε ένα γράφημα  $G$  χωρίζεται σε δύο μεγάλες κατηγορίες αλγορίθμων.

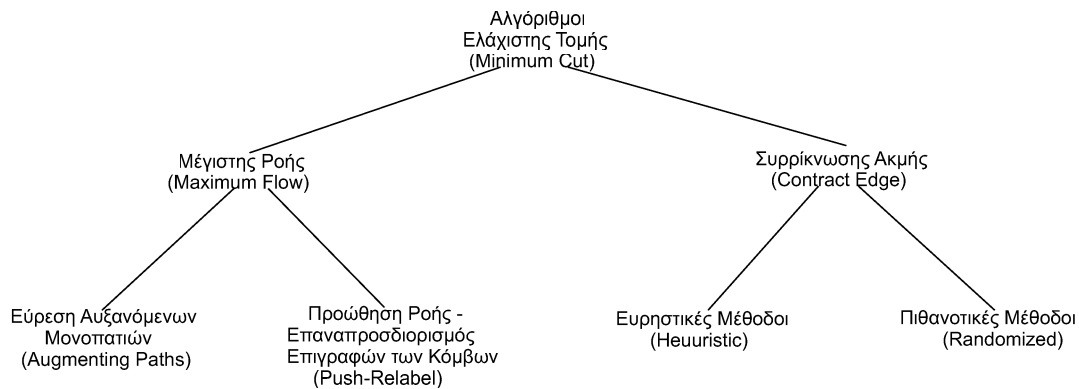
A) Η πρώτη κατηγορία αλγορίθμων επιλύει το πρόβλημα εύρεσης ελάχιστης τομής βασισζόμενη στο θεώρημα των Ford και Fulkerson το οποίο αποδεικνύει ότι η τιμή της

μέγιστης  $(s, t)$ -ροής (maximum  $(s, t)$ -flow) είναι ίση με την τιμή της ελάχιστης  $(s, t)$ -τομής (minimum  $(s, t)$ -cut), για οποιουδήποτε κόμβους  $s$  και  $t$ . Το θεώρημα αυτό αποτελεί τη βάση όλων των αλγορίθμων που εντάσσονται στην κατηγορία αυτή.

B) Η δεύτερη κατηγορία αλγορίθμων εύρεσης ελάχιστης τομής ενός γραφήματος  $G$ , βασίζεται στις επαναληπτικές συρρικνώσεις ακμών μεγάλου βάρους οι οποίες δεν συμμετέχουν στην ελάχιστη τομή.

Οι αλγόριθμοι της πρώτης κατηγορίας μπορούν να υπολογίσουν ελάχιστες τομές τόσο σε κατευθυνόμενα όσο και σε μη κατευθυνόμενα γραφήματα, ενώ οι αλγόριθμοι της δεύτερης κατηγορίας μπορούν να υπολογίσουν τομές μόνο σε μη κατευθυνόμενα γραφήματα.

Αντίστοιχα οι δύο μεγάλες αυτές κατηγορίες, μπορούν να διαιρεθούν σε μικρότερες όπως φαίνεται στο Σχήμα 3.1.



Σχήμα 3.1: Τεχνικές αλγορίθμων που επιλύουν το πρόβλημα ελάχιστης τομής.

### 3.1 Αλγόριθμος Nagamochi-Ibaraki

Στην υποενότητα αυτή αναφέρουμε εν συντομία τον αλγόριθμο των Nagamochi-Ibaraki [10]. Ο αλγόριθμος τους ποτέ δε καταφεύγει σε υπολογισμούς μέγιστης ροής. Ο αλγόριθμος τους εκτελείται σε φάσεις και διατηρεί μια τιμή  $\hat{\lambda}$ , η οποία αντιστοιχεί στη χωρητικότητα ελάχιστης τομής που έχει βρεθεί μέχρι τη δεδομένη στιγμή. Σε κάθε φάση, ο αλγόριθμος αναγνωρίζει μια ακμή  $uw$  για την οποία ικανοποιείται η σχέση  $\lambda_{u,w}(G) \geq \hat{\lambda}$ . Εφόσον μια τέτοια ακμή βρεθεί στο γράφημα, γνωρίζουμε ότι η ακμή αυτή είναι ασφαλής για συρρίκνωση. Επαναλαμβάνοντας τη διαδικασία αυτή  $n-1$  φορές, τότε λύνεται το πρόβλημα ελάχιστης τομής. Συνεπώς χρησιμοποιείται μια ρουτίνα ContractSafe, η οποία συρρικνώνει την ασφαλή

ακμή και επιστρέφει το συρρικνωμένο γράφημα  $G \setminus \{u, w\}$  (contracted graph) μαζί με τη νέα εκτίμηση της ελάχιστης τομής. Σε υψηλού επιπέδου ψευδοκώδικα, ο αλγόριθμος NI περιγράφεται πιο κάτω.

ΑΛΓΟΡΙΘΜΟΣ NI( $G(V, E)$ )  
 $\hat{\lambda} = \min\{c(u) : u \in V\}$   
**Ενόσω** ( $|V| \geq 3$ )  
      $(G, \hat{\lambda}) \leftarrow \text{ContractSafe}(G, \hat{\lambda})$   
**Επέστρεψε**  $\hat{\lambda}$

### Αλγόριθμος 3: Αλγόριθμος Nagamochi-Ibaraki

Η ρουτίνα *ContractSafe* υλοποιείται με αναζήτηση στο γράφημα  $G(V, E)$ , ξεκινώντας από ένα οποιοδήποτε κόμβο του γραφήματος. Αρχικά θεωρούμε ότι οι ακμές του γραφήματος δεν έχουν σαρωθεί (scanned = FALSE) και ότι οι κόμβοι του γραφήματος δεν έχουν επισκεφθεί (visited = FALSE). Ο αλγόριθμος, διατηρεί μεταβλητές  $r(u)$  για κάθε κόμβο  $u$  και  $q(e)$  για κάθε ακμή  $e$ , όπου το  $r(u)$  είναι το άθροισμα των χωρητικότητων των ακμών ανάμεσα στον κόμβο  $u$  και τους κόμβους που έχουμε ήδη επισκεφθεί και  $q(e)$  (για κάθε ακμή  $uw$ ) είναι η τιμή του  $r(w)$  όταν η ακμή  $e$  σαρώνεται με αρχικό κόμβο τον κόμβο  $u$ . Η αναζήτηση πάντα επιλέγει έναν κόμβο  $u$  τον οποίο δεν έχει ακόμη επισκεφθεί με τη μέγιστη τιμή  $r(u)$  και σαρώνει όλες τις εξερχόμενες ακμές του. Έστω  $x, y$  ο προτελευταίος και ο τελευταίος κόμβος που έχουμε επισκεφθεί, αντίστοιχα. Όπως έχειδειχθεί στο [10], η τετριμμένη τομή  $\{y\}$  είναι η ελάχιστη  $(x, y)$ -τομή, και συνεπώς η τιμή του  $\hat{\lambda}$  ανανεώνεται με την ελάχιστη ποσότητα από τα  $\hat{\lambda}$  και  $c(y)$  καθώς επίσης η ακμή  $xy$  συρρικνώνεται. Η ποσότητα  $c(y)$  εκφράζει το άθροισμα των βαρών των ακμών που προσπίπτουν στον κόμβο  $y$ . Συνεπώς σε κάθε φάση συρρικνώνονται τουλάχιστον μια ακμές και απαιτούνται το πολύ  $n - 2$  επαναλήψεις της ρουτίνας *ContractSafe* μέχρι να προκύψει γράφημα με δύο μόνο κόμβους.

Σε κάθε εκτέλεση της ρουτίνας *ContractSafe* συρρικνώνεται μια (τουλάχιστον) ακμή και αυτή είναι η τελευταία ακμή που σαρώνεται. Μια εκτέλεση της *ContractSafe* απαιτεί  $O(m + n \log n)$  και επομένως η ασυμπτωτική πολυπλοκότητα του αλγορίθμου είναι  $O(mn + n^2 \log n)$ . Οι ίδιοι πρότειναν μια δεύτερη υλοποίηση το 1994 [9] η οποία ενσωματώνει δύο επιπλέον ευριστικές.

**Πρώτη ευριστική** Συρρικνώνεται κάθε ακμή  $e = uw$  με  $q(e) \geq \hat{\lambda}$ , αφού ουσιαστικά το  $q(e)$  αποτελεί ένα κάτω φράγμα της ποσότητας  $\lambda_{u,w}(G)$ . Η παρατήρηση αυτή επιταχύνει



ΡΟΥΤΙΝΑ CONTRACTSAFE( $G(V,E),\hat{\lambda}$ )

**Για κάθε**  $u \in V$

$r(u) \leftarrow 0$

mark  $u$  unvisited

**Για κάθε**  $e \in E$

mark  $e$  unscanned

**Ενόσω** υπάρχει κόμβος unvisited επέλεξε τον κόμβο  $u$  με το μεγαλύτερο  $r(u)$

$\alpha \leftarrow \alpha + c(u) - 2r(u)$  (\*)

$\hat{\lambda} \leftarrow \min\{\alpha, \hat{\lambda}\}$

**Για κάθε** unscanned  $e = \{u, w\}$

$r(w) \leftarrow r(u) + c(u, w)$

$q(e) \leftarrow r(w)$

mark  $e$  scanned

mark  $u$  visited

**Για κάθε**  $e = \{u, w\}$  με  $q(e) \geq \hat{\lambda}$  (\*\*)

$G \leftarrow G / (u, w)$  με νέο κόμβο  $u'$

$\hat{\lambda} = \min\{c(u'), \hat{\lambda}\}$

**Επέστρεψε**  $\hat{\lambda}$

#### Αλγόριθμος 4: Ρουτίνα ContractSafe

τον αλγόριθμο, εφόσον κάθε επιπρόσθετη συρρίκνωση ακμής μειώνει τον αριθμό των απαιτούμενων φάσεων κατά ένα. Επιπλέον ο αλγόριθμος υπολογίζει τις ποσότητες  $q$  ούτως η αλλιώς και επομένως η εφαρμογή της ευριστικής δεν έχει κανένα υπολογιστικό κόστος. Ο κώδικας της ευριστικής αυτής φαίνεται στο σημείο (\*\*) παρακάτω. Επομένως χρησιμοποιώντας την πρώτη ευριστική αποφεύγουμε τους υπολογισμούς μέγιστης  $(u, w)$ -ροής για να αποφασίσουμε κατά πόσο μια ακμή  $uw$  μπορεί να συρρικνωθεί ή όχι.

**Δεύτερη ευριστική** Βασίζεται στην παρατήρηση ότι κατά τη προσπέλαση κάποιου κόμβου  $u$ , ένα συνδεδεμένο σύνολο κόμβων  $V'$  έχει ήδη προσπελαστεί. Ανιχνεύοντας κάθε φορά μια τομή που ορίζεται κάθε φορά από το σύνολο  $V'$  της οποίας η ποσότητα  $c(V')$  είναι μικρότερη από  $\hat{\lambda}$ , τότε η τιμή του  $\hat{\lambda}$  ενημερώνεται ανάλογα. Ο κώδικας για τη συγκεκριμένη ευριστική ονομάζεται  $\alpha$ -heuristic και φαίνεται στο σημείο (\*) στη ρουτίνα *ContractSafe*.

### 3.2 Αλγόριθμος Karger-Stein

Ο αλγόριθμος των Karger-Stein [20] (KS), παρόμοια με τον αλγόριθμο των NI, επαναληπτικά συρρικνώνει ακμές. Αντίθετα με τον αλγόριθμο NI, ο αλγόριθμος KS μπορεί κάποιες φορές να συρρικνώσει ακμή η οποία να ανήκει στην ελάχιστη τομή. Συγκεκριμένα ο KS επιλέγει κάθε φορά τυχαία την ακμή που πρόκειται να συρρικνώσει. Η ιδέα στην οποία βασίζεται ο συγκεκριμένος αλγόριθμος είναι ότι σε ένα γράφημα με  $n$  κόμβους και τιμή ελάχιστης τομής  $c$ , ο ελάχιστος βαθμός του γραφήματος είναι τουλάχιστον  $c$  και οι ακμές του γραφήματος είναι τουλάχιστον  $\frac{nc}{2}$ . Προκύπτει ότι μια ακμή που επιλέγεται τυχαία, έχει πιθανότητα να συμμετέχει στην ελάχιστη τομή είναι το πολύ  $2/n$ . Συνεπώς η πιθανότητα η συγκεκριμένη ακμή να συμμετέχει στην ελάχιστη τομή είναι μικρή. Επομένως είναι πιθανόν να συρρικνωθεί μεγάλος αριθμός ακμών χωρίς να συρρικνωθεί η ελάχιστη τομή. Συγκεκριμένα, αν σε κάθε φάση συρρικνώνεται μια ακμή, τότε η ελάχιστη τομή του γραφήματος επιβιώνει με πιθανότητα  $\Omega(n^{-2})$ .

$$(1 - \frac{2}{n})(1 - \frac{2}{n-1}) \dots (1 - \frac{2}{3}) = \binom{n}{2}^{-1} = \Omega(n^{-2}).$$

Ένας προφανής αλγόριθμος χρησιμοποιώντας αυτή την ιδέα είναι η εφαρμογή μιας επαναληπτικής διαδικασίας κατά την οποία συρρικνώνεται σε κάθε επανάληψη μια ακμή μέχρι να προκύψει ένα γράφημα με δύο κόμβους και μια ακμή. Το βάρος της ακμής που απομένει μετά τις επαναληπτικές συρρικνώσεις αποτελεί τελικά και το συνολικό βάρος της ελάχιστης τομής. Στο συγκεκριμένο αλγόριθμο όμως υπάρχει ένα ζήτημα. Η πρώτη ακμή που συρρικνώνεται έχει πολύ μικρή πιθανότητα να συμμετέχει στην ελάχιστη τομή ( $\frac{2}{n}$ ), ενώ αντίθετα η τελευταία ακμή που συρρικνώνεται έχει μεγάλη πιθανότητα να συμμετέχει στην ελάχιστη τομή ( $\frac{2}{3}$ ). Συνεπώς ο αλγόριθμος έχει αρχικά καλή απόδοση αλλά στην συνέχεια αποδίδει λιγότερο. Ένας τρόπος να αυξηθεί η πιθανότητα επιτυχίας του αλγορίθμου είναι: έπειτα από μερική συρρίκνωση του γραφήματος, εφαρμόζεται ένας (πιθανόν πιο αργός) αλγόριθμος, ο οποίος έχει μεγαλύτερη πιθανότητα επιτυχίας στο εναπομένον γράφημα. Για παράδειγμα θα μπορούσε να χρησιμοποιηθεί ένας από τους ντεντερμινιστικούς αλγορίθμους υπολογισμού ελάχιστης τομής (NI), έπειτα από συρρικνώσεις που μειώνουν τους κόμβους του αρχικού γραφήματος σε  $m^{\frac{1}{3}}$ .

Ωστόσο, μια καλύτερη παρατήρηση είναι ότι μεγαλύτερη επιτυχία μπορούν να δώσουν δύο δοκιμές του απλού αλγορίθμου συρρίκνωσης που αναφέρθηκε πιο πάνω. Έπειτα σε κάθε δοκιμή, συρρικνώνεται μερικώς το γράφημα, τόσο ώστε η πιθανότητα η ελάχιστη τομή να επιβιώσει να μην είναι τόσο μικρή. Με συρρίκνωση του γραφήματος τόσο ώστε οι κόμβοι του να μειωθούν σε  $\frac{n}{2^2}$ , η πιθανότητα επιβίωσης της ελάχιστης τομής είναι μεγαλύτερη από

ΑΛΓΟΡΙΘΜΟΣ KS(G,N)

**Αν**  $n = 2$

**Τότε** επέστρεψε το βάρος της τομής του  $G$

**Διαφορετικά επανέλαβε** δύο φορές

$G' \leftarrow \text{Contract}(G, \frac{n}{2^{\frac{1}{2}}})$

$KS(G', \frac{n}{2^{\frac{1}{2}}})$

Αλγόριθμος 5: Επαναληπτικός αλγόριθμος συρρίκνωσης

$\frac{1}{2}$ . Επομένως ο αναμενόμενος αριθμός των επιτυχών συρρικνώσεων σε δύο δοκιμές είναι ίσος με ένα. Η παρατήρηση αυτή οδηγεί σε έναν καλύτερο αλγόριθμο ο οποίος περιγράφεται από τον υψηλού επιπέδου ψευδοκώδικα του Αλγορίθμου 5.

Ο επαναληπτικός αλγόριθμος συρρίκνωσης KS εκτελείται σε χρόνο  $O(n^2 \log n)$ . Γίνονται δύο ανεξάρτητες συρρικνώσεις του  $G$  έως ότου οι κόμβοι του  $G$  να μειωθούν σε  $\frac{n}{2^{\frac{1}{2}}}$  και στη συνέχεια σε κάθε μια εφαρμόζεται αναδρομικά η διαδικασία συρρίκνωσης στο γράφημα που προκύπτει. Αποδεικνύεται ότι ο αλγόριθμος KS βρίσκει μια ελάχιστη τομή με πιθανότητα  $\Omega(\frac{1}{\log n})$ .

### 3.3 Αλγόριθμος Padberg-Rinaldi

Οι Padberg-Rinaldi (PR) [23] εισήγαγαν κάποιες ευριστικές βελτιώσεις στον αλγόριθμο των Gomory-Hu.

Υπενθυμίζουμε στο σημείο αυτό ότι, ο αλγόριθμος των Gomory-Hu εκτελεί  $n-1$  υπολογισμούς μέγιστης  $(s, t)$ -ροής για την εύρεση της ελάχιστης τομής του  $G$ , η οποία διαμερίζει το γράφημα σε δύο ανεξάρτητα υπογραφήματα  $H_1, H_2$ . Μια απλή έκδοση του αλγορίθμου των Gomory-Hu βασίζεται στην εξής ιδέα: δοθέντος ενός ζεύγους κόμβων  $s, t$  και της ελάχιστης  $(s, t)$ -τομής τους  $A$ , τότε είτε το  $\lambda_{s,t}(G) = \lambda(G)$ , είτε στην ελάχιστη τομή του  $G$  οι κόμβοι  $s, t$  ανήκουν στο ίδιο υπογράφημα ( $H_1$  ή  $H_2$ ). Αρχικά επιλέγεται ένας κόμβος πηγή  $s$  τυχαία. Έπειτα, μέχρι το σύνολο  $s$  να αποτελείται από όλους τους κόμβους του γραφήματος επαναλαμβάνονται τα ακόλουθα τρία βήματα.

- A) Επιλέγεται ένας κόμβος προορισμός  $t$  ο οποίος δεν ανήκει στο  $s$  και υπολογίζεται η ελάχιστη  $(s, t)$ -τομή. Η ελάχιστη  $(s, t)$ -τομή μπορεί να υπολογιστεί με έναν από τους αλγορίθμους εύρεσης μέγιστης ροής που φαίνονται στον Πίνακα 2.1.

B) Αν η τιμή της ελάχιστης  $(s, t)$ -τομής  $\lambda_{s,t}(G)$  που υπολογίστηκε είναι μικρότερη από την προηγούμενη ελάχιστη τομή, τότε η συγκεκριμένη τομή (και η τιμή της) αποθηκεύεται ως η ελάχιστη τομή του γραφήματος.

Γ) Οι κόμβοι  $s, t$  συρρικνώνονται στον κόμβο  $s$ .

Οι βελτιώσεις των Padberg-Rinaldi βασίζονται σε επιπρόσθετους ελέγχους που επιτρέπουν συγκεκριμένες συρρικνώσεις κόμβων χωρίς την εφαρμογή υπολογισμών μέγιστων ροών. Στην χειρότερη περίπτωση ο αλγόριθμος PR θα χρειαστεί να κάνει  $n - 1$  υπολογισμούς μέγιστων ροών. Σε κάποιες κατηγορίες προβλημάτων, οι ευριστικές επιτρέπουν σε πολλούς κόμβους να συρρικνωθούν, γεγονός που μειώνει σημαντικά το χρόνο εκτέλεσης.

Τα λήμματα που ακολουθούν περιγράφουν τις ευριστικές που χρησιμοποιούν.

**Λήμμα 3.1.** Έστω  $\hat{\lambda}$  ένα άνω φράγμα στην ελάχιστη τομή του  $G$ ,  $\lambda(G)$ . Αν  $u, w \in V$ , ικανοποιεί μια από τις παρακάτω συνθήκες:

$$(PR1) \quad c(u, w) \geq \hat{\lambda},$$

$$(PR2) \quad c(u) \leq 2c(u, w) \quad \text{ή} \quad c(w) \leq 2c(u, w),$$

τότε μια από τις ακόλουθες συνθήκες πρέπει να ισχύει:

1.  $\{u\}$  είναι μια ελάχιστη τομή,
2.  $\{w\}$  είναι μια ελάχιστη τομή,
3. Η ακμή  $\{u, w\}$  είναι η μοναδική ακμή της ελάχιστης τομής,
4.  $\exists$  μια ελάχιστη τομή  $(A, \bar{A})$  με τον  $u \in A$  και τον  $w \in \bar{A}$ .

Ο PR1 έλεγχος λέει ότι αν υπάρχει ένα άνω φράγμα της ελάχιστης τομής  $\hat{\lambda}$  και βρεθεί ακμή  $uw$  με  $c(u, w) \geq \hat{\lambda}$ , τότε η ακμή  $uw$  είτε είναι η μοναδική ακμή που συμμετέχει στην ελάχιστη τομή ( $c(u, w) = \hat{\lambda}$ ), είτε η ακμή αυτή δεν συμμετέχει στην ελάχιστη τομή και μπορεί να συρρικνωθεί.

Για την κατανόηση του PR2 ελέγχου, θεωρούμε μια τομή  $(A, \bar{A})$  με  $u \in A$  και  $w \in \bar{A}$ , και υποθέτουμε ότι ισχύει η σχέση  $c(u) \leq 2c(u, w)$  (η άλλη περίπτωση είναι συμμετρική). Επίσης υποθέτουμε ότι οι κόμβοι  $u$  και  $w$  δεν απομονώνονται με την εφαρμογή της ελάχιστης τομής. Έστω επίσης ότι το σύνολο  $N$  αποτελεί το σύνολο των γειτόνων του κόμβου  $w$ . Τότε ισχύει ότι,

$$\begin{aligned}
c(w) &\leq 2c(u, w) \\
\sum_{x \in N} c(w, x) &\leq 2c(u, w) \\
\sum_{x \in N - \{u\}} c(w, x) + c(u, w) &\leq 2c(u, w) \\
\sum_{x \in N - \{u\}} c(w, x) &\leq c(u, w)
\end{aligned}$$

Αν στην συνέχεια ο κόμβος  $w$  μεταφερθεί από το σύνολο  $\bar{A}$  στο σύνολο  $A$ , τότε η ακμή  $uw$  δεν θα ανήκει πλέον στην τομή και επιπλέον οι κόμβοι που προσπίπτουν στον  $w$  και θα είναι στην τομή είναι το πολύ  $\sum_{x \in N - \{u\}} c(w, x)$ . Το οποίο με βάση την εξίσωση 3.3 είναι το πολύ  $(c(u, w))$ . Συνεπώς η τιμή της τομής που προκύπτει με την μετακίνηση του κόμβου  $w$  είναι μικρότερη της αρχικής,  $c(A \cup w) \leq c(A)$ . Επομένως, στην περίπτωση αυτή το γράφημα πρέπει να έχει ελάχιστη τομή στην οποία οι κόμβοι  $u, w$  θα βρίσκονται στην ίδια πλευρά της τομής, γι'αυτό και είναι ασφαλής η συρρίκνωση της ακμής  $uw$ . Βέβαια όταν γίνεται ο έλεγχος PR2, υπάρχει και το ενδεχόμενο ότι οι κόμβοι  $u$  και  $w$  να αποτελούν μια ελάχιστη τομή. Γι'αυτό και πριν από οποιαδήποτε συρρίκνωση, υπολογίζονται οι ποσότητες  $c(u)$  και  $c(w)$ . Αν κάποια από αυτές έχει τιμή μικρότερη από την τιμή  $\hat{\lambda}$  τότε η τιμή  $\hat{\lambda}$  ανανεώνεται κατάλληλα.

Ένα άλλο σύνολο (πιο ακριβών υπολογιστικά ελέγχων) δίνεται από το ακόλουθο λήμμα.

**Λήμμα 3.2.** Έστω  $\hat{\lambda}$  ένα άνω φράγμα στην ελάχιστη τομή του  $G$ ,  $\lambda(G)$ . Αν  $v, w \in V$ , ικανοποιεί μια από τις παρακάτω συνθήκες:

$$(PR3) \quad \exists u \text{ τέτοιο ώστε } c(v) \leq 2\{c(v, w) + c(v, u)\} \text{ και } c(w) \leq 2\{c(v, w) + c(w, u)\}$$

$$(PR4) \quad c(v, w) + \sum_u \min(c(v, u), c(w, u)) \geq \hat{\lambda}$$

τότε μια από τις ακόλουθες συνθήκες πρέπει να ισχύει:

1.  $\{u\}$  είναι μια ελάχιστη τομή,
2.  $\{w\}$  είναι μια ελάχιστη τομή,
3.  $\exists$  μια ελάχιστη τομή  $(A, \bar{A})$  με τον  $v \in A$  και τον  $w \in A$ .

Για όλα τα ζεύγη κόμβων,  $v, w$  οι έλεγχοι PR3 και PR4 μπορούν να γίνουν σε χρόνο  $O(n)$ . Οι δύο έλεγχοι εφαρμόζονται ταυτόχρονα, και το άθροισμα στον έλεγχο PR4 υπολογίζεται ενόσω γίνεται ο έλεγχος PR3. Στον έλεγχο PR4, για κάθε κοινό γείτονα  $u$  των κόμβων  $v, w$ , επιλέγω την ακμή  $vu, wu$  με το ελάχιστο βάρος για τον υπολογισμό του αθροίσματος.

Ο αλγόριθμος PR εφαρμόζει επαναληπτικά συρρικνώσεις κόμβων είτε κάνοντας υπολογισμούς μέγιστης  $(s, t)$ -ροής (όπου συρρικνώνει τους κόμβους  $s, t$ ), είτε συρρικνώνοντας τους κόμβους που ικανοποιούν τους ελέγχους PR1, PR2, PR3, PR4 που περιγράφονται στα Λήμματα 3.1, 3.2. Η επαναληπτική διαδικασία επαναλαμβάνεται έως ότου οι κόμβοι του γραφήματος μειωθούν σε δύο. Πριν από κάθε συρρίκνωση, υπολογίζεται η τιμή της τομής. Αν η τιμή αυτή είναι μικρότερη από το άνω φράγμα  $\hat{\lambda}$  τότε το άνω φράγμα ανανεώνεται κατάλληλα. Η τιμή  $\hat{\lambda}$ , μπορεί να οριστεί αρχικά με τιμή τον μικρότερο βαθμό του γραφήματος. Τέλος η διαδικασία επιστρέφει την τιμή  $\hat{\lambda}$  που αποτελεί τελικά και τη τιμή της ελάχιστης τομής και το ένα από τα δύο σύνολα που θα προκύψουν με την εφαρμογή της διαμέρισης.

### 3.4 Αλγόριθμος Stoer-Wagner

Η εύρεση της ελάχιστης τομής σε ένα μη κατευθυνόμενο γράφημα με ακμικά βάρη αποτελεί θεμελιώδες αλγοριθμικό πρόβλημα. Ουσιαστικά διαμερίζει ένα γράφημα  $G(V, E)$  σε δύο υπογραφήματα, έτσι ώστε η συνδεσιμότητα μεταξύ των δύο υπογραφημάτων (cut weight) να είναι η ελάχιστη. Κάθε ακμή  $e$  του  $G$  έχει μία πραγματική θετική τιμή βάρους  $w(e)$ .

Το 1997 οι Stoer-Wagner [26] παρουσίασαν ένα ντεντερμινιστικό αλγόριθμο εύρεσης ελάχιστης τομής σε ένα μη κατευθυνόμενο γράφημα με βάρη. Ο αλγόριθμος αυτός είναι πολύ απλός και είναι εύκολος να εφαρμοστεί. Σε αντίθεση με άλλους αλγορίθμους εύρεσης ελάχιστης τομής, που είχαν βρεθεί μέχρι και το 1997, ο αλγόριθμος αυτός δε χρησιμοποιεί τεχνικές ροής. Σε γενικές γραμμές, ο αλγόριθμος αποτελείται από  $|V|$  φάσεις, κάθε μία από τις οποίες είναι μια αναζήτηση της μέγιστης γειτνίασης των κόμβων (maximum adjacency).

Το βασικό κλειδί του αλγορίθμου είναι η παρατήρηση ότι, αν υπάρχει η δυνατότητα να βρεθούν δύο κορυφές  $s$  και  $t$  και το βάρος της ελάχιστης  $(s, t)$ -τομής, τότε το πρόβλημα λύνεται με τη βοήθεια του παρακάτω θεωρήματος.

**Θεώρημα 3.1.** Έστω  $s$  και  $t$  δύο κόμβοι του γραφήματος  $G$  και  $G \setminus \{s, t\}$  το γράφημα που προκύπτει από τη συνένωση των κόμβων  $s$  και  $t$ , τότε η ελάχιστη τομή του  $G$  μπορεί να ληφθεί παίρνοντας το μικρότερο από τις ποσότητες ελάχιστης  $(s, t)$ -τομής του  $G$  και της ελάχιστης τομής του  $G \setminus \{s, t\}$ .

Είναι προφανής η απόδειξη του πιο πάνω θεωρήματος. Αν η ελάχιστη τομή του  $G$  χωρίζει τους κόμβους  $s, t$  τότε η ελάχιστη  $(s, t)$ -τομή του  $G$  είναι και η ελάχιστη τομή του

**MinimumCutPhase( $G, w, \alpha$ )** $A \leftarrow \{ \alpha \}$ **Ενόσω**  $A \neq V$ 

πρόσθεσε στο  $A$  την κορυφή που έχει μεγαλύτερη συνδεσιμότητα με τη λίστα  $A$   
 αποθήκευσε το cut της φάσης αυτής, συρρίκνωσε το γράφημα  $G$  (συνένωση των δύο  
 τελευταίων κορυφών που προστέθηκαν στο  $A$ )

## Αλγόριθμος 6: MinimumCutPhase

$G$ . Διαφορετικά, η ελάχιστη τομή του  $G$  είναι η ελάχιστη τομή του  $G \setminus \{s, t\}$ .

Επομένως, δημιουργείται μια διαδικασία η οποία υπολογίζει αυθαίρετα μια ελάχιστη  $(s, t)$ -τομή του  $G$ . Αυτή χρησιμοποιείται επαναληπτικά έως ότου υπολογισθεί η ελάχιστη τομή του  $G$  και είναι γνωστή στη βιβλιογραφία ως αναζήτηση μέγιστης γειτνίασης (maximum adjacency search ή maximum cardinality search). Η διαδικασία παρουσιάζεται παρακάτω.

