ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ,ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ

Προσομοίωση κυκλωμάτων με χρήση κατάτμησης γράφων για την επίλυση των γραμμικών συστημάτων

Circuit simulation with the use of graph partitioning for linear systems solving

Διπλωματική Εργασία

Κωνσταντάς Ι. Χρήστος

Επιβλέποντες Καθηγητές : Ευμορφόπουλος Νέστωρας
Επίκουρος Καθηγητής

Αντωνόπουλος Χρήστος
Επίκουρος Καθηγητής

Βόλος, Σεπτέμβριος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ,
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

Προσομοίωση κυκλωμάτων με χρήση κατάτμησης γράφων για την επίλυση των γραμμικών συστημάτων

Διπλωματική Εργασία

Κωνσταντάς Ι. Χρήστος

Επιβλέποντες : Ευμορφόπουλος Νέστωρας
Επίκουρος Καθηγητής

Χρήστος Δ. Αντωνόπουλος
Επίκουρος Καθηγητής

Εγκρίθηκε από την διμελή εξεταστική επιτροπή την 26 / 2014

....	....
Ν. Ευμορφόπουλος	Χ. Δ.Αντωνόπουλος
Επίκουρος Καθηγητής	Επίκουρος Καθηγητής

Διπλωματική Εργασία για την απόκτηση του Διπλώματος του Μηχανικού Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας, στα πλαίσια του Προγράμματος Προπτυχιακών Σπουδών του Τμήματος Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας.

……………………………..

Χρήστος Κωνσταντάς

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων Πανεπιστημίου Θεσσαλίας

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ,ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ

Προσομοίωση κυκλωμάτων με χρήση κατάτμησης γράφων για την επίλυση των γραμμικών συστημάτων

Circuit simulation with the use of graph partitioning for linear systems solving

Διπλωματική Εργασία

Κωνσταντάς Ι. Χρήστος

Επιβλέποντες Καθηγητές : Ευμορφόπουλος Νέστωρας
Επίκουρος Καθηγητής

Αντωνόπουλος Χρήστος
Επίκουρος Καθηγητής

Βόλος, Σεπτέμβριος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ,
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

Προσομοίωση κυκλωμάτων με χρήση κατάτμησης γράφων για την επίλυση των γραμμικών συστημάτων

Διπλωματική Εργασία

Κωνσταντάς Ι. Χρήστος

Επιβλέποντες : Ευμορφόπουλος Νέστωρας
Επίκουρος Καθηγητής

Χρήστος Δ. Αντωνόπουλος
Επίκουρος Καθηγητής

Εγκρίθηκε από την διμελή εξεταστική επιτροπή την 26 / 2014

....	....
Ν. Ευμορφόπουλος	Χ. Δ.Αντωνόπουλος
Επίκουρος Καθηγητής	Επίκουρος Καθηγητής

Διπλωματική Εργασία για την απόκτηση του Διπλώματος του Μηχανικού Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας, στα πλαίσια του Προγράμματος Προπτυχιακών Σπουδών του Τμήματος Μηχανικών Η/Υ, Τηλεπικοινωνιών και Δικτύων του Πανεπιστημίου Θεσσαλίας.

……………………………..

Χρήστος Κωνσταντάς

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων Πανεπιστημίου Θεσσαλίας

Στην οικογένεια & στους φίλους μου

Θα ήθελα να ευχαριστήσω θερμά όλου όσους συνέβαλαν στο να περατωθεί η συγκεκριμένη διπλωματική εργασία και ιδιαίτερα τους επιβλέποντες καθηγητές μου κ.Νέστωρα Ευμορφόπουλο και κ.Χρήστο Αντωνόπουλο οι οποίοι στάθηκαν δίπλα μου απο την αρχή της επιλογής του θέματος μέχρι και την παρουσίαση.
Επίσης θα ήθελα να ευχαριστήσω και όλους τους συναδέρφους που ήταν μαζί μου αυτό το διάστημα στο εργαστήριο το οποίο εργαζόμουν και μου παρείχαν άμεση στήριξη σε οτι χρειαζόμουν.
Τέλος, ενα πολύ μεγάλο ευχαριστώ θα ήθελα να προσφέρω στους φίλους και στην κοντινή μου οικογένεια καθώς στάθηκαν κοντά αυτές τις απαιτητικές μέρες της εργασίας μου για την διπλωματική εργασία.

Χρήστος Κωνσταντάς
Βόλος, 2014

Περιεχόμενα

Κατάλογος πινάκων

Κατάλογος σχημάτων

Κατάλογος Συντομογραφιών

CG Conjugate Gradient

CAD Computer Aided Design

MLRB Multilevel Recursive Bisection

API Application Programming Interface

CSR Compressed Storage Format

MNA Modified Nodal Analysis

NA Nodal Analysis

SPD Symetric Positive Definite

Περίληψη

Κατά τη σύγχρονη διαδικασία παραγωγής κυκλωμάτων έχουμε πλέον τη δυνατότητα να προσομοιώνουμε τη συμπεριφορά τους πριν την παραγωγή. Η προσομοίωση αυτή γίνεται με τη βοήθεια των σύγχρονων υπολογιστικών συστημάτων τα οποία είναι σε θέση να μας δώσουν μια αρκετά ακριβή συμπεριφορά του πραγματικού κυκλώματος.
Η συνεχής ανάγκη να παράγουμε κυκλώματα τα οποία θα έχουν ανοχή σε λάθη και η συμπεριφορά τους να είναι όπως επιθυμεί ο σχεδιαστής, οδήγησε στη ολοένα και πιο ακριβή μοντελοποίηση των κυκλωμάτων σε περιβάλλοντα προσομοίωσης. Μαζί όμως με την αυξανόμενη ακρίβεια έχουμε και την αυξανόμενη υπολογιστική ισχύ που χρειάζονται τα συστήματα να προσομοιώσουν τη συμπεριφορά. Αυτή είναι και η πρόκληση προς τους σημερινούς μηχανικούς οι οποίοι καλούνται να βρούν τεχνικές επίλυσης που ολοκληρώνονται σε λογικό χρόνο για πολύ μεγάλα συστήματα.
Βασικός παράγοντας στην προσομοίωση κυκλωμάτων είναι η επίλυση γραμμικών και μη συστημάτων σε ρεαλιστικό χρόνο. Τα σημερινά ρεαλιστικά συστήματα ξεκινάνε απο κάποιες εκατοντάδες χιλιάδες στοιχεία και φτάνουν εύκολα σε μερικά εκατομμύρια. Σε τέτοια συστήματα οι τεχνικές επίλυσης θα πρέπει να είναι όσο πιο αποδοτικές γίνεται απο θεωρητικής απόψεως αλλά και απο τη μεριά της υλοποίησης.

Κεφάλαιο 1

Εισαγωγή

1.1 Περιγραφή του προβλήματος

Τα τελευταία χρόνια υπάρχει μια ραγδαία αύξηση στον αριθμό των ηλεκτρικών και ηλεκτρονικών στοιχείων που χρησιμοποιούμε στις συσκευές μας απο τις μικρότερες και πιο απλές μέχρι τις μεγαλύτερες και πιο περίπλοκες. Το μέγεθος και η πολυπλοκότητα των συστημάτων αυξήθηκε τόσο πολύ που πλέον καθίσταται αδύνατο το να υπολογιστεί η συμπεριφορά ενός συστήματος χωρίς τη χρήση κάποιου υπολογιστικού συστήματος ή αλλιώς προσομοιωτή.

Οι προσομοιωτές αναλαμβάνουν να υπολογίσουν με βάση κάποιες εξισώσεις και μεταβλητές τη συμπεριφορά που θα έχει το σύστημά μας αν το κατασκευάσουμε. Όπως καταλαβαίνουμε, θα ήταν πολύ χρονοβόρο αλλά και με μεγάλο κόστος το να κατασκευάσουμε ένα σύστημα το οποίο τελικά δεν θα παρουσιάζει τη συμπεριφορά για το οποίο το κατασκευάσαμε. Για τον λόγο αυτό έχει γίνει πολύ μεγάλη έρευνα στο να μοντελοποιηθούν η συμπεριφορά του κάθε στοιχείου ξεχωριστά αλλά και η αλληλεπίδραση μεταξύ τους. Η μοντελοποίηση αυτή οδήγησε στα εργαλεία Computer Aided Design (CAD) τα οποία αναλαμβάνουν να προσφέρουν μια διεπαφή προς το χρήστη η οποία θα διευκολύνει τη μοντελοποίηση αυτή καθώς επίσης θα την κάνει προσιτή και στο ευρύ φάσμα των μηχανικών.

Όπως και στους περισσότερους τομείς έτσι και στον τομέα κατασκευής ηλεκτρικών και ηλεκτρονικών συστημάτων με τη βοήθεια των ηλεκτρονικών υπολογιστών, φτάσαμε στο σημείο όπου τα συστήματα που θέλουμε να προσομοιώσουμε να είναι τόσο περίπλοκα που ακόμη και τα υπολογιστικά συστήματα να χρειάζονται αρκετούς πόρους και χρόνο για να περατώσουν τη προσομοίωση. Έτσι λοιπόν τα τελευταία χρόνια εκτός από το να αναθέτουμε τα προβλήματα αυτά της προσομοίωσης σε όλο και πιο ισχυρά μηχανήματα, αρχίσαμε να σχεδιάζουμε πιο αποδοτικούς αλγορίθμους οι οποίοι μας δίνουν πιο γρήγορα αποτελέσματα.

Ακόμη όμως και σε αυτή την εποχή όπου μπορούμε να κατασκευάσουμε υπολογιστικά συστήματα με πολλές δυνατότητες και πολύ μικρό κόστος, υπάρχει η ανάγκη να εξελίξουμε τις μεθόδους που χρησιμοποιούμε κατά την προσομοίωση κυκλωμάτων. Η κλίμακα των προβλημάτων που καλούμαστε να λύσουμε στην καθημερινότητα δεν αλλάζει γραμμικά και έτσι πολύ συχνά φτάνουμε στο σημείο να μην μπορούμε να λύσουμε προβλήματα σε πραγματικό χρόνο. Η έρευνά μας σαν μηχανικοί στρέφεται πλέον στο να βρούμε ακόμη πιο αποδοτικές μεθόδους που αξιοποιούν στο έπακρο τις δυνατότητες των συστημάτων που χρησιμοποιούμε.

Στον τομέα της προσομοίωσης κυκλωμάτων μπορούμε πολύ γρήγορα να παρατηρήσουμε ότι ο βασικός παράγοντας πολυπλοκότητας και ο οποίος καταναλώνει και τους περισσότερους πόρους στο σύστημά μας είναι η επίλυση γραμμικών και μη συστημάτων που προκύπτουν κατά τη μοντελοποίηση των κυκλωμάτων.

1.2 Σκοπός της εργασίας

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι να μελετήσει μια ακόμη προσέγγιση στην επίλυση των γραμμικών συστημάτων. Όπως τονίσαμε, τα γραμμικά συστήματα που προκύπτουν αποτελούν και το πιο πολύπλοκο κομμάτι του συστήματος προσομοίωσης και για τον λόγο αυτό είναι ένας τομέας που οποιαδήποτε βελτίωση αντικατοπτρίζεται άμεσα στο σύστημα. Έτσι λοιπόν προσπαθήσαμε όχι μόνο να υλοποιήσουμε μια διαφορετική προσέγγιση για την επίλυση των συστημάτων αλλά να το κάνουμε και με τον πιο δυνατό αποτελεσματικό τρόπο.

Στόχος μας βέβαια δεν είναι μόνο να μειώσουμε το χρόνο εκτέλεσης αλλά όπως θα δούμε να το πετύχουμε χωρίς να θυσιάσουμε τίποτα απο την ακρίβεια της λύσης. Έτσι μέρος της διπλωματικής ήταν να παρατηρήσουμε και να κοστολογήσουμε με βάση τους πόρους που χρησιμοποιούμε αλλά και τον χρόνο που χρειάζεται το σύστημά μας για να περατώσει τη λειτουργία της προσομοίωσης. Δεν είναι ο στόχος μας μια βελτίωση στο χρόνο μόνο απο τη μεριά της υλοποίησης αλλά μια βελτίωση του χρόνου ή καλύτερα τον αριθμό επαναλήψεων που χρειάζεται το σύστημά μας να προσεγγίσει τη καλύτερη δυνατή λύση.

1.3 Διάρθρωση της διπλωματικής Εργασίας

Το Κεφάλαιο 2 μας δίνει μια θεωρητική βάση που χρειαζόμαστε για να κατανοήσουμε τον τρόπο με τον οποίο πραγματοποιείται η γραμμική επίλυση των συστημάτων.

Στο Κεφάλαιο 3 μπορούμε να δούμε όλη τη διαδικασία της κατάτμησης του γράφου που συμβολίζουμε το κύκλωμά μας, καθώς και μια πιο θεωρητική σκοπιά των αλγορίθμων που χρησιμοποιούνται.

Το Κεφάλαιο 4 αναλύει με αρκετά επεξηγηματικό τρόπο τη διαδικασία που ακολουθήσαμε κατά την υλοποίηση της εφαρμογής μας που πραγματοποιεί την προσομοίωση κυκλωμάτων.

Μετά και την περιγραφή της υλοποίησης, στο Κεφάλαιο 5 γίνεται μια αναφορά στα αποτελέσματα που λάβαμε δοκιμάζοντας διάφορα πειράματα πάνω στο σύστημά μας και καταγράψαμε τη συμπεριφορά του κάθε στοιχείου της εφαρμογής μας.

Τέλος η παρούσα διπλωματική εργασία ολοκληρώνεται με το Κεφάλαιο 6 όπου αναφέρουμε τη συνολική εικόνα που λάβαμε εργαζόμενοι πάνω στο πεδίο της διπλωματικής μας εργασίας.

Κεφάλαιο 2

Γραμμική επίλυση συστημάτων

2.1 Επίλυση γραμμικών συστημάτων

Η προσομοίωση κυκλωμάτων αναφέρεται στην προσέγγιση της συμπεριφοράς ενός συνόλου κυκλωματικών στοιχείων που έχουν κάποια αλληλεπίδραση μεταξύ τους. Η αναπαράσταση ενός κυκλώματος στον προσομοιωτή δεν είναι κάτι άλλο από ένα γραμμικό γραμμικό σύστημα. Συνεπώς, το πρόβλημα της προσομοίωσης ανάγεται τελικά σε ένα πρόβλημα επίλυσης γραμμικών συστημάτων.

