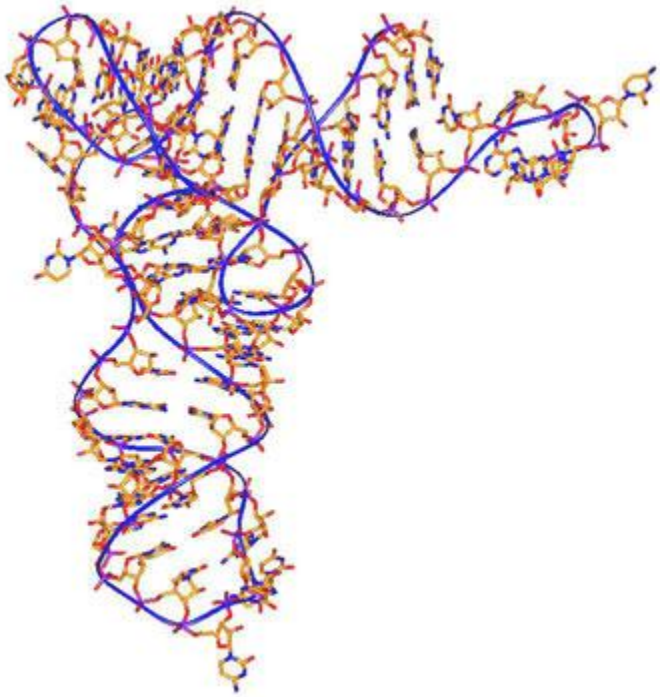
ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



Σχεδίαση και υλοποίηση σε αναδιατασσόμενη λογική του αλγορίθμου Zucker.

ΕΚΠΟΝΗΣΗ :

ΣΜΕΡΔΗΣ ΜΙΛΤΙΑΔΗΣ : Α. Μ. 2002030002

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ **3**

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ **6**

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ **8**

ΕΙΣΑΓΩΓΗ **10**

ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ **11**

RNA 11
Δομή *11*
Ρόλος *13*
ΔΕΥΤΕΡΟΤΑΓΗΣ ΔΟΜΗ ΤΟΥ RNA 14
Τι είναι η δευτεροταγής δομή *14*
Πού χρειάζεται *14*
Πώς υπολογίζεται *15*
Ενέργεια Gibbs 15
Ισχυροί και ασθενής δεσμοί 16
ΑΛΓΟΡΙΘΜΟΙ ΓΙΑ RNA FOLDING 16
Αλγόριθμοι που εστιάζουν στα ζεύγη των βάσεων *16*
Αλγόριθμοι που εστιάζουν στις περιοχές που σχηματίζουν τα ζεύγη των βάσεων *22*
Αλγόριθμοι που λαμβάνουν υπόψη τα pseudoknots *22*
Γιατί χρησιμοποιείται ο αλγόριθμος του Zucker *22*

ΠΕΡΙΓΡΑΦΗ ΑΛΓΟΡΙΘΜΟΥ **23**

ΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ 23
K-LOOP DECOMPOSITION 24
1-loop (Hairpin loop) *25*
2-loop *26*
Stack pair 26
Bulge 26
Interior loop 27
K-loop για K>2 (Multi-loop) *27*
Exterior loop (0-loop) *28*
Pseudoknots *28*
ΠΕΙΡΑΜΑΤΙΚΑ ΔΕΔΟΜΕΝΑ 29
Stacking Energies *30*
Terminal mismatch stacking energies (hairpin loops) *30*
Terminal mismatch stacking energies (interior loops) *31*
Single base stacking energies (dangle) *32*
Loop destabilizing energies *33*
Ειδικές περιπτώσεις για Interior Loop *34*
1×1 loop energies 34
1×2 loop energies 36
2×2 loop energies 37
Ειδικές περιπτώσεις για Hairpin Loop *37*
Tetra-loops 38
Tri-loops 38
CCC-Loop 39
GGG-Loop 39
Miscellaneous energies *40*
ΥΠΟΛΟΓΙΣΜΟΣ ΕΝΕΡΓΕΙΩΝ ΤΩΝ LOOPS 41
Ενέργεια για Hairpin loop *41*
Ενέργεια για Bulge *41*

Αριστερό Bulge ... 42
Δεξιό Bulge ... 42
Ενέργεια για Interior loop ... *42*
Ενέργεια για Multi loops ... *42*
ΑΝΑΔΡΟΜΙΚΕΣ ΣΧΕΣΕΙΣ ΓΙΑ ΤΟΝ ΥΠΟΛΟΓΙΣΜΟ ΤΩΝ ΥΠΟ-ΠΡΟΒΛΗΜΑΤΩΝ ... 43
Πίνακας V ... *44*
Πίνακας L ... *45*
Πίνακας W ... *46*
ΕΞΑΡΤΗΣΕΙΣ ΜΕΤΑΞΥ ΔΕΔΟΜΕΝΩΝ ... 47
Πίνακας V ... *47*
Πίνακας L ... *48*
Πίνακας W ... *49*
ΣΤΡΑΤΗΓΙΚΗ ΥΠΟΛΟΓΙΣΜΟΥ ΤΩΝ ΥΠΟ-ΠΡΟΒΛΗΜΑΤΩΝ ΓΙΑ ΤΗΝ ΕΞΑΛΕΙΨΗ ΤΩΝ ΕΞΑΡΤΗΣΕΩΝ ... 50
ΕΞΑΓΩΓΗ ΤΗΣ ΔΕΥΤΕΡΟΤΑΓΗΣ ΔΟΜΗΣ ΑΠΟ ΤΑ ΥΠΟ-ΠΡΟΒΛΗΜΑΤΑ ... 51
Αλγόριθμος οπισθοδρόμησης (Trace Back) ... *52*
Οπισθοδρόμηση στον πίνακα W ... 52
Οπισθοδρόμηση στον πίνακα L ... 53
Οπισθοδρόμηση στον πίνακα V ... 54
Οπισθοδρόμηση σε όλους τους πίνακες ... 55

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ... 56

ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ ... 56
Base ... *56*
Pair ... *56*
Energy ... *56*
LEnergy ... *56*
Indx ... *57*
Data ... *57*
Addr ... *57*
ctype ... *57*
ΓΕΝΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ... 58
Μείωση συνολικού MAC ... *61*
Μείωση συνολικού MAC – Πίνακας V ... 62
Μείωση συνολικού MAC – Πίνακας W ... 63
Μείωση συνολικού MAC – Πίνακας L ... 64
Βελτίωση κατανάλωσης τιμών ... *65*
ΕΠΙΠΕΔΑ ΥΠΟΛΟΓΙΣΜΟΥ ... 66
Repository για τα πειραματικά δεδομένα (EDRR) ... *67*
Επιλογή στοιχείων προς επίλυση ... 69
Αποθήκευση RNA ... 69
Αποθήκευση ενεργειών ... 69
Σχηματική αναπαράσταση EDRR ... 69
Calculators ... *70*
VCalculators ... 71
WCalculators ... 73
LCalculators ... 75
Calculator Controllers ... *76*
VCalculator Controller ... 78
WCalculator Controller ... 80
LCalculator Controller ... 82
Calculation Limiter ... *84*
Υπολογισμός ορίου για τον πίνακα V ... 85
Υπολογισμός ορίου για τον πίνακα W ... 86
Υπολογισμός ορίου για τον πίνακα L ... 86
Memory Controller ... *88*

Συνολικη σχεδιαση 90

ΥΛΟΠΟΙΗΣΗ 91

Αριθμητικη συστηματος (Problem Sizing) 91
Base *91*
Pair *91*
Energy *92*
LEnergy *92*
Indx *92*
Data *93*
Addr *94*
ctype *94*
Αποθηκευση υπο-προβληματων 95
Αναδιάταξη πινάκων *95*
Συστημα Αναφορας 97
Αξιοποίηση διαθέσιμου χώρου *97*
Κατανάλωση πόρων για διάφορα επίπεδα παραλληλησμού 98

ΑΠΟΔΟΣΗ ΣΥΣΤΗΜΑΤΟΣ 99

Αποδοση αλγοριθμου σε υλικο 99
Υπολογισμός speed up σε σχέση με την αύξηση των παράλληλων υπολογισμών. *100*
Αποδοση αλγοριθμου στο λογισμικο αναφορας 101
Σύγκριση με το υλικό για την περίπτωση χωρίς παραλληλισμό *102*
Σύγκριση με το υλικό για την περίπτωση με δύο παράλληλους υπολογισμούς *103*
Σύγκριση με το υλικό για την περίπτωση με τέσσερις παράλληλους υπολογισμούς *104*
Αποδοση αλγοριθμου απο το πακετο UNAFold 105
Σύγκριση με το υλικό για την περίπτωση χωρίς παραλληλισμό *106*
Σύγκριση με το υλικό για την περίπτωση με δύο παράλληλους υπολογισμούς *107*
Σύγκριση με το υλικό για την περίπτωση με τέσσερις παράλληλους υπολογισμούς *108*
Εκτιμηση χρονου εκτελεσης υλικου για μεγαλυτερα μεγεθη RNA. 109
Εκτίμηση απόδοσης σε υλικό. *111*
Υπολογισμός εκτιμώμενου speed up σε σχέση με την αύξηση των παράλληλων υπολογισμών. *112*
Σύγκριση εκτιμώμενων τιμών για την περίπτωση χωρίς παραλληλισμό με software *113*
Σύγκριση εκτιμώμενων τιμών για την περίπτωση δύο παράλληλων υπολογισμών με software *114*
Σύγκριση εκτιμώμενων τιμών για την περίπτωση τεσσάρων παράλληλων υπολογισμών με software *115*
Αποδοση αλγοριθμου για μεγαλυτερα επιπεδα παραλληλισμου 116
Αποδοση αλγοριθμου για το μεγιστο επιπεδο παραλληλισμου. 119
Βελτιστο επιπεδο παραλληλισμου για διαφορα μεγεθη RNA 120
Απαιτησεις βελτιστου επιπεδου παραλληλισμου 121
Επίδραση στο ρολόι *121*
Επίδραση στα BRams *121*
Βελτιστα επιπεδα παραλληλισμου για διαφορα boards 122
Εκτίμηση speed up σε σχέση με το λογισμικό *123*

ΣΥΜΠΕΡΑΣΜΑΤΑ - ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ 124

REFERENCES 125

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Figure 1 - Κεντρικο δογμα μοριακης βιολογιας (αρχικη εμφανιση ορου) 10
Figure 2 - Κεντρικο δογμα μοριακης βιολογιας (σημερα) 10
Figure 3 - Διαφορα RNA απο DNA 12
Figure 4 - RNA-11 14
Figure 5 - Nussinov - Αλγοριθμος οπισθοδρομησης 20
Figure 6 – Nussinov – Αποτελεσμα 21
Figure 7 - Παρουσιαση k-loops πανω στο RNA 24
Figure 8 - 1-loop : Hairpin loop 25
Figure 9 - 2-loop : Stack pair 26
Figure 10 - 2-loop : Bulge 26
Figure 11 - 2-loop Interior loop 27
Figure 12 - 3-loop : Multi-loop 27
Figure 13 - 0-loop : Exterior Loop 28
Figure 14 - Pseudoknot 28
Figure 15 - Stacking Energy 30
Figure 16 - Terminal Mismatch Stacking Energies (hairpin loop) 30
Figure 17 - Terminal Mismatch Stacking Energies (interior loop) 31
Figure 18 - Dange5 Energies 32
Figure 19 - Dange3 Energies 32
Figure 20 - Hairpin Loop Size Calculation 33
Figure 21 - Bulge Loop Size Calculation 33
Figure 22 - Interior Loop Size Calculation 34
Figure 23- 1x1 Interior Loop Energies 35
Figure 24 - 1x2 Interior Loop Energies 36
Figure 25 - 2x1 Interior Loop Energies 36
Figure 26 - 2x2 Interior Loop Energies 37
Figure 27 - CCC-Loop 39
Figure 28 - GGG-Loop 39
Figure 29 - Εξαρτησεις του στοιχειου V(i,j) 47
Figure 30 - Εξαρτησεις του στοιχειου L(i,j) 48
Figure 31 - Εξαρτησεις του στοιχειου W(i,j) 49
Figure 32 - Αλγοριθμος οπισθοδρομησης για τον πινακα W 52
Figure 33 - Αλγοριθμος οπισθοδρομησης για τον πινακα L 53
Figure 34 - Αλγοριθμος οπισθοδρομησης για τον πινακα V 54
Figure 35 - Αλγοριθμος οπισθοδρομησης για ολους τους πινακες 55
Figure 36 - Κοινα στοιχεια μεταξυ γειτονικων υπολογισμων 61
Figure 37 - Εξαρτησεις για παραλληλο υπολογισμο στον πινακα V 62
Figure 38 - Εξαρτησεις για παραλληλο υπολογισμο στον πινακα W 63
Figure 39 - Εξαρτησεις για παραλληλο υπολογισμο στον πινακα L 64
Figure 40 - Σχηματικη αναπαρασταση του EDRR 69
Figure 41 – Μεταβαση καταστασεων στα Calculators 70
Figure 42 - Σχηματικη αναπαρασταση του VCalculator 71
Figure 43 - Σχηματικη αναπαρασταση του WCalculator 73
Figure 44 - Σχηματικη αναπαρασταση του LCalculator 75
Figure 45 - Μεταβαση καταστασεων στους Controllers 76
Figure 46 - Στοιχεια για την αιτηση V(10,20,4) 77
Figure 47 - Σχηματικη αναπαρασταση του VCalculator Controller 78
Figure 48 - Αιτησεις VCalculatorController για στοιχεια απο τον πινακα V 79
Figure 49 - Αιτησεις VCalculatorController για στοιχεια απο τον πινακα W 79
Figure 50 - Σχηματικη αναπαρασταση του WCalculator Controller 80
Figure 51 - Αιτησεις VCalculatorController για στοιχεια απο ολους τους πινακες 81

FIGURE 52 - ΣΧΗΜΑΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΟΥ LCALCULATOR CONTROLLER 82
FIGURE 53 - ΑΙΤΗΣΕΙΣ VCALCULATORCONTROLLER ΓΙΑ ΣΤΟΙΧΕΙΑ ΑΠΟ ΤΟΝ ΠΙΝΑΚΑ L 83
FIGURE 54 - ΣΧΗΜΑΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΟΥ CALCULATION LIMITER 84
FIGURE 55 - ΣΧΗΜΑΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΟΥ MEMORY CONTROLLER 88
FIGURE 56 - ΣΧΗΜΑΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΣΥΝΟΛΙΚΗΣ ΣΧΕΔΙΑΣΗΣ 90
FIGURE 57 - ΣΧΗΜΑΤΙΚΗ ΑΝΑΠΑΡΑΣΤΑΣΗ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ PAIRING(B1,B2) 91
FIGURE 58 - ΘΕΣΕΙΣ ΣΤΟΙΧΕΙΩΝ ΣΤΟ ΑΝΑΔΙΑΤΕΤΑΓΜΕΝΟ ΔΙΑΝΥΣΜΑ 96
FIGURE 59 – XUP VIRTEX II PRO BOARD 98
FIGURE 60 - ΡΥΘΜΟΣ ΑΥΞΗΣΗΣ ΧΡΟΝΟΥ ΕΚΤΕΛΕΣΗΣ (HARDWARE) 99
FIGURE 61 - SPEED UP ΔΙΑΦΟΡΕΤΙΚΩΝ ΕΠΙΠΕΔΩΝ ΠΑΡΑΛΛΗΛΙΣΜΟΥ 100
FIGURE 62 - ΡΥΘΜΟΣ ΑΥΞΗΣΗΣ ΧΡΟΝΟΥ ΕΚΤΕΛΕΣΗΣ (SOFTWARE) 101
FIGURE 63 - SPEED UP ΥΛΙΚΟΥ (1) ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ 102
FIGURE 64 - SPEED UP ΥΛΙΚΟΥ (2) ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ 103
FIGURE 65 - SPEED UP ΥΛΙΚΟΥ (4) ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ 104
FIGURE 66 - ΡΥΘΜΟΣ ΑΥΞΗΣΗΣ ΧΡΟΝΟΥ ΕΚΤΕΛΕΣΗΣ (SOFTWARE) 105
FIGURE 67 - SPEED UP ΥΛΙΚΟΥ (1) ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ 106
FIGURE 68 - SPEED UP ΥΛΙΚΟΥ (2) ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ 107
FIGURE 69 - SPEED UP ΥΛΙΚΟΥ (4) ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ 108
FIGURE 70 - ΣΥΓΚΡΙΣΗ ΕΚΤΙΜΩΜΕΝΩΝ ΤΙΜΩΝ ΜΕ ΤΙΣ ΠΡΑΓΜΑΤΙΚΕΣ 110
FIGURE 71 – ΕΚΤΙΜΩΜΕΝΟΣ ΡΥΘΜΟΣ ΑΥΞΗΣΗΣ ΧΡΟΝΟΥ ΕΚΤΕΛΕΣΗΣ (HARDWARE) 111
FIGURE 72 - SPEED UP ΔΙΑΦΟΡΕΤΙΚΩΝ ΕΠΙΠΕΔΩΝ ΠΑΡΑΛΛΗΛΙΣΜΟΥ 112
FIGURE 73 – ΕΚΤΙΜΩΜΕΝΟ SPEED UP ΥΛΙΚΟΥ (1) ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ 113
FIGURE 74 - ΕΚΤΙΜΩΜΕΝΟ SPEED UP ΥΛΙΚΟΥ (2) ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ 114
FIGURE 75 - ΕΚΤΙΜΩΜΕΝΟ SPEED UP ΥΛΙΚΟΥ (4) ΣΕ ΣΧΕΣΗ ΜΕ ΤΟ ΛΟΓΙΣΜΙΚΟ 115

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Table 1 – Nussinov - Αρχικοποιηση Πινακα 17
Table 2 – Nussinov - Τελικος Πινακας 19
Table 3 – Nussinov - Αποτελεσματα συναρτησης a 19
Table 4 – Nussinov - Μονοπατι Οπισθοδρομησης 21
Table 5 - Αριθμος στοιχειων για τον υπολογισμο του V(i,j) 47
Table 6 - Αριθμος στοιχειων για τον υπολογισμο του L(i,j) 48
Table 7 - Αριθμος στοιχειων για τον υπολογισμο του W(i,j) 49
Table 8 - Συγκεντρωτικα στοιχεια για τον αριθμο των απαιτουμενων στοιχειων για τον υπολογισμο των W(i,j), V(i,j) και L(i,j) 50
Table 9 - MAC για διαφορα μεγεθη RNA 60
Table 10 - Μεσος αριθμος στοιχειων για παραλληλο υπολογισμο στον πινακα V 62
Table 11 - Μεσος αριθμος στοιχειων για παραλληλο υπολογισμο στον πινακα W 63
Table 12 - Διαχωρισμος αλγοριθμου σε υποσυστηματα 66
Table 13 - Διεπαφη εισοδου (EDRR) 67
Table 14 - Διεπαφη εξοδου (EDRR) 67
Table 15 - Τυποι υπολογισμου (EDRR) 68
Table 16 - Καταστασεις λειτουργιας Calculators 70
Table 17 - Εισοδοι για τα VCalculators 71
Table 18 - Εξοδοι για τα VCalculators 72
Table 19 - Εξαγωγη τυπου υπολογισμου για τα VCalculators 72
Table 20 - Εισοδοι για τα WCalculators 73
Table 21 - Εξοδοι για τα WCalculators 74
Table 22 - Εξαγωγη τυπου υπολογισμου για τα WCalculators 74
Table 23 - Εισοδοι για τα LCalculators 75
Table 24 - Εξοδοι για τα LCalculators 75
Table 25 - Καταστασεις λειτουργιας των Controllers 76
Table 26 - Εισοδοι του VCalculator Controller 78
Table 27 - Εξοδοι του VCalculator Controller 79
Table 28 - Εισοδοι του WCalculator Controller 80
Table 29 - Εξοδοι του WCalculator Controller 81
Table 30 - Εισοδοι του LCalculator Controller 82
Table 31 - Εξοδοι του LCalculator Controller 83
Table 32 - Εισοδοι του Calculation Limiter 84
Table 33 – Εξοδοι του Calculation Limiter 84
Table 34 - Διαμορφωση οριου υπολογισμου για τον πινακα V 85
Table 35 - Διαμορφωση οριου υπολογισμου για τον πινακα W 86
Table 36 - Διαμορφωση οριου υπολογισμου για τον πινακα L 86
Table 37 - Εισοδοι του Memory Controller 89
Table 38 - Έξοδοι του Memory Controller 89
Table 39 - Κωδικοποιηση δεδομενων τυπου Base 91
Table 40 - Κωδικοποιηση δεδομενων τυπου Pair 91
Table 41 - Κωδικοποιηση δεδομενων τυπου ctype 94
Table 42 - Αριθμος στοιχειων προς αποθηκευση σε σχεση με το μεγεθος του RNA 95
Table 43 - Αριθμος απαιτουμενων BRAM για καθε ενεργεια 97
Table 43 – Καταναλωση πορων στο Board Virtex-2 Pro XC2VP30 98
Table 44 - Χρονοι εκτελεσης υλικου 99
Table 45 - Συγκριση διαφορετικων επιπεδων παραλληλισμου 100
Table 46 - Χρονοι εκτελεσης λογισμικου 101
Table 47 - Συγκριση λογισμικου με υλικο (1) 102
Table 48 - Συγκριση λογισμικου με υλικο (2) 103
Table 49 - Συγκριση λογισμικου με υλικο (4) 104

Table 50 - Χρονοι εκτελεσης λογισμικου 105
Table 51 - Συγκριση λογισμικου με υλικο (1) 106
Table 52 - Συγκριση λογισμικου με υλικο (2) 107
Table 53 - Συγκριση λογισμικου με υλικο (4) 108
Table 54 - Χρονοι εκτελεσης υλικου 111
Table 55 - Συγκριση διαφορετικων επιπεδων παραλληλισμου 112
Table 57 - Συγκριση λογισμικου με εκτιμηση υλικου (1) 113
Table 57 - Συγκριση λογισμικου με εκτιμηση υλικου (2) 114
Table 58 - Συγκριση λογισμικου με εκτιμηση υλικου (4) 115
Table 59 - Αριθμος προσβασεων μνημης 116
Table 60 - Συγκριση θεωριτικων και πειραματικων δεδομενων για speed up. 116
Table 61 - Αριθμος προσβασεων μνημης για 1,2,4,8 και 16 παραλληλους υπολογισμους 117
Table 62 - Αριθμος προσβασεων μνημης για 32,64,128,256 και 512 παραλληλους υπολογισμους 117
Table 63 – Speed up για 2,4,8 και 16 παραλληλους υπολογισμους 118
Table 64 – Speed up για 32,64,128,256 και 512 παραλληλους υπολογισμους 118
Table 65 – Μεγιστο speed up για διαφορα μεγεθη RNA 119
Table 66 – Μεση ποσοστιαια χρηση πυρηνων για μεγιστο speed up 119
Table 67 – Speed up και αξιοποιηση πυρηνων για παραλληλισμο ισο με το βελτιστο speed up 120
Table 68 – Speed up και αξιοποιηση πυρηνων για παραλληλισμο ισο με το μισο του βελτιστο speed up 120
Table 69 - Απαιτουμενος αριθος Block RAMs για την υλοποιηση των πυρηνων 121
Table 69 – Διαθεσιμα Block Rams και μεγιστος αριθμος πυρηνων για διαφορα Boards 122
Table 69 – Speed up και μεση ποσοστιαια χρηση πυρηνων για διαφορα Boards 122
Table 69 – Speed up σε σχεση με το λογισμικο για διαφορα Boards 123

ΕΙΣΑΓΩΓΗ

Το Κεντρικό δόγμα της Μοριακής Βιολογίας αποτελεί μια αναπαράσταση των πιθανών τρόπων ροής της γενετικής πληροφορίας. Ο όρος προτάθηκε από τον Francis Crick το 1958 [1] ορίζοντας τρείς πιθανούς τρόπους ροής της γενετικής πληροφορίας

1. Αντιγραφή του DNA.
2. Μεταγραφή του DNA.

   Μεταφορά δηλαδή της γενετικής πληροφορίας από μορφή DNA σε μορφή αγγελιοφόρου RNA (mRNA).

3. Μετάφραση

   Έκφραση δηλαδή της πληροφορίας στη γλώσσα των αμινοξέων (πρωτεΐνες), με βάση το γενετικό κώδικα.

Οι οποίοι συνοψίζονται στο ακόλουθο σχήμα

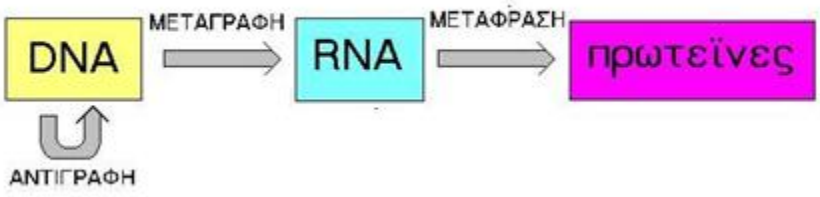


Figure 1 - Κεντρικό δόγμα μοριακής βιολογίας (αρχική εμφάνιση όρου)

Στις μέρες μας, οι ερευνητές της Βιολογίας γνωρίζουν ότι υπάρχουν επιπλέον τρόποι ροής της γενετικής πληροφορίας. Πιο συγκεκριμένα, έχουν εντοπιστεί ορισμένοι ιοί που έχουν ως γενετικό υλικό RNA και χρησιμοποιούν ένα ειδικό ένζυμο, την αντίστροφη μεταγραφάση, προκειμένου να χρησιμοποιήσουν αυτό το RNA και να συνθέσουν DNA. Επίσης, έχει διαπιστωθεί ότι σε κάποιους ιούς το RNA μπορεί να αυτό-διπλασιάζεται.

Έτσι ο όρος επανα-ορίστηκε το 1970 [2] σύμφωνα με τον οποίο, η γενετική πληροφορία ρέει από τα νουκλεϊκά οξέα (το DNA και RNA) προς τις πρωτεΐνες (Figure 2).

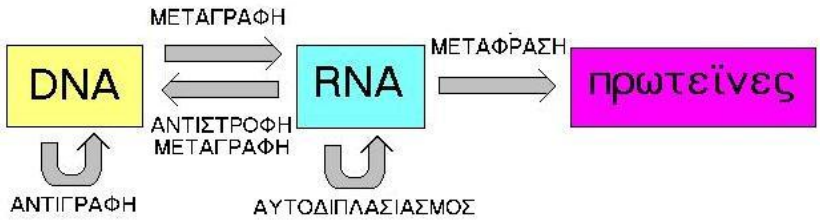


Figure 2 - Κεντρικό δόγμα μοριακής βιολογίας (σήμερα)

Όπως γίνεται φανερό και από το παραπάνω σχήμα το RNA έχει καταλυτικό ρόλο στην ροή της γενετικής πληροφορίας και στην διαδικασία της πρωτεϊνοσύνθεσης γενικότερα. Αποτελεί δομικό συστατικό των ριβοσωμάτων καθόσον επιτελεί σημαντικές μεταφορές αμινοξέων που προορίζονται για την Πρωτεϊνοσύνθεση. Το RNA μεταγράφεται από το DNA με τη βοήθεια κυρίως ενός ενζύμου που ονομάζεται RNA πολυμεράση και στη συνέχεια επεξεργάζεται με έναν αριθμό άλλων δευτερευόντων ενζύμων. Στη συνέχεια χρησιμοποιείται σαν την βάση για την μετάφραση των γονιδίων σε πρωτεΐνες, μεταφέροντας αμινοξέα στα ριβοσώματα για να δημιουργήθουν πρωτεΐνες [3]. Η κατανόηση του ρόλου και της λειτουργίας του RNA είναι ζωτικής σημασίας στην μελέτη της διαδικασίας αυτής.

Κύριο ρόλο στην λειτουργία του RNA έχει η δομή του στον χώρο, όπου σε συνδυασμό με το υψηλό κόστος της κρυσταλλογραφικής ανάλυσης με περίθλαση ακτινών X [4] και τους περιορισμούς μεγέθους για την Φασματοσκοπία NMR [5], δημιουργεί την ανάγκη του αλγοριθμικού υπολογισμού της δομής του RNA με βάση την χημική του ένωση. Το πρώτο βήμα σε αυτήν διαδικασία και το θέμα αυτής της εργασίας είναι ο υπολογισμός της δευτερεύουσας δομής.

Το μεγάλο μειονέκτημα της αλγοριθμικής προσέγγισης του προβλήματος είναι ο υψηλός υπολογιστικός φόρτος. Στα πλαίσια της εργασίας αυτής χρησιμοποιήσαμε την πλέον γνωστή και ευρέως αποδεχτή μέθοδο του Zucker [6], σε μια προσπάθεια για βελτίωση του χρόνου εκτέλεσης του αλγορίθμου με την αποτύπωσή του σε ένα σύστημα αναδιατασσόμενης λογικής.

ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ

RNA

Το Ριβονουκλεϊκό οξύ, ή ορθότερα ριβοζονουκλεϊκό οξύ, και συντομογραφικά RNA, είναι μία τις δύο κατηγορίες των πολυμερών νουκλεϊκών οξέων. Έχει σημαντικό ρόλο στη διαδικασία της μετάφρασης, από την έτερη κατηγορία, το δεοξυριβονουκλεϊκό οξύ (DNA), σε πρωτεϊνικά προϊόντα και χαρακτηρίζεται ως ο "αγγελιοφόρος" του DNA και των πρωτεϊνικών συμπλεγμάτων που είναι γνωστά σαν ριβοσώματα στο κυτταρόπλασμα του κυττάρου. Έτσι το RNA μαζί με το DNA αποτελούν το γενετικό υλικό των οργανισμών [3].

Δομή

Από χημικής άποψης το RNA είναι όμοιο με το DNA. Και οι δύο αυτές κατηγορίες νουκλεϊκών οξέων είναι μακρομοριακές ενώσεις μεγάλου μοριακού βάρους. Το μακρομόριο του RNA αποτελείται από επαναλαμβανόμενες δομικές μονάδες τα νουκλεοτίδια. Το μόριο του RNA περιλαμβάνει (όπως και του DNA), τέσσερις τύπους νουκλεοτιδίων που συνδέονται μεταξύ τους με 3'-5' φωσφοδιεστερικούς δεσμούς.

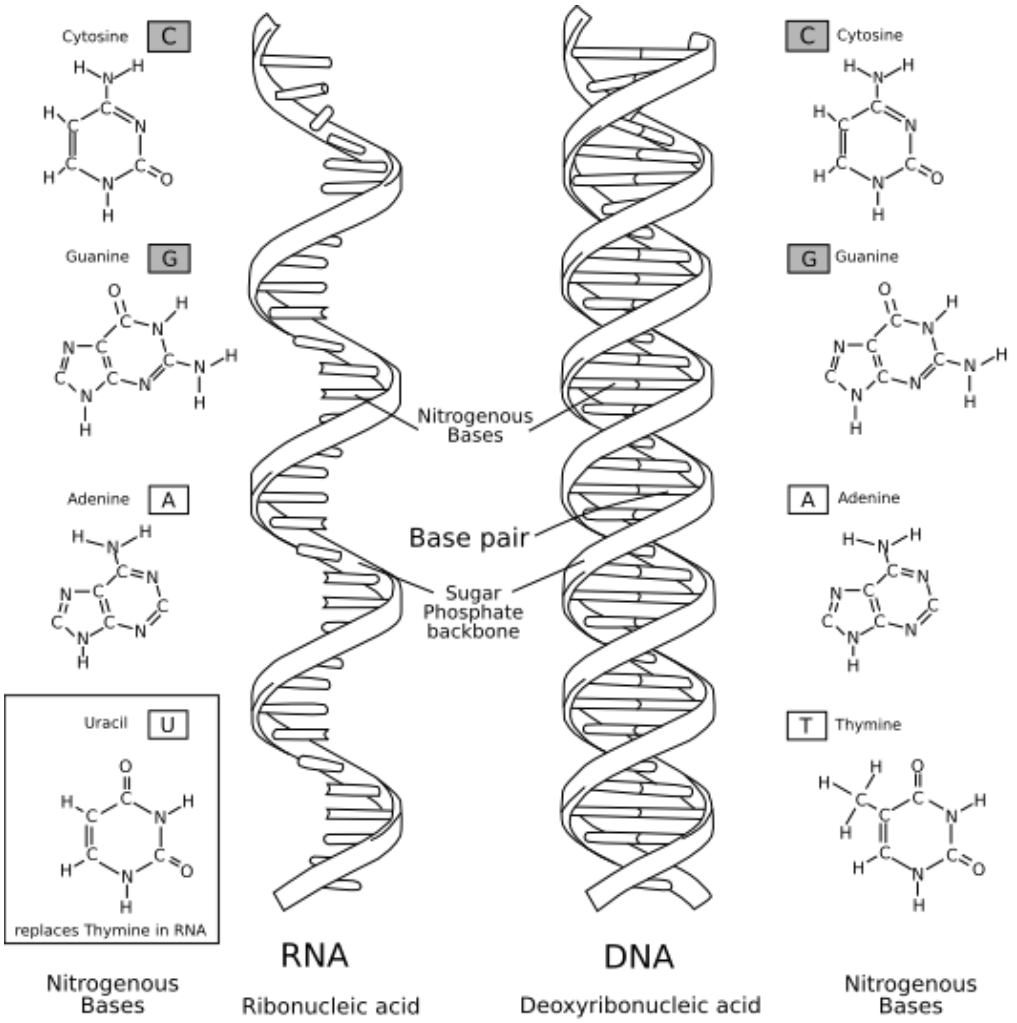


Figure 3 - Διαφορά RNA από DNA

Σε αντίθεση με το DNA, τα RNA μόρια είναι ουσιαστικά μονόκλωνα με την έννοια ότι δεν αποτελούνται από συμπληρωματικές αλυσίδες. Μέσα όμως στα μονόκλωνα μόρια παρουσιάζονται αναδιπλώσεις που οφείλονται σε δεσμούς υδρογόνου που αναπτύσσονται μεταξύ συμπληρωματικών βάσεων. Στις περιοχές αυτές (όπου το RNA γίνεται δίκλωνο) η δομή θυμίζει την A-μορφή του DNA και ονομάζεται A-RNA ή RNA-11 επειδή για μία πλήρης στροφή απαιτούνται 11 ζεύγη βάσεων, το βήμα της έλικας είναι 30A και τέλος, τα επίπεδα των ζευγών βάσεων εμφανίζουν κλίση περίπου 14°. Τα RNA μόρια είναι πολύ πιο ετερογενή από το DNA και η ετερογένεια στη δομή αντανακλά την ετερογένεια στη λειτουργικότητα όπου συναντάμε από απλά μεταφορικά μόρια μέχρι μόρια RNA με ενζυμική δράση.

Βασική επίσης διαφορά είναι ότι το σάκχαρο στα νουκλεοτίδια του είναι η ριβόζη, εξ ου και η ονομασία τους ριβονουκλεοτίδια, αντί της δεοξυριβόζης στο DNA, και ότι περιέχει την πυριμιδίνη ουρακίλη αντί της θυμίνης (που υπάρχει στο μόριο του DNA), χωρίς να είναι γνωστός ο λόγος της τελευταίας αυτής διαφοράς.

Ρόλος

Ο ρόλος του RNA ποικίλει ανάλογα με την δομή του και χωρίζεται στις παρακάτω κύριες κατηγορίες

1. Αγγελιοφόρο RNA ή mRNA
   Είναι το RNA που μεταφέρει τη γενετική πληροφορία από το DNA στα σημεία των ριβοσωμάτων για την πρωτεΐνοσύνθεση του κυττάρων. Στα ευκαρυωτικά κύτταρα όταν το mRNA μεταγράφεται από το DNA, στη συνέχεια επεξεργάζεται πριν εξαχθεί από τον πυρήνα στο κυτταρόπλασμα, όπου συνδέεται με τα ριβοσώματα και μεταφράζεται στην αντίστοιχη πρωτεΐνη με τη βοήθεια του μεταφορικού RNA (tRNA). Στα προκαρυωτικά κύτταρα, στα οποία δεν υπάρχει σαφής διάκριση μεταξύ πυρήνα και κυτταροπλάσματος το mRNA μπορεί να συνδεθεί με τα ριβοσώματα ενώ μεταγράφεται από το DNA. Μετά από κάποιο χρονικό διάστημα το γενετικό μήνυμα υποβαθμίζεται στα συστατικά του νουκλεοτίδια συνήθως με τη βοήθεια ειδικών ενζύμων.

2. Το Μεταφορικό RNA ή tRNA
   Είναι μία μικρή αλυσίδα RNA με περίπου 74-95 νουκλεοτιδίων που μεταφέρει ειδικά αμινοξέα σε μια επεκτεινόμενη πολυπεπτιδική αλυσίδα στα ριβοσώματα του κυττάρου όπου γίνεται η πρωτεΐνοσύνθεση κατά τη διάρκεια της μετάφρασης του κυττάρου. Το tRNA διαθέτει ειδικούς υποδοχείς για την πρόσδεση αμινοξέων και μια περιοχή αντικωδωνίου για την αναγνώριση του κωδωνίου που δεσμεύεται με την ειδική ακολουθία του αγγελιοφόρου RNA με ισχυρούς δεσμούς υδρογόνου. Είναι ένας τύπος μη κωδικοποιούμενου RNA.

3. Το Ριβοσωμικό RNA ή rRNA
   Είναι ένας τύπος RNA των ριβοσωμάτων που καταλύει την πρωτεΐνοσύνθεση στο κύτταρο. Τα ευκαρυωτικά ριβοσώματα αποτελούν το εργοστάσιο πρωτεΐνοσύνθεσης του κυττάρου και περιέχουν τέσσερα διαφορετικά μόρια rRNA: τα 18S, 5.8S, 28S, και 5S rRNA. Τρία από τα αυτά τα μόρια rRNA συντίθενται στον πυρηνίσκο του κυττάρου. Τα διαφορετικά είδη rRNA είναι πολυάριθμα στο κύτταρο και αποτελούν το 80% των ολικών RNA σε ένα τυπικό ευκαρυωτικό κύτταρο. Στο κυτταρόπλασμα το ριβοσωμικό RNA συνδυάζεται με ειδικές πρωτεΐνες του πλάσματος και συνθέτουν ένα νουκλεοπρωτεΐνικό σύμπλεγμα που ονομάζεται ριβόσωμα. Το ριβόσωμα συνδέεται με το mRNA και είναι υπεύθυνο για την πρωτεΐνική σύνθεση, όπου και λαμβάνει χώρα. Ορισμένα ριβοσώματα μπορούν να είναι συνδεδεμένα με μία μονή αλυσίδα mRNA σε όλη τη διάρκεια της ζωής τους.

4. Το καταλυτικό RNA
   Το καταλυτικό RNA ή ριβοζύμη είναι ένα είδος RNA που καταλύει μια χημική αντίδραση. Παρότι το RNA περιέχει μόνο τέσσερις νιτρογενείς βάσεις σε σχέση με τα 20 αμινοξέα που απαντώνται στις πρωτεΐνες, ορισμένα είναι ικανά να καταλύουν ειδικές χημικές αντιδράσεις που λαμβάνουν χώρα στο κύτταρο. Τέτοιες αντιδράσεις περιλαμβάνουν την διάσπαση και επανένωση άλλων μορίων RNA και επίσης η κατάλυση του πεπτιδικού δεσμού που λαμβάνει χώρα στα ριβοσώματα του κυτταροπλάσματος.

Δευτεροταγής δομή του RNA

Η δευτεροταγής δομή του RNA είναι το δεύτερο επίπεδο οργάνωσης του μακρομορίου του RNA και περιέχει πληροφορία για την αναδίπλωση του [7].

Τι είναι η δευτεροταγής δομή

Η αλληλουχία των βάσεων στο μακρομόριο του RNA δεν σχηματίζει μια ενεργειακά σταθερή χημική ένωση, σαν συνέπεια αυτού είναι η δημιουργία ομοιοπολικών δεσμών μεταξύ των νουκλεοτίδιων οι οποίες την σταθεροποιούν. Οι δεσμοί δημιουργούνται μεταξύ ακριβώς δύο νουκλεοτίδιων με το κάθε νουκλεοτίδιο να παίρνει μέρος το πολύ σε ένα ζεύγος. Το σύνολο των ζευγών αποτελεί την δευτεροταγή δομή του RNA.

Στο σημείο αυτό αξίζει να σημειωθεί ότι η δευτεροταγής δομής δεν είναι στατική αλλά "ταλαντώνεται" γύρω από ένα σημείο "ισορροπίας", καθώς ανά πάσα στιγμή ζεύγη μπορούν να δημιουργηθούν ή να διασπαστούν. Κατά την δημιουργία του RNA, είτε μέσω της μετάφρασης από το DNA είτε μέσω του αυτοδιπλασιασμού, το μακρομόριο δεν έχει αποκτήσει κάποια δομή στον χώρο. Αυτό γίνεται στην συνέχεια μέσω της δημιουργίας και διάσπασης ζευγών, όπου σταδιακά συγκλίνει σε κάποια σταθερή ένωση. Όμως η εξάρτηση του "σημείου σύγκλισης" από την θερμοκρασία και την περιεκτικότητα του διαλύματος στο οποίο περιέχεται σε κατιόντα νατρίου (αλάτι), μεγέθη από την φύση τους μεταβλητά, οδηγεί σε ανατροφοδότηση της διαδικασίας αυτής.

Πού χρειάζεται

Η δευτεροταγής δομή αποτελεί την πρώτη εκτίμηση της αναπαράστασης του μακρομορίου στον χώρο. Στο σημείο αυτό παρατηρούνται οι συμπληρωματικές αλυσίδες που δημιουργούνται στο μόριο του RNA (A-RNA ή RNA-11). Στο παρακάτω σχήμα παρουσιάζεται μια τέτοια περιοχή.

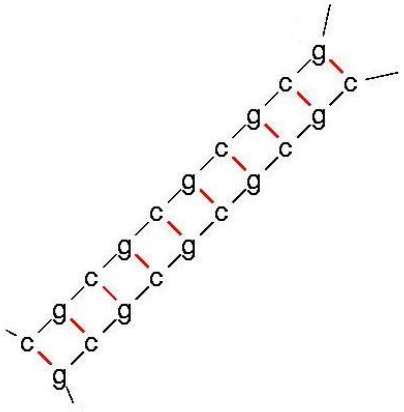


Figure 4 - RNA-11

Οι περιοχές αυτές είναι γενικά πολύ σταθερές και παραμένουν αμετάβλητες για μικρές διαφοροποιήσεις του περιβάλλοντος. Στην συνέχεια με τον υπολογισμό της τριτοταγής και τεταρτοταγής δομής μπορούμε να έχουμε μια ποιο αναλυτική αναπαράσταση.

Πώς υπολογίζεται

Εν γένει υπάρχουν δύο τρόποι υπολογισμού της δευτεροταγής δομής.

1. Με απευθείας παρατήρηση του μακρομορίου.
   Στην κατηγορία αυτή, η πληροφορία εξάγεται απευθείας από το ίδιο το μακρομόριο. Υπάρχουν δύο μέθοδοι

   a. Κρυσταλλογραφική ανάλυση με περίθλαση ακτινών Χ
      Η μέθοδος αυτή απαιτεί εξεζητημένα μηχανήματα πολύ υψηλού κόστους και εξειδικευμένο προσωπικό για την χρήση τους. Υπάρχουν ελάχιστα εργαστήρια τα οποία είναι σε θέση να εκτελέσουν τέτοιου είδους αναλύσεις και στις περισσότερες περιπτώσεις το κόστος είναι απαγορευτικό.

   b. Φασματοσκοπία NMR
      Η μέθοδος αυτή είναι σημαντικά ποιο φτηνή από την κρυσταλλογραφική ανάλυση, όμως η τωρινή τεχνολογία επιτρέπει την ανάλυση περιορισμένου μεγέθους μακρομορίων, με μέγιστο μερικές εκατοντάδες νουκλεοτίδια, το οποίο την καθιστά μη αποτελεσματική.

2. Αλγοριθμικός υπολογισμός
   Στην δεύτερη κατηγορία αντιστοιχούν οι αλγοριθμικές μέθοδοι του υπολογισμού της δευτερεύουσας δομής. Το κύριο πρόβλημα σε αυτήν την κατηγορία αλγορίθμων είναι το υψηλό κόστος σε χρόνο καθώς και μια σημαντική απόκλιση από την πραγματική δομή.

   Οι αλγόριθμοι αυτοί εκμεταλλεύονται το γεγονός ότι η δευτεροταγής δομή συγκλίνει σε μία σταθερή ένωση και προσπαθούν να βρούν την δομή εκείνη η οποία ελαχιστοποιεί την ενέργεια Gibbs.

Για μικρές αλλαγές του περιβάλλοντα χώρου (θερμοκρασία, κατιόντα νατρίου), οι αλλαγές στην δευτερεύουσα δομή είναι εξίσου μικρές και έτσι η λειτουργία του μένει ανεπηρέαστη, αντίθετα για μεγάλες αλλαγές θα επηρεαστεί και η λειτουργία του μακρομορίου και για την μελέτη της απαιτείται να επαναϋπολογιστεί η δευτερεύουσα δομή λαμβάνοντας υπόψη τις αλλαγές αυτές.

Ενέργεια Gibbs

Η ενέργεια Gibbs ή αλλιώς free energy εκφράζει την δύναμη η οποία είναι ικανή να προκαλέσει χημικές αντιδράσεις. Αντιστοιχεί στο έργο το οποίο μπορεί να εξαχθεί από μία αντιστρεπτή ενέργεια σε ένα κλειστό θερμοδυναμικό σύστημα (ισόθερμο και ισοβαρής). Είναι προφανές ότι η ελαχιστοποίηση την ενέργειας αυτής οδηγεί σε σταθεροποίηση της χημικής ένωσης [8].

Ισχυροί και ασθενής δεσμοί

Οι δεσμοί μεταξύ των νουκλεοτιδίων δεν είναι ισοδύναμοι. Η πρώτη κατηγορία ζευγών είναι τα ζεύγη Watson-Crick τα οποία σχηματίζουν ισχυρούς δεσμούς. Αποτελούν την πλειοψηφία των δεσμών στο μακρομόριο του RNA και συνεισφέρουν θετικά στην μείωση της ενέργειας Gibbs και στην σταθεροποίηση της ένωσης. Τα ζεύγη αυτά είναι τα παρακάτω

- A-U (Αδεδίνη - Ουρακίλη)
- C-G (Κυτοσίνη - Γουανίνη)

Η δεύτερη κατηγορία είναι τα ζεύγη Wobble τα οποία σχηματίζουν ασθενής δεσμούς. Τα ζεύγη αυτά είναι λιγότερα σε πλήθος και στις περισσότερες περιπτώσεις συνεισφέρουν θετικά στην μείωση της ενέργειας Gibbs, ενώ σε ειδικές περιπτώσεις μπορεί και να αποσταθεροποιούν το μακρομόριο. Ακόμα και στην δεύτερη περίπτωση ζεύγη αυτού του είδους μπορεί να υπάρχουν, αν οδηγούν στην συνολική μείωση της ελεύθερης ενέργειας. Τα ζεύγη αυτά είναι τα παρακάτω

1. G-U (Γουανίνη - Ουρακίλη)

Τέλος οι υπόλοιποι συνδυασμοί έχουν αρνητική επίδραση στην σταθεροποίηση του RNA, αυξάνοντας την ενέργεια Gibbs και δημιουργούνται μόνο σε ειδικές περιπτώσεις.

Αλγόριθμοι για RNA folding

Η αλγοριθμική πρόβλεψη της δευτεροταγής δομής του RNA είναι μια ανοιχτή και αρκετά ενεργή ερευνητική περιοχή. Έχουν ήδη προταθεί δεκάδες αλγόριθμοι ενώ νέοι αλγόριθμοι δημιουργούνται με σκοπό την βελτίωση τόσο της ποιότητας των αποτελεσμάτων όσο και της μείωσης του υπολογιστικού φόρτου.

Η πλειονότητα των αλγορίθμων αποτελεί τροποποιήσεις και βελτιστοποιήσεις ήδη υπάρχων αλγορίθμων. Στην πράξη ανήκουν σε τρείς κλάσεις αλγορίθμων κάθε μία με διαφορετικά χαρακτηριστικά.

1. Αλγορίθμους που εστιάζουν στα ζεύγη των βάσεων
2. Αλγορίθμους που εστιάζουν στις περιοχές που σχηματίζουν τα ζεύγη των βάσεων
3. Αλγόριθμοι που λαμβάνουν υπόψη τα pseudoknots

Οι τρείς κλάσεις των αλγορίθμων αντιστοιχούν σε μια σταδιακή εξέλιξη τόσο της πολυπλοκότητας και του υπολογιστικού φόρτου όσο και την ποιότητας των αποτελεσμάτων.

Αλγόριθμοι που εστιάζουν στα ζεύγη των βάσεων

Η πρώτη κατηγορία αλγορίθμων και η πρώτη που έκανε την εμφάνιση της για τον υπολογισμό της δευτεροταγής δομής αποτελείται από αλγορίθμους που εστιάζουν στα ζεύγη βάσεων. Κύριος αντιπρόσωπος αυτής της κλάσης αλγορίθμων είναι ο αλγόριθμος Nussinov [9], ο οποίος προτάθηκε το 1978 από την Dr Ruth Nussinov, καθώς αποτελεί την πρώτη και απλοϊκή προσπάθεια για αλγοριθμική προσέγγιση του υπολογισμού της δευτεροταγής δομής του RNA.

Οι αλγόριθμοι αυτής της κατηγορίας είναι αλγόριθμοι δυναμικού προγραμματισμού. Βασική ιδέα είναι ο υπολογισμός της βέλτιστης δομής για όλες της υπό-ακολουθίες και στην συνέχεια ο συνδυασμός τους για την δημιουργία της τελικής λύσης. Από την φύση του ο αλγόριθμος αγνοεί την ύπαρξη των pseudoknots, καθώς αντιμετωπίζει κάθε υπακολουθία σαν ένα ανεξάρτητο πρόβλημα.

Για την αποθήκευση των υπό-προβλημάτων γίνεται χρήση ενός άνω τριγωνικού πίνακα, έστω E, με μέγεθος ίσο με τον αριθμό των νουκλεοτιδίων που απαρτίζουν μακρομόριο του RNA. Κάθε στοιχείο του πίνακα δεν περιέχει την ακριβή δευτεροταγής δομή αλλά ένα "score" για την σύγκριση μεταξύ της ποιότητας των λύσεων. Η ίδια η λύση θα δημιουργηθεί μετά από μελέτη των αποτελεσμάτων του πίνακα.

Η εξαγωγή της λύσης γίνεται σε τρία βήματα

Βήμα πρώτο – Αρχικοποίηση πίνακα

Η αρχικοποίηση του πίνακα περιλαμβάνει τον ορισμό κάποιας τιμής στην κύρια διαγώνιο και σε έναν αριθμό από δευτερεύουσες διαγώνιους, συνήθως τρεις. Ο λόγος που γίνεται αυτό έχει να κάνει με της φυσικές ιδιότητες του RNA καθώς νουκλεοτίδια με απόσταση μικρότερη ή ίση του τέσσερα δεν είναι δυνατόν να δημιουργήσουν ζεύγος. Η τιμή αυτή είναι συνήθως παραμετροποιήσιμη επιτρέποντας τον ορισμό της ελάχιστης απόστασης μεταξύ δύο βάσεων οι οποίες μπορούν να δημιουργήσουν ζεύγος.

