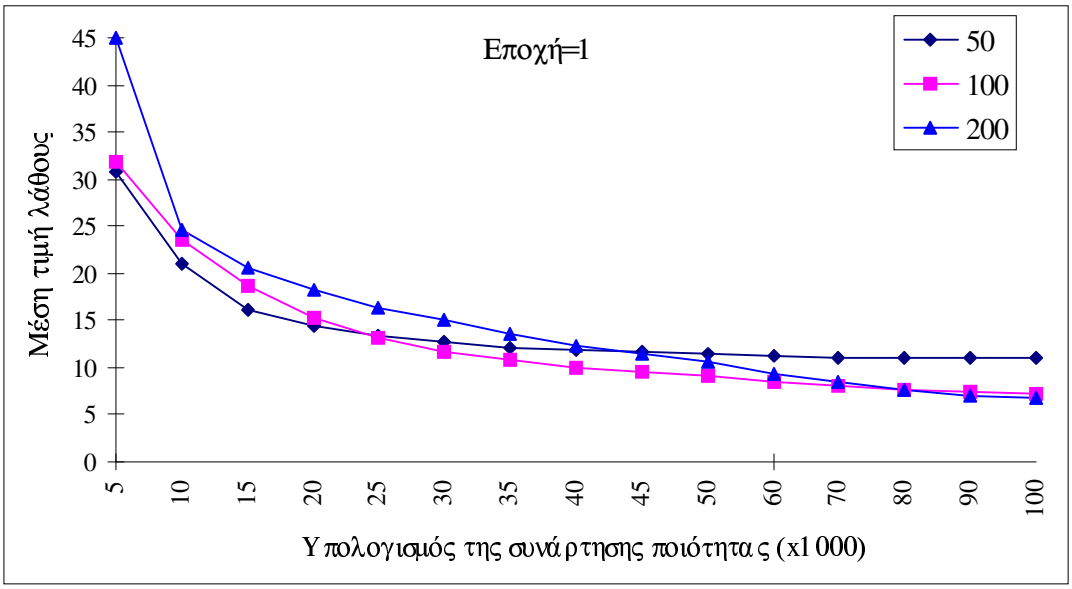
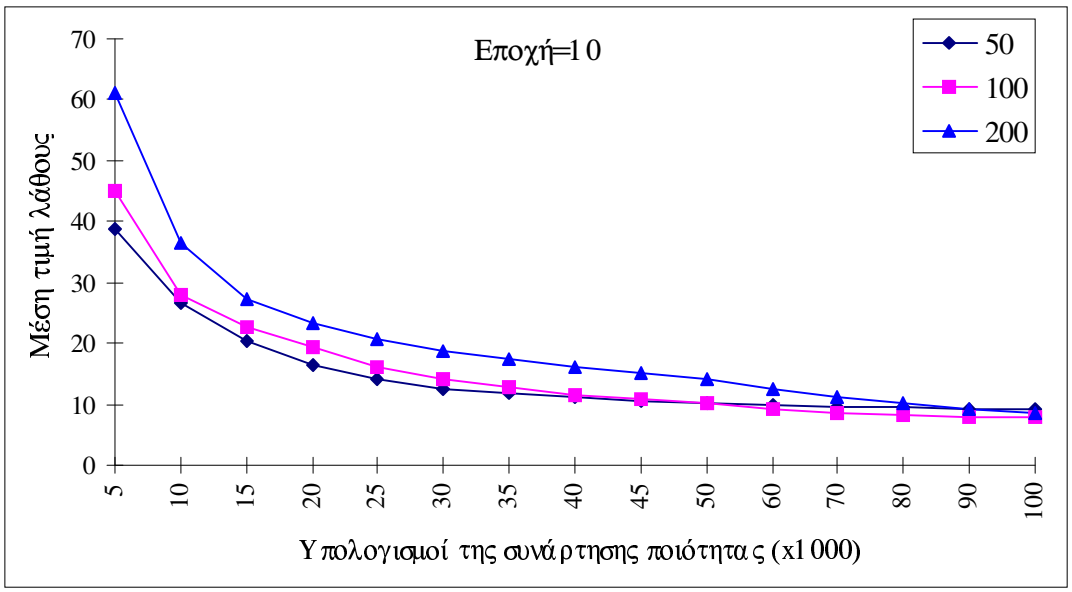
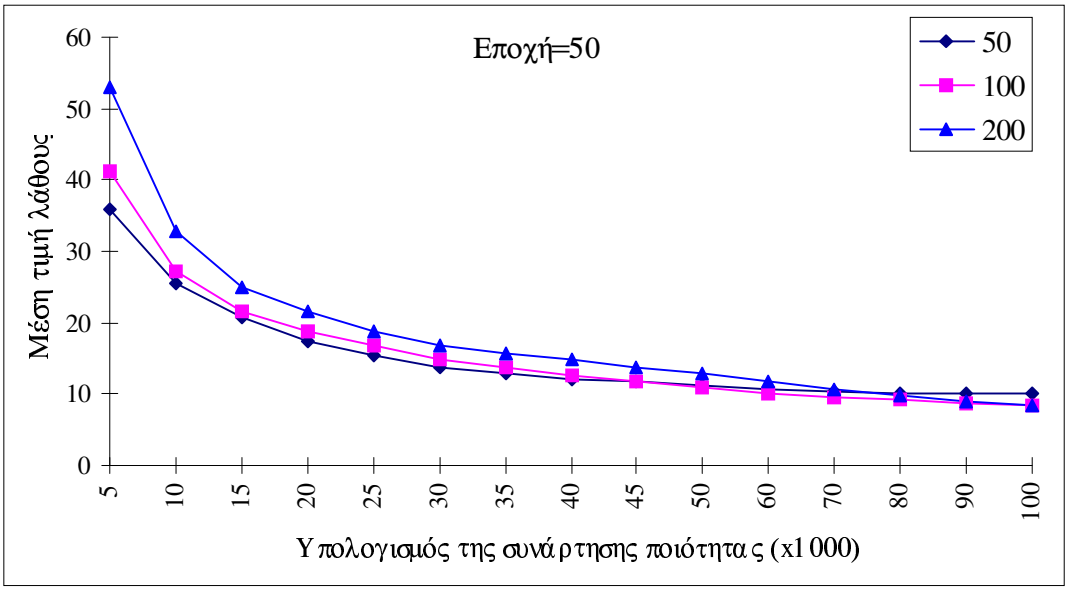
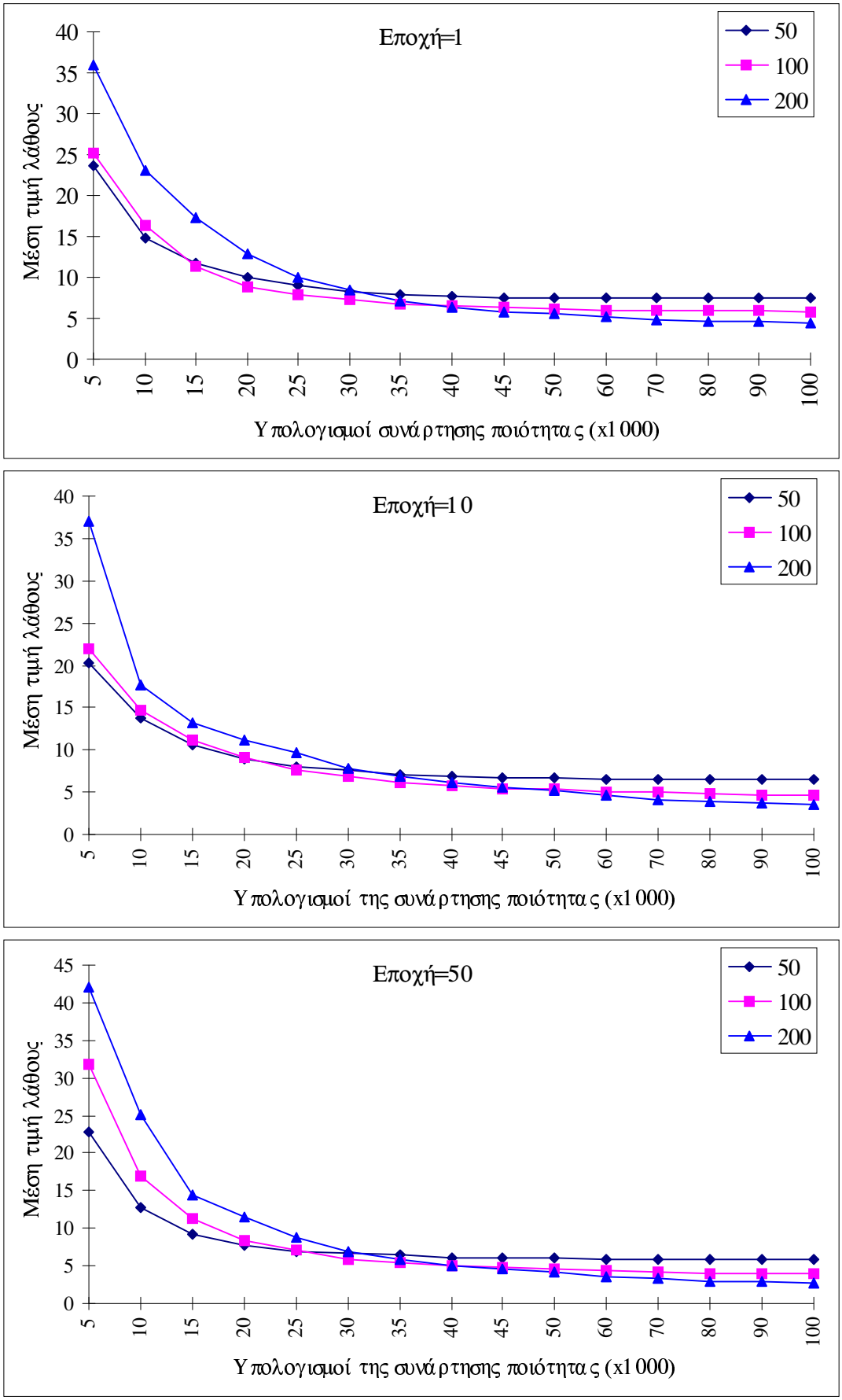
6.2.3. Πρόβλημα του Περιοδεύοντος Πωλητή 67
7. Συμπεράσματα 72
Βιβλιογραφία 76
Παράρτημα Α: Απόδοση Όρων 86
Παράρτημα Β: Πρόβλημα Περιοδεύοντος Πωλητή 88
Παράρτημα Γ: Περιγραφή του Transputer IMS T800 90
Παράρτημα Δ: Περιγραφή της κάρτας Fast9 93