Λόγω της παραπάνω παραδοχής καταλαβαίνουμε ότι πρέπει να δώσουμε τη μεγαλύτερη προσοχή και βάση μας στη εύρεση μεθόδων επίλυσης γραμμικών συστημάτων που να παρουσιάζουν διάφορα χαρακτηριστικά. Ένα βασικό χαρακτηριστικό που θα θέλαμε είναι η επίλυση να γίνεται με αποδοτικό τρόπο. Με τον όρο αποδοτικό αναφερόμαστε στο χρόνο που κάνει η διαδικασία της επίλυσης να βρει τη λύση σε σχέση με τον αριθμό των στοιχείων που υπάρχουν στον πίνακα. Ένα δεύτερο χαρακτηριστικό μπορεί να είναι η ακρίβεια της λύσης που θα βρούμε. Πολλές από τις μεθόδους προσεγγίζουν τη λύση και συνεπώς θα πρέπει να ορίσουμε το βάθος της σύγκλισης σε δεκαδικά ψηφία.

Επισκόπηση των μεθόδων επίλυσης γραμμικών συστημάτων

Υπάρχουν 2 βασικοί τρόποι επίλυσης γραμμικών συστημάτων. Ο πρώτος και πιο συνηθισμένος για μικρά συστήματα είναι ο άμεσος τρόπος ενώ για μεγάλα υπολογιστικά συστήματα έχουμε τους επαναληπτικούς τρόπους επίλυσης. Η διαφορά τους έγκειται στο γεγονός ότι οι άμεσοι τρόποι προσπαθούν με οποιοδήποτε κόστος να βρουν τη ακριβή λύση σε αντίθεση με τις επαναληπτικές μέθοδοι οι οποίες προσπαθούν να συγκλίνουν στη λύση. Αν το γραμμικό σύστημα είναι πολύ περίπλοκο είναι σχεδόν αδύνατο να χρησιμοποιήσουμε άμεσες μεθόδους για να βρούμε τη λύση. Ο λόγος είναι ότι λόγο της πολυπλοκότητας του αλγορίθμου ακόμη και σε πολύ ισχυρά μηχανήματα, ο χρόνος που μπορεί να πάρει για να βρεθεί η τελική λύση είναι απροσέγγιστος.

Επαναληπτικές μέθοδοι επίλυσης

Τα περισσότερα σύγχρονα υπολογιστικά συστήματα χρησιμοποιούν τις επαναληπτικές μεθόδους σαν μεθόδους επίλυσης γραμμικών συστημάτων γιατί η εύρεση μιας προσέγγισης της λύσης επιτυγχάνεται σε πολύ καλύτερο χρόνο. Πιο συγκεκριμένα με τον όρο ”επαναληπτικές” αναφερόμαστε σε μια ευρεία χρήση τεχνικών που προσπαθούν με διαδοχικές προσεγγίσεις να αποκτήσουν πιο ακριβές λύσεις του γραμμικού συστήματος σε κάθε βήμα [1].

Υπάρχουν 2 βασικές κατηγορίες επαναληπτικών μεθόδων: οι **Στατικές** και οι **Μη στατικές**. Οι στατικές μέθοδοι είναι πιο παλιές, ευκολότερες στη κατανόηση και την υλοποίηση αλλά όχι τόσο αποτελεσματικές. Οι μη-στατικές είναι σχετικά νεότερες και η ανάλυση τους είναι πιο σύνθετη αλλά με καλύτερα αποτελέσματα.

Με βάση τα παραπάνω εμείς επιλέξαμε για την υλοποίησή μας μια μη-στατική μέθοδο και συγκεκριμένα την Conjugate Gradient (CG).

Επαναληπτική μέθοδος σύγκλισης: CG

Το όνομα της παραπάνω τεχνικής απορρέει από το γεγονός οτι παράγει μια ακολουθία από συζυγή (conjugate) ή διαφορετικά ορθογώνια διανύσματα. Τα διανύσματα αυτά είναι τα υπόλοιπα από την κάθε επανάληψη. Υπάρχουν επίσης και οι κλίσεις (∇ gradients) από μια τετραγωνική συνάρτηση, η ελαχιστοποίηση της οποίας είναι αντίστοιχη με το να λύσουμε το γραμμικό σύστημα.Η CG είναι πολύ αποτελεσματική όταν ο πίνακας των συντελεστών είναι Symetric Positive Definite (SPD) γεγονός που μας εξυπηρετεί στη συγκεκριμένη ανάλυση αφού οι πίνακες στους οποίους αναφερόμαστε είναι εν γένει SPD. Να τονίσουμε ότι για τα κυκλώματα πίνακες SPD σημαίνει ότι περιέχουν μόνο στοιχεία της ομάδας 1 και πιο συγκεκριμένα μόνο αντιστάσεις, πυκνωτές και πηγές ρεύματος.

Η μέθοδος CG σε βάθος

Η CG μέθοδος είναι όπως τονίσαμε μια πολύ αποτελεσματική μέθοδος για SPD συστήματα. Είναι από τις παλαιότερες και τις πιο γνωστές μεθόδους. Η μέθοδος εξελίσσεται με το να παράγει διανύσματα σε κάθε επανάληψη(δηλαδή προσεγγίσεις της λύσης), το υπόλοιπο μεταξύ της προσέγγισης της λύσης και της πραγματικής λύσης και οδηγίες ανάγνωσης οι οποίες χρησιμοποιούνται για να ανανεώνουν τη προσέγγιση της λύσης και το υπόλοιπο. Συνεπώς, ακόμη και αν τα μέγεθος του γραμμικού συστήματος είναι πολύ μεγάλο, μόνο ένας πολύ μικρός αριθμός από διανύσματα χρειάζεται να διατηρείται στη μνήμη. Σε κάθε επανάληψη της μεθόδου, δυο εσωτερικά γινόμενα πραγματοποιούνται με σκοπό να υπολογίσουμε τιμές οι οποίες είναι προορισμένες να κάνουν τις ακολουθίες να ικανοποιούν κάποια κριτήρια ορθογωνιότητας. Σε ένα SPD σύστημα, οι συνθήκες αυτές υπονοούν ότι η απόσταση από την πραγματική λύση ελαχιστοποιείται σε κάποια νόρμα.

Τα διανύσματα επανάληψης $x^{(i)}$ ανανεώνονται σε κάθε επανάληψη πολλαπλασιάζοντάς τα με a_i από το διάνυσμα οδηγιών $p^{(i)}$:

$$x^{(i)} = x^{(i-1)} + a_i p^{(i)} \tag{2.1}$$

Algorithm 2 Ένας αφαιρετικός αλγόριθμος που δείχνει τη μαθηματική προσέγγιση της λύσης της CG

procedure ComputeCG(A)
    Υπολόγισε $r^{(0)} = b - Ax^{(0)}$ για μια αρχική προσέγγιση $x^{(0)}$
    for $i = 1, 2, ...$ **do**
        solve $Mz^{(i-1)} = r^{(i-1)}$
        $\rho_{i-1} = r^{(i-1)^T} z^{(i-1)}$
        if $i = 1$ **then**
            $p^{(1)} = z^{(0)}$
        else
            $\beta_{i-1} = \rho_{i-1}/\rho_{i-2}$
            $p^{(i)} = z^{(i-1)} + \beta_{i-1} p^{(i-1)}$
        endif
        $q^{(i)} = Ap^{(i)}$
        $a_i = \rho_{i-1}/p^{(i)^T} q^{(i)}$
        $x^{(i)} = x^{(i-1)} + a_i p^{(i)}$
        $r^{(i)} = r^{(i-1)} - a_i q^{(i)}$
        Έλενξε εαν συγκλίνει και συνέχισε αν είναι απαραίτητο
    end

Συνεπώς το διάνυσμα με τα υπόλοιπα $r^{(i)} = b - Ax^{(i)}$ ανανεώνονται σαν:

$$r^{(i)} = r^{(i-1)} - a_i q^{(i)} q^{(i)} = Ap^{(i)} \tag{2.2}$$

Η επιλογή $a = a_i = r^{(i-1)^T} z^{(i-1)}/p^{(i)^T} Ap^{(i)}$ ελαχιστοποιεί $r^{(i)^T} A^{-1} r^{(i)}$ από όλες τις επιλογές του a στη συνάρτηση 2.2

Το διάνυσμα που περιέχει τις οδηγίες εύρεσης, ανανεώνεται χρησιμοποιώντας το διάνυσμα που περιέχει τα υπόλοιπα

$$p^{(i)} = r^{(i)} + \beta_{i-1} p^{(i-1)} \tag{2.3}$$

όπου η επιλογή $\beta_i = (i)^T r^{(i)}/r^{(i-1)^T} r^{(i-1)}$ εξασφαλίζει ότι $p^{(i)}$ και $Ap^{(i-1)}$ ή ισοδύναμα $r^{(i)}$ και $r^{(i-1)}$ είναι ορθογώνια. Στην πραγματικότητα ένας μπορεί να δείξει οτι η επιλογή του β_i κάνει $p^{(i)}$ και $r^{(i)}$ ορθογώνια σε όλα τα προηγούμενα $Ap^{(j)}$ και $r^{(j)}$ αντίστοιχα.

2.2 Σύνδεση της επίλυσης των γραμμικών συστημάτων με την υλοποίηση μας

Από την παραπάνω παρουσίαση που κάναμε για τις μεθόδους επίλυσης αυτό που πρέπει να κρατήσουμε είναι η σύνδεση τους με την εφαρμογή μας. Πιο συγκεκριμένα πρέπει να δούμε ποια είναι τα σημεία τα οποία θα επικεντρώσουμε την προσοχή μας καθώς επίσης και τα σημεία τα οποία τελικώς θα μεταβληθούν.

Στη δικιά μας εφαρμογή επιλέξαμε να χρησιμοποιήσουμε και συνεπώς να υλοποιήσουμε τη μέθοδο της CG. Οι λόγοι αναφέρθηκαν σε προηγούμενη υποενότητα και το μόνο που έχουμε να προσθέσουμε είναι ότι τα συστήματα τα οποία θα ασχοληθούμε θα είναι εν γένει SPD και έτσι η παραπάνω μέθοδος είναι ιδανική.

Στον παραπάνω αλγόριθμο δεν έγινε καμιά απολύτως αναφορά στο βασικό σημείο της εφαρμογής μας το οποίο είναι η κατασκευή ενός προρυθμιστή. Πιο συγκεκριμένα ο προρυθμιστής στο παραπάνω αλγόριθμο 1 δεν είναι κάτι άλλο από τον πίνακα . Όπως βλέπουμε στη γραμμή 4 του αλγορίθμου γίνεται η επίλυση του συστήματος και προσπαθούμε να βρούμε μια προσέγγιση της λύσης καθώς και να παράγουμε το διάνυσμα που περιέχει τα υπόλοιπα.

Χρήση προρυθμιστή για τη γρηγορότερη επίλυση του συστήματος

Ο ρυθμός σύγκλισης από τις επαναληπτικές μεθόδους εξαρτάται από τις ιδιότητες των συχνοτήτων ενός πίνακα συντελεστών. Συνεπώς κάποιος μπορεί να προσπαθήσει να κάνει μετασχηματισμό ενός γραμμικού συστήματος σε κάποιο το οποίο θα είναι ισοδύναμο με την έννοια ότι θα έχει την ίδια λύση, αλλά θα παρουσιάζει πολύ καλύτερες ιδιότητες. Ένας προρυθμιστής είναι ένας πίνακας που πραγματοποιεί έναν τέτοιο μετασχηματισμό.

Για παράδειγμα, αν ένας πίνακας M προσεγγίζει τον πίνακα συντελεστών A με κάποιον τρόπο, το μετασχηματισμένο σύστημα

$$M^{-1}Ax = M^{-1}b \tag{2.4}$$

θα έχει την ίδια λύση με το πραγματικό σύστημα $Ax = b$ αλλά οι ιδιότητες των συχνοτήτων του πίνακα $M^{-1}A$ μπορεί να είναι πολύ καλύτερες.

Στη σχεδίαση ενός προρυθμιστή βρισκόμαστε ενώπιον δύο επιλογών. Η πρώτη έχει να κάνει με την εύρεση ενός πίνακα M που προσεγγίζει τον πίνακα A και για τον οποίο το σύστημα είναι ευκολότερο προς επίλυση από το αρχικό σύστημα που περιείχε τον A. Η δεύτερη επιλογή που έχουμε αναφέρεται στην εύρεση ενός πίνακα M που προσεγγίζει τον A^{-1} έτσι ώστε μόνο ένας πολλαπλασιασμός με το M να χρειάζεται. Οι περισσότεροι όπως και ο δικός μας αναφέρεται στη πρώτη επιλογή προρυθμιστών.

Είδη προρυθμιστών

Καθώς η διπλωματική μας είναι άρρηκτα συνδεδεμένη με τους προρυθμιστές, σε αυτό το σημείο θα αναφέρουμε κάποια παραδείγματα προρυθμιστών.

Προρυθμιστης Jacobi

Ο πιο απλός προρυθμιστής που μπορεί να υπάρξει είναι ο διαγώνιος του πίνακα:

$$m_{i,j} = \begin{cases} a_{i,j} & \text{if} i = j \\ 0 & \text{διαφορετικά} \end{cases}$$

Ο παραπάνω προρυθμιστής είναι γνωστός σαν Jacobi προρυθμιστής. Φυσικά τα αποτελέσματα σύγκλισης που δίνει δεν είναι και τα καλύτερα δυνατά αλλά είναι αρκετά εύκολος στην υλοποίηση του και δεν καταλαμβάνει και πολύ χώρο στη μνήμη.

Προρυθμιστης με χρήση κατάτμησης γράφων

Στην υλοποίηση μας κατά την παρούσα διπλωματική θα εξετάσουμε την δημιουργία ενός διαφορετικού προρυθμιστή. Εκτός της κατασκευής του προρυθμιστή θα εξετάσουμε την απόδοση που έχει στο σύστημά μας με βασικό στόχο να βρούμε αν η λύση μας συγκλίνει πιο γρήγορα από λύσεις που περιλαμβάνουν διαφορετικούς προρυθμιστές. Φυσικά εκτός από τον αριθμό των βημάτων που απαιτούνται για τη σύγκλιση θα ελέγξουμε και τους πόρους που χρειάζεται ο συγκεκριμένος προρυθμιστής να κατασκευαστεί σε σχέση με άλλους.

