
Δρομολόγηση Φωλιασμένων Βρόχων
με τη χρήση Μεθόδων
Υπολογιστικής Γεωμετρίας

Ιωάννης Π. Δροσίτης
Εθνικό Μετσόβιο Πολυτεχνείο



Εθνικό Μετσόβιο Πολυτεχνείο
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Υπολογιστικών Συστημάτων

ΔΡΟΜΟΛΟΓΗΣΗ ΦΩΛΙΑΣ ΜΕΝΩΝ ΒΡΟΧΩΝ

ΜΕ ΤΗ ΧΡΗΣΗ ΜΕΘΟΔΩΝ

ΥΠΟΛΟΓΙΣΤΙΚΗΣ ΓΕΩΜΕΤΡΙΑΣ

Διδακτορική Διατριβή

Επιβλέπων Καθηγητής:

Γεώργιος Παπακωνσταντίνου, Καθηγητής Ε. Μ. Πολυτεχνείου

Συμβουλευτική Επιτροπή:

Γεώργιος Παπακωνσταντίνου	Καθηγητής Ε. Μ. Πολυτεχνείου
Παναγιώτης Τσανάκας	Καθηγητής Ε. Μ. Πολυτεχνείου
Βασίλειος Μάγκλαρης	Καθηγητής Ε. Μ. Πολυτεχνείου

Εξεταστική Επιτροπή:

Γεώργιος Παπακωνσταντίνου	Καθηγητής Ε. Μ. Πολυτεχνείου
Παναγιώτης Τσανάκας	Καθηγητής Ε. Μ. Πολυτεχνείου
Βασίλειος Μάγκλαρης	Καθηγητής Ε. Μ. Πολυτεχνείου
Ευστάθιος Ζάχος	Καθηγητής Ε. Μ. Πολυτεχνείου
Ανδρέας Σταφυλοπάτης	Καθηγητής Ε. Μ. Πολυτεχνείου
Κεμάλ Πεκμεστζή	Καθηγητής Ε. Μ. Πολυτεχνείου
Νεκτάριος Κοζύρης	Λέκτορας Ε. Μ. Πολυτεχνείου

Η στοιχειοθεσία του κειμένου έγινε με το πρόγραμμα L^AT_EX και τις ελληνικές γραμματοσειρές CB Greek του Claudio Beccari για το πωάτο babel.

Copyright© 2002 by Ioannis P. Drositis

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without permission of the author.

*στους γονείς μου
Παναγιώτη και Φωτεινή
στις αδερφές μου
Πέπη και Αγγελίνα
στη γιαγιά μας*

*στους αγαπητούς μου
Αχιλλέα και Θάλεια*

Περίληψη

Πρωταρχικό πρόβλημα στο χώρο της παράλληλης επεξεργασίας, αποτελεί η δρομολόγηση φωλιασμένων βρόχων. Η παρούσα διατριβή μελετά τη δρομολόγηση φωλιασμένων βρόχων με ομοιόμορφες εξαρτήσεις, εστιάζοντας στις γεωμετρικές ιδιότητες των εξαρτήσεων και του χώρου δεικτών.

Παρουσιάζεται αρχικά, ένας νέος αλγόριθμος απεικόνισης σε αρχιτεκτονικές κατανεμημένης μνήμης. Η τεχνική που ακολουθεί βασίζεται στην απεικόνιση σε συστολικές διατάξεις. Μετά τον κατάλληλο μετασχηματισμό του χώρου δεικτών, απαιτείται η διαμέριση και η απεικόνισή του στη δεδομένη αρχιτεκτονική. Η μέθοδος διαμέρισης που προτείνεται, αξιοποιεί τις διευθύνσεις των συνόρων του μετασχηματισμένου χώρου. Παρότι δεν παρουσιάζει τη βέλτιστη πολυπλοκότητα, επιτυγχάνει αποδοτική απεικόνιση σε επεξεργαστές, ισοκατανέμοντας τον φόρτο εργασίας.

Μελετώνται κατόπιν, οι γεωμετρικές ιδιότητες του χώρου δεικτών, για τα προβλήματα φωλιασμένων βρόχων με ομοιόμορφες εξαρτήσεις. Σχηματίζονται οι εξισώσεις που δίνουν την κυματομορφή εκτέλεσης για κάθε χρονική στιγμή. Παρά το γεγονός ότι απαιτείται ακέραιος γραμμικός προγραμματισμός για τη λύση των εξισώσεων αυτών, είναι η πρώτη φορά που παρουσιάζονται σε κλειστή μορφή. Σημειώνονται επιπλέον ειδικές περιπτώσεις, κατά τις οποίες οι κυματομορφές εκτέλεσης είναι απολύτως ομοιόμορφες και επαναλαμβανόμενες. Στις περιπτώσεις αυτές, τα διανύσματα εξάρτησης του αλγορίθμου επαληθεύουν κάποιες ανισότητες και η πολυπλοκότητα υπολογισμού των κυματομορφών εκτέλεσης μειώνεται δραματικά: υπολογίζονται με πολυωνυμικό ή ακόμα και γραμμικό κόστος.

Με βάση την ανάλυση των γεωμετρικών ιδιοτήτων, αναδεικνύεται η χρησιμότητα και η εφαρμογή τους στα προβλήματα δρομολόγησης φωλιασμένων βρόχων. Παρουσιάζεται, αρχικά, ένας αλγόριθμος δρομολόγησης, ο οποίος αφορά τα προβλήματα που ακολουθούν το μοντέλο μναδιαίου κόστους υπολογισμού - μηδενικού κόστους επικοινωνίας - *UET*. Η δρομολόγηση που προτείνεται είναι ασυμπτωτικά βέλτιστη και εμφανίζει μικρότερο κόστος ανά επανάληψη από τις ήδη υπάρχουσες μεθόδους.

Στη συνέχεια, παρουσιάζεται μια μέθοδος αναγωγής των προβλημάτων που ακολουθούν το μοντέλο μοναδιαίου κόστους υπολογισμού - μοναδιαίου κόστους επικοινωνίας - *UET-UCT*, σε ισοδύναμα προβλήματα μηδενικού κόστους επικοινωνίας -έννοια ισοδυνάμου *UET* πρόβληματος. Επιτυγχάνεται έτσι, η δρομολόγηση των προβλημάτων που εμπεριέχουν κόστος επικοινωνίας, τόσο από τη μέθοδο που προτείνουμε, όσο και από οποιαδήποτε μέθοδο της βιβλιογραφίας που αναφέρεται σε προβλήματα *UET*.

Στις μεθόδους δρομολόγησης που παρουσιάζονται, ακολουθείται μια εναλλακτική μέθοδος υπολογισμού του διανύσματος δρομολόγησης *P*. Η μέθοδος αυτή βασίζεται αποκλειστικά στις σχέσεις που συνδέουν τα διανύσματα εξάρτησης και τον τρόπο που αυτά επιτρέπουν την εκτέλεση των στιγμιοτύπων του χώρου δεικτών. Αποτελεί μια παραλλαγή της μεθόδου υπολογισμού του κυρτού περιγράμματος συνόλου σημείων στο *n*-διάστατο χώρο, ο οποίος ακούει στο όνομα *QuickHull*. Το ιδιαίτερο χαρακτηριστικό αυτής της εναλλακτικής μεθόδου υπολογισμού του διανύσματος δρομολόγησης, είναι ότι παρουσιάζει εξαιρετικά χαμηλή πολυπλοκότητα, σε σύγκριση με τις υπάρχουσες μεθόδους της βιβλιογραφίας.

Προτείνεται τέλος, η βασική ιδέα για τη γενίκευση των παραπάνω γεωμετρικών μεθόδων δρομολόγησης στις *n* διαστάσεις.

Για την αξιολόγηση των μεθόδων δρομολόγησης της παρούσας διατριβής, πραγματοποιήθηκε μια σειρά υλοποιήσεων. Με τη βοήθεια των υλοποιήσεων αυτών, συγκρίνεται η προτεινόμενη γεωμετρική μέθοδος δρομολόγησης τόσο με τη μέθοδο των υπερεπιπέδων, όσο και με την εξαντλητική μέθοδο δρομολόγησης, την επικαλούμενη βέλτιστη. Τα αποτελέσματα που εξάγονται παρατίθενται με τη μορφή πινάκων και διαγραμμάτων.

Abstract

One of the major problems in the area of parallel processing, is scheduling nested loops. This thesis studies the problem of scheduling nested loops with uniform dependencies, focusing onto the *geometric properties* of the index space and the dependence vectors of the problem.

Initially, a new algorithm for mapping onto distributed memory architectures is presented. The followed technique is based on mapping onto systolic arrays. After the index space transformation is performed, the transformed index space needs to be partitioned and mapped to the given architecture. The partitioning is made along the directions of the transformed index space boundaries. This method achieves optimal load balancing between processors, although scheduling complexity is not optimal.

After that, the geometric properties of the index space considered in the uniform nested loop problems are studied. The formal expressions that give the *execution waveform* of any time instance are presented. Although they can only be solved by integer linear programming techniques, it is the first time they are computed in closed form. Some special cases are also noted. In these cases, certain inequalities between the algorithm dependence vectors hold and the complexity of computing the execution waveform decreases dramatically: it can be computed in polynomial or even linear time.

Based on the above geometric properties, a technique for scheduling any nested loop problem that follows the *unit execution - zero communication time model - UET* is presented. This scheduling method is nearly optimal.

In addition to the above, we introduce a method of reducing problems that consider the *unit execution - unit communication time model - UET-UCT* to pure UET equivalent ones. This is one of the main contributions of our work, as it enables any scheduling technique of the bibliography for the UET cases, to apply to any problem that considers unit communication

cost.

During the scheduling algorithms presented in this thesis, a new method of computing the hyperplane vector Π is concerned. This method is explicitly based on the algorithm dependence vectors and on how these vectors allow the execution of the index space points. Actually, it is a revision of the convex hull computing method, called **QuickHull**. It should be noted that, in comparison with all other methods of the bibliography, it achieves quite a small complexity.

Finally, the generalization of the above scheduling techniques for n -dimensional problems is presented.

Certain simulation programs have been implemented, in order to testify the scheduling methods of this thesis. The proposed geometric method is compared with the hyperplane one, as well as with the exhaustive scheduling procedure, also called optimal. The comparison results are analytically presented.

Περιεχόμενα

Περίληψη	v
Abstract	vii
Κατάλογος Σχημάτων	xv
Κατάλογος Πινάκων	xvii
Αντί Προλόγου	xix
1 Εισαγωγή	1
1.1 Αντικείμενο της Διατριβής	2
1.2 Οργάνωση της Διατριβής	3
1.3 Συμβολή της Διατριβής	5
1.4 Δημοσιεύσεις	7
2 Βασικές Έννοιες & Ορισμοί	9
2.1 Το Μοντέλο του Προβλήματος	9
2.2 Χρονικές και Γεωμετρικές περιοχές	11
2.3 Βιβλιογραφική Επισκόπηση	15
2.3.1 Η Μέθοδος των Υπερεπιπέδων	15
2.3.2 Εξελίξεις της Μεθόδου των Υπερεπιπέδων	16
2.3.3 Προσδορισμός του Βέλτιστου Γραμμικού Μετασχηματισμού	17

2.3.4 Μη Γραμμικές Δρομολογήσεις	19
2.3.5 Δρομολόγηση με Κόστος Επικοινωνίας	19
3 Απεικόνιση Βρόχων σε Αρχιτεκτονικές Κατανεμημένης Μνήμης	23
3.1 Μετασχηματισμός του Χώρου Δεικτών	24
3.2 Ορολογία και Ορισμοί	25
3.2.1 Μετασχηματισμοί Αλγορίθμων	27
3.3 Τεχνικές Διαμέρισης και Απεικόνισης	30
3.4 Απεικόνιση και Κόστος Επικοινωνίας	31
3.5 Κόστος Επικοινωνίας Απεικονίσεων & Τομών	34
3.6 Καθορισμός των Πιθανών Διευθύνσεων Τομής	36
3.7 Αξιολόγηση και Επιλογή Απεικόνισης	39
3.7.1 Προσδιορισμός Κόστους Επικοινωνίας Τομών	39
3.7.2 Προσδιορισμός Βέλτιστης Απεικόνισης	40
3.7.3 Προσδιορισμός Επιφάνειας Τομής	42
3.8 Παράδειγμα Απεικόνισης	44
4 Υπολογισμός των Περιγραμμάτων Εκτέλεσης	53
4.1 Προσδιορισμός των Περιγραμμάτων Εκτέλεσης	54
4.1.1 Σχηματισμός του Αρχικού Σχήματος Κυματομορφής	54
4.1.2 Προσδιορισμός του Αρχικού Περιγράμματος	56
4.1.3 Προσδιορισμός των Περιγραμμάτων Ανώτερης Τάξης	59
4.2 Αναλυτική Περιγραφή των Περιγραμμάτων	61
4.3 Αναλυτική Περιγραφή Προβλημάτων	63
4.4 Συνθήκες Ομοιομορφίας των Σχημάτων Κυματομορφής	64
4.5 Πολυπλοκότητα Υπολογισμού των Περιγραμμάτων	68
5 Γεωμετρική Δρομολόγηση στις Δύο Διαστάσεις	73
5.1 Γενική Περιγραφή της Μεθόδου	73
5.2 Ορολογία και Ορισμοί	75

5.3 Περιγραφή Κώνων και Περιοχών	78
5.4 Προσδιορισμός των Κώνων	80
5.5 Προσδιορισμός των Περιοχών	83
5.6 Γεωμετρική Δρομολόγηση με Κώνους & Περιοχές	84
5.7 Συσχέτιση με το Πρόβλημα του Convex Hull	87
5.7.1 Εναλλακτικός Υπολογισμός των Cone Vectors	88
5.7.2 Εναλλακτικός Ορισμός των Περιοχών	91
5.8 Θέματα Εγγυρότητας & Βελτιστοποίησης	91
5.8.1 Εγκυρότητα με βάση τα Κυρτά Περιγράμματα	94
5.8.2 Βελτιστοποίηση	96
5.9 Πολυπλοκότητα	99
6 Γεωμετρική Δρομολόγηση Προβλημάτων UET-UCT	101
6.1 Το Μοντέλο UET-UCT	103
6.2 Σχηματισμός των Περιγραμμάτων Εκτέλεσης	104
6.3 Αναγωγή UET-UCT Προβλημάτων σε Ισοδύναμα UET	109
6.4 Δρομολόγηση Προβλημάτων UET-UCT	111
6.5 Εγκυρότητα & Βελτιστοποίηση	114
6.6 Πολυπλοκότητα	115
7 Γεωμετρική Δρομολόγηση σε Περισσότερες Διαστάσεις	117
7.1 Ορολογία & Ορισμοί	118
7.2 Υπολογισμός των Μη-Τετριμένων Κώνων	122
7.3 Υπολογισμός των Τετριμένων Κώνων	123
7.4 Γεωμετρική Δρομολόγηση στις n Διαστάσεις	126
7.5 Πολυπλοκότητα	128
8 Τλοποιήσεις και Αποτελέσματα	129
8.1 Τρισδιάστατη Απεικόνιση Εκτέλεσης (cubePlaying)	129
8.2 Γεωμετρική Δρομολόγηση Αλγορίθμων (crazySched)	131

8.3 Αποτελέσματα & Συγκρίσεις	134
8.4 Σχολιασμός των Αποτελεσμάτων	138
9 Συμπεράσματα και Προτάσεις	143
Α' Αντιστοιχία Ελληνικών-Αγγλικών Όρων	145
Β' Οι Αλγόριθμοι Κυρτού Περιγράμματος DivideHull2d και QuickHull	147
B'.1 Ο Αλγόριθμος DivideHull2d	147
B'.2 Ο Αλγόριθμος QuickHull	150
B'.3 Το πρόγραμμα qhull	156
Βιβλιογραφία	157

Κατάλογος Σχημάτων

2.1	Τα σύνολα ECT_i και Pat_i , $1 \leq i \leq 4$, για αλγόριθμο με 3 εξαρτήσεις	12
2.2	Απόλυτες και σχετικές διαφορικές εξαρτήσεις	14
3.1	Χώρος δεικτών αλγορίθμου διάστασης 3	25
3.2	Μετασχηματισμένος χώρος δεικτών 2 διαστάσεων	26
3.3	Αρχικός χώρος δεικτών, διάστασης 3 και ο αντίστοιχος μετασχηματισμένος . .	29
3.4	Μετασχηματισμένος χώρος διάστασης 2 και η προς απεικόνιση διάταξη	30
3.5	Οι μέθοδοι απεικόνισης LPGS και GPLS	32
3.6	Απεικόνιση με βάση δύο διαμερίσεις, σε διάταξη με 3 επεξεργαστές	33
3.7	Τυπολογισμός της παραμέτρου ρ	35
3.8	Εύρεση των συνοριακών γραμμών μέσω του Αλγορίθμου 3.1	38
3.9	Εναλλακτικοί τρόποι διαμέρισης ενός μετασχηματισμένου χώρου δεικτών . . .	41
3.10	Εύρεση μήκους τομής σε μετασχηματισμένο χώρο δεικτών διάστασης 2	43
3.11	Τρισδιάστατος χώρος δεικτών παραδείγματος απεικόνισης	45
3.12	Ο μετασχηματισμένος χώρος δεικτών του παραδείγματος	47
3.13	Διαμέριση κατά τις διευθύνσεις των συνόρων του χώρου	49
3.14	Η διαμέριση και η απεικόνιση που προτείνονται για το παράδειγμά μας	50
4.1	Σχηματισμοί υποχώρων λόγω των διανυσμάτων εξάρτησης	55
4.2	Σχηματισμός του αρχικού pattern ως τομή υποχώρων	57
4.3	Σχηματισμός του περιγράμματος του pattern του βήματος i	60

4.4	Τα περιγράμματα των 3 πρώτων patterns, μέσω των διαφορικών εξαρτήσεων	61
4.5	Περίγραμμα εκτέλεσης που παρουσιάζει πλήρη ομοιομορφία	65
4.6	Το περίγραμμα του κάθε σχήματος χυματομορφής ως σύνθεση ξεχωριστών τμημάτων	66
4.7	Επεξήγηση της συνθήκης ομοιομορφίας (Θεώρημα 4.1)	68
4.8	Μία δεύτερη συνθήκη ομοιόμορφου περιγράμματος	69
5.1	Σχηματισμός των συνόλων ECT_i και των γεωμετρικών περιοχών R_i	75
5.2	Ο χώρος δεικτών ενός αλγορίθμου με 6 διανύσματα εξάρτησης	77
5.3	Εντοπισμός σημείου σε χώρο δεικτών με 3 κώνους	79
5.4	Υπολογισμός των διανυσμάτων κώνων και σχηματισμός των κώνων	82
5.5	Διαμέριση του χώρου δεικτών σε περιοχές	84
5.6	Παράδειγμα 5.1: γεωμετρική δρομολόγηση αλγορίθμου με 5 εξαρτήσεις	86
5.7	Υπολογισμός των cone vectors μέσω του υπολογισμού του convex hull	88
5.8	Επεξήγηση του Θεωρήματος 5.1	89
5.9	Η ακολουθία των κυρτών περιγραμμάτων	90
5.10	Εξήγηση της εγκυρότητας της γεωμετρικής μεθόδου	92
5.11	Εξήγηση της βέλτιστης δρομολόγησης της γεωμετρικής μεθόδου	94
5.12	Ιδιότητα των εσωτερικών σημείων των κυρτών περιγραμμάτων	95
5.13	Επεξήγηση του βελτίστου της γεωμετρικής μεθόδου	97
5.14	Το βέλτιστο της γεωμετρικής μεθόδου αναλυτικά	98
6.1	Δρομολόγηση σε αρχιτεκτονικές UET-UCT	102
6.2	Περιγράμματα εκτέλεσης σε αλγόριθμο UET-UCT	105
6.3	Αντιπαραβολή εκτέλεσης UET-UCT και UET ισοδυνάμου (I)	110
6.4	Αντιπαραβολή εκτέλεσης UET-UCT και UET ισοδυνάμου (II)	111
6.5	Χωρισμός περιοχών και δρομολόγηση προβλήματος UCT	113
7.1	Τετριμένος κώνος σε χώρο τριών διαστάσεων	119
7.2	Εσωτερικές και εξωτερικές πλευρές κώνου στις τρεις διαστάσεις	121

7.3	Τυπερεπίπεδο δρομολόγησης για τετριμμένο κώνο	125
7.4	Τυπερεπίπεδα μη-τετριμμένων και τετριμμένων κώνων στις 2 διαστάσεις	126
8.1	Παράδειγμα εκτέλεσης του προγράμματος <code>cubePlaying (I)</code>	130
8.2	Παράδειγμα εκτέλεσης του προγράμματος <code>cubePlaying (II)</code>	131
8.3	Παράδειγμα εκτέλεσης του προγράμματος <code>cubePlaying (III)</code>	132
8.4	Περιγραφή και λειτουργία της πλατφόρμας <code>crazySched</code>	134
8.5	Εκτέλεση με τη μέθοδο των υπερεπιπέδων και τη γεωμετρική	135
8.6	Συγκριτικά εκτέλεσης αλγορίθμων ανά σημείο για αριθμό εξαρτήσεων 3-6 . . .	138
8.7	Συγκριτικά εκτέλεσης αλγορίθμων ανά σημείο για αριθμό εξαρτήσεων 7-10 . . .	139
8.8	Συγκριτικά εκτέλεσης αλγορίθμων ανά σημείο για αριθμό εξαρτήσεων 11-15 .	140
B'.1	Παράδειγμα εφαρμογής του αλγορίθμου <code>DivideHull2d</code>	149
B'.2	Εφαρμογή του αλγορίθμου <code>QuickHull</code> στις δύο διαστάσεις	152
B'.3	Ορίζοντας πολυτόπου ως προς σημείο στις τρεις διαστάσεις	153
B'.4	Πρόσθεση σημείου σε κυρτό περίγραμμα	154
B'.5	Γράφος συγκρούσεων κατά την κατασκευή κυρτού περιγράμματος	155
B'.6	Συνεπίπεδες πλευρές κατά την δημιουργία κυρτού περιγράμματος	155

Κατάλογος Πινάκων

8.1 Σύγκριση των μεθόδων δρομολόγησης για μέγεθος προβλήματος 100 σε κάθε διάσταση	136
8.2 Σύγκριση των μεθόδων δρομολόγησης για μέγεθος προβλήματος 500 σε κάθε διάσταση	136
8.3 Σύγκριση των μεθόδων δρομολόγησης για μέγεθος προβλήματος 1.000 σε κάθε διάσταση	137
A'.1 Αντιστοιχία ελληνικών-αγγλικών όρων	145
A'.2 Αντιστοιχία ελληνικών-αγγλικών όρων (συνέχεια)	146

Αντί Προλόγου

Η διδακτορική αυτή διατριβή εκπονήθηκε στον τομέα *Τεχνολογίας Υπολογιστών* & *Πληροφορικής* του τμήματος *Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών* του Εθνικού Μετσοβίου Πολυτεχνείου. Περιλαμβάνει την έρευνα και τα αποτελέσματα που προέκυψαν κατά τη διάρκεια των μεταπτυχιακών μου σπουδών στο *Εργαστήριο Υπολογιστικών Συστημάτων* του εν λόγω τμήματος.

Το αντικείμενο με το οποίο ασχολήθηκα κατά τη διάρκεια των μεταπτυχιακών μου σπουδών, είναι η *Παράλληλη Επεξεργασία* και ιδιαίτερα, οι αλγόριθμοι δρομολόγησης φωλιασμένων βρόχων με βάση τις γεωμετρικές ιδιότητες που παρουσιάζει η αναπαράσταση τους. Η μελέτη των ιδιοτήτων αυτών, ανοίγει νέους τρόπους προσέγγισης κλασικών προβλημάτων του χώρου, εισάγοντας αποτελέσματα του χώρου της *Υπολογιστικής Γεωμετρίας*.

Μέσα από τις γραμμές αυτές ωστε να εκφράσω τις ευχαριστίες μου σε ένα πλήθος ανθρώπων που, όχι μόνο με βοήθησαν, αλλά συνέβαλλαν αποφασιστικά στην πραγματοποίηση της παρούσας διατριβής. Πρώτα απ' όλους ωστε να ευχαριστήσω τον καθηγητή μου κ. Γεώργιο Παπακωνσταντίνου, ο οποίος είναι και ο επιβλέπων καθηγητής της παρούσας διατριβής, για την επιστημονική επίβλεψη την οποία μου παρείχε. Κυρίως όμως τον ευχαριστώ για την κατανόηση και τη συμπαράσταση την οποία μου έδειξε, ιδίως σε περιόδους απογοητεύσεων και δυσκολιών.

Τις ωρεμές μου ευχαριστίες ωστε να εκφράσω και στον δεύτερο καθηγητή του εργαστηρίου μας κ. Παναγιώτη Τσανάκα, τόσο για τη βοήθεια που μου παρείχε, όσο και για τον εμπλουτισμό της «κομπιουτεριστικής» μας μονοτονίας. Ιδιαίτερως ωστε να ευχαριστήσω τους κ. Ευστάθιο Ζάχο και Νεκτάριο Κοζύρη, για τα πολλά που έμαθα στο πλευρό τους και για τους νέους ορίζοντες που μου ανοίχθηκαν με τη βοήθεια τους. Οφείλω ακόμη ένα μεγάλο ευχαριστώ στους καθηγητές μου Γιάννη Μαΐστρο και Ανδρέα Σταφυλοπάτη για τη συμπαράσταση και την αγάπη που μου έδειξαν.

Ιδιαιτέρως όμως θα ήθελα να σταθώ στους αγαπητούς φίλους και συνεργάτες μου Παναγιώτη Θεοδωρόπουλο, Θοδωρή Ανδρόνικο, Νίκο Βλάσση, Γιώργο Οικονομάκο, Άγγελο Θάνο, Νατάσσα Μανουσοπούλου, Ανδρέα Κουλούρη, Γιάννη Πουλάκη, Γιώργο Γκούμα, Άρη Σωτηρόπουλο και Στέργιο Στεργίου, οι οποίοι όχι μόνο μου συμπαραστάθηκαν ποικιλοτρόπως, αλλά συνέβαλαν σημαντικά στην αποπεράτωση της διατριβής μου. Ιδιαιτέρως ευχαριστώ τον αγαπητό φίλο Γιάννη Παναγόπουλο για την καθοριστική συμβολή του στην υλοποίηση των μεθόδων της παρούσας διατριβής και όχι μόνο. Ευχαριστώ πολύ παιδιά!

Η εργασία αυτή αφιερώνεται σε όσους επιμένουν να μοχθούν παρά τη μη καταξίωση των επιστημονικών τους ερευνών και σε όσους αγαπώ και νιώθω κοντά μου.

Καλή ανάγνωση!

Αθήνα, Μάιος 2002
Ιωάννης Π. Δροσίτης

Κεφάλαιο 1

Εισαγωγή

Λόγω της θεαματικής προόδου της τεχνολογίας κατά την τελευταία εικοσαετία, παρατηρήθηκαν σημαντικές αλλαγές, τόσο στη διαθεσιμότητα, όσο και στην αρχιτεκτονική των υπολογιστικών μηχανών. Με άλλα λόγια, παλαιότερα εψεωρείτο «πολυτέλεια» η δέσμευση πολλών μονάδων επεξεργασίας, για τη δημιουργία μιας ισχυρής υπολογιστικής μηχανής.

Στις μέρες μας, υπάρχει η δυνατότητα σύνθεσης τέτοιων ισχυρών υπολογιστικών μηχανών, με σχετικά χαμηλό κόστος. Έτσι, οι αρχιτεκτονικές οι οποίες διαθέτουν πλήθος μονάδων επεξεργασίας, κάθε άλλο παρά ιδιαίτερη περίπτωση αποτελούν πλέον.

Από την άλλη πλευρά, υπάρχουν προγράμματα των οποίων η εκτέλεση –ακόμα και σε υπερ-υπολογιστές– διαρκεί ώρες ή ακόμα και μέρες. Η πλέον χρονοβόρα πηγή καινοτέρησης στο μεγαλύτερο ποσοστό των προβλημάτων αυτών, είναι οι επαναληπτικές δομές εκτέλεσης που περιέχει· δομές που, στη συντριπτική πλειοψηφία τους, είναι πολλαπλοί φωλιασμένοι βρόχοι (*multiple nested loops*).

Αρκετές τεχνικές και μέθοδοι, σε διάφορους τομείς της επιστήμης των υπολογιστών, παρουσιάζουν εξαιρετικά αυξημένο υπολογιστικό κόστος, λόγω των πολλαπλών φωλιασμένων βρόχων που αποτελούν τον πυρήνα τους. Υποδεικνύεται, με τον τρόπο αυτό, η σημαντικότητα της όσο το δυνατόν γρηγορότερης εκτέλεσης των φωλιασμένων βρόχων, σε οποιαδήποτε προσπάθεια επιτάχυνσης της εκτέλεσης ενός προγράμματος.

1.1 Αντικείμενο της Διατριβής

Ο τομέας της επιστήμης των υπολογιστών που ασχολείται με τα προβλήματα αυτά, είναι η παράλληλη επεξεργασία. Στη θεματική περιοχή της εντάσσεται η παραλληλοποίηση αλγορίθμων, η σχεδίαση, μελέτη και αξιοποίηση παράλληλων και κατανεμημένων αρχιτεκτονικών, η μελέτη των τρόπων επικοινωνίας και διασύνδεσης αυτών, κ.α.

Όσον αφορά τους αλγορίθμους, μελετώνται αφενός οι τρόποι και οι τεχνικές μετατροπής των κλασικών αλγορίθμων σε παράλληλους, εκμεταλλεύομενοι το γεγονός της διαθεσιμότητας πολλαπλών μονάδων επεξεργασίας. Αφετέρου, λόγω των νέων δεδομένων των διαθέσιμων αρχιτεκτονικών, γίνεται προσπάθεια επανασχεδίασης τους, με σκοπό την εξαγωγή του εγγενούς παραλληλισμού που τυχόν παρουσιάζουν. Μιλάμε τότε για μια εξ' αρχής σχεδίαση του αλγορίθμου και όχι για την παραλληλοποίηση του αντίστοιχου σειριακού.

Μεγάλο τμήμα των αλγορίθμων που παρουσιάζονται στην περιοχή της παράλληλης επεξεργασίας, εστιάζονται στην παραλληλοποίηση της εκτέλεσης φωλιασμένων βρόχων. Αυτό συμβαίνει για δύο λόγους κυρίως. Αφενός, γιατί αποτελούν τα πλέον χρονοβόρα τμήματα των προγραμμάτων που απαιτούν μεγάλη υπολογιστική ισχύ, όπως προείπαμε και αφετέρου, γιατί προσφέρονται συνήθως για παραλληλοποίηση, λόγω της κανονικότητας και της ομοιομορφίας που παρουσιάζουν. Αυτό συμβαίνει γιατί, σε ένα σύστημα πολλών μονάδων επεξεργασίας, είναι ασύγκριτα ευκολότερη η επεξεργασία και η εκτέλεση του ίδιου τμήματος κώδικα (με διαφορετικές μεταβλητές, προφανώς) από όλες τις μονάδες, παρά η εκτέλεση διαφορετικών τμημάτων κώδικα από την κάθε μια, λόγω των ιδιόμορφων συνθηκών συγχρονισμού που απαιτούνται –θεωρούμε πάντα ότι αναφερόμαστε σε τμήματα κώδικα με εξαρτήσεις από το ένα στο άλλο.

Η διαδικασία αποδοτικής εκτέλεσης ενός προγράμματος, σε κάποιο σύγχρονο παράλληλο υπολογιστή, περιλαμβάνει συνήθως, τα εξής στάδια:

- εξαγωγή του εγγενούς παραλληλισμού σε όλα τα επίπεδα της εφαρμογής,
- εύρεση των επιμέρους εργασιών και χρονική δρομολόγηση τους στον ελάχιστο συνολικό χρόνο και
- αποδοτική διαμέριση των επιμέρους εργασιών στα διάφορα υπολογιστικά στοιχεία της παράλληλης αρχιτεκτονικής.

Επομένως, στην προσπάθεια επίτευξης της βέλτιστης επιτάχυνσης της εκτέλεσης ενός προγράμματος σε κάποιον παράλληλο υπολογιστή, απαιτείται τόσο η θεωρητική παραλληλοποίηση των

χρονοβόρων τμημάτων του, όσο και η μελέτη των ιδιαιτεροτήτων της αρχιτεκτονικής του παράλληλου υπολογιστή.

Η παρούσα διατριβή ασχολείται με τη θεωρητική βέλτιστη παραλληλοποίηση αλγορίθμων περιγραφών που περιέχουν τέτοιους φωλιασμένους βρόχους, υπό το πρίσμα των γεωμετρικών ιδιοτήτων του χώρου στον οποίο αναφέρονται. Χρησιμοποιεί ορισμένες μεθόδους και κάποια συμπεράσματα από το χώρο της υπολογιστικής γεωμετρίας, τα οποία επιτρέπουν την προσέγγιση θεμελιωδών εννοιών του χώρου, με τελείως διαφορετικό τρόπο. Προσθέτει στη βιβλιογραφία της παράλληλης επεξεργασίας μια σειρά μεθόδων και αποτελεσμάτων, που προάγουν την αποδοτική υλοποίηση των αλγορίθμων που βασίζονται στη δομή των φωλιασμένων βρόχων.

Απότερος στόχος μας είναι οι μέθοδοι αυτές να εφαρμοστούν σε παράλληλους μεταγλωτιστές ή άλλα εργαλεία ανάπτυξης παράλληλων εφαρμογών, κάνοντας έτσι τη διαδικασία της αυτόματης παραλληλοποίησης αποδοτικότερη και ευκολότερη.

Στα κεφάλαια που ακολουθούν, παρουσιάζονται οι μέθοδοι που αναπτύχθηκαν κατά την εκπόνηση της παρούσας διατριβής και γίνεται σύγκριση με τις αντίστοιχες μεθόδους της βιβλιογραφίας.

1.2 Οργάνωση της Διατριβής

Η παρούσα διατριβή μελετά τη δρομολόγηση βρόχων με ομοιόμορφες εξαρτήσεις, εστιάζοντας στις γεωμετρικές ιδιότητες των εξαρτήσεων και του χώρου δεικτών.

Στο παρόν Κεφάλαιο συνοψίζεται το αντικείμενο τις διατριβής, ο τρόπος με τον οποίο οργανώνεται και οι δημοσιεύσεις που προέκυψαν κατά τη διάρκεια της εκπόνησής της. Στο **Κεφάλαιο 2** περιγράφεται το μοντέλο του προβλήματος το οποίο μελετάται και σημειώνονται οι βασικοί ορισμοί που χρησιμοποιούνται στη συνέχεια. Γίνεται αναφορά στις αντίστοιχες μεθόδους της βιβλιογραφίας και στα αποτελέσματα στα οποία κατέληξαν αυτές.

Το ουσιαστικό μέρος της διατριβής αρχίζει με το **Κεφάλαιο 3**. Παρουσιάζεται αρχικά, ένας νέος αλγόριθμος απεικόνισης σε αρχιτεκτονικές κατανεμημένης μνήμης. Η βασική ιδέα της μεθόδου αντιστοιχεί με αυτή της απεικόνισης σε συστολικές διατάξεις. Ο χώρος δεικτών του φωλιασμένου βρόχου, αρχικά μετασχηματίζεται, χρησιμοποιώντας μεθόδους της βιβλιογραφίας. Ο μετασχηματισμένος χώρος δεικτών που προκύπτει, παρουσιάζει χαρακτηριστικά τα οποία επιτρέπουν την αποδοτικότερη απεικόνισή του στη διαθέσιμη αρχιτεκτονική. Η μέθοδος που προτείνεται, διαμερίζει τον μετασχηματισμένο χώρο και απεικονίζει τα τμήματα που προ-

κύπτουν στις διαθέσιμες μονάδες επεξεργασίας, αξιοποιώντας τις διευθύνσεις των συνόρων του μετασχηματισμένου χώρου. Παρότι δεν παρουσιάζει τη βέλτιστη πολυπλοκότητα, σε ορισμένες περιπτώσεις επιτυγχάνει αποδοτικότερη απεικόνιση από τις υπάρχουσες μεθόδους, ισοχατανέμοντας τον φόρτο εργασίας του αρχικού προβλήματος.

Στο **Κεφάλαιο 4** μελετώνται οι γεωμετρικές ιδιότητες του χώρου δεικτών για τα προβλήματα φωλιασμένων βρόχων με ομοιόμορφες και θετικές εξαρτήσεις. Από τη μελέτη αυτή προκύπτει ότι η διάδοση της εκτέλεσης μεταξύ των σημείων του χώρου δεικτών, ακολουθεί εξαρτόμενα και επαναλαμβανόμενα σχήματα, τα οποία ονομάζονται *σχήματα κυματομορφής εκτέλεσης (patterns)*. Εισάγεται η γεωμετρική έννοια των διανυσμάτων κυματομορφής (*pattern vectors*), τα οποία αποτελούν τα διανύσματα εξάρτησης που διαμορφώνουν το περίγραμμα των σχημάτων εκτέλεσης. Με βάση αυτά τα διανύσματα, σχηματίζονται οι εξισώσεις που περιγράφουν την κυματομορφή εκτέλεσης, για κάθε χρονική στιγμή. Παρά το γεγονός ότι απαιτείται ακέραιος γραμμικός προγραμματισμός για τη λύση των εξισώσεων αυτών, είναι η πρώτη φορά που παρουσιάζονται σε κλειστή μορφή, έστω και για την ειδική περίπτωση των θετικών εξαρτήσεων. Σημειώνονται επιπλέον ειδικές περιπτώσεις, κατά τις οποίες οι κυματομορφές εκτέλεσης είναι απολύτως ομοιόμορφες και επαναλαμβανόμενες. Στις περιπτώσεις αυτές, η πολυπλοκότητα υπολογισμού τους μειώνεται δραματικά και υπολογίζονται με γραμμικό ή πολυωνυμικό κόστος.

Εισάγεται, στη συνέχεια, η έννοια των κώνων και των διανυσμάτων κώνων (*cone vectors*). Οποιοδήποτε σημείο του χώρου δεικτών μπορεί να εκφραστεί ως γραμμικός συνδυασμός k διανυσμάτων από αυτά, έτσι ώστε το αντίστοιχο άθροισμα των συντελεστών να είναι μέγιστο. Τα διανύσματα κώνων αποτελούν το γεωμετρικό ανάλογο του μακρύτερου μονοπατιού (*longest path*).

Με βάση τα διανύσματα κώνων, παρουσιάζεται στο **Κεφάλαιο 5**, ένας αλγόριθμος δρομολόγησης για τα προβλήματα που ακολουθούν το μοντέλο μοναδιάσιου κόστους υπολογισμού - μηδενικού κόστους επικοινωνίας - *UEC*. Ο χώρος δεικτών διαμερίζεται σε γεωμετρικά οριζόμενες περιοχές, οι οποίες αντιστοιχούν στα υποσύνολα του χώρου δεικτών που πρέπει να δρομολογηθούν σε κάθε χρονική στιγμή. Η δρομολόγηση που προτείνεται είναι ασυμπτωτικά βέλτιστη και υπερτερεί από τις ήδη υπάρχουσες μεθόδους, όσον αφορά τον χρόνο εκτέλεσης κάθε σημείου. Επέκταση αυτών αποτελεί και η μέθοδος υπολογισμού του βέλτιστου διανύσματος δρομολόγησης Π αξιοποιώντας τα διανύσματα των κώνων, η οποία χρησιμοποιείται κατά τη δρομολόγηση. Βασίζεται στον υπολογισμό του κυρτού περιγράμματος (*convex hull*) και η πολυπλοκότητα της είναι σημαντικά χαμηλότερη των ήδη υπαρχόντων μεθόδων της βιβλιογραφίας.

Μέχρις αυτού του σημείου θεωρήσαμε την ιδανική περίπτωση κατά την οποία έχουμε αρχιτεκτονικές με αμελητέο κόστος επικοινωνίας, ή ισοδύναμα, προβλήματα με εξαιρετικά υψηλό

κόστος υπολογισμού ανά επανάληψη του βρόχου (fine-grain). Στο **Κεφάλαιο 6** περιγράφεται μια πιο ρεαλιστική προσέγγιση του προβλήματος: η περίπτωση κατά την οποία το κόστος επικοινωνίας είναι συγκρίσιμο με αυτό του υπολογισμού. Παρουσιάζεται μια μέθοδος αναγωγής των προβλημάτων που ακολουθούν το μοντέλο *μοναδιαίου κόστους υπολογισμού - μοναδιαίου κόστους επικοινωνίας - UET-UCT*, σε ισοδύναμα προβλήματα μηδενικού κόστους επικοινωνίας -ισοδύναμα *UET προβλήματα*. Επιτυγχάνεται έτσι, η δρομολόγηση και των προβλημάτων που εμπεριέχουν κόστος επικοινωνίας, τόσο από τη μέθοδο που προτείνουμε, όσο και από οποιαδήποτε μέθοδο της βιβλιογραφίας για προβλήματα UET. Είναι η πρώτη φορά στη βιβλιογραφία που παρουσιάζεται μια τέτοια μέθοδος αναγωγής.

Στο **Κεφάλαιο 7** παρουσιάζονται οι γενικές ιδέες για τη γενίκευση της παραπάνω μεθόδου δρομολόγησης στις *n* διαστάσεις. Τα προγράμματα που υλοποιήθηκαν στο πλαίσιο της παρούσας διατριβής παρουσιάζονται στο **Κεφάλαιο 8**. Το πρώτο ονομάζεται *cubePlaying* και παρουσιάζει το περίγραμμα εκτέλεσης του δεδομένου προβλήματος στον τρισδιάστατο χώρο. Το δεύτερο αναφέρεται ως *crazySched* και αποτελεί μια ομάδα προγραμμάτων παραγωγής, δρομολόγησης και σύγκρισης δρομολογήσεων. Μέσω αυτής της πλατφόρμας επετεύχθει η πρακτική σύγκριση της μεθόδου μας με την επιχρατέστερη της βιβλιογραφίας, τη μέθοδο των υπερεπιπέδων.

Τέλος, τα συμπεράσματα της παρούσας διατριβής, καθώς και κάποιες σκέψεις για περαιτέρω έρευνα στο χώρο, αναλύονται και σχολιάζονται στο **Κεφάλαιο 9**. Για λόγους πληρότητας παρατίθενται στο παρότρημα οι δύο αλγόριθμοι υπολογισμού κυρτού περιγράμματος *QuickHull* και *DivideHull*. Οι αλγόριθμοι αυτοί προέρχονται από το χώρο της υπολογιστικής γεωμετρίας και αποτελούν τη βάση δύο αλγορίθμων που παρουσιάζονται στην παρούσα διατριβή.

1.3 Συμβολή της Διατριβής

Συνοπτικά, οι καινοτομίες που παρουσιάζονται στην παρούσα διατριβή είναι:

- Ένας νέος αλγόριθμος απεικόνισης σε συστολικές διατάξεις. Η ιδιαιτερότητα του αλγορίθμου αυτού είναι ότι απεικονίζει σε μονάδες επεξεργασίας, αξιοποιώντας τις συνοριακές διευθύνσεις του (μετασχηματισμένου) χώρου δεικτών. Με τον τρόπο αυτό επιτυγχάνεται βέλτιστη ισοχατανομή του φόρτου εργασίας στις μονάδες επεξεργασίας, ακόμα και στις περιπτώσεις που έχουμε κεκλιμένες συνοριακές διευθύνσεις [FN00, DKPT99].
- Μελετώνται, από τη γεωμετρική τους σκοπιά, οι ιδιότητες του χώρου δεικτών για τα προβλήματα φωλιασμένων βρόχων με ομοιόμορφες και θετικές εξαρτήσεις. Εισάγεται η έννοια

των διανυσμάτων κυματομορφής (pattern vectors), τα οποία αποτελούν τα διανύσματα εξάρτησης που επηρεάζουν την κυματομορφή εκτέλεσης. Με βάση αυτά, σχηματίζονται οι εξισώσεις οι οποίες δίνουν την κυματομορφή εκτέλεσης, για κάθε χρονική στιγμή. Σημειώνονται ειδικές περιπτώσεις, κατά τις οποίες οι κυματομορφές εκτέλεσης είναι απολύτως ομοιόμορφες και επαναλαμβανόμενες [DAA⁺01].

- Παρουσιάζεται μια εναλλακτική μέθοδος δρομολόγησης, για τα προβλήματα που περιλαμβάνουν διπλά φωλιασμένους βρόχους, η οποία βασίζεται στη γεωμετρική διαμέριση του χώρου δεικτών [DAK⁺01]. Η εν λόγω διαμέριση, γίνεται με βάση ορισμένα διανύσματα εξάρτησης, τα οποία καλούνται διανύσματα κώνων (cone vectors) και είναι αυτά που καθορίζουν, κατ' ουσίαν την εκτέλεση. Η μέθοδος αναφέρεται σε προβλήματα όπου το κόστος υπολογισμού θεωρείται μοναδιαίο, ενώ το κόστος επικοινωνίας μηδενικό (αρχιτεκτονικές που ακολουθούν το μοντέλο UET).
- Παρουσιάζεται μια μέθοδος αναγωγής των προβλημάτων που εμπεριέχουν επικοινωνία σε ισοδύναμα που δεν περιέχουν επικοινωνία, για προβλήματα διπλά φωλιασμένων βρόχων [PAD01, DAM⁺02]. Συγκεκριμένα, προβλήματα όπου το κόστος επικοινωνίας θεωρείται συγκρίσιμο με το κόστος υπολογισμού, ακολουθούν δηλαδή το μοντέλο UET-UCT, είναι δυνατόν να αναχθούν σε προβλήματα χωρίς κόστος επικοινωνίας (UET). Προφανώς, το μετασχηματισμένο πρόβλημα έχει διαφορετικές εξαρτήσεις από το αρχικό, παρουσιάζει όμως, εντελώς αντίστοιχο χρόνο εκτέλεσης.
- Στις δύο παραπάνω μεθόδους χρησιμοποιείται μια εναλλακτική μέθοδος υπολογισμού του βέλτιστου διανύσματος χρονοδρομολόγησης Π . Η μέθοδος αυτή βασίζεται στον υπολογισμό του χυρτού περιγράμματος συνόλου σημείων. Η πολυπλοκότητα που επιτυγχάνει είναι εντυπωσιακά μικρότερη από αυτές των μεθόδων της βιβλιογραφίας, λόγω του βέλτιστου του αλγορίθμου στον οποίον βασίζεται.¹
- Αναλύονται οι βασικές ιδέες που επιτρέπουν τη γενίκευση των παραπάνω μεθόδων και σε μεγαλύτερης διάστασης προβλήματα [DAK⁺02].

¹Βασίζεται στον βέλτιστο ντετερμινιστικό αλγόριθμο υπολογισμού του χυρτού περιγράμματος συνόλου σημείων στο n -διάστατο χώρο QuickHull [BDH96].

1.4 Δημοσιεύσεις

Στη συνέχεια, αναφέρονται συγκεντρωτικά οι δημοσιεύσεις που προέκυψαν από τη διατριβή αυτή, οι οποίες και περιλαμβάνουν τη μελέτη και τα συμπεράσματα που προέκυψαν.

ΔΙΕΘΝΗ ΠΕΡΙΟΔΙΚΑ

- G. Papakonstantinou, T. Andronikos and I. Drositis, **On Parallelization of UET/UET-UCT Loops**, *Neural Parallel and Scientific Computations*, vol. 9(3-4), pp. 279-318, September & December 2001.
- I. Drositis, N. Koziris, G. Papakonstantinou and P. Tsanakas, **A Systolic Approach to Loop Partitioning and Mapping into Fixed Size Distributed Memory Architectures**, *Advances in Informatics*, by D. I. Fotiadis and S. D. Nikolopoulos editors, pp. 83–96, World Scientific Publishing, University of Ioannina, Greece, 2000.

ΔΙΕΘΝΗ ΣΥΝΕΔΡΙΑ

- I. Drositis, T. Andronikos, M. Kalathas, G. Papakonstantinou and N. Koziris, **Optimal Loop Parallelization in n -Dimensional Index Spaces**, to appear in *The 2002 International Conference on Parallel and Distributed Processing Techniques and Applications - PDPTA '02*, Las Vegas, Nevada, USA.
- I. Drositis, T. Andronikos, G. Manis, G. Papakonstantinou and N. Koziris, **Geometric Scheduling of 2D UET/UET-UCT Uniform Dependence Loops**, *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing - PDP 2002*, pp. 343–350, Gran Canaria Island, Spain, IEEE Press, 2002.
- I. Drositis, T. Andronikos, A. Kokorogiannis, G. Papakonstantinou and N. Koziris, **Geometric Scheduling of 2D Uniform Dependence Loops**, *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS 2001*, pp. 259–264, Korea, 2001.
- I. Drositis, T. Andronikos, A. Alexandridi, G. Papakonstantinou and N. Koziris, **Geometric Pattern Prediction and Scheduling of Uniform Dependence Loops**, *Proceedings of the 5th Hellenic European Conference on Computer Mathematics and its Applications - HERCMA 2001*, Athens, 2001.

- I. Drositis, G. Goumas, N. Koziris, P. Tsanakas and G. Papakonstantinou, **Evaluation of Loop Grouping Methods based on Orthogonal Projection Spaces**, *Proceedings of the 2000 International Conference on Parallel Processing ICPP-2000*, Toronto, Canada, pp. 469–476, IEEE Press, 2000.
- I. Drositis, N. Koziris, G. Papakonstantinou and P. Tsanakas, **A Systolic Approach to Loop Partitioning and Mapping into Fixed Size Distributed Memory Architectures**, *Proceedings of the 7th Panhellenic Conference on Informatics*, Ioannina, Greece, pp. IV18–IV28, University of Ioannina, 1999.

ΤΕΧΝΙΚΕΣ ΑΝΑΦΟΡΕΣ

- I. Drositis, T. Andronikos and G. Papakonstantinou, **2D Pattern Prediction and Scheduling**, *Technical Report CSLAB-TR-00-01*, Computing Systems Laboratory - N.T.U.A., 2000.

ΤΠΟ ΠΡΟΕΤΟΙΜΑΣΙΑ

- I. Drositis, T. Andronikos, I. Panagopoulos and G. Papakonstantinou, **Optimal Loop Parallelization with and without Communication Cost**.

Κεφάλαιο 2

Βασικές Έννοιες & Ορισμοί

Στο κεφάλαιο αυτό παρουσιάζουμε τις βασικές έννοιες των προβλημάτων που θα ασχοληθούμε και δίνουμε τους γενικούς ορισμούς της σημειολογίας που θα ακολουθηθεί.

Τα σύνολα των πραγματικών και των ακεραίων συμβολίζονται ως \mathbb{R} και \mathbb{Z} , αντίστοιχα. Τα σύνολα των μη αρνητικών πραγματικών και των μη αρνητικών ακεραίων, συμβολίζονται ως \mathbb{R}^+ και \mathbb{Z}^+ , ενώ ως \mathbb{R}^n και \mathbb{Z}^n συμβολίζονται οι n -οστές καρτεσιανές δυνάμεις των πραγματικών και των ακεραίων, αντίστοιχα. Έντονη γραφή χρησιμοποιείται όταν αναφερόμαστε σε διανύσματα ή σημεία του n -διάστατου χώρου.

2.1 Το Μοντέλο του Προβλήματος

Οι αλγόριθμοι με τους οποίους ασχολούμαστε έχουν τη μορφή φωλιασμένων βρόχων με ομοιόμορφες εξαρτήσεις. Στο μοντέλο αυτό αναφερθήκαμε περιγραφικά στο προηγούμενο κεφάλαιο ακολουθείται, γενικά, ο συμβολισμός της σχετικής βιβλιογραφίας [Lam74, Mol93, ST91, AKT⁺95].

Όπως προείπαμε, η προσοχή μας εστιάζεται σε αλγορίθμους οι οποίοι έχουν τη μορφή n -φωλιασμένων βρόχων (n -nested loops). Περιοριζόμαστε σε μια τυπική περιγραφή τους. Η μορφή των βρόχων αυτών είναι ως εξής:

```
for (i0 = l0; i0 ≤ u0; i0++) {  
    ...  
    for (in-1 = ln-1; in-1 ≤ un-1; in-1++) {  
        S1(i);
```

```

    ...
     $S_k(\mathbf{i});$ 
}
...
}
```

όπου i_j , $0 \leq j \leq n - 1$, είναι οι μεταβλητές των βρόχων και l_j και u_j είναι ακέραιες σταθερές που αντιπροσωπεύουν τα άνω και κάτω όρια των μεταβλητών βρόχου.

Οι εντολές S_m , $1 \leq m \leq k$ είναι εντολές ανάθεσης, για τις οποίες θεωρούμε ορισμένες επιπλέον παραδοχές. Δεν περιέχουν εντολές I/O, ούτε εντολές μεταφοράς του ελέγχου έξω από τον βρόχο και δεν περιέχουν υπορουτίνες ή συναρτήσεις που τροποποιούν δεδομένα.¹

Το διάνυσμα δεικτών ορίζεται ως το διάνυσμα διάστασης n : $\mathbf{i} = (i_0, i_1, \dots, i_{n-1})$. Κάθε εντολή ανάθεσης S_i είναι της μορφής:

$$v_0 = E(v_1, v_2, \dots, v_p)$$

όπου v_0 είναι μια μεταβλητή εξόδου και v_r είναι οι μεταβλητές εισόδου, όπου $1 \leq r \leq p$. E είναι μια οποιαδήποτε έκφραση των μεταβλητών εισόδου. Κάθε μία από αυτές τις μεταβλητές είναι συνάρτηση του διανύσματος δεικτών: $v_0 = v_0(\mathbf{i})$ και $v_r = v_r(\mathbf{i})$. Ο χώρος δεικτών (index/iteration space) J^n ορίζεται ως εξής:

$$J^n = \{\mathbf{i} = (i_0, \dots, i_{n-1}), \text{ όπου } i_j \in \mathbb{Z} \text{ και } l_j \leq i_j \leq u_j, 0 \leq j \leq n - 1\}.$$

Κάθε σημείο του n -διάστατου αυτού χώρου αποτελεί ξεχωριστό στιγμιότυπο του σώματος του βρόχου. Η γεωμετρική μορφή του χώρου δεικτών είναι αυτή ενός παραλληλεπιπέδου στο n -διάστατο χώρο. Το πλήθος των σημείων που περιέχονται σ' αυτόν είναι:

$$|J^n| = \prod_{i=0}^{n-1} (u_i - l_i + 1).$$

Σημειώνουμε ότι τα σημεία του J^n είναι διατεταγμένα λεξικογραφικά· το σύμβολο ' $<_n$ ' χρησιμοποιείται για να δηλώσει αυτή τη (γραμμική) διάταξη.

Εντολές που ανήκουν στο ίδιο ή σε διαφορετικό στιγμιότυπο του βρόχου, υπάρχει περίπτωση να αναφέρονται στις ίδιες μεταβλητές. Λόγω αυτού του γεγονότος, ενδέχεται να υπάρχουν εξαρτήσεις μεταξύ των μεταβλητών. Έστω π.χ. ότι η μεταβλητή A αρχικοποιήθηκε στην επανάληψη *i*

¹Οι παραδοχές αυτές έγιναν για πρώτη φορά από τον Lamport [Lam74].

και χρησιμοποιείται στην επανάληψη j . Τότε, για την μεταβλητή A υπάρχει η εξάρτηση $\mathbf{d}_A \in \mathbb{Z}^n$, όπου $\mathbf{d}_A = \mathbf{j} - \mathbf{i}$.

Γενικά, μια κατευθυνόμενη εξάρτηση μεταξύ δύο στιγμιοτύπων $S_j(\mathbf{i}_1)$ και $S_j(\mathbf{i}_2)$, χαρακτηρίζεται από το διάνυσμα εξάρτησης $\mathbf{d}_j = \mathbf{i}_2 - \mathbf{i}_1$.

Στην έρευνα αυτή ακολουθείται το μοντέλο των ομοιόμορφων εξαρτήσεων, όπως ορίζεται στο άρθρο [MF86]. Θεωρείται δηλαδή, ότι οι υπάρχουσες εξαρτήσεις είναι σταθερές σε όλο το χώρο δεικτών. Μια οποιαδήποτε εξάρτηση αναφέρεται ως \mathbf{d}_i , $1 \leq i \leq m$. Θεωρούμε επίσης, τον πίνακα εξαρτήσεων D , ο οποίος περιέχει ως στήλες όλα τα διανύσματα εξάρτησης.

Οι αλγόριθμοι που ακολουθούν το μοντέλο που παρουσιάστηκε παραπάνω, καθορίζονται πλήρως από το χώρο δεικτών τους J^n και τον πίνακα εξαρτήσεων τους D . Ένας τέτοιος αλγόριθμος θα αναφέρεται στη συνέχεια ως $A(J^n, D)$.

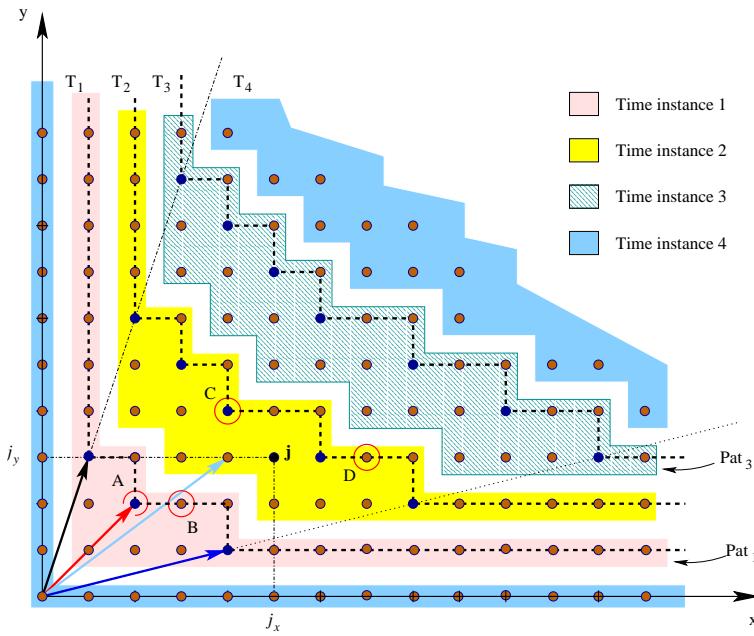
2.2 Χρονικές και Γεωμετρικές περιοχές

Ο χώρος δεικτών ενός φωλιασμένου βρόχου με ομοιόμορφες εξαρτήσεις, μπορεί να χωριστεί σε μια ακολουθία ECT_0, ECT_1, \dots από (ζεχωριστά) υποσύνολα, τέτοια ώστε το ECT_k να περιέχει ακριβώς εκείνα τα σημεία των οποίων ο νωρίτερος χρόνος εκτέλεσης (*earliest computation time - ECT*) είναι k . Για δεδομένο σημείο του χώρου j , η χρονική στιγμή μετά το πέρας της οποίας μπορεί να εκτελεστεί, καθορίζεται ως εξής: είναι αυτή κατά την οποία όλα τα σημεία από τα οποία εξαρτάται το j , είναι υπολογισμένα (και άρα διαθέσιμα). Έστω t η χρονική στιγμή αυτή. Λέμε τότε ότι $j \in ECT_t$. Εξ' ορισμού τα (προ-υπολογισμένα) σημεία του συνόρου αποτελούν το σύνολο ECT_0 . Τα σύνολα ECT_i καλούνται χρονικές περιοχές.

Η γεωμετρική αναπαράσταση των χρονικών περιοχών, σχηματίζει μια ακολουθία κλειστών πολυγώνων, τα σύνορα των οποίων μοιάζουν με κυματομορφές και καλούνται σχήματα κυματομορφής εκτέλεσης (*patterns*). Παραδείγματα τέτοιων σχημάτων φαίνονται στο σχήμα 2.1. Τα σχήματα αυτά οριοθετούν τις γεωμετρικές υποπεριοχές του χώρου δεικτών, οι οποίες έχουν τον ίδιο χρόνο εκτέλεσης.

Από τη μελέτη του σχήματος 2.1, παρατηρούμε ότι δεν συμβάλλουν όλα τα διανύσματα εξάρτησης στον σχηματισμό των περιγραμάτων των σχημάτων αυτών, αλλά μόνον ορισμένα, όπως θα δούμε αναλυτικότερα στη συνέχεια.

Σε όλη τη διάρκεια της μελέτης μας, θεωρούμε ότι το κάθε σημείο εκτελείται κατά τη μηχρότερη δυνατή στιγμή (*earliest computation time*), λαμβάνοντας υπ' όψιν όλα τα διανύσματα



Σχήμα 2.1: Θεωρούμε φωλιασμένο βρόχο με 3 διανύσματα εξάρτησης $\mathbf{d}_1 = (1, 3)$, $\mathbf{d}_2 = (2, 2)$, $\mathbf{d}_3 = (4, 1)$ και $\mathbf{d}_4 = (4, 3)$. Κάθε σκιαγραφημένο πολύγωνο αναπαριστά το υποσύνολο του J το οποίο μπορεί να εκτελεστεί τη συγκεκριμένη χρονική στιγμή. Παρουσιάζονται τα σχήματα κυματομορφής εκτέλεσης (patterns) των χρονικών στιγμών 1 έως 4. Τα σημεία του άξονα αντιστοιχούν στη χρονική στιγμή $t = 0$ (προϋπολογισμένα σημεία).

εξάρτησης. Με άλλα λόγια, μόλις είναι διαθέσιμα όλα τα σημεία από τα οποία εξαρτάται. Αυτό επιβεβαιώνει τη βέλτιστη εκτέλεση για το συνολικό χώρο δεικτών. Θεωρείται επίσης, ότι η εκτέλεση μιας οποιασδήποτε επανάληψης απαιτεί μια μονάδα χρόνου και μόνον (μοντέλο Μοναδιαίας Χρονικής Εκτέλεσης - Unit Execution Time model - UET).

Ακολουθούν οι τυπικοί ορισμοί των εννοιών που θα χρησιμοποιηθούν κατά τη διάρκεια της παρούσας μελέτης:

- ο χώρος δεικτών J διαιρείται σε μη-επικαλυπτόμενα χρονικά υποσύνολα, τα οποία συμβολίζονται ECT_i , $i \geq 0$, τέτοια ώστε το ECT_i περιέχει τα σημεία του J , των οποίων η μικρότερη δυνατή χρονική στιγμή υπολογισμού είναι i : εξ' ορισμού, το ECT_0 περιέχει τα συνοριακά (προϋπολογισμένα) σημεία.
- Θεωρούμε ένα φωλιασμένο βρόχο, μαζί με τα διανύσματα εξάρτησης του και τον χώρο

δεικτών. Τα γεωμετρικά σχήματα που περικλείουν τα ECT_i καλούνται **σχήματα κυματομορφής εκτέλεσης (patterns)** και συμβολίζονται Pat_i . Το Pat_1 καλείται αρχικό σχήμα κυματομορφής (initial pattern), ενώ με Pat_0 συμβολίζουμε το σύνορο των προϋπολογισμένων σημείων. Στο σχήμα 2.1 παρουσιάζονται τα σχήματα κυματομορφής για τις τέσσερις πρώτες χρονικές στιγμές.

- **Περίγραμμα του σχήματος κυματομορφής (pattern outline)** καλείται το τμήμα του συνόρου του Pat_k το οποίο το χωρίζει από το Pat_{k+1} . Ορίζεται και ως το σύνορο του Pat_k κατά τη διεύθυνση εκτέλεσης. Τα περιγράμματα τεσσάρων σχημάτων κυματομορφής σημειώνονται με διακεκομένη γραμμή στο σχήμα 2.1. Συμβολίζονται ως pat_i , $i \geq 1$ και τα σημεία που τα αποτελούν είναι λεξικογραφικά ταξινομημένα. Καλούνται και **περιγράμματα εκτέλεσης**. Για λόγους πληρότητας θεωρούμε $pat_0 = \langle (0, 0) \rangle$.
- Έστω $\mathbf{d} = (d_x, d_y)$ ένα σημείο στις δύο διαστάσεις· ορίζουμε ως $J_{\mathbf{d}}$ το σύνολο:

$$J_{\mathbf{d}} = \{\mathbf{j} = (j_x, j_y) \in J, \text{ όπου } d_x \leq j_x \text{ και } d_y \leq j_y\}.$$

- $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m$ είναι τα διανύσματα εξάρτησης του αλγορίθμου. Ο αριθμός τους δηλώνεται ως m . Στη συνέχεια αυτού του κεφαλαίου, χωρίς βλάβη της γενικότητας, θεωρούμε ότι τα διανύσματα εξάρτησης είναι ταξινομημένα ως προς τη γωνία που σχηματίζουν με τον άξονα των x . Έτσι, ως \mathbf{d}_1 θεωρείται αυτό που βρίσκεται πιο κοντά στον άξονα των y , ενώ ως \mathbf{d}_m αυτό που βρίσκεται πιο κοντά στον άξονα των x . Γι' αυτό και τα διανύσματα αυτά αναφέρονται και ως \mathbf{d}_{xmin} και \mathbf{d}_{ymin} , αντίστοιχα. Ως DV συμβολίζουμε την ακολουθία τους:

$$DV = \langle \mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{m-1}, \mathbf{d}_m \rangle = \langle \mathbf{d}_{xmin}, \mathbf{d}_2, \dots, \mathbf{d}_{m-1}, \mathbf{d}_{ymin} \rangle.$$

- Ως ακολουθία των διανυσμάτων κυματομορφής (pattern vectors) $PV = \langle \mathbf{p}_1, \dots, \mathbf{p}_k \rangle$, ορίζεται η ακολουθία των διανυσμάτων εξάρτησης, τα οποία επαληθεύουν την ακόλουθη σχέση:

$$J_{\mathbf{d}_1} \cup \dots \cup J_{\mathbf{d}_m} = J_{\mathbf{p}_1} \cup \dots \cup J_{\mathbf{p}_k} \text{ και } k \text{ ελάχιστο.}$$

Ο αριθμός τους δηλώνεται ως k (όπου $k \leq m$). Με δυο λόγια, είναι τα διανύσματα εξάρτησης τα οποία επηρεάζουν την κυματομορφή της εκτέλεσης.

- **Σημεία κυματομορφής (pattern points)** ονομάζονται τα σημεία τα οποία καθορίζουν την πολυγωνική γραμμή του περιγράμματος του κάθε σχήματος κυματομορφής (pattern).

Τα υπόλοιπα σημεία, απλά συμπληρώνουν τη γραμμή αυτή. Θεωρούμε, π.χ. τα σημεία A , B , C και D του σχήματος 2.1. Τα A και C είναι σημεία κυματομορφής (pattern points), ενώ τα B και D δεν είναι. Σημειώνουμε ότι τα σημεία κυματομορφής του αρχικού pattern, είναι τα τερματικά σημεία των διανυσμάτων κυματομορφής.

- Ως διαφορικές εξαρτήσεις ορίζονται τα διανύσματα διαφορών των k διανυσμάτων κυματομορφής. Είναι το πολύ $m - 1$ και χωρίζονται σε δύο είδη, όπως απεικονίζονται στο σχήμα 2.2:

- απόλυτα διαφορικά διανύσματα εξάρτησης: $\mathbf{a}_i = \mathbf{p}_{i+1} - \mathbf{p}_1$, $1 \leq i \leq k - 1$, και
- σχετικά διαφορικά διανύσματα εξάρτησης: $\mathbf{b}_i = \mathbf{p}_{i+1} - \mathbf{p}_i$, $1 \leq i \leq k - 1$.