Παράρτημα Ε: 98
Παράρτημα ΣΤ 103
Α. Περιεχόμενα του αρχείου διαμόρφωσης της τοπολογίας σύνδεσης των transputers 103
Β. Αρχεία διαμόρφωσης των παράλληλων υλοποιήσεων. 104
Παράρτημα Ζ 112
Α. Περιγραφή των αρχείων 112
Β. Εκτέλεση των προσομοιώσεων 115







Σχήμα 6.13. Βελτίωση του μέσου λάθους για το μοντέλο νησίδων (Συνάρτηση Rastrigin)



Σχήμα 6.14. Βελτίωση του μέσου λάθους για τους ΣΠΔΕΣ (Συνάρτηση Rastrigin)

6.2.3. Πρόβλημα του Περιοδεύοντος Πωλητή

Το πρόβλημα του περιοδεύοντος πωλητή (TSP) χρησιμοποιήθηκε για να επιβεβαιωθεί η υπεροχή των ΣΠΔΕΣ έναντι του μοντέλου νησίδων, αλλά και για να δοκιμαστεί η απόδοση των ΣΠΔΕΣ σε συνδυαστικά (combinatorial) προβλήματα. (Όπως έχει ήδη τονιστεί, κάθε υπο-πληθυσμός του μοντέλου νησίδων έχει την συμπεριφορά του υπο-πληθυσμού 1 των ΣΠΔΕΣ).