Σε κάθε φάση του αλγορίθμου δημιουργείται μια διάταξη γειτνίασης κόμβων  $A$ . Αρχικά το σύνολο  $A$  περιέχει ένα τυχαίο κόμβο και επαναληπτικά προστίθεται σε αυτό ένας καινούργιος κόμβος έως ότου όλοι οι κόμβοι του  $G$  εισαχθούν σε αυτό. Σε κάθε επανάληψη, προστίθεται ένας κόμβος εκτός του συνόλου  $A$  που είναι περισσότερο συνδεδεμένος με το  $A$ . Δηλαδή, προστίθεται ο κόμβος  $z$  για τον οποίο ισχύει

$$z \notin A \text{ και } w(A, z) = \max\{w(A, y) \mid y \notin A\},$$

όπου  $w(A, z)$  είναι το άθροισμα των βαρών όλων των ακμών ανάμεσα στο σύνολο  $A$  και του κόμβου  $z$ . Στο τέλος κάθε φάσης οι δύο κόμβοι που προστέθηκαν τελευταίοι στο σύνολο  $A$  συνενώνονται, δηλαδή, οι δύο κόμβοι αντικαθίστανται με ένα νέο κόμβο, και οι ακμές από τους δύο αυτούς κόμβους προς ένα οποιοδήποτε άλλο κόμβο αντικαθίστανται με μια ακμή το βάρος της οποίας είναι το άθροισμα των δύο επί μέρους ακμών. Πιθανόν βρόγχοι στο γράφημα που προκύπτουν απομακρύνονται. Η τομή του συνόλου του  $V$  που χωρίζει τον κόμβο που προστέθηκε τελευταίος στο σύνολο  $A$  από το υπόλοιπο γράφημα αποτελεί το *cut-of-the-face*. Το μικρότερο *cut-of-the-face* αποτελεί την ελάχιστη τομή του  $G$ , και αποτελεί το αποτέλεσμα του αλγορίθμου.

Όπως παρατηρούμε στον Αλγόριθμο 7, ο κόμβος εκκίνησης παραμένει ο ίδιος καθ' όλη τη διάρκεια του αλγορίθμου.

**MinimumCut(G, w, α)**

Ενόσω  $|V| > 1$

**MinimumCutPhase(G, w, α)**

Αν το cut-of-the-phase είναι μικρότερο από το τρέχον

Τότε θέσε το cut-of-the-phase ως την ελάχιστη τομή του γραφήματος

### Αλγόριθμος 7: Minimum Cut

Για την απόδειξη της ορθότητας του αλγόριθμου, παρατίθεται το ακόλουθο λήμμα.

**Λήμμα 3.3.** Κάθε cut-of-the-phase είναι η ελάχιστη  $(s, t)$ -τομή στο τρέχον γράφημα, όπου  $s$  και  $t$  οι κόμβοι που προστέθηκαν τελευταίοι στη διάταξη γειτνίασης.

Σε κάθε φάση το *cut-of-the-phase* αποτελεί την ελάχιστη  $(s, t)$ -τομή στο τρέχον γράφημα, όπου  $s$  και  $t$  οι δυο κόμβοι που προστίθενται τελευταίοι στη διάταξη γειτνίασης τη συγκεκριμένη φάση. Αυτό αποδεικνύεται εύκολα από το γεγονός ότι σε κάθε φάση οι κόμβοι του γραφήματος ταξινομούνται γραμμικά, ξεκινώντας από τον κόμβο  $\alpha$  και καταλήγοντας στους κόμβους  $s$  και  $t$ . Έστω μια τυχαία  $(s, t)$ -τομή  $C$  στο τρέχον γράφημα, αρκεί να αποδειχθεί ότι είναι τουλάχιστον τόσο μεγάλη όσο και το *cut-of-the-phase*.

Έστω ένας ενεργός (active) κόμβος  $v! = \alpha$  (ενεργός είναι ένας κόμβος όταν αυτός και ο κόμβος που προστέθηκε πριν από αυτόν ανήκουν σε διαφορετικά μέρη στη τομή  $C$ ). Το συνολικό βάρος της τομής  $C$  συμβολίζεται με  $w(C)$ , το σύνολο των κόμβων που προστέθηκαν πριν από τον  $v$  συμβολίζεται με  $A_v$ , η τομή του συνόλου  $A_v \cup \{v\}$  συμβολίζεται με  $C_v$  και το βάρος της με  $w(C_v)$ . Αποδεικνύεται με επαγωγή στο σύνολο των ενεργών κόμβων ότι για κάθε ενεργό κόμβο  $v$  ισχύει:

$$w(A_v, v) \leq w(C_v)$$

Για τον πρώτο ενεργό κόμβο, η ανισότητα ισχύει με ισότητα. Έστω ότι η ανισότητα ισχύει για κάθε ενεργό κόμβο που προστίθεται μέχρι και τον ενεργό κόμβο  $v$ , και έστω ότι ο  $u$  είναι ο επόμενος ενεργός κόμβος που προστίθεται. Τότε ισχύει το εξής:

$$w(A_u, u) = w(A_v, u) + w(A_u \setminus A_v, u)$$

Επειδή ο κόμβος  $v$  επιλέγηκε πριν από τον κόμβο  $u$  είναι προφανές ότι ισχύει  $w(A_v, u) \leq w(A_v, v)$ , γιατί σύμφωνα με τον αλγόριθμο σε κάθε επανάληψη επιλέγεται ο κόμβος που έχει μεγαλύτερη συνδεσιμότητα με το σύνολο  $A_v$ . Με βάση την επαγωγική υπόθεση ισχύει  $w(A_v, v) \leq w(C_v)$ . Επιπλέον, επειδή ο  $u$  είναι ένας ενεργός κόμβος, οι ακμές ανάμεσα



στο σύνολο  $A_u \setminus A_v$  και του κόμβου  $u$  συνδέουν διαφορετικά μέρη της τομής  $C$  και συνεπώς επηρεάζουν μόνο την ποσότητα  $w(C_u)$  και όχι την ποσότητα  $w(C_v)$ . Επομένως ισχύει ότι:

$$w(A_u, u) = w(C_v) + w(A_u \setminus A_v, u) \leq w(C_u)$$

Όπου η ισότητα ισχύει όταν ο κόμβος  $u$  δε συνδέεται με κανένα από τους κόμβους στο σύνολο το οποίο ο κόμβος  $v$  ανήκει ή συνδέεται μόνο με τον κόμβο  $v$ . Συμπερασματικά, αφού ο κόμβος  $t$  είναι πάντα ένας ενεργός κόμβος ως αναφορά τη τομή  $C$ , ισχύει ότι  $w(A_t, t) \leq w(C_t)$ . Άρα το *cut-of-the-face* αποτελεί και την ελάχιστη  $(s, t)$ -τομή. Η πολυπλοκότητα χρόνου του αλγόριθμου *MinimumCut* (Αλγόριθμος 7) είναι ουσιαστικά ίση με  $|V|$  φορές την πολυπλοκότητα χρόνου της διαδικασίας *MinimumCutPhase* (Αλγόριθμος 6), η οποία καλείται σε κάθε φάση σε γράφημα μειωμένου μεγέθους. Αρκεί να δειχθεί ότι μια εκτέλεση της *MinimumCutPhase* χρειάζεται το πολύ  $O(|E| + |V| \log |V|)$  οδηγώντας σε συνολική πολυπλοκότητα  $O(|V||E| + |V|^2 \log |V|)$

Το κλειδί ώστε κάθε φάση να υλοποιηθεί αποτελεσματικά είναι το κατά πόσο εύκολα μπορεί να επιλεγεί ο επόμενος κόμβος (ο πιο ισχυρά συνδεδεμένος) που πρόκειται να εισαχθεί στο σύνολο  $A$ . Δηλαδή, ο επόμενος κόμβος που εισάγεται είναι ο κόμβος με τη μεγαλύτερη γειτνίαση ως προς το σύνολο  $A$ . Κατά τη διάρκεια εκτέλεσης της φάσης, όλοι οι κόμβοι που δεν έχουν εισαχθεί στο σύνολο  $A$ , τοποθετούνται σε μία ουρά προτεραιότητας (priority queue) με βάση ένα πεδίο κλειδί. Το κλειδί ενός κόμβου  $u$  είναι το άθροισμα των βαρών των ακμών που συνδέουν τον κόμβο αυτό με το σύνολο  $A$  και συμβολίζεται ως  $w(A, u)$ . Κάθε φορά που ένας κόμβος  $u$  προστίθεται στο σύνολο  $A$ , η ουρά ανανεώνεται:

- Ο κόμβος  $u$ , διαγράφεται από την ουρά.
- Το κλειδί κάθε κόμβου  $w$  που δεν ανήκει στο  $A$  και συνδέεται με τον κόμβο  $u$  πρέπει να μειωθεί κατά το βάρος της ακμής  $uw$ .

### 3.5 Αλγόριθμος Brinkmeier

Όπως και οι αλγόριθμοι *NI* και *SW*, έτσι και ο αλγόριθμος του Brinkmeier [31] χρησιμοποιεί συρρικνώσεις ακμών για την εύρεση της ελάχιστης τομής σε ένα μη κατευθυνόμενο γράφημα  $G(V, E)$ , όπου  $n = |V|$  και  $|E| = m$ . Κάθε ακμή του γραφήματος, χαρακτηρίζεται από μη αρνητικές πραγματικές τιμές βαρών. Στον αλγόριθμο αυτόν προτείνονται δύο αλλαγές στον αλγόριθμο *SW*, ο οποίος υπολογίζει μια διάταξη των κόμβων

(maximum adjacency order (MA\_order)) σε μη κατευθυνόμενα γραφήματα και έπειτα συρρικνώνει τους τελευταίους δύο κόμβους στη διάταξη αυτή. Επαναλαμβάνοντας το αυτό  $n-1$  φορές υπολογίζεται η ελάχιστη τομή του  $G$ . Οι αλλαγές που προτείνονται, βελτιώνουν τον προηγούμενο αλγόριθμο σε πολυπλοκότητα χρόνου.

Οι αλλαγές που προτείνονται είναι:

1. Συρρικνώνει περισσότερα από ένα ζεύγη κόμβων, αν αυτό είναι δυνατόν, σε μια επανάληψη. Η ίδια τεχνική χρησιμοποιήθηκε και από τους *NI* (1994) αλλά ο αλγόριθμος αυτός δεν βασίζεται στο MA\_order. Το επιτυγχάνει ουσιαστικά αυτό χαλαρώνοντας τους περιορισμούς της διάταξης MA\_order προκύπτοντας έτσι μια LA\_order (lax adjacency order). Στην διάταξη δηλαδή, εισάγει όπως και πριν τον κόμβο  $z$  για τον οποίο ισχύει  $w(A, z) = \max\{w(A, y) | y \notin A\}$ , όπου  $w(A, z)$  είναι το άθροισμα των βαρών όλων των ακμών ανάμεσα στο σύνολο  $A$  (lax adjacency order) και του  $z$ , με τη διαφορά ότι αν  $w(A, z) \geq \delta_G$  τότε ο κόμβος  $z$  συρρικνώνεται με τον τελευταίο κόμβο που εισήχθηκε στη διάταξη. Οι συρρικνώσεις των κόμβων μπορούν να γίνονται ταυτόχρονα με την εισαγωγή τους στη διάταξη, γιατί οι συρρικνώσεις των κόμβων δεν επηρεάζουν την συνδεσιμότητα (adjacency) των κόμβων.
2. Αν τα βάρη των άκμων του γραφήματος  $G$ , είναι ακέραιοι αριθμοί, τότε μπορεί να χρησιμοποιηθεί μια εναλλακτική δομή δεδομένων (Priority Queues with Threshold), για την κατασκευή της LA\_order διάταξης, η οποία βελτιώνει τον χρόνο εκτέλεσης.

Αν εφαρμοστεί στον αλγόριθμο *SW* η πρώτη αλλαγή που προτείνεται, τότε τα ακόλουθα απλά παραδείγματα περιγράφουν την συμπεριφορά του αλγόριθμου σε δύο απλές περιπτώσεις.

**Παράδειγμα 1** Έστω ένα γράφημα  $G(V, E)$  με κόμβους  $u_1, \dots, u_n$  και ακμές  $(u_i, u_{i+1})$  για  $\leq i < n$ , δηλαδή, ένα μονοπάτι. Τότε η διάταξη  $u_1, \dots, u_n$  είναι μια LA\_order και ο κόμβος  $u_1$  είναι ο κόμβος με τον ελάχιστο βαθμό 1. Συνεπώς, ο αλγόριθμος συρρικνώνει όλους τους κόμβους του  $G$  στην πρώτη και μοναδική επανάληψη. Άρα υπολογισμός της ελάχιστης τομής του συγκεκριμένου γραφήματος, απαιτεί χρόνο  $O(n + m + \log n)$ .

**Παράδειγμα 2** Έστω ένα γράφημα  $G(V, E)$  με κόμβους  $u_1, \dots, u_n$  και ακμές  $(u_i, u_{i+1})$  για  $\leq i < n$  και  $(u_n, u_1)$ , δηλαδή, έναν κύκλο. Τότε η διάταξη  $u_1, \dots, u_n$  είναι μια LA\_order και όλοι οι κόμβοι του γραφήματος έχουν ίδιο ελάχιστο βαθμό 2. Εφόσον, οι συνδεσιμότητα των κόμβων  $u_i$ , εκτός από τον κόμβο  $u_n$  είναι 1, τότε μόνο οι κόμβοι  $u_{n-1}$  και  $u_n$  θα

συρρικνωθούν. Το γράφημα  $G \setminus \{n - 1, n\}$ , αποτελεί πάλι κύκλο με ένα λιγότερο κόμβο σε σύγκριση με το αρχικό γράφημα. Επομένως, ο αλγόριθμος απαιτεί  $n - 1$  επαναλήψεις για τον υπολογισμό της ελάχιστης ροής. Όπως ακριβώς και ο αλγόριθμος *SW*.

Η δεύτερη αλλαγή που προτείνουν αφορά την δομή στην οποία αποθηκεύουν, σε κάθε μια από τις  $n - 1$  επαναλήψεις, τις γειτνιάσεις των κόμβων (όπου επιλέγουν σε κάθε βήμα τον κόμβο που με την μεγαλύτερη συνδεσιμότητα με την τρέχουσα διάταξη *LAorder*). Ορίζουν την νέα δομή, με την παρατήρηση ότι δεν υπάρχει καμία διάκριση ανάμεσα στους κόμβους με συνδεσιμότητα μεγαλύτερη από ένα κατώφλι  $\tau$  (το κατώφλι παίρνει τιμή ίση με  $\delta_G$ , δηλαδή τον ελάχιστο βαθμό του γραφήματος). Η δομή αυτή επιτρέπει τις ίδιες λειτουργίες όπως και η ουρά προτεραιότητας, δηλαδή εισαγωγή, διαγραφή του στοιχείου με την μεγαλύτερη συνδεσιμότητα και ανανέωση της προτεραιότητας ενός κόμβου, μόνο που η διαγραφή του στοιχείου με τη μεγαλύτερη συνδεσιμότητα διαφέρει ελαφρώς. Στην προκειμένη περίπτωση η ενέργεια αυτή επιστρέφει το στοιχείο με τη μέγιστη προτεραιότητα μόνο αν η μέγιστη συνδεσιμότητα είναι μικρότερη από το ορισμένο κατώφλι. Η ουρά προτεραιότητας με κατώφλι  $\tau$ , αποτελείται από  $\tau$  κάδους,  $B_j$ , όπου  $0 \leq j \leq \tau - 1$ . Ο κάδος  $B_j$  περιέχει όλες τους κόμβους με προτεραιότητα  $j$ . Επομένως διατηρώντας κάθε φορά ένα δείκτη προς τον μη κενό κάδο τιμής  $j_{max}$ , η διαγραφή του κόμβου με τη μεγαλύτερη συνδεσιμότητα, επιτυγχάνεται αφαιρώντας την πρώτη τιμή του κάδου με  $B_{j_{max}}$ . Αν το στοιχείο που αφαιρείται αποτελεί και το μοναδικό στοιχείο του κάδου, τότε αναζητήσε τον επόμενο μη κενό κάδο (ξεκινώντας από τον κάδο  $j_{max} - 1$ ). Στην χειρότερη περίπτωση αυτή η αναζήτηση απαιτεί χρόνο  $O(\tau)$ . Επομένως σε ένα γράφημα  $G(V, E)$  με ακέραιες τιμές βαρών, η δεύτερη αλλαγή μπορεί να μειώσει τον χρόνο εκτέλεσης του αλγορίθμου σε  $O(\delta_G n^2)$ .

Πίνακας 3.1: Αλγόριθμοι Εύρεσης Ελάχιστης Τομής

| Αναφορά                      | Πολυπλοκότητα                                      |
|------------------------------|--|
| Gomory and Hu (1961)         | $O(n^2 m \log(n^2/m))$                             |
| Padberg and Rinaldi (1988)   | $O(n^2 m \log(n^2/m))$                             |
| Hao and Orlin (1992)         | $O(nm \log(n^2/m))$                                |
| Nagamochi and Ibaraki (1992) | $O(nm + n^2 \log n)$                               |
| Nagamochi and Ibaraki (1994) | $O(nm + n^2 \log n)$                               |
| Karger and Stein (1996)      | $O(n^2 \log^3 n)$                                  |
| Stoer and Wagner (1997)      | $O(nm + n^2 \log n)$                               |
| Brinkmeier (2000)            | $O(\max(\log n, \min(m/n, \delta_G/\epsilon))n^2)$ |

## ΚΕΦΑΛΑΙΟ 4

# ΟΜΑΔΟΠΟΙΗΣΗ-ΚΑΤΑΤΜΗΣΗ ΕΙΚΟΝΑΣ

---

### 4.1 Μέθοδοι Ομαδοποίησης

### 4.2 Μέθοδοι Κατάτμησης Εικόνας

---

## 4.1 Μέθοδοι ομαδοποίησης

### 4.1.1 Η διαδικασία ομαδοποίησης

Η ομαδοποίηση είναι η οργάνωση μιας συλλογής από δείγματα-στοιχεία σε συστάδες (clusters) με βάση κάποιο μέτρο ομοιότητας. Τα στοιχεία συνήθως περιγράφονται σαν διανύσματα τιμών και αναπαρίστανται ως σημεία σε έναν πολυδιάστατο χώρο. Στοιχεία που ανήκουν στην ίδια ομάδα παρουσιάζουν μεγαλύτερη ομοιότητα από ότι στοιχεία που ανήκουν σε διαφορετικές ομάδες. Οι τεχνικές ομαδοποίησης κατατάσσονται στις τεχνικές μάθησης χωρίς επίβλεψη. Στις αυτές τα στοιχεία (δεδομένα του προβλήματος) ομαδοποιούνται χωρίς καμία γνώση για τις προϋπάρχουσες ομάδες. Συνεπώς, η ομαδοποίηση είναι απόλυτα οδηγούμενη από τα δεδομένα και παράγεται μόνο από αυτά. Τα βήματα που ακολουθούνται κατά την διαδικασία της ομαδοποίησης είναι:

1. Αναπαράσταση των στοιχείων (Pattern representation) η οποία μπορεί να περιλαμβάνει παραγωγή νέων χαρακτηριστικών ή επιλογή μέρους των χαρακτηριστικών των στοιχείων
2. Ορισμός του μέτρου ομοιότητας μεταξύ των δεδομένων (Similarity measure definition)

3. Ομαδοποίηση των δεδομένων (Clustering)
4. Αφαίρεση δεδομένων όταν χρειάζεται (Data abstraction)
5. Προσδιορισμός και εκτίμηση του αποτελέσματος (Assessment of output)

Το αποτέλεσμα της ομαδοποίησης, μπορεί να είναι απόλυτα καθορισμένο (ομαδοποίηση των δεδομένων σε ξένες μεταξύ τους κλάσεις), ή χαλαρό (όπου κάθε στοιχείο δεν ανήκει μόνο σε μία κλάση αλλά είναι μέλος πολλών κλάσεων με κάποιο βαθμό σε κάθε μια).

#### 4.1.2 Τεχνικές ομαδοποίησης

Οι αλγόριθμοι ομαδοποίησης μπορεί να είναι:

1. Ιεραρχικοί (hierarchical), όπου παράγονται εμφωλευμένες ομάδες μετά από διαδικασίες διαχωρισμού (top-down) ή συγχώνευσης (bottom-up) με βάση το μέτρο ομοιότητας μεταξύ των στοιχείων διαφορετικών ομάδων.
2. Διαμεριστικοί (partitional ή non-hierarchical) οι οποίοι προσπαθούν να βελτιώσουν επαναληπτικά την ποιότητα της ομαδοποίησης, βάσει κάποιων κριτηρίων ελέγχου. Στοχεύουν στο να διαχωρίσουν τα δεδομένα με τέτοιο τρόπο ώστε να βελτιστοποιείται το κριτήριο με το οποίο γίνεται η ομαδοποίηση.
3. “Σκληροί” και “χαλαροί”. Ένας “σκληρός” αλγόριθμος τοποθετεί κάθε στοιχείο σε μία και μόνο ομάδα, σε αντίθεση με τους “χαλαρούς” αλγόριθμους οι οποίοι δίνουν σε κάθε στοιχείο για κάθε ομάδα έναν βαθμό που εκφράζει κατά πόσο το στοιχείο αυτό ανήκει στην ομάδα αυτή.
4. Αυξητικοί, όπου το σύνολο των δεδομένων προς ομαδοποίηση είναι πολύ μεγάλο και οι περιορισμοί που υπάρχουν στον χρόνο εκτέλεσης και τον διαθέσιμο χώρο μνήμης επηρεάζουν την αρχιτεκτονική του αλγορίθμου. Με την αύξηση όμως της πληροφορίας, υπήρξε η ανάγκη για εύρεση αλγορίθμων οι οποίοι ελαχιστοποιούν τον αριθμό σαρώσεων των δεδομένων, μειώνουν τον αριθμό των στοιχείων που εξετάζονται ή μειώνουν το μέγεθος των δομών που χρησιμοποιούνται κατά την εκτέλεση του αλγορίθμου.

#### 4.1.3 Επιλογή κατάλληλης μεθόδου ομαδοποίησης

Υπάρχει πληθώρα αλγορίθμων ομαδοποίησης στην βιβλιογραφία και ένα σημαντικό ερώτημα είναι πως αποφασίζουμε ποιος είναι ο κατάλληλος αλγόριθμος που πρέπει να χρησιμο-

ποιήσουμε για την επίλυση κάποιου προβλήματος. Κάποια κριτήρια επιλογής κατάλληλου αλγόριθμου ομαδοποίησης βασίζονται στις εξείς ιδιότητες:

1. στον τρόπο με τον οποίο σχηματίζονται οι ομάδες,
2. στη δομή που έχουν τα δεδομένα προς επεξεργασία,
3. στην ευαισθησία που έχει ο αλγόριθμος σε αλλαγές που δεν επηρεάζουν τα δεδομένα.

Το κυρίως πρόβλημα ενός αλγόριθμου ομαδοποίησης είναι ότι δεν μπορεί για όλες τις περιπτώσεις δεδομένων να εφαρμοστεί και να αναδείξει επιτυχώς την ποικιλία δομών που εμφανίζονται ειδικά σε πολυδιάστατα σύνολα δεδομένων. Για παράδειγμα ο αλγόριθμος ομαδοποίησης K-Μέσων βρίσκει αποκλειστικά σφαιρικές ομάδες. Για το λόγο αυτό είναι χρήσιμη όποια πληροφορία μπορούμε να έχουμε σχετικά με το τι είναι πιθανό να υπάρχει στα δεδομένα, είτε για να δώσουμε παραπάνω πληροφορία στον αλγόριθμο που χρησιμοποιούμε ώστε να μπορέσει να την εκμεταλλευτεί, είτε να μας βοηθήσει να επιλέξουμε έναν καταλληλότερο αλγόριθμο.

#### 4.1.4 Μέτρα ομοιότητας

Για όλες τις τεχνικές ομαδοποίησης, είναι σημαντικό να ορίζεται ένα μέτρο ομοιότητας (ή συνάρτηση ομοιότητας  $sim(x, y)$ ) το οποίο αποτιμά την ομοιότητα μεταξύ δύο δεδομένων  $x$  και  $y$ , με μια αριθμητική ποσότητα. Η έννοια της ομοιότητας είναι συμπληρωματική της απόστασης  $dist(x, y)$ , που εκφράζει την ανομοιότητα των αντικειμένων. Όταν ισχύει  $sim(x, y) \in [0, 1]$  εύκολα μετατρέπουμε τη συνάρτηση ομοιότητας σε συνάρτηση απόστασης στο ίδιο διάστημα (και αντίστροφα):  $dist(x, y) = 1 - sim(x, y)$ . Το πιο γνωστό μέτρο ανομοιότητας που χρησιμοποιείται είναι η Ευκλείδεια απόσταση η οποία ορίζεται ως εξής:

$$D(x, y) = \sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

Άλλοι τύποι που δίνουν την απόσταση μεταξύ δύο δεδομένων μπορεί να είναι και η απόσταση Manhattan:

$$D(x, y) = \sum_{i=1}^k |x_i - y_i|$$

ή το μέγιστο της διαφοράς σε κάθε διάσταση:

$$D(x, y) = \max_{i=1}^k |x_i - y_i|$$

Η ευκλείδεια απόσταση χρησιμοποιείται ευρέως σε περιπτώσεις λίγων διαστάσεων και έχει καλά αποτελέσματα όταν τα δεδομένα κατηγοριοποιούνται σε συμπαγείς και αρκετά απομονωμένες ομάδες. Ένα πρόβλημα που παρουσιάζει είναι ότι στις πολλές διαστάσεις το χαρακτηριστικό το οποίο παρουσιάζει την μεγαλύτερη διαφοροποίηση από τα άλλα κυριαρχεί και αποπροσανατολίζει το τελικό αποτέλεσμα. Εδώ πρόκειται για αυτό που συνήθως αναφέρεται ως “κατάρρα” των πολλών διαστάσεων. Στην συνέχεια περιγράφονται κάποιοι βασικοί αλγόριθμοι ομαδοποίησης.

#### 4.1.5 Αλγόριθμος K-μέσων

Ο αλγόριθμος k-μέσων είναι ένας πολύ απλός και πολύ διαδομένος διαμεριστικός αλγόριθμος ο οποίος ελαχιστοποιεί το τετραγωνικό σφάλμα. Ο αλγόριθμος ξεκινά με μια τυχαία διαμέριση σε ομάδες και συνεχώς τοποθετεί στοιχεία στις ομάδες με βάση την απόσταση των στοιχείων από το κεντροειδές της ομάδας. Αυτό σταματάει μέχρι να ικανοποιηθεί κάποιο κριτήριο το οποίο μπορεί να είναι η ελαχιστοποίηση της συνάρτησης τετραγωνικού λάθους ή η μη διαφοροποίηση των ομάδων από κάποια επανάληψη και μετά. Ο αλγόριθμος αυτός είναι δημοφιλής εξαιτίας της απλότητας υλοποίησής του.

Μια δυσκολία του αλγορίθμου είναι στο να εντοπίσει μη σφαιρικές ομάδες, ομάδες με διαφορετικές πυκνότητες και ομάδες διαφορετικών μεγεθών. Ένα άλλο μειονέκτημα του αλγορίθμου είναι ότι εξαρτάται από την αρχικοποίηση. Αν η επιλογή αυτή δεν είναι αρκετά προσεκτική τότε το κριτήριο τετραγωνικού σφάλματος εγκλωβίζεται σε τοπικά ελάχιστα κάνοντας την τελική επιλογή ομάδας ανεπιτυχή. Για παράδειγμα, αν κατά την αρχικοποίηση αναθέσει δύο αντικείμενα μίας κατηγορίας σε περισσότερες από μία ομάδες ως αρχικά κέντρα, τότε είναι πολύ πιθανό η κατηγορία αυτή να περιγράφεται επίσης από δύο ομάδες στην τελική λύση, ενώ αντίστοιχα τουλάχιστον μία άλλη ομάδα θα περιέχει στοιχεία από δύο κατηγορίες. Όσο αυξάνονται οι ομάδες του προβλήματος τόσο πιθανότερο είναι το φαινόμενο αυτό.

#### 4.1.6 Συσσωρευτικός Αλγόριθμος Ομαδοποίησης

Ο συσσωρευτικός αλγόριθμος είναι ένας επαναληπτικός αλγόριθμος, ο οποίος σε κάθε επανάληψη του αλγορίθμου οι δυο ομάδες που είναι “περισσότερο όμοιες” σχηματίζουν μία ενιαία ομάδα. Μειώνεται έτσι ο τρέχον αριθμός ομάδων του αλγορίθμου, ο οποίος τερματίζει όταν σχηματιστούν  $M$  ομάδες κειμένων.

Υπάρχουν τρεις βασικοί τρόποι υπολογισμού της ομοιότητας μεταξύ δύο ομάδων (inter-

ΑΛΓΟΡΙΘΜΟΣ K-ΜΕΣΩΝ

1. Επιλογή  $k$  κεντροειδών ομάδων τα οποία αποτελούν και τα μόνα στοιχεία των  $k$  των επιλεγμένων ομάδων
2. Τοποθέτησε κάθε στοιχείο στην πιο κοντινή ομάδα μετά από υπολογισμό της απόστασης του σημείου από το κεντροειδές της ομάδας
3. Υπολόγισε το νέο κεντροειδές
4. Αν το κριτήριο τερματισμού δεν ικανοποιείται, πήγαινε στο Βήμα 2.

Αλγόριθμος 8: Αλγόριθμος K-μέσων

cluster similarity) οι οποίοι και διαχωρίζουν τις μεθόδους:

1. Απλού Συνδέσμου (Single-Link). Η ομοιότητα υπολογίζεται ως η μέγιστη ομοιότητα μεταξύ ενός ζεύγους αντικειμένων των ομάδων.
2. Πλήρους Συνδέσμου (Complete-Link). Η ομοιότητα υπολογίζεται ως η ελάχιστη ομοιότητα μεταξύ ενός ζεύγους κειμένων των ομάδων.
2. Μέσου Όρου (Average-Link, Group Average). Η ομοιότητα υπολογίζεται ως η μέση ομοιότητα μεταξύ όλων των ζευγών κειμένων από τις δύο ομάδες.

Η τρίτη προσέγγιση λαμβάνει υπόψη όλη την ομάδα και δίνει καλύτερα αποτελέσματα.

ΣΥΣΣΩΡΕΥΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΟΜΑΔΟΠΟΙΗΣΗΣ

1. Θεώρησε  $N$  ομάδες οι οποίες περιέχουν ένα αντικείμενο η κάθε μία
2. Υπολόγισε τον πίνακα ομοιότητας μεταξύ ομάδων
3. Θέσε  $K = N$
4. Όσο  $K > M$ ,
5. Βρες τις δύο κοντινότερες ομάδες  $C_i, C_j$
6. Αν  $i \leq j$  Συνένωσε τις  $C_i, C_j$  στην  $C_i$ ,
7. Αλλιώς Συνένωσε τις  $C_i, C_j$  στην  $C_j$
8. Ενημέρωσε τον πίνακα ομοιότητας για την συγχωνευμένη ομάδα
8. Θέσε  $K = K - 1$

Αλγόριθμος 9: Συσσωρευτικός Αλγόριθμος Ομαδοποίησης



## 4.2 Κατάτμηση Εικόνας

Η κατάτμηση εικόνας αποτελεί βασική λειτουργία στο τομέα της ανάλυσης εικόνας. Η βασική αυτή λειτουργία διαμερίζει μια διακριτή εικόνα σε ανεξάρτητες περιοχές, έτσι ώστε κάθε περιοχή που προκύπτει από την κατάτμηση να είναι ομοιογενής και οι ανεξάρτητες περιοχές να είναι μεταξύ τους ανομοιογενείς.