Στην συνέχεια φαίνεται ο πίνακας μετά την αρχικοποίηση. Ο πίνακας έχει αρχικοποιηθεί με την τιμή 0, ενώ με – συμβολίζονται τα τετράγωνα τα οποία δεν χρησιμοποιούνται. Τέλος τα κενά τετράγωνα αντιπροσωπεύουν θέσεις η οποίες δεν έχουν υπολογιστεί ακόμα.

0	0	0	0															
-	0	0	0	0														
-	-	0	0	0	0													
-	-	-	0	0	0	0												
-	-	-	-	0	0	0	0											
-	-	-	-	-	0	0	0	0										
-	-	-	-	-	-	0	0	0	0									
-	-	-	-	-	-	-	0	0	0	0								
-	-	-	-	-	-	-	-	0	0	0	0							
-	-	-	-	-	-	-	-	-	0	0	0	0						
-	-	-	-	-	-	-	-	-	-	0	0	0	0					
-	-	-	-	-	-	-	-	-	-	-	0	0	0	0				
-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0			
-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0		
-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0	
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0

Table 1 – Nussinov - Αρχικοποιήση Πίνακα

Βήμα δεύτερο – Συμπλήρωση των κενών τετραγώνων.
Στην συνέχεια ο αλγόριθμος γεμίζει αναδρομικά τις κενές θέσεις του πίνακα με βάση τις παρακάτω αναδρομικές σχέσεις.

$$E(i,j) = \begin{cases} E(i+1, j-1) + a(i,j) & (1) \\ E(i+1, j) & (2) \\ E(i, j-1) & (3) \\ E(i,k) + E(k+1, j), \forall k \in (i, j-1) & (4) \end{cases}$$

Σε προβλήματα μεγιστοποίησης επιλέγεται η περίπτωση με την μεγαλύτερη τιμή ενώ σε προβλήματα ελαχιστοποίησης η περίπτωση με την μικρότερη τιμή. Σε περίπτωση ισοτιμίας έχουμε δύο ισοδύναμες δευτεροταγείς δομές και μπορούμε να επιλέξουμε όποια θέλουμε.

Από φυσικής άποψης οι παραπάνω περιπτώσεις έχουν ως εξής

1) Προσθήκη του ζεύγους (i,j) στην υπό-ακολουθία $(i+1, j-1)$
   Η συνάρτηση $a(\cdot,\cdot)$ ορίζει ένα βάρος στο ζεύγος (i,j) και καθορίζει το κατά πόσο η δημιουργία του ζεύγους είναι δυνατή.
2) Προσθήκη της βάσης (i) χωρίς ζεύγος στην υπό-ακολουθία $(i+1, j)$
3) Προσθήκη της βάσης (j) χωρίς ζεύγος στην υπό-ακολουθία $(i, j-1)$
4) Συνδυασμός δύο βέλτιστων υπό-ακολουθιών (i,k) και $(k+1, j)$

Η διαφοροποίηση των αλγορίθμων αυτής της κλάσεις έχει να κάνει με τον τρόπο που υπολογίζεται η συνάρτηση $a(\cdot,\cdot)$.

Ο αλγόριθμος Nussinov προσπαθεί να μεγιστοποιήσει τον αριθμό των ζευγών και χρησιμοποιεί την παρακάτω σχέση.

$$a(i,j) = \begin{cases} 1 \text{ αν τα } i \text{ και } j \text{ μπορούν να σχηματίσουν ζεύγος} \\ 0 \text{ αλλιώς} \end{cases}$$

Ενώ παραλλαγές του αλγορίθμου περιλαμβάνουν "βαθμολόγηση" των ζευγαριών με βάση την συνεισφορά στην ελαχιστοποίηση της ενέργειας Gibbs.

$$a(i,j) = \begin{cases} -3 \text{ για ζεύγη Watson} - \text{Crick} \\ -1 \text{ για ζεύγη Wobble} \\ 0 \text{ αλλιώς} \end{cases}$$

Στην συνέχεια φαίνεται ο συμπληρωμένος πίνακας με βάση τον αλγόριθμο του Nussinov και για την ακολουθία r(5*'AGGGAAAUCCC3')*

	A	*G*	*G*	*G*	*A*	*A*	*A*	*A*	*U*	*C*	*C*	*C*
A	0	0	0	0	**0**	**0**	**0**	**0**	**1**	**2**	**3**	**4**
G	-	0	0	0	0	**0**	**0**	**0**	**1**	**2**	**3**	**4**
G	-	-	0	0	0	0	**0**	**0**	**1**	**2**	**3**	**3**
G	-	-	-	0	0	0	0	**0**	**1**	**2**	**2**	**2**
A	-	-	-	-	0	0	0	0	**1**	**1**	**1**	**1**
A	-	-	-	-	-	0	0	0	0	**1**	**1**	**1**
A	-	-	-	-	-	-	0	0	0	0	**1**	**1**
A	-	-	-	-	-	-	-	0	0	0	0	**1**
U	-	-	-	-	-	-	-	-	0	0	0	0
C	-	-	-	-	-	-	-	-	-	0	0	0
C	-	-	-	-	-	-	-	-	-	-	0	0
C	-	-	-	-	-	-	-	-	-	-	-	0

Table 2 – Nussinov - Τελικός Πίνακας

καθώς και τα αποτελέσματα της συνάρτησης $a(i,j)$ για κάθε περίπτωση

	A	*G*	*G*	*G*	*A*	*A*	*A*	*A*	*U*	*C*	*C*	*C*
A	0	0	0	0	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
G	-	0	0	0	0	**0**	**0**	**0**	**0**	**1**	**1**	**1**
G	-	-	0	0	0	0	**0**	**0**	**0**	**1**	**1**	**1**
G	-	-	-	0	0	0	0	**0**	**0**	**1**	**1**	**1**
A	-	-	-	-	0	0	0	0	**1**	**0**	**0**	**0**
A	-	-	-	-	-	0	0	0	0	**0**	**0**	**0**
A	-	-	-	-	-	-	0	0	0	0	**0**	**0**
A	-	-	-	-	-	-	-	0	0	0	0	**0**
U	-	-	-	-	-	-	-	-	0	0	0	0
C	-	-	-	-	-	-	-	-	-	0	0	0
C	-	-	-	-	-	-	-	-	-	-	0	0
C	-	-	-	-	-	-	-	-	-	-	-	0

Table 3 – Nussinov - Αποτελέσματα συνάρτησης a

Βήμα τρίτο – Εξαγωγή της λύσης.

Για την εξαγωγή της λύσης χρησιμοποιείται ένας αλγόριθμός οπισθοδρόμησης ο οποίος ξεκινώντας από την θέση $(0, L-1)$, όπου L ο αριθμός των νουκλεοτιδίων που σχηματίζουν το RNA, ακολουθεί το αντίστροφο μονοπάτι από εκείνο με το οποίο ακολουθήθηκε στο βήμα δύο και ανιχνεύει τα ζεύγη. Συγκεκριμένα ο αλγόριθμος είναι ο εξής.

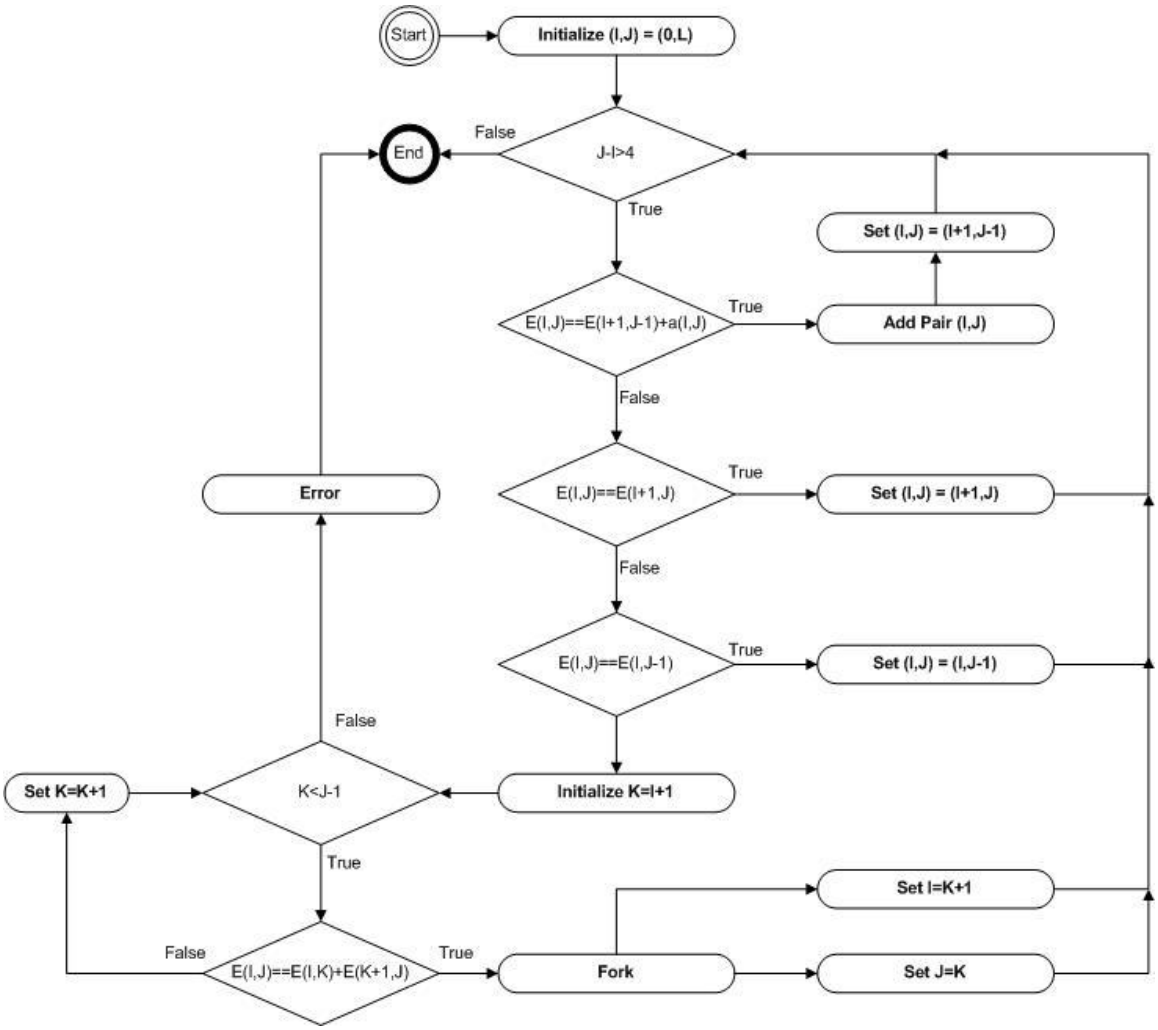


Figure 5 - Nussinov - Αλγόριθμος οπισθοδρόμησης

Ο αλγόριθμος οπισθοδρόμησης εκτελεί ακριβώς την αντίστροφη λειτουργία από εκείνη που χρησιμοποιήθηκε για να γεμίσει τον πίνακα. Σε κάθε στιγμή ελέγχει όλες τις διαφορετικές περιπτώσεις έτσι ώστε να βρει με ποια από τις τέσσερις αναδρομικές σχέσεις χρησιμοποιήθηκε για την δημιουργία της δεδομένης λύσης. Όταν βρεθεί ο πρώτος κανόνας τότε γνωρίζουμε ότι το συγκεκριμένο ζεύγος θα ανήκει και στην δευτεροταγή δομή του συνολικού RNA.

Στην συνέχεια φαίνεται η τελική λύση καθώς και το μονοπάτι οπισθοδρόμησης

	A	*G*	*G*	*G*	*A*	*A*	*A*	*A*	*U*	*C*	*C*	*C*
A	0	0	0	0	**0**	**0**	**0**	**0**	**1**	**2**	**3**	*4*
G	-	0	0	0	0	**0**	**0**	**0**	**1**	**2**	**3**	*4*
G	-	-	0	0	0	0	**0**	**0**	**1**	**2**	*3*	**3**
G	-	-	-	0	0	0	0	**0**	**1**	*2*	**2**	**2**
A	-	-	-	-	0	0	0	0	*1*	**1**	**1**	**1**
A	-	-	-	-	-	0	0	0	0	**1**	**1**	**1**
A	-	-	-	-	-	-	0	0	0	0	**1**	**1**
A	-	-	-	-	-	-	-	0	0	0	0	**1**
U	-	-	-	-	-	-	-	-	0	0	0	0
C	-	-	-	-	-	-	-	-	-	0	0	0
C	-	-	-	-	-	-	-	-	-	-	0	0
C	-	-	-	-	-	-	-	-	-	-	-	0

Table 4 – Nussinov - Μονοπάτι Οπισθοδρόμησης

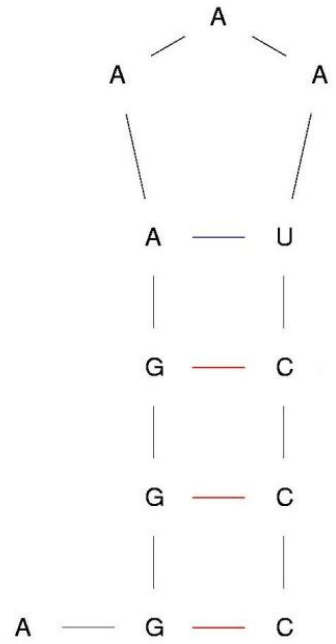


Figure 6 – Nussinov – Αποτέλεσμα

Αλγόριθμοι που εστιάζουν στις περιοχές που σχηματίζουν τα ζεύγη των βάσεων

Οι αλγόριθμοι αυτής της κατηγορίας αποτελούν την φυσική εξέλιξη των αλγορίθμων της προηγούμενης κατηγορίας σε μια προσπάθεια για βελτίωση της ποιότητας των αποτελεσμάτων. Η βασική δομή του αλγορίθμου έχει παραμείνει σταθερή, πρόκειται για αλγόριθμους δυναμικού προγραμματισμού με τα ίδια τρία στάδια. Η μεγάλη διαφοροποίηση έγκειται στο γεγονός ότι βέλτιστη λύση δεν επιδιώκεται μελετώντας τα ίδια τα ζεύγη αλλά τις υπό-περιοχές που σχηματίζουν τα ζεύγη, τα λεγόμενα loop.

Ο κύριος λόγος για αυτήν την διαφοροποίηση είναι ότι η μείωση στην ενέργεια Gibbs που προκαλεί κάθε ένα από τα loop είναι εύκολα μετρήσιμη μέσω πειραματικών δεδομένων. Η διαφορά μπορεί να φαίνεται μικρή όμως η μέχρι τώρα διαισθητική εκτίμηση της βέλτιστης δομής (όσο δυνατόν περισσότερα ζεύγη) η οποία εκφράζεται από τους προγενέστερους αλγορίθμους, αντικαταστάθηκε με ένα ακριβές μαθηματικό μοντέλο για τον υπολογισμό της, επιτυγχάνοντας πολύ ποιο ακριβής λύσεις, με τίμημα την αύξηση της πολυπλοκότητας και του υπολογιστικού φόρτου των αλγορίθμων.

Πάραυτα η λύσεις που προσφέρουν οι αλγόριθμοι αυτοί εξακολουθούν να αποτελούν εκτιμήσεις και όχι την πραγματική λύση. Κύριος λόγος είναι ότι τα pseudoknots δεν λαμβάνονται υπόψη, επιτρέποντας έτσι την ανάλυση των υπό-ακολουθιών του RNA ως ανεξάρτητα υπό-προβλήματα, μια αναγκαία θυσία για να μπορέσουν οι αλγόριθμοι να αναλυθούν μέσω ενός αλγορίθμου δυναμικού προγραμματισμού. Ένας δεύτερος σημαντικός λόγος είναι η απόκλιση των πειραματικών δεδομένων που χρησιμοποιούν οι αλγόριθμοι από τις πραγματικές τιμές, ειδικά στις περιπτώσεις loop υψηλής τάξης

Κύριος αντιπρόσωπος αυτής της κατηγορίας αλγορίθμων είναι η μέθοδος του Zucker [6]. Ο αλγόριθμος αυτός αποτελεί κύριο αντικείμενο της εργασίας αυτής και θα αναλυθεί διεξοδικά στο επόμενο κεφάλαιο.

Αλγόριθμοι που λαμβάνουν υπόψη τα pseudoknots

Στην κατηγορία αυτή ανήκουν αλγόριθμοι οι οποίοι προσπαθούν με κάποιο τρόπο να υπολογίσουν την ακριβή δευτεροταγής δομή συμπεριλαμβάνοντας στον υπολογισμό τα pseudoknots. Είναι μια αρκετά ανομοιογενής κλάση αλγορίθμων η οποία περιλαμβάνει λύσεις από θεωρία γράφων, γραμματικές χωρίς συμφραζόμενα και γενετικούς αλγορίθμους.

Η εισαγωγή των pseudoknots στον υπολογισμό της δευτεροταγής δομής εισάγει ένα τεράστιο υπολογιστικό κόστος με αποτέλεσμα ο αλγόριθμοι αυτοί να απαιτούν ένα μη ρεαλιστικό χρόνο εκτέλεσης.

Γιατί χρησιμοποιείται ο αλγόριθμος του Zucker

Οι τρείς κατηγορίες αλγορίθμων αποτελούν μια σταδιακή εξέλιξη των αλγορίθμων προς όσο δυνατών καλύτερες και ακριβέστερες λύσεις, με παράλληλη αύξηση της πολυπλοκότητας τους καθώς και την ανάγκης για πόρους, κυρίως για κύρια μνήμη. Ο αλγόριθμος του Zucker βρίσκεται στο μέσο αυτής της "ψαλίδας" προσφέροντας όσο το δυνατό καλύτερες λύσεις μέσα σε λογικά περιθώρια χρόνου, καθιστώντας τον την προτιμώμενη λύση από ερευνητές της μοριακής βιολογίας.

ΠΕΡΙΓΡΑΦΗ ΑΛΓΟΡΙΘΜΟΥ

Ορισμός προβλήματος

Ορισμός του RNA

Στα πλαίσια της εργασίας αυτής, μακρομόριο του RNA θα θεωρηθεί ως μια ακολουθία χαρακτήρων

$$x = (x_0 x_1 x_2 \dots x_{L-2} x_{L-1})$$

Όπου

$x_i \in \{A, C, G, U\}, \forall i$ και L το μέγεθος του RNA.

Επιπλέον η αναγραφή μιας ακολουθίας RNA θα γίνεται με βάση την παρακάτω μορφή

r(5*'AGGGAAAUCCC3')*

Όπου ο χαρακτήρας 'r' συμβολίζει ότι η ακολουθία αντιστοιχεί σε μακρομόριο RNA (έναντι του 'd' που χρησιμοποιείται για το DNA) ενώ οι χαρακτήρες *5'* και *3'* συμβολίζουν την αρχή και το τέλος της ακολουθίας αντίστοιχα.

Ορισμός δευτεροταγής δομής του RNA

Η δευτεροταγής δομή του RNA είναι ένα σύνολο P από διατεταγμένα ζεύγη της μορφής (i, j) τα οποία ικανοποιούν τους παρακάτω περιορισμούς

1. $0 \leq i < j < L - 1$
   Ο κανόνας αυτός περιορίζει τα ζεύγη έτσι ώστε να αντιστοιχούν σε εφικτούς συνδυασμούς τιμών και παράλληλα αποτρέπει την διπλή καταγραφή ενός ζεύγους τοποθετώντας πάντα το στοιχείο με τον μικρότερο δείκτη "πρώτο" στο ζεύγος.

2. $j - i > 3$
   Ο δεύτερος κανόνας πηγάζει από τους φυσικούς περιορισμούς του RNA. Η αλυσίδα των βάσεων είναι αρκετά δύσκαμπτη και έτσι δεν είναι δυνατών να σχηματιστούν ζεύγη με απόσταση μικρότερη από τέσσερα.

   (Υποσημείωση : αν και η τιμή τέσσερα είναι γενικά αποδεκτή, ο αλγόριθμός επιτρέπει την τροποποίηση της ελάχιστης απόστασης για την δημιουργία ενός ζεύγους)

3. $\{i, j\} \cap \{k, l\} = \emptyset, \forall (i, j), (k, l) \in P \cap (i, j) \neq (k, l)$
   Ο τελευταίος κανόνας χρησιμοποιείται για να αποτρέψει την δημιουργία δύο ζευγών με κάποια κοινή βάση. Κάτι τέτοιο δεν είναι δυνατόν να συμβεί.

Επιπλέον για κάθε ζεύγος (i, j) θα θεωρούμε ως 5' βάση την βάση i και ως 3' βάση την βάση j.

Ορισμός εμφωλευμένης δευτεροταγής δομής του RNA

Μια εμφωλευμένη δευτεροταγή δομή για το RNA έχει επιπλέον την ιδιότητα ότι για οποιαδήποτε δύο διαφορετικά ζεύγη $\forall(i,j),(k,l) \in P \cap (i,j) \neq (k,l)$, χωρίς βλάβη της γενικότητας έστω $i < k$ ισχύει ένας από τους δύο περιορισμούς

1. $i < j < k < l$
   Όπου το ζεύγος (i,j) προηγείται του (k,l).

2. $i < k < l < j$
   Όπου το ζεύγος (i,j) περιέχει το (k,l).

Μια εμφωλευμένη δευτεροταγής δομή αποτρέπει την δημιουργία pseudoknots και επιτρέπει τον χωρισμό της σε k-loop.

K-loop decomposition

Οι περιοχές που σχηματίζονται μεταξύ των ζευγών σε μια εμφωλευμένη δευτεροταγή δομή του RNA, αναλύονται και μελετώνται με βάση την θεωρία των k-loop [10].

Στην παρακάτω εικόνα οι περιοχές αυτές παρουσιάζονται αριθμημένες με την σειρά που εμφανίζονται απ την αρχή της ακολουθίας (5 ') προς το τέλος της (3 ').

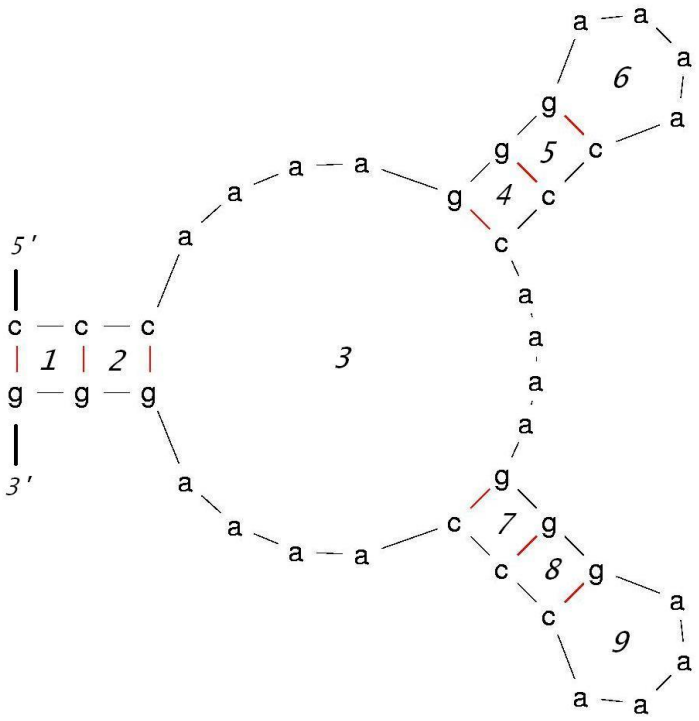


Figure 7 - Παρουσίαση k-loops πάνω στο RNA

Ορισμός k-loop

- Αν τα στοιχεία (i, j) αποτελούν ζεύγος, τότε το στοιχείο k θα είναι προσβάσιμο από το (i, j) αν $i < k < j$ και δεν υπάρχει ζεύγος (i', j') έτσι ώστε $i < i' < k < j' < j$

- Ένα ζεύγος (i', j') είναι προσβάσιμο από κάποιο άλλο, έστω (i, j) αν και οι δύο βάσεις που το αποτελούν είναι προσβάσιμες από αυτό.

- Το σύνολο των k-1 ζευγών και k' βάσεων που είναι προσβάσιμα από ένα ζεύγος (i, j) ονομάζονται k-loop τερματιζόμενο από το (i, j)

- Ο αριθμός των βάσεων που περιέχονται σε ένα k-loop αποτελούν το μέγεθος του.

- Ο αριθμός των ζευγών που περιέχονται σε ένα k-loop συν ένα αποτελούν την τάξη του.

Έτσι και σύμφωνα με το προηγούμενο σχήμα τα loops 1-2-4-5-7-8 είναι τάξης δύο με μέγεθος μηδέν, τα 6 και 9 τάξης ένα και μέγεθος τέσσερα, ενώ το loop 3 είναι τάξης τρία με μέγεθος 12.

Στην συνέχεια παρουσιάζονται οι βασικές δομές των k-loops.

1-loop (Hairpin loop)

Η πρώτη κατηγορία k-loops αποτελείται από τα επονομαζόμενα hairpin loops. Είναι τα loops με τάξη ένα, της μικρότερης δυνατής τάξης για κάποιο loop. Το μέγεθός τους είναι τουλάχιστον τρία, με βάση την ελάχιστη απόσταση μεταξύ δύο βάσεων οι οποίες μπορούν να σχηματίσουν ζεύγος, ενώ συνήθως είναι μικρά σε μέγεθος.

Στην παρακάτω εικόνα παρουσιάζεται ένα Hairpin loop με μέγεθος πέντε.

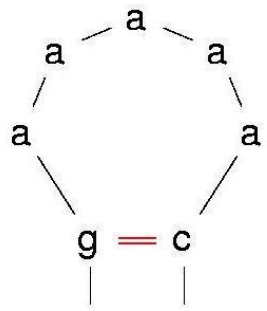


Figure 8 - 1-loop : Hairpin loop

Τα Loops αυτά έχουν αρνητική επίδραση στην σταθεροποίηση του μακρομορίου αφού η συνεισφορά τους στην συνολική ενέργεια Gibbs είναι θετική. Παρόλα αυτά συναντούνται σε μεγάλο αριθμό σε οποιαδήποτε δευτεροταγή δομή αφού δεν είναι δυνατό να δημιουργηθούν ζεύγη χωρίς την ύπαρξη τους.

2-loop

Τα k-loops δεύτερης τάξης χωρίζονται σε τρείς κατηγορίες ανάλογα με το μέγεθος τους ή ακριβέστερα με την κατανομή των βάσεων στο loop.

Stack pair

Η ποιο βασική δομή είναι το Stack Pair, το 2-loop με μέγεθος μηδέν. Όπως γίνεται φανερό και από την παρακάτω εικόνα, αποτελείτε από δύο ζεύγη των οποίων τόσο οι 5' βάσεις όσο και οι 3' βάσεις είναι διαδοχικές στο μακρομόριο του RNA.

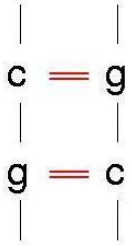


Figure 9 - 2-loop : Stack pair

Τα loop αυτά συνεισφέρουν θετικά στην σταθεροποίηση του μακρομορίου. Η συμβολή τους στην σταθεροποίηση του μακρομορίου είναι καταλυτική, αφού αποτελούν την ποιο σταθερή δομή που μπορεί να βρεθεί στην δευτεροταγή δομή. Συνήθως σχηματίζουν μεγάλες αλυσίδες από συνεχόμενα stack pair, μια δομή παρόμοια με το μακρομόριο του DNA.

Bulge

Στην συνέχεια έχουμε τα Bulge. Αποτελούνται από loop τάξης δύο με μέγεθος μεγαλύτερο το ενός. Βασική προϋπόθεση για τον χαρακτηρισμό ενός 2-loop ως Bulge είναι είτε οι 5' βάσεις είτε οι 3' βάσεις των ζευγών να είναι διαδοχικές στο μακρομόριο του RNA. Σε περίπτωση που οι 5' βάσεις είναι διαδοχικές έχουμε ένα Δεξί Bulge, ενώ αν οι 3' βάσεις είναι διαδοχικές έχουμε Αριστερό Bulge.

Στην παρακάτω εικόνα παρουσιάζεται ένα Bulge μεγέθους ένα. Δεν μπορούμε να ξέρουμε αν πρόκειται για ένα δεξί ή αριστερό Bulge γιατί δεν γνωρίζουμε ποια είναι η αρχή και ποιο το τέλος του μακρομορίου.

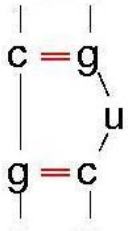


Figure 10 - 2-loop : Bulge

Τα Bulges κατά κύριο λόγο συνεισφέρουν θετικά στην σταθεροποίηση του μακρομορίου. Ώμος όσο αυξάνεται το μέγεθός τους τόσο αυξάνεται η συνεισφορά στην συνολική ενέργεια Gibbs με αποτέλεσμα Bulges μεγάλου μεγέθους να αποσταθεροποιούν το μακρομόριο.

Interior loop

Τέλος έχουμε τα Interior loops τα οποία είναι k-loops τάξης δύο με μέγεθος μεγαλύτερο του δύο. Αντίστοιχα με τα Bulges για να θεωρηθεί ένα 2-loop ως Interior loop θα πρέπει τοσο οι 5’ βάσεις όσο και οι 3’ βάσεις των ζευγών να μην είναι διαδοχικές στο μακρομόριο του RNA.

Στην παρακάτω εικόνα παρουσιάζεται ένα συμμετρικό Interior loop μεγέθους 8. Ένα Interior loop καλείτε συμμετρικό όταν η απόσταση μεταξύ των 5’ βάσεων είναι ίση με την απόσταση των 3’ βάσεων.

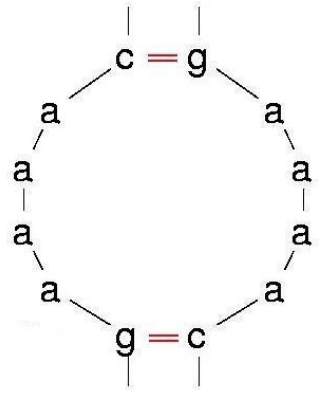


Figure 11 - 2-loop Interior loop

Σε γενικές γραμμές τα Interior loops συνεισφέρουν αρνητικά στην σταθεροποίηση του μακρομορίου. Ειδικές περιπτώσεις ενδέχεται να μειώσουν την συνολική ενέργεια Gibbs αλλά κατά κύριο λόγο τα Interior loops είναι λίγα σε αριθμό και η ύπαρξή τους συνήθως οδηγεί στην δημιουργία μιας μεγάλης αλυσίδας από Stacked Pairs.

K-loop για K>2 (Multi-loop)

Όλα τα υπόλοιπα loops (τάξης μεγαλύτερης του δύο) κατηγοριοποιούνται ως Multi-loops. Στην παρακάτω εικόνα παρουσιάζεται ένα Multi-loop τάξης τρία με μέγεθος δεκαπέντε.

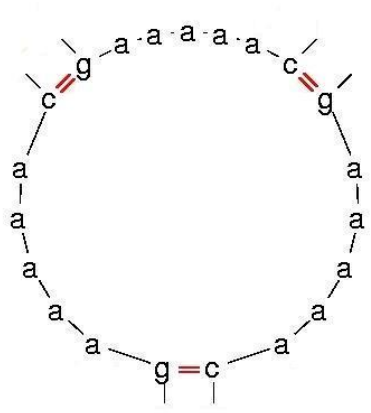


Figure 12 - 3-loop : Multi-loop

Exterior loop (0-loop)

Το Exterior loop ή αλλιώς 0-loop στην πραγματικότητα δεν αποτελεί ένα k-loop. Χρησιμοποιείται για να οριστεί η ενέργεια των ελευθέρων βάσεων και ζευγών δηλαδή των βάσεων και ζευγών που δεν είναι προσβάσιμα από κανένα άλλο ζεύγος. Τα στοιχεία αυτά απατώνται συνήθως στην αρχή και στο τέλος της ακολουθίας του RNA, χωρίς όμως αυτό να είναι δεσμευτικό.

Στην παρακάτω εικόνα παρουσιάζεται ένα Exterior loop το οποίο περιέχει ένα ζεύγος και πέντε βάσεις.

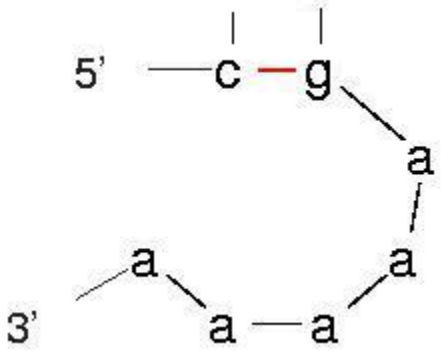


Figure 13 - 0-loop : Exterior Loop

Pseudoknots

Τα Pseudoknots αποτελούν δομές οι οποίες δεν μπορούν να περιγραφούν από την θεωρία των k-loops.

Στην παρακάτω εικόνα παρουσιάζεται ένα Pseudoknot.

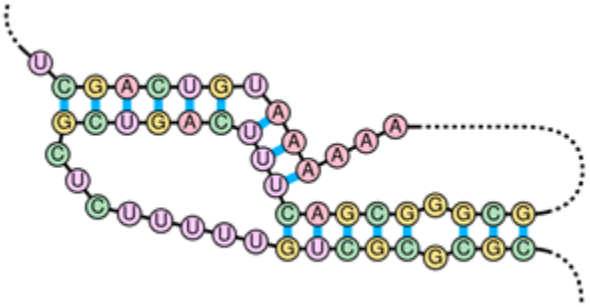


Figure 14 - Pseudoknot

Πειραματικά δεδομένα

Ο αλγόριθμος του Zucker υπολογίζει την συνεισφορά για κάθε K-loop με βάση ενός συνόλου πειραματικών δεδομένων [11] [12]. Τα δεδομένα αυτά αποτελούν μετρήσεις της συνεισφοράς στην συνολική ενέργεια Gibbs διάφορων σχηματισμών, πάνω σε πραγματικά μακρομόρια RNA.

Συγκεκριμένα χρησιμοποιούνται τα ακόλουθα πειραματικά δεδομένα.

- **Stacking energies**
- **Terminal mismatch stacking energies για τα Hairpin Loops**
- **Terminal mismatch stacking energies για τα Bulges**
- **Terminal mismatch stacking energies για τα Interior Loops**
- **Single base stacking energies**
- **Loop destabilizing energies**
- **Tetraloops**
- **Triloops**
- **1x1 Interior loops**
- **1x2 Interior loops**
- **2x2 Interior loops**

Οι τιμές αυτές δεν αντιστοιχούν στα ίδια τα k-loop καθώς το πλήθος των πιθανών συνδυασμών είναι απαγορευτικά μεγάλο. Αντ. αυτού η ενέργεια υπολογίζεται από το άθροισμα ενός συνόλου επιμέρους ενεργειών οι οποίες εξαρτώνται από την τάξη και το μέγεθος του κάθε loop.

Τα πειραματικά δεδομένα αναφέρονται σε βασικούς σχηματισμούς όπως στις ενέργειες των ζευγών που κλείνουν και περιέχονται σε κάποιο loop, τον αριθμό και την κατανομή των βάσεων που περιέχονται σε αυτά και ένα μεγάλο σύνολο ειδικών περιπτώσεων.

Τα δεδομένα είναι οργανωμένα σε μορφή πολυδιάστατων πινάκων, έτσι ώστε να γίνεται ευκολότερη η πρόσβαση στις τιμές αυτές. Κάθε διάσταση του πίνακα αντιστοιχεί στην τιμή κάποιας βάσης ή ζεύγους που ανήκει στον σχηματισμό που αναφέρεται η εκάστοτε ενέργεια, έτσι αν ο σχηματισμός περιέχει τέσσερις βάσεις ο αντίστοιχος πίνακας θα είναι τεσσάρων διαστάσεων.

Stacking Energies

Ο πίνακας αυτός είναι διαστάσεων $4 \times 4 \times 4 \times 4$ και περιέχει τις ενέργειες που αντιστοιχούν στα stacked pairs (2-loop με μέγεθος μηδέν). Στον πίνακα λαμβάνονται υπόψη όλοι οι δυνατοί συνδυασμοί, ακόμα και οι μη επιτρεπτοί, έτσι για όλα τα μη έγκυρα loops η τιμή της ενέργειας Gibbs λαμβάνει "άπειρη" τιμή.

Η τιμή αυτή εξαρτάται και από τα δύο ζεύγη που απαρτίζουν το loop.

Στην συνέχεια η αναφορά στην ενέργεια αυτή θα γίνεται μέσω της συνάρτησης $es(\cdot,\cdot,\cdot,\cdot)$. Η δε σειρά τον ορισμάτων θα είναι ίδια με την σειρά των βάσεων στο μακρομόριο του RNA έτσι για το παρακάτω σχήμα η αναφορά στην ενέργεια του θα είναι $es(i,k,l,j)$ ή αντίστοιχα $es(g,c,g,c)$.

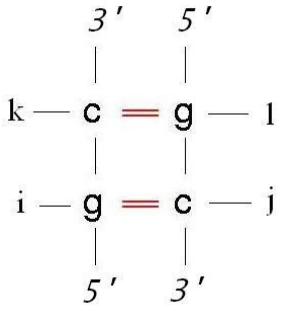


Figure 15 - Stacking Energy

Terminal mismatch stacking energies (hairpin loops)

Ο πίνακας αυτός είναι διαστάσεων $4 \times 4 \times 4 \times 4$ και περιέχει την ενέργεια που λαμβάνει το ζεύγος που τερματίζει ένα hairpin loop. Η ενέργεια αυτή υπολογίζεται τόσο με βάση τις βάσεις που δημιουργούν το ζεύγος όσο και με τις προσκείμενες αυτών βάσεις. Έτσι αν το loop τερματίζεται με το ζεύγος (i,j) οι βάσεις $i, i+1, j-1, j$ θα χρησιμοποιηθούν για την επιλογή της κατάλληλης ενέργειας από τον πίνακα αυτό.

Στην συνέχεια για την αναφορά σε αυτήν την ενέργεια, θα χρησιμοποιούμε την συνάρτηση $hltm(\cdot,\cdot,\cdot,\cdot)$

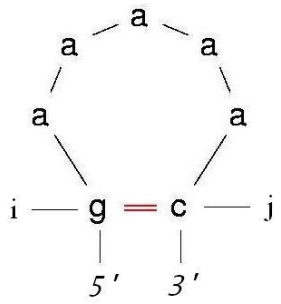


Figure 16 - Terminal Mismatch Stacking Energies (hairpin loop)

Αντίστοιχα με την περίπτωση των Stacking Energies η αναφορά στην ενέργεια αυτή για το παραπάνω σχήμα είναι $hltm(i,i+1,j-1,j)$ ή αντίστοιχα $hltm(g,a,a,c)$.

Terminal mismatch stacking energies (interior loops)

Ο πίνακας αυτός είναι διαστάσεων $4 \times 4 \times 4 \times 4$ και περιέχει την ενέργεια που λαμβάνουν τα ζεύγη που σχηματίζουν ένα interior loop. Η ενέργεια αυτή υπολογίζεται τόσο με βάση τις βάσεις που δημιουργούν το ζεύγος όσο και με τις προσκείμενες αυτών βάσεις. Έτσι για την επιλογή της κατάλληλης ενέργειας αν το loop τερματίζεται με το ζεύγος (i, j) θα χρησιμοποιηθούν οι βάσεις $i, i+1, j-1, j$, ενώ αν το ζεύγος (k, l) περιέχεται στο loop θα χρησιμοποιηθούν οι βάσεις $k-1, k, l, l+1$.

Στην συνέχεια για την αναφορά σε αυτήν την ενέργεια, θα χρησιμοποιούμε την συνάρτηση $iltm(\cdot,\cdot,\cdot,\cdot)$

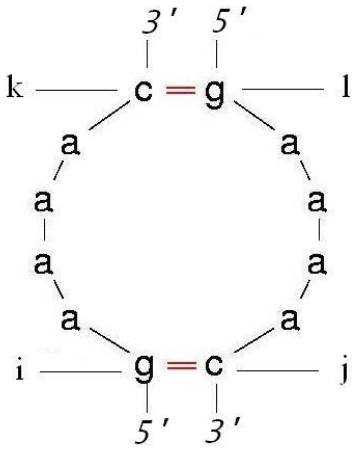


Figure 17 - Terminal Mismatch Stacking Energies (interior loop)

Μια μεγάλη διαφοροποίηση της ενέργειας αυτής από την υπόλοιπες είναι ότι χρησιμοποιείται για τον υπολογισμό της ενέργειας και των δύο ζευγών που δημιουργούν το Interior Loop. Όσον αφορά το ζεύγος που κλείνει το Loop, (i, j), ισχύει η ίδια σύμβαση με τις υπόλοιπες ενέργειες και θα έχουμε $iltm(i, i+1, j-1, j)$ ή αντίστοιχα $iltm(g, a, a, c)$.

Για το δεύτερο ζεύγος όμως η αντιστοιχία αυτή δεν ισχύει, αφού κάτι τέτοιο θα αντέστρεφε τους ρόλους των βάσεων (οι βάσεις που συμμετείχαν στο ζεύγος θα έπαιρναν την θέση εκείνων που δεν συμμετείχαν και αντίστροφα). Έτσι η αντιστοιχία για το ζεύγος (k, l) είναι η εξής $iltm(l, l+1, k-1, k)$ ή αντίστοιχα $iltm(g, a, a, c)$.

Single base stacking energies (dangle)

Σε αυτήν την κατηγορία περιέχονται δύο πίνακες διαστάσεων $4 \times 4 \times 4$ και περιέχουν την ενέργεια που λαμβάνει μια βάση προσκείμενη σε ένα ζεύγος. Οι δύο πίνακες αναφέρονται στις βάσεις που βρίσκονται πριν και μετά από κάποια βάση που απαρτίζει το ζεύγος αντίστοιχα. Για τον υπολογισμό την ενέργειας αυτής λαμβάνονται υπόψη οι βάσεις που απαρτίζουν το ζεύγος και η ελεύθερη βάση.

Στην συνέχεια η αναφορά στις ενέργειες αυτές θα γίνεται μέσω την συναρτήσεων $d5(\cdot,\cdot,\cdot)$ και $d3(\cdot,\cdot,\cdot)$ αντίστοιχα.

Συνολικά υπάρχουν τέσσερις διαφορετικές περιπτώσεις για Single Base Stacking καθώς η ελεύθερη βάση μπορεί να βρίσκεται είτε πριν είτε μετά από οποιαδήποτε βάση που συμμετέχει στο ζεύγος. Στην πρώτη περίπτωση έχουμε τα Single Base Stacking τα οποία προηγούνται κάποιας βάσης του ζεύγους.

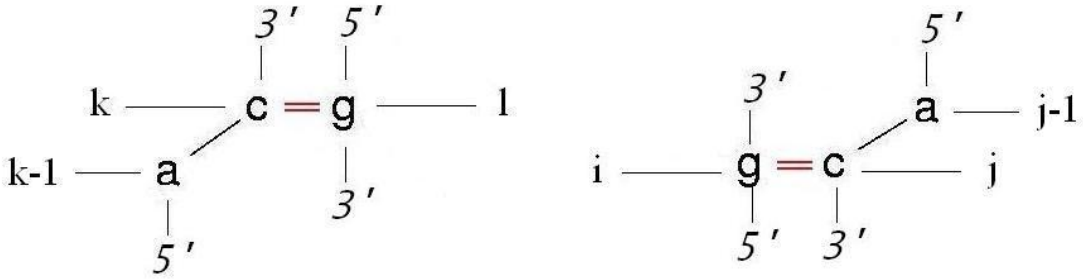


Figure 18 - Dange5 Energies

Κατά σύμβαση κατά την αναφορά στην ενέργεια αυτή θα παρέχουμε πρώτα τις βάσεις που ανήκουν στο ζεύγος και στην συνέχεια την ελεύθερη βάση. Έτσι στην πρώτη περίπτωση θα έχουμε $d5(k,l,k-1)$ ή αντίστοιχα $d5(c,g,a)$ ενώ στην δεύτερη περίπτωση έχουμε $d5(j,i,j-1)$ ή αντίστοιχα $d5(c,g,a)$. Παρατηρούμε πως αντίστοιχα με την περίπτωση στο ανεστραμμένο ζεύγος στα Interior loops οι βάσεις που συμμετέχουν στο ζεύγος είναι αντίστροφα τοποθετημένες.

Στην συνέχεια ακολουθούν τα Single Base Stacking τα οποία έπονται κάποιας βάσης του ζεύγους.

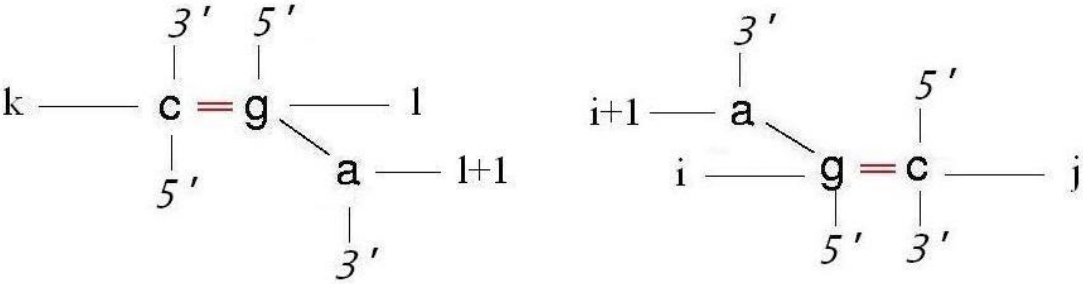


Figure 19 - Dange3 Energies

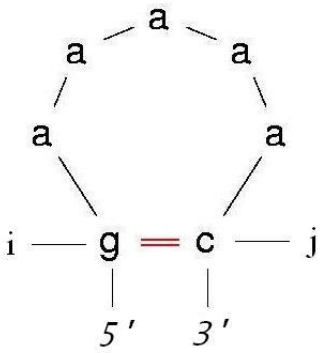
Αντίστοιχα για την αναφορά στην ενέργεια αυτή έχουμε $d3(k,l,l+1)$ ή αντίστοιχα $d5(c,g,a)$ για την πρώτη περίπτωση και $d5(j,i,i+1)$ ή αντίστοιχα $d5(c,g,a)$ για την δεύτερη.

Loop destabilizing energies

Στην συνέχεια έχουμε τρεις πίνακες διαστάσεων 1×30 οι οποίοι περιέχουν τον παράγοντα αποσταθεροποίησης των hairpin loops, bulge loops, interior loops για μεγέθη από ένα μέχρι και τριάντα. Ο παράγοντας για μεγαλύτερα μεγέθη υπολογίζεται προσεγγιστικά με την προσθήκη του όρου $1.75 \times RT \times ln(l_s/30)$, όπου l_s το μέγεθος του loop, στην τιμή που αντιστοιχεί για μέγεθος 30.

Η αναφορά στις ενέργειες αυτές θα γίνεται μέσω των συναρτήσεων $hlde(\cdot)$, $blde(\cdot)$ και $ilde(\cdot)$ για τα Hairpin loop, bulge και interior loop αντίστοιχα, με βάση το μέγεθος του loop.

Ο υπολογισμός του μεγέθους του κάθε Loop έχει ως εξής

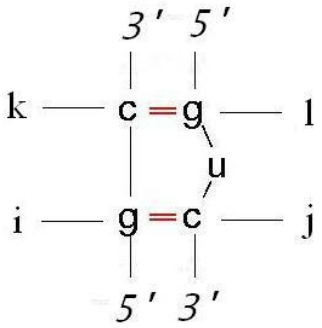


Το μέγεθος ενός Hairpin loop υπολογίζεται από την διαφορά (απόσταση) των δύο βάσεων που σχηματίζουν το ζεύγος που τερματίζει το loop.

$$j - i - 1$$

Από την απόσταση των βάσεων αφαιρείται ένα για να πάρουμε τον αριθμό των βάσεων που μεσολαβούν μεταξύ τους.

Figure 20 - Hairpin Loop Size Calculation

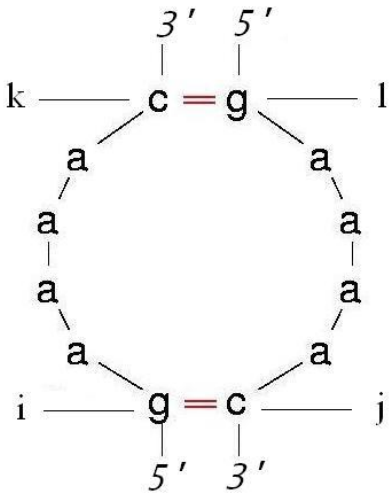


Όσον αφορά τα Bulge το μέγεθος προκύπτει είτε από την απόσταση μεταξύ των βάσεων l και j είτε μεταξύ των βάσεων i και k για τα δεξιά και αριστερά Bulge αντίστοιχα.

$$\begin{cases} j - l - 1\,, \text{για} \quad \text{δεξιά} \quad \text{Bulge} \\ k - i - 1, \text{για αριστερά Bulge} \end{cases}$$

Επιπλέον για την αποφυγή του ελέγχου για δεξί ή αριστερό Bulge μπορούμε να προσθέσουμε τα δύο μεγέθη καθώς ένα από τα δύο θα είναι μηδέν.

Figure 21 - Bulge Loop Size Calculation



Τέλος για την περίπτωση των Interior loops έχουμε

$$j - l + k - i - 2$$

Το οποίο αποτελεί το άθροισμα των βάσεων ο οποίες μεσολαβούν μεταξύ των βάσεων l και j και των βάσεων i και k.
Επιπλέον για Interior loop ορίζεται ένα επιπλέον μέγεθος, ο βαθμός ασυμμετρίας, ως

$$|j + i - l - k|$$

Το οποίο ορίζει την διαφορά μεταξύ των βάσεων από την αριστερή και την δεξιά πλευρά του loop.

Figure 22 - Interior Loop Size Calculation

Στην συνέχεια θα αναφερόμαστε σε Interior loops με βαθμό ασυμμετρίας μηδέν ως συμμετρικά.

Ειδικές περιπτώσεις για Interior Loop

Στην συνέχεια ακολουθούν τρείς πίνακες οι οποίοι περιέχουν τις ενέργειες για τα Interior loop μεγέθους μικρότερου ή ίσου με τέσσερα και μέγιστο βαθμό ασυμμετρίας ένα.

Τα συγκεκριμένα loops σχηματίζουν πολύ ισχυρούς δεσμούς επιτρέποντας τον σχηματισμό ζευγών μεταξύ των "ελεύθερων" βάσεων που περιέχονται στο Loop. Οι ενέργειες αυτές αποτελούν την συνολική ενέργεια του loop.

1×1 loop energies

Η πρώτη κατηγορία είναι τα 1×1 Interior loops. Πρόκειται για συμμετρικά Interior loops μεγέθους δύο. Τα πειραματικά δεδομένα αποθηκεύονται σε έναν πίνακα μεγέθους $7 \times 4 \times 4 \times 7$.

Παρατηρούμε ότι ο παραπάνω πίνακας περιέχει διαστάσεις μεγέθους 7. Αυτές αντιστοιχούν σε κάποιο υπαρκτό ζεύγος. Οι επτά δυνατοί συνδυασμοί είναι οι ακόλουθοι

1. AU
2. UA
3. CG
4. GC
5. GU
6. UG
7. MISMATCH

Όπου η τελευταία περίπτωση δεν εκφράζει κάποιο πραγματικό ζεύγος αλλά την αδυναμία δημιουργίας ζεύγους.