Η ιδέα πίσω από τον προρυθμιστή είναι ότι προσπαθούμε να κατασκευάσουμε έναν πίνακα ο οποίος να προσεγγίζει όσο το δυνατόν καλύτερα το αρχικό σύστημα με αποτέλεσμα η αρχική λύση που θα βρούμε να είναι όσο το δυνατόν πιο κοντά στη λύση που ψάχνουμε. Φυσικά αυτό δεν θα είχε πολύ νόημα αν λύναμε το προσεγγιστικό σύστημα όπως και το αρχικό γιατί δεν θα καταφέρναμε κάτι. Σε αυτό το σημείο είναι που χρησιμοποιούμε τη θεωρία γράφων για καταφέρουμε να λύσουμε πιο αποδοτικά το σύστημά μας [8].

Με τη χρήση γράφων μπορούμε να χωρίσουμε το αρχικό μεγάλο σύστημα που απαιτεί πολλούς πόρους και επαναλήψεις για να λυθεί, σε πολλά μικρότερα τα οποία λύνονται με πιο απλό και γρήγορο τρόπο. Για να γίνει το παραπάνω πρέπει φυσικά να μετατρέψουμε τον πίνακα σε γράφο και να πραγματοποιήσουμε την κατάτμηση.

Έχοντας στη διάθεσή μας πολλά μικρά υποσυστήματα μπορούμε πλέον να εφαρμόσουμε οποιαδήποτε αναλυτική ή επαναληπτική μέθοδο για να τα λύσουμε και να συγκεντρώσουμε το αποτέλεσμα της λύσης. Πριν γίνει η επίλυση το μόνο που πρέπει να κάνουμε είναι να αφαιρέσουμε οποιαδήποτε σύνδεση μεταξύ των επιμέρους υποσυστημάτων έτσι ώστε να είμαστε σίγουροι ότι το κάθε υποσύστημα είναι ξεχωριστό και ανεξάρτητο.

Μετά το πέρας της επίλυσης έχουμε στα χέρια μας ένα δεξί μέλος το οποίο είναι πολύ κοντά στην πραγματική λύση και έτσι υποθέτουμε ότι θα συγκλίνει πολύ γρήγορα στην πραγματική λύση.

Εάν κάνουμε μια πρώιμη εκτίμηση του κόστους παραγωγής του προρυθμιστή μπορούμε πολύ γρήγορα να δούμε οτι σίγουρα είναι πολύ πιο απαιτητική η παραγωγή του συγκεκριμένου προρυθμιστή απο τον προηγούμενο Jacobi. Σε βάθος χρόνου όμως οι επαναλήψεις που θα χρειαστούν για να επιτευχθεί η τελική λύση περιμένουμε να είναι πολύ λιγότερες. Έτσι μπορεί να έχουμε επένδυση πόρων στη κατασκευή του προρυθμιστή, αλλά αυτή η επένδυση περιμένουμε να αποδώσει αργότερα στο συνολικό χρόνο.

Κεφάλαιο 3

Κατάτμηση γράφου

3.1 Εισαγωγή στην κατάτμηση γράφου

Η κατάτμηση γράφου αποτελεί την διαδικασία κατά την οποία λαμβάνουμε σαν είσοδο έναν γράφο σε πλήρη μορφή και προσπαθούμε να σχηματίσουμε υπογράφους χωρίς όμως να διασπάσουμε την συνοχή του αρχικού γράφου. Γενικότερα ο βασικός σκοπός της κατάτμησης είναι να σχηματίσουμε ομάδες απο στοιχεία τα οποία ομοιάζουν μεταξύ τους με βάση έναν παράγοντα που θα θέσουμε. Στη γενική περίπτωση της κατάτμησης γράφου, το αποτέλεσμα που θα θέλουμε να επιτύχουμε μπορεί να διαφέρει απο εφαρμογή σε εφαρμογή.

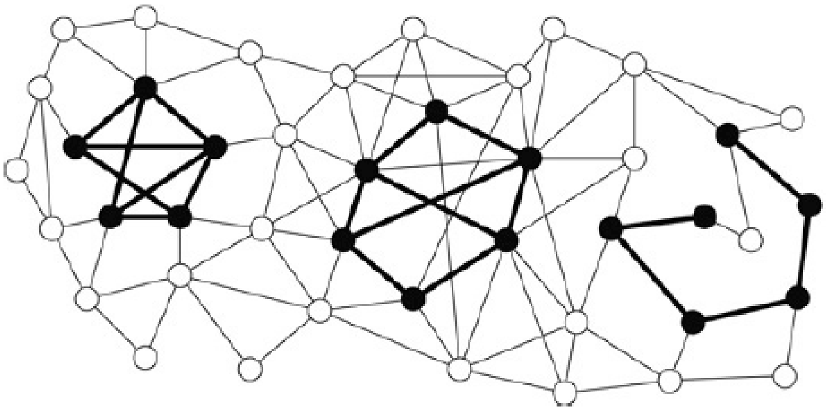
Στη συνέχεια της διπλωματικής εργασίας θα αναφερθούμε πολλές φορές στον όρο γράφο και για τον λόγο αυτό καλό θα ήταν να δώσουμε έναν σύντομο ορισμό. Γράφος είναι μια δομή η οποία αποτελείται απο ένα σύνολο κορυφών ή αλλιώς αποκαλούμενων και ως κόμβων και ένα σύνολο ακμών οι οποίες είναι συνδέσεις μεταξύ ζευγάρια κορυφών. Σκοπός της κατάτμησης γράφου είναι να χωρίσουμε τις κορυφές σε ομάδες. Οι ομάδες που θα σχηματιστούν θα πρέπει να είναι αυστηρά συνδεδεμένες υπο την έννοια οτι κάθε κόμβος θα πρέπει να είναι συνδεδεμένος με κάποιον άλλο κόμβο της ίδιας ομάδας. [7]

Στόχοι της κατάτμησης

Όπως τονίσαμε και παραπάνω, κάθε κατάτμηση που γίνεται σε έναν γράφο θα πρέπει να έχει μια μετρική η οποία θα καθορίσει και το τελικό αποτέλεσμα της κατάτμησης. Στην περίπτωση της δικιάς μας υλοποίησης ο στόχος που θέλαμε να επιτύχουμε απο την κατάτμηση είναι ο παρακάτω:

Βασικός στόχος που βάζουμε κατα την κατάτμηση γράφου που πραγματοποιούμε είναι η δημιουργία συστάδων κόμβων με υψηλή εσωτερική (intra) και χαμηλή εξωτερική (inter) συνδεσιμότητα

Ολόκληρη η ανάλυση που θα κάνουμε στη συνέχεια θα έχει σαν βασικό γνώμονα τον παραπάνω στόχο. Ο στόχος που θέσαμε δεν αποτελεί μια τυχαία σύμβαση αλλά μια ανάγκη η οποία έχει να κάνει με την μετέπειτα πορεία της επίλυσης του συστήματος. Πιο συγκεκριμένα όπως θα δούμε και σε μεγαλύτερο βάθος στη συνέχεια, όσο μεγαλύτερη είναι η προσέγγιση του παραπάνω στόχου, τόσο πιο ακριβής θα είναι και η προσέγγιση της λύσης που θα βρούμε αρχικά και συνεπώς θα είναι και πιο γρήγορη η επίλυση του τελικού συστήματος.



Σχήμα 3.1: Μια τυπική κατάτμηση γράφου. Φαίνονται οι συστάδες απο κόμβους που έχουν σχηματιστεί μετά την ανάλυση που έγινε

Εκτός του βασικού στόχου που θέσαμε παραπάνω, υπάρχουν και άλλοι παράμετροι που διαδραματίζουν βασικό ρόλο κατά την κατάτμηση του γράφου. Μια επίσης σημαντική παράμετρος είναι και η παρακάτω:

Επιπλέον στόχος της κατάτμησης μας είναι να δημιουργήσουμε συστάδες ή κοινότητες κόμβων [3, 6] οι οποίες θα είναι σε μεγέθη συγκρίσιμα μεταξύ τους και με κοντινό αριθμό κόμβων μεταξύ τους.

Ο παραπάνω στόχος αναφέρεται περισσότερο στη υλοποίηση της υλοποίησης μας η οποία βασίζεται στο γεγονός ότι οι υπογράφοι που σχηματίζονται είναι σε παρόμοια μεγέθη και παρόμοια πολυπλοκότητα. Σε περίπτωση που δεν λαμβάναμε υπ' όψιν μας τον παραπάνω παράγοντα υπήρχε περίπτωση να σχηματιστούν υπογράφοι ανομοιογενείς και τελικά ο βασικός στόχος της κατάτμησης της πολυπλοκότητας τους να αποτύγχανε.

3.2 Αλγόριθμοι κατάτμησης

Υπάρχει μια πληθώρα αλγορίθμων οι οποίοι πραγματοποιούν κατάτμηση γράφου. Εμείς όμως θέλουμε όπως τονίσαμε το αποτέλεσμα της κατάτμησης να οδηγήσει στην καλύτερη δυνατή λύση και να ακολουθεί τους στόχους που έχουμε θέσει. Υπάρχουν πολλές μέθοδοι για την κατάτμηση γράφου που βασίζονται σε διαφορετικές τεχνικές. Το αποτέλεσμα και τα πλεονεκτήματα του κάθε αλγορίθμου διαφέρουν και για τον λόγο αυτό θα πρέπει να είμαστε πολύ προσεκτικοί στην επιλογή του αλγορίθμου που θα χρησιμοποιήσουμε.

Ευριστικοί αλγόριθμοι κατάτμησης

Μια μεγάλη υποκατηγορία των αλγορίθμων κατάτμησης αποτελούν οι αλγόριθμοι οι οποίοι βασίζονται σε ευριστικές μεθόδους. Το βασικό χαρακτηριστικό των ευριστικών αλγορίθμων είναι ότι βρίσκουν μια λύση πιο γρήγορα από τους αναλυτικούς αλγορίθμους. Ο λόγος της ταχύτητας των αλγορίθμων αυτών είναι ότι μπορεί να μην βρουν μια απόλυτη λύση αλλά μια προσέγγιση της πραγματικής λύσης με μια ανοχή που θα καθορίσουμε εμείς. Οι αλγόριθμοι αυτοί είναι απαραίτητοι σε προβλήματα των οποίων η αναλυτική λύση είναι τόσο περίπλοκη και απαιτητική που δεν μπορεί να περατωθεί σε πραγματικό χρόνο.

Οι ευριστικοί αλγόριθμοι εκτός των πολλών πλεονεκτημάτων που παρέχουν σε θέματα ταχύτητας έχουν και πολλούς παράγοντες που τους καθιστούν ασταθείς και μη χρησιμοποιήσιμους σε όλα τα είδη των προβλημάτων. Πολύ συνοπτικά θα αναφέρουμε κάποια από τα μειονεκτήματα τους έτσι ώστε να βασίσουμε και την τελική μας απόφαση στο θέμα της επιλογής ενός κατάλληλου αλγορίθμου κατάτμησης.

Ενα πρώτο αρνητικό χαρακτηριστικό των παραπάνω ευριστικών αλγορίθμων είναι ότι εξαρτάται το τελικό αποτέλεσμά τους και η απόδοση της λύσης από την αρχική συνθήκη που θα δώσουμε ή από την αρχική επιλογή των σημείων έναρξης του αλγορίθμου. Οι αλγόριθμοι της παραπάνω κατηγορίας, βασίζονται στην επιλογή ενός σημείου έναρξης του αλγορίθμου (για παράδειγμα στην κατάτμηση γράφου ξεκινάμε τον αλγόριθμο από κάποια αρχικά ζεύγη κόμβων). Το παραπάνω γεγονός όπως μπορούμε να καταλάβουμε διαισθητικά μπορεί να προκαλέσει πολλά προβλήματα στο τελικό αποτέλεσμα της λύσης μας καθώς θα λαμβάνουμε διαφορετική απόδοση η οποία θα μεταβάλλεται βάση της αρχικής (τυχαίας πολλές φορές) επιλογής αρχικού σημείου.

Ενα επίσης χαρακτηριστικό των αλγορίθμων που δρούν με ευριστικό τρόπο είναι οτι υπάρχει περίπτωση να παγιδευθούν λόγω της τοπικότητας που παρουσιάζουν και να μας επιστρέψουν λύσεις οι οποίες απέχουν απο την ιδανική λύση που μπορούμε να λάβουμε. Η εύρεση της καλύτερης δυνατής λύσης εξαρτάται όπως τονίσαμε και παραπάνω στην επιλογή των αρχικών συνθηκών ή σημείων εκκίνησης του αλγορίθμου. Συνεπώς, για την δικιά μας εφαρμογή, κατά την οποία η κατάτμηση γράφου είναι σημαντικότατη, δεν μπορούμε να έχουμε έναν τέτοιο παράγοντα.