Πολλές τεχνικές κατάτμησης εικόνας είναι διαθέσιμες στη βιβλιογραφία. Μερικές από αυτές τις τεχνικές χρησιμοποιούν μόνο το γκριζο ιστόγραμμα επιπέδων και κάποιες άλλες χρησιμοποιούν τις χωρικές λεπτομέρειες της εικόνας. Οι περισσότερες από αυτές τις τεχνικές δεν είναι κατάλληλες για θορυβώδεις εικόνες. Μερικές εργασίες έχουν γίνει χρησιμοποιώντας το τυχαίο μοντέλο Markov (MRF) το οποίο είναι ανθεκτικό στο θόρυβο, αλλά είναι υπολογιστικά περίπλοκο. Η βιβλιογραφία που αφορά τεχνικές κατάτμησης που εφαρμόζονται σε έγχρωμες εικόνες δεν είναι τόσο μεγάλη όσο αυτή που αφορά τεχνικές που εφαρμόζονται σε εικόνες που περιέχουν μόνο αποχρώσεις του γκρι.

Από τις διάφορες μεθόδους που χρησιμοποιούνται για κατάτμηση εικόνας, μόνο λίγες από αυτές χρησιμοποιούν θεωρία γραφημάτων. Όταν χρησιμοποιείται γραφοθεωρητική τεχνική για την επίλυση του προβλήματος, τότε απαιτείται κάποιου είδους προεπεξεργασία κατά την οποία η διακριτή εικόνα μοντελοποιείται με ένα γράφημα. Οι κόμβοι του γραφήματος αναπαριστούν τα εικονοστοιχεία (pixel) της εικόνας (ή ομάδες εικονοστοιχείων της εικόνας). Το βάρος ακμής ορίζεται ως μια ποσότητα συσχέτισης μεταξύ γειτονικών ζευγών εικονοστοιχείων. Για παράδειγμα, αν η συνάρτηση βάρους ορίζεται για κάθε ζεύγος εικονοστοιχείων της εικόνας τότε το γράφημα που προκύπτει είναι ένα πλήρες γράφημα με βάρη,  $K_n$ , ενώ αν ορίζεται μόνο σε οριζόντια ή κατακόρυφα γειτονικά εικονοστοιχεία της εικόνας, τότε το γράφημα που προκύπτει είναι ένα πλέγμα (*grid* γράφημα). Έπειτα, αφού η εικόνα του προβλήματος μοντελοποιηθεί με ένα γράφημα, τότε το πρόβλημα κατάτμησης εικόνας ανάγεται σε πρόβλημα διαμέρισης (partitioning) του γραφήματος  $G$ . Μια διαμέριση του γραφήματος  $G$ , είναι μια συλλογή από ανεξάρτητα υποσύνολα,  $A_1, A_2, \dots, A_k$ , τέτοια ώστε  $A_i \subset V, i = 1, \dots, k$ , και  $\bigcup_{i=1}^k A_i = V$ .

Η κατάτμηση μιας εικόνας μπορεί να γίνει με βάση διάφορα κριτήρια όπως το χρώμα, την υφή, την φωτεινότητα, την κίνηση αν πρόκειται για video, ή κάποιο συνδυασμό των παραπάνω χαρακτηριστικών.

Στην συνέχεια, παρουσιάζονται κάποιες μέθοδοι κατάτμησης εικόνας, που χρησιμοποιούν θεωρία γραφών και επομένως το πρόβλημα κατάτμησης ανάγεται σε διαμέριση γραφήματος.

### 4.2.1 Μέθοδος Wu-Leahy

Ο αλγόριθμος των Wu-Leahy [27], διαμερίζει το γράφημα χρησιμοποιώντας ως κριτήριο διαμέρισης τη συνάρτηση  $SCut$ . Έστω η τομή που διαμερίζει το σύνολο των κόμβων  $V$  σε ένα υποσύνολο  $A \subset V$  και σε ένα υποσύνολο  $B \subset V$ , όπου  $A + B = V$ . Η τιμή  $SCut$  (το οποίο λέγεται και κόστος διαμέρισης), είναι το άθροισμα των βαρών των ακμών που συμμετέχουν στην τομή, δηλαδή των ακμών που έχουν το ένα άκρο στο υποσύνολο  $A$  και το άλλο άκρο στο υποσύνολο  $B$ .

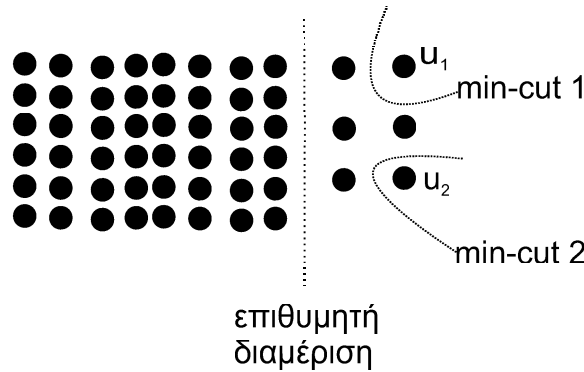
$$SCut(A, B) = \sum_{u \in A, v \in B} c(u, v).$$

Η ελάχιστη τιμή της συνάρτησης υπολογίζεται με θεωρία ροής και συγκεκριμένα με τη βοήθεια του δέντρου ελάχιστης τομής (minimum cut tree)  $T$ . Στη μέθοδο αυτή το δέντρο το κατασκευάζουν με βάση τον αλγόριθμο των Gomory-Hu. Μια διαμέριση συμβαίνει όταν υπάρχει στο αντίστοιχο δέντρο ελάχιστης τομής  $T$  μια ακμή με βάρος-κόστος μικρότερο από ένα προκαθορισμένο κατώφλι. Ο αλγόριθμος αυτός απαιτεί την επίλυση αρκετών προβλημάτων εύρεσης μέγιστης ροής. Ο αριθμός των προβλημάτων αυτών αυξάνει γραμμικά ως προς το πλήθος των κόμβων του γραφήματος (το πλήθος των κόμβων είναι της τάξης των δεκάδων χιλιάδων όταν το γράφημα μοντελοποιεί μια εικόνα). Για το λόγο αυτό, η άμεση εφαρμογή του αλγορίθμου των Gomory-Hu είναι αποδοτική μόνο σε πολύ μικρές εικόνες.

Σε μεγάλες εικόνες, χρησιμοποιούν μια ιεραρχική τεχνική κατά την οποία γίνονται συρρικνώσεις κόμβων πριν την κατασκευή του ισοδύναμου δέντρου ελάχιστης τομής. Η προεπεξεργασία αυτή μειώνει σημαντικά το γράφημα στο οποίο πρόκειται να εφαρμοστεί στη συνέχεια ο αλγόριθμος των Gomory-Hu. Η διαμέριση που επιλέγεται με την μέθοδο που περιγράφηκε, ελαχιστοποιεί το συνολικό κόστος της διαμέρισης, επομένως αποτελεί μια ακριβής λύση ελαχιστοποίησης του μέτρου  $SCut$  και διαμέρισης του  $G$ . Για μερικές εισόδους όμως, η ελαχιστοποίηση του κριτηρίου αυτού, έχει ως αποτέλεσμα πολλές διαμερίσεις να περιλαμβάνουν πολύ λίγους κόμβους. Μια λύση που προτείνουν, για την επίλυση αυτού του προβλήματος, είναι ένα επιπλέον βήμα συνένωσης των ομάδων που προκύπτουν με πλήθος κόμβων λιγότερο από 5 στις γειτονικές ομάδες με τις οποίες έχουν την μεγαλύτερη ομοιότητα.

Ένα απλό παράδειγμα στο οποίο φαίνεται ο λόγος για τον οποίο η ελαχιστοποίηση αυτού του κριτηρίου μπορεί να οδηγήσει στην διάσπαση μικρών συνόλων ή ακόμη και τετριμμένων συνόλων (singletons), φαίνεται στο παράδειγμα που απεικονίζεται στο Σχήμα 4.1. Εξ'

ορισμού η τιμή του  $SCut$  μεταβάλλεται ανάλογα με τον αριθμό των ακμών που συμμετέχουν στην τομή. Στο παράδειγμα θεωρούμε ότι το βάρος των ακμών είναι αντιστρόφως ανάλογο με την απόσταση των δύο στοιχείων. Παρατηρούμε ότι, η διάσπαση του στοιχείου  $n_1$  ή του στοιχείου  $n_2$  ελαχιστοποιεί το κριτήριο  $SCut$ . Ενώ η επιθυμητή διαμέριση δίνει μεγαλύτερη τιμή  $SCut$ . Επομένως ο αλγόριθμος διαμέρισης, σε ένα και μόνο βήμα, δε δίνει το επιθυμητό αποτέλεσμα της ομαδοποίησης.



Σχήμα 4.1: Παράδειγμα στο οποίο η ελαχιστοποίηση του κριτηρίου  $SCut$  οδηγεί σε μια κακή διαμέριση του γραφήματος.

#### 4.2.2 Μέθοδος Shi-Malik

Στον αλγόριθμο Shi-Malik [28] ορίζεται ένα μέτρο συσχέτισης μεταξύ των υπογραφημάτων  $A$  και  $B$  που προκύπτουν από τη τομή που διαμερίζει το σύνολο των κόμβων  $V$  σε ένα υποσύνολο  $A \subset V$  και σε ένα υποσύνολο  $B \subset V$ , όπου  $A+B = V$ . Το βάρος των ακμών που συνδέουν κόμβους στο  $A$  με κόμβους στο  $B$  συμβολίζεται με την ποσότητα  $Assoc(A, B)$ . Στην συνέχεια το γράφημα διαμερίζεται ελαχιστοποιώντας το κανονικοποιημένο κόστος διαμέρισης, θέλοντας να αποφύγουν την τάση του προηγούμενου αλγόριθμου να διαμερίσει σε πολλά μικρά υποσύνολα κόμβων. Όρισαν δηλαδή τις σχέσεις που ακολουθούν:

$$Assoc(A, V) = \sum_{u \in A, v \in V} w(u, v)$$

$$NCut(A, B) = \frac{SCut(A, B)}{Assoc(A, V)} + \frac{SCut(A, B)}{Assoc(B, V)}$$

Εντούτοις, ο υπολογισμός της ποσότητας  $NCut$ , είναι πιο περίπλοκος από τον υπολογισμό της ποσότητας  $SCut$  και επιπλέον ο υπολογισμός αυτός έχει αποδειχθεί ότι είναι NP-complete. Για το λόγο αυτό δίνουν προσεγγιστικές λύσεις στο πρόβλημα της ελαχιστοποίησης. Ο αλγόριθμος τους αποτελείται από τα πιο κάτω βήματα:

1. Μοντελοποίηση της εικόνας με ένα γράφημα  $G(V, E)$  και ορισμός των βαρών των ακμών ως την ομοιότητα μεταξύ των δύο κόμβων που συνδέουν.
2. Επίλυση του συστήματος  $(D - W)x = \lambda Dx$  αναζητώντας τα ιδιοδιανύσματα με τις μικρότερες ιδιοτιμές.
3. Χρησιμοποίησε το ιδιοδιάνυσμα με τη δεύτερη μικρότερη ιδιοτιμή για την διαμέριση του γραφήματος. Η διαμέριση πιο συγκεκριμένα γίνεται επιλέγοντας το κατάλληλο σημείο διαίρεσης που δίνει το μικρότερο  $NCut$ .
4. Αποφασίζει πιο είναι το κατάλληλο τμήμα της διαμέρισης που πρέπει να διαμερίσει στην επόμενη επανάληψη.

Η αλγεβρική συνδεσιμότητα ενός γραφήματος  $G$ , ορίζεται ως η δεύτερη μικρότερη ιδιοτιμή του Laplacian πίνακα. Ο Laplacian πίνακας είναι ο πίνακας που προκύπτει από την αφαίρεση του πίνακα γειτνίασης  $W$  του γραφήματος (όπου το στοιχείο  $W_{i,j}$  του πίνακα  $W$  είναι το βάρος της ακμής που συνδέει τους κόμβους  $i, j$ ) από τον διαγώνιο πίνακα  $D$  (τα στοιχεία της κύριας διαγωνίου  $d_i$  είναι το άθροισμα των βαρών των ακμών που προσπίπτουν στον κόμβο  $i$ ). Η τιμή της ιδιοτιμής αυτής είναι μεγαλύτερη του μηδενός αν και μόνο αν το γράφημα είναι συνεκτικό. Με άλλα λόγια, το πλήθος των εμφανίσεων μιας μηδενικής ιδιοτιμής, στον Laplacian πίνακα, δηλώνει και το πλήθος των συνεκτικών συνιστωσών του γραφήματος.

### 4.2.3 Μέθοδος Scanlon-Deo

Παρουσίασαν έναν προσεγγιστικό αλγόριθμο διαμέρισης. Κατά τη διάρκεια της ανάλυσης χαρακτηριστικών, παράγεται ένα πλήρες γράφημα με βάρη,  $K_n$ . Οι κόμβοι του γραφήματος αντιστοιχούν στα εικονοστοιχεία ή σε ομάδες εικονοστοιχείων, και τα βάρη των ακμών αντιστοιχούν στη ομοιότητα των κόμβων του οποίους συνδέουν. Προτείνουν ένα άλλο κριτήριο διαμέρισης  $BCut(A, B) = \frac{SCut(A, B)}{Edges(A, B)}$ , όπου το  $Edges(A, B)$  είναι το πλήθος των ακμών που συμμετέχουν στην τομή του συνόλου  $V$  στα υποσύνολα  $A$  και  $B$ . Ο αλγόριθμος των Gomory-Hu δε μπορεί να χρησιμοποιηθεί για την ελαχιστοποίηση αυτού του κριτηρίου. Η επιλογή του ελάχιστου κόστους από  $2^{n-1}$  πιθανές διαμερίσεις, χρειάζεται εκθετικό χρόνο.

Η μέθοδος που προτείνουν προσπαθεί να καθορίσει το maximal συνδετικό δέντρο (maximal spanning tree) (MST). Από τις  $2^{n-1}$  πιθανές διαμερίσεις, μόνο οι τομές που προκύπτουν από την αφαίρεση των ακμών του MST αποτιμούνται. Τα βήματα του αλγορίθμου που προτείνουν είναι:

1. Υπολογισμός του MST έστω  $T$  του αρχικού γραφήματος  $G$ .
2. Για κάθε ακμή του δέντρου  $T$ , αντικατέστησε το βάρος της ακμής  $e$  με το κόστος της διαμέρισης που προκύπτει με την αφαίρεση της ακμής  $e$ .
3. Χρησιμοποίησε την ακμή  $e$  για να διαμερίσεις το γράφημα αν η ακμή  $e$  είναι εσωτερική στο  $T$  και το βάρος-κόστος της είναι μικρότερο από μια τιμή  $\delta$ .

Η διαδικασία αφαίρεσης ακμών από το  $T$ , εκτελείται αφού προηγουμένως ταξινομηθούν οι ακμές του  $T$  κατά αύξουσα σειρά με βάση το βάρος τους. Στη συνέχεια, από την ταξινομημένη λίστα ακμών αφαιρούνται τόσες ακμές όσες ανάλογα με το πλήθος των ομάδων που απαιτεί το πρόβλημα. Οι συνεκτικές συνιστώσες του  $T$  που προκύπτουν μετά την αφαίρεση των ακμών αποτελούν και τις τελικές ομάδες του προβλήματος. Ο αλγόριθμος υπολογισμού του  $T$  παρουσιάζεται στην Ενότητα 1.2.

#### 4.2.4 Κατάτμηση Εικόνας με μεθόδους Ομαδοποίησης

Στην συνέχεια, παρουσιάζονται κάποιες μέθοδοι κατάτμησης εικόνας, που χρησιμοποιούν θεωρία γράφων αλλά η κατάτμηση τελικά επιτυγχάνεται με τον αλγόριθμο ομαδοποίησης  $K$ -μέσων. Πιο συγκεκριμένα παρουσιάζονται οι αλγόριθμοι:

1. Average Association (μεγιστοποίηση του κριτηρίου  $Nassoc$ )
2. Average Cut (ελαχιστοποίηση του κριτηρίου  $ACut$ )
3. Normalized Cut (ελαχιστοποίηση του κριτηρίου  $NCut$ )

οι οποίοι ελαχιστοποιούν τις πιο κάτω ποσότητες αντίστοιχα:

**Κριτήριο  $Nassoc$**

$$Nassoc = \max_{A \subset V, B \subset V} \frac{assoc(A,A)}{|A|} + \frac{assoc(B,B)}{|B|}$$

**Κριτήριο  $ACut$**

$$Nassoc = \min_{A \subset V, B \subset V} \frac{cut(A,B)}{|A|} + \frac{cut(A,B)}{|B|}$$

**Κριτήριο  $NCut$**

$$Nassoc = \min_{A \subset V, B \subset V} \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

Έχει δειχθεί ότι η ελαχιστοποίηση του κριτηρίου  $NCut$  μεγιστοποιεί το κριτήριο  $Nassoc$ . Έστω ότι το πρόβλημα που καλούμαστε να επιλύσουμε είναι η ομαδοποίηση των εικονοστοιχείων μια εικόνας σε  $k$  ομάδες. Αναπαριστούμε την εικόνα με ένα γράφημα και

υπολογίζουμε τους κατάλληλους πίνακες (που απαιτούνται ανάλογα με τη μέθοδο που θέλουμε να χρησιμοποιήσουμε) από το γράφημα. Οι αλγόριθμοι *Average association*, *Average Cut* και *Normalized Cut* λύνουν το σύστημα ιδιοτιμών  $Wx = \lambda x$ ,  $(D - W)x = \lambda x$  και  $(D - W)x = \lambda Dx$  αντίστοιχα. Ο πίνακας  $W$  είναι ο πίνακας βαρών ή γειτνίασης του γραφήματος (το στοιχείο  $W(i, j)$  είναι το βάρος που αντιστοιχεί στην ακμή που συνδέει τους κόμβους  $i, j$ ) και ο πίνακας  $D$  είναι ο διαγώνιος πίνακας όπου τα στοιχεία της διαγωνίου του αποτελούν τη συνολική σύνδεση του κόμβου  $i$  προς όλους τους άλλους (το στοιχείο  $D(i, i) = \sum_j w(i, j)$ ).

Οι αλγόριθμοι *Average Cut* και *Normalized Cut* επιλύουν το σύστημα ιδιοτιμών και βρίσκουν τις  $k$  μικρότερες ιδιοτιμές και στη συνέχεια τα  $k$  ιδιοδιανύσματα  $(u_1, \dots, u_k)$  που αντιστοιχούν σε αυτές τις ιδιοτιμές. Στην συνέχεια δημιουργούν ένα πίνακα  $U = [u_1, \dots, u_k] \in \mathbb{R}^{N \times k}$ . Έπειτα κανονικοποιούν κάθε γραμμή του πίνακα  $U$  ώστε να έχουν μοναδιαίο μήκος και κάθε κανονικοποιημένη γραμμή αποτελεί γραμμή του πίνακα  $Y$ . Δηλαδή  $Y_{ij} = \frac{U_{ij}}{(\sum_j U_{ij}^2)^{\frac{1}{2}}}$ . Έπειτα, θεωρώντας κάθε γραμμή του πίνακα  $Y$  ως ένα σημείο στον  $\mathbb{R}^k$ , χρησιμοποιείται ο κλασικός αλγόριθμος ομαδοποίησης  $k$ -μέσων για την ομαδοποίηση τους σε  $k$  ομάδες. Τέλος το αρχικό σημείο  $x_i$  θα ανήκει στην ομάδα  $j$  αν και μόνο αν η γραμμή  $i$  του πίνακα  $Y$  ανήκει στην ομάδα  $j$ .

Ο αλγόριθμος *Average association*, ακολουθεί ακριβώς την ίδια διαδικασία με την διαφορά ότι δεν παίρνει τα  $k$  μικρότερα ιδιοδιανύσματα αλλά τα  $k$  μεγαλύτερα. Η υπόλοιπη διαδικασία που ακολουθείται είναι ακριβώς η ίδια.

Αναφέρουμε ότι οι αλγόριθμοι αυτοί είναι γνωστοί στη βιβλιογραφία και ως *spectral clustering* αλγόριθμοι επειδή βασίζονται στις αλγεβρικές ιδιότητες των γραφημάτων.

## ΚΕΦΑΛΑΙΟ 5

# ΕΦΑΡΜΟΓΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ STOER-WAGNER ΣΤΗΝ ΚΑΤΑΤΜΗΣΗ ΕΙΚΟΝΑΣ

---

### 5.1 Αλγόριθμος Stoer-Wagner

### 5.2 Εφαρμογή Αλγορίθμου SW στην Κατάτμηση Εικόνας

---

## 5.1 Αλγόριθμος των Stoer-Wagner

Το 1997 οι Stoer και Wagner παρουσίασαν ένα ντεντερμινιστικό αλγόριθμο εύρεσης ελάχιστης τομής σε ένα μη κατευθυνόμενο γράφημα  $G$  με κόμβους  $V(G)$  και ακμές  $E(G)$  με ακμικά βάρη  $w_e$ , όπου  $e \in E(G)$ . Ο αλγόριθμος αυτός είναι πολύ απλός και είναι εύκολος στην υλοποίηση. Σε αντίθεση με άλλους αλγόριθμους που επιλύουν το πρόβλημα εύρεσης ελάχιστης τομής που είχαν βρεθεί μέχρι και το 1997, δε χρησιμοποιεί τεχνικές ροής. Σε γενικές γραμμές, ο αλγόριθμος αποτελείται από  $|V(G)|$  φάσεις. Σε κάθε φάση κατασκευάζεται μια διάταξη γειτνίασης  $A$  (maximum adjacency order).

Η εύρεση της ελάχιστης τομής σε ένα μη κατευθυνόμενο γράφημα με βάρη αποτελεί θεμελιώδες αλγοριθμικό πρόβλημα. Ουσιαστικά διαμερίζει ένα γράφημα με  $|V|$  κόμβους σε δύο υπογραφήματα, έτσι ώστε το άθροισμα των βαρών των ακμών που συμμετέχουν στην τομή (cut weight) να ελαχιστοποιείται.

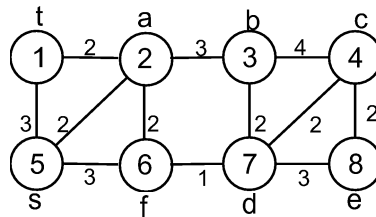
Ο αλγόριθμος βασίζεται στην παρατήρηση στο ότι αν έχουμε δύο κόμβους  $s$  και  $t$  του γραφήματος  $G$  και το γράφημα  $G/\{s, t\}$  που προκύπτει από τη συνένωση των κόμβων  $s$  και

$t$ , τότε η ελάχιστη τομή του  $G$  μπορεί να ληφθεί παίρνοντας το μικρότερο από τις ποσότητες ελάχιστης  $(s, t)$ -τομής του  $G$  και της ελάχιστης τομής του  $G/\{s, t\}$  (βλ. Κεφάλαιο 3.4).

Στο κεφάλαιο αυτό, υπενθυμίζουμε τη διαδικασία που ακολουθεί ο αλγόριθμος παραθέτοντας το παράδειγμα που ακολουθεί.

### Παράδειγμα Αλγόριθμου Εύρεσης Ελάχιστης Τομής Stoer-Wagner (SW)

Για τον υπολογισμό της ελάχιστης τομής του μη-κατευθυνόμενου γραφήματος  $G$  που φαίνεται στο Σχήμα 5.1, ακολουθούνται τα πιο κάτω βήματα.

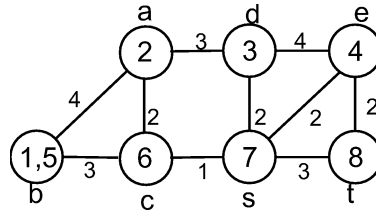


Σχήμα 5.1: Το γράφημα  $G$  στο οποίο πρόκειται να εφαρμόσουμε τον αλγόριθμο SW.

**Βήμα 1:** Αρχικά εκτελείται η υπορουτίνα *MinimumCutPhase* που περιγράφεται από τον Αλγόριθμο 6 που περιγράφεται στο Κεφάλαιο 3.4 με είσοδο το γράφημα  $G$  και έναν αρχικό κόμβο  $a = 2$ . Στη συνέχεια δημιουργείται η διάταξη γειτνίασης  $A$  η οποία αρχικά περιέχει μόνο τον κόμβο  $a = 2$ . Υπολογίζονται οι γειτνιάσεις των κόμβων του συνόλου  $V(G) - 2$  και εισάγονται σε μία ουρά προτεραιότητας με κλειδί τη γειτνίαση του κάθε κόμβου του συνόλου αυτού με τον κόμβο  $a = 2$  (στην γενική περίπτωση το κλειδί είναι η γειτνίαση του κόμβου με όλους τους κόμβους που έχουν ήδη εισαχθεί στη διάταξη  $A$ ). Στην συνέχεια, επιλέγεται ο κόμβος με την μέγιστη συνδεσιμότητα από την ουρά -στο παράδειγμα μας ο κόμβος  $b$ - και εισάγεται στο τέλος της διάταξης  $A$  (αφού διαγραφεί από την ουρά προτεραιότητας). Έπειτα για κάθε κόμβο  $u$ , στην ουρά προτεραιότητας, γείτονα του  $b$ , αυξάνεται το κλειδί του κατά μια ποσότητα ίση με το βάρος της ακμής που συνδέει τον  $u$  με τον  $b$ . Στην συνέχεια, επιλέγεται ο κόμβος  $c$ , από την ουρά προτεραιότητας, και εισάγεται στο τέλος της διάταξης  $A$ . Τα κλειδιά των κόμβων-γειτόνων του  $c$ , αυξάνονται με τον ίδιο τρόπο που περιγράψαμε πιο πάνω. Η διαδικασία αυτή επαναλαμβάνεται έως ότου διαγράψουμε και τον τελευταίο κόμβο από την ουρά προτεραιότητας. Επομένως, η διάταξη που προκύπτει είναι  $A = \{a, b, c, d, e, f, s, t\}$ . Έχει δειχθεί ότι το κλειδί του τελευταίου κόμβου που εισήχθη στην διάταξη  $A$  είναι ίσο με την ελάχιστη  $(s, t)$ -τομή, όπου  $s, t$  είναι οι δύο τελευταίοι κόμβοι που εισήχθησαν στην διάταξη  $A$ . Σύμφωνα με το Θεώρημα 3.1, η ελάχιστη τομή του  $G$  είναι είτε η ελάχιστη  $(s, t)$ -τομή, είτε η ελάχιστη τομή του γραφήματος  $G \setminus \{s, t\}$  που

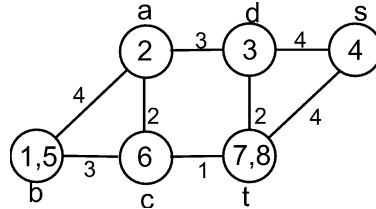


προκύπτει μετά την συρρίκνωση των κόμβων  $s, t$ . Επομένως, μια υποψήφια ελάχιστη τομή του γραφήματος  $G$  αποτελεί η ελάχιστη  $(s, t)$ -τομή κατά την οποία το σύνολο των κόμβων του γραφήματος  $G$  χωρίζεται στα υποσύνολα  $\{1\}, \{2, 3, 4, 5, 6, 7, 8\}$  και έχει τιμή ίση με  $w = 5$ .



Σχήμα 5.2: Το γράφημα  $G\{s, t\}$  το οποίο προκύπτει μετά τη συρρίκνωση των κόμβων  $s, t$ .

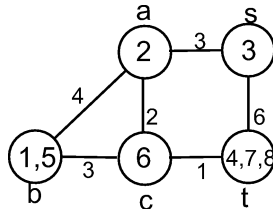
**Βήμα 2:** Έπειτα η πιο πάνω διαδικασία επαναλαμβάνεται για το γράφημα  $G\{s, t\}$  που φαίνεται στο Σχήμα 5.2. Προκύπτει η διάταξη  $A = \{a, b, c, d, e, s, t\}$  και επομένως η υποψήφια ελάχιστη τομή του  $G$  που αποτελείται από τα υποσύνολα  $\{8\}, \{1, 2, 3, 4, 5, 6, 7\}$  και έχει τιμή με  $w = 5$ . Έπειτα οι τελευταίοι κόμβοι της διάταξης συρρικνώνονται και προκύπτει το γράφημα που φαίνεται στο Σχήμα 5.3.



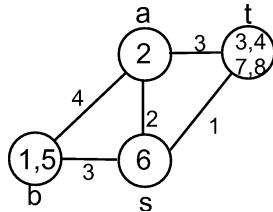
Σχήμα 5.3: Το γράφημα που προκύπτει μετά τη δεύτερη εκτέλεση της MinCutPhase.

**Βήμα 3:** Έπειτα η διαδικασία επαναλαμβάνεται για το γράφημα που φαίνεται στο Σχήμα 5.3. Προκύπτει η υποψήφια ελάχιστη τομή του  $G$  που αποτελείται από τα υποσύνολα  $\{7, 8\}, \{1, 2, 3, 4, 5, 6\}$  και έχει τιμή με  $w = 7$ . Έπειτα οι τελευταίοι κόμβοι της διάταξης συρρικνώνονται και προκύπτει το γράφημα που φαίνεται στο Σχήμα 5.4.

**Βήμα 4:** Έπειτα η διαδικασία επαναλαμβάνεται για το γράφημα που φαίνεται στο Σχήμα 5.4. Προκύπτει η υποψήφια ελάχιστη τομή του  $G$  που αποτελείται από τα υποσύνολα  $\{4, 7, 8\}, \{1, 2, 3, 5, 6\}$  και έχει τιμή με  $w = 7$ . Μετά τη συρρίκνωση των δύο τελευταίων κόμβων της διάταξης προκύπτει το γράφημα που φαίνεται στο Σχήμα 5.5.

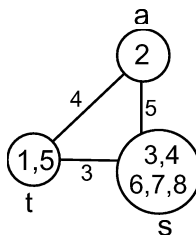


Σχήμα 5.4: Το γράφημα που προκύπτει μετά την τρίτη εκτέλεση της MinCutPhase.



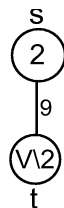
Σχήμα 5.5: Το γράφημα που προκύπτει μετά την τέταρτη εκτέλεση της MinCutPhase.

**Βήμα 5:** Έπειτα η διαδικασία επαναλαμβάνεται για το γράφημα που φαίνεται στο Σχήμα 5.5. Προκύπτει η υποψήφια ελάχιστη τομή του  $G$  που αποτελείται από τα υποσύνολα  $\{3, 4, 7, 8\}$ ,  $\{1, 2, 5, 6\}$  και έχει τιμή με  $w = 4$ . Έπειτα οι τελευταίοι κόμβοι της διάταξης συρρικνώνονται και προκύπτει το γράφημα που φαίνεται στο Σχήμα 5.6.



Σχήμα 5.6: Το γράφημα που προκύπτει μετά την πέμπτη εκτέλεση της MinCutPhase.

**Βήμα 6:** Η υποψήφια ελάχιστη τομή που προκύπτει από το γράφημα που φαίνεται στο Σχήμα 5.6 είναι η  $\{1, 5\}$ ,  $\{2, 3, 4, 6, 7, 8\}$  και έχει τιμή με  $w = 7$ . Το γράφημα που προκύπτει μετά τη συρρίκνωση φαίνεται στο Σχήμα 5.7.-



Σχήμα 5.7: Το γράφημα που προκύπτει μετά την έκτη εκτέλεση της MinCutPhase.