Στην συνέχεια θα χρησιμοποιούμε την συνάρτηση

$$pairing(b1, b2) = \begin{cases} AU\ when\ b1 = A\ and\ b2 = U \\ UA\ when\ b1 = U\ and\ b2 = A \\ CG\ when\ b1 = C\ and\ b2 = G \\ GC\ when\ b1 = G\ and\ b2 = C \\ GU\ when\ b1 = G\ and\ b2 = U \\ UG\ when\ b1 = U\ and\ b2 = G \\ MISMATCH\ when\ others \end{cases}$$

για να αναφερθούμε στο ζεύγος που μπορεί να δημιουργηθεί από δύο βάσεις.

Έτσι για το παρακάτω σχήμα

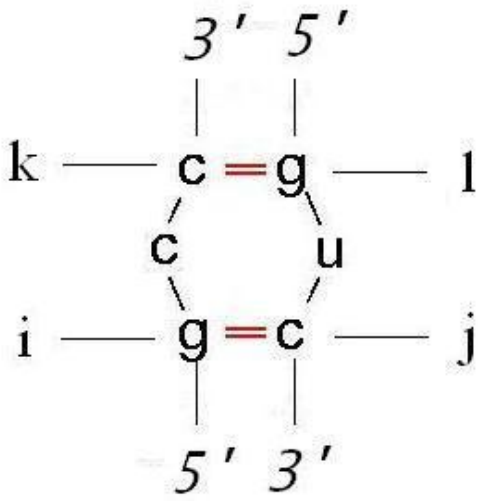


Figure 23- 1x1 Interior Loop Energies

η αναφορά στην ενέργεια αυτή θα γίνεται μέσω της συνάρτησης

$$ei1x1(pairing(i, j), pairing(k, l), i + 1, j - 1)$$

Ή ισοδύναμα

$$ei1x1(gc, cg, c, u)$$

1×2 loop energies

Στην συνέχεια έχουμε τα 1×2 Interior loops. Πρόκειται για μη-συμμετρικά Interior loops μεγέθους τρία. Τα πειραματικά δεδομένα αποθηκεύονται σε έναν πίνακα μεγέθους $7 \times 4 \times 4 \times 4 \times 7$.

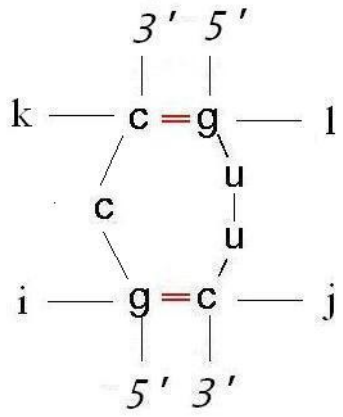


Figure 24 - 1x2 Interior Loop Energies

Η αναφορά στην ενέργεια αυτή θα γίνεται μέσω της συνάρτησης

$$ei1x2(pairing(i,j), pairing(k,l), i+1, l+1, j-1)$$

Ή ισοδύναμα : $ei1x2(gc, cg, c, u, u)$

Αντίστοιχα στην περίπτωση των 2x1 interior loops

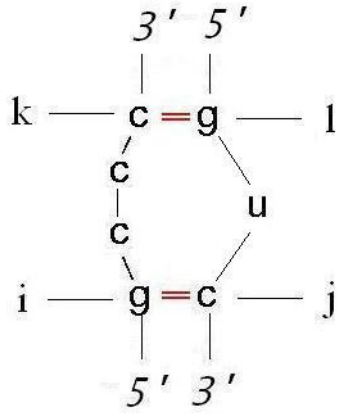


Figure 25 - 2x1 Interior Loop Energies

χρησιμοποιείται η εξής σύμβαση

$$ei1x2(pairing(l,k), pairing(j,i), l+1, i+1, k-1)$$

Ή ισοδύναμα : $ei1x2(gc, cg, u, c, c)$

2×2 loop energies

Τέλος για τις ειδικές περιπτώσεις των Interior loops έχουμε τα 2x2 loops. Πρόκειται για συμμετρικά Interior loops μεγέθους τέσσερα. Τα πειραματικά δεδομένα αποθηκεύονται σε έναν πίνακα μεγέθους $7 \times 4 \times 4 \times 4 \times 4 \times 7$.

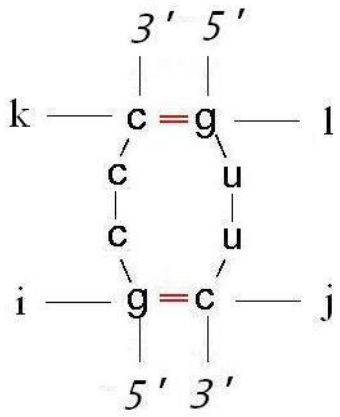


Figure 26 - 2x2 Interior Loop Energies

Για την αναφορά στην ενέργεια αυτή θα χρησιμοποιήσουμε την συνάρτηση

$$ei2x2(pairing(i,j), pairing(k,l), i+1, k-1, l+1, j-1)$$

Ή ισοδύναμα

$$ei2x2(gc, cg, u, c, c)$$

Ειδικές περιπτώσεις για Hairpin Loop

Στην γενική περίπτωση τα hairpin loop σχηματίζουν μη ενεργειακά σταθερά loops, υπάρχουν όμως κάποιοι συνδυασμοί βάσεων οι οποίοι παρουσιάζουν σημαντική σταθερότητα. Οι συνδυασμοί αυτοί συναντώνται σε hairpin loops μεγέθους τρία και τέσσερα και αποτελούν τα tri-loops και tetra-loops αντίστοιχα. Κάθε loop που ανήκει σε αυτές τις κατηγορίες λαμβάνει έναν επιπλέον σταθερό παράγοντα σταθεροποίησης, κοινό για όλα τα tetra-loops και τα tri-loops.

Επιπλέον loops που σχηματίζονται αποκλειστικά από βάσεις τύπου C ή περιέχουν μια αλυσίδα από τρείς συνεχόμενες βάσεις τύπου G λαμβάνουν ένα επιπλέον παράγοντα σταθεροποίησης, ο οποίος ορίζεται ως μέρος της παραμετροποίησης του αλγορίθμου.

Tetra-loops

Τα hairpin loops που ανήκουν σε αυτήν την κατηγορία είναι τα εξής

AGAAAU	**CGCGAG**	**UGCAAA**	**GUCCGU**
AGCAAU	**CUCCGG**	**UGAGAA**	**GGUAAU**
AGAGAU	**CGUAAG**	**UGUGAA**	**GCUUGU**
AGUGAU	**CCUUGG**	**UGGAAA**	**GAUUUU**
AGGAAU	**CAUUUG**	**UUUCGA**	**GUUUAU**
AUUCGU	**CUUUAG**	**UUACGA**	**UGAAAG**
AUACGU	**GGAAAC**	**UGCGAA**	**UGCAAG**
AGCGAU	**GGCAAC**	**UUCCGA**	**UGAGAG**
AUCCGU	**GGAGAC**	**UGUAAA**	**UGUGAG**
AGUAAU	**GGUGAC**	**UCUUGA**	**UGGAAG**
ACUUGU	**GGGAAC**	**UAUUUA**	**UUUCGG**
AAUUUU	**GUUCGC**	**UUUUAA**	**UUACGG**
AUUUAU	**GUACGC**	**GGAAAU**	**UGCGAG**
CGAAAG	**GGCGAC**	**GGCAAU**	**UUCCGG**
CGCAAG	**GUCCGC**	**GGAGAU**	**UGUAAG**
CGAGAG	**GGUAAC**	**GGUGAU**	**UCUUGG**
CGUGAG	**GCUUGC**	**GGGAAU**	**UAUUUG**
CGGAAG	**GAUUUC**	**GUUCGU**	**UUUUAG**
CUUCGG	**GUUUAC**	**GUACGU**	
CUACGG	**UGAAAA**	**GGCGAU**	

Παρατηρούμε ότι για την κατάταξη των hairpin loop στην ειδική κατηγορία των tetra-loops χρησιμοποιούνται τόσο οι βάσεις που περιέχονται στο loop όσο και το ζεύγος που κλείνει το loop.

Η αναφορά στην ενέργεια αυτή θα γίνεται μέσω της συνάρτησης

$$tetra(i, i+1, i+2, j-2, j-1, j)$$

Όπου (i, j) το ζεύγος που τερματίζει το hairpin loop. Η τιμή της συνάρτησης αυτής θα είναι μηδέν στην περίπτωση που οι βάσεις δεν σχηματίζουν κάποιο hairpin loop ή τον παράγοντα σταθεροποίησης αντίθετα.

Tri-loops

Σε αυτήν την κατηγορία δεν έχουν ανακοινωθεί ακόμα τα πειραματικά αποτελέσματα.

Για να καλυφθεί η περίπτωση αυτή δημιουργήθηκε η συνάρτηση

$$tri(i, i+1, i+2, j-1, j)$$

Η οποία επιστρέφει πάντα μηδέν.

CCC-Loop

Στην περίπτωση όπου όλες οι βάσεις που περιέχονται στο hairpin loop είναι τύπου C το Loop ονομάζεται CCC-Loop και λαμβάνει έναν επιπλέον παράγοντα σταθεροποίησης.

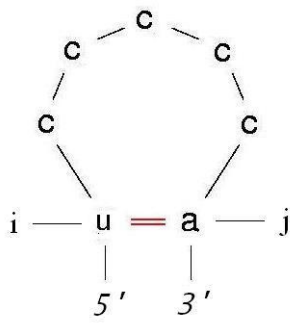


Figure 27 - CCC-Loop

Ο επιπλέον παράγοντας σταθεροποίησης υπολογίζεται με βάση το μέγεθος του hairpin loop.

$$cccloop(size) = \begin{cases} c\_hairpin\_of\_3 \; when \; size = \; 3 \\ c\_hairpin\_intercept \; + size \times c\_hairpin\_slope \; when \; others \end{cases}$$

Όπου οι σταθερές

- $c\_hairpin\_of\_3$
- $c\_hairpin\_intercept$
- $c\_hairpin\_slope$

Ορίζονται ως μέρος της παραμετροποίησης του αλγορίθμου.

GGG-Loop

Τέλος αν οι τρείς τελευταίες βάσεις είναι τύπου G το loop ονομάζεται GGG-Loop

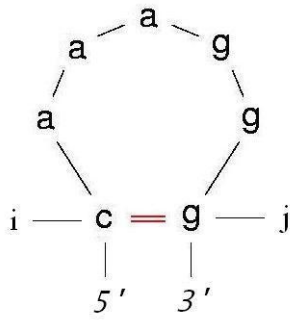


Figure 28 - GGG-Loop

και λαμβάνει έναν επιπλέον παράγοντα σταθεροποίησης, ο οποίος ορίζεται ως μέρος της παραμετροποίησης του αλγορίθμου.

Miscellaneous energies

Τέλος υπάρχει ένας αριθμός μεταβλητών οι οποίες χρησιμοποιούνται για την παραμετροποίηση του συστήματος.

Οι μεταβλητές αυτές είναι οι εξής

1. Η ελάχιστη απόσταση μεταξύ βάσεων που μπορούν να σχηματίσουν ζεύγος

   Η προκαθορισμένη καθώς και ελάχιστη τιμή που μπορεί να πάρει η παράμετρος αυτή είναι τέσσερα, ορίζοντας έτσι ως τρία το ελάχιστο μέγεθος για ένα hairpin loop.

2. Ο παράγοντας σταθεροποίησης για τα tetraloops και triloops

   Η προκαθορισμένη τιμή για αυτήν την παράμετρο είναι -2.0.

3. Ο παράγοντας σταθεροποίησης για τα GGG-Loops

   Η προκαθορισμένη τιμή για αυτήν την παράμετρο είναι μηδέν, αγνοώντας έτσι την ειδική περίπτωση των GGG-Loops.

4. Οι μεταβλητές για τον υπολογισμό του παράγοντα σταθεροποίησης στα CCC-Loop

   a. $c\_hairpin\_of\_3$
   b. $c\_hairpin\_intercept$
   c. $c\_hairpin\_slope$

   Οι προκαθορισμένες τιμές για τις μεταβλητές αυτές είναι μηδέν, αγνοώντας έτσι την ειδική περίπτωση των CCC-Loop.

5. Και οι μεταβλητές για τον υπολογισμό της ενέργειας για τα Multiloops

   a. Offset
   b. Free base penalty
   c. Helix penalty

   Η προκαθορισμένες τιμές για τις παραμέτρους αυτές είναι 4.6, 0.4 και 0.1 για τα Offset, Free base penalty και Helix penalty αντίστοιχα στην περίπτωση των Exterior loop και 10.1, -0.3, -0.3 στην περίπτωση των multi-loops. Η διαφοροποίηση αυτή γίνεται έτσι ώστε το helix penalty να έχει αρνητική τιμή και να μπορέσουμε να αποφύγουμε κάποιους ελέγχους στο στάδιο του υπολογισμού.

Υπολογισμός ενεργειών των loops

Για τον υπολογισμό της ενέργειας του κάθε loop τα πειραματικά δεδομένα θα πρέπει να συνδυαστούν, ανάλογα με της τάξη και το μέγεθος τους. Στην συνέχεια παρουσιάζεται ο τρόπος με τον οποίο υπολογίζεται η ενέργεια αυτή σε κάθε περίπτωση [11] [12].

Ενέργεια για Hairpin loop

Η ενέργεια για hairpin-loops υπολογίζεται ως το άθροισμα των παρακάτω όρων

1. Terminal mismatch stacking energy για τα hairpin loop
2. Loop destabilizing energy για τα hairpin loop
3. Τον παράγοντα σταθεροποίησης για κάθε μία από τις τέσσερις ειδικές περιπτώσεις των Hairpin loops

    a. Tetraloops
    b. Triloops
    c. CCC-Loops
    d. GGG-Loops

Σύμφωνα με τις συναρτήσεις που ορίστηκαν για την αναφορά στα πειραματικά δεδομένα, ο υπολογισμός της ενέργειας του loop ανάγεται στον υπολογισμό της συνάρτησης

$$eh(i,j) = hltm(i,i+1,j-1,j) + hlde(j-i-1) + tetra(i,i+1,i+2,j-2,j-1,j) + tri(i,i+1,i+2,j-1,j) + ehbonus(i,j)$$

όπου

$$ehbonus(i,j)\begin{cases} cccloop(j-i-1), & \text{όταν το Loop είναι } CCC-Loop \\ gggloop, & \text{όταν το Loop είναι } CCC-Loop \\ 0, & \text{αλλιώς} \end{cases}$$

Και (i,j) το ζεύγος του τερματίζει το hairpin loop.

Ενέργεια για Bulge

Στην περίπτωση των bulge η ενέργεια του loop είναι ίση με την ενέργεια αποσταθεροποίησης των bulges (Loop destabilizing energy) με εξαίρεση τα loops μεγέθους ένα στα οποία προστίθεται η ενέργεια για staking pair μεταξύ των δύο ζευγών που σχηματίζουν το Loop.

Επιπλέον υπάρχει μια μικρή διαφοροποίηση μεταξύ των δεξιών και αριστερών bulges όσων αφορά τον υπολογισμό του μεγέθους τους. Έτσι έχουμε δύο διαφορετικές συναρτήσεις. Και στις δύο περιπτώσεις θεωρούμε ότι το ζεύγος που τερματίζει το loop είναι το (i,j) ενώ το (k,l) περιέχεται στο loop.

Αριστερό Bulge

Για τα αριστερά bulge το μέγεθος υπολογίζεται ως $k-i-1$ και έτσι έχουμε την παρακάτω συνάρτηση

$$ebl(i,j,k,l) = \begin{cases} blde(k-i-1) + es(i,k,l,j), & \text{όταν } k-i-1=1 \\ blde(k-i-1) + \quad 0 \quad , & \text{αλλιώς} \end{cases}$$

Δεξιό Bulge

Για τα δεξιά Bulge το μέγεθος υπολογίζεται ως $j-l-1$ και έχουμε αντίστοιχα

$$ebr(i,j,k,l) = \begin{cases} blde(j-l-1) + es(i,k,l,j), & \text{όταν } j-l-1=1 \\ blde(j-l-1) + \quad 0 \quad , & \text{αλλιώς} \end{cases}$$

Ενέργεια για Interior loop

Στην συνέχεια για τον υπολογισμό της ενέργειας των interior loop έχουμε να επιλέξουμε ανάμεσα σε πέντε διαφορετικές τιμές, τις τέσσερις ειδικές περιπτώσεις των 1×1, $1\times2, 2\times1$ και 2×2 loops και την γενική περίπτωση για τις υπόλοιπες περιπτώσεις. Στις τέσσερις πρώτες η ενέργεια δίνεται έτοιμη από τα δεδομένα που έχουμε διαθέσιμα ενώ για την τελευταία θα πρέπει να υπολογιστεί το άθροισμα των παρακάτω όρων

1. Terminal mismatch stacking energy για το ζεύγος που τερματίζει το loop
2. Terminal mismatch stacking energy για το ζεύγος που περιέχεται στο loop
3. Loop destabilizing energy για τα Interior loop

Έτσι ο υπολογισμός της ενέργειας του loops ανάγεται στον υπολογισμό της παρακάτω συνάρτησης

$$ei(i,j,k,l) = \begin{cases} ei1x1(pairing(i,j), pairing(k,l), i+1, j-1) & ,\text{όταν } k-i=2 \text{ και } j-l=2 \\ ei1x2(pairing(i,j), pairing(k,l), i+1, l+1, j-1) & ,\text{όταν } k-i=2 \text{ και } j-l=3 \\ ei1x2(pairing(l,k), pairing(j,i), l+1, i+1, k-1) & ,\text{όταν } k-i=3 \text{ και } j-l=2 \\ ei2x2(pairing(i,j), pairing(k,l), i+1, k-1, l+1, j-1), & \text{όταν } k-i=3 \text{ και } j-l=3 \\ \quad iltm(i, i+1, j-1, j) + iltm(l, l+1, k-1, k) & \\ \quad +ilde(j-l+k-i-2) & ,\text{αλλιώς} \end{cases}$$

Όπου (i,j) το ζεύγος που τερματίζει το Loop και (k,l) το ζεύγος που περιέχεται σε αυτό.

Ενέργεια για Multi loops

Τέλος για τον υπολογισμό των k-loops τάξεως μεγαλύτερης του δύο χρησιμοποιείται ο παρακάτω προσεγγιστικός τύπος.

$$\text{"Offset"} + \text{"Free base penalty"} \times k' + \text{"Helix penalty"} \times k$$

Όπου k η τάξη και k' το μέγεθος του Multi-loop. Επιπλέον στο τελικό αποτέλεσμα προστίθεται και οι ενέργειες $d5(\cdot,\cdot,\cdot)$ και $d3(\cdot,\cdot,\cdot)$ που σχηματίζονται μεταξύ των ζευγών που περιέχονται στο Multi-loop και των προσκείμενων σε αυτά ελευθέρων βάσεων.

Αναδρομικές σχέσεις για τον υπολογισμό των υπό-προβλημάτων

Μέχρι αυτό το σημείο έχουν αναλυθεί οι βασικές δομές που "αναγνωρίζει" ο αλγόριθμος καθώς και ο τρόπος υπολογισμού της ενέργειας του κάθε ενός από αυτούς με χρήση των πειραματικών δεδομένων που είναι διαθέσιμα. Στην συνέχεια θα πρέπει τα k-loop να συνδυαστούν με τέτοιο τρόπο έτσι ώστε να εξαχθεί η βέλτιστη δομή, δηλαδή η δομή με την ελάχιστη ενέργεια Gibbs.

Μια μέθοδος άπληστης αναζήτησης (brute force), όπως αναζήτηση σε βάθος ή πλάτος, επιφέρει έναν εκθετικό χρόνο εκτέλεσης στην εύρεση της βέλτιστης λύσης. Για τον αλγόριθμο του Zucker όπου μια τυπική είσοδος αποτελείται από RNA μεγέθους από μερικές εκατοντάδες έως μερικές χιλιάδες νουκλεοτίδια η μέθοδος αυτή είναι μη αποτελεσματική καθώς πολύ γρήγορα ο αλγόριθμός φτάνει σε μη ρεαλιστικούς χρόνους εκτέλεσης.

Για την εξομάλυνση αυτού του προβλήματος ο αλγόριθμος χρησιμοποιεί μια προσέγγιση δυναμικού προγραμματισμού αποθηκεύοντας τα ενδιάμεσα αποτελέσματα με σκοπό την επαναχρησιμοποίηση τους [11] [12]. Έτσι κερδίζουμε σε χρόνο εκτέλεσης επιτυγχάνοντας πολυωνυμικούς χρόνους εκτελέσεις εις βάρος τις ανάγκης για μεγάλη ποσότητα κύριας μνήμης για την αποθήκευση των ενδιάμεσων αποτελεσμάτων.

Ο αλγόριθμος χρησιμοποιεί τρεις άνω τριγωνικούς πίνακες διαστάσεων $L \times L$, $W(\cdot,\cdot)$, $V(\cdot,\cdot)$, $L(\cdot,\cdot)$ όπου L ο αριθμός των βάσεων στο μακρομόριο του RNA. Κάθε στοιχείο από αυτούς τους τρείς πίνακες αντιστοιχεί στην ενέργεια Gibbs κάποιας δομής, συγκεκριμένα

- Το στοιχείο $W(i,j)$ αντιστοιχεί στην ενέργεια Gibbs του βέλτιστου προβλήματος r(5$'b_i \ldots b_j 3'$)
- Το στοιχείο $V(i,j)$ αντιστοιχεί στην ενέργεια Gibbs του βέλτιστου προβλήματος r(5$'b_i \ldots b_j 3'$), θεωρώντας ότι οι βάσεις i,j σχηματίζουν ζεύγος
- Το στοιχείο $L(i,j)$ αντιστοιχεί στην ενέργεια Gibbs του βέλτιστου προβλήματος r(5$'b_i \ldots b_j 3'$), το οποίο προέκυψε από συνένωση δύο άλλων βέλτιστων προβλημάτων σε σειρά.

Από τα παραπάνω γίνεται εμφανές ότι ο πίνακας W είναι εκείνος ο οποίος περιέχει την πραγματική βέλτιστη τιμή ενώ οι πίνακες V, L περιέχουν προσωρινές τιμές κάποιου υποσυνόλου των περιπτώσεων της βέλτιστης δομής, οι οποίες θα χρησιμοποιηθούν αργότερα.

Στην συνέχεια παρουσιάζονται οι αναδρομικές σχέσεις για τον υπολογισμό των τιμών αυτών, όπου χάριν απλότητας θα θεωρήσουμε ότι

- a : "Offset"
- b : "Free base penalty"
- c : "Helix penalty"

Πίνακας V

Στον πίνακα αυτό αποθηκεύονται οι βέλτιστες τιμές για τις υπό-ακολουθίες των οποίων οι δύο βάσεις στα άκρα τους σχηματίζουν ζεύγος. Για τον υπολογισμό των τιμών αυτού του πίνακα χρησιμοποιείται η παρακάτω φόρμουλα.

$$V(i,j) = min \begin{cases} eh(i,j) & (1) \\ es(i,j) + V(i+1, j-1) & (2) \\ ebl(i,j,k) + V(k, j-1), \quad \forall k \in [i+2, j-4) & (3) \\ ebr(i,j,k) + V(i+1, k), \quad \forall k \in [i+5, j-1) & (4) \\ ei(i,j,k,l) + V(k,l), \quad \forall k,l: i+1 < k, l < j-1, l-k > 3 & (5) \\ a + c + L(i+1, j-1) & (6) \\ a + c + L(i+2, j-1) + d5(i,j,i+1) + b & (7) \\ a + c + L(i+1, j-2) + d3(i,j,j-1) + b & (8) \\ a + c + L(i+2, j-2) + d5(i,j,i+1) + d3(i,j,j-1) + 2b & (9) \end{cases}$$

Παρατηρούμε ότι ο υπολογισμός της τιμής κάθε στοιχείου του πίνακα ανάγεται στην επιλογή της καλύτερης τιμής ανάμεσα από εννέα διαφορετικές περιπτώσεις. Αξίζει να σημειωθεί ότι κάθε μία από αυτές τις περιπτώσεις αντιστοιχεί στην εισαγωγή ενός καινούριου k-loop σε κάποια προηγούμενη βέλτιστη δομή.

Περίπτωση 1: Η βέλτιστη υπό-ακολουθία είναι ένα Hairpin loop.

Στην περίπτωση αυτή η βέλτιστη υπό-ακολουθία αποτελεί ένα Hairpin loop. Ο υπολογισμός της ενέργειας της γίνεται μέσω της συνάρτησης $eh(i,j)$ όπως έχει ήδη αναλυθεί.

Περίπτωση 2: Εισαγωγή ενός Stacked Pair στην βέλτιστη υπό-ακολουθία r(5'b_{i+1}...b_{j-1}3')

Πέραν της βασικής περίπτωσης (Hairpin loop) η ενέργεια της βέλτιστης υπό-ακολουθίας πρέπει να υπολογιστεί χρησιμοποιώντας αποτελέσματα από προηγούμενα υπό-προβλήματα.

Στην περίπτωση αυτή χρησιμοποιούμε την ενέργεια $es(i,j)$ για να πάρουμε την ενέργεια του Stacked Pair και την προσθέτουμε στην ενέργεια $V(i+1, j-1)$ η οποία μας παρέχει την ενέργεια για το υπόλοιπο τμήμα της υπό-ακολουθίας.

Περίπτωση 3 και 4 : Εισαγωγή ενός Bulge σε κάποια βέλτιστη υπό-ακολουθία

Στις δύο αυτές περιπτώσεις έχουμε περισσότερες από μία πιθανές λύσεις καθώς το bulge μπορεί να περιέχει μεταβλητό αριθμό βάσεων. Έτσι θα πρέπει να υπολογιστεί κάθε μία από αυτές έτσι ώστε να λάβουμε την βέλτιστη.

Σε κάθε περίπτωση υπολογίζουμε την ενέργεια του Bulge με βάση τις συναρτήσεις $ebl(i,j,k)$ και $ebr(i,j,k)$ για αριστερά και δεξιά Bulge αντίστοιχά και τα προσθέτουμε στην αντίστοιχη τιμή για το υπόλοιπο τμήμα της υπό-ακολουθίας $V(k, j-1)$ και $V(i+1, k)$.

Περίπτωση 5: Εισαγωγή ενός Interior Loop σε κάποια βέλτιστη υπό-ακολουθία

Αντίστοιχα με τα Bulge υπολογίζονται και οι ενέργειες για τα Interior loop. Υπολογίζουμε την ενέργεια του loop, $ei(i,j,k,l)$, για κάθε πιθανό Loop και προσθέτουμε σε αυτή την αντίστοιχη τιμή για το υπόλοιπο τμήμα της υπό-ακολουθίας $V(k,l)$.

Περιπτώσεις 6, 7, 8 και 9: Εισαγωγή Multi-loop σε κάποια βέλτιστη δομή

Στις τέσσερις τελευταίες περιπτώσεις έχουμε την εισαγωγή ενός Multi-loop. Κύρια διαφορά με τις προηγούμενες περιπτώσεις είναι ότι προσεγγιστικός τύπος για τον υπολογισμό της ενέργειας του multi-loop δεν μπορεί να υπολογιστεί, καθώς δεν αποθηκεύεται η ενδιάμεση δομή αλλά μόνο η ενέργειά της. Κάτι τέτοιο θα απατούσε πολύ μεγαλύτερη ανάγκη για χώρο καθιστώντας τον αλγόριθμο μη αποδοτικό για μεγάλες ακολουθίες RNA.

Έτσι για τον υπολογισμό της ενέργειας των multi-loops οι αντίστοιχες ενέργειες ενσωματώνονται σε διάφορα σημεία του αλγορίθμου έτσι ώστε προοδευτικά να πάρουμε το σωστό αποτέλεσμα. Σε αυτό το σημείο θα υποθέσουμε ότι λαμβάνοντας την ενέργεια του προηγούμενου υπό-προβλήματος, π.χ. $L(i+1, j-1)$, θα πάρουμε μια τιμή η οποία θα περιέχει τόσο το "Helix penalty" όσο και το "Free base penalty" για όλες της βάσεις και τα ζεύγη που περιέχονται στο υπό δημιουργία loop.

Στην τιμή αυτή θα προσθέσουμε την σταθερά $a + c$ δηλαδή το "Offset" και το "Helix penalty" για το τελευταίο ζεύγος, το ζεύγος που κλείνει το loop.

Οι τέσσερις διαφορετικές περιπτώσεις αντιστοιχούν στην ύπαρξη ή όχι κάποιας ελεύθερης βάσης $i+1$ ή $j-1$ δίπλα στο ζεύγος (i, j), έτσι ώστε να προστεθεί η ενέργεια $d5$ και/ή $d3$ μεταξύ τους καθώς και "Free base penalty" για εκείνη την βάση.

Τέλος παρατηρούμε ότι για τις περιπτώσεις 2 έως και 5 η τιμή του υπό-προβλήματος λαμβάνεται από τον πίνακα V ενώ στις περιπτώσεις 6 έως και 9 από τον πίνακα L. Αυτό γίνεται γιατί για τον υπολογισμό ενός Multi-loop θα πρέπει να εξασφαλίσουμε ότι θα περιέχονται τουλάχιστον δύο ζεύγη στο loop επιπλέον τα άκρα του υπό-προβλήματος δεν σχηματίζουν ζεύγος. Έτσι επιλέγοντας μια τιμή από τον πίνακα L του οποίου τα αποτελέσματα λαμβάνονται από την σύνθεση δύο διαφορετικών υπό-προβλημάτων εξασφαλίζουμε αυτές τις δύο προϋποθέσεις. Αντίστοιχα στην περίπτωση των 2-Loop επειδή η τιμή που χρησιμοποιούμε αντιστοιχεί στην ενέργεια που λαμβάνει η υπό-ακολουθία η οποία τερματίζεται από το ζεύγος που περιέχεται στο Loop, παίρνουμε την τιμή από τον πίνακα V.

Πίνακας L

Στον πίνακα αυτό αποθηκεύονται οι βέλτιστες τιμές για τις υπό-ακολουθίες οι οποίες προκύπτουν από την συνένωση δύο άλλων βέλτιστων υπό-ακολουθιών σε σειρά. Ο ρόλος του πίνακα αυτού είναι να δημιουργήσει τις προϋποθέσεις για την δημιουργία multi-loops καθώς και να μειώσει τον συνολικό υπολογιστικό χρόνο αφού κάθε αποτέλεσμα θα χρησιμοποιηθεί συνολικά τουλάχιστον πέντε φορές.

Οι βέλτιστες υπό-ακολουθίες λαμβάνονται από τον πίνακα W έτσι ώστε να περιέχουν την συνολική βέλτιστη λύση για κάθε υπό-πρόβλημα.

$$L(i,j) = \min_k W(i,k) + W(k+1,j)\ ,\quad \forall k \in [i+4, j-4)$$

Πίνακας W

Στον πίνακα αυτό αποθηκεύονται οι βέλτιστες τιμές για το σύνολο των υπό-ακολουθιών. Για τον υπολογισμό των τιμών του πίνακα W χρησιμοποιείται η παρακάτω φόρμουλα.

$$W(i,j) = min \begin{cases} V(i,j) & (1) \\ L(i,j) & (2) \\ b + c + d3(i+1,j,i) + V(i+1,j) & (3) \\ b + c + d5(i,j-1,j) + V(i,j-1) & (4) \\ b + W(i+1,j) & (5) \\ b + W(i,j-1) & (6) \\ 2b + c + d3(i+1,j,i) + d5(i,j-1,j) + V(i+1,j-1) & (7) \end{cases}$$

Παρατηρούμε ότι ο υπολογισμός ανάγεται στην επιλογή της καλύτερης μεταξύ εφτά περιπτώσεων.

Περίπτωση 1 και 2 : Επιλογή της αντίστοιχης ενέργειας από τους πίνακες V και L

Όντας υπερσύνολο των περιπτώσεων που αντιστοιχούν στους πίνακες V και L το αποτέλεσμα επιλέγεται το αποτέλεσμα από τους δύο πίνακες για να καλυφτούν αυτές οι περιπτώσεις.

Περιπτώσεις 3, 4 και 7 : Εισαγωγή ελεύθερης βάσης προσκείμενη σε κάποιο ζεύγος

Στην συνέχεια έχουμε την περίπτωση της εισαγωγής ελεύθερης βάσεις δεξιά, αριστερά ή και από τις δύο πλευρές κάποιου ζεύγους. Στην περίπτωση αυτή προσθέτουμε το "Helix penalty" για το ζεύγος έτσι ώστε να σχηματίζεται προοδευτικά το κόστος για τα Multi-loops καθώς και η ενέργεια $d5$ και/ή $d3$ και το "Free base penalty" για κάθε μια από τις βάσεις οι οποίες θα εισαχθούν.

Η ενέργεια για το υπόλοιπο της υπό-ακολουθία παρέχεται από τον πίνακα V καθώς απαιτείται οι βάσεις να εισαχθούν δίπλα σε κάποιο ζεύγος.

Περίπτωση 5 και 6 : Εισαγωγή ελεύθερης βάσης προσκείμενη σε άλλη ελεύθερη βάση

Σε αυτήν την περίπτωση προστίθεται μόνο το "Free base penalty" καθώς οι υπόλοιπες τιμές λαμβάνονται αναδρομικά χρησιμοποιώντας την τιμή από τον πίνακα W.

ΠΡΟΣΟΧΗ : Ενώ ο αλγόριθμος επιτρέπει την εισαγωγή μιας ελεύθερης βάσεις δίπλα σε κάποιο ζεύγος χωρίς την προσθήκη των ενεργειών $d3$ ή $d5$, π.χ. με την επιλογή του κανόνα 5 έχοντας $W(i+1,j) \equiv V(i,j)$ κάτι τέτοιο αποτρέπεται επειδή η ενέργειες $d3$ και $d5$ συνεισφέρουν θετικά στην σταθεροποίηση του μακρομορίου και επομένως η επιλογή του κανόνα 3 θα είναι πάντα προτιμητέα.

Εξαρτήσεις μεταξύ δεδομένων

Για τον υπολογισμό των παραπάνω τιμών είναι αναγκαίο οι τιμές οι οποίες χρησιμοποιούνται να έχουν ήδη υπολογιστεί, αλλιώς τα αποτελέσματα θα είναι λανθασμένα. Στην συνέχεια παρουσιάζονται γραφικά οι εξαρτήσεις μεταξύ των τιμών που θέλουμε να υπολογίσουμε και των τιμών οι οποίες χρειάζονται για τον υπολογισμό της έτσι ώστε να εξάγουμε κάποια στρατηγική επίλυσης.

Οι εξαρτήσεις αυτές παρουσιάζονται ως ένας άνω τριγωνικός πίνακας με κορυφή το σημείο (i, j) δηλαδή την θέση του στοιχείου που θέλουμε να υπολογίσουμε. Πάνω στον πίνακα αυτό θα σημειωθούν τα στοιχεία τα οποία πρέπει να έχουν προϋπολογιστεί σε προηγούμενα βήματα του αλγορίθμου.

Πίνακας V

Για τον υπολογισμό ενός στοιχείου του πίνακα V χρησιμοποιούνται, από τον πίνακα L, τα στοιχεία $L(i+1, j-1)$, $L(i+1, j-2)$, $L(i+2, j-2)$ και $L(i+1, j-2)$, καθώς και όλα τα στοιχεία $V(k, l)$ για τα οποία ισχύει $k < i$, $l < j$ και $l - k > 3$, από τον πίνακα V.

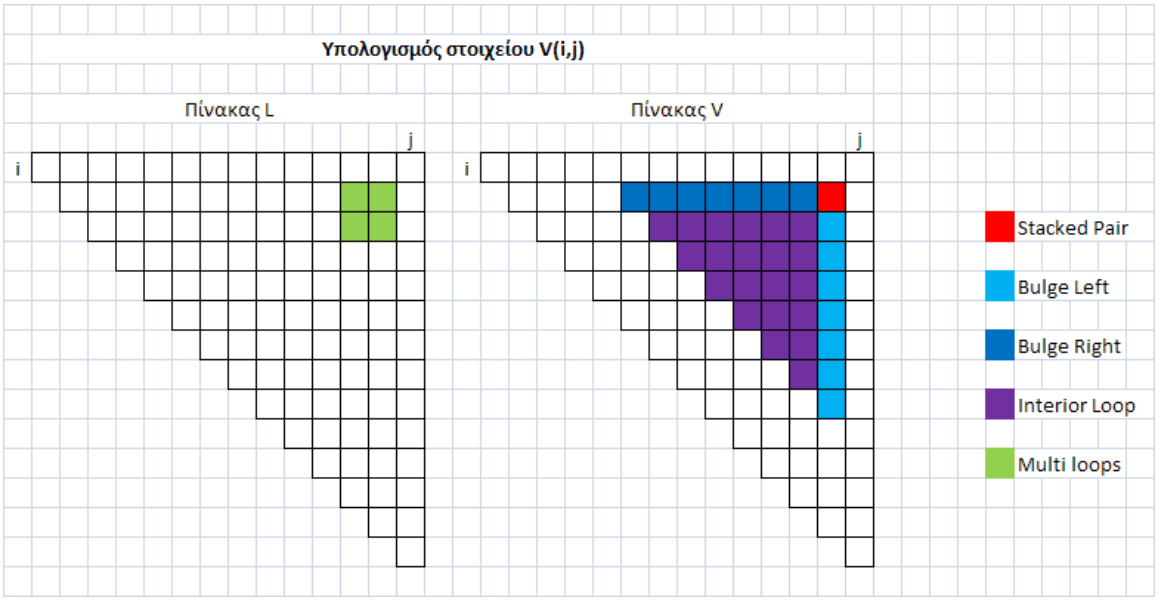


Figure 29 - Εξαρτήσεις του στοιχείου V(i,j)

Στην συνέχεια παρουσιάζεται ο αριθμός των στοιχείων που χρησιμοποιούνται από τον κάθε πίνακα για τον υπολογισμό της τελικής λύσης, σε σχέση με την διαφορά $j - i$, η οποία καθορίζει το «ύψος» της πυραμίδας.

$j - i$	Στοιχεία από τον πίνακα L	Στοιχεία από τον πίνακα V
< 4	-	-
$4, 5$	0	$\frac{(j-i-5)\cdot(j-i-4)}{2}$
6	1	
7	3	
> 7	4	

Table 5 - Αριθμός στοιχείων για τον υπολογισμό του V(i,j)

Πίνακας L

Ο πίνακας L χρησιμοποιεί στοιχεία μόνο από τον πίνακα W.

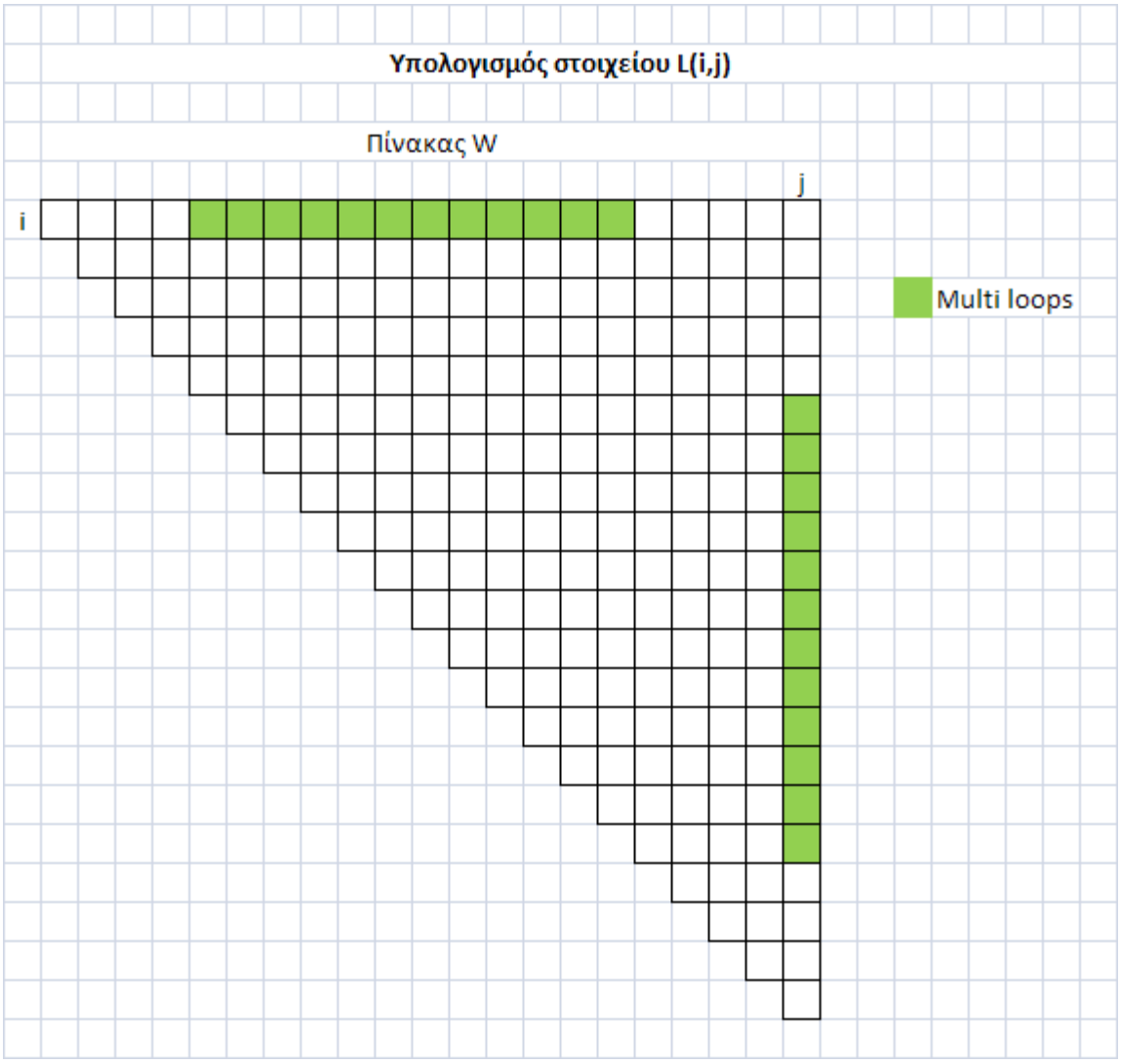


Figure 30 - Εξαρτήσεις του στοιχείου L(i,j)

Στην συνέχεια παρουσιάζεται ο αριθμός των στοιχείων που χρησιμοποιούνται για τον υπολογισμό της τελικής λύσης, σε σχέση με την διαφορά $j - i$, η οποία καθορίζει το «ύψος» της πυραμίδας.

$j - i$	Στοιχεία από τον πίνακα W
≤ 8	-
> 8	$2 \cdot (j - i - 8)$

Table 6 - Αριθμός στοιχείων για τον υπολογισμό του L(i,j)

Πίνακας W

Ο πίνακας *W* χρησιμοποιεί στοιχεία από όλους τους πίνακες.

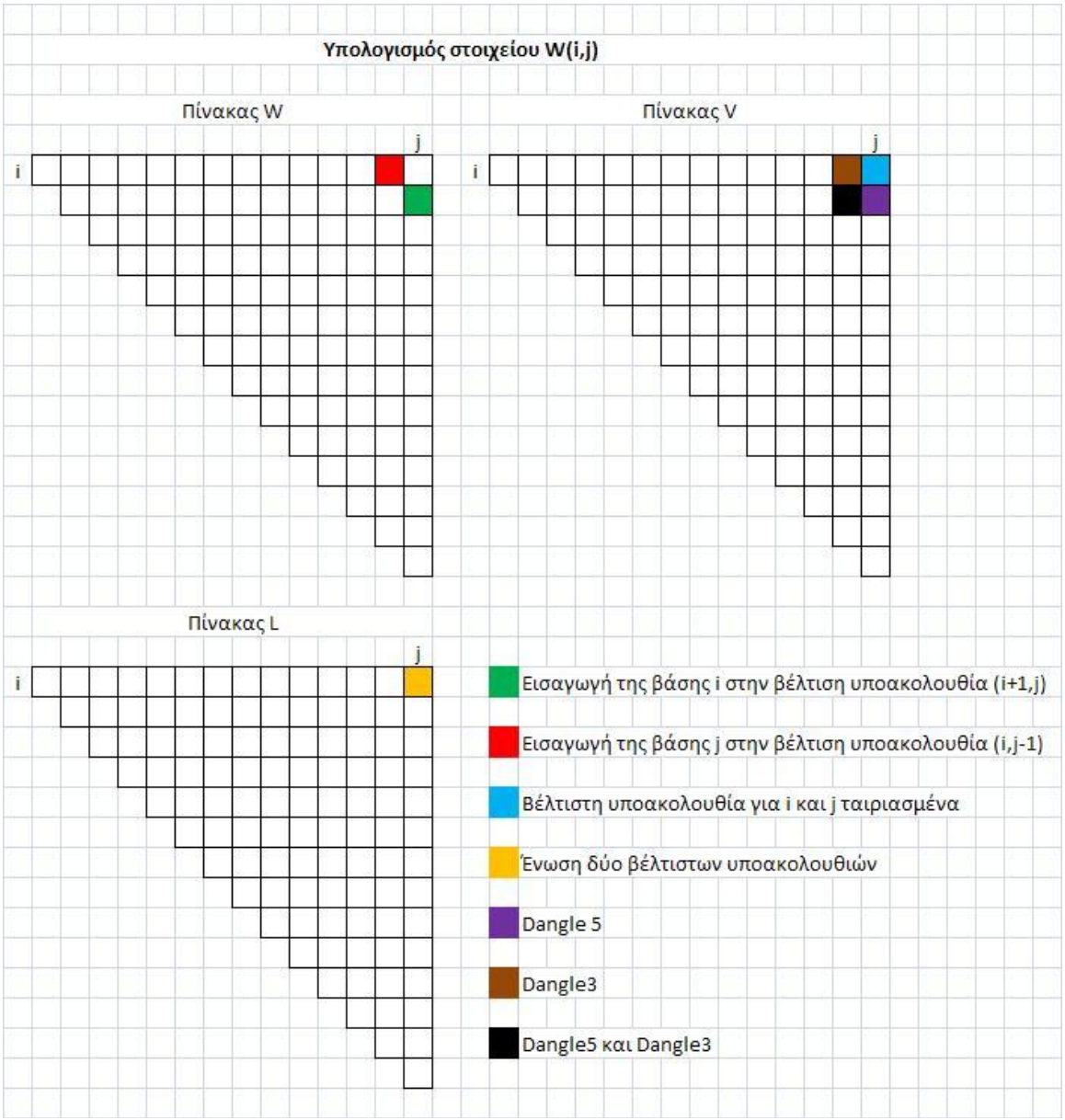


Figure 31 - Εξαρτήσεις του στοιχείου W(i,j)

Στην συνέχεια παρουσιάζεται ο αριθμός των στοιχείων που χρησιμοποιούνται για τον υπολογισμό της τελικής λύσης, σε σχέση με την διαφορά $j - i$.

$j - i$	Στοιχεία από τον πίνακα *W*	Στοιχεία από τον πίνακα *V*	Στοιχεία από τον πίνακα *L*
< 4	-	-	-
4	0	1	1
5	2	3	1
> 5	2	4	1

Table 7 - Αριθμός στοιχείων για τον υπολογισμό του W(i,j)

Συγκεντρωτικά ο αριθμός των στοιχείων που χρειάζονται για τον υπολογισμό κάθε των στοιχείων κάποιου πίνακα.

$j-i$	Πίνακας	Στοιχεία από τον πίνακα W	Στοιχεία από τον πίνακα V	Στοιχεία από τον πίνακα L	Συνολο
< 4	W, V, L	-	-	-	-
4	W	0	1	1	2
	V	-	0	0	0
	L	0	-	-	0
5	W	2	3	1	6
	V	-	0	0	0
	L	0	-	-	0
6	W	2	4	1	7
	V	-	1	1	2
	L	0	-	-	0
7	W	2	4	1	7
	V	-	3	3	6
	L	0	-	-	0
8	W	2	4	1	7
	V	-	6	4	10
	L	0	-	-	0
> 8	W	2	4	1	7
	V	-	$\frac{(j-i-5)\cdot(j-i-4)}{2}$	4	$\frac{(j-i-5)\cdot(j-i-4)}{2} + 4$
	L	$2 \cdot (j-i-8)$	-	-	$2 \cdot (j-i-8)$

Table 8 - Συγκεντρωτικά στοιχεία για τον αριθμό των απαιτούμενων στοιχείων για τον υπολογισμό των W(i,j), V(i,j) και L(i,j)

Στρατηγική υπολογισμού των υπό-προβλημάτων για την εξάλειψη των Εξαρτήσεων

Για την εξάλειψη των εξαρτήσεων θα εισαχθεί μια στρατηγική επίλυσης των υπό-προβλημάτων σύμφωνα με την οποία τα υπό-προβλήματα θα υπολογίζονται με τέτοια σειρά έτσι ώστε για κάθε ένα από αυτά να έχουν προϋπολογιστεί όλες οι απαραίτητες τιμές. Εν γένει σε προβλήματα δυναμικού προγραμματισμού υπάρχουν δύο ενδεικτικοί τρόποι για την σειρά επίλυσης, με την πρώτη να επιλύει τα υπό-προβλήματα ανά στήλη και την δεύτερη ανά διαγώνιους.

Αναλυτικότερα για την πρώτη περίπτωση, με επίλυση ανά στήλη και αριθμώντας τα στοιχεία του πίνακα ως εξής

11	12	13	14	15	…
	22	23	24	25	…
		33	34	35	…
			44	45	…
				55	…
					…

Η σειρά επίλυσης θα είναι

$$11 \rightarrow$$
$$22 \rightarrow 12 \rightarrow$$
$$33 \rightarrow 23 \rightarrow 13 \rightarrow$$
$$44 \rightarrow 34 \rightarrow 24 \rightarrow 14 \rightarrow$$
$$55 \rightarrow 45 \rightarrow 35 \rightarrow 25 \rightarrow 15 \rightarrow \cdots$$

Ενώ για την δεύτερη περίπτωση, με επίλυση ανά διαγώνιο, η σειρά επίλυσης θα είναι

$$11 \rightarrow 22 \rightarrow 33 \rightarrow 44 \rightarrow 55 \rightarrow \cdots \rightarrow$$
$$12 \rightarrow 23 \rightarrow 34 \rightarrow 45 \rightarrow \cdots \rightarrow$$
$$13 \rightarrow 24 \rightarrow 35 \rightarrow \cdots \rightarrow$$
$$14 \rightarrow 25 \rightarrow \cdots \rightarrow$$
$$15 \rightarrow \cdots$$

Στα πλαίσια της εργασίας αυτής, αν και η πρώτη περίπτωση αποτελεί μια απλούστερη και ευκατανόητη λύση, επιλέχθηκε η δεύτερη περίπτωση γιατί, όπως θα φανεί και αργότερα, απλοποιεί την παραλληλοποίηση του προβλήματος.