Εδώ κάθε σειρά πειραμάτων αποτελείται από 450 πειράματα (συνολικά 900 πειράματα). Οι τιμές εποχής που χρησιμοποιήθηκαν είναι 1, 10 και 50. Οι τιμές μεγέθους πληθυσμού είναι 50, 100 και 200 άτομα ανά υπο-πληθυσμό. Τα αποτελέσματα είναι οι μέσοι όροι από 50 εκτελέσεις για κάθε πείραμα.

Όπως έχει ήδη τονιστεί, η επιλογή της εξελικτικής συμπεριφοράς κάθε υπο-πληθυσμού των ΣΠΔΕΣ δεν είναι η βέλτιστη δυνατή και δεν είναι προσανατολισμένη στο πρόβλημα TSP. Δεν χρησιμοποιούμε κωδικοποίηση ή τελεστές που βοηθούν στην καλύτερη διερεύνηση του χώρου λύσεων, έτσι ώστε η δομή των ΣΠΔΕΣ να είναι παρόμοια και στα τρία προβλήματα..

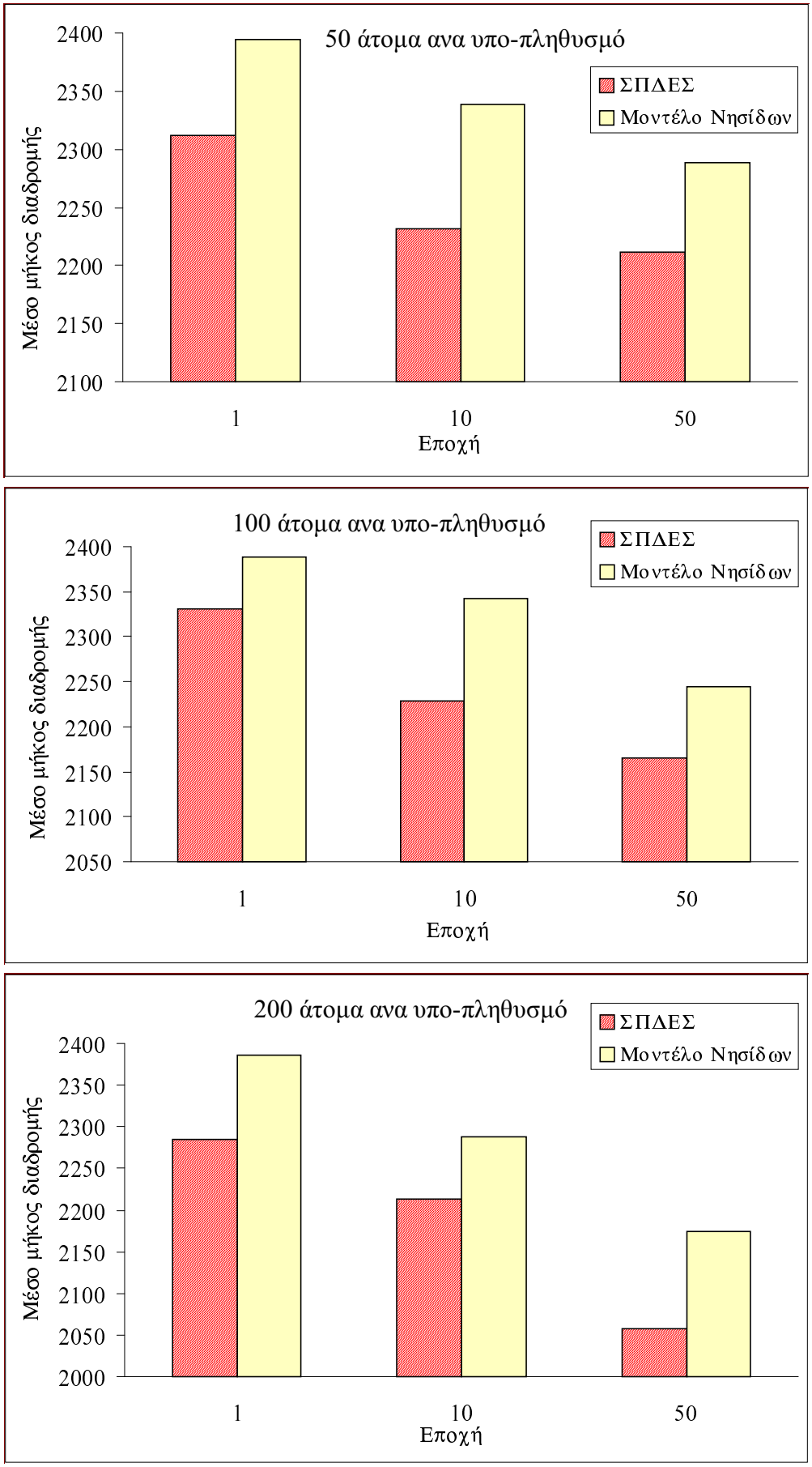
Και σε αυτό το πρόβλημα παρατηρείται καλύτερη απόδοση των ΣΠΔΕΣ έναντι του μοντέλου νησίδων (σχ. 6.15, 6.17).