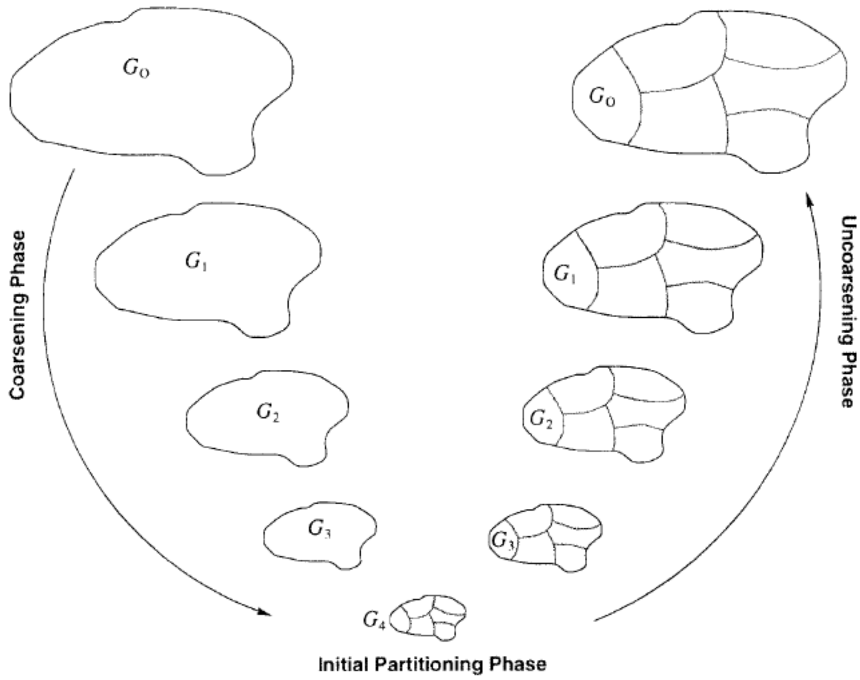
Σε αυτό το σημείο θα ήταν καλό να τονίσουμε οτι τα παραπάνω αρνητικά χαρακτηριστικά μπορούν να εμφανιστούν και σε άλλους αλγορίθμους οι οποίοι δεν είναι απόλυτα ευριστικοί αλλά σε πολύ μικρότερη κλίμακα. Έτσι είναι σημαντικό να αναφέρουμε έναν τρόπο αντιμετώπισης τέτοιων φαινομένων. Κάτι το οποίο μπορούμε να κάνουμε είναι ένα είδος επαναληπτικής επίλυσης του αρχικού προβλήματος με διαφορετικές αρχικές συνθήκες. Με τον τρόπο αυτό μπορούμε να λάβουμε διαφορετικά αποτελέσματα απο την κάθε επίλυση και να κρατήσουμε αυτή που παρουσιάζει την καλύτερη λύση.

Η παραπάνω επαναληπτική προσέγγιση δεν μπορεί να μας διασφαλίσει οτι θα λάβουμε την καλύτερη δυνατή λύση. Επίσης αποτελεί μια πολύ σημαντική επένδυση του των πόρων του συστήματος καθώς και θα αυξήσει και τον χρόνο εκτέλεσης κατά έναν σημαντικό παράγοντα.

Μέθοδοι πολλαπλών επιπέδων κατάτμησης

Εκτός των ευριστικών μεθόδων μια άλλη πολύ σημαντική κατηγορία κατάτμησης αποτελεί και η κατάτμηση κατά επίπεδα [4, **?**]. Σε αυτή τη προσέγγιση μειώνουμε το μέγεθος του γράφου με το συνδυάζουμε σταδιακά κοντινούς (coarsening) κόμβους οδηγώντας τον γράφο σε μια πιο τραχιά(coarse) μορφή. Όταν φτάσουμε σε ένα επίπεδο στο οποίο δεν μπορούμε να συνδυάσουμε άλλους κόμβους, πραγματοποιούμε τον χωρισμό του γράφου.

Όταν φτάσουμε στο σημείο όπου ο γράφος δεν μπορεί να συρρικνωθεί άλλο, μπορούμε να επιλέξουμε την μέθοδο που θα χρησιμοποιήσουμε για την κατάτμηση του γράφου. Γενικά θα πούμε οτι οι μέθοδοι που μπορούν να χρησιμοποιηθούν είναι δυο. Η πρώτη αφορά τη πιο απλή έκδοση του αλγορίθμου και ονομάζεται Multilevel Recursive Bisection (MLRB). Η μέθοδος αυτή σπάει το κάθε γράφο σε δυο υπογράφους. Σε περίπτωση που χρειαζόμαστε κατάτμηση για παραπάνω απο δυο υπογράφους θα πρέπει να συνεχίσουμε τη διαδικασία για το κάθε υπογράφο που έχει παραχθεί, παράγοντας και άλλους υπογράφους με ακόμη μικρότερο μέγεθος. Η παραπάνω διαδικασία μπορεί να συνεχιστεί μέχρι να προσεγγίσουμε ένα επιθυμητό αποτέλεσμα αριθμού υπογράφων που χρειαζόμαστε.



Σχήμα 3.2: Οι πολλαπλές φάσεις ενός πολυ-επιπεδικού k-δρόμων(k-way) αλγόριθμος κατάτμησης. Κατά τη διαδικασία της συμπύκνωσης(coarsening) το μέγεθος του γράφου μειώνεται ακολουθιακά. Στην αρχική φάση της κατάτμησης, μια κατάτμηση κ-δρόμων του μικρότερου γράφου κατασκευάζεται(στην συγκεκριμένη περίπτωση έχουμε 6-δρόμων κατάτμηση. Κατά τη διαδικασία της αποσυμπίκωνσης(uncoarsening) του γράφου, η κατάτμηση βελτιστοποιείται(refinement) καθώς γίνεται προβολή σε μεγαλύτερο γράφο

Μια διαφορετική μέθοδο που μπορούμε να χρησιμοποιήσουμε είναι η άμεση κατάτμηση κ-δρόμων (k-way) του συμπυκνωμένου γράφου χωρίς να κάνουμε επαναληπτικές κατατμήσεις. Η μέθοδος αυτή έχει το βασικό πλεονέκτημα έναντι της MLRB στο θέμα της ταχύτητας κατάτμησης καθώς με μια μοναδική συμπύκνωση(coarsening) του γράφου επιτυγχάνουμε k διαφορετικούς υπογράφους. Φυσικά, η παραπάνω μέθοδος είναι αρκετά πιο πολύπλοκη κατά την αντίστροφη διαδικασίας αποσυμπίκνωσης(uncoarsening) του γράφου, η πολυπλοκότητα είναι αρκετά μεγαλύτερη απο την προηγούμενη μέθοδο όπου είχαμε μόνο δυο διαφορετικούς υπογράφους.

Σε αυτό το σημείο να τονίσουμε οτι το εργαλείο το οποίο θα χρησιμοποιήσουμε για να κάνουμε την κατάτμηση του γράφου χρησιμοποιεί την παραπάνω γενική μέθοδο κατάτμησης γράφου με χρήση πολλών επιπέδων κατάτμησης.

3.3 Εργαλείο METIS

Στην παρούσα διπλωματική εργασία χρησιμοποιήσαμε το εργαλείο METIS για να πραγματοποιήσουμε την κατάτμηση γράφου. Το εργαλείο αυτό επιλέχθηκε για λόγους που θα αναπτύξουμε στη συνέχεια.

Το εργαλείο METIS αποτελεί μια ολοκληρωμένη βιβλιοθήκη η οποία έχει υλοποιηθεί απο το Karypis Lab και αποτελεί μια απο τις πιο βασικές επιλογές στον τομέα της κατάτμησης γράφου. Σαν εργαλείο παρέχει διάφορες επιλογές κατάτμησης που έχουν υλοποιηθεί με πολύ αποδοτικό τρόπο με αποτέλεσμα να προσφέρουν μια ελκυστική επιλογή για κάθε είδους εφαρμογή που χρειάζεται κατάτμηση γράφου.

Ενσωμάτωση εργαλείου

Το εργαλείο METIS παρέχει πολλές επιλογές προς τη χρήση του, δίνοντας τη δυνατότητα στο χρήστη να επιλέξει. Ένα απο τα βασικά πλεονεκτήματα του εργαλείου είναι οτι προσφέρει έτοιμες εφαρμογές οι οποίες λαμβάνουν ένα αρχείο και πραγματοποιούν όλες τις άλλες λειτουργίες με τελικό βήμα την παραγωγή ενός αρχείου που περιέχει την κατάτμηση. Οι εφαρμογές οι οποίες είναι υλοποιημένες, εκτός της διευκόλυνσης που παρέχουν στο χρήστη, παράγουν και πολύ πληροφορία που έχει να κάνει με τον χρόνο εκτέλεσης καθώς και με την ποιότητα της κατάτμησης.

Οι εφαρμογές οι οποίες προσφέρονται είναι οι εξής: κατάτμηση γράφου, κατάτμηση πλέγματος(mesh), πραγματοποίηση μείωσης της τάξης των αραιών πινάκων καθώς και μετατροπή απο πλέγμα σε γράφο κατάλληλο για χρήση απο το METIS.

Στην περίπτωση της παρούσας διπλωματικής επιλέξαμε να χρησιμοποιήσουμε τη βιβλιοθήκη που παρέχει το εργαλείο για την κατασκευή της εφαρμογής μας. Η απόφαση αυτή μας οδήγησε στην κατασκευή μιας πιο ολοκληρωμένης εφαρμογής η οποία λαμβάνει τα δεδομένα άμεσα χωρίς να υπάρχει ενδιάμεση αποθήκευση σε αρχεία. Φυσικά για να επιτευχθεί μια τέτοια ενσωμάτωση της βιβλιοθήκης σε δικιά μας εφαρμογή, χρειάστηκε να μελετηθεί πολύ προσεκτικά το πρότυπο το οποίο ακολουθεί η βιβλιοθήκη για τις δομές δεδομένων καθώς και τα βήματα αρχικοποίησης τα οποία χρειάζονται πριν την εκτέλεση της κατάτμησης καθαυτής.

Περιγραφή του METIS API

Όπως τονίσαμε και παραπάνω το εργαλείο παρέχει μια ολοκληρωμένη βιβλιοθήκη η οποία μπορεί να ενσωματωθεί σε οποιαδήποτε εφαρμογή.

Για την εισαγωγή της βιβλιοθήκης το πρώτο βήμα που πρέπει να κάνουμε είναι η αναφορά στο header file της βιβλιοθήκης.Το αρχείο header το οποίο πρέπει να κάνουμε εισαγωγή(include) στον κώδικά μας είναι η metis.h. Το αρχείο αυτό περιέχει τις δηλώσεις των συναρτήσεων που μπορεί να χρησιμοποιήσει ο χρήστης για να κάνει τις λειτουργίες που περιγράψαμε πιο πάνω καθώς και ορίζει κάποιους βασικούς τύπους τους οποίους μπορούμε να προσαρμόσουμε εύκολα για διαφορετικά συστήματα των 32 ή 64 bit.

Διαφορετικές επιλογές κατάτμησης

Το εργαλείο METIS παρέχει πολλές και διάφορες επιλογές που έχουν να κάνουν με την κατάτμηση του γράφου και το αποτέλεσμα που θέλουμε να πετύχουμε. Να τονίσουμε ότι κάθε διαφορετική εφαρμογή μπορεί να έχει διαφορετικούς στόχους που μπορεί να θέλει να επιτύχει η κατάτμηση και να ταιριάζουν σε αυτή.

Αρκετός χρόνος επενδύθηκε σε αυτό το σημείο έτσι ώστε να βρεθούν οι κατάλληλες επιλογές για τη δικιά μας εφαρμογή, τις οποίες θα τις παραθέσουμε στη συνέχεια. Η διαδικασία που ακολουθήθηκε εκτός από διαισθητική για την επιλογή των διαφόρων τιμών των παραμέτρων με βάση τον στόχο που θέλουμε να επιτύχουμε, ήταν και πειραματική για να έχουμε το καλύτερο δυνατό αποτέλεσμα.

Το METIS χρησιμοποιεί ένα διάνυσμα το οποίο παρέχει όλες τις επιλογές που κάνει ο χρήστης και θα επηρεάσουν το τελικό αποτέλεσμα της κατάτμησης. Στη συνέχεια παρουσιάζουμε συνοπτικά τις διάφορες επιλογές που υπάρχουν με μία σύντομη περιγραφή καθώς και με την τελική επιλογή που κάναμε στο πρόγραμμά μας.

Η βασική επιλογή που παρέχει το σύστημα έχει να κάνει με το είδος της μεθόδου που θα ακολουθήσει για την κατάτμηση του γράφου. Όπως τονίσαμε και παραπάνω έχουμε να επιλέξουμε μεταξύ δυο διαφορετικών μεθόδων της MLRB και της αναδρομικής κατάτμησης 2 τμημάτων. Η επιλογή μας έχει να κάνει κυρίως με το αποτέλεσμα και όχι με τον χρόνο εκτέλεσης. Επιλέγουμε λοιπόν να θυσιάσουμε λίγο απο το χρόνο εκτέλεσης για ένα καλύτερο μελλοντικό αποτέλεσμα.

Μια ακόμη σημαντική επιλογή που μας αφορά έχει να κάνει με την προτεραιότητα που θα δώσουμε στη μείωση των ακμών που υπάρχουν μεταξύ των διαφορετικών κατατμήσεων έναντι της συνολικής μείωσης των ακμών στο εσωτερικό. Σε προηγούμενη αναφορά μας δώσαμε το πόσο σημαντικό είναι να έχουμε όσο το δυνατόν μικρή επικοινωνία μεταξύ των κατατμήσεων και έτσι η επιλογή είναι εύκολη.

.

Δομή του γράφου

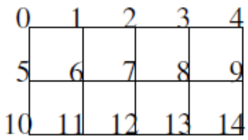
Κατά την ενσωμάτωση του γράφου θα πρέπει να προσαρμόσουμε τα δεδομένα τα οποία έχουμε σε μορφή MNA πίνακα στη μορφή που τα ζητάει το εργαλείο. Ένας απο τους βασικούς λόγους που επιλέξαμε να χρησιμοποιήσουμε την κατάτμηση σαν τεχνική ήταν οτι δεν χρειάζεται να κάνουμε μεγάλο πλήθος αλλαγών στη δομή που ήδη έχουμε.

Όλες οι συναρτήσεις οι οποίες αναφέρονται στη κατάτμηση γράφου ή και στη διάταξη αραιών πινάκων λαμβάνουν σαν είσοδο μια δομή που περιγράφει τα γειτονικά στοιχεία του κάθε κόμβου(adjacency structure) καθώς και τα βάρη των κόμβων και ακμών αν υπάρχουν.

Η δομή γειτνίασης ενός γράφου αποθηκεύεται σε μορφή CSR. Η CSR μορφή χρησιμοποιείται ευρέως για την αποθήκευση αραιών πινάκων. Στη μορφή αυτή ο πίνακας γειτνίασης μιας δομής ενός γράφου με n κορυφές και m ακμές αντιποσοπεύεται απο δύο πίνακες xadj και adjncy. Ο xadj πίνακας έχει μέγεθος n + 1 ενώ ο πίνακας adjncy έχει μέγεθος 2m(για κάθε ακμή μεταξύ δύο κόμβων v και u αποθηκεύουμε και τις δύο $(\mathrm{v,u})$ και $(\mathrm{u,v})$).