Εξαγωγή της δευτεροταγής δομής από τα υπό-προβλήματα

Μετά των υπολογισμό των υπό-προβλημάτων σειρά έχει η εξαγωγή της τελικής λύσης. Όμοια με τον αλγόριθμο Nussinov [9], η διαδικασία αυτή γίνεται μέσω ενώ αλγορίθμου οπισθοδρόμησης. Ο αλγόριθμος αυτός εκτελεί την ίδια λειτουργία ανάστροφα, ξεκινώντας δηλαδή από την τελική λύση και προχωρώντας προς τα μικρότερα υπό-προβλήματα και ανιχνεύει τον κανόνα ο οποίος χρησιμοποιήθηκε για τον υπολογισμό την λύσης. Αν ο κανόνας περιλαμβάνει και την εισαγωγή ενός ζεύγους τότε αυτό το ζεύγος θα ανήκει και στην τελική λύση.

Αλγόριθμος οπισθοδρόμησης (Trace Back)

Λόγο της ύπαρξης τριών πινάκων η διαδικασία αυτή είναι αρκετά περίπλοκη. Αρχικά θα παρουσιαστούν τα τρία βασικά τμήματα του αλγορίθμου, οι διαδικασίες για οπισθοδρόμηση σε κάθε πίνακα χωριστά και στο τέλος ο συνολικός αλγόριθμος.

Οπισθοδρόμηση στον πίνακα W

Η διαδικασία ξεκινά από τον πινάκα W καθώς και αυτός περιέχει την συνολικά βέλτιστη λύση. Το στάδιο αυτό περιλαμβάνει την αρχικοποίηση του αλγορίθμου οπισθοδρόμησης ορίζοντας τις τιμές I και J και στην συνέχεια ελέγχει όλες τις δυνατές περιπτώσεις για τον υπολογισμό του στοιχείου $W(I,J)$.

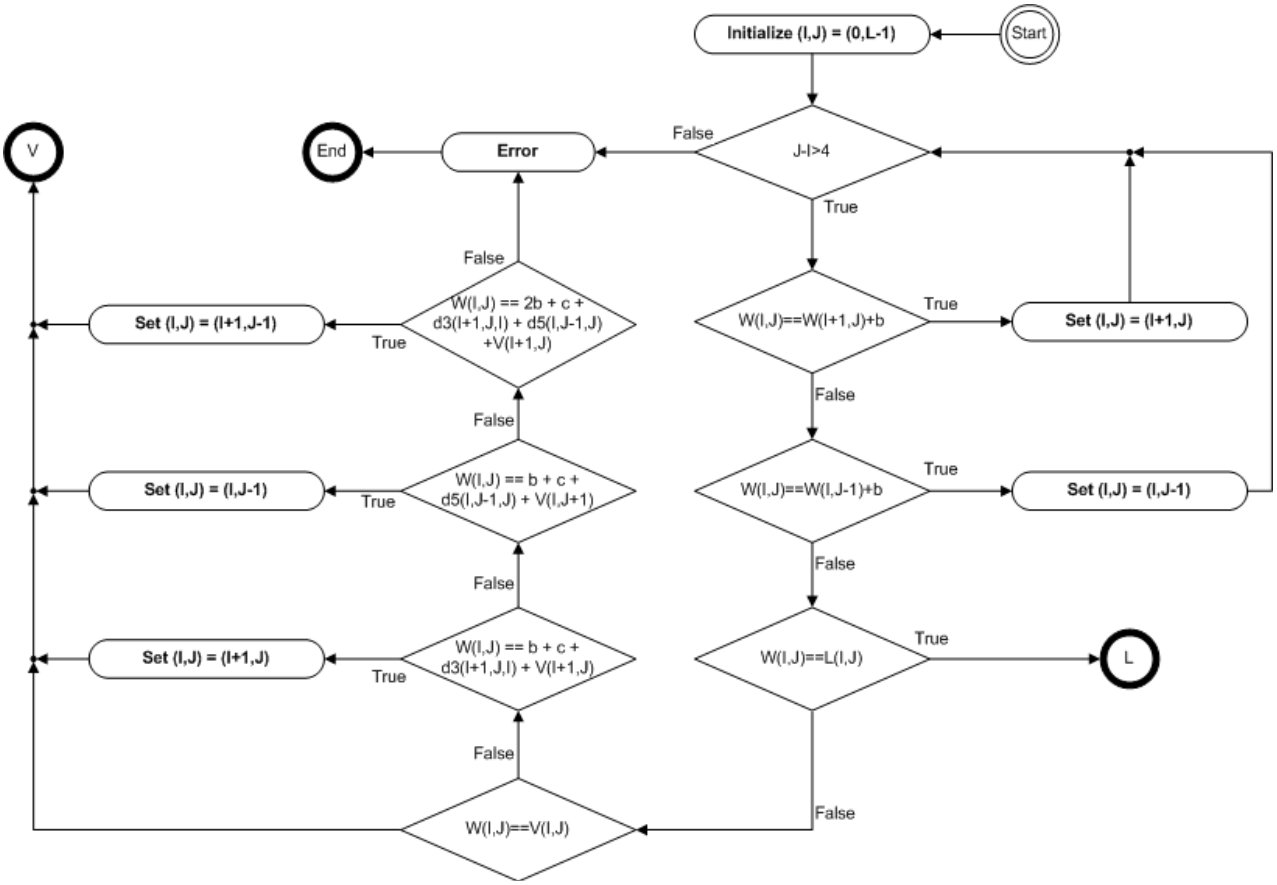


Figure 32 - Αλγόριθμος οπισθοδρόμησης για τον πίνακα W

Σε περίπτωση που βρεθεί ο σωστός κανόνας ο αλγόριθμος ανανεώνει τις τιμές για τις μεταβλητές I και J και προχωράει στον αντίστοιχο πίνακα. Οι τερματικές καταστάσεις L και V ορίζουν αυτήν ακριβώς την μετάβαση σε κάποιο άλλο τμήμα του αλγορίθμου που είναι υπεύθυνο για την εκτέλεση της οπισθοδρόμησης στον πίνακα αυτό.

Κατά την οπισθοδρόμηση στον πίνακα W δεν ανιχνεύεται κανένα ζεύγος, καθώς οι κανόνες είτε εισαγάγουν κάποια ελεύθερη βάση σε κάποια προηγούμενη λύση είτε λαμβάνουν την τιμή που έχει υπολογιστεί στους δύο άλλους πίνακες.

Επιπλέον λαμβάνονται υπόψη και δύο περιπτώσεις λάθους η περίπτωση ο αλγόριθμος να φτάσει σε κάποια υπό-ακολουθία η οποία δεν μπορεί να έχει κάποια δομή, περίπτωση $J - I \leq 4$ και την περίπτωση να μην βρεθεί καμία δυνατή περίπτωση για την λήψη την τιμής $W(I, J)$.

Οπισθοδρόμηση στον πίνακα L

Σε αυτό το τμήμα του αλγορίθμου έχουμε ήδη αρχικοποιημένες τις μεταβλητές I και J, καθώς πρέπει να έχει προηγηθεί κάποιο άλλο στάδιο για να χρειαστεί να γίνει οπισθοδρόμηση στον πίνακα L και λόγω του ότι υπάρχουν περισσότερα από ένα σημεία στον αλγόριθμο από τα οποία μπορούμε να εκκινήσουμε την διαδικασία αυτή, την ανάθεση των μεταβλητών στις σωστές τιμές την αναλαμβάνει το προηγούμενο στάδιο, όποιο και αν είναι αυτό.

Για την εκτέλεση της οπισθοδρόμησης ο αλγόριθμος θα ελέγξει και πάλι όλους τους πιθανούς συνδυασμούς οι οποίοι μπορούν να αποτελέσουν την δομή που περιγράφει το στοιχείο $L(I, J)$, και στην συνέχεια την ανάθεση της οπισθοδρόμησης στο αντίστοιχο τμήμα.

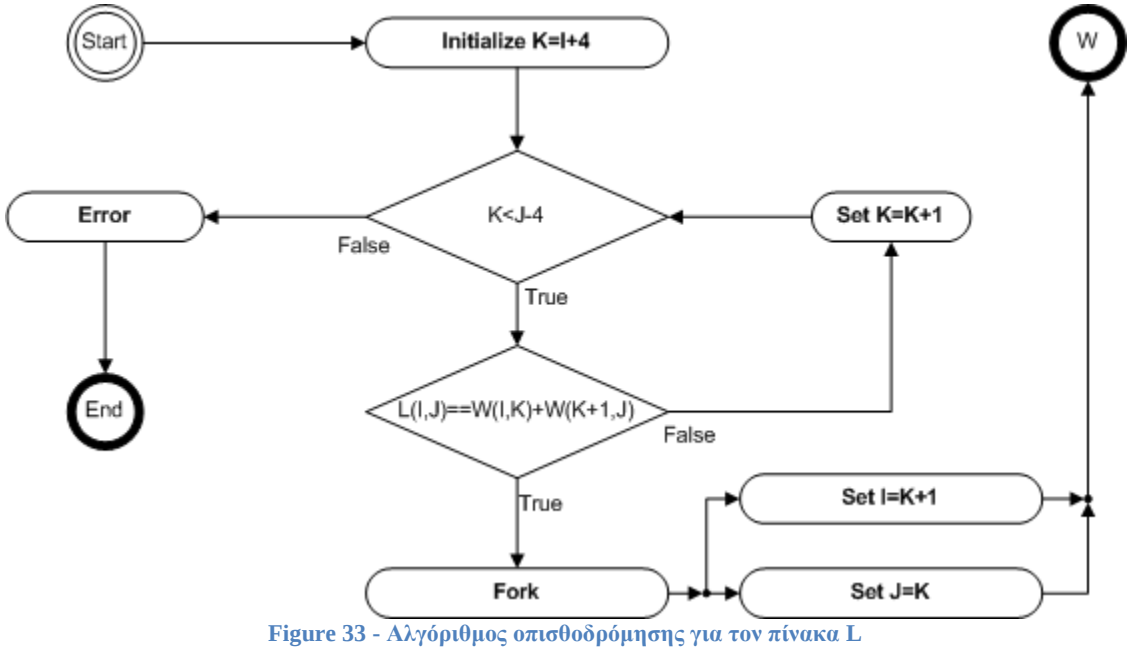


Figure 33 - Αλγόριθμος οπισθοδρόμησης για τον πίνακα L

Ο αλγόριθμος ξεκινά μια επαναληπτική διαδικασία για να ελέγξει όλους τους πιθανούς συνδυασμούς υπό-προβλημάτων. Όταν βρεθεί ο συνδυασμός ο οποίος χρησιμοποιήθηκε για τον υπολογισμό του στοιχείου $L(I, J)$ ο αλγόριθμος εκτελεί αναδρομικά οπισθοδρόμηση και για τα δύο υπό-προβλήματα.

Σε περίπτωση που κανένας συνδυασμός δεν βρεθεί τότε ο αλγόριθμος τερματίζει με σφάλμα καθώς η μόνη περίπτωση τερματισμού είναι η ανίχνευση κάποιου Hairpin Loop.

Κατά την οπισθοδρόμηση στον πίνακα L δεν ανιχνεύεται κανένα ζεύγος, καθώς οι κανόνες αποτελούν συνένωση σε σειρά δύο ανεξάρτητων υπό-προβλημάτων.

Οπισθοδρόμηση στον πίνακα V

Το τελευταίο τμήμα του αλγορίθμου αποτελεί την οπισθοδρόμηση στον πίνακα V και αποτελεί το μοναδικό τμήμα στο οποίο ανιχνεύονται τα ζεύγη. Η προσθήκη του ζεύγους στην τελική λύση γίνεται με την έναρξη του αλγορίθμου καθώς γνωρίζουμε ότι τα στοιχεία (I, J) θα σχηματίζουν ζεύγος. Επιπλέον το κομμάτι αυτό περιέχει τον επιτυχή τερματισμό του αλγορίθμου, σε περίπτωση που ανιχνευθεί ένα Hairpin Loop καθώς τότε και μόνο τότε έχουμε φτάσει στο άκρο της δευτεροταγής δομής του RNA.

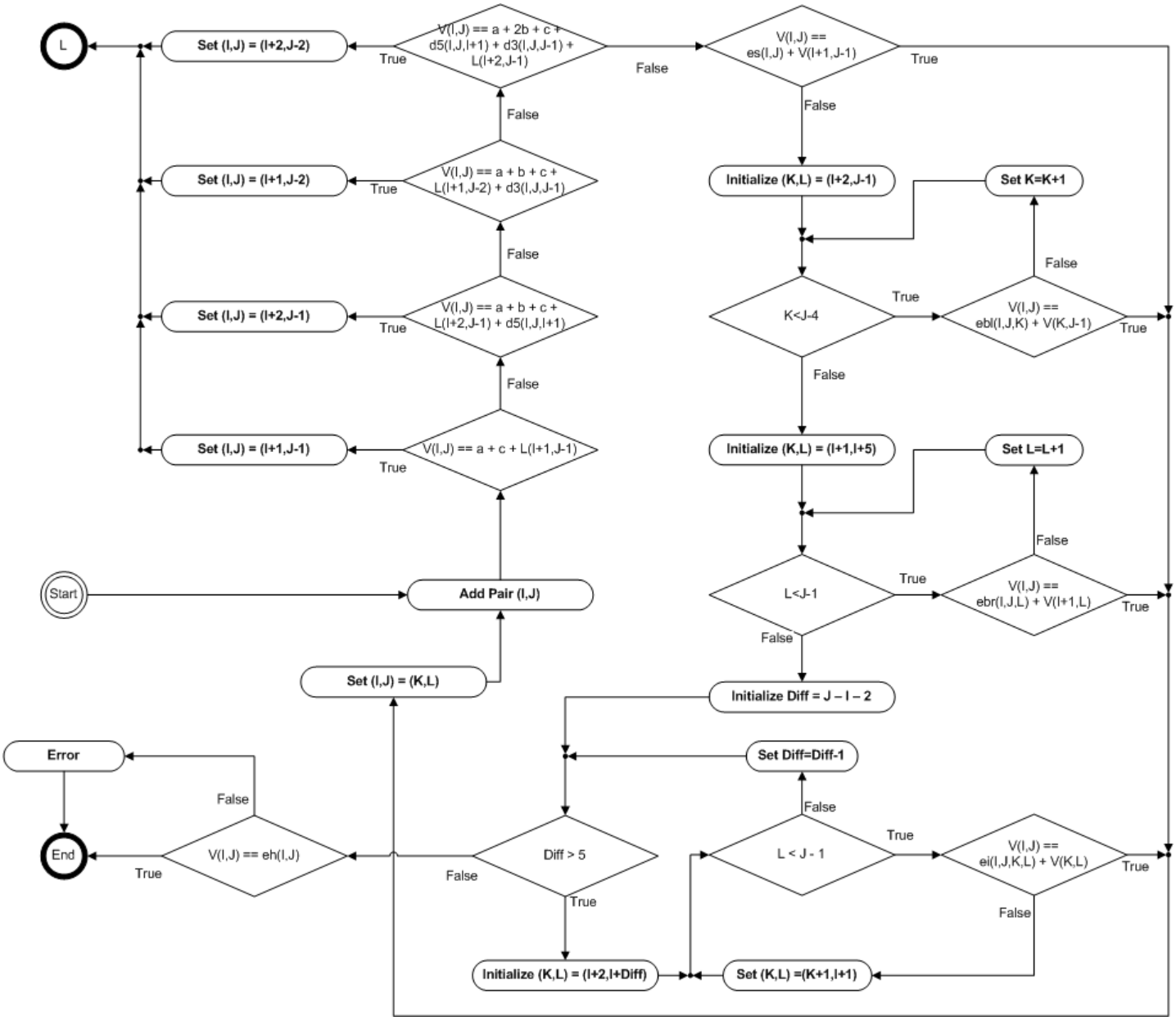


Figure 34 - Αλγόριθμος οπισθοδρόμησης για τον πίνακα V

Αντίστοιχα και με τα προηγούμενα τμήματα ο αλγόριθμος ελέγχει όλες τις δυνατές περιπτώσεις και αναθέτει την εκτέλεση της οπισθοδρόμησης στο κατάλληλο τμήμα ενώ τερματίζει με σφάλμα αν καμία από τις περιπτώσεις δεν ταιριάζει με την δομή που περιγράφει η τιμή $V(I, J)$

Οπισθοδρόμηση σε όλους τους πίνακες

Στην συνέχεια ακολουθεί ο συνολικός αλγόριθμος οπισθοδρόμησης, ο οποίος αποτελείται από συνένωση των τριών τμημάτων που αναλυθήκαν προηγουμένως με κατάλληλες συνδέσεις μεταξύ τους έτσι ώστε να γίνεται σωστά η μεταφορά του ελέγχου.

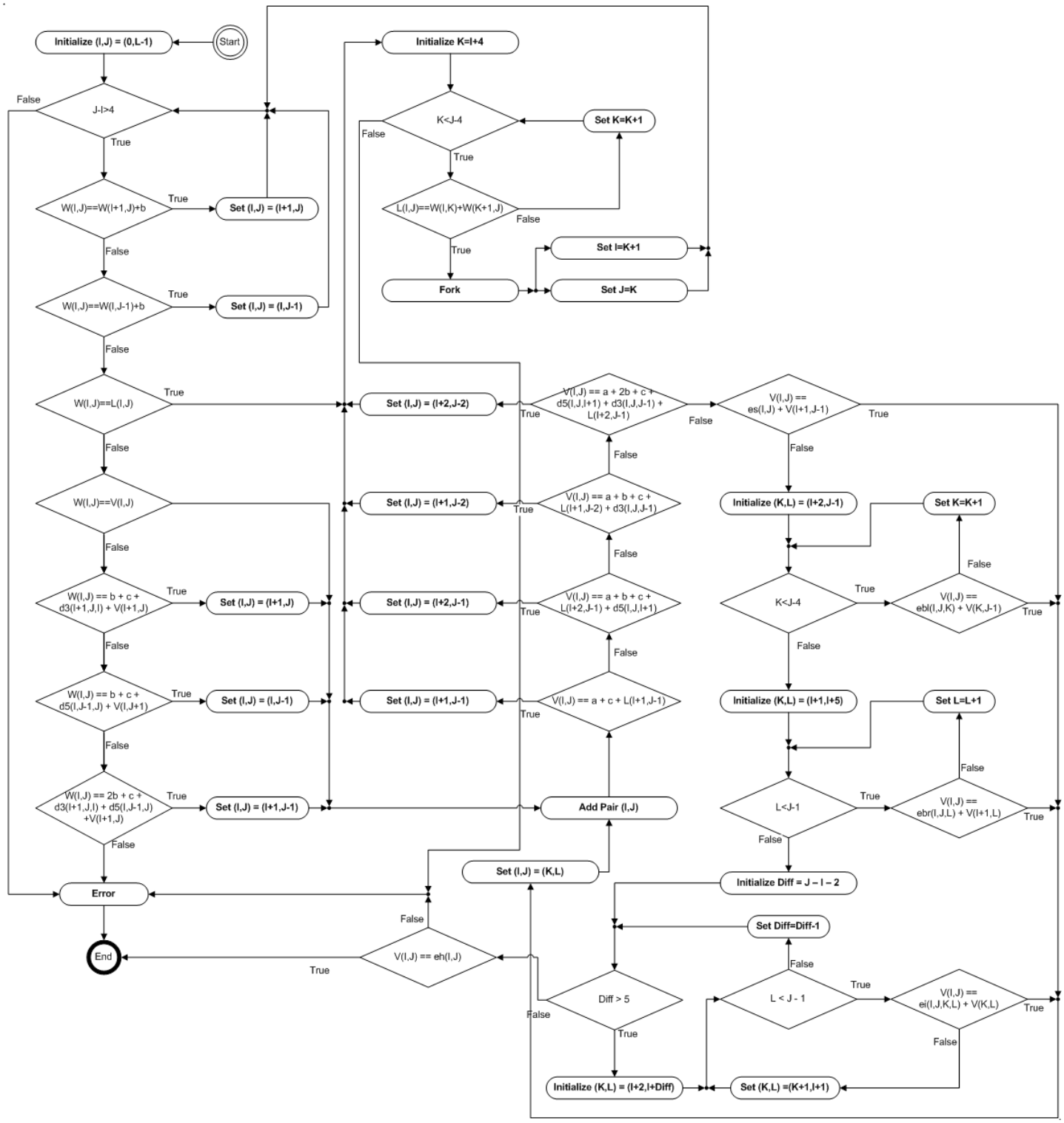


Figure 35 - Αλγόριθμος οπισθοδρόμησης για όλους τους πίνακες

ΑΡΧΙΤΕΚΤΟΝΙΚΗ

Τύποι δεδομένων

Κατά την σχεδίαση του συστήματος χρησιμοποιήθηκαν οι ακόλουθοι τύποι δεδομένων.

Base

Τα δεδομένα αυτού του τύπου μπορούν να λάβουν τιμές από το σύνολο τιμών

$$\{A, C, G, U\}$$

και χρησιμοποιούνται για να αναπαραστήσουν τις βάσεις που αποτελούν το μακρομόριο του RNA.

Pair

Τα δεδομένα αυτού του τύπου χρησιμοποιούνται για να αναπαραστήσουν τα πιθανά ζεύγη που μπορούν να δημιουργηθούν μεταξύ των βάσεων στο μακρομόριο του RNA και λαμβάνουν τιμές από το σύνολο τιμών

$$\{AU, CG, GC, UA. GC, UG, MISMATCH\}$$

Με την τιμή $MISMATCH$ να αντιστοιχεί στην περίπτωση όπου δεν μπορεί να δημιουργηθεί κάποιο ζεύγος.

Energy

Τα δεδομένα αυτού του τύπου χρησιμοποιούνται για να αναπαραστήσουν τις τιμές των πειραματικών δεδομένων. Οι τιμές αυτές είναι δεκαδικές με ένα δεκαδικό ψηφίο και ανήκουν εύρος τιμών

$$[-3.4,7.4]$$

Επειδή όμως οι τιμές αυτές ενδέχεται να αλλάξουν για τον υπολογισμό της δευτεροταγής δομής σε διαφορετικές θερμοκρασίες και περιβάλλοντα χώρο το εύρος αυτό θα επεκταθεί σε

$$[-25.6,25.5]$$

Με τις τιμές -25.6 και 25.5 να αντιστοιχούν στο αρνητικό και θετικό άπειρο αντίστοιχα.

LEnergy

Τα δεδομένα αυτού του τύπου χρησιμοποιούνται για να αναπαραστήσουν τις τιμές που μπορεί να λάβει η ενέργεια ενός K-Loop. Μιας και στην χειρότερη περίπτωση η ενέργεια αυτή θα προκύψει από το άθροισμα τεσσάρων τιμών τύπου Energies το εύρος τιμών θα πρέπει να είναι ίσο με

$$[4 \times -25.6,4 \times 25.5] = [-102.4,102]$$

Indx
Τα δεδομένα αυτού του τύπου χρησιμοποιούνται για να αριθμηθούν οι βάσεις που αποτελούν το μακρομόριο του RNA.

Η αρίθμηση ξεκινά από το μηδέν το οποίο αντιστοιχεί στο πρώτο στοιχείο στην ακολουθία των βάσεων και αυξάνεται κατά ένα για να αρίθμηση το επόμενο στοιχείο. Έτσι για Indx ίσο με δέκα θα αντιστοιχεί στην εντεκάτη βάση.

Το εύρος τιμών των δεδομένων αυτών ανήκει στο

$$[0, L)$$

Όπου L ο συνολικός αριθμός των βάσεων.

Data
Τα δεδομένα αυτού του τύπου χρησιμοποιούνται για να αναπαραστήσουν την ενέργεια κάποιας υπό-ακολουθίας του RNA. Το εύρος τιμών των δεδομένων αυτού του τύπου εξαρτάται τόσο από το εύρος των τιμών για τα δεδομένα τύπου Energy όσο και από τον αριθμό των βάσεων.

Στην χειρότερη περίπτωση μπορούν να σχηματιστούν $\frac{L-3}{2}$ K-Loops, όπου στην χειρότερη περίπτωση, κάθε ένα από αυτά θα έχει τιμή είτε -25.5 είτε 25.4.

Έτσι με μια χονδρική εκτίμηση του εύρους τιμών λαμβάνουμε το εξής

$$[(-12.7 \cdot L + 76.5), (12.7 \cdot L - 76.3)]$$

Όπου L ο συνολικός αριθμός των βάσεων.

Addr
Τέλος τα δεδομένα αυτού του τύπου χρησιμοποιούνται για να αναπαραστήσουν την θέση μνήμης στην οποία θα αποθηκευτούν τα δεδομένα των πινάκων $W(\cdot,\cdot), V(\cdot,\cdot), L(\cdot,\cdot)$.

Εν γένει ο αριθμός αυτός εξαρτάται από το μέγεθος της μνήμης η οποία είναι διαθέσιμη, μπορούμε όμως να γίνει μια εκτίμηση για τον χώρο που θα χρειαστεί η αποθήκευση των υπό-προβλημάτων.

Κάθε πίνακας αποτελείτε από L^2 στοιχεία και επομένως χρειαζόμαστε να αριθμήσουμε $3 \cdot L^2$ διευθύνσεις, έτσι το εύρος τιμών των δεδομένων αυτών ανήκει στο

$$[0, 3 \cdot L^2)$$

ctype
Επιπλέον χρησιμοποιήθηκε και ένας τύπος δεδομένων για την μεταφορά δεδομένων ελέγχου.

Γενικές κατευθύνσεις αρχιτεκτονικής

Κύριο χαρακτηριστικό του αλγορίθμου είναι ο υψηλός αριθμός προσβάσεων στην μνήμη. Σύμφωνα με τα στοιχεία από τον πίνακα 8, σελίδα 51, μπορούμε να υπολογίσουμε τον συνολικό αριθμό προσβάσεων στην μνήμη αθροίζοντας των αριθμό των στοιχείων μνήμης που χρειάζεται για τον υπολογισμό κάθε στοιχείου για όλα τα στοιχεία.

Ο συνολικός αριθμός δίνεται από τον παρακάτω τύπο

$$MAC = \sum_{Diff=0}^{L-1} AccessCount(Diff) \cdot ProblemCount(Diff)$$

Όπου τα στοιχεία που απαρτίζουν την παραπάνω συνάρτηση αναλύονται στον παρακάτω πίνακα

Στοιχείο συνάρτησης	Περιγραφή	Τιμή
MAC	Ο αριθμός των συνολικών προσβάσεων στην μνήμη για τον υπολογισμό όλων των υπό-προβλημάτων.	-
L	Ο συνολικός αριθμός στοιχείων στο RNA.	Η τιμή αυτή είναι σταθερή και ίση με τον αριθμό των στοιχείων στο RNA.
$Diff$	Η διαφορά $j - i$	Η τιμή αυτή λαμβάνει όλες της τιμές στο διάστημα $[0, L)$ έτσι ώστε να αθροιστούν οι τιμές για όλα μεγέθη υποπροβλημάτων.
$AccessCount(Diff)$	Το άθροισμα των συνολικών αριθμών προσβάσεων στην μνήμη των τριών πινάκων για τα στοιχεία μεγέθους $Diff$.	Η τιμή της συνάρτησης αυτής υπολογίζεται από τον πίνακα 8, σελίδα 51
$ProblemCount(Diff)$	Ο αριθμός των υπό-προβλημάτων μεγέθους $Diff$	Η τιμή της συνάρτησης αυτής είναι ίση με $L - Diff$

Έτσι ο συνολικός αριθμός προσβάσεων στην μνήμη παίρνει την παρακάτω μορφή

$$MAC = \sum_{Diff=9}^{L-1} \left[(L - Diff) \cdot \left(\frac{(Diff - 5) \cdot (Diff - 4)}{2} + 4 + 7 + 2 \cdot (Diff - 8) \right) \right] + [2 \cdot (L - 4) + 6 \cdot (L - 5) + 9 \cdot (L - 6) + 13 \cdot (L - 7) + 17 \cdot (L - 8)]$$

Απλοποιώντας την παραπάνω σχέση λαμβάνουμε

$$MAC = 47 \cdot L - 319 - \frac{1}{2}\sum_{Diff=9}^{L-1}(Diff - L)\cdot(Diff - 5)\cdot(Diff - 4) + \sum_{Diff=9}^{L-1}(Diff - L)\cdot(2\cdot Diff - 5) \Leftrightarrow$$

$$MAC = 47 \cdot L - 319 - \frac{1}{2}\left[-\frac{1}{12}\cdot L^4 + \frac{18}{12}\cdot L^3 - \frac{119}{12}\cdot L^2 + \frac{582}{12}\cdot L - \frac{2160}{12}\right] + \left[\frac{2}{6}\cdot L^3 - \frac{15}{6}\cdot L^2 - \frac{179}{6}\cdot L + \frac{1368}{6}\right] \Leftrightarrow$$

$$MAC = \frac{1128}{24}\cdot L - \frac{7656}{24} + \frac{1}{24}\cdot L^4 - \frac{18}{24}\cdot L^3 + \frac{119}{24}\cdot L^2 - \frac{582}{24}\cdot L + \frac{2160}{24} + \frac{8}{24}\cdot L^3 - \frac{60}{24}\cdot L^2 - \frac{716}{24}\cdot L + \frac{5472}{24} \Leftrightarrow$$

$$MAC = \frac{1}{24}\cdot L^4 - \frac{10}{24}\cdot L^3 + \frac{59}{24}\cdot L^2 - \frac{170}{24}\cdot L - \frac{24}{24}$$

Οπότε ο αριθμός των προσβάσεων στην μνήμη για τον υπολογισμό των πινάκων του αλγορίθμου Zucker για RNA μεγέθους L, $L > 8$ βάσεων δίνεται από τον τύπο

$$\boxed{MAC = \frac{L^4 - 10\cdot L^3 + 59\cdot L^2 - 170\cdot L - 24}{24}}$$

Η παραπάνω εξίσωση επιβεβαιώνεται και πρακτικά, μετρώντας των αριθμό των στοιχείων τα οποία διαβάζονται από την μνήμη κατά την διάρκεια του αλγορίθμου.

Λόγο του μικρού χρόνου επεξεργασίας για κάθε στοιχείο μνήμης* το MAC αποτελεί μια πολύ καλή εκτίμηση για τον χρόνο εκτέλεσης.

Για κάθε στοιχείο εκτελούνται οι παρακάτω πράξεις

1. *Μια σύγκριση μεταξύ των δεικτών (k,l) για το στοιχείο που διαβάστηκε από την μνήμη και των δεικτών (i,j) για το στοιχείο που υπολογίζεται, για να εξαχθεί ο τύπος του υπολογισμού.*
2. *Μια σειρά από προσθέσεις, από καμία έως και τέσσερις.*
3. *Μια σύγκριση για την ανίχνευση της ελάχιστης τιμής.*

Στον παρακάτω πίνακα παρουσιάζονται διάφορες τιμές για το MAC σε σχέση με το μέγεθος του RNA.

L	MAC
100	3.773.874
200	63.430.249
300	326.469.124
400	1.040.390.499
500	2.552.694.374
600	5.310.880.749
700	9.862.449.624
800	16.854.900.999
900	27.035.734.874
1.000	41.252.451.249
2.000	663.343.152.499
3.000	3.363.772.103.749
4.000	10.640.039.304.999
5.000	25.989.644.756.249
6.000	53.910.088.457.499
7.000	99.898.870.408.749
8.000	170.453.490.609.999
9.000	273.071.449.061.249
10.000	416.250.245.762.499

Table 9 - MAC για διάφορα μεγέθη RNA

Όπως γίνεται εμφανές και από τα παραπάνω στοιχεία, για την βελτίωση της απόδοσης του αλγορίθμου πρέπει να γίνει μια αποδοτική διαχείριση των στοιχείων στην μνήμη.

Από την φύση του αλγορίθμου, μια προσπάθεια για την τοπική αποθήκευση τιμών με σκοπό την επαναχρησιμοποίηση τους σε μετέπειτα υπολογισμούς (χρονική τοπικότητα), δεν έχει θετικά αποτελέσματα μιας και ο αριθμός των στοιχείων που απαιτούνται για κάθε υπολογισμό είναι πολύ μεγάλος, ενώ παράλληλα οι εξαρτήσεις μεταξύ των δεδομένων αποτρέπουν τον υπολογισμό τους με τέτοιο τρόπο έτσι ώστε να επιτρέπεται η επαναχρησιμοποίηση τους. Έτσι απαιτείται μια απαγορευτικά μεγάλη αποθήκευση τιμών, τάξης συγκρίσιμη με εκείνης που χρειάζεται για την αποθήκευση των πινάκων.

Αντίθετα τα στοιχεία που διαβάζονται από την μνήμη βρίσκονται «κοντά» (χωρική τοπικότητα) επιτρέποντας την μαζική ανάγνωσή τους. Επιπλέον το σύνολο των στοιχείων είναι γνωστό εκ τον προτέρων επιτρέποντας έτσι την δημιουργία μιας στρατηγικής ανάγνωσης η οποία θα μειώσει τον αριθμό των προσβάσεων στην μνήμη.

Έτσι οι κατευθυντήριες γραμμές για την επιτάχυνση του αλγορίθμου, στα πλαίσια της εργασίας αυτής, ήταν η μείωση του MAC με την επαναχρησιμοποίηση των στοιχείων της μνήμης περισσότερες από μία φορές και η αύξηση της κατανάλωσης των τιμών αυτών σε ρυθμό ίσο με τον ρυθμό ανάγνωσης.

Μείωση συνολικού MAC

Η μείωση αυτή μπορεί να επιτευχθεί επαναχρησιμοποιώντας κάθε στοιχείο μνήμης για περισσότερους από έναν υπολογισμούς. Επειδή όμως η αποθήκευσή του σε κάποια κρυφή μνήμη με σκοπό την χρήση του αργότερα δεν είναι αποδοτική, έγινε μια προσπάθεια για την δημιουργία των προϋποθέσεων οι οποίες θα επέτρεπαν την άμεση χρήση του στοιχείου πολλές φορές χωρίς την ενδιάμεση αποθήκευσή του.

Κύρια ευθύνη για τον υψηλό αριθμό προσβάσεων στην μνήμη έχει ο υπολογισμός του στοιχείου V. Για δύο διπλανά στοιχεία στην ίδια διαγώνιο η πλειοψηφία των στοιχείων που χρειάζονται είναι κοινά.

Στο παρακάτω σχήμα φαίνονται τέσσερα γειτονικά τετράγωνα ως προς την διαγώνιο και ο αριθμός των «χρήσεων» κάθε στοιχείου.

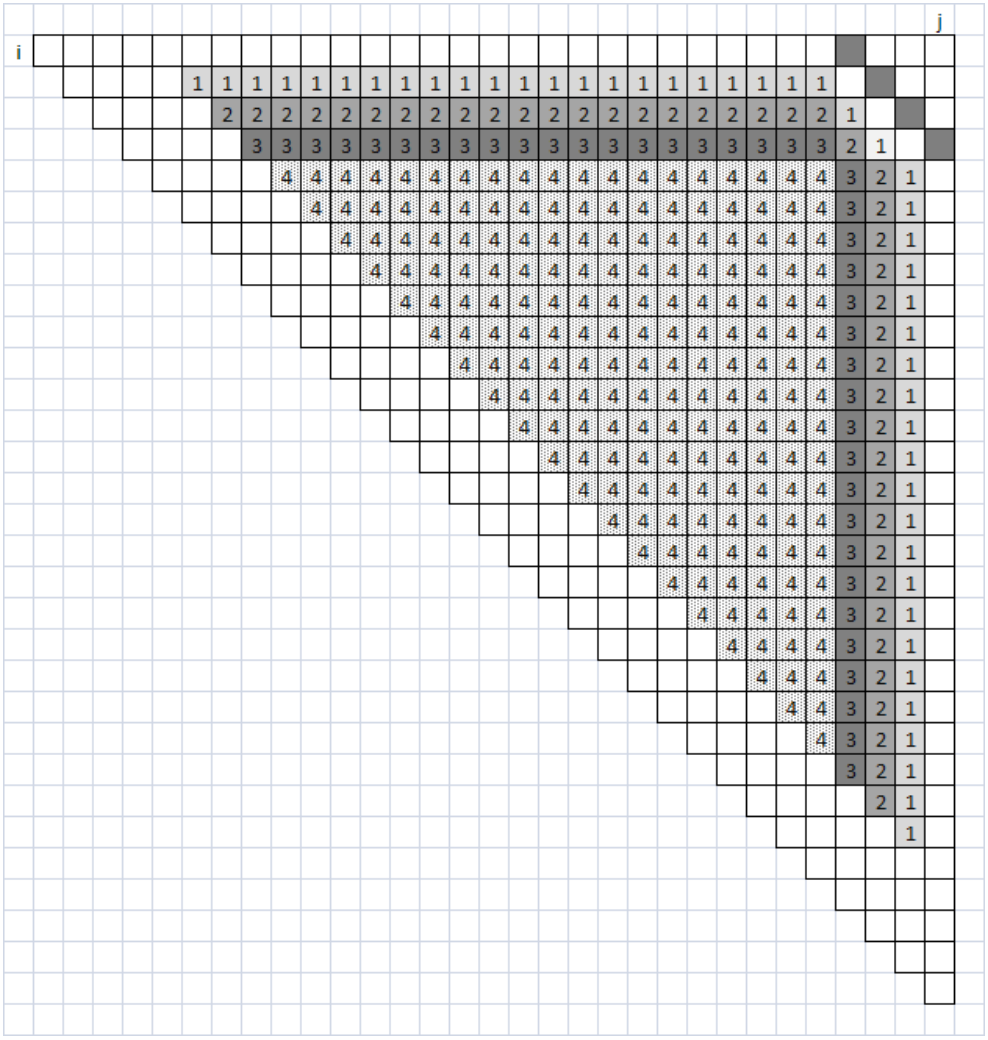


Figure 36 - Κοινά στοιχεία μεταξύ γειτονικών υπολογισμών

Όπως φαίνεται και από το παραπάνω σχήμα αν υπολογίσουμε τα στοιχεία αυτά ταυτόχρονα αποφεύγεται ένας σημαντικός αριθμός προσβάσεων στην μνήμη.

Μείωση συνολικού MAC – Πίνακας V

Αντίστοιχα με την περίπτωση όπου δεν υπήρχε κάποιος παράλληλος υπολογισμός τιμών, παρουσιάζονται στο παρακάτω σχήμα τόσο οι εξαρτήσεις του υπολογισμού όσο και πόσες φορές χρησιμοποιείται κάθε στοιχείο.

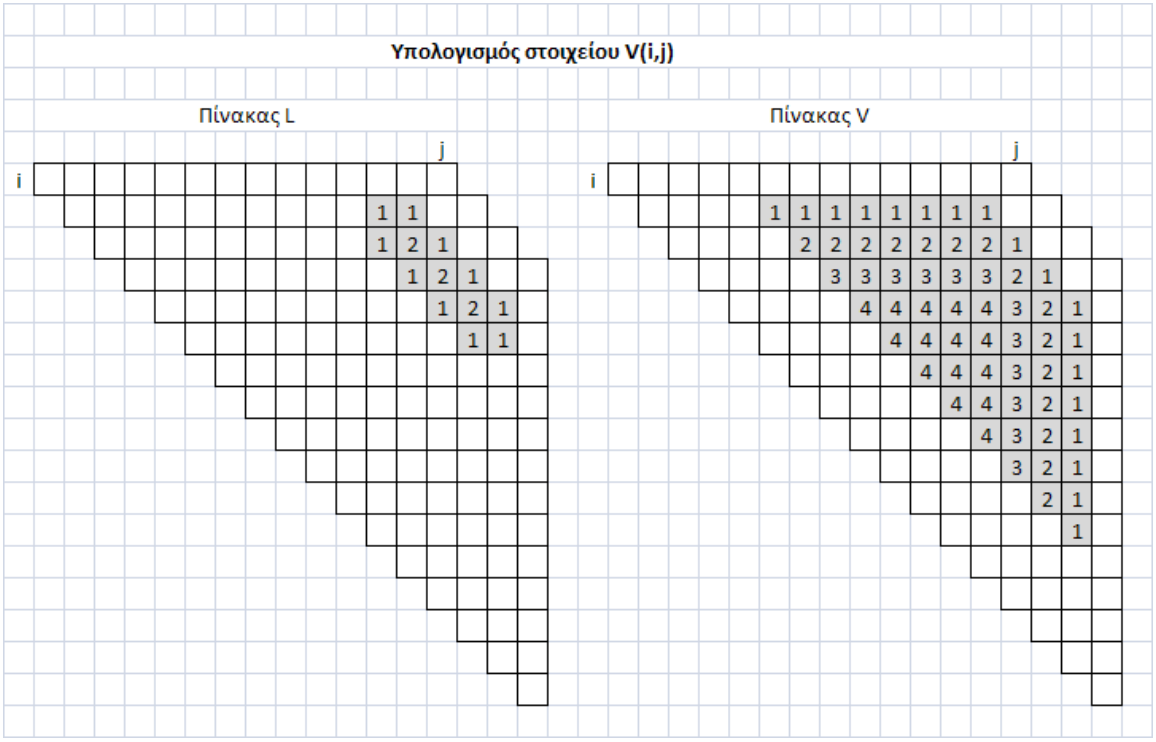


Figure 37 - Εξαρτήσεις για παράλληλο υπολογισμό στον πίνακα V

Στην συνέχεια παρουσιάζεται το MAC/C για τον υπολογισμό ενός στοιχείου, σε σχέση με την διαφορά $j-i$ και για διαφορετικό αριθμό παράλληλων υπολογισμών C.

$j-i$	$MAC/1$	Δύο παράλληλοι υπολογισμοί.		Τέσσερις παράλληλοι υπολογισμοί		Οκτώ παράλληλοι υπολογισμοί	
		$MAC/2$	$\frac{MAC/2}{MAC/1}$	$MAC/4$	$\frac{MAC/4}{MAC/1}$	$MAC/8$	$\frac{MAC/8}{MAC/1}$
100	4.564	2.331	0,511	1.214	0,266	656	0,144
200	19.114	9.656	0,505	4.927	0,258	2.562	0,134
300	43.664	21.981	0,503	11.139	0,255	5.718	0,131
400	78.214	39.306	0,502	19.852	0,254	10.125	0,129
500	122.764	61.631	0,502	31.064	0,253	15.781	0,128
600	177.314	88.956	0,502	44.777	0,252	22.687	0,128
700	241.864	121.281	0,501	60.989	0,252	30.843	0,127
800	316.414	158.606	0,501	79.702	0,252	40.250	0,127
900	400.964	200.931	0,501	100.914	0,251	50.906	0,127
1.000	495.514	248.256	0,501	124.627	0,251	62.812	0,127

Table 10 - Μέσος αριθμός στοιχείων για παράλληλο υπολογισμό στον πίνακα V

Τα αποτελέσματα έχουν διαιρεθεί με τον αριθμό των παράλληλων υπολογισμών για να εξαχθεί η μέση τιμή των προσβάσεων στην μνήμη ανά στοιχείο.

Μείωση συνολικού MAC – Πίνακας W

Στην συνέχεια παρουσιάζονται οι εξαρτήσεις του υπολογισμού και ο αριθμός των χρήσεων κάθε στοιχείου για τον υπολογισμό στον πίνακα W.

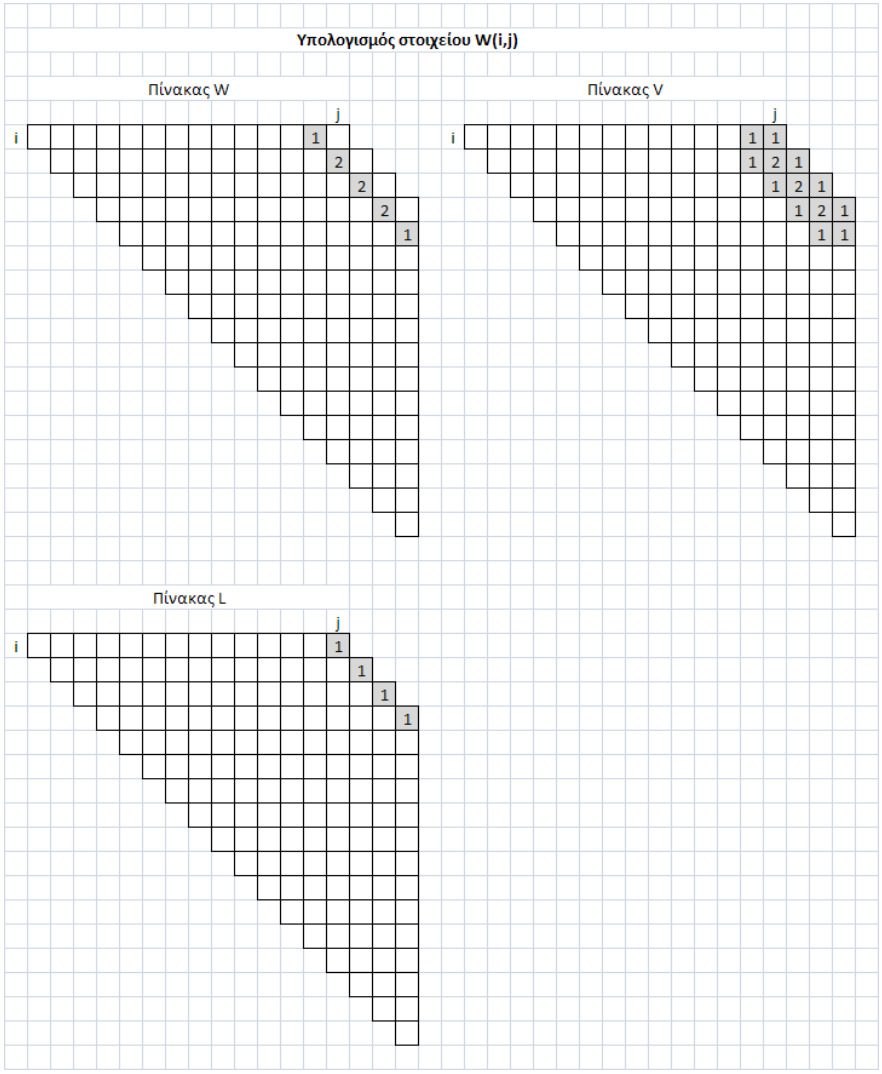


Figure 38 - Εξαρτήσεις για παράλληλο υπολογισμό στον πίνακα W

Στην συνέχεια παρουσιάζεται το MAC/C για τον υπολογισμό ενός στοιχείου, σε σχέση με την διαφορά $j-i$ και για διαφορετικό αριθμό παράλληλων υπολογισμών C.

$j-i$	$MAC/1$	Δύο παράλληλοι υπολογισμοί.		Τέσσερις παράλληλοι υπολογισμοί		Οκτώ παράλληλοι υπολογισμοί	
		$MAC/2$	$\frac{MAC/2}{MAC/1}$	$MAC/4$	$\frac{MAC/4}{MAC/1}$	$MAC/8$	$\frac{MAC/8}{MAC/1}$
4	2	2	1,0	2	1,0	2	1,0
5	6	5	0,833	4	0,75	4	0,708
6	7	6	0,857	5	0,786	5	0,75
100	7	6	0,857	5	0,786	5	0,75
1000	7	6	0,857	5	0,786	5	0,75

Table 11 - Μέσος αριθμός στοιχείων για παράλληλο υπολογισμό στον πίνακα W

Μείωση συνολικού MAC – Πίνακας L

Τέλος παρουσιάζονται οι εξαρτήσεις του υπολογισμού και ο αριθμός των χρήσεων κάθε στοιχείου για τον υπολογισμό στον πίνακα W.

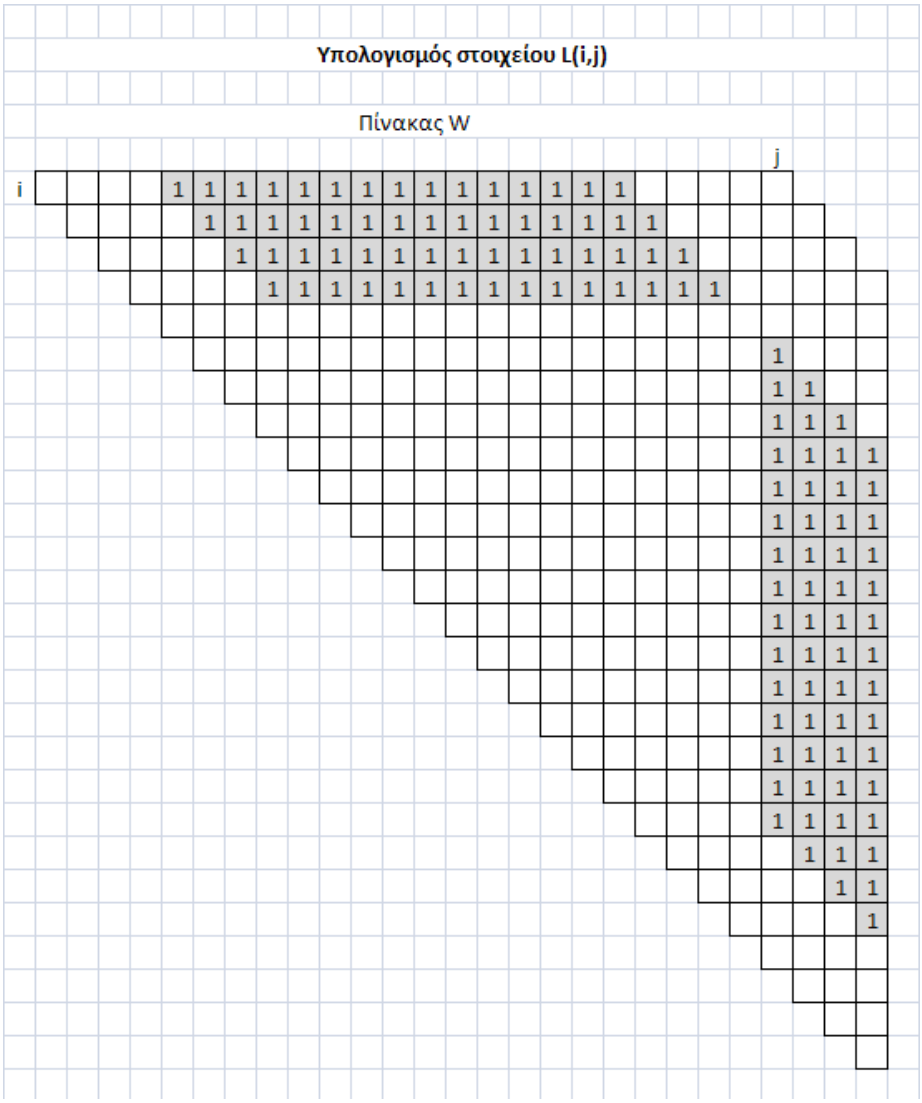


Figure 39 - Εξαρτήσεις για παράλληλο υπολογισμό στον πίνακα L

Όπως γίνεται φανερό και από το παραπάνω σχήμα ο παράλληλος υπολογισμός δεν έχει κάποια επίδραση στην μείωση του MAC. Θα χρησιμοποιηθεί όμως μιας και αυξάνει την χωρική τοπικότητα.

Βελτίωση κατανάλωσης τιμών

Η μείωση των συνολικών προσβάσεων στην μνήμη, ένα αδιαμφισβήτητα θετικό αποτέλεσμα, δεν έχει αντίστοιχη συνεισφορά στην μείωση του συνολικού χρόνου εκτέλεσης καθώς ταυτόχρονα αυξάνει τις απαιτήσεις των υπολογισμών. Πράγματι, χωρίς τον παράλληλο υπολογισμό στοιχείων το σύστημα επεξεργασίας θα έπρεπε να διαχειρίζεται ένα στοιχείο κάθε στιγμή, τώρα θα πρέπει να είναι σε θέση να αυξήσει τον ρυθμό επεξεργασίας τόσες φορές όσες και ο αριθμός των παράλληλων υπολογισμών.