**Βήμα 7:** Στη τελευταία επανάληψη του αλγορίθμου (συνολικά  $n - 1$  επαναλήψεις) προκύπτει η υποψήφια τομή που αντιστοιχεί στη διαμέριση  $\{2\}$ ,  $V(G) - \{2\}$  και έχει τιμή με  $w = 9$ .

Από όλες τις  $n - 1$  υποψήφιες ελάχιστες τομές που υπολογίστηκαν σε κάθε φάση του αλγορίθμου SW επιλέγουμε την υποψήφια τομή η οποία έχει το μικρότερο βάρος τομής. Αυτή αποτελεί και την ελάχιστη τομή του γραφήματος  $G$ . Στο παράδειγμα η διαμέριση στην οποία προκύπτουν τα υποσύνολα  $\{3, 4, 7, 8\}$  και  $\{1, 2, 5, 6\}$  αποτελεί και την ελάχιστη τομή του  $G$  και έχει τιμή  $w = 4$ .

## 5.2 Εφαρμογή Αλγορίθμου SW στην Κατάτμηση Εικόνας

Το 1993 οι Wu-Leahy [27] (MCT) πρότειναν έναν αλγόριθμο βέλτιστης διαμέρισης γραφήματος σε  $k$  υπογραφήματα έτσι ώστε η συνδεσιμότητα (ομοιότητα) μεταξύ διαφορετικών ομάδων να ελαχιστοποιείται. Δηλαδή το κριτήριο διαμέρισης που χρησιμοποιούν είναι το κριτήριο ελάχιστης τομής. Όπως αναφέραμε και σε προηγούμενες ενότητες, το κριτήριο αυτό μειονεκτεί δεδομένου ότι τείνει να διαμερίσει το γράφημα σε δύο ανεξάρτητα υπογραφήματα εκ των οποίων το ένα επάγεται είτε από έναν τετριμμένο κόμβο, είτε από ένα μικρό πλήθος κόμβων. Για να αποφύγουν τέτοιου είδους διαμερίσεις επιλέγουν μια συνάρτηση βαρών η οποία φθίνει γρήγορα στο μηδέν. Επομένως μια τομή στην οποία συμμετέχουν λίγες ακμές που συνδέουν εικονοστοιχεία με μικρές διαφορές φωτεινότητας, έχει μεγαλύτερη χωρητικότητα από μια τομή στην οποία συμμετέχουν πολλές ακμές που συνδέουν εικονοστοιχεία με μεγάλες διαφορές φωτεινότητας.

Επιλύουν το πρόβλημα εύρεσης ελάχιστης τομής με τον αλγόριθμο Gomory-Hu, ο οποίος χρησιμοποιεί  $n - 1$  υπολογισμούς μέγιστης ροής. Γεγονός που κάνει την εφαρμογή του σε εικόνα με μεγάλο πλήθος εικονοστοιχείων ( $> 10000$ ), δύσκολη. Παρόλα αυτά πρότειναν κάποιους τρόπους βελτίωσης της πολυπλοκότητας χρόνου του αλγορίθμου.

Το 2005 οι Flake-Tarjan-Tsioutsoulouklis [29], χρησιμοποίησαν δέντρα ελάχιστης τομής για την εύρεση κοινοτήτων διαδικτύου (web communities). Κατασκευάζουν ένα δέντρο ελάχιστης τομής με βάση τον αλγόριθμο Gomory-Hu. Σε αντίθεση με τον αλγόριθμο MCT, δεν προσπαθούν να ελαχιστοποιήσουν μόνο το άθροισμα των βαρών που συμμετέχουν στην τομή αλλά παράλληλα με αυτό προσπαθούν να μεγιστοποιήσουν και το άθροισμα των βαρών των ακμών στα υπογραφήματα που προκύπτουν μετά τη διαμέριση. Αυτό το επιτυγχάνουν εισάγοντας, μια παράμετρο  $\alpha$  η οποία αποτελεί ένα συνδυαστικό κριτήριο

διαμέρισης (bicriterion). Το κριτήριο αυτό, λέγεται συνδυαστικό γιατί εξασφαλίζει τόσο την μικρή συνδεσιμότητα μεταξύ των ομάδων που προκύπτουν αλλά ταυτόχρονα και την μεγάλη συνδεσιμότητα μεταξύ των κόμβων μιας ομάδα. Επομένως, με τη βοήθεια αυτού του κριτηρίου αποφεύγουν την δημιουργία πολλών και μικρών ομάδων.

Αναπτύξαμε έναν ιεραρχικό αλγόριθμο διαμέρισης γραφήματος (HSW), ο οποίος βασίζεται επίσης σε αλγόριθμους εύρεσης ελάχιστης τομής. Συγκεκριμένα βασίζεται στον αλγόριθμο ελάχιστης τομής Stoer-Wagner ο οποίος δεν χρησιμοποιεί θεωρία ροής (όπως ο αλγόριθμος Gomory-Hu) αλλά σε τεχνικές συρρίκνωσης ακμών και τον εφαρμόσαμε στο πρόβλημα ομαδοποίησης-κατάταξης εικόνας. Ο αλγόριθμος είναι ιεραρχικός και σε κάθε βήμα διαμερίζει ένα (υπό)γράφημα (το οποίο αναπαριστά την εικόνα του προβλήματος) σε δύο ανεξάρτητα υπογραφήματα (ομάδες-περιοχές της εικόνας). Επομένως σε κάθε επίπεδο του αλγορίθμου, ο αριθμός των υπογραφημάτων (ομάδων) αυξάνεται κατά ένα.

Τα Βασικά ζητήματα που μας απασχόλησαν κατά την ανάπτυξη του αλγορίθμου HSW ήταν:

- Αναπαράσταση της εικόνας με γράφημα.
- Καθορισμός κατάλληλης συνάρτησης βαρών που να εκφράζει την ομοιότητα των εικονοστοιχείων.
- Πως θα μπορούσαμε να επιλύσουμε μερικώς το πρόβλημα που παρουσιάζει το κριτήριο ελάχιστης τομής.
- Επιλογή κατάλληλου υπογραφήματος προς διαμεριση.

### 5.2.1 Αναπαράσταση της Εικόνας με Γράφημα

Μια εικόνα αναπαρίσταται με ένα μη κατευθυνόμενο γράφημα  $G$  το οποίο ανήκει στην κλάση των πλεγμάτων (Grid). Κάθε στοιχείο της εικόνας  $(i, j)$  αναπαρίσταται με έναν κόμβο στο γράφημα  $G$ . Σε κάθε κόμβο  $v$  προσπίπτουν δύο ακμές οι οποίες συνδέουν τον κόμβο  $v$  με τον οριζόντια και κατακόρυφα γειτονικό κόμβο του αντίστοιχα. Δύο κόμβοι  $v, u$  είναι οριζόντια γειτονικοί αν αντιστοιχούν στα εικονοστοιχεία  $(i, j), (i, j + 1)$  αντίστοιχα. Λέμε ότι οι κόμβοι  $v, u$  είναι κατακόρυφα γειτονικοί αν αυτοί αντιστοιχούν στα εικονοστοιχεία  $(i, j), (i + 1, j)$ . Στη συνέχεια ορίσαμε μια συνάρτηση βαρών η οποία εκφράζει την ομοιότητα ομοιότητα γειτονικών κόμβων του  $G$  με βάση το χρώμα των εικονοστοιχείων που αυτά αντιπροσωπεύουν. Όσο η διαφορά φωτεινότητας των εικονοστοιχείων που αντιστοιχούν στους κόμβους  $v, u$  μεγαλώνει τόσο το βάρος που αντιστοιχεί στην ακμή  $vu$  μειώνεται.

## 5.2.2 Καθορισμός Συνάρτησης βαρών

Γενικά, όπως αναφέραμε και σε προηγούμενα κεφάλαια, η διαμέριση με το κριτήριο ελάχιστης τομής μπορεί να μην είναι ικανοποιητική για κάποια προβλήματα, δεδομένου ότι μπορεί να προκύψουν δύο τμήματα  $A$  και  $B$  όπου  $A = \{u\}$ ,  $B = V - \{u\}$ , και  $u$  ο κόμβος του  $G$  με τον ελάχιστο βαθμό βαρών (weight degree). Μια λύση στο πρόβλημα αυτό είναι να οριστεί συνάρτηση βάρους (ακμών) η οποία να φθίνει απότομα στο μηδέν ως προς τη διαφορά έντασης φωτεινότητας των εικονοστοιχείων που συνδέει. Με τον τρόπο αυτό μια τομή στην οποία συμμετέχουν πολλές ακμές των οποίων τα βάρη τείνουν στο μηδέν μπορεί να προτιμηθεί αντί της απομόνωσης του κόμβου με τον ελάχιστο βαθμό.

Για παράδειγμα μια τέτοια συνάρτηση ομοιότητας μπορεί να είναι η ακόλουθη

$$SCap = e^{-\left(\frac{x_{i,j} - x_{i+1,j}}{\sigma}\right)^2} \quad (5.1)$$

Μια τέτοια συνάρτηση όμως μπορεί να έχει ως αποτέλεσμα απώλεια σημαντικής πληροφορίας της εικόνας, διότι δεν λαμβάνει υπόψη το μέγεθος της διαφοράς φωτεινότητας αν αυτή ξεπεράσει μια συγκεκριμένη τιμή. Επίσης η συνάρτηση αυτή χρησιμοποιεί μια παράμετρο  $\sigma$  η οποία είναι σταθερή σε όλη την εικόνα και επομένως μπορεί η τιμή αυτή να δίνει καλά αποτελέσματα σε κάποιες περιοχές της εικόνας και σε κάποιες άλλες όχι.

Για να αποφύγουμε την απώλεια πληροφορίας χρησιμοποιούμε συνάρτηση βαρών η οποία φθίνει αργά στο μηδέν. Στη συνάρτηση χρησιμοποιούμε μια παράμετρο η οποία εξαρτάται από τη περιοχή της εικόνας την οποία πρόκειται να διαμερίσουμε. Η παράμετρος αυτή ορίζεται έτσι ώστε στις ακμές που συνδέουν κόμβους που αντιστοιχούν σε εικονοστοιχεία με τη μεγαλύτερη διαφορά φωτεινότητας (στο υπογράφημα στο οποίο εργαζόμαστε σε κάθε βήμα) να αντιστοιχίζεται το μικρότερο βάρος.

$$MyCap^H = \exp\left(\frac{-D_{i,j}^H}{MaxIntensityDifference}\right), \quad \text{βάρος οριζόντιας ακμής} \quad (5.2)$$

$$MyCap^V = \exp\left(\frac{-D_{i,j}^V}{MaxIntensityDifference}\right), \quad \text{βάρος κατακόρυφης ακμής} \quad (5.3)$$

Οι ποσότητες  $D_{i,j}^H$  και  $D_{i,j}^V$  ορίζονται με τους ακόλουθους τύπους:

$$D_{i,j}^H = |\delta \cdot (x_{i,j} - x_{i,j+1}) + (x_{i-1,j} - x_{i-1,j+1}) + (x_{i+1,j} - x_{i+1,j+1}) + (x_{i,j-1} - x_{i,j+2})| \quad (5.4)$$

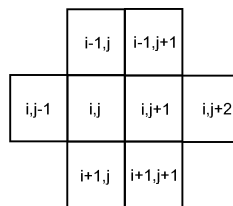
$$D_{i,j}^V = |\delta \cdot (x_{i,j} - x_{i+1,j}) + (x_{i,j-1} - x_{i+1,j-1}) + (x_{i,j+1} - x_{i+1,j+1}) + (x_{i-1,j} - x_{i+2,j})| \quad (5.5)$$

όπου η ποσότητα  $x_{i,j}$  συμβολίζει την φωτεινότητα του εικονοστοιχείου  $(i, j)$ .

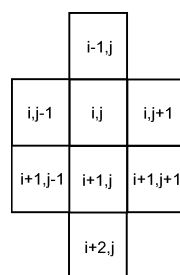
Η παράμετρος *MaxIntensityDifference* είναι ίση με τη μεγαλύτερη διαφορά ομοιότητας μεταξύ των στοιχείων του τμήματος της εικόνας που πρόκειται να διαμερίσουμε. Μετά από κάθε διαμέριση η ποσότητα *MaxIntensityDifference* έχει διαφορετική τιμή σε κάθε τμήμα εικόνας (υπογράφημα) που προκύπτει. Για το λόγο αυτό μετά από κάθε διαμέριση τα βάρη των ακμών των υπογραφημάτων που προκύπτουν επαναπροσδιορίζονται .

Επίσης για το καθορισμό του βάρους της ακμής  $v_1v_2$  λαμβάνονται υπόψη οι διαφορές φωτεινότητας ζευγών κόμβων στην γειτονιά των κόμβων  $v_1$  και  $v_2$ , με σκοπό τα βάρη των ακμών να επηρεάζονται λιγότερο από τυχόν θόρυβο στην εικόνα.

Πιο συγκεκριμένα, για τον καθορισμό του βάρους  $w(v_1, v_2)$  της ακμής  $v_1v_2$  που συνδέει τους κόμβους  $v_1$  και  $v_2$  που αντιστοιχούν στα εικονοστοιχεία  $(i, j)$  και  $(i, j + 1)$ , χρησιμοποιείται η γειτονιά των εικονοστοιχείων που φαίνονται στο Σχήμα 5.8. Ενώ για τον καθορισμό του βάρους  $w(v_1, v_2)$  της ακμής  $v_1v_2$  που συνδέει τους κόμβους  $v_1$  και  $v_2$  που αντιστοιχούν στα εικονοστοιχεία  $(i, j)$  και  $(i + 1, j)$ , χρησιμοποιείται η γειτονιά των εικονοστοιχείων που φαίνονται στο Σχήμα 5.9.



Σχήμα 5.8: Η γειτονιά των εικονοστοιχείων  $(i, j)$  και  $(i, j + 1)$  που συμμετέχει για τον καθορισμό του βάρους της ακμής που συνδέει τους αντίστοιχους κόμβους τους.



Σχήμα 5.9: Η γειτονιά των εικονοστοιχείων  $(i, j)$  και  $(i + 1, j)$  που συμμετέχει για τον καθορισμό του βάρους της ακμής που συνδέει τους αντίστοιχους κόμβους τους.

Η παράμετρος  $\delta$  χρησιμοποιείται ώστε να λαμβάνεται περισσότερο υπόψη η διαφορά φωτεινότητας των  $v_1, v_2$  σε σχέση με τις υπόλοιπες διαφορές. Ιδανικά στην εικόνα θα θέλαμε να χωρίσουμε δύο ομάδες αν αυτές μεταξύ τους έχουν μικρή ομοιότητα. Πράγματι η ποσότητα που αντιστοιχίζεται στα βάρη των ακμών (ομοιότητα) θα είναι μικρότερη όταν

οι διαφορές στη γειτονιά των στοιχείων είναι μεγάλες και αντίθετα τα βάρη ακμών είναι μεγαλύτερα όταν οι διαφορές στη γειτονιά των στοιχείων είναι μικρές. Η συνάρτηση αυτή είναι εκθετική αλλά θέλοντας να μην τείνει απότομα στο μηδέν, όπως η συνάρτηση 5.1 (στην περίπτωση αυτή ο αλγόριθμος δεν επιλέγει κατ' ανάγκη μια διαμέριση στην οποία οι διαφορές φωτεινότητας είναι οι μεγαλύτερες, δεδομένου ότι όταν οι διαφορές υπερβούν ένα όριο τιμών η συνάρτηση δίνει πάντα τιμή μηδέν) δεν χρησιμοποιείται ως εκθέτης στην εκθετική συνάρτηση το άθροισμα των διαφορών αλλά μια αναλογία ως προς την μέγιστη διαφορά φωτεινότητας στο υπογράφημα στο οποίο εργαζόμαστε. Επομένως, σε κάθε επανάληψη τα βάρη επαναπροσδιορίζονται ώστε η νέα τους τιμή να είναι συναρτήσει της μέγιστης διαφοράς φωτεινότητας των στοιχείων της κάθε ομάδας.

### 5.2.3 Επίλυση του Προβλήματος του Κριτηρίου Ελάχιστης τομής

Παρόλα αυτά, επιθυμώντας να δώσουμε προτεραιότητα στην διαμέριση που απομονώνει κάθε φορά το τμήμα που έχει την καθολικά μεγαλύτερη διαφορά στο υπογράφημα στο οποίο εργαζόμαστε, το πρόβλημα να οδηγηθούμε σε μια διαμέριση στην οποία προκύπτουν ομάδες με ελάχιστο πλήθος κόμβων παραμένει άλυτο. Σε αντίθετη περίπτωση, χρησιμοποιώντας τη συνάρτηση  $SCap$  για τον καθορισμό των βαρών το πρόβλημα περιορίζεται σημαντικά.

Για το λόγο αυτό τροποποιήσαμε τον αλγόριθμο  $SW$  σύμφωνα με κάποιες παρατηρήσεις. Ο αλγόριθμος εκτελεί  $n - 1$  φορές την υπορουτίνα  $MinimumCutPhase$ . Σε κάθε βήμα, υπολογίζει μια υποψήφια ελάχιστη  $(s, t)$ -τομή. Έπειτα ο αλγόριθμος  $SW$  από όλες τις υποψήφιες ελάχιστες  $(s, t)$ -τομές επιλέγει την τομή αυτή που ελαχιστοποιεί το κριτήριο ελάχιστης τομής. Σύμφωνα με όλα όσα είπαμε παραπάνω, η ελαχιστοποίηση του κριτηρίου αυτού οδηγεί σε μια διαμέριση στην οποία το ένα τμήμα αποτελείται από έναν κόμβο ή μικρό αριθμό κόμβων. Επομένως μια βελτίωση που προτείνουμε είναι να επιλέγουμε μια από τις υποψήφιες τομές όχι με βάση το κριτήριο ελάχιστης τομής, αλλά με βάση ένα διαφορετικό κριτήριο το οποίο θα λαμβάνει υπόψη και το πλήθος των κόμβων-στοιχείων που περιέχονται στις ομάδες που προκύπτουν. Προτείνουμε ως κριτήριο επιλογής της τομής, που διαμερίζει το σύνολο  $V$  σε δύο τμήματα  $A$  και  $B$ , την ελαχιστοποίηση της ποσότητας  $MyCut(A, B)$ :

$$MyCut(A, B) = \frac{SCut(A, B)}{(|C| + 1) \cdot \min\{|A|, |B|\}}$$

όπου η ποσότητα  $SCut(A, B)$  είναι ίση με το άθροισμα των βαρών που συμμετέχουν στην τομή,  $|C|$  είναι το πλήθος των ακμών που συμμετέχουν στην τομή και  $\min\{|A|, |B|\}$  είναι το πλήθος των κόμβων του μικρότερου τμήματος που προκύπτει από τη διαμέριση.

Παρατηρήσαμε ότι χρησιμοποιώντας τον συνδυασμό της συνάρτησης *MyCap* και του κριτηρίου διαμέρισης *MyCut* τα αποτελέσματα που λαμβάνονται από τον ιεραρχικό αλγόριθμο κατάτμησης εικόνας που προτείνουμε είναι συγκριτικά καλύτερα με αυτά που θα είχαμε αν χρησιμοποιούσαμε βάρη που να τείνουν απότομα στο μηδέν.

#### 5.2.4 Επιλογή Κατάλληλου Υπογράφηματος προς Διαμέριση

Ο αλγόριθμος μας όπως είπαμε είναι ιεραρχικός και χρησιμοποιεί τον τροποποιημένο αλγόριθμο SW (ModSW) (τροποποίηση κριτηρίου επιλογής μιας τομής από τις  $n - 1$  υποψήφιας) για την διαμέριση ενός υπογράφηματος σε δύο τμήματα. Αρχικά ο ModSW εφαρμόζεται στο γράφημα  $G$  και στα επόμενα επίπεδα της ιεραρχίας εφαρμόζεται στο υπογράφημα (ομάδα εικονοστοιχείων) του προηγούμενου επιπέδου με το μεγαλύτερο τετραγωνικό σφάλμα. Σε κάθε επανάληψη ο αλγόριθμος απομονώνει το τμήμα το οποίο συνδέεται λιγότερο με τα στοιχεία της ομάδας στην οποία ανήκει. Δηλαδή επιλέγει το υπογράφημα προς διαμέριση το οποίο έχει το μεγαλύτερο τετραγωνικό σφάλμα, σε σχέση με τα υπόλοιπα υπογράφηματα (ομάδες) που έχουν προκύψει μέχρι στιγμής. Επομένως, ο αλγόριθμος σε κάθε επανάληψη μειώνει το τετραγωνικό σφάλμα της ομαδοποίησης. Ο αλγόριθμος, τερματίζει είτε όταν το τετραγωνικό σφάλμα έχει μειωθεί πέραν από ένα κατώφλι το οποίο ορίζει ο χρήστης, είτε όταν προκύψουν οι επιθυμητές ομάδες.

Το τετραγωνικό σφάλμα μιας ομάδας  $i$  ( $SSE_i$ ) ορίζεται ως το άθροισμα των τετραγωνικών αποστάσεων των φωτεινοτήτων των εικονοστοιχείων της ομάδας  $i$  από τη μέση φωτεινότητα της ομάδας στην οποία ανήκουν. Έστω ότι το υπογράφημα  $C_i$  επάγεται από τους κόμβους  $V_{c_i}$  που αντιστοιχούν στα εικονοστοιχεία της ομάδας  $i$ , και χαρακτηρίζονται από την ποσότητα  $d$  (την φωτεινότητα του αντίστοιχου εικονοστοιχείου).

$$SSE_i = \sum_{d \in C_i} (d - \mu_i)^2, \text{ όπου } \mu_i = \frac{1}{|C_i|} \sum_{d \in C_i} d$$

Πιο απλά επιλέγεται το πιο ανομοιογενές τμήμα ως προς τις φωτεινότητες των εικονοστοιχείων στο τμήμα αυτό.

Στο σημείο αυτό, καλό είναι να αναφέρουμε ότι θα μπορούσαμε να επιλέξουμε την υποψήφια ομάδα προς διαμέριση με μια σειρά άλλων κριτηρίων. Θα περιοριστούμε στο να αναφέρουμε μόνο τρία από αυτά συμπεριλαμβανομένου και του κριτηρίου που εμείς επιλέξαμε, ώστε να δικαιολογήσουμε και την επιλογή μας.

#### Κριτήρια Επιλογής Κατάλληλης Ομάδας προς Διαμέριση



- A. Τετραγωνικό σφάλμα, το οποίο δίνει μια γενική εικόνα της ομάδας (σαν να είχαμε πλήρες γράφημα) η πολυπλοκότητα χρόνου για τον υπολογισμό αυτού του κριτηρίου είναι  $O(n)$ .
- B. Διάμετρο ομάδας, υπολογίζοντας δηλαδή την απόσταση των σημείων της ομάδας που απέχουν περισσότερο. Η πολυπλοκότητα χρόνου για τον υπολογισμό αυτής της ποσότητας είναι  $O(n^2)$ .
- Γ. Την ακτίνα της ομάδας, μέγιστη απόσταση στοιχείου ομάδας από το μέσο στοιχείο της ομάδας με πολυπλοκότητα χρόνου  $O(n)$  ή από το ενδιάμεσο στοιχείο με πολυπλοκότητα χρόνου  $O(n \log n)$

Παρατηρούμε ότι, τα κριτήρια Β και Γ είναι ευαίσθητα στο θόρυβο γιατί αρκεί μόνο ένα θορυβώδες εικονοστοιχείο για να αυξηθεί η ακτίνα ή η διάμετρος μιας ομάδας.

Το κριτήριο Α είναι απλό και πιο περιγραφικό διότι λαμβάνει υπόψη όλα τα στοιχεία της ομάδας. Η μέθοδος μας χρησιμοποιώντας αυτό το κριτήριο μπορεί να θεωρηθεί υβριδική δεδομένου ότι χρησιμοποιεί πλέγμα από τη μια για την αναπαράσταση μιας ομάδας (τιμήμα εικόνας), απλοποιώντας το πρόβλημα, ενώ το κριτήριο επιλογής της ομάδας προς διαμέριση, γίνεται λαμβάνοντας υπόψη την γενική εικόνα της ομάδας (όπως και στην περίπτωση που θα είχαμε πλήρες γράφημα).

Δυστυχώς όμως, ο αριθμός των ομάδων που απαιτείται για τον διαχωρισμό βασικών χαρακτηριστικών της εικόνας παραμένει αρκετά μεγάλος. Δηλαδή για να διαχωρίσουμε μια εικόνα σε ένα αριθμό βασικών ομάδων πρέπει να συμπεριληφθούν και αρκετές διαμερίσεις οι οποίες μπορεί να μην αποτελούν ουσιαστική ή χρήσιμη πληροφορία για κάποια προβλήματα, αλλά παρόλα αυτά το ένα από τα δύο τμήματα που πρόκειται να διαχωρίσει να είναι τόσο μικρό που να ικανοποιεί το κριτήριο διαμέρισης. Η ευαισθησία αυτή του αλγορίθμου δεν είναι πάντα αρνητική. Αντίθετα σε κάποια προβλήματα μπορεί να είναι και το ζητούμενο. Μπορεί το αποτέλεσμα του αλγορίθμου να χρησιμοποιηθεί και μέσω μιας μετά-επεξεργασίας να λάβουμε τις ομάδες που μας ενδιαφέρουν. Με άλλα λόγια η διαδικασία μπορεί να θεωρηθεί ως και ένα στάδιο προ-επεξεργασίας. Η μετά-επεξεργασία μπορεί να γίνει με έναν αλγόριθμο ομαδοποίησης (συνένωσης ομάδων).

### Προσεγγίσεις:

- Συνένωση των μικρών ομάδων ( $< k$ -μέγεθος)
- Εφαρμογή της διαδικασίας Labeling (στην περίπτωση χρειάζονται κάποιες εικόνες

παραδείγματα για το πρόβλημα το οποίο λύνουμε, για τις οποίες έχουμε μια αντιστοιχία των φωτεινότητων σε κατηγορίες). Στη περίπτωση αυτή όλες οι ομάδες που προκύπτουν αντιστοιχίζονται στις πιο κοντινές περιοχές (ως προς τη μέση διαφορά φωτεινότητας τους) της εικόνας-παράδειγμα.

- Εφαρμογή του συσσωρευτικού αλγόριθμου ομαδοποίησης (Agglomerative) ή κάποιου άλλου αλγόριθμου ομαδοποίησης, όπως για παράδειγμα του  $K$ -μέσων. Κάτι τέτοιο δε θα είχε καλά αποτελέσματα αν εφαρμοζόταν σε προβλήματα στα οποία η χωρική τοπολογία των αντικειμένων είναι σημαντική και επιπλέον στην περίπτωση που χρησιμοποιείται ο  $K$ -μέσων το αποτέλεσμα θα εξαρτιόταν σε μεγάλο βαθμό από την αρχικοποίηση.
- Προτείνουμε συνένωση των γειτονικών ομάδων με τον αλγόριθμο MinCutTree (MCT) ακολουθώντας τη διαδικασία που περιγράφεται στον Αλγόριθμο 10.

### 5.2.5 Μερική Συνένωση των Ομάδων με Αλγόριθμο MCT (MIN CUT TREE)

Ο ιεραρχικός αλγόριθμος που αναπτύξαμε βελτιώνει σημαντικά το πρόβλημα που παρουσιάζει η διαμέριση η οποία χρησιμοποιεί το κριτήριο ελάχιστης τομής, παρόλα αυτά το αποτέλεσμα που προκύπτει αποτελείται από αρκετές ομάδες. Προτείνουμε συνένωση των ομάδων αυτών με τον αλγόριθμο MCT αν επιλύουμε προβλήματα στα οποία μας ενδιαφέρει η χωρική τοπολογία των περιοχών της εικόνας. Η διαδικασία που ακολουθείται είναι η εξής:

ΑΛΓΟΡΙΘΜΟΣ ModHSW-MCT( $G(V,E)$ )

**Βήμα 1:** Εκτέλεσε τον ModHSW ιεραρχικό αλγόριθμο.

**Βήμα 2:** Δημιούργησε ένα γράφημα  $G_{ModHSW}$  ως εξής:

Για κάθε ομάδα  $i$  που προκύπτει από την εκτέλεση του ModHSW ιεραρχικού αλγορίθμου, δημιούργησε κόμβο  $v_i$ .

Δημιούργησε την ακμή  $v_i v_j$  αν και μόνο αν η ομάδα  $i$  και η ομάδα  $j$  είναι γειτονικές στο αρχικό γράφημα  $G$ .

Δώσε βόρος στην ακμή  $v_i v_j$  ίσο με το άθροισμα των βαρών των ακμών που συνδέουν τις κόμβους της ομάδας  $i$  με κόμβους της ομάδας  $j$ .

**Βήμα 3:** Εκτέλεσε τον MCT αλγόριθμο.

**Βήμα 4:** Επέστρεψε τις ομάδες που προκύπτουν από τον MCT.

Αλγόριθμος 10: Αλγόριθμος των ModHSW-MCT.

Η υποποίηση του ιεραρχικού αλγορίθμου έγινε σε LEDA. Ο υπολογισμός του τετραγωνικού σφάλματος σε κάθε ομάδα απαιτεί  $O(n)$ , και ο επαπροσδιορισμός των βαρών γίνεται σε  $O(m)$ . Η τροποποίηση στον αλγόριθμο SW δεν αυξάνει την πολυπλοκότητα του αλγορίθμου και επομένως η πολυπλοκότητα του επαναληπτικού αλγορίθμου που προτείνουμε είναι τελικά  $O(Mnm + Mn^2 \log n)$ , όπου το  $M$  συμβολίζει τον αριθμό των ομάδων που προκύπτουν και επομένως το πλήθος των επαναλήψεων του ιεραρχικού αλγορίθμου.

## ΚΕΦΑΛΑΙΟ 6

# ΑΛΓΟΡΙΘΜΟΣ ΟΜΑΔΟΠΟΙΗΣΗΣ ΜΕ ΔΕΝΤΡΑ ΕΛΑΧΙΣΤΗΣ ΑΠΟΚΟΠΗΣ

---

6.1 Βασική Ιδέα

6.2 Επιλογή  $\alpha$

6.3 Ευρεστική Μέθοδος για τον Αλγόριθμο Cut-clustering

6.4 Ιεραρχικός Αλγόριθμος Cut-clustering

---

### 6.1 Βασική Ιδέα

Παρουσιάζεται ένας νέος αλγόριθμος ομαδοποίησης ο οποίος βασίζεται σε τεχνικές μέγιστης ροής και συγκεκριμένα σε δέντρα ελάχιστης τομής. Οι αλγόριθμοι μέγιστης ροής είναι γρήγοροι και απλοί. Η βασική ιδέα των τεχνικών ομαδοποίησης με βάση τη θεωρία μέγιστης ροής, είναι η δημιουργία ομάδων με μικρή συνδεσιμότητα μεταξύ τους (μικρή ομοιότητα ανάμεσα στις ομάδες που δημιουργούνται) και μεγάλη συνδεσιμότητα εσωτερικά σε κάθε μια από αυτές. Γενικά αυτό εξασφαλίζει την ισχυρή συνδεσιμότητα στο εσωτερικό της κάθε ομάδας και ένα ισχυρό κριτήριο για μια καλή ομαδοποίηση.