Η απόδοση, τόσο των ΣΠΔΕΣ όσο και του μοντέλου νησίδων, είναι καλύτερη χρησιμοποιώντας μεγαλύτερη τιμή *εποχής*, επιτυγχάνοντας μικρότερο μέσο μήκος διαδρομής. Η χρήση της τιμής *εποχής* 50 είχε τα καλύτερα αποτελέσματα Την χειρότερη απόδοση είχε η χρήση της τιμής *εποχής* 1, δηλαδή όταν οι υπο-πληθυσμοί επικοινωνούν αμέσως μετά την δημιουργία της νέας γενιάς (σχ. 6.15).

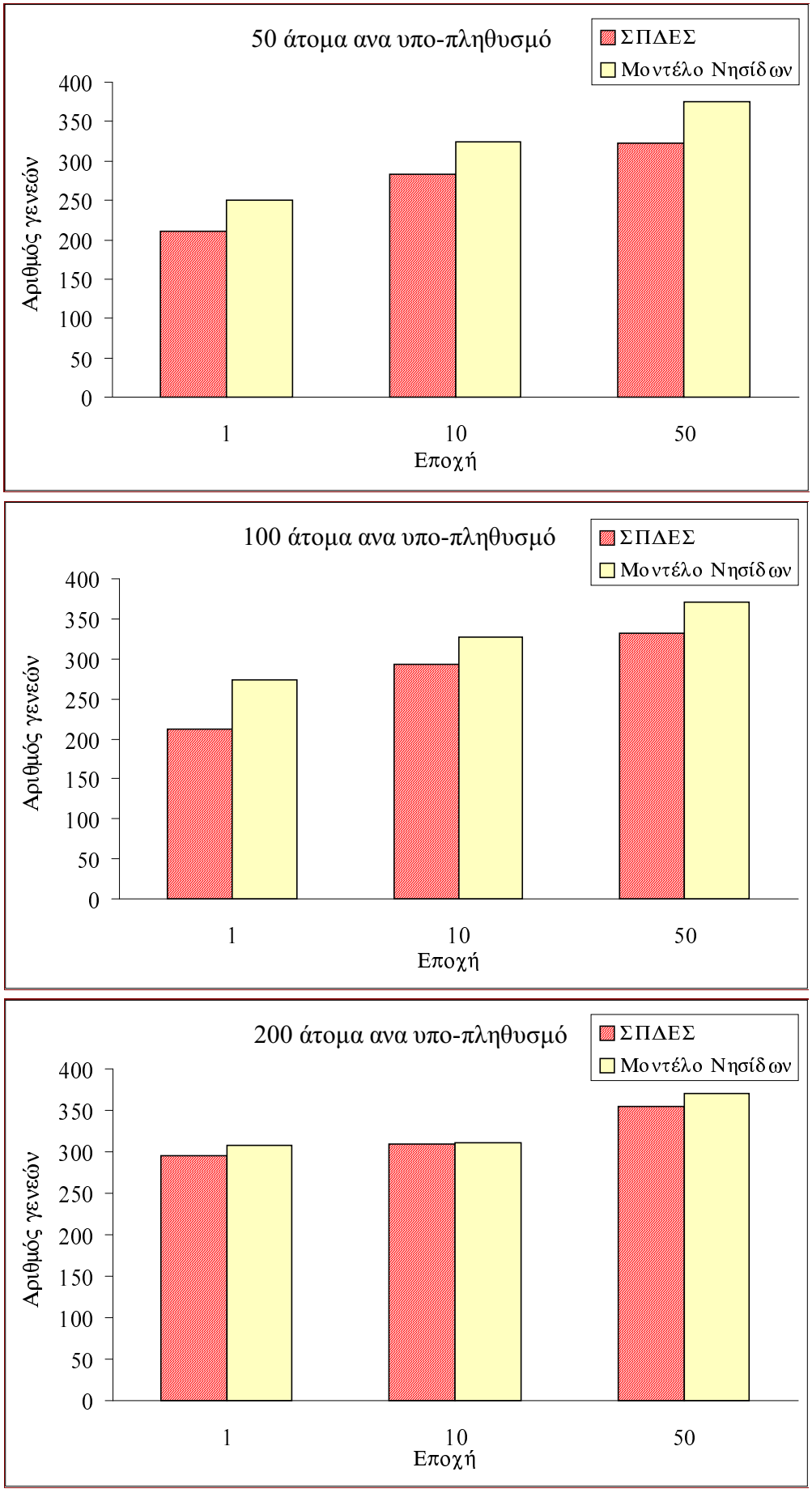
Καθώς όμως αυξάνεται η τιμή *εποχής*, μειώνεται η ταχύτητα σύγκλισης και των δύο μοντέλων. Έτσι η χρήση της τιμής *εποχής* 1 συγκλίνει με μεγαλύτερη ταχύτητα απαιτώντας μικρότερο αριθμό γενεών (επομένως λιγότερο χρόνο) για όλα τα μεγέθη υπο-πληθυσμού που χρησιμοποιήθηκαν. Όσο αυξάνεται η τιμή *εποχής* που χρησιμοποιείται τόσο αυξάνεται και ο απαιτούμενος αριθμός γενεών για να συγκλίνει ο αλγόριθμος. Από τις τρεις τιμές που χρησιμοποιήθηκαν η μεγαλύτερη τιμή (*εποχή*=50) είχε την βραδύτερη εξέλιξη (σχ. 6.16).

Αντίθετα με την εμφανή επίδραση της τιμής εποχής, παρατηρούμε ότι η χρήση διαφορετικών μεγεθών πληθυσμού δεν έχει μεγάλη επίδραση στην απόδοση των δύο μοντέλων κυρίως στις μικρές τιμές *εποχής* (1,10). Στην τιμή εποχής 50 παρατηρείται αύξηση της απόδοσης (μικρότερο μήκος διαδρομής) καθώς αυξάνεται το μέγεθος πληθυσμού (σχ. 6.17). Αυτό οφείλεται στο ότι το πρόβλημα είναι αρκετά δύσκολο (κυρίως λόγω της κωδικοποίησης και των εξελικτικών συμπεριφορών που επιλέξαμε) με αποτέλεσμα να μην προλαβαίνουν οι υπο-πληθυσμοί να σχηματίσουν χρήσιμα τμήματα κατασκευής όταν η επικοινωνία μεταξύ τους είναι πολύ συχνή.