Για κάθε ένα από τα στοιχεία που διαβάζονται από την μνήμη χρειάζονται οι παρακάτω πράξεις.

Ενέργεια	Περιγραφή
Υπολογισμός του τύπου του υπολογισμού.	Το βήμα αυτό απαιτεί μια σύγκριση μεταξύ των δεικτών i, j που αντιστοιχούν στο στοιχείο που υπολογίζεται και των δεικτών k, l που αντιστοιχούν στο στοιχείο που διαβάστηκε από την μνήμη.
Μετάφραση των δεικτών σε βάσεις.	Στο βήμα αυτό οι αριθμητικές τιμές για τους δείκτες μεταφράζονται στις τιμές των βάσεων τις οποίες αντιπροσωπεύουν. Συνολικά γίνονται οκτώ μεταφράσεις για τα στοιχεία $i, i+1, k-1, k, l, l+1, j-1, j$ Κάθε μετάφραση ισοδυναμεί με την ανάγνωση της τιμής της βάσης από την μνήμη που βρίσκεται αποθηκευμένη η ακολουθία του RNA.
Ανάγνωση των πειραματικών τιμών.	Οι τιμές των βάσεων που λήφθηκαν από το προηγούμενο βήμα χρησιμοποιούνται για την ανάγνωση των επιμέρους πειραματικών δεδομένων. Οι τιμές διαβάζονται από την μνήμη στην οποία βρίσκονται αποθηκευμένες οι τιμές αυτές.
Εξαγωγή αποτελέσματος.	Τέλος υπολογίζεται το αποτέλεσμα με την πρόσθεση των τιμών των πειραματικών δεδομένων στην τιμή που αντιστοιχεί στο συγκεκριμένο υπό-πρόβλημα.

Εξαίρεση στον παραπάνω κανόνα αποτελεί ο υπολογισμός για το Hairpin loop μιας και δεν συναθροίζεται σε αυτό η τιμή κάποιου προηγούμενου υπό-προβλήματος και ο υπολογισμός των στοιχείων για τον πίνακα L καθώς ο υπολογισμός αυτός απαιτεί την άθροιση μεταξύ δύο στοιχείων από την μνήμη.

Από τις παραπάνω ενέργειες τόσο ο υπολογισμός του τύπου του υπολογισμού όσο και η εξαγωγή του αποτελέσματος δεν αποτελούν κάποιο πρόβλημα καθώς οι πράξεις αυτές μπορούν να γίνουν παράλληλα. Αντίθετα οι μετάφραση των δεικτών και η ανάγνωση των πειραματικών αποτελεσμάτων, επειδή απαιτεί την ανάγνωση τιμών από την μνήμη, πρέπει να γίνουν σειριακά για κάθε ένα από τα στοιχεία.

Για την αντιμετώπιση του προβλήματος αυτού δημιουργήθηκε ένα Repository για τα πειραματικά δεδομένα το οποίο περιέχει πολλαπλά αντίγραφα των τιμών αυτών όσο και της ακολουθίας του RNA σε μικρές τοπικές μνήμες. Έτσι όχι μόνο γίνεται δυνατή η παράλληλη επεξεργασία των δεδομένων αλλά και αποφεύγονται οι προσβάσεις στην κύρια μνήμη για την ανάγνωση τους. Στην συνέχεια θα αναλυθεί η δομή του Repository.

Επίπεδα υπολογισμού

Ο διαχωρισμός του υπολογισμού των επιμέρους λύσεων αναδεικνύει την ανάγκη για την τμηματοποίηση του αλγορίθμου σε μικρότερα υποσυστήματα, έτσι ώστε να απλοποιηθεί η διαδικασία σχεδίασης και υλοποίησης. Έχοντας ήδη διαχωρίσει Repository για τα πειραματικά δεδομένα και το RNA (EDRR), ως υπεύθυνο για τον υπολογισμό των επιμέρους λύσεων, ο περεταίρω διαχωρισμός έγινε με τέτοιο τρόπο έτσι ώστε κάθε τμήμα του αλγορίθμου να χρησιμοποιεί το προηγούμενο.

Συνολικά ο διαχωρισμός έχει ως εξής

Τμήμα	Ρόλος	Περιγραφή
EDRR	Ο υπολογισμός της ενέργειας των επιμέρους λύσεων.	Το τμήμα αυτό περιέχει όλους τους υπολογισμούς που απαιτούνται για την επεξεργασία των δεδομένων. Δεν έχει γνώση για το πώς ή πού θα χρησιμοποιηθούν τα στοιχεία ούτε με την σειρά που θα υπολογιστούν.
Calculators	Η επιλογή της βέλτιστης λύσης για κάποιο δοσμένο υπό-πρόβλημα.	Τα Calculators παρέχουν τα σωστά δεδομένα στο EDRR έτσι ώστε να υπολογιστούν σταδιακά όλες οι επιμέρους λύσεις για κάθε υπό-προβλημα και συγκρατούν την καλύτερη. Έχουν γνώση για των στοιχείων που απαιτούνται για τον υπολογισμό. Αλλά δεν συγκρατούν πληροφορία για τα στοιχεία που επεξεργάζονται και δεν είναι σε θέση να γνωρίζουν αν έχουν λάβει όλα τα στοιχεία.
Calculator Controllers	Η τροφοδοσία των Calculators με τα δεδομένα που χρειάζονται.	Οι Controllers αυτοί αναλαμβάνουν την αρχικοποίηση των Calculators έτσι ώστε να υπολογίσουν την τιμή για κάποιο συγκεκριμένο υπό-πρόβλημα, την μετέπειτα τροφοδοσία τους με τις τιμές που χρειάζονται και την αποθήκευση της βέλτιστης λύσης στους αντίστοιχους πίνακες. Γνωρίζει για την σειρά με την οποία πρέπει να υπολογιστούν τα υπό-προβλήματα καθώς και για τα στοιχεία που θα χρειαστεί ο υπολογισμός.
Calculation Limiter	Ο έλεγχος των Calculator Controllers έτσι ώστε να μην παραβιαστούν οι εξαρτήσεις μεταξύ των δεδομένων.	Τέλος ο Calculation Limiter παρέχει έναν κεντρικό έλεγχο επί των Controllers περιορίζοντας τον υπολογισμό των στοιχείων για να εξασφαλίσει την τήρηση των εξαρτήσεων. Ο Limiter γνωρίζει το σημείο μέχρι το οποίο έχουν υπολογίζει όλοι οι Controllers και παρέχει ένα όριο, μέχρι το οποίο είναι ασφαλής ο υπολογισμός.
Memory Controller	Η ανάγνωση και εγγραφή στοιχείων στην μνήμη.	Επιπλέον υπάρχει ένα σύστημα διαχείρισης μνήμης το οποίο δεν θεωρείται τμήμα της αρχιτεκτονικής μιας και εξαρτάται απόλυτα από την μνήμη που θα χρησιμοποιηθεί. Ρόλος της είναι η ανάγνωση και εγγραφή στοιχείων στην μνήμη όπως αυτά απαιτούνται από τους Calculator Controllers.

Table 12 - Διαχωρισμός αλγορίθμου σε υποσυστήματα

Repository για τα πειραματικά δεδομένα (EDRR)

Το EDRR αποτελεί το χαμηλότερου επιπέδου υποσύστημα, το οποίο αναλαμβάνει τον υπολογισμό της τιμής για κάθε μία μεμονωμένη λύση. Κύρια κατεύθυνση σχεδιασμού είναι η εκτέλεση πολλών παράλληλων υπολογισμών με έναν σταθερό ρυθμό επεξεργασίας.

Η επικοινωνία με τα υπόλοιπα υποσυστήματα και συγκεκριμένα με τα Calculators γίνεται μέσο πολλαπλών διεπαφών τόσο για την εισαγωγή των στοιχείων τα οποία απαιτούνται για τον υπολογισμό όσο και για την επιστροφή των αποτελεσμάτων. Οι πολλαπλές διεπαφές δεν έχουν άμεση σχέση με τον αριθμό των παράλληλων υπολογισμών καθώς δεν θα είναι ποτέ όλες ενεργές. Αντίθετα ο ρόλος τους είναι να εξασφαλίσουν ένα αποκλειστικό σημείο πρόσβασης από κάθε Calculator και να παρέχει έναν κεντρικό σύστημα προτεραιοτήτων για την επιλογή του στοιχείου που πρόκειται να υπολογιστεί.

Οι είσοδοι του υποσυστήματος αυτού φαίνονται στον παρακάτω πίνακα

Είσοδος	Τύπος	Περιγραφή
val	$Data$	Η τιμή που διαβάστηκε από την μνήμη. Αντιστοιχεί στο στοιχείο στην θέση (k, l), ενώ η πληροφορία για από ποιον πίνακα προέρχεται έχει ενσωματωθεί στην είσοδο $type$
(i, j)	$Index, Index$	Η θέση του στοιχείου που αντιστοιχεί στο υπό-πρόβλημα στο οποίο ανήκει η λύση η οποία θα υπολογιστεί. Χρησιμοποιείται για την ανάκληση των ενεργειών οι οποίες αντιστοιχούν στο ζεύγος που κλείνει το K-loop.
(k, l)	$Index, Index$	Η θέση του υπό-προβλήματος στο οποίο αντιστοιχεί η τιμή της εισόδου val. Χρησιμοποιείται για την ανάκληση των ενεργειών οι οποίες αντιστοιχούν στο ζεύγος που περιέχεται στο Loop.
$type$	$ctype$	Ο τύπος του υπολογισμού ο οποίος θα εκτελεστεί. Η τιμή αυτή επηρεάζει τον τρόπο με τον οποίο θα υπολογιστεί το αποτέλεσμα.

Table 13 - Διεπαφή εισόδου (EDRR)

Αντίστοιχα οι έξοδοι του υποσυστήματος φαίνονται στον παρακάτω πίνακα

Έξοδος	Τύπος	Περιγραφή
res	$Data$	Η τιμή της επιμέρους λύσης για το υπό-πρόβλημα που αντιστοιχεί στην θέση (i, j)

Table 14 - Διεπαφή εξόδου (EDRR)

Ο τύπος $ctype$ χρησιμοποιείται για την μεταφορά δεδομένων ελέγχου και συγκεκριμένα τον τύπο του υπολογισμού που πρέπει να εκτελεστεί. Οι διαθέσιμοι διαφορετικοί υπολογισμοί που θα πρέπει να είναι σε θέση να εκτελέσει το EDRR είναι οι εξής

ctype	Αποτέλεσμα
NoCalculation	-
HairpinLoop	$eh(i,j)$
StackedPair	$es(i,j) + val$
BulgeLeft	$ebl(i,j,k) + val$
BulgeRight	$ebr(i,j,k) + val$
InteriorLoop	$ei(i,j,k,l) + val$
InteriorLoop1x1	$ei1x1(pairing(i,j), pairing(k,l), i+1, j-1) + val$
InteriorLoop1x2	$ei1x2(pairing(i,j), pairing(k,l), i+1, l+1, j-1) + val$
InteriorLoop2x1	$ei1x2(pairing(l,k), pairing(j,i), l+1, i+1, k-1) + val$
InteriorLoop2x2	$ei2x2(pairing(i,j), pairing(k,l), i+1, k-1, l+1, j-1) + val$
JoinStructure1x1	$a + c + val$
JoinStructure1x2	$a + c + val + d5(i,j,i+1) + b$
JoinStructure2x1	$a + c + val + d3(i,j,j-1) + b$
JoinStructure2x2	$a + c + val + d5(i,j,i+1) + d3(i,j,j-1) + 2b$
DangleLeft	$b + c + d5(i,j-1,j) + val$
DangleRight	$b + c + d3(i+1,j,i) + val$
DangleBoth	$2b + c + d3(i+1,j,i) + d5(i,j-1,j) + val$
AppendBase	$val + b$
NULLCalculation	val

Table 15 - Τύποι υπολογισμού (EDRR)

Οι παραπάνω περιπτώσεις καλύπτουν το σύνολο του απαραίτητων υπολογισμών που είναι απαραίτητο για την συνολική εξαγωγή των τιμών των υπό-προβλημάτων. Στην συνέχεια θα αναλυθεί περισσότερο η δομή του EDRR.

Επιλογή στοιχείων προς επίλυση

Το EDRR σχεδιάστηκε έτσι ώστε κάθε Calculator να έχει ένα αποκλειστικό σημείο σύζευξης μέσω του οποίου να μπορεί να στείλει τα δεδομένα που χρειάζεται για να υπολογιστούν. Επειδή ώμος ως επί το πλείστον οι διεπαφές αυτές δεν χρησιμοποιούνται συνεχώς, το EDRR δεν παρέχει για κάθε μία σύζευξη ένα διαφορετικό πυρήνα υπολογισμού αλλά αντίθετα πολυπλέκει στον χρόνο τις «αιτήσεις» από τα Calculator σε ένα μικρότερο σύνολο πυρήνων. Οι πυρήνες αυτοί είναι ανεξάρτητοι μεταξύ τους και ο κάθε ένας είναι σε θέση να εκτελέσει όλους τους δυνατούς συνδυασμούς πράξεων.

Αποθήκευση RNA

Απαραίτητο στοιχείο στον υπολογισμό του αποτελέσματος αποτελεί η μετάφραση των δεικτών $i, i+1, k-1, k, l, l+1, j-1, j$ στις αντίστοιχες τιμές βάσεων. Για την επίτευξη ενός σταθερού ρυθμού υπολογισμών θα πρέπει όλες οι τιμές αυτές να μεταφράζονται ταυτόχρονα. Έτσι το RNA αποθηκεύεται σε πολλαπλά αντίγραφα, με κάθε πυρήνα να διαθέτει τα δικά του αντίγραφα, σε εσωτερικές μνήμες έτσι ώστε να γίνεται άμεσα η μετάφραση των στοιχείων.

Αποθήκευση ενεργειών

Αντίθετα με την αποθήκευση του RNA τα πειραματικά δεδομένα δεν χρειάζεται να αποθηκεύονται σε πολλαπλά αντίγραφα καθώς μόνο μία τιμή θα χρειαστεί από τον κάθε πίνακα. Κάθε σύνολο πειραματικών δεδομένων θα αποθηκεύεται σε δικιά του μνήμη με το αντίστοιχο σύνολο διευθυνσιοδότησης.

Σχηματική αναπαράσταση EDRR

Στο παρακάτω σχήμα παρατηρούνται τα βασικά στοιχεία του EDRR. Ο PriorityMultiplexer για την επιλογή μεταξύ των δεδομένων εισόδου, το RNA Repository για την μετάφραση και το ExperimentalDataRepository για την ανάκτηση των πειραματικών τιμών και την μεταξύ τους συνάθροιση και η πρόσθεση της τιμής αυτής στην τιμή val για την εξαγωγή της τελικής λύσης.

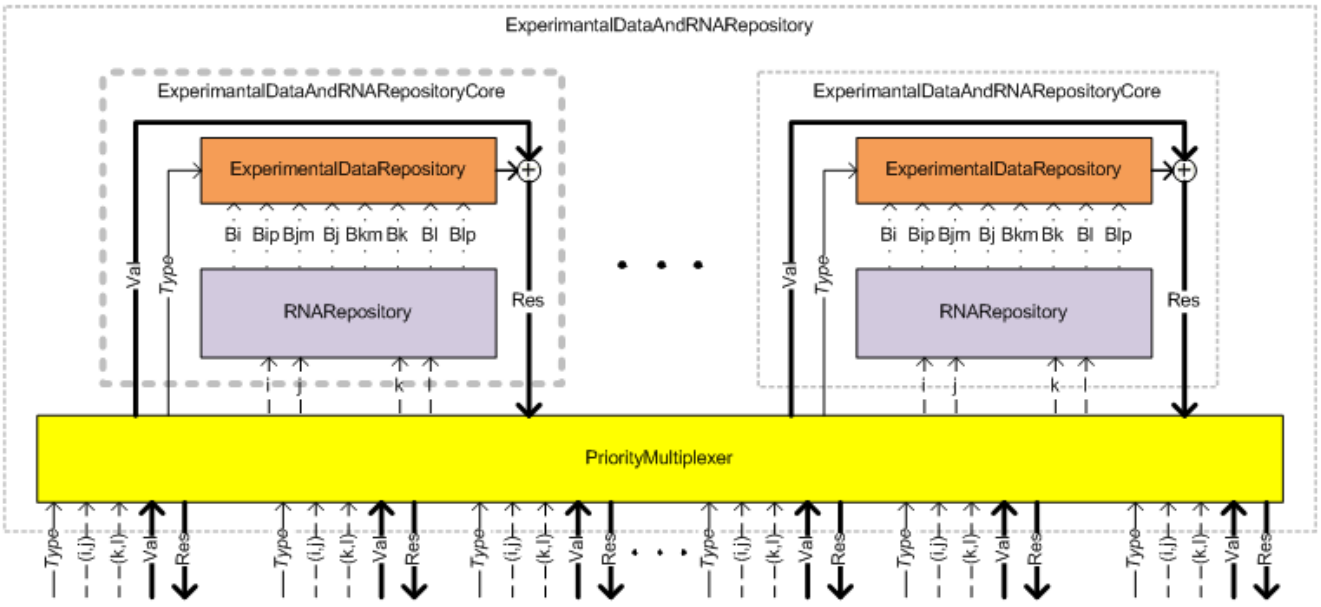


Figure 40 - Σχηματική αναπαράσταση του EDRR

Calculators

Τα $Calculators$ έχουν τέσσερα στάδια λειτουργίας κάθε ένα από τα οποία επηρεάζει τον τρόπο με τον οποίο λειτουργούν.

Κατάσταση	Περιγραφή	Μετάβαση κατάστασεων
$IDLE$	Στην κατάσταση αυτή τα $Calculators$ είναι ανενεργά και δεν δέχονται κανένα στοιχείο από τους $Controllers$ ούτε παρέχουν στοιχεία για υπολογισμό στο $EDRR$.	Η μοναδική περίπτωση για να περάσει κάποιο $Calculator$ σε άλλη κατάσταση είναι μέσω της αρχικοποίησης του μέσω ενός σήματος $init$. Ταυτόχρονα με την αρχικοποίηση παρέχονται και κάποια δεδομένα αρχικοποίησης όπως το ζεύγος (i, j) το οποίο περιέχει την θέση του στοιχείου το οποίο αρχικοποιείται για να υπολογίσει.
$BUSY$	Η κατάσταση αυτή είναι η κατάσταση κατά την οποία εκτελούνται οι υπολογισμοί. Τα $Calculators$ σε αυτήν την κατάσταση δέχονται στοιχεία από τους $Controllers$ για τα οποία υπολογίζουν τον τύπο υπολογισμού και αναθέτουν στο $EDRR$ την επεξεργασία των δεδομένων. Παράλληλα δέχονται τα αποτελέσματα από το $EDRR$ και διατηρούν κάθε φορά την ελάχιστη τιμή.	Τα $Calculators$ δεν διατηρούν κάποια πληροφορία για τα στοιχεία που απαιτεί ο υπολογισμός ούτε για τα στοιχεία που έχουν ήδη επεξεργαστεί, έτσι βασίζονται στους $Controllers$ για αυτήν την πληροφορία διαμέσου του σήματος $finish$ το οποίο ενημερώνει το $Calculator$ ότι όλα τα στοιχεία έχουν δοθεί.
$TERMINATED$	Η κατάσταση αυτή είναι μια εσωτερική κατάσταση στην οποία παραμένει το $Calculator$ μέχρι να λάθει όλες τις τιμές από το $EDRR$ και να συγκεντρώσει το τελικό αποτέλεσμα. Στην κατάσταση αυτή δεν υπάρχει καμία ανταλλαγή δεδομένων με τους $Controllers$ ενώ η επικοινωνία με το $EDRR$ συνεχίζεται κανονικά.	Η μετάβαση στην επόμενη κατάσταση γίνεται από το ίδιο το $Calculator$ όταν έχει υπολογιστεί το ζητούμενο αποτέλεσμα.
$RESULT$	Τέλος στην κατάσταση αυτή το $Calculator$ ενημερώνει τον $Controller$ ότι το αποτέλεσμα είναι διαθέσιμο και περιμένει μέρχι την ανάγνωσή του.	Η μετάβαση στην επόμενη κατάσταση συμβαίνει ταυτόχρονα με την ανάγνωση του αποτελέσματος.

Table 16 - Καταστάσεις λειτουργίας Calculators

Στην συνέχεια παρουσιάζεται το διάγραμμα μετάβασης καταστάσεων.

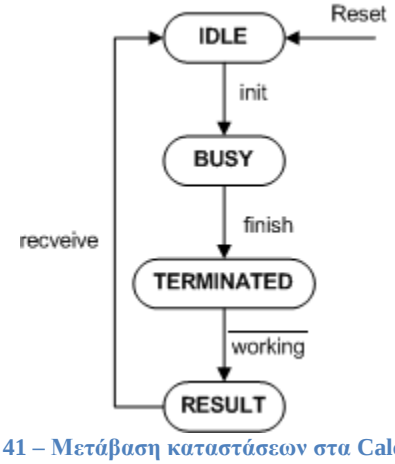


Figure 41 – Μετάβαση καταστάσεων στα Calculators

Πέρα των γενικών κατευθύνσεων για τα $Calculators$ δεν παύουν να υπάρχουν ιδιομορφίες ανάμεσα στα τρία ήδη, εκείνα τα οποία υπολογίζουν στοιχεία για τους πίνακες V, W και L. Στην συνέχεια θα παρουσιαστούν αναλυτικά οι διάφοροι τύποι.

VCalculators

Τα $Calculators$ αυτού του τύπου υπολογίζουν τις τιμές για τον πίνακα V. Η συγκεκριμένη κατηγορία ακολουθεί, χωρίς καμία διαφοροποίηση, την βασική λειτουργία των $Calculators$.

Γραφικά μπορούμε να δούμε τα $VCalculators$ ως εξής

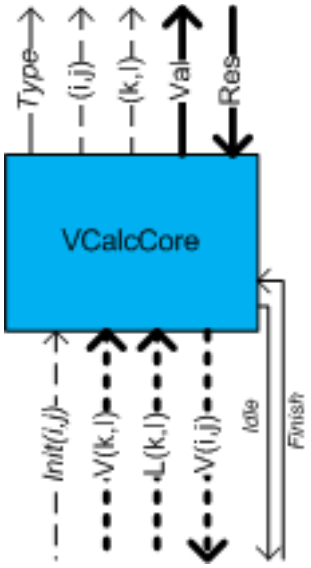


Figure 42 - Σχηματική αναπαράσταση του VCalculator

Οι είσοδοι των $VCalculators$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Είσοδος	Σύνδεση	Τύπος	Περιγραφή
$Init(i,j)$	$Controller$	$Bool(Index, Index)$	Το σήμα, $init$, για την αρχικοποίηση του $Calculator$ καθώς και η θέση του στοιχείου που πρέπει να υπολογιστεί.
$V(k,l)$	$Controller$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου που αντιστοιχεί στο υπό-πρόβλημα από τον πίνακα V το οποίο θα επεξεργαστεί.
$L(k,l)$	$Controller$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου που αντιστοιχεί στο υπό-πρόβλημα από τον πίνακα L το οποίο θα επεξεργαστεί.
Res	$EDRR$	$Data$	Η τιμή από το $EDRR$ όπως υπολογίστηκε για τα δεδομένα που στάλθηκαν.
$Finish$	$Controller$	$Bool$	Το σήμα το οποίο ενημερώνει το $Calculator$ ότι όλα τα στοιχεία έχουν σταλθεί από τον $Controller$

Table 17 - Εισόδοι για τα VCalculators

Αντίστοιχα οι εξόδοι των $VCalculators$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Έξοδος	Σύνδεση	Τύπος	Περιγραφή
$V(i,j)$	$Controller$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου που υπολογίστηκε από το $Calculator$
val	$EDRR$	$Data$	Είτε η τιμή του στοιχείου $V(k,l)$ είτε του στοιχείου $L(k,l)$, όπως αυτά δόθηκαν για επεξεργασία.
(i,j)	$EDRR$	$Index, Index$	Η θέση του στοιχείου που υπολογίζει την δεδομένη στιγμή το $Calculator$
(k,l)	$EDRR$	$Index, Index$	Η θέση του υπό-προβλήματος στο οποίο αντιστοιχεί η τιμή της εισόδου val.
$type$	$EDRR$	$ctype$	Ο τύπος του υπολογισμού ο οποίος θα εκτελεστεί. Η τιμή αυτή επηρεάζει τον τρόπο με τον οποίο θα υπολογιστεί το αποτέλεσμα.
$Idle$	$Controller$	$Bool$	Το σήμα το οποίο ενημερώνει τον $Controller$ αν το $Calculator$ είναι ανενεργό ή όχι.

Table 18 - Εξόδοι για τα VCalculators

Στην συνέχεια ακολουθεί η διαδικασία εξαγωγής του τύπου του υπολογισμού από τα στοιχεία (i,j), (k,l) καθώς και τον πίνακα στον οποίο ανήκει η τιμή.

Πίνακας	Τιμή k	Τιμή l	Τύπος υπολογισμού
V	$i+1$	$j-1$	*StackedPair*
	$i+1$	$< j-1$	*BulgeRight*
	$> i+1$	$j-1$	*BulgeLeft*
	$i+2$	$j-2$	*InteriorLoop1x1*
	$i+2$	$j-3$	*InteriorLoop1x2*
	$i+3$	$j-2$	*InteriorLoop2x1*
	$i+3$	$j-3$	*InteriorLoop2x2*
	$i+2$	$< j-3$	*InteriorLoop*
	$i+3$	$< j-3$	
	$> i+3$	$< j-3$	
	$> i+3$	$j-2$	
	$> i+3$	$j-3$	
	αλλιώς		NoCalculation
L	$i+2$	$j-2$	*JoinStructure1x1*
	$i+2$	$j-3$	*JoinStructure1x2*
	$i+3$	$j-2$	*JoinStructure2x1*
	$i+3$	$j-3$	*JoinStructure2x2*
	αλλιώς		NoCalculation

Table 19 - Εξαγωγή τύπου υπολογισμού για τα VCalculators

Όπως φαίνεται και από το παραπάνω σχήμα τα $VCalculator$ αγνοούν τα στοιχεία τα οποία δεν χρειάζονται για τον υπολογισμό, επιτρέποντας έτσι στους Controllers να στέλνουν και λανθασμένα στοιχεία χωρίς επιπτώσεις στο τελικό αποτέλεσμα.

WCalculators

Τα $Calculators$ αυτού του τύπου υπολογίζουν τις τιμές για τον πίνακα W. Η συγκεκριμένη κατηγορία ακολουθεί, χωρίς καμία διαφοροποίηση, την βασική λειτουργία των $Calculators$.

Γραφικά μπορούμε να δούμε τα$WCalculators$ ως εξής

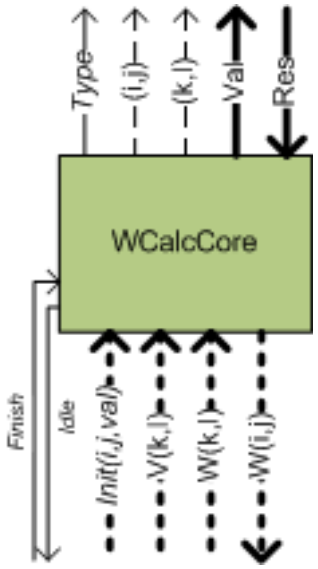


Figure 43 - Σχηματική αναπαράσταση του WCalculator

Οι είσοδοι των $VCalculators$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Είσοδος	Σύνδεση	Τύπος	Περιγραφή
$Init(i,j,val)$	$Controller$	$Bool(Index, Index, Data)$	Το σήμα, $init$, για την αρχικοποίηση του $Calculator$ καθώς και η θέση του στοιχείου που πρέπει να υπολογιστεί. Επιπλέον η αρχικοποίηση περιέχει την τιμή του στοιχείου $L(i,j)$ και συμπεριλαμβάνεται ως λύση του υπό-προβλήματος.
$V(k,l)$	$Controller$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου που αντιστοιχεί στο υπό-πρόβλημα από τον πίνακα V το οποίο θα επεξεργαστεί.
$W(k,l)$	$Controller$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου που αντιστοιχεί στο υπό-πρόβλημα από τον πίνακα W το οποίο θα επεξεργαστεί.
Res	$EDRR$	$Data$	Η τιμή από το $EDRR$ όπως υπολογίστηκε για τα δεδομένα που στάλθηκαν.
$Finish$	$Controller$	$Bool$	Το σήμα το οποίο ενημερώνει το $Calculator$ ότι όλα τα στοιχεία έχουν σταλθεί από τον $Controller$

Table 20 - Εισόδοι για τα WCalculators

Αντίστοιχα οι εξόδοι των $WCalculators$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Έξοδος	Σύνδεση	Τύπος	Περιγραφή
$W(i,j)$	$Controller$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου που υπολογίστηκε από το $Calculator$
val	$EDRR$	$Data$	Είτε η τιμή του στοιχείου $V(k,l)$ είτε του στοιχείου $L(k,l)$, όπως αυτά δόθηκαν για επεξεργασία.
(i,j)	$EDRR$	$Index, Index$	Η θέση του στοιχείου που υπολογίζει την δεδομένη στιγμή το $Calculator$
(k,l)	$EDRR$	$Index, Index$	Η θέση του υπό-προβλήματος στο οποίο αντιστοιχεί η τιμή της εισόδου val.
$type$	$EDRR$	$ctype$	Ο τύπος του υπολογισμού ο οποίος θα εκτελεστεί. Η τιμή αυτή επηρεάζει τον τρόπο με τον οποίο θα υπολογιστεί το αποτέλεσμα.
$Idle$	$Controller$	$Bool$	Το σήμα το οποίο ενημερώνει τον $Controller$ αν το $Calculator$ είναι ανενεργό ή όχι.

Table 21 - Εξόδοι για τα WCalculators

Στην συνέχεια ακολουθεί η διαδικασία εξαγωγής του τύπου του υπολογισμού από τα στοιχεία (i,j), (k,l) καθώς και τον πίνακα στον οποίο ανήκει η τιμή.

Πίνακας	Τιμή k	Τιμή l	Τύπος υπολογισμού
V	i	j	*NULLCalculation*
	i	$j-1$	*DangleRight*
	$i+1$	j	*DangleLeft*
	$i+1$	$j-1$	*DangleBoth*
	αλλιώς		NoCalculation
W	i	$j-1$	AppendBase
	$i+1$	j	
	αλλιώς		NoCalculation

Table 22 - Εξαγωγή τύπου υπολογισμού για τα WCalculators

Όπως φαίνεται και από το παραπάνω σχήμα τα $WCalculator$ αγνοούν τα στοιχεία τα οποία δεν χρειάζονται για τον υπολογισμό, επιτρέποντας έτσι στους Controllers να στέλνουν και λανθασμένα στοιχεία χωρίς επιπτώσεις στο τελικό αποτέλεσμα.

LCalculators

Τα $Calculators$ αυτού του τύπου υπολογίζουν τις τιμές για τον πίνακα L. Η συγκεκριμένη κατηγορία αποτελεί εξαίρεση ανάμεσα στα $Calculators$ των άλλων δύο τύπων για δύο βασικούς λόγους. Πρώτον δεν χρησιμοποιεί το $EDRR$ καθώς δεν απαιτεί πειραματικά δεδομένα για τον υπολογισμό της τελικής λύσης και δεύτερον λαμβάνει τα στοιχεία ανά ζεύγη.

Γραφικά μπορούμε να δούμε τα $LCalculators$ ως εξής

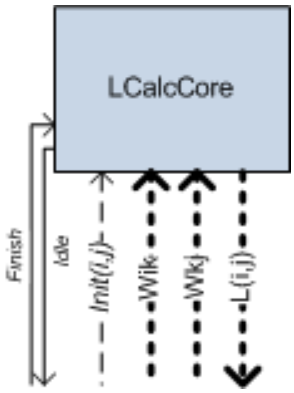


Figure 44 - Σχηματική αναπαράσταση του LCalculator

Οι είσοδοι των $LCalculators$, καθώς και πληροφορία για τον τύπο τους, φαίνονται στον παρακάτω πίνακα. Καθότι όλες οι συνδέσεις γίνονται με τον $Controller$ έχει απαλειφθεί το πεδίο "σύνδεση".

Είσοδος	Τύπος	Περιγραφή
$Init(i,j)$	$Bool(Index, Index)$	Το σήμα, $init$, για την αρχικοποίηση του $Calculator$ καθώς και η θέση του στοιχείου που πρέπει να υπολογιστεί.
Wik	$Data$	Η τιμή του στοιχείου που αντιστοιχεί στο υπό-πρόβλημα $W(i,k)$
Wkj	$Data$	Η τιμή του στοιχείου που αντιστοιχεί στο υπό-πρόβλημα $W(k+1,j)$
$Finish$	$Bool$	Το σήμα το οποίο ενημερώνει το $Calculator$ ότι όλα τα στοιχεία έχουν σταλθεί από τον $Controller$

Table 23 - Εισόδοι για τα LCalculators

Αντίστοιχα οι εξόδοι των $LCalculators$ και η πληροφορία για τον τύπο τους.

Έξοδος	Τύπος	Περιγραφή
$L(i,j)$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου που υπολογίστηκε από το $Calculator$
$Idle$	$Bool$	Το σήμα το οποίο ενημερώνει τον $Controller$ αν το $Calculator$ είναι ανενεργό ή όχι.

Table 24 - Εξόδοι για τα LCalculators

Σημαντική διαφοροποίηση σε σχέση με τα άλλα $Calculators$ αποτελεί επίσης το γεγονός ότι τα στοιχεία λαμβάνονται ως απλές τιμές, χωρίς τους δείκτες για την θέση τους στοιχείου. Επιπλέον τα στοιχεία Wik και Wkj απαιτήται να φτάνουν στο $LCalculator$ την ίδια στιγμή.

Calculator Controllers

Μοναδική ευθύνη των $CalculatorControllers$, για συντομία $Controllers$, είναι η αρχικοποίηση των $Calculator$ η τροφοδοσία τους με όλες τις τιμές που χρειάζονται για τον υπολογισμό και η αποθήκευση των αποτελεσμάτων. Οι $Controllers$, όπως και τα $Calculators$ έχουν τέσσερα στάδια λειτουργίας κάθε ένα από τα οποία επηρεάζει τον τρόπο με τον οποίο λειτουργούν.

Κατάσταση	Περιγραφή	Μετάβαση κατάστασεων
$INITIALIZE$	Στην κατάσταση οι $Controllers$ αρχικοποιούν όσα από τα $Calculators$ πρόκειται να πάρουν μέρος στον επόμενο υπολογισμό. Ο αριθμός τους μπορεί να ποικίλει από ένα έως και όλα τα διαθέσιμα $Calculators$. Σε περίπτωση όπου δεν μπορεί να εκτελεστεί κανένας υπολογισμός οι $Controllers$ περιμένουν σε αυτήν την κατάσταση μέχρις ότου δημιουργηθούν οι συνθήκες που θα επιτρέψουν τον επόμενο υπολογισμό.	Η μοναδική περίπτωση για να περάσει κάποιος $Controller$ σε άλλη κατάσταση είναι η επιτυχημένη αρχικοποίηση όλων των $Calculator$. Οι $Controllers$ μέσω του σήματος ελέγχου $idle$ που λαμβάνουν μπορούν να γνωρίζουν αν τα ζητούμενα $Calculator$ έχουν πράγματι αρχικοποιηθεί και περιμένουν δεδομένα.
$SEND$	Η κατάσταση αυτή είναι η κατάσταση κατά την οποία εκτελούνται οι υπολογισμοί. Οι $Controllers$ αρχικά στέλνουν αιτήματα προς τον $MemoryController$ και προωθούν τα δεδομένα στα $Calculators$.	Όταν όλα τα στοιχεία έχουν περαστεί στα $Calculator$ οι $Controllers$ αυτόματα περνάν στην επόμενη κατάσταση λειτουργίας.
$FINALIZE$	Η κατάσταση αυτή είναι μια εσωτερική κατάσταση στην οποία οι $Controllers$ ενημερώνουν τα $Calculators$ ότι έχουν λάβει όλα τα δεδομένα.	Η μετάβαση στην επόμενη κατάσταση γίνεται όταν όλα τα $Calculators$ ενημερωθούν.
$RECEIVE$	Τέλος στην κατάσταση αυτή οι $Controllers$ λαμβάνουν το τελικό αποτέλεσμα από τα $Calculators$ και το προωθούν στον $MemoryController$ για να αποθηκευτούν στην μνήμη.	Η μετάβαση στην επόμενη κατάσταση συμβαίνει ταυτόχρονα με την ανάγνωση του αποτελέσματος.

Table 25 - Καταστάσεις λειτουργίας των Controllers

Στην συνέχεια παρουσιάζεται το διάγραμμα μετάβασης καταστάσεων.

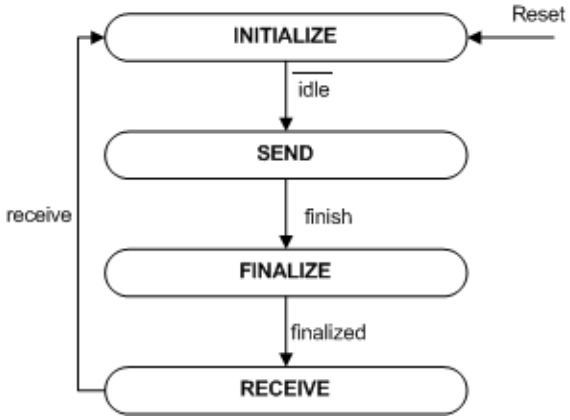


Figure 45 - Μετάβαση καταστάσεων στους Controllers

Η πρόσβαση στα στοιχεία πινάκων V, W και L γίνεται μέσω μιας μορφής αίτησης από τους $Controllers$ στον $MemoryController$. Η αίτηση περιέχει την θέση του πρώτου στοιχείου, (i, j), καθώς και το πλήθος των στοιχείων που θέλει να λάβει, $count$. Η πρόσβαση σε στοιχεία των διαφορετικών πινάκων γίνεται μέσω ξεχωριστών διεπαφών κάθε μία για την ανάγνωση στοιχείων από τον πίνακα στον οποίο αντιστοιχεί.

Στην συνέχεια θα γράφουμε τις αιτήσεις στην μορφή $\Pi(i, j, count)$ όπου, $\Pi \in \{V, W, L\}$ και δηλώνει θα την διεπαφή για την ανάγνωση στοιχείων από τον πίνακα Π, (i, j) την θέση του πρώτου στοιχείου και $count$ το πλήθος των στοιχείων.

Ο $MemoryController$ στην επιστρέφει τα στοιχεία σειριακά με πρώτο το στοιχείο (i, j) και στην συνέχεια τα επόμενα στοιχεία στην διαγώνιο που ανήκει το πρώτο στοιχείο.

Έτσι για την αίτηση $V(10,20,4)$ ο $Controller$ θα λάβει τα στοιχεία

$$V(10,20),\ V(11,21),\ V(12,22) \text{ και } V(13,23)$$

Γραφικά τα στοιχεία αυτά καταλαμβάνουν την ακόλουθη θέση στον πίνακα

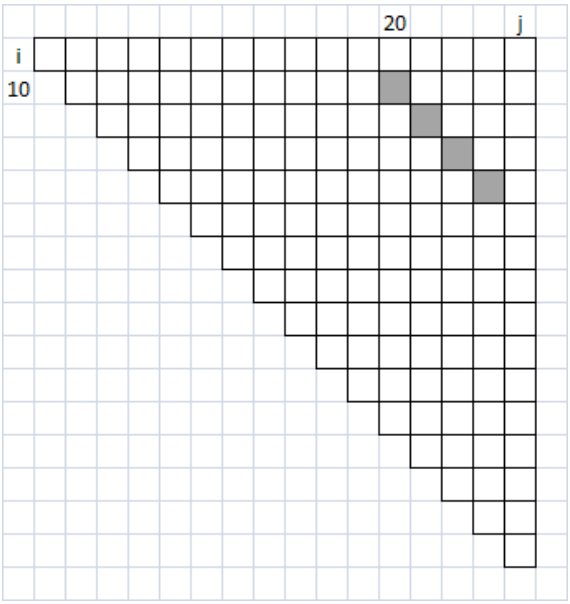


Figure 46 - Στοιχεία για την αίτηση V(10,20,4)

Πέρα των γενικών κατευθύνσεων για τους $Controllers$ δεν παύουν να υπάρχουν ιδιομορφίες ανάμεσα στα τρία ήδη, εκείνα τα οποία διαχειρίζονται τα στοιχεία για των υπολογισμό των πινάκων V, W και L. Στην συνέχεια θα παρουσιαστούν αναλυτικά οι διάφοροι τύποι.

Έχοντας να περιγράψουμε μονάδες με πολλαπλά σημεία εισόδου και εξόδου, ένα για κάθε $Calculator$, θα δηλώνουμε θα σήματα αυτά ως δύναμη εις την C, όπου C θα θεωριθεί ο αριθμός των $Calculator$ τα οποία είναι συνδεδεμένα σε κάθε $Controller$.

VCalculator Controller

Οι $Controllers$ αυτού του τύπου διαχειρίζονται τα στοιχεία για τον υπολογισμό των τιμών του πίνακα V. Η συγκεκριμένη κατηγορία ακολουθεί, χωρίς καμία διαφοροποίηση, την βασική λειτουργία των $Controllers$.

Γραφικά μπορούμε να δούμε τους $VCalculatorControllers$ ως εξής

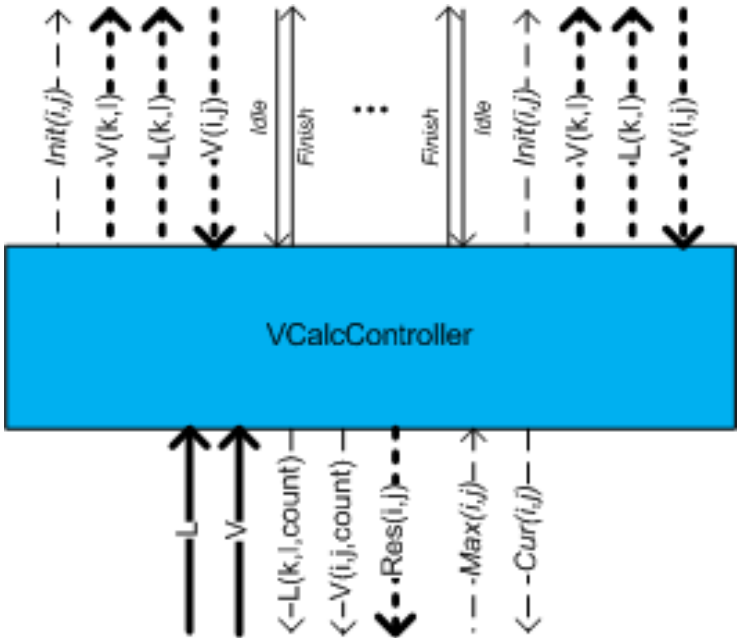


Figure 47 - Σχηματική αναπαράσταση του VCalculator Controller

Οι είσοδοι των $VCalculatorControllers$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Είσοδος	Σύνδεση	Τύπος	Περιγραφή
$Max(i,j)$	$CalculationLimiter$	$(Index, Index)$	Το μέγιστο σημείο το οποίο μπορεί να υπολογίσει ο $Controller$ χωρίς την παραβίαση των εξαρτήσεων.
V	$MemoryController$	$Data$	Τα δεδομένα που "ζήτησε" ο $Controller$ από τον πίνακα V.
L	$MemoryController$	$Data$	Τα δεδομένα που "ζήτησε" ο $Controller$ από τον πίνακα L.
$V(i,j)^C$	$Calculator$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου που υπολογίστηκε από το κάθε $Calculator$
$idle^C$	$Calculator$	$Bool$	Το σήμα το οποίο ενημερώνει τον $Controller$ αν το $Calculator$ είναι ανενεργό ή όχι.

Table 26 - Εισόδοι του VCalculator Controller

**Οι είσοδοι που έχουν δύναμη εις την C αντιστοιχούν σε πολλαπλές εισόδους μία για κάθε ένα $Calculator$.*

Αντίστοιχα οι έξοδοι των $VCalculators$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Έξοδος	Σύνδεση	Τύπος	Περιγραφή
$Cur(i,j)$	$CalculationLimiter$	$(Index, Index)$	Το σημείο μέχρι το οποίο έχει υπολογίσει και αποθηκεύσει στην μνήμη ο $Controller$.
$Init(i,j)^C$	$Calculator$	$Bool(Index, Index)$	Το σήμα, $init$, για την αρχικοποίηση των $Calculators$ καθώς και η θέση του στοιχείου που πρέπει να υπολογίσει κάθε ένα από αυτά.
$V(k,l)^C$	$Calculator$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου από τον πίνακα V το οποίο θα αποσταλεί στα $Calculators$ για επεξεργασία.
$L(k,l)^C$	$Calculator$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου από τον πίνακα L το οποίο θα αποσταλεί στα $Calculators$ για επεξεργασία.
$Finish^C$	$Calculator$	$Bool$	Το σήμα το οποίο ενημερώνει τα $Calculators$ ότι όλα τα στοιχεία έχουν σταλθεί.
$L(k,l,count)$	$MemoryController$	$(Index, Index, Index)$	Αίτηση προς τον $MemoryController$ για λήψη $count$ στοιχείων με πρώτο το στοιχείο (k,l) από τον πίνακα L.
$V(k,l,count)$	$MemoryController$	$(Index, Index, Index)$	Αίτηση προς τον $MemoryController$ για λήψη $count$ στοιχείων με πρώτο το στοιχείο (k,l) από τον πίνακα V.
$Res(k,l)$	$MemoryController$	$Data(Index, Index)$	Η θέση και η τιμή ενός από τα στοιχεία τα οποία υπολογίστηκαν από κάποιο $Calculator$.

Table 27 - Εξόδοι του VCalculator Controller

**Οι είσοδοι που έχουν δύναμη εις την C αντιστοιχούν σε πολλαπλές εισόδους μία για κάθε ένα $Calculator$.*

Βασική λειτουργία των $Controllers$ αποτελεί η παραγωγή των αιτήσεων για ανάκτηση στοιχείων από τους πίνακες. Στην συνέχεια παρουσιάζονται οι αλγόριθμοι που χρησιμοποιεί ο $VCalculatorController$.

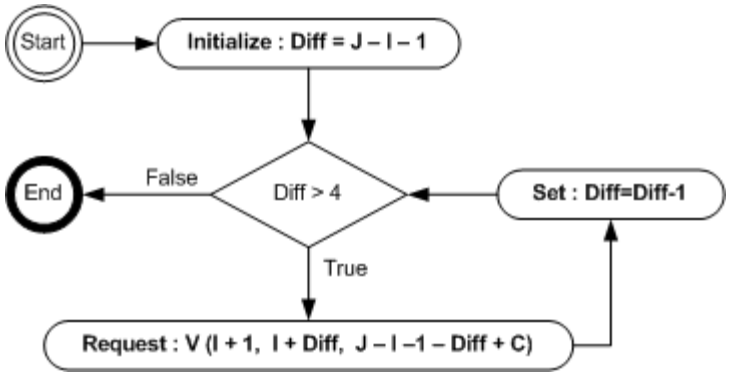


Figure 48 - Αιτήσεις VCalculatorController για στοιχεία από τον πίνακα V

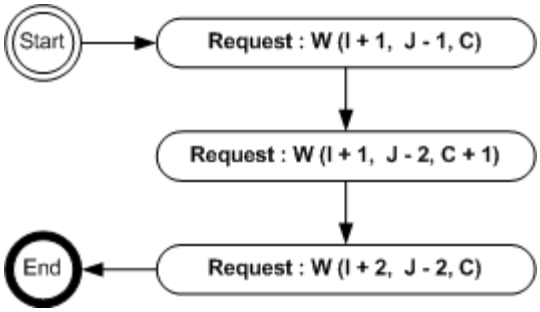


Figure 49 - Αιτήσεις VCalculatorController για στοιχεία από τον πίνακα W

Όπου (I,J) το πρώτο στοιχείο που υπολογίζεται και C ο αριθμός των $VCalculator$ που χρησιμοποιούνται.

WCalculator Controller

Οι $Controllers$ αυτού του τύπου διαχειρίζονται τα στοιχεία για τον υπολογισμό των τιμών του πίνακα W . Η συγκεκριμένη κατηγορία ακολουθεί, χωρίς καμία διαφοροποίηση, την βασική λειτουργία των $Controllers$.

Γραφικά μπορούμε να δούμε τους $WCalculatorControllers$ ως εξής

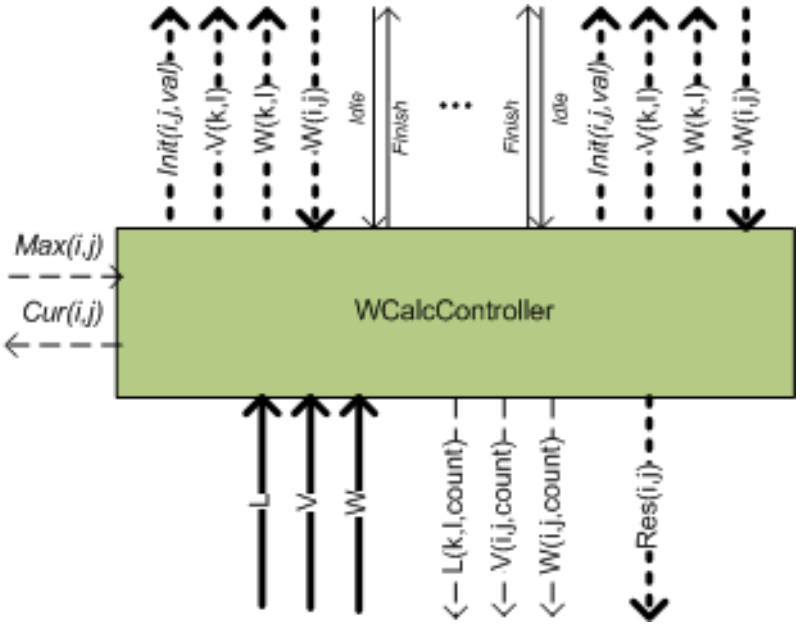


Figure 50 - Σχηματική αναπαράσταση του WCalculator Controller

Οι είσοδοι των $WCalculatorControllers$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Είσοδος	Σύνδεση	Τύπος	Περιγραφή
$Max(i,j)$	$CalculationLimiter$	$(Index, Index)$	Το μέγιστο σημείο το οποίο μπορεί να υπολογίσει ο $Controller$ χωρίς την παραβίαση των εξαρτήσεων.
V	$MemoryController$	$Data$	Τα δεδομένα που "ζήτησε" ο $Controller$ από τον πίνακα V.
L	$MemoryController$	$Data$	Τα δεδομένα που "ζήτησε" ο $Controller$ από τον πίνακα L.
W	$MemoryController$	$Data$	Τα δεδομένα που "ζήτησε" ο $Controller$ από τον πίνακα W.
$W(i,j)^C$	$Calculator$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου που υπολογίστηκε από το κάθε $Calculator$
$idle^C$	$Calculator$	$Bool$	Το σήμα το οποίο ενημερώνει τον $Controller$ αν το $Calculator$ είναι ανενεργό ή όχι.