Η δομή γειτνίασης του γράφου αποθηκεύεται με τον παρακάτω τρόπο. Υποθέτοντας ότι η αρίθμηση των κορυφών ξεκινάει από το 0, τότε η λίστα γειτνίασης της κορυφής i αποθηκεύεται στον πίνακα adjncy ξεκινώντας από το δείκτη xadj[i] και τελειώνει (χωρίς να το περιλαμβάνει) στο δείκτη xadj[i+1]. Αυτό σημαίνει ότι για κάθε κόμβο i, η λίστα γειτνίασης αποθηκεύεται σε διαδοχικές θέσεις στον πίνακα adjncy και ο πίνακας xadj χρησιμοποιείται να δείχνει που ξεκινάει και που τελειώνει. Θα παραθέσουμε και ένα γράφημα μαζί με την αναπαράστασή του.

Να τονίσουμε σε αυτό το σημείο ότι η διάταξη του αραιού πίνακα CSR είναι γενικής μορφής η



Σχήμα 3.3: Ένα απλό γράφημα

xadj	0 2 5 8 11 13 16 20 24 28 31 33 36 39 42 44
adjncy	1 5 0 2 6 1 3 7 2 4 8 3 9 0 6 10 1 5 7 11 2 6 8 12 3 7 9 13 4 8 14 5 11 6 10 12 7 11 13 8 12 14 9 13

Σχήμα 3.4: Η αναπαράσταση του γραφήματος σε μορφή CSR

οποία όπως τονίσαμε χρησιμοποιείται και από τις περισσότερες εφαρμογές που εμπεριέχουν αραιούς πίνακες.

Σύνδεση του MNA πίνακα με το εργαλείο METIS

Ο προσομοιωτής μας χρησιμοποιεί σαν βασική δομή αποθήκευσης τον τύπο των αραιών πινάκων της μορφής **CRS!** (**CRS!**). Συνεπώς, η χρήση των διάφορων συναρτήσεων του εργαλείου METIS μπορούν να γίνουν χωρίς μεγάλες διαφορές στις δομές δεδομένων. Φυσικά όμως όπως θα επισημάνουμε στη συνέχεια πρέπει να κάνουμε κάποιες διαφορές στα δεδομένα που αποθηκεύουμε στους αραιούς πίνακες.

Σε αυτό το σημείο πρέπει να επισημάνουμε ότι ο τύπος δεδομένων που χρησιμοποιεί και παρέχει το εργαλείο METIS για την αποθήκευση των βαρών των ακμών και των κορυφών είναι ακέραιοι αριθμοί(integers). Το παραπάνω γεγονός προκαλεί διάφορα προβλήματα στην υλοποίησή μας. Όπως θα δούμε και παρακάτω, οι τιμές που λαμβάνουμε κατά τη δημιουργία του MNA πίνακα είναι τύπου κινητής υποδιαστολής(double).

Τα δεδομένα που έχουμε από την κατασκευή του MNA γίνονται άμεση αντιστοίχηση με το είδος των δεδομένων που ζητάει το εργαλείο METIS για την αναπαράσταση ενός γράφου. Πιο συγκεκριμένα θεωρούμε οτι ο MNA πίνακας δεν είναι τίποτα άλλο απο μια αναπαράσταση ενός γράφου όπου οι ακμές είναι τα στοιχεία ενός κυκλώματος ενώ οι κορυφές οι κόμβοι ενός κυκλώματος. Έτσι μπορούμε πολύ εύκολα να χρησιμοποιήσουμε την ίδια αναπαράσταση.

Το πρόβλημα που προέκυπτε ήταν οτι υπάρχει η διαφορά μεταξύ των τύπων δεδομένων του εργαλείου METIS με τους αντίστοιχους τύπους δεδομένων του **MNA!** (**MNA!**). Έτσι χρειάζεται να αφιερώσουμε κάποιον από το χρόνο του συστήματος για να κάνουμε τις απαραίτητες μετατροπές. Σε αυτό το σημείο θα αναφέρουμε πολύ συνοπτικά τις αλλαγές που πρέπει να γίνουν καθώς γίνεται εκτεταμένη αναφορά στο επόμενο κεφάλαιο.

Η βασική αλλαγή που πρέπει να κάνουμε είναι να μετατρέψουμε τα βάρη των κορυφών από τύπου κινητής υποδιαστολής(double) σε τύπο ακεραίου(integer) για να μπορεί να υποστηριχθεί απο το εργαλείο. Επίσης κάτι άλλο που πρέπει να κάνουμε είναι να θέσουμε μηδέν στα στοιχεία της διαγωνίου καθώς αντιπροσωπεύουν ακμές προς τον εαυτό τους κάτι το οποίο δεν ισχύει και θα προκαλούσε προβλήματα στο γράφημά μας.

Κεφάλαιο 4

Υλοποίηση της προσομοίωσης

4.1 Κατασκευή του προσομοιωτή

Η υλοποίησή μας ξεκινάει από το σημείο όπου λαμβάνουμε ένα αρχείο το οποίο περιγράφει τη κυκλωματική συνδεσμολογία με κάποιο συγκεκριμένο πρότυπο το οποίο είναι κοινό για τις περισσότερες εφαρμογές προσομοίωσης κυκλωμάτων.

Το κάθε στοιχείο έχει το δικό του κωδικό όνομα καθώς και μια συγκεκριμένη μορφή με την οποία ορίζουμε τις διάφορες παραμέτρους του στοιχείο. Όπως γνωρίζουμε υπάρχουν πολλά είδη κυκλωματικών στοιχείων τα οποία πρέπει να συμπεριλάβουμε στον αναλυτή μας.

Ανάλυση του αρχείου περιγραφής του κυκλώματος

Όπως τονίσαμε το πρώτο στάδιο της ανάλυσης είναι να διαβάσουμε το αρχείο εισόδου και να μεταφέρουμε την πληροφορία του στον προσομοιωτή μας. Συνεπώς, διατρέχουμε όλες τις γραμμές του αρχείο και ανάλογα με το κάθε στοιχείο που προκύπτει, διαβάζουμε και τις διάφορες παραμέτρους εισόδου.

Το κάθε στοιχείο που διαβάζουμε καταχωρείται σε μία κοινή δομή λίστας έτσι ώστε να μην χρειάζεται να χρησιμοποιήσουμε μελλοντικά το αρχείο εισόδου. Η λίστα αυτή περιέχει μια καταχώρηση για κάθε κυκλωματικό στοιχείο που διαβάσαμε απο το αρχείο.

Στη συνέχεια θα πρέπει να τονίσουμε ότι στο κάθε κυκλωματικό στοιχείο που διαβάζουμε υπάρχουν και οι κόμβοι στους οποίους το στοιχείο είναι συνδεδεμένο. Έτσι θα πρέπει κατά την ανάλυση μας να εντοπίσουμε τους κόμβους αυτούς και να προσδιορίσουμε ποια στοιχεία είναι συνδεδεμένα μεταξύ τους. Για να διευκολυνθούμε σε αυτό το κομμάτι της διαδικασίας, επιστρατεύουμε την δομή αποθήκευσης του πίνακα κατακερματισμού (hash table) για να μας βοηθήσει στην ευκολότερη οργάνωση των στοιχείων.

Ο πίνακας κατακερματισμού έχει πολλά πλεονεκτήματα σαν δομή αποθήκευσης δεδομένων με τα βασικά του να είναι η επίδοση στο χώρο που καταναλώνει αλλά και στην ταχύτητα ανάγνωσης των δεδομένων. Εμείς όμως στην υλοποίησή μας εκμεταλλευόμαστε τη λειτουργία του πίνακα κατακερματισμού να λειτουργεί με τη βοήθεια κλειδιών για τον εντοπισμό των δεδομένων.

Χρήση του πίνακα κατακερματισμού

Στον πίνακα κατακερματισμού αποθηκεύουμε με τη σειρά τον κάθε κόμβο ο οποίος συνδέεται με κάθε καινούριο στοιχείο που διαβάζουμε. Με αυτόν το τρόπο δημιουργούμε μια νέα αρίθμηση για τον κάθε κόμβο που μας διευκολύνει να εντοπίσουμε πολύ εύκολα στοιχεία τα οποία είναι συνδεδεμένα μεταξύ τους. Έτσι κάθε φορά που διαβάζουμε ένα καινούριο στοιχείο, χρησιμοποιώντας το όνομα των κόμβων με τους οποίους είναι συνδεδεμένος τους εντοπίζουμε εάν υπάρχουν ήδη στη συνδεσμολογία μας ή τους καταχωρούμε αν δεν υπάρχουν.

Για τον κάθε κόμβο χρησιμοποιούμε το όνομά τους για να δούμε αν υπάρχουν ήδη μέσα στον πίνακα. Αν υπάρχει κάποιος κόμβος μας επιστρέφεται η παρούσα τιμή που έχει λάβει ο κόμβος στο σύστημά μας και την οποία θα κρατήσει κατά τη δημιουργία του MNA πίνακα. Σε περίπτωση που ο κόμβος είναι καινούριος, δημιουργείται μια καινούρια καταχώρηση στον πίνακα κατακερματισμού και επιστρέφεται πάλι η καινούρια τιμή του κόμβου.

Να τονίσουμε ότι η παραπάνω διαδικασία εξυπηρετεί πολλές ανάγκες του συστήματός μας. Η πρώτη και κύρια ανάγκη είναι ότι το αρχείο το οποίο λαμβάνουμε σαν είσοδο μπορεί να έχει σαν ονόματα κόμβων και στοιχεία τα οποία δεν είναι ακέραιοι αριθμοί αλλά και ονόματα. Έτσι κάνουμε εύκολα μια μετατροπή από ένα όνομα σε έναν αριθμό ο οποίος μπορεί να χρησιμοποιηθεί άμεσα στην κατασκευή του πίνακα MNA.

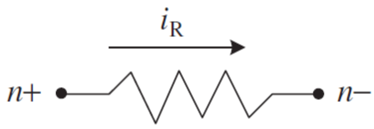
Ένα επίσης πολύ σημαντικό χαρακτηριστικό του πίνακα κατακερματισμού στην υλοποίηση μας είναι ότι διευκολύνει την κατασκευή του MNA πίνακα καθώς συγκεντρώνει όλα τα στοιχεία μαζί χωρίς να δημιουργούνται κενές καταχωρήσεις στον πίνακα οι οποίες μπορεί να υπήρχαν αν απλώς μεταφέραμε τους κόμβους που είχε καταχωρημένος ο χρήστης στο αρχείο εισόδου.

Στο τέλος της παραπάνω διαδικασίας, έχουμε έναν πίνακα ο οποίος περιέχει από μία φορά όλους τους κόμβους του συστήματος χωρίς επαναληπτικές καταχωρήσεις και χωρίς να υπάρχουν κενά στην αρίθμηση των κόμβων. Στη συνέχεια μπορούμε πολύ εύκολα να κατασκευάσουμε τον MNA πίνακα.

Ο MNA πίνακας

Ο **MNA!** είναι ένας πίνακας που προκύπτει απο την ανάλυση των κυκλωματικών στοιχείων και είναι μια επέκταση της Nodal Analysis (NA). Στο σημείο αυτό θα περιγράψουμε συνοπτικά τον τρόπο σύνθεσης του MNA πίνακα ο οποίος αποτελεί βασικό συστατικό της υλοποίησης μας.

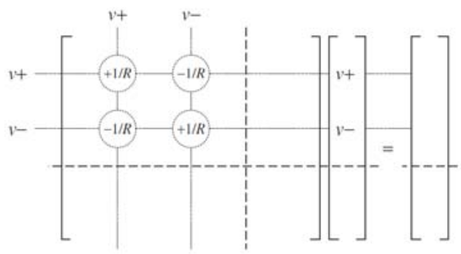
Αρχικά να τονίσουμε οτι μετά τη θεωρητική ανάλυση που έχει γίνει για τη δημιουργία του πίνακα, εμείς δεν χρειάζεται να ασχοληθούμε ξανά με τους νόμους ρευμάτων και κόμβων. Συνεπώς το μόνο που έχουμε να κάνουμε είναι να ακολουθήσουμε το πρότυπο που παρέχεται για την κατασκευή του. Για κάθε κυκλωματικό στοιχείο παρέχεται ένα είδος σφραγίδας που μας λέει ακριβώς σε ποιο σημείο και με τι βάρος θα συνεισφέρει στη δομή που θα αντιπροσωπεύει τον MNA πίνακα. Έχοντας το πρότυπο για το κάθε στοιχείο μπορούμε να κατασκευάσουμε τον πίνακα άμεσα σχεδόν τη στιγμή που διαβάζουμε το στοιχείο απο το αρχείο εισόδου.



Σχήμα 4.1: Ενας αντιστάτης με το ρεύμα που τον διαπερνά να συμβολίζεται σαν i_R και οι κόμβοι σαν $n+$ για τον θετικό και $n-$ για τον αρνητικό

	v^+	v^-	RHS
v^+	$1/R$	$-1/R$	
v^-	$-1/R$	$1/R$	

Σχήμα 4.2: Ένας πιο συγκεντρωτικός πίνακας για τη συνεισφορά του στοιχείου



Σχήμα 4.3: Η σφραγίδα του στοιχείου με τη συνεισφορά του σε διάφορα σημεία του πίνακα. Ο συγκεκριμένος πίνακας αναφέρεται σε αντιστάτη του group 1 ο οποίος δεν αλλάζει συμπεριφορά στο πεδίο του χρόνου

Όπως παρατηρούμε και απο τα παραπάνω σχήματα που παραθέτουμε ένα απλά παράδειγμα κυκλωματικού στοιχείου, η διαδικασία είναι αρκετά άμεση και γίνεται σε γραμμικό χρόνο. Επίσης εδώ φαίνεται και η χρησιμότητα του πίνακα κατακερματισμού οπού μπορούμε πολύ εύκολα να πάρουμε την αρίθμηση του κάθε κόμβου και μάλιστα με πολύ καλή απόδοση.

Να τονίσουμε σε αυτό το σημείο οτι υπάρχουν στοιχεία τα οποία επηρεάζουν και το δεξί μέλος του συστήματος (όπως για παράδειγμα οι πηγές ρεύματος). Τα στοιχεία αυτά έχουν αντίστοιχες σφραγίδες που μας περιγράφουν τις αλλαγές στο δεξί μέλος του συστήματος.