- Δεδομένης μιας ακολουθίας σημείων $A = \langle \mathbf{j}_1, \mathbf{j}_2, \dots, \mathbf{j}_a \rangle$ και ενός διανύσματος \mathbf{d} , ορίζεται:

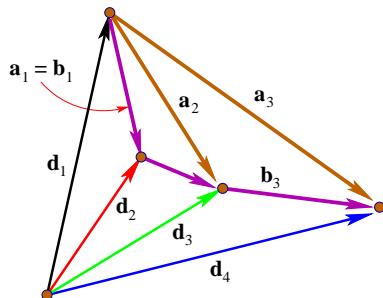
$$A + \mathbf{d} = \langle \mathbf{j}_1 + \mathbf{d}, \mathbf{j}_2 + \mathbf{d}, \dots, \mathbf{j}_a + \mathbf{d} \rangle.$$

- Δεδομένης μιας ακολουθίας σημείων $A = \langle \mathbf{j}_1, \mathbf{j}_2, \dots, \mathbf{j}_a \rangle$ και μιας ακολουθίας διανυσμάτων $B = \langle \mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_b \rangle$, ορίζεται:

$$A + B = \langle A + \mathbf{d}_1, A + \mathbf{d}_2, \dots, A + \mathbf{d}_b \rangle.$$

- Δεδομένων δύο ακολουθίων σημείων $A = \langle \mathbf{j}_1, \mathbf{j}_2, \dots, \mathbf{j}_a \rangle$ και $B = \langle \mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_b \rangle$, ορίζουμε την ένωση τους ως εξής:

$$A \cup B = \langle \mathbf{j}_1, \mathbf{j}_2, \dots, \mathbf{j}_a, \mathbf{i}_1, \mathbf{i}_2, \dots, \mathbf{i}_b \rangle.$$



Σχήμα 2.2: Απόλυτες και σχετικές διαφορικές εξαρτήσεις, για αλγόριθμο με 4 διανύσματα εξάρτησης.

Στο σημείο αυτό, ορίζουμε τα προβλήματα, πτυχές των οποίων μελετούμε και προσπαθούμε να προσεγγίσουμε στην παρούσα διατριβή. Δεδομένου ενός φωλιασμένου βρόχου, με χώρο δεικτών J και ακολουθία διανυσμάτων εξάρτησης DV , ζητείται:

- **Πρόβλημα Κυματομορφής:** να υπολογιστεί το περίγραμμα της κυματομορφής εκτέλεσης i pat_{*i*}, το οποίο αντιστοιχεί στη χρονική στιγμή $t = i \in \mathbb{N}$.
- **Πρόβλημα Δρομολόγησης:** να υπολογιστεί η χρονική στιγμή t_j , κατά την οποία είναι δυνατόν να εκτελεστεί ένα δεδομένο σημείου j του χώρου δεικτών.

Το πρόβλημα της κυματομορφής μελετάται στο Κεφάλαιο 4, ενώ με διαφορετικά ζητήματα του προβλήματος της δρομολόγησης, ασχολούνται τα υπόλοιπα κεφάλαια της διατριβής.

2.3 Βιβλιογραφική Επισκόπηση

Κατά την διαδικασία παραλληλοποίησης ενός φωλιασμένου βρόχου, το μεγαλύτερο πρόβλημα είναι αυτό της εύρεσης μιας έγκυρης και κατά το δυνατόν βέλτιστης δρομολόγησης των επαναλήψεων του βρόχου. Η εγκυρότητα αφορά τη διατήρηση των εξαρτήσεων του βρόχου. Το βέλτιστο της δρομολόγησης αφορά την απόκλιση από τη βέλτιστη δυνατή εκτέλεση. Λέμε ότι επιτυγχάνεται η βέλτιστη δυνατή εκτέλεση, όταν κάθε υπολογισμός πραγματοποιείται όταν και μόνον όταν όλα τα δεδομένα που απαιτεί είναι διαθέσιμα. Χωρίς, δηλαδή, επιπλέον καθυστέρηση.

2.3.1 Η Μέθοδος των Υπερεπιπέδων

Η πρώτη ιδέα δρομολόγησης φωλιασμένων βρόχων, χρησιμοποιώντας ένα γραμμικό μετασχηματισμό, παρουσιάστηκε από τον Lamport το 1974 [Lam74]. Την μέθοδο αυτή την ονόμασε μέθοδο των υπερεπιπέδων (*hyperplane method*) και βασίζεται στην εύρεση μιας γραμμικής απεικόνισης των δεικτοδοτημένων στιγμιοτύπων εκτέλεσης του βρόχου στο σύνολο των φυσικών αριθμών.

Οι δεικτοδοτημένοι υπολογισμοί του βρόχου οργανώνονται, με τον τρόπο αυτό, σε διακριτές ομάδες –τα υπερεπίπεδα– ούτως ώστε, όλες οι επαναλήψεις που ανήκουν στο ίδιο υπερεπίπεδο να μπορούν να εκτελεστούν ταυτόχρονα. Παράλληλα, η απεικόνιση διατηρεί τις εξαρτήσεις δεδομένων που εμφανίζονται στα διάφορα στιγμιότυπα εκτέλεσης του αρχικού βρόχου. Αν δηλαδή, δυο επαναλήψεις συνδέονται με σχέση εξάρτησης, τότε ανήκουν σε διαφορετικά υπερεπίπεδα.

Το αδιαμφισβήτητο πλεονέκτημα της γραμμικής αυτής μεθόδου, είναι ότι διατηρεί την κανονικότητα της ακολουθιακής εκτέλεσης και ροής των δεδομένων και κατά την παράλληλη εκτέλεση.

Προσφέρεται δηλαδή, για απεικόνιση σε παράλληλες αρχιτεκτονικές που εμφανίζουν κανονικότητα στη δομή τους.

Σημειώνουμε ότι το μαθηματικό μοντέλο που παρουσιάστηκε από τον Lampert, πραγματοποιούσε απεικόνιση του χώρου J^n των δεικτοδοτημένων επαναλήψεων, σε χώρο χρονικών στιγμιοτύπων \mathbb{N}^k . Βιβλιογραφικά, όμως, επικράτησε το $k = 1$, όπως θα φανεί και στη συνέχεια.

2.3.2 Εξελίξεις της Μεθόδου των Υπερεπιπέδων

Μετά τον Lampert, η μέθοδος του γραμμικού μετασχηματισμού, απετέλεσε αντικείμενο μελέτης πολλών ερευνητών του χώρου, οι οποίοι προσπάθησαν αφενός να βελτιώσουν το θεωρητικό πλαίσιο ορισμού του υπερεπιπέδου και αφετέρου να βρούνε αποδοτικούς αλγορίθμους εύρεσης των βέλτιστων υπερεπιπέδων.

Οι Moldovan και Fortes [MF86] επισήμαναν ότι σε ορισμένες περιπτώσεις, είναι δυνατόν να εκτελεστεί την ίδια χρονική στιγμή, ένα πλήθος διαδοχικών χρονικών υπερεπιπέδων. Αυτό εξαρτάται τόσο από την επιλογή του διανύσματος δρομολόγησης P , όσο και από τον τύπο των εξαρτήσεων που παρουσιάζει ο προς δρομολόγηση αλγόριθμος. Το πλήθος αυτό ορίστηκε από τους Shang και Fortes [SF91] και Sheu και Tai [ST91], ως η μετατόπιση (*displacement*) του συγκεκριμένου μετασχηματισμού. Εισάγοντας την έννοια της μετατόπισης, ορίστηκε και ο μαθηματικός τύπος της βέλτιστης γραμμικής χρονικής δρομολόγησης [MF86].

Μια σημαντική βελτίωση της παραλλαγής των Sheu και Tai [ST91] παρουσιάστηκε στην εργασία [TKP00]. Η βέλτιστη χρονική δρομολόγηση διατηρείται, ακολουθώντας την τακτική των υπερεπιπέδων. Για την ομαδοποίηση των επαναλήψεων του βρόχου που εκτελούνται στον ίδιο επεξεργαστή, επιλέγεται ένα από τα διανύσματα εξάρτησης. Το διάνυσμα εξάρτησης αυτό, επιλέγεται ούτως ώστε να μεγιστοποιεί τις αναφορές στον ίδιο επεξεργαστή. Έτσι, επιτυγχάνεται ελαχιστοποίηση της επικοινωνίας μεταξύ των επεξεργαστών, ενώ διατηρείται η βέλτιστη χρονική δρομολόγηση. Η μέθοδος αυτή αναφέρεται στη βιβλιογραφία ως *Chain Grouping*.

Οι Darte, Khachiyan και Robert, απέδειξαν ότι η δρομολόγηση που επιτυγχάνεται μέσω τέτοιων γραμμικών μετασχηματισμών, διαφέρει από τη βέλτιστη κατά σταθερά, για χώρους ικανοποιητικά μεγάλους [DR91]. Μάλιστα, η σταθερά αυτή εξαρτάται από τη δομή των εξαρτήσεων και όχι από το μέγεθος του χώρου. Το αποτέλεσμα αυτό είναι αρκετά σημαντικό, καθώς δείχνει ότι οι γραμμικές δρομολογήσεις, εκτός από την κανονικότητα που διατηρούν, είναι και σχεδόν βέλτιστες.

Κατόπιν αυτής της διαπίστωσης, οι περισσότερες μέθοδοι που παρουσιάστηκαν στη βιβλιογρα-

φία, για τη δρομολόγηση φωλιασμένων βρόχων, βασίζονται στην εύρεση ενός τέτοιου γραμμικού μετασχηματισμού.

2.3.3 Προσδορισμός του Βέλτιστου Γραμμικού Μετασχηματισμού

Προέκυψε με τον τρόπο αυτό, η ανάγκη εύρεσης του βέλτιστου υπερεπιπέδου. Αυτού, δηλαδή, που δίνει τη δρομολόγηση με τη μικρότερη απόκλιση από τη βέλτιστη δυνατή εκτέλεση. Το υπερεπιπέδο δρομολόγησης καθορίζεται πλήρως από το κάθετο διάνυσμα σ' αυτό. Το διάνυσμα αυτό καλείται διάνυσμα δρομολόγησης και συμβολίζεται ως P . Συνήθως είναι $P \in \mathbb{N}^n$.

Στην εργασία [MF86] παρουσιάστηκε μια μέθοδος εύρεσης γραμμικών χρονικών δρομολογήσεων, η οποία στηρίζεται στην εξαντλητική αναζήτηση όλων των δυνατών γραμμικών μετασχηματισμών. Για τον περιορισμό της αναζήτησης τέθηκαν τεχνητοί περιορισμοί, όσον αφορά το μέγεθος του διανύσματος δρομολόγησης P . Επειδή, όμως, οι περιορισμοί αυτοί δεν έχουν φυσική σημασία, υπάρχει ο κίνδυνος της μη εύρεσης της (συνολικά) βέλτιστης λύσης.

Στην εργασία [SF91] παρουσιάστηκε ένας αλγόριθμος εύρεσης ενός συνόλου υποψηφίων βέλτιστων διανυσμάτων δρομολόγησης. Ο αλγόριθμος αυτός παρουσιάζει εκθετική πολυπλοκότητα, τόσο ως προς τη διάσταση n του προβλήματος (βάθος βρόχου), όσο και ως προς των αριθμών εξαρτήσεων m . Η μέθοδος τους βασίζεται στη διαμέριση του χώρου όλων των γραμμικών απεικονίσεων σε υποσύνολα και την επίλυση με μεθόδους γραμμικού προγραμματισμού σε κάθε ένα από αυτά. Η ένωση όλων των λύσεων δίνει και το σύνολο των υποψηφίων γραμμικών μετασχηματισμών που οδηγούν στο βέλτιστο χρόνο εκτέλεσης. Το πλεονέκτημα της μεθόδου είναι ότι έχει χαμηλότερη πολυπλοκότητα από την επίλυση του καθολικού προβλήματος γραμμικού προγραμματισμού, στον συνολικό χώρο. Επιπλέον, δίνει ένα σύνολο υποψηφίων γραμμικών δρομολογήσεων, το οποίο είναι ανεξάρτητο από τα όρια εκτέλεσης των δεικτοδοτημένων επαναλήψεων του βρόχου. Η καλύτερη δρομολόγηση μπορεί να προκύψει εύκολα,² κατά τον χρόνο εκτέλεσης του βρόχου, όταν τα όρια του γίνουν γνωστά. Προτείνεται η πραγματοποίηση της ανάλυσης κατά τη μεταγλώττιση και η επιλογή της βέλτιστης δρομολόγησης κατά την εκτέλεση.

Οι προηγούμενες μέθοδοι βέλτιστης χρονικής δρομολόγησης, αντιμετωπίζουν την κάθε δεικτοδοτημένη επανάληψη ως ενιαίο σώμα, το οποίο δεν διαμερίζεται, αλλά δρομολογείται ολόκληρο να εκτελεστεί σε κάποιο από τα υπολογιστικά στοιχεία της διαθέσιμης αρχιτεκτονικής. Οι *Darte* και *Vivien* παρουσίασαν μια μέθοδο εύρεσης ξεχωριστού διανύσματος χρονικής δρομολόγησης

²Αναζητώντας π.χ. το μικρότερο χρόνο εκτέλεσης του τερματικού σημείου του βρόχου που προκύπτει από όλες τις πιθανές δρομολογήσεις.

Π για κάθε επιμέρους έκφραση (statement) του σώματος του βρόχου [DR95]. Η μέθοδος αυτή, απευθύνεται κυρίως σε πιο γενικές, μη ομοιόμορφες αρχιτεκτονικές, στις οποίες η κοινή δρομολόγηση ολόκληρου του σώματος του βρόχου δεν οδηγεί πάντα στο βέλτιστο αποτέλεσμα.

Όλες οι παραπάνω μέθοδοι έχουν το μειονέκτημα της αναζήτησης της βέλτιστης χρονικής δρομολόγησης, αγνοώντας τη δομή του χώρου δεικτών του βρόχου και του ρόλου των διανυσμάτων εξάρτησης. Βασίζονται είτε σε εξαντλητικές αναζητήσεις, είτε σε γραμμικό προγραμματισμό με βελτιστοποιήσεις και περιορισμούς, αναζητώντας ουσιαστικά στο χώρο όλων των δυνατών γραμμικών απεικονίσεων. Η ύπαρξη των συγκεκριμένων διανυσμάτων εξάρτησης και οι δομικές ιδιότητες που αυτά δημιουργούν στο χώρο των δεικτοδοτημένων στιγμιοτύπων ενός βρόχου, δεν λαμβάνονται υπόψη, παρά μόνο για την εγκυρότητα του υποψήφιου μετασχηματισμού. Τα συγκεκριμένα διανύσματα εξάρτησης, όμως, είναι αυτά που καυθορίζουν τη διάδοση της εκτέλεσης.

Στην εργασία [AKPT95] παρουσιάζεται μια μέθοδος εύρεσης της βέλτιστης γραμμικής δρομολόγησης για διδιάστατους φωλιασμένους βρόχους, η οποία βασίζεται στις γεωμετρικές ιδιότητες του συγκεκριμένου χώρου δεικτοδοτημένων επαναλήψεων και των διανυσμάτων εξάρτησης που περιέχει. Η μέθοδος αυτή έχει πολυωνυμική πολυπλοκότητα ως προς τον αριθμό m των διανυσμάτων εξάρτησης, υπερέχοντας κατά πολύ της μεθόδου των Shang και Fortes [SF91], η οποία έχει εκθετική πολυπλοκότητα ως προς m , για διδιάστατους βρόχους.

Γενίκευση αυτής της μεθόδου παρουσιάζεται στην παρούσα διατριβή. Μελετώντας την εξέλιξη της εκτέλεσης ενός φωλιασμένου βρόχου στο n -διάστατο χώρο, παρατηρούμε ότι εξακολουθούν να υπάρχουν διανύσματα που οριοθετούν τα κατώτατα τμήματα της κυματομορφής που σχηματίζεται, με τρόπο ανάλογο της παραπάνω μεθόδου. Το πρόβλημα εύρεσης των τμημάτων αυτών, ανάγεται σε αυτό του υπολογισμού ενός κυρτού περιγράμματος (convex hull). Η διάσταση του περιγράμματος ταυτίζεται με το βάθος των φωλιασμένων βρόχων n , ενώ το πλήθος των σημείων που το αποτελούν, εξαρτάται γραμμικά από τον αριθμό των διανυσμάτων εξάρτησης m .

Η πολυπλοκότητα υπολογισμού τέτοιων περιγραμμάτων είναι σημαντικά μικρότερη των αντίστοιχων εξαντλητικών μεθόδων που περιγράψαμε παραπάνω. Για προβλήματα διάστασης έως $n = 3$ είναι $\mathcal{O}(m \log m)$, ανεξάρτητο δηλαδή, της διάστασης! Για μεγαλύτερης διάστασης προβλήματα είναι εκθετική, της μορφής $\mathcal{O}(m^{\lfloor n/2 \rfloor} \log m)$, όπου m ο αριθμός των διανυσμάτων εξάρτησης και n η διάσταση του χώρου [Cha95, BDH96].

2.3.4 Μη Γραμμικές Δρομολογήσεις

Μια γραμμική χρονική δρομολόγηση έχει αρκετά πλεονεκτήματα. Τα κυριότερα είναι η ομοιόμορφη οργάνωση σε χρονικά υπερεπίπεδα και άρα η διατήρηση της κανονικότητας, η εύκολη εφαρμογή της λόγω γραμμικότητας, ο σχεδόν βέλτιστος χρόνος εκτέλεσης ανεξάρτητα από το μέγεθος του βρόχου και η εύκολη απεικόνιση της σε μεγάλο αριθμό επεξεργαστών. Από την άλλη πλευρά, αυτή η μέθοδος δρομολόγησης δεν οδηγεί πάντα σε βέλτιστα αποτελέσματα. Παραδείγματα τέτοιων περιπτώσεων είναι αυτές όπου πρόκειται να γίνει απεικόνιση σε αρχιτεκτονικές με υπολογίσιμο κόστος επικοινωνίας μεταξύ των επεξεργαστών και αυτές που οι βρόχοι έχουν αυθαιρέτου τύπου εξαρτήσεις. Σε τέτοιες περιπτώσεις πρέπει να εφαρμοστούν πιο γενικές μέθοδοι.

Στην εργασία [KPT96] παρουσιάστηκε μια εναλλακτική μέθοδος εύρεσης της βέλτιστης χρονικής δρομολόγησης για γενικού τύπου φωλιασμένους βρόχους, με βάση γραφοθεωρητικές ιδιότητες. Ο χώρος δεικτών αναπαρίσταται από έναν κατευθυνόμενο ακυκλικό γράφο προήγησης διεργασιών. Ο αλγόριθμος για την εύρεση της βέλτιστης χρονικής δρομολόγησης απαιτεί πολυωνυμικό χρόνο, ως προς τον αριθμό των επαναλήψεων. Η μέθοδος αυτή εκμεταλλεύεται πλήρως τη μη κανονικότητα που εμφανίζεται στα άκρα οποιουδήποτε πολυδιάστατου χώρου δεικτών, ούτως ώστε να απεικονίζει στον ελάχιστο αριθμό επεξεργαστών. Το σημαντικό της μειονέκτημα είναι η πολυπλοκότητα της.

Σε αντίστοιχες περιπτώσεις μη ομοιόμορφων εξαρτήσεων, οι μέθοδοι ομοιομορφοποίησης τους είναι εξ' ίσου μη αποδοτικές, καθώς δημιουργούν τεχνητές καθυστερήσεις συγχρονισμών, ώστε να βρεθεί η ελάχιστη δυνατή μορφή κανονικότητας. Ενδεικτικά σημειώνουμε τις μεθόδους των Tzen και Ni [TN93] και Shang, Hodzic και Chen [SHC96].

2.3.5 Δρομολόγηση με Κόστος Επικοινωνίας

Οι παραπάνω μέθοδοι αναφέρονται σε αρχιτεκτονικές με μηδενικό ή αμελητέο κόστος επικοινωνίας (zero communication cost). Το μόνο κόστος που λαμβάνεται υπ' όψιν είναι αυτό του υπολογισμού του κάθε σημείου του βρόχου (computation cost). Το κόστος αυτό θεωρείται μοναδιαίο. Το εν λόγω μοντέλο αναφέρεται στη βιβλιογραφία ως μοντέλο μοναδιαίου κόστους υπολογισμού - μηδενικού κόστους επικοινωνίας (*Unit Execution Time model - UET*).

Υπάρχουν όμως αρχιτεκτονικές οι οποίες επιφέρουν κόστος επικοινωνίας και μάλιστα συγχρίσιμο του κόστους υπολογισμού. Γι' αυτές τις αρχιτεκτονικές θεωρούμε το μοντέλο του μο-

ναδιαίου κόστους υπολογισμού - μοναδιαίου κόστους επικοινωνίας (*Unit Execution Time - Unit Communication Time model - UET-UCT*). Τέτοιες αρχιτεκτονικές είναι τα πλέγματα (*grids*).

Γενικά, το πρόβλημα της δρομολόγησης εργασιών (task scheduling) είναι από τα δυσκολότερα στο χώρο της παραλληλης επεξεργασίας. Το γενικό πρόβλημα της δρομολόγησης έχει αποδειχτεί ότι είναι NP-πλήρες από το 1975 [Ull75]. Κατόπιν αυτού, προτάθηκαν διάφορες ευριστικές μέθοδοι και προσεγγιστικοί αλγόριθμοι. Οι Παπαδημητρίου και Γιαννακάνης απέδειξαν την μη προσεγγισμότητα (intractability) του γενικού προβλήματος δρομολόγησης ενός γράφου διεργασιών με οποιοδήποτε κόστος επικοινωνίας και υπολογισμού [PY90]. Πρότειναν δε μια έξυπνη ευριστική μέθοδο, η οποία εγγυάτται χρόνο εκτέλεσης διπλάσιο του βέλτιστου, στη χειρότερη περίπτωση. Ένας δεύτερος ευριστικός αλγόριθμος δρομολόγησης παρουσιάστηκε από τους Γερασούλη και Yang [GY93]. Η μέθοδος που πρότειναν ονομάζεται *dominant sequence clustering* και βασίζεται στο χρίσιμο μονοπάτι (critical path) των διεργασιών.

Από την άλλη πλευρά, περιορίζοντας το γενικό πρόβλημα δρομολόγησης, είναι δυνατόν να έχουμε προσεγγίσμες περιπτώσεις γι' αυτό. Για παράδειγμα, οι Jung, Κιρούσης και Σπυράκης, παρουσίασαν έναν πολυωνυμικό αλγόριθμο ο οποίος βρίσκει το βέλτιστο χρόνο εκτέλεσης όταν το κόστος επικοινωνίας είναι σταθερό και επιτρέπεται η πολλαπλή ή διπλή εκτέλεση εργασιών (task duplication) [JKS89].

Όταν οι χρόνοι υπολογισμού και επικοινωνίας περιορίζονται στη μονάδα, είναι γνωστό ότι η γενική δρομολόγηση UET-UCT γράφων με περιορισμένο αριθμό επεξεργαστών είναι NP-πλήρες πρόβλημα, όπως απέδειξαν οι Rayward-Smith [RS87] και Picouleau [Pic92]. Ακόμη και στην περίπτωση που δεν έχουμε περιορισμό στον αριθμό των επεξεργαστών και όταν δεν επιτρέπεται διπλή εκτέλεση, το γενικό πρόβλημα εξακολουθεί να παραμένει μη προσεγγίσιμο (intractable). Στην περίπτωση που επιτρέπεται η διπλή εκτέλεση, οι Colin και Chretienne παρουσιάσαν μια βέλτιστη πολυωνυμική δρομολόγηση για γράφους με UET και SCT (small communication times), το οποίο περιλαμβάνει την περίπτωση UCT [CC91].

Και στον χώρο των πλεγμάτων (*grids*), το πρόβλημα της δρομολόγησης είναι πολυωνυμικά προσεγγίσιμο σε ορισμένες περιπτώσεις. Συγκεκριμένα, στην περίπτωση που ο χώρος δεικτών έχει τη μορφή n -διάστατου πλέγματος, είναι δυνατή η δρομολόγηση UET και UET-UCT προβλημάτων σε πολυωνυμικό χρόνο, είτε διαθέτουμε περιορισμένο αριθμό επεξεργαστών είτε όχι [AKPT99]. Υπολογίζεται μάλιστα και ο ελάχιστος αριθμός επεξεργαστών που επιτυχάνει τη βέλτιστη δρομολόγηση.

Όπως μπορεί να παρατηρήσει και ο αναγνώστης από τα παραπάνω, δεν αναφέρεται στη βιβλιογραφία καμία μέθοδος η οποία να επιτυγχάνει τη δρομολόγηση UET/UET-UCT βρόχων. Όλες οι μέθοδοι δρομολόγησης, είτε αναφέρονται σε βρόχους με μηδενικό κόστος επικοινωνίας, είτε περιλαμβάνουν κόστος επικοινωνίας και αναφέρονται σε γράφους εργασιών.

Αυτό το κενό έρχεται να συμπληρώσει η παρούσα διατριβή, της οποίας το κυρίως τμήμα είναι η γενική δρομολόγηση UET/UET-UCT βρόχων. Για πρώτη φορά αναδεικνύεται η αντιστοιχία μεταξύ ενός UET-UCT βρόχου και ενός ιδεατού, ισοδυνάμου UET.

Κεφάλαιο 3

Απεικόνιση Βρόχων σε Αρχιτεκτονικές Κατανεμημένης Μνήμης

Ξεκινώντας τη γεωμετρική προσέγγιση των προβλημάτων παράλληλης επεξεργασίας, ασχοληθήκαμε με το πρόβλημα της απεικόνισης φωλιασμένων βρόχων σε αρχιτεκτονικές κατανεμημένης μνήμης, συγκεκριμένου μεγέθους. Η μέθοδος που προτείνεται βασίζεται στην απεικόνιση βρόχων σε συστολικές διατάξεις.

Από τον αρχικό χώρο δεικτών (*initial index space*), μέσω ενός γραμμικού μετασχηματισμού, προκύπτει ο ιδεατός χώρος των μονάδων επεξεργασίας (*virtual array of cells*). Ο γραμμικός μετασχηματισμός που εφαρμόζεται επιλέγεται από τη βιβλιογραφία. Σκοπός του μετασχηματισμού είναι να δημιουργηθεί ένας νέος χώρος δεικτών, με χαρακτηριστικά συγγενή της διανθέσιμης διάταξης επεξεργαστών, διατηρώντας την απαιτούμενη αλληλουχία των υπολογισμών. Ο χώρος που προκύπτει καλείται μετασχηματισμένος χώρος δεικτών ή ιδεατός χώρος επεξεργαστών, γιατί στην ιδανική περίπτωση, διανέτουμε μια μονάδα επεξεργασίας (επεξεργαστή) ανά υπολογισμό.

Δεδομένου ότι σπάνια συμβαίνει να διανέτουμε μια διάταξη επεξεργαστών εντελώς ανάλογη με τον μετασχηματισμένο χώρο δεικτών, απαιτείται κάποια διαμέριση του μετασχηματισμένου χώρου δεικτών σε ένα πεπερασμένο αριθμό τμημάτων (*clusters*). Ο αριθμός των τμημάτων είναι ίσος με τον αριθμό των πραγματικών διανθέσιμων επεξεργαστών. Έτσι, ο κάθε επεξεργαστής εκτελεί μια ομάδα υπολογισμών, ένα τμήμα δηλαδή, του ιδεατού χώρου επεξεργαστών.

Στο κεφάλαιο αυτό προτείνεται μια γεωμετρική μεθόδος διαμέρισης του μετασχηματισμένου χώρου δεικτών, η οποία ελαχιστοποιεί την επικοινωνία μεταξύ των τμημάτων που προκύπτουν. Ταυτόχρονα, εξασφαλίζεται ισοκατανομή των επαναλήψεων υπολογισμού στους διαθέσιμους επεξεργαστές.

Η βασική ιδέα της μεθόδου είναι η πραγματοποίηση της διαμέρισης κατά επιλεγμένες διεύθυνσεις, ώστε να ελαχιστοποιείται η επικοινωνία μεταξύ των τμημάτων και να ισοκατανέμεται ο ιδεατός χώρος επεξεργασίας (*load balancing*). Στη συνέχεια, το κάθε τμήμα ανατίθεται σε ξεχωριστό επεξεργαστή, ο οποίος περιορίζεται κυρίως σε υπολογισμούς, δεδομένου ότι οι απαιτήσεις επικοινωνίας έχουν ελαχιστοποιηθεί.

Η προτεινόμενη απεικόνιση μειώνει τη συνολική καθυστέρηση λόγω επικοινωνίας, ενώ χρησιμοποιεί συγκεκριμένο αριθμό επεξεργαστών από μια διάταξη διάστασης $n - 1$, όπου n η διάσταση του αρχικού χώρου δεικτών.

3.1 Μετασχηματισμός του Χώρου Δεικτών

Στο πλαίσιο της παρούσας εργασίας δεν ασχολούμαστε με τη φάση του μετασχηματισμού. Θεωρούμε ότι ένας πίνακας μετασχηματισμού T έχει ήδη επιλεγεί. Στη βιβλιογραφία έχει επιχρατήσει να επιλέγεται πίνακας T διαστάσεων $n \times n$. Επιπλέον, ορίζεται από δύο επιμέρους υποπίνακες, ως εξής:

$$T_{n \times n} = \begin{bmatrix} \Pi_{1 \times n} \\ S_{n-1 \times n} \end{bmatrix}$$

όπως αναλύονται στην εργασία [MF86].

Ο υποπίνακας Π είναι πίνακας γραμμής και καυθορίζει τη χρονική εκτέλεση των σημείων του μετασχηματισμένου χώρου δεικτών J' , όπου $J'^n = T \cdot J^n$. Ονομάζεται και διάνυσμα δρομολόγησης *scheduling vector* ή *hyperplane vector*, επειδή είναι κάθετο στο υπερεπίπεδο δρομολόγησης. Ο υποπίνακας S είναι ο πίνακας χωρικού μετασχηματισμού. Μέσω αυτού απεικονίζονται τα στιγμιότυπα του μετασχηματισμένου χώρου στη διάταξη επεξεργαστών $S \cdot J^n$, με διάσταση $n - 1$.¹ Ο μετασχηματισμός πίνακας εξαρτήσεων D' επιβάλλεται να περιέχει ομόσημες τιμές στην πρώτη γραμμή, δεδομένου ότι αναφερόμαστε σε έγκυρο μετασχηματισμό.

Για τις απαιτήσεις της μεθόδου μας, θεωρούμε ότι η επιλογή του T δεν είναι απλώς έγκυρη, αλλά είναι και η βέλτιστη. Αυτό σημαίνει ότι:

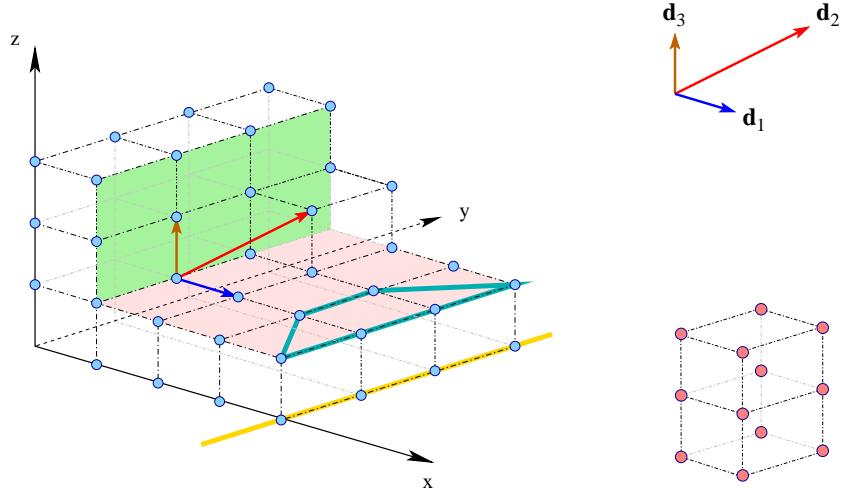
¹Ο περισσότερο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στις εργασίες [KPT96] και [ST91], όπου αναλύονται οι επιλογές των Π και S .

- ο πίνακας Π δίνει το υπερεπίπεδο δρομολόγησης με το μικρότερο χρονικό όνοιγμα (βέλτιστος χρόνος εκτέλεσης),
- ο S απεικονίζει στη βέλτιστη διάταξη επεξεργαστών, όσον αφορά το πλήθος των επεξεργαστών και τις συνδέσεις επικοινωνίας (με δεδομένο την εκτέλεση εντός των ορίων του βέλτιστου χρόνου).

3.2 Ορολογία και Ορισμοί

Στην παράγραφο αυτή εξηγούμε την ορολογία που θα χρησιμοποιήσουμε στην πορεία αυτού του κεφαλαίου. Για πληρέστερη εισαγωγή στον χώρο και καλύτερη κατανόηση, προτείνονται τα άρθρα [MF86], [ST91] και [DR94], όπως επίσης και το τεχνικό δελτίο [AKT⁺95].

Κατά τη διάρκεια αυτού του κεφαλαίου, το πρόθεμα “υπέρ-” χρησιμοποιείται κατά κόρον. Συμβολίζεται και ως “hyper-” και προσδίδει την ιδιότητα των n διαστάσεων. Η ορολογία που ακολουθεί, αντιγράφεται από αυτή του τρισδιάστατου χώρου, για τον οποίο παρουσιάζεται στο σχήμα 3.1:

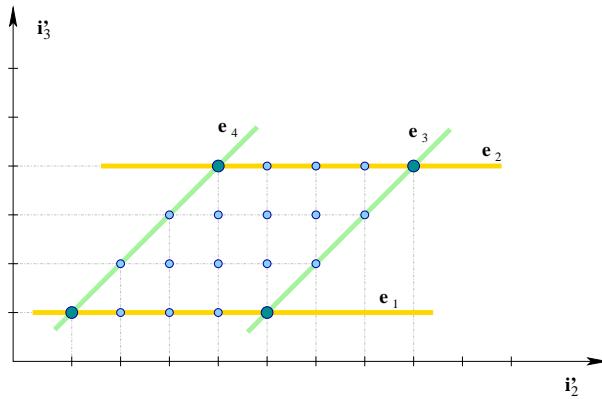


Σχήμα 3.1: Αλγόριθμος με (αρχικό) χώρο δεικτών 3 διαστάσεων. Σημειώνονται οι εξαρτήσεις, δύο υπερεπίπεδα, μία συνοριακή υπερευθεία, ένα υπερπολύγωνο και ένα πλέγμα επεξεργαστών, επίσης τριών διαστάσεων.

- **υπέρ-χώρος:** ο n -διάστατος χώρος

- **υπερ-επίπεδο:** ένα επίπεδο στο n -διάστατο χώρο· ένας γραμμικός υποχώρος διάστασης $n - 1$
- **υπερ-πολύγωνο:** ένα υποσύνολο ενός υπερεπιπέδου διάστασης $n - 1$
- **υπερ-ευθεία:** μια ευθεία στο n -διάστατο χώρο· ένα γραμμικός υποχώρος διάστασης $n - 2$
- **υπερ-πλέγμα:** ένα πλέγμα υπολογιστικών μονάδων στον χώρο διάστασης $n - 1$.

Λαμβάνοντας υπ' όψιν τα παραπάνω, ο αρχικός και ο μετασχηματισμένος χώρος είναι υπερχώροι. Ο προβαλλόμενος μετασχηματισμένος χώρος $S \cdot J^n$ –χωρίς τη χρονική συντεταγμένη– είναι συνήθως υπερεπίπεδο και υποδηλώνει την ιδεατή διάταξη επεξεργαστών. Το περίγραμμα του σχηματίζει ένα υπερπολύγωνο. Οι υπερπλευρές του πολυγώνου αυτού ορίζουν έναν αριθμό υπερευθειών που ονομάζονται συνοριακές υπερευθείες και οριοθετούν το σύνορο του (σχήματα 3.1 και 3.2):



Σχήμα 3.2: Μετασχηματισμένος χώρος δεικτών 2 διαστάσεων. Σημειώνεται το υπερπολύγωνο που σχηματίζεται και οι συνοριακές υπερευθείες που το αποτελούν.

- **συνοριακή υπερευθεία:** μια υπερευθεία με τις ακόλουθες δύο ιδιότητες:

1. τουλάχιστον $n - 1$ σημεία του μετασχηματισμένου χώρου δεικτών ανήκουν σ' αυτήν (γεωμετρικός ορισμός γραμμής) και
2. όλα τα υπόλοιπα σημεία του χώρου ανήκουν στο ίδιο ημι-υπερεπίπεδο.

Για τη μοντελοποίηση κάθε διαμέρισης του μετασχηματισμένου χώρου δεικτών, ορίζουμε τα παρακάτω:

- **τομή:** μια οποιαδήποτε υπερευθεία η οποία είναι παράλληλη με ένα ζευγάρι συνοριακών υπερευθειών και χωρίζει τον μετασχηματισμένο χώρο δεικτών σε δύο ημι-υπερεπίπεδα
- **πολλαπλή τομή:** ένα μη-κενό σύνολο από τομές, οι οποίες είναι παράλληλες μεταξύ τους και βρίσκονται σε ίσες αποστάσεις: ο ισομερής τμηματισμός του χώρου εξασφαλίζει τον ισόποσο καταμερισμό του υπολογιστικού φόρτου
- **απεικόνιση:** ένα σύνολο από πολλαπλές τομές, στο οποίο δεν υπάρχουν δύο τομές παράλληλες στην ίδια υπερευθεία
- **κόστος επικοινωνίας τομής:** το σύνολο των διανυσμάτων εξάρτησης που τέμνονται από την εν λόγω τομή.

3.2.1 Μετασχηματισμοί Αλγορίθμων

Η κεντρική ιδέα πίσω από τη μέθοδο του Moldovan [MF86] είναι ο μετασχηματισμός ενός αλγορίθμου $A(J^n, D)$ σε έναν ισοδύναμο $A'(J'^k, D')$, με μικρότερο χρόνο εκτέλεσης. Η έννοια της ισοδυναμίας δύο αλγορίθμων, για τους σκοπούς της παρούσας εργασίας, δίνεται από τον ακόλουθο ορισμό.

Ορισμός 3.1 Δύο αλγόριθμοι $A(J^n, D)$ και $A'(J'^k, D')$, λέγεται ότι είναι ισοδύναμοι, εάν και μόνον εάν ικανοποιούνται όλα τα παρακάτω:

1. Για κάθε εντολή ανάθεσης στον A , υπάρχει μια αντίστοιχη εντολή στον A' .
2. Οι αλγόριθμοι A και A' παρουσιάζουν ισοδυναμία εισόδου/εξόδου (για την ίδια είσοδο, δίνουν την ίδια έξοδο).
3. Ο χώρος J'^k αποτελεί μετασχηματισμό του αρχικού χώρου J^n : $J'^k = T(J^n)$, όπου T ένας αμφιμονοσήμαντος μετασχηματισμός.
4. Ο πίνακας εξαρτήσεων D' του A' , δίνεται από τον ίδιο μετασχηματισμό: $D' = T(D)$.

Μια ειδική περίπτωση μετασχηματισμών, οι οποίοι χρησιμοποιούνται στους αλγορίθμους που μας ενδιαφέρουν, είναι οι γραμμικοί μετασχηματισμοί. Αυτού του είδους οι μετασχηματισμοί ορίζονται πλήρως μέσω ενός πίνακα μετασχηματισμού T , διάστασης $n \times k$. Παρουσιάζουν τις ακόλουθες ιδιότητες:

- Κάθε σημείο $i \in J^n$ αντιστοιχεί με το μετασχηματισμένο σημείο $i' = T \cdot i$.

- Για τον πίνακα εξαρτήσεων ισχύει, αντίστοιχα $D' = T \cdot D$.

Κατόπιν αυτού, ορίζουμε τη γραμμική δρομολόγηση (*linear schedule*), ως ένα γραμμικό μετασχηματισμό με ορισμένους επιπλέον περιορισμούς, όπως παρουσιάζεται στη συνέχεια.²

Ορισμός 3.2 Δεδομένου αλγορίθμου $A(J^n, D)$, μια γραμμική δρομολόγηση Π ορίζεται με την προϋπόθεση ότι ισχύουν τα εξής:

1. Ο πίνακας Π είναι γραμμικός.

2. Ο πίνακας Π διατηρεί τις εξαρτήσεις του χώρου δεικτών:

$$\text{αν } i_1, i_2 \in J^n \text{ και } i_2 = i_1 + d_j, 1 \leq j \leq m, \text{ τότε } \Pi \cdot i_2 > \Pi \cdot i_1.$$

Ισοδύναμα: $\Pi \cdot d_j > 0$, για κάθε $1 \leq j \leq m$.

3. Η συνάρτηση $f(i)$, η οποία δίνει τη χρονική στηγμή κατά την οποία εκτελείται το σημείο i , δίνεται ως εξής:

$$f(i) = \left\lfloor \frac{\Pi \cdot i - t_0}{disp(\Pi)} \right\rfloor$$

όπου t_0 είναι είναι η αρχή του χρόνου και $disp(\Pi) = \min\{\Pi \cdot d_j, 1 \leq j \leq m\}$.

Από τον ορισμό αυτό προκύπτει το θεώρημα που ακολουθεί και αναφέρεται στο συνολικό χρόνο εκτέλεσης του παραλληλοποιημένου αλγορίθμου. Η απόδειξη του υπάρχει στο άρθρο [MF86].

Θεώρημα 3.1 Ο συνολικός χρόνος εκτέλεσης του παραλληλοποιημένου αλγορίθμου $A(J^n, D)$, δεδομένου του πίνακα δρομολόγησης Π , είναι:

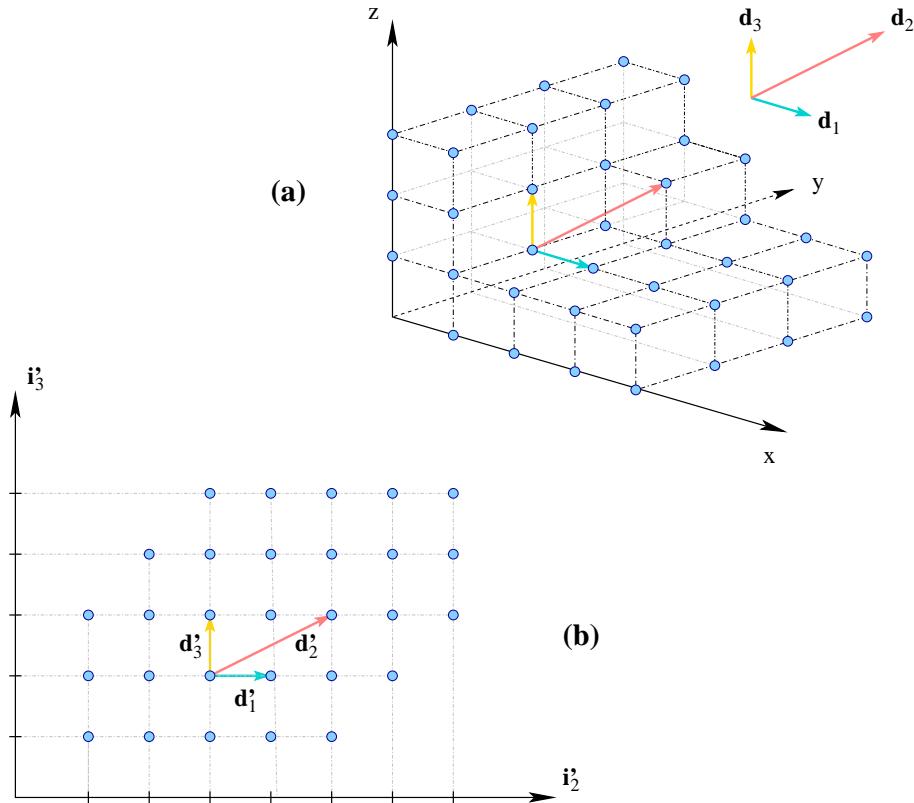
$$t_{fin} = \left\lceil \frac{\max\{\Pi \cdot i \mid i \in J^n\} - \min\{\Pi \cdot i \mid i \in J^n\}}{disp(\Pi)} \right\rceil + 1$$

Ορισμός 3.3 Έστω \mathcal{L} η κλάση των έγκυρων γραμμικών δρομολογήσεων, για έναν αλγόριθμο $A(J^n, D)$ και $\Pi \in \mathcal{L}$. Έστω $t_{fin}(\Pi)$ ο συνολικός χρόνος εκτέλεσης του αλγορίθμου, για τη χρονοδρομολόγηση Π . Μια γραμμική δρομολόγηση $\Pi^* \in \mathcal{L}$ είναι βέλτιστη (*optimal*) αν και μόνον εάν:

$$t_{fin}(\Pi^*) = \min\{t_{fin}(\Pi) \mid \Pi \in \mathcal{L}\}.$$

Το αποτέλεσμα του μετασχηματισμού του χώρου δεικτών J^n είναι ένας νέος χώρος J' , γενικά μικρότερης διάστασης. Ένα τέτοιο παράδειγμα μετασχηματισμού φαίνεται στο σχήμα 3.3.

²Για περισσότερες πληροφορίες περί των γραμμικών μετασχηματισμών, ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στα άρθρα [MF86], [ST91] και [AKT⁺95].



Σχήμα 3.3: Μετασχηματισμός του χώρου δεικτών J^3 ενός αλγορίθμου στον αντίστοιχο J'^2 .

Στο πρώτο τμήμα του σχήματος παρουσιάζεται ο χώρος δεικτών του αλγορίθμου J^3 . Στο δεύτερο τμήμα, παρουσιάζεται ο μετασχηματισμένος χώρος δεικτών. Για λόγους απλότητας δεν σχηματίζονται οι ολόκληροι οι δύο χώροι.

Το κέρδος που αποκομίζουμε από ένα τέτοιο μετασχηματισμό, είναι η δυνατότητα αποδοτικότερης παραλληλοποίησης. Αυτό επιτυγχάνεται επιλέγοντας τον μετασχηματισμό κατά τέτοιον τρόπο ώστε, τα χαρακτηριστικά του μετασχηματισμένου χώρου δεικτών (διαστάσεις, πλήθος σημείων, απαιτήσεις επικοινωνίας, κ.α.) να ταιριάζουν –κατά το δυνατόν– με αυτά της διαθέσιμης παράλληλης αρχιτεκτονικής.

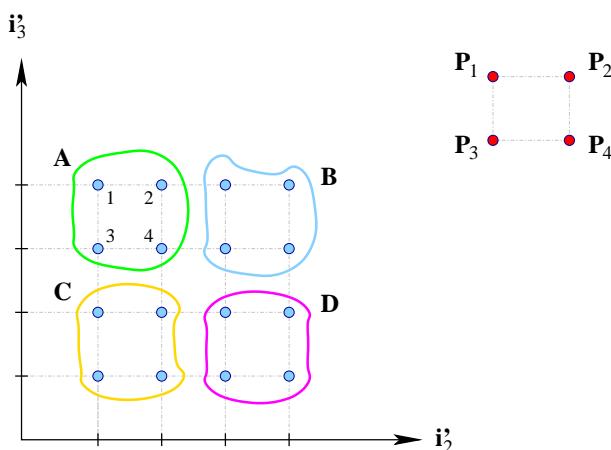
Το πρόβλημα που αναδεικνύεται στη συνέχεια, είναι αυτό της απεικόνισης των μετασχηματισμένων στιγμιοτύπων του βρόχου, σε υπολογιστικές μονάδες/επεξεργαστές, πεπερασμένου πλήθους. Εν γένει, είτε δεν διαθέτουμε πλήθος υπολογιστικών μονάδων ίσο με το πλήθος των

στιγμιοτύπων του βρόχου, είτε ο φόρτος του κάθε υπολογισμού είναι αρκετά μικρός ώστε να μην απαιτείται μια υπολογιστική μονάδα ανά στιγμιότυπο εκτέλεσης.

Απαιτείται λοιπόν, μια μεθοδολογία διαμέρισης του μετασχηματισμένου χώρου δεικτών, τόσο σε συγκεκριμένο πλήθος τμημάτων, όσο και σε τμήματα καθιορισμένων διαστάσεων. Κάτι τέτοιο έχει ως αποτέλεσμα την αποτελεσματικότερη και αποδοτικότερη παραλληλοποίηση του αλγορίθμου μας.

3.3 Τεχνικές Διαμέρισης και Απεικόνισης

Η μέθοδος διαμέρισης που περιγράφεται στη συνέχεια, βασίζεται στην απεικόνιση φωλιασμένων βρόχων σε συστολικές διατάξεις. Θεωρούμε ότι διαθέτουμε κάποια MIMD αρχιτεκτονική, με επεξεργαστές διασυνδεδεμένους με τη μορφή πλέγματος.



Σχήμα 3.4: Ο μετασχηματισμένος χώρος δεικτών και η διαθέσιμη διάταξη επεξεργαστών, για δεδομένο αλγόριθμο. Αναζητείται η μέθοδος απεικόνισης (mapping).

Μετά την φάση του μετασχηματισμού, στις περιπτώσεις που μας δίνεται ένας πεπερασμένος αριθμός επεξεργαστών, η μεθοδολογία διαμέρισης (*partitioning*) και απεικόνισης (*mapping*) καλείται να απεικονίσει τον μετασχηματισμένο χώρο δεικτών στη διαθέσιμη διάταξη υπολογιστικών μονάδων (βλ. σχήμα 3.4). Σε διαφορετική περίπτωση, όπου διαθέτουμε αριθμό επεξεργαστών ίσο με το πλήθος των στιγμιοτύπων υπολογισμού, η βέλτιστη εκτέλεση προκύπτει προφανώς, από την ένα-προς-ένα ανάθεση.

Η πιο διαδεδομένη μέθοδος απεικόνισης σε συστολικές διατάξεις, είναι η επονομαζόμενη LPGS (*Locally Parallel Globally Sequential* [DR94]). Δεδομένου ότι ο αριθμός των διαθέσιμων επεξεργαστών είναι μικρότερος από τον αριθμό των στιγμιοτύπων υπολογισμού, ο μετασχηματισμένος χώρος διαμερίζεται σε τμήματα (blocks), των οποίων το μέγεθος είναι ίσο με τις διαστάσεις της συστολικής μας διάταξης. Σε κάθε βήμα της εκτέλεσης, η συστολική διάταξη εκτελεί ένα block (globally sequential). Στο εσωτερικό του block, οι υπολογισμοί εκτελούνται παράλληλα (locally parallel), κάθε ένας από ξεχωριστό επεξεργαστή. Μετά την εκτέλεση του ενός τμήματος, η συστολική διάταξη προχωρεί στην εκτέλεση του επόμενου κ.ο.κ. Ένα τέτοιο παράδειγμα εκτέλεσης φαίνεται στο σχήμα 3.5[a].

Η μέθοδος αυτή δεν έχει ιδιαίτερες απαιτήσεις σε μνήμη, δεδομένου του ότι η ροή δεδομένων είναι συνεχής και περιορισμένη. Υπάρχει όμως, μεγάλη επιβάρυνση επικοινωνίας μεταξύ των επεξεργαστών. Η μέθοδος αυτή είναι ιδανική για συστολικές διατάξεις, όπου το κόστος επικοινωνίας είναι μικρό, ενώ δεν προτείνεται για αρχιτεκτονικές κατανεμημένης μνήμης.

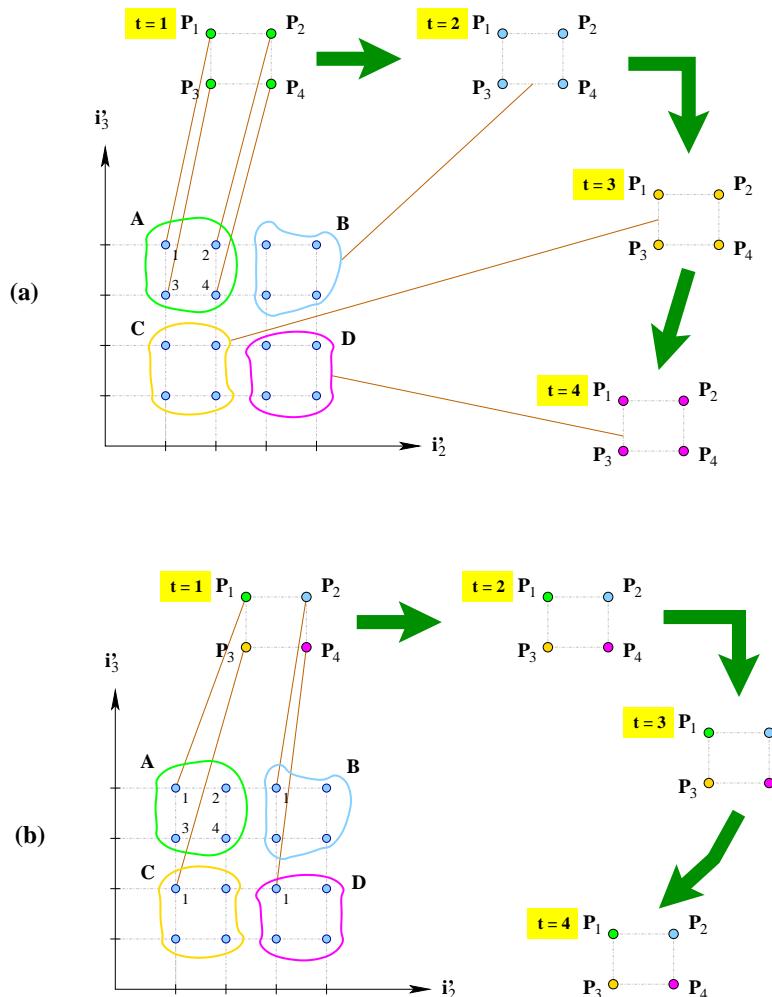
Στην προτεινόμενη μέθοδο, ακολουθείται η αντίθετη προσέγγιση. Επιθυμούμε τη μείωση των απαιτήσεων για επικοινωνία, με σκοπό την όσο το δυνατόν αποδοτικότερη παραλληλοποίηση. Και στη μέθοδο που προτείνουμε, ο μετασχηματισμένος χώρος διαμερίζεται σε τμήματα. Στην περίπτωση μας όμως, ο αριθμός των τμημάτων ισούται με τον αριθμό των διαθέσιμων επεξεργαστών (βλ. σχήμα 3.5[b]). Τα σημεία που περιέχει κάθε τμήμα εκτελούνται σειριακά (locally sequential), ενώ κάθε στιγμή εκτελείται ένα σημείο από το κάθε block (globally parallel).

Η διαμέριση του μετασχηματισμένου χώρου δεικτών γίνεται έτσι ώστε να ελαχιστοποιείται η επικοινωνία μεταξύ των τμημάτων. Για το λόγο αυτό επιλέγονται συγκεκριμένες διευθύνσεις τομής. Προφανώς, υπάρχουν περισσότερες απαιτήσεις για τοπική μνήμη, οι οποίες όμως, στις κατανεμημένες αρχιτεκτονικές επαρκούν. Σε τέτοιου είδους αρχιτεκτονικές, οι προσπελάσεις της τοπικής μνήμης παρουσιάζουν αμελητέο κόστος.

3.4 Απεικόνιση και Κόστος Επικοινωνίας

Η διαμέριση του μετασχηματισμένου χώρου δεικτών μπορεί να γίνει κατά οποιαδήποτε διεύθυνση. Λόγω των εξαρτήσεων του αλγορίθμου όμως, το κόστος της επικοινωνίας μεταξύ των τμημάτων, το οποίο απαιτείται σε κάθε περίπτωση, δεν είναι το ίδιο.

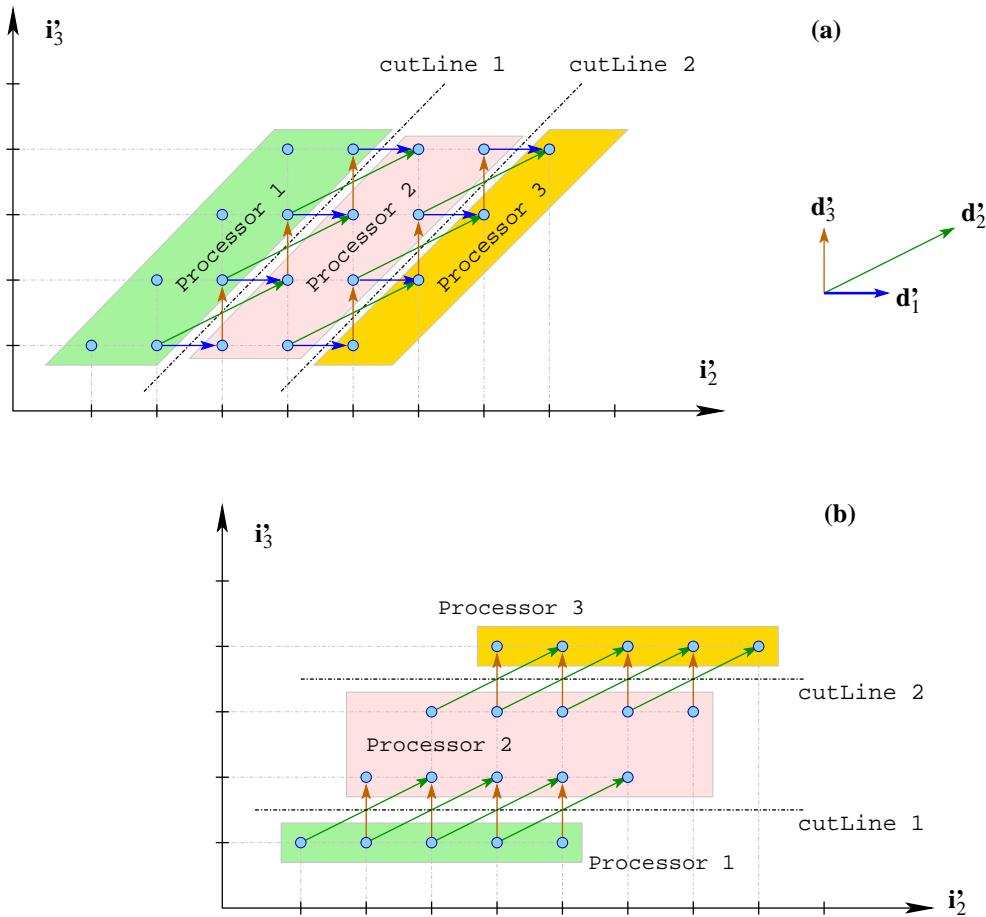
Στο σχήμα 3.6 παρουσιάζονται δύο τρόποι διαμέρισης και απεικόνισης του ίδιου μετασχηματισμένου χώρου δεικτών σε τρεις επεξεργαστές. Στην πρώτη εκδοχή, ο χώρος κόβεται παράλληλα



Σχήμα 3.5: Οι δύο μεθοδολογίες απεικόνισης: (a) η μέθοδος *Locally Parallel Globally Sequential (LPGS)*· ο αριθμός των στιγμοτύπων επανάληψης, που περιέχει κάθε τμήμα της διαμέρισης, ισούται με τον αριθμό των διαθέσιμων επεξεργαστών (b) η μέθοδος *Globally Parallel Locally Sequential (GPLS)*· εδώ ο αριθμός των τμημάτων της διαμέρισης ισούται με τον αριθμό των διαθέσιμων επεξεργαστών.

με το πλάγιο όριο του. Το συνολικό κόστος επικοινωνίας (μεταξύ των τμημάτων) είναι 20, ενώ οι επεξεργαστές εκτελούν 8, 8 και 4 στιγμότυπα, αντίστοιχα.

Στη δεύτερη περίπτωση, ο χώρος κόβεται παράλληλα με το οριζόντιο όριο του. Το συνολικό κόστος επικοινωνίας είναι 16, ενώ οι επεξεργαστές εκτελούν 5, 10 και 5 στιγμότυπα του



Σχήμα 3.6: Απεικόνιση με βάση δύο διαμερίσεις: (a) κατά τη διεύθυνση του πλαγίου ορίου του χώρου και (b) κατά τη διεύθυνση του οριζοντίου ορίου του χώρου. Παρατηρούμε τις διαφορές τόσο στις απαιτήσεις επικοινωνίας, όσο και στην χρησιμοποίηση του κάθε επεξεργαστή. Και στις δύο περιπτώσεις θεωρούμε διαθέσιμους 3 επεξεργαστές.

βρόχου, αντίστοιχα. Παρατηρούμε τη διαφορά τόσο στην απαίτηση επικοινωνίας, όσο και στον καταμερισμό του φορτίου, για τις δύο διαμερίσεις.

Με τη μέθοδο διαμέρισης και απεικόνισης που παρουσιάζεται στη συνέχεια, επιδιώκουμε την κατάλληλη επιλογή των διευθύνσεων διαμέρισης, ούτως ώστε η απαίτηση επικοινωνίας να ελαχιστοποιείται και να επιτυγχάνεται όσο το δυνατόν βέλτιστος καταμερισμός φόρτου στους επεξεργαστές.

Κατά την πορεία της παρουσίασης της μεθόδου, πρέπει να έχουμε υπ' όψιν μας τα εξής:

- ο πίνακας T έχει προεπιλεγεί ούτως ώστε να αποτελεί τον βέλτιστο πίνακα μετασχηματισμού για τις ανάγκες του προβλήματός μας (εκτέλεση στον ελάχιστο χρόνο, με συγκεκριμένο αριθμό επεξεργαστών),
- ο μετασχηματισμός χώρος δεικτών είναι ένα υπερπολύγωνο, με ακμές οι οποίες είναι παράλληλες κατά ζεύγη,
- ο πραγματικός χώρος των επεξεργαστών είναι ένα n -διάστατο πλέγμα· στο χώρο αυτό θα γίνει η απεικόνιση του μετασχηματισμένου χώρου και
- η βέλτιστη τομή θα δοθεί με όρους συνεχούς χώρου (όχι διακριτού).

3.5 Κόστος Επικοινωνίας Απεικονίσεων & Τομών

Το κόστος επικοινωνίας μιας απεικόνισης, υπολογίζεται ως το άθροισμα από τα επιμέρους κόστη των ξεχωριστών τομών που την συνθέτουν. Το επιμέρους κόστος μιας τομής, ορίζεται ως ο αριθμός των μετασχηματισμένων διανυσμάτων εξάρτησης τα οποία διαπερνούν την υπερευθεία που την αποτελεί. Ο ορισμός αυτός φαίνεται εποπτικά στο σχήμα 3.6, για δύο διαφορετικές τομές.

Η παρακάτω έκφραση υπολογίζει το κόστος επικοινωνίας –σύμφωνα με τον παραπάνω ορισμό– το οποίο επιφέρει μια τομή:

$$c = \sum_{i=1}^m \frac{|\mathbf{p} \cdot \mathbf{d}'_i|}{\|\mathbf{p}\| \cdot \|\mathbf{d}'_i\|} \cdot \ell \cdot \rho \quad (3.1)$$

όπου m είναι ο αριθμός των διανυσμάτων εξάρτησης, \mathbf{p} είναι ένα διάνυσμα κάθετο στην ευθεία της τομής, \mathbf{d}'_i είναι ένα μετασχηματισμένο διάνυσμα εξάρτησης, $\|\mathbf{u}\|$ είναι η Ευκλίδεια νόρμα του διανύσματος \mathbf{u} , ℓ είναι το μήκος του τμήματος της υπερευθείας που αντιστοιχεί στην τομή και βρίσκεται εντός των ορίων του μετασχηματισμένου χώρου και ρ είναι η πυκνότητα των διανυσμάτων \mathbf{d}'_i κατά μήκος μιας υπερευθείας κάθετης σ' αυτά.

Το κλάσμα, στο εσωτερικό του αιθροίσματος της εξίσωσης 3.1, ισούται με το συνημίτονο της γωνίας μεταξύ των διανυσμάτων \mathbf{p} και \mathbf{d}'_i . Το γεγονός αυτό εκφράζει την παρατήρηση ότι διευθύνσεις τομής όσο το δυνατόν παράλληλες με τα μετασχηματισμένα διανύσματα εξάρτησης, τέμνονται από λιγότερα τέτοια διανύσματα και άρα δίνουν μικρότερο κόστος επικοινωνίας.

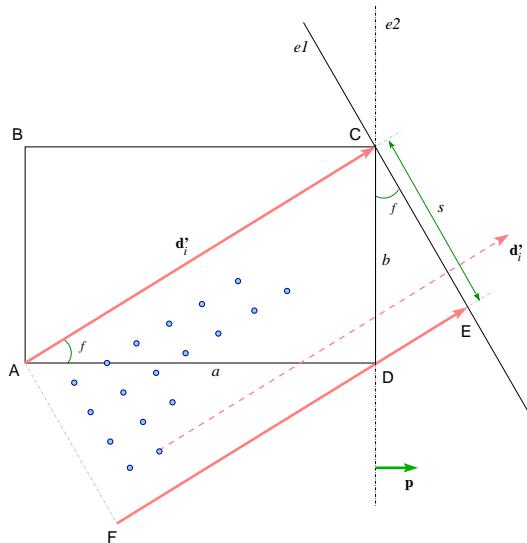
Ο παράγοντας $|\mathbf{p} \cdot \mathbf{d}'_i|$ δηλώνει το βάθος υπολογισμού για κάθε διάνυσμα εξάρτησης κατά τη διεύθυνση του \mathbf{p} . Δηλαδή, όταν η πιθανή διεύθυνση τομής είναι κάθετη στο διάνυσμα εξάρτησης \mathbf{d}'_i , επιφέρει το μεγαλύτερο δυνατό κόστος επικοινωνίας. Όσο πλησιάζει την παράλληλη διεύθυνση με το \mathbf{d}'_i ($\mathbf{p} \perp \mathbf{d}'_i$), τόσο η εν λόγω τομή δεν επηρεάζει το κόστος επικοινωνίας λόγω αυτού του διανύσματος.

Θεώρημα 3.2 Η παράμετρος ρ για το διάνυσμα \mathbf{d}'_i , είναι ίση με την Ευκλίδεια νόρμα του \mathbf{d}'_i :

$$\rho = \|\mathbf{d}'_i\| \quad (3.2)$$

Απόδειξη

Θα δούθει μια απλή γεωμετρική απόδειξη για τις τρεις διαστάσεις (για $n = 3$). Αντίστοιχα μπορεί να γίνει η απόδειξη και για μεγαλύτερες διαστάσεις. Θεωρούμε το μετασχηματισμένο διάνυσμα εξάρτησης $\mathbf{d}'_i = (a, b)$, με μήκος AC , όπως φαίνεται στο σχήμα 3.7. Θεωρούμε ότι ε_2 είναι η κάθετη ευθεία στο διάνυσμα \mathbf{d}'_i . Ένα από τα τμήματα της ευθείας που ανήκουν στο



Σχήμα 3.7: Υπολογισμός της παραμέτρου ρ για το μετασχηματισμένο διάνυσμα εξάρτησης $\mathbf{d}'_i = (a, b)$.