Table 28 - Εισόδοι του WCalculator Controller

**Οι είσοδοι που έχουν δύναμη εις την C αντιστοιχούν σε πολλαπλές εισόδους μία για κάθε ένα $Calculator$.*

Αντίστοιχα οι έξοδοι των $WCalculators$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Έξοδος	Σύνδεση	Τύπος	Περιγραφή
$Cur(i,j)$	$CalculationLimiter$	$(Index, Index)$	Το σημείο μέχρι το οποίο έχει υπολογίσει και αποθηκεύσει στην μνήμη ο $Controller$.
$Init(i,j,val)^C$	$Calculator$	$Bool(Index, Index, Data)$	Το σήμα, $init$, για την αρχικοποίηση των $Calculators$ καθώς και η θέση του στοιχείου που πρέπει να υπολογίσει κάθε ένα από αυτά. Επιπλέον μαζί με την αρχικοποίηση παρέχεται και η τιμή $val = L(i,j)$.
$V(k,l)^C$	$Calculator$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου από τον πίνακα V το οποίο θα αποσταλεί στα $Calculators$ για επεξεργασία.
$W(k,l)^C$	$Calculator$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου από τον πίνακα W το οποίο θα αποσταλεί στα $Calculators$ για επεξεργασία.
$Finish^C$	$Calculator$	$Bool$	Το σήμα το οποίο ενημερώνει τα $Calculators$ ότι όλα τα στοιχεία έχουν σταλθεί.
$L(k,l,count)$	$MemoryController$	$(Index, Index, Index)$	Αίτηση προς τον $MemoryController$ για λήψη $count$ στοιχείων με πρώτο το στοιχείο (k,l) από τον πίνακα L.
$V(k,l,count)$	$MemoryController$	$(Index, Index, Index)$	Αίτηση προς τον $MemoryController$ για λήψη $count$ στοιχείων με πρώτο το στοιχείο (k,l) από τον πίνακα V.
$W(k,l,count)$	$MemoryController$	$(Index, Index, Index)$	Αίτηση προς τον $MemoryController$ για λήψη $count$ στοιχείων με πρώτο το στοιχείο (k,l) από τον πίνακα W.
$Res(k,l)$	$MemoryController$	$Data(Index, Index)$	Η θέση και η τιμή ενός από τα στοιχεία τα οποία υπολογίστηκαν από κάποιο $Calculator$.

Table 29 - Εξόδοι του WCalculator Controller

**Οι είσοδοι που έχουν δύναμη εις την C αντιστοιχούν σε πολλαπλές εισόδους μία για κάθε ένα $Calculator$.*

Στην συνέχεια παρουσιάζεται ο αλγόριθμος που χρησιμοποιήται για την παραγωγή των αιτήσεων στην μνήμη από τον $WCalculatorController$.

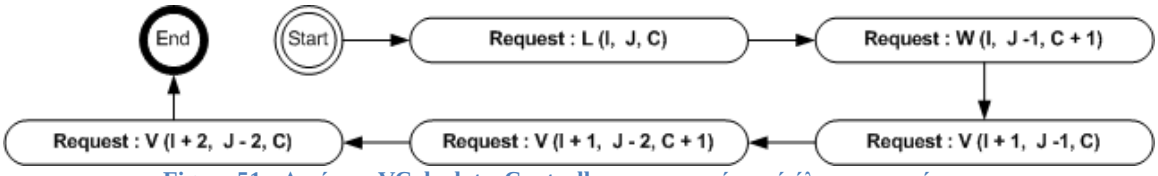


Figure 51 - Αιτήσεις VCalculatorController για στοιχεία από όλους τους πίνακες

Όπου (I,J) το πρώτο στοιχείο που υπολογίζεται και C ο αριθμός των $WCalculator$ που χρησιμοποιούνται.

LCalculator Controller

Οι $Controllers$ αυτού του τύπου διαχειρίζονται τα στοιχεία για τον υπολογισμό των τιμών του πίνακα L. Η συγκεκριμένη κατηγορία ακολουθεί, χωρίς καμία διαφοροποίηση, την βασική λειτουργία των $Controllers$.

Γραφικά μπορούμε να δούμε τους $LCalculatorControllers$ ως εξής

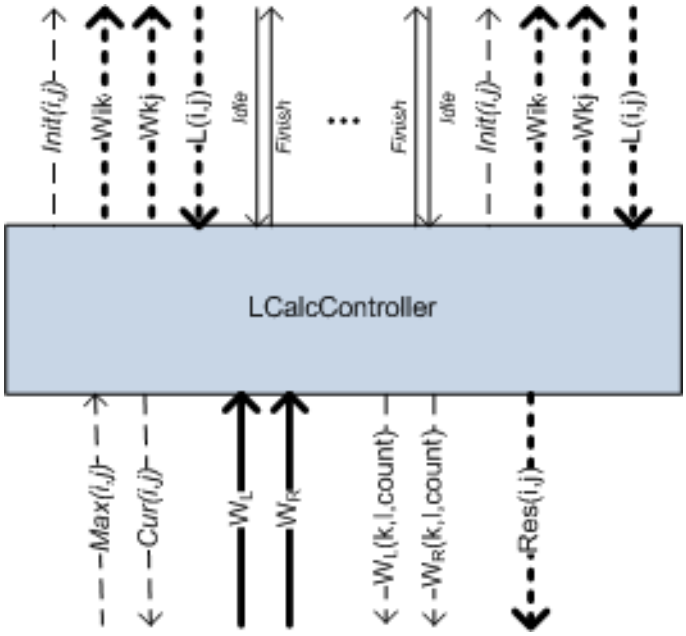


Figure 52 - Σχηματική αναπαράσταση του LCalculator Controller

Οι είσοδοι των $LCalculatorControllers$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Είσοδος	Σύνδεση	Τύπος	Περιγραφή
$Max(i,j)$	$CalculationLimiter$	$(Index, Index)$	Το μέγιστο σημείο το οποίο μπορεί να υπολογίσει ο $Controller$ χωρίς την παραβίαση των εξαρτήσεων.
W_L	$MemoryController$	$Data$	Τα δεδομένα που "ζήτησε" ο $Controller$ από τον πίνακα W.
W_R	$MemoryController$	$Data$	Τα δεδομένα που "ζήτησε" ο $Controller$ από τον πίνακα W.
$L(i,j)^C$	$Calculator$	$Data(Index, Index)$	Η θέση και η τιμή του στοιχείου που υπολογίστηκε από το κάθε $Calculator$
$idle^C$	$Calculator$	$Bool$	Το σήμα το οποίο ενημερώνει τον $Controller$ αν το $Calculator$ είναι ανενεργό ή όχι.

Table 30 - Εισόδοι του LCalculator Controller

**Οι είσοδοι που έχουν δύναμη εις την C αντιστοιχούν σε πολλαπλές εισόδους μία για κάθε ένα $Calculator$.*

Αντίστοιχα οι έξοδοι των $LCalculators$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Έξοδος	Σύνδεση	Τύπος	Περιγραφή
$Cur(i,j)$	$CalculationLimiter$	$(Index, Index)$	Το σημείο μέχρι το οποίο έχει υπολογίσει και αποθηκεύσει στην μνήμη ο $Controller$.
$Init(i,j)^C$	$Calculator$	$Bool(Index, Index)$	Το σήμα, $init$, για την αρχικοποίηση των $Calculators$ καθώς και η θέση του στοιχείου που πρέπει να υπολογίσει κάθε ένα από αυτά.
${W_{ik}}^C$	$Calculator$	$Data$	Η τιμή του στοιχείου $W(i,k)$
${W_{kj}}^C$	$Calculator$	$Data$	Η τιμή του στοιχείου $W(k+1,j)$
$Finish^C$	$Calculator$	$Bool$	Το σήμα το οποίο ενημερώνει τα $Calculators$ ότι όλα τα στοιχεία έχουν σταλθεί.
$W_L(k,l,count)$	$MemoryController$	$(Index, Index, Index)$	Αίτηση προς τον $MemoryController$ για λήψη $count$ στοιχείων με πρώτο το στοιχείο (k,l) από τον πίνακα W.
$W_R(k,l,count)$	$MemoryController$	$(Index, Index, Index)$	Αίτηση προς τον $MemoryController$ για λήψη $count$ στοιχείων με πρώτο το στοιχείο (k,l) από τον πίνακα W.
$Res(k,l)$	$MemoryController$	$Data(Index, Index)$	Η θέση και η τιμή ενός από τα στοιχεία τα οποία υπολογίστηκαν από κάποιο $Calculator$.

Table 31 - Εξόδοι του LCalculator Controller

**Οι είσοδοι που έχουν δύναμη εις την C αντιστοιχούν σε πολλαπλές εισόδους μία για κάθε ένα $Calculator$.*

Βασική λειτουργία των $Controllers$ αποτελεί η παραγωγή των αιτήσεων για ανάκτηση στοιχείων από τους πίνακες. Στην συνέχεια παρουσιάζονται οι αλγόριθμοι που χρησιμοποιεί ο $LCalculatorController$.

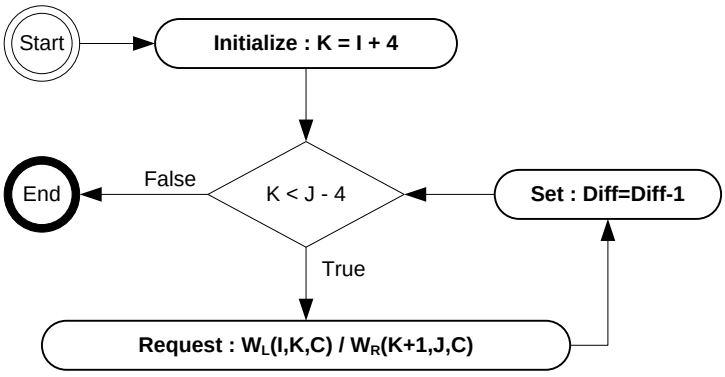


Figure 53 - Αιτήσεις VCalculatorController για στοιχεία από τον πίνακα L

Όπου (I,J) το πρώτο στοιχείο που υπολογίζεται και C ο αριθμός των $LCalculator$ που χρησιμοποιούνται.

Calculation Limiter

Οι τρείς $Controllers$ σχεδιάστηκαν με τέτοιο τρόπο ώστε να μπορούν να λειτουργήσουν ανεξάρτητα ο ένας από τον άλλο και καθώς δεν υπάρχει καμία επικοινωνία μεταξύ τους κάθε $Controller$ προσπαθεί συνεχών να υπολογίσει νέα στοιχεία αγνοώντας την πρόοδο των άλλων. Αποτέλεσμα αυτού είναι η πιθανότητα παραβίασης των εξαρτήσεων μεταξύ των δεδομένων και εξαγωγής λανθασμένων τιμών εφόσον η πρόοδος στον υπολογισμό στοιχείων σε κάθε πίνακα εξαρτάται αποκλειστικά από την σειρά εξυπηρέτησης των αιτημάτων προς των $MemoryController$.

Το πρόβλημα αυτό καλείται να επιλύσει ο $Calculation\ Limiter$ υλοποιώντας μια γέφυρα επικοινωνίας μεταξύ των τριών $Controllers$. Δεχόμενος το σημείο προόδου υπολογισμού για τους τρείς πίνακες ορίζει ένα ασφαλές όριο για κάθε έναν από τους τρείς $Controller$.

Γραφικά μπορούμε να δούμε τον $Limiter$ ως εξής

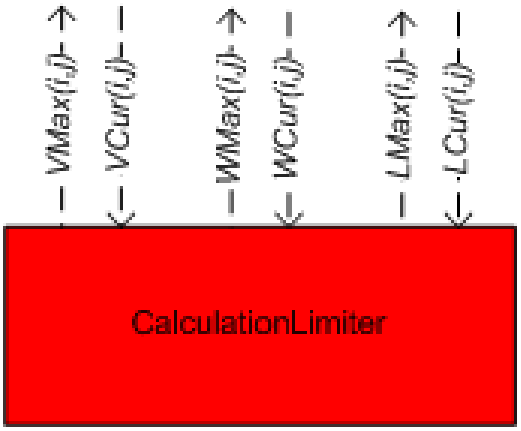


Figure 54 - Σχηματική αναπαράσταση του Calculation Limiter

Οι είσοδοι του $Limiter$ στον παρακάτω πίνακα.

Είσοδος	Σύνδεση	Περιγραφή
	$VCalculationController$	Το μέγιστο σημείο που έχει υπολογιστεί για τον πίνακα V
$WCur(i,j)$	$WCalculationController$	Το μέγιστο σημείο που έχει υπολογιστεί για τον πίνακα W
$LCur(i,j)$	$LCalculationController$	Το μέγιστο σημείο που έχει υπολογιστεί για τον πίνακα L

Table 32 - Είσοδοι του Calculation Limiter

Αντίστοιχα οι έξοδοι του $Limiter$ φαίνονται στον παρακάτω πίνακα.

Έξοδος	Σύνδεση	Περιγραφή
$VMax(i,j)$	$VCalculationController$	Το μέγιστο σημείο υπολογισμού για τον πίνακα V
$WMax(i,j)$	$WCalculationController$	Το μέγιστο σημείο υπολογισμού για τον πίνακα W
$LMax(i,j)$	$LCalculationController$	Το μέγιστο σημείο υπολογισμού για τον πίνακα L

Table 33 – Εξόδοι του Calculation Limiter

Στην συνέχεια θα χρησιμοποιηθούν οι ακόλουθές μεταβλητές για την ευκολότερη χρήση των σημείων προόδου (αριστερά) και σημείων ασφαλούς υπολογισμού (δεξιά).

Μεταβλητή	Αντιστοιχία
$VCur_i$	Την γραμμή της θέσης $VCur(i,j)$
$VCur_j$	Την στήλη της θέσης $VCur(i,j)$
$WCur_i$	Την γραμμή της θέσης $WCur(i,j)$
$WCur_j$	Την στήλη της θέσης $WCur(i,j)$
$LCur_i$	Την γραμμή της θέσης $LCur(i,j)$
$LCur_j$	Την στήλη της θέσης $LCur(i,j)$

Μεταβλητή	Αντιστοιχία
$VMax_i$	Την γραμμή της θέσης $VMax(i,j)$
$VMax_j$	Την στήλη της θέσης $VMax(i,j)$
$WMax_i$	Την γραμμή της θέσης $WMax(i,j)$
$WMax_j$	Την στήλη της θέσης $WMax(i,j)$
$LMax_i$	Την γραμμή της θέσης $LMax(i,j)$
$LMax_j$	Την στήλη της θέσης $LMax(i,j)$

Με βάση τις αντιστοιχίες αυτές θα δοθεί η λογική με την οποία υπολογίζεται το όριο μέχρι το οποίο κάθε $Controller$ μπορεί να υπολογίσει χωρίς πρόβλημα για κάθε πίνακα.

Υπολογισμός ορίου για τον πίνακα V

Ο πίνακας V αξιοποιεί στοιχεία από τους πίνακες L και V. Η σειρά με την οποία υπολογίζονται τα στοιχεία εξασφαλίζει ότι δεν θα τηρηθούν όλες οι εξαρτήσεις μεταξύ των στοιχείων του ίδιου πίνακα, επομένως αρκεί να οριστεί ένα όριο σε σχέση με το σημείο προόδου για τον πίνακα L.

Για τον υπολογισμό του στοιχείου $V(i,j)$ χρειάζεται να έχει υπολογιστεί και το στοιχείο $L(i+1, j-1)$ καθώς λόγω της σειράς υπολογισμού το στοιχείο αυτό είναι το τελευταίο από τα ζητούμενα το οποίο θα υπολογιστεί.

Η αντιστροφή των εξαρτήσεων μας δίνει ότι αν έχει υπολογιστεί το στοιχείο $L(i,j)$ τότε είναι ασφαλές να υπολογιστεί μέχρι και το στοιχείο $V(i-1, j+1)$. Παρόλο που στην γενική περίπτωση το όριο αυτό είναι σωστό, αν το $L(i,j)$ βρίσκεται είτε στην πρώτη γραμμή είτε στην τελευταία στήλη λαμβάνουμε ένα όριο το οποίο δεν βρίσκεται μέσα στα όρια του πίνακα. Έτσι ξεχωρίζουμε τέσσερις διαφορετικές περιπτώσεις

Περιπτώσεις		$VMax_i$	$VMax_j$
$LCur_i = 0$	$LCur_j = L-1$	0	$L-1$
$LCur_i = 0$	$LCur_j < L-1$	$L-2-LCur_j$	$L-1$
$LCur_i > 0$	$LCur_j = L-1$	$LCur_i - 2$	$LCur_j + 0$
$LCur_i > 0$	$LCur_j < L-1$	$LCur_i - 1$	$LCur_j + 1$

Table 34 - Διαμόρφωση ορίου υπολογισμού για τον πίνακα V

Κατά την πρώτη περίπτωση έχουν υπολογιστεί όλα τα στοιχεία του πίνακα L και επομένως μπορούν να υπολογιστούν όλα τα στοιχεία από τον πίνακα V. Στην δεύτερη και τρίτη περίπτωση έχουμε στοιχεία από τον πίνακα τα οποία βρίσκονται στην πρώτη γραμμή και τελευταία στήλη αντίστοιχα. Τα στοιχεία αυτά δεν χρησιμοποιούνται για τον υπολογισμό στοιχείων του πίνακα V και το όριο ορίζεται να είναι ίσο με το όριο που θα είχε το προηγούμενο στοιχείο του πίνακα L. Τέλος έχουμε την γενική περίπτωση σύμφωνα με την οποία υπολογίζεται το όριο.

Υπολογισμός ορίου για τον πίνακα W

Ο πίνακας W αξιοποιεί στοιχεία από τους όλους τους πίνακες. Η σειρά με την οποία υπολογίζονται τα στοιχεία εξασφαλίζει ότι δεν θα τηρηθούν όλες οι εξαρτήσεις μεταξύ των στοιχείων του ίδιου πίνακα, επομένως αρκεί να οριστεί ένα όριο σε σχέση με το σημείο προόδου για τους πίνακες L και V.

Για τον υπολογισμό του στοιχείου $W(i,j)$ χρειάζεται να έχουν υπολογιστεί μέχρι και τα στοιχεία $L(i,j)$ και $V(i,j)$ καθώς λόγο της σειράς υπολογισμού τα στοιχεία αυτό είναι το τελευταία από τα ζητούμενα κάθε πίνακα τα οποία θα υπολογιστούν.

Έτσι για την εξαγωγή του ορίου χρειάζεται να κρατήσουμε το ελάχιστο σημείο προόδου ανάμεσα σε εκείνα των πινάκων L και V. Τα οποία μας δίνουν τις δύο παρακάτω περιπτώσεις.

Περιπτώσεις		$\boldsymbol{WMax_i}$	$\boldsymbol{WMax_j}$
$LCur_i - LCur_j < VCur_i - VCur_j$		$LCur_i$	$LCur_j$
$LCur_i - LCur_j > VCur_i - VCur_j$		$VCur_i$	$VCur_j$
$LCur_i - LCur_j = VCur_i - VCur_j$	$LCur_j < VCur_j$	$LCur_i$	$LCur_j$
$LCur_i - LCur_j = VCur_i - VCur_j$	$LCur_j \geq VCur_j$	$VCur_i$	$VCur_j$

Table 35 - Διαμόρφωση ορίου υπολογισμού για τον πίνακα W

Στην πρώτη περίπτωση ο υπολογισμός στον πίνακα L βρίσκεται σε προηγούμενη διαγώνιο, άρα αποτελεί και το ποίο αυστηρό όριο. Στην δεύτερη ισχύει το αντίθετο με τον υπολογισμό στον πίνακά V να βρίσκεται σε προηγούμενη διαγώνιο. Στις δύο τελευταίες περιπτώσεις ο υπολογισμός και στους δύο πίνακες είναι στην ίδια διαγώνιο και πλέον χρησιμοποιείται το όριο με την μικρότερη αριθμητική τιμή.

Υπολογισμός ορίου για τον πίνακα L

Ο πίνακας L αξιοποιεί στοιχεία μόνο από τον πίνακα W. Για τον υπολογισμό του στοιχείου $L(i,j)$ χρειάζεται να έχει υπολογιστεί και το στοιχείο $W(i+5,j)$ καθώς λόγο της σειράς υπολογισμού το στοιχείο αυτό είναι το τελευταίο από τα ζητούμενα το οποίο θα υπολογιστεί.

Η αντιστροφή των εξαρτήσεων μας δίνει ότι αν έχει υπολογιστεί το στοιχείο $W(i,j)$ τότε είναι ασφαλές να υπολογιστεί μέχρι και το στοιχείο $L(i-5,j)$. Παρόλο που στην γενική περίπτωση το όριο αυτό είναι σωστό, αν το $L(i,j)$ βρίσκεται ανάμεσα στις πρώτες πέντε γραμμές λαμβάνουμε ένα όριο το οποίο δεν βρίσκεται μέσα στα όρια του πίνακα. Έτσι ξεχωρίζουμε δύο διαφορετικές περιπτώσεις

Περιπτώσεις	$\boldsymbol{LMax_i}$	$\boldsymbol{LMax_j}$
$WCur_i < 5$	$max\left(L - 5 - WCur_j + WCur_i\ , 0\right)$	$L-1$
$WCur_i > 4$	$WCur_i - 5$	$WCur_j$

Table 36 - Διαμόρφωση ορίου υπολογισμού για τον πίνακα L

Συγκεντρωτικά τα όρια για όλους τους πίνακες, σε μορφή συνάρτησης, είναι τα παρακάτω.

$$VMax_i = \begin{cases} 0 & , \text{αν } LCur_i = 0 \text{ και } LCur_j = L-1 \\ L-2-LCur_j & , \text{αν } LCur_i = 0 \text{ και } LCur_j < L-1 \\ LCur_i - 2 & , \text{αν } LCur_i > 0 \text{ και } LCur_j = L-1 \\ LCur_i - 1 & , \text{αν } LCur_i > 0 \text{ και } LCur_j < L-1 \end{cases}$$

$$VMax_j = \begin{cases} LCur_j & , \text{αν } LCur_i > 0 \text{ και } LCur_j = L-1 \\ LCur_j + 1 & , \text{αν } LCur_i > 0 \text{ και } LCur_j < L-1 \\ L-1 & , \quad \text{αλλιώς} \end{cases}$$

$$WMax_i = \begin{cases} LCur_i & , \text{ αν } LCur_i - LCur_j < VCur_i - VCur_j \\ VCur_i & , \text{ αν } LCur_i - LCur_j > VCur_i - VCur_j \\ \min(LCur_i, VCur_i) & , \text{ αν } LCur_i - LCur_j = VCur_i - VCur_j \end{cases}$$

$$WMax_j = \begin{cases} LCur_j & , \text{ αν } LCur_i - LCur_j < VCur_i - VCur_j \\ VCur_j & , \text{ αν } LCur_i - LCur_j > VCur_i - VCur_j \\ \min\left(LCur_j, VCur_j\right) & , \text{ αν } LCur_i - LCur_j = VCur_i - VCur_j \end{cases}$$

$$LMax_i = \begin{cases} \max\left(L-5-WCur_j + WCur_i \ , 0\right) & , \text{ αν } WCur_i < 5 \\ WCur_i - 5 & , \text{ αν } WCur_i > 4 \end{cases}$$

$$LMax_j = \begin{cases} L-1 & , \text{ αν } WCur_i < 5 \\ WCur_j & , \text{ αν } WCur_i > 4 \end{cases}$$

Memory Controller

Τέλος έχουμε τον $MemoryController$, ο οποίος αναλαμβάνει την εξυπηρέτηση των αιτήσεων από τους $CalculatorControllers$.

Γραφικά μπορούμε να δούμε τον $MemoryController$ ως εξής

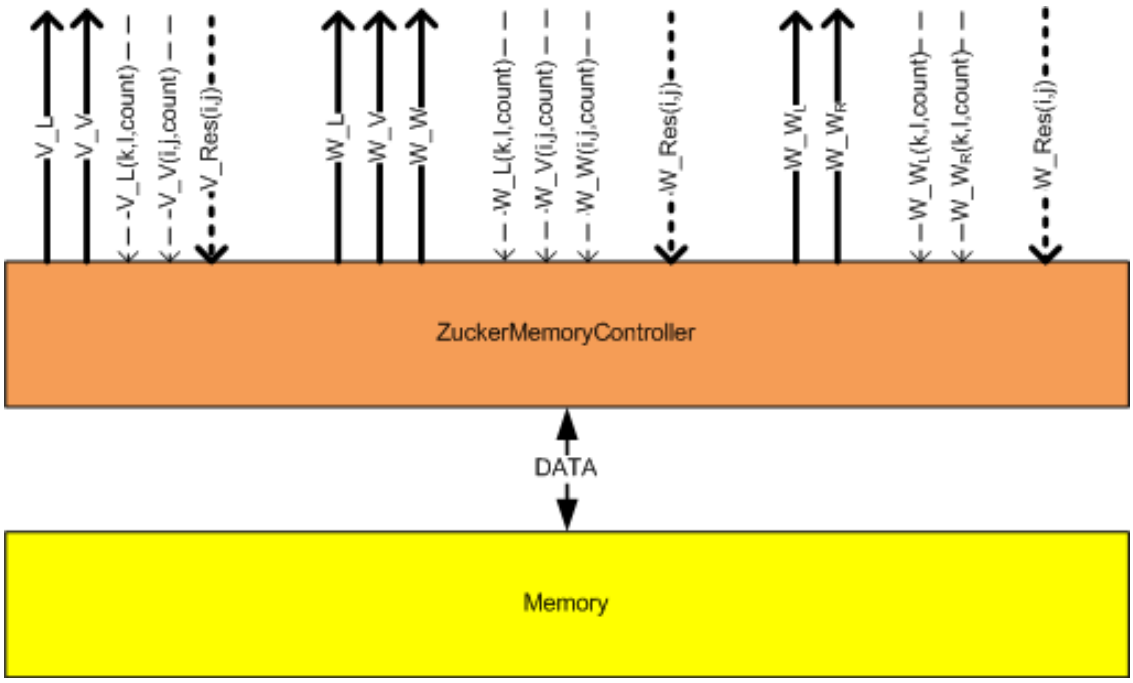


Figure 55 - Σχηματική αναπαράσταση του Memory Controller

Στο παραπάνω σχήμα παρατηρούμε ότι ο $MemoryController$ αποτελείται από δύο τμήματα.

Το πρώτο τμήμα αντιστοιχεί στον $ZuckerMemoryController$. ο οποίος αναλαμβάνει καταρχήν την επιλογή ενός από τα αιτήματα για επεξεργασία και την επικοινωνία με την μνήμη που περιέχει τα δεδομένα έτσι ώστε να ληφθούν τα σωστά δεδομένα. Η επικοινωνία αυτή εξαρτάται από τον τύπο, μέγεθος και πλήθος μνημών η οποία χρησιμοποιείται κάθε φορά.

Το δεύτερο τμήμα είναι το ίδιο το σύστημα μνήμης στο οποίο αποθηκεύονται τα δεδομένα των τριών πινάκων.

Η σειρά με την οποία εξυπηρετούνται τα αιτήματα δεν επηρεάζει την ορθότητα των αποτελεσμάτων, μπορεί όμως σε πολλές περιπτώσεις να μειώσει τον συνολικό χρόνο εκτέλεσης ενσωματώνοντας διάφορες τεχνικές όπως caching, pre-fetching και εξυπηρέτηση πολλών αιτήσεων με κοινά στοιχεία.

Οι είσοδοι του $MemoryController$, καθώς και πληροφορία για τον τύπο και τον $Controller$ με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Είσοδος	$Controller$	Τύπος	Περιγραφή
$V\_L(k,l,count)$	$VCalculator$	$(Index, Index, Index)$	Αίτηση για ανάγνωση δεδομένων από τον πίνακα L
$V\_V(k,l,count)$	$VCalculator$	$(Index, Index, Index)$	Αίτηση για ανάγνωση δεδομένων από τον πίνακα V
$V\_Res(k,l)$	$VCalculator$	$Data(Index, Index)$	Αίτηση για αποθήκευση αποτελεσμάτων στον πίνακα V
$W\_L(k,l,count)$	$WCalculator$	$(Index, Index, Index)$	Αίτηση για ανάγνωση δεδομένων από τον πίνακα L
$W\_W(k,l,count)$	$WCalculator$	$(Index, Index, Index)$	Αίτηση για ανάγνωση δεδομένων από τον πίνακα W
$W\_V(k,l,count)$	$WCalculator$	$(Index, Index, Index)$	Αίτηση για ανάγνωση δεδομένων από τον πίνακα V
$W\_Res(k,l)$	$WCalculator$	$Data(Index, Index)$	Αίτηση για αποθήκευση αποτελεσμάτων στον πίνακα V
$L\_W_L(k,l,count)$	$LCalculator$	$(Index, Index, Index)$	Αίτηση για ανάγνωση δεδομένων από τον πίνακα W
$L\_W_R(k,l,count)$	$LCalculator$	$(Index, Index, Index)$	Αίτηση για ανάγνωση δεδομένων από τον πίνακα W
$L\_Res(k,l)$	$LCalculator$	$Data(Index, Index)$	Αίτηση για αποθήκευση αποτελεσμάτων στον πίνακα L

Table 37 - Εισόδοι του Memory Controller

Αντίστοιχα οι έξοδοι των $LCalculators$, καθώς και πληροφορία για τον τύπο και το σύστημα με το οποίο συνδέονται, φαίνονται στον παρακάτω πίνακα.

Είσοδος	$Controller$	Τύπος	Περιγραφή
V_L	$VCalculator$	$Data$	Δεδομένα ανάγνωσης της αίτησης $V\_L(k,l,count)$ από τον πίνακα L
V_V	$VCalculator$	$Data$	Δεδομένα ανάγνωσης της αίτησης $V\_V(k,l,count)$ από τον πίνακα V
W_L	$WCalculator$	$Data$	Δεδομένα ανάγνωσης της αίτησης $W\_L(k,l,count)$ από τον πίνακα L
W_W	$WCalculator$	$Data$	Δεδομένα ανάγνωσης της αίτησης $W\_W(k,l,count)$ από τον πίνακα W
W_V	$WCalculator$	$Data$	Δεδομένα ανάγνωσης της αίτησης $W\_V(k,l,count)$ από τον πίνακα V
L_W_L	$LCalculator$	$Data$	Δεδομένα ανάγνωσης της αίτησης $L\_W_L(k,l,count)$ από τον πίνακα W
L_W_R	$LCalculator$	$Data$	Δεδομένα ανάγνωσης της αίτησης $L\_W_R(k,l,count)$ από τον πίνακα W

Table 38 - Έξοδοι του Memory Controller

Συνολική σχεδίαση

Στην συνέχεια παρουσιάζεται σχηματικά η συνολική σχεδίαση του συστήματος.

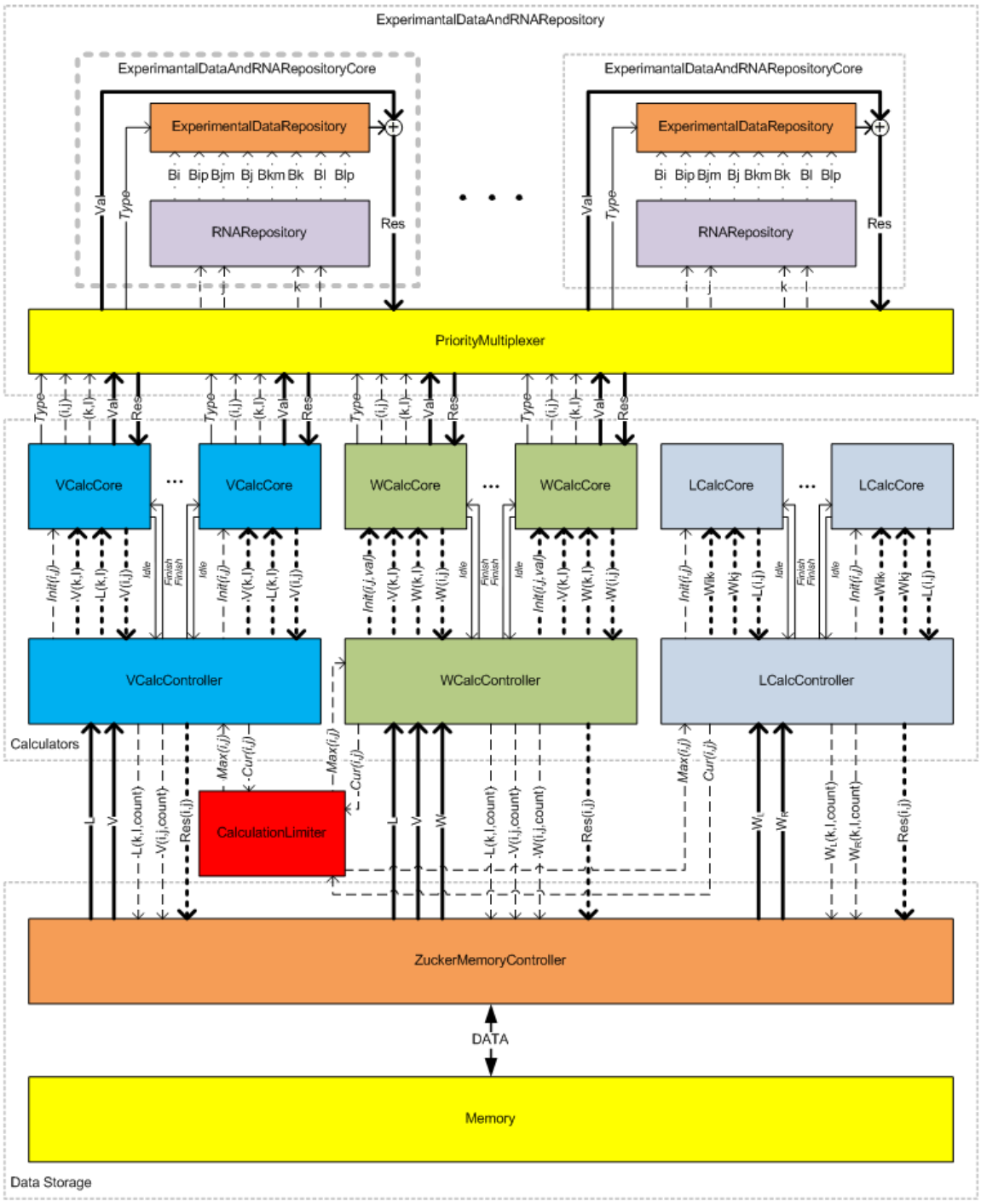


Figure 56 - Σχηματική αναπαράσταση συνολικής σχεδίασης

Στο παραπάνω σχήμα φαίνονται τα επιμέρους τμήματα στα οποία διασπάστηκε ο αλγόριθμος καθώς και η μεταξύ τους σύνδεση.

ΥΛΟΠΟΙΗΣΗ

Αριθμητική συστήματος (Problem Sizing)

Αρχικό στάδιο για την υλοποίηση του συστήματος αποτελεί η αναπαράσταση των τύπων δεδομένων οι οποίοι θα χρησιμοποιηθούν σε κάποια φυσική δομή.

Base

Για την αναπαράσταση των δεδομένων αυτών θα χρησιμοποιηθεί ένας μη προσημασμένος ακέραιος μεγέθους δύο bit και θα υποθέσουμε την ακόλουθη κωδικοποίηση των τιμών αυτών.

Τιμή βάσης	δεκαδική αναπαράσταση	δυαδική αναπαράσταση
A	0	00
C	1	01
G	2	10
U	3	11

Table 39 - Κωδικοποίηση δεδομένων τύπου Base

Pair

Για την αναπαράσταση των δεδομένων αυτών θα χρησιμοποιηθεί ένας μη προσημασμένος ακέραιος μεγέθους τριών bit και θα υποθέσουμε την ακόλουθη κωδικοποίηση των τιμών αυτών.

Τιμή ζεύγους	δεκαδική αναπαράσταση	δυαδική αναπαράσταση
AU	0	000
CG	1	001
GC	2	010
UA	3	011
GU	4	100
UG	5	101
$MISMATCH$	6	110

Table 40 - Κωδικοποίηση δεδομένων τύπου Pair

Και η συνάρτηση $pairing(\cdot,\cdot)$ λαμβάνει την μορφή

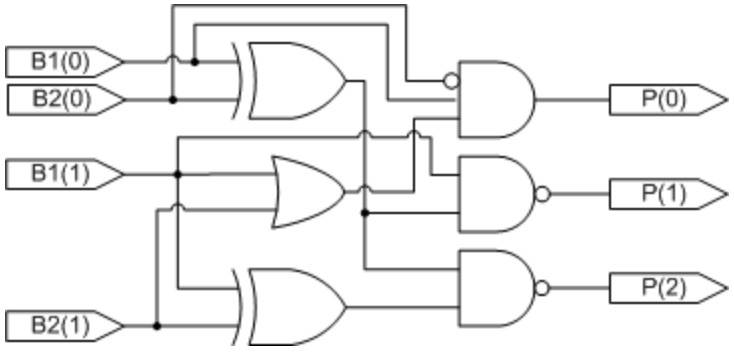


Figure 57 - Σχηματική αναπαράσταση της συνάρτησης pairing(b1,b2)

όπου $B1(1)$ και $B1(0)$ το πρώτο και το δεύτερο bit της πρώτης βάσης, $B2(1)$ και $B2(0)$ το πρώτο και το δεύτερο bit της δεύτερης βάσης και $P(2)$, $P(1)$ και $P(0)$ τα τρία bit του ζεύγους.

Energy

Οι τιμές οι οποίες πρέπει να αναπαρασταθούν είναι δεκαδικές τιμές με ένα δεκαδικό ψηφίο. Η περιγραφή των τιμών αυτών ως αριθμούς σταθερής υποδιαστολής δεν είναι θεμιτή καθώς θα υπάρχει αλλοίωση των δεδομένων, αφού για παράδειγμα ο δεκαδικός αριθμός 0.1, απαιτεί άπειρα ψηφία για την αναπαράσταση του στο δυαδικό σύστημα.

Προς αποφυγή του φαινομένου αυτού οι αριθμοί θα περιγραφούν ως αριθμοί κινητής υποδιαστολής. Έτσι ο αριθμός 3.1 θα αποθηκεύεται σαν 31×10^{-1}, ένα σύνολο δύο ακέραιων τιμών, έναν για την αναπαράσταση των ψηφίων και έναν για τον ορισμό του δεκαδικού σημείου. Σε μια προσπάθεια για την μείωση του κόστους αποθήκευσης και επεξεργασίας των στοιχείων αυτών, θα παραλείψουμε τον δεύτερο ακέραιο, τον εκθετικό δηλαδή όρο 10^{-1}, και θα αναπαραστήσουμε τα στοιχεία μόνο με το ακέραιο μέρος τους, υποθέτοντας ότι όλα έχουν κοινό εκθετικό όρο 10^{-1}.

Η παράλειψη αυτή μπορεί να γίνει χωρίς καμία απώλεια πληροφορίας δεδομένου ότι ο μόνες πράξεις που εκτελούνται κατά την διάρκεια του αλγορίθμου είναι, πρόσθεση και σύγκριση, πράξεις οι οποίες δεν επηρεάζουν την τιμή του εκθετικού όρου.

Για την αναπαράσταση του ακέραιου μέρους των τιμών αυτών θα χρησιμοποιηθεί ένας προσημασμένος ακέραιος μεγέθους εννέα $bits$. Η τιμή αυτή μας δίνει ένα εύρος τιμών

$$[-256,255]$$

Το οποίο υπέρ-καλύπτει το διάστημα τιμών

$$[-3.4,7.4]$$

Το οποίο θέλουμε να αναπαραστήσουμε αφήνοντας και περιθώρια τυχόν αλλαγές των τιμών αυτών.

LEnergy

Η τιμές αυτού του τύπου αντιστοιχούν στην ενέργεια ενός K-loop. Η τιμή αυτή θα πρέπει να είναι σε θέση να κρατήσει το αποτέλεσμα της πρόσθεσης μηδέν έως τεσσάρων στοιχείων τύπου $Energy$, και έτσι για την αναπαράσταση του χρησιμοποιείται ένας προσημασμένος ακέραιος μεγέθους έντεκα $bits$.

Indx

Οι τιμές αυτού του τύπου δεν έχουν σταθερό μέγεθος, αλλά εξαρτώνται από τον αριθμό των βάσεων στο μακρομόριο του RNA. Επειδή τα RNA που καλούνται να υπολογιστούν είναι ανομοιογενή όσων αφορά το μέγεθός τους, η τιμή αυτή επηρεάζει το μέγιστο μέγεθος των RNA που μπορούν τα υπολογιστούν.

Η τιμές αυτού του τύπου αναπαριστώνται ως ένας μη προσημασμένος ακέραιος μεγέθους

$$\log_2(L-1)+1\ bits$$

Όπου L ο συνολικός αριθμός των βάσεων.

Στην συνέχεια ακολουθεί ένας πίνακας ο οποίος παρουσιάζει το αναγκαίο μέγεθος του $Indx$ για τον υπολογισμό κάποιου μεγέθους RNA.

Μέγεθος *RNA* σε βάσεις	**Απαραίτητο μέγεθος *Indx* σε *bits***	**Μέγεθος *RNA* σε βάσεις**	**Απαραίτητο μέγεθος *Indx* σε *bits***
1.000	10	6.000	13
2.000	11	8.000	13
3.000	12	10.000	14
4.000	12	12.000	14

Το μέγιστο μέγεθος RNA που καλείται να επιλύσει ο αλγόριθμος είναι μεγέθους 12000 βάσεων και επομένως χρειάζεται $Indx$ μεγέθους δεκατεσσάρων $bits$.

Data

Αντίστοιχα με τα δεδομένα τύπου $Indx$ και τα στοιχεία $Data$, εξαρτώνται από τον αριθμό των βάσεων στο μακρομόριο του RNA. Η τιμές αυτές εκφράζουν την ενέργεια που μπορεί να λάβει η δευτεροταγής δομή του RNA και ανάλογα με το μέγεθος αυτού μπορεί εν δυνάμει να λάβει πολύ μικρότερες τιμές.

Η χειρότερη περίπτωση δίνεται από τον σχηματισμό $\frac{L-3}{2}$ K-Loops, τον μέγιστο αριθμό για ένα RNA με L βάσεις. Με κάθε K-Loop λαμβάνει την ελάχιστη δυνατή τιμή, -256. Επιπλέον τα δεδομένα αυτού του τύπου πρέπει να υποστηρίζουν και θετικές τιμές για δομές που αποσταθεροποιούν το RNA.

Έτσι για την αναπαράσταση των $Data$ χρησιμοποιήθηκε ένας προσημασμένος ακέραιος μεγέθους τουλάχιστον

$$\log_2(256 \times L - 769) + 2\ bits$$

Όπου L ο συνολικός αριθμός των βάσεων.

Στην συνέχεια ακολουθεί ένας πίνακας ο οποίος παρουσιάζει το αναγκαίο μέγεθος των $Data$ για τον υπολογισμό κάποιου μεγέθους RNA.

Μέγεθος *RNA* σε βάσεις	**Απαραίτητο μέγεθος *Data* σε *bits***	**Μέγεθος *RNA* σε βάσεις**	**Απαραίτητο μέγεθος *Data* σε *bits***
1.000	19	7.000	22
2.000	20	8.000	22
3.000	21	9.000	23
4.000	21	10.000	23
5.000	22	11.000	23
6.000	22	12.000	23

Το μέγιστο μέγεθος RNA που καλείται να επιλύσει ο αλγόριθμος είναι μεγέθους 12000 βάσεων και επομένως χρειάζονται $Data$ μεγέθους είκοσι τριών $bits$.

Addr

Αντίστοιχα με τα δεδομένα τύπου $Indx$ και $Data$ τα στοιχεία $Addr$, εξαρτώνται από τον αριθμό των βάσεων στο μακρομόριο του RNA. Καθώς οι τιμές αυτές εκφράζουν την θέση ενός στοιχείου στην μνήμη χρειάζεται να αριθμήσουν $3 \cdot \frac{(L-3)(L-4)}{2}$ στοιχεία (καθώς τα για κάθε πίνακα τα στοιχεία για τα οποία ισχύει $j < i + 4$) δεν χρησιμοποιούνται.

Έτσι για την αναπαράσταση των $Data$ χρησιμοποιήθηκε ένας προσημασμένος ακέραιος μεγέθους τουλάχιστον

$$\log_2\left(3 \cdot \frac{(L-3)(L-4)}{2} - 1\right) + 1\ bits$$

ctype

Για την αναπαράσταση του τύπου υπολογισμού που θα εκτελέσει το $EDRR$ θα χρησιμοποιηθεί ένας μη προσημασμένος ακέραιος μεγέθους πέντε $bits$ και θα υποθέσουμε την ακόλουθη κωδικοποίηση των τιμών αυτών.

ctype	δεκαδική αναπαράσταση	δυαδική αναπαράσταση
NoCalculation	0	00000
HairpinLoop	1	00001
StackedPair	*4*	00100
BulgeLeft	6	00110
BulgeRight	5	00101
InteriorLoop	7	00111
InteriorLoop1x1	12	01100
InteriorLoop1x2	13	01101
InteriorLoop2x1	14	01110
InteriorLoop2x2	15	01111
JoinStructure1x1	28	11100
JoinStructure1x2	29	11101
JoinStructure2x1	30	11110
JoinStructure2x2	31	11111
DangleLeft	9	01001
DangleRight	10	01010
DangleBoth	11	01000
AppendBase	23	10111
NULLCalculation	16	10000

Table 41 - Κωδικοποίηση δεδομένων τύπου ctype

Ολοκληρώνοντας έτσι την αναπαράσταση των δεδομένων.

Αποθήκευση υπό-προβλημάτων

Σημαντικό πρόβλημα αποτελεί η αποθήκευση των υπό-προβλημάτων καθώς καταλαμβάνουν μεγάλη ποσότητα μνήμης. Στον παρακάτω πίνακα παρουσιάζεται ο αριθμός των στοιχείων που χρειάζεται να αποθηκευτούν σε σχέση με τον αριθμό των βάσεων στο μακρομόριο του RNA για δύο περιπτώσεις, την περίπτωση που αποθηκεύεται ολόκληρος ο πίνακα και την περίπτωση όπου αποθηκεύονται μόνο τα χρήσιμα στοιχεία, εκείνα δηλαδή για τα οποία ισχύει $j \geq i + 4$.

Μέγεθος RNA σε βάσεις (L)	Αριθμός στοιχείων ολόκληρου του πίνακα $(3 \cdot L^2)$	*Χώρος αποθήκευσης ολόκληρου του πίνακα*	Αριθμός χρήσιμων στοιχείων $\left(3 \cdot \frac{(L-3)(L-4)}{2}\right)$	*Χώρος αποθήκευσης χρήσιμων στοιχείων*
1.000	3.000.000	*11,45 MB*	1.486.530	*5,68 MB*
2.000	12.000.000	*45,78 MB*	5.973.030	*22,79 MB*
3.000	27.000.000	*103,00 MB*	13.459.530	*51,35 MB*
4.000	48.000.000	*183,11 MB*	23.946.030	*91,35 MB*
5.000	75.000.000	*286,11 MB*	37.432.530	*142,80 MB*
6.000	108.000.000	*411,99 MB*	53.919.030	*205,69 MB*
7.000	147.000.000	*560,77 MB*	73.405.530	*280,02 MB*
8.000	192.000.000	*732,43 MB*	95.892.030	*365,80 MB*
9.000	243.000.000	*926,98 MB*	121.378.530	*463,03 MB*
10.000	300.000.000	*1.144,41 MB*	149.865.030	*571,67 MB*
11.000	363.000.000	*1.384,74 MB*	181.351.530	*691,81 MB*
12.000	432.000.000	*1.647,95 MB*	215.838.030	*823,36 MB*

Table 42 - Αριθμός στοιχείων προς αποθήκευση σε σχέση με το μέγεθος του RNA

Παρατηρείται μια σημαντική μείωση του απαραίτητης μνήμης στην περίπτωση όπου αποθηκεύονται μόνο τα χρήσιμα στοιχεία της τάξεως του 50%. Έτσι τα στοιχεία των πινάκων θα αναδιαταχθούν σε ένα διάνυσμα, με τέτοιο τρόπο έτσι ώστε να διατηρηθούν μόνο τα στοιχεία που χρειάζονται.

Αναδιάταξη πινάκων

Η αναδιάταξη των πινάκων εκτός από την μείωση του απαραίτητου χώρου αποθήκευσης μπορεί να συμβάλει και στην χρονική βελτίωση του αλγορίθμου τοποθετώντας τα στοιχεία που πρόκειται να διαβαστούν ανά πάσα στιγμή σε σειρά επιτρέποντας στον $MemoryController$ μια ποίο αποδοτική ανάγνωση και εγγραφή στοιχείων.

Στο προηγούμενο κεφάλαιο η ανάγνωση και εγγραφή στοιχείων ορίστηκε να γίνεται πάνω στα στοιχεία των διαγωνίων, οπότε μια βέλτιστη αναδιάταξη θα τοποθετούσε τα στοιχεία των διαγωνίων σε σειρά.

Η αναδιάταξη του πίνακα γίνεται μέσω της συνάρτησης

$$Pos(i,j) = (j - i - 4) * L + \frac{(j-i)}{2} - \frac{(j-i)\cdot(j-i)}{2} + i + 6$$

Όπου (i,j) η θέση του στοιχείου στον πίνακα και $Pos(i,j)$ η θέση του στοιχείου στο αναδιατεταγμένο διάνυσμα και L ο μέγιστος αριθμός βάσεων για κάποιο RNA.

Στον παρακάτω πίνακα παρουσιάζονται οι θέσεις των στοιχείων του πίνακα στο αναδιατεταγμένο διάνυσμα.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	-	-	-	-	0	12	23	33	42	50	57	63	68	72	75	77
1	-	-	-	-	-	1	13	24	34	43	51	58	64	69	73	76
2	-	-	-	-	-	-	2	14	25	35	44	52	59	65	70	74
3	-	-	-	-	-	-	-	3	15	26	36	45	53	60	66	71
4	-	-	-	-	-	-	-	-	4	16	27	37	46	54	61	67
5	-	-	-	-	-	-	-	-	-	5	17	28	38	47	55	62
6	-	-	-	-	-	-	-	-	-	-	6	18	29	39	48	56
7	-	-	-	-	-	-	-	-	-	-	-	7	19	30	40	49
8	-	-	-	-	-	-	-	-	-	-	-	-	8	20	31	41
9	-	-	-	-	-	-	-	-	-	-	-	-	-	9	21	32
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	10	22
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	11
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Figure 58 - Θέσεις στοιχείων στο αναδιατεταγμένο διάνυσμα

Στο παραπάνω σχήμα έχουν αριθμηθεί οι γραμμές και οι στήλες έτσι ώστε να γίνονται εμφανείς οι τιμές των (i, j). Τα στοιχεία που δεν περιέχουν κάποια τιμή δεν υπάρχουν στο αναδιατεταγμένο διάνυσμα καθώς δεν χρησιμοποιούνται από τον αλγόριθμο, ενώ για τα υπόλοιπα παρατηρείτε πως η αναδιάταξη αυτή διατηρεί τα στοιχεία των διαγωνίων σε διπλανά σημεία.

Παράλληλα με την μείωση της ανάγκης για αποθηκευτικό χώρο και της αύξησης της χωρικής τοπικότητας, εισάγεται και η ανάγκη του υπολογισμού της θέσης του πρώτου στοιχείου κάθε αίτησης προς τον $MemoryController$ στο αναδιατεταγμένο διάνυσμα.