Η μέθοδος δουλεύει ως εξής: ένας τεχνητός κόμβος  $t$  προστίθεται στο γράφημα και συνδέεται με όλους τους κόμβους του γραφήματος. Οι μέγιστες ροές έπειτα υπολογίζονται στο νέο γράφημα που προκύπτει μετά την προσθήκη του κόμβου  $t$ . Για ένα γράφημα  $G$ ,

υπάρχει ένα γράφημα με βάρη  $T_G$ , το οποίο ονομάζεται δέντρο ελάχιστης τομής του  $G$ . Το δέντρο τομής ορίζεται από το σύνολο κόμβων  $V(G)$  και έχει την ιδιότητα ότι μπορούμε να βρούμε την ελάχιστη τομή αφαιρώντας από το δέντρο την ακμή με το ελάχιστο βάρος. Παρόλο που οι ροές και οι τομές ορίζονται καλά τόσο στα κατευθυνόμενα όσο και στα μη κατευθυνόμενα γραφήματα, εντούτοις τα δέντρα ελάχιστης αποκοπής ορίζονται μόνο στα μη κατευθυνόμενα γραφήματα. Γι' αυτό και η μέθοδος που αναλύεται στην ενότητα αυτή περιορίζεται μόνο στα μη κατευθυνόμενα γραφήματα. Έστω  $G(V, E)$  ένα μη κατευθυνόμενο υπογράφημα και έστω  $s, t \in V$  δύο κόμβοι του γραφήματος  $G$ . Έστω  $(S, T)$  η ελάχιστη τομή ανάμεσα στους κόμβους  $s$  και  $t$ , κατά την οποία  $s \in S$  και  $t \in T$ . Το  $S$  αποτελεί την κοινότητα (community) του  $s$  στο  $G$  σε σχέση με το  $t$ . Αν η ελάχιστη τομή των κόμβων  $s, t$  δεν είναι μοναδική, τότε επιλέγεται η αποκοπή αυτή η οποία ελαχιστοποιεί το μέγεθος της κοινότητας  $S$ . Σε αυτή την περίπτωση αποδεικνύεται ότι το  $S$  είναι μοναδικό.

**Λήμμα 6.1.** Σε ένα μη κατευθυνόμενο γράφημα  $G(V, E)$ , το σύνολο  $S$  ορίζει μια κοινότητα του κόμβου  $s$  σε σχέση με τον κόμβο  $t$ , αν ικανοποιείται η ακόλουθη σχέση.

$$\sum_{v \in S} w(u, v) > \sum_{v \in V - S} w(u, v), \forall u \in S - \{s\}.$$

Η απόδειξη του πιο πάνω λήμματος είναι απλή. Επειδή  $S$  είναι η κοινότητα του  $s$  σε σχέση με το  $t$ , η τομή  $(S, V - S)$  είναι και η ελάχιστη στο γράφημα  $G$ . Συνεπώς, κάθε κόμβος  $w \in S - \{s\}$  έχει μεγαλύτερη συνδεσιμότητα με τους κόμβους του συνόλου  $S$  σε σχέση με τους κόμβους του συνόλου  $V - S$ . Διαφορετικά, η μετακίνηση ενός κόμβου  $w$  στο σύνολο  $V - S$  θα οδηγούσε σε μικρότερη τομή ανάμεσα στους κόμβους  $s$  και  $t$ .

Μια κοινότητα διαδικτύου (web community) ορίζεται ως μια συλλογή κόμβων οι οποίοι ικανοποιούν την εξής ιδιότητα:

$$\sum_{v \in S} w(u, v) > \sum_{v \in V - S} w(u, v), \forall u \in S.$$

Συμπεραίνουμε ότι η κοινότητα  $S$ , η οποία βασίζεται στις ελάχιστες τομές, αποτελεί και κοινότητα διαδικτύου. Ο μοναδικός κόμβος που μπορεί να παραβιάσει αυτή την ιδιότητα είναι ο κόμβος  $s$ . Σε αυτή τη περίπτωση, είναι καλό να χρησιμοποιείται ως κοινότητα του  $s$  μια μεγαλύτερη κοινότητα η οποία έχει ως υποσύνολο το σύνολο  $S$ .

Στη περίπτωση όμως του αλγορίθμου ομαδοποίησης ο κόμβος  $t$  δεν είναι δεδομένος. Για το λόγο αυτό προστίθεται τεχνητά ένας κόμβος  $t$ , ο οποίος συνδέεται με κάθε κόμβο του  $G$  με μη κατευθυνόμενη ακμή βάρους  $\alpha$ . Έπειτα η κοινότητα  $S$  ορίζεται όπως και

πρροηγουμένως σε σχέση με τον τεχνητό κόμβο  $t$ . Είναι σημαντικό, στο σημείο αυτό να διατυπωθεί το σημαντικό Θεώρημα 6.1.

**Θεώρημα 6.1.** Έστω  $G(V, E)$  ένα μη κατευθυνόμενο γράφημα,  $s \in V$  ένας κόμβος πηγή,  $t$  ένας τεχνητός κόμβος με ακμές βάρους  $\alpha$  προς όλους τους κόμβους του  $G$ . Έστω  $S$  η κοινότητα του  $s$  σε σχέση με τον κόμβο  $t$ . Τότε για κάθε μη κενά υποσύνολα  $P$  και  $Q$ , για τα οποία ισχύει  $P \cup Q = S$  και  $P \cap Q = \emptyset$ , τα ακόλουθα όρια διατηρούνται πάντα:

$$\frac{c(S, V - S)}{|V - S|} \leq \alpha \leq \frac{c(P, Q)}{\min(|P|, |Q|)}$$

Σύμφωνα με το Θεώρημα 6.1, η παράμετρος  $\alpha$  χρησιμοποιείται ως το άνω φράγμα στη συνδεσιμότητα μεταξύ των κοινοτήτων  $S$  και  $V - S$  καθώς και ως κάτω φράγμα στην εσωτερική συνδεσιμότητα της κοινότητας  $S$ .

**Επέκταση (expansion):** η επέκταση ενός υποσυνόλου  $S$  του  $V$  ορίζεται ως η ελάχιστη τιμή του  $\Phi(S) = \frac{c(P, Q)}{\min(|P|, |Q|)}$ , για όλα τις τομές  $(P, Q)$  του  $S$ . Επιπλέον, η κοινότητα του  $G$  σε σχέση με τον τεχνητό κόμβο  $t$  θα έχει επέκταση τουλάχιστον  $\alpha$ . Το αριστερό μέρος της ανισότητας φράσσει την συνδεσιμότητα μεταξύ των κοινοτήτων και επιπλέον εγγυάται ότι οι κοινότητες μεταξύ τους θα είναι αποσυνδεδεμένες. Επίσης το  $\alpha$  μπορεί να χρησιμοποιηθεί για τον έλεγχο των δύο κριτηρίων. Η επιλογή του  $\alpha$  και το πως αυτό επηρεάζει την κοινότητα αναφέρεται στην Ενότητα 6.2. Το Θεώρημα 6.1 αποδεικνύεται με μια σειρά από λήμματα. Σημαντικό είναι, στο σημείο αυτό να ορίσουμε το γράφημα  $G_\alpha$  ως το επεκτεταμένο γράφημα του  $G$ , μετά την προσθήκη του τεχνητού κόμβου  $t$  ο οποίος συνδέεται με όλους τους κόμβους του γραφήματος με ακμές βάρους  $\alpha$ .

**Λήμμα 6.2.** Έστω  $s$  και  $t \in V$  δύο κόμβοι του γραφήματος  $G$  και  $S$  η κοινότητα του  $s$  σε σχέση με τον κόμβο  $t$ . Τότε, υπάρχει κάποιο δέντρο ελάχιστης τομής (min-cut tree)  $T_G$  του  $G$ , και μια ακμή  $ab \in T_G$ , ώστε η αφαίρεση της ακμής  $ab$  να οδηγεί στην διαμέριση  $S$  και  $V - S$ .

**Λήμμα 6.3.** Έστω  $T_G$  το δέντρο ελάχιστης τομής του  $G(V, E)$ , και  $uw$  μια ακμή του  $T_G$ . Η ακμή  $uw$  οδηγεί στην τομή  $(U, W)$  στο γράφημα  $G$ , όπου το  $u \in U$ ,  $w \in W$ . Για κάθε τομή  $(U1, U2)$  του  $U$ , όπου  $U1, U2$  μη κενά σύνολα,  $u \in U1$ ,  $U1 \cup U2 = U$ , και  $U1 \cap U2 = \emptyset$ . Τότε,

$$c(W, U2) \leq c(U1, U2)$$

Το Λήμμα 6.3 αποδεικνύεται ως εξής, δεδομένου ότι η ακμή  $uw$  είναι μια ακμή του  $T_G$ , τότε αυτή προσδιορίζει την ελάχιστη τομή  $(U, W)$  ανάμεσα στους κόμβους  $u$  και  $w$

του γραφήματος  $G$ . Επιπλέον έστω η τομή  $(U1, W \cup U2)$ . Δεδομένου ότι  $u \in U1$  και  $w \in W$ , η τομή  $(U1, W \cup U2)$  είναι επίσης τομή ανάμεσα στους κόμβους  $u$  και  $w$ , αλλά όχι απαραίτητα η ελάχιστη. Επιπλέον,

$$\begin{aligned} c(U, W) &\leq c(U1, W \cup U2), \\ c(U1 \cup U2, W) &\leq c(U1, W \cup U2), \\ c(U1, W) + c(U2, W) &\leq c(U1, W) + c(U1, U2), \\ c(U2, W) &\leq c(U1, U2) \end{aligned}$$

**Λήμμα 6.4.** Έστω  $G_\alpha$  το γράφημα επέκτασης του  $G$  και  $S$  η κοινότητα του  $s$  σε σχέση με τον τεχνητό κόμβο  $t$ . Για κάθε μη κενά σύνολα  $P$  και  $Q$ , τέτοια ώστε  $P \cup Q = S$  και  $P \cap Q = \emptyset$ , τότε ισχύει η πιο κάτω σχέση,

$$\alpha \leq \frac{c(P, Q)}{\min(|P|, |Q|)}$$

Έστω  $T_{G_\alpha}$  το ελάχιστο δέντρο τομής του  $G_\alpha$ , και  $s't'$  μια ακμή του δέντρου  $T_{G_\alpha}$  η αφαίρεση της οποίας οδηγεί στην τομή  $S$  και  $V - S \cup \{t\}$  στο γράφημα  $G_\alpha$ . Σύμφωνα με το Λήμμα 6.4, ένα δέντρο  $T_{G_\alpha}$  υπάρχει, και με βάση τον ορισμό του δέντρου ελάχιστης τομής, η ακμή  $s't'$  η οποία βρίσκεται στο μονοπάτι που συνδέει τους κόμβους  $s$  και  $t$  στο  $T_{G_\alpha}$ . Έπειτα, το σύνολο  $S$  διαχωρίζεται σε δύο υποσύνολα  $P$  και  $Q$  τέτοια ώστε ο κόμβος  $s' \in P$ . Τότε εφαρμόζοντας το Λήμμα 6.3,

$$\begin{aligned} c(V - S \cup \{t\}, Q) &\leq c(P, Q), \\ c(V - S, Q) + c(\{t\}, Q) &\leq c(P, Q), \\ c(\{t\}, Q) &\leq c(P, Q), \\ \alpha|Q| &\leq c(P, Q), \\ \alpha \cdot \min(|P|, |Q|) &\leq c(P, Q) \end{aligned}$$

Η αντικατάσταση της ποσότητας  $c(\{t\}, Q)$  με την ποσότητα  $\alpha|Q|$  είναι σωστή γιατί ο κόμβος  $t$  συνδέεται με κάθε κόμβο του συνόλου  $V$  με ακμή βάρους  $\alpha$ .

**Λήμμα 6.5.** Έστω  $G_\alpha$  το γράφημα επέκτασης του  $G(V, E)$  και  $S$  η κοινότητα του  $s$  σε σχέση με τον τεχνητό κόμβο  $t$ . Τότε η ακόλουθη ανισότητα ισχύει πάντα:

$$\frac{c(S, V - S)}{|V - S|} \leq \alpha$$

Η διαδικασία απόδειξης του πιο πάνω Λήμματος είναι παρόμοια με την προηγούμενη απόδειξη. Έστω ότι  $T_{G_\alpha}$  το ελάχιστο δέντρο τομής του  $G_\alpha$ , το οποίο περιέχει την ακμή  $s't'$ , η αφαίρεση της οποίας οδηγεί στη τομή  $S$  και  $V - S \cup \{t\}$ . Με βάση τον ορισμό του

δέντρου ελάχιστης τομής,  $(s', t')$  αντιστοιχεί στην ελάχιστη τομή ανάμεσα στους κόμβους  $s'$  και  $t'$  στο γράφημα  $G_\alpha$ . Η τομή αυτής της ελάχιστης τομής είναι ίση με  $c(S, V - S \cup \{t\})$ . Μια τομή  $(V, \{t\})$  στο γράφημα  $G_\alpha$ , διαχωρίζει επίσης τους κόμβους  $s$  και  $t$  αλλά δεν είναι απαραίτητα και μια ελάχιστη τομή μεταξύ των κόμβων αυτών. Αυτό μπορεί να γραφεί ως:

$$\begin{aligned} c(S, V - S \cup \{t\}) &\leq c(V, \{t\}), \\ c(S, V - S) + c(S, \{t\}) &\leq c(V - S, \{t\}) + c(S, \{t\}), \\ c(S, V - S) &\leq c(V - S, \{t\}), \\ c(S, V - S) &\leq \alpha|V - S| \end{aligned}$$

Η τελευταία ανισότητα δίνει το Λήμμα 6.5 και οφείλεται στο γεγονός ότι ο κόμβος  $t$  συνδέεται με όλους τους κόμβους του γραφήματος με ακμή βάρους  $\alpha$ . Τα Λήμματα 6.4 και 6.5 αποδεικνύουν το Θεώρημα 6.1.

Το Θεώρημα 6.1 μπορεί να χρησιμοποιηθεί είτε στην εύρεση των υψηλής ποιότητας κοινοτήτων του  $G$ , ή στην ανάπτυξη ενός γενικού αλγορίθμου ομαδοποίησης ο οποίος βασίζεται στο δέντρο ελάχιστης τομής του γραφήματος  $G$ . Η ιδέα του Αλγόριθμου είναι η επέκταση του γραφήματος  $G$  στο  $G_\alpha$ , εύρεση του δέντρου ελάχιστης τομής  $T_{G_\alpha}$ , αφαίρεση του τεχνητού κόμβου  $t$  από το δέντρο ελάχιστης τομής και οι συνεκτικές συνιστώσες που προκύπτουν από την αφαίρεση αυτή αποτελούν και τις ομάδες του  $G$ .



CUTCLUSTERINGALGORITHM( $G(V,E), \alpha$ )  
**Έστω**  $V' = V \cup t$   
**Για** όλους τους κόμβους  $u \in V$   
 Συνέδεσε τον  $t$  με τον  $u$  με ακμή βάρους  $\alpha$   
**Έστω**  $G'(V', E')$  το γράφημα επέκταση του  $G$  μετά την προσθήκη του τεχνητού κόμβου  $t$   
 Υπολόγισε το δέντρο ελάχιστης αποκοπής  $T'$  του  $G'$   
 Αφαίρεσε τον κόμβο  $t$  από το  $T'$   
**Επέστρεψε** όλες τις συνεκτικές συνιστώσες που προκύπτουν ως τις ομάδες του  $G$

### Αλγόριθμος 11: Αλγόριθμος Cut-clustering

## 6.2 Επιλογή $\alpha$

Η επιλογή του  $\alpha$  στο επεκτεταμένο γράφημα  $G_\alpha$ , παίζει σημαντικό ρόλο στην ποιότητα των ομάδων που παράγονται.

Όσο το  $\alpha$  τείνει στο 0, η ελάχιστη τομή ανάμεσα στους κόμβους  $t$  και κάθε άλλο κόμβο  $u$  του  $G$  θα είναι η τετριμμένη τομή  $(t, V)$ , η οποία απομονώνει τον κόμβο  $t$  από τους υπόλοιπους κόμβους του γραφήματος. Επομένως το αποτέλεσμα του αλγόριθμου στην περίπτωση αυτή θα ήταν μια μοναδική ομάδα, το ίδιο το γράφημα  $G$ , σε περίπτωση που αυτό είναι συνδεδεμένο. Από την άλλη πλευρά, αν το  $\alpha$  τείνει στο άπειρο, τότε το δέντρο ελάχιστης τομής  $T_{G_\alpha}$  που προκύπτει από το γράφημα επέκτασης  $G_\alpha$ , έχει ρίζα τον κόμβο  $t$  και παιδιά του όλοι οι κόμβοι του γραφήματος. Συνεπώς μετά την αφαίρεση του κόμβου  $t$  από το  $T_{G_\alpha}$ , παράγονται  $n$  τετριμμένες ομάδες, όσες και οι κόμβοι του γραφήματος.

Για τιμές του  $\alpha$  ανάμεσα στις δύο αυτές ακραίες τιμές του  $\alpha$ , οι ομάδες που θα προκύψουν θα είναι από 1 μέχρι και  $n$ . Η ακριβής τιμή του  $\alpha$  εξαρτάται από τη δομή του γραφήματος  $G$  και την κατανομή των βαρών στις ακμές του. Το σημαντικό, είναι ότι όσο η τιμή του  $\alpha$  αυξάνεται ο αριθμός των ομάδων δεν φθίνει.

## 6.3 Ευριστική Μέθοδος για τον Αλγόριθμο Cut-clustering

Ο συνολικός χρόνος του βασικού Αλγόριθμου 11 Cut-clustering είναι ίσος με το χρόνο υπολογισμού του ελάχιστου δέντρου τομής, καθώς και τον ελάχιστο επιπλέον χρόνο που χρειάζεται για την εξαγωγή των υποδέντρων κάτω από τον κόμβο  $t$ . Αλλά ο υπολογισμός

του ελάχιστου δέντρου τομής είναι ισοδύναμος με  $n - 1$  υπολογισμούς των μέγιστων ροών, στην χειρότερη περίπτωση. Γι' αυτό τον λόγο χρησιμοποιείται μια ευριστική μέθοδος για γρήγορη εύρεσης ομάδων, συχνά σε χρόνο ανάλογο του συνολικού πλήθους των ομάδων.

**Λήμμα 6.6.** Έστω  $u_1, u_2 \in V$ , και  $S_1, S_2$  οι αντίστοιχες κοινότητες τους σε σχέση με τον κόμβο  $t$  στο γράφημα  $G_\alpha$ . Τότε είτε τα  $S_1, S_2$  είναι ξεχωριστά υποσύνολα ή ένα από αυτά είναι υποσύνολο του άλλου.

Αν τα σύνολα  $S_1$  και  $S_2$  επικαλύπτονται χωρίς το ένα να περιέχει το άλλο, τότε είτε το  $S_1 \cap S_2$  ή  $S_1 - S_2$  αποτελεί μικρότερη κοινότητα για τον κόμβο  $u_1$ , ή συμμετρικά, είτε το  $S_1 \cap S_2$  ή  $S_2 - S_1$  αποτελεί μικρότερη κοινότητα για τον κόμβο  $u_2$ . Γι' αυτό τα σύνολα  $S_1$  και  $S_2$  είναι είτε ξεχωριστά σύνολα ή το ένα από αυτά είναι υποσύνολο του άλλου. Μια πιο κοντινή ματιά στον βασικό αλγόριθμο cut-clustering δείχνει ότι αρκεί η εύρεση των ελάχιστων τομών που αντιστοιχούν σε ακμές προς το  $t$  στο δέντρο ελάχιστης τομής  $T_{G_\alpha}$ . Γι' αυτό, αντί να υπολογίζεται ολόκληρο το δέντρο ελάχιστης τομής του γραφήματος  $G_\alpha$  χρησιμοποιείται το Λήμμα 6.6 με στόχο την μείωση των υπολογισμών ελάχιστης τομής που απαιτούνται. Αν η τομή ανάμεσα σε δύο κόμβους  $u$  και  $t$  οδηγεί στην κοινότητα  $S$ , τότε δε χρησιμοποιείται κανένας άλλος κόμβος της κοινότητας  $S$  ως κόμβος πηγή για υπολογισμό ελάχιστης τομής με τον κόμβο  $t$ , επειδή σύμφωνα με το προηγούμενο λήμμα δε μπορούμε να δημιουργήσουμε μικρότερες κοινότητες. Αντίθετα οι κόμβοι αυτοί μαρκάρονται ως κόμβοι της κοινότητας  $S$  και αν αργότερα η κοινότητα  $S$  γίνει μέρος μιας μεγαλύτερης κοινότητας  $S'$ , τότε οι κόμβοι της κοινότητας  $S$  μαρκάρονται ως κόμβοι της κοινότητας  $S'$ . Προφανώς ο αλγόριθμος αυτός βρίσκει τις μεγαλύτερες κοινότητες ανάμεσα στον κόμβο  $t$  και τους κόμβους  $V$ .

Η ευριστική που χρησιμοποιήθηκε εξαρτάται από την επιλογή του επόμενου κόμβου πηγή (για τον οποίο θα βρούμε την ελάχιστη τιμή του με τον κόμβο  $t$ ). Αν επιλεγεί κόμβος πηγή που θα έχει ως αποτέλεσμα την παραγωγή μεγαλύτερης ομάδας, τόσο λιγότεροι υπολογισμοί ελάχιστης τομής θα χρειαστούν. Πριν από κάθε υπολογισμό ελάχιστης τομής, οι κόμβοι ταξινομούνται σύμφωνα με το αθροιστικό βάρος των ακμών που προσπίπτουν σε αυτούς κατά φθίνουσα σειρά. Κάθε φορά, υπολογίζεται η ελάχιστη τομή ανάμεσα στον επόμενο μη σημαδεμένο κόμβο και τον  $t$ . Η διαδικασία που περιγράφηκε μειώνει τους υπολογισμούς μέγιστων ροών περίπου στο πλήθος των ομάδων του  $G$ , γεγονός που αυξάνει την ταχύτητα του αλγορίθμου σημαντικά.

## 6.4 Ιεραρχικός Αλγόριθμος Cut-clustering

Στην ενότητα 6.1, παρουσιάστηκε ένας αλγόριθμος ομαδοποίησης των κόμβων του  $G$  δοθέντος μιας τιμής  $\alpha$ . Μόλις ολοκληρωθεί η διαδικασία ομαδοποίησης, κάθε ομάδα συρρικνώνεται σε ένα μοναδικό κόμβο και ο αλγόριθμος επαναλαμβάνεται στο γράφημα που προκύπτει μετά τη συρρίκνωση. Για τη συρρίκνωση ενός συνόλου κόμβων, αυτοί αντικαθίστανται με ένα μοναδικό καινούριο κόμβο, πιθανοί βρόγχοι διαγράφονται και παράλληλες ακμές αντικαθίστανται με μια μοναδική ακμή βάρους ίσου με το άθροισμα των βαρών τους.

Έπειτα εφαρμόζοντας τον βασικό αλγόριθμο που περιγράφηκε στην Ενότητα 6.1 στο γράφημα που προέκυψε μετά τη συρρίκνωση, απλά επιλέγεται ένα καινούριο  $\alpha$ , και η νέα ομαδοποίηση, αντιστοιχεί σε ομαδοποίηση των ομάδων του  $G$  που προέκυψαν στην προηγούμενη επανάληψη. Επαναλαμβάνοντας τη διαδικασία αυτή για κάποιο αριθμό φορών, κτίζεται μια ιεραρχία από ομάδες. Το καινούριο  $\alpha$  που επιλέγεται έχει μικρότερη τιμή από αυτή του προηγούμενου βήματος, διαφορετικά η νέα ομαδοποίηση θα ήταν ίδια με την προηγούμενη, γιατί κάθε συρρικνωμένος κόμβος στην προκειμένη περίπτωση θα αποτελέσει ξεχωριστή ομάδα.

Η ποιότητα των ομάδων σε κάθε επίπεδο της ιεραρχίας είναι ίδια με αυτή του αρχικού βασικού αλγορίθμου, η οποία εξαρτάται από την επιλογή του  $\alpha$ . Όμως το μέτρο επέκτασης δεν είναι πλέον πάνω σε όλους τους κόμβους του γραφήματος αλλά πάνω στις ομάδες που προέκυψαν. Ο επαναληπτικός αλγόριθμος σταματά είτε όταν το πλήθος των ομάδων που προκύπτουν είναι το επιθυμητό, είτε όταν το αποτέλεσμα του αλγορίθμου είναι μια μοναδική ομάδα η οποία περιέχει όλους τους κόμβους του γραφήματος.

Ο ιεραρχικός αλγόριθμος παρέχει ένα τρόπο εξέτασης του γραφήματος  $G$  σε πολλά επίπεδα. Στα χαμηλότερα επίπεδα το  $\alpha$  έχει μεγάλη τιμή και οι ομάδες είναι μικρές και πυκνές. Στα ψηλότερα επίπεδα, οι ομάδες είναι μεγαλύτερες σε μέγεθος και πιο αραιές. Επίσης οι ομάδες στα ψηλότερα επίπεδα είναι πάντα υπερσύνολα των ομάδων του χαμηλότερου επιπέδου. Αυτό δηλώνεται σαφώς από το ακόλουθο λήμμα.

**Λήμμα 6.7.** Έστω  $\alpha_1 > \alpha_2 > \dots > \alpha_{max}$  μια ακολουθία παραμετρικών τιμών οι οποίες συνδέουν το  $t$  με το  $V$  στο γράφημα  $G_{\alpha_i}$ . Έστω  $\alpha_{max+1} \leq \alpha_{max}$  αρκετά μικρό ώστε να οδηγήσει σε μοναδική ομάδα στο  $G$  και  $\alpha_0 \geq \alpha_1$  αρκετά μεγάλο ώστε να οδηγήσει κάθε κόμβος του γραφήματος να αποτελεί και μια ομάδα. Τότε όλες οι τιμές  $\alpha_{i+1}$ , για κάθε  $0 \leq i \leq max$ , οδηγούν σε ομάδες στο  $G$  οι οποίες είναι υπερσύνολα των ομάδων που δημιουργήθηκαν από κάθε  $\alpha_i$ , και όλες αυτές οι ομαδοποιήσεις σχηματίζουν ένα ιεραρχικό δέντρο κατά τις ομαδοποιήσεις του  $G$ .

```

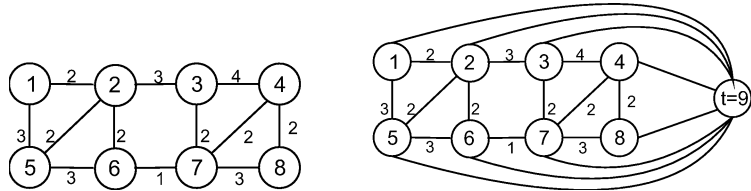
HIERARCHICALCUTCLUSTERINGALGORITHM(G(V,E))
Έστω  $G^0 = G$ 
Για ( $i = 0; ; i++$ )
    Θέσε νέα μικρότερη τιμή  $\alpha_i$ 
    Κάλεσε CutClusterigAlgorithm( $G^i, \alpha$ )
    Αν ((το πλήθος των ομάδων που προκύπτουν είναι το επιθυμητό) ή
        (το αποτέλεσμα του αλγορίθμου είναι μια μοναδική ομάδα
        η οποία περιέχει όλους τους κόμβους του γραφήματος))
        Διέκοψε
        Συρρίκνωσε τις ομάδες και δημιούργησε το γράφημα  $G^{i+1}$ 
Επέστρεψε όλες τις ομάδες σε όλα τα επίπεδα

```

Αλγόριθμος 12: Ιεραρχικός Αλγόριθμος Cut-clustering

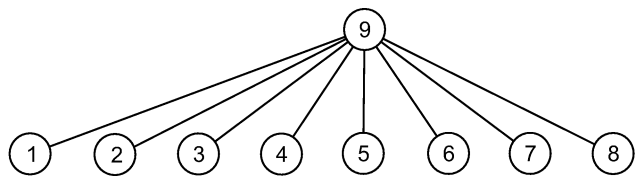
### 6.4.1 Παράδειγμα Ιεραρχικού Αλγορίθμου Cut-Clustering

Έστω το αρχικό γράφημα  $G(V, E)$ , όπου  $|V| = 8$  και  $|E| = 12$ . Στο γράφημα αυτό προστίθεται ένας τεχνητός κόμβος  $t$ , ο οποίος ενώνεται με όλους τους κόμβους στο  $V$  με ακμή βάρους  $\alpha$ . Στο Σχήμα 6.1 αριστερά φαίνεται το αρχικό γράφημα  $G$  και δίπλα το γράφημα  $G_\alpha$  που προέκυψε μετά την προσθήκη του τεχνητού κόμβου  $t$ .



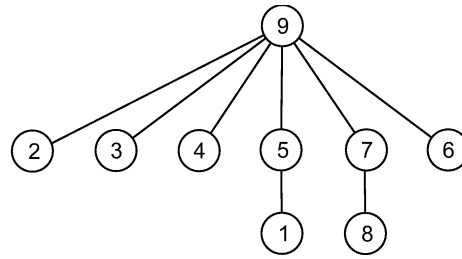
Σχήμα 6.1: Αρχικό γράφημα  $G$  και το αντίστοιχο  $G_\alpha$  μετά την προσθήκη του τεχνητού κόμβου  $t$ .

Στην συνέχεια του αλγορίθμου υπολογίζεται το δέντρο ελάχιστης τομής  $T_{G_\alpha}$  (για παράδειγμα με τον αλγόριθμο των Gomory-Hu). Το δέντρο ελάχιστης τομής διαφέρει ανάλογα με την επιλογή του  $\alpha$ . Για παράδειγμα το δέντρο ελάχιστης τομής που προκύπτει για  $3 \leq \alpha \leq \infty$  είναι αυτό που φαίνεται στο πιο κάτω Σχήμα 6.2.



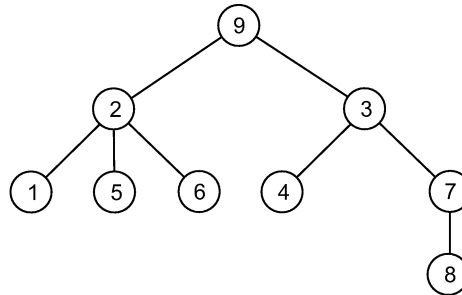
Σχήμα 6.2: Δέντρο Ελάχιστης Τομής  $T_{G_\alpha}$  για τιμές του  $\alpha$  στο διάστημα  $3 \leq \alpha \leq \infty$ .

Το δέντρο ελάχιστης τομής που προκύπτει για  $\alpha = 2$  είναι αυτό που φαίνεται στο πιο κάτω Σχήμα 6.3.



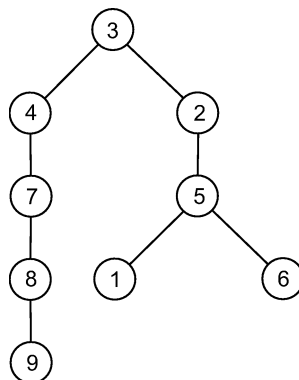
Σχήμα 6.3: Δέντρο Ελάχιστης Τομής  $T_{G_\alpha}$  για  $\alpha = 2$ .

Το δέντρο ελάχιστης τομής που προκύπτει για  $\alpha = 1$  είναι αυτό που φαίνεται στο πιο κάτω Σχήμα 6.4.



Σχήμα 6.4: Δέντρο Ελάχιστης Τομής  $T_{G_\alpha}$  για  $\alpha = 1$ .