μετασχηματισμένο χώρο, είναι το τμήμα CE . Θεωρούμε επίσης, ότι η πυκνότητα είναι σταθερή σε όλο το μήκος της ευθείας. Στην περίπτωση αυτή, η πυκνότητα ρ των \mathbf{d}'_i που διαπερνούν την ευθεία, ισούται με τον αριθμό w των διανυσμάτων που διαπερνούν το CE , προς το μήκος s του CE , δηλαδή: $\rho = w/s = w/|CE|$. Για τον υπολογισμό του αριθμού w , των διανυσμάτων \mathbf{d}'_i

που διαπερνούν το τμήμα CE , αρκεί να παρατηρήσουμε ότι η αρχή τους πρέπει να βρίσκεται στο εσωτερικό του παραλληλογράμμου $ACEF$. Το εμβαδό όμως, του $ACEF$ είναι ίσο με αυτό του $ABCD$ και άρα είναι $a \cdot b$. Δεδομένου ότι κάθε σημείο εκκίνησης, οποιουδήποτε διανύσματος εξάρτησης έχει ακέραιες συντεταγμένες και δεν υπάρχουν κενά στον χώρο δεικτών, καταλήγουμε στο συμπέρασμα ότι ο αριθμός των διανυσμάτων \mathbf{d}'_i που εκκινούν από το $ACEF$ είναι ίσος με το εμβαδόν του $ACEF$, δηλαδή: $w = a \cdot b$.³ Επίσης, το μήκος του τμήματος CE υπολογίζεται, σύμφωνα με το σχήμα 3.7, από την σχέση:

$$s = b \cdot \cos(f) = b \cdot \frac{|AD|}{|AC|} = b \cdot \frac{a}{\|\mathbf{d}'_i\|} = \frac{a \cdot b}{\|\mathbf{d}'_i\|}$$

Ως συνέπεια αυτού, η πυκνότητα ρ γράφεται πλέον ως εξής:

$$\rho = \frac{w}{s} = \frac{a \cdot b}{\frac{a \cdot b}{\|\mathbf{d}'_i\|}} = \|\mathbf{d}'_i\|$$

→

Λόγω του θεωρήματος, η συνάρτηση κόστους 3.1 μπορεί να γραφεί με την ακόλουθη μορφή:

$$c = \ell \cdot \sum_{i=1}^m \frac{|\mathbf{p} \cdot \mathbf{d}'_i|}{\|\mathbf{p}\|} \quad (3.3)$$

Η παραπάνω έκφραση δίνει έναν ευριστικό τρόπο αξιολόγησης της κάθε απεικόνισης. Ιδιαίτερα όταν ο μετασχηματισμένος χώρος είναι ένα υπερ-παραλληλόγραμμο, μπορούμε να εφαρμόσουμε την παραπάνω συνάρτηση για όλες τις δυνατές απεικονίσεις και έτσι να καταλήξουμε σ' αυτή με το μικρότερο κόστος επικοινωνίας. Ποιές όμως είναι οι πιθανές διευθύνσεις τομής;

3.6 Καθορισμός των Πιθανών Διευθύνσεων Τομής

Αφού καθορίσαμε το πως μετράμε το κόστος κάθε τομής και κάθε απεικόνισης, το ζητούμενο τώρα είναι να βρούμε όλες τις πιθανές διευθύνσεις τομής. Γνωρίζοντας αυτές τις διευθύνσεις, μπορούμε να υπολογίσουμε το κόστος για καθεμιά.

³Παρά την ισότητα των παραπάνω εμβαδών, λόγω του ασυνεχούς/διακριτού του χώρου δεικτών, ο αριθμός των σημείων που περιέχονται στα δύο παραλληλόγραμμα ενδέχεται να είναι διαφορετικός. Η διαφορά αυτή καθίσταται αμελητέα αν θεωρήσουμε χώρο δεικτών σημαντικού μεγέθους, γεγονός που συμβαίνει στην πλειοψηφία των προβλημάτων.

Για την όσο το δυνατόν μικρότερη επιβάρυνση επικοινωνίας, ως πιθανές διευθύνσεις τομής επιλέγουμε ανάμεσα σ' αυτές των συνόρων του μετασχηματισμένου χώρου. Αυτό εξηγείται ως εξής. Λόγω της ύπαρξης των συνόρων, ένα τουλάχιστον από τα διανύσματα εξάρτησης θα είναι παράλληλο με κάθε μία από τις συνοριακές διευθύνσεις. Ως αποτέλεσμα αυτού, οι διευθύνσεις τομής οι οποίες είναι παράλληλες με τα σύνορα του χώρου, επιφέρουν κόστος επικοινωνίας μηδενικό, τουλάχιστον ως προς ένα διάνυσμα εξάρτησης (αυτό που είναι παράλληλο με το αντίστοιχο σύνορο). Γι' αυτό το λόγο επιλέγουμε ως πιθανές διευθύνσεις τομής αυτές που είναι παράλληλες με τις συνοριακές γραμμές.

Ο αλγόριθμος που ακολουθεί βρίσκει όλες τις πιθανές διευθύνσεις που μπορεί να έχει μια συνοριακή υπερευθεία, δηλαδή ένα σύνορο του μετασχηματισμένου χώρου.

Αλγόριθμος 3.1

προσδιορισμός των διευθύνσεων των συνοριακών γραμμών

Εισοδος: τα όρια των βρόχων $l_i, u_i, 0 \leq i \leq n - 1$ και ο πίνακας μετασχηματισμού T .

Εξοδος: τα ζεύγη των συνοριακών γραμμών του μετασχηματισμένου χώρου, έστω β .

1. Ορίζουμε τον πίνακα V , διάστασης $n \times 2^n$, ο οποίος περιέχει ως στήλες, όλες τις 2^n μεταθέσεις των συντεταγμένων των συνοριακών σημείων του αρχικού χώρου:

$$V = \begin{bmatrix} l_0 & l_0 & \dots & u_0 & u_0 \\ l_1 & l_1 & \dots & u_1 & u_1 \\ \vdots & \vdots & & \vdots & \vdots \\ l_{n-2} & l_{n-2} & \dots & u_{n-2} & u_{n-2} \\ l_{n-1} & u_{n-1} & \dots & l_{n-1} & u_{n-1} \end{bmatrix}$$

2. Με την βοήθεια του πίνακα μετασχηματισμού T , υπολογίζουμε τον μετασχηματισμό του πίνακα V :

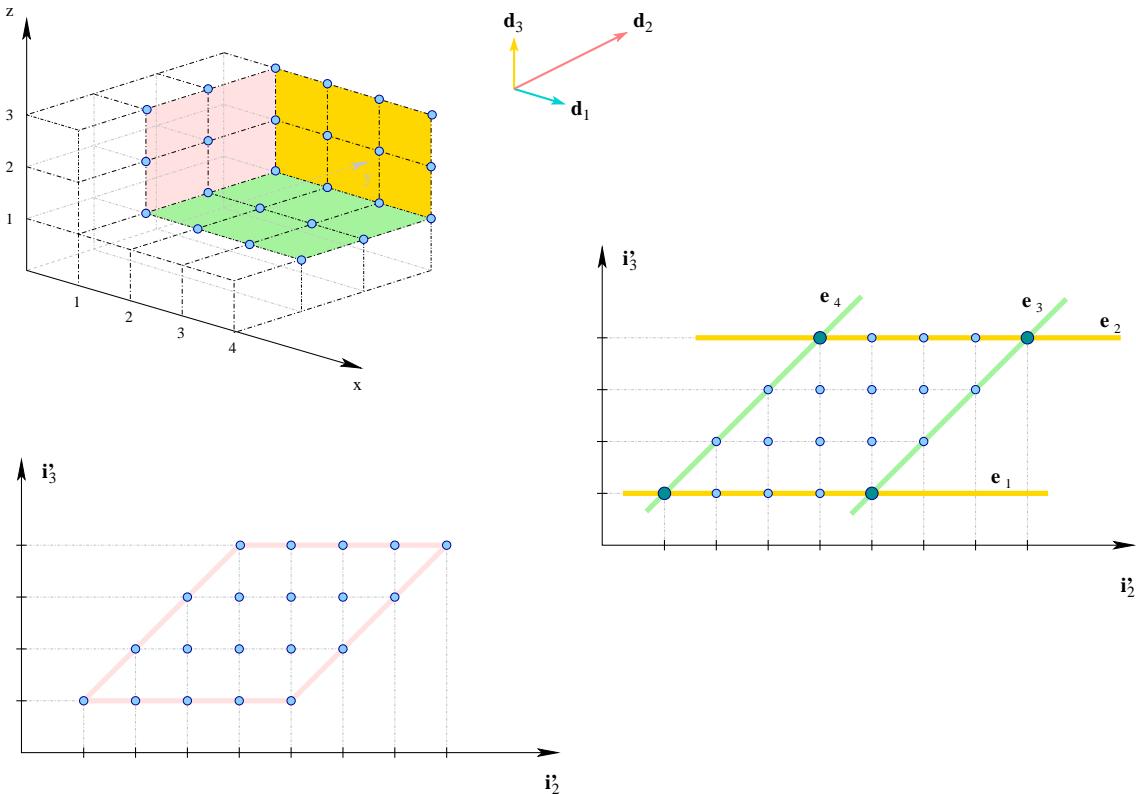
$$V' = T \cdot V.$$

3. Ορίζουμε τον πίνακα W , ο οποίος προκύπτει από τον V' , αν αφαιρέσουμε την πρώτη γραμμή του (η γραμμή αυτή αποτελεί τη συντεταγμένη χρόνου). Ο πίνακας W είναι διάστασης $(n - 1) \times 2^n$ και αναπαριστά τις συντεταγμένες των συνοριακών σημείων του αρχικού χώρου, στον μετασχηματισμένο χώρο δεικτών. Κάθε στήλη του W αντιστοιχεί σε ένα σήμειο.
4. Υπολογίζεται το κυρτό περίγραμμα (convex hull) των σημείων που περιέχονται στον πίνακα W . Το αποτέλεσμα είναι ένα υπερπολύγωνο. Οι υπερευθείες που ορίζονται από τις πλευρές του πολυγώνου αυτού, αποτελούν τις ζητούμενες συνοριακές υπερευθείες.
5. Δεδομένου ότι οι πλευρές του κυρτού περιγράμματος δίνονται σε ζεύγη παράλληλων υπερευθειών, μπορούν να αναπαρασταθούν από ζεύγη ανισοτήτων της μορφής:

$$\gamma \leq a_1 x'_1 + a_2 x'_2 + \dots + a_{n-1} x'_{n-1} + a_n \leq \delta$$

Έστω β ο αριθμός των ζευγών αυτών.

Ο αλγόριθμος περιγράφεται στο σχήμα 3.8, όπου εφαρμόζεται σε ένα πρόβλημα φωλιασμένων βρόχων διάστασης δύο.



Σχήμα 3.8: Εύρεση των συνοριακών γραμμών μετασχηματισμένου χώρου, μέσω του Αλγορίθμου 3.1: προσδιορίζονται τα μετασχηματισμένα σημεία των ορίων του αρχικού χώρου και υπολογίζεται το κυρτό τους περίγραμμα. Οι ακμές του αποτελούν τμήματα των ζητούμενων συνοριακών γραμμών.

Με την εφαρμογή του παραπάνω αλγορίθμου, έχουμε στη διάθεση μας τις ανισώσεις των ορίων του μετασχηματισμένου χώρου για το εκάστοτε πρόβλημα. Οι ανισώσεις αυτές δίνονται σε ζεύγη ως β θεωρούμε το πλήνος τους.

3.7 Αξιολόγηση και Επιλογή Απεικόνισης

Για να βρούμε όλες τις δυνατές απεικονίσεις, αρκεί να βρούμε όλους τους δυνατούς τρόπους κατά τους οποίους, ο μετασχηματισμένος χώρος δεικτών μπορεί να ανατεθεί στο χώρο των επεξεργαστών. Η διάταξη των επεξεργαστών αναπαρίσταται από ένα διάνυσμα π , $n - 1$ στοιχείων. Κάθε στοιχείο υποδηλώνει τον αριθμό των μονάδων επεξεργασίας που διατίθενται στην κάθε διάσταση.⁴ Ισούται δηλαδή, με τον αριθμό των τομών, ως προς τη διάσταση αυτή, συν ένα, όπως θα φανεί στη συνέχεια. Το ξ υποδηλώνει τις διαφορετικές, μη μηδενικές τιμές του π και το ψ_j τις τιμές αυτές, όπου $1 \leq j \leq \xi$.

Θεωρούμε δηλαδή, ότι η κάθε απεικόνιση αποτελείται από μια σειρά από πολλαπλές τομές (multiple cuts), οι οποίες συμβολίζονται $mCut_i$. Το i δηλώνει τη συνοριακή ευθεία, παράλληλα με την οποία γίνεται η κατάτμηση από την πολλαπλή τομή. Η κάθε πολλαπλή τομή με τη σειρά της, αποτελείται από j τομές (cuts), οι οποίες συμβολίζονται $cut_{i,j}$. Το j δηλώνει τον αύξοντα αριθμό της τομής. Συνοψίζοντας:

$$mCut_i = \{cut_{i,1}, cut_{i,2}, \dots, cut_{i,\xi}\}$$

Σύμφωνα με τα παραπάνω, η διαδικασία απεικόνισης χωρίζεται σε δύο φάσεις:

- τον προϋπολογισμό των –κοινών για όλες τις απεικονίσεις– συντελεστών αξιολόγησης $mcCost_{i,j}$, του κόστους επικοινωνίας κάθε επιμέρους τομής της πολλαπλής τομής $mCut_i$
- τον καθεαυτό υπολογισμό του κόστους κάθε μιας απεικόνισης: αυτό γίνεται αιθροίζοντας τα επιμέρους κόστη επικοινωνίας των πολλαπλών τομών, από τις οποίες δημιουργείται.

3.7.1 Προσδιορισμός Κόστους Επικοινωνίας Τομών

Ο αλγόριθμος που ακολουθεί υπολογίζει τους συντελεστές αξιολόγησης $mcCost_{i,j}$. Οι συντελεστές αυτοί εκφράζουν το κόστος επικοινωνίας το οποίο επιφέρει η τυχόν κατάτμηση του μετασχηματισμένου χώρου δεικτών, σε ψ_j τμήματα, κατά τη διεύθυνση του i -οστού ζεύγους συνοριακών γραμμών.

⁴Αν το πλέγμα των επεξεργαστών έχει διάσταση μεγαλύτερη από $n - 1$, ορισμένα από τα στοιχεία του π θα πρέπει να έχουν την τιμή 1.

Αλγόριθμος 3.2

προ-υπολογισμός κόστους πολλαπλών τομών

Εισοδος: τα β ζεύγη των συνοριακών γραμμών του μετασχηματισμένου χώρου και οι εξαρτήσεις $\mathbf{d}'_1, \dots, \mathbf{d}'_m$.

Εξοδος: οι συντελεστές αξιολόγησης κόστους επικοινωνίας $mcCost_{i,j}$.

- για κάθε ζεύγος i συνοριακών γραμμών (binding h-lines)

$$\gamma_i \leq a_{i,1}x'_1 + a_{i,2}x'_2 + \dots + a_{i,n-1}x'_{n-1} + a_{i,n} \leq \delta_i, \text{ όπου } 1 \leq i \leq \beta$$

1. έστω \mathbf{p}_i το κάθετο διάνυσμα στη διεύθυνση αυτή: $\mathbf{p}_i = (a_{i,1}, a_{i,2}, \dots, a_{i,n-1}, a_{i,n})$
2. υπολογίζεται ο παράγοντας κόστους $depCost_i$, του αλγορίθμου μας, κατά τη διεύθυνση την κάθετη στο \mathbf{p}_i :

$$depCost_i = \frac{1}{\|\mathbf{p}_i\|} \sum_{j=1}^m |\mathbf{p}_i \cdot \mathbf{d}'_j|$$

3. για όλους τους διαφορετικούς αριθμούς ψ_j , $1 \leq j \leq \xi$, του διανύσματος των επεξεργαστών:

▷ υπολογίζεται το κόστος της πολλαπλής τομής, η οποία τέμνει τον μετασχηματισμένο χώρο σε ψ_j τμήματα, με διεύθυνση την κάθετη στο \mathbf{p}_i :

$$mcCost_{i,j} = depCost_i \cdot \sum_{k=1}^{\psi_j - 1} cutArea \left(\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_\beta, \frac{\delta_i - \gamma_i}{\psi_j} \right)$$

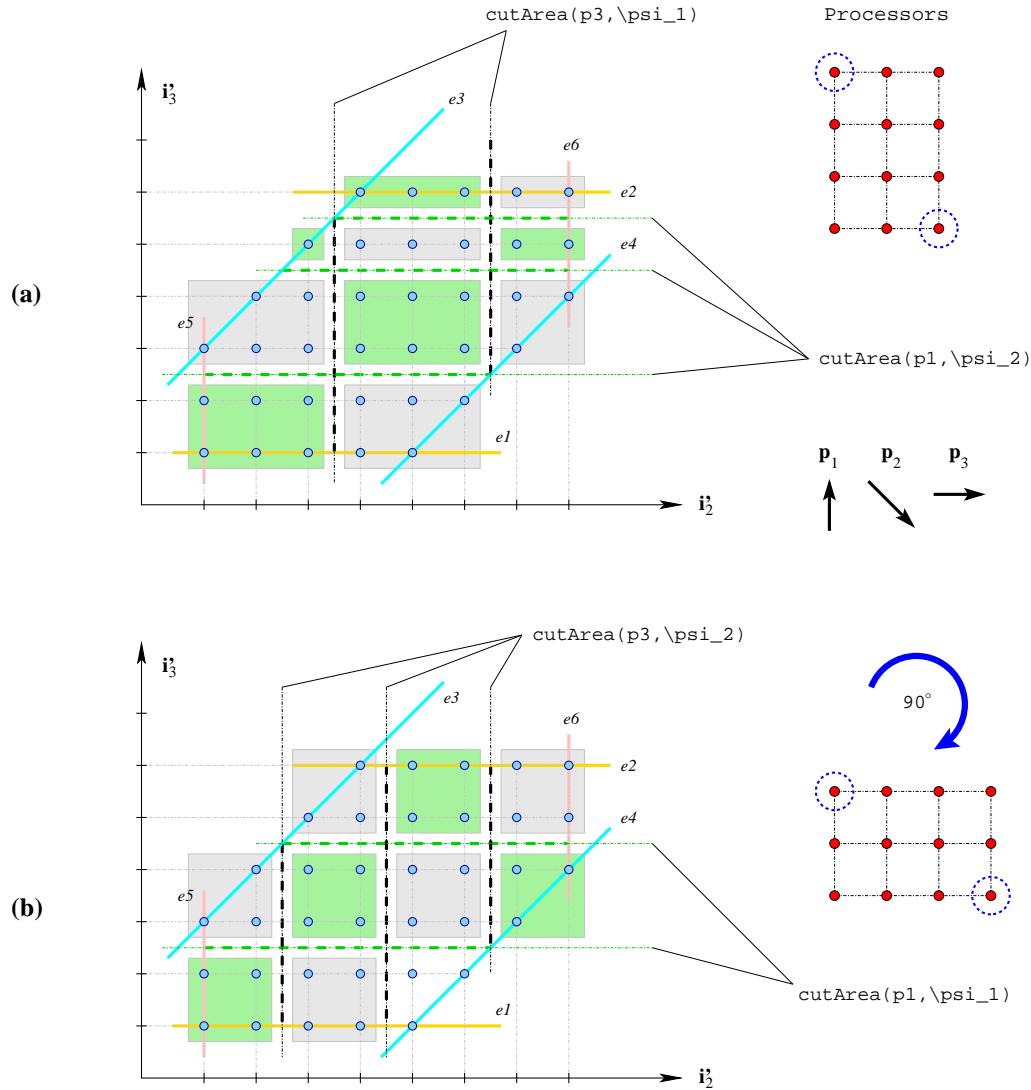
όπου $cutArea$, η συνάρτηση που υπολογίζει το μήκος της τομής που ανήκει στον (μετασχηματισμένο) χώρο δεικτών υλοποιείται με τον Αλγόριθμο 3.4, ο οποίος παρουσιάζεται στο τέλος της παρούσας ενότητας.

Επισημαίνεται η δυνατότητα προσαρμογής του προσανατολισμού της διάταξης των επεξεργαστών στον μετασχηματισμένο χώρο, με σκοπό την αποδοτικότερη απεικόνιση. Συγκεκριμένα, κατά τον οποιονδήποτε τμηματισμό του χώρου, γίνεται έλεγχος της επιβάρυνσης επικοινωνίας για όλα τα ψ_j , σύμφωνα με το βήμα 3 του αλγορίθμου. Ελέγχεται δηλαδή, αν προσφέρεται η διαμέριση του χώρου κατά την συγκεκριμένη διάσταση της διάταξης των επεξεργαστών. Ο έλεγχος πραγματοποιείται για κάθε διάσταση.

Στο σχήμα 3.9 παρουσιάζονται δύο εναλλακτικές απεικονίσεις του ίδιου μετασχηματισμένου χώρου. Παρατηρούμε την αποδοτικότερη χρησιμοποίηση των επεξεργαστών στην δεύτερη περίπτωση (b).

3.7.2 Προσδιορισμός Βέλτιστης Απεικόνισης

Με βάση τον προσδιορισμό του κόστους όλων των πιθανών κατατμήσεων, υπολογίζεται η βέλτιστη απεικόνιση, επιλέγοντας από αυτές με τη μικρότερη επιβάρυνση επικοινωνίας. Οι λεπτομέρειες παρουσιάζονται στον αλγόριθμο που ακολουθεί.



Σχήμα 3.9: Εναλλακτικοί τρόποι διαμέρισης ενός μετασχηματισμένου χώρου δεικτών. Στην δεύτερη περίπτωση, η διάταξη των επεξεργαστών στρέφεται κατά 90° . παρατηρούμε τον αποδοτικότερο καταμερισμό των υπολογισμών στους επεξεργαστές.

Αλγόριθμος 3.3

προσδιορισμός της βέλτιστης απεικόνισης

Εισοδος: οι συντελεστές $mcCost_{i,j}$ και ο πίνακας επεξεργαστών $\pi = (\psi_1, \dots, \psi_\xi)$.

Εξοδος: η βέλτιστη απεικόνιση $M_{opt} = (m_{opt1}, \dots, m_{opt\beta})$.

- για κάθε έγκυρη απεικόνιση $M_k = (m_1, m_2, \dots, m_\beta)$:

1. αρχικοποιείται το κόστος επικοινωνίας της:

$$\text{cost}_k := 0$$

2. για κάθε ζεύγος j συνοριακών γραμμών, $1 \leq j \leq \beta$:

▷ αν $m_i > 1$ τότε:

$$\text{cost}_k := \text{cost}_k + mcCost_{i,j}, 1 \leq j \leq \xi, \text{ έτσι ώστε } \psi_j = m_i$$

3. καταγράφεται η απεικόνιση με το μικρότερο κόστος επικοινωνίας
-

Στο σχήμα 3.10 παρουσιάζεται σχηματικά η διαμέριση ενός μετασχηματισμένου χώρου δεικτών, παράλληλα με ένα από τα ζεύγη των συνοριακών του ευθειών. Σημειώνονται όλα τα κάθετα διανύσματα στις συνοριακές γραμμές, καθώς και η διάταξη των διαθέσιμων επεξεργαστών. Για τη συγκεκριμένη διάταξη είναι $\pi = (\psi_1, \psi_2) = (3, 4)$. Παρατηρούμε ότι απαιτείται ένας ακόμη τμηματισμός (κάθετα ως προς ένα ακόμη ζεύγος συνοριακών ευθειών) ώστε να ολοκληρωθεί η απεικόνιση, διότι στην κατακόρυφη διάσταση έχουμε 7 επαναλήψεις προς υπολογισμό, ενώ διαθέτουμε 4 επεξεργαστές.

3.7.3 Προσδιορισμός Επιφάνειας Τομής

Ο αλγόριθμος που ακολουθεί υπολογίζει το μήκος τομής, που εμπίπτει στον μετασχηματισμένο χώρο δεικτών. Πρόκειται για την εύρεση της τομής ενός υποχώρου διάστασης $n - 2$, με τον μετασχηματισμένο χώρο (διάστασης $n - 1$).

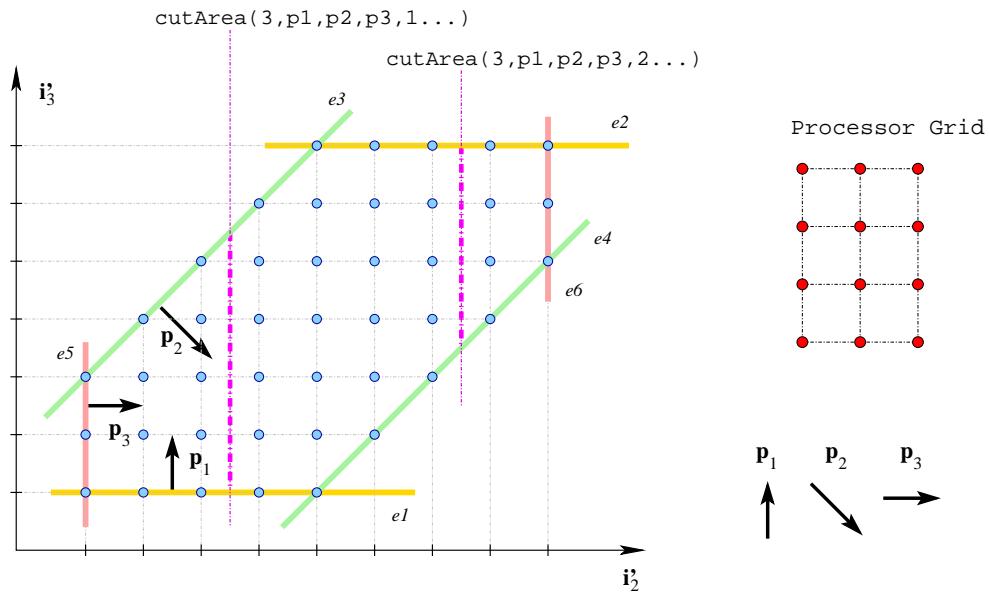
Αλγόριθμος 3.4

υπολογισμός επιφάνειας τομής

Εισοδος: τα κάθετα διανύσματα p_i στις συνοριακές γραμμές, το κάθετο διάνυσμα στην τομή $\mathbf{p} = (p_1, \dots, p_{n-1})$ και η συντεταγμένη p_n η εξίσωση της ευθείας της τομής είναι $p_1x'_1 + p_2x'_2 + \dots + p_{n-1}x'_{n-1} = p_n$.

Εξοδος: το μήκος του ευθυγράμμου τμήματος της υπερευθείας που ανήκει στο μετασχηματισμένο χώρο.

1. έστω P το σύνολο των σημείων που ορίζουν το ευθύγραμμο τμήμα της τομής: αρχικά $P = \emptyset$
2. για όλους τους συνδυασμούς των $n - 2$ συνοριακών γραμμών που δεν είναι κάθετες στο \mathbf{p} (για όλα τα $\mathbf{p}_i \neq \mathbf{p}$):
 - 2.1. προστίθεται η νέα υπερευθεία: οι εξισώσεις ευθειών γίνονται $n - 1$
 - 2.2. επιλύεται το $(n - 1) \times (n - 1)$ διάστασης σύστημα, για τον υπολογισμό του σημείου τομής των $n - 1$ αυτών γραμμών



Σχήμα 3.10: Εύρεση μήκους τομής σε μετασχηματισμένο χώρο δεικτών δύο διαστάσεων. Παρουσιάζεται και η διαδέσιμη διάταξη επεξεργαστών.

2.3. αν η λύση ικανοποιεί και τις εξισώσεις των υπόλοιπων συνοριακών γραμμών, το σημείο προστίθεται στο P . σε διαφορετική περίπτωση απορρίπτεται

3. υπολογίζεται το μήκος του ευθυγράμμου τμήματος που ορίζεται από τα σημεία του P

Στο σχήμα 3.10 παρουσιάζεται σχηματικά η λειτουργία του αλγορίθμου. Πρόκειται ουσιαστικά, για την εύρεση της τομής ενός υποχώρου διάστασης $n - 2$, με τον μετασχηματισμένο χώρο (διάστασης $n - 1$). Για αρχικό χώρο 3 διαστάσεων, προκύπτει μετασχηματισμένος χώρος 2 διαστάσεων και η τομή είναι ευθεία (διάσταση 1).

Δεδομένου ότι όλα τα σημεία του συνόλου P ανήκουν στην ευθεία κατά την οποία πραγματοποιείται η τομή, ορίζεται ένα ευθύγραμμο τμήμα με ακριβώς δύο από αυτά.⁵ Στη γενική περίπτωση προβλημάτων διάστασης n , τα τμήματα των υπεργραμμών είναι πεπερασμένοι υποχώροι διάστασης $n - 2$. Και στην περίπτωση αυτή, παρά το γεγονός ότι το σύνολο P μπορεί να αποτελείται από $k \geq n - 1$ σημεία, ο υπολογισμός του υπερμήκους που καλύπτεται από αυτόν, μπορεί να υπολογιστεί χρησιμοποιώντας ακριβώς $n - 1$ σημεία.

⁵Στην περίπτωση όπου $n = 3$, το σύνολο P θα αποτελείται από δύο σημεία και μόνον, αφού όλα τα υπόλοιπα σημεία θα έχουν απορριφθεί. Αντιθέτως, για $n > 3$, ο αριθμός των σημείων του P μπορεί να είναι μεγαλύτερος.

Λόγω των ανωτέρω, καθίσταται απαραίτητη η δυνατότητα υπολογισμού του υπερμήκους, για μεγαλύτερες διαστάσεις, ξεκινώντας από το Ευκλίδειο μήκος. Στην περίπτωση αυτή, το ζητούμενο μέγεθος είναι το εμβαδό που περικλείεται από τα δούντα σημεία. Για λόγους απλότητας καλείται γενικευμένο μήκος. Στις δύο διαστάσεις εκφυλίζεται στο Ευκλίδειο μήκος.

Ο αλγόριθμος που ακολουθεί, υπολογίζει αναδρομικά την επιφάνεια τομής υποχώρου διάστασης $n - 2$ με τον μετασχηματισμένο χώρο (διάστασης $n - 1$), ξεκινώντας από το Ευκλίδειο μήκος για $n = 3$. Προφανώς αντικαθιστά το βήμα 3 του Αλγορίθμου 3.4, όταν αναφερόμαστε στη γενική περίπτωση, όπου $n \neq 3$.

Αλγόριθμος 3.5

υπολογισμός γενικευμένου μήκους

Εισόδος: τα σημεία του συνόλου P

Εξόδος: το γενικευμένο μήκος του τμήματος της υπερευθείας που ορίζεται από τα n -διάστατα σημεία του P

1. για $n = 3$, ορίζεται ως η Ευκλίδεια απόσταση των δύο πιο απομακρυσμένων σημείων
2. στην περίπτωση όπου $n > 3$:
 - 2.1. αφαιρείται τυχαία ένα σημείο, έστω u
 - 2.2. υπολογίζεται το γενικευμένο μήκος ℓ του τμήματος που ορίζεται από τα $n - 2$ σημεία (προκειται για χώρο διάστασης $n - 3$)
 - 2.3. το σημείο u προβάλλεται στο επίπεδο που ορίζεται από τα $n - 2$ αυτά σημεία· προκύπτει η προβολή του u'
 - 2.4. υπολογίζεται η Ευκλίδεια απόσταση d μεταξύ των u και u'
 - 2.5. το γενικευμένο μήκος είναι το γινόμενο $\ell \cdot d$

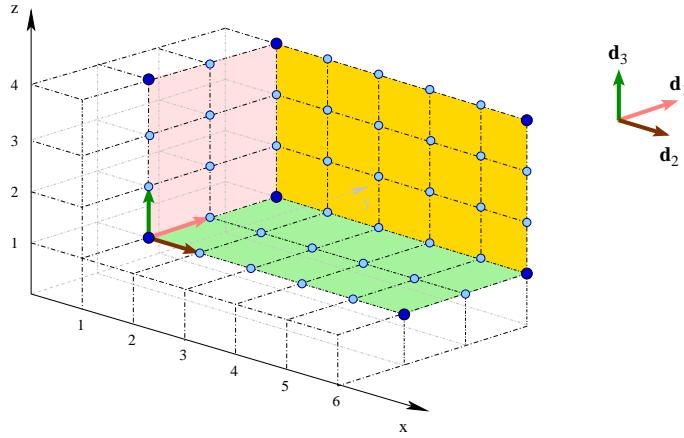
Για την καλύτερη κατανόηση και επεξήγηση της μεθόδου, παραθέτουμε το ακόλουθο παράδειγμα ενός φωλιασμένου βρόχου, με διάσταση 3.

3.8 Παράδειγμα Απεικόνισης

Θεωρούμε τον ακόλουθο φωλιασμένο βρόχο τριών διαστάσεων:

```
for (i0 = 1; i0 ≤ 6; i0++)
    for (i1 = 1; i1 ≤ 4; i1++)
        for (i2 = 1; i2 ≤ 3; i2++)
            a(i0, i1, i2) = a(i0, i1 - 1, i2) + a(i0 - 1, i1, i2) + a(i0, i1, i2 - 1);
```

Αναζητούμε τη βέλτιστη απεικόνιση του βρόχου αυτού, σε μια τετραγωνική διάταξη επεξεργαστών, μεγέθους 3×3 , με δυνατότητα επικοινωνίας οριζόντια και κατακόρυφα. Μια τέτοια διάταξη εικονίζεται στο σχήμα 3.12. Είναι δηλαδή, $\pi = (\psi_1, \psi_1) = (3, 3)$, στο σύνολο εννέα επεξεργαστές.



Σχήμα 3.11: Ο τρισδιάστατος χώρος δεικτών του παραδείγματος χάριν απλότητας, δεν σημειώνονται όλα τα σημεία. Σκιαγραφούνται τρεις από τις πλευρές του χώρου και σημειώνονται οι κορυφές τους (ακραία σημεία του αρχικού χώρου).

Αρχικά εντοπίζονται οι υπάρχουσες εξαρτήσεις: το κάθε σημείο του βρόχου υπολογίζεται χρησιμοποιώντας τρία, ήδη υπολογισμένα, σημεία. Είναι $\mathbf{d}_1 = (0, 1, 0)$, $\mathbf{d}_2 = (1, 0, 0)$ και $\mathbf{d}_3 = (0, 0, 1)$. Ο πίνακας εξαρτήσεων D , καθώς και ο πίνακας μετασχηματισμού T που δίνεται είναι:

$$D = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

Ο πίνακας μετασχηματισμού επελέγει ούτως ώστε να παρουσιάζονται εντονότερα τα χαρακτηριστικά της παρούσας μέθοδου. Στο τέλος του παραδείγματος παραθέτουμε μερικούς ακόμη έγκυρους πίνακες μετασχηματισμού.

Αν $\mathbf{i} = (i_0, i_1, i_2)$ το διάνυσμα δεικτών στον αρχικό χώρο, θεωρούμε $\mathbf{i}' = (i'_0, i'_1, i'_2)$ το

διάνυσμα δεικτών στον μετασχηματισμένο χώρο:

$$\mathbf{i}' = \begin{bmatrix} i'_0 \\ i'_1 \\ i'_2 \end{bmatrix} = T \cdot \mathbf{i} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} i_0 \\ i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} i_0 + i_1 + i_2 \\ i_0 + i_1 \\ i_1 + i_2 \end{bmatrix}.$$

Αντίστοιχα βρίσκουμε και τον μετασχηματισμένο πίνακα εξαρτήσεων D' , ως γινόμενο του αρχικού πίνακα εξαρτήσεων με τον πίνακα μετασχηματισμού:

$$D' = T \cdot D = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}.$$

Η πρώτη γραμμή, του μετασχηματισμένου πίνακα εξαρτήσεων, δείχνει τη χρονική εξάρτηση ως προς το κάθε διάνυσμα εξάρτησης και αγνοείται. Ο υπολοιπόμενος πίνακας έχει ως στήλες τα νέα διανύσματα εξάρτησης στον μετασχηματισμένο χώρο. Στο παρόντα μας είναι:

$$D' = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \Rightarrow \begin{cases} \mathbf{d}'_1 = (1, 1) \\ \mathbf{d}'_2 = (1, 0) \\ \mathbf{d}'_3 = (0, 1) \end{cases}$$

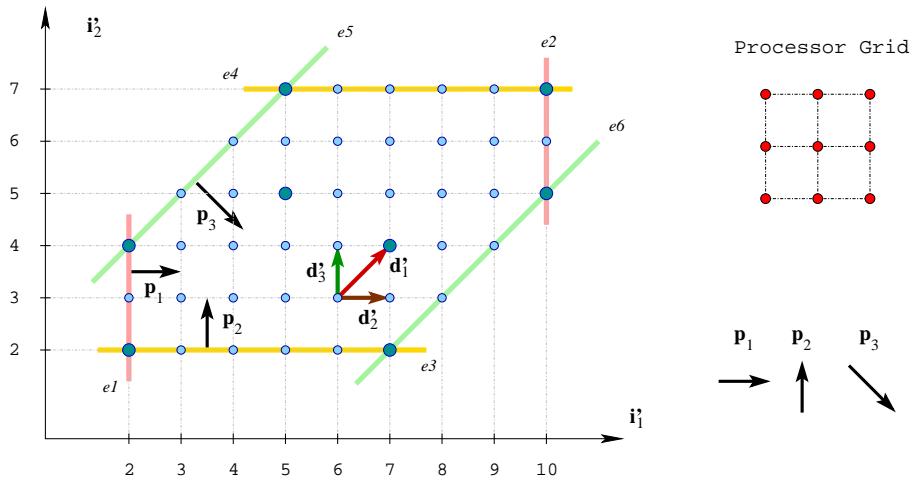
Στα σχήματα 3.11 και 3.12 παρουσιάζονται ο χώρος δεικτών $J^3 = \{(i_0, i_1, i_2) \mid i_j \in \mathbb{Z}, \text{όπου } 1 \leq i_0 \leq 6, 1 \leq i_1 \leq 3, 1 \leq i_2 \leq 4, 0 \leq j \leq 2\}$, καθώς και ο μετασχηματισμένος χώρος, που προκύπτει με την εφαρμογή του παραπάνω μετασχηματισμού T .

Ακολουθώντας τον Αλγόριθμο 3.1, σχηματίζουμε τους πίνακες που απαιτούνται για τον αναλυτικό προσδιορισμό των συνοριακών γραμμών:

$$V = \begin{bmatrix} 1 & 1 & 1 & 1 & 6 & 6 & 6 & 6 \\ 1 & 1 & 4 & 4 & 1 & 1 & 4 & 4 \\ 1 & 3 & 1 & 3 & 1 & 3 & 1 & 3 \end{bmatrix} \Rightarrow V' = T \cdot V = \begin{bmatrix} 3 & 5 & 6 & 8 & 8 & 10 & 11 & 13 \\ 2 & 2 & 5 & 5 & 7 & 7 & 10 & 10 \\ 2 & 4 & 5 & 7 & 2 & 4 & 5 & 7 \end{bmatrix} \Rightarrow$$

$$W = \begin{bmatrix} 2 & 2 & 5 & 5 & 7 & 7 & 10 & 10 \\ 2 & 4 & 5 & 7 & 2 & 4 & 5 & 7 \end{bmatrix}.$$

Οι στήλες του πίνακα W δίνουν τα πιθανά ακρότατα σημεία του μετασχηματισμένου χώρου, όπως φαίνεται και στο σχήμα 3.12. Παίρνοντας το κυρτό τους περίγραμμα, προσδιορίζουμε τα



Σχήμα 3.12: Ο μετασχηματισμένος χώρος δεικτών (δύο διαστάσεων) του παραδείγματος μας. Σημειώνονται οι μετασχηματισμοί των ακραίων σημείων του αρχικού χώρου, οι συνοριακές γραμμές που αντά ορίζουν (σημειώνονται και τα κάθετα διανύσματα σ' αυτές) και οι μετασχηματισμένες εξαρτήσεις. Ο διαθέσιμος χώρος επεξεργαστών είναι μια τετραγωνική διάταξη 3×3 : $\pi = (\psi_1, \psi_1) = (3, 3)$.

σημεία αυτά: $(2, 2), (7, 2), (10, 5), (10, 7), (5, 7)$ και $(2, 4)$. Το χυρτό περίγραμμα που σχηματίζεται είναι ένα πολύγωνο, με πλευρές παράλληλες ανά δύο. Οι ανισότητες οι οποίες περιγράφουν το εσωτερικό του πολυγώνου αυτού είναι:

$$\text{Ζευγάρι 1: } 2 \leq i'_1 \leq 10 \quad -\text{με κάθετο διάνυσμα το } \mathbf{p}_1 = (0, 1) \Leftrightarrow \begin{cases} \varepsilon_1: & i'_1 - 2 = 0 \\ \varepsilon_2: & i'_1 - 10 = 0 \end{cases}$$

$$\text{Ζευγάρι 2: } 2 \leq i'_2 \leq 7 \quad -\text{με κάθετο διάνυσμα το } \mathbf{p}_2 = (1, 0) \Leftrightarrow \begin{cases} \varepsilon_3: & i'_2 - 2 = 0 \\ \varepsilon_4: & i'_2 - 7 = 0 \end{cases}$$

$$\text{Ζευγάρι 1: } -2 \leq i'_3 \leq 5 \quad -\text{με κάθετο διάνυσμα το } \mathbf{p}_3 = (1, -1) \Leftrightarrow \begin{cases} \varepsilon_5: & i'_3 + 2 = 0 \\ \varepsilon_6: & i'_3 - 5 = 0 \end{cases}$$

όπου i'_3 , ένας δείκτης που αντιστοιχεί στη διάσχιση του μετασχηματισμένου χώρου, κατά τη διεύθυνση του \mathbf{d}'_1 : $i'_3 = i'_1 - i'_2$. Προφανώς υπάρχει ένα τουλάχιστον ζεύγος συνοριακών γραμμών παράλληλο με κάθε διάνυσμα εξάρτησης.

Μετά τον υπολογισμό των συνόρων, γνωρίζουμε όλες τις πιθανές διευθύνσεις τομής. Στο παράδειγμα μας είναι τρεις και, δεδομένου ότι αναζητούμε την επιθυμητή διαμέριση του χώρου με

τη μικρότερη επιβάρυνση επικοινωνίας, επιλέγουμε τις δύο από αυτές με τη μικρότερη απαιτούμενη επικοινωνία.

Αρχικά υπολογίζουμε τους συντελεστές κόστους για όλες τις δυνατές τομές. Σημειώνουμε ότι απαιτείται το “κόψιμο” του χώρου σε τρία τμήματα, αφού οι διαθέσιμοι επεξεργαστές είναι τρεις σε κάθε διάσταση. Εφαρμόζοντας τον Αλγόριθμο 3.2, υπολογίζουμε τους συντελεστές κόστους για τυχόν “τομή” του χώρου παράλληλα με το κάθε ένα από τα τρία ζεύγη συνοριακών γραμμών, δεδομένου ότι: $\mathbf{p}_1 = (0, 1)$, $\mathbf{p}_2 = (1, 0)$ και $\mathbf{p}_3 = (1, 1)$:

$$\begin{aligned} depCost_1 &= \frac{1}{|\mathbf{p}_1|} \cdot (|\mathbf{p}_1 \cdot \mathbf{d}'_1| + |\mathbf{p}_1 \cdot \mathbf{d}'_2| + |\mathbf{p}_1 \cdot \mathbf{d}'_3|) = \frac{1}{1} \cdot (|1| + |1| + |0|) = 2 \\ depCost_2 &= \frac{1}{|\mathbf{p}_2|} \cdot (|\mathbf{p}_2 \cdot \mathbf{d}'_1| + |\mathbf{p}_2 \cdot \mathbf{d}'_2| + |\mathbf{p}_2 \cdot \mathbf{d}'_3|) = \frac{1}{1} \cdot (|1| + |0| + |1|) = 2 \\ depCost_3 &= \frac{1}{|\mathbf{p}_3|} \cdot (|\mathbf{p}_3 \cdot \mathbf{d}'_1| + |\mathbf{p}_3 \cdot \mathbf{d}'_2| + |\mathbf{p}_3 \cdot \mathbf{d}'_3|) = \frac{1}{\sqrt{2}} \cdot (|0| + |1| + |-1|) = \sqrt{2}. \end{aligned}$$

Στο σχήμα 3.13, φαίνεται η επικοινωνία που απαιτείται για τον τμηματισμό του μετασχηματισμένου χώρου ως προς κάθε διεύθυνση. Μια πλήρης διαμέριση απαιτεί τομές ως προς δύο, μη παράλληλα, σύνορα του χώρου. Το μήκος που αντιστοιχεί σε κάθε τομή είναι το τμήμα της ευθείας το οποίο ανήκει στο χώρο δεικτών (ευθύγραμμα τμήματα l_1, l_2, \dots, l_6 του σχήματος).

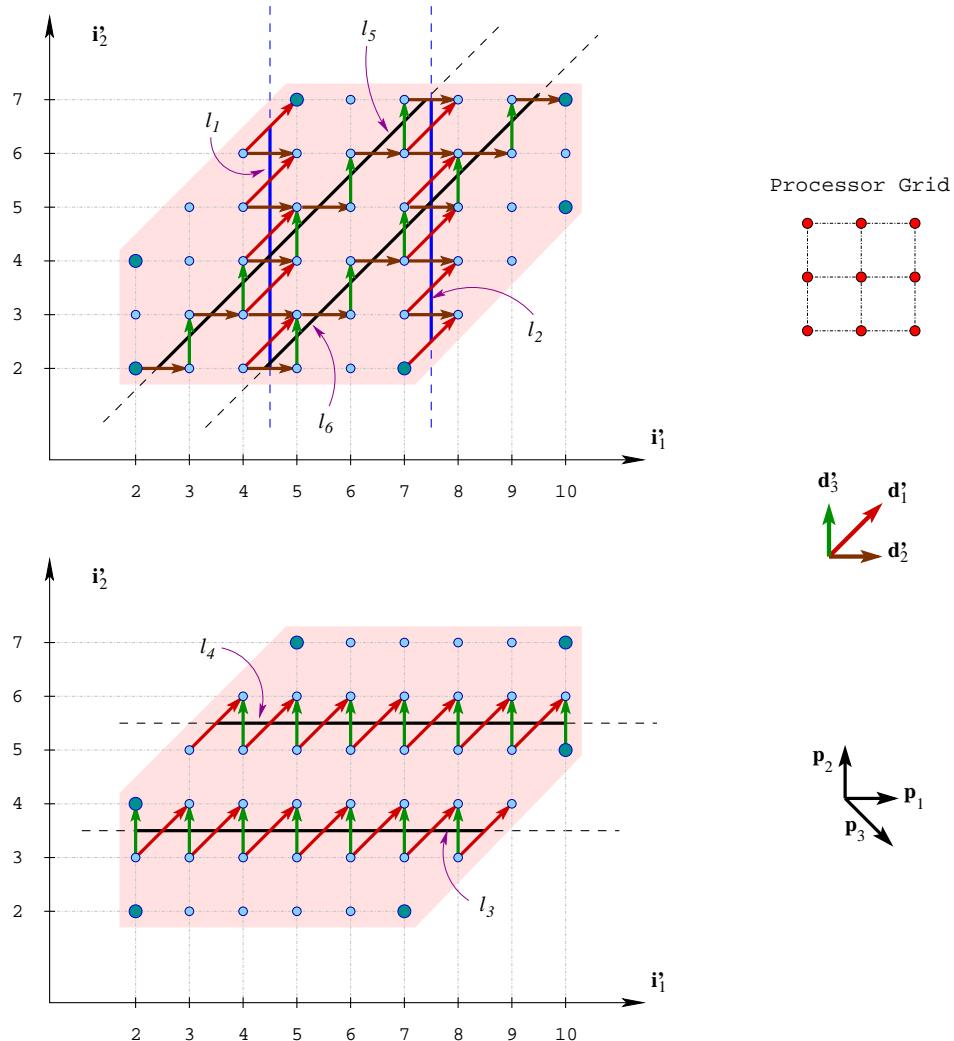
Στην συνέχεια, υπολογίζουμε το συνολικό κόστος διαμέρισης⁶ του χώρου σε τρία τμήματα ($\pi = (3, 3) \Rightarrow \psi_1 = 3$), κατά την κάθε διεύθυνση:

$$\begin{aligned} mcCost_{1,1} &= depCost_1 \cdot (|l_1| + |l_2|) = 2 \cdot (4.5 + 4.5) = 18 \\ mcCost_{2,1} &= depCost_2 \cdot (|l_3| + |l_4|) = 2 \cdot (6.5 + 6.5) = 26 \\ mcCost_{3,1} &= depCost_3 \cdot (|l_5| + |l_6|) = \sqrt{2} \cdot (5\sqrt{2} + 5\sqrt{2}) = 20 \end{aligned}$$

όπου τα μήκη των τμημάτων υπολογίζονται με τη βοήθεια των Αλγορίθμων 3.4 και 3.5.

Τστερα από τα παραπάνω, δεν είναι δύσκολο να διαπιστώσει κανείς ότι η βέλτιστη απεικόνιση –αυτή που παρουσιάζει το μικρότερο κόστος επικοινωνίας– αποτελείται από τις δύο διαμερίσεις

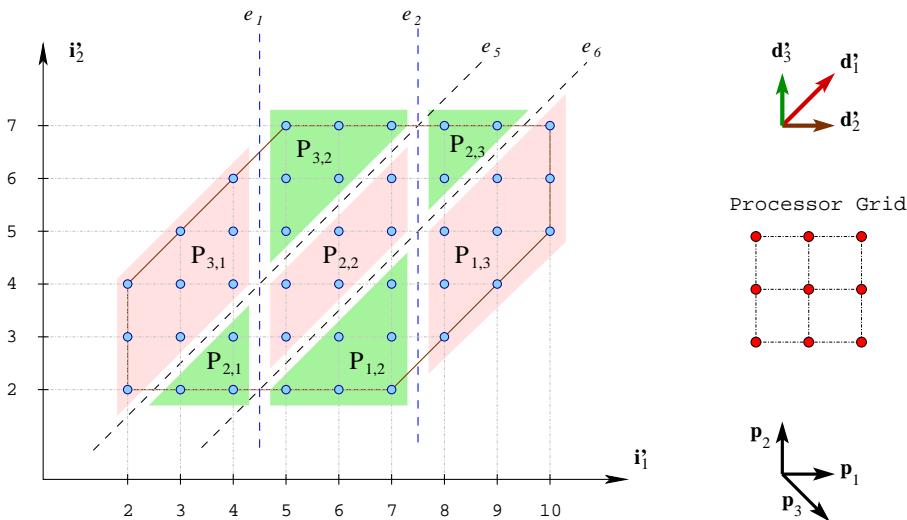
⁶Ως πραγματικό κόστος επικοινωνίας, θεωρούμε τον αριθμό των διανυσμάτων εξάρτησης που τέμνονται από τις ευθείες διαμέρισης. Λόγω του ότι ο χώρος μας είναι διακριτός και όχι συνεχής, παρουσιάζεται διαφορά μεταξύ του υπολογίζομένου κόστους και του πραγματικού. Η διαφορά αυτή γενικά είναι μικρή και δεν επηρεάζει την κρίση μας, όσον αφορά το κόστος τομής, γιατί μας ενδιαφέρει μόνο συγκριτικά. Το πραγματικό κόστος τομής παρουσιάζεται στο σχήμα 3.13 και είναι 20, 28 και 22 μονάδες επικοινωνίας, για τις τρεις διευθύνσεις τομής, αντίστοιχα.



Σχήμα 3.13: Διαμέριση του μετασχηματισμένου χώρου κατά τις τρεις διευθύνσεις των συνόρων του και αντίστοιχο κόστος επικοινωνίας. Στο πρώτο σχήμα παρουσιάζεται μια πλήρης διαμέριση, η οποία πραγματοποιείται κάθετα προς τα p_1 και p_3 . Στο δεύτερο σχήμα παρουσιάζεται μια μη πλήρης διαμέριση, κάθετα προς το p_2 .

του χώρου, κάθετα με το πρώτο και κάθετα με το τρίτο ζευγάρι συνόρων. Αυτή η διαμέριση επιλέγεται από τον Αλγόριθμο 3.3 και παρουσιάζεται αναλυτικά στο σχήμα 3.14.

Το υπολογιζόμενο συνολικό κόστος επικοινωνίας είναι $mcCost_{1,1} + mcCost_{3,1} = 38$ μονάδες



Σχήμα 3.14: Η διαμέριση και η απεικόνιση που προτείνονται για το παράδειγμά μας.

επικοινωνίας, ενώ το πραγματικό είναι 42. ■

Σημειώνουμε στο σημείο αυτό, ότι όταν ο μετασχηματισμένος χώρος δεν είναι παραλληλόγραμμο ($n - 1$ διαστάσεων), δεν αξιοποιείται το σύνολο των διαθέσιμων υπολογιστικών μονάδων. Στο παράδειγμα μας π.χ., οι επεξεργαστές $P_{1,1}$ και $P_{3,3}$ δεν χρησιμοποιούνται καθόλου.

Όσον αφορά την εκλογή του πίνακα μετασχηματισμού T , ακολουθήθηκε η μέθοδος των Lee και Kedem, για την επιλογή του βέλτιστου μετασχηματισμού κατά την απεικόνιση σε συστολικές διατάξεις [LK90]. Η εν λόγω μέθοδος, για το παράδειγμα μας, δίνει τους ακόλουθους πίνακες μετασχηματισμού:

$$T_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad T_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad T_3 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad T_4 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Οι πίνακες αυτοί καταλήγουν σε συστολικές διατάξεις με 42, 24, 12 και 12 μονάδες επεξεργασίας, αντίστοιχα. Και οι τέσσερις ακολουθούν το βέλτιστο υπερεπίπεδο δρομολόγησης $\Pi = (1, 1, 1)$. Ο βέλτιστος μετασχηματισμός, για τις ανάγκες του προβλήματος μας, είναι ο T_4 . Ο πίνακας αυτός επιτυγχάνει απεικόνιση στον ελάχιστο αριθμό επεξεργαστών, σύμφωνα με την εργασία [AKPT99] και προτιμάται έναντι του T_3 , διότι παράγει συστολική διάταξη που απαιτεί μόνο δύο εξωτερικές συνδέσεις επικοινωνίας (δύο μετασχηματισμένα διανύσματα εξάρτησης).

Στο παράδειγμα μας επιλέξαμε τον πίνακα T_1 , επειδή ο μετασχηματισμένος χώρος που παράγεται είναι αρκετά μεγάλος ώστε να γίνει καλύτερα αντιληπτή η μέθοδος που προτείνουμε.

Κεφάλαιο 4

Υπολογισμός των Περιγραμμάτων Εκτέλεσης

Κατά την εκτέλεση ενός φωλιασμένου βρόχου, η διάδοση των υπολογισμών υπήρξε ανέκαθεν πρόβλημα εξέχουσας σημασίας. Σε προβλήματα συνοριακών τιμών, του τύπου της διάδοσης θερμότητας ή δυναμικού, μας ενδιαφέρει όχι μόνο αυτή καθεαυτή η διάδοση, αλλά και η οικριβής περιγραφή της κυματομορφής κατά την εξέλιξη του φαινομένου.

Στο κεφάλαιο αυτό μελετάται ο σχηματισμός των κυματομορφών εκτέλεσης (execution patterns) για τα προβλήματα φωλιασμένων βρόχων με ομοιόμορφες εξαρτήσεις. Όπως θα δούμε, ο οικριβής και αναλυτικός προσδιορισμός της κυματομορφής, οδηγεί σε ένα πρόβλημα ακέραιου γραμμικού προγραμματισμού. Το ενδιαφέρον του προβλήματος έγκειται στο ότι, σε κάθε περίπτωση, μπορούμε να βρούμε μια προσεγγιστική περιγραφή αυτής της κυματομορφής σε πολυωνυμικό χρόνο, αποφεύγοντας τον ακέραιο γραμμικό προγραμματισμό.

Μελετώνται οι σχέσεις που πρέπει να ισχύουν, ώστε αυτή η προσεγγιστική περιγραφή να ταυτίζεται με την οικριβή. Στις περιπτώσεις αυτές έχουμε αλγορίθμους με ιδιαίτερα χαρακτηριστικά, των οποίων η οικριβής κυματομορφή μπορεί να υπολογιστεί σε πολυωνυμικό ή ακόμη και γραμμικό χρόνο. Τα ιδιαίτερα αυτά χαρακτηριστικά επισημαίνονται και διατυπώνονται με τη μορφή σχέσεων μεταξύ των διανυσμάτων εξάρτησης.

Τα αποτελέσματα αυτού του κεφαλαίου, αποτελούν τη βάση για τη διατύπωση νέων τεχνικών δρομολόγησης, οι οποίες λαμβάνουν υπ' όψιν και τυχόν καθυστερήσεις επικοινωνίας· με τις τεχνικές αυτές θα ασχοληθούμε στα επόμενα κεφάλαια.

4.1 Προσδιορισμός των Περιγραμμάτων Εκτέλεσης

Αρχικά θα προσπαθήσουμε να προσεγγίσουμε τον τρόπο σχηματισμού των σχημάτων της κυματομορφής εκτέλεσης (patterns) και των περιγραμμάτων τους (pattern outlines). Σκοπός μας είναι να περιγράψουμε τον σχηματισμό του περιγράμματος pat_j της τυχαίας χρονικής στιγμής j . Όπως θα φανεί στη συνέχεια, το οποιοδήποτε περίγραμμα εξαρτάται μόνο από το αρχικό περίγραμμα εκτέλεσης pat_1 και από τα διανύσματα που το αποτελούν.

Η βασική ιδέα πίσω από τα σχήματα κυματομορφής και τα περιγράμματα τους, είναι ότι σχηματίζονται από την τομή πεπερασμένων υποχώρων του J . Αρχικά, υποδεικνύουμε τους υποχώρους που συμβάλλουν στον σχηματισμό του κάθε σχήματος κυματομορφής και στη συνέχεια, παρουσιάζουμε δύο αλγορίθμους που υπολογίζουν τα αντίστοιχα περιγράμματα. Η έρευνα μας καταλήγει σε ένα κλειστό τύπο ο οποίος περιγράφει το περίγραμμα εκτέλεσης pat_j για την οποιαδήποτε χρονική στιγμή j .

Η μελέτη μας ξεκινάει από τον προσδιορισμό του αρχικού σχήματος κυματομορφής, δεδομένης της ακολουθίας των διανυσμάτων εξάρτησης.

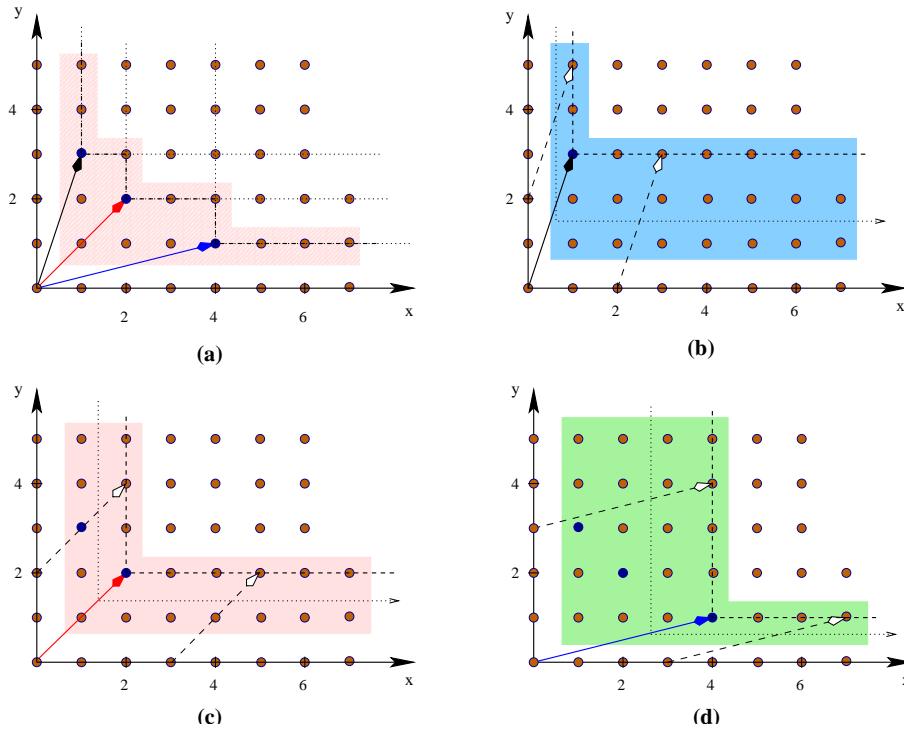
4.1.1 Σχηματισμός του Αρχικού Σχήματος Κυματομορφής

Ο σχηματισμός του αρχικού pattern δείχνει τη λογική πίσω από τη μέθοδο μας. Δεδομένων των προϋπολογισμένων σημείων (σημεία των αξόνων και σημεία εκτός του χώρου μας), κάθε διάνυσμα εξάρτησης περιορίζει τα σημεία του χώρου που μπορούν να υπολογιστούν την πρώτη χρονική στιγμή $t = 1$ (βλ. σχήμα 4.1).

Έστω S_{j-1} τα ήδη υπολογισμένα σημεία κατά τη χρονική στιγμή $t = j$ και $S_j(\mathbf{d})$ το σύνολο των σημείων που μπορούν να υπολογιστούν κατά τη χρονική στιγμή αυτή, λόγω του διανύσματος \mathbf{d} . Τότε, τη χρονική στιγμή $t = 1$, εξ' αιτίας του διανύσματος εξάρτησης \mathbf{d}_i , μπορούν να υπολογιστούν τα σημεία που ανήκουν στο σύνολο $S_1(\mathbf{d}_i) = S_0 + \mathbf{d}_i$. Το σύνολο όλων των σημείων τα οποία μπορούν να υπολογιστούν τη χρονική στιγμή $t = 1$ δίνονται αν πάρουμε την τομή όλων αυτών των συνόλων, με τους επιμέρους περιορισμούς, δηλαδή:

$$S_1 = S_1(\mathbf{d}_1) \cap S_1(\mathbf{d}_2) \cap \dots \cap S_1(\mathbf{d}_m) = \bigcap_{i=1}^m (S_0 + \mathbf{d}_i) \quad (4.1)$$

Τυποδεικνύεται από αυτά η σχέση που συνδέει τον υπολογισμό του αρχικού σχήματος κυματομορφής, με τον υπολογισμό της τομής πεπερασμένου αριθμού υποχώρων (αυτών που το



Σχήμα 4.1: Το αρχικό *pattern* ενός αλγορίθμου με 3 διανύσματα εξάρτησης. Οι υποχώροι των σημείων που μπορούν να εκτελεστούν για $t = 1$, λόγω κάθε διανύσματος εξάρτησης (b-d). Η τομή των τριών αυτών υποχώρων σχηματίζει το αρχικό σχήμα κυματομορφής και το περιγραμμα του (a).

δημιουργούν). Ο αριθμός των υποχώρων αυτών ισούται με το πλήθος των διανυσμάτων εξάρτησης.

Επισημαίνουμε τη συμβολή των άκρων των διανυσμάτων \mathbf{d}_i στο σχηματισμό του αντίστοιχου περιγράμματος. Τα σημεία αυτά ονομάζονται σημεία κυματομορφής (*pattern points*) και είναι αυτά που αρκούν για τον ακριβή καθορισμό, τόσο του αρχικού σχήματος κυματομορφής, όσο και του περιγράμματός του. Τα υπόλοιπα σημεία συμπληρώνουν την τευχλασμένη γραμμή. Κατόπιν αυτού, τα διανύσματα κυματομορφής (*pattern vectors*) ορίζονται ως εξής:

Ορισμός 4.1 Ως διανύσματα κυματομορφής (*pattern vectors*) $\mathbf{p}_1, \dots, \mathbf{p}_k$, ορίζουμε το μικρότερο υποσύνολο των διανυσμάτων εξάρτησης για το οποίο ισχύει:

$$\bigcap_{i=1}^k (S_0 + \mathbf{p}_i) = \bigcap_{i=1}^m (S_0 + \mathbf{d}_i), \text{ óπου } k \leq m. \quad (4.2)$$

Βλέπουμε λοιπόν, ότι το πρόβλημα εύρεσης του αρχικού σχήματος κυματομορφής και του περιγράμματος του, ανάγεται στον προσδιορισμό των διανυσμάτων κυματομορφής από τα διανύσματα εξάρτησης. Η σχέση 4.2 όμως, δεν οδηγεί άμεσα σε κάποιο τρόπο αναγνώρισης και προσδιορισμού των διανυσμάτων κυματομορφής.

Μια διαφορετική προσέγγιση για τον προσδιορισμό των διανυσμάτων κυματομορφής, μπορεί να γίνει μέσω της “μεταφοράς” του χώρου δεικτών, όπως ορίστηκε στο Κεφάλαιο 2: $J_{\mathbf{d}} = \{\mathbf{j} = (j_x, j_y) \in J, \text{όπου } d_x \leq j_x \text{ και } d_y \leq j_y\}$, όπου $\mathbf{d} = (d_x, d_y)$, ένα διάνυσμα εξάρτησης. Χρησιμοποιώντας αυτόν τον ορισμό, μπορούμε να ορίσουμε τα διανύσματα κυματομορφής με έναν ισοδύναμο τρόπο:

Ορισμός 4.2 Ως διανύσματα κυματομορφής (*pattern vectors*) $\mathbf{p}_1, \dots, \mathbf{p}_k$, ορίζουμε το μικρότερο υποσύνολο των διανυσμάτων εξάρτησης για το οποίο ισχύει:

$$J_{\mathbf{p}_1} \cup J_{\mathbf{p}_2} \cup \dots \cup J_{\mathbf{p}_k} = J_{\mathbf{d}_1} \cup J_{\mathbf{d}_2} \cup \dots \cup J_{\mathbf{d}_m}, \text{ όπου } k \leq m. \quad (4.3)$$

Οι δύο αυτοί ορισμοί υποδηλώνουν την χαρακτηριστική ιδιότητα των διανυσμάτων κυματομορφής: είναι τα διανύσματα εξάρτησης τα οποία καθορίζουν την κυματομορφή εκτέλεσης. Τα υπόλοιπα διανύσματα εξάρτησης δεν την επηρεάζουν, είτε λόγω μεγέθους, είτε λόγω κλίσης (βλ. σχήμα 4.2).

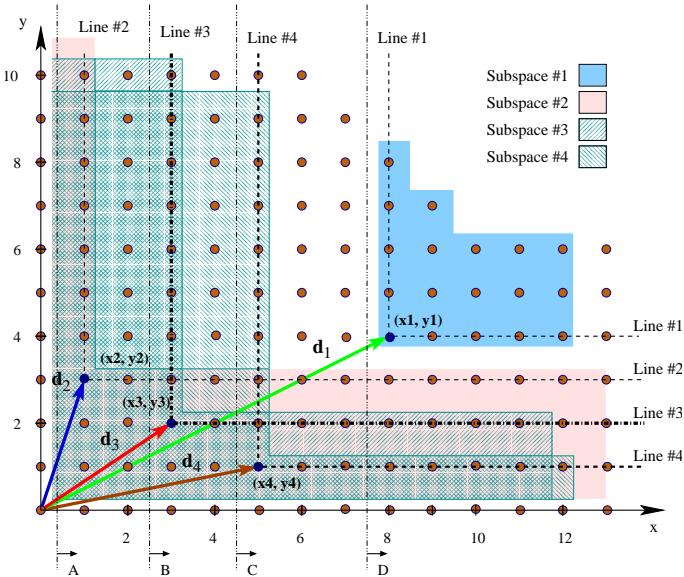
Ένα πιο πολύπλοκο παράδειγμα σχηματισμού του αρχικού pattern, από τους υποχώρους των διανυσμάτων εξάρτησης, φαίνεται στο σχήμα 4.2.

4.1.2 Προσδιορισμός του Αρχικού Περιγράμματος

Παρουσιάζουμε στην παράγραφο αυτή, δύο αλγορίθμους οι οποίοι λύνουν αποδοτικά το πρόβλημα του προσδιορισμού των διανυσμάτων κυματομορφής (pattern vectors), σύμφωνα με τους ορισμούς που παρουσιάστηκαν παραπάνω.

Πηγάζουν από τους Ορισμούς 4.1 και 4.2. Ο πρώτος προσδιορίζει την τομή m υποχώρων στις δύο διαστάσεις, ακολουθώντας την τεχνική της σάρωσης του επιπέδου (τεχνική *plane-sweeping*¹). Ο αλγόριθμος αυτός είναι από την περιοχή της υπολογιστικής γεωμετρίας και το κόστος του είναι πολυωνυμικό. Ο δεύτερος εφαρμόζει πιστά την ιδιότητα της σχέσης 4.3.