Ο αλγόριθμος δεν πρακτικά δυνατόν να λειτουργήσει με τις νέες διευθύνσεις στοιχείων και εξακολουθεί να στέλνει αιτήματα τις μορφής $(i, j, count)$. Έτσι θα πρέπει κάθε φορά να υπολογίζεται η ποσότητα $Pos(i, j)$ και στην συνέχεια να εκτελείται η ανάγνωση/εγγραφή σειριακά από εκείνη την θέση.

Το κόστος αυτής της πράξης είναι αμελητέο καθώς μπορεί να εκτελεστεί σε ένα ξεχωριστό pipeline επίπεδο καθυστερώντας τον συνολικό χρόνο εκτέλεσης κατά έναν κύκλο ρολογιού.

Σύστημα Αναφοράς

Για την υλοποίηση του αλγορίθμου χρησιμοποιήθηκε ένα σύστημα αναδιατασσόμενης λογικής και συγκεκριμένα το board Virtex-2 Pro XC2VP30.

Το Board μας προσφέρει

Στοιχείο	Ποσότητα	Περιγραφή
Logic Cells	13.696	Κάθε cell περιέχει δύο Flip-Flop και δύο LUT τεσσάρων εισόδων. Τα στοιχεία αυτά μπορούν να προγραμματιστούν έτσι ώστε να εκτελέσουν οποιαδήποτε (σχεδόν) λειτουργία.
Block Rams	2.448KB	Η μνήμη είναι οργανωμένη σε 136 Block Rams των 18KB το κάθε ένα. Τα στοιχεία αυτά χρησιμοποιούνται για την αποθήκευση δεδομένων.

Η μνήμη που παρέχει το board είναι αρκετή για την κάλυψη των απαιτήσεων για το $EDRR$ αλλά όχι και για την αποθήκευση των στοιχείων για τους τρείς πίνακες. Όπου για την αποθήκευση τους θα πρέπει να χρησιμοποιηθεί κάποια εξωτερική μνήμη.

Αξιοποίηση διαθέσιμου χώρου

Ο κύριος παράγοντας ο οποίος επηρεάζει την ταχύτητα του αλγορίθμου είναι ο αριθμός των παράλληλων υπολογισμών που μπορεί να εκτελέσει ο αλγόριθμος.

Για κάθε παράλληλο υπολογισμό χρειάζεται και ένας πυρήνας υπολογισμού στο $EDRR$, για την υλοποίηση του οποίου χρειάζονται να αποθηκευτούν οι παρακάτω ποσότητες

Στοιχείο	Αριθμός BRAM
Stacking Energies	1
Terminal mismatch stacking energies (hairpin loops)	
Single base stacking energies (dangle)	
Loop destabilizing energies	3
1x1 Loops	1
1x2 Loops	4
2x2 Loops	16
Interior Loop	1
RNA	4

Table 43 - Αριθμός απαιτούμενων BRAM για κάθε ενέργεια

Με χρήση Dual Port BRAM οι μνήμες αυτές μπορούν να επαναχρησιμοποιηθούν για την υλοποίηση ενός δεύτερου πυρήνα, με εξαίρεση τα BRAM στα οποία αποθηκεύονται τα Interior Loops και το RNA καθώς ήδη χρησιμοποιούνται και τα δύο ports.

Έτσι χρειάζονται 35 BRAM ανά δύο πυρήνες, ορίζοντας το μέγιστο αριθμό παράλληλων υπολογισμών σε έξι. Χρησιμοποιήθηκαν όμως μόνο τέσσερις, καθώς οι απαιτήσεις σε λογική ξεπερνούσαν τα όρια του board, αξιοποιώντας έτσι το 94% των διαθέσιμων Logic Cells.

Ο προγραμματισμός του έγινε μέσω των εργαλείων CAD της Xilinx και με χρήση της γλώσσας περιγραφής υλικού VHDL.

Κατανάλωση πόρων για διάφορα επίπεδα παραλληλησμού.

Στον παρακάτω πίνακα παρουσιάζεται η κατανάλωση των πόρων στο Board Virtex-2 Pro XC2VP30 για έναν, δύο και τέσσερις παράλληλους υπολογισμούς.

Πόροι		**Περίπτωση χωρίς παραλληλισμό**	**Δύο παράλληλοι υπολογισμοί.**	**Τέσσερις παράλληλοι υπολογισμοί.**
Brams	EDRR	30 / 22%	35 / 26%	70 / 51%
	Πίνακες	106 / 78%	101 / 74%	66 / 49%
	Σύνολο	136 / 100%	136 / 100%	136 / 100%
Slices		*61%*	*72%*	*94%*
Ρολόι		106 MHZ	103 MHz	101 MHz
Μέγιστο μέγεθος RNA		272 Βάσεις	266 Βάσεις	215 Βάσεις

Table 44 – Κατανάλωση πόρων στο Board Virtex-2 Pro XC2VP30

Το σύστημα αυτό δεν διαθέτει κάποια εξωτερική μνήμη και έτσι τα BRams χωρίστηκαν για την υλοποίηση των πυρήνων υπολογισμού στο EDRR και την αποθήκευση των πινάκων V,W και L. Έτσι με την αύξηση των παράλληλων υπολογισμών μειώνεται ο χώρος για τους τρείς πίνακες με αποτέλεσμα να μειωθεί και το μέγεθος των RNA τα οποία μπορούν να επιλυθούν.

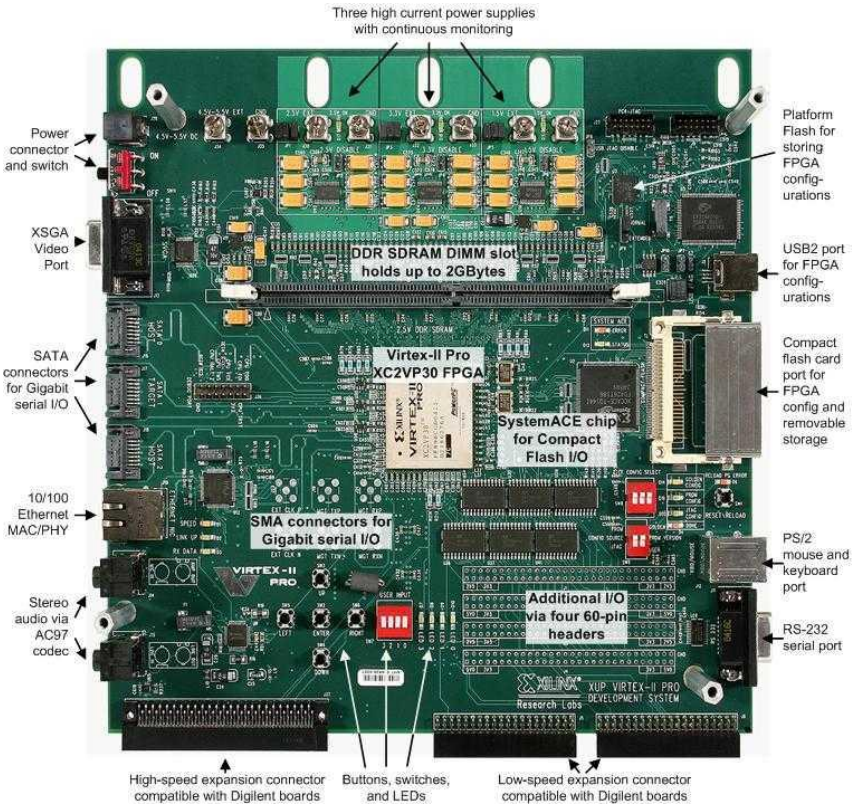


Figure 59 – XUP Virtex II Pro Board

ΑΠΟΔΟΣΗ ΣΥΣΤΗΜΑΤΟΣ

Η μέτρηση της απόδοσης του συστήματος έγινε τόσο στο ίδιο το σύστημα μέσω εργαλείων προσομοίωσης όσο και σε λογισμικό για την σύγκριση των αποτελεσμάτων.

Απόδοση αλγορίθμου σε υλικό

Για την σχεδίαση το ρολόι του συστήματος ήταν στα 100MHz και ο αλγόριθμος εκτελέστηκε για διάφορα μεγέθη και αριθμό παράλληλων υπολογισμών, C, ενώ τα αποτελέσματα λήφθηκαν από εκτέλεσή του σε προσομοιωτή.

Μέγεθος *RNA* σε βάσεις	**Χρόνος εκτέλεσης Hardware (1) (ms)**	**Χρόνος εκτέλεσης Hardware (2) (ms)**	**Χρόνος εκτέλεσης Hardware (4) (ms)**
50	26,81	16,41	11,22
75	134,75	78,18	49,93
100	423,93	238,44	145,75
125	1.031,87	568,65	337,14
150	2.135,11	1.159,84	672,34
175	3.949,30	2.122,56	1.209,37
200	6.729,11	3.586,87	2.015,99

Table 45 - Χρόνοι εκτέλεσης υλικού

Γραφικά οι τιμές αυτές αντιστοιχούν στην παρακάτω γραφική παράσταση

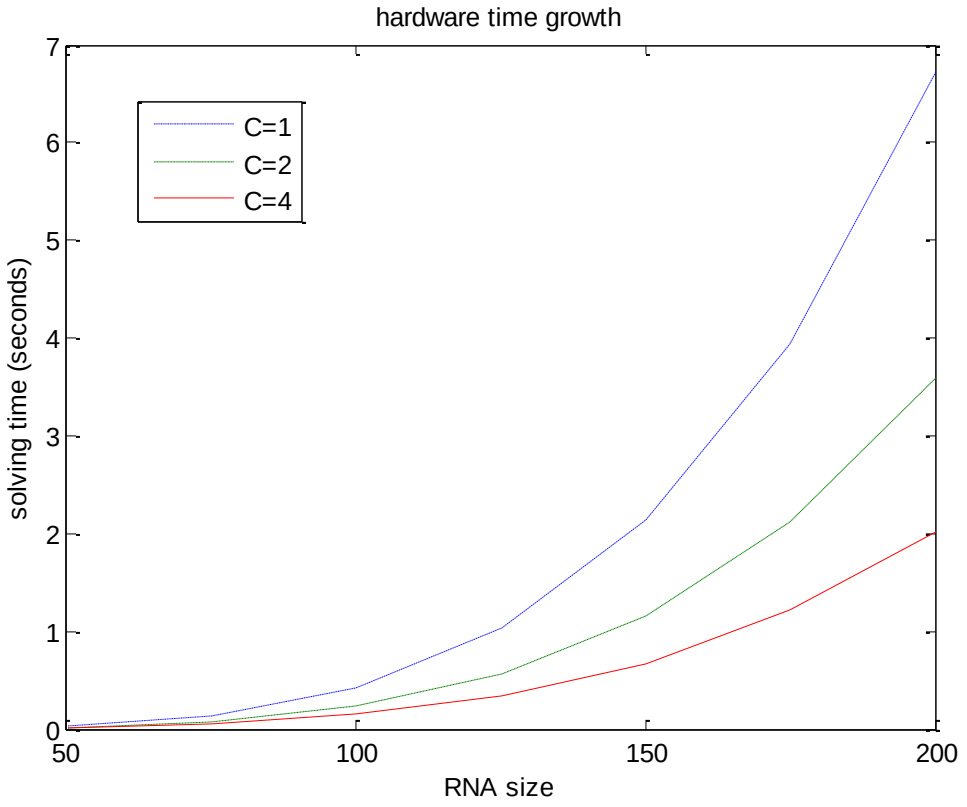


Figure 60 - Ρυθμός αύξησης χρόνου εκτέλεσης (hardware)

Υπολογισμός speed up σε σχέση με την αύξηση των παράλληλων υπολογισμών.

Στην συνέχεια παρουσιάζεται το speed up του υλικού με την αύξηση των παράλληλων υπολογισμών σε σχέση με την περίπτωση χωρίς παράλληλο υπολογισμό, σε μια προσπάθεια να φανεί το κέρδος απο την αύξηση των παράλληλων υπολογισμών.

Μέγεθος *RNA* σε βάσεις	speed up Hardware (2)	speed up Hardware (4)
50	1,6338	2,3895
75	1,7236	2,6988
100	1,7779	2,9086
125	1,8146	3,0607
150	1,8409	3,1756
175	1,8606	3,2656
200	1,8760	3,3379

Table 46 - Σύγκριση διαφορετικών επιπέδων παραλληλισμού

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Χρόνος εκτέλεσης Hardware (1)}}{\text{Χρόνος εκτέλεσης Hardware } (C)}$$

Τα speed up φαίνεται στην παρακάτω γραφική παράσταση

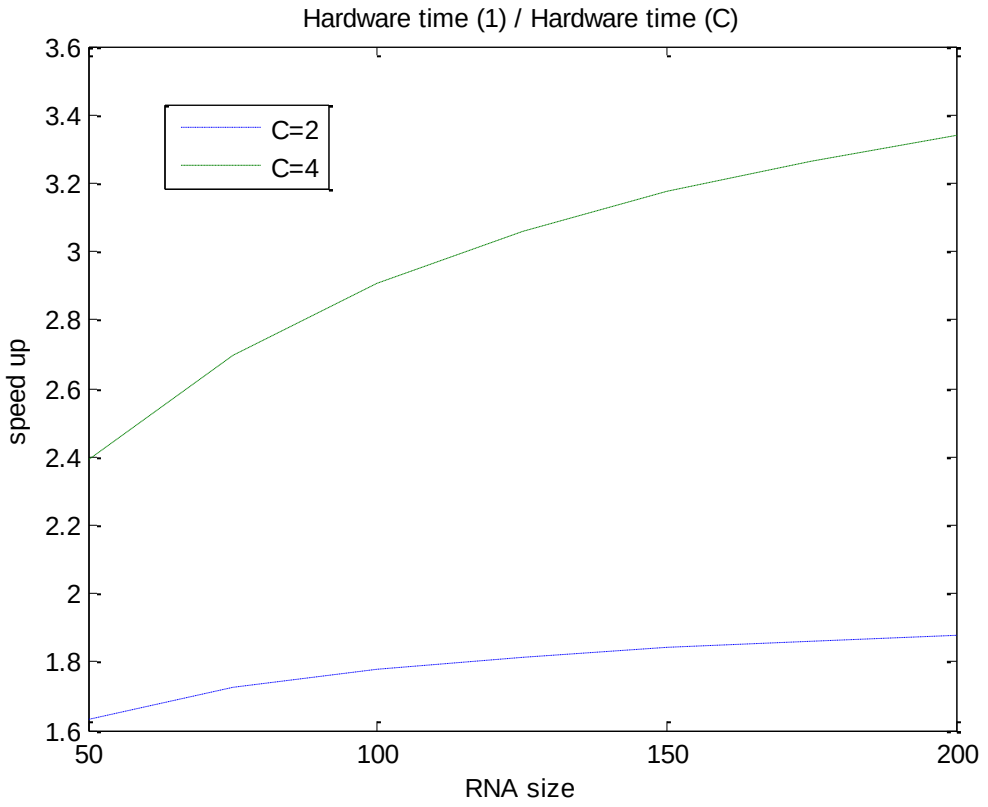


Figure 61 - speed up διαφορετικών επιπέδων παραλληλισμού

Απόδοση αλγορίθμου στο λογισμικό αναφοράς

Το λογισμικό αυτό δημιουργήθηκε με σκοπό την κατανόηση του αλγορίθμου και την πιστοποίηση της λειτουργίας του υλικού. Πρόκειται για μια μη βελτιστοποιημένη έκδοση του αλγορίθμου.

Ο αλγόριθμος εκτελέστηκε σε έναν προσωπικό υπολογιστή με επεξεργαστή Intel Pentium D στα 2.80 GHz και 2 GB μνήμη Ram. Το πρόγραμμα εκτελέστηκε δέκα φορές για διάφορα μεγέθη RNA και λήφθηκαν οι παρακάτω χρόνοι.

Μέγεθος *RNA* σε βάσεις	**Ελάχιστος χρόνος εκτέλεσης software (ms)**	**Μέσος χρόνος εκτέλεσης software (ms)**	**Μέγιστος χρόνος εκτέλεσης software (ms)**
50	15	**15,6**	16
75	281	**284,4**	297
100	1.391	**1.404,7**	1.422
125	4.156	**4.179,7**	4.250
150	9.625	**9.653,1**	9.796
175	19.187	**19.265,5**	19.703
200	34.407	**34.715,7**	35.313

Table 47 - Χρόνοι εκτέλεσης λογισμικού

Γραφικά οι τιμές αυτές αντιστοιχούν στην παρακάτω γραφική παράσταση

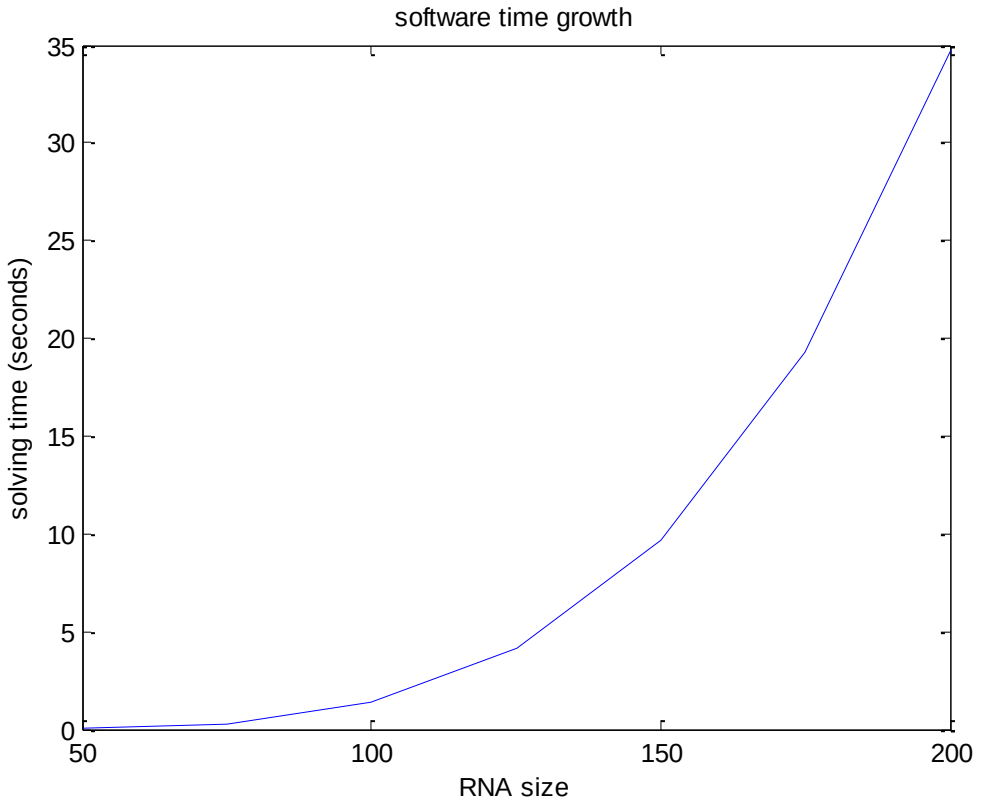


Figure 62 - Ρυθμός αύξησης χρόνου εκτέλεσης (software)

Σύγκριση με το υλικό για την περίπτωση χωρίς παραλληλισμό

Μέγεθος *RNA* σε βάσεις	Μέσος χρόνος εκτέλεσης software (ms)	Χρόνος εκτέλεσης Hardware (1) (ms)	speedup
50	15,6	26,81	0,5819
75	284,4	134,75	2,1106
100	1.404,7	423,93	3,3135
125	4.179,7	1.031,87	4,0506
150	9.653,1	2.135,11	4,5211
175	19.265,5	3.949,30	4,8782
200	34.715,7	6.729,11	5,1590

Table 48 - Σύγκριση λογισμικού με υλικό (1)

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Χρόνος εκτέλεσης Software}}{\text{Χρόνος εκτέλεσης Hardware}}$$

Το speed up φαίνεται στην παρακάτω γραφική παράσταση

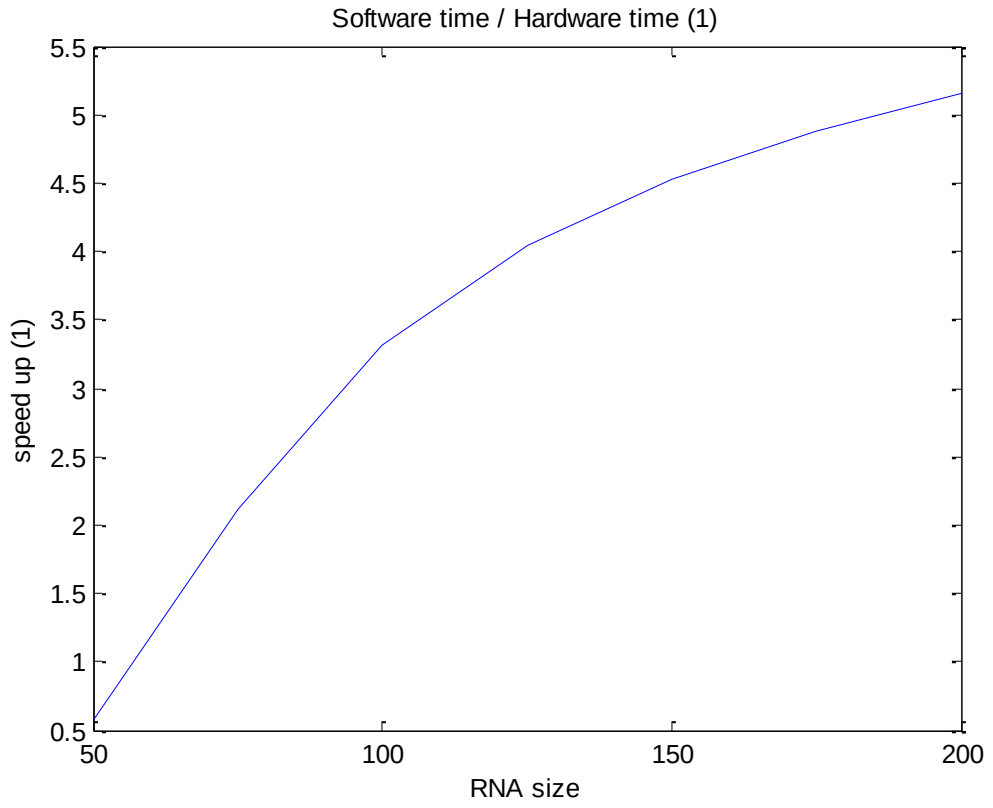


Figure 63 - speed up υλικού (1) σε σχέση με το λογισμικό

Σύγκριση με το υλικό για την περίπτωση με δύο παράλληλους υπολογισμούς

Μέγεθος *RNA* σε βάσεις	Μέσος χρόνος εκτέλεσης software (ms)	Χρόνος εκτέλεσης Hardware (2) (ms)	speedup
50	15,6	16,41	0,9506
75	284,4	78,18	3,6378
100	1.404,7	238,44	5,8912
125	4.179,7	568,65	7,3502
150	9.653,1	1.159,84	8,3228
175	19.265,5	2.122,56	9,0765
200	34.715,7	3.586,87	9,6785

Table 49 - Σύγκριση λογισμικού με υλικό (2)

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Χρόνος εκτέλεσης Software}}{\text{Χρόνος εκτέλεσης Hardware}}$$

Το speed up φαίνεται στην παρακάτω γραφική παράσταση

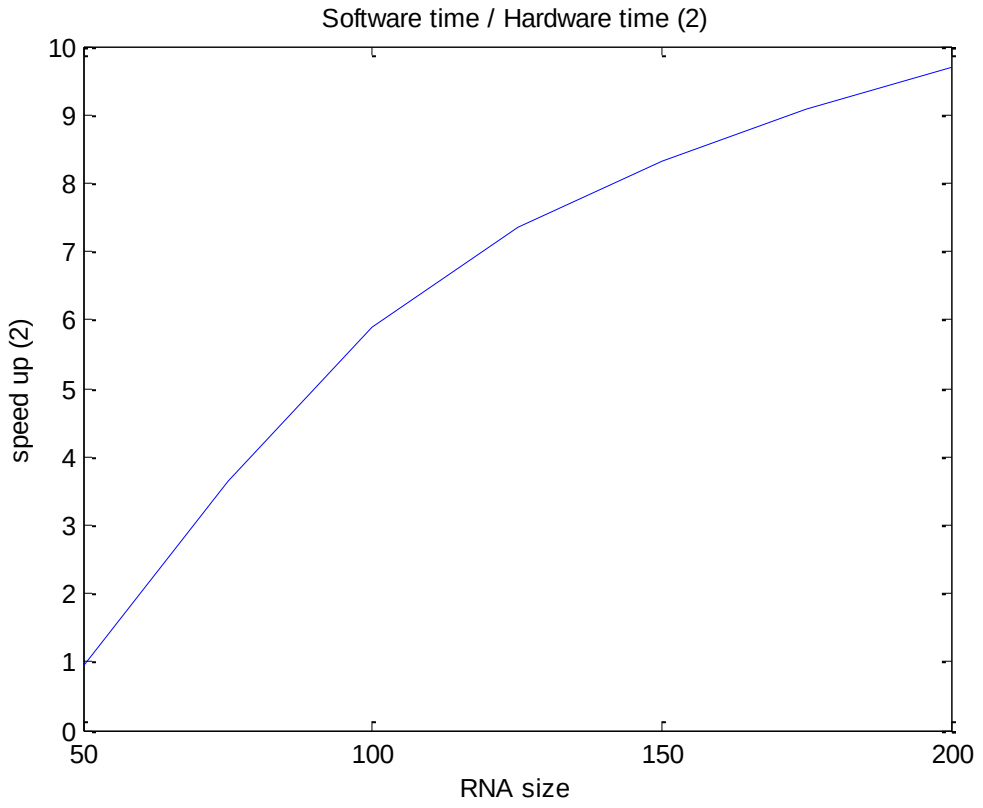


Figure 64 - speed up υλικού (2) σε σχέση με το λογισμικό

Σύγκριση με το υλικό για την περίπτωση με τέσσερις παράλληλους υπολογισμούς

Μέγεθος *RNA* σε βάσεις	Μέσος χρόνος εκτέλεσης software (ms)	Χρόνος εκτέλεσης Hardware (4) (ms)	speedup
50	15,6	11,22	1,3904
75	284,4	49,93	5,6960
100	1.404,7	145,75	9,6377
125	4.179,7	337,14	12,3975
150	9.653,1	672,34	14,3575
175	19.265,5	1.209,37	15,9302
200	34.715,7	2.015,99	17,2202

Table 50 - Σύγκριση λογισμικού με υλικό (4)

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Χρόνος εκτέλεσης Software}}{\text{Χρόνος εκτέλεσης Hardware}}$$

Το speed up φαίνεται στην παρακάτω γραφική παράσταση

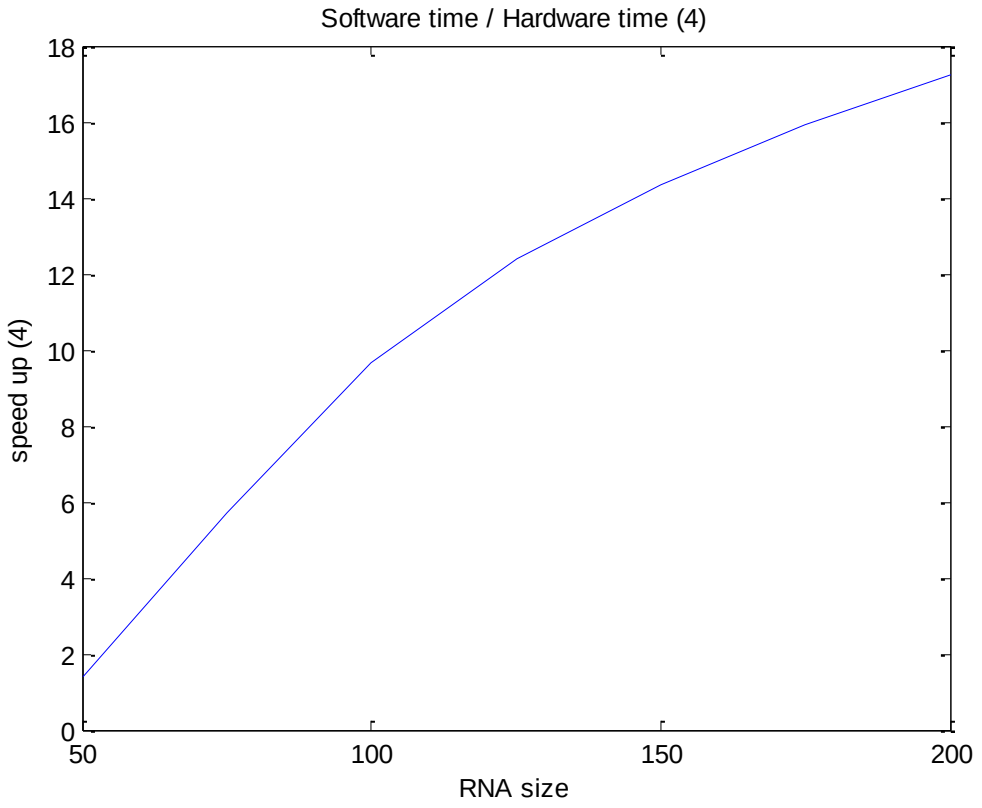


Figure 65 - speed up υλικού (4) σε σχέση με το λογισμικό

Απόδοση αλγορίθμου από το πακέτο UNAFold

Το πακέτο UNAFold [13] περιέχει μια πλήρως βελτιστοποιημένη έκδοση του αλγορίθμου του Zucker.

Ο αλγόριθμος εκτελέστηκε σε έναν προσωπικό υπολογιστή με επεξεργαστή Intel Pentium D στα 2.80 GHz και 2 GB μνήμη Ram. Το πρόγραμμα εκτελέστηκε δέκα φορές για διάφορα μεγέθη RNA και λήφθηκαν οι παρακάτω χρόνοι.

Μέγεθος *RNA* σε βάσεις	**Ελάχιστος χρόνος εκτέλεσης software (ms)**	**Μέσος χρόνος εκτέλεσης software (ms)**	**Μέγιστος χρόνος εκτέλεσης software (ms)**
50	109	**110,9**	125
75	109	**120,3**	140
100	171	**196,9**	235
125	234	**239,3**	250
150	547	**565,7**	578
175	890	**901,6**	921
200	1.609	**1.628,2**	1.656

Table 51 - Χρόνοι εκτέλεσης λογισμικού

Γραφικά οι τιμές αυτές αντιστοιχούν στην παρακάτω γραφική παράσταση

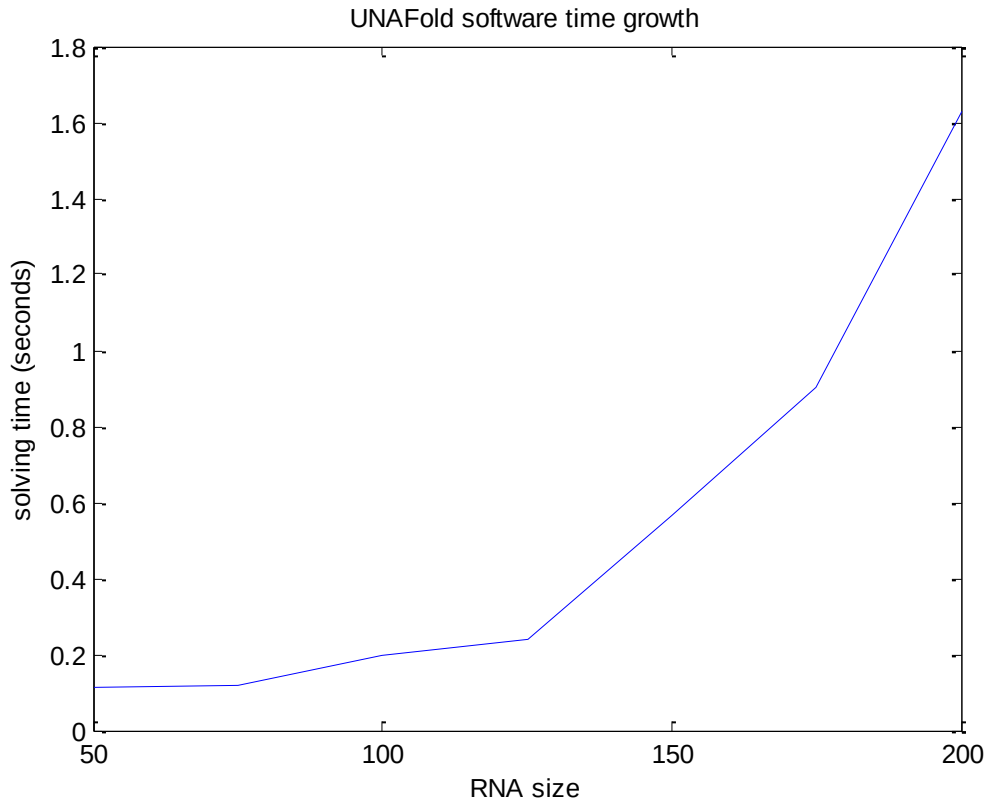


Figure 66 - Ρυθμός αύξησης χρόνου εκτέλεσης (software)

Σύγκριση με το υλικό για την περίπτωση χωρίς παραλληλισμό

Μέγεθος *RNA* σε βάσεις	Μέσος χρόνος εκτέλεσης software (UNAFold) (ms)	Χρόνος εκτέλεσης Hardware (1) (ms)	speedup
50	110,9	26,81	4,1365
75	120,3	134,75	0,8928
100	196,9	423,93	0,4645
125	239,3	1.031,87	0,2319
150	565,7	2.135,11	0,2650
175	901,6	3.949,30	0,2283
200	1.628,2	6.729,11	0,2420

Table 52 - Σύγκριση λογισμικού με υλικό (1)

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Χρόνος εκτέλεσης Software}}{\text{Χρόνος εκτέλεσης Hardware}}$$

Το speed up φαίνεται στην παρακάτω γραφική παράσταση

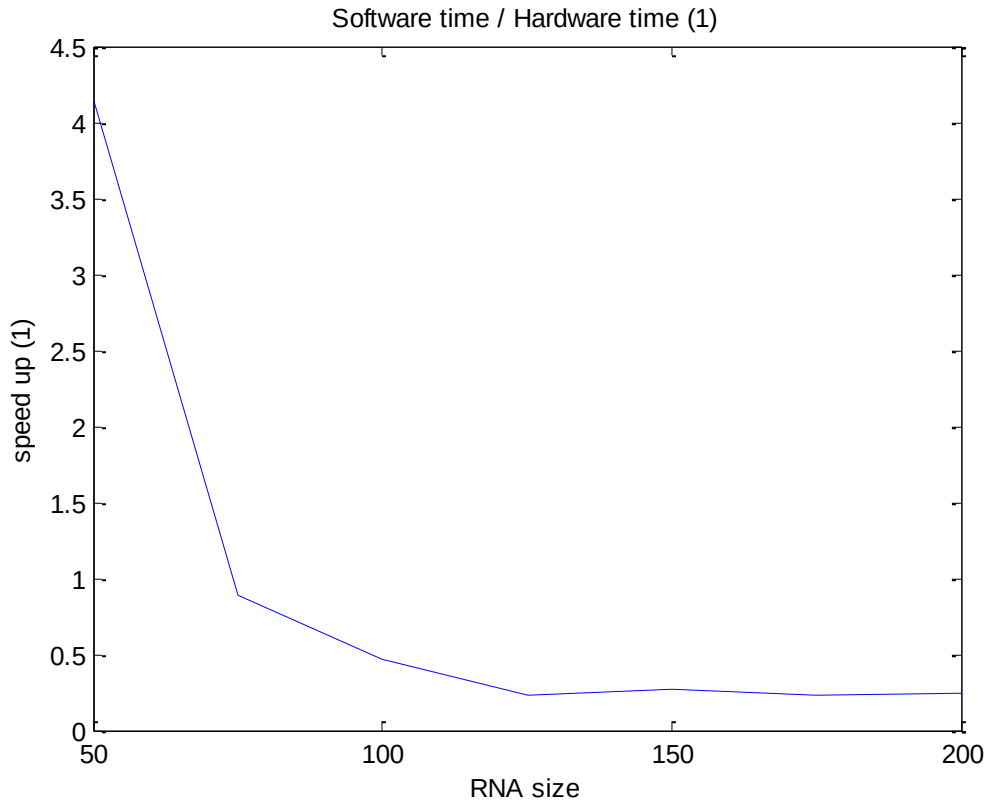


Figure 67 - speed up υλικού (1) σε σχέση με το λογισμικό

Σύγκριση με το υλικό για την περίπτωση με δύο παράλληλους υπολογισμούς

Μέγεθος *RNA* σε βάσεις	Μέσος χρόνος εκτέλεσης software (UNAFold) (ms)	Χρόνος εκτέλεσης Hardware (2) (ms)	speedup
50	110,9	16,41	6,7581
75	120,3	78,18	1,5388
100	196,9	238,44	0,8258
125	239,3	568,65	0,4208
150	565,7	1.159,84	0,4877
175	901,6	2.122,56	0,4248
200	1.628,2	3.586,87	0,4539

Table 53 - Σύγκριση λογισμικού με υλικό (2)

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Χρόνος εκτέλεσης Software}}{\text{Χρόνος εκτέλεσης Hardware}}$$

Το speed up φαίνεται στην παρακάτω γραφική παράσταση

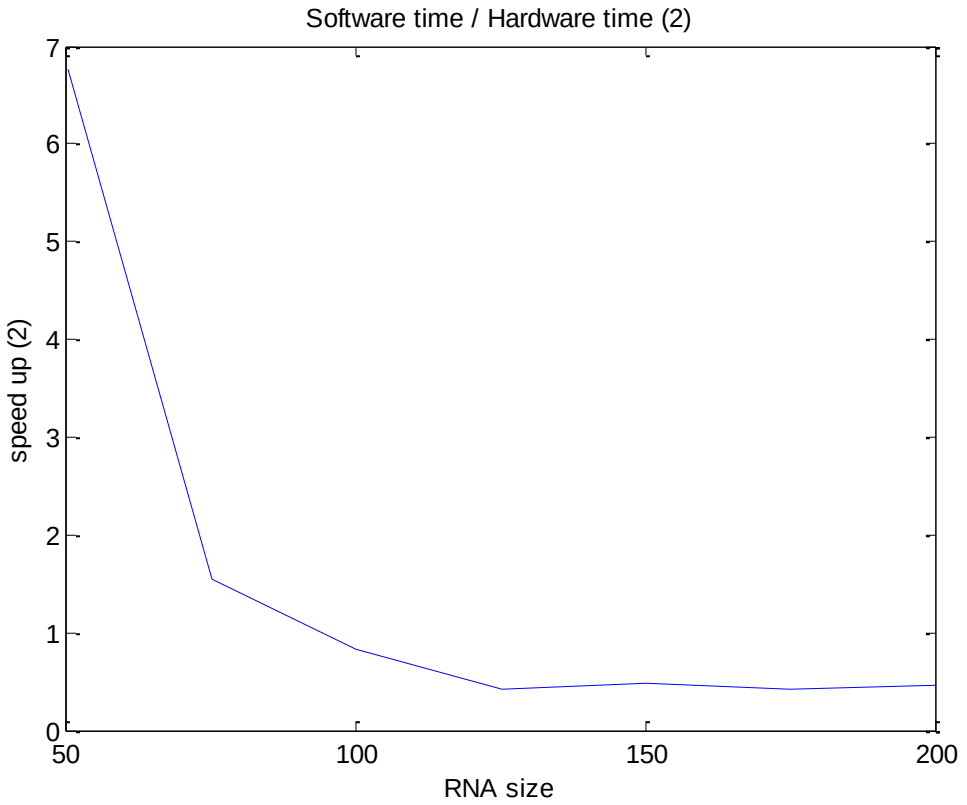


Figure 68 - speed up υλικού (2) σε σχέση με το λογισμικό

Σύγκριση με το υλικό για την περίπτωση με τέσσερις παράλληλους υπολογισμούς

Μέγεθος *RNA* σε βάσεις	Μέσος χρόνος εκτέλεσης software (UNAFold) (ms)	Χρόνος εκτέλεσης Hardware (4) (ms)	speedup
50	110,9	11,22	9,8841
75	120,3	49,93	2,4094
100	196,9	145,75	1,3509
125	239,3	337,14	0,7098
150	565,7	672,34	0,8414
175	901,6	1.209,37	0,7455
200	1.628,2	2.015,99	0,8076

Table 54 - Σύγκριση λογισμικού με υλικό (4)

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Χρόνος εκτέλεσης Software}}{\text{Χρόνος εκτέλεσης Hardware}}$$

Το speed up φαίνεται στην παρακάτω γραφική παράσταση

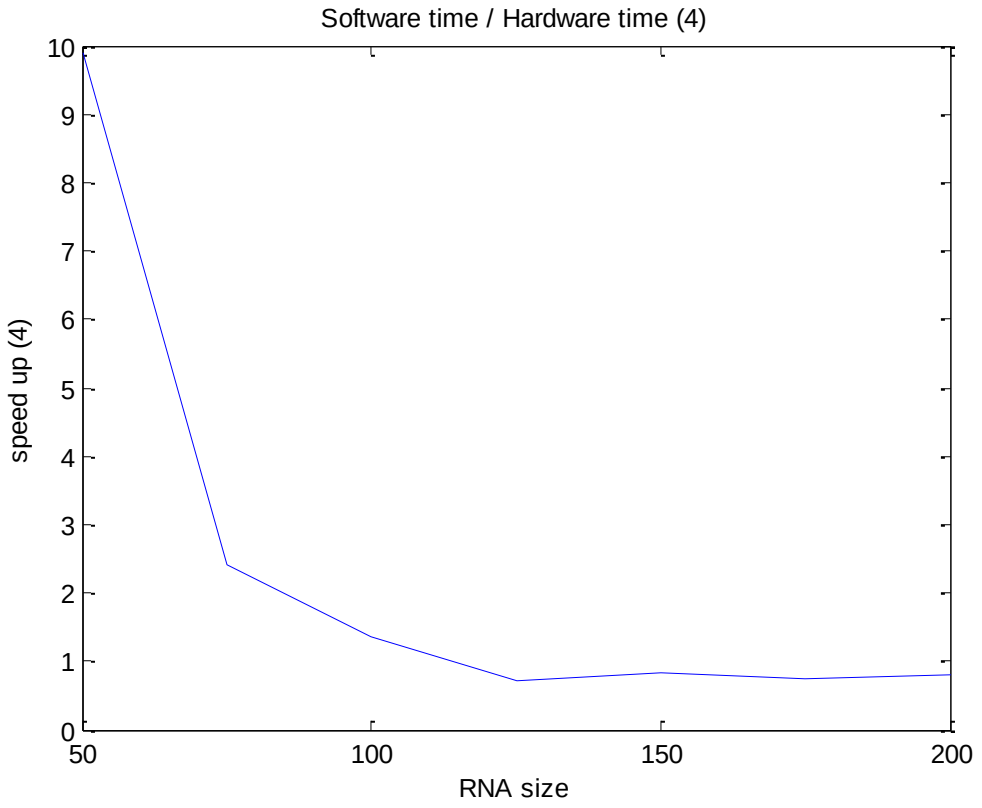


Figure 69 - speed up υλικού (4) σε σχέση με το λογισμικό

Εκτίμηση χρόνου εκτέλεσης υλικού για μεγαλύτερα μεγέθη RNA.

Η εξαγωγή αποτελεσμάτων για μεγαλύτερα μεγέθη RNA δεν είναι δυνατή καθώς η προσομοίωση για μεγάλα χρονικά διαστήματα απαιτεί τεράστια ποσότητα μνήμης, έτσι γίνεται μια εκτίμηση της απόδοσης του αλγορίθμου με βάση τα δείγματα που έχουν ήδη ληφθεί. Η εκτίμηση αυτή θα γίνει με τον υπολογισμό μιας συνάρτησης, με την μέθοδο των ελαχίστων τετραγώνων, η οποία με βάση το μέγεθος του RNA θα παράγει τον χρόνο εκτέλεσης.

Η συνάρτηση αυτή γνωρίζουμε ότι θα είναι πολυωνυμική τετάρτου βαθμού, καθώς ο χρόνος εκτέλεσης έχει γραμμική σχέση με τον αριθμό των προσβάσεων στην μνήμη και ο αριθμός αυτός είναι ίσος με

$$MAC = \frac{L^4 - 10 \cdot L^3 + 59 \cdot L^2 - 170 \cdot L - 24}{24}$$

Με την βοήθεια του εργαλείου Matlab οι συναρτήσεις αυτές υπολογίστηκαν ως

$$Est1(L) = [0{,}0001 \cdot L^3 - 0{,}0012 \cdot L^2 + 0{,}0133 \cdot L - 0{,}4071] \cdot 10^{-4}$$
$$Est2(L) = [0{,}0003 \cdot L^3 - 0{,}0033 \cdot L^2 + 0{,}0064 \cdot L + 0{,}1786] \cdot 10^{-4}$$
$$Est4(L) = [0{,}0005 \cdot L^3 - 0{,}0040 \cdot L^2 - 0{,}0129 \cdot L + 0{,}7571] \cdot 10^{-4}$$

Όπου παρατηρείται η απαλοιφή του όρου L^4 καθώς ο συντελεστής του υπολογίστηκε σε μηδέν. Αυτό συμβαίνει επειδή ο συντελεστής αυτός είναι πολύ μικρός.

Στην συνέχεια υπολογίζουμε το μέσο τετραγωνικό σφάλμα για τις συναρτήσεις αυτές, ώστε να φανεί πόσο καλή εκτίμηση θα έχουμε για τα αποτελέσματα. Το μέσο τετραγωνικό σφάλμα ορίζεται ως το άθροισμα των τετραγώνων των διαφορών των πραγματικών τιμών από τις εκτιμώμενες τιμές.

$$ErrXX = \sum_{L} \big(EstXX(L) - RealXX(L)\big)^2$$

Όπου $RealXX(L)$ τα πειραματικά αποτελέσματα που λάβαμε από την προσομοίωση του αλγορίθμου.

Και λαμβάνουμε τα παρακάτω σφάλματα για κάθε μία από τις τέσσερις περιπτώσεις

$$Err1 = 5{,}0093 \cdot 10^{-12}$$
$$Err2 = 6{,}3698 \cdot 10^{-12}$$
$$Err4 = 2{,}9685 \cdot 10^{-12}$$
$$Err8 = 2{,}7829 \cdot 10^{-12}$$

Το μέσο τετραγωνικό σφάλμα σε όλες τις περιπτώσεις είναι πολύ μικρό, εξασφαλίζοντας ότι η εκτίμηση αυτή θα έχει ρεαλιστικές τιμές.

Στην συνέχεια παρουσιάζονται τα εκτιμώμενα αποτελέσματα σε σχέση με τις πραγματικές τιμές για να επιβεβαιωθεί και γραφικά η ορθότητα των εκτιμήσεων.

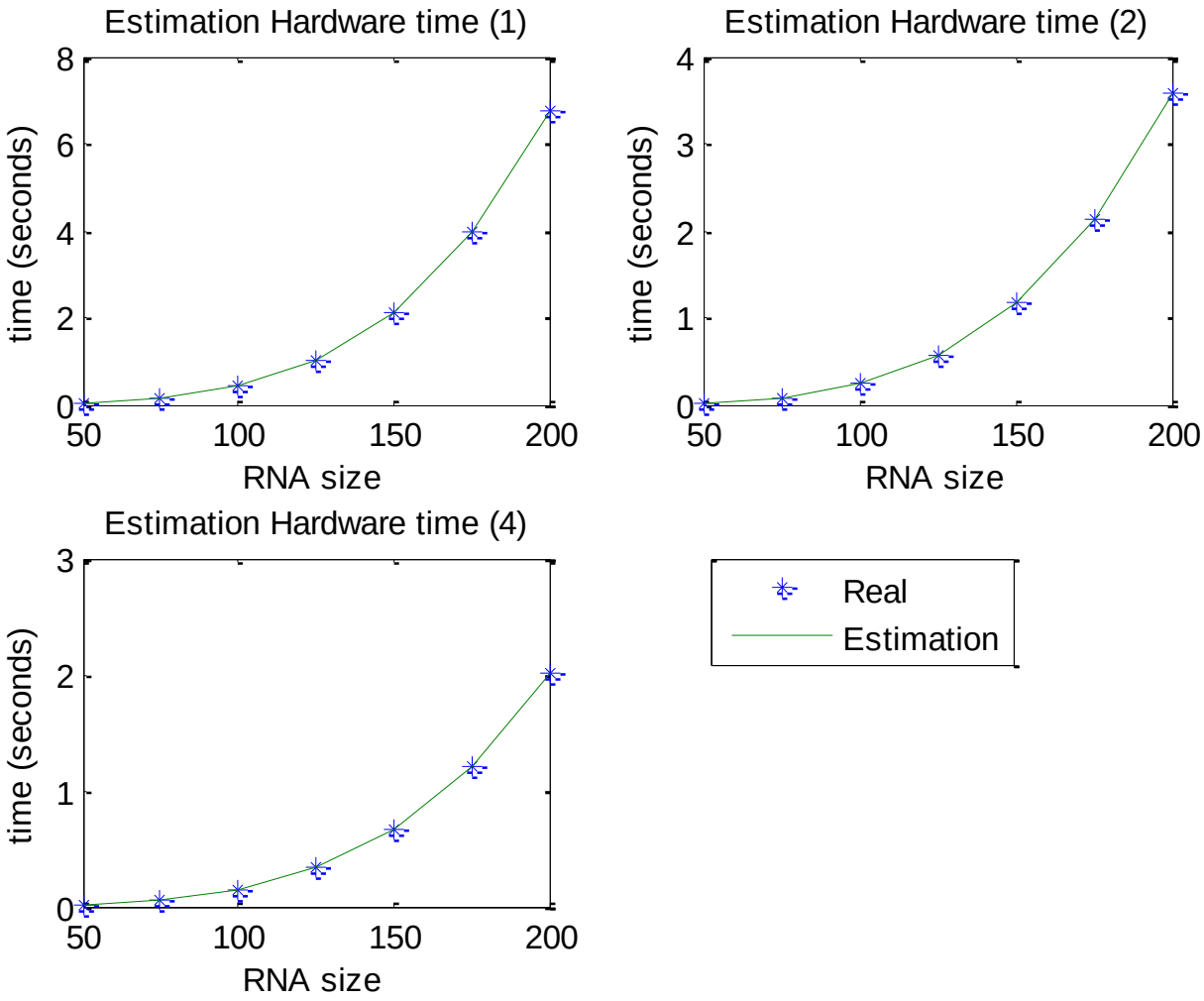


Figure 70 - Σύγκριση εκτιμώμενων τιμών με τις πραγματικές

Παρόλο που η εκτίμηση των τιμών αυτών έγινε με μεγάλη ακρίβεια οι συναρτήσεις αυτές εξακολουθούν να έχουν ένα σφάλμα το οποίο αναμένεται να αυξάνεται όσο οι τιμές του μεγέθους του RNA "απομακρύνονται" από το διάστημα [0,200].

Έτσι για τον περιορισμό του σφάλματος θα πάρουμε εκτιμήσεις τιμών στο διάστημα

$$[100,800]$$

Το οποίο είναι αρκετά μικρό έτσι ώστε να έχουμε ένα αμελητάιο σφάλμα και αρκετά μεγάλο για να λάβουμε κάποια χρήσιμα αποτελέσματα.

Εκτίμηση απόδοσης σε υλικό.