Το δέντρο ελάχιστης τομής που προκύπτει για  $\alpha = 0$  είναι αυτό που φαίνεται στο πιο κάτω Σχήμα 6.5.



Σχήμα 6.5: Δέντρο Ελάχιστης Τομής  $T_{G_\alpha}$  για  $\alpha = 0$ .

Σε κάθε περίπτωση, η ομαδοποίηση που προκύπτει εξαρτάται άμεσα από την επιλογή της παραμέτρου  $\alpha$ . Πράγματι επιβεβαιώνεται ότι αν η τιμή του  $\alpha$  τείνει στο άπειρο, τότε

το αποτέλεσμα της ομαδοποίησης θα είναι  $|V|$  ομάδες όσο και το πλήθος των κόμβων του γραφήματος. Όπως παρατηρεί κανείς η αφαίρεση του κόμβου  $t$  από το δέντρο στο Σχήμα 6.2 θα έχει ως αποτέλεσμα  $|V|$  τετριμμένες ομάδες, κάθε μια από τις οποίες αποτελείται από ένα μοναδικό κόμβο του γραφήματος. Τα ελάχιστα δέντρα τομής που βασίζονται σε επέκταση γραφήματος, εξασφαλίζουν ένα τρόπο παραγωγής ποιοτικών ομάδων και εξαγωγής ισχυρών συνεκτικών συνιστωσών. Η μοναδική παράμετρος  $\alpha$ , μπορεί να χρησιμοποιηθεί ως ένα αυστηρό όριο στην τιμή επέκτασης της ομαδοποίησης ενώ ταυτόχρονα εξυπηρετεί και ως όριο του συνολικού βάρους μεταξύ των ομάδων που δημιουργούνται. Με τον τρόπο αυτό δηλαδή έχουν μετατρέψει ένα συνδυαστικό κριτήριο σε μια μοναδική παράμετρο.

Το πρόβλημα αναγώρισης κοινοτήτων είναι NP-complete πρόβλημα, για το λόγο αυτό επιλύουν το πρόβλημα περιορίζοντας τις κοινότητες που αναγνωρίζουν. Πιο συγκεκριμένα, ο αλγόριθμος τους αναγνωρίζει κοινότητες που ικανοποιούν ένα αυστηρό περιορισμό που επιβάλλεται από την παράμετρο  $\alpha$ . Ο αλγόριθμος δεν εγγυάται ούτε καν την εύρεση όλων των κοινοτήτων που ικανοποιούν τα όρια που θέτουν, αλλά μόνο την εύρεση των κοινοτήτων που είναι προφανή (είναι ορατά) μέσα στο δέντρο τομής του επεκτεταμένου γραφήματος. Γι' αυτό και η κύρια δύναμη του αλγόριθμου τους προκύπτει ουσιαστικά από την κύρια αδυναμία του (που περιγράφηκε πιο πάνω): αποφεύγει την υπολογιστική δυσκολία του αλγορίθμου (NP-completeness) εστιάζοντας στις αυστηρές απαιτήσεις που οδηγούν σε υψηλής ποιότητα ομαδοποίηση.

**Πλεονεκτήματα** Όλες οι παραλλαγές του βασικού Αλγορίθμου 11 είναι σχετικά γρήγορες (ιδιαίτερα αυτής που περιγράφηκε στην Ενότητα 6.3), η οποία είναι και εύκολη στην υλοποίηση και δίνει εύρωστα αποτελέσματα. Η ευκαμψία στην επιλογή της παραμέτρου  $\alpha$  και συνεπώς η δυνατότητα καθορισμού της ποιότητας των ομάδων που παράγονται είναι ακόμη ένα πλεονέκτημα του αλγορίθμου.

**Μειονεκτήματα** Ένα όμως μειονέκτημα του αλγορίθμου είναι το γεγονός ότι δεν μπορεί ο αλγόριθμος να παραμετροποιηθεί με βάση τον αριθμό και το μέγεθος των ομάδων. Δηλαδή η ομάδες που προκύπτουν είναι φυσικό αποτέλεσμα του αλγορίθμου και δεν μπορούν να οριστούν εξ' αρχής ως επιθυμητές, εκτός και αν αυτές αναζητηθούν μέσω μιας σειράς επαναλήψεων. Επίσης ο αλγόριθμος δεν επιτρέπει ένα αποτέλεσμα ομαδοποίησης κατά το οποίο οι ομάδες να επικαλύπτονται (overlap). Πρόβλημα το οποίο εμφανίζεται σε δεδομένα στα οποία ένας κόμβος μπορεί να ανήκει σε περισσότερες από μία κατηγορίες. Αυτό όμως το ζήτημα αποτελεί γενικά μειονέκτημα όλων των αλγορίθμων ομαδοποίησης που παράγουν

ισχυρές ή μη συνδεδεμένες ομάδες.

# ΚΕΦΑΛΑΙΟ 7

## ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

---

### 7.1 Περιγραφή πειραμάτων

### 7.2 Πειράματα

---

### 7.1 Περιγραφή πειραμάτων

Υλοποιήσαμε έναν ιεραρχικό αλγόριθμο κατάτμησης εικόνας HSW, χρησιμοποιώντας τη μέθοδο SW (μέθοδος εύρεσης ελάχιστης τομής). Τροποποιήσαμε τον αλγόριθμο ελάχιστης τομής SW, ώστε το κριτήριο διαμέρισης να μην είναι η ελάχιστη τομή αλλά το κριτήριο *MyCut* που περιγράφεται από την πιο κάτω εξίσωση:

$$MyCut(A, B) = \frac{SCut(A, B)}{(|C| + 1) \cdot \min\{|A|, |B|\}}.$$

Εφ' εξής θα αναφερόμαστε στον τροποποιημένο ιεραρχικό αλγόριθμο που προτείνουμε ως ModHSW.

Υλοποιήσαμε επίσης τον βασικό αλγόριθμο ομαδοποίησης του [27], ο οποίος είναι μη ιεραρχικός και στον οποίο βασίζεται μια ιεραρχική εφαρμογή που προτείνουν (η οποία δίνει καλύτερη πολυπλοκότητα χρόνου σε μεγάλα γραφήματα), η εφαρμογή είναι πρακτική σε γραφήματα με περιορισμένο πλήθος κόμβων. Η βασική του ιδέα βασίζεται σε δέντρα ελάχιστης τομής (Minimum Cut Trees) τα οποία υπολογίζονται με τον γνωστό αλγόριθμο των Gomory-Hu. Ο αλγόριθμος των Gomory-Hu δεν είναι αποδοτικός ως αναφορά το χρόνο σε μεγάλο μεγέθους γραφήματα εξαιτίας ότι περιλαμβάνει ένα πλήθος υπολογισμών μέγιστης



ροής, το οποίο αυξάνει γραμμικά με τον αριθμό των κόμβων του γραφήματος. Θα αναφερόμαστε σε αυτόν τον αλγόριθμο κατάτμησης ως MCT. Εξαιτίας ότι βασίζεται σε δέντρα ελάχιστης τομής, το κριτήριο διαμέρισης του αλγορίθμου MCT, είναι το SCut (κριτήριο διαμέρισης με βάση την ελάχιστη τομή).

Τέλος υλοποιήσαμε και ένα τρίτο αλγόριθμο [29], ο οποίος είναι αλγόριθμος ομαδοποίησης και βασίζεται σε δέντρα ελάχιστης τομής. Ο συγκεκριμένος έχει εφαρμοστεί σε κοινότητες διαδικτύου (web communities). Παρόλο που βασίζεται σε δέντρα ελάχιστης τομής, διαφοροποιείται από τον προηγούμενο αλγόριθμο στο κριτήριο διαμέρισης το οποίο δεν είναι η ελάχιστη τομή, αλλά ένα συνδυαστικό κριτήριο (bicriterion) που ορίζεται με τη βοήθεια μιας παραμέτρου  $\alpha$ . Θα αναφερόμαστε σε αυτόν τον αλγόριθμο κατάτμησης ως MCT- $\alpha$ .

Υλοποιήσαμε και τους τρεις αλγόριθμους έγινε σε LEDA και τους εφαρμόσαμε στο πρόβλημα κατάτμησης εικόνας με σκοπό να συγκρίνουμε τα αποτελέσματα τους και να παρατηρήσουμε τη συμπεριφορά τους κάτω από διάφορες συνθήκες. Σε κάθε περίπτωση, χρησιμοποιείται ένα στάδιο προεπεξεργασίας.

**Στάδιο Προεπεξεργασίας:** Διαβάζουμε το δοθέν αρχείο εικόνας, το οποίο μπορεί να είναι οποιασδήποτε μορφής (bmp, jpg κ.α), μέσω της Matlab. Μια εικόνα περιγράφεται από  $N \times M$  εικονοστοιχεία. Κάθε εικονοστοιχείο έχει μια τιμή, η οποία εκφράζει τη φωτεινότητα του (επίπεδο του γκρι). Στην εργασία αυτή θεωρούμε επίπεδα που κωδικοποιούνται με 8 bits, δηλαδή μπορούμε να έχουμε 256 διαφορετικές τιμές φωτεινότητας (0 – 255) για κάθε εικονοστοιχείο. Όπου το 0 είναι το μαύρο και το 255 το λευκό, αντίστοιχα. Από τα αρχεία εικόνων εξάγονται στοιχειώδη χαρακτηριστικά (φωτεινότητες των στοιχείων της). Έπειτα, δημιουργείται ένα αρχείο περιγραφής των στοιχείων της εικόνας με χαρακτήρες ASCII, το οποίο αποτελεί και την είσοδο σε έναν από τους 3 αλγορίθμους κατάτμησης που προαναφέραμε για την επίλυση του προβλήματος κατάτμησης εικόνας.

Στη συνέχεια ο αλγόριθμος κατάτμησης που χρησιμοποιείται, διαβάζει το αρχείο εισόδου και αναπαριστά την εικόνα σε ένα μη κατευθυνόμενο γράφημα  $G$ . Πιο αναλυτικά, η αναπαράσταση μιας εικόνας σε ένα γράφημα γίνεται ως εξής: Κάθε κόμβος του γραφήματος αντιστοιχεί σε ένα στοιχείο της εικόνας (pixel) και υπάρχει ακμή  $e$  με άκρα τους κόμβους  $v_1, v_2$  αν και μόνο αν τα στοιχεία  $p_1$  και  $p_2$  που αντιστοιχούν στους κόμβους  $v_1$  και  $v_2$  είναι γειτονικά (είτε σε οριζόντια είτε σε κατακόρυφη κατεύθυνση). Δηλαδή σε κάθε περίπτωση το γράφημα  $G$  ανήκει στην κλάση των πλεγμάτων (Grid).

Σε κάθε αλγόριθμο, το αποτέλεσμα της κατάτμησης αποτελείται από ένα σύνολο ανεξάρτητων υπογραφημάτων. Πιο συγκεκριμένα, οι κόμβοι που ανήκουν σε ένα υπογράφημα

αντιστοιχούν στα εικονοστοιχεία που ανήκουν σε μια ομάδα. Για την αναπαράσταση του αποτελέσματος, υπολογίζουμε τη μέση τιμή των φωτεινότητων των εικονοστοιχείων κάθε ομάδας και θεωρούμε πως όλα τα εικονοστοιχεία που ανήκουν στην ίδια ομάδα, μπορούν να περιγραφούν από τη μέση τιμή τους.

Η έξοδος των αλγορίθμων είναι ένα αρχείο περιγραφής των στοιχείων της τεμαχισμένης εικόνας με χαρακτήρες ASCII, στο οποίο σε κάθε εικονοστοιχείο αντιστοιχεί τιμή φωτεινότητας ίση με τη μέση τιμή της ομάδας στην οποία ανήκει. Τέλος, το αρχείο αυτό το διαβάζουμε μέσω της Matlab για να δούμε οπτικά τα αποτελέσματα της κατάτμησης.

Πίνακας 7.1: Περιγραφή Συμβόλων

| Σύμβολο      | Περιγραφή  |
|--------------|--|
| $M$          | το πλήθος των ομάδων που προκύπτουν μετά το αποτέλεσμα της κατάτμησης εικόνας  |
| $MyCut$      | το κριτήριο διαμέρισης που προτείνουμε   |
| $SCut$       | το κριτήριο διμέρισης ελάχιστης τομής  |
| $SCap$       | εχθετικά βάρη που φθίνουν απότομα στο μηδέν με παράμετρο $\sigma$ θεωρούμε ότι έχει τιμή $\sigma = 10$ στα περισσότερα πειράματά μας |
| $MyCap$      | εχθετικά βάρη που προτείνουμε αλλά δε φθίνουν απότομα στο μηδέν  |
| $HSW$        | ιεραρχικός αλγόριθμος κατάτμησης χρησιμοποιώντας τη μέθοδο SW και επομένως το κριτήριο SCut  |
| $ModHSW$     | ιεραρχικός αλγόριθμος κατάτμησης χρησιμοποιώντας τη μέθοδο ModSW και επομένως το κριτήριο διαμέρισης MyCut                           |
| $MCT$        | αλγόριθμος κατάτμησης [27] που χρησιμοποιεί δέντρα ελάχιστης τομής και επομένως το κριτήριο διαμέρισης SCut                          |
| $MCT-\alpha$ | αλγόριθμος κατάτμησης [29] που χρησιμοποιεί δέντρα ελάχιστης τομής, χρησιμοποιώντας συνδυαστικό κριτήριο διαμέρισης $\alpha$         |

Στην συνέχεια παρουσιάζουμε διάφορες εικόνες και τα αποτελέσματα των αλγορίθμων αυτών αναλύοντας ταυτόχρονα τη συμπεριφορά τους.

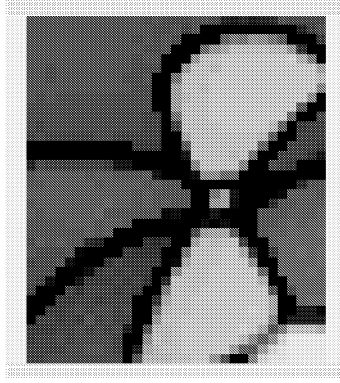
## 7.2 Πειράματα

**Πείραμα 1:** Στο πείραμα αυτό θεωρούμε μια εικόνα μικρού μεγέθους η οποία αποτελείται από περιοχές οι οποίες χωρίζονται με ευδιάκριτες ακμές. Η εικόνα περιέχει περιοχές με ίδια τιμή μέσης φωτεινότητας. Αναφέρουμε εδώ ότι αν σε μια τέτοια εικόνα χρησιμοποιούσαμε τον κλασικό αλγόριθμο  $K$ -μέσων τότε το αποτέλεσμα της κατάτμησης (αν δεν υπήρχε πρόβλημα αρχικοποίησης) θα ήταν τρεις ομάδες. Πιο συγκεκριμένα στη μια ομάδα

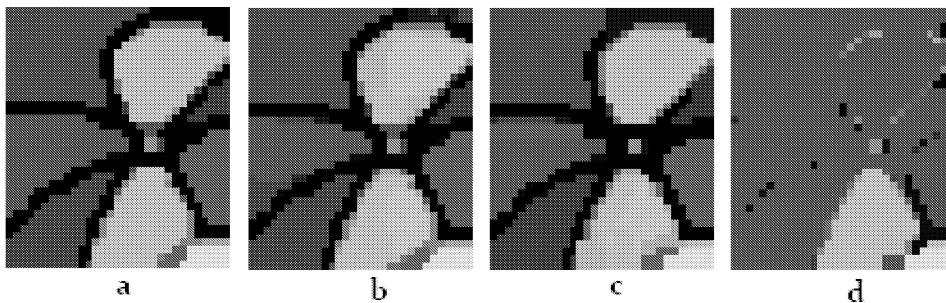
ανήκουν όλα τα εικονοστοιχεία που ανήκουν στις ακμές (όλες οι ακμές έχουν μέση τιμή 0), στη δεύτερη θα ανήκουν τόσο τα εικονοστοιχεία του φόντου (background) όσο και τα εικονοστοιχεία του φιόγκου που ανήκουν σε περιοχές με μέση φωτεινότητα ίση με τη μέση φωτεινότητα του φόντου, ενώ στην τρίτη ομάδα ανήκουν όλα τα υπόλοιπα εικονοστοιχεία της εικόνας. Ένα πλεονέκτημα των μεθόδων που χρησιμοποιούν τομές γραφημάτων είναι ότι λαμβάνουν υπόψη και την χωρική τοπολογία των εικονοστοιχείων (βέβαια και σε άλλες μεθόδους η χωρική τοπολογία μπορεί να ληφθεί υπόψη και με κατάλληλο ορισμό των βαρών, αλλά αυτό δε θα μας αποσχολήσει στην παρουσίαση αυτή). Συγκεκριμένα στην εικόνα αυτή υπάρχουν 5 περιοχές με παρόμοιες αποχρώσεις. Πράγματι με την εφαρμογή και των τριών αλγορίθμων παρατηρούμε ότι το αποτέλεσμα της κατάτμησης δεν πρόκειται να δώσει ομάδα στην οποία να περιέχονται τόσο εικονοστοιχεία του φόντου όσο και εικονοστοιχεία του φιόγκου (γενικά και οι πέντε περιοχές θα ανήκουν σε άλλη ομάδα του αποτελέσματος), παρά το γεγονός ότι έχουν ίδια φωτεινότητα. Στο συγκεκριμένο παράδειγμα αυτό είναι εφικτό γιατί, για παράδειγμα, η περιοχή του φιόγκου και του φόντου χωρίζονται με μια ευδιάκριτη ακμή και επομένως οι περιοχές αυτές δε θεωρούνται γειτονικές (αφού το γράφημα είναι Grid).

Εκτελούμε τον αλγόριθμο ModHSW-MyCap για  $M = 30$  και  $M = 60$ . Παρατηρούμε ότι τα αποτελέσματα του αλγόριθμου, χρησιμοποιώντας τον συνδυασμό των βαρών και του κριτηρίου διαμέρισης που εμείς ορίσαμε, είναι καλύτερα από τα αποτελέσματα που λαμβάνουμε εκτελώντας τον αλγόριθμο HSW-SCap δηλαδή με κριτήριο διαμέρισης SCut αλλά ορίζοντας τα βάρη με τέτοιο τρόπο ώστε να τείνουν απότομα στο μηδέν SCap (παρόμοια προσέγγιση με τον αλγόριθμο [27] (MCT), στον οποίο το κριτήριο διαμέρισης είναι το SCut και τα βάρη ακμών φθίνουν απότομα στο μηδέν). Από την άλλη πλευρά βέβαια, τα αποτελέσματα που μας δίνει ο αλγόριθμος HSW-MyCap είναι χειρότερο από το αποτέλεσμα του HSW-SCap γιατί δίνει πολύ περισσότερες τετριμμένες ομάδες ή ομάδες με λιγότερο από 5 στοιχεία.

Στη συνέχεια εκτελούμε τον αλγόριθμο MCT ορίζοντας (το κριτήριο του αλγορίθμου είναι το SCut) και  $M = 60$ . Παρατηρούμε ότι, αν ορίσουμε τα SCap βάρη το αποτέλεσμα του αλγορίθμου είναι καλύτερο σε σύγκριση με το αν ορίσουμε MyCap. Αυτό συμβαίνει γιατί το κριτήριο του MCT είναι η ελάχιστη τομή, γεγονός που όπως αναφέραμε και σε προηγούμενα κεφάλαια προμοδοτεί διαμερίσεις στις οποίες προκύπτουν μικρές ομάδες (είναι σύνηθες να προκύπτουν singletons). Έτσι όταν χρησιμοποιούμε βάρη τα οποία δεν τείνουν απότομα στο μηδέν προκύπτει τομή στην οποία συμμετέχουν όσο το δυνατό λιγότερες ακμές με μικρό βάρος. Αντίθετα, όταν η συνάρτηση βαρών φθίνει γρήγορα στο μηδέν ο



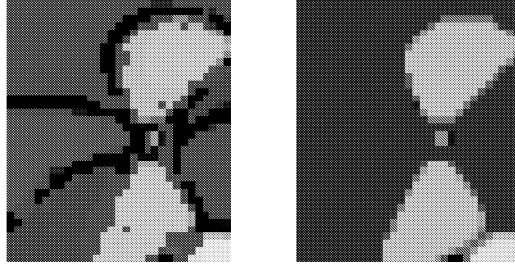
Σχήμα 7.1: Αρχική εικόνα.



Σχήμα 7.2: a) ModHSW-MyCap και  $M=30$ . b) ModHSW-MyCap και  $M=60$ . c) HSW-SCap και  $M=60$ , d) HSW-MyCap και  $M=60$

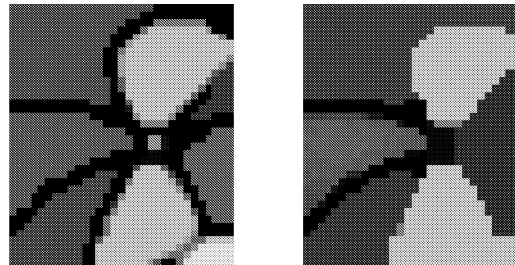
διαχωρισμός λίγων ακμών που συνδέουν εικονοστοιχεία με μικρές διαφορές φωτεινότητας γίνεται πιο ακριβός (ως προς το άθροισμα των βαρών των ακμών που προσπίπτουν σε αυτόν) σε σχέση με τον διαχωρισμό πολλών ακμών ανάμεσα σε εικονοστοιχεία με μεγάλες διαφορές φωτεινότητας. Παρόλα αυτά ακόμη και στην περίπτωση που χρησιμοποιούμε τα βάρη SCap προκύπτουν επίσης μικρές ομάδες, με τη διαφορά ότι είναι λιγότερες σε σύγκριση με το αποτέλεσμα των βαρών MyCap. Τέλος, συγκρίνοντας τα αποτελέσματα του MCT με αυτά του ModHSW αλγορίθμου, παρατηρούμε ότι στο δεύτερο τα χαρακτηριστικά της εικόνας που λαμβάνονται στις 60 επαναλήψεις (ομάδες) είναι περισσότερα σε σύγκριση με τον πρώτο.

Τέλος, εκτελούμε τον αλγόριθμο MCT-α στην ίδια εικόνα και επιλέγουμε εμπειρικά μια κατάλληλη τιμή για την παράμετρο  $\alpha$ . Παρατηρούμε ότι και αυτός ο αλγόριθμος εξαρτάται πολύ από τον καθορισμό των βαρών. Τα αποτελέσματα του αλγορίθμου είναι συγκρίσιμα με τον ιεραρχικό αλγόριθμο που προτείνουμε και καλύτερα από τον αλγόριθμο MCT. Το αποτέλεσμα αυτό είναι αναμενόμενο, δεδομένου ότι ο MCT διαμερίζει το γράφημα με το κριτήριο SCap χωρίς να λαμβάνει καθόλου υπόψη το μέγεθος των ομάδων που προκύπτουν,



Σχήμα 7.3: Αριστερά MCT-SCar και  $M=60$ . Δεξιά MCT-MyCar και  $M=60$ .

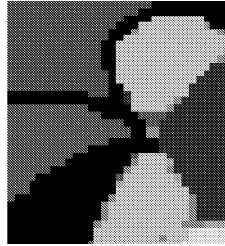
σε αντίθεση με τους άλλους δύο. Επαναλαμβάνουμε στο σημείο αυτό ότι ο αλγόριθμος MCT- $\alpha$ , επιλέγει μια διαμέριση κατά την οποία προκύπτουν ομάδες με εσωτερική συνδεσιμότητα μεγαλύτερη από τη τιμή  $\alpha$  και συνδεσιμότητα ανάμεσα στις ομάδες μικρότερη από τη παράμετρο  $\alpha$ .



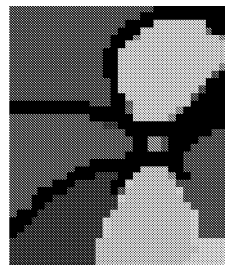
Σχήμα 7.4: Αριστερά MCT- $\alpha$ ,  $\alpha = 20$  και  $M=64$ . Δεξιά MCT- $\alpha$   $\alpha = 46$  και  $M=161$ .

**Πείραμα 2:** Στον αλγόριθμο ModSW με MyCut και MyCar εφαρμόσαμε μια επιπλέον τροποποίηση στον τρόπο με τον οποίο επιλέγουμε τη κατάλληλη διαμέριση. Ονομάζουμε αυτή τη τροποποίηση force τροποποίηση, στην οποία ορίζουμε ένα κατώφλι. Χρησιμοποιούμε το κατώφλι αυτό ως εξής: αν κατά τη διάρκεια κατασκευής της διάταξης συνδεσιμότητας εισαχθεί κόμβος  $v$  με συνδεσιμότητα μικρότερη ίση από ένα ορισμένο κατώφλι  $t = 0$  τότε όλοι οι κόμβοι που δεν έχουν ακόμη εισαχθεί στη διάταξη έχουν συνδεσιμότητα μηδέν με τους κόμβους που έχουν ήδη εισαχθεί στη διάταξη. Επομένως, στο σημείο αυτό αναγκάζουμε την συγκεκριμένη φάση του αλγορίθμου να τερματίσει και θέτουμε ως ελάχιστη τομή της φάσης, τη διαμέριση που προκύπτει με το ένα υποσύνολο να είναι όλοι οι κόμβοι που έχουν εισαχθεί στη διάταξη και στο άλλο υποσύνολο να είναι οι κόμβοι που παραμένουν στην ουρά προτεραιότητας, συμπεριλαμβανομένου και του κόμβου  $v$ . Στο σημείο αυτό απλά αναφέρουμε ότι σε περίπτωση που επιλύονται προβλήματα για τα οποία έχουμε κάποια γνώση, τότε μπορεί να οριστεί ένα κατάλληλο κατώφλι το οποίο να είναι πιο προσεγγιστικό. Ακόμη και στη περίπτωση που ορίσαμε αυστηρά ένα κατώφλι στην

τετριμμένη περίπτωση που τα υποσύνολα που προκύπτουν είναι εντελώς ξένα μεταξύ τους, παρατηρήσαμε ότι ο χρόνος εκτέλεσης βελτιώνεται. Αυτό ισχύει γιατί όταν βρεθεί ένας τέτοιος κόμβος που ικανοποιεί τη πιο πάνω συνθήκη, αν για παράδειγμα είμαστε στη φάση κατά την οποία υπολογίζεται η 2<sup>η</sup> υποψήφια τομή δε χρειάζεται να υπολογιστούν όλες οι υπόλοιπες  $n - 3$  υποψήφιες τομές, γεγονός που βελτιώνει σημαντικά το χρόνο εκτέλεσης του αλγορίθμου.



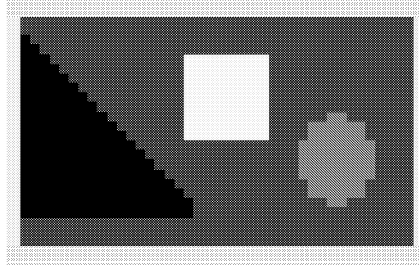
Σχήμα 7.5: ModHSW-MyCap,  $M=15$ .



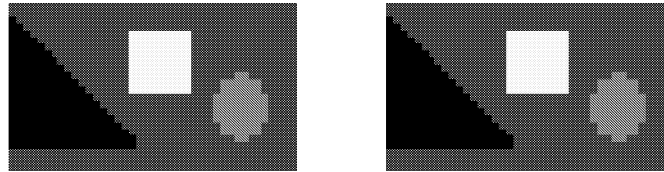
Σχήμα 7.6: ModHSW-MyCap,  $M=15$ . Ορίζοντας κατώφλι  $t = 0$  κατά τη διάρκεια της κατασκευής της διάταξης συνδεσιμότητας.

**Πείραμα 3:** Πήραμε μια εικόνα με 4 περιοχές σταθερής φωτεινότητας. Ο αλγόριθμος MCT με βάρη SCap δίνει το επιθυμητό αποτέλεσμα, δηλαδή τις 4 περιοχές μετά από ακριβώς 4 επαναλήψεις. Ορίζοντας, στον ίδιο αλγόριθμο, βάρη MyCap χρειάζονται 7 επαναλήψεις για να προκύψουν οι ομάδες. Συγκεκριμένα τρεις από τις ομάδες που προκύπτουν αποτελούν τετριμμένους κόμβους που ανήκουν στην περιοχή του τριγώνου (γωνίες τριγώνου) και οι άλλες 4 είναι οι 4 περιοχές που αναζητάμε. Πιο συγκεκριμένα το συνολικό βάρος ενός κόμβου στη γωνία του τριγώνου -με το τρόπο που ορίσαμε τα βάρη- είναι μικρότερο σε σχέση με το συνολικό βάρος της τομής που χωρίζει δύο περιοχές διαφορετικής φωτεινότητας.

Στο ίδιο παράδειγμα εκτελέσαμε και τον αλγόριθμο ModSW, τόσο με τα βάρη SCap όσο και με τα βάρη MyCap. Τα αποτελέσματα που προκύπτουν είναι τα ίδια και στις δύο



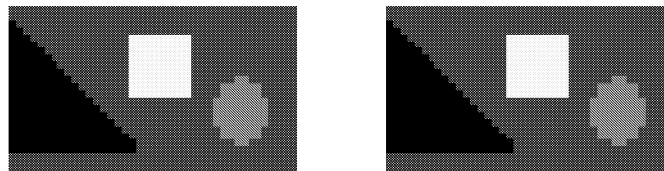
Σχήμα 7.7: Αρχική εικόνα.



Σχήμα 7.8: Αριστερά MCT-SCar και  $M=4$ . Δεξιά MCT-MyCar και  $M=7$ .

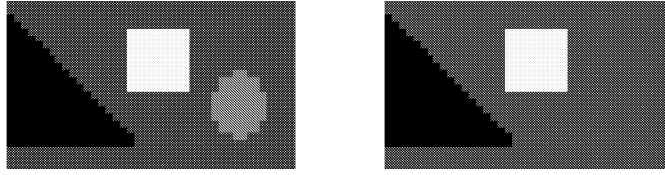
περιπτώσεις. Σε ακριβώς 4 επαναλήψεις λαμβάνονται οι τέσσερις περιοχές της εικόνας-παράδειγμα. Συμπεραίνουμε λοιπόν ότι, το αποτέλεσμα του αλγόριθμου που προτείνουμε δεν εξαρτάται από τα βάρη όπως και ο αλγόριθμος MCT.

Για το λόγο αυτό, στα πειράματα που θα ακολουθήσουν στην συνέχεια όταν εκτελούμε τον αλγόριθμο MCT, θα χρησιμοποιούμε τα βάρη SCar.



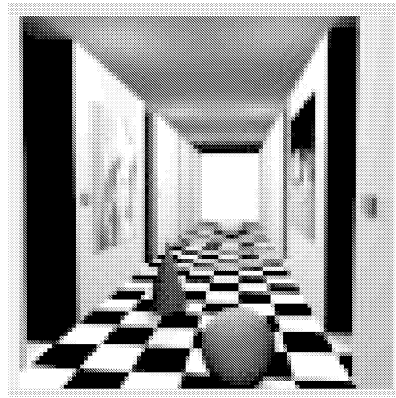
Σχήμα 7.9: Αριστερά ModHSW-SCar και  $M=4$ . Δεξιά MoHdSW-MyCar και  $M=4$ .