¹Για περισσότερες πληροφορίες για τους αλγορίθμους αυτού του είδους, ο ενδιαφερόμενος αναγνώστης παραπέμπεται στο βιβλίο [dBvKOS97]



Σχήμα 4.2: Σχηματισμός του αρχικού pattern μέσω της τομής των υποχώρων των διανυσμάτων εξάρτησης. Μόνο τα διανύσματα \mathbf{d}_2 , \mathbf{d}_3 και \mathbf{d}_4 συντελούν στη δημιουργία του περιγράμματος.

Ως είσοδο παίρνουν m σημεία $\mathbf{d}_1, \dots, \mathbf{d}_m$ και επιστρέφουν το μικρότερο υποσύνολο τους, το οποίο επαληθεύει τη σχέση:

$$J_{\mathbf{p}_1} \cup J_{\mathbf{p}_2} \cup \dots \cup J_{\mathbf{p}_k} = J_{\mathbf{d}_1} \cup J_{\mathbf{d}_2} \cup \dots \cup J_{\mathbf{d}_m} \quad (4.4)$$

Στην περίπτωση αυτή λέμε ότι τα $\mathbf{p}_1, \dots, \mathbf{p}_k$ αποτελούν το **minimum cover** των $\mathbf{d}_1, \dots, \mathbf{d}_m$ και το συμβολίζουμε ως εξής:

$$\langle \mathbf{p}_1, \dots, \mathbf{p}_k \rangle = \mathcal{MC}(\langle \mathbf{d}_1, \dots, \mathbf{d}_m \rangle).$$

Αλγόριθμος 4.1

Minimum Cover - σάρωση επιπέδου

INPUT: m 2D σημεία $\langle \mathbf{j}_1, \dots, \mathbf{j}_m \rangle$.

OUTPUT: Τα k 2D σημεία $\langle \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k \rangle$, $k \leq m$, τέτοια ώστε $\langle \mathbf{p}_1, \dots, \mathbf{p}_k \rangle = \mathcal{MC}(\langle \mathbf{j}_1, \dots, \mathbf{j}_m \rangle)$.

1. Αρχικά $\mathcal{MC}(\langle \mathbf{j}_1, \dots, \mathbf{j}_m \rangle) = \langle \rangle$ –η κενή ακολουθία.
2. Ξεκινώντας από την αρχή των αξόνων, σαρώνουμε το επίπεδο προς τα δεξιά με μια κάθετη ευθεία, η οποία ονομάζεται ευθεία σάρωσης.

3. Καθώς κινούμαστε κατά μήκος του άξονα των x , προσθέτουμε στο minimum cover το πρώτο από τα σημεία που συναντάμε· έτσι έχουμε: $\mathcal{MC}(\langle j_1, \dots, j_m \rangle) = \langle p_1 \rangle$.
4. Συνεχίζουμε να μετακινούμε την ευθεία σάρωσης προς τα δεξιά. Για κάθε σημείο που συναντάμε, ελέγχεται η γ-συντεταγμένη του· αν είναι μικρότερη από την γ-συντεταγμένη του τελευταίου σημείου που προστέθηκε, προστίθεται και το τρέχον σημείο.
5. Επαναλαμβάνεται το Βήμα 4, έως ότου προσπεράσουμε όλα τα σημεία.
6. Η ακολουθία που σχηματίζεται αποτελεί το minimum cover των αρχικών σημείων:

$$J_{p_1} \cup \dots \cup J_{p_k} = J_{j_1} \cup \dots \cup J_{j_m}.$$

Το πλεονέκτημα αυτού του πρώτου αλγορίθμου είναι ότι έχει πολυπλοκότητα $\mathcal{O}(m \log m)$, ενώ το κυριότερο μειονέκτημα του είναι ότι αναφέρεται στις δύο διαστάσεις.

Για τον υπολογισμό του minimum cover σημείων ανώτερης διάστασης, χρησιμοποιούμε τον δεύτερο αλγόριθμο, ο οποίος εφαρμόζεται σε οποιαδήποτε διάσταση αλλά η πολυπλοκότητα που παρουσιάζει είναι σημαντικά χειρότερη $\Theta(m^2)$.

Αλγόριθμος 4.2

Minimum Cover - n-D

INPUT: Ακολουθία m σημείων $Q = \langle j_1, \dots, j_m \rangle$.

OUTPUT: Τα k σημεία $\langle p_1, p_2, \dots, p_k \rangle$, $k \leq m$, τέτοια ώστε $\langle p_1, \dots, p_k \rangle = \mathcal{MC}(\langle j_1, \dots, j_m \rangle)$.

1. Αρχικά $\mathcal{MC}(Q) = \langle \rangle$ –η κενή ακολουθία.

2. Ελέγχουμε τα δύο πρώτα σημεία του Q :

- αν $j_1 \in J_{j_2} \Rightarrow$ προσθέτουμε το j_2 στο $\mathcal{MC}(Q)$
- αν $j_2 \in J_{j_1} \Rightarrow$ προσθέτουμε το j_1 στο $\mathcal{MC}(Q)$
- αν τίποτε από τα δύο δεν ισχύει $\Rightarrow \mathcal{MC}(Q) = \langle j_1, j_2 \rangle$.

3. Έστω $q = j_3$ το τρίτο σημείο του Q . Για όλα τα σημεία p_i που έχουν τοποθετηθεί ήδη στο $\mathcal{MC}(Q)$:

- εάν $p_i \in J_q \Rightarrow$ αφαιρούμε το p_i από το $\mathcal{MC}(Q)$ και προσθέτουμε το q
- εάν $p_i \notin J_q$ και $q \notin J_{p_i} \Rightarrow$ προσθέτουμε το q στο $\mathcal{MC}(Q)$

4. Επαναλαμβάνεται το Βήμα 3 για όλα τα υπόλοιπα σημεία j_4, j_5, \dots, j_m του Q .

5. Η ακολουθία που σχηματίζεται αποτελεί το minimum cover των αρχικών σημείων:

$$J_{\mathbf{p}_1} \cup \dots \cup J_{\mathbf{p}_k} = J_{\mathbf{j}_1} \cup \dots \cup J_{\mathbf{j}_m}.$$

Κατόπιν αυτών, για τον υπολογισμό του περιγράμματος του αρχικού pattern ενός αλγορίθμου με ακολουθία διανυσμάτων εξάρτησης $\langle \mathbf{d}_1, \dots, \mathbf{d}_m \rangle$, δεν έχουμε παρά να υπολογίσουμε το minimum cover των διανυσμάτων εξάρτησης, με τη βοήθεια ενός εκ των παραπάνω αλγορίθμων. Είναι λοιπόν:

$$pat_1 = \mathcal{MC}(\langle \mathbf{d}_1, \dots, \mathbf{d}_m \rangle) = \langle \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k \rangle = PV \quad (4.5)$$

Εξ' ορισμού τα διανύσματα αυτά αποτελούν και τα διανύσματα κυματομορφής (pattern vectors) του αλγορίθμου.

4.1.3 Προσδιορισμός των Περιγραμμάτων Ανώτερης Τάξης

Μια παρόμοια διαδικασία τομής υποχώρων, μπορεί να εφαρμοστεί για τον προσδιορισμό του περιγράμματος του σχήματος κυματομορφής της οποιασδήποτε χρονικής $t = i$.

Θεωρούμε όλους τους υποχώρους οι οποίοι λαμβάνουν μέρος στο σχηματισμό του και πάρουμε την τομή τους. Οι εν λόγω υποχώροι, δεν είναι άλλοι από τις k μετατοπίσεις του περιγράμματος του $(i - 1)$ -pattern, μία για το κάθε διάνυσμα κυματομορφής. Αυτή η διαδικασία σχηματισμού παρουσιάζεται στο σχήμα 4.3 και περιγράφεται λεπτομερώς από τον αλγόριθμο που ακολουθεί. Βασίζεται στον αλγόριθμο του minimum cover και υπολογίζει το περίγραμμα του σχήματος κυματομορφής της χρονικής στιγμής i .

Αλγόριθμος 4.3

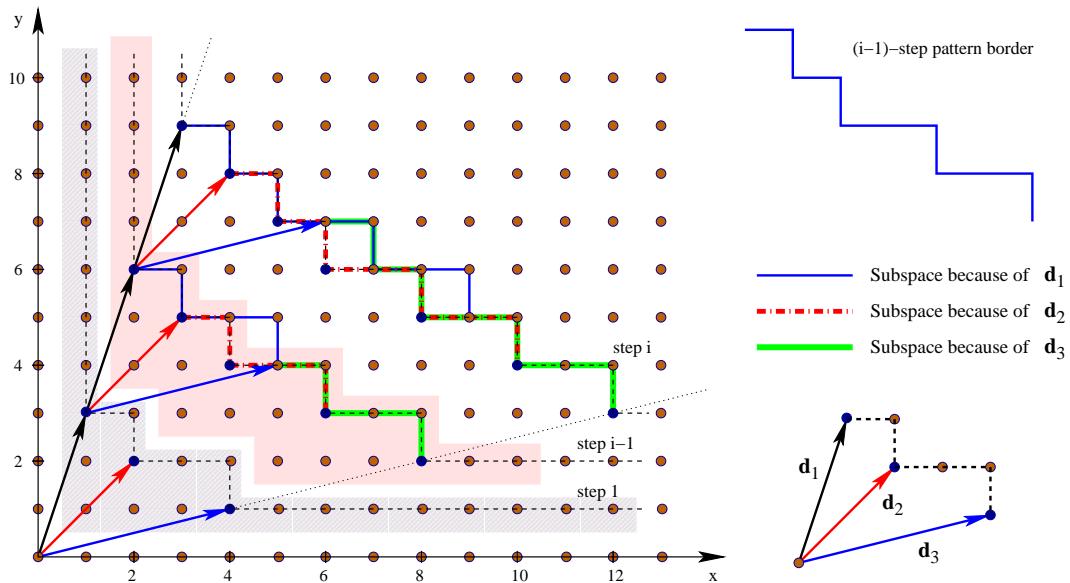
Σχηματισμός του i περιγράμματος pat_i

INPUT: Η ακολουθία των εξαρτήσεων $DV = \langle \mathbf{d}_1, \dots, \mathbf{d}_m \rangle$ και ο αριθμός του βήματος $i \in \mathbb{N}$, $i > 0$.

OUTPUT: Η ακολουθία των περιγραμμάτων όλων των patterns, μέχρι το i .

- | | |
|---|--|
| 1. $pat_1 = PV = \mathcal{MC}(DV)$ | <i>–υπολογίζονται τα k pattern vectors</i> |
| 2. αν $i = 1 \Rightarrow$ επιστρέφεται το pat_1 | |
| 3. αν $i \neq 1 \Rightarrow pat_i = \mathcal{MC}(pat_{i-1} + PV)$ | <i>–παρατηρείστε ότι $pat_i \leq m \cdot pat_{i-1}$</i> |

Παρά το γεγονός ότι ο αλγόριθμος δουλεύει αναδρομικά, είναι ο πρώτος που παρουσιάζεται στη βιβλιογραφία για τον προσδιορισμό των περιγραμμάτων των κυματομορφών εκτέλεσης,



Σχήμα 4.3: Σχηματισμός του περιγράμματος του *pattern* του βήματος i , για αλγόριθμο με 3 *pattern vectors*: η τομή όλων των περιγραμμάτων που προκύπτουν από τις μετακινήσεις του περιγράμματος του προηγούμενου βήματος, κατά όλα τα *pattern vectors*. Παρουσιάζεται και το αρχικό *pattern*.

αλγορίθμων φωλιασμένων βρόχων. Προφανώς, λόγω της αναδρομής, υπολογίζονται και τα περιγράμματα όλων των προηγούμενων χρονικών στιγμών $1, 2, \dots, i - 1$.

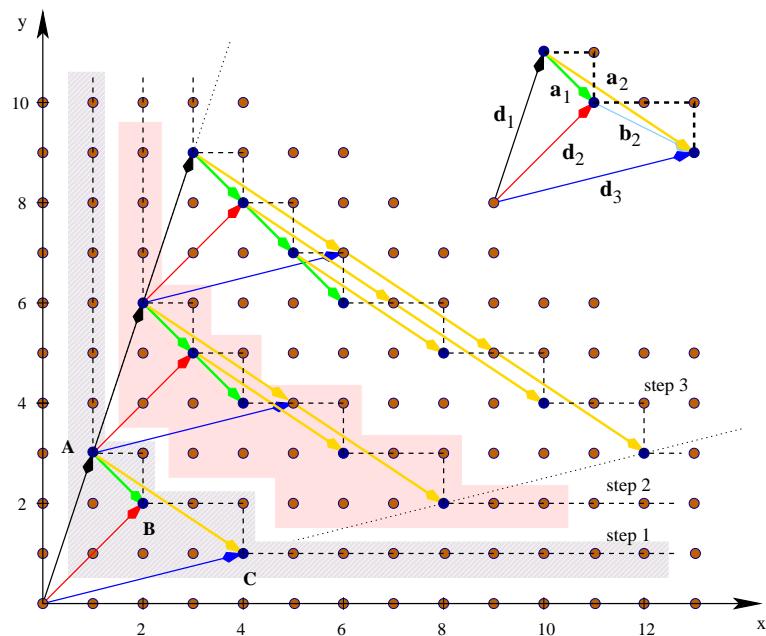
Σημειώνουμε επίσης, ότι ορισμένα σημεία του περιγράμματος i , υπολογίζονται παραπόνω από μία φορά (από διαφορετικά σημεία του προηγούμενου περιγράμματος). Το γεγονός αυτό ομολογεί τη μη-αποδοτικότητα της μεθόδου, παρότι είναι το καλύτερο που μπορούμε να επιτύχουμε διατηρώντας τη γενικότητα της.

Σε ορισμένες ειδικές περιπτώσεις αλγορίθμων, τα περιγράμματα των κυματομορφών εκτέλεσης είναι δυνατόν να υπολογιστούν σε πολυωνυμικό χρόνο και απευθείας, αποφεύγοντας την αναδρομή. Όπως θα δούμε στη συνέχεια, αυτό συμβαίνει όταν τα διανύσματα εξάρτησης πληρούν ορισμένες συνθήκες. Κάποιες από τις ειδικές αυτές περιπτώσεις αναφέρονται στην εργασία [DAA⁺01]: στο πλαίσιο της παρούσας διατριβής θα αναφερθούμε σε ορισμένες απ' αυτές σε επόμενη ενότητα.

4.2 Αναλυτική Περιγραφή των Περιγραμμάτων

Η περιγραφή των σχημάτων κυματομορφής και των περιγραμμάτων τους, όπως δόθηκε στην προηγούμενη παράγραφο, εισηγείται την προφανή εξαντλητική μέθοδο εύρεσης τους: αναπτύσσοντας όλους τους συνδυασμούς των διανυσμάτων κυματομορφής (pattern vectors), από την αρχή των αξόνων.

Με άλλα λόγια, τα διανύσματα κυματομορφής είναι αυτά τα οποία παράγουν το εσωτερικότερο περίγραμμα των έτοιμων προς εκτέλεση σημείων. Γι' αυτό και οριοθετούν τις χρονικές περιοχές και περικλείουν και το σύνολο του χώρου.



Σχήμα 4.4: Τα περιγράμματα των 3 πρώτων patterns, σχηματίζομενα από τις διαφορικές εξαρτήσεις a_i , $i = 1, 2, \dots, m - 1$ και $d_{xmin} \equiv d_1$.

Το περίγραμμα του σχήματος κυματομορφής του βήματος i , σχηματίζεται ακολουθώντας τα pattern vectors που ξεκινούν από το περίγραμμα του $(i - 1)$ -pattern και κρατώντας τα εσωτερικότερα από τα σημεία που προκύπτουν, όπως φαίνεται στο σχήμα 4.3. Με τη μορφή

σχέσεως, είναι:

$$pat_i = \begin{cases} \mathcal{MC}(DV) = \langle \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k \rangle = PV, & \text{για } i = 1 \\ \mathcal{MC}(pat_{i-1} + PV), & \text{για } i > 1 \end{cases} \quad (4.6)$$

όπου ο τελεστής $\mathcal{MC}(.)$ εφαρμόζει ένα εκ των Αλγορίθμων 4.1, 4.2 στο προκύπτον σύνολο και εξασφαλίζει την εσωτερικότητα στην οποία αναφερθήκαμε παραπάνω.

Από τις εκφράσεις των διαφορικών εξαρτήσεων όμως, μπορούμε να εκφράσουμε τα διανύσματα εξάρτησης και τα διανύσματα κυματομορφής, ως συνάρτηση των \mathbf{a}_i :

$$\mathbf{d}_i = \mathbf{d}_{xmin} + \mathbf{a}_{i-1}, \text{ για } 2 \leq i \leq m \quad (4.7)$$

$$\mathbf{p}_i = \begin{cases} \mathbf{d}_{xmin}, & \text{για } i = 1 \\ \mathbf{d}_{xmin} + \mathbf{a}_{i-1}, & \text{για } 2 \leq i \leq k \end{cases} \quad (4.8)$$

όπως εξηγείται στο σχήμα 4.4.

Με τη βοήθεια αυτών των εξισώσεων, μπορούμε να εκφράσουμε όλα τα σημεία που ανήκουν στο περίγραμμα pat_1 , ως συνάρτηση των διαφορικών εξαρτήσεων:

$$\mathbf{p}_j = \mathbf{d}_{xmin} + k_1 \cdot \mathbf{a}_1 + \dots + k_{m-1} \cdot \mathbf{a}_{m-1}, \quad (4.9)$$

όπου:

$$k_i \in \mathbb{N} \text{ και } k_1 + k_2 + \dots + k_{m-1} \leq 1, \quad 1 \leq i \leq m, \text{ για } 1 \leq j \leq k.$$

Έτσι, το περίγραμμα του αρχικού (για τη χρονική στιγμή $t = 1$) σχήματος κυματομορφής pat_1 περιέχει τα σημεία $\mathbf{j} = (j_x, j_y)$, τα οποία προκύπτουν από τις σχέσεις:

$$\begin{aligned} &\triangleright d_{xmin_x} + k_1 \cdot a_{1_x} + \dots + k_{m-1} \cdot a_{m-1_x} \leq j_x \\ &\triangleright d_{xmin_y} + k_1 \cdot a_{1_y} + \dots + k_{m-1} \cdot a_{m-1_y} = j_y \\ &\triangleright j_y = \min_{j_x} (για \text{ ένα } j_x \text{ μπορεί να έχουμε πολλαπλά } j_y) \\ &\triangleright k_1 + k_2 + \dots + k_{m-1} \leq 1 = t. \end{aligned} \quad (4.10)$$

Από την σχέση 4.6, έχουμε ότι το περίγραμμα οποιασδήποτε χρονικής στιγμής προκύπτει από το προηγούμενό του. Μπορούμε λοιπόν, να γενικεύσουμε τη σχέση 4.10, ώστε να δίνονται τα σημεία του περιγράμματος για οποιαδήποτε χρονική στιγμή. Το περίγραμμα του σχήματος κυματομορφής pat_t , το οποίο αντιστοιχεί στη χρονική στιγμή t , περιέχει όλα τα σημεία $\mathbf{j} =$

(j_x, j_y) , ούτως ώστε:

$$\begin{aligned} & \triangleright t \cdot d_{xmin_x} + k_1 \cdot a_{1_x} + \dots + k_{m-1} \cdot a_{m-1_x} \leq j_x \\ & \triangleright t \cdot d_{xmin_y} + k_1 \cdot a_{1_y} + \dots + k_{m-1} \cdot a_{m-1_y} = j_y = minimum \\ & \triangleright k_1 + k_2 + \dots + k_{m-1} \leq t, \text{ όπου } k_i \in \mathbb{N} \text{ και } 1 \leq i \leq m-1. \end{aligned} \quad (4.11)$$

Οι παραπάνω εκφράσεις είναι οι πρώτες που περιγράφουν αναλυτικά το περίγραμμα εκτέλεσης φωλιασμένου βρόχου, σε οποιαδήποτε χρονική στιγμή t . Με τη βοήθεια τους ορίζονται αναλυτικά τα δύο σχετικά προβλήματα της βιβλιογραφίας.

4.3 Αναλυτική Περιγραφή Προβλημάτων

Στην προηγούμενη ενότητα παρουσιάσαμε τις εξισώσεις που δίνουν τα περιγράμματα των σχημάτων κυματομορφής εκτέλεσης (patterns). Με βάση αυτές, μπορούμε να περιγράψουμε τα δύο σχετικά προβλήματα της βιβλιογραφίας αναλυτικά:

Πρόβλημα Εύρεσης Κυματομορφής Εκτέλεσης Δεδομένης της χρονικής στιγμής t , επιθυμούμε να βρούμε όλα τα σημεία του χώρου δεικτών τα οποία σχηματίζουν το περίγραμμα pat_t κατά την εν λόγω χρονική στιγμή t . Τα σημεία $\mathbf{j} = (j_x, j_y) \in pat_t$ δίνονται από τις σχέσεις:

$$\begin{aligned} & \triangleright t \cdot d_{xmin_x} + k_1 \cdot a_{1_x} + \dots + k_{m-1} \cdot a_{m-1_x} \leq j_x \\ & \triangleright t \cdot d_{xmin_y} + k_1 \cdot a_{1_y} + \dots + k_{m-1} \cdot a_{m-1_y} = j_y = minimum \\ & \triangleright k_1 + \dots + k_{m-1} \leq t, \text{ όπου } k_i \in \mathbb{N} \text{ και } 1 \leq i \leq m-1 \end{aligned} \quad (4.12)$$

Πρόβλημα Δρομολόγησης Δεδομένου οποιουδήποτε σημείου $\mathbf{j} = (j_x, j_y)$ του χώρου δεικτών, επιθυμούμε να βρούμε τη χρονική στιγμή t , κατά την οποία είναι δυνατόν να εκτελεστεί. Η χρονική στιγμή t προσδιορίζεται με βάση τους περιορισμούς:

$$\begin{aligned} & \triangleright t \cdot d_{xmin_x} + k_1 \cdot a_{1_x} + \dots + k_{m-1} \cdot a_{m-1_x} \leq j_x \\ & \triangleright t \cdot d_{xmin_y} + k_1 \cdot a_{1_y} + \dots + k_{m-1} \cdot a_{m-1_y} \geq j_y \\ & \triangleright k_1 + k_2 + \dots + k_{m-1} \leq t, \text{ } k_i \in \mathbb{N}, \text{ } 1 \leq i \leq m-1, \text{ και} \\ & \triangleright t = minimum \end{aligned} \quad (4.13)$$

Από τις περιγραφές των δύο προβλημάτων, μπορεί κανείς να παρατηρήσει ότι επιλύονται αναλυτικά με μεθόδους ακέραιου γραμμικού προγραμματισμού. Παρότι δεν επιδέχονται αποδοτική επίλυση, είναι η πρώτη φορά που σχηματίζονται σε κλειστή μορφή.

4.4 Συνθήκες Ομοιομορφίας των Σχημάτων Κυματομορφής

Όσον αφορά τη μορφή των σχημάτων κυματομορφής εκτέλεσης (patterns), οι σχέσεις 4.12 δίνουν την αναλυτική περιγραφή τους στη γενική περίπτωση. Υπό ειδικές συνθήκες, η παραπάνω έκφραση απλοποιείται σε μεγάλο βαθμό.

Στις περιπτώσεις αυτές το περίγραμμα της κυματομορφής είναι ομοιόμορφο ή/και επαναλαμβανόμενο. Η κατασκευή του περιγράμματος- i είναι προφανής, όπως στην περίπτωση του σχήματος 4.5. Για την κατασκευή του περιγράμματος της χρονικής στιγμής $t = i$, επαναλαμβάνεται το αρχικό περίγραμμα i φορές.

Θα περιγράψουμε μία συνθήκη η οποία καθιστά την κυματομορφή εκτέλεσης πλήρως ομοιόμορφη. Η συνθήκη αφορά τα διανύσματα εξάρτησης του φωλιασμένου βρόχου. Στην περίπτωση που η συνθήκη αυτή ισχύει, το περίγραμμα της χρονικής στιγμής i pat_i , είναι δυνατόν να υπολογιστεί –απλά– παίρνοντας το αρχικό περίγραμμα i φορές.

Λήμμα 4.1 Θεωρούμε φωλιασμένο βρόχο με δύο διανύσματα εξάρτησης \mathbf{a}'_1 και \mathbf{a}'_2 και τις κλίσεις τους² g_1 και g_2 , αντίστοιχα. Αν $g_1 \geq g_2$ και $a'_{1y} \geq a'_{2y}$, τότε τα σημεία τα οποία πιθανώς συνεισφέρουν στον σχηματισμό του περιγράμματος είναι (όπως στο σχήμα 4.6):

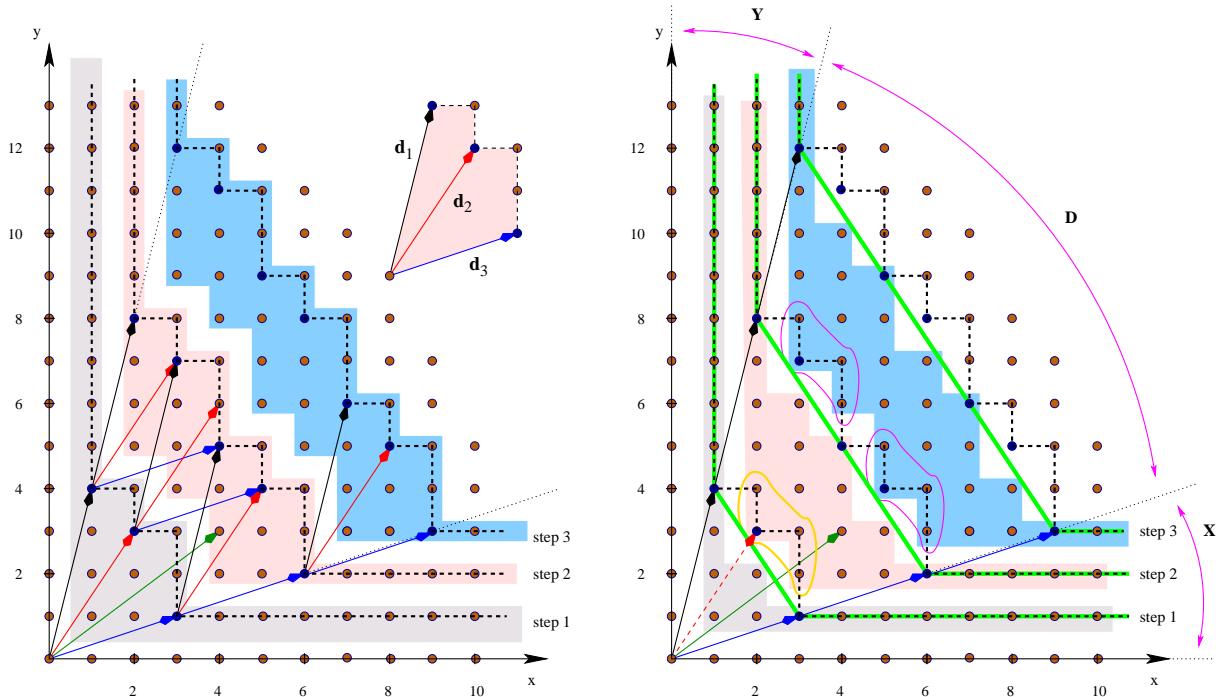
- το τερματικό σημείο του διανύσματος $i \cdot \mathbf{a}'_1$, όπου $0 \leq i \leq n$,
- τα τερματικά σημεία των διανυσμάτων $i \cdot \mathbf{a}'_2$, τα οποία ανήκουν στα τρίγωνα που σχηματίζονται από το \mathbf{a}'_1 .

Απόδειξη³

Από τα διαφορικά διανύσματα εξάρτησης \mathbf{a}'_1 , \mathbf{a}'_2 το \mathbf{a}'_1 είναι αυτό με τη μεγαλύτερη κλίση. Λόγω αυτού, τα τερματικά σημεία των διανυσμάτων $i \cdot \mathbf{a}'_1$, πάντα ύπα αποτελούν σημεία του αντίστοιχου περιγράμματος i (είναι τα σημεία με τη μικρότερη συντεταγμένη x). Όλα τα υπόλοιπα πιθανά

²Σύμφωνα με το σύστημα συντεταγμένων των σχημάτων 4.6 και 4.7.

³Βασίζεται στις σχέσεις της παραγράφου 4.2.



Σχήμα 4.5: Αλγόριθμος με 4 διανύσματα εξάρτησης, $\mathbf{d}_1 = (1, 4)$, $\mathbf{d}_2 = (2, 3)$, $\mathbf{d}_3 = (3, 1)$ και $\mathbf{d}_4 = (4, 3)$, το οποίο δεν είναι διάνυσμα κυματομορφής. Το περίγραμμα του pattern είναι ομοιόμορφο: το περίγραμμα της χρονικής στιγμής i , υπολογίζεται παίρνοντας το αρχικό περίγραμμα i φορές.

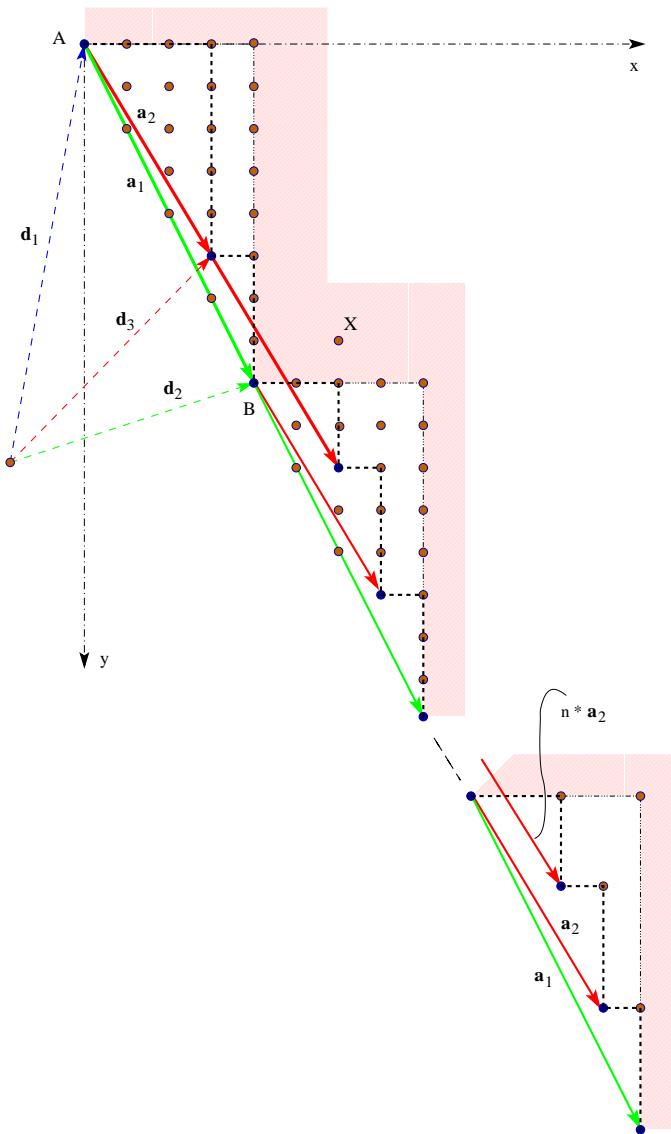
σημεία, δίνονται από την έκφραση:

$$k_1 \cdot \mathbf{a}'_1 + k_2 \cdot \mathbf{a}'_2, \text{ όπου } 0 \leq k_1, k_2 \leq i \text{ και } k_1 + k_2 = t,$$

για την χρονική στιγμή t . Επίσης, αν ένα σημείο X ανήκει στην γραμμοσκιασμένη περιοχή, δεν επηρεάζει το περίγραμμα, αφού υπάρχει ένα δεύτερο σημείο με μικρότερες συντεταγμένες x και y , το οποίο εξαφανίζει την επιρροή του (σημείο B στο σχήμα). \dashv

Λήμμα 4.2 Θεωρούμε τα $m - 1$ διαφορικά διανύσματα εξαρτήσεων \mathbf{a}_i και τις κλίσεις τους g_i , όπου $g_1 \geq g_i$, $2 \leq i \leq m - 1$. Χωρίς βλάβη της γενικότητας θεωρούμε $a_{1y} \geq a_{iy}$. Στην περίπτωση αυτή, το προηγούμενο Λήμμα επαληθεύεται για $\mathbf{a}'_1 = \mathbf{a}_1$ και \mathbf{a}'_2 οποιονδήποτε γραμμικό συνδυασμό της μορφής:

$$k_2 \cdot \mathbf{a}_2 + k_3 \cdot \mathbf{a}_3 + \dots + k_{m-1} \cdot \mathbf{a}_{m-1}, \text{ όπου } 0 \leq k_i \leq m - 1 \text{ και } k_2 + k_3 + \dots + k_{m-1} = n.$$



Σχήμα 4.6: Το περίγραμμα του κάθε σχήματος κυματομορφής εξαρτάται από τις κλίσεις των διαφορικών διανυσμάτων εξάρτησης. Αποτελείται από ξεχωριστά τμήματα, τα οποία περικλείονται σε τρίγωνα που σχηματίζονται από τα \mathbf{d}_{xmin} και \mathbf{d}_{ymin} . Τα ακριβή σημεία του περιγράμματος καθορίζονται από ένα γραμμικό συνδυασμό των διαφορικών διανυσμάτων εξάρτησης.

Απόδειξη

Προφανής από το προηγούμενο Λήμμα και τον ορισμό των διανυσμάτων \mathbf{a}_i . +

Θεώρημα 4.1 Θεωρούμε ένα φωλιασμένο βρόχο με k pattern vectors $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k$. Έστω $\mathbf{a}_i = \mathbf{p}_i - \mathbf{p}_{i+1}$ τα $k-1$ διαφορικά διανύσματα εξάρτησης, με τιμές κλίσεων g_i . Άν, για όλα τα i , $2 \leq i \leq k-1$, ισχύουν οι σχέσεις:

- $g_1 \geq 2 \cdot g_i$
 - $a_{1y} \geq a_{iy}$
 - $a_{ix} \geq \frac{a_{1x}}{2}$,
- (4.14)

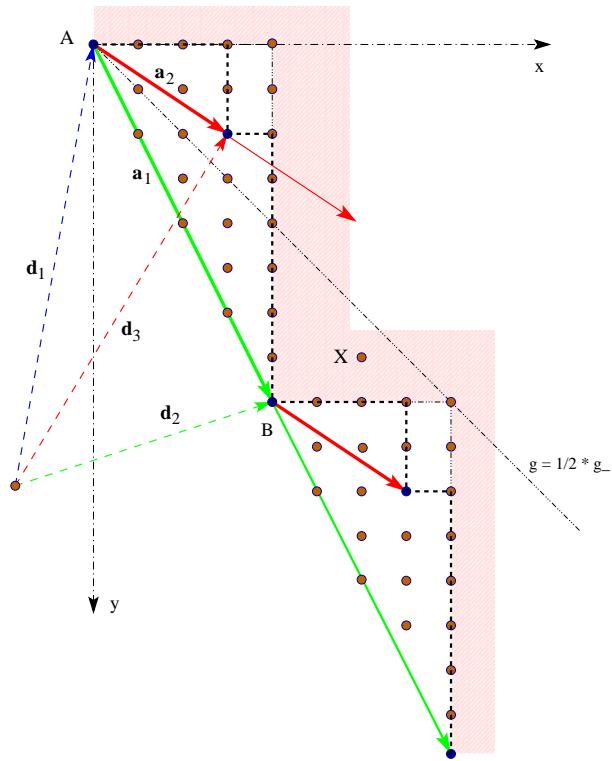
τότε, το περίγραμμα του pattern οποιασδήποτε χρονικής στιγμής, είναι επαναλαμβανόμενο και μπορεί να προσδιοριστεί επακριβώς επαναλαμβάνοντας το αρχικό του τμήμα. Αποτελείται δηλαδή, από επαναλήψεις του ίδιου “υποπεριγράμματος”. Σημειώνουμε ότι το υποπεριγραμμα αυτό δεν είναι άλλο από το αρχικό pattern, αυτό της πρώτης χρονικής στιγμής.

Απόδειξη

Κάθε γραμμικός συνδυασμός των διαφορικών διανυσμάτων εξάρτησης \mathbf{a}_i , καταλήγει στη γραμμοσκιασμένη περιοχή, λόγω της συνολικής κλίσης g_{sum} . Είναι γι' αυτήν: $2 \cdot g_{sum} \leq g_1$ και $a_{sum_x} \geq a_{1x}$, λόγω της τρίτης σχέσης. Έτσι, για το περίγραμμα του pattern συμβάλλουν μόνο τα διανύσματα \mathbf{a}_i . Κάθε διάνυσμα της μορφής $k_i \cdot \mathbf{a}_i$, λόγων των τριών παραπάνω σχέσεων, σχηματίζει ένα μόνον σημείο σε κάθε τρίγωνο, στην ίδια (σχετικά με την αρχή του κάθε τριγώνου) θέση. Σύμφωνα με τα παραπάνω και λόγω των προηγουμένων λημμάτων, το περίγραμμα του οποιουδήποτε pattern είναι επαναλαμβανόμενο και αποτελείται από επαναλήψεις του αρχικού περιγράμματος. +

Ανάλογες ειδικές περιπτώσεις ομοιομορφίας των περιγραμμάτων μπορούν να αποδειχτούν, βασιζόμενοι σε αντίστοιχες σχέσεις. Στο σχήμα 4.8 δίνεται μια ακόμη περίπτωση ομοιομορφίας. Το αρχικό περίγραμμα χωρίζεται σε πεπερασμένο αριθμό (διαφορετικών) υποπεριγραμμάτων. Παίρνοντας το πρώτο υποπεριγραμμα i φορές, κατόπιν το δεύτερο i φορές, κ.οχ., σχηματίζεται το ακριβές περίγραμμα που αντιστοιχεί στην χρονική στιγμή $t = i$.

Δεδομένων των διαφορικών διανυσμάτων εξάρτησης \mathbf{a}_i και των αντίστοιχων κλίσεων g_i , $1 \leq i \leq m-1$, αποδεικνύεται ότι οι περιορισμοί που συνθέτουν αυτή την περίπτωση ομοιομορφίας είναι οι εξής:



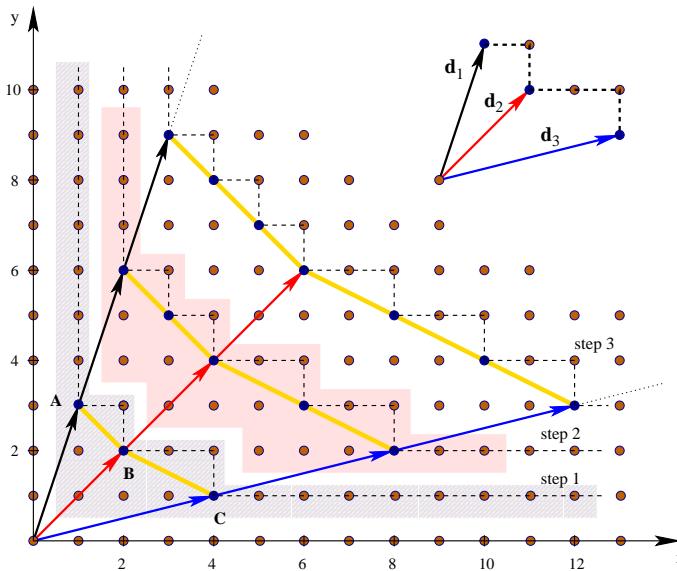
Σχήμα 4.7: Εξήγηση του Θεωρήματος 4.1· θεωρούμε το διαφορικό διάνυσμα εξάρτησης $\mathbf{a}_{g_{max}} \equiv \mathbf{a}_1$, ως αυτό με τη μεγαλύτερη τιμή κλίσης g_{max} . Αν οι κλίσεις όλων των υπολοίπων διαφορικών διανυσμάτων είναι μικρότερες από το μισό της g_{max} , τότε το περίγραμμα κάθε περιγράμματος παράγεται από ένα επαναλαμβανόμενο “υποπερίγραμμα”.

- $g_1 > g_2 > \dots > g_{m-1}$ και
- $a_{1y} < a_{2y} < \dots < a_{(m-1)y}$.

Η ομοιομορφία που μόλις περιγράψαμε ισχύει και στην περίπτωση όπου όλα τα διανύσματα εξάρτησης αποτελούν διανύσματα κώνων, όπως αυτά ορίζονται στην παράγραφο 5.2.

4.5 Πολυπλοκότητα Τυπολογισμού των Περιγραμμάτων

Στην παράγραφο αυτή θα μελετήσουμε την πολυπλοκότητα υπολογισμού των περιγραμμάτων, σύμφωνα με τις μεθόδους που περιγράψαμε. Πιο συγκεκριμένα, θα υπολογίσουμε το χρονικό



Σχήμα 4.8: Μία δεύτερη συνθήκη ομοιόμορφου περιγράμματος: το περίγραμμα της χρονικής στιγμής i σχηματίζεται παίρνοντας τα επιμέρους τμήματα του αρχικού περιγράμματος i φορές.

κόστος που απαιτείται για τον προσδιορισμό των διανυσμάτων κυματομορφής (pattern vectors) –Αλγόριθμοι 4.1, 4.2– και για τον προσδιορισμό του περιγράμματος κυματομορφής (pattern outline) της τυχαίας χρονικής στιγμής t –Αλγόριθμος 4.3.

Η είσοδος του Αλγορίθμου 4.1 είναι μια ακολουθία m σημείων. Για να προχωρήσουμε στη σάρωση των σημείων, τα ταξινομούμε λεξικογραφικά, ως προς τις συντεταγμένες τους x και y . Προσομοιώνουμε έτσι, την χρυμμένη ταξινόμηση που υπονοεί ο αλγόριθμος. Η ταξινόμηση αυτή επιτυγχάνεται αυτόματα σαρώνοντας τον χώρο προς τη θετική φορά του άξονα των x .

Συνολικά λοιπόν, έχουμε μια ταξινόμηση των m σημείων και μια επιπλέον σάρωση, κατά αύξουσα σειρά της συντεταγμένης x . Έτσι, η πολυπλοκότητα $Cx(4.1)$ προσδιορισμού των διανυσμάτων κυματομορφής στο επίπεδο είναι:

$$Cx(4.1) = \Theta(m \log m) + \Theta(m) = \Theta(m \log m). \quad (4.15)$$

Στον Αλγόριθμο 4.2 δεν έχουμε ταξινόμηση. Όμως, η συνθήκη εισαγωγής του κάθε σημείου στο minimum cover, απαιτεί τον έλεγχό του με όλα τα ήδη υπάρχοντα σημεία σ' αυτό. Αναλογιζόμενοι ότι ο κάθε έλεγχος, μεταξύ δύο σημείων απαιτεί n συγκρίσεις –αναφερόμαστε στις n διαστάσεις– η πολυπλοκότητα του προσδιορισμού των διανυσμάτων κυματομορφής στις n

διαστάσεις είναι:

$$Cx(4.2) = \Theta(m^2 \cdot n) = Cx(\mathcal{MC}_{m,n}). \quad (4.16)$$

Η πολυπλοκότητα αυτή αναφέρεται και ως η πολυπλοκότητα υπολογισμού του minimum cover συνόλου m σημείων στο n -διάστατο χώρο $Cx(\mathcal{MC}_{m,n})$.

Για τον Αλγόριθμο 4.3, η πολυπλοκότητα ως υπολογιστεί αναδρομικά, δεδομένου ότι για τον υπολογισμό του περιγράμματος t , πρέπει να έχει υπολογιστεί και το περίγραμμα $t - 1$, κ.ο.κ. Έστω $Cx(pat_t)$ η πολυπλοκότητα υπολογισμού του περιγράμματος της χρονικής στιγμής t . Για την αρχική χρονική στιγμή $t = 1$ είναι:

$$Cx(pat_1) = \Theta(m^2 \cdot n)$$

δεδομένου ότι εφαρμόζεται ένας εκ των Αλγορίθμων 4.1, 4.2 στα διανύσματα εξάρτησης.

Για το περίγραμμα pat_t , της τυχαίας χρονικής στιγμής t , είναι $|pat_t| = m^t$, στη χειρότερη περίπτωση. Αναζητούμε την πολυπλοκότητα υπολογισμού του περιγράμματος $Cx(pat_{t+1})$, για τη χρονική στιγμή $t + 1$. Τα πιθανά σημεία του pat_{t+1} είναι τα $m^t \cdot m$ που προκύπτουν αν στο κάθε σημείο του pat_t , προσθέσουμε όλα τα pattern vectors.⁴ Στα πιθανά αυτά σημεία εφαρμόζεται ο αλγόριθμος του minimum cover:

$$\begin{aligned} Cx(pat_{t+1}) &= Cx(pat_t) + Cx(\mathcal{MC}_{|pat_t| \cdot m, n}) \\ &= Cx(pat_t) + Cx(\mathcal{MC}_{m^t \cdot m, n}) \\ &= Cx(pat_t) + Cx(\mathcal{MC}_{m^{t+1}, n}) \\ &= Cx(pat_t) + \Theta(m^{2(t+1)} \cdot n) \end{aligned} \quad (4.17)$$

Η αναδρομική αυτή έκφραση μπορεί να μετατραπεί σε μη αναδρομική, εκφράζοντας τις πολυπλοκότητες για όλες τις χρονικές στιγμές από 1 έως t και προσθέτοντας τις σχέσεις. Προκύπτει έτσι, ότι η πολυπλοκότητα υπολογισμού του περιγράμματος της t χρονικής στιγμής, n -διάστατου φωλιασμένου βρόχου, με m διανύσματα εξάρτησης, δίνεται από την έκφραση:

$$Cx(pat_t) = \Theta(m^{2t} \cdot n). \quad (4.18)$$

Για την ειδική περίπτωση των προβλημάτων δύο διαστάσεων, η παραπάνω έκφραση της πολυπλοκότητας απλοποιείται σύμφωνα με την έκφραση:

$$Cx(pat_t) = \Theta(m^t \cdot \log(m^t)). \quad (4.19)$$

⁴Τα σημεία του pat_t είναι το πολύ m^t και τα pattern vectors είναι $k \leq m$. Έτσι είναι $|pat_{t+1}| = m^t \cdot m = m^{t+1}$.

Τα παραπάνω αποτελέσματα αποδεικνύουν την εκθετική πολυπλοκότητα της μεθόδου εύρεσης των περιγραμμάτων που περιγράψαμε.

Κεφάλαιο 5

Γεωμετρική Δρομολόγηση στις Δύο Διαστάσεις

Στο κεφάλαιο αυτό προτείνεται μια τεχνική δρομολόγησης, η οποία βασίζεται στις γεωμετρικές ιδιότητες του χώρου δεικτών. Η μέθοδος παρουσιάζει ασυμπτωτικά βέλτιστο χρόνο εκτέλεσης και πολυωνυμική πολυπλοκότητα, γεγονός που την καθιστά αποδοτική, σε σύγκριση με τις αντίστοιχες μεθόδους της βιβλιογραφίας, οι οποίες πραγματοποιούν μετασχηματισμό του χώρου δεικτών, χρησιμοποιώντας μεθόδους γραμμικής άλγεβρας [MF86, ST91, DR94, TKP00].

Παρότι η εν λόγω μέθοδος αναφέρεται σε φωλιασμένους βρόχους οποιασδήποτε διάστασης, στο κεφάλαιο αυτό αναφερόμαστε σε βρόχους διάστασης δύο. Αυτό γίνεται, τόσο για την καλύτερη κατανόηση της μεθόδου, όσο και για την ιδιαίτερη παρουσίαση ορισμένων χαρακτηριστικών της. Τα χαρακτηριστικά αυτά είναι ιδιαίτερα εμφανή στις δύο διαστάσεις.

5.1 Γενική Περιγραφή της Μεθόδου

Η βασική ιδέα της μεθόδου είναι η γεωμετρική διαμέριση του χώρου δεικτών σε μια ακολουθία περιοχών (*regions*) $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \dots$ και η παράλληλη εκτέλεση των σημείων κάθε περιοχής. Επιθυμητή επιλογή περιοχών είναι η προφανής $\mathcal{R}_i = ECT_i$, για $i \geq 1$. Αυτή η επιλογή όμως, δεν είναι ιδιαίτερα ελκυστική, για έναν κυρίως λόγο:

- τα γεωμετρικά όρια των χρονικών περιοχών ECT_i δεν είναι προσδιορίσημα με άμεσο και αποδοτικό τρόπο· το γεγονός αυτό καθιστά δύσκολο τον καθορισμό των σημείων κάθε

περιοχής και άρα την άμεση δρομολόγησή τους.

Από την άλλη πλευρά, για να εξασφαλίζεται η ορθότητα της εκτέλεσης, επιβάλλεται η μη διαταραχή των εξαρτήσεων. Για τις οριζόμενες περιοχές δηλαδή, πρέπει να ισχύουν εξισώσεις της μορφής:

$$\begin{aligned} \mathcal{R}_1 &\subseteq ECT_1 \\ \mathcal{R}_2 &\subseteq ECT_2 \cup ECT_1 \\ &\vdots \\ \mathcal{R}_i &\subseteq \bigcup_{t=1}^i ECT_t. \end{aligned} \tag{5.1}$$

Όπως θα αποδειχτεί στη συνέχεια,¹ η τεχνική δρομολόγησης αυτή είναι ασυμπτωτικά βέλτιστη, δεδομένου ότι ο χρόνος εκτέλεσης που ανατίθεται σε κάθε σημείο είναι το πολύ ένα μεγαλύτερος από τον νωρίτερο χρόνο εκτέλεσής του. Στην πραγματικότητα δηλαδή, η σχέση 5.1 είναι:

$$\mathcal{R}_i \subseteq ECT_i \cup ECT_{i-1}. \tag{5.2}$$

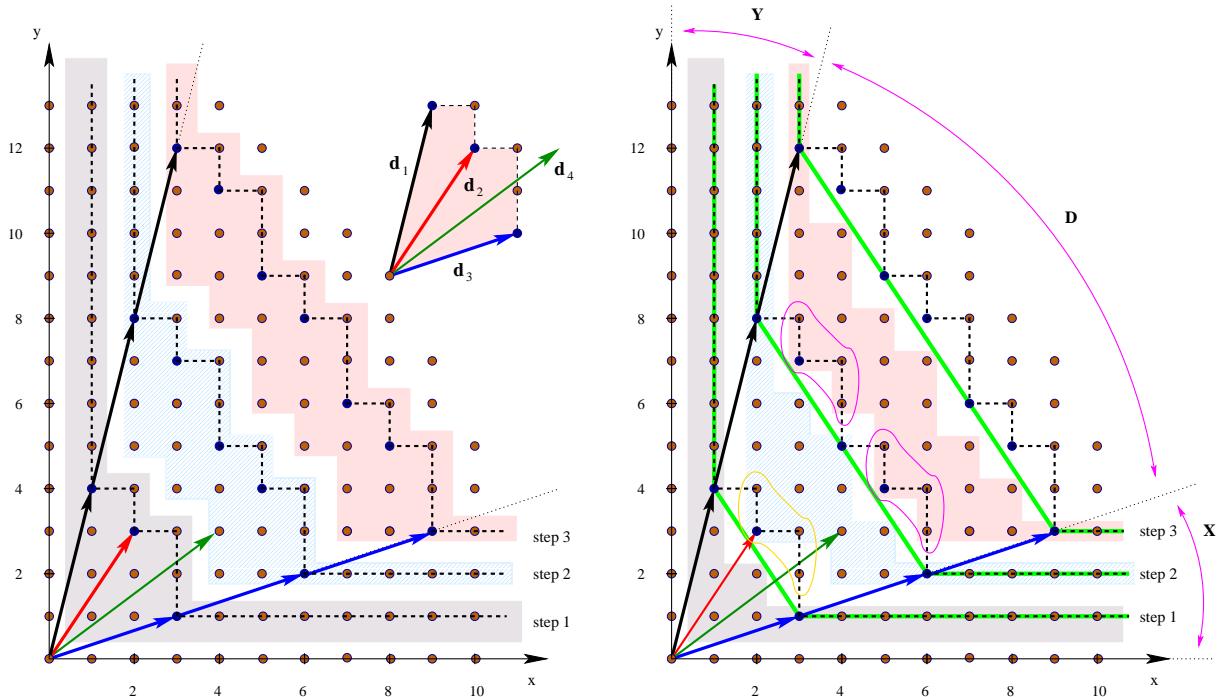
Από την έκφραση αυτή, εξηγείται και η σχέση που συνδέει τα \mathcal{R}_i και τα ECT_i ή Pat_i , τα οποία αναλύσαμε στο προηγούμενο κεφάλαιο.

Στο σχήμα 5.1 παρουσιάζεται ο ακριβής υπολογισμός των συνόλων ECT_i , ακολουθώντας τη μέθοδο που παρουσιάστηκε στο προηγούμενο κεφάλαιο (αριστερά). παρουσιάζεται επίσης, ο προσεγγιστικός υπολογισμός των περιοχών \mathcal{R}_i (δεξιά). Η βασική διαφορά των δύο μεθόδων έγκειται στα εξής:

- η μεν πρώτη επιτυγχάνει τον ακριβή υπολογισμό των συνόλων ECT_i : δεν επιτρέπει όμως τον προσδιορισμό του συνόλου ECT_j στο οποίο ανήκει το τυχαίο σημείο p_j , με αποδοτικό τρόπο, λόγω της πολύπλοκης (γεωμετρικής) μορφής των ECT_i .
- η δεύτερη, παρά το γεγονός ότι $\mathcal{R}_i \subseteq ECT_i \cup ECT_{i-1}$, επιτρέπει τον εύκολο προσδιορισμό της περιοχής που ανήκει το οποιοδήποτε σημείο, λόγω του τραπεζοειδούς (γεωμετρικού) σχήματος των περιοχών.

Η νέα αυτή μέθοδος δρομολόγησης δεν λειτουργεί αναδρομικά, γεγονός που μειώνει δραματικά την πολυπλοκότητα της.

¹Ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει και στο [DAP00].



Σχήμα 5.1: Ο σχηματισμός των συνόλων ECT_i και \mathcal{R}_i για έναν αλγόριθμο με 4 διανύσματα εξάρτησης. Με την εξαντλητική μέθοδο του προηγούμενου κεφαλαίου λαμβάνονται τα ακριβή σύνολα ECT_i (αριστερά). Με τη μέθοδο που θα παρουσιαστεί στη συνέχεια, ο χώρος δεικτών χωρίζεται σε μια ακολουθία συνόλων, με τραπεζοειδή περιγράμματα, τα οποία ονομάζονται περιοχές \mathcal{R}_i (δεξιά).

5.2 Ορολογία και Ορισμοί

Η μέθοδος μας βασίζεται στις ιδιότητες των διανυσμάτων εξάρτησης και του χώρου δεικτών. Θεωρούμε αυστηρώς θετικά διανύσματα εξάρτησης, χωρίς περιορισμό της γενικότητας. Δεδομένης της ακολουθίας DV των διανυσμάτων εξάρτησης $DV = \langle \mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m \rangle$, δίνουμε τους ακόλουθους ορισμούς:

Κώνος: Ο κυρτός υποχώρος ο οποίος σχηματίζεται από δύο διανύσματα \mathbf{a} και \mathbf{b} (στις 2 διαστάσεις) καλείται κώνος και συμβολίζεται ως $Con(\mathbf{a}, \mathbf{b})$. Τυπικότερα, ορίζεται ως:

$$Con(\mathbf{a}, \mathbf{b}) = \{ \mathbf{j} \in \mathbb{N}^2 \mid \mathbf{j} = \lambda_1 \cdot \mathbf{a} + \lambda_2 \cdot \mathbf{b}, \text{ όπου } \lambda_1, \lambda_2 \geq 0 \}.$$

Κώνος- Y : Ο κώνος ο οποίος σχηματίζεται από τον άξονα των y και το διάνυσμα εξάρτη-

σης με την μικρότερη συντεταγμένη x \mathbf{d}_{xmin} (βλέπε σχήμα 5.1). Συμβολίζεται και ως $ConY(\mathbf{d}_{xmin})$.

Κώνος-X: Ο κώνος ο οποίος σχηματίζεται από το διάνυσμα εξάρτησης με την μικρότερη συντεταγμένη y \mathbf{d}_{ymin} και τον άξονα των x . Συμβολίζεται και ως $ConX(\mathbf{d}_{ymin})$.

Κώνος-D: Ο κώνος $ConD$ ο οποίος σχηματίζεται από τα διανύσματα \mathbf{d}_{xmin} και \mathbf{d}_{ymin} :

$$ConD = Con(\mathbf{d}_{xmin}, \mathbf{d}_{ymin}).$$

Διανύσματα Κώνων (cone vectors) $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_r$ είναι αυτά τα διανύσματα εξάρτησης τα οποία έχουν την ακόλουθη ιδιότητα. Σχηματίζουν μια λεξικογραφικά ταξινομημένη ακολουθία $CV = \langle \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_r \rangle$, τέτοια ώστε:

- $\mathbf{d} = \lambda_1 \cdot \mathbf{c}_i + \lambda_2 \cdot \mathbf{c}_{i+1}$, όπου $\lambda_1 + \lambda_2 > 1$, για κάθε $\mathbf{d} \in DV$, $\mathbf{d} \neq \mathbf{c}_i, \mathbf{c}_{i+1}$ και για ακριβώς ένα ζεύγος $\langle \mathbf{c}_i, \mathbf{c}_{i+1} \rangle$.

Ο αριθμός των διανυσμάτων των κώνων συμβολίζεται ως r . Σημειώνουμε ότι τα διανύσματα εξάρτησης \mathbf{d}_{xmin} και \mathbf{d}_{ymin} πάντα συμπεριλαμβάνονται στην ακολουθία CV ($\mathbf{c}_1 = \mathbf{d}_{xmin}$ και $\mathbf{c}_r = \mathbf{d}_{ymin}$).

Περιοχή X_i είναι το σύνολο:

$$X_i = \{\mathbf{j} = (j_x, j_y) \in ConX \mid (i - 1) \cdot d_{xmin_y} < j_y < i \cdot d_{xmin_y}, \text{ όπου } i > 0\}.$$

Περιοχή Y_i είναι το σύνολο:

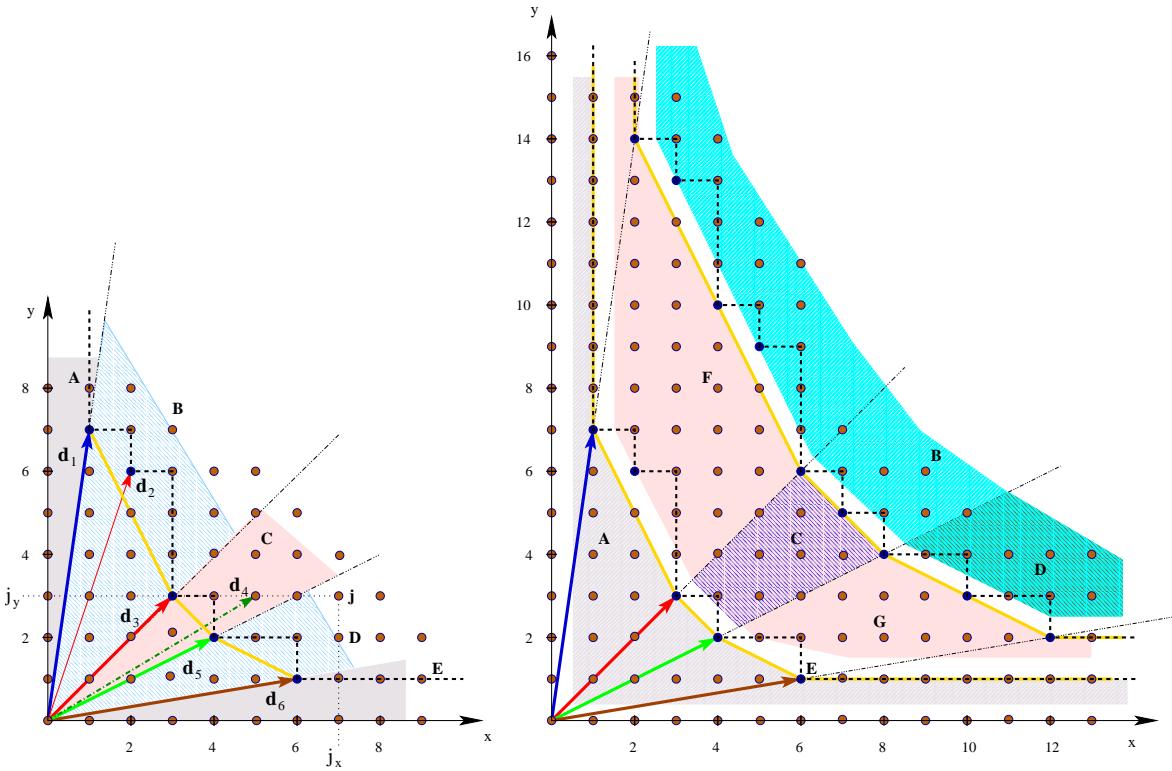
$$Y_i = \{\mathbf{j} = (j_x, j_y) \in ConY \mid (i - 1) \cdot d_{ymin_x} < j_x < i \cdot d_{ymin_x}, \text{ όπου } i > 0\}.$$

Περιοχή Κώνου (cone region): Ως i περιοχή ενός κώνου $Con(\mathbf{a}, \mathbf{b})$ είναι το σύνολο (βλ. σχήμα 5.2):

$$Con_i(\mathbf{a}, \mathbf{b}) = \{\mathbf{j} \in Con(\mathbf{a}, \mathbf{b}) \mid \mathbf{j} = \lambda_1 \cdot \mathbf{a} + \lambda_2 \cdot \mathbf{b}, \text{ όπου } i < \lambda_1 + \lambda_2 \leq i + 1\}.$$

Περιοχή (region): Ως περιοχή \mathcal{R}_i του χώρου δεικτών, ορίζεται η ένωση των περιοχών i όλων των σχηματιζόμενων κώνων:

$$\mathcal{R}_i = \bigcup_{j=1}^{r-1} Con_i(\mathbf{c}_j, \mathbf{c}_{j+1}) \cup X_i \cup Y_i.$$



Σχήμα 5.2: Ο χώρος δεικτών ενός αλγορίθμου με 6 διανύσματα εξάρτησης $\mathbf{d}_1 = (1, 7)$, $\mathbf{d}_2 = (2, 6)$, $\mathbf{d}_3 = (3, 3)$, $\mathbf{d}_4 = (5, 3)$, $\mathbf{d}_5 = (4, 2)$ και $\mathbf{d}_6 = (6, 1)$. Οι κώνοι (αριστερά) και οι περιοχές που ορίζονται (δεξιά), χρωματίζονται/γραμμοσκιάζονται αντίστοιχα.

Για την καλύτερη κατανόηση των παραπάνω ορισμών, παρουσιάζουμε στο σχήμα 5.2 τον χωρισμό του χώρου δεικτών ενός αλγορίθμου με έξι διανύσματα εξάρτησης, σε κώνους και περιοχές. Η ακολουθία των διανυσμάτων των κώνων είναι $CV = \langle \mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_5, \mathbf{d}_6 \rangle$.

Σημειώνουμε επίσης δύο περιοχές κώνων, την $Con_2(\mathbf{d}_3, \mathbf{d}_5)$ και την $Con_3(\mathbf{d}_5, \mathbf{d}_6)$, καθώς και τις πρώτες τρεις περιοχές \mathcal{R}_1 , \mathcal{R}_2 και \mathcal{R}_3 του χώρου δεικτών. Τα σημεία A και E ανήκουν στην περιοχή \mathcal{R}_1 , τα σημεία F , C και G ανήκουν στην \mathcal{R}_2 και τα σημεία B και D ανήκουν στην \mathcal{R}_3 .

5.3 Περιγραφή Κώνων και Περιοχών

Η μέθοδος που παρουσιάζεται στις επόμενες παραγράφους, διαμερίζει τον χώρο δεικτών σε πεπερασμένο αριθμό κώνων και περιοχών. Οι κώνοι μεταξύ τους δεν έχουν κοινά σημεία και συνολικά καλύπτουν το χώρο δεικτών. Το ίδιο ισχύει και για τις οριζόμενες περιοχές.

Τα διανύσματα εξάρτησης που σχηματίζουν τους κώνους –στους οποίους διαμερίζεται ο χώρος δεικτών– ονομάζονται διανύσματα κώνων (*cone vectors*). Εξ’ ορισμού, κάθε σημείο το οποίο περιέχεται σε έναν από τους κώνους μπορεί να γραφεί ως γραμμικός συνδυασμός των διανύσματων που σχηματίζουν τον κώνο με θετικούς συντελεστές. Διαισθητικά, τα διανύσματα κώνων καθορίζουν τη σειρά εκτέλεσης μεταξύ των σημείων που ανήκουν στον κώνο, καθώς ορίζουν το βέλτιστο υπερεπίπεδο δρομολόγησης για τον κώνο, όπως αποδεικνύεται στο [AKT⁺95].

Χάριν παραδείγματος, θεωρούμε τον κώνο που σχηματίζεται από τα διανύσματα \mathbf{d}_{xmin} και \mathbf{d}_{ymin} στο σχήμα 5.3 και ένα σημείο του χώρου $\mathbf{j} = (j_x, j_y)$. Αν $\mathbf{j} = \lambda_1 \cdot \mathbf{d}_{xmin} + \lambda_2 \cdot \mathbf{d}_{ymin}$, τότε διαχωρίζουμε τις ακόλουθες περιπτώσεις:²

- αν $\lambda_1 > 0$ και $\lambda_2 > 0$, τότε $\mathbf{j} \in Con(\mathbf{d}_{xmin}, \mathbf{d}_{ymin})$ π.χ. $A \in Con(\mathbf{d}_{xmin}, \mathbf{d}_{ymin})$
- αν $\lambda_1 > 0$ και $\lambda_2 < 0$, τότε $\mathbf{j} \in ConY$ π.χ. $B \in ConY$
- αν $\lambda_1 < 0$ και $\lambda_2 > 0$, τότε $\mathbf{j} \in ConX$. π.χ. $C \in ConX$

Στην περίπτωση που έχουμε περισσότερους κώνους, οι περιπτώσεις 2 και 3 υποδηλώνουν ότι το σημείο μας βρίσκεται αριστερά και δεξιά του κώνου $Con(\mathbf{d}_{xmin}, \mathbf{d}_{ymin})$, αντίστοιχα. Η καλύτερα, δεν ανήκει στον εν λόγω κώνο, αλλά βρίσκεται (έξω από αυτόν) προς την πλευρά του διανύσματος στο οποίο αντιστοιχεί ο θετικός συντελεστής.