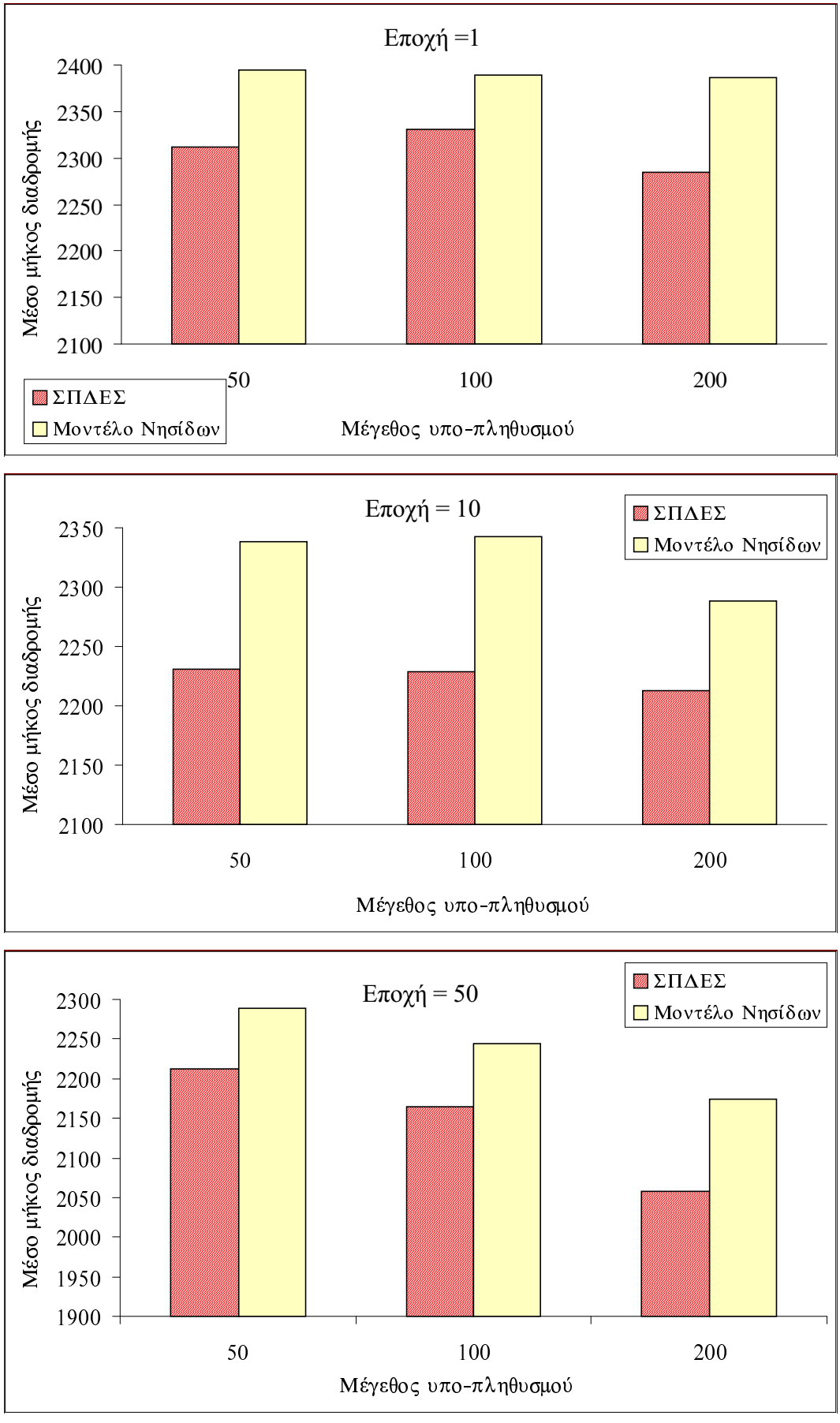
Αν και η συχνή επικοινωνία μεταξύ των υπο-πληθυσμών αυξάνει τις επικοινωνίες μεταξύ των επεξεργαστών, επειδή ο χρόνος επεξεργασίας είναι συγκριτικά πολύ μεγάλύτερος, η επίδρασή των επικοινωνιών στην ταχύτητα των πειραμάτων είναι σχετικά μικρή. Το σχήμα 6.18 δείνει ενδεικτικά, τους χρόνους των πειραμάτων για τιμή εποχής 10. Ὅπως αναμενόταν τα πειράματα με μεγάλο μέγεθος πληθυσμού απαιτούν περισσότερο χρόνο.