Το ίδιο ευαίσθητος στα βάρη είναι και ο αλγόριθμος MCT- $\alpha$ . Το αποτέλεσμα του, εξαρτάται επίσης και από τον καθορισμό του  $\alpha$ . Για παράδειγμα, αν ορίσουμε μια πολύ μεγάλη τομή στο  $\alpha$ , τέτοια ώστε να τείνει στο άπειρο, το αποτέλεσμα του MCT- $\alpha$  είναι τόσες ομάδες όσες και το πλήθος των εικονοστοιχείων της εικόνας. Για  $\alpha = 20$  λαμβάνουμε το επιθυμητό αποτέλεσμα, δηλαδή, τις 4 ομάδες σταθερής φωτεινότητας. Καθώς η τιμή του  $\alpha$ , μειώνεται, το αποτέλεσμα που προκύπτει αποτελείται από λιγότερες ομάδες οι οποίες αποτελούν συνένωση των ομάδων που προκύπτουν θέτωντας μεγαλύτερες τιμές στο  $\alpha$ . Στην περίπτωση που το  $\alpha = 0$ , τότε προκύπτει μια ομάδα που αποτελείται από όλους τους κόμβους του γραφήματος.



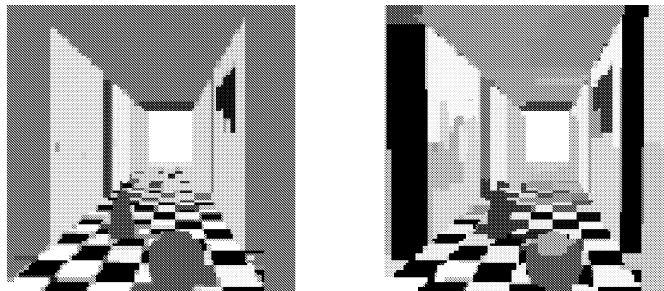
Σχήμα 7.10: Αριστερά MCT- $\alpha$ -SCap,  $\alpha = 20$  και  $M=4$ . Δεξιά MCT- $\alpha$ -SCap,  $\alpha = 1$  και  $M=3$ .

**Πείραμα 4:** Στο πείραμα εδώ θεωρούμε μια μικρή εικόνα ( $92 \times 92$ ), με πολλές περιοχές. Υπάρχουν πολλές περιοχές ίδιας φωτεινότητας. Κάθε μια από αυτές, με βάση την ιδιότητα που περιγράψαμε πιο πάνω των γραφοθεωρητικών μεθόδων σε συνδιασμό με το γεγονός ότι στην παρουσίαση αυτή αναπαριστούμε μια εικόνα με γράφημα πλέγμα, ανήκει και σε διαφορετική ομάδα στο αποτέλεσμα. Η εικόνα έχει ελάχιστο θόρυβο.



Σχήμα 7.11: Αρχική εικόνα.

Εκτελώντας τον αλγόριθμο MCT-SCap για 400 ομάδες παρατηρούμε ότι πολλά χαρακτηριστικά της εικόνας δεν έχουν βρεθεί. Αντίθετα, εκτελώντας τον αλγόριθμο ModHSW-MyCap παρατηρούμε ότι για 115 ομάδες, τα χαρακτηριστικά που λαμβάνονται στην εικόνα αποτέλεσμα είναι πολύ περισσότερα σε σύγκριση με τον MCT.



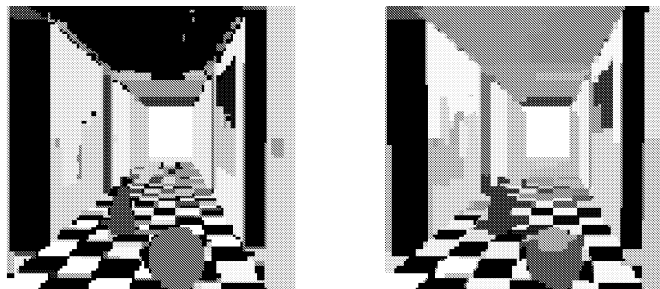
Σχήμα 7.12: Αριστερά MCT-SCap και  $M=400$ . Δεξιά ModHSW-MyCap και  $M=115$ .



Αντίθετα, με τον αλγόριθμο ModHSW-MyCap σε 115 επαναλήψεις παίρνουμε πολύ καλύτερα αποτελέσματα. Σχεδόν καταφέραμε να πάρουμε όλη τη σημαντική πληροφορία της εικόνας. Αυξάνοντας τις ομάδες του αποτελέσματος παίρνουμε και περισσότερη πληροφορία σε κάθε ένα από τους τρεις αλγόριθμους που παρουσιάζουμε.

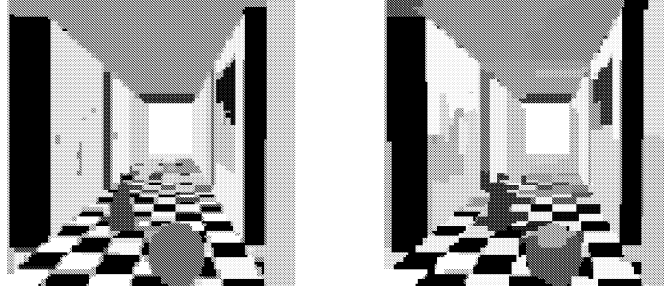


Σχήμα 7.13: Αριστερά MCT-SCap, και  $M=1500$ . Δεξιά ModHSW-MyCap και  $M=115$ .



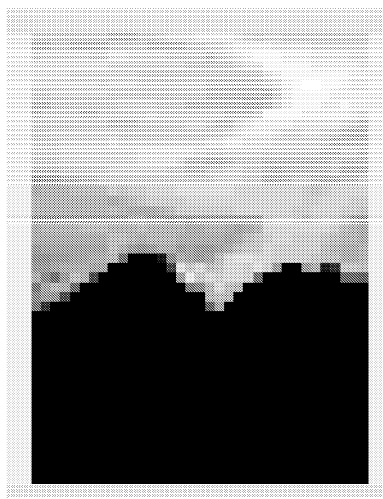
Σχήμα 7.14: Αριστερά MCT- $\alpha$ -SCap,  $\alpha = 30$  και  $M=1400$ . Δεξιά ModHSW-MyCap και  $M=115$ .

Εκτελώντας τον αλγόριθμο MCT- $\alpha$  με παράμετρο  $\alpha = 30$ , παίρνω πολλά χαρακτηριστικά της εικόνας αλλά τα εικονοστοιχεία που ανήκουν στο ταβάνι της αρχικής εικόνας δεν έχουν μπει σε κάποια ομάδα αντίθετα κάθε ένα από αυτά αποτελεί και μια ομάδα στην εικόνα αποτέλεσμα. Πράγματι αυτό μπορούμε να το δούμε οπτικά, δίνοντας χρώμα 0 στις ομάδες του αποτελέσματος που αποτελούνται από ένα και μοναδικό κόμβο. Επειδή λοιπόν προκύπτουν πολλές ομάδες με ένα κόμβο, οι τελικές ομάδες που προκύπτουν είναι 1400. Μειώνοντας κατάλληλα την τιμή του  $\alpha$  μπορούμε να πετύχουμε αποτέλεσμα με λιγότερες ομάδες. Το καλύτερο αποτέλεσμα που καταφέραμε να πετύχουμε είναι θέτοντας  $\alpha = 20$  σύμφωνα με την οποία προκύπτουν 171 ομάδες. Μειώνοντας περισσότερο την τιμή του  $\alpha$  χάνουμε πολλή πληροφορία στην εικόνα γιατί συνενώνονται με παρόμοιες γειτονικές ομάδες (βλ. Κεφάλαιο 6). Πάλι παρατηρούμε ότι τα αποτελέσματα του ModHSW-MyCap είναι συγκρίσιμα με τον αλγόριθμο MCT- $\alpha$ .



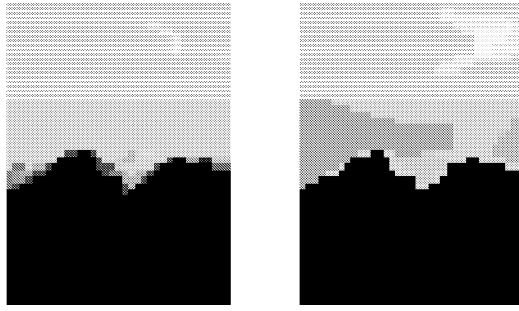
Σχήμα 7.15: Αριστερά MCT- $\alpha$ -SCap,  $\alpha = 20$  και  $M=171$ . Δεξιά ModHSW-MyCap και  $M=115$ .

**Πείραμα 5:** Στο πείραμα αυτό θεωρούμε μια μικρού μεγέθους εικόνα ( $50 \times 40$ ) η οποία έχει δύο ευδιάκριτες περιοχές (βουνό, ουρανός). Η περιοχή του ουρανού αποτελείται από μικρές υποπεριοχές οι οποίες έχουν μεταξύ τους μικρές διαφορές φωτεινότητας. Εκτελώντας τον αλγόριθμο MCT με 100 ομάδες, παρατηρούμε ότι δε μπορούμε να εξάγουμε στο αποτέλεσμα τα βασικά χαρακτηριστικά της εικόνας. Οι διαφορές στις υποπεριοχές του ουρανού δεν είναι τόσο μεγάλες ώστε το βάρος στις ακμές που συνδέουν αυτές τις υποπεριοχές να φθίνει στο μηδέν ως προς την διαφορά ομοιότητας. Με αποτέλεσμα τα βάρη των ακμών που συνδέουν κόμβους που αντιστοιχούν σε εικονοστοιχεία στο όριο του βουνού να έχουν μικρότερο βάρος. Άρα στην προκειμένη περίπτωση πριμοδοτούνται διαμερίσεις στις οποίες το ένα εκ των δύο τμημάτων αποτελείται από μόνο έναν κόμβο που αντιστοιχεί σε εικονοστοιχείο στο όριο του βουνού. Γιαυτό παρατηρούμε ότι οι περισσότερες ομάδες του αποτελέσματος βρίσκονται στο όριο του βουνού.



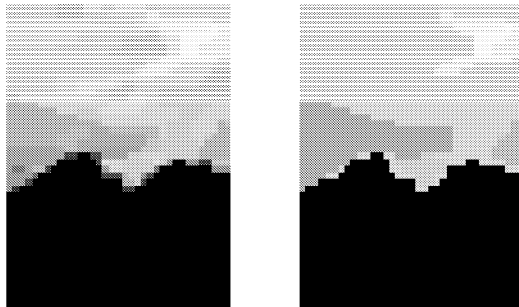
Σχήμα 7.16: Αρχική εικόνα.

Εκτελώντας τον ModHSW-MyCap αλγόριθμο για  $M = 10$  η πληροφορία που λαμ-

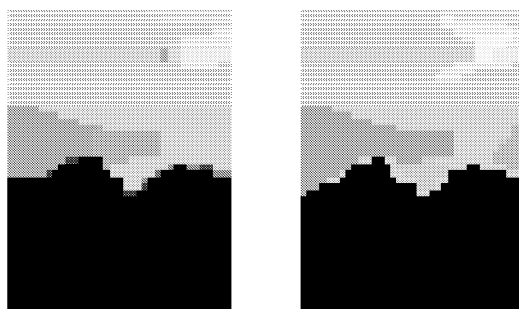


Σχήμα 7.17: Αριστερά MCT-SCap, και  $M=100$ . Δεξιά ModHSW-MyCap και  $M=100$  στο αποτέλεσμα του οποίου εφαρμόσαμε τη διαδικασία MCT-SCap και  $M=10$ .

βάνεται στην εικόνα αποτέλεσμα είναι ικανοποιητική. Παρόλα αυτά τον εκτελέσαμε για  $M = 100$  και έπειτα εφαρμόσαμε τον αλγόριθμο MCT-SCap για  $M = 10$  και το αποτέλεσμα που πήραμε ήταν ακόμη καλύτερο.



Σχήμα 7.18: Αριστερά ModHSW-MyCap, και  $M=100$ . Δεξιά ModHSW-MyCap και  $M=100$  στο αποτέλεσμα του οποίου εφαρμόσαμε τη διαδικασία MCT-SCap και  $M=10$ .



Σχήμα 7.19: Αριστερά ModHSW-MyCap, και  $M=10$ . Δεξιά ModHSW-MyCap και  $M=100$  στο αποτέλεσμα του οποίου εφαρμόσαμε τη διαδικασία MCT-SCap και  $M=10$ .

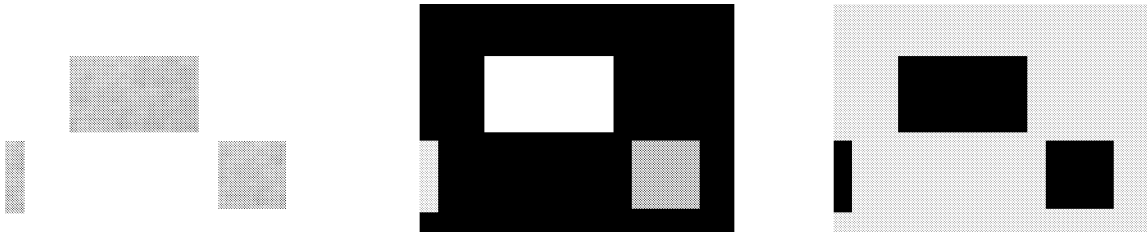
Όπως αναφέραμε και στο Κεφάλαιο 5.2, θα μπορούσαμε να εφαρμόσουμε μια διαδικασία συνένωσης των γειτονικών ομάδων με τον αλγόριθμο MCT-SCap ακολουθώντας τη

διαδικασία που περιγράφεται στον Αλγόριθμο 10. Παρόλο που ο ModHSW-MyCap, μειώνει σημαντικά τον αριθμό των ομάδων που προκύπτουν με μικρό πλήθος κόμβων (συχνά singletons), εντούτοις εξακολουθεί να δίνει πολλές μικρές ομάδες. Αν δεν είναι απαραίτητο να πάρουμε ένα αποτέλεσμα με λεπτομερή περιγραφή της εικόνας, τότε οι ομάδες του αποτελέσματος μπορούν να μειωθούν σημαντικά κρατώντας τα βασικότερα χαρακτηριστικά της εικόνας. Το πείραμα αυτό αποτελεί ένα απλό παράδειγμα το οποίο δείχνει ότι η μεταεπεξεργασία που προτείνουμε μπορεί να αποτελέσει λύση στο πρόβλημα μας. Η οποιαδήποτε συνένωση των ομάδων που προκύπτουν με τον ModHSW-MyCap αλγόριθμο, δεν θα έλυσε το πρόβλημα ικανοποιητικά γιατί θα χάναμε τη σημαντική ιδιότητα που έχουν οι γραφοθεωρητικές μέθοδοι να λαμβάνουν υπόψη και τη χωρική τοπολογία των στοιχείων της εικόνας κατά τη διάρκεια της διαμέρισης.

**Πείραμα 6:** Ο αλγόριθμος ο οποίος προτείνουμε, έχει μια επιπλέον πολύ σημαντική ιδιότητα. Επειδή επιτυγχάνει διαμέριση του γράφηματος χρησιμοποιώντας επαναληπτικά μια διάταξη συνδεσιμότητας, με βάση την οποία συνενώνει δύο κόμβους  $s$ ,  $t$  που εισάγονται τελευταίοι στη διάταξη, μπορεί να δώσει διαμέριση κατά την οποία προκύπτουν δύο υπογράφηματα  $A$ ,  $B$ , έτσι ώστε το υπογράφημα  $A$  να μην είναι συνεκτικό. Δηλαδή, μπορεί να προκύψει ομάδα που να περιέχει δύο ανεξάρτητες περιοχές στην εικόνα. Αυτό μπορεί για παράδειγμα να συμβεί όταν ο κόμβος  $t$  έχει προκύψει μετά από διαδοχικές συρρικνώσεις κόμβων μιας ομοιόμορφης περιοχής και αντίστοιχα ο κόμβος  $s$  από διαδοχικές συρρικνώσεις κόμβων μιας άλλης περιοχής. Επομένως κατά μετά τη συνένωση των υπερκόμβων  $s$  και  $t$  προκύπτει ο υπερκόμβος  $t$  ο οποίος περιέχει τους κόμβους δύο ανεξάρτητων υπογραφήματων. Άρα η διαμέριση στην οποία προκύπτει ότι ο κόμβος  $t$  ανήκει στο ένα τμήμα της διαμέρισης και όλοι οι υπόλοιποι κόμβοι στο άλλο, δίνει την περίπτωση που αναφέρουμε. (Μια τέτοια διαμέριση μπορεί να προκύψει μόνο όταν το κριτήριο διαμέρισης είναι το MyCut. Αν το κριτήριο διαμέρισης είναι το SCut τότε η  $(s, t)$ -τομή είναι η ελάχιστη (ελαχιστοποιεί τη ποσότητα SCut), αφού τα υπογράφηματα που επάγονται από τους κόμβους  $s$  και  $t$  είναι ανεξάρτητα και επομένως  $SCut = 0$ .) Τα ανεξάρτητα υπογράφηματα που επάγονται από τους κόμβους του  $t$  έχουν συνδεσιμότητα μηδέν (επειδή το γράφημα είναι πλέγμα) και επομένως με την εκτέλεση του ModHSW-MyCap σε αυτό το μη συνεκτικό υπογράφημα τελικά προκύπτει διαμέριση που χωρίζει τα δύο ανεξάρτητα υπογράφηματα σε διαφορετικές ομάδες.

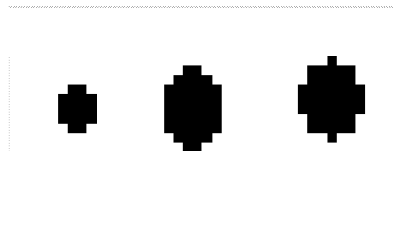
Στη συνέχεια δίνουμε ένα παράδειγμα στο οποίο μπορεί αυτό να βρει εφαρμογή. Έστω μια εικόνα η οποία περιλαμβάνει τρεις φλούδες πορτοκαλιού. Αν ο ModHSW-MyCap εφαρ-

μοστεί για  $M = 2$  τότε όλες οι φλούδες πορτοκαλιού προκύπτουν σε μια ομάδα και το φόντο στην άλλη. Αν όμως, ο ModHSW-MyCap εφαρμοστεί για  $M = 4$  τότε κάθε φλούδα πορτοκαλιού αποτελεί μια ομάδα στο αποτέλεσμα.

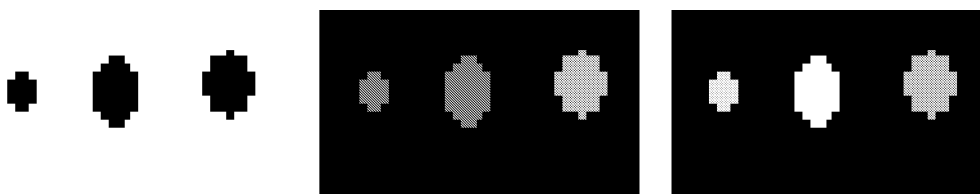


Σχήμα 7.20: Αρχική εικόνα. Εκτέλεση ModHSW-MyCap, και  $M=4$ . Εκτέλεση ModHSW-MyCap και  $M=2$ .

Στη συνέχεια, δίνουμε μια απλή εικόνα, στην οποία υπάρχουν τρεις μπάλες ίδιας και σταθερής φωτεινότητας. Πράγματι, αν εκτελεστεί ο ModHSW-MyCap το αποτέλεσμα της πρώτης επανάληψης είναι μια ομάδα η οποία περιλαμβάνει όλες τις μπάλες και μια δεύτερη ομάδα με το φόντο. Στη δεύτερη επανάληψη σπάει η ομάδα με τις μπάλες σε δύο ομάδες χωρίζοντας την μια μπάλα σε μια ομάδα και τις άλλες δύο στην άλλη. Στην τρίτη και τελευταία επανάληψη χωρίζεται η ομάδα με τις δύο μπάλες σε δύο ομάδες, όπου κάθε μια περιέχει μια μπάλα.



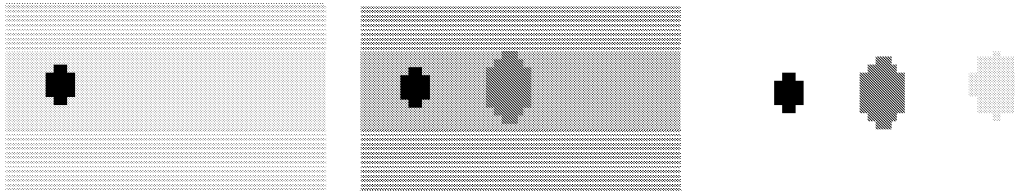
Σχήμα 7.21: Αρχική εικόνα.



Σχήμα 7.22: Εκτέλεση ModHSW-MyCap με  $M=2$ . Εκτέλεση ModHSW-MyCap με  $M=3$ . Εκτέλεση ModHSW-MyCap με  $M=4$ .

Αντίθετα αν εκτελεστεί ο HSW-MyCap το αποτέλεσμα της πρώτης επανάληψης είναι μια ομάδα η οποία περιλαμβάνει τη μικρότερη μπάλα και μια δεύτερη ομάδα με όλα τα υπόλοιπα

εικονοστοιχεία. Στη δεύτερη επανάληψη σπάει η ομάδα με το μεγαλύτερο τετραγωνικό σφάλμα (δηλαδή αυτή στην οποία περιέχονται εικονοστοιχεία από διαφορετικές περιοχές φωτεινότητας), δίνοντας μια ομάδα με την δεύτερη μικρότερη μπάλα και όλα τα υπόλοιπα εικονοστοιχεία στην άλλη. Αντίστοιχα στην τελευταία επανάληψη χωρίζεται και η τελευταία μπάλα από τα εικονοστοιχεία που ανήκουν στο φόντο.



Σχήμα 7.23: Εκτέλεση HSW-MyCap με  $M=2$ . Εκτέλεση HSW-MyCap με  $M=3$ . Εκτέλεση HSW-MyCap με  $M=4$ .

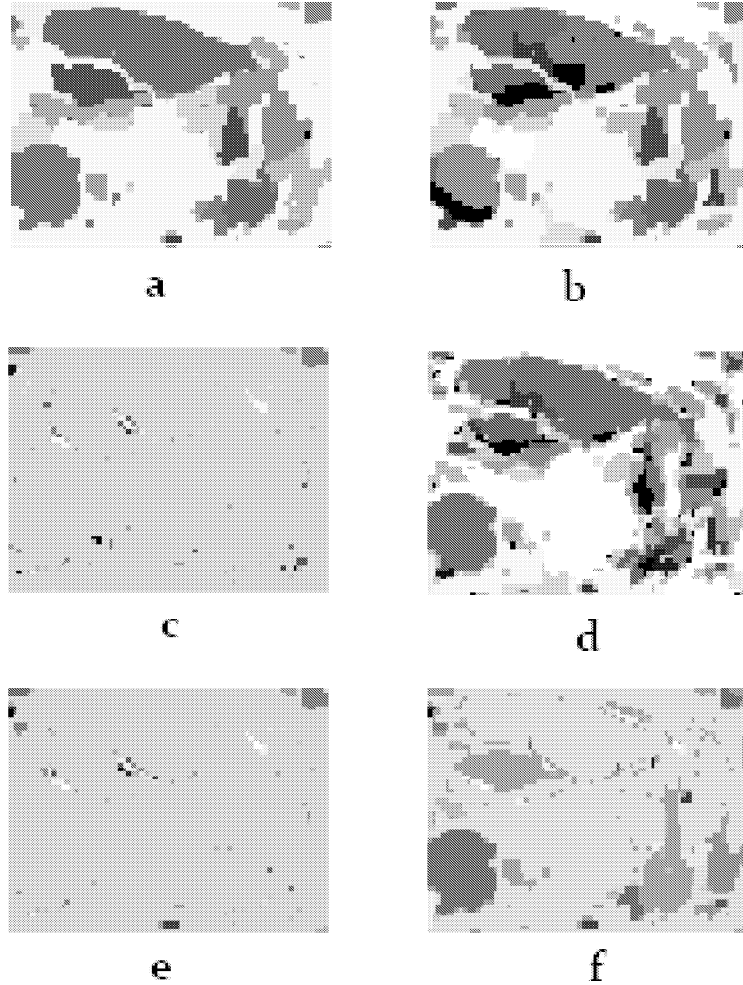
**Πείραμα 7:** Στο πείραμα αυτό χρησιμοποιούμε μια εικόνα με πολύ λεπτομέρεια. Αποτελείται γενικά από τρεις περιοχές χρωμάτων, αλλά κάθε μια πέτρα στο ποτάμι αποτελεί μια ανεξάρτητη περιοχή. Επομένως χρειάζεται να τρέξουμε τον αλγόριθμο για πολλές ομάδες ώστε να πάρουμε την πληροφορία της εικόνας. Όπως αναφέραμε και στο προηγούμενο παράδειγμα με τον αλγόριθμο ModHSW-MyCap μπορούμε να πάρουμε και ομάδες που περιέχουν περισσότερες από μια πέτρες αλλά με κατάλληλη αύξηση των επαναλήψεων, και αν το πρόβλημα το απαιτεί, μπορούμε να χωρίσουμε τις πέτρες σε διαφορετικές ομάδες.



Σχήμα 7.24: Αρχική εικόνα.

Πιο κάτω βλέπουμε τα αποτελέσματα των αλγορίθμων σε αυτό το παράδειγμα εικόνας.

Παρατηρούμε ότι στον αλγόριθμο MCT- $\alpha$ -SCap, τα περισσότερα χαρακτηριστικά της εικόνας τα προσεγγίζονται καλύτερα με τιμή παραμέτρου  $\alpha = 40$ . Αυτό συμβαίνει διότι όσο μειώνεται η τιμή της παραμέτρου μειώνεται επίσης και το πλήθος των ομάδων αφού κάποιες ομάδες στο αποτέλεσμα συνενώνονται όπως φαίνεται στο Σχήμα 7.25(d).



Σχήμα 7.25: a) ModHSW-MyCap για  $M = 50$ . b) Εκτέλεση ModHSW-MyCap για  $M = 100$ . c) HSW-MyCap για  $M = 100$ . d) MCT- $\alpha$ -SCap,  $\alpha = 40$  προκύπτουν ομάδες  $M = 209$ . e) MCT-MyCap για  $M = 200$ . f) MCT-SCap για  $M = 200$ .

Στο Σχήμα 7.25(c), φαίνονται τα αποτελέσματα του HSW-MyCap. Το κριτήριο διαμέρισης που χρησιμοποιεί είναι το κριτήριο ελάχιστης τομής και επομένως χρησιμοποιώντας τα βάρη MyCap, στο αποτέλεσμα προκύπτουν πολλές μικρές ομάδες.

Για τον λόγο αυτό προκύπτει παρόμοιο αποτέλεσμα και κατά την εκτέλεση του MCT-MyCap που φαίνεται στο Σχήμα 7.25(e). Μάλιστα, προκύπτουν ακόμη περισσότερες ομάδες που αποτελούνται από ένα και μοναδικό στοιχείο.

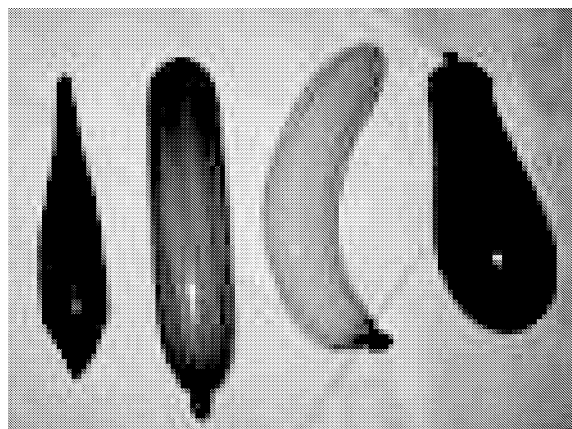
Στον ίδιο αριθμό επαναλήψεων, χρησιμοποιώντας τα βάρη SCap (Σχήμα 7.25(f)) λαμβάνονται περισσότερα χαρακτηριστικά στο αποτέλεσμα. Αξίζει να αναφέρουμε εδώ ότι μειώνοντας την τιμή της παραμέτρου  $\sigma$  το αποτέλεσμα βελτιώνεται περισσότερο. Αλλά είναι ένα ζήτημα το πώς καθορίζουμε κατάλληλα την τιμή της (εμείς τη θεωρούμε σε όλα τα

πειράματα μας σταθερή  $\sigma = 10$ ).

Στα Σχήματα 7.25(a), 7.25(b) φαίνονται τα αποτελέσματα του ModHSW-MyCap τα οποία είναι ικανοποιητικά ακόμη και για  $M = 50$ , ενώ όσο αυξάνεται η τιμή του  $M$ , λαμβάνουμε περισσότερη πληροφορία στο αποτέλεσμα (λεπτομέρεια στην εικόνα).

**Πείραμα 8:** Στο πείραμα αυτό χρησιμοποιήσαμε μια εικόνα με τέσσερα διαφορετικά αντικείμενα, τρία από τα οποία έχουν παρόμοια φωτεινότητα ενώ το τέταρτο έχει φωτεινότητα παρόμοια με το φόντο. Ο κλασικός αλγόριθμος ομαδοποίησης  $K$ -μέσων για  $M = 5$  δίνει τα αποτελέσματα που βλέπουμε στο Σχήμα 7.27(a). Η ομαδοποίηση γίνεται με βάση το χρώμα των εικονοστοιχείων. Θα μπορούσε ο  $K$ -μέσων να λάβει υπόψη και τη χωρική τοπολογία, όπως και οι γραφοθεωρητικές μέθοδοι που παρουσιάσαμε, ορίζοντας μια πιο σύνθετη συνάρτηση βαρών.

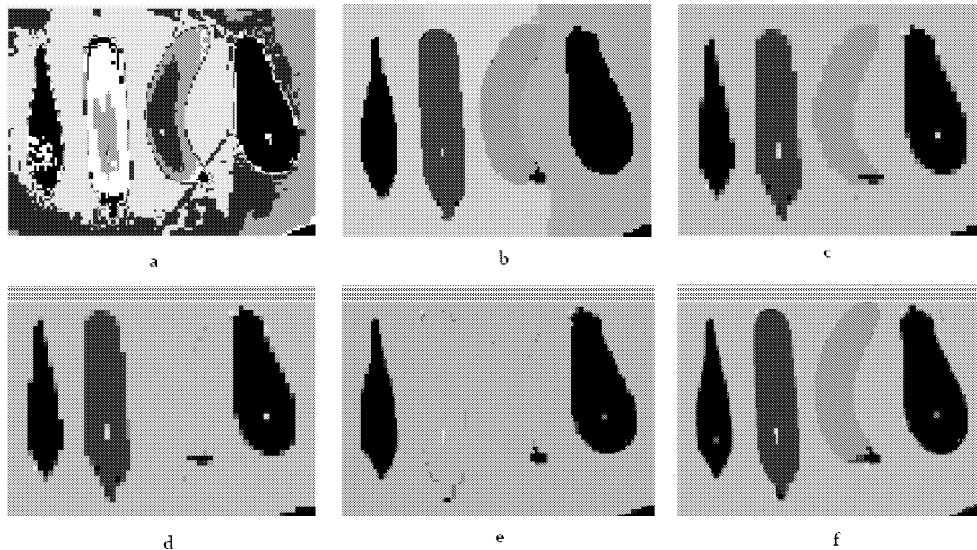
Εκτελώντας τον αλγόριθμο ModHSW-MyCap ώστε να προκύψουν στο τελικό αποτέλεσμα  $M = 5$  ομάδες, παρατηρήσαμε ότι διαχωρίστηκαν τα τρία από τα τέσσερα αντικείμενα (κάθε σε ξεχωριστές ομάδες) και το φόντο. Αντίθετα η μπανάνα δεν βρέθηκε γιατί η φωτεινότητά της είναι πολύ κοντά στη φωτεινότητα του φόντου. Επομένως το κόστος αποκοπής της μπανάνας από το φόντο είναι ακριβό (δεδομένου ότι τα βάρη MyCap δεν φθίνουν απότομα στο μηδέν). Για να συμπεριληφθεί και η μπανάνα στο αποτέλεσμα θα πρέπει να αυξήσουμε το πλήθος των επαναλήψεων συνεπώς και το πλήθος των ομάδων  $M$ . Παρατηρούμε ότι για  $M = 20$ , διαχωρίστηκαν ορθά όλα τα αντικείμενα όπως φαίνεται στο Σχήμα 7.27(b).