Θεωρούμε πάλι τον αλγόριθμο του σχήματος 5.3. Λύνοντας την εξίσωση $A = \lambda_1 \cdot \mathbf{d}_{xmin} + \lambda_2 \cdot \mathbf{d}_{ymin}$ ως προς (λ_1, λ_2) , παίρνουμε:

$$(\lambda_1, \lambda_2) = \left(\frac{17}{11}, \frac{9}{11} \right) \Rightarrow A \in Con(\mathbf{d}_{xmin}, \mathbf{d}_{ymin}).$$

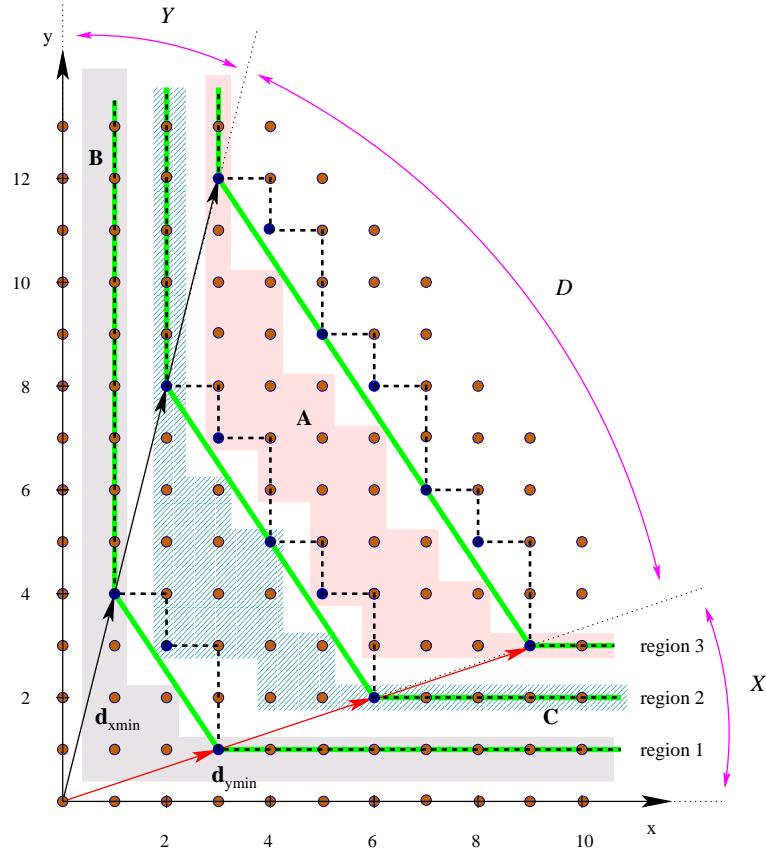
Αντίστοιχα, για το σημείο B παίρνουμε:

$$(\lambda_1, \lambda_2) = \left(\frac{35}{11}, \frac{-8}{11} \right) \Rightarrow B \in ConY.$$

²Χωρίς βλάβη της γενικότητας, θεωρούμε ότι τα \mathbf{d}_{xmin} και \mathbf{d}_{ymin} είναι λεξικογραφικά ταξινομημένα.

Τέλος, για το σημείο C παίρνουμε:

$$(\lambda_1, \lambda_2) = \left(\frac{-5}{11}, \frac{42}{11} \right) \Rightarrow C \in ConX.$$



Σχήμα 5.3: Εντοπισμός σημείου σε χώρο δεικτών με 3 κώνους, όπου $\mathbf{d}_{xmin} = (1, 4)$ και $\mathbf{d}_{ymin} = (3, 1)$.

Η βασική ιδέα πίσω από τον αλγόριθμο δρομολόγησης είναι η ακόλουθη: αν το σημείο j ανήκει γεωμετρικά στην περιοχή R_i , τότε είναι δυνατόν να εκτελεστεί τη χρονική στιγμή i , δεδομένου ότι $R_i \subseteq ECT_i \cup ECT_{i-1}$, από τη σχέση 5.2.

Έτσι, για να βρούμε πότε θα εκτελεστεί το σημείο j , δεν έχουμε παρά να προσδιορίσουμε την περιοχή στην οποία ανήκει. Το πρώτο βήμα είναι ο προσδιορισμός του κώνου, στον οποίον ανήκει το σημείο j . π.χ. αν ανήκει στους κώνους $ConX$, $ConY$, ή $Con(\mathbf{c}_i, \mathbf{c}_{i+1})$, για κάποιο

από τα ζεύγη των διανυσμάτων κώνων. Αυτό γίνεται λύνοντας τις εξισώσεις

$$\mathbf{j} = \lambda_1 \cdot \mathbf{c}_i + \lambda_2 \cdot \mathbf{c}_{i+1}, \text{ όπου } 1 \leq i \leq r-1,$$

και ελέγχοντας το πρόσημο των λ_1 και λ_2 . Για λεπτομέρειες, δείτε τον Αλγόριθμο 5.2.

Μετά τον προσδιορισμό του κώνου, υπολογίζεται η ακριβής περιοχή στην οποία ανήκει το σημείο \mathbf{j} . Γίνεται χρήση των ακόλουθων εκφράσεων, όπως φαίνεται και στο σχήμα 5.3:

- ▷ **αν** $\mathbf{j} \in ConY$: $[\# \pi \rho \iota \chi \dot{\eta}_S] = \lceil \frac{j_x}{d_{xmin_x}} \rceil$ $\pi.\chi. B \in \mathcal{R}_1$
- ▷ **αν** $\mathbf{j} \in ConX$: $[\# \pi \rho \iota \chi \dot{\eta}_S] = \lceil \frac{j_y}{d_{ymin_y}} \rceil$ $\pi.\chi. C \in \mathcal{R}_2$
- ▷ **αν** $\mathbf{j} \in Con(\mathbf{d}_{xmin}, \mathbf{d}_{ymin})$: $[\# \pi \rho \iota \chi \dot{\eta}_S] = \lceil \lambda_1 + \lambda_2 \rceil$, όπου $\mathbf{j} = \lambda_1 \cdot \mathbf{d}_{xmin} + \lambda_2 \cdot \mathbf{d}_{ymin}$,
όπως παραπάνω. $\pi.\chi. A \in \mathcal{R}_3$

Για το εν λόγω σχήμα είναι $A \in \mathcal{R}_3$, δεδομένου ότι $\lceil \frac{17}{11} + \frac{9}{11} \rceil = 3$, $B \in \mathcal{R}_1$, αφού $\lceil \frac{1}{1} \rceil = 1$ και $C \in \mathcal{R}_2$, αφού $\lceil \frac{2}{1} \rceil = 2$.

Συνοψίζοντας, ο χώρος δεικτών διαμερίζεται σε μια ακόλουθα από κώνους και ο κάθε κώνος τμηματίζεται περαιτέρω σε περιοχές. Στο εσωτερικό κάθε κώνου, ο αριθμός της περιοχής καθορίζει τη χρονική στιγμή κατά την οποία εκτελούνται τα σημεία που ανήκουν σ' αυτή. Η όλη διαδικασία περιγράφεται από τα ακόλουθα βήματα:

1. Από τα διανύσματα εξάρτησης προσδιορίζονται τα διανύσματα κώνων (*cone vectors*)· με βάση τα διανύσματα αυτά, ο χώρος δεικτών διαμερίζεται σε κώνους και περιοχές.
2. Η καθεαυτό δρομολόγηση μπορεί να πραγματοποιηθεί, προσδιορίζοντας τον κώνο και την περιοχή που ανήκουν τα δεδομένα σημεία.

Δεδομένου ενός φωλιασμένου βρόχου, με ακόλουθα διανυσμάτων εξάρτησης $DV = \langle \mathbf{d}_1, \dots, \mathbf{d}_m \rangle$, κάθε ένα από τα παραπάνω βήματα παρουσιάζεται στη συνέχεια ως ξεχωριστός αλγόριθμος.

5.4 Προσδιορισμός των Κώνων

Για τον σχηματισμό των κώνων απαιτείται ο υπολογισμός των διανυσμάτων των κώνων. Η διαδικασία υπολογισμού τους αποτελεί το κυρίως τμήμα της μεθόδου και συνάμα αποτελεί το κομμάτι με τη μεγαλύτερη πολυπλοκότητα.

Παρουσιάζουμε έναν αλγόριθμο από το χώρο της υπολογιστικής γεωμετρίας, ο οποίος υπολογίζει τα διανύσματα αυτά. Ως είσοδο παίρνει την ακολουθία των διανυσμάτων κυματομορφής (pattern vectors) $PV = \langle \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k \rangle$ και υπολογίζει τη ζητούμενη ακολουθία των διανυσμάτων κώνων CV .³

Για τον υπολογισμό της ακολουθίας PV , χρησιμοποιούμε τον *Αλγόριθμο 4.2* του προηγουμένου κεφαλαίου, με είσοδο την ακολουθία DV . Κατά τη διάρκεια του αλγορίθμου, χρησιμοποιείται η βοηθητική ακολουθία CV' , όπου $CV' = \langle \mathbf{p}_2, \dots, \mathbf{p}_{k-1} \rangle = CV - \{\mathbf{d}_{xmin}, \mathbf{d}_{ymin}\}$.

Αλγόριθμος 5.1

Υπολογισμός των διανυσμάτων κώνων

DivideConeVector($PV, \mathbf{d}_{xmin}, \mathbf{d}_{ymin}$)

INPUT: Η ακολουθία των k pattern vectors $PV = \langle \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k \rangle$ του αλγορίθμου.

OUTPUT: Η ακολουθία των r cone vectors $CV = \langle \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_r \rangle$, όπου $r \leq k \leq m$.

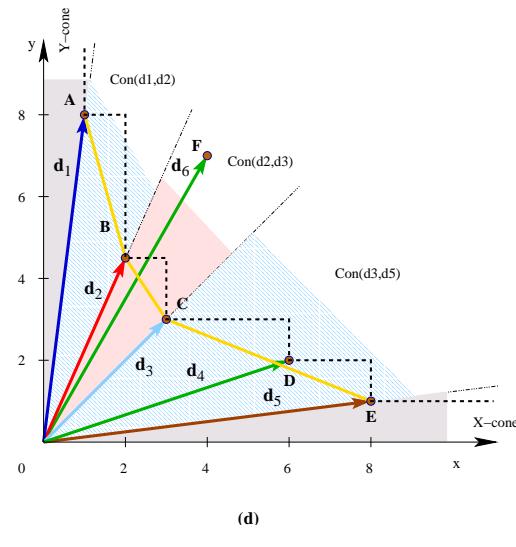
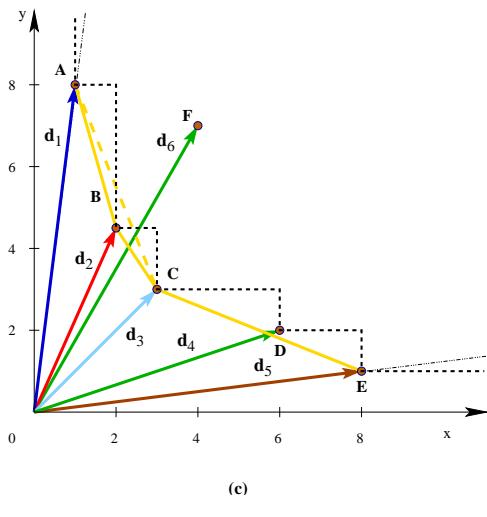
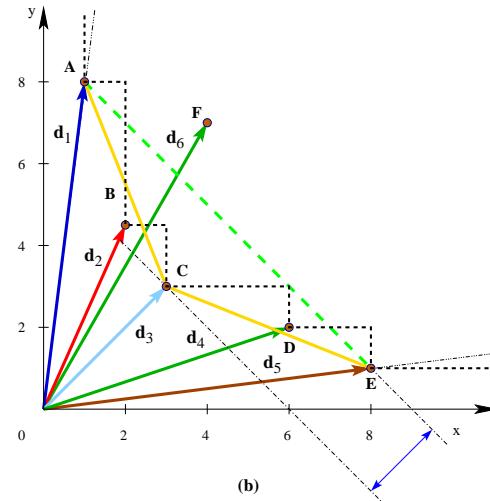
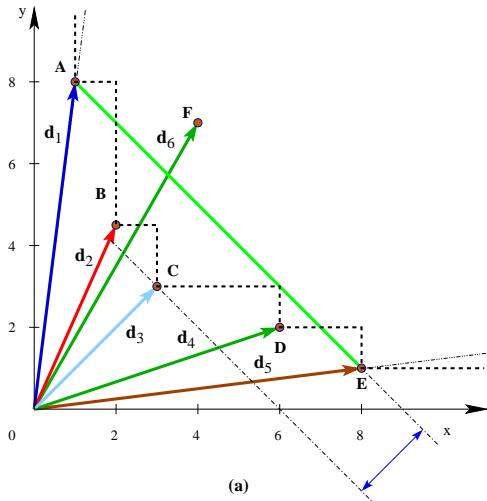
1. Έστω $CV' = PV - \{\mathbf{d}_{xmin}, \mathbf{d}_{ymin}\}$.
2. Αφαιρούνται από το CV' όλα τα διανύσματα εξάρτησης, τα οποία υπερβαίνουν τη γραμμή που ορίζεται από το ευθύγραμμο τμήμα $\overline{\mathbf{d}_{xmin} \mathbf{d}_{ymin}}$.⁴
3. Αν $CV' = \emptyset$, τότε επιστρέφεται η ακολουθία $\langle \mathbf{d}_{xmin}, \mathbf{d}_{ymin} \rangle$.
4. Διαφορετικά, αν $CV' = \langle \mathbf{c} \rangle$, τότε επιστρέφεται η ακολουθία $\langle \mathbf{d}_{xmin}, \mathbf{c}, \mathbf{d}_{ymin} \rangle$.
5. Αν κανένα από τα παραπάνω δεν ισχύει:
Έστω $s = (d_{ymin_y} - d_{xmin_y}) / (d_{ymin_x} - d_{xmin_x})$. Έστω $\mathbf{p}_i = (p_{i_x}, p_{i_y}) \in PV$ το διάνυσμα το οποίο ελαχιστοποιεί την έκφραση $p_{i_y} - s \cdot p_{i_x}$ (το σημείο αυτό καλείται *furthest point*).
6. Έστω $CV'_l = \langle \mathbf{c} \in CV' : c_x < p_{i_x} \wedge \mathbf{c} \in \langle \mathbf{d}_{xmin}, \mathbf{p}_i \rangle \rangle$ και
 $CV'_r = \langle \mathbf{c} \in CV' : c_x > p_{i_x} \wedge \mathbf{c} \in \langle \mathbf{p}_i, \mathbf{d}_{ymin} \rangle \rangle$.
7. Επιστρέφεται η συνένωση των ακολουθιών $\text{DivideConeVector}(\langle \mathbf{d}_{xmin}, CV'_l \rangle, \mathbf{d}_{xmin}, \mathbf{p}_i)$ και $\text{DivideConeVector}(\langle CV'_r, \mathbf{d}_{ymin} \rangle, \mathbf{p}_i, \mathbf{d}_{ymin})$.

Για την καλύτερη κατανόηση του αλγορίθμου, στο σχήμα 5.4 παρουσιάζουμε ένα παράδειγμα εφαρμογής του. Το εν λόγω πρόβλημα έχει έξι διανύσματα εξάρτησης $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_6$. Είναι $PV = \langle \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_4, \mathbf{d}_5 \rangle$.

Σε πρώτη φάση, σχηματίζεται το ευθύγραμμο τμήμα \overline{AE} που συνδέει τα τερματικά σημεία των \mathbf{d}_{xmin} και \mathbf{d}_{ymin} (a). Τα διανύσματα \mathbf{d}_1 και \mathbf{d}_5 εισάγονται στην ακολουθία των cone

³Σημειώνουμε ότι $\mathbf{p}_1 = \mathbf{d}_{xmin}$ και $\mathbf{p}_k = \mathbf{d}_{ymin}$.

⁴Το ευθύγραμμο τμήμα το οποίο συνδέει τα τερματικά σημεία των \mathbf{d}_{xmin} και \mathbf{d}_{ymin} .



Σχήμα 5.4: **a)** Τα $\mathbf{d}_{x_{min}}$ και $\mathbf{d}_{y_{min}}$ εισάγονται στην CV και υπολογίζεται το πιο απομακρυσμένο σημείο. **b)** Το διάνυσμα που αντιστοιχεί σ' αυτό, επιλέγεται επίσης για εισαγωγή στην ακολουθία. **c)** Ο αλγόριθμος συνεχίζει στους δύο υποκώνους στους οποίους χωρίζει το νέο διάνυσμα των αρχικό κώνων. Στον $Con(\mathbf{d}_1, \mathbf{d}_3)$ προστίθεται ένα ακόμη διάνυσμα, ενώ στον $Con(\mathbf{d}_3, \mathbf{d}_5)$ η διαδικασία ολοκληρώθηκε. **d)** Παρουσιάζονται οι κώνοι που σχηματίζονται.

vectors: $CV = \langle \mathbf{d}_1, \mathbf{d}_5 \rangle$. Από τα εναπομείναντα σημεία, επιλέγεται το πιο απομακρυσμένο από το ευθύγραμμο τμήμα (προς την αρχή των αξόνων): σημείο C . Το διάνυσμα εξάρτησης \mathbf{d}_3 , το

οποίο έχει τερματικό σημείο το C , εισάγεται στην ακολουθία (b): $CV = \langle \mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_5 \rangle$.

Στη συνέχεια, λόγω του διανύσματος \mathbf{d}_3 , ο αρχικός κώνος $Con(\mathbf{d}_1, \mathbf{d}_5)$ χωρίζεται στους δύο υποκώνους $Con(\mathbf{d}_1, \mathbf{d}_3)$ και $Con(\mathbf{d}_3, \mathbf{d}_5)$. Ο αλγόριθμος επαναλαμβάνεται αναδρομικά στους δύο αυτούς κώνους. Πρώτα εφαρμόζεται στον $Con(\mathbf{d}_1, \mathbf{d}_3)$, όπου προστίθεται στην ακολουθία το διάνυσμα \mathbf{d}_2 (c). Στον κώνο $Con(\mathbf{d}_3, \mathbf{d}_5)$ η διδικασία έχει ολοκληρωθεί. Τελικά η ακολουθία των διανύσματων κώνων περιλαμβάνει τα διανύσματα $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ και \mathbf{d}_5 και οι σχηματιζόμενοι κώνοι είναι 5 (d).

5.5 Προσδιορισμός των Περιοχών

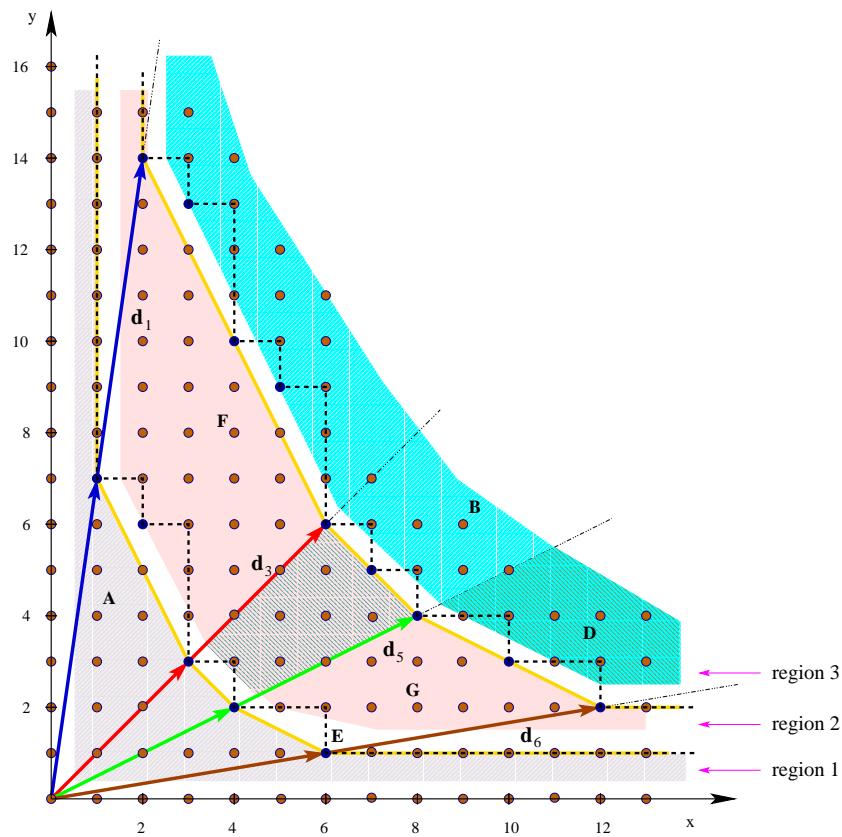
Έχοντας υπολογίσει τα διανύσματα κώνων για το δούέν πρόβλημα, οποιαδήποτε περιοχή του χώρου δεικτών αλλά και κάθε κώνου, μπορεί να υπολογιστεί άμεσα.

Για τον προσδιορισμό της περιοχής R_1 του χώρου δεικτών, εφαρμόζονται αρχικά τα διανύσματα κώνων (cone vector) στην αρχή των αξόνων. Από το άκρο του \mathbf{d}_{xmin} φέρνουμε ημιευθεία παράλληλη με τον άξονα των y , κατά τη διεύθυνση αύξησης των y . Αντίστοιχα, από το άκρο του \mathbf{d}_{ymin} φέρνουμε ημιευθεία παράλληλη με τον άξονα των x , κατά τη διεύθυνση αύξησης των x . Κατόπιν, ενώνουμε όλα τα διαδοχικά ζεύγη των cone vectors, με ευθύγραμμα τμήματα. Η περιοχή R_1 ορίζεται από την τεθλασμένη γραμμή που προκύπτει και τους άξονες, μη συμπεριλαμβανομένων των αξόνων (σχήμα 5.5).

Εντελώς ανάλογα, για τον προσδιορισμό της περιοχής i , εφαρμόζονται τα cone vectors στην αρχή των αξόνων, i και $i-1$ φορές. Φέρνουμε ημιευθείες από τα άκρα των $i\mathbf{d}_{xmin}, (i-1)\mathbf{d}_{xmin}$ και των $i\mathbf{d}_{ymin}, (i-1)\mathbf{d}_{ymin}$, παράλληλα με τους άξονες των y και x αντίστοιχα, όπως προηγουμένως. Κατόπιν, ενώνουμε με ευθύγραμμα τμήματα όλα τα διαδοχικά ζεύγη των i και $i-1$ πολλαπλασίων των cone vectors.

Η περιοχή R_i ορίζεται από τις δύο τεθλασμένες γραμμές που προκύπτουν, χωρίς να περιλαμβάνει τα σημεία που ανήκουν στην γραμμή $i-1$. Στο σχήμα 5.5 παρουσιάζονται οι περιοχές R_1, R_2 και R_3 , για φωλιασμένο βρόχο με τέσσερα διανύσματα κώνων (cone vectors).

Για την εύρεση της περιοχής i ενός κώνου, εφαρμόζουμε ένα τμήμα της διαδικασίας που μόλις περιγράψαμε. Έστω δύο διανύσματα κώνων \mathbf{a} και \mathbf{b} . Αναζητούμε την περιοχή $Con_i(\mathbf{a}, \mathbf{b})$. Εφαρμόζουμε στην αρχή των αξόνων τα i και τα $i-1$ πολλαπλασία των διανύσματων \mathbf{a} και \mathbf{b} . Φέρνουμε τα ευθύγραμμα τμήματα που ενώνουν το i πολλαπλάσιο του \mathbf{a} με το i πολλαπλάσιο



Σχήμα 5.5: Διαμέριση του χώρου δεικτών σε περιοχές, γνωρίζοντας τα διανύσματα κώνων.

του **b** και το $i - 1$ πολλαπλάσιο του **a** με το $i - 1$ πολλαπλάσιο του **b**.

Η περιοχή του κώνου που βρίσκεται μεταξύ των δύο αυτών ευθυγράμμων τμημάτων, μη συμπεριλαμβανομένων των σημείων που ανήκουν στην γραμμή $i - 1$, αποτελεί τη ζητούμενη περιοχή $Con_i(\mathbf{a}, \mathbf{b})$. Στο σχήμα 5.5 παρουσιάζονται οι περιοχές $Con_2(\mathbf{d}_3, \mathbf{d}_5)$ και $Con_3(\mathbf{d}_5, \mathbf{d}_6)$.

5.6 Γεωμετρική Δρομολόγηση με Κώνους & Περιοχές

Δεδομένων των αλγορίθμων υπολογισμού των κώνων και των περιοχών του χώρου δεικτών ενός φωλιασμένου βρόχου, η δρομολόγηση των σημείων του χώρου καθίσταται προφανής. Υπολογίζοντας τα διανύσματα κώνων, ο χώρος δεικτών του φωλιασμένου βρόχου διαμερίζεται σε κώνους και περιοχές.

Έπειτα, δεδομένου οποιουδήποτε σημείου j του χώρου, προσδιορίζεται ο κώνος $Con(\mathbf{c}_k, \mathbf{c}_{k+1})$ στον οποίο ανήκει. Στη συνέχεια, προσδιορίζεται η περιοχή του κώνου $Con_i(\mathbf{c}_k, \mathbf{c}_{k+1})$, όπου ειδικότερα ανήκει. Κατόπιν αυτού, η εκτέλεση του σημείου j δρομολογείται κατά τη χρονική στιγμή $t = i$. Οι λεπτομέρειες περιέχονται στον αλγόριθμο που ακολουθεί.

Αλγόριθμος 5.2

Γεωμετρική Δρομολόγηση στις 2 Διαστάσεις

INPUT: Το σύνολο των εξαρτήσεων $DS = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m\}$ και ένα σημείο του χώρου δεικτών j .

OUTPUT: Η χρονική στιγμή t_j κατά την οποία το σημείο j θα εκτελεστεί.

1. Εφαρμόζεται ο Αλγόριθμος 4.1 στο σύνολο DS , οπότε υπολογίζεται η ακολουθία των pattern vectors PV και τα διανύσματα \mathbf{d}_{xmin} και \mathbf{d}_{ymin} .
2. Υπολογίζονται τα r cone vectors, εφαρμόζοντας τον Αλγόριθμο 5.1 στην ακολουθία PV . Μετά από αυτό το βήμα είναι: $CV = \langle \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_r \rangle$, όπου $\mathbf{c}_1 = \mathbf{d}_{xmin}$ και $\mathbf{c}_r = \mathbf{d}_{ymin}$.
3. Εύρεση του κώνου στον οποίο ανήκει το σημείο j : για κάθε ζεύγος $(\mathbf{c}_i, \mathbf{c}_{i+1})$ διαδοχικών cone vectors $\mathbf{c}_i, \mathbf{c}_{i+1} \in CV$, επιλύεται η εξίσωση

$$\mathbf{j} = \lambda_{i1} \cdot \mathbf{c}_i + \lambda_{i2} \cdot \mathbf{c}_{i+1}$$

ως προς τα λ_{i1} και λ_{i2} :

- αν για τα $(\mathbf{d}_{xmin}, \mathbf{c}_2)$ είναι $\lambda_{11} \geq 0$ και $\lambda_{12} < 0$, τότε $\mathbf{j} \in ConY$
- αν για τα $(\mathbf{c}_{r-1}, \mathbf{d}_{ymin})$ είναι $\lambda_{r-1,1} < 0$ και $\lambda_{r-1,2} \geq 0$, τότε $\mathbf{j} \in ConX$
- αν $\lambda_{i1} \geq 0$ και $\lambda_{i2} \geq 0$, τότε $\mathbf{j} \in Con(\mathbf{c}_i, \mathbf{c}_{i+1})$.

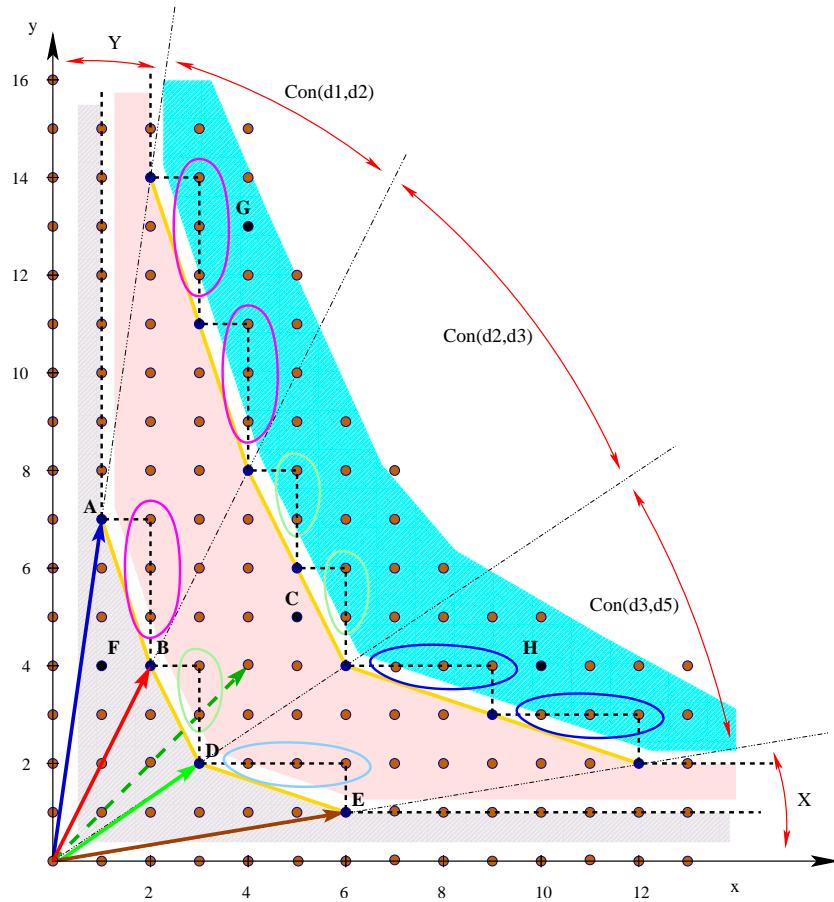
4. Γνωρίζοντας τον κώνο στον οποίο ανήκει το σημείο j , δρομολογείται η εκτέλεσή του σύμφωνα με τα ακόλουθα:

- ▷ αν $\mathbf{j} \in ConY \Rightarrow t_j = \lceil \frac{j_x}{d_{xmin_x}} \rceil = \lceil \lambda_{12} \rceil$
- ▷ αν $\mathbf{j} \in ConX \Rightarrow t_j = \lceil \frac{j_y}{d_{ymin_y}} \rceil = \lceil \lambda_{r-1,1} \rceil$
- ▷ αν $\mathbf{j} \in Con(\mathbf{c}_i, \mathbf{c}_{i+1}) \Rightarrow t_j = \lceil \lambda_{i1} + \lambda_{i2} \rceil$.

Το ακόλουθο παράδειγμα επιδεικνύει την διαδικασία δρομολόγησης που μόλις περιγράφηκε από τον Αλγόριθμο 5.2.

Παράδειγμα 5.1

Θεωρούμε τον διπλά φωλιασμένο βρόχο του σχήματος 5.6. Εμπεριέχει πέντε διανύσματα εξάρτησης $\mathbf{d}_1 = (1, 7)$, $\mathbf{d}_2 = (2, 4)$, $\mathbf{d}_3 = (3, 2)$, $\mathbf{d}_4 = (4, 4)$ και $\mathbf{d}_5 = (6, 1)$.



Σχήμα 5.6: Χώρος δεικτών αλγορίθμου με 5 διανύσματα εξάρτησης $\mathbf{d}_1 = (1, 7)$, $\mathbf{d}_2 = (2, 4)$, $\mathbf{d}_3 = (3, 2)$, $\mathbf{d}_4 = (4, 4)$ και $\mathbf{d}_5 = (6, 1)$. Οι 5 κώνοι που σχηματίζονται χρωματίζονται/σκιαγραφούνται ανάλογα· σημειώνονται και οι 3 πρώτες περιοχές.

Εφαρμόζοντας τα δύο πρώτα βήματα του Αλγορίθμου 5.2, σχηματίζεται η ακολουθία των cone vectors: $CV = \langle \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_5 \rangle$. Με τη βοήθεια των διανυσμάτων αυτών, ο χώρος δεικτών χωρίζεται σε 5 κώνους: $ConY$, $Con(\mathbf{d}_1, \mathbf{d}_2)$, $Con(\mathbf{d}_2, \mathbf{d}_3)$, $Con(\mathbf{d}_3, \mathbf{d}_5)$ και $ConX$.

Έστω ότι επιθυμούμε να δρομολογήσουμε την εκτέλεση του σημείου $C = (5, 5)$. Σχηματί-

Ζονται οι εξισώσεις:

$$\text{εντοπισμός του } C: \begin{cases} Con(\mathbf{d}_1, \mathbf{d}_2) : \lambda_1 \cdot \mathbf{d}_1 + \lambda_2 \cdot \mathbf{d}_2 = (5, 5) \Rightarrow \lambda_1 = -1 \text{ και } \lambda_2 = 3 \\ Con(\mathbf{d}_2, \mathbf{d}_3) : \lambda_1 \cdot \mathbf{d}_2 + \lambda_2 \cdot \mathbf{d}_3 = (5, 5) \Rightarrow \lambda_1 = 5/8 \text{ και } \lambda_2 = 5/4 \\ Con(\mathbf{d}_3, \mathbf{d}_5) : \lambda_1 \cdot \mathbf{d}_3 + \lambda_2 \cdot \mathbf{d}_5 = (5, 5) \Rightarrow \lambda_1 = 25/9 \text{ και } \lambda_2 = -5/9. \end{cases}$$

Από τα παραπάνω, υποδηλώνεται ότι το σημείο C βρίσκεται στον κώνο $Con(\mathbf{d}_2, \mathbf{d}_3)$. Για αυτόν τον κώνο είναι: $\lceil \lambda_1 + \lambda_2 \rceil = \lceil 15/8 \rceil = 2$: το σημείο C δηλαδή, μπορεί να εκτελεστεί τη χρονική στιγμή $t_C = 2$.

Ακολουθώντας ανάλογη διαδικασία για τα σημεία $F = (1, 4)$, $G = (4, 13)$ και $H = (10, 4)$, μπορούμε να επαληθεύσουμε τις στιγμές εκτέλεσης τους:

$$\begin{aligned} F: \quad \lambda_1 \cdot \mathbf{d}_1 + \lambda_2 \cdot \mathbf{d}_2 &= (1, 4) \Rightarrow \lambda_1 = 2/5 \text{ και } \lambda_2 = 3/10 \Rightarrow F \in Con(\mathbf{d}_1, \mathbf{d}_2) \text{ και } t_F = 1 \\ G: \quad \lambda_1 \cdot \mathbf{d}_1 + \lambda_2 \cdot \mathbf{d}_2 &= (4, 13) \Rightarrow \lambda_1 = 1 \text{ και } \lambda_2 = 3/2 \Rightarrow G \in Con(\mathbf{d}_1, \mathbf{d}_2) \text{ και } t_G = 3 \\ H: \quad \lambda_1 \cdot \mathbf{d}_3 + \lambda_2 \cdot \mathbf{d}_5 &= (10, 4) \Rightarrow \lambda_1 = 14/9 \text{ και } \lambda_2 = 8/9 \Rightarrow H \in Con(\mathbf{d}_3, \mathbf{d}_5) \text{ και } t_H = 3. \end{aligned}$$

Στο σχήμα 5.6 παρουσιάζεται η διαμέριση του χώρου δεικτών σε περιοχές. Σημειώνονται τα διανύσματα εξάρτησης που σχηματίζουν τους απαιτούμενους κώνους (cone vectors). Η εκτέλεση ξεκινάει από τη χρονική στιγμή $t = 1$. Με διακεκομένη γραμμή σημειώνονται τα ακριβή όρια των περιοχών ECT_i .

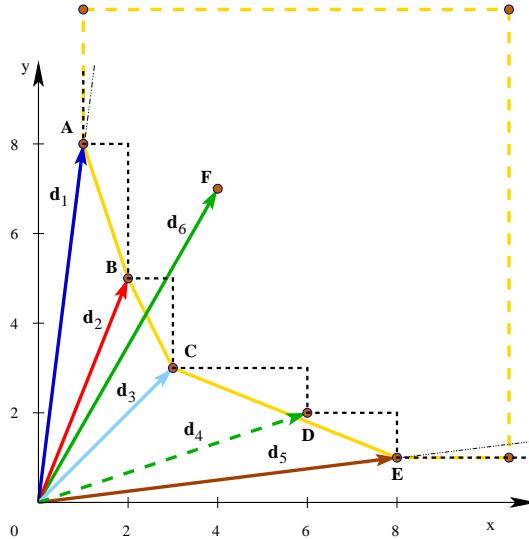
Παρατηρούμε ότι η εκτέλεση ορισμένων σημείων επιβραδύνεται από το βέλτιστο κατά 1 χρονική στιγμή. Αυτό γίνεται γιατί η ακολουθία των R_i προέρχεται από την ακολουθία των ECT_i , σύμφωνα με την σχέση 5.2:

$$R_i = ECT_i \cup ECT_{i-1}.$$

Αν το προς δρομολόγηση σημείο ανήκει στη χρονική περιοχή ECT_i , δρομολογείται κατά τη μικρότερη δυνατή στιγμή -βέλτιστη εκτέλεση. Αν ανήκει στη χρονική περιοχή ECT_{i-1} , η δρομολόγηση του καθυστερεί κατά μία χρονική στιγμή -σχεδόν βέλτιστη εκτέλεση. Το γεγονός αυτό υποδεικνύει ότι η τεχνική δρομολόγησης που προτείνεται είναι ασυμπτωτικά βέλτιστη, όπως θα αποδειχτεί και στη συνέχεια. ■

5.7 Συσχέτιση με το Πρόβλημα του Convex Hull

Στο σημείο αυτό παρουσιάζουμε τη συσχέτιση που εμφανίζεται κατά τον χωρισμό του χώρου δεικτών σε κώνους και περιοχές, με το πρόβλημα υπολογισμού του κυρτού περιγράμματος (convex hull) συνόλου σημείων.



Σχήμα 5.7: Το κυρτό περίγραμμα που σχηματίζουν οι απολήξεις των διανυσμάτων εξάρτησης και τα συνοριακά σημεία (d_{xmin_x}, y_{max}) , (x_{max}, y_{max}) και (x_{max}, d_{ymin_y}) . Τα σημεία που συμμετέχουν στο κυρτό περίγραμμα –πλήν των συνοριακών– αποτελούν τα διανύσματα κώνων του αλγορίθμου.

5.7.1 Εναλλακτικός Υπολογισμός των Cone Vectors

Έστω ότι αναζητούμε τα διανύσματα κώνων (cone vectors) για δεδομένο αλγόριθμο με έξι διανύσματα εξάρτησης. Αντί του Αλγορίθμου 5.1 που προτείναμε, μπορεί επίσης να εφαρμοστεί η διαδικασία του σχήματος 5.7. Θεωρούμε τις απολήξεις των διανυσμάτων εξάρτησης του αλγορίθμου. Θεωρούμε επίσης, τα σημεία που προκύπτουν αν προβάλλουμε τις απολήξεις των d_{xmin} και d_{ymin} στις ευθείες οι οποίες ορίζουν τα άνω φράγματα του χώρου. Θεωρούμε και το τερματικό σημείο του χώρου. Έστω ότι αυτά τα τρία επιπλέον σημεία είναι τα (d_{xmin_x}, y_{max}) , (x_{max}, y_{max}) και (x_{max}, d_{ymin_y}) , όπου x_{max} , y_{max} τα άνω όρια των συντεταγμένων x και y αντίστοιχα.

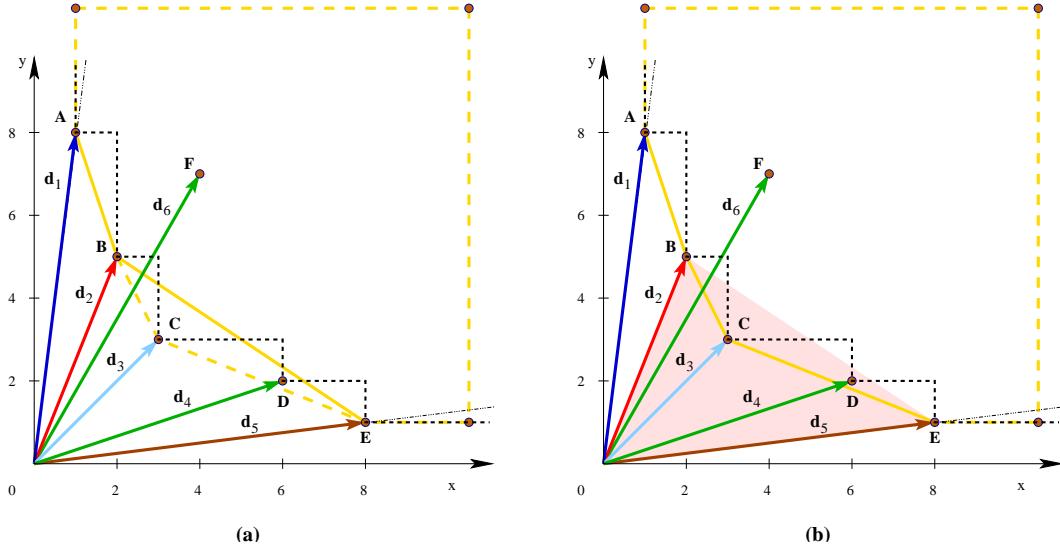
Το κυρτό περίγραμμα των σημείων αυτών, αποτελείται από τα διανύσματα κώνων του αλγορίθμου, συν τα τρία συνοριακά σημεία, σύμφωνα με το θεώρημα που ακολουθεί. Αν αφαιρεθούν αυτά από το κυρτό περίγραμμα, απομένουν τα ζητούμενα σημεία.

Θεώρημα 5.1 Έστω φωλιασμένος βρόχος με ακολουθία διανυσμάτων εξάρτησης

$$DV = \langle d_1, d_2, \dots, d_{m-1}, d_m \rangle = \langle d_{xmin}, d_2, \dots, d_{m-1}, d_{ymin} \rangle.$$

Έστω (x_{max}, y_{max}) το τερματικό σημείο του χώρου δεικτών και (d_{xmin_x}, y_{max}) και (x_{max}, d_{ymin_y})

οι προβολές των $\mathbf{d}_{x_{min}}$ και $\mathbf{d}_{y_{min}}$ στις ευθείες $x = x_{max}$ και $y = y_{max}$. Έστω \mathcal{CH} το κυρτό περίγραμμα των απολήξεων των διανυσμάτων εξάρτησης και των τριών αυτών σημείων. Το τυχαίο διάνυσμα εξάρτησης $\mathbf{d}_\ell \in DV$ περιλαμβάνεται στο \mathcal{CH} αν και μόνον εάν είναι διάνυσμα κώνου του αλγορίθμου.



Σχήμα 5.8: Επεξήγηση του Θεωρήματος 5.1: όλα τα διανύσματα κώνων του αλγορίθμου αποτελούν σημεία του κυρτού περιγράμματος \mathcal{CH} και αντίστροφα.

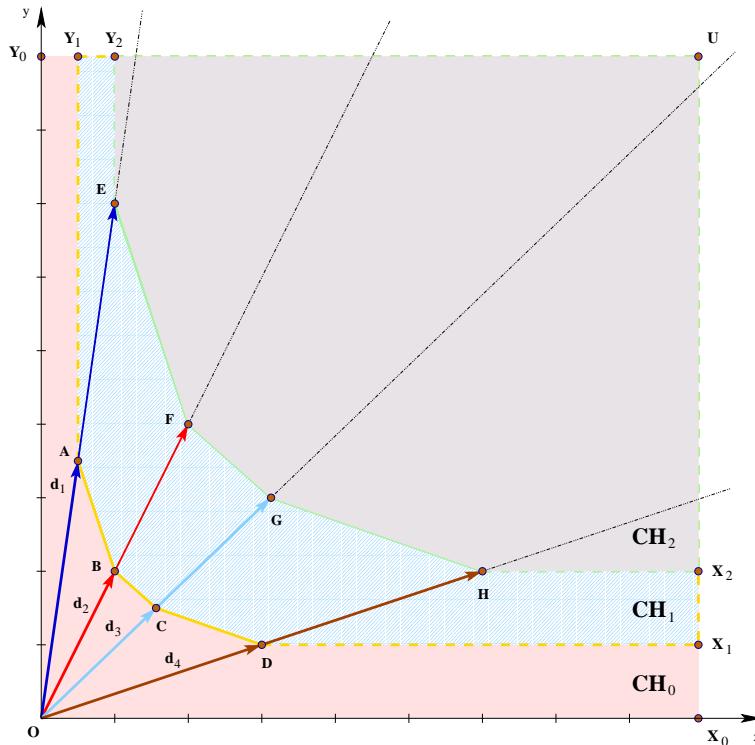
Απόδειξη

Θα δείξουμε αρχικά ότι όλα τα διανύσματα κώνων περιέχονται στο \mathcal{CH} . Έστω ότι το διάνυσμα κώνου \mathbf{d}_ℓ δεν ανήκει στο κυρτό περίγραμμα. Αυτό συνεπάγεται ότι βρίσκεται έξω από αυτό, όπως στο σχήμα 5.8(a), με $\mathbf{d}_\ell = \mathbf{d}_3$. Από τον ορισμό του κυρτού περιγράμματος όμως, αυτό καθίσταται άτοπο, γιατί όλα τα σημεία του κυρτού περιγράμματος πρέπει να περιέχονται μέσα σ' αυτό. Θα δείξουμε τώρα ότι όλα τα σημεία που αποτελούν το κυρτό περίγραμμα είναι διανύσματα κώνων. Έστω ότι το διάνυσμα $\mathbf{d}_3 \in \mathcal{CH}$ δεν ανήκει στα διανύσματα κώνων. Στην περίπτωση αυτή, και αν ισχύει η διάταξη του σχήματος 5.8(b), θα μπορεί να εκφραστεί ως προς τα εκατέροιν διανύσματα κώνων ως εξής: $\mathbf{d}_3 = \lambda_1 \cdot \mathbf{d}_2 + \lambda_2 \cdot \mathbf{d}_5$, όπου $\lambda_1 + \lambda_2 > 1$.⁵ Αυτό όμως δεν συμβαίνει, μιας και περιλαμβάνεται στο τρίγωνο που ορίζουν. Άρα το \mathbf{d}_3 ανήκει στα

⁵ Αυτή η ιδιότητα εξ' ορισμού ισχύει για οποιοδήποτε διάνυσμα εξάρτησης ως προς τα διανύσματα κώνων που ορίζουν το μικρότερο κώνο στον οποίο ανήκει.

διανύσματα κώνων. Συνεπώς, όλα τα διανύσματα κώνων του αλγορίθμου αποτελούν σημεία του κυρτού περιγράμματος CH και αντίστροφα.

Σημειώνουμε ότι ο *Αλγόριθμος 5.1* για τον υπολογισμό των διανυσμάτων των κώνων, αποτελεί παραλλαγή του αλγορίθμου *QuickHull*,⁶ ο οποίος προτείνεται για τον υπολογισμό του κυρτού περιγράμματος συνόλου σημείων στις n διαστάσεις [BDH96].



Σχήμα 5.9: Ορισμός των περιοχών του χώρου δεικτών μέσω μιας ακολουθίας κυρτών περιγραμμάτων.

⁶Για περισσότερες πληροφορίες για τη μέθοδο αυτή, ο ενδιαφερόμενος αναγνώστης μπορεί να ανατρέξει στα [Dro01], [Cha95] και [KS86].

5.7.2 Εναλλακτικός Ορισμός των Περιοχών

Χρησιμοποιώντας την ιδέα του κυρτού περιγράμματος, μπορούμε να ορίσουμε μια ακολουθία κυρτών περιγραμμάτων

$$\langle \mathcal{CH}_0, \mathcal{CH}_1, \dots, \mathcal{CH}_i, \dots \rangle$$

ως εξής. Ως \mathcal{CH}_0 , ορίζουμε το κυρτό περιγράμμα που προκύπτει παίρνοντας το τερματικό σημείο $\mathbf{U} = (x_{max}, y_{max})$, την αρχή των αξόνων $\mathbf{O} = (0, 0)$ και τις προβολές της στις τερματικές συνοριακές γραμμές $\mathbf{Y}_0 = (0, y_{max})$, $\mathbf{X}_0 = (x_{max}, 0)$, όπως φαίνεται στο σχήμα 5.9. Είναι δηλαδή:

$$\mathcal{CH}_0 = Conv(P_0), \text{ όπου } P_0 = \{\mathbf{O}, \mathbf{X}_0, \mathbf{U}, \mathbf{Y}_0\},$$

όπου $Conv(P_0)$ το κυρτό περιγράμμα των σημείων του συνόλου P_0 .

Αν $\mathbf{Y}_1, \mathbf{X}_1$ οι προβολές των απολήξεων των διανυσμάτων εξάρτησης \mathbf{d}_{xmin} και \mathbf{d}_{ymin} στα επίπεδα τα οποία ορίζουν τα άνω φράγματα του χώρου, ορίζεται ως:

$$\mathcal{CH}_1 = Conv(P_1), \text{ όπου } P_1 = \{\{\mathbf{c}_j \in CV \mid 1 \leq j \leq r\}, \mathbf{X}_1, \mathbf{U}, \mathbf{Y}_1\}.$$

Γενικά, αν $\mathbf{Y}_i, \mathbf{X}_i$ οι προβολές των απολήξεων των διανυσμάτων $i \cdot \mathbf{d}_{xmin}$ και $i \cdot \mathbf{d}_{ymin}$ στα επίπεδα τα οποία ορίζουν τα άνω φράγματα του χώρου, ορίζεται ως:

$$\mathcal{CH}_i = Conv(P_i), \text{ όπου } P_i = \{\{i \cdot \mathbf{c}_j \mid \mathbf{c}_j \in CV, 1 \leq j \leq r\}, \mathbf{X}_i, \mathbf{U}, \mathbf{Y}_i\}.$$

Μέσω των ορισμών αυτών, μπορούμε εναλλακτικά να ορίσουμε τις περιοχές \mathcal{R}_i ως εξής:

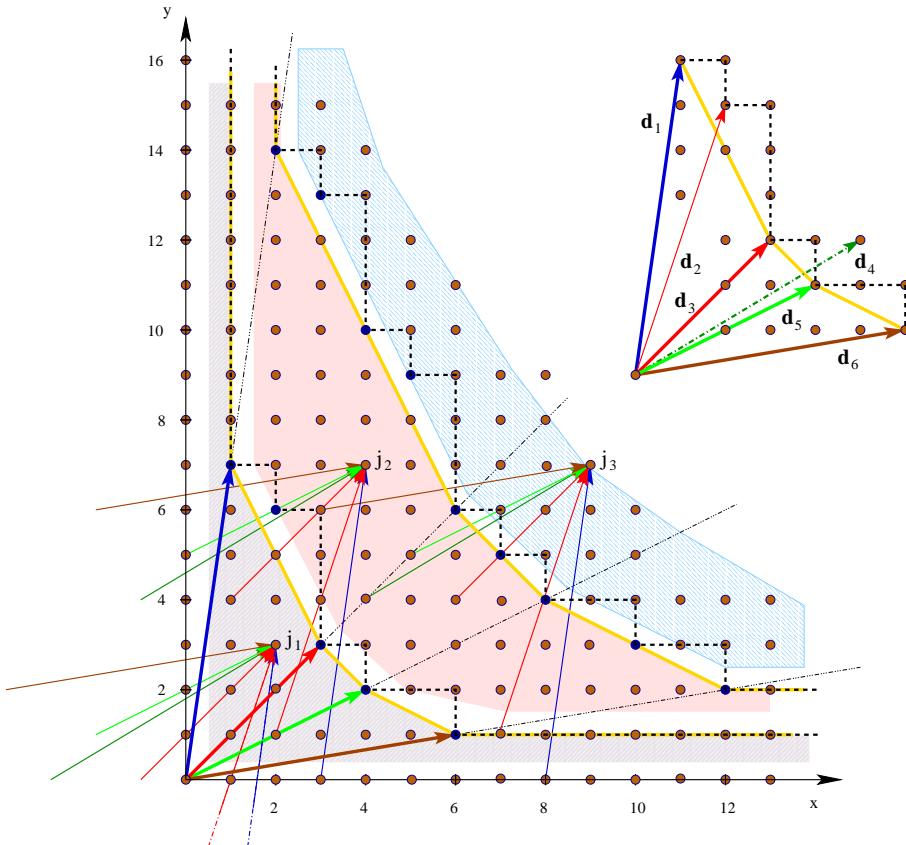
$$\mathcal{R}_i = \mathcal{CH}_{i-1} - \mathcal{CH}_i, \text{ για } i \geq 1. \quad (5.3)$$

5.8 Θέματα Εγγυρότητας & Βελτιστοποίησης

Είναι προφανές ότι ο αλγόριθμος δρομολόγησης που προτείνουμε είναι έγκυρος, δεδομένου ότι σέβεται και διατηρεί τις υπάρχουσες εξαρτήσεις μεταξύ των σημείων του χώρου δεικτών.

Οποιοδήποτε σημείο j το οποίο ανήκει στην περιοχή \mathcal{R}_i , εκτελείται τη χρονική στιγμή i . Το γεγονός ότι το σημείο j ανήκει στην περιοχή \mathcal{R}_i σημαίνει ότι $ECT(j) \leq i$. Ας το δούμε αυτό αναλυτικότερα.

Στο σχήμα 5.10 θεωρούμε το χώρο δεικτών του *Παραδείγματος 5.1*. Οι ακολουθίες των διανυσμάτων κυματομορφής και των διανυσμάτων των κώνων (pattern και cone vectors) είναι



Σχήμα 5.10: Ο χώρος δεικτών του Παραδείγματος 5.1. Ο χώρος δεικτών χωρίζεται σε περιοχές, ενώ οι διακεκομμένες γραμμές οριοθετούν τα σύνορα των συνόλων ECT_i . Σημειώνονται όλα τα διανύσματα εξάρτησης, όπως και τα διανύσματα κυματομορφής και κώνων.

$PV = \langle \mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \mathbf{d}_5, \mathbf{d}_6 \rangle$ και $CV = \langle \mathbf{d}_1, \mathbf{d}_3, \mathbf{d}_5, \mathbf{d}_6 \rangle$. Ο χώρος δεικτών χωρίζεται σε περιοχές· οι διακεκομμένες γραμμές οριοθετούν τα σύνορα των συνόλων ECT_i , των σημείων δηλαδή, με μικρότερη δυνατή/βέλτιστη χρονική στιγμή εκτέλεσης i . Θεωρούμε τις περιπτώσεις όπου $\mathbf{j} = \mathbf{j}_1 \in \mathcal{R}_1$, $\mathbf{j} = \mathbf{j}_2 \in \mathcal{R}_2$ και $\mathbf{j} = \mathbf{j}_3 \in \mathcal{R}_3$.

- Έστω σημείο $\mathbf{j} \in \mathcal{R}_1$. Τότε, όπως φαίνεται στο σχήμα 5.10, θα είναι $\mathbf{j} - \mathbf{d} \notin J^2$ για κάθε $\mathbf{d} \in PV$. Έτσι $ECT(\mathbf{j}) = 1$ και $\mathcal{R}_1 \subseteq ECT_1$.
- Θεωρούμε στη συνέχεια, το σημείο $\mathbf{j} \in \mathcal{R}_i$. Τότε, όπως μπορεί να δει κανείς από το

σχήμα 5.10, είναι:

$$\mathbf{j} - \mathbf{d} \in \bigcup_{l=1}^{i-1} \mathcal{R}_l \text{ ή } \mathbf{j} - \mathbf{d} \notin J^2 \text{ για κάθε } \mathbf{d} \in PV.$$

Αν $\mathbf{j}' = \mathbf{j} - \mathbf{d} \in \bigcup_{l=1}^{i-1} \mathcal{R}_l$, τότε $ECT(\mathbf{j}') \leq i - 1$ και το σημείο \mathbf{j}' εκτελέστηκε κατά την τελευταία χρονική στιγμή $i - 1$. Μπορούμε λοιπόν, να εκτελέσουμε το σημείο \mathbf{j} κατά τη χρονική στιγμή i , χωρίς να παραβιάζουμε καμία συνθήκη εξάρτησης. Είναι δηλαδή:

$$ECT(\mathbf{j}) \leq i \text{ και } \mathcal{R}_i \subseteq \bigcup_{k=1}^i ECT_k. \quad (5.4)$$

Τα παραπάνω πιστοποιούν την εγκυρότητα της μεθόδου δρομολόγησης που παρουσιάστηκε. Ας δούμε τώρα την επίδοση της γεωμετρικής μεθόδου, όσον αφορά το χρόνο εκτέλεσης. Είναι η δρομολόγηση που επιτυγχάνεται βέλτιστη και ποια είναι η χειρότερη περίπτωση;

Ο αλγόριθμος μας καθυστερεί την εκτέλεση κάθε σημείου \mathbf{j} , σε σχέση με τη βέλτιστη δυνατή εκτέλεσή του, το πολύ κατά μία χρονική στιγμή. Αυτό σημαίνει ότι:

$$\mathbf{j} \in \mathcal{R}_i \Rightarrow ECT(\mathbf{j}) = i - 1 \text{ ή } ECT(\mathbf{j}) = i \quad (5.5)$$

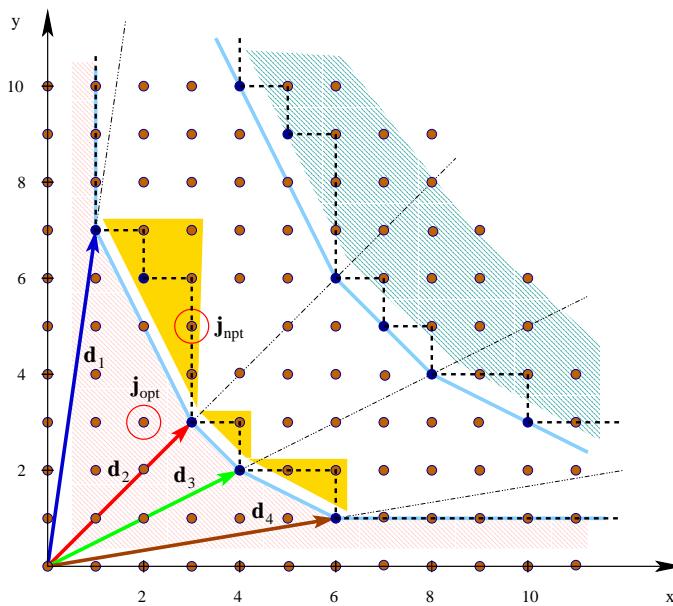
και

$$\mathcal{R}_i \subseteq ECT_{i-1} \cup ECT_i. \quad (5.6)$$

Άρα, η δρομολόγηση που επιτυγχάνεται είναι ασυμπτωτικά βέλτιστη, καθώς η εκτέλεση κάθε σημείου επιβραδύνεται το πολύ κατά σταθερά.

Για να γίνει καλύτερα αντιληπτό αυτό, παρουσιάζουμε ένα στιγμιότυπο της εκτέλεσης στο σχήμα 5.11. Για τα σημεία τα οποία ανήκουν στο ECT_i και δεν περιέχονται στην περιοχή \mathcal{R}_i , η εκτέλεση καθυστερεί κατά μία χρονική στιγμή (σχεδόν βέλτιστη εκτέλεση). π.χ. το σημείο \mathbf{j}_{opt} στο σχήμα.

Αντιθέτως, τα σημεία του ECT_i που περιέχονται στην περιοχή \mathcal{R}_i , εκτελούνται κατά τη χρονική στιγμή i (βέλτιστη εκτέλεση). π.χ. το σημείο \mathbf{j}_{opt} στο σχήμα που προαναφέραμε. Στο εν λόγω σχήμα σημειώνονται όλα τα σημεία, των οποίων η εκτέλεση καθυστερεί όλα τα υπόλοιπα σημεία εκτελούνται κατά τη μικρότερη χρονική στιγμή που είναι δυνατή η εκτέλεσή τους (earliest computation time).



Σχήμα 5.11: Σχέδιον βέλτιστη δρομολόγηση του Παραδείγματος 5.1. Για τα σημεία του ECT_1 , τα οποία περιλαμβάνονται στα χρωματισμένα τρίγωνα της περιοχής \mathcal{R}_2 , η εκτέλεση καθυστερεί από τη βέλτιστη κατά μία χρονική στιγμή· π.χ. το j_{npt} . Τα σημεία αυτά εκτελούνται τη χρονική στιγμή $t = 2$, ενώ μπορούν να εκτελεστούν τη χρονική στιγμή $t = 1$. Όλα τα υπόλοιπα σημεία του ECT_1 εκτελούνται κατά τη μικρότερη δυνατή χρονική στιγμή ($t = 1$).

Αποδεικνύεται με τον τρόπο αυτό ότι η τεχνική δρομολόγησης που προτείνουμε είναι σχεδόν βέλτιστη, όσον αφορά το χρόνο εκτέλεσης, με σταθερή καθυστέρηση μιας χρονικής στιγμής το πολύ.

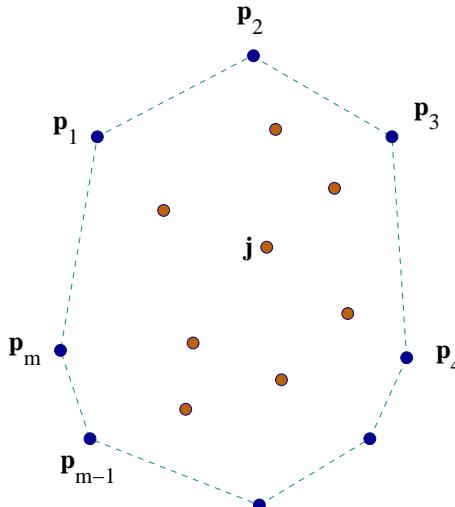
5.8.1 Εγκυρότητα με βάση τα Κυρτά Περιγράμματα

Με βάση των ορισμό των περιοχών που δίνεται από τη σχέση 5.3

$$\mathcal{R}_i = \mathcal{CH}_i - \mathcal{CH}_{i-1}, \text{ για } i \geq 1,$$

αποδεικνύεται στη συνέχεια η εγκυρότητα της μεθόδου. Ότι δηλαδή για την εκτέλεση οποιουδήποτε σημείου που ανήκει στην περιοχή \mathcal{R}_i , απαιτούνται μόνο σημεία τα οποία ανήκουν σε προηγούμενες περιοχές και είναι ήδη υπολογισμένα.

Η απόδειξη βασίζεται στην ιδιότητα που συνδέει οποιοδήποτε εσωτερικό σημείο σε κυρτό



Σχήμα 5.12: Οποιοδήποτε σημείο που περικλείεται σε κυρτό περίγραμμα, μπορεί να εκφραστεί ως γραμμικός συνδυασμός των σημείων που το αποτελούν, με θετικούς συντελεστές, οι οποίοι έχουν άθροισμα μονάδα.

περίγραμμα με τα σημεία που το αποτελούν. Έστω το κυρτό περίγραμμα του σχήματος 5.12. Το οποιοδήποτε εσωτερικό του σημείο \mathbf{j} μπορεί να εκφραστεί ως γραμμικός συνδυασμός των σημείων που αποτελούν το κυρτό περίγραμμα, με θετικούς συντελεστές και τέτοιους ώστε να έχουν άθροισμα μονάδα [Zie95, Grü67, BNO01]. Υπάρχουν δηλαδή, $k_1, k_2, \dots, k_m \in \mathbb{R}$ τέτοια ώστε:

$$\mathbf{j} = \sum_{i=1}^m k_i \cdot \mathbf{p}_i, \text{ όπου } k_i \geq 0 \text{ και } \sum_{i=1}^m k_i = 1. \quad (5.7)$$

Για δεδομένο αλγόριθμο διπλά φωλιασμένων βρόχων, θεωρούμε σημείο \mathbf{j}_k του χώρου δεικτών, το οποίο βρίσκεται στο εσωτερικό του κυρτού περιγράμματος \mathcal{CH}_k . Σύμφωνα με τη σχέση 5.7, το εν λόγω σημείο \mathbf{j}_k συνδέεται με τα σημεία του συνόρου του \mathcal{CH}_k με την εξής σχέση:

$$\mathbf{j}_k = \sum_{i=1}^r \lambda_i \cdot \mathbf{c}_i + \lambda_x \cdot \mathbf{X}_k + \lambda_y \cdot \mathbf{Y}_k + \lambda_u \cdot \mathbf{U}, \text{ όπου } \sum_{i=1}^r \lambda_i + \lambda_x + \lambda_y + \lambda_u = 1$$

με τους συντελεστές $\lambda_i, \lambda_x, \lambda_y, \lambda_u$ μη αρνητικούς.

Επίσης, για οποιοδήποτε διάνυσμα εξάρτησης $\mathbf{d} \in PV$, με $\mathbf{d} \in Con(\mathbf{c}_\ell, \mathbf{c}_{\ell+1})$, λόγω της

βασικής ιδιότητας των διανυσμάτων των κώνων ισχύει η σχέση:

$$\mathbf{d} = \sum_{i=1}^r \mu_i \cdot \mathbf{c}_i, \text{ όπου } \mu_i = 0 \text{ για κάθε } \mu_i \neq \mu_\ell, \mu_{\ell+1} \text{ και } \mu_\ell + \mu_{\ell+1} \geq 1.$$

Από τις δύο αυτές σχέσεις μπορούμε να βρούμε μια έκφραση για το σημείο $\mathbf{j} - \mathbf{d}$ του χώρου δεικτών, από το οποίο εξαρτάται το σημείο \mathbf{j} :

$$\mathbf{j} - \mathbf{d} = \sum_{i=1}^r (\lambda_i - \mu_i) \cdot \mathbf{c}_i + \lambda_x \cdot X_k + \lambda_y \cdot Y_k + \lambda_u \cdot U \quad (5.8)$$

Στην έκφραση αυτή της διαφοράς, οι συντελεστές έχουν άθροισμα s , όπου:

$$\begin{aligned} s &= \sum_{i=1}^r (\lambda_i - \mu_i) + \lambda_x + \lambda_y + \lambda_u \\ &= \underbrace{\sum_{i=1}^r \lambda_i + \lambda_x + \lambda_y + \lambda_u}_{=1} - \underbrace{\sum_{i=1}^r \mu_i}_{\geq 1} \Rightarrow s \leq 0 \end{aligned} \quad (5.9)$$

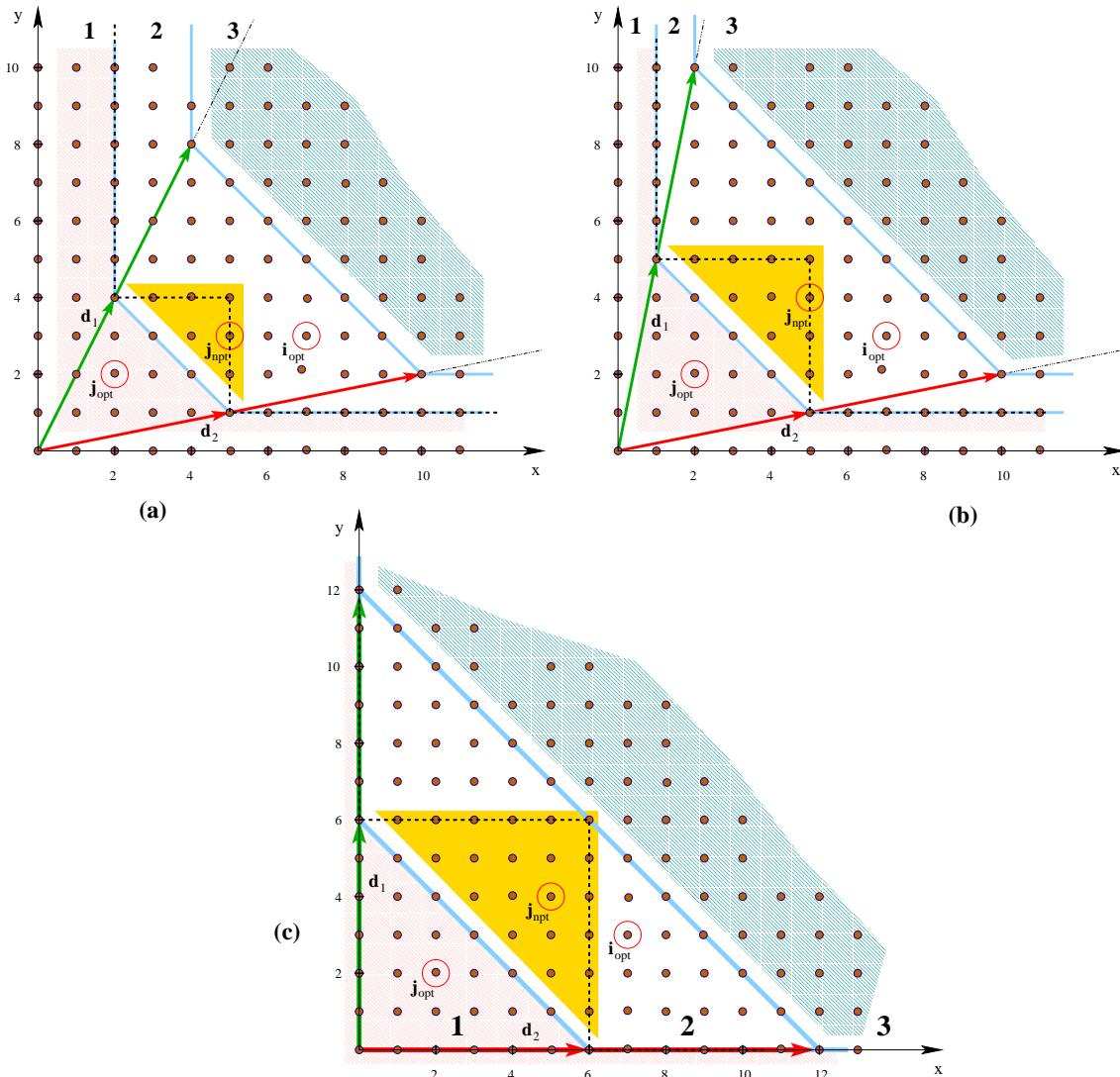
Λόγω των αρχικών περιορισμών για τα λ_i και μ_i , δεν υπάρχουν κάποιες τιμές γι' αυτά ούτως ώστε να ισχύει $s = 1$. Αυτό συνεπάγεται ότι για οποιοδήποτε σημείο \mathbf{j} του χώρου δεικτών και για οποιοδήποτε διάνυσμα εξάρτησης \mathbf{d} , το σημείο $\mathbf{j} - \mathbf{d}$ δεν περικλείεται στο κυρτό περίγραμμα \mathcal{CH}_k –όπου περικλείεται το \mathbf{j} . Άρα η παραπάνω προτεινόμενη διαμέριση σε περιοχές είναι έγκυρη.