Με βάση τις συναρτήσεις για την εκτίμηση του χρόνου εκτέλεσης παρουσιάζονται οι τιμές για διάφορα μεγέθη RNA στο διάστημα [100,800].

Μέγεθος *RNA* σε βάσεις	**Εκτιμώμενος Χρόνος εκτέλεσης Hardware (1) (ms)**	**Εκτιμώμενος Χρόνος εκτέλεσης Hardware (2) (ms)**	**Εκτιμώμενος Χρόνος εκτέλεσης Hardware (4) (ms)**
100	400	238,4	145,7
200	6.700	3.586,9	2.016,0
300	34.000	17.745,2	9.635,9
400	107.200	55.413,3	29.531,5
500	261.400	134.290,8	70.729,7
600	541.800	277.077,4	144.758,3
700	1.003.200	511.472,7	265.645,8
800	1.710.800	870.176,0	449.922,0

Table 55 - Χρόνοι εκτέλεσης υλικού

Γραφικά οι τιμές αυτές αντιστοιχούν στην παρακάτω γραφική παράσταση

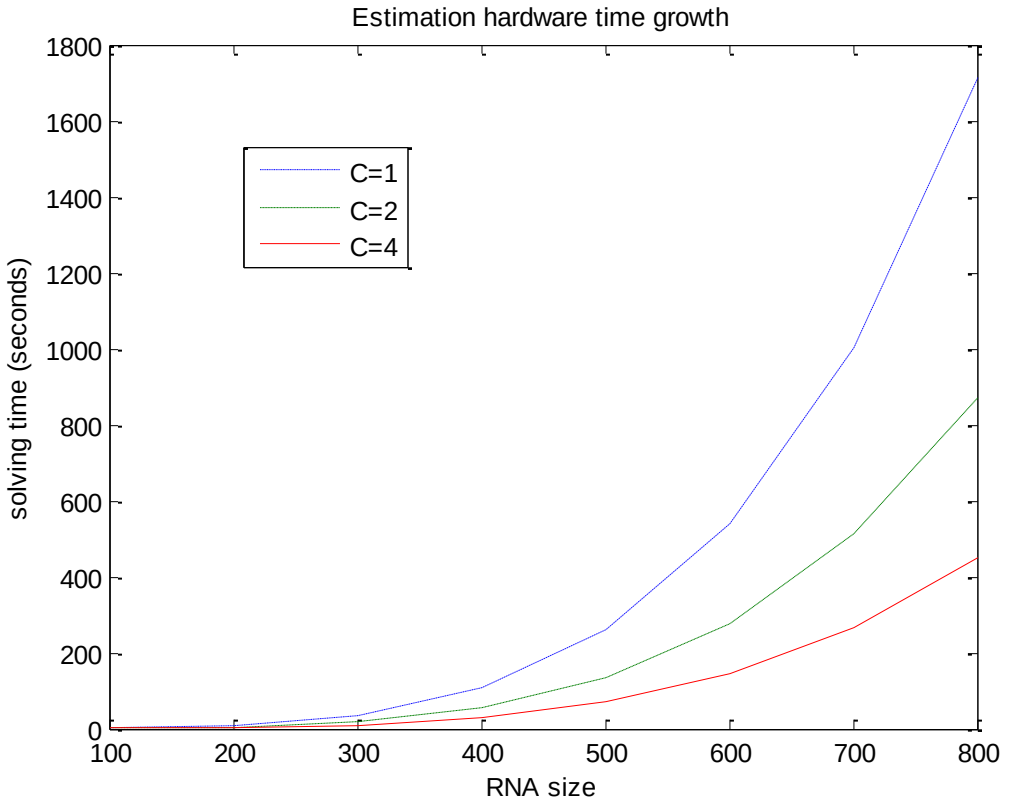


Figure 71 – Εκτιμώμενος ρυθμός αύξησης χρόνου εκτέλεσης (hardware)

Υπολογισμός εκτιμώμενου speed up σε σχέση με την αύξηση των παράλληλων υπολογισμών.

Στην συνέχεια παρουσιάζεται το speed up του υλικού με την αύξηση των παράλληλων υπολογισμών σε σχέση με την περίπτωση χωρίς παράλληλο υπολογισμό για τις εκτιμώμενες τιμές στο διάστημα [100,800]

Μέγεθος *RNA* σε βάσεις	Εκτιμώμενο speed up Hardware (2)	Εκτιμώμενο speed up Hardware (4)
100	1,7780	2,9087
200	1,8760	3,3379
300	1,9141	3,5249
400	1,9342	3,6294
500	1,9467	3,6962
600	1,9552	3,7425
700	1,9614	3,7765
800	1,9661	3,8025

Table 56 - Σύγκριση διαφορετικών επιπέδων παραλληλισμού

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Εκτιμώμενος χρόνος εκτέλεσης Hardware } (1)}{\text{Εκτιμώμενος χρόνος εκτέλεσης Hardware } (C)}$$

Τα speed up φαίνεται στην παρακάτω γραφική παράσταση

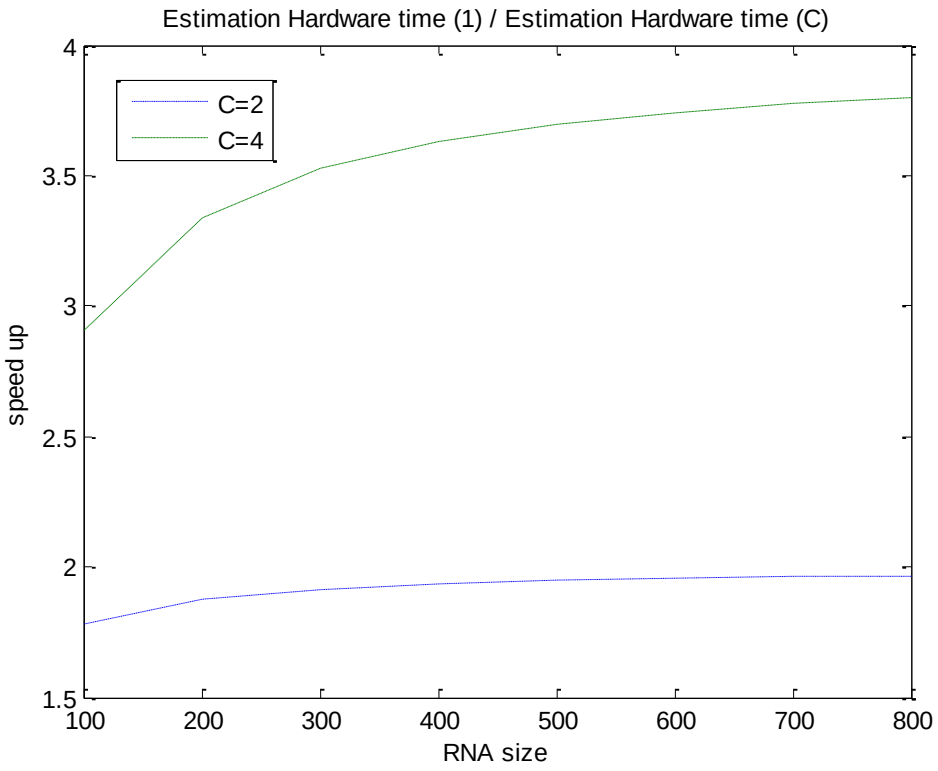


Figure 72 - speed up διαφορετικών επιπέδων παραλληλισμού

Σύγκριση εκτιμώμενων τιμών για την περίπτωση χωρίς παραλληλισμό με software

Μέγεθος *RNA* σε βάσεις	Μέσος χρόνος εκτέλεσης software (UNAFold) (ms)	Εκτιμώμενος Χρόνος εκτέλεσης Hardware (1) (ms)	speedup
100	198,4	400	0,4680
200	1.595,3	6.700	0,2371
300	6.937,4	34.000	0,2042
400	40.111,0	107.200	0,3742
500	89.679,7	261.400	0,3430
600	196.861,0	541.800	0,3634
700	381.368,8	1.003.200	0,3801
800	518.221,8	1.710.800	0,3029

Table 57 - Σύγκριση λογισμικού με εκτίμηση υλικού (1)

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Χρόνος εκτέλεσης Software}}{\text{Εκτιμώμενος χρόνος εκτέλεσης Hardware}}$$

Το speed up φαίνεται στην παρακάτω γραφική παράσταση

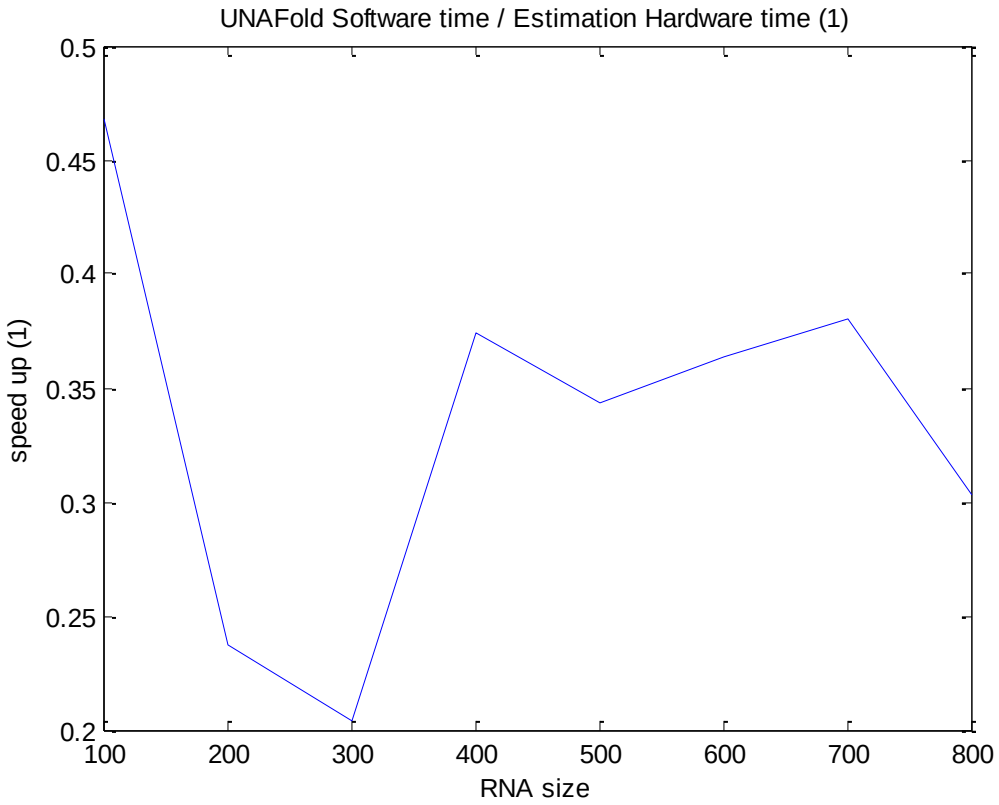


Figure 73 – Εκτιμώμενο speed up υλικού (1) σε σχέση με το λογισμικό

Σύγκριση εκτιμώμενων τιμών για την περίπτωση δύο παράλληλων υπολογισμών με software

Μέγεθος *RNA* σε βάσεις	Μέσος χρόνος εκτέλεσης software (UNAFold) (ms)	Εκτιμώμενος Χρόνος εκτέλεσης Hardware (2) (ms)	speedup
100	198,4	238,4	0,8321
200	1.595,3	3.586,9	0,4448
300	6.937,4	17.745,2	0,3909
400	40.111,0	55.413,3	0,7239
500	89.679,7	134.290,8	0,6678
600	196.861,0	277.077,4	0,7105
700	381.368,8	511.472,7	0,7456
800	518.221,8	870.176,0	0,5955

Table 58 - Σύγκριση λογισμικού με εκτίμηση υλικού (2)

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Χρόνος εκτέλεσης Software}}{\text{Εκτιμώμενος χρόνος εκτέλεσης Hardware}}$$

Το speed up φαίνεται στην παρακάτω γραφική παράσταση

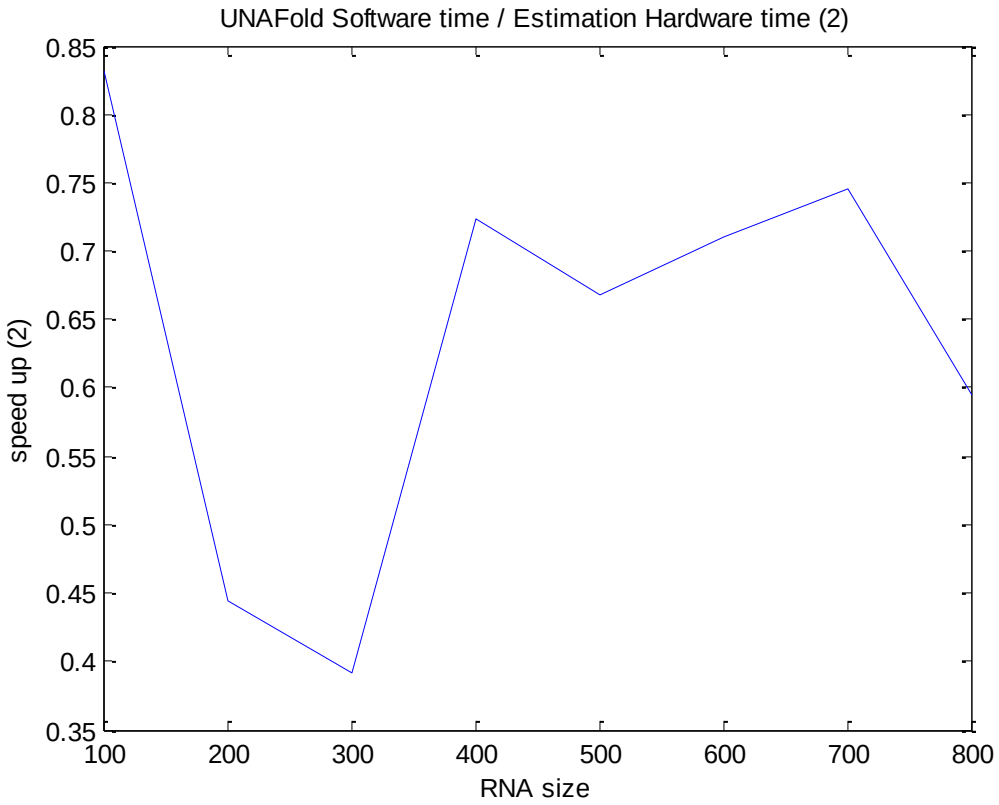


Figure 74 - Εκτιμώμενο speed up υλικού (2) σε σχέση με το λογισμικό

Σύγκριση εκτιμώμενων τιμών για την περίπτωση τεσσάρων παράλληλων υπολογισμών με software

Μέγεθος *RNA* σε βάσεις	Μέσος χρόνος εκτέλεσης software (UNAFold) (ms)	Εκτιμώμενος Χρόνος εκτέλεσης Hardware (4) (ms)	speedup
100	198,4	145,7	1,3613
200	1.595,3	2.016,0	0,7913
300	6.937,4	9.635,9	0,7200
400	40.111,0	29.531,5	1,3582
500	89.679,7	70.729,7	1,2679
600	196.861,0	144.758,3	1,3599
700	381.368,8	265.645,8	1,4356
800	518.221,8	449.922,0	1,1518

Table 59 - Σύγκριση λογισμικού με εκτίμηση υλικού (4)

Όπου για την εξαγωγή του speedup χρησιμοποιήθηκε ο εξής τύπος

$$\frac{\text{Χρόνος εκτέλεσης Software}}{\text{Εκτιμώμενος χρόνος εκτέλεσης Hardware}}$$

Το speed up φαίνεται στην παρακάτω γραφική παράσταση

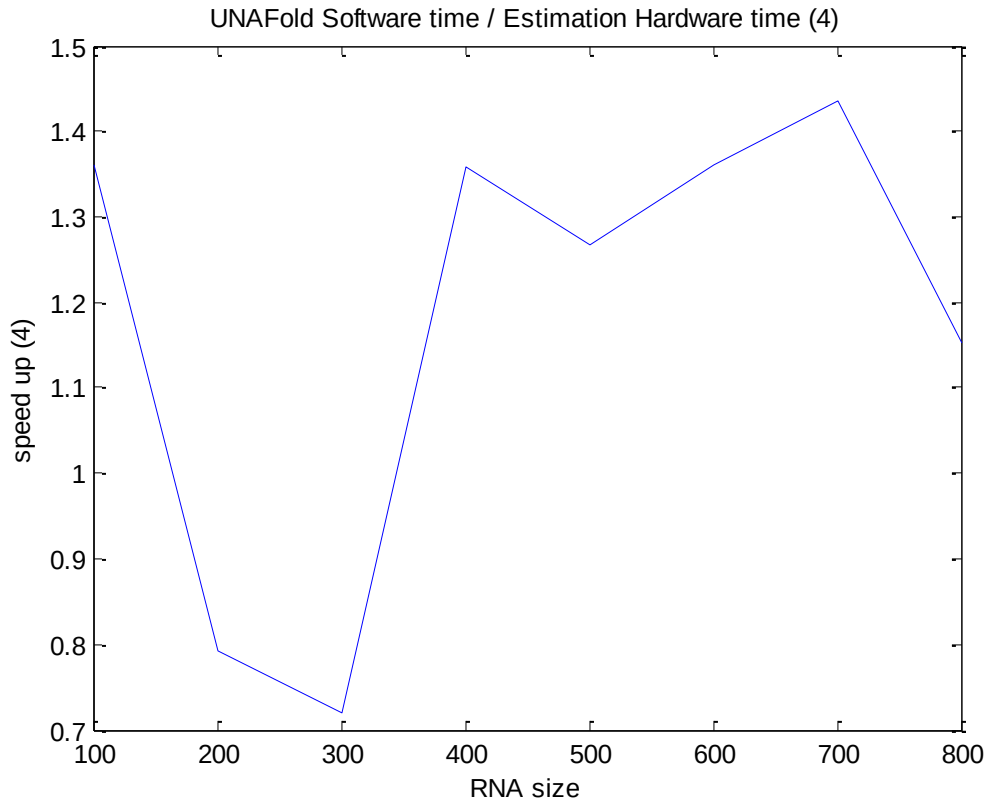


Figure 75 - Εκτιμώμενο speed up υλικού (4) σε σχέση με το λογισμικό

Απόδοση αλγορίθμου για μεγαλύτερα επίπεδα παραλληλισμού.

Παρόλο που στο Board Virtex-2 Pro XC2VP30 δεν στάθηκε δυνατή η εισαγωγή παραπάνω απο τέσσερις πυρήνες υπολογισμού, είναι σημαντική η πρόβλεψη της συμπεριφοράς του αλγορίθμου για μεγαλύτερα επίπεδα παραλληλισμού. Έτσι θα χρησιμοποιηθεί ο αριθμός προσβάσεων στην μνήμη (MAC) που απαιτούνται για την επίλυση του προβλήματος για διάφορα επίπεδα παραλληλισμού, για την εξαγωγή ενός speed up σε σχέση με την περίπτωση χωρις παραλληλισμό.

Αρχικά θα εξακριβωθεί αν ο αριθμός των προσβάσεων στην μνήμη αποτελεί ένα καλό κριτήριο για την εκτίμηση της συμπεριφοράς του αλγορίθμου. Έτσι θα υπολογιστεί θεωριτικά το speed up για την περίπτωση των δύο και τεσσάρων παράλληλων υπολογισμών και στην συνέχεια θα συγκριθεί με το speed up απο τα πειραματικά δεδομένα του πίνακα 45.

Ο αριθμός των προσβάσεων στην μνήμη για την περίπτωση χωρίς παραλληλησμό (MAC1), με δύο παράλληλους υπολογισμούς (MAC2), και τέσσερις παράλληλους υπολογισμούς (MAC4), παρουσιάζονται στον παρακάτω πίνακα.

L	MAC1	MAC2	MAC4
50	214.124	134.726	95.148
75	1.155.874	680.433	443.011
100	3.773.874	2.141.376	1.325.679
125	9.396.249	5.208.958	3.116.198
150	19.741.749	10.769.901	6.285.273
175	36.919.749	19.906.233	11.401.261
200	63.430.249	33.895.301	19.130.179

Table 60 - Αριθμός προσβάσεων μνήμης

Ενώ το speed up που προκύπτει απο τα παραπάνω νούμερα σε αντιδιαστολή με το speed up απο τον πίνακα 45 είναι

L	MAC1/MAC2	speed up Hardware (2)	MAC1/MAC2 / speed up Hardware (2)	MAC1/MAC4	speed up Hardware (4)	MAC1/MAC4 / speed up Hardware (4)
50	1,5893	1,6338	0,9728	2,2504	2,3895	0,9418
75	1,6987	1,7236	0,9856	2,6091	2,6988	0,9668
100	1,7624	1,7779	0,9913	2,8467	2,9086	0,9787
125	1,8039	1,8146	0,9941	3,0153	3,0607	0,9852
150	1,8330	1,8409	0,9957	3,1410	3,1756	0,9891
175	1,8547	1,8606	0,9968	3,2382	3,2656	0,9916
200	1,8714	1,8760	0,9975	3,3157	3,3379	0,9933

Table 61 - Σύγκριση θεωριτικών και πειραματικών δεδομένων για speed up.

Απο τον παραπάνω πίνακα παρατηρήτε μία μικρή απόκλιση μεταξύ των speed up τα οποία έχουν υπολογιστεί θεωριτικά και πειραματικά η οποία μειώνεται καθώς αυξάνεται το μέγεθος του RNA συγκλίνοντας στην ίδια τιμή.

Ο αριθμός των προσβάσεων στην μνήμη μπορεί να χρησιμοποιηθεί για την εκτίμηση με μεγάλη ακρίβεια της απόδοσης του αλγορίθμου για μεγαλύτερα επίπεδα υπολογισμού.

Στην συνέχεια παρουσιάζονται δύο πίνακες με τον αριθμό προσβάσεων στην μνήμη για μεγέθη RNA στο διάστημα [100,1000] και για ένα μεγάλο εύρως παράλληλων υπολογισμών.

L	MAC1	MAC2	MAC4	MAC8	MAC16
100	3.773.874	2.141.376	1.325.679	918.935	717.771
200	63.430.249	33.895.301	19.130.179	11.752.323	8.072.803
300	326.469.124	170.761.726	92.913.429	54.000.085	34.565.017
400	1.040.390.499	538.240.651	287.175.429	161.662.223	98.944.423
500	2.552.694.374	1.311.832.076	691.416.179	381.238.735	226.211.021
600	5.310.880.749	2.717.036.001	1.420.135.679	771.729.623	447.614.803
700	9.862.449.624	5.029.352.426	2.612.833.929	1.404.634.885	800.655.767
800	16.854.900.999	8.574.281.351	4.434.010.929	2.363.954.523	1.329.083.923
900	27.035.734.874	13.727.322.776	7.073.166.679	3.746.188.535	2.082.899.271
1.000	41.252.451.249	20.913.976.701	10.744.801.179	5.660.336.923	3.118.351.803

Table 62 - Αριθμός προσβάσεων μνήμης για 1,2,4,8 και 16 παράλληλους υπολογισμούς

L	MAC32	MAC64	MAC128	MAC256	MAC512
100	621.605	581.707	577.121	577.121	577.121
200	6.251.859	5.379.019	5.012.475	4.964.371	4.964.371
300	24.890.683	20.139.373	17.933.729	17.170.875	17.161.621
400	67.663.095	52.177.567	44.745.075	41.607.275	41.168.871
500	148.819.111	110.366.871	91.628.175	83.233.675	80.986.121
600	285.733.739	205.145.723	165.553.403	147.142.203	140.705.499
700	498.906.979	348.514.121	274.276.051	239.061.753	225.055.899
800	811.963.823	554.033.631	426.328.055	364.994.307	339.036.299
900	1.251.654.255	836.830.979	631.017.805	531.264.607	487.646.699
1.000	1.847.853.259	1.213.591.371	898.434.699	744.804.907	675.887.099

Table 63 - Αριθμός προσβάσεων μνήμης για 32,64,128,256 και 512 παράλληλους υπολογισμούς

Απο τους παραπάνω πίνακες γίνεται φανερός ο περιορισμός στην παραλληλοποίηση του προβλήματος καθώς για μεγάλα επίπεδα παραλληλησμού (256,512) και για μικρά μεγέθη RNA, δεν υπάρχει βελτίωση στον αριθμό των προσβάσεων στην μνήμη ενώ για μεγαλύτερα RNA η βελτίωση δεν είναι ανάλογη με την αύξηση των παράλληλων υπολογισμών.

Ο λόγος για το φαινόμενο αυτό είναι ο περιορισμός στον υπολογισμό στοιχείων πάνω στην ίδια διαγώνιο. 'Ετσι μπορούν να υπολογιστούν το πολύ τόσα στοιχεία όσα και ο αριθμός των στοιχείων στην διαγώνιο, με μέγιστο τα $L-5$ στοιχεία. Επιπλέον καθώς ο αριθμός των στοιχείων στις διαγώνιους μειώνεται πλησιάζοντας την κορυφή του άνω τριγωνικού πίνακα μειώνεται και ο αριθμός των παράλληλων υπολογισμών, φτάνοντας μέχρι και την τιμή 1 για το τελευταίο στοιχείο.

Στην συνέχεια παρουσιάζεται το speed up όπως αυτό προκύπτει απο την μείωση των προσβάσεων στην μνήμη, σε σχέση με την περίπτωση χωρίς παραλληλισμό.

L	MAC1/MAC2	MAC1/MAC4	MAC1/MAC8	MAC1/MAC16	MAC1/MAC32
100	1,7624	2,8467	4,1068	5,2578	6,0711
200	1,8714	3,3157	5,3973	7,8573	10,1458
300	1,9118	3,5137	6,0457	9,4451	13,1161
400	1,9329	3,6228	6,4356	10,5149	15,3760
500	1,9459	3,6920	6,6958	11,2846	17,1530
600	1,9547	3,7397	6,8818	11,8648	18,5868
700	1,9610	3,7746	7,0214	12,3180	19,7681
800	1,9658	3,8013	7,1300	12,6816	20,7582
900	1,9695	3,8223	7,2169	12,9799	21,6000
1.000	1,9725	3,8393	7,2880	13,2289	22,3245

Table 64 – Speed up για 2,4,8 και 16 παράλληλους υπολογισμούς

L	MAC1/MAC64	MAC1/MAC128	MAC1/MAC256	MAC1/MAC512
100	6,4876	6,5391	6,5391	6,5391
200	11,7922	12,6545	12,7771	12,7771
300	16,2105	18,2042	19,0130	19,0232
400	19,9394	23,2515	25,0050	25,2713
500	23,1292	27,8593	30,6690	31,5201
600	25,8883	32,0796	36,0935	37,7447
700	28,2986	35,9581	41,2548	43,8222
800	30,4222	39,5351	46,1785	49,7141
900	32,3073	42,8446	50,8894	55,4412
1.000	33,9920	45,9159	55,3869	61,0345

Table 65 – Speed up για 32,64,128,256 και 512 παράλληλους υπολογισμούς

Απο την μειωμένη βελτίωση του αλγορίθμου, για μεγάλη αύξηση των παράλληλων υπολογισμών, προκύπτει το ερώτημα για το μέγιστο speed up που μπορεί να πετύχει η αρχιτεκτονική καθώς και ο βέλτιστος αριθμός παράλληλων υπολογισμών για κάθε περίπτωση.

Έτσι θα υπολογιστεί για διάφορα μεγέθη του RNA το speed up που θα έχει χρησιμοποιώντας τον μέγιστο δυνατό αριθμό παράλληλων υπολογισμών, $L-5$ όπου L το μέγεθος του RNA.

Απόδοση αλγορίθμου για το μέγιστο επίπεδο παραλληλισμού.

Στον παρακάτω πίνακα παρουσιάζεται το μέγιστο speed up του αλγορίθμου απο την μέγιστη παραλληλοποίηση του αλγορίθμου. Ο αριθμός των παράλληλων υπολογισμών που μπορούν να εκτελεστούν σε κάθε περίπτωση είναι ίση με το μέγεθος του RNA μείων πέντε, καθώς αυτός είναι ο αριθμός των στοιχείων της μεγαλύτερης διαγωνίου που θα υπολογιστεί.

Μέγεθος *RNA* σε βάσεις	**Μέγιστο speed up**	**Μέγεθος *RNA* σε βάσεις**	**Μέγιστο speed up**
100	6,5391	1.000	62,7679
200	12,7771	2.000	125,2667
300	19,0232	3.000	187,7664
400	25,2713	4.000	250,2662
500	31,5201	5.000	312,7661
600	37,7694	6.000	375,2660
700	44,0188	7.000	437,7659
800	50,2684	8.000	500,2659
900	56,5181	9.000	562,7659

Table 66 – Μέγιστο speed up για διάφορα μεγέθη RNA

Απο τον παραπάνω πίνακα γίνεται φανερό ότι δεν είναι αποδοτική η χρήση τόσο μεγάλου επιπέδου υπολογισμού καθώς η μεγάλη πληοψηφία των calculators μένουν ανεκμετάλευτοι.

Στον παρακάτω πίνακα παρουσιάζεται το μέσο ποσοστό χρήσης των πυρήνων υπολογισμού. Ο υπολογισμός του γίνεται με την διαίρεση του speed up, το οποιο εκφράζει τον μέσο αριθμό υπολογισμό ανα κύκλο ρολογιού, με τον αριθμό των πυρήνων.

Μέγεθος *RNA* σε βάσεις	**Μέση ποσοστιαία χρήση πυρήνων (%)**	**Μέγεθος *RNA* σε βάσεις**	**Μέση ποσοστιαία χρήση πυρήνων (%)**
100	6,8833	1000	6,3083
200	6,5524	2000	6,2790
300	6,4485	3000	6,2693
400	6,3978	4000	6,2645
500	6,3677	5000	6,2616
600	6,3478	6000	6,2596
700	6,3336	7000	6,2583
800	6,3231	8000	6,2572
900	6,3149	9000	6,2564

Table 67 – Μέση ποσοστιαία χρήση πυρήνων για μέγιστο speed up

Έτσι προκύπτει ότι κατα μέσο όρο μόνο το έξι με εφτά τις εκατό των συνολικών διαθέσιμων πυρήνων υπολογισμού χρησιμοποιούνται για κάποιον υπολογισμό, το οποίο δεν αποτελεί καθόλου καλή αξοιοποίηση του χώρου που καταλαμβάνουν οι πυρήνες αυτοί.

Βέλτιστο επίπεδο παραλληλισμού για διάφορα μεγέθη RNA

Στην συνέχεια έγινε μια προσπάθεια έτσι ώστε να διατηρηθεί ένα καλό speed up με έναν αριθμό πυρήνων υπολογισμού οι οποίοι θα αξιοποιούνται αποδοτικά. Η πρώτη προσπάθεια ήταν να χρησιμοποιηθούν τόσοι πυρήνες υπολογισμού όσο και το speed up που λαμβάνουμε απο την περίπτωση του μέγιστου υπολογισμού.

Μέγεθος *RNA* σε βάσεις	Αριθμός πυρήνων υπολογισμού	Speed up	Μέση ποσοστιαία χρήση πυρήνων (%)	Μέγεθος *RNA* σε βάσεις	Αριθμός πυρήνων υπολογισμού	Speed up	Μέση ποσοστιαία χρήση πυρήνων (%)
100	6	3,5796	59,6601	1000	62	33,4295	53,9186
200	12	6,8225	56,8541	2000	125	66,8347	53,4678
300	19	10,3633	54,5434	3000	187	99,9569	53,4529
400	25	13,6167	54,4667	4000	250	133,3622	53,3449
500	31	16,8705	54,4211	5000	312	166,4848	53,3605
600	37	20,1246	54,3909	6000	375	199,8901	53,3040
700	44	23,6643	53,7825	7000	437	233,0128	53,3210
800	50	26,9195	53,8390	8000	500	266,4181	53,2836
900	56	30,1746	53,8832	9000	562	299,5409	53,2991

Table 68 – Speed up και αξιοποίηση πυρήνων για παραλληλισμό ίσο με το βέλτιστο speed up

Η μείωση των πυρήνων συντέλεσε σε μια μείωση του speed up περίπου στο μισό ενώ ταυτόχρονα αυξήθηκε η αξιοποίηση των πυρήνων περίπου εννέα φορές.

Η μέση ποσοστιαία χρήση των πυρήνων αν και σημαντικά βελτιωμένη δεν είναι αρκετή και έτσι ο αριθμός των πυρήνων υπολογισμού μειώθει επιπλέον στο μισό.

Μέγεθος *RNA* σε βάσεις	Αριθμός πυρήνων υπολογισμού	Speed up	Μέση ποσοστιαία χρήση πυρήνων (%)	Μέγεθος *RNA* σε βάσεις	Αριθμός πυρήνων υπολογισμού	Speed up	Μέση ποσοστιαία χρήση πυρήνων (%)
100	3	2,3624	78,7476	1000	31	21,8404	70,4528
200	6	4,4636	74,3933	2000	62	43,4428	70,0690
300	9	6,5717	73,0184	3000	93	65,0458	69,9418
400	12	8,6814	72,3448	4000	125	87,1340	69,7072
500	15	10,7917	71,9449	5000	156	108,7373	69,7034
600	18	12,9024	71,6802	6000	187	130,3406	69,7009
700	22	15,5067	70,4851	7000	218	151,9440	69,6991
800	25	17,6179	70,4716	8000	250	174,0312	69,6125
900	28	19,7291	70,4612	9000	281	195,6347	69,6209

Table 69 – Speed up και αξιοποίηση πυρήνων για παραλληλισμό ίσο με το μισό του βέλτιστο speed up

Στην περίπτωση αυτή η αξιοποίηση των πυρήνων υπολογισμού έχει αυξηθεί ακόμα περισσότερο και κυμαίνεται περίπου στο 70%. Αποτελεί ένα καλό επίπεδο χρησιμοποίησης πόρων και παρέχει ένα αρκετά υψηλό speed up.

Απαιτήσεις βέλτιστου επίπεδου παραλληλισμού

Όπως είναι λογικό η αύξηση απο τέσσερις σε μερικές εκατοντάδες, ακόμα και δεκάδες πυρήνες υπολογισμού επιφέρει δραματική επίδραση στην ανάγκη της σχεδίασης σε πόρους. Έτσι αναμένεται κάποια επίδραση στο ρολόι του συστήματος, στον συνολικό αριθμό slices και φυσικά στα BRams.

Επίδραση στο ρολόι

Καθώς οι πυρήνες αυτοί λειτουργούν ανεξάρτητα, η επίδραση στο ρολόι θα οφείλεται στην αύξηση του fanout για τα δεδομένα εισόδου, καθώς τα στοιχεία τα οποία θα διαβάζονται απο την μνήμη θα πρέπει να τροφοδοτιθούν σε όλους τους πυρήνες ταυτόχρονα. Η αρνητική αυτή επίδραση, μπορεί ώς ένα σημείο να αποφευχθεί, με την εισαγωγή μιας δεντροηδούς δομής απο καταχωριτές οι οποίοι θα μεταφέρουν τα δεδομένα στους πυρήνες περιορίζοντας το fanout σε μικρά επίπεδα. Η επίδραση των καταχωρητών στον συνολικό χρόνο εκτέλεσης θα έχουν αμελητέα επίδραση, καθώς θα επεκτείνουν το μήκος του pipeline κρύβοντας έτσι τον χρόνο μεταφοράς των δεδομένων.

Επίδραση στα BRams

Για την υλοποίηση ενός ζεύγους απο πυρήνες απαιτούνται 35 Brams, έτσι προστέτοντας έναν ακόμα πυρήνα για τις περιπτώσεις με περιττό αριθμό πυρήνων έχουμε τις παρακάτω απαιτήσεις σε BRams.

Μέγεθος *RNA* σε βάσεις	**Αριθμός πυρήνων υπολογισμού**	**Απαιτούμενος αριθμός BRams**	**Μέγεθος *RNA* σε βάσεις**	**Αριθμός πυρήνων υπολογισμού**	**Απαιτούμενος αριθμός BRams**
100	4	70	1000	32	560
200	6	105	2000	62	1.085
300	10	175	3000	92	1.610
400	12	210	4000	126	2.205
500	16	280	5000	158	2.765
600	18	315	6000	188	3.290
700	22	385	7000	218	3.815
800	26	455	8000	250	4.375
900	28	490	9000	282	4.935

Table 70 - Απαιτούμενος αριθός Block RAMs για την υλοποίηση των πυρήνων

Ο αριθμός των απαιτούμενων BRams για μεγάλα μεγέθη RNA αυξάνεται πέρα απο τα επιτρεπτά όρια, ακόμα και για τα μεγαλύτερα Boards. Έτσι θα πρέπει με βάση τον μέγιστο επιτρεπτό αριθμό πυρήνων να υπολογιστεί ένα speed up για διάφορα Board.

Βέλτιστα επίπεδα παραλληλισμού για διάφορα boards

Στο σημείο αυτό ο κρίσιμος πόρος είναι ο αριθμός των Block Rams. Στον ακόλουθο πίνακα παρουσιάζεται ο μέγιστος αριθμός Block Rams για διάφορα boards τα οποία είναι διαθέσιμα στην αγορά καθώς και ο μέγιστος αριθμός πυρήνων υπολογισμού ο οποίος μπορεί να υλοποιηθεί.

Board	BRams	Πυρήνες
XC5VLX110T	296	16
XC5VLX330T	648	36
XC5VSX240T	1,032	58

Table 71 – Διαθέσιμα Block Rams και μέγιστος αριθμός πυρήνων για διάφορα Boards

Στην συνέχεια και με βάση τον αριθμό τον μέγιστο αριθμό πυρήνων για κάθε Board υπολογίζεται το speed up σε σχέση με την περίπτωση χωρίς παραλληλισμό.

Μέγεθος *RNA* σε βάσεις	XC5VLX110T		XC5VLX330T		XC5VSX240T	
	Speedup	Μέση ποσοστιαία χρήση πυρήνων (%)	Speedup	Μέση ποσοστιαία χρήση πυρήνων (%)	Speedup	Μέση ποσοστιαία χρήση πυρήνων (%)
100	5,2578	*32,8611*	6,1701	*17,1391*	6,4511	*11,1225*
200	7,8573	*49,1080*	10,4793	*29,1091*	11,6073	*20,0125*
300	9,4451	*59,0317*	13,7035	*38,0653*	15,8313	*27,2954*
400	10,5149	*65,7181*	16,2050	*45,0139*	19,3518	*33,3651*
500	11,2846	*70,5286*	18,2020	*50,5611*	22,3299	*38,4999*
600	11,8648	*74,1553*	19,8330	*55,0918*	24,8820	*42,8999*
700	12,3180	*76,9873*	21,1903	*58,8618*	27,0933	*46,7126*
800	12,6816	*79,2600*	22,3373	*62,0479*	29,0279	*50,0480*
900	12,9799	*81,1241*	23,3194	*64,7761*	30,7345	*52,9906*
1000	13,2289	*82,6808*	24,1698	*67,1383*	32,2513	*55,6057*
2000	14,4814	*90,5085*	28,9187	*80,3298*	41,4553	*71,4747*
3000	14,9541	*93,4631*	30,9471	*85,9642*	45,8120	*78,9862*
4000	15,2024	*95,0148*	32,0722	*89,0895*	48,3525	*83,3664*
5000	15,3554	*95,9712*	32,7875	*91,0764*	50,0166	*86,2356*
6000	15,4592	*96,6197*	33,2824	*92,4512*	51,1910	*88,2604*
7000	15,5341	*97,0884*	33,6452	*93,4589*	52,0643	*89,7660*
9000	15,6358	*97,7205*	34,1414	*94,8373*	53,2759	*91,8551*

Table 72 – Speed up και μέση ποσοστιαία χρήση πυρήνων για διάφορα Boards

Όπως φαίνεται και απο τον παραπάνω πίνακα, η μέση ποσοστιαία χρήση των πυρήνων αυξάνεται πάνω απο το 90%.

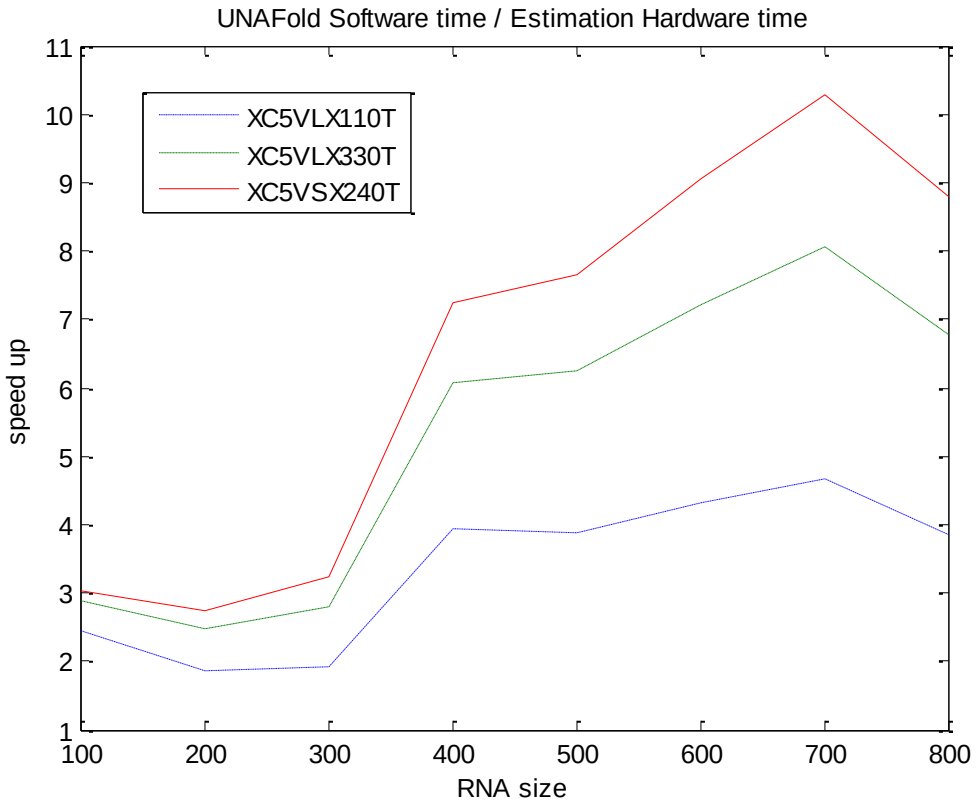
Εκτίμηση speed up σε σχέση με το λογισμικό

Υπο την υπόθεση ότι η εφαρμογή της σχεδίασης στα τρία boards που αναλήθηκαν παραπάνω θα εξακολουθήσει να έχει ρολόι 100Mhz και χρησιμοποιόντας τα δεδομένα απο τον πίνακα 57, για την σύγκριση της περίπτωσης της σχεδίασης χωρίς παραλληλισμό το το λογισμικό απο το πακέτο UNAFold μπορούμε να έχουμε μια εκτίμηση για την απόδοση του αλγοριθμου σε μεγαλύτερα board σε σχέση με το λογισμικό.

Μέγεθος *RNA* σε βάσεις	**XC5VLX110T Speedup**	**XC5VLX330T Speedup**	**XC5VSX240T Speedup**
100	2,4606	2,8876	3,0191
200	1,8630	2,4846	2,7521
300	1,9287	2,7983	3,2328
400	3,9347	6,0639	7,2414
500	3,8706	6,2433	7,6592
600	4,3117	7,2073	9,0421
700	4,6821	8,0544	10,2982
800	3,8413	6,7660	8,7925

Table 73 – Speed up σε σχέση με το λογισμικό για διάφορα Boards

Το speed up αναμένεται να αυξηθεί για μεγαλύτερα μεγέθη RNA καθότι αυξάνεται η μέση ποσοστιαία χρήση των πυρήνων.



ΣΥΜΠΕΡΑΣΜΑΤΑ - ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Από την μελέτη των αποτελεσμάτων μπορούμε να εξάγουμε μερικά χρήσιμα συμπεράσματα για την αρχιτεκτονική του αλγορίθμου του Zucker σε υλικό.

Καταρχήν γίνεται εμφανές ότι η αύξηση των παράλληλων υπολογισμών επιφέρει μεγάλη βελτίωση στον χρόνο εκτέλεσης του αλγορίθμου. Η βελτίωση αυτή (speedup) είναι αύξουσα σε σχέση με το μέγεθος του RNA συγκλίνοντας, για RNA μεγάλου μεγέθους στον αριθμό των παράλληλων υπολογισμών. Η αύξηση αυτή ήταν αναμενόμενη καθώς υπάρχει αντίστοιχη μείωση των προσβάσεων στην μνήμη (MAC).

Η σύγκριση των αποτελεσμάτων με τους χρόνους εκτέλεσης του λογισμικού αναφοράς φανερώνουν ότι για μικρά μεγέθη RNA το λογισμικό υπερτερεί σε σχέση με το υλικό ενώ στην συνέχεια ακολουθεί μια ταχύτατη βελτίωση του υλικού. Κύριος παράγοντας για το φαινόμενο αυτό είναι η κρυφή μνήμη του επεξεργαστή στον οποίο εκτελείται το λογισμικό. Παρόλο που απορρίφθηκε από την αρχιτεκτονική του συστήματος ως μη αποτελεσματικός τρόπος βελτίωσης του αλγορίθμου, για μικρά RNA η κρυφή μνήμη μπορεί να επιφέρει σημαντική βελτίωση, όσο όμως αυξάνεται το μέγεθος του RNA τόσο μειώνεται η επίδρασή της. Η βελτίωση (speedup) του υλικού σε σχέση με το λογισμικό αναμένεται να συγκλίνει σε κάποια σταθερή τιμή για μεγαλύτερα μεγέθη RNA όπου επίδραση της κρυφής μνήμης θα είναι αμελητέα.

Η σύγκριση των αποτελεσμάτων με τους χρόνους εκτέλεσης με το λογισμικό από το πακέτο UNAFold παρουσιάζει ακριβώς τα αντίθετα αποτελέσματα, με το λογισμικό να είναι πολύ ποιο αργό από το υλικό για μικρά RNA και στην συνέχεια να βελτιώνεται όσο μεγαλώνει το μέγεθος του RNA. Τα αποτελέσματα μπορεί να είναι παραπλανητικά καθώς όπως φαίνεται και από τις εκτιμώμενες τιμές η βελτίωση αυτή διαρκεί ένα μικρό διάστημα τιμών και για μεγαλύτερα RNA η βελτίωση του λογισμικού μειώνεται και σταθεροποιείται σε κάποια σταθερό εύρος τιμών. Η περίεργη αυτή αυξομείωση της βελτίωσης οφείλεται σε ένα μεγάλο σταθερό κόστος που εκτελεί ο αλγόριθμος για τον υπολογισμό των πειραματικών δεδομένων στο περιβάλλον (θερμοκρασία, περιεκτικότητα διαλύματος σε κατιόντα νατρίου και γενικά παράγοντες που επηρεάζουν τις ενέργειες αυτές) το οποίο ορίζεται κατά την εκτέλεση του. Η υλοποίηση στο λογισμικό δεν περιέχει αυτό το κόστος καθώς οι ενέργειες είναι υπολογισμένες και αποθηκευμένες σε κατάλληλες μνήμες στο σύστημα. Το κόστος αυτό επιδρά στον υπολογισμό μικρών μεγεθών RNA ενώ σε μεγαλύτερα η αύξηση στον χρόνο εκτέλεσης είναι μηδαμινή σε σχέση με τον χρόνο εκτέλεσης του αλγορίθμου. Στην συνέχεια η σταδιακή βελτίωση του υλικού οφείλεται και πάλι στην επίδραση της κρυφής μνήμης.

Η απαλοιφή των υπολογισμών των ενεργειών των πειραματικών δεδομένων εισάγει μια μείωση του χρόνου εκτέλεσης, κυρίως για μικρά RNA αλλά ταυτόχρονα αποτελεί ένα σημαντικό μειονέκτημα του αλγορίθμου καθώς για τον υπολογισμό ενεργειών σε άλλες συνθήκες περιβάλλοντος απαιτείται επαναπρογραμματισμός του board. Μία πιθανή λύση θα ήταν ο προγραμματισμός των ενεργειών με την σύνδεση του Board με κάποιο Interface το οποίο θα παρέχει τις ενέργειες αυτές. Επιπλέον ο αλγόριθμος του Zucker έχει ανανεωθεί παρέχοντας διαφορετικό τρόπο υπολογισμού των K-loops και συγκεκριμένα των Multi-loops οι οποίοι δεν έχουν ενσωματωθεί στην αρχιτεκτονική.

Εν κατακλείδι στα πλαίσια της εργασίας αυτής μελετήθηκε κυρίως ο τρόπος διαχείρισης των στοιχείων για των υπολογισμό πινάκων δυναμικού προγραμματισμού και ο παραλληλισμός των αναγνώσεων από τους πίνακες για την μείωση του συνολικού χρόνου εκτέλεσης. Οι πίνακες αναδιατάχθηκαν με τέτοιο τρόπο έτσι ώστε να απλοποιηθεί η διαδικασία ανάκτησης δεδομένων και να επιτραπεί η αποδοτική ανάγνωση των στοιχείων σε ριπές, ενώ τα ίδια τα στοιχεία θα χρησιμοποιηθούν για πολλούς παράλληλους υπολογισμούς μειώνοντας τον συνολικό αριθμό προσβάσεων την μνήμη. Η εργασία αυτή μπορεί να αποτελέσει την βάση για την μοντελοποίηση παρόμοιων προβλημάτων δυναμικού προγραμματισμού.

REFERENCES

[1] Crick, F.H.C. (1958): *On Protein Synthesis.* Symp. Soc. Exp. Biol. XII, 139-163.

[2] Crick, F. (1970): *Central Dogma of Molecular Biology.* Nature 227, 561-563. PMID 4913914

[3] http://en.wikipedia.org/wiki/RNA

[4] Shantanu Sharma,1 Feng Ding,1 and Nikolay V. Dokholyan *iFoldRNA: Three-dimensional RNA Structure Prediction and Folding*, Bioinformatics **24** (2008), pp. 1951–1952.

[5] Boris Fürtig, Christian Richter, Jens Wöhnert, Harald Schwalbe, *"NMR-spectroscopy of RNA"*, Chembiochem, 4, 936-962.

[6] Zuker M, Stiegler P. (1981) *Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids* Res. Jan 10;9(1):133-48

[7] http://en.wikipedia.org/wiki/Secondary_structure

[8] http://en.wikipedia.org/wiki/Gibbs_free_energy

[9] R. Nussinov, G. Piecznik, J. R. Grigg and D. J. Kleitman, (1978) *Algorithms for loop matchings*, SIAM Journal on Applied Mathematics 35, 68-82.

[10] R. B. Lyngsø, M. Zuker, and C. N. S. Pedersen. (1999) *Internal loops in RNA secondary structure prediction.* In Proceedings of the 3rd Annual International Conference on Computational Molecular Biology (RECOMB).

[11] R.B. Lyngsø, M. Zuker, and C.N.S. Pedersen. (1999) *An Improved Algorithm for RNA Secondary Structure Prediction.* Tech-report BRICS RS-99-15.

[12] Mathews DH, Sabina J, Zuker M, Turner DH (1999) *Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure*. J Mol Biol. 288(5):911-40.

[13] N. R. Markham and M. Zuker. *Unafold: Software for nucleic acid folding and hybridization.* In Jonathan M. Keith, editor, Bioinformatics, Volume II. Structure, Functions and Applications, number 453 in Methods in Molecular Biology, chapter 1, pages 3–31. Humana Press, Totowa, New Jersey, 2008.