Σχήμα 7.26: Αρχική εικόνα.

Η εκτέλεση του MCT- $\alpha$ -SCap με παράμετρο  $\alpha = 15$  δίνει στο αποτέλεσμα και τις πέντε περιοχές της εικόνας, αλλά δίνει και πολλές μικρές περιοχές στα όρια των αντικειμένων.





Σχήμα 7.27: a) K-μέσων για  $M = 4$ . b) Εκτέλεση ModHSW-MyCap για  $M = 20$ . c) MCT- $\alpha$ -SCap,  $\alpha = 15$  προκύπτουν ομάδες  $M = 90$ . d) MCT- $\alpha$ -SCap,  $\alpha = 10$  προκύπτουν ομάδες  $M = 84$ . e) MCT-SCap για  $M = 300$ . f) MCT-SCap για  $M = 500$ .

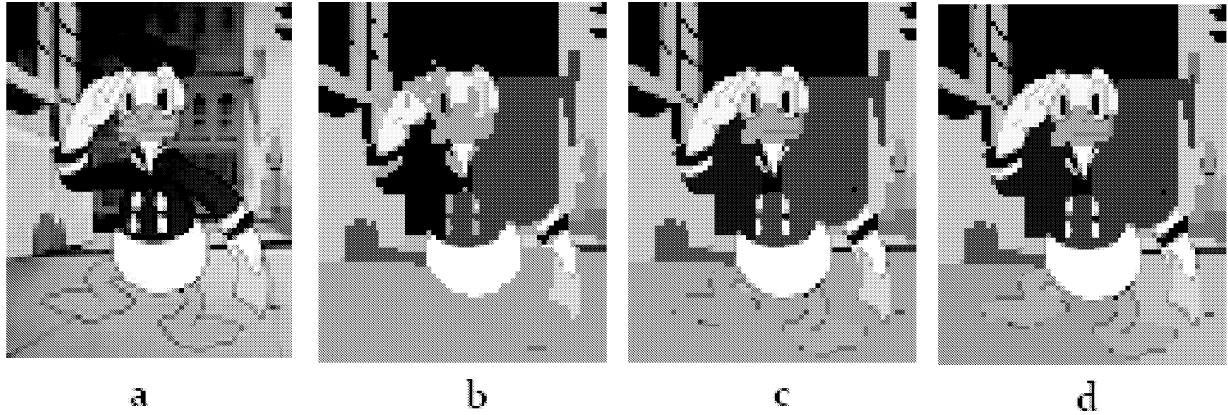
Αυτό είναι λογικό γιατί, σύμφωνα με τον ορισμό της κοινότητας, κάθε στοιχείο της πρέπει να συνδέεται με τα στοιχεία στο εσωτερικό της κοινότητας περισσότερο από ότι με τα στοιχεία που δεν ανήκουν σε αυτή. Αναπαριστώντας λοιπόν την εικόνα με ένα πλέγμα, τα μόνα στοιχεία μιας περιοχής που μπορεί να παραβιάζουν αυτόν τον περιορισμό είναι τα στοιχεία που βρίσκονται στα όρια των αντικειμένων. Συνεπώς, πολλά από αυτά τα στοιχεία δημιουργούν μικρές κοινότητες, που αποτελούν μικρές ομάδες του αποτελέσματος στα όρια των αντικειμένων. Στη συνέχεια, μειώνουμε τη τιμή της παραμέτρου θέλοντας να μειώσουμε τις ομάδες του αποτελέσματος. Παρατηρούμε όμως ότι το πρόβλημα με τις μικρές κοινότητες παραμένει και χάνουμε μια μεγάλη περιοχή, την περιοχή της μπανάνας, η οποία συνενώνεται με το φόντο. Καταλήγουμε έτσι, ότι το καλύτερο αποτέλεσμα το επιτυγχάνουμε με  $a = 15$  όπως φαίνεται στο Σχήμα 7.27(c), στο οποίο όμως προκύπτουν πολλές μικρές ομάδες στα όρια των αντικειμένων.

Με την εκτέλεση του MCT-SCap χρειάζεται να προκύψει ένας μεγάλος αριθμός ομάδων στο αποτέλεσμα ώστε να πάρουμε όλα τα αντικείμενα της εικόνας. Πράγματι, για  $M = 300$ , καταφέραμε να πάρουμε μόνο δύο από τα τέσσερα αντικείμενα και το φόντο. Αυτό συμβαίνει διότι στα όρια των αντικειμένων έχουμε μικρές ομοιότητες όπου ο αλγόριθμος τείνει να αποκόπτει μικρές περιοχές στα όρια των αντικειμένων. Για  $M = 500$ , λαμβάνονται όλα τα αντικείμενα της εικόνας.

Καταλήγουμε στο συμπέρασμα ότι η υβριδική μέθοδος (διατηρεί μικρή πληροφορία εξαιτίας του γραφήματος που χρησιμοποιεί για την αναπαράσταση αλλά επιλέγει το υπογράφημα προς διαμέριση θεωρώντας κάθε υπογράφημα πλήρες) που προτείνουμε δεν επηρεάζεται από το γεγονός ότι η εικόνα αναπαρίσταται με ένα πλέγμα, τουλάχιστον όχι σε τόσο μεγάλο βαθμό όσο οι άλλες δύο μέθοδοι. Παρόλα αυτά αξίζει να δούμε και τη συμπεριφορά του αλγορίθμου αναπαριστώντας την εικόνα με άλλο τύπο γραφήματος το οποίο θα διατηρεί περισσότερη πληροφορία από την εικόνα.

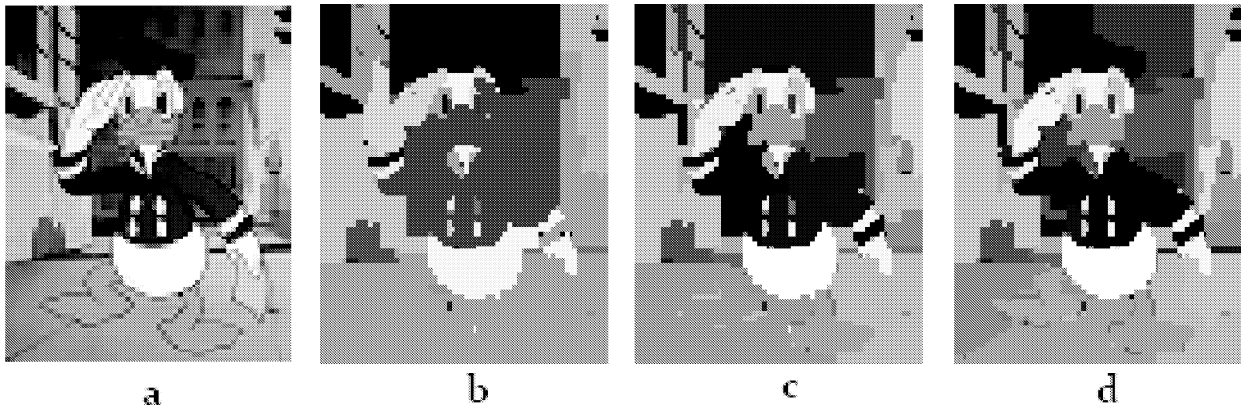
**Πείραμα 9:** Χρησιμοποιήσαμε μια εικόνα με πολλές περιοχές, κάποιες από τις οποίες παρουσιάζουν μεγαλύτερη ανομοιογένεια με τις γειτονικές τους περιοχές και κάποιες άλλες λιγότερη. Εφαρμόσαμε τον αλγόριθμο MCT- $\alpha$  με τα βάρη SCap τρεις φορές ορίζοντας κάθε φορά διαφορετική τιμή στη παράμετρο  $\alpha$ . Όσο η τιμή  $\alpha$  αυξάνει συλλέγεται περισσότερη λεπτομέρεια στο αποτέλεσμα. Παρόλα αυτά, ένα μειονέκτημα του αλγορίθμου είναι ότι δε είναι δυνατόν να παραμετροποιηθεί ως προς το πλήθος των παραγόμενων ομάδων, με αποτέλεσμα να αυξάνεται σημαντικά ο αριθμός των ομάδων. Επίσης το  $\alpha$  το οποίο φράσσει άνω και κάτω την συνδεσιμότητα ανάμεσα σε δύο ομάδες που προκύπτουν και τη συνδεσιμότητα στο εσωτερικό της κάθε ομάδας αντίστοιχα, έχει ίδια τιμή σε όλη την εικόνα μη λαμβάνοντας υπόψη ότι η εικόνα μπορεί να περιλαμβάνει περιοχές με διαφορετικά χαρακτηριστικά. Επομένως, με μικρές τιμές  $\alpha$  λαμβάνουμε λιγότερες ομάδες, οι οποίες αποτελούν πιο ισχυρές κοινότητες αλλά σε αυτές δε συμπεριλαμβάνονται οι περιοχές που παρουσιάζουν μικρότερη ανομοιογένεια με τις γειτονικές τους ομάδες (λιγότερο ισχυρές κοινότητες). Με την αύξηση του  $\alpha$  λαμβάνονται και λιγότερο ισχυρές κοινότητες αλλά το πλήθος των ομάδων αυξάνει σημαντικά. Τα αποτελέσματα του αλγορίθμου για τιμές  $\alpha = 10$ ,  $\alpha = 20$  και  $\alpha = 30$  φαίνονται στο Σχήμα 7.28.

Στη συνέχεια εκτελέσαμε τον αλγόριθμο ModHSW-MyCap. Σύμφωνα με τη συνάρτηση καθορισμού βαρών που χρησιμοποιούμε, επιλέγουμε κάθε φορά να αποκόψουμε τις περιοχές με τη μεγαλύτερη ανομοιογένεια ως προς τις γειτονικές τους, λόγο του ότι δίνουμε μικρότερο βάρος σε ακμές όπου η αναλογία των διαφορών των φωτεινοτήτων που συμμετέχουν στο καθορισμό του βάρους ως προς τη μεγαλύτερη διαφορά φωτεινότητας που παρουσιάζεται στην περιοχή της εικόνας την οποία πρόκειται να διαμερίσουμε είναι μεγάλη. Άρα ο αλγόριθμος, αφού επιλέξει την περιοχή  $A$  με τη μεγαλύτερη ανομοιογένεια (μεγαλύτερο τετραγωνικό σφάλμα) στο αποτέλεσμα της προηγούμενης επανάληψης, αποκόπτει από την  $A$  την περιοχή  $B$  με την μεγαλύτερη ανομοιογένεια ως προς τα υπόλοιπα στοιχεία της περιοχής  $A$ .



Σχήμα 7.28: a) Αρχική εικόνα. b) MCT- $\alpha$ -SCap,  $\alpha = 10$  προκύπτουν ομάδες  $M = 81$ . c) MCT- $\alpha$ -SCap,  $\alpha = 20$  προκύπτουν ομάδες  $M = 340$ . d) MCT- $\alpha$ -SCap,  $\alpha = 30$  προκύπτουν ομάδες  $M = 352$ .

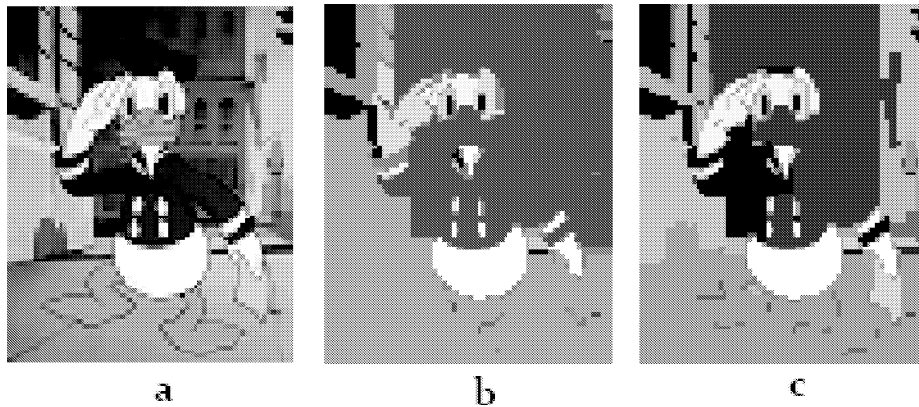
Για  $M = 40$  λαμβάνονται τα πιο έντονα τμήματα ενώ αυξάνοντας το  $M$  λαμβάνουμε όλα και περισσότερη πληροφορία από την εικόνα. Τα αποτελέσματα που παίρνουμε για  $M = 120$  είναι πολύ καλύτερα από αυτά που δίνει ο MCT- $\alpha$ -SCap για  $\alpha = 30$  με  $M = 352$ .



Σχήμα 7.29: a) Αρχική εικόνα. b) ModHSW-MyCap για  $M = 40$ . c) ModHSW-MyCap για  $M = 80$ . d) ModHSW-MyCap για  $M = 120$ .

Ο αλγόριθμος MCT-SCap δίνει για  $M = 400$  πολύ λιγότερη πληροφορία από τους δύο προηγούμενους αλγόριθμους.

**Πείραμα 10:** Σε όλα τα προηγούμενα πειράματα παρατηρήσαμε ότι ο αλγόριθμος MCT-SCap παράγει πάρα πολλές ομάδες μέχρι να προκύψει ικανοποιητική διαμέριση της εικόνας. Το 1999 οι Scanlon-Deo [2] χρησιμοποίησαν τον αλγόριθμο τους (τον οποίο δεν χρησιμοποιήσαμε στην πειραματική μας μελέτη αλλά αναφέρουμε την τεχνική που χρησιμοποιεί



Σχήμα 7.30: a) Αρχική εικόνα. b) MCT-SCar και  $M = 100$ . c) MCT-SCar και  $M = 400$ .

στο Κεφάλαιο 4.2) στην εικόνα που φαίνεται στο Σχήμα 7.31 και πήραν αποτέλεσμα με  $M = 115$  χρησιμοποιώντας το δικό τους κριτήριο διαμέρισης BCut. Στον αλγόριθμο τους χρησιμοποίησαν πλήρες γράφημα σε αντίθεση με το πλέγμα που χρησιμοποιούμε στα δικά μας πειράματα και όρισαν συνάρτηση ακμικών βαρών η οποία λαμβάνει υπόψη εκτός από τη διαφορά των φωτεινοτήτων και την χωρική τοπολογία των εικονοστοιχείων.

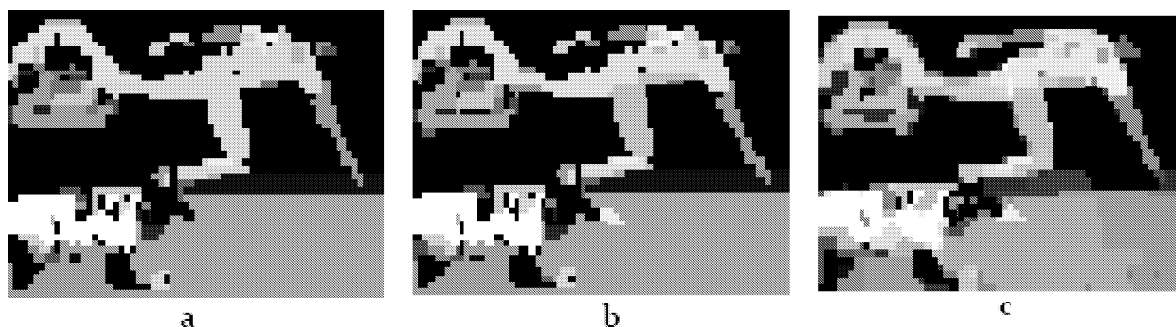


Σχήμα 7.31: a) Αρχική εικόνα με μείωση στο 40%

Εφαρμόσαμε στην ίδια εικόνα (αλλά με σμίκρυνση της στο 40%) τους αλγόριθμους ModHSW-MyCut και MCT- $\alpha$ -SCar. Στον αλγόριθμο ModHSW-MyCut μετά από 115 επαναλήψεις πήραμε ικανοποιητικό αποτέλεσμα, συγκρίσιμο με το αποτέλεσμα των Scanlon-Deo. Παρόλο που εμείς αναπαριστούμε την εικόνα με πλέγμα (λιγότερη πληροφορία).

Ο αλγόριθμος MCT- $\alpha$ -SCar με παράμετρο  $\alpha = 5$  και  $\alpha = 10$  έδωσε ικανοποιητικό αποτέλεσμα αλλά σε 233 και 256 ομάδες, το οποίο σημαίνει ότι απομόνωσε πολλά εικονοστοιχεία.

Τα πιο πάνω αποτελέσματα εφαρμόστηκαν πάνω στην εικόνα του Σχήματος 7.31 η



Σχήμα 7.32: a) MCT- $\alpha$ -SCap,  $\alpha = 5$  προκύπτουν ομάδες  $M = 233$ . b) MCT- $\alpha$ -SCap,  $\alpha = 10$  προκύπτουν ομάδες  $M = 256$ . c) ModHSW-MyCap για  $M = 115$ .

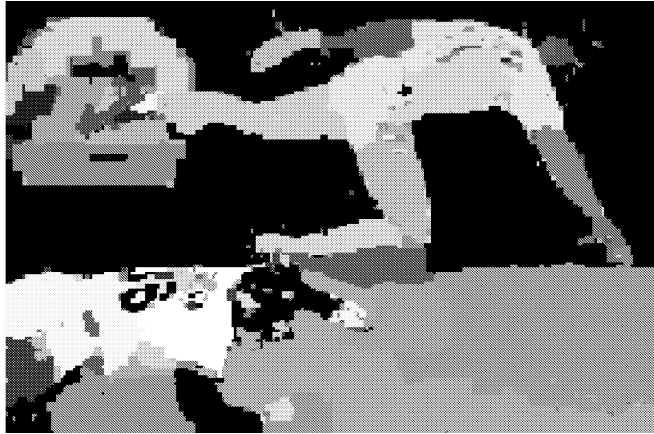
οποία προέκυψε με μείωση της εικόνας τους Σχήματος στο 40%. Δηλαδή τα πιο πάνω αποτελέσματα προέκυψαν από εικόνα από την οποία έχει χαθεί ορισμένη πληροφορία κατά τη συμπίεση.



Σχήμα 7.33: a) Αρχική εικόνα

Εφαρμόζοντας τον αλγόριθμο ModHSW-MyCap για  $M = 115$  στην αρχική εικόνα παρατηρούμε ότι τα αποτελέσματα είναι πολύ ικανοποιητικά και ότι το πλήθος των ομάδων του αποτελέσματος δεν εξαρτάται από το μέγεθος της εικόνας στην οποία εφαρμόζεται ο αλγόριθμος.

Με βάση την πιο πάνω παρατήρηση θα ήταν ενδιαφέρον να μελετηθεί κατά πόσο μπορούμε να κατασκευάσουμε μια εικόνα χαμηλότερης ανάλυσης από την αρχική (για παράδειγμα χρησιμοποιώντας την μέθοδο που χρησιμοποιείται κατά τη κατασκευή μιας γκαουσιανής πυραμίδας) και να εφαρμόσουμε σε αυτήν τον αλγόριθμο κατάτμησης ModHSW-MyCap και έπειτα για κάθε υπογράφημα που προκύπτει (για κάθε ομάδα στην εικόνα χαμηλότερης ανάλυσης), να βρίσκουμε το αντίστοιχο επαγόμενο υπογραφήμα στην αρχική εικόνα. Κατά αυτόν τον τρόπο έχουμε μια αρχική διαμέριση της εικόνας σε τμήματα (υπογραφήματα) τα



Σχήμα 7.34: Εκτέλεση του ModHSW-MyCap για  $M = 115$  στην αρχική εικόνα.

οποία με τον ίδιο αλγόριθμο ModHSW-MyCap έχουμε τη δυνατότητα να συνεχίσουμε να διαμερίζουμε. Κάτι τέτοιο θα βελτιώνε σημαντικά το χρόνο δεδομένου ότι ο ModHSW-MyCap αρχικά εργάζεται σε μειωμένου μεγέθους γράφημα, και έπειτα δουλεύει με τα αποτελέσματα (τα οποία δεν είναι παρά επαγόμενα υπογραφήματα του αρχικού γραφήματος) τα οποία είναι και πάλι μειωμένου μεγέθους.

## ΚΕΦΑΛΑΙΟ 8

# ΣΥΜΠΕΡΑΣΜΑΤΑ - ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

---

### 8.1 Συμπεράσματα

### 8.2 Μελλοντική εργασία

---

### 8.1 Συμπεράσματα

Στην παρούσα εργασία μελετήσαμε αλγόριθμους εύρεσης ελάχιστης τομής γραφήματος. Παρατηρήσαμε ότι το πρόβλημα αυτό βρίσκει πολλές εφαρμογές στο πρόβλημα κατάτμησης εικόνας. Μελετήσαμε δύο αλγόριθμους κατάτμησης εικόνας που υπάρχουν στη βιβλιογραφία, οι οποίοι βασίζονται σε διαμερίσεις γραφήματος και συγκεκριμένα χρησιμοποιούν αλγόριθμους εύρεσης ελάχιστης τομής που βασίζονται σε τεχνικές ροής. Αναπτύξαμε ένα ιεραρχικό αλγόριθμο διαμέρισης γραφήματος χρησιμοποιώντας αλγόριθμο εύρεσης ελάχιστης τομής ο οποίος δε χρησιμοποιεί τεχνικές ροής αλλά βασίζεται σε διαδοχικές συρρικνώσεις ακμών. Εφαρμόσαμε και τους τρεις αυτούς αλγόριθμους στο πρόβλημα κατάτμησης εικόνας και μελετήσαμε τη συμπεριφορά τους. Παρατηρήσαμε ότι ο ιεραρχικός αλγόριθμος που αναπτύξαμε εφαρμόζεται ικανοποιητικά στο πρόβλημα κατάτμησης εικόνας ακόμη και σε εικόνες που περιλαμβάνουν πολλές περιοχές με διαφορετικά χαρακτηριστικά. Αυτό συμβαίνει γιατί ορίσαμε τα βάρη χρησιμοποιώντας μια παράμετρο η οποία εξαρτάται σε κάθε επανάληψη από την περιοχή την οποία διαμερίζουμε (σε κάθε επανάληψη τα βάρη επαναπροσδιορίζονται). Επίσης περιορίσαμε σημαντικά το πρόβλημα που παρουσιάζουν οι άλλες δύο μέθοδοι κατά το οποίο προκύπτουν πολλές μικρές ομάδες. Αυτό το πετύχαμε αλλάζοντας το κριτήριο διαμέρισης του αλγόριθμου εύρεσης ελάχιστης τομής με τέτοιο τρόπο

ώστε να λαμβάνουμε υπόψη και το μέγεθος των ομάδων που προκύπτουν μετά τη διαμέριση. Παρόλα αυτά εξακολουθούν να προκύπτουν αρκετές μικρές ομάδες στο αποτέλεσμα (πολύ λιγότερες από τις άλλες δύο μεθόδους) τις οποίες συνενώνουμε με τον Αλγόριθμο 10 διατηρώντας έτσι τη χωρική τοπολογία των περιοχών.

## 8.2 Μελλοντική εργασία

Στα μελλοντικά μας σχέδια εντάσσεται καταρχήν η μελέτη της συμπεριφοράς της ιεραρχικής μεθόδου που αναπτύξαμε όταν η αναπαράσταση της εικόνας γίνεται με ένα γράφημα το οποίο διατηρεί περισσότερη πληροφορία, για παράδειγμα πλήρες γράφημα.

Επίσης κατά πόσο θα μπορούσαμε να βελτιώσουμε τη πολυπλοκότητα χρόνου του αλγόριθμου αποδοτικά. Μια ιδέα που έχουμε με την οποία η πολυπλοκότητα χρόνου βελτιώνεται σημαντικά είναι να εφαρμόσουμε τον αλγόριθμο σε μια εικόνα χαμηλότερης ανάλυσης που προκύπτει από την αρχική. Για παράδειγμα θα μπορούσαμε να μειώσουμε την ανάλυση μιας εικόνας κάνοντας συνέλιξη στην αρχική εικόνα με ένα γκαουσιανό πυρήνα, κρατώντας έτσι τα βασικά χαρακτηριστικά της και στη συνέχεια εφαρμόζοντας δειγματοληψία στην εικόνα που προκύπτει μετά τη συνέλιξη. Αφού εφαρμοστεί λοιπόν ο αλγόριθμος στην εικόνα μειωμένου μεγέθους (χαμηλής ανάλυσης), σε κάθε υπογράφημα που προκύπτει βρίσκουμε το αντίστοιχο επαγόμενο υπογραφήμα στην αρχική εικόνα πάνω στο οποίο εφαρμόζουμε τον αλγόριθμο για επιπλέον διαμερίση της εικόνας.



## ΒΙΒΛΙΟΓΡΑΦΙΑ

---

- [1] Z. Wu and R. Leahy. “*An optimal graph theoretic approach to data clustering: theory and it’s application to image segmentation*”. IEEE Trans. Pattern Anal. Mach. Intell. 15: 1101-1113, 1993.
- [2] J. Scanlon and N.Deo. “*Graph-theoretic algorithms for image segmentation*”, In: IEEE International Symposium on Circuits and Systems,6: 141-144, 1999.
- [3] O. Veksler. “*Image segmentation by nested cuts*”, In Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1: 339-344, 2000.
- [4] G. W. Flake, R. E. Tarjan and K. Tsioutsoulis. “*Graph clustering and minimum cut trees*”, Internet Mathematics, 1: 355-378, 2004.
- [5] B. Saha, P. Mitra. “*Fast Incremental Minimum-Cut Based Algorithm for Graph Clustering*”.
- [6] R. E. Gomory and T. C. Hu. “*Multi-terminal network flows*”, J-SIAM, 9: 551-570, 1961.
- [7] M. Stoer and F. Wagner. “*A simple min-cut algorithm*”, Journal of the ACM, 44: 585-591, 1997.
- [8] J. Esmonds and R. M. Karp. “*Theoretical improvements in algorithmic efficiency of network flow problem*”, Journal of the ACM, 19: 248-264, 1972.
- [9] H. Nagamochi, T. Ono and T. Ibaraki. “*Implementing an efficient minimum capacity cut algorithm*”, Math. Program, 67: 325-341, 1994.
- [10] H. Nagamochi and T. Ibaraki. “*Computing edge-connectivity in multigraphs and capacitated graphs*”, SIAM Journal on Discrete Mathematics, 5: 54-66, 1992.

- [11] N. Curet, M. Gaston and J. DeVinney. “*An Efficient Network Flow Code for Finding All Minimum Cost  $s$ - $t$  Cutsets*”, to appear in Computers and Operations Research, 2001. R. K. Ahuja , J. B. Orlin , R. E. Tarjan. “*Improved time bounds for the maximum flow problem*”, *SIAM Journal on Computing*, 18: 939-954, Oct. 1989
- [12] J. Cheriyan and T. Hagerup. “*A randomized maximum flow algorithm*”, In Proc. FOCS 30, 118-123, 1989.
- [13] J. Cheriyan , T. Hagerup , K. Mehlhorn. “*Can a maximum flow be computed in  $o(nm)$  time?*”, Proceedings of the seventeenth international colloquium on Automata, languages and programming, 235-248, 1990.
- [14] A. V. Goldberg and R. E. Tarjan. “*A new approach to the maximum flow problem*”, *J. Assoc. Comput. Mach.*, 35: 921-940,1988.
- [15] A. V. Goldberg. “*A new approach to the max-flow algorithm*”. Technical Report MIT/LCS/TM-291, Laboratory for Computer Science, MIT, Cambridge, MA, 1985.
- [16] V. King, S. Rao and R. Tarjan. “*A Faster Deterministic Maximum Flow Algorithm*”. *Journal of Algorithms*, 17: 447-474, 1994.
- [17] J. Hao, J. B. Orlin. “*A faster algorithm for finding the minimum cut in a directed graph*”. *Journal of Algorithms*, 17: 424-446, 1994.
- [18] J. Hao. “*A Faster Algorithm for Finding the Minimum Cut of a Graph*”. Unpublished manuscript, 1991.
- [19] D. R. Karger. “*Minimum cuts in near-linear time*”, Proceedings of the 28th annual ACM symposium on Theory of computing, 56-63, 1996.
- [20] D. R. Karger, C. Stein. “*An  $O(n^2)$  algorithm for minimum cuts*”, Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, 757-765, 1993.
- [21] D. R. Karger. “*A randomized fully polynomial approximation schene for the all terminal network reliability problem*”, In Proc. 27th Annual ACM Symposium on Theory of Computing, 11-17, 1995.
- [22] L. R. Ford, Jr. and D.R. Fulkerson. “*Maximal Flow Through a Network*”. *Canadian Journal of Math.*, 8: 399-404,1956.

- [23] M. Padberg and G. Rinaldi. “*An Efficient Algorithm for the Minimum Capacity Cut Problem*”, 1988.
- [24] N. D. Curet, J. D. Vinney, M. E. Gaston. “*An efficient network flow code for finding all minimum cost s-t cutsets*”, 2000.
- [25] D. L. Applegate and W. J. Cook. “*Personal communication*”, 1996.
- [26] M. Stoer and F. Wagner. “*A simple min-cut algorithm*”, Journal of the ACM, 44: 585-591, 1997.
- [27] Z. Wu and R. Leahy. “*An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation*”. IEEE Trans. on Pattern Analysis and Machine Intelligence, 15: 1101-1113, 1993.
- [28] J. Shi and J. Malik. “*Normalized cuts and image segmentation*”, IEEE Conf. on Computer Vision and Pattern Recognition, 731-737,1997.
- [29] G. W. Flake, R. E. Tarjan and K. Tsioutsoulis. “*Graph clustering and minimum cut trees*”, Internet Mathematics, 1: 355-378, 2004.
- [30] D. Gusfield. “*Very simple method for all pairs network flow analysis*”. SIAM J. Comput. 19: 143-155, 1990.
- [31] M. Brinkmeier. “*A Simple and Fast Min-Cut Algorithm*”. Theory Comput. Syst. 41: 369-380, 2007.

## ΒΙΟΓΡΑΦΙΚΟ

---

Η Αυγούστα Θεοδούλου γεννήθηκε το 1982 στην Κύπρο (Λάρνακα), όπου και ολοκλήρωσε τις λυκειακές της σπουδές το 2000. Το 2000 ξεκίνησε τις σπουδές της στο Τμήμα Πληροφορικής του Πανεπιστημίου Ιωαννίνων, τις οποίες και ολοκλήρωσε το 2005. Από τον Σεπτέμβριο του 2005 είναι μεταπτυχιακή φοιτήτρια του Τμήματος Πληροφορικής του Πανεπιστημίου Ιωαννίνων.