5.8.2 Βελτιστοποίηση

Εμβαθύνοντας περισσότερο στο σχήμα 5.11, έχουμε τις τρεις περιπτώσεις του σχήματος 5.13. Δοθέντος ενός κώνου $Con(\mathbf{d}_1, \mathbf{d}_2)$, και εφαρμόζοντας την γεωμετρική δρομολόγηση που προτείνεται, τα σημεία που δεν εκτελούνται την μικρότερη δυνατή στιγμή, είναι αυτά που περιέχονται στο χρωματισμένο τρίγωνο (a).

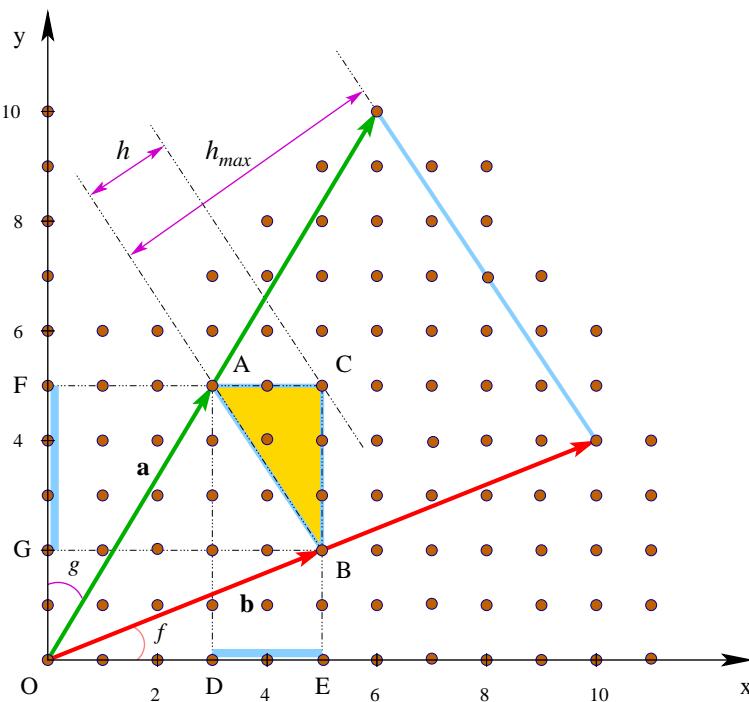
Όσο τα διανύσματα που σχηματίζουν τον κώνο ανοίγουν (πλησιάζουν τους άξονες), τόσο το τρίγωνο αυτό μεγαλώνει (b). Παρατηρούμε όμως, ότι τα σημεία που περιέχονται σ' αυτό, βρίσκονται στη περιοχή \mathcal{R}_2 . Το τρίγωνο αυτό γίνεται μέγιστο όταν τα διανύσματα βρίσκονται πάνω στους άξονες (c). Σε αυτή την περίπτωση, το άνω άκρο του τριγώνου εφάπτεται με το άνω άκρο της περιοχής \mathcal{R}_2 . Ούτε σε αυτή την περίπτωση όμως δεν εισχωρεί στην περιοχή \mathcal{R}_3 .

Η τυπικότερη απόδειξη αυτού γίνεται μέσω του σχήματος 5.14. Για δεδομένο κώνο $Con(\mathbf{a}, \mathbf{b})$, το ύψος του τριγώνου h εκφράζεται ως συνάρτηση των διανυσμάτων \mathbf{a} και \mathbf{b} που δημιουργούν τον



Σχήμα 5.13: Δοθέντος ενός κώνου $\text{Con}(\mathbf{d}_1, \mathbf{d}_2)$, τα σημεία που δεν εκτελούνται κατά τη μικρότερη δυνατή στιγμή, περιέχονται στο χρωματισμένο τρίγωνο (a). Όσο τα διανύσματα που σχηματίζουν τον κώνο ανοίγουν (πλησιάζοντας τους άξονες), τόσο το τρίγωνο αυτό μεγαλώνει (b). Στη χειρότερη των περιπτώσεων, τα διανύσματα βρίσκονται πάνω στους άξονες και το τρίγωνο είναι το μέγιστο δυνατό (c).

κώνο και των γωνιών f και g που σχηματίζουν αυτά με τους άξονες. Αναλυτικά, εκφράζοντας



Σχήμα 5.14: Δοθέντος ενός κώνου $Con(\mathbf{a}, \mathbf{b})$, τα σημεία που δεν εκτελούνται την μικρότερη δυνατή στιγμή, είναι αυτά που περιέχονται στο χρωματισμένο τρίγωνο. Αν f και g οι γωνίες που σχηματίζουν με τους άξονες, το ύψος του τριγώνου εξαρτάται από το μήκος των διανυσμάτων και των γωνιών αυτών.

το εμβαδόν του τριγώνου με δύο τρόπους είναι:

$$\frac{1}{2} \cdot h \cdot |AB| = \frac{1}{2} \cdot |AC| \cdot |BC| = \frac{1}{2} \cdot |DE| \cdot |FG| \quad (5.10)$$

Ισχύει όμως και ότι:

$$\begin{aligned} |DE| &= |\mathbf{b}| \cdot \cos(f) - |\mathbf{a}| \cdot \sin(g) \\ |FG| &= |\mathbf{a}| \cdot \cos(g) - |\mathbf{b}| \cdot \sin(f) \end{aligned} \quad (5.11)$$

Από τις δύο αυτές σχέσεις το h εκφράζεται ως προς τα μέτρα των διανυσμάτων του κώνου και σε συνάρτηση με τις γωνίες f και g :

$$h = \frac{|AC| \cdot |BC|}{|AB|} = \frac{2|\mathbf{a}| \cdot |\mathbf{b}| \cdot \cos(f-g) - |\mathbf{b}|^2 \cdot \sin(2f) - |\mathbf{a}|^2 \cdot \sin(2g)}{2\sqrt{|\mathbf{b}|^2 + |\mathbf{a}|^2 - 2|\mathbf{a}| \cdot |\mathbf{b}| \cdot \sin(f+g)}} \quad (5.12)$$

Από την έκφραση αυτή βλέπουμε ότι το h μεγιστοποιείται όταν $f = g = 0$, όταν δηλαδή

έχουμε τα διανύσματα επάνω στους άξονες. Στην περίπτωση αυτή είναι:

$$h = h_{max} = \frac{|\mathbf{a}| \cdot |\mathbf{b}|}{\sqrt{|\mathbf{b}|^2 + |\mathbf{a}|^2}} \quad (5.13)$$

Για οποιεσδήποτε άλλες τιμές των $|\mathbf{a}|$, $|\mathbf{b}|$, f και g , το h είναι μικρότερο του h_{max} . Τα σημεία του τριγώνου λοιπόν, σε καμία περίπτωση δεν φτάνουν στην περιοχή \mathcal{R}_3 . Το πολύ να φτάσουν μέχρι το όριο της περιοχής \mathcal{R}_2 .

Αποδεικνύεται έτσι, ότι η δρομολόγηση του κάθε σημείου με την προτεινόμενη γεωμετρική μέθοδο, καθυστερεί το πολύ κατά μία χρονική στιγμή.

5.9 Πολυπλοκότητα

Στην παράγραφο αυτή θα μελετήσουμε την πολυπλοκότητα των αλγορίθμων που συνθέτουν τη γεωμετρική μέθοδο δρομολόγησης. Πιο συγκεκριμένα, θα υπολογίσουμε το χρονικό κόστος που απαιτείται για τον προσδιορισμό των διανυσμάτων κώνων (cone vectors) –Αλγόριθμος 5.1– και για τη γεωμετρική δρομολόγηση δεδομένου σημείου σε χώρο δύο διαστάσεων –Αλγόριθμος 5.2.

Ο Αλγόριθμος 5.1 αποτελεί παραλλαγή του αλγορίθμου DivideHull2d που παρουσιάστηκε από τον Chan [Cha95] και αναφέρεται στον υπολογισμό κυρτού περιγράμματος συνόλου n σημείων στο επίπεδο. Για λόγους πληρότητας παρουσιάζεται στο Παράρτημα B'. Ο αλγόριθμος αυτός παρουσιάζει πολυπλοκότητα $\mathcal{O}(n \log h)$, όπου n το πλήθος των σημείων της εισόδου (σημεία για τα οποία αναζητείται το κυρτό περίγραμμα) και h το πλήθος των σημείων που αποτελούν το κυρτό περίγραμμα προφανώς $h \leq n$.⁷

Εντελώς αντίστοιχα, ο υπολογισμός των διανυσμάτων κώνων παρουσιάζει πολυπλοκότητα $\mathcal{O}(k \log r)$, όπου k το πλήθος των διανυσμάτων της εισόδου (διανύσματα κυματομορφής) και r το πλήθος των διανυσμάτων της εξόδου (διανύσματα κώνων). Σημειώνουμε ότι $r \leq k \leq m$, όπου m τα διανύσματα εξάρτησης:

$$Cx(5.1) = \mathcal{O}(k \log r), \text{ όπου } r \leq k \leq m. \quad (5.14)$$

Το πρώτο βήμα του Αλγορίθμου 5.2 αποτελεί ο υπολογισμός των διανυσμάτων κυματομορφής, μέσω του Αλγορίθμου 4.1. Στη συνέχεια υπολογίζονται τα διανύσματα κώνων εφαρμόζοντας τον Αλγόριθμο 5.1. Κατόπιν, η επίλυση των r συστημάτων 2×2 απαιτεί γραμμικό χρόνο $\mathcal{O}(r)$ και

⁷Για την απόδειξη της εν λόγω πολυπλοκότητας ο ενδιαφερόμενος αναγνώστης παραπέμπεται στο [Cha95].

η δρομολόγηση πραγματοποιείται ευθύς αμέσως. Είναι λοιπόν, για την πολυπλοκότητα $Cx(5.2)$ της γεωμετρικής δρομολόγησης στις δύο διαστάσεις:

$$\begin{aligned}
 Cx(5.2) &= Cx(4.1) + Cx(5.1) + \mathcal{O}(r) \\
 &= \Theta(m \log m) + \mathcal{O}(k \log r) + \mathcal{O}(r), \text{ óπου } r \leq k \leq m \\
 &= \Theta(m \log m).
 \end{aligned} \tag{5.15}$$

Κεφάλαιο 6

Γεωμετρική Δρομολόγηση Προβλημάτων UET-UCT

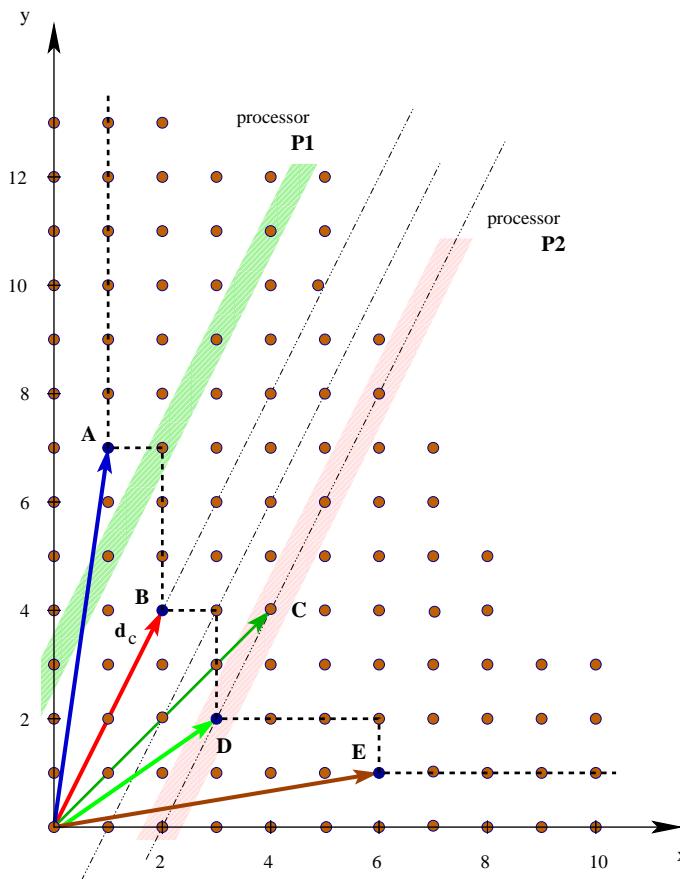
Μέχρι αυτού του σημείου, θεωρήσαμε σιωπηρά ότι αναφερόμαστε σε αρχιτεκτονικές, στις οποίες το κόστος επικοινωνίας μεταξύ δύο μονάδων επεξεργασίας είναι αμελητέο έναντι του κόστους υπολογισμού, το οποίο θεωρείται μοναδιαίο. Το μοντέλο αυτών των αρχιτεκτονικών είναι γνωστό ως μοντέλο μοναδιαίου κόστους υπολογισμού - μηδενικού κόστους επικοινωνίας (*unit execution - zero communication time model - UET*). Σύμφωνα με το μοντέλο αυτό, ένας υπολογισμός για τον οποίο τα δεδομένα που απαιτεί είναι διαθέσιμα κατά το τέλος της χρονικής στιγμής $t = i$, εκτελείται ευθύς αμέσως, κατά την χρονική στιγμή $t = i + 1$.

Θεωρούμε τώρα ένα πιο ρεαλιστικό μοντέλο, όπου η επικοινωνία μεταξύ των μονάδων επεξεργασίας απαιτεί μία επιπλέον χρονική στιγμή. Το μοντέλο αυτό συναντάται στη βιβλιογραφία ως μοντέλο μοναδιαίου κόστους υπολογισμού - μοναδιαίου κόστους επικοινωνίας (*unit execution - unit communication time model - UET-UCT*).

Η βασική διαφορά με το μοντέλο UET είναι το επιπλέον κόστος επικοινωνίας για κάθε ανταλλαγή δεδομένων μεταξύ δύο οποιωνδήποτε επεξεργαστών. Αυτό σημαίνει ότι δύο διαδοχικά σημεία, ως προς το διάνυσμα εξάρτησης d , μπορούν να υπολογιστούν σε διαδοχικές χρονικές στιγμές, μόνον εφόσον εκτελούνται στον ίδιο επεξεργαστή σε κάθε άλλη περίπτωση απαιτείται μία επιπλέον χρονική στιγμή για την επικοινωνία.

Τα παραπάνω υποδηλώνουν ότι στα UET-UCT προβλήματα, υπάρχουν συγκεκριμένες ακολουθίες σημείων οι οποίες υπολογίζονται από τον ίδιο επεξεργαστή. Κάθε τέτοια ακολουθία

σημείων μπορεί να υθεωρηθεί ως μια αλυσίδα υπολογισμών, η οποία δημιουργείται λόγω της ύπαρξης ενός διανύσματος εξάρτησης, όπως φαίνεται στο σχήμα 6.1. Το διάνυσμα αυτό αναφέρεται ως διάνυσμα επικοινωνίας (*communication vector*) \mathbf{d}_c και είναι το μόνο που δεν επιφέρει κόστος επικοινωνίας. Όλα τα υπόλοιπα διανύσματα εξάρτησης απαιτούν ένα κόστος επικοινωνίας μιας επιπλέον χρονικής στιγμής.



Σχήμα 6.1: Στις αρχιτεκτονικές που ακολουθούν το μοντέλο UET-UCT, επιβάλλεται καθυστέρηση μιας χρονικής στιγμής κατά μήκος διανυσμάτων εξάρτησης, εκτός του \mathbf{d}_c . Οι αλυσίδες εκτέλεσης που σχηματίζονται στο χώρο δεικτών, κατά μήκος του \mathbf{d}_c , εκτελούνται από την ίδια μονάδα επεξεργασίας. Στο παράδειγμα μας είναι $\mathbf{d}_c = (2, 4)$.

Στο κεφάλαιο αυτό παρουσιάζουμε τη γενίκευση των γεωμετρικών μεθόδων που περιγράψαμε στα προηγούμενα κεφάλαια, ώστε να αναφέρονται και σε προβλήματα που περιλαμβάνουν

μοναδιαίο κόστος επικοινωνίας: στα προβλήματα φωλιασμένων βρόχων δηλαδή, που αναφέρονται σε αρχιτεκτονικές UET-UCT.

6.1 Το Μοντέλο UET-UCT

Το κύριο χαρακτηριστικό του μοντέλου UET-UCT, είναι το επιπλέον κόστος επικοινωνίας το οποίο εισάγεται από οποιαδήποτε επικοινωνία, μεταξύ διαφορετικών μονάδων επεξεργασίας. Θεωρούμε επίσης, ότι κάθε αλυσίδα υπολογισμών που σχηματίζεται από το διάνυσμα \mathbf{d}_c , ανατίθενται σε ξεχωριστή μονάδα επεξεργασίας, όπως φαίνεται στο σχήμα 6.1. Έτσι, όλα τα διανύσματα εξάρτησης εκτός του \mathbf{d}_c , επιφέρουν μια χρονική μονάδα καθυστέρησης.

Τα χρονικά υποσύνολα του χώρου δεικτών ECT_t , τα οποία αποτελούνται από τα σημεία με earliest computation time ίσο με t , για τα προβλήματα UET-UCT, ορίζονται αναδρομικά ως εξής:

- $ECT_0 = \{\text{συνοριακά/προϋπολογισμένα σημεία}\}$
- $ECT_1 = \{j \in J \mid j - \mathbf{d}_i \in ECT_0, \text{ για κάθε } \mathbf{d}_i \in DV\}$
- $ECT_t = \{j \in J \mid j - \mathbf{d}_i \in ECT_{t-2}, \text{ για κάθε } \mathbf{d}_i \neq \mathbf{d}_c \text{ και } j - \mathbf{d}_c \in ECT_{t-1}\}, \text{ για } t \geq 2.$

Η ιδέα πίσω από τους παραπάνω ορισμούς είναι ότι: τα σημεία του συνόλου ECT_t προκύπτουν από τα σημεία του ECT_{t-1} μέσω του διανύσματος \mathbf{d}_c και από τα σημεία του ECT_{t-2} μέσω οποιουδήποτε άλλου διανύσματος εξάρτησης. Η επιπλέον αυτή χρονική στιγμή δηλώνει το κόστος επικοινωνίας –μια χρονική στιγμή.

Και για τη μελέτη αυτού του μοντέλου χρησιμοποιούμε το συμβολισμό που παρουσιάστηκε στα προηγούμενα κεφάλαια. Ορίζουμε ακόμη τα παρακάτω.

- Δεδομένης της ακολουθίας των διανυσμάτων εξάρτησης $DV = \langle \mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m \rangle$, ορίζουμε ως $\mathcal{MC}(DV) = \langle \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k \rangle$, το αποτέλεσμα της εφαρμογής του Αλγορίθμου 4.2 (minimum cover στις n -διαστάσεις) στην ακολουθία DV . Ως συνέπεια αυτού, για κάθε διάνυσμα $\mathbf{p}_i = (p_{ix}, p_{iy}) \in \mathcal{MC}(DV)$ δεν υπάρχει κάποιο διάνυσμα $\mathbf{d}_j = (d_{jx}, d_{jy}) \in DV$ τέτοιο ώστε $d_{jx} \leq p_{ix}$ και $d_{jy} < p_{iy}$, για $1 \leq i \leq k$ και $1 \leq j \leq m$.
- $PV = \mathcal{MC}(DV) = \langle \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k \rangle = \langle \mathbf{p}_i \rangle$, $1 \leq i \leq k$.

- $DV_2 = \langle \mathbf{d}_1, \dots, \mathbf{d}_{c-1}, \mathbf{d}_{c+1}, \dots, \mathbf{d}_m \rangle$.
- $DV_{UET} = \langle \mathbf{d}_1, \dots, \mathbf{d}_{c-1}, 2\mathbf{d}_c, \mathbf{d}_{c+1}, \dots, \mathbf{d}_m \rangle$.

Σημειώνουμε ότι η ακολουθία PV περιέχει όλα τα διανύσματα κυματομορφής (pattern vectors) και η ακολουθία DV_2 όλα τα διανύσματα κυματομορφής, εκτός του \mathbf{d}_c . Η ακολουθία DV_{UET} αποτελεί μια παραλλαγή της ακολουθίας των διανυσμάτων κυματομορφής, η οποία στη θέση του \mathbf{d}_c έχει το διάνυσμα $2\mathbf{d}_c$.

6.2 Σχηματισμός των Περιγραμμάτων Εκτέλεσης

Η μελέτη μας ξεκινάει από την αναζήτηση μιας έκφρασης, η οποία δίνει το περίγραμμα του σχήματος κυματομορφής (pattern) για την τυχαία χρονική στιγμή t , δεδομένου ότι αναφερόμαστε σε αρχιτεκτονική UET-UCT. Αρχίζουμε και στην περίπτωση αυτή από το αρχικό περίγραμμα. Από τον ορισμό του συνόλου ECT_1 , κάθε σημείο $\mathbf{p} \in pat_1$ δίνεται από την παρακάτω έκφραση –παρατηρούμε την ομοιότητα με την περίπτωση UET:

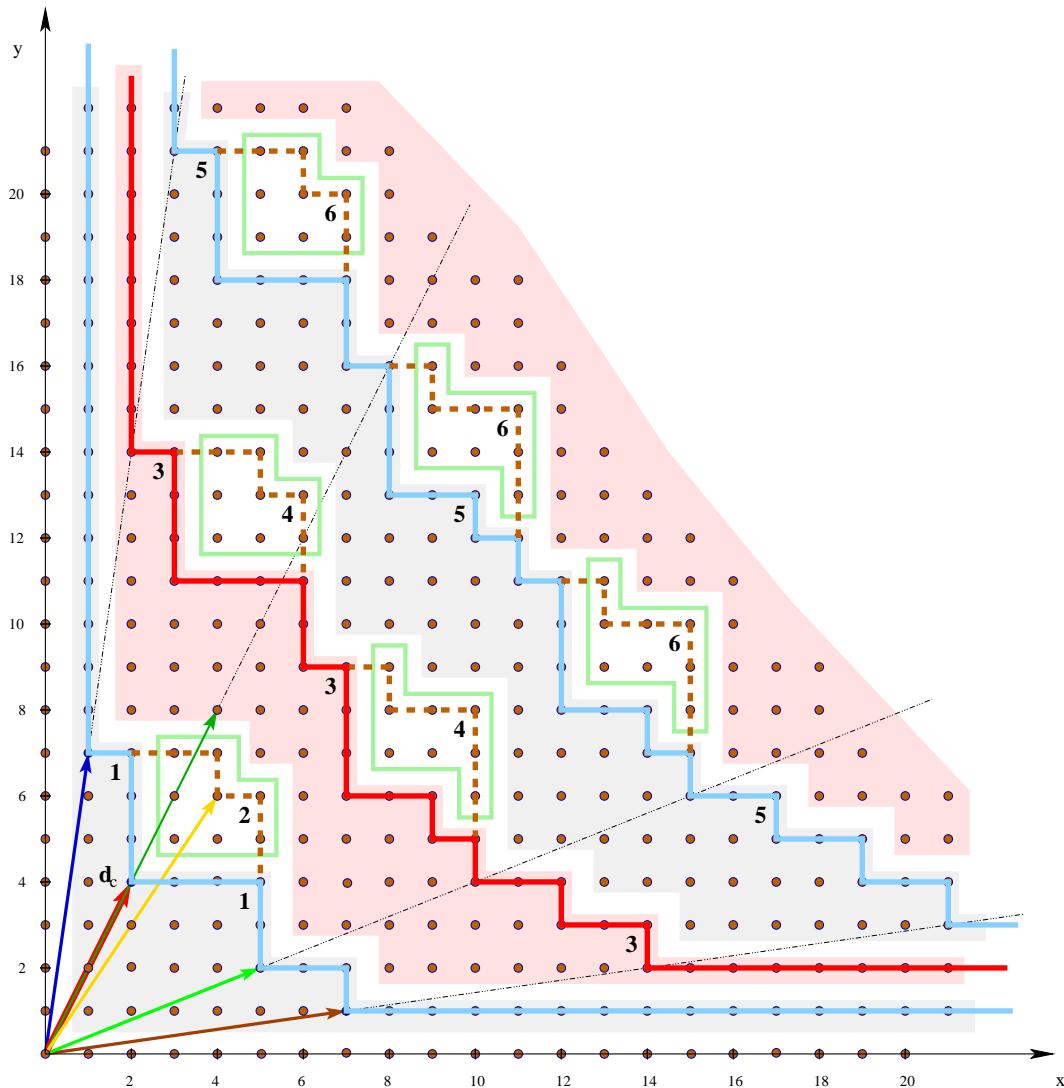
$$\begin{aligned} \mathbf{p} &= \{i_1 \cdot \mathbf{p}_1 + i_2 \cdot \mathbf{p}_2 + \dots + i_k \cdot \mathbf{p}_k\} \\ i_1 + i_2 + \dots + i_k &= 1 \end{aligned} \tag{6.1}$$

Τα διανύσματα \mathbf{p}_i , $1 \leq i \leq k$, αποτελούν τα διανύσματα κυματομορφής (pattern vectors) του προβλήματος.

Η διαφορά από την περίπτωση του UET γίνεται έντονα εμφανής κατά τον προσδιορισμό του σχήματος κυματομορφής της t χρονικής στιγμής, όπου $t > 1$. Το σχήμα αυτό εξαρτάται από τα δύο προηγούμενα σχήματα κυματομορφής, αφού σε κάθε επεξεργαστή P τη χρονική στιγμή t , τα διαθέσιμα προς εκτέλεση σημεία είναι αυτά που εξαρτώνται από:

- τα σημεία που υπολογίστηκαν τη στιγμή $t - 1$ στον ίδιο επεξεργαστή P και από
- τα σημεία που υπολογίστηκαν τη στιγμή $t - 2$ σε οποιονδήποτε άλλον επεξεργαστή –κάθε ένα από τα σημεία αυτά, χρειάζεται μια επιπλέον χρονική στιγμή για να φτάσει στον επεξεργαστή P .

Για τα περιγράμματα των σχημάτων κυματομορφής (pattern outlines) έχουμε τα εξής. Από



Σχήμα 6.2: Η εξέλιξη της εκτέλεσης ενός αλγορίθμου με 5 διανύσματα εξάρτησης $\mathbf{d}_1 = (1, 7)$, $\mathbf{d}_2 = (2, 4)$, $\mathbf{d}_3 = (4, 6)$, $\mathbf{d}_4 = (5, 2)$ και $\mathbf{d}_5 = (7, 1)$, ακολουθώντας το μοντέλο UET-UCT με $\mathbf{d}_c = \mathbf{d}_2 = (2, 4)$. Με κλειστή γραμμή σημειώνονται τα σύνολα ECT_i , για i άρτιο.

τους ορισμούς των χρονικών περιοχών ECT_t , προκύπτουν απευθείας οι παρακάτω εκφράσεις:

$$\begin{aligned} pat_0 &= \langle (0, 0) \rangle \\ pat_1 &= \langle \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k \rangle = PV \\ patt_t &= \mathcal{MC}(\langle pat_{t-2} + DV_2 \rangle \cup \langle pat_{t-1} + \mathbf{d}_c \rangle). \end{aligned} \tag{6.2}$$

Τα παραπάνω διαμορφώνουν μια τελείως διαφορετική εξέλιξη για τα περιγράμματα εκτέλεσης των αλγορίθμων αυτών. Ένα παράδειγμα τέτοιων περιγραμμάτων φαίνεται στο σχήμα 6.2. Σημειώνονται τα χρονικά σύνολα ECT_t για τις πρώτες έξι χρονικές στιγμές, όπως επίσης και τα αντίστοιχα περιγράμματα.

Τα λήμματα και τα θεωρήματα που ακολουθούν, αποδεικνύουν την αναλυτική έκφραση του περιγράμματος του γενικού σχήματος κυματομορφής, για την περίπτωση αλγορίθμων UET-UCT.

Λήμμα 6.1 *Ta minimum cover των ακολουθιών $\langle PV + \mathbf{d}_c \rangle \cup DV_2$ και DV_{UET} συμπίπτουν:*

$$\mathcal{MC}(\langle PV + \mathbf{d}_c \rangle \cup DV_2) = \mathcal{MC}(DV_{UET}).$$

Απόδειξη

Προφανώς ισχύει $PV = \mathcal{MC}(DV) = \mathcal{MC}(DV_2 \cup \mathbf{d}_c)$, αφού οι ακολουθίες DV και $DV_2 \cup \mathbf{d}_c$ περιέχουν τα ίδια στοιχεία. Ο τελεστής $\mathcal{MC}(.)$ απλώς τις αναδιατάσσει. Για τον ίδιο λόγο είναι και $\mathcal{MC}(PV + \mathbf{d}_c) = \mathcal{MC}(\langle DV_2 + \mathbf{d}_c \rangle \cup 2\mathbf{d}_c)$. Από αυτά προκύπτει ότι:

$$\begin{aligned} \mathcal{MC}(\langle PV + \mathbf{d}_c \rangle \cup DV_2) &= \mathcal{MC}(\langle \langle DV_2 + \mathbf{d}_c \rangle \cup 2\mathbf{d}_c \rangle \cup DV_2) \\ &= \mathcal{MC}(DV_2 \cup 2\mathbf{d}_c) \\ &= \mathcal{MC}(DV_{UET}). \end{aligned}$$

Από την πρώτη εξίσωση στη δεύτερη πηγαίνουμε λόγω του τελεστή $\mathcal{MC}(.)$. Η εφαρμογή του υπαγορεύει την απόρριψη των σημείων $\langle DV_2 + \mathbf{d}_c \rangle$ έναντι των $2\mathbf{d}_c$ και DV_2 , λόγω του ότι περιέχονται στον χώρο που τα τελευταία ορίζουν. \dashv

Λήμμα 6.2 *Ta minimum cover των ακολουθιών $\langle DV_{UET} + \mathbf{d}_c \rangle \cup \langle PV + DV_2 \rangle$ και $\langle PV + DV_2 \rangle$ συμπίπτουν επίσης:*

$$\mathcal{MC}(\langle DV_{UET} + \mathbf{d}_c \rangle \cup \langle PV + DV_2 \rangle) = \mathcal{MC}(PV + DV_{UET}).$$

Απόδειξη

Προφανώς ισχύει ότι $\mathcal{MC}(DV_{UET} + \mathbf{d}_c) = \mathcal{MC}(\langle DV_2 + \mathbf{d}_c \rangle \cup 3\mathbf{d}_c)$ και $\mathcal{MC}(PV + DV_2) = \mathcal{MC}(\langle DV_2 \cup \mathbf{d}_c \rangle + DV_2) = \mathcal{MC}(\langle DV_2 + DV_2 \rangle \cup \langle DV_2 + \mathbf{d}_c \rangle)$, όπως και παραπάνω. Από αυτά προκύπτει ότι:

$$\begin{aligned} \mathcal{MC}(\langle DV_{UET} + \mathbf{d}_c \rangle \cup \langle PV + DV_2 \rangle) &= \mathcal{MC}(\langle DV_2 + \mathbf{d}_c \rangle \cup 3\mathbf{d}_c \cup \langle DV_2 + DV_2 \rangle \cup \langle DV_2 + \mathbf{d}_c \rangle) \\ &= \mathcal{MC}(\langle DV_2 + \mathbf{d}_c \rangle \cup \langle DV_2 + DV_2 \rangle \cup 3\mathbf{d}_c) \end{aligned} \quad (6.3)$$

Ισχύει επίσης, ότι:

$$\begin{aligned}
 \mathcal{MC}(PV + DV_{UET}) &= \mathcal{MC}(\langle DV_2 \cup \mathbf{d}_c \rangle + \langle DV_2 \cup 2\mathbf{d}_c \rangle) \\
 &= \mathcal{MC}(\langle DV_2 + DV_2 \rangle \cup \langle DV_2 + \mathbf{d}_c \rangle \cup \langle DV_2 + 2\mathbf{d}_c \rangle \cup 3\mathbf{d}_c) \\
 &= \mathcal{MC}(\langle DV_2 + \mathbf{d}_c \rangle \cup \langle DV_2 + DV_2 \rangle \cup 3\mathbf{d}_c)
 \end{aligned} \tag{6.4}$$

Τα δεύτερα μέλη των εξισώσεων 6.3 και 6.4 συμπίπτουν, άρα ισούνται και τα πρώτα. \dashv

Χρησιμοποιώντας τα παραπάνω λήματα, οι εκφράσεις 6.2 μπορούν να μετασχηματιστούν, σύμφωνα με το θεώρημα που ακολουθεί.

Θεώρημα 6.1 Θεωρούμε φωλιασμένο βρόχο, με ακολουθία διανυσμάτων εξάρτησης $DV = \langle \mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m \rangle$ και ακολουθία των διανυσμάτων κυματομορφής $PV = \langle \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k \rangle$. Το περίγραμμα pat_t του σχήματος κυματομορφής για τη χρονική στιγμή t , δίνεται από τις σχέσεις:

$$\begin{aligned}
 pat_0 &= \langle (0, 0) \rangle \\
 pat_{2i+1} &= \mathcal{MC}(pat_{2i} + PV), \quad t = 2i + 1 \\
 pat_{2i+2} &= \mathcal{MC}(pat_{2i} + DV_{UET}), \quad t = 2i + 2
 \end{aligned} \tag{6.5}$$

Απόδειξη

Θα ακολουθήσουμε την επαγωγική μέθοδο. Ελέγχουμε εάν ισχύει για $i = 0$. Λόγω των Λημμάτων 6.1 και 6.2, οι εκφράσεις ορισμού των περιγραμμάτων στην περίπτωση UET-UCT (σχέσεις 6.2) δίνουν τα εξής:

$$\begin{aligned}
 pat_1 &= \mathcal{MC}(pat_0 + PV) = PV \\
 pat_2 &= \mathcal{MC}(\langle pat_0 + DV_2 \rangle \cup \langle pat_1 + \mathbf{d}_c \rangle) \\
 &= \mathcal{MC}(DV_2 \cup \langle PV + \mathbf{d}_c \rangle) \\
 &= \mathcal{MC}(DV_{UET}) \\
 &= \mathcal{MC}(pat_0 + DV_{UET})
 \end{aligned} \tag{6.6}$$

Θεωρούμε τώρα ότι ισχύει για $i = k - 1$, ότι αληθεύουν δηλαδή οι παραπάνω εκφράσεις για τα pat_{2i+1} και pat_{2i+2} :

$$pat_{2k-1} = \mathcal{MC}(pat_{2(k-1)} + PV) \tag{6.7}$$

$$pat_{2k} = \mathcal{MC}(pat_{2(k-1)} + DV_{UET}) \tag{6.8}$$

Θα αποδείξουμε ότι ισχύουν και για $i = k$, ότι δηλαδή:

$$pat_{2k+1} = \mathcal{MC}(pat_{2k} + PV) \quad (6.9)$$

$$pat_{2k+2} = \mathcal{MC}(pat_{2k} + DV_{UET}) \quad (6.10)$$

Από τις εκφράσεις 6.2 και χρησιμοποιώντας τις παραπάνω, θεωρούμενες ως αληθείς προτάσεις, έχουμε:

$$\begin{aligned} pat_{2k+1} &= \mathcal{MC}(\langle pat_{2k-1} + DV_2 \rangle \cup \langle pat_{2k} + \mathbf{d}_c \rangle) \\ &= \mathcal{MC}(\langle \langle pat_{2(k-1)} + PV \rangle + DV_2 \rangle \cup \langle \langle pat_{2(k-1)} + DV_{UET} \rangle + \mathbf{d}_c \rangle) \\ &= \mathcal{MC}(\langle pat_{2(k-1)} + PV \rangle + DV_{UET}) -\text{λόγω του Λήμματος 6.1} \\ &= \mathcal{MC}(\langle pat_{2(k-1)} + DV_{UET} \rangle + PV) \\ &= \mathcal{MC}(pat_{2k} + PV) \end{aligned} \quad (6.11)$$

$$\begin{aligned} pat_{2k+2} &= \mathcal{MC}(\langle pat_{2k} + DV_2 \rangle \cup \langle pat_{2k+1} + \mathbf{d}_c \rangle) \\ &= \mathcal{MC}(\langle pat_{2k} + DV_2 \rangle \cup \langle \langle pat_{2k} + PV \rangle + \mathbf{d}_c \rangle) -\text{λόγω του Λήμματος 6.2} \\ &= \mathcal{MC}(pat_{2k} + DV_{UET}) \end{aligned} \quad (6.12)$$

Αφού ισχύει και για $i = k$, θα ισχύει και για κάθε $i \in \mathbb{N}$, δηλαδή το θεώρημα επαληθεύεται. \dashv

Το θεώρημα αυτό αποδεικνύει ότι τα περιγράμματα των σχημάτων κυματομορφής μπορούν να εκφραστούν αναλυτικά και στην περίπτωση των προβλημάτων UET-UCT. Επιπλέον, τα περιγράμματα που αντιστοιχούν σε περιττή χρονική στιγμή, συμπίπτουν με τα περιγράμματα που θα προέκυπταν από υποθετικό UET αλγόριθμο, με ακολουθία διανυσμάτων εξάρτησης την εξής:

$$DV_{UET} = \langle \mathbf{d}_1, \dots, \mathbf{d}_{c-1}, 2\mathbf{d}_c, \mathbf{d}_{c+1}, \dots, \mathbf{d}_m \rangle.$$

Αυτήν δηλαδή, η οποία αντί του διανύσματος χωρίς κόστος επικοινωνίας \mathbf{d}_c , περιέχει το πολλαπλάσιο του $2\mathbf{d}_c$.

Για ένα δεδομένο UET-UCT πρόβλημα, το UET πρόβλημα, το οποίο περιέχει όλα τα διανύσματα εξάρτησης του αρχικού, εκτός του \mathbf{d}_c και αντί αυτού περιέχει το $2\mathbf{d}_c$, καλείται *ισοδύναμο UET πρόβλημα* του αρχικού.

6.3 Αναγωγή UET-UCT Προβλημάτων σε Ισοδύναμα UET

Λόγω του θεωρήματος της προηγούμενης πραγράφου, για κάθε UET-UCT αλγόριθμο φωλιασμένων βρόχων, μπορούμε να βρούμε τον ισοδύναμο UET. Η έννοια της ισοδυναμίας έγκειται στην ισότητα που παρουσιάζουν τα περιγράμματα των κυματομορφών της εκτέλεσης τους, κατά τις άρτιες χρονικές στιγμές του UET-UCT προβλήματος. Το περίγραμμα κάθε χρονικής στιγμής $t_{uct} = 2i$ του UET-UCT προβλήματος, ταυτίζεται με αυτό της χρονικής στιγμής $t_{equiv} = i$ του ισοδυνάμου UET.

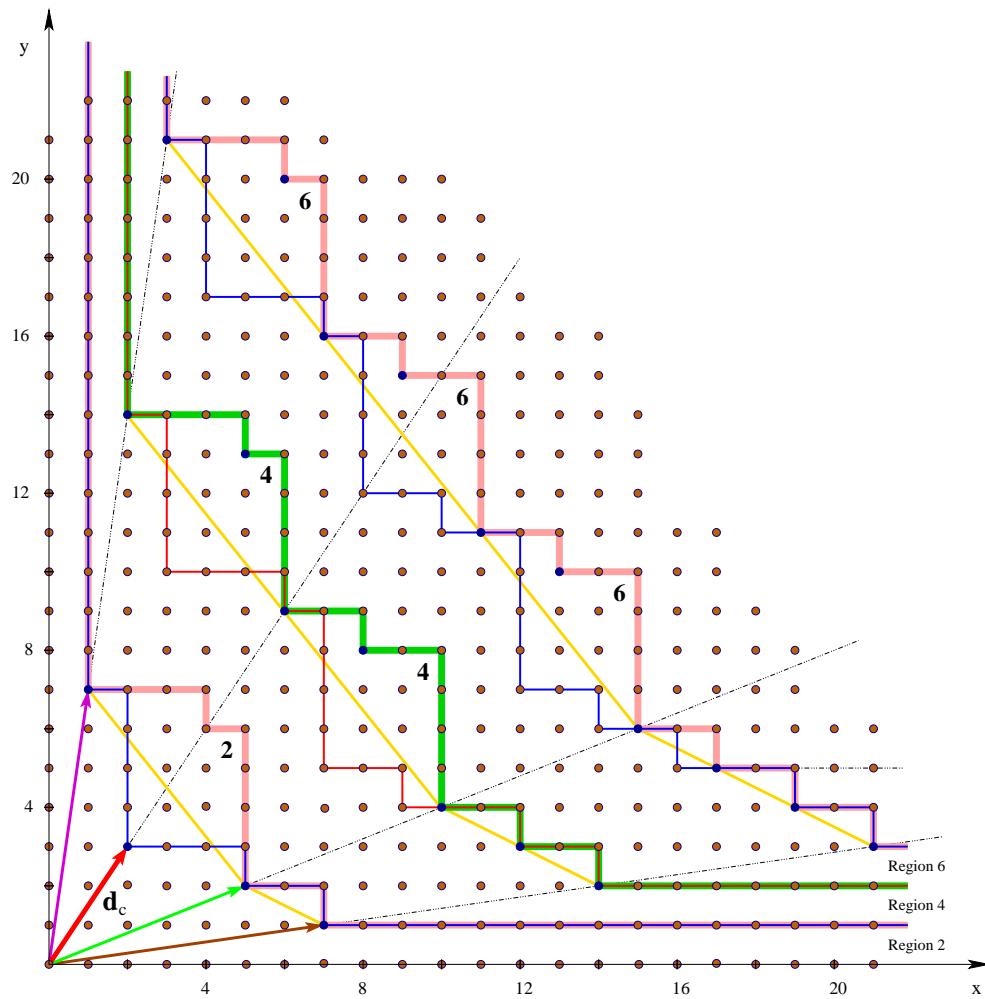
Για να γίνει αυτό καλύτερα κατανοητό, αντιπαραβάλλετε τα σχήματα 6.3 και 6.4. Στο πρώτο θεωρούμε ένα UET-UCT πρόβλημα: σημειώνονται οι κυματομορφές διάδοσης της εκτέλεσης και τα περιγράμματα τους, για τις πρώτες έξι χρονικές στιγμές. Σημειώνεται το διάνυσμα εξάρτησης \mathbf{d}_c , το οποίο αποτελεί το διάνυσμα εξάρτησης που δεν επιφέρει κόστος επικοινωνίας. Στο δεύτερο σχήμα θεωρούμε την εκτέλεση του ισοδυνάμου UET προβλήματος, με τα αντίστοιχα σχήματα κυματομορφής και τα περιγράμματα τους. Αντί του \mathbf{d}_c περιλαμβάνεται στα διανύσματα εξάρτησης το $2\mathbf{d}_c$.

Παρατηρούμε την ισοδυναμία των περιγραμμάτων των $2i$ χρονικών στιγμών του προβλήματος UET-UCT, με τα i περιγράμματα του UET προβλήματος, όπου $i = 1, 2, 3, \dots$. Αυτή η ισοδυναμία υποδηλώνει ότι μπορούμε να δρομολογήσουμε την εκτέλεση οποιουδήποτε προβλήματος UET-UCT, σύμφωνα με το ισοδύναμό του. Για το ισοδύναμο UET πρόβλημα, μπορούμε να χρησιμοποιήσουμε οποιαδήποτε μέθοδο της βιβλιογραφίας. Θα περιοριστούμε στον γεωμετρικό αλγόριθμο δρομολόγησης που παρουσιάστηκε στο προηγούμενο κεφάλαιο. Κάθε σημείο του χώρου δεικτών, μπορεί να δρομολογηθεί ανάλογα με τον κώνο και την περιοχή που ανήκει.

Δοθέντος ενός σημείου j του UET-UCT αλγορίθμου, υπολογίζουμε αρχικά τη χρονική στιγμή $t_{equiv} = i$, κατά την οποία θα εκτελείτο αν είχαμε το ισοδύναμο UET πρόβλημα. Η χρονική αυτή στιγμή αντιστοιχεί με τις χρονικές στιγμές $t_{uct} = 2i$ και $t'_{uct} = 2i + 1$, του αρχικού μας προβλήματος. Απαιτείται ένας επιπλέον έλεγχος, ο οποίος θα διαχωρίζει τις δύο αυτές περιπτώσεις για το σημείο j .

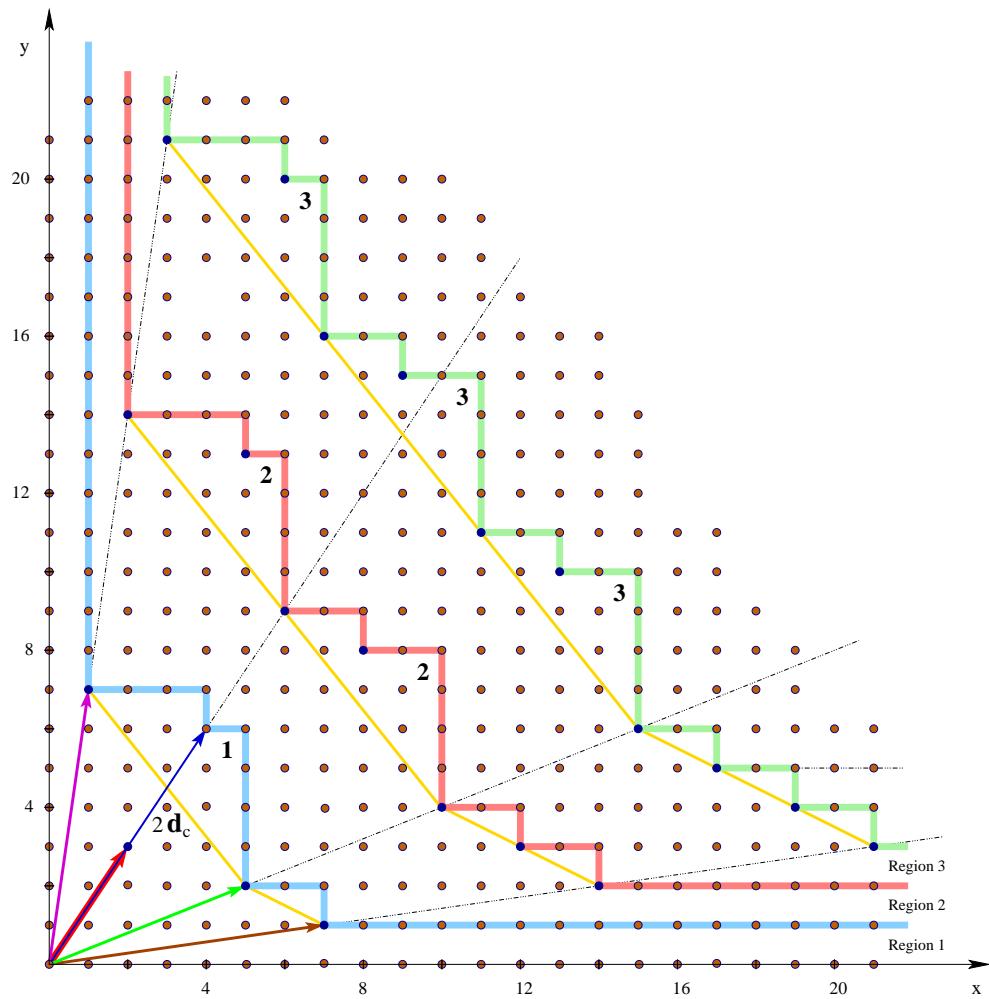
Κατόπιν αυτών, η διαδικασία δρομολόγησης ενός σημείου στο UET-UCT πρόβλημα, περιλαμβάνει τα εξής τρία βήματα, αντί των δύο του UET:

- εντοπισμός της περιοχής t_{equiv} στην οποία ανήκει το σημείο στο ισοδύναμο UET πρόβλημα,
- μέσω ενός επιπλέον ελέγχου, καθορισμός της ακριβούς χρονικής στιγμής $t_{uct} = 2t_{equiv}$ ή $t'_{uct} = 2t_{equiv} + 1$, κατά την οποία το εν λόγω σημείο πρέπει να δρομολογηθεί.



Σχήμα 6.3: Η διάδοση της εκτέλεσης ενός προβλήματος UET-UCT, με διανύσματα εξάρτησης $\mathbf{d}_1 = (1, 7)$, $\mathbf{d}_2 = (2, 3)$, $\mathbf{d}_3 = (5, 2)$ και $\mathbf{d}_4 = (7, 1)$, με $\mathbf{d}_c = \mathbf{d}_2$. Η δρομολόγηση του ισοδυνάμου UET προβλήματος παρουσιάζεται στο σχήμα 6.4.

- τέλος, δρομολόγηση του σημείου κατά την υπολογισθείσα χρονική στιγμή t_{uct} ή t'_{uct} .



Σχήμα 6.4: Ισοδύναμο UET για το πρόβλημα του σχήματος 6.3. Τα διανύσματα εξάρτησης είναι τα $\mathbf{d}_1 = (1, 7)$, $\mathbf{d}_2 = 2\mathbf{d}_c = (4, 6)$, $\mathbf{d}_3 = (5, 2)$ και $\mathbf{d}_4 = (7, 1)$. Παρατηρούμε την ισοδυναμία των i -περιγραμμάτων αυτού του αλγορίθμου, με τα αντίστοιχα $2i$ του UET-UCT, για $i = 1, 2, 3, \dots$

6.4 Δρομολόγηση Προβλημάτων UET-UCT

Αυτό που χρειαζόμαστε είναι ένας επιπλέον έλεγχος, ο οποίος θα διαχωρίζει ένα σημείο j που ανήκει στην άρτια χρονική στιγμή $t_{uct} = 2t_{equiv}$, από το j' το οποίο ανήκει στην περιττή χρονική στιγμή $t'_{uct} = 2t_{equiv} + 1$.

Χρησιμοποιώντας τα αποτελέσματα του Θεωρήματος 6.1, μπορούμε να εκφράσουμε το pat_{2k+1} ως συνάρτηση του pat_{2i} και του pat_{2i+2} : ως συνάρτηση δηλαδή, μόνο άρτιων περιγραμμάτων. Συγκεκριμένα, είναι:

$$\begin{aligned}
 pat_{2i+1} &= \mathcal{MC}(pat_{2i} + PV) = \mathcal{MC}(\langle pat_{2i} + DV_2 \rangle \cup \langle pat_{2i} + \mathbf{d}_c \rangle) \\
 &= \mathcal{MC}(\langle pat_{2i} + DV_2 \rangle \cup \langle pat_{2i} + \mathbf{d}_c \rangle \cup \langle pat_{2i} + 2\mathbf{d}_c \rangle) \\
 &= \mathcal{MC}(\langle pat_{2i} + DV_2 \rangle \cup \langle pat_{2i} + 2\mathbf{d}_c \rangle \cup \langle pat_{2k} + \mathbf{d}_c \rangle) \\
 &= \mathcal{MC}(\langle pat_{2i} + DV_{UET} \rangle \cup \langle pat_{2i} + \mathbf{d}_c \rangle) \\
 &= \mathcal{MC}(\mathcal{MC}(\langle pat_{2i} + DV_{UET} \rangle) \cup \langle pat_{2i} + \mathbf{d}_c \rangle) \\
 &= \mathcal{MC}(pat_{2i+2} \cup \langle pat_{2i} + \mathbf{d}_c \rangle)
 \end{aligned} \tag{6.13}$$

Από την έκφραση αυτή υποδηλώνεται ότι αν το σημείο j ανήκει στην περιοχή t_{equiv} του ισοδυνάμου UET προβλήματος, θα ανήκει στη χρονική στιγμή $t'_{uct} = 2t_{equiv} + 1$, αν και μόνο αν το σημείο $j - \mathbf{d}_c$ ανήκει επίσης στην περιοχή t_{equiv} . Αν δεν συμβαίνει αυτό, τότε ανήκει στη χρονική στιγμή $t_{uct} = 2t_{equiv}$ του αρχικού προβλήματος. Το γεγονός αυτό αναπαρίσταται ξεκάθαρα στο σχήμα 6.5.

Ο αλγόριθμος που ακολουθεί πραγματοποιεί τη δρομολόγηση που περιγράψαμε, δουλεύοντας σε προβλήματα UET-UCT. Αποτελεί αναθεώρηση του Αλγορίθμου 5.2. Ο χώρος δεικτών τμηματίζεται σε κώνους και περιοχές, όπως συνάρτηση τα διανύσματα εξάρτησης του ισοδυνάμου UET προβλήματος. Το προς δρομολόγηση σημείο εντοπίζεται στον κώνο και την περιοχή στα οποία ανήκει, στο ισοδύναμο πρόβλημα. Στη συνέχεια, πραγματοποιείται ο επιπλέον έλεγχος για τον καθορισμό της ακριβούς χρονικής στιγμής, κατά την οποία θα δρομολογηθεί το σημείο, στο αρχικό πρόβλημα.

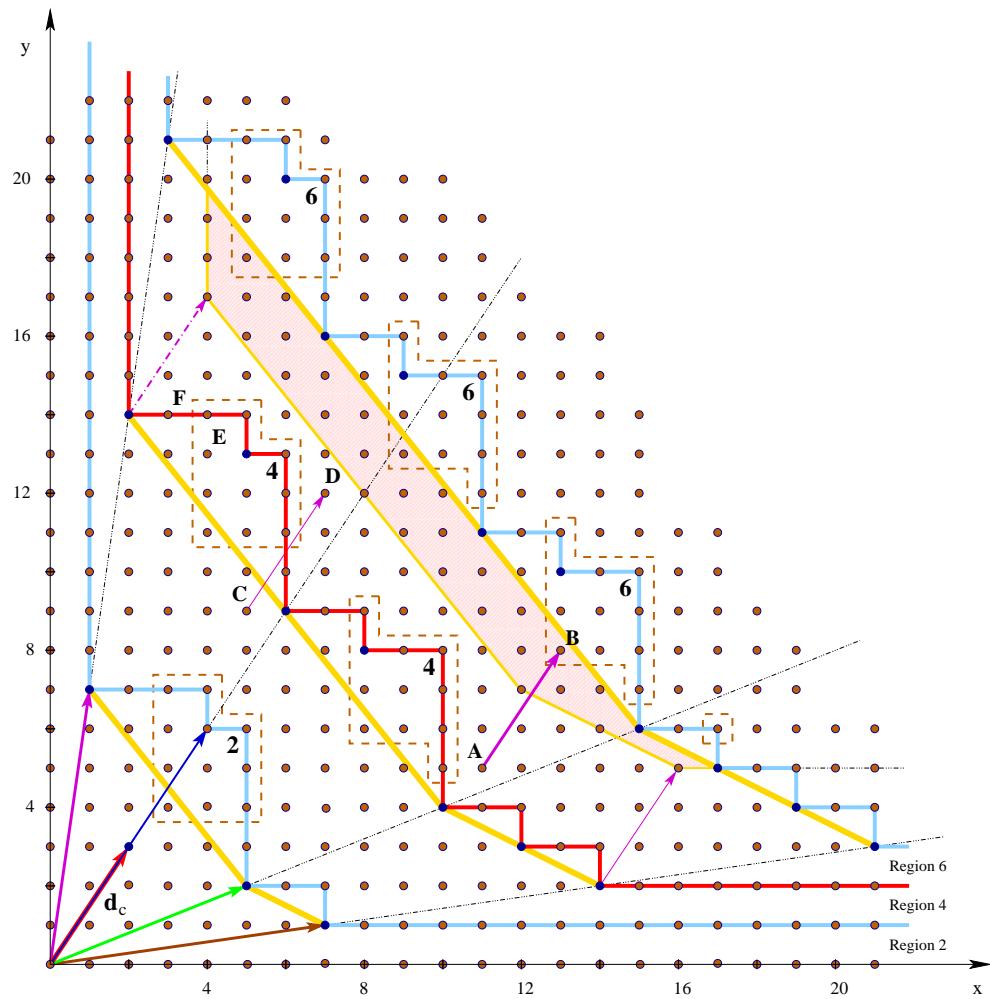
Αλγόριθμος 6.1

Δρομολόγηση ενός UET-UCT βρόχου

INPUT: Η ακολουθία των εξαρτήσεων $DV = \langle \mathbf{d}_1, \dots, \mathbf{d}_m \rangle$, το διάνυσμα \mathbf{d}_c και ένα σημείο j του χώρου.

OUTPUT: Η χρονική στιγμή $t_{uct}(j)$ κατά την οποία θα εκτελεστεί το σημείο j .

- Το διάνυσμα επικοινωνίας \mathbf{d}_c αφαιρείται από το DV και στη θέση του τοποθετείται το $2\mathbf{d}_c$. δημιουργείται έτσι η ακολουθία εξαρτήσεων $DV_{UET} = \langle \mathbf{d}_1, \dots, \mathbf{d}_{c-1}, 2\mathbf{d}_c, \mathbf{d}_{c+1}, \dots, \mathbf{d}_m \rangle$ -ισοδύναμο UET πρόβλημα.
- Εφαρμόζεται ο Αλγόριθμος 5.2 στην ακολουθία DV_{UET} για το σημείο j . υπολογίζεται έτσι η χρονική στιγμή $t_{equiv}(j)$. Οι υποψήφιες χρονικές στιγμές για το σημείο j είναι η $2t_{equiv}(j) - 1$ και η $2t_{equiv}(j)$.



Σχήμα 6.5: Χωρισμός περιοχών προβλήματος UCT και δρομολόγηση του. Τα διανύσματα εξάρτησης είναι τα $\mathbf{d}_1 = (1, 7)$, $\mathbf{d}_2 = (2, 3)$, $\mathbf{d}_3 = (5, 2)$ ανδ $\mathbf{d}_5 = (7, 1)$. Κόστος επικοινωνίας δεν υπάρχει κατά τη διεύθυνση του \mathbf{d}_2 ($\mathbf{d}_c = \mathbf{d}_2$). Οι ανοιχτές πολυγωνικές γραμμές οριοθετούν τα σύνολα των σημείων $\mu \in ECT \leq 2k$, όπου $k = 1, 2, 3, \dots$. Τα πολύγωνα με τη διακεκομμένη γραμμή, περικλείουν τα σημεία $\mu \in ECT = 2k$: το γραμμοσκιασμένο πολύγωνο περιέχει τα σημεία τα οποία θα εκτελεστούν κατά την χρονική στιγμή $t_{uct} = 6$.

3. Ο Αλγόριθμος 5.2 εφαρμόζεται στην ακολουθία DV_{UET} και στο σημείο $\mathbf{j} - \mathbf{d}_c$: υπολογίζεται έτσι η χρονική στιγμή $t_{equiv}(\mathbf{j} - \mathbf{d}_c)$:

- ▷ αν $t_{equiv}(\mathbf{j} - \mathbf{d}_c) = t_{equiv}(\mathbf{j}) \Rightarrow t_{uct}(\mathbf{j}) = 2t_{equiv}(\mathbf{j})$, ενώ
- ▷ αν $t_{equiv}(\mathbf{j} - \mathbf{d}_c) \neq t_{equiv}(\mathbf{j}) \Rightarrow t_{uct}(\mathbf{j}) = 2t_{equiv}(\mathbf{j}) - 1$.

Ο αλγόριθμος επεξηγείται στο σχήμα 6.5. Οι περιοχές αναφέρονται στις άρτιες χρονικές στιγμές και μόνο, αφού αντιστοιχούν με τα περιγράμματα των άρτιων χρονικών στιγμών. Τα σημεία B και D εντοπίζονται στην περιοχή 6, χρησιμοποιώντας τον Αλγόριθμο 5.2. Λόγω αυτού, οι πιθανές χρονικές στιγμές για την εκτέλεση τους είναι η 5η και η 6η.

Όμως, το σημείο $A = B - \mathbf{d}_c$ βρίσκεται στην προηγούμενη περιοχή, ενώ το σημείο $D = C - \mathbf{d}_c$ βρίσκεται στην ίδια περιοχή. Έτσι, το σημείο C θα εκτελεστεί τη χρονική στιγμή $t_{uct}(C) = 5$ και το σημείο B τη χρονική στιγμή $t_{uct}(B) = 6$.

Η ιδέα πίσω από αυτή τη “μετακίνηση” του pat_{2k} κατά \mathbf{d}_c είναι η εξής: παρά το γεγονός ότι το \mathbf{d}_c δεν επιφέρει κόστος επικοινωνίας, δεν πάγει να υπάρχει ως εξάρτηση, η οποία μπορεί να διαχωρίζει δύο διαδοχικές χρονικές στιγμές. Αυτό φαίνεται ξεκάθαρα στο σχήμα μας. Δεν είναι υπολογιστικά ορθό να παρουσιάζεται μια περιοχή μεγαλύτερη (γεωμετρικά) ως προς κάποια διάσταση, από οποιοδήποτε διάνυσμα εξάρτησης. Αυτό θα είχε ως επακόλουθο δύο διαδοχικά σημεία ως προς αυτό το διάνυσμα, να ανήκουν στην ίδια χρονική περιοχή, το οποίο προφανώς είναι άτοπο.

Οι περιοχές που ορίζονται στα προβλήματα UET-UCT έχουν εύρος μεγαλύτερο του \mathbf{d}_c . Τον επιμέρους χωρισμό τους σε δύο χρονικές στιγμές, αναλαμβάνει το διάνυσμα αυτό, γιατί είναι το μόνο που δεν περιέχεται στον μετασχηματισμένο UET αλγόριθμο.

6.5 Εγκυρότητα & Βελτιστοποίηση

Η εγκυρότητα του αλγορίθμου μας αποδεικνύεται από τη μη διαταραχή των εξαρτήσεων. Από τη μη ύπαρξη δηλαδή, περιοχών με εύρος μικρότερο από οποιαδήποτε εξάρτηση. Αυτό επιτυγχάνεται λαμβάνοντας υπ' όψιν όλες τις εξαρτήσεις του UET-UCT αλγορίθμου για τον καθορισμό των περιοχών. Όλες οι εξαρτήσεις πλην του \mathbf{d}_c –οι οποίες σχηματίζουν το ισοδύναμο UET πρόβλημα– εξασφαλίζουν ότι στις περιοχές που ορίζονται δεν υπάρχει περίπτωση να συνυπάρχουν σημεία τα οποία να εξαρτώνται μεταξύ τους από οποιαδήποτε από αυτές.

Η εξάρτηση \mathbf{d}_c χωρίζει τις σχηματιζόμενες περιοχές σε δύο υπο-περιοχές, μεταξύ των οποίων εξασφαλίζεται η ανεξαρτησία και ως προς \mathbf{d}_c .

Ας δούμε τώρα το πόσο βέλτιστη είναι η δρομολόγηση του αλγορίθμου μας. Τα σημεία που αντιστοιχούν στις άρτιες χρονικές στιγμές, καθυστερούν το πολύ κατά μία χρονική στιγμή. Αυτό προκύπτει λόγω του ισοδυνάμου UET αλγορίθμου, στον οποίο η μέγιστη καθυστέρηση από τη βέλτιστη είναι το πολύ μια επιπλέον χρονική στιγμή, όπως δείξαμε σε προηγούμενο κεφάλαιο (σημείο E του σχήματος 6.5).

Τα σημεία των περιττών χρονικών στιγμών καθυστερούν το πολύ κατά δύο χρονικές στιγμές. Αυτό προκύπτει ως εξής. Έστω το σημείο F του σχήματος 6.5. Η εκτέλεση του σημείου αυτού, λόγω της εφαρμογής της γεωμετρικής δρομολόγησης, καθυστερεί κατά μία χρονική στιγμή στο ισοδύναμο UET πρόβλημα: Θα μπορούσε να εκτελεστεί κατά τη χρονική στιγμή $t = 2$ και εκτελείται κατά τη χρονική στιγμή $t' = 3$ –στο ισοδύναμο UET πρόβλημα πάντα. Λόγω της ύπαρξης του \mathbf{d}_c όμως, καθυστερεί μια επιπλέον χρονική στιγμή στο αρχικό UET-UCT πρόβλημα. Αν δηλαδή, ένα τυχαίο σημείο του αρχικού αλγορίθμου, εκτελεστεί κατά τη βέλτιστη χρονική στιγμή στο ισοδύναμο UET πρόβλημα, το πολύ να καθυστερήσει λόγω του \mathbf{d}_c μια επιπλέον χρονική στιγμή. Συνολικά δηλαδή, μια χρονική στιγμή. Αν όμως, έχει ήδη καθυστερήσει κατά μία χρονική στιγμή, η εκτέλεση του μπορεί να επιβραδυνθεί για ακόμα μία χρονική στιγμή λόγω του \mathbf{d}_c . Συνολικά δηλαδή, κατά δύο χρονικές στιγμές.

Έτσι, ο αλγόριθμος δρομολόγησης που προτείνεται για τα προβλήματα UET-UCT είναι ασυμπτωτικά βέλτιστος, με σταθερή καθυστέρηση το πολύ δύο χρονικών στιγμών.

6.6 Πολυπλοκότητα

Η πολυπλοκότητα της δρομολόγησης προβλημάτων φωλιασμένων βρόχων με μη-μηδενικό κόστος επικοινωνίας –Αλγόριθμος 6.1– είναι ουσιαστικά η ίδια με αυτή της γεωμετρικής δρομολόγησης χωρίς κόστος επικοινωνίας –Αλγόριθμος 5.2. Αυτό γιατί ο ένας αλγόριθμος αποτελεί τη διπλή εφαρμογή του άλλου.

Για την πολυπλοκότητα $Cx(6.1)$ της δρομολόγησης προβλημάτων με κόστος επικοινωνίας μη-μηδενικό είναι λοιπόν:

$$Cx(6.1) = \Theta(m \log m) \quad (6.14)$$

όπου m το πλήθος των διανυσμάτων εξάρτησης του αλγορίθμου.