Να τονίσουμε οτι στην παρούσα υλοποίηση αναφερόμαστε μόνο σε στοιχεία τα οποια ανήκουν στο group 1. Τα στοιχεία που ανήκουν στο group 1 είναι τα στοχεία αντίστασης, ανεξάρτητες πηγές ρεύματος, πυκνωτές και πηγές ρεύματος ελεγχόμενες απο τάση. Οποιοδήποτε διαφορετικό στοιχείο μεταβάλει διαφορετικά τον MNA πίνακα με τρόπο που τον καθιστά μη SPD. Ο βασικός τρόπος που επιλέξαμε για να λύσουμε το γραμμικό σύστημα αναφέρεται μόνο σε πίνακες οι οποίοι παρουσιάζουν την παραπάνω ιδιότητα. Φυσικά υπάρχει η περίπτωση να υπάρχουν κυκλωματικά στοιχεία τα οποία δεν ανήκουν στο group 1 αλλά να μην αλλάζουν τη συμπεριφορά του πίνακα και έτσι να παραμένει SPD. Εμείς όμως θα θεωρήσουμε οτι στη γενική περίπτωση το σύστημα μας θα είναι SPD ετσι ώστε να μπορούμε να εφαρμόσουμε τις μεθόδους επίλυσης που αναφέρονται σε αυτά τα συστήματα. Η μετάβαση σε λύση που δεν απαιτεί SPD πίνακες δεν είναι πολύ δύσκολη, δεν καλύπτεται όμως στα πλαίσια αυτής της διπλωματικής.

Κατασκευή του γράφου απο τον πίνακα MNA

Σε προηγούμενο κεφάλαιο 3 είχαμε αναφερθεί στη διαφορά των δεδομένων μεταξύ του πίνακα MNA και του αραιού πίνακα που αντιπροσωπεύει τον γράφο. Είχαμε τονίσει οτι θα χρειαστεί μια προεργασία των δεδομένων, έτσι ώστε να είναι ο πίνακας σε θέση να μας δώσει τα σωστά αποτελέσματα κατάτμησης. Η προεργασία που χρειάζεται πάνω στα δεδομένα είναι η εξής:

- **Μετατροπή όλων των μη μηδενικών καταχωρήσεων απο αριθμούς κινητής υποδιαστολής(double) σε αριθμό 1**. Το μόνο που χρειαζόμαστε για να γίνει ο κατακερματισμός του γράφου είναι να ξέρουμε αν υπάρχει κάποια σύνδεση μεταξύ δυο κόμβων. Δεν θα υπάρχει κάποια σημαντική αλλαγή αν το βάρος είναι κάτι διαφορετικό απο μονάδα. Επίσης να τονίσουμε οτι δεν μπορεί να υποστηρίξει το εργαλείο τη δυνατότητα να έχουμε βάρη κόμβων σε δεκαδική μορφή
- **Αφαίρεση όλων των στοιχείων τα οποια βρίσκονται στη κύρια διαγώνιο του MNA πίνακα**. Στη κύρια διαγώνιο του MNA πίνακα υπάρχει το άθροισμα των στοιχείων της κάθε γραμμής αντίστοιχα. Σε μια δομή γράφου όμως μια τιμή στα διαγώνια στοιχεία θα σήμανε οτι υπάρχει ακμή που ξεκινάει και καταλήγει στον ίδιο κόμβο. Αυτό προφανώς δεν είναι σωστό για το σύστημά μας.

Απο τις παραπάνω διαδικασίες που πρέπει να κάνουμε για να μετατρέψουμε τον πίνακα σε γράφο, η δεύτερη είναι κάπως πιο απαιτητική απο ότι φαίνεται αρχικά. Αν δούμε τη δομή με την οποία αποθηκεύεται ο αραιός πίνακας, κάθε γραμμή του πίνακα βρίσκεται σε διπλανές θέσεις. Έτσι εκεί που τελειώνει η μια γραμμή ξεκινάει η επόμενη και ο πίνακας των δεικτών που έχουμε για να βρίσκουμε την κάθε γραμμή βασίζεται σε αυτή την αρχή. Για να διατρέξουμε τα στοιχεία μιας γραμμής ξεκινάμε απο το σημείο που μας δείχνει ο δείκτης της γραμμής και πηγαίνουμε μέχρι και το στοιχείο που μας λέει ο δείκτης της επόμενης γραμμής. Έτσι με αυτόν τον έμμεσο τρόπο μπορούμε πολύ εύκολα να διατρέξουμε τα στοιχεία της κάθε γραμμής.

Με βάση την παραπάνω παραδοχή που κάνουμε καταλαβαίνουμε οτι δεν είναι και πολύ εύκολο να αφαιρέσουμε ένα στοιχείο απο τον πίνακα όπως θα κάναμε σε άλλη περίπτωση ενός πλήρους πίνακα. Με βάση αυτό το πρόβλημα που έχουμε αλλά και απο το γεγονός οτι θα χρειαστούμε την αρχική πληροφορία που έχει ο MNA πίνακας πήραμε την απόφαση να δημιουργήσουμε έναν νέο πίνακα που θα χρησιμοποιηθεί για να γίνει η κατάτμηση του γράφου, με τα χαρακτηριστικά που θέλουμε.

Αρχικοποίηση των ορισμάτων και κατάτμηση του γράφου

Πριν γίνει οποιαδήποτε διαδικασία κατάτμησης θα πρέπει να αρχικοποιήσουμε όλες τις παραμέτρους στις οποίες αναφερθήκαμε στο 3. Το εργαλείο METIS έχει την δικιά του δομή την οποία πρέπει να αρχικοποιήσουμε πριν κάνουμε την κατάτμηση. Στην εφαρμογή μας υπάρχει η δυνατότητα ο χρήστης να διαλέγει για κάθε εκτέλεση τις επιλογές που θέλει να τρέξει καθώς είχαμε αναφερθεί στο πόσο σημαντικές είναι οι επιλογές που έχει το εργαλείο για το αποτέλεσμα της κατάτμησης.

Αφού έχουμε αρχικοποιήσει τις επιλογές της κατάτμησης, μπορούμε να προχωρήσουμε στην κατάτμηση καθ'αυτή. Όπως είχαμε τονίσει σε προηγούμενο κεφάλαιο, έχουμε δύο διαφορετικές επιλογές κατάτμησης, οι οποίες τρέχουν ακριβώς με τις ίδιες παραμέτρους και έτσι δεν χρειάζεται κάποια προσαρμογή απο μέρους μας. Αφού τελειώσει η εκτέλεση του εργαλείου, μας επιστρέφεται σε μια θέση μνήμης που έχουμε ορίσει εμείς, ένα διάνυσμα το οποίο περιέχει την τιμή του κάθε τομέα που ανήκει η κάθε κορυφή. Έτσι μπορούμε άμεσα να αντιστοιχήσουμε τον κάθε κόμβο με τον τομέα που ανήκει.

Μετά το πέρας της παραπάνω διαδικασίας, θα λάβουμε έναν ακέραιο για να μπορέσουμε να ελέγξουμε εαν υπήρχε κάποιο πρόβλημα με την κατάτμηση καθώς και τον αριθμό των κατατμήσεων που κατάφερε να παράξει το εργαλείο.

Αποκοπή των ενώσεων μεταξύ των κατατμήσεων

Μια απο τις βασικές αρχές τις οποίες βασίζεται η κατασκευή του προρυθμιστή μέσω της επίλυσης υποσυστημάτων, είναι οτι δεν υπάρχουν συνδέσεις μεταξύ του κάθε υποσυστήματος με τα υπόλοιπα. Έτσι κάθε τομέας θα πρέπει να μην έχει καμία ακμή η οποία θα ξεκινάει απο κόμβο ενός υπογράφου και θα καταλήγει σε κόμβο διαφορετικού υπογράφου.

Σε αυτό το σημείο μπορούμε να καταλάβουμε και τον λόγο για τον οποίο θέλαμε να επιτύχουμε όσο το δυνατόν μικρότερο αριθμό συνδέσεων μεταξύ διαφορετικών υπογράφων(edge-cut). Ο λόγος είναι πλεόν προφανής και έχει να κάνει με το γεγονός ότι οι συνδέσεις αυτές θα αποκοπούν. **Η αποκοπή των συνδέσεων θα οδηγήσει σε μείωση της ακρίβειας της λύσης και συνεπώς θα αυξήσει τα βήματα που θα χρειαστούν για να συγκλίνει η λύση μας.** Συνεπώς, είναι πολύ σημαντικό να τονίσουμε και πάλι οτι χρειαζόμαστε τεχνικές κατάτμησης οι οποίες παράγουν κατατμήσεις με χαμηλή συνδεσιμότητα μεταξύ των υπογράφων(clusters).

Πριν γίνει οποιαδήποτε άλλη κίνηση πρέπει να αποκόψουμε τον κάθε υπογράφο απο όλους τους άλλους. Για να γίνει η αποκοπή, θα χρειαστεί να εντοπίσουμε τις ακμές που υπάρχουν και να τις αφαιρέσουμε. Το πρόβλημα που προκύπτει είναι τι θα κάνουμε με το βάρος της κάθε ακμής. **Το βάρος της κάθε ακμής θα πρέπει να διαμοιραστεί στους κόμβους απο τους οποίους προέρχεται η ακμή αυτή.** Έτσι το συνολικό βάρος θα μείνει ίδιο αλλά η κατάτμηση του βάρους θα αλλάξει. Υπάρχουν διάφορες τεχνικές για να επιλεχθεί το ποσοστό που θα πρέπει να μοιραστεί σε κάθε έναν απο τους δύο κόμβους που ανήκει η ακμή. Η τεχνική που θα χρησιμοποιήσουμε εμείς είναι και η πιο απλή και έχει να κάνει με το ισομερές διαμέρισμα του βάρους κα.

Κατασκευή του προ-ρυθμιστή

Σε αυτό το σημείο έχουμε έτοιμες τις συστάδες(clusters) των κόμβων και έτσι μπορούμε να προχωρήσουμε στην κατασκευή του προρυθμιστή. Για την κατασκευή του πίνακα που θα αντιπροσωπεύει τον προρυθμιστή, υπάρχουν δυο διαφορετικές τεχνικές που μπορούμε να χρησιμοποιήσουμε και η καθε μια έχει τα δικά της πλεονεκτήματα και μειονεκτήματα.

Μια τεχνική που μπορούμε να χρησιμοποιήσουμε για να κατασκευάσουμε τον προρυθμιστή είναι να χρησιμοποιήσουμε έναν πίνακα απο δείκτες που θα δείχνουν σε στοιχεία που ανήκουν στην ίδια κατάτμηση. Έτσι με αυτόν το τρόπο δεν χρειάζεται να γίνει οποιαδήποτε μετακίνηση ή αντιγραφή στοιχείων απο τον αρχικό πίνακα σε κάποιο άλλο. Επίσης μπορούμε να χρησιμοποιήσουμε τον ίδιο πίνακα επαναληπτικά για διαφορετικούς τομείς του γράφου. Έτσι ο επιπλέον χώρος που θα χρησιμοποιήσουμε θα είναι μηδενικός. Επίσης η λύση θα συντεθεί σταδιακά χρησιμοποιόντας τις λύσεις του κάθε υπο-γράφου ξεχωριστά όπως θα δείξουμε και στη συνέχεια.

Μια διαφορετική μέθοδος είναι να κατασκευάσουμε καινούριες δομές στις οποίες θα αποθηκεύουμε το κάθε σύστημα ξεχωριστά. Επίσης εκτός απο τον κάθε υπογράφο που θα αποθηκεύουμε στις καινούριες δομές, θα χρειαστούμε και ένα διάνυσμα για τον κάθε υπογράφο που θα είναι το δεξί μας μέλος. Όπως καταλαβαίνουμε η μέθοδος αυτή χρειάζεται έναν αρκετά μεγαλύτερο αποθηκευτικό χώρο καθώς και αντιγραφές που χρειάζονται να γίνουν για να κατασκευαστούν τα συστήματα σε νέες δομές. Έτσι όμως έχουμε ένα πολύ μεγάλο κέρδος το οποίο είναι οτι θα έχουμε συγκεντρωμένα τα δεδομένα και με τη σειρά ανάλογα με τον πιο τομέα ανήκουν και τό ίδιο θα ισχύσει και για το δεξί μέλος.

Για να δούμε πιο συγκεκριμένα πως θα κατασκευαστεί ο προρυθμιστής μας απο το κάθε υποσύστημα ξεχωριστά. Σε προηγούμενο κεφάλαιο είχαμε τονίσει οτι ο προρυθμιστής αναλαμβάνει να βρεί μια πρώτη προσέγγιση της λύσης η οποία βελτιώνεται με κάθε επανάληψη. Το σύστημα που λύνουμε είναι το παρακάτω:

$$M * z = r$$

όπου είναι ο πίνακας του προρυθμιστή, z είναι μια προσέγγιση της λύσης και r είναι το υπόλοιπο μεταξύ της πραγματικής λύσης και της λύσης που βρήκαμε(residual). Στο παραπάνω σύστημα ο προρυθμιστής αποτελείται απο ένα μοναδικό πίνακα.

Στη περίπτωση που εξετάζουμε, έχουμε να λύσουμε πολλά διαφορετικά συστήματα και να συνδυάσουμε τις λύσεις τους για να σχηματίσουμε τον προρυθμιστή. Για το λόγο αυτό σχηματίζουμε έναν διαγώνιο block πίνακα ο οποίος στη διαγώνιο του αποτελείται από πίνακες του κάθε υποσυστήματος και εκτός της διαγώνιο του δεν θα πρέπει να υπάρχει κάτι διαφορετικό απο μηδενικές τιμές. Το σύστημα που θα προκύψει θα είναι:

$$\begin{bmatrix} M_1 & \emptyset & \dots & \emptyset \\ \emptyset & M_2 & \dots & \emptyset \\ \emptyset & \dots & M_{N-1} & \emptyset \\ \emptyset & \dots & \emptyset & M_N \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} \Rightarrow \begin{Bmatrix} M_1 z_1 = r_1 \\ M_2 z_2 = r_2 \\ \vdots \\ M_N z_N = r_N \end{Bmatrix}$$

Από το παραπάνω σύστημα αυτά που αναγνωρίζουμε είναι τα $_1$ με M_N τα οποία αποτελούν τα υποσυστήματα που έχουμε κατασκευάσει.

Το πολύ σημαντικό που παρατηρούμε απο το παραπάνω σύστημα είναι οτι πρέπει τα στοιχεία του κάθε υποσυστήματος να αλλάξουν θέση και να εισέλθουν στον πίνακα του προρυθμιστή με τη σειρά που ανήκουν στο σύστημα. Αυτό προϋποθέτει ότι θα πρέπει να ξαναγίνει η διαδικασία της αρίθμησης έτσι ώστε να ταιριάξουν τα στοιχεία στις καινούριες θέσεις τους. Επίσης πρέπει να αλλάξουμε και την αρίθμηση και στα στοιχεία του δεξιού μέλους.

Αφού έχει γίνει η ανατοποθέτηση των στοιχείων μπορούμε πλέον να συνεχίσουμε στην επίλυση του κάθε συστήματος και να συγκεντρώσουμε τα στοιχεία του πίνακα z τα οποία θα χρησιμοποιήσουμε και παρακάτω για την εύρεση της λύσης μας.

Το πολύ σημαντικό που έχουμε επιτύχει είναι να κατασκευάσουμε έναν προρυθμιστή ο οποίος ταιριάζει πολύ περισσότερο στο σύστημα μας από έναν γενικού σκοπού προρυθμιστή καθώς έχει κατασκευαστεί από την επίλυση τμηματικά του ολοκληρωμένου συστήματος. Ακόμη και η αρχική λύση που θα πάρουμε σαν προσέγγιση μπορεί να μην είναι απόλυτα ακριβής όσο η πραγματική λύση αλλά σίγουρα θα είναι πολύ κοντά και έτσι δεν θα χρειαστούν πολλές επαναλήψεις του συνολικού συστήματος.

Να τονίσουμε επίσης ότι ο χώρος που θα χρειαστεί για τον block διαγώνιο πίνακα δεν θα έχει κάποια βασική επιβάρυνση καθώς εφόσον δεν θα έχει άλλα στοιχεία μη μηδενικά εκτός από τα στοιχεία της διαγωνίου του και θα είναι αποθηκευμένος σε αραιή μορφή. Στην πραγματικότητα ο καινούριος πίνακας θα έχει λιγότερα στοιχεία από τον αρχικό καθώς θα έχουμε αποκόψει τις συνδέσεις που υπάρχουν μεταξύ των στοιχείων που ανήκουν σε διαφορετική κατάτμηση.

Συνολική επίλυση του συστήματος

Για να λύσουμε το συνολικό σύστημα δεν έχουμε να κάνουμε κάτι παραπάνω απο το να χρησιμοποιήσουμε τον προρυθμιστή και την αρχική προσεγγιστική λύση σαν είσοδο στη CG και να συνεχίσουμε κανονικά την επίλυση μας. Δεν χρειάζεται να γίνει κάποια βασική αλλαγή στη συνέχεια.

Είμαστε σίγουροι ότι μετά το πέρας της κατασκευής του παραπάνω προρυθμιστή, η σύγκλιση θα είναι πολύ πιο γρήγορη και έτσι η επένδυση χρόνου και πόρων που θα έχουμε κάνει θα αποδώσει τελικά κατά τη γρήγορη επίλυση του συστήματος.

Επιπλέον παρατηρήσεις της υλοποίησης

Για να κλείσουμε το κεφάλαιο αυτό θα τονίσουμε ότι ένα πολύ μεγάλο κέρδος της προσέγγισης μας είναι ότι αν έχουμε μια ανάλυση που αλλάζει κάποια στοιχεία μέσα στο σύστημα λόγω κάποιων AC τάσεων ή ρευμάτων, δεν χρειάζεται να λύσουμε ξανά όλο το σύστημα και να κατασκευάσουμε τον προρυθμιστή από την αρχή. Το μόνο που έχουμε να κάνουμε είναι να μαρκάρουμε τα υποσυστήματα που έχουν μεταβληθεί και να λύσουμε ξανά μόνο αυτά. Συνεπώς έχουμε κέρδος έναντι κάποιων προρυθμιστών που χρειάζονται να κατασκευαστούν από την αρχή για κάθε αλλαγή που συμβαίνει στο σύστημα.

Κάτι το οποίο δεν έχει τονιστεί επίσης είναι ότι έχουμε πολύ μεγάλο κέρδος και σε υλοποιήσεις όπου τα αρχεία εισόδου είναι πολύ μεγάλα και μπορεί να μην χωράνε στη κύρια μνήμη. Το μεγάλο μέγεθος αρχείων δεν είναι κάτι το σπάνιο καθώς σε ρεαλιστικά συστήματα συμβαίνει πολύ συχνά. Ακόμη και η αραιή μορφή δεν μας διασφαλίζει ότι ο χώρος που θα χρειαστούμε θα είναι διαθέσιμος στη κύρια μνήμη του συστήματος μας.

Το παραπάνω πρόβλημα της μνήμης αντιμετωπίζεται αρκετά εύκολα με το σύστημα που έχουμε σχεδιάσει. Κάθε επίλυση των υποσυστημάτων είναι ανεξάρτητη απο τις υπόλοιπες. Συνεπώς δεν χρειάζεται να έχουμε όλα τα υποσυστήματα συνεχώς στη κύρια μνήμη. Μπορούμε λοιπόν να κάνουμε ανεξάρτητη επίλυση του κάθε συστήματος και συνεπώς να μπορούμε να λύσουμε όσο μεγάλα συστήματα γίνεται για να κατασκευάσουμε έναν προρυθμιστή. Φυσικά θα πρέπει να φροντίσουμε στη συνέχεια να μετατρέψουμε τη διαδικασία επίλυσης και αυτή σε μια διαδικασία που να μπορεί να λειτουργεί με το παραπάνω μοντέλο.

Τέλος να τονίσουμε ότι η κάθε μια από τις παραπάνω επιλύσεις των υποσυστημάτων είναι ανεξάρτητη από τις υπόλοιπες και επίσης αναφέρονται σε ασυσχέτιστες θέσεις μνήμης. Το παραπάνω γεγονός μας οδηγεί πολύ εύκολα στο να κάνουμε οποιαδήποτε διαδικασία παραλληλισμού της επίλυσης των δεδομένων και μάλιστα χωρίς να επενδύσουμε πολύ χρόνο στην αλλαγή των αλγορίθμων επίλυσης καθώς είναι αρκετά εμφανή τα σημεία που μπαίνει ο παραλληλισμός.

Κεφάλαιο 5

Εκτίμηση της λύσης

Τα αρχεία ελέγχου που θα χρησιμοποιήσουμε στη συνέχεια για την εκτίμηση των διάφορων λύσεων μας βρίσκονται στον πίνακα 5.1.

Name	i	n	r	s	v	l	nonZeros
ibmpg1	10774	30638	30027	14208	14308	2	147.315
ibmpg2	37926	127238	208325	1298	330	5	833.960
ibmpg3	201054	851584	1401572	461	955	5	3.656.107
ibmpg4	276976	953583	1560645	11682	962	6	6.244.504

Πίνακας 5.1: τον πίνακας και τα αρχεία τα λάβαμε απο το

5.1 Εκτίμηση της κατάτμησης γράφου

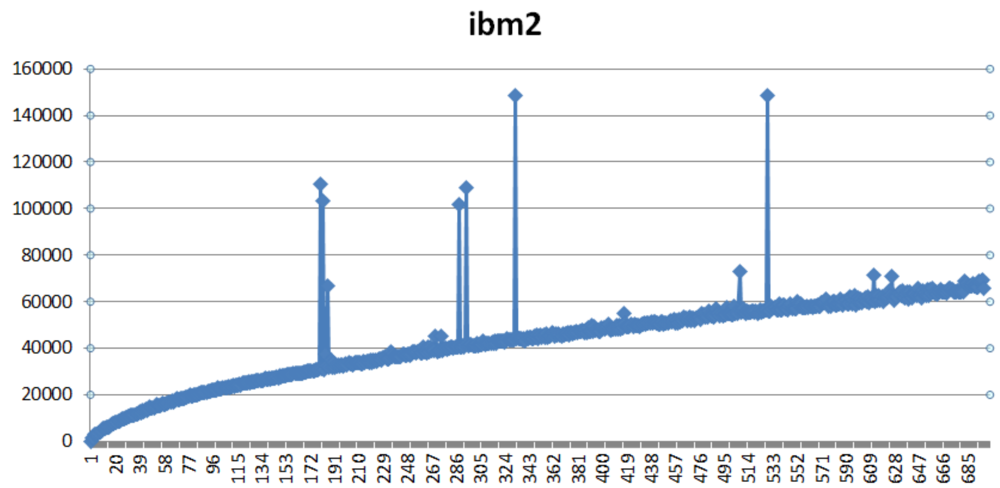
Ένα απο τα βασικά τμήματα της υλοποίησης μας είναι και η κατάτμηση του γράφου. Πρέπει λοιπόν σε αυτό το σημείο να αναφερθούμε σε κάποια στοιχεία που θα μας δώσουν μια καλύτερη εκτίμηση της διανομής των πόρων της επίλυσης μας.

Απόδοση τομής των υπογράφων

Όπως τονίσαμε σε προηγούμενα κεφάλαια, ενα απο τα σημαντικότερα χαρακτηριστικά που θέλουμε να έχει η κατάτμηση του γράφου μας είναι οτι θα πρέπει οι ακμές που υπάρχουν μεταξύ διαφορετικά υποσυστήματα/υπογράφους να είναι όσο το δυνατόν λιγότερες. Έτσι μια απο τις πρώτες διαδικασίες που κάναμε είναι να δούμε την απόδοση που έχει η κατάτμηση με βάση το πλήθος των κατατμήσεων που ζητάμε σε συνάρτηση με το μέγεθος που έχει ο γράφος.

Απο τα στοιχεία που μαζέψαμε, παρατηρήσαμε ένα πολύ ενδιαφέρον φαινόμενο το οποίο είχε να κάνει με την εμφάνιση κάποιων υπερβολικών τιμών τομής(edge-cut) σε τυχαία σημεία. Το γεγονός αυτό είναι αξιοσημείωτο καθώς μπορεί να προσπαθήσουμε να προσεγγίσουμε ένα σύστημα ξεκινώντας απο καθόλου αποδοτική κατάτμηση στην αρχή. Συνέπως μια άσχημη κατάτμηση μπορεί να οδηγήσει και σε άσχημη σύγκλιση.

Στο γράφημα 5.1 μπορούμε να δούμε την συμπεριφορά που παρουσιάζει η κατάτμηση γράφου σε σχέση με τον αριθμό των κατατμήσεων που επιλέγουμε. Μπορούμε να παρατη-



Σχήμα 5.1: Η συσχέτηση μεταξύ της τομής του γράφου σε σχέση με τον αριθμό των κατατμήσεων που ζητάμε

ρήσουμε διάφορα φαινόμενα απο το παραπάνω γράφημα.

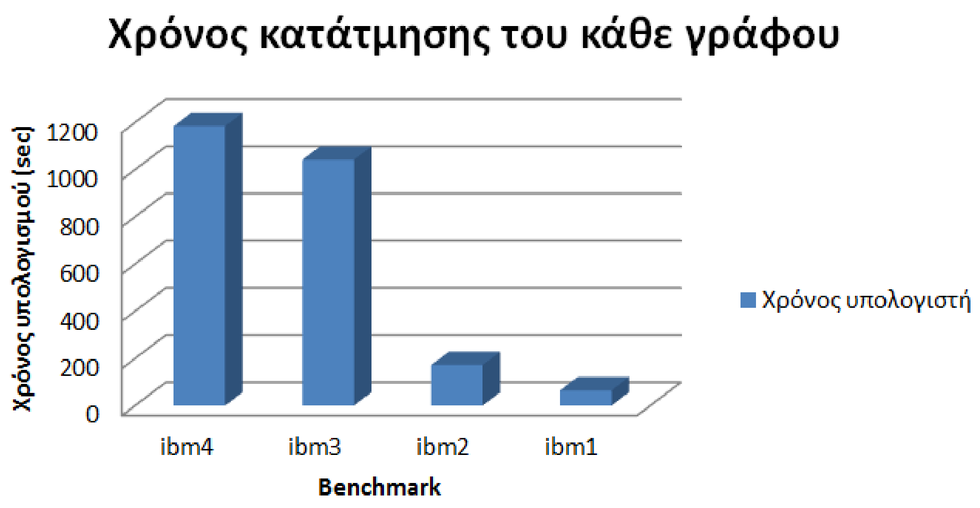
Αρχικά αυτό που παρατηρούμε είναι οτι η τομή του γράφου αυξάνεται με βάση τον αριθμό των κατατμήσεων. Το παραπάνω συμπέρσμα είναι λογικό να ισχύει καθώς περισσότερες κατατμήσεις σημαίνει οτι θα υπάρχουν και περισσότερες συνδέσεις μεταξύ των κατατμήσεων. Βέβαια όπως παρατηρούμε, η συμπεριφορά του γράφου δεν είναι γραμμική. Η μη γραμμικότητα του γράφου μας δίνει την ευκαιρία να βρούμε μια κατάτμηση που να έχει αρκετές κατατμήσεις και άρα η κάθε κατάτμηση να μην είναι πολύ μεγάλη σε μέγεθος, αλλά και η τομή να μην αυξάνεται πάρα πολύ. Πρέπει λοιπόν για μεγάλα κυκλώματα να γίνεται μια προεργασία για να εντοπίζονται τέτοια σημεία.

Μια ακόμη παρατήρηση που μπορεί να γίνει είναι οτι εμφανίζονται κάποια σημεία όπου η κατάτμηση παρουσιάζει αστάθεια. Η αστάθεια αυτή μπορεί να οφείλεται σε πολλούς παράγοντες καθώς υπάρχει και ένας παράγοντας τυχαιότητας στη διαδικασία κατάτμησης. Είναι λοιπόν πολύ σημαντικό να γίνεται μια προεργασία για να αναγνωρίζουμε αν βρισκόμαστε σε ένα τέτοιο σημείο αστάθειας γιατί διαφορετικά τα επόμενα βήματα θα παρουσιάσουν πολύ φτωχά αποτελέσματα.