Κεφάλαιο 7

Γεωμετρική Δρομολόγηση σε Περισσότερες Διαστάσεις

Για τη γενίκευση της γεωμετρικής δρομολόγησης σε προβλήματα μεγαλυτέρων διαστάσεων, αρκεί η γενίκευση των αλγορίθμων του προηγουμένου κεφαλαίου στις n διαστάσεις. Το κυρίως πρόβλημα έγκειται σε δύο σημεία. Το πρώτο αφορά τον προσδιορισμό των διανυσμάτων των κώνων (cone vectors) στη γενικότερη περίπτωση των n διαστάσεων. Η μέθοδος προσδιορισμού που παρουσιάστηκε στο Κεφάλαιο 5, αφορούσε την περίπτωση των δύο διαστάσεων, πετυχαίνοντας βέβαια πολύ χαμηλή πολυπλοκότητα.

Για την επίλυση αυτού του προβλήματος καταφεύγουμε σε μια μέθοδο προσδιορισμού του κυρτού περιγράμματος συνόλου σημείων στο n -διάστατο χώρο, η οποία είναι γνωστή στη βιβλιογραφία με το όνομα *QuickHull* [BDH96]. Η μέθοδος αυτή προέρχεται από το χώρο της υπολογιστικής γεωμετρίας: παρουσιάζει τη βέλτιστη πολυπλοκότητα και μας επιτρέπει τον προσδιορισμό των διανυσμάτων των κώνων στη γενικότερη περίπτωση των n διαστάσεων. Για τους κώνους που προκύπτουν, προσδιορίζονται επίσης και τα βέλτιστα υπερεπίπεδα εκτέλεσης, γεγονός το οποίο πραγματοποιείται για πρώτη φορά στη βιβλιογραφία για προβλήματα n διαστάσεων.

Το δεύτερο επιμέρους πρόβλημα που ανακύπτει είναι αυτό της δρομολόγησης των επαναλήψεων του χώρου δεικτών που δεν ανήκουν στους σχηματιζόμενους κώνους. Αρχικά προσδιορίζεται το περίγραμμα των σχηματιζόμενων κώνων. Κατόπιν αυτού, τα σημεία τα οποία δεν ανήκουν σ' αυτούς προσδιορίζονται αποδοτικά. Τέλος, η δρομολόγηση διαφοροποιείται για τα σημεία που ανήκουν στους κώνους και για τα εξωτερικά.

7.1 Ορολογία & Ορισμοί

Στην περίπτωση των n διαστάσεων, δεν μας αρκούν τα διανύσματα των κώνων για τον ορισμό των κώνων και των περιοχών του προβλήματός μας. Αυτό γιατί στις μεγαλύτερες διαστάσεις, για τον ορισμό των κώνων, δεν αρκεί η διάταξη των διανυσμάτων των κώνων με βάση απλώς την κλίση τους. Δεν αρκεί π.χ. τρία διανύσματα κώνων να είναι συνεχόμενα ως προς μια από τις συντεταγμένες τους ώστε να ορίζουν ένα κώνο.

Για το λόγο αυτό απαιτείται η διατήρηση ορισμένων επιπλέον πληροφοριών, όπως το πλήθος των κώνων, η ακολουθία των διανυσμάτων που ορίζουν τον καθένα απ' αυτούς, οι πλευρές (facets) που σχηματίζονται, κ.λπ. Θα περιοριστούμε στο ελάχιστο πλήθος πληροφοριών που απαιτούνται με τη χρήση αυτών προσδιορίζονται εύκολα και οι υπόλοιπες.

Οι ορισμοί που ακολουθούν αποτελούν γενίκευση των αντίστοιχων ορισμών για τις δύο διαστάσεις. Έστω η ακολουθία DV των διανυσμάτων εξάρτησης $DV = \langle \mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m \rangle$. Ορίζονται τα εξής:

Κώνος: Ο κυρτός υποχώρος ο οποίος σχηματίζεται από i διανύσματα $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_q \in \mathbb{N}^n$, καλείται κώνος και συμβολίζεται ως $Con(\mathbf{a}_1, \dots, \mathbf{a}_q)$. Τυπικότερα, ορίζεται ως:

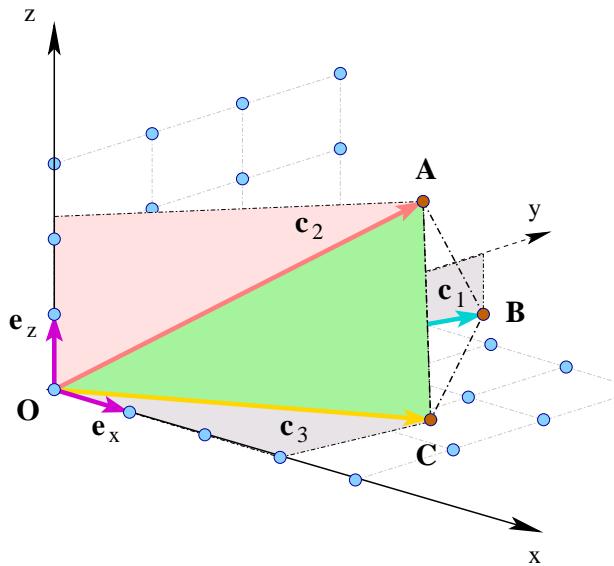
$$Con(\mathbf{a}_1, \dots, \mathbf{a}_q) = \{ \mathbf{j} \in \mathbb{N}^n \mid \mathbf{j} = \lambda_1 \mathbf{a}_1 + \dots + \lambda_q \mathbf{a}_q, \text{ όπου } \lambda_1, \dots, \lambda_q \geq 0 \}.$$

Σε ορισμένες περιπτώσεις, όταν τα διανύσματα που ορίζουν τον κώνο δεν παρουσιάζουν ιδιαίτερο ενδιαφέρον, χρησιμοποιούμε τον συμβολισμό Con_i για να δηλώσουμε απλούστερα τον κώνο i .

Μη-Τετριμμένος Κώνος: Ως μη-τετριμμένο κώνο (*non-trivial cone*) καλούμε τον κώνο που ορίζεται αποκλειστικά από διανύσματα εξάρτησης.

Τετριμμένος Κώνος: Ως τετριμμένο κώνο (*trivial cone*) καλούμε τον κώνο που ορίζεται από διανύσματα εξάρτησης και τουλάχιστον ένα μοναδιαίο διάνυσμα των αξόνων. Ο κώνος στην περίπτωση αυτή συμβολίζεται ως $TrCon(\mathbf{a}_1, \dots, \mathbf{a}_q, \mathbf{e}_s, \dots, \mathbf{e}_t)$, όπου τα \mathbf{a}_i είναι διανύσματα εξάρτησης και τα \mathbf{e}_i είναι μοναδιαία διανύσματα αξόνων. Συνολικά αποτελούν τα διανύσματα που σχηματίζουν τον κώνο (σήμα 7.1). Και στην περίπτωση των τετριμμένων κώνων χρησιμοποιούμε, ορισμένες φορές, τον συμβολισμό $TrCon_j$ για να δηλώσουμε τον τετριμμένο κώνο j .

Σύνολο Κώνων: Το σύνολο των μη-τετριμμένων και των τετριμμένων κώνων ονομάζεται σύνολο κώνων και συμβολίζεται ως CS . παρουσιάζει δε τις ακόλουθες ιδιότητες:



Σχήμα 7.1: Απεικονίζεται ο τετριμμένος κώνος $TrCon(\mathbf{c}_2, \mathbf{c}_3, \mathbf{e}_x, \mathbf{e}_z)$ που ορίζεται από τα διανύσματα κώνων \mathbf{c}_2 και \mathbf{c}_3 και τα μοναδιαία διανύσματα \mathbf{e}_x και \mathbf{e}_z .

1. κάθε σημείο του χώρου δεικτών $\mathbf{j} \in J^n$ ανήκει σε κάποιο τετριμμένο κώνο $TrCon_i \in CS$, είτε σε έναν από τους μη-τετριμμένους κώνους $Con_i \in CS$,
2. αν $\mathbf{j} \in Con_i = Con(\mathbf{c}_p, \dots, \mathbf{c}_q)$, τότε $\mathbf{j} = \lambda_p \mathbf{c}_p + \dots + \lambda_q \mathbf{c}_q$ και $\lambda_p + \dots + \lambda_q = maximum$.

Προφανώς, είναι δυνατόν να εκφράσουμε το \mathbf{j} ως γραμμικό συνδυασμό διαφόρων υποσυνόλων του συνόλου των διανυσμάτων εξάρτησης. Όμως, το άθροισμα των συντελεστών για οποιονδήποτε άλλο συνδυασμό δεν θα ξεπερνάει το προαναφερθέν μέγιστο.

Διανύσματα Κώνων: Ως διανύσματα κώνων (*cone vectors*) ορίζονται τα διανύσματα εξάρτησης τα οποία συμμετέχουν στο σχηματισμό του συνόλου των κώνων. Σημειώνουμε ότι τα διανύσματα εξάρτησης με τη μικρότερη συντεταγμένη ως προς κάθε διάσταση, περιλαμβάνονται στα διανύσματα κώνων. Στις n διαστάσεις ($n \geq 3$) δεν έχει νόημα η διάταξη των διανυσμάτων αυτών. Για το λόγο αυτό χρησιμοποιούμε το συμβολισμό του συνόλου, αντί αυτού της ακολουθίας, για την αναπαράσταση τους: $CV = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_r\}$.

Πλευρά: Ως πλευρά (*facet*) στις n διαστάσεις, ορίζεται ένα πολύγωνο με k κορυφές, όπου $k \geq n$ και την εξής επιπλέον ιδιότητα: οποιεσδήποτε n από τις κορυφές του δεν ανήκουν

στην ίδια ευθεία, ενώ ανήκουν στο ίδιο επίπεδο. Μια πλευρά f με k κορυφές $\mathbf{a}_1, \dots, \mathbf{a}_k$ στις n διαστάσεις συμβολίζεται ως $f = \langle \mathbf{a}_1, \dots, \mathbf{a}_k \rangle$. Η σειρά αναγραφής των κορυφών είναι δεξιόστροφη ή αριστερόστροφη. Όταν πρόκειται για μεμονωμένη πλευρά, δεν έχει σημασία. Όταν πρόκειται όμως για πλευρά κάποιου σχήματος, π.χ. κώνου, τότε ακολουθεί τη διάταξη η οποία έχει ως αποτέλεσμα το κάθετο διάνυσμα της πλευράς να κατευθύνεται προς το εσωτερικό του σχήματος.

Εσωτερικές Πλευρές: Θεωρούμε το σύνολο των απολήξεων των διανυσμάτων εξάρτησης και του τερματικού σημείου του χώρου \mathbf{U} . Ως εσωτερικές πλευρές (*inner facets*) ορίζουμε τις πλευρές του κυρτού περιγράμματος του συνόλου αυτού, οι οποίες δεν περιέχουν το τερματικό σημείο \mathbf{U} (σχήμα 7.2). Η ακολουθία των εσωτερικών πλευρών ενός αλγορίθμου ορίζεται ως:

$$\mathcal{IF} = \langle if_1, if_2, \dots \rangle.$$

Εξωτερικές Πλευρές: Θεωρούμε το σύνολο των απολήξεων των διανυσμάτων εξάρτησης και της αρχής των αξόνων \mathbf{O} . Ως εξωτερικές πλευρές (*outer facets*) ορίζουμε τις πλευρές του κυρτού περιγράμματος του συνόλου αυτού, οι οποίες περιέχουν την αρχή των αξόνων \mathbf{O} . Η ακολουθία των εξωτερικών πλευρών για έναν αλγόριθμο ορίζεται ως εξής:

$$\mathcal{OF} = \langle of_1, of_2, \dots \rangle.$$

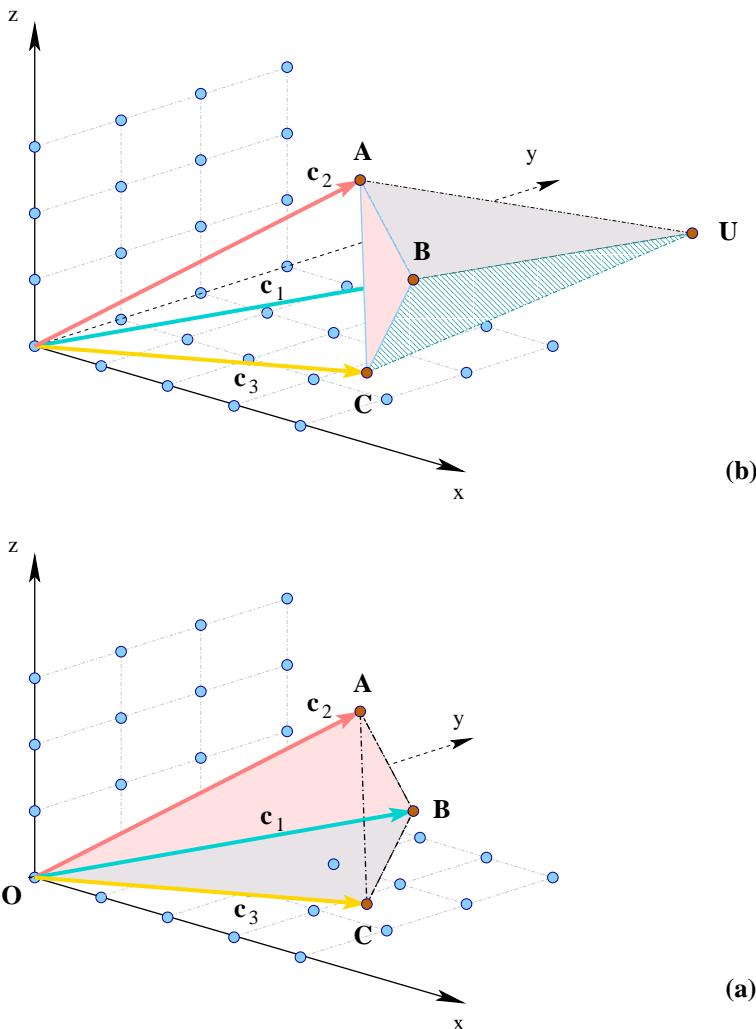
Κάθετο Διάνυσμα Πλευράς: Έστω n γραμμικώς ανεξάρτητα σημεία $\mathbf{p}_1, \dots, \mathbf{p}_n \in J^n$ και $f = \langle \mathbf{p}_1, \dots, \mathbf{p}_n \rangle$, η πλευρά που ορίζεται από αυτά. Το κάθετο μοναδιαίο διάνυσμα στην f ορίζεται ως $\mathbf{n}(\mathbf{p}_1, \dots, \mathbf{p}_n)$. Η φορά του διανύσματος αυτού ακολουθεί τον κανόνα του δεξιόστροφου κοχλία: κατευθύνεται προς το εσωτερικό του σχήματος που ανήκει η πλευρά στην οποία αναφερόμαστε. Το μήκος του είναι μοναδιαίο.

Περιοχή Μη-Τετριμμένου Κώνου: Έστω ο μη-τετριμμένος κώνος $Con_\ell = Con(\mathbf{a}_1, \dots, \mathbf{a}_q)$. Οι περιοχή i του κώνου Con_ℓ καλείται το σύνολο:

$$Reg_i(\mathbf{a}_1, \dots, \mathbf{a}_q) = \{ \mathbf{j} \in Con_\ell \mid \mathbf{j} = \lambda_1 \mathbf{a}_1 + \dots + \lambda_q \mathbf{a}_q, \text{ όπου } i < \lambda_1 + \dots + \lambda_q \leq i+1 \}.$$

Περιοχή Τετριμμένου Κώνου: Έστω ο τετριμμένος κώνος $TrCon_\ell(\mathbf{a}_p, \dots, \mathbf{a}_q, \mathbf{e}_s, \dots, \mathbf{e}_t)$ και έστω $\mathbf{e} = \mathbf{e}_s + \dots + \mathbf{e}_t$. Η περιοχή i του τετριμμένου κώνου $TrCon_\ell$ ορίζεται ως εξής:

$$TrReg_i(\mathbf{a}_p, \dots, \mathbf{a}_q, \mathbf{e}_s, \dots, \mathbf{e}_t) = \{ \mathbf{j} \in TrCon_\ell \mid [\mathbf{j} \cdot \mathbf{n}(\mathbf{a}_p, \dots, \mathbf{a}_q, \mathbf{a}_p + \mathbf{e}, \dots, \mathbf{a}_q + \mathbf{e})] = i \}.$$



Σχήμα 7.2: Εσωτερικές και εξωτερικές πλευρές κώνου στις τρεις διαστάσεις. Για τον κώνο του σχήματος έχουμε μόνο μία εσωτερική πλευρά: την τριγωνική ABC (b). Εξωτερικές πλευρές έχουμε τρεις, επίσης τριγωνικές: OAB , OCB και OCA (a).

Περιοχή: Ως περιοχή R_i ορίζεται η ένωση των περιοχών i όλων των τετριμμένων και των μη-τετριμμένων κώνων.

Ένα παράδειγμα τετριμμένου κώνου βλέπουμε στο σχήμα 7.1, ενώ όλοι οι κώνοι στους οποίους αναφερθήκαμε στα προηγούμενα κεφάλαια ήταν μη-τετριμμένοι. Μη-τετριμμένος είναι

και ο κώνος $Con(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ στο ίδιο σχήμα.

Στο σχήμα 7.2 διαχρίνεται ένα παράδειγμα εσωτερικών και εξωτερικών πλευρών. Απεικονίζεται ένας κώνος ο οποίος σχηματίζεται από τρία διανύσματα. Έχουμε μία μόνο εσωτερική πλευρά, την τριγωνική ABC , ενώ εξωτερικές πλευρές έχουμε τρεις, επίσης τριγωνικές OAB , OBC και OCA .

Γενικά, έχουμε μία εσωτερική πλευρά ανά κώνο. Η πλευρά αυτή έχει κορυφές τις απολήξεις των διανυσμάτων που ορίζουν τον κώνο και ορίζει το βέλτιστο επίπεδο δρομολόγησης για τον εν λόγω κώνο, με τον ίδιο ακριβώς τρόπο όπως στην περίπτωση των δύο διαστάσεων. Σημειώνουμε ότι αν ακολουθηθεί η μέθοδος των υπερεπιπέδων, το σύνολο των πιθανών βέλτιστων υπερεπιπέδων δρομολόγησης Π_i που προκύπτει, είναι αυτό που αποτελείται από τα κάθετα διανύσματα στις εσωτερικές πλευρές.

Οι εξωτερικές πλευρές έχουν βοηθητικό χαρακτήρα. Συμβάλλουν στον καθορισμό των υπερεπιπέδων δρομολόγησης των τετριμένων κώνων. Από αυτές εξαρτάται δηλαδή, η δρομολόγηση των σημείων που δεν ανήκουν σε κανέναν από τους σχηματιζόμενους μη-τετριμένους κώνους, όπως θα φανεί στη συνέχεια.

7.2 Υπολογισμός των Μη-Τετριμμένων Κώνων

Για τη δρομολόγηση απαιτείται ο προσδιορισμός των μη-τετριμένων κώνων. Στην περίπτωση των n διαστάσεων όμως, δεν αρκούν τα διανύσματα των κώνων (cone vectors) για τον καθορισμό τους. Απαιτείται ο προσδιορισμός των εσωτερικών πλευρών (inner facets) που ορίστηκαν παραπάνω.

Ο υπολογισμός των εσωτερικών πλευρών γίνεται χρησιμοποιώντας τον αλγόριθμο κυρτού περιγράμματος QuickHull. Για κάθε εσωτερική πλευρά που υπολογίζεται, σημειώνονται και τα διανύσματα εξάρτησης που σχηματίζουν τον εν λόγω κώνο. Η μέθοδος που παρουσιάζεται αποτελεί γενίκευση της μεθόδου της παραγράφου 5.7.1. Προσδιορίζεται το κυρτό περίγραμμα των απολήξεων των διανυσμάτων αυτών και του τερματικού σημείου του χώρου δεικτών $\mathbf{U} = (i_{1_{max}}, i_{2_{max}}, \dots)$. Από το κυρτό περίγραμμα που προκύπτει αφαιρείται το τερματικό σημείο \mathbf{U} . Οι πλευρές που προκύπτουν αποτελούν τις ζητούμενες εσωτερικές πλευρές.

Από την κάθε εσωτερική πλευρά σχηματίζεται ένας μη-τετριμένος κώνος. Είναι αυτός που ορίζεται από τα διανύσματα εξάρτησης, τα οποία αποτελούν τις κορυφές της αντίστοιχης πλευράς. Τα διανύσματα που σχηματίζουν το σύνολο των πλευρών (και άρα και το σύνολο των κώνων), αποτελούν τα διανύσματα κώνων.

Για δεδομένο τερματικό σημείο \mathbf{U} και σύνολο διανυσμάτων εξάρτησης $DS = \{\mathbf{d}_1, \dots, \mathbf{d}_m\}$, ο αλγόριθμος που ακολουθεί προσδιορίζει τις εσωτερικές πλευρές, τους κώνους που σχηματίζονται και τα διανύσματα κώνων.

Αλγόριθμος 7.1

εύρεση των εσωτερικών πλευρών (*inner facets*)

INPUT: Τα διανύσματα εξάρτησης $DS = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m\}$ και το τερματικό σημείο \mathbf{U} του αλγορίθμου.

OUTPUT: Η ακολουθία των εσωτερικών πλευρών $\mathcal{IF} = \langle if_1, if_2, \dots \rangle$.

1. Έστω $DS' = DS \cup \mathbf{U}$.

2. Προσδιορίζεται το χυρτό περιγραμμα \mathcal{CH}_{in} των σημείων που περιέχονται στο DS' :

$$\mathcal{CH}_{in} = conv(DS').$$

Εστω $\mathcal{F} = \langle f_1, f_2, \dots \rangle$ η ακολουθία των πλευρών του χυρτού περιγράμματος \mathcal{CH}_{in} .

3. Η ακολουθία των εσωτερικών πλευρών προκύπτει από την ακολουθία των πλευρών του χυρτού περιγράμματος αν αφαιρεθούν οι πλευρές που περιέχουν το τερματικό σημείο \mathbf{U} :

$$\mathcal{IF} = \langle if_1, if_2, \dots \rangle = \langle f_i \in \mathcal{F} \mid \mathbf{U} \notin f_i \rangle.$$

4. Το σύνολο των κορυφών κάθε εσωτερικής πλευράς ορίζει ένα κώνο. Το σύνολο των μη-τετριμένων κώνων προσδιορίζεται ως εξής:

$$NonTrCS = \{Con_i(\mathbf{a}_{i1}, \mathbf{a}_{i2}, \dots) \mid \mathbf{a}_{ij} \in if_i\}.$$

5. Τα διανύσματα των κώνων προκύπτουν από τους μη-τετριμένους κώνους, ως το σύνολο των διανυσμάτων που τους ορίζουν:

$$CV = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_r\}.$$

Με τον αλγόριθμο αυτό υπολογίζονται τα επίπεδα δρομολόγησης όλων των μη-τετριμένων κώνων. Δεν είναι άλλα από αυτά που ορίζονται από τις εσωτερικές πλευρές.

7.3 Υπολογισμός των Τετριμμένων Κώνων

Για τον προσδιορισμό των τετριμένων κώνων αρκεί ο προσδιορισμός του συνόλου των εξωτερικών πλευρών. Για κάθε εξωτερική πλευρά ορίζεται ένας τετριμένος κώνος. Το πλήθος των

διανυσμάτων που τον αποτελούν εξαρτάται από τη συγκεκριμένη πλευρά και εν γένει διαφέρει για κάθε τετριμένο κώνο.

Για τον προσδιορισμό των εξωτερικών πλευρών, προσθέτουμε στο σύνολο των διανυσμάτων των κώνων την αρχή των αξόνων \mathbf{O} και υπολογίζουμε το κυρτό περίγραμμα. Οι πλευρές του περιγράμματος αυτού οι οποίες περιέχουν την αρχή των αξόνων αποτελούν τις ζητούμενες εξωτερικές πλευρές. Τα διανύσματα που ορίζουν την κάθε πλευρά με κάποια από τα μοναδιαία διανύσματα των αξόνων, ορίζουν τον τετριμένο κώνο που αντιστοιχεί στην πλευρά αυτή.

Ο αλγόριθμος που ακολουθεί προσδιορίζει τις εξωτερικές πλευρές και στη συνέχεια το σύνολο των τετριμένων κώνων.

Αλγόριθμος 7.2

εύρεση των εξωτερικών πλευρών (outer facets)

INPUT: Τα διανύσματα των κώνων $CV = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_r\}$ και η αρχή των αξόνων \mathbf{O} .

OUTPUT: Η ακολουθία των εξωτερικών πλευρών $\mathcal{OF} = \langle of_1, of_2, \dots \rangle$.

1. Έστω $CV' = CV \cup \mathbf{O}$.
2. Προσδιορίζεται το κυρτό περίγραμμα \mathcal{CH}_{out} των σημείων που περιέχονται στο CV' :

$$\mathcal{CH}_{out} = conv(CV').$$

Έστω $\mathcal{F} = \langle f_1, f_2, \dots \rangle$ η ακολουθία των πλευρών του κυρτού περιγράμματος \mathcal{CH}_{out} .

3. Η ακολουθία των εξωτερικών πλευρών προκύπτει από την ακολουθία των πλευρών του κυρτού περιγράμματος: είναι αυτές που περιέχουν την αρχή των αξόνων \mathbf{O} :

$$\mathcal{OF} = \langle of_1, of_2, \dots \rangle = \langle f_i \in \mathcal{F} \mid \mathbf{O} \in f_i \rangle.$$

4. Έστω $\mathbf{U}_1, \dots, \mathbf{U}_n$ οι προβολές του τερματικού σημείου \mathbf{U} στους άξονες x_1, \dots, x_n .

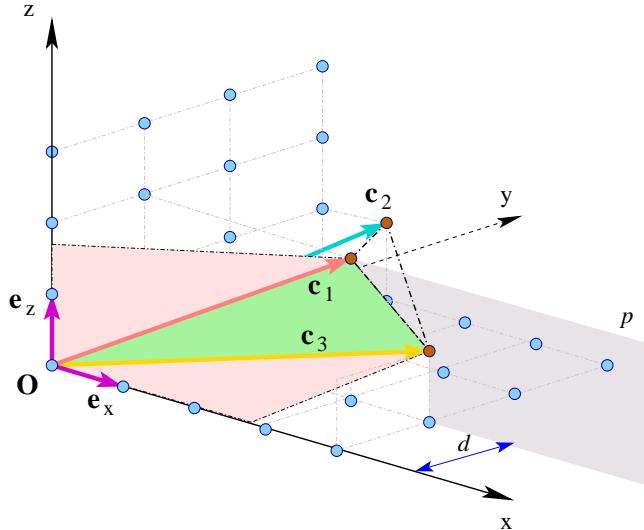
- Για κάθε εξωτερική πλευρά $of_i = \langle \mathbf{a}_{i1}, \dots, \mathbf{a}_{iq} \rangle$, επιλέγονται όλα τα σημεία \mathbf{U}_j τα οποία ανήκουν στον αρνητικό υπόχωρο, σχετικά με το υπερεπίπεδο που ορίζεται από την of_i .¹ Τα σημεία αυτά καλούνται $\mathbf{U}_s, \dots, \mathbf{U}_t$.
- Έστω $\mathbf{e}_s, \dots, \mathbf{e}_t$ τα μοναδιαία διανύσματα τα οποία αντιστοιχούν στα σημεία $\mathbf{U}_s, \dots, \mathbf{U}_t$. Ο τετριμένος κώνος $TrCon_i$ που αντιστοιχεί σε αυτή την (εξωτερική) πλευρά of_i ορίζεται ως εξής:

$$TrCon_i = Con(\mathbf{a}_{i1}, \dots, \mathbf{a}_{iq}, \mathbf{e}_s, \dots, \mathbf{e}_t).$$

¹ Υπολογίζεται με τον τρόπο αυτό η προσανατολισμένη απόσταση όλων των τερματικών σημείων των αξόνων από τη συγκεκριμένη εξωτερική πλευρά.

5. Η διαδικασία επαναλαμβάνεται για όλες τις εξωτερικές πλευρές, οπότε προσδιορίζεται το σύνολο των τετριμμένων κώνων:

$$TrCS = \{TrCon_1, TrCon_2, \dots\}.$$

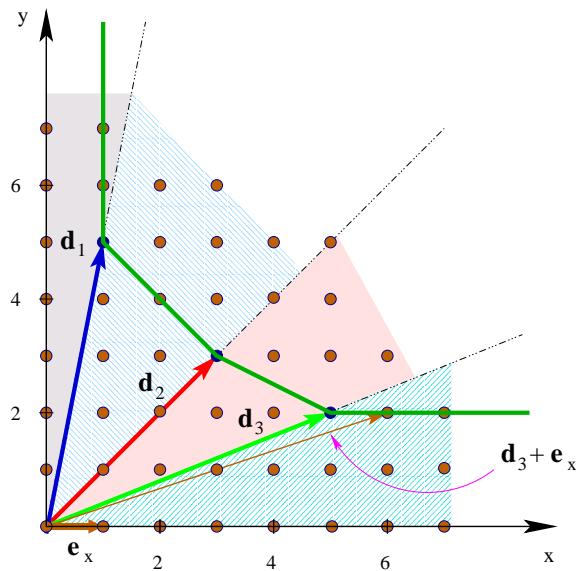


Σχήμα 7.3: Απεικονίζεται ο τετριμμένος κώνος $TrCon(\mathbf{e}_z, \mathbf{c}_1, \mathbf{c}_3, \mathbf{e}_x)$ και το επίπεδο χρονοδρομολόγησης για τον κώνο αυτό p . Το p είναι παράλληλο στο επίπεδο xz και περνάει από τα άκρα των \mathbf{c}_1 και \mathbf{c}_3 . Σημειώνεται η απόσταση d μεταξύ των δύο επιπέδων.

Στο βήμα 4 του αλγορίθμου υπολογίζεται η προσανατολισμένη απόσταση όλων των τερματικών σημείων των αξόνων από τη συγκεκριμένη εξωτερική πλευρά. Το αρνητικό πρόσημο υποδηλώνει ότι το εν λόγω σημείο ανήκει στο εξωτερικό αυτής της πλευράς και άρα είναι δυνατόν να ορίζει κώνο με αυτήν. Αντίθετα, το θετικό πρόσημο σημαίνει ότι το τερματικό σημείο βρίσκεται από την πλευρά της ofi που ορίζονται άλλοι κώνοι, τετριμένοι ή μη-τετριμένοι. Στην περίπτωση αυτή ο ορισμός ενός ακόμη κώνου είναι περιττός.

Στην περίπτωση των τετριμμένων κώνων, το επίπεδο δρομολόγησης ορίζεται με διαφορετικό τρόπο. Έστω ο τετριμμένος κώνος $TrCon_i = Con(\mathbf{a}_{i1}, \dots, \mathbf{a}_{ip}, \mathbf{e}_s, \dots, \mathbf{e}_t)$ και $\mathbf{e} = \mathbf{e}_s + \dots + \mathbf{e}_t$. Το υπερεπίπεδο δρομολόγησης για τον κώνο $TrCon_i$ είναι αυτό που ορίζεται από τα σημεία $\mathbf{a}_{i1}, \dots, \mathbf{a}_{ip}, \mathbf{a}_{i1} + \mathbf{e}, \dots, \mathbf{a}_{ip} + \mathbf{e}$.

Για να γίνει καλύτερα κατανοητός ο καθορισμός του επιπέδου δρομολόγησης στην περιοχή ενός τετριμμένου κώνου, ο αναγνώστης παραπέμπεται στα σχήματα 7.3 και 7.4. Παρουσιάζονται



Σχήμα 7.4: Στο σχήμα αυτό απεικονίζονται τα επίπεδα δρομολόγησης των τετριμένων και των μη-τετριμένων κώνων ενός αλγορίθμου διάστασης 2. Για τον τετριμένο κώνο $TrCon(\mathbf{d}_3, \mathbf{e}_x)$ το επίπεδο δρομολόγησης δίνεται από τα σημεία \mathbf{d}_3 και $\mathbf{d}_3 + \mathbf{e}_x$.

τα επίπεδα δρομολόγησης τετριμένων και μη-τετριμένων κώνων, για έναν αλγόριθμο διάστασης 3 και για έναν αλγόριθμο διάστασης 2, αντίστοιχα.

Για τους μη-τετριμένους κώνους, το επίπεδο δρομολόγησης ορίζεται από τα άκρα των διανυσμάτων που ορίζουν τον κώνο, π.χ. για τους κώνους $Con(\mathbf{d}_1, \mathbf{d}_2)$ και $Con(\mathbf{d}_2, \mathbf{d}_3)$ του σχήματος 7.4. Στην περίπτωση ενός τετριμένου κώνου, καθορίζεται από τα άκρα των διανυσμάτων εξάρτησης που τον αποτελούν και από τα μετατοπισμένα –κατά το άθροισμα των μοναδιαίων διανυσμάτων που συμμετέχουν στον κώνο– άκρα.

Έτσι, στον κώνο $TrCon(\mathbf{d}_3, \mathbf{e}_x)$ του σχήματος 7.4, καθορίζεται από τα άκρα των διανυσμάτων \mathbf{d}_3 και $\mathbf{d}_3 + \mathbf{e}_x$. Κατά τον ίδιο τρόπο, για τον κώνο $TrCon(\mathbf{c}_1, \mathbf{c}_3, \mathbf{e}_x, \mathbf{e}_z)$ του σχήματος 7.3, καθορίζεται από τα άκρα των διανυσμάτων $\mathbf{c}_1, \mathbf{c}_3$ και $\mathbf{c}_1 + \mathbf{e}, \mathbf{c}_3 + \mathbf{e}$, όπου $\mathbf{e} = \mathbf{e}_x + \mathbf{e}_z$.

7.4 Γεωμετρική Δρομολόγηση στις n Διαστάσεις

Μετά τον υπολογισμό των μη-τετριμένων και των τετριμένων κώνων, είναι δυνατόν να πραγματοποιηθεί η δρομολόγηση κάθε σημείου του χώρου δεικτών.

Για τα σημεία που ανήκουν σε μη-τετριμμένο κώνο, η δρομολόγηση γίνεται με τρόπο ανάλογο με τις δύο διαστάσεις. Το προς δρομολόγηση σημείο εκφράζεται ως γραμμικός συνδυασμός των διανυσμάτων που ορίζουν τον κώνο· παίρνοντας το άθροισμα των συντελεστών του γραμμικού συνδυασμού, προκύπτει η χρονική στιγμή που πρέπει να δρομολογηθεί το εν λόγω σημείο.

Για ένα σημείο που δεν ανήκει σε κανένα μη-τετριμμένο κώνο, αρχικά προσδιορίζεται ο τετριμμένος κώνος στον οποίον ανήκει, με βάση τις εξωτερικές πλευρές. Στη συνέχεια, δρομολογείται ανάλογα με την απόσταση του από το επίπεδο δρομολόγησης του τετριμμένου κώνου στον οποίο ανήκει. Ο αλγόριθμος που ακολουθεί υλοποιεί τη γεωμετρική δρομολόγηση στις n διαστάσεις, για προβλήματα χωρίς κόστος επικοινωνίας.

Αλγόριθμος 7.3

γεωμετρική δρομολόγηση στις n διαστάσεις

INPUT: Τα διανύσματα εξάρτησης $DS = \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m\}$, το τερματικό σημείο \mathbf{U} και ένα σημείο $\mathbf{j} \in J$.

OUTPUT: Η χρονική στιγμή t_j κατά την οποία δρομολογείται το \mathbf{j} .

1. Εφαρμόζεται ο Αλγόριθμος 7.1 στο σύνολο DS , οπότε προσδιορίζονται οι εσωτερικές πλευρές $\mathcal{IF} = \langle if_1, if_2, \dots \rangle$, οι μη-τετριμμένοι κώνοι $NonTrCS = \{Con_1, Con_2, \dots\}$ και τα διανύσματα κώνων $CV = \{\mathbf{c}_1, \dots, \mathbf{c}_r\}$.
2. Εφαρμόζεται ο Αλγόριθμος 7.2 στο σύνολο CV , οπότε προσδιορίζονται και οι εξωτερικές πλευρές $\mathcal{OF} = \langle of_1, of_2, \dots \rangle$.
3. Το οποιοδήποτε σημείο $\mathbf{j} \in J$ εντοπίζεται σε έναν τετριμμένο ή μη-τετριμμένο κώνο.
4. Αν το σημείο \mathbf{j} ανήκει σε έναν μη-τετριμμένο κώνο, εκφράζεται ως γραμμικός συνδυασμός των διανυσμάτων που ορίζουν τον κώνο, έστω $\mathbf{a}_1, \dots, \mathbf{a}_q$. προσδιορίζονται οι συντελεστές $\lambda_1, \dots, \lambda_q$. Το άθροισμα τους δίνει τη ζητούμενη χρονική στιγμή:

$$t_j = \lceil \lambda_1 + \lambda_2 + \dots + \lambda_q \rceil.$$

5. Σε διαφορετική περίπτωση, το σημείο \mathbf{j} ανήκει στον τετριμμένο κώνο $TrCon_i$, ο οποίος αντιστοιχεί στην εξωτερική πλευρά of_i . Έστω \mathbf{n}_i το κάθετο μοναδιαίο διάνυσμα της of_i . Η ζητούμενη χρονική στιγμή δίνεται από τη σχέση:

$$t_j = \lceil \mathbf{j} \cdot \mathbf{n}_i \rceil.$$

7.5 Πολυπλοκότητα

Στην παράγραφο αυτή θα μελετήσουμε την πολυπλοκότητα των αλγορίθμων που συνθέτουν τη γεωμετρική μέθοδο δρομολόγησης στις n διαστάσεις. Πιο συγκεκριμένα, θα υπολογίσουμε το χρονικό κόστος που απαιτείται για τον προσδιορισμό των διανυσμάτων των κώνων (cone vectors) –Αλγόριθμος 7.1– και για τη γεωμετρική δρομολόγηση σημείων σε χώρο n διαστάσεων –Αλγόριθμος 7.3.

Το κύριο μέρος του Αλγορίθμου 7.1 είναι ο προσδιορισμός του κυρτού περιγράμματος των διανυσμάτων κυματομορφής και του τερματικού σημείου. Για τον προσδιορισμό του κυρτού περιγράμματος χρησιμοποιείται ο αλγόριθμος QuickHull [BDH96], ο οποίος παρατίθεται και στο Παράρτημα Β'. Τα σημεία για τα οποία υπολογίζεται το κυρτό περίγραμμα είναι $m + 1$ και τα σημεία που αποτελούν το κυρτό περίγραμμα είναι $r + 1$, όπου $r \leq m$.

Κατόπιν αυτών, η πολυπλοκότητα $Cx(7.1)$ προσδιορισμού των εσωτερικών πλευρών, των κώνων και των διανυσμάτων των κώνων στις n διαστάσεις είναι:

$$Cx(7.1) = \begin{cases} \mathcal{O}(m \log m), & n \leq 3 \\ \mathcal{O}(mf_h/h), & n \geq 4 \end{cases} \quad (7.1)$$

όπου m ο αριθμός των διανυσμάτων εξάρτησης, h ο αριθμός των σημείων που προσπελαύνονται και f_h ο μέγιστος αριθμός πλευρών (facets) για h κορυφές. Σημειώνουμε ότι στη δυσμενέστερη περίπτωση είναι $f_r = \Theta(r^{\lfloor n/2 \rfloor})$, όπου r ο αριθμός των διανυσμάτων κώνων και $r \leq m$ [Cha95].

Κεφάλαιο 8

Υλοποιήσεις και Αποτελέσματα

Στο κεφάλαιο αυτό παρουσιάζονται δύο υλοποιήσεις που πραγματοποιήθηκαν στα πλαίσια της παρούσας διατριβής. Η πρώτη συνετέλεσε στην κατανόηση της επέκτασης της μεθόδου από τις δύο διαστάσεις στις τρεις και κατ' επέκτασιν στις n . Το πρόγραμμα αυτό ακούει στο όνομα `cubePlaying`.¹

Η δεύτερη υλοποίηση² αφορά την παραγωγή τυχαίων αλγορίθμων φωλιασμένων βρόχων και τη δρομολόγηση αυτών σύμφωνα με τις μεθόδους που παρουσιάστηκαν στην παρούσα διατριβή. Κατόπιν, πραγματοποιούνται συγχρίσεις όσον αφορά τους χρόνους εκτέλεσης, τόσο σε σχέση με την επιχρατέστερη μέθοδο των υπερεπιπέδων, όσο και με την εξαντλητική (βέλτιστη) εκτέλεση. Η δεύτερη αυτή υλοποίηση ακούει στο όνομα `crazySched`.

Τα αποτελέσματα των συγχρίσεων της υλοποίησης αυτής παρατίθενται στην Παράγραφο 8.3, μετά την παρουσίαση των υλοποιήσεων.

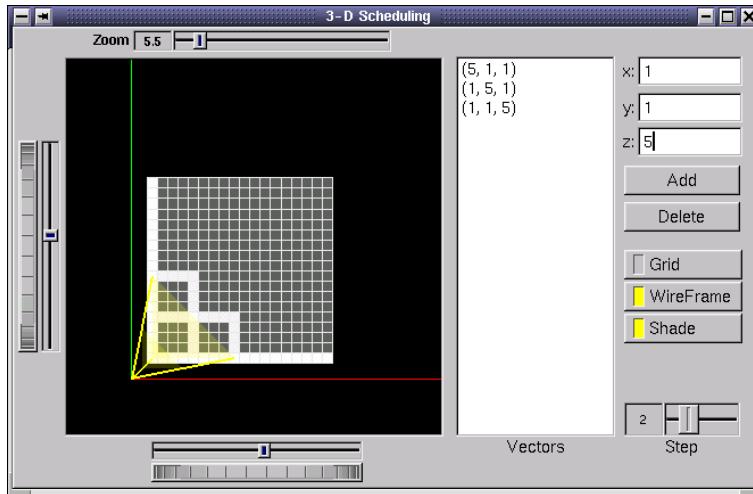
8.1 Τρισδιάστατη Απεικόνιση Εκτέλεσης (`cubePlaying`)

Το πρόγραμμα `cubePlaying` λειτουργεί ως εξής. Δέχεται ως είσοδο το σύνολο των διανυσμάτων εξάρτησης του αλγορίθμου και προσομοιώνει την εκτέλεσή του ακολουθώντας το μοντέλο UET. Στη συνέχεια, απεικονίζει τα περιγράμματα κυματομορφής (pattern outlines) της χρονι-

¹ Υλοποιήθηκε σε συνεργασία με τον Στέργιο Στεργίου, ως εργασία του μεταπτυχιακού μαθήματος *Παραλληλης Επεξεργασίας*.

² Η κυρίως υλοποίηση πραγματοποιήθηκε από τον Ιωάννη Παναγόπουλο και τροποποιήθηκε κατάλληλα για τις ανάγκες τις παρούσας διατριβής.

κής στιγμής που επιλέγει ο χρήστης στον τρισδιάστατο χώρο. Δίνεται στο χρήστη η δυνατότητα μεγέθυνσης της ορατής περιοχής, καθώς και πλήρεις δυνατότητες στροφής των αξόνων.



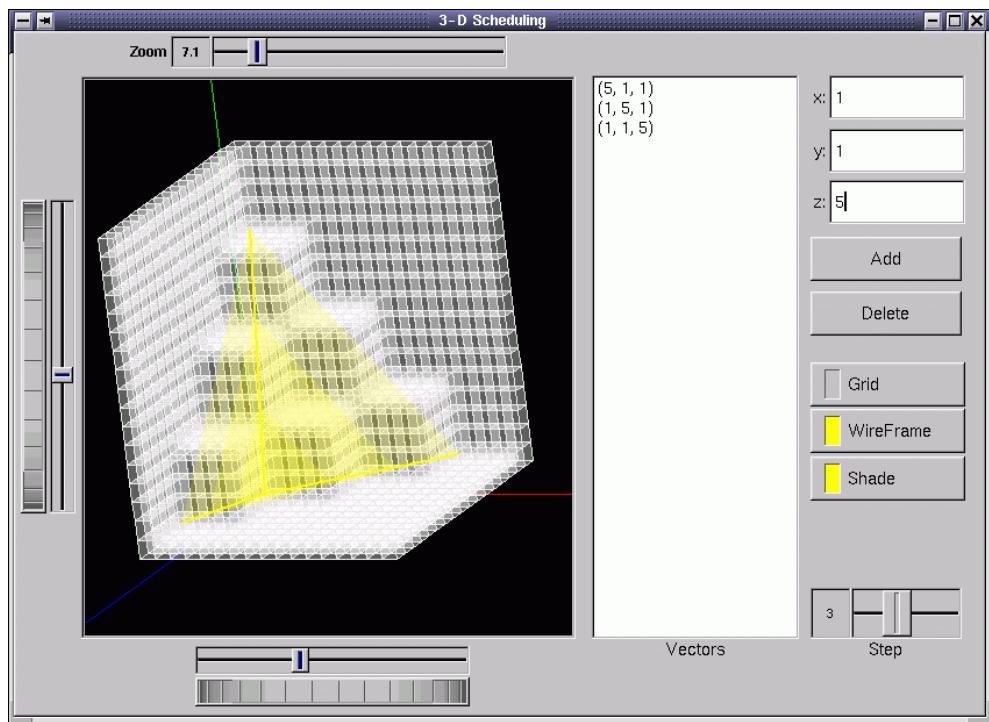
Σχήμα 8.1: Απεικόνιση της κυματομορφής εκτέλεσης ενός αλγορίθμου με τρία διανύσματα εξάρτησης, με τη χρήση του προγράμματος `cubePlaying`. Απεικονίζεται η προβολή του περιγράμματος της χρονικής στιγμής $t = 2$ στο επίπεδο xy .

Στα σχήματα 8.1, 8.2 και 8.3 παρουσιάζονται τρία στιγμιότυπα από την εκτέλεση του προγράμματος `cubePlaying`. Στα δύο πρώτα σχήματα παρουσιάζονται δύο βήματα από την εκτέλεση ενός τρισδιάστατου αλγορίθμου με διανύσματα εξάρτησης τα $\mathbf{d}_1 = (5, 1, 1)$, $\mathbf{d}_2 = (1, 5, 1)$ και $\mathbf{d}_3 = (1, 1, 5)$. Τα διανύσματα αυτά είναι και διανύσματα κώνων για τον εν λόγω αλγόριθμο.

Στο σχήμα 8.1 παρουσιάζεται η προβολή του περιγράμματος της χρονικής στιγμής $t = 2$ στο επίπεδο xy . Στο σχήμα 8.2 απεικονίζεται το περίγραμμα της χρονικής στιγμής $t = 3$. Σημειώνονται με αχνό χρώμα τα διαδοχικά επίπεδα δρομολόγησης του μοναδικού κώνου που σηματίζεται (από τα παραπάνω τρία διανύσματα κώνων) και στα δύο σχήματα.

Στο σχήμα 8.3 απεικονίζεται ένα στιγμιότυπο εκτέλεσης του προγράμματος `cubePlaying` για έναν δεύτερο αλγόριθμο, με διανύσματα κώνων τα $\mathbf{d}_1 = (5, 1, 1)$, $\mathbf{d}_2 = (1, 1, 5)$ και $\mathbf{d}_3 = (2, 2, 2)$. Διακρίνονται και στην περίπτωση αυτή τα υπερεπίπεδα δρομολόγησης.

Για το interface του προγράμματος χρησιμοποιήθηκε το περιβάλλον σχεδίασης `fltk`, ενώ για την τρισδιάστατη απεικόνιση της εκτέλεσης χρησιμοποιήθηκε η βιβλιοθήκη `OpenGL`. Η υλοποίηση



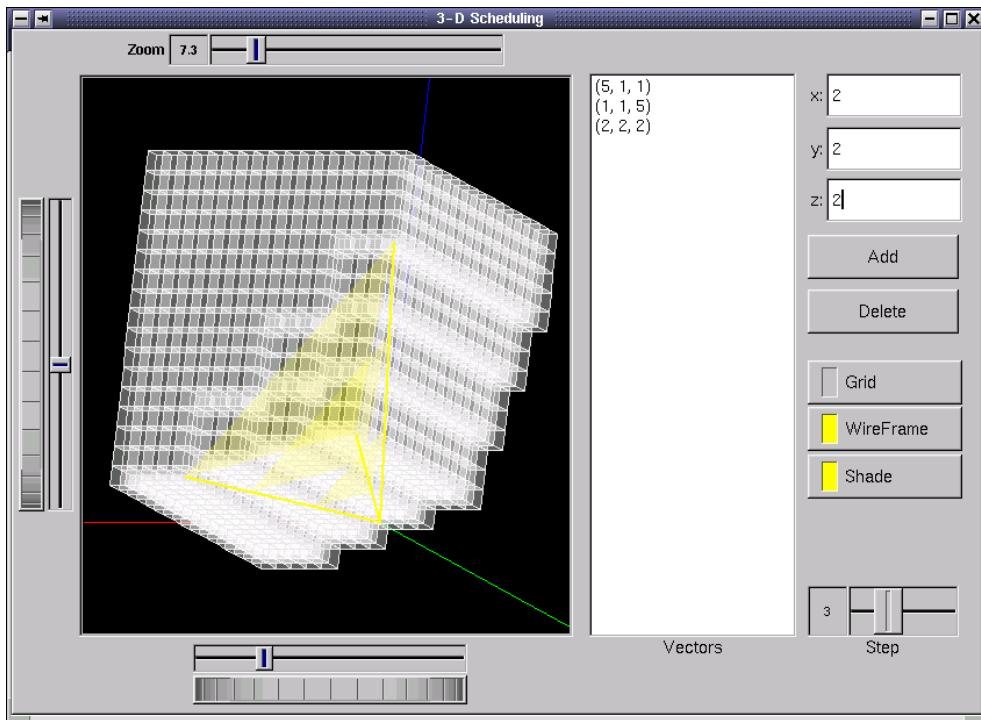
Σχήμα 8.2: Το περίγραμμα της χρονικής στιγμής $t = 3$, για τον αλγόριθμο του σχήματος 8.1, μέσω του προγράμματος `cubePlaying`.

έγινε σε περιβάλλον Linux.

8.2 Γεωμετρική Δρομολόγηση Αλγορίθμων (crazySched)

Πρόκειται για μια ολοκληρωμένη πλατφόρμα δρομολόγησης και αξιολόγησης αλγορίθμων φωλιασμένων βρόχων, μέ ή χωρίς κόστος επικοινωνίας. Αποτελείται από τα εξής επιμέρους προγράμματα:

- `generate [-f file] <#algs> <problem size> <#deps> <value range> [-c]`
το πρόγραμμα αυτό γεννά έναν αριθμό αλγορίθμων, με τυχαία διανύσματα εξάρτησης συγκεκριμένου εύρους. Είναι δυνατόν να πάρει είσοδο από αρχείο μέσω της παραμέτρου `[-f]`. Η παράμετρος `[-c]` καθορίζει το αν οι παραγόμενοι αλγόριθμοι θα έχουν κόστος επικοινωνίας ή όχι (αλγόριθμοι UET-UCT ή απλοί UET). Στην πρώτη περίπτωση, διάνυσμα



Σχήμα 8.3: Το περίγραμμα της χρονικής στιγμής $t = 3$, για έναν δεύτερο αλγόριθμο με τρία διανύσματα κώνων, μέσω του προγράμματος *cubePlaying*.

επικοινωνίας (communication vector) θεωρείται το δεύτερο διάνυσμα εξάρτησης, εκτός αν επιλεγεί κάποιο άλλο από τον χρήστη. Με την εκτέλεση του προγράμματος αυτού παράγονται τα εξής αρχεία:

- **deps**: περιέχει τα διανύσματα εξάρτησης για κάθε αλγόριθμο που παράγεται,
- **deps_equiv**: δημιουργείται μόνο στην περίπτωση UET-UCT αλγορίθμων (παράμετρος **[-c]**) και περιέχει τα διανύσματα εξάρτησης των ισοδυνάμων UET προβλημάτων,
- **sched_simple**: περιέχει το αποτέλεσμα της εξαντλητικής δρομολόγησης των παραγόμενων αλγορίθμων.

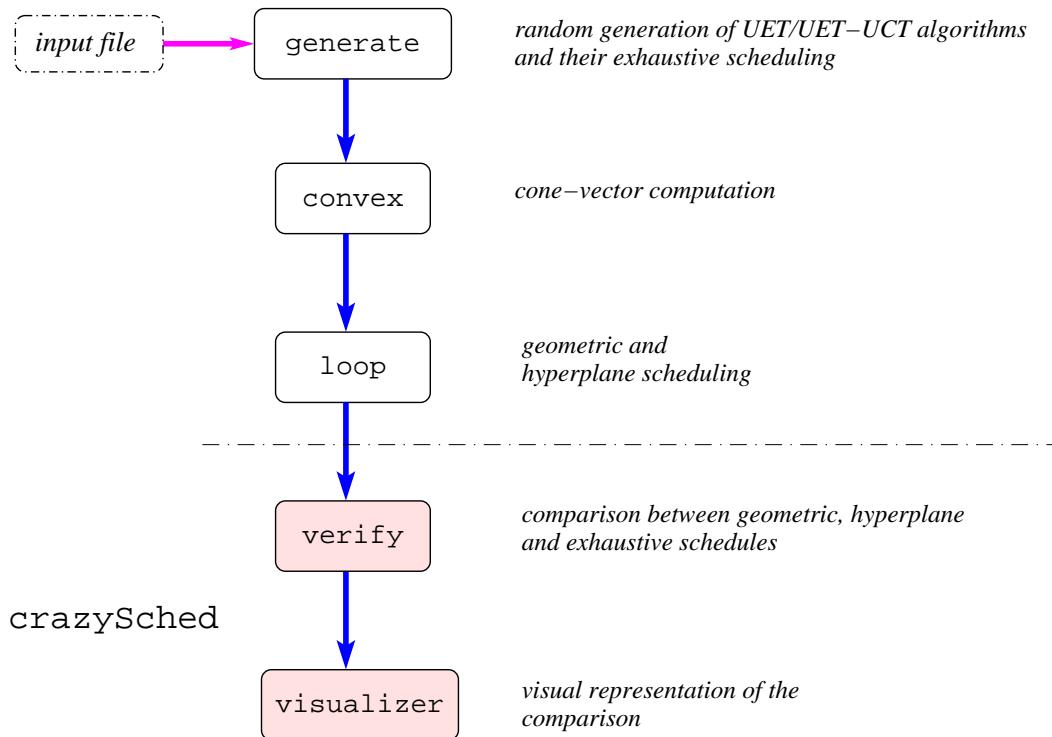
• **convex [-c]**

το πρόγραμμα αυτό, για κάθε έναν από τους αλγορίθμους του αρχείου **deps/deps_equiv**, προσδιορίζει τα διανύσματα κώνων και τα τοποθετεί με αντίστοιχη σειρά στο αρχείο **vectors**.

Χρησιμοποιεί το πρόγραμμα `qhull` και συγκεκριμένα το εκτελέσιμο `qconvex`. Περισσότερες πληροφορίες για τα προγράμματα αυτά αναφέρονται στην Παράγραφο Β'.3 του παραπάνω ματιών. Το αρχείο που παράγει είναι το:

- **vectors**: περιέχει τα διανύσματα κώνων (cone vectors) για κάθε έναν από τους παραγόμενους αλγορίθμους. Στην περίπτωση αλγορίθμων UET-UCT, περιέχει τα διανύσματα κώνων των ισοδυνάμων UET αλγορίθμων (χρήση παραμέτρου `[-c]`).
- **loop [-c]**
το πρόγραμμα αυτό, για κάθε έναν από τους αλγορίθμους του αρχείου `deps/deps_equiv`, πραγματοποιεί χρονοδρομολόγηση ως εξής. Αν πρόκειται για UET αλγόριθμο, τον δρομολογεί ακολουθώντας την γεωμετρική δρομολόγηση και τη μέθοδο των υπερεπιπέδων. Αν πρόκειται για αλγόριθμο UET-UCT, τον χρονοδρομολογεί ακολουθώντας μόνο τη γεωμετρική μέθοδο. Παράγει τα εξής αρχεία:
 - `sched_qhull`: περιέχει το αποτέλεσμα της δρομολόγησης των παραγόμενων αλγορίθμων ακολουθώντας τη μέθοδο γεωμετρικής δρομολόγησης,
 - `sched_p`: περιέχει το αποτέλεσμα της δρομολόγησης των παραγόμενων αλγορίθμων ακολουθώντας τη μέθοδο των υπερεπιπέδων –το αρχείο αυτό παράγεται μόνο στην περίπτωση αλγορίθμων UET.
- **verify <file1> <file2>**
το πρόγραμμα αυτό συγχρίνει τις δρομολογήσεις που περιέχονται στα αρχεία `file1` και `file2`. Προσδιορίζει το ποσοστιαίο λάθος, καθώς και τα απόλυτα λάθη ανά πρόβλημα. Στο τέλος εξάγει τον μέσο όρο των αποκλίσεων για το σύνολο των προβλημάτων. Τα αρχεία εισόδου μπορεί να είναι οποιαδήποτε δύο εκ των `sched_simple`, `sched_qhull` και `sched_p`.
- **visualizer**
το πρόγραμμα αυτό, παίρνοντας ως είσοδο τα αποτελέσματα του `verify`, αναπαριστά τις εν λόγω δρομολογήσεις σε περιβάλλον παραθύρων X-Windows. Ο χρήστης αποκτά μια εποπτική εικόνα των αποκλίσεων κατά τη δρομολόγηση.

Η λειτουργία και η αλληλεπίδραση των προγραμμάτων αυτών παρουσιάζεται στο σχήμα 8.4.



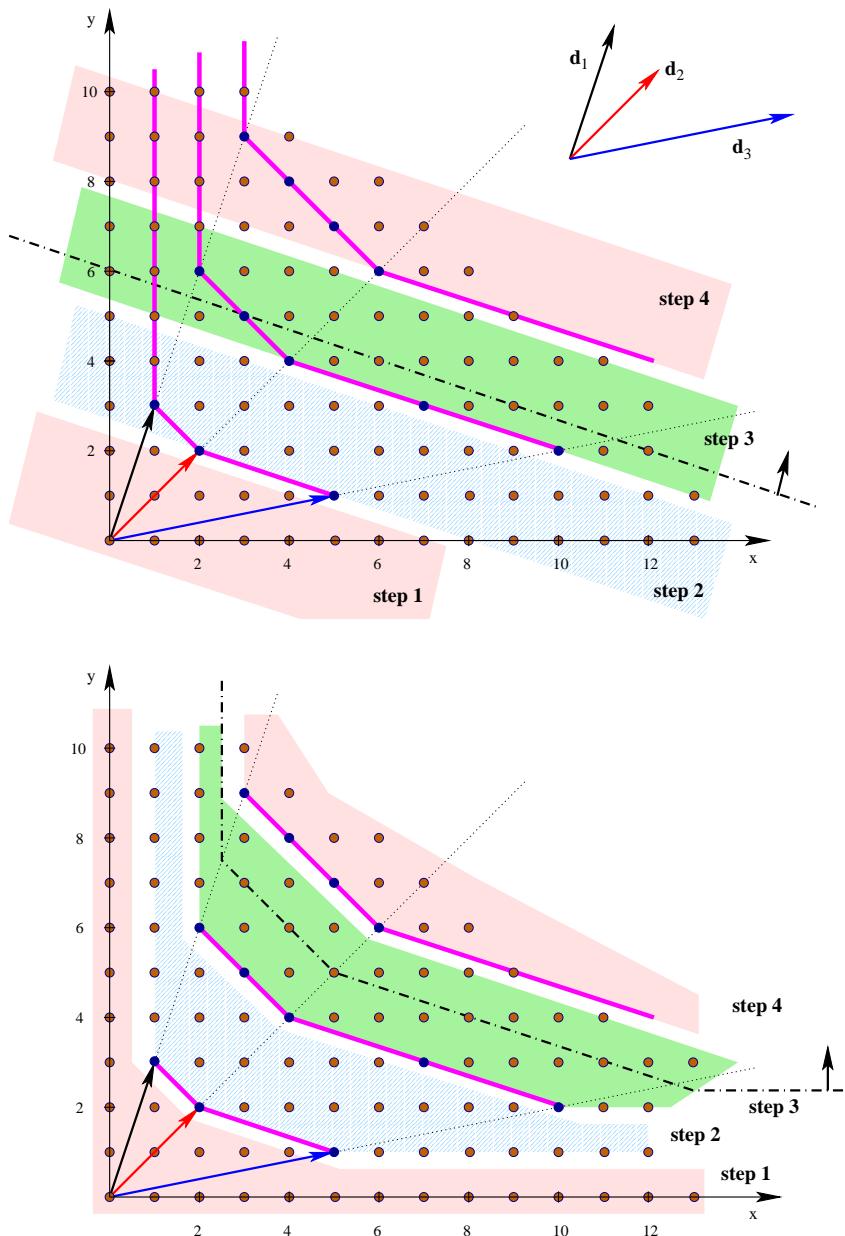
Σχήμα 8.4: Αξιολόγηση αλγορίθμων με τη χρήση της πλατφόρμας προγραμμάτων `crazySched`.

8.3 Αποτελέσματα & Συγκρίσεις

Για την αξιολόγηση των μεθόδων που παρουσιάστηκαν στην παρούσα διατριβή, πραγματοποιήθηκαν μια σειρά από πειράματα. Στα πειράματα αυτά έγινε σύγχριση της γεωμετρικής δρομολόγησης με την μέθοδο των υπερεπιπέδων, όπως και με την εξαντλητική μέθοδο. Οι μετρήσεις έγιναν τόσο σε αλγορίθμους χωρίς κόστος επικοινωνίας (μοντέλο UET), όσο και σε άλλους με κόστος επικοινωνίας (μοντέλο UET-UCT).

Τα πειράματα μας κινήθηκαν στα πλαίσια των δύο διαστάσεων και σε χώρους δεικτών μέχρι μεγέθους 1.000×1.000 , με πλήθος διανυσμάτων εξάρτησης έως 15. Στους πίνακες που ακολουθούν παρουσιάζονται ορισμένα ενδεικτικά αποτελέσματα των μετρήσεων που πραγματοποιήθηκαν, για προβλήματα μεγέθους 100, 500 και 1000, ανά διάσταση. Συγκεντρωτικά αποτελέσματα παρουσιάζονται στα διαγράμματα που ακολουθούν.

Σημειώνεται το ποσοστό των σημείων για τα οποία η εκτέλεση καθυστερεί έναντι του βέλ-



Σχήμα 8.5: Γραφική αναπαράσταση της εκτέλεσης με βάση τη γεωμετρική μέθοδο (κάτω) και τη μέθοδο των υπερεπιπέδων (επάνω). Στη μέθοδο των υπερεπιπέδων, παρατηρούμε την αναπόφευκτα μεγάλη καθυστέρηση της εκτέλεσης ορισμένων σημείων που βρίσκονται κοντά στους άξονες.

τιστου. Στην περίπτωση των προβλημάτων UET, σημειώνεται η καθυστέρηση της γεωμετρικής μεθόδου και της μεθόδου των υπερεπιπέδων έναντι της ιδανικής. Για τα προβλήματα UET-UCT, σημειώνεται το ποσοστό των σημείων των οποίων η εκτέλεση καθυστερεί έναντι της ιδανικής εκτέλεσης. Οι τιμές αναφέρονται στο μέσο όρο αποτελεσμάτων κατά τη δρομολόγηση 20 αλγορίθμων.

Αριθμός Εξαρτήσεων	UET		UET-UCT
	υπερεπιπέδων	γεωμετρική	γεωμετρική
3 - 6	9.01%	0.33%	6.16%
7 - 10	8.19%	0.24%	3.99%
11 - 15	8.43%	0.09%	2.41%

Πίνακας 8.1: Σύγκριση της μεθόδου γεωμετρικής δρομολόγησης και της μεθόδου των υπερεπιπέδων με την εξαντλητική (ιδανική). Σημειώνεται το ποσοστό των σημείων του χώρου των οποίων η εκτέλεση καθυστερεί, έναντι της ιδανικής χρονικής στιγμής εκτέλεσής τους. Θεωρούμε χώρο δεικτών μεγέθους 100×100 .

Σημειώνεται επιπλέον ότι, ενώ στη γεωμετρική μέθοδο δρομολόγησης το τυχόν λάθος είναι ένα ή δύο, στη μέθοδο των υπερεπιπέδων μπορεί να είναι πολύ μεγαλύτερο. Εντούτοις, στη σύγκριση που παραγματοποιήθηκε υεωρήθηκαν ως ιδίου μεγέθους λάθη. Αν υπολογίσουμε και την ακριβή απόκλιση των δρομολογήσεων από το ιδανικό, τα αποτελέσματα θα είναι δραματικότερα, καθώς η μέθοδος των υπερεπιπέδων καθυστερεί κατά πολύ τη δρομολόγηση των σημείων που βρίσκονται κοντά στους άξονες (βλέπε σχήμα 8.5).

Αριθμός Εξαρτήσεων	UET		UET-UCT
	υπερεπιπέδων	γεωμετρική	γεωμετρική
3 - 6	9.51%	0.31%	5.31%
7 - 10	9.48%	0.17%	2.30%
11 - 15	9.07%	0.16%	3.30%

Πίνακας 8.2: Σύγκριση της μεθόδου γεωμετρικής δρομολόγησης και της μεθόδου των υπερεπιπέδων με την εξαντλητική (ιδανική). Σημειώνεται το ποσοστό των σημείων του χώρου των οποίων η εκτέλεση καθυστερεί, έναντι της ιδανικής χρονικής στιγμής εκτέλεσής τους. Θεωρούμε χώρο δεικτών μεγέθους 500×500 .

Αριθμός Εξαρτήσεων	UET		UET-UCT
	υπερεπιπέδων	γεωμετρική	γεωμετρική
3 - 6	9.93%	0.20%	3.41%
7 - 10	9.00%	0.15%	3.38%
11 - 15	9.63%	0.10%	2.20%

Πίνακας 8.3: Σύγκριση της μεθόδου γεωμετρικής δρομολόγησης και της μεθόδου των υπερεπιπέδων με την εξαντλητική (ιδανική). Σημειώνεται το ποσοστό των σημείων του χώρου των οποίων η εκτέλεση καθυστερεί, έναντι της ιδανικής χρονικής στιγμής εκτέλεσής τους. Θεωρούμε χώρο δεικτών μεγέθους 1.000×1.000 .