Απο τα παραπάνω συμπεράσματα μπορούμε να δούμε οτι είναι πολύ σημαντικό να γίνεται σε πολύ μεγάλα κυκλώματα μια προεργασία για να αποφύγουμε άσχημα φαινόμενα. Το πολύ ενθαρρυντικό είναι οπως θα δούμε και σε ένα γράφημα πιο κάτω, οτι η κατάτμηση είναι σταθερή στο χρόνο και εξαρτάται μόνο απο το μέγεθος του γράφου και όχι απο τον αριθμό των κατατμήσεων που διαλέγουμε. Επίσης η διαδικασία της κατάτμησης περιλαμβάνει ένα πολύ μικρό ποσοστό της συνολικής ανάλυσης. Συμπερασματικά υπάρχει η δυνατότητα να γίνει μια προεργασία χωρίς να έχει πολύ μεγάλη επίπτωση στο χρόνο και η οποία θα μας διασφαλίσει οτι δεν θα πέσουμε σε ασταθή κατάσταση.

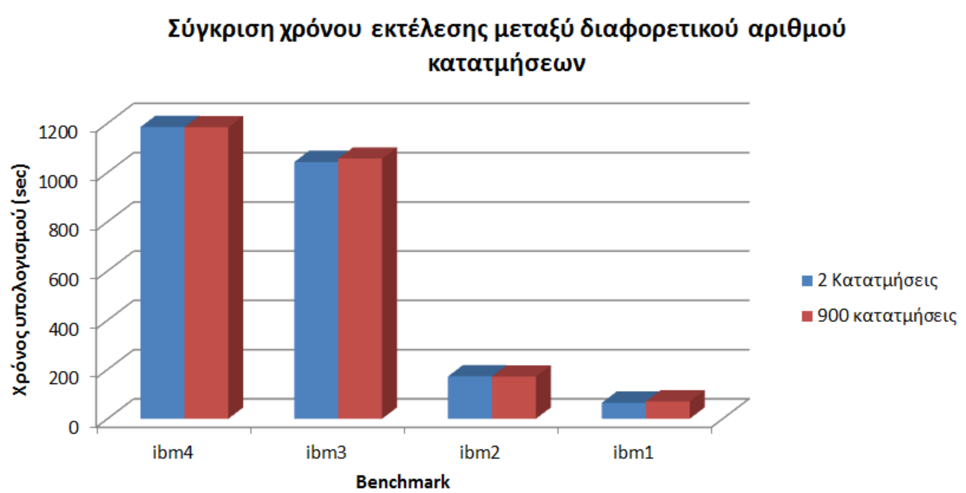
Απόδοση χρόνου κατάτμησης των υπογράφων

Ένα πολύ σημαντικό σημείο της εφαρμογή μας είναι ο χρόνος εκτέλεσης της κατάτμησης του γράφου. Πρέπει η διαδικασία κατάτμησης να είναι όσο το δυνατόν μικρότερη σε σχέση με την επίλυση. Στο γράφημα 5.2 μπορούμε να δούμε τη σύγκριση χρόνου για το κάθε κύκλωμα. Τα στοιχεία του κυκλώματος βρίσκονται στον πίνακα 5.1.



Σχήμα 5.2: Ο χρόνος εκτέλεσης του κάθε αρχείου εισόδου

Η βασική παρατήρηση που μπορεί να γίνει είναι οτι ο χρόνος που λαμβάνει η κατάτμηση σχετίζεται άμεσα απο τα μη μηδενικά στοιχεία που υπάρχουν στο κύκλωμα.



Σχήμα 5.3: Ο χρόνος εκτέλεσης του κάθε αρχείου εισόδου

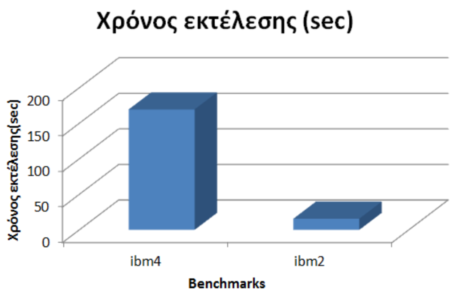
Απο το γράφημα 5.3 μπορούμε να δούμε τη σύγκριση μεταξύ του χρόνου που παίρνει για να κάνει κατάτμηση σε διαφορετικό αριθμό κατατμήσεων. Στο γράφημα αυτό βλέπουμε δύο εντελώς διαφορετικές επιλογές με τη μία να ζητάει έναν πολύ μικρό αριθμό κατατμήσεων (2 πιο συγκεκριμένα) και την άλλη να ζητάει ένα πολύ μεγάλο αριθμό κατατμήσεων (900 πιο συγκεκριμένα). Το πολυ σημαντικό που παρατηρούμε είναι οτι ο χρόνος κατάτμησης είναι σχεδόν ίδιος και στις δυο επιλογές. Το παραπάνω φαινόμενο ταιριάζει άμεσα με την θεωρία που υπάρχει πίσω απο την κατάτμηση του εργαλείου METIS.

Με βάση το γράφημά 5.3 μπορούμε εύκολα να θεωρήσουμε οτι μπορεί να γίνεται ένα είδος αρχικής εκτίμησης για να δούμε αν βρισκόμαστε σε σημείο όπου η τομή του γράφου(edge-cut) είναι σε σταθερό σημείο.

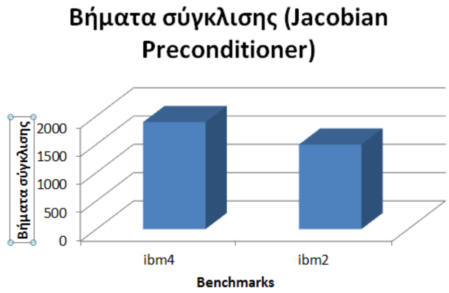
Μπορούμε να πούμε σαν παρατήρηση οτι σε σχέση με την εκτέλεση πού είναι σε κλίμακα λεπτών, η κατάτμηση είναι αρκετά μικρότερη καθώς κυμαίνεται σε τάξη δευτερολέπτων. Το παραπάνω γεγονός είναι πολύ ενθαρρυντικό καθώς μας επιτρέπει να χρησιμοποιήσουμε την κατάτμηση για τη δημιουργία προρυθμιστή χωρίς πολύ μεγάλη επένδυση χρόνου.

Χρόνος εκτέλεσης εύρεσης της λύσης

Η πιο σημαντική μετρική είναι ο χρόνος που κάνει να λυθεί το σύστημα με τη χρήση της μεθόδου CG. Ο χρόνος αυτός επηρρεάζεται άμεσα οπως έχουμε τονίσει απο το πόσο καλός είναι ο προρυθμιστής που θα χρησιμοποιήσουμε. Επίσης όπως θα δούμε και στα γραφήματα, ο προρυθμιστής επηρεάζει και τον αριθμό των βημάτων που απαιτούνται για να συγκλίνει η λύση.



Σχήμα 5.4: Ο χρόνος της εκτέλεσης του κάθε αρχείου.



Σχήμα 5.5: Τα βήματα που πρέπει να γίνουν για να γίνει επιτυχημένη σύγκλιση.

Για τις παραπάνω μετρήσεις χρησιμοποιήθηκε ακρίβεια 6 δεκαδικών ψηφίων για να προσδιορίσουμε ποια θα είναι η ακρίβεια την οποία θέλουμε να επιτύχουμε για να είμαστε ικανοποιημένοι με τη λύση.

Παρατηρούμε απο το γράφημα 5.4 οτι ο χρόνος εκτέλεσης εξαρτάται άμεσα απο τον αριθμό των μη μηδενικών ψηφίων. Επίσης μπορούμε να δούμε οτι του μεγαλύτερου κυκλώματος μια μοναδική λύση παίρνει γύρω στα 170 δευτερόλεπτα. Το παραπάνω νούμερο είναι απαγορευτικό αν θέλουμε να κάνουμε παραπάνω απο μία ανάλυση του συστήματος όπως για παράδειγμα γίνεται όταν έχουμε πηγές εναλλασσόμενου ρεύματος ή τάσης. Μπορούμε να φανταστούμε οτι το πρόβλημα γίνεται σχεδόν άλυτο καθώς θα χρειαστούμε μέρες να προσομοιώσουμε μια συμπεριφορά η οποία έχει πολλές εναλλαγές και πολλές επιλύσεις του συστήματος.

Βλέπουμε λοιπόν με το παραπάνω γράφημα, πόσο σημαντικό γίνεται το γεγονός να έχουμε υλοποιήσεις οι οποίες θα συγκλίνουν γρήγορα. Επίσης φαίνεται το πόσο σημαντικό είναι να επενδύσουμε χρόνο στην κατασκευή ενος προρυθμιστή που θα επιτυγχάνει γρήγορη σύγκλιση.

Ο βασικός λόγος που παραθέσαμε τα παραπάνω στατιστικά είναι για να δούμε πόσο θα είναι το κόστος που θα πρέπει να επενδύσουμε για την κατασκευή του προρυθμιστή και αν αξίζει. Το κάθε υποσύστημα που θα κατασκευάσουμε επιδιώκουμε να έχει γύρω στους 1000 με 1500 κόμβους. Η επίλυση ενός τέτοιου συστήματος είναι σε τάξη κάποιων δεκάδων ms. Συνεπώς η συνολική επίλυση όλων των συστημάτων θα κυμαίνεται στις τάξεις του δευτερολέπτου.

Αφού σχεδιάσουμε τον προρυθμιστή σε χρόνο κοντά στα 4 με 5 δευτερόλεπτα (για το πολύ μεγάλο κύκλωμα) μετά θα πρέπει να συνεχίσουμε με την επίλυση ολόκληρου του συστήματος. Μην ξεχνάμε όμως οτι η λύση που ήδη θα έχουμε βρεί και θα είναι μια προσέγγιση της πραγματικής λύσης, θα πρέπει να είναι αρκετά κοντά στην πραγματική λύση. Συνεπώς τα βήματα της συνολικής επίλυσης θα πρέπει να είναι αρκετά χαμηλά. Ολες οι παραπάνω παραδοχές δεν μπορούν να επιβεβαιωθούν σε αυτό το στάδιο καθώς η εφαρμογή μας έφτασε μέχρι και το σημείο οπου λύνουμε το κάθε υποσύστημα και συγκεντρώνουμε τα δεδομένα. Συνεπώς έχουμε κάποια στατιστικά στοιχεία για το χρόνο που λαμβάνει να λυθεί το κάθε υποσύστημα αλλά δεν έχουμε στοιχεία για το συνολικό χρόνο επίλυσης ολόκληρου του συστήματος.

Βλέπουμε οτι με απλούς υπογισμούς το σύστημα μας βρίσκει αρκετά πιο γρήγορα μια λύση σε σχέση με ένα σύστημα που χρησιμοποιεί έναν πιο απλό προρυθμιστή.

Κεφάλαιο 6

Επίλογος

Στην παρούσα διπλωματική εργασία είδαμε αναλυτικά την κατασκευή ενός προσομοιωτή κυκλωμάτων από το διάβασμα ενός αρχείου περιγραφής συνδεσμολογίας μέχρι και την επίλυση του συστήματος. Δώσαμε μια περιγραφή της θεωρίας πίσω απο την επίλυση CG και εξηγήσαμε τις βασικές αρχές των προρυθμυιστών και της σημασίας του στην επίλυση των συστημάτων. Στη συνέχεια αναφερθήκαμε στη κατάτμηση γράφου και στο εργαλείο METIS το οποίο χρησιμοποιήσαμε για να την πραγματοποιήσουμε. Σε επόμενο κεφάλαιο αναφερθήκαμε εξονυχιστικά στην δικιά μας υλοποίηση του προρυθμιστή, τις τεχνικές ανάπτυξης που εφαρμόσαμε και τις δυσκολίες που αντιμετωπίσαμε. Τέλος, αναφερθήκαμε στα αποτελέσματα που πήραμε κατά την πειραματική ανάλυση.

Στα αποτελέσματα είδαμε οτι η κατάτμηση δεν καταλαμβάνει σημαντικό χρονικό διάστημα και τα οφέλη που προσφέρει είναι πολλά. Είδαμε οτι τα συστήματα που παράγονται είναι αρκετά μικρά έτσι ώστε να ολοκληρώνονται σε πολύ μικρό χρόνο και έτσι ακόμη και αν έχουμε ένα πολύ μεγάλο όγκο συστημάτων η επίλυση θα γίνει σε τάξεις κάποιων λίγων δευτερολέπτων. Επίσης αναφερθήκαμε και στο γεγονός οτι δεν χρειάζεται να επιλύουμε όλο το σύστημα κάθε φορά που αλλάζει κάτι για να παράγουμε τον προρυθμιστή αλλά μόνο το υποσύστημα που ανήκει η κάθε αλλαγή. Τέλος, ένα πολύ σημαντικό πλεονέκτημα είναι οτι δεν χρειάζεται να φορτωθεί όλο το σύστημα στη κύρια μνήμη καθώς η κάθε επίλυση είναι ατομική και ξεχωριστή στα δεδομένα που παράγει απο τις υπόλοιπες.

6.1 Μελλοντικές προεκτάσεις

Πιθανές προεκτάσεις στην παρούσα εργασία θα μπορούσαν να 'ναι:

- Ολοκλήρωση του συστήματος έτσι ώστε να δούμε την πραγματική απόδοση σε ταχύτητα σύγκλισης.
- Παραλληλοποίηση διαφόρων σταδίων της υλοποίησης.
- Επέκταση της υλοποίησης έτσι ώστε να υποστηρίζει μεταβατική ανάλυση.

Βιβλιογραφία

[1] Richard Barrett, Michael W Berry, Tony F Chan, James Demmel, June Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, and Henk Van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods*, volume 43. Siam, 1994.

[2] Charles M Fiduccia and Robert M Mattheyses. A linear-time heuristic for improving network partitions. In *Design Automation, 1982. 19th Conference on*, pages 175–181. IEEE, 1982.

[3] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[4] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[5] Brian W Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical journal*, 49(2):291–307, 1970.

[6] Mark EJ Newman and Michelle Girvan. Mixing patterns and community structure in networks. In *Statistical mechanics of complex networks*, pages 66–87. Springer, 2003.

[7] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[8] Daniel A Spielman and Shang-Hua Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *arXiv preprint cs/0607105*, 2006.