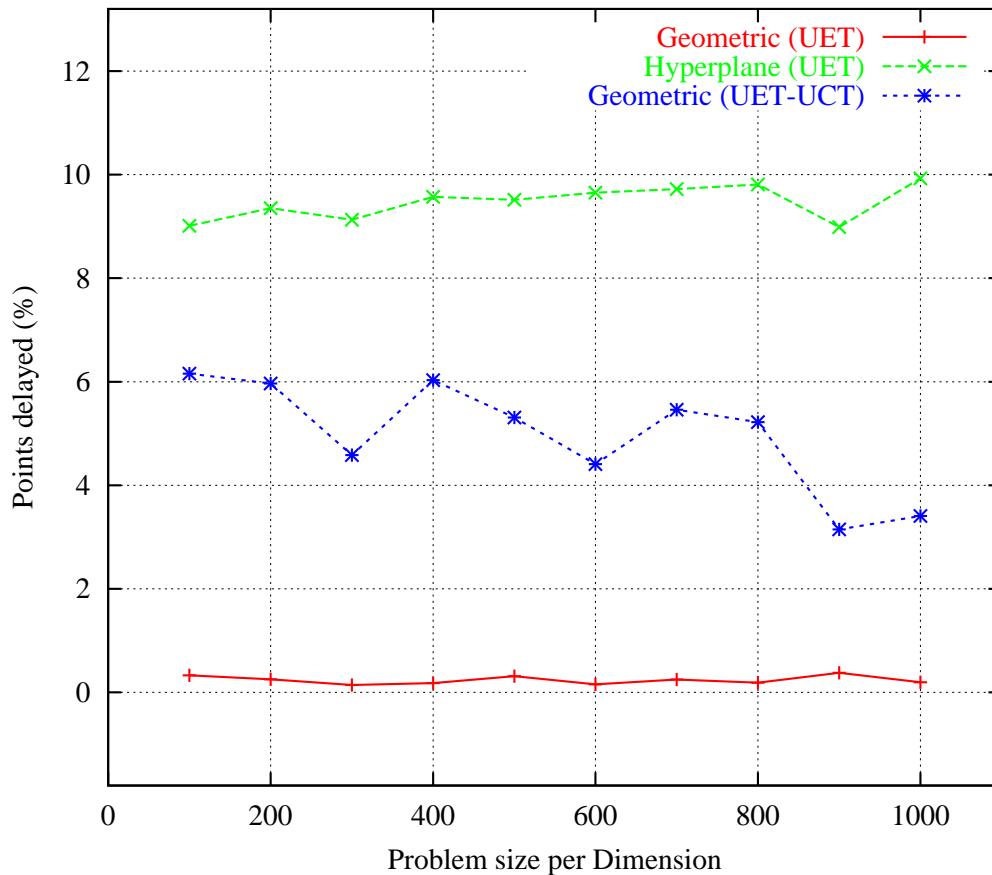
Συνολικά πραγματοποιήθηκαν συγκρίσεις για μέγεθος προβλημάτων από 100 έως 1000 ανά διάσταση, για 3 έως 15 διανύσματα εξάρτησης. Στα διαγράμματα που ακολουθούν παραθέτουμε τα συγκεντρωτικά αποτελέσματα των συγκρίσεων αυτών.

Συγκεκριμένα, εκτελούμε 20 αλγορίθμους με αριθμό διανυσμάτων 3-6 και μέγεθος 100-1000, με βήμα 100. Η δρομολόγηση πραγματοποιείται πρώτον με τη γεωμετρική μέθοδο, δεύτερον με τη μέθοδο των υπερεπιπέδων και τρίτον εξαντλητικά. Αυτή η τελευταία μέθοδος δίνει τη χρονικά βέλτιστη εκτέλεση κάθε σημείου και η απόκλιση απ' αυτήν αποτελεί το μέτρο σύγκρισης και αποτελεσματικότητας των άλλων δύο. Σημειώνονται τα σημεία των οποίων η εκτέλεση καθυστερεί έναντι του βελτίστου. Στο σχήμα 8.6 παρατίθενται οι ποσοστιαίες αποκλίσεις (μέσες τιμές για 20 αλγορίθμους) από το βέλτιστο, τόσο για τη γεωμετρική μέθοδο χρονοδρομολόγησης, όσο και για τη μέθοδο των υπερεπιπέδων.

Η ίδια διαδικασία πραγματοποιείται και στην περίπτωση που έχουμε κόστος επικοινωνίας (μοντέλο UET-UCT), με τη διαφορά ότι δεν έχουμε τη μέθοδο των υπερεπιπέδων. Η γεωμετρική μέθοδος, στην περίπτωση αυτή, συγκρίνεται μόνο με την εξαντλητική. Οι ποσοστιαίες αποκλίσεις της γεωμετρικής δρομολόγησης έναντι της εξαντλητικής, για την περίπτωση που έχουμε κόστος επικοινωνίας, σημειώνονται επίσης στο σχήμα 8.6.

Στα σχήματα 8.7 και 8.8 παρουσιάζονται τα αντίστοιχα ποσοστά για αλγορίθμους με 7-10 και 11-15 διανύσματα εξάρτησης αντίστοιχα.

Non-optimally Scheduled Instances for Loop Algorithms with 3-6 Dependence Vectors
(mean values for 20 scheduled algorithms)

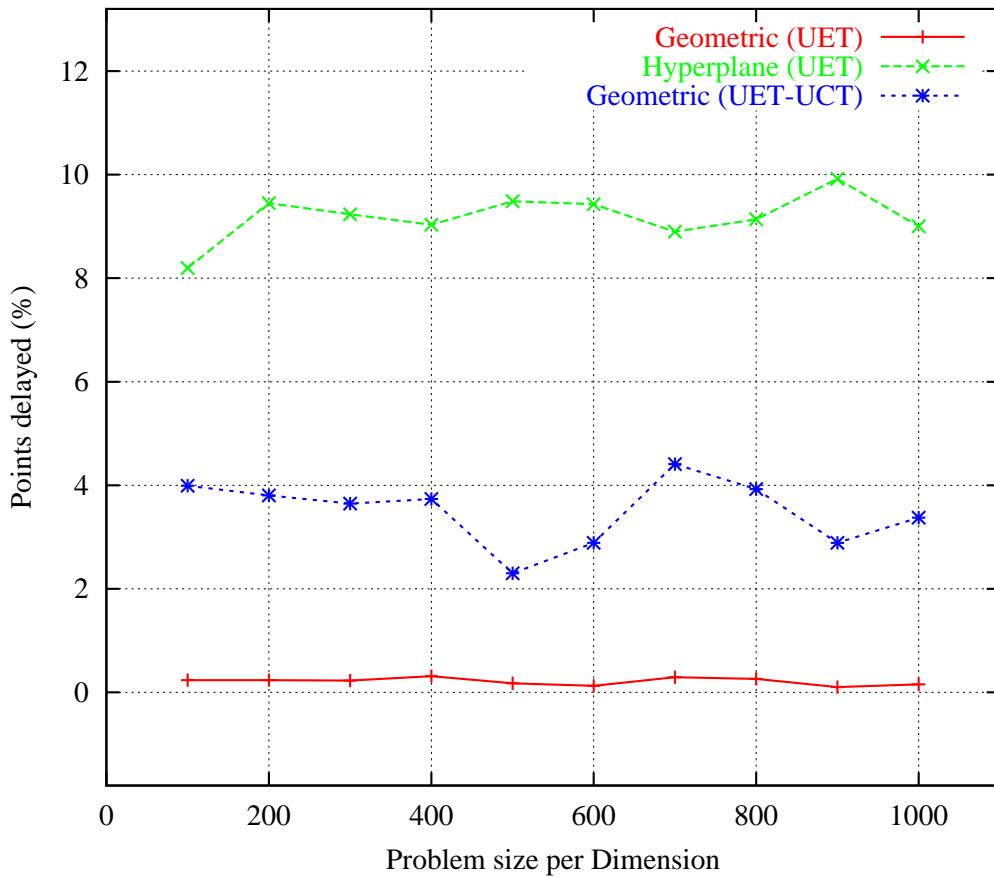


Σχήμα 8.6: Συγκριτικά ποσοστά καθυστέρησης της εκτέλεσης ανά σημείο, για αλγορίθμους φωλιασμένων βρόχων με 3-6 διανύσματα εξάρτησης. Απεικονίζονται τα ποσοστά λάθους της γεωμετρικής μεθόδου και της μεθόδου των υπερεπιπέδων σε σχέση με την εξαντλητική (μοντέλο UET). Απεικονίζεται επίσης το ποσοστό λάθους της γεωμετρικής μεθόδου σε σχέση με την εξαντλητική και για το μοντέλο εκτέλεσης UET-UCT.

8.4 Σχολιασμός των Αποτελεσμάτων

Από τα συγκριτικά αποτελέσματα που εμφανίζονται στους Πίνακες 8.1, 8.2 και 8.3 και τα διαγράμματα των σχημάτων 8.6, 8.7 και 8.8, συμπεραίνουμε τα εξής:

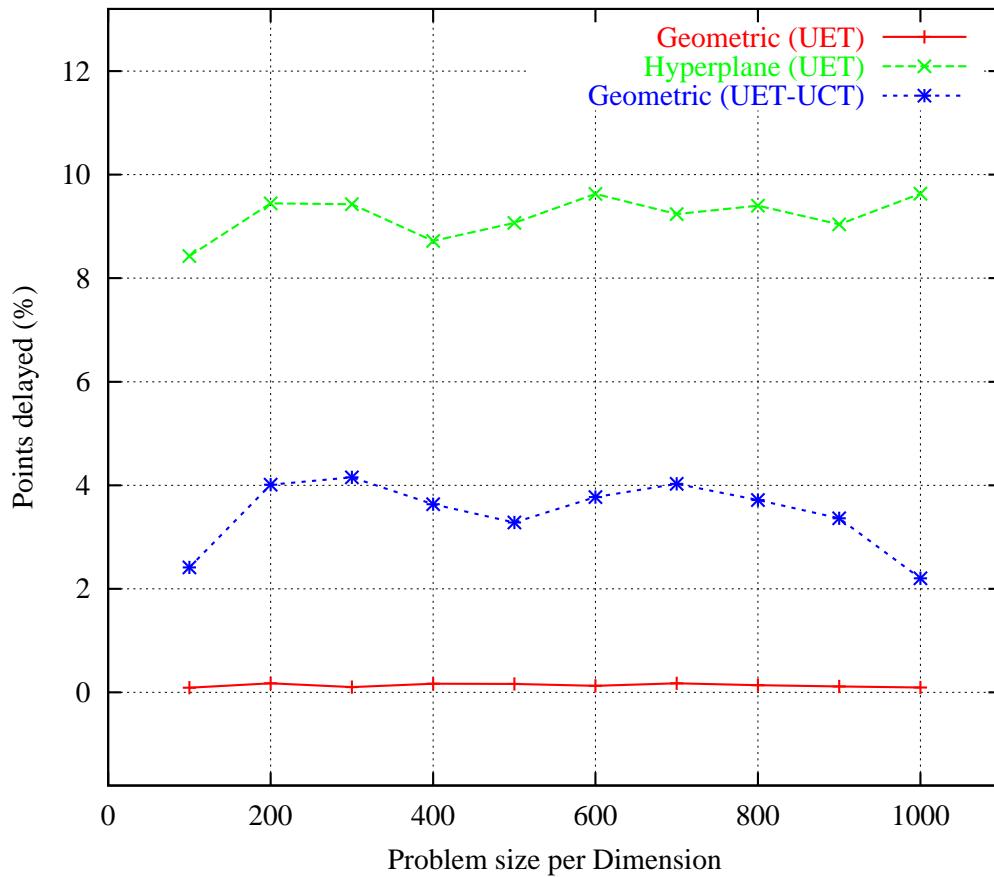
Non-optimally Scheduled Instances for Loop Algorithms with 7-10 Dependence Vectors
(mean values for 20 scheduled algorithms)



Σχήμα 8.7: Συγκριτικά ποσοστά καθυστέρησης της εκτέλεσης ανά σημείο, για αλγορίθμους φωλιασμένων βρόχων με 7-10 διανύσματα εξάρτησης. Απεικονίζονται τα ποσοστά λάθους της γεωμετρικής μεθόδου και της μεθόδου των υπερεπιπέδων σε σχέση με την εξαντλητική (μοντέλο UET). Απεικονίζεται επίσης το ποσοστό λάθους της γεωμετρικής μεθόδου σε σχέση με την εξαντλητική και για το μοντέλο εκτέλεσης UET-UCT.

- η γεωμετρική μέθοδος δρομολόγησης υπερτερεί της μεθόδου των υπερεπιπέδων, όσον αφορά την προσέγγιση της βέλτιστης χρονικής στιγμής εκτέλεσης για κάθε σημείο.
- μεταξύ των δύο μεθόδων, το ποσοστό των σημείων που δρομολογούνται κατά την βέλτιστη

Non-optimally Scheduled Instances for Loop Algorithms with 11-15 Dependence Vectors
(mean values for 20 scheduled algorithms)



Σχήμα 8.8: Συγκριτικά ποσοστά καθυστέρησης της εκτέλεσης ανά σημείο, για αλγορίθμους φωλιασμένων βρόχων με 11-15 διανύσματα εξάρτησης. Απεικονίζονται τα ποσοστά λάθους της γεωμετρικής μεθόδου και της μεθόδου των υπερεπιπέδων σε σχέση με την εξαντλητική, για το μοντέλο UET. Απεικονίζεται επίσης το ποσοστό λάθους της γεωμετρικής μεθόδου σε σχέση με την εξαντλητική και για το μοντέλο εκτέλεσης UET-UCT.

χρονική στιγμή γι' αυτά (earliest computation time), διαφέρει κατά μία τάξη μεγέθους·

- κατά τη δρομολόγηση UET-UCT αλγορίθμων, η απόκλιση από τη βέλτιστη εκτέλεση πραγματοποιείται για πολύ μικρό αριθμό σημείων (μέχρι 7%).

- όσο αυξάνει ο αριθμός των διανυσμάτων εξάρτησης, η συμπεριφορά της γεωμετρικής μεθόδου βελτιώνεται και στην περίπτωση του UET αλλά και στην περίπτωση του UET-UCT στην περίπτωση της UET εκτέλεσης μάλιστα, προσεγγίζει κατά πολύ τη βέλτιστη δρομολόγηση.

Κεφάλαιο 9

Συμπεράσματα και Προτάσεις

Ο κύριος σκοπός αυτής της διατριβής ήταν να μελετήσει τις γεωμετρικές ιδιότητες του χώρου δεικτών και των διανυσμάτων εξάρτησης φωλιασμένων βρόχων, καθώς και τον τρόπο που αυτές επηρεάζουν τη δρομολόγησή τους. Ο σκοπός αυτός επετεύχθη, όχι μόνο για τα προβλήματα που δεν περιέχουν κόστος επικοινωνίας, αλλά και γι' αυτά που το κόστος επικοινωνίας είναι συγχρίσιμο με το κόστος υπολογισμού.

Κάτω από το πρίσμα αυτό, οι καινοτομίες που παρουσιάστηκαν στην παρούσα διατριβή είναι συνοπτικά οι εξής:

- Ένας αλγόριθμος απεικόνισης σε συστολικές διατάξεις· η ιδιαιτερότητα του είναι ότι απεικονίζει σε μονάδες επεξεργασίας, αξιοποιώντας τις διευθύνσεις του χώρου δεικτών. Με τον τρόπο αυτό επιτυγχάνεται βέλτιστη ισοκατανομή του φόρτου εργασίας στις μονάδες επεξεργασίας, ακόμα και στις περιπτώσεις που έχουμε κεκλιμένες συνοριακές διευθύνσεις.
- Σχηματίστηκαν οι εξισώσεις οι οποίες δίνουν την χυματομορφή εκτέλεσης, για κάθε χρονική στιγμή. Έγινε η επισήμανση ορισμένων ειδικών περιπτώσεων, κατά τις οποίες οι χυματομορφές εκτέλεσης είναι απολύτως ομοιόμορφες και επαναλαμβανόμενες, γεγονός που επιτρέπει την ακριβή πρόβλεψη της χυματομορφής εκτέλεσης σε πολυωνυμικό ή ακόμη και γραμμικό χρόνο.
- Παρουσιάστηκε μια εναλλακτική μέθοδος δρομολόγησης, για τα προβλήματα που περιλαμβάνουν διπλά φωλιασμένους βρόχους, η οποία βασίζεται στη γεωμετρική διαμέριση του χώρου δεικτών. Η εν λόγω διαμέριση έγινε με βάση ορισμένα διανύσματα εξάρτησης τα

οποία καλούνται διανύσματα κώνων (cone vectors) και είναι αυτά που καθορίζουν, κατ' ουσίαν τη εκτέλεση. Η μέθοδος αναφέρεται σε προβλήματα όπου το κόστος υπολογισμού θεωρείται μοναδιαίο και το κόστος επικοινωνίας μηδενικό (μοντέλο UET).

- Παρουσιάστηκε μια μέθοδος αναγωγής των προβλημάτων που εμπεριέχουν επικοινωνία σε ισοδύναμα που δεν περιέχουν επικοινωνία. Τα προβλήματα όπου το κόστος επικοινωνίας θεωρείται συγχρίσιμο με το κόστος υπολογισμού (μοντέλο UET-UCT), είναι δυνατόν να αναχθούν σε προβλήματα χωρίς κόστος επικοινωνίας (καθαρά UET προβλήματα). Προφανώς, το μετασχηματισμένο πρόβλημα έχει διαφορετικές εξαρτήσεις από το αρχικό, παρουσιάζει όμως εντελώς αντίστοιχους χρόνους εκτέλεσης.
- Στις δύο παραπάνω μεθόδους, χρησιμοποιήθηκε μια εναλλακτική μέθοδος υπολογισμού του βέλτιστου διανύσματος χρονοδρομολόγησης Π . Η μέθοδος αυτή βασίστηκε στον υπολογισμό του κυρτού περιγράμματος συνόλου σημείων. Η πολυπλοκότητα που επιτυγχάνει είναι εντυπωσιακά μικρότερη από αυτές των μεθόδων της βιβλιογραφίας, λόγω του βέλτιστου του αλγορίθμου στον οποίον βασίζεται.
- Αναλύθηκαν οι βασικές ιδέες που επιτρέπουν τη γενίκευση των παραπάνω μεθόδων και σε προβλήματα μεγαλυτέρων διαστάσεων.

Οστόσο, υπάρχουν ορισμένα ανοιχτά θέματα στην περιοχή, τα οποία χρίζουν περαιτέρω μελέτης και μπορούν να θεωρηθούν επεκτάσεις των μεθόδων και των προοπτικών της παρούσας διατριβής. Έτσι, προτείνονται τα ακόλουθα:

- Ο τρόπος που επηρεάζει το διάνυσμα επικοινωνίας (communication vector) το χρόνο εκτέλεσης. Το ζητούμενο είναι το διάνυσμα επικοινωνίας να επιλέγεται ούτως ώστε να ελαχιστοποιείται ο χρόνος αυτός.
- Η ανάπτυξη μεθόδων απεικόνισης των γεωμετρικών αλγορίθμων σε διατάξεις πολυεπεξεργασίας. Πιθανές τέτοιες διατάξεις είναι τόσο οι παραδοσιακές συστολικές, όσο και ορισμένες σύχρονες διατάξεις μνημών, οι οποίες έχουν τη δυνατότητα στοιχειώδους επεξεργασίας σε κάθε στοιχείο τους.

Παράρτημα Α'

Αντιστοιχία Ελληνικών-Αγγλικών Όρων

Για διευκόλυνση του αναγνώστη, παραθέτουμε ένα συγκεντρωτικό πίνακα αντιστοιχισης των ελληνικών όρων που εμφανίζονται στο κείμενο και των αντίστοιχων αγγλικών. Έγινε προσπάθεια να ακολουθηθεί η ορολογία που χρησιμοποιείται σε αντίστοιχα ελληνικά κείμενα. Παρατίθενται αλφαριθμητικά.

Ελληνικός Όρος	Αγγλικός Όρος	Συμβολισμός
απεικόνιση	mapping	
γεωμετρικές περιοχές	regions	$\mathcal{R}_i, i = 1, 2, \dots$
γραμμική δρομολόγηση	linear schedule	Π
γραμμικός μετασχηματισμός	linear transformation	
διαμέριση	partitioning	
διάνυσμα επικοινωνίας	communication vector	\mathbf{d}_c
διανύσματα εξάρτησης	dependence vectors	$\mathbf{d}_i, 1 \leq i \leq m$
διανύσματα κυματομορφής	pattern vectors	$\mathbf{p}_i, 1 \leq i \leq k$

Πίνακας Α'.1: Αντιστοιχία ελληνικών-αγγλικών όρων στην παρούσα διατριβή.

Ελληνικός 'Όρος	Αγγλικός 'Όρος	Συμβολισμός
διανύσματα κώνων	cone vectors	$\mathbf{c}_i, 1 \leq i \leq r$
διαφορικές εξαρτήσεις	difference dependence vectors	$\mathbf{a}_i, \mathbf{b}_i, 1 \leq i \leq k - 1$
δρομολόγηση/χρονοδρομολόγηση	scheduling	
εξαρτήσεις	dependencies	$\mathbf{d}_i, 1 \leq i \leq m$
εξωτερικές πλευρές	outer facets	
εσωτερικές πλευρές	inner facets	
ιδεατός χώρος επεξεργαστών	virtual array of cells	J'
μετασχηματισμένος χώρος δεικτών	transformed index space	J'
ομοιόμορφες εξαρτήσεις	uniform dependencies	
περιγράμματα εκτέλεσης	pattern outlines	$pat_j, j = 0, 1, 2, \dots$
περιγράμματα σχημάτων κυματομορφής	pattern outlines	$pat_j, j = 0, 1, 2, \dots$
περιοχές	regions	$\mathcal{R}_i, i = 1, 2, \dots$
περιοχές κώνων	cone regions	
πλέγμα επεξεργαστών	mesh, grid	
πλευρές	facets	
σημεία κυματομορφής	pattern points	
σχήματα κυματομορφής εκτέλεσης	patterns	$Pat_j, j = 0, 1, 2, \dots$
τετριμμένος κώνος	trivial cone	
τμήμα του χώρου δεικτών	cluster	
υπερεπίπεδο	hyperplane	
φωλιασμένος βρόχος	nested loop	
χρονικές περιοχές/υποπεριοχές	time subspaces	$ECT_i, i = 1, 2, \dots$
χώρος δεικτών	index space	J^n

Πίνακας Α'.2: Αντιστοιχία ελληνικών-αγγλικών όρων στην παρούσα διατριβή (συνέχεια).

Παράρτημα Β'

Οι Αλγόριθμοι Κυρτού Περιγράμματος DivideHull2d και QuickHull

Για λόγους πληρότητας και λόγω της αντιστοιχίας με τους αλγορίθμους προσδιορισμού των διανυσμάτων κώνων (cone vectors) –Αλγόριθμοι 5.2 και 7.1-7.3– παρουσιάζουμε τους αλγορίθμους κυρτού περιγράμματος DivideHull2d και QuickHull.

Παρουσιάζουμε επίσης, μια ολοκληρωμένη εφαρμογή που υλοποιεί τον υπολογισμό κυρτών περιγραμάτων με τη μέθοδο QuickHull. Η εφαρμογή αυτή ακούει στο όνομα qhull και ένα μέρος της χρησιμοποιείται από την πλατφόρμα crazySched που υλοποιήσαμε.

B'.1 Ο Αλγόριθμος DivideHull2d

Ο εν λόγω αλγόριθμος ονομάζεται *Simplified “Ultimate Planar Convex Hull Algorithm”* στη σχετική βιβλιογραφία και παρουσιάστηκε από τον Chan [Cha95]. Αποτελεί απλοποίηση της μεθόδου που πρωτοπαρουσιάστηκε το 1986 από τους Kirkpatrick και Seidel [KS86], με τίτλο “The Ultimate Planar Convex Hull Algorithm”.

Η πολυπλοκότητά του είναι η βέλτιστη για τις δύο διαστάσεις $\mathcal{O}(n \log h)$, όπου n ο αριθμός των σημείων για τα οποία αναζητούμε το κυρτό περίγραμμα και h ο αριθμός των σημείων που συμμετέχουν σ' αυτό.

Όπως θα φανεί στη συνέχεια, κατά την εν λόγω μέθοδο το κυρτό περίγραμμα υπολογίζεται σε δύο φάσεις. Στην πρώτη φάση υπολογίζεται το άνω κυρτό περίγραμμα και στη δεύτερη φάση το κάτω. Οι φάσεις είναι εντελώς αντίστοιχες, γι' αυτό και περιοριζόμαστε στην παρουσίαση της μιας φάσης μόνο.

Ως P θεωρούμε το σύνολο εισόδου· βιοηθητικά ορίζονται ως \mathbf{p}_l και \mathbf{p}_r τα δύο ακραία σημεία (δεξιά και αριστερό αντίστοιχα) του συνόλου εισόδου P . Ορίζεται επίσης, το σύνολο P^* των σημείων του P εκτός των \mathbf{p}_l και \mathbf{p}_r : $P^* = P - \{\mathbf{p}_l, \mathbf{p}_r\}$. Για δύο σημεία \mathbf{p} και \mathbf{q} , ως $\overline{\mathbf{pq}}$ ορίζεται το ευθύγραμμο τμήμα που ενώνει τα δύο σημεία και ως p_x, p_y οι συντεταγμένες x και y αντίστοιχα του σημείου \mathbf{p} . Σημειώνουμε ότι ο αλγόριθμος λειτουργεί αναδρομικά.

DivideHull2d(P^* , \mathbf{p}_l , \mathbf{p}_r)

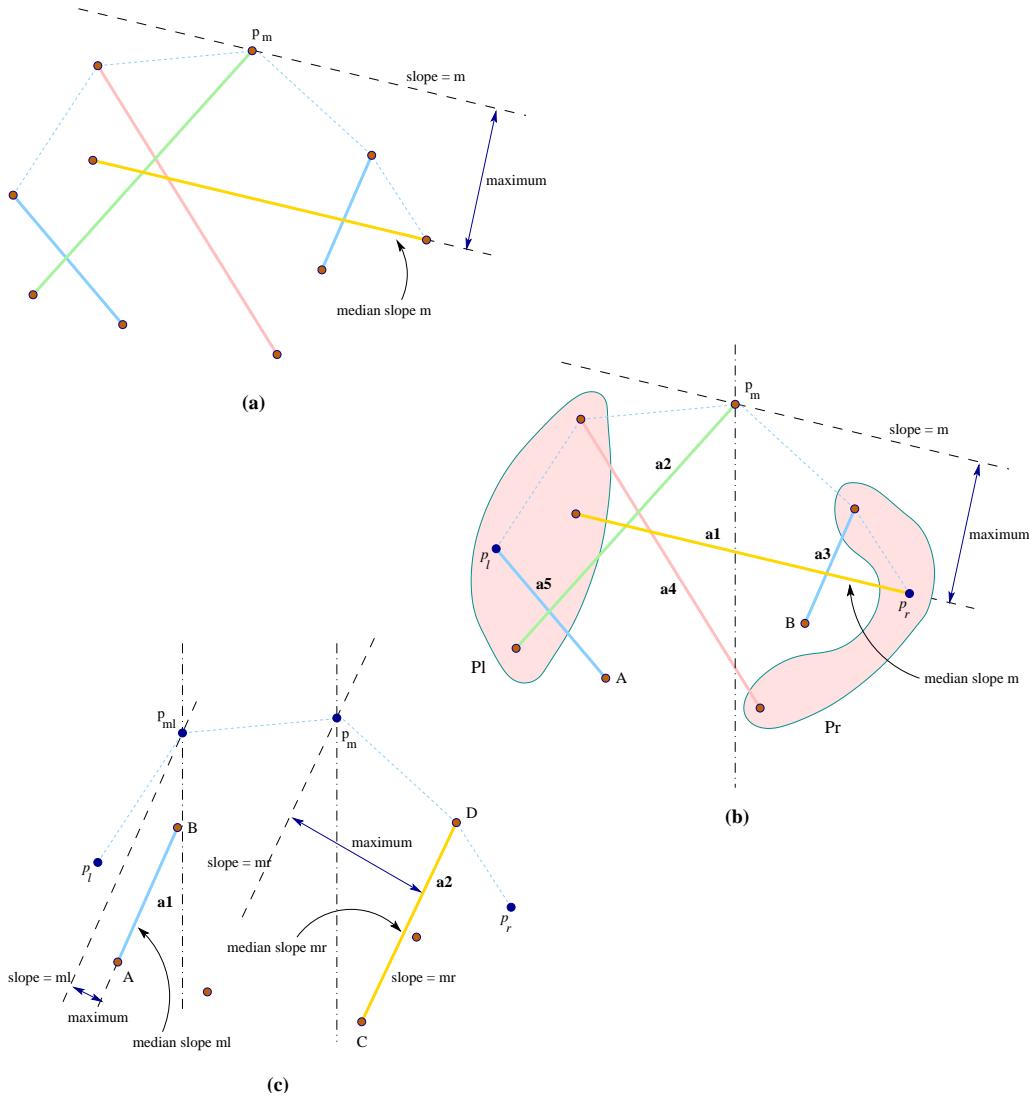
1986-95

ΕΙΣΟΔΟΣ: δεδομένου συνόλου σημείων $P \subseteq \mathbb{N}^2$, με $|P| = n$ και $P^* = P - \{\mathbf{p}_l, \mathbf{p}_r\}$

ΕΞΟΔΟΣ: μια ακολουθία ακμών, η οποία αποτελεί το άνω κυρτό περίγραμμα του συνόλου P

1. Αφαιρούνται από το P^* όλα τα σημεία τα οποία βρίσκονται κάτω από το $\overline{\mathbf{p}_l \mathbf{p}_r}$
2. Ελέγχεται το σύνολο P^* :
 - αν $P = \emptyset$, τότε επιστρέφεται το $\langle \overline{\mathbf{p}_l \mathbf{p}_r} \rangle$
 - αν $P = \{\mathbf{p}\}$, τότε επιστρέφεται το $\langle \overline{\mathbf{p} \mathbf{p}}, \overline{\mathbf{p} \mathbf{p}} \rangle$
3. Επιλέγονται με τυχαίο τρόπο $\lfloor n/2 \rfloor$ ξεχωριστά ζευγάρια σημείων $\{(s_1, t_1), \dots, (s_{\lfloor n/2 \rfloor}, t_{\lfloor n/2 \rfloor})\}$ από το P^* και ταξινομούνται ούτως ώστε: $s_{ix} < t_{ix}$
4. Έστω $m_i = (t_{iy} - s_{iy}) / (t_{ix} - s_{ix})$, $i = 1, 2, \dots, \lfloor n/2 \rfloor$ και m ο μέσος των $\langle m_1, m_2, \dots, m_{\lfloor n/2 \rfloor} \rangle$
5. Έστω \mathbf{p}_m το σημείο του P το οποίο μεγιστοποιεί τη διαφορά $p_{my} - m \cdot p_{mx}$
6. Έστω $P_l^* = \{\mathbf{p} \in P^* : p_x < p_{mx}\} - \{t_i : m_i \leq m\}$ και
 $P_r^* = \{\mathbf{p} \in P^* : p_x > p_{mx}\} - \{s_i : m_i \geq m\}$
7. Ελέγχεται το p_m :
 - αν $\mathbf{p}_m = \mathbf{p}_r$, τότε επιστρέφεται το DivideHull2d(P_l^* , \mathbf{p}_l , \mathbf{p}_r)
 - αν $\mathbf{p}_m = \mathbf{p}_l$, τότε επιστρέφεται το DivideHull2d(P_r^* , \mathbf{p}_l , \mathbf{p}_r)
 - διαφορετικά, επιστρέφεται η ένωση των
 $\text{DivideHull2d}(P_l^*, \mathbf{p}_l, \mathbf{p}_m)$ και $\text{DivideHull2d}(P_r^*, \mathbf{p}_m, \mathbf{p}_r)$

Στο σχήμα B'.1 εξηγείται ο παραπάνω αλγόριθμος για ένα σύνολο 10 σημείων στις δύο διαστάσεις. Ο αλγόριθμος εφαρμόζεται στα επιμέρους υποσύνολα του αρχικού. Παρουσιάζονται οι δύο πρώτες επαναλήψεις.



Σχήμα B'.1: Τα τρία πρώτα βήματα του Απλοποιημένου “Ultimate Convex Hull Algorithm” για δεδομένο σύνολο σημείων στο επίπεδο. **a)** Το σύνολο των σημείων χωρίζεται σε ζεύγη, τα οποία σχηματίζουν ευθύγραμμα τμήματα με αντίστοιχες κλίσεις. Υπολογίζεται το ευθύγραμμο τμήμα με τη μέση κλίση m καθώς και το σημείο p_m με την μεγαλύτερη απόσταση από το τμήμα αυτό. **b)** Με βάση το σημείο p_m χωρίζεται το αρχικό σύνολο P στα P_l και P_r , ανάλογα με τις κλήσεις των τμημάτων που ανήκουν και τη θέση τους ως προς το p_m . **c)** Η διαδικασία επαναλαμβάνεται για τα δύο υποσύνολα (divide-and-conquer).

B'.2 Ο Αλγόριθμος QuickHull

Ο αλγόριθμος QuickHull μελετήθηκε και ανακαλύφτηκε ανεξάρτητα από διάφορους ερευνητές κατά το τέλος της δεκαετία του '70. Οι παράλληλες προσεγγίσεις σημειώνονται στην πρόσφατη εργασία [BDH96], όπου παρουσιάζεται ο γενικός αλγόριθμος και για μεγαλύτερες διαστάσεις. Η πολυπλοκότητα του, για τις δύο διαστάσεις, είναι $\mathcal{O}(nr)$, όπου r , ο αριθμός των κορυφών που επεξεργάζεται ο αλγόριθμος. Σημειώνουμε ότι είναι ο πρώτος αλγόριθμος που παρουσιάστηκε ο οποίος εξαρτάται από την έξοδο. Στην παρούσα εργασία αναφερόμαστε στην εν λόγω παραλλαγή των Barder, Dobkin και Hundanpaa [BDH96], τόσο επειδή είναι η πιο πρόσφατη, όσο και γιατί περιγράφει τον γενικευμένο αλγόριθμο για διάσταση $d \geq 2$.

Ο αλγόριθμος αυτός σε κάθε βήμα προσθέτει μια κορυφή στο τμήμα του κυρτού περιγράμματος που έχει ήδη υπολογιστεί: δουλεύει δηλαδή αυξητικά (incremental). Εφαρμόζει την τακτική *beneath-beyond* [Grü67, BDH96] για τον εντοπισμό της επόμενης πιθανής κορυφής, αναζητώντας την πιο απομακρυσμένη σύμφωνα με συγκεκριμένα κριτήρια.

Η πολυπλοκότητα του εξαρτάται από την έξοδο και είναι $\mathcal{O}(n \log r)$ για $d \leq 3$, όπου r ο αριθμός των σημείων που επεξεργάζεται ο αλγόριθμος. Εν γένει τα σημεία αυτά είναι περισσότερα από h , από τα σημεία δηλαδή, τα οποία αποτελούν το κυρτό περίγραμμα –την έξοδο.

Ο αλγόριθμος εφαρμόζεται και σε μεγαλύτερες διαστάσεις, με αρκετά χειρότερη πολυπλοκότητα, βέβαια: $\mathcal{O}(n \cdot f_r/r)$ για $d \geq 4$, όπου f_r ο μέγιστος αριθμός πλευρών για r σημεία. Σημειώνουμε ότι με τον όρο πλευρές αναφερόμαστε σε πολύγωνα ($d - 1$)-διαστάσεων.

QuickHull

1976-96

ΕΙΣΟΔΟΣ: ένα σύνολο n σημείων στον d -διάστατο χώρο

ΕΞΟΔΟΣ: το υποσύνολο των σημείων αυτών, το οποίο αποτελεί το κυρτό περίγραμμα του αρχικού συνόλου

1. Επιλέγουμε τυχαία $d + 1$ σημεία και κατασκευάζουμε τις (τριγωνικές) πλευρές που σχηματίζονται από αυτά. Ο αριθμός των πλευρών αυτών είναι:

$$\frac{(d+1)!}{3! \cdot (d+1-3)!} = \frac{(d+1)!}{6 \cdot (d-2)!} = \frac{d}{6} \cdot (d^2 - 1)$$

Κατασκευάζεται έτσι, το (αρχικό) κυρτό περίγραμμα των $d + 1$ σημείων.

2. Για κάθε μία από τις σχηματιζόμενες πλευρές (*faces*) F_i :

- ελέγχουμε τα εναπομείναντα σημεία. Αν βρίσκονται επάνω από την πλευρά, τοποθετούνται στο σύνολο *εξωτερικών σημείων* της $out(F_i)$ –σημειώνουμε ότι το κάθε σημείο εισάγεται σε ένα τέτοιο σύνολο και μόνον

3. Για κάθε πλευρά F_i με μη κενό σύνολο εξωτερικών σημείων $out(F_i)$:

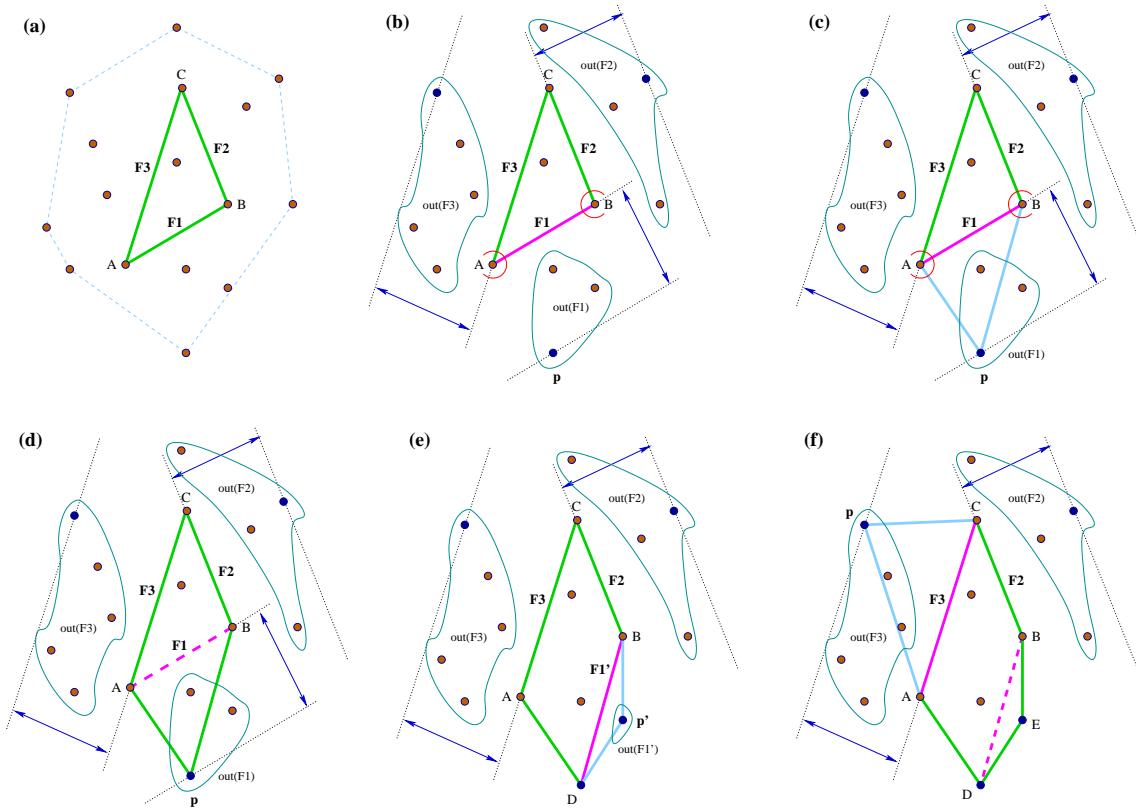
- βρίσκουμε το σημείο p του $out(F_i)$ με την μεγαλύτερη απόσταση από την πλευρά F_i
- αρχικοποιούμε το σύνολο των ορατών πλευρών V_p από το p ως εξής: $V_p = \{F_i\}$
- για κάθε γειτονική πλευρά N των πλευρών του V_p
 - αν το σημείο p είναι πάνω από την N , η πλευρά N προστίθεται στο V_p
 - δημιουργείται έτσι η ευρύτερη ορατή περιοχή πλευρών από το σημείο p
- το σύνορο του V_p λέγεται ορίζοντας του p , και δεν είναι άλλο από το σύνολο των ακμών, οι οποίες περιβάλλουν τις ορατές (από το p) πλευρές· έστω H το σύνολο αυτό
- για κάθε ακμή R του H :
 - κατασκευάζεται μια νέα πλευρά από τα άκρα του R και το p , η οποία και συνδέεται με τις γειτονικές τις πλευρές
- για κάθε μία από τις νέες πλευρές F'_j που δημιουργούνται:
 - ελέγχονται όλα τα εναπομείναντα σημεία των συνόλων εξωτερικών σημείων των πλευρών που περιέχονται στο V_p : αν βρίσκονται επάνω από την πλευρά F'_j , τοποθετούνται στο σύνολο εξωτερικών της σημείων $out(F'_j)$
- αφαιρούνται από το κυρτό περίγραμμα οι πλευρές που ανήκουν στο V_p

4. Αν υπάρχει πλευρά με μη κενό σύνολο εξωτερικών σημείων, το βήμα 3 επαναλαμβάνεται· σε διαφορετική περίπτωση, το ζητούμενο κυρτό περίγραμμα έχει υπολογιστεί.

Ο αλγόριθμος επεξηγείται στο σχήμα Β'.2, όπου εφαρμόζεται σε ένα δεδομένο σύνολο σημείων στο επίπεδο ($d = 2$).

Η πολυπλοκότητα του αλγορίθμου παραμένει $\mathcal{O}(n \log r)$ και στις τρεις διαστάσεις, για σύνολο σημείων P μεγέθους n , όπου r τα σημεία που επεξεργάζεται ο αλγόριθμος. Θα περιγράψουμε τις διαφορές της διαδικασίας στις μεγαλύτερες διαστάσεις, από την αντίστοιχη στις δύο διαστάσεις, σχηματικά. Τόσο ο αλγόριθμος στις τρεις διαστάσεις, όσο και κάποιες δομές δεδομένων για την πρακτική υλοποίηση του, περιγράφονται αναλυτικά στα [BDH96, dBvKOS97].

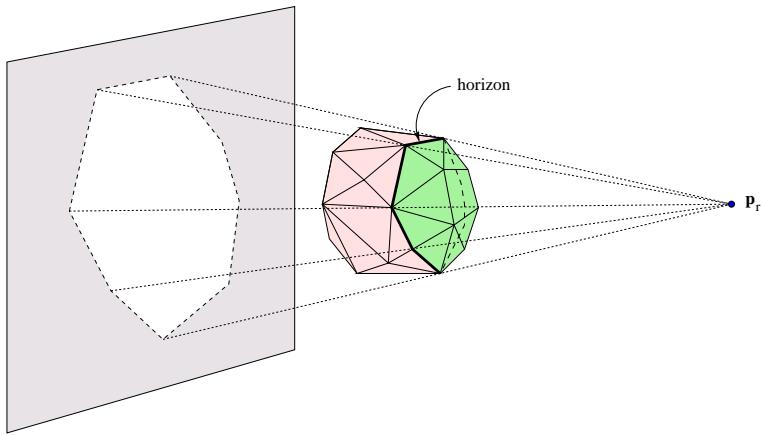
Η κύρια διαφορά με τις δύο διαστάσεις έγκειται στον κανονισμό της ορατής περιοχής και του ορίζοντα (βλέπε σχήμα Β'.3). Θεωρούμε σύνολο $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\} \subset E^3$. Ο αλγόριθμος αρχίζει επιλέγοντας τέσσερα σημεία του P , μη συνεπίπεδα. Αυτό μπορεί να γίνει πολύ εύκολα, ες εξής. Επιλέγονται τα δύο πρώτα σημεία του P , \mathbf{p}_1 και \mathbf{p}_2 . Έπειτα, διατρέχονται τα υπόλοιπα σημεία έως ότου να βρεθεί ένα σημείο \mathbf{p}_3 μη συνευθειακό με τα δύο πρώτα. Κατόπιν, αναζητείται το τέταρτο σημείο \mathbf{p}_4 , για το οποίο απαιτείται να μην είναι συνεπίπεδο με τα τρία πρώτα. Χωρίς βλάβη της γενικότητας, θεωρούμε ότι τα πρώτα τέσσερα σημεία του P πληρούν τα παραπάνω



Σχήμα B'.2: Τα έξι πρώτα βήματα του αλγορίθμου QuickHull για δεδομένο σύνολο σημείων στο επίπεδο.
a) Επιλέγονται τα σημεία A, B και C ως αρχικό κυρτό περίγραμμα. **b)** Για τις σχηματιζόμενες πλευρές F_1, F_2 και F_3 , δημιουργούνται τα αντίστοιχα σύνολα εξωτερικών σημείων και σημειώνονται τα ακρότατα σημεία· για την F_1 είναι το σημείο p . **c,d)** Για την πλευρά F_1 , από το p και τον ορίζοντα του (ακμή \overline{AB}) σχηματίζονται δυο νέες πλευρές, οι οποίες προστίθενται στο περίγραμμα, από το οποίο αφαιρείται η πλευρά F_1 . **e)** Η διαδικασία συνεχίζεται για τη νέα πλευρά F'_1 : ακρότατο σημείο είναι το p' , μέσω του οποίου προστίθενται δυο ακόμη πλευρές \overline{ED} και \overline{EB} , ενώ αφαιρείται η \overline{BD} . **f)** Ο αλγόριθμος συνεχίζεται για την πλευρά F_3 .

—είναι μη συνεπίπεδα. Θεωρούμε, επίσης, ότι τα εναπομείναντα σημεία p_5, \dots, p_n είναι σε τυχαία σειρά.

Ο αλγόριθμος λειτουργεί αυξητικά (incrementally), δουλεύοντας με ένα σημείο από τα εναπομείναντα του P , κάθε φορά. Για δεδομένο ακέραιο $r \geq 1$, έστω $P_r = \{p_1, p_2, \dots, p_r\}$. Σε κάποιο βήμα του αλγορίθμου, έστω r , πρέπει να προστεθεί το σημείο p_r στο κυρτό περίγραμμα του



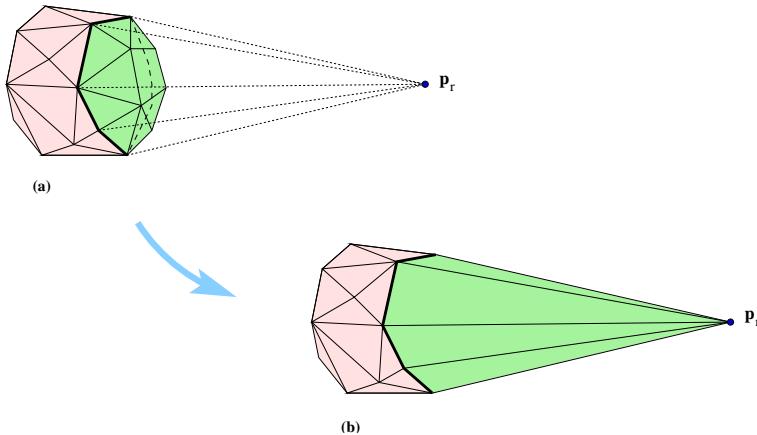
Σχήμα B'.3: Ο ορίζοντας και η ορατή περιοχή από ένα σημείο σε ένα πολύτοπο τριών διαστάσεων.

P_{r-1} , με άλλα λόγια, να μετασχηματιστεί το $\text{conv}(P_{r-1}) = \mathcal{CH}(P_{r-1})$ σε $\text{conv}(P_r) = \mathcal{CH}(P_r)$. Προφανώς, προκύπτουν δύο περιπτώσεις:

- Αν το p_r βρίσκεται εντός του $\mathcal{CH}(P_{r-1})$ ή επάνω στο σύνορο του, τότε $\mathcal{CH}(P_{r-1}) = \mathcal{CH}(P_r)$, οπότε δεν απαιτείται να γίνει κάτι επιπλέον.
- Υποθέτουμε, τώρα, ότι το p_r βρίσκεται εκτός του $\mathcal{CH}(P_{r-1})$. Στην περίπτωση αυτή, βλέποντας από το p_r το κυρτό περίγραμμα $\mathcal{CH}(P_{r-1})$, κάποιες πλευρές του θα είναι ορατές και κάποιες όχι. Οι ορατές πλευρές σχηματίζουν μια επιφάνεια, η οποία καλείται ορατή περιοχή του p_r στο κυρτό περίγραμμα $\mathcal{CH}(P_{r-1})$. Το περίγραμμα αυτής της περιοχής στο $\mathcal{CH}(P_{r-1})$ καλείται ορίζοντας του p_r (βλέπε σχήμα B'.3).

Ο ορίζοντας του p_r αποτελεί καθοριστικό παράγοντα για τη μετατροπή του $\mathcal{CH}(P_{r-1})$ σε $\mathcal{CH}(P_r)$: αποτελεί το όριο μεταξύ του τμήματος του $\mathcal{CH}(P_{r-1})$ το οποίο παραμένει –οι μη ορατές πλευρές– και του τμήματος το οποίο πρέπει να αντικατασταθεί –οι ορατές περιοχές. Οι ορατές πλευρές πρέπει να αντικατασταθούν από νέες, οι οποίες θα συνδέουν το σημείο p_r με τις ακμές του ορίζοντα (σχήμα B'.4).

Το πρόβλημα το οποίο ανακύπτει είναι ο αποδοτικός προσδιορισμός των ορατών περιοχών, για δεδομένο σημείο του P . Η προφανής λύση του προβλήματος, είναι να ελέγξουμε το σημείο το οποίο θέλουμε να προσθέσουμε, ως προς κάθε πλευρά του ήδη δημιουργημένου κυρτού περιγράμματος. Για το σημείο p_r απαιτείται κόστος $\mathcal{O}(r)$, δηλαδή συνολικά έχουμε πολυπλοκότητα $\mathcal{O}(n^2)$.



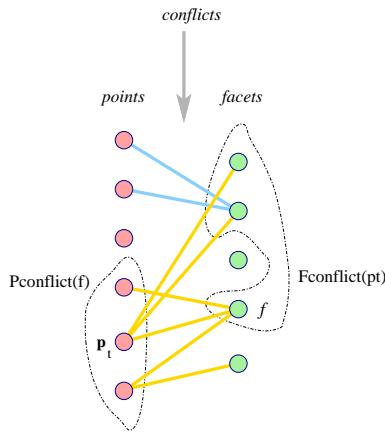
Σχήμα B'.4: Δημιουργία του $\mathcal{CH}(P_r)$ από το $\mathcal{CH}(P_{r-1})$, προσθέτοντας το σημείο p_r . Αφού βρεθεί ο ορίζοντας του p_r στο $\mathcal{CH}(P_{r-1})$, αφαιρείται η ορατή περιοχή από το κυρτό περίγραμμα (a). Κατόπιν, προστίθενται νέες πλευρές, οι οποίες σχηματίζονται από το σημείο p_r και τις ακμές του ορίζοντα (b).

Το εν λόγω πρόβλημα μπορεί να λυθεί αποδοτικά κρατώντας επιπλέον πληροφορίες, τόσο για τα σημεία του P , όσο και για τις πλευρές του $\mathcal{CH}(P_{r-1})$. Οι πληροφορίες αυτές θα πρέπει να ενημερώνονται σε κάθε βήμα του αλγορίθμου. Πολλοί ερευνητές για το σκοπό αυτό, χρησιμοποιούν μια επιπλέον δομή, η οποία καλείται γράφος συγκρούσεων (*conflict-graph*).¹

Για κάθε πλευρά f του τρέχοντος κυρτού περιγράμματος $\mathcal{CH}(P_r)$, αποθηκεύεται το σύνολο $P_{conflict}(f) \subseteq \{p_{r+1}, p_{r+2}, \dots, p_n\}$, το οποίο περιέχει τα σημεία τα οποία βλέπουν την f (και δεν έχουν προστεθεί ακόμη στο κυρτό περίγραμμα). Αντίστοιχα, για κάθε σημείο p_t , με $t > r$ (σημείο δηλαδή που δεν έχει ελεγχθεί ακόμη για το κυρτό περίγραμμα), αποθηκεύεται το σύνολο $F_{conflict}(p_t)$ των πλευρών του $\mathcal{CH}(P_r)$, οι οποίες είναι ορατές από αυτό. Λέμε ότι ένα σημείο $p \in P_{conflict}(f)$ είναι σε σύγκρουση με την πλευρά f , γιατί δεν μπορούν να συνυπάρξουν στο κυρτό περίγραμμα — με την πρόσθεση οποιοδήποτε σημείου $p \in P_{conflict}(f)$, η πλευρά f πρέπει να αφαιρεθεί.

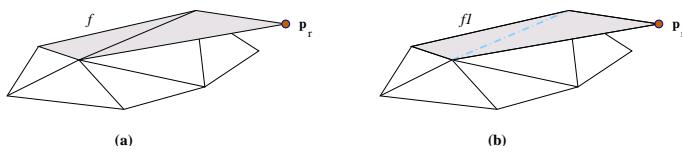
Τα σύνολα συγκρούσεων διατηρούνται στον γράφο συγκρούσεων \mathcal{G} , ως εξής. Πρόκειται για διμερή (*bipartite*) γράφο, ο οποίος περιέχει ένα σύνολο κόμβων για τα σημεία του P , τα οποία δεν έχουν εισαχθεί ακόμη στο κυρτό περίγραμμα και ένα δεύτερο σύνολο κόμβων για τις πλευρές του μέχρι τώρα κυρτού περιγράμματος. Οι συγκρούσεις μεταξύ σημείων και πλευρών σημειώνονται ως ακμές του γράφου (σχήμα B'.5).

¹Η δομή αυτή περιγράφεται αναλυτικά στο [dBvKOS97].



Σχήμα B'.5: Γράφος συγκρούσεων κατά τη διάρκεια κατασκευής κυρτού περιγράμματος συνόλου σημείων. Για το σημείο p_t , σημειώνεται το σύνολο $F_{conflict}(p_t)$ και αντίστοιχα για την πλευρά f , το σύνολο $P_{conflict}(f)$.

Στην παραπάνω ανάλυση υπάρχει ένα σημείο στο οποίο δεν έγινε αναφορά: τί γίνεται στην περίπτωση όταν οποία προστίθεται μια πλευρά συνεπίπεδη με μια ήδη υπάρχουσα. Στην περίπτωση αυτή ωστόπει η νέα τριγωνική πλευρά να ενωθεί με την ήδη υπάρχουσα. Στο σχήμα B'.6 παρουσιάζεται μια τέτοια περίπτωση. Η νέα τριγωνική πλευρά ωστόπει να ενωθεί με την υπάρχουσα συνεπίπεδη της f και να αφαιρεθούν οι επιπλέον ακμές.



Σχήμα B'.6: Συνεπίπεδες πλευρές κατά την δημιουργία κυρτού περιγράμματος. Η νέα τριγωνική πλευρά ενώνεται με την υπάρχουσα f και δημιουργείται η νέα (μη τριγωνική πλέον) πλευρά f_1 .

Η πολυπλοκότητα του αλγορίθμου για μεγαλύτερες διαστάσεις είναι $\mathcal{O}(n \cdot f_r / r)$, για διάσταση $d \geq 4$, όπου f_r ο μέγιστος αριθμός πλευρών για r σημεία.

B'.3 Το πρόγραμμα qhull

Για τον ενδιαφερόμενο αναγνώστη που δεν θέλει να μείνει μόνο στο θεωρητικό μέρος των κυρτών περιγραμμάτων και των πολυτόπων, σημειώνουμε το πρόγραμμα `qhull`. Δεν είναι παρά μια υλοποίηση του αλγορίθμου QuickHull, η οποία συνάμα δημιουργεί και όλες τις δομές δεδομένων που συνήθως απαιτούνται (λίστα κορυφών, πλευρών, λίστα γειτονικών πλευρών για κάθε πλευρά, κ.α.).

Το εν λόγω πρόγραμμα αναπτύχθηκε στο πανεπιστήμιο της Minnesota και επιτρέπει τον υπολογισμό κυρτών περιγραμμάτων, διαγραμμάτων Voronoi και τριγωνοποιήσεων Delaunay. Ακούει στα ονόματα `qhull` και `qconvex` και είναι προσπελάσιμο μέσω των ακόλουθων διευθύνσεων:

```
<http://www.geom.umn.edu/locate/qhull>
<ftp://geom.umn.edu/pub/software/qhull3-0.zip>
<http://www.geom.umn.edu/~bradb/qhull-news.html>
```

Αξίζει να πειραματιστείτε! Το πρόγραμμα είναι ιδιαίτερα εύχρηστο λόγω των χαρακτηριστικών του: δέχεται είσοδο από αρχείο, δουλεύει για οποιαδήποτε διάσταση και η έξοδος του καθορίζεται από τον χρήστη (αν θέλει μόνο σημεία, ή μόνο πλευρές ή/και ακμές, κ.α.). Η λεπτομέρειά της καθορίζεται επίσης, από τον χρήστη (λεπτομέρειες γειτνίασης πλευρών-κορυφών, πλευρών-ακμών, κ.α.).

Στο πλαίσιο της παρούσας διατριβής χρησιμοποιήθηκε κατά κόρον η εφαρμογή `qconvex` του προγράμματος αυτού, για τον προσδιορισμό των διαγυσμάτων των κώνων. Αυτό έγινε μέσω του προγράμματος `loop` το οποίο περιγράφεται στην Παράγραφο 8.2.

Βιβλιογραφία

- [AKPT95] T. Andronikos, N. Koziris, G. Papakonstantinou, and P. Tsanakas. An efficient algorithm for the optimal linear schedule of uniform dependence algorithms. In *Proceedings of the First ECPD International Conference on Advanced Robotics and Intelligent Automation*, pages 140–145, Athens, Greece, 1995.
- [AKPT99] T. Andronikos, N. Koziris, G. Papakonstantinou, and P. Tsanakas. Optimal scheduling for UET/UET-UCT generalized n -dimensional grid task graphs. *Journal of Parallel and Distributed Computing*, 57(2):140–165, May 1999.
- [AKT⁺95] T. Andronikos, N. Koziris, Z. Tsatsoulis, G. Papakonstantinou, and P. Tsanakas. Lower bounds for parallel time and number of processors for the class of uniform dependence algorithms. Technical Report DSCLAB-TR-95-03, Digital Systems and Computers Laboratory - N.T.U.A., 1995.
- [AR96] N. M. Amato and E. A. Ramos. On computing voronoi diagrams by divide-prune-and-conquer. In *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, pages 166–175, 1996.
- [Bay99] V. Bayer. Survey of algorithms for the convex hull problem, March 1999. Department of Computer Science, Oregon State University.
- [BDH96] C. B. Barber, D. P. Dobkin, and H. Hundanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, December 1996.
- [BNO01] D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convexity, Duality and Lagrange Multipliers*. Lecture Notes, Massachusetts Institute of Technology, Spring 2001.

- [CC91] J. Y. Colin and P. Chretienne. CPM scheduling with small communication delays and task duplication. *European Journal of Operational Research*, 39:680–684, 1991.
- [Cha93] B. Chazelle. An optimal convex hull algorithm for point sets in any fixed dimension. *Discrete & Computational Geometry*, 10:377–409, 1993.
- [Cha95] T. M. Chan. *Output-Sensitive Construction of Convex Hulls*. PhD thesis, The University of British Columbia, October 1995.
- [CK70] D. R. Chand and S. S. Kapur. An algorithm for convex polytopes. *Journal of the Association of Computing Machinery*, 17:78–86, 1970.
- [Cla94] K. L. Clarkson. More output-sensitive geometric algorithms. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 695–702, 1994.
- [CM95] B. Chazelle and J. Matoušek. Derandomizing an output-sensitive convex hull algorithm in three dimensions. *Computational Geometry: Theory and Applications*, 5:27–32, 1995.
- [CS89] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, ii. *Discrete & Computational Geometry*, 4:387–421, 1989.
- [CSY95] T. M. Chan, J. Snoeyink, and C.-K. Yap. Output-sensitive construction of polytopes in four dimensions and voronoi diagrams in three. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 282–291, 1995.
- [DAA⁺01] I. Drositis, T. Andronikos, A. Alexandridi, G. Papakonstantinou, and N. Koziaris. Geometric pattern prediction and scheduling of uniform dependence loops. In *Proceedings of the 5th Hellenic European Conference on Computer Mathematics and its Applications - HERCMA 2001*, Athens, 2001.
- [DAK⁺01] I. Drositis, T. Andronikos, A. Kokorogiannis, G. Papakonstantinou, and N. Koziaris. Geometric scheduling of 2-D uniform dependence loops. In *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS 2001*, pages 259–264, Korea, 2001.

- [DAK⁺02] I. Drositis, I. Andronikos, M. Kalathas, G. Papakonstantinou, and N. Koziris. Optimal loop parallelization in n -dimensional index spaces. In (*to appear*) *The 2002 International Conference on Parallel and Distributed Processing Techniques and Applications - PDPTA'02*, Las Vegas, Nevada, USA, 2002.
- [DAM⁺02] I. Drositis, T. Andronikos, G. Manis, G. Papakonstantinou, and N. Koziris. Geometric scheduling of 2-D UET/UET-UCT uniform dependence loops. In *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing - PDP 2002*, pages 343–350, Gran Canaria Island, Spain, 2002. IEEE Press.
- [DAP00] I. Drositis, T. Andronikos, and G. Papakonstantinou. 2-D pattern prediction and scheduling. Technical Report CSLAB-TR-00-01, Computing Systems Laboratory - N.T.U.A., 2000.
- [dBvKOS97] M. de Berg, M. van Kreveld, M. Overmans, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, 1997.
- [DGK⁺00] I. Drositis, G. Goumas, N. Koziris, P. Tsanakas, and Papakonstantinou. Evaluation of loop grouping methods based on orthogonal projection spaces. In *Proceedings of the 2000 International Conference on Parallel Processing ICPP-2000*, pages 469–476, Toronto, Canada, 2000. IEEE Press.
- [DK83] D.P. Dobkin and D. G. Kirkpatrick. Fast detection of polyedral intersection. *Theoretical Computer Science*, 27:241–253, 1983.
- [DK90] D.P. Dobkin and D. G. Kirkpatrick. Determining the separation of preprocessed polyedra: a unified approach. In *Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, volume 443, pages 440–443, Lecture Notes in Computer Science, 1990. Springer-Verlag.
- [DKPT99] I. Drositis, N. Koziris, G. Papakonstantinou, and P. Tsanakas. A systolic approach to loop partitioning and mapping into fixed size distributed memory architectures. In *Proceedings of the 7th Panhellenic Conference on Informatics*, pages IV18–IV28, Ioannina, Greece, 1999. University of Ioannina.
- [DR91] L. Khachiyan Darte and Y. Robert. Linear scheduling is nearly optimal. *Parallel Processing Letters*, 1.2:73–81, 1991.

- [DR94] A. Darte and Y. Robert. Mapping uniform loop nests onto distributed memory architectures. *Parallel Computing*, 20:679–710, 1994.
- [DR95] A. Darte and Y. Robert. Affine-by-statement scheduling of uniform and affine loops nests over parametric domains. *Journal of Parallel and Distributed Computing*, 29:43–59, 1995.
- [Dro01] I. Drositis. Survey of algorithms for the convex hull problem and their complexity, June 2001. Department of Computer Science, National Technical University of Athens.
- [EC92] I. Emiris and J. Canny. An efficient approach to removing geometric degeneracies. In *Proceedings of the Eighth Annual ACM Symposium on Computational Geometry*, pages 74–82, 1992.
- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, 1987.
- [EM90] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics*, 9(1):66–104, 1990.
- [Epp91] D. Eppstein. Dynamic three-dimensional linear programming. In *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, pages 488–494, 1991.
- [Eri96] J. Erickson. New lower bounds for convex hull problems in odd dimensions. In *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, pages 1–9, 1996.
- [ES91] H. Edelsbrunner and W. Shi. An $O(n \log^2 h)$ time algorithm for the three-dimensional convex hull problem. *SIAM Journal on Computing*, 20:259–277, 1991.
- [FN00] D. I. Fotiadis and S. D. Nikolopoulos, editors. *Advances in Informatics*. World Scientific Publishing, University of Ioannina, Greece, 2000.
- [Gra67] R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1:132–133, 1967.

- [Grü67] B. Grünbaum. *Convex Polytopes*. John Wiley & Sons, London, 1967.
- [GY93] A. Gerasoulis and T. Yang. On the granularity and clustering of directed acyclic task graphs. *IEEE Transactions on Parallel and Distributed Systems*, 4(6):686–701, 1993.
- [Jar73] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2:18–21, 1973.
- [JKS89] H. Jung, L. Kirousis, and P. Spirakis. Lower bounds and efficient algorithms for multiprocessor scheduling of DAGs with communication delays. In *Proceedings of the 1st ACM SPAA 1989*, pages 254–264, 1989.
- [KPT96] N. Koziris, G. Papakonstantinou, and P. Tsanakas. Optimal time and efficient space free scheduling for nested loops. *The Computer Journal*, 39(5):439–448, 1996.
- [KS86] D. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM Journal on Computing*, 15:287–299, 1986.
- [Lam74] L. Lamport. The parallel execution of DO loops. *Communications of the ACM*, 37(2):83–93, February 1974.
- [LK90] P.-Z. Lee and Z. M. Kedem. Mapping nested loop algorithms into multidimensional systolic arrays. *IEEE Transactions on Parallel and Distributed Systems*, 1:64–76, January 1990.
- [Mat93] J. Matoušek. Linear optimization queries. *Journal of Algorithms*, 14:432–448, 1993.
- [MF86] D. Moldovan and J. Fortes. Partitioning and mapping algorithms into fixed size systolic arrays. *IEEE Transactions on Computers*, C-35(1):1–11, 1986.
- [Mol93] D. I. Moldovan. *Parallel Processing: From Applications to Systems*. Morgan Kaufmann Publishers, 1993.
- [MS71] P. McMullen and G. C. Shephard. *Convex Polytopes and the Upper Bound Conjecture*. Cambridge University Press, 1971.

- [Mul93] M. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, N.J., 1993.
- [O'R94] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.
- [PAD01] G. Papakonstantinou, T. Andronikos, and I. Drositis. On parallelization of UET/UET-UCT loops. *Neural, Parallel and Scientific Computations*, 9(3-4):279–318, September & December 2001.
- [PH77] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the Association of Computing Machinery*, 20:87–93, 1977.
- [Pic92] P. Picouleau. *Etude de Problèmes d'Optimization dans les Systèmes Distribués*. PhD thesis, Université Pierre et Marie Curie, 1992.
- [PS85] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.
- [PY90] C. Papadimitriou and M. Yannakakis. Toward an architecture-independent analysis of parallel algorithms. *SIAM Journal of computers, Extended Abstract in Proceedings STOC 1988*, 19:322–328, 1990.
- [RS87] V. J. Rayward-Smith. UET scheduling with unit interprocessor communication delays and unlimited number of processors. *Discrete Applied Mathematics*, 18:55–71, 1987.
- [Sei81] R. Seidel. A convex hull algorithm optimal for point sets in even dimensions. Technical Report 81-14, Department of Computer Science, University of British Columbia, Vancouver, B.C., 1981.
- [Sei86] R. Seidel. Constructing higher-dimensional convex hulls at logarithmic cost per face. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 404–413, 1986.
- [Sei91] R. Seidel. Small-dimensional linear programming and convex hull made easy. *Discrete & Computational Geometry*, 6:423–434, 1991.

- [SF91] W. Shang and J. Fortes. Time optimal linear schedules for algorithms with uniform dependencies. *IEEE Transactions on Computers*, 40(6):723–742, 1991.
- [SHC96] W. Shang, E. Hodzic, and Z. Chen. On uniformization of affine dependence algorithms. *IEEE Transactions on Parallel and Distributed Systems*, 45(7):827–840, July 1996.
- [ST91] J. Sheu and T. Tai. Partitioning and mapping nested loops on multiprocessors systems. *IEEE Transactions on Parallel and Distributed Systems*, 24:430–439, 1991.
- [Swa85] G. F. Swart. Finding the convex hull facet by facet. *Journal of Algorithms*, 6:17–48, 1985.
- [TKP00] P. Tsanakas, N. Koziris, and G. Papakonstantinou. Chain Grouping: A method for partitioning loops onto mesh-connected processor arrays. *IEEE Transactions on Parallel and Distributed Systems*, 11(9):941–955, September 2000.
- [TN93] T. H. Tzen and L. Ni. Dependence uniformization: A loop parallelization technique. *IEEE Transactions on Parallel and Distributed Systems*, 4(5):547–558, May 1993.
- [Ull75] J. Ullman. NP-complete scheduling problems. *Journal of Computer and System Sciences*, 10:384–393, 1975.
- [Zie95] Günter M. Ziegler. *Lectures on Polytopes*. Springer-Verlag, New York, 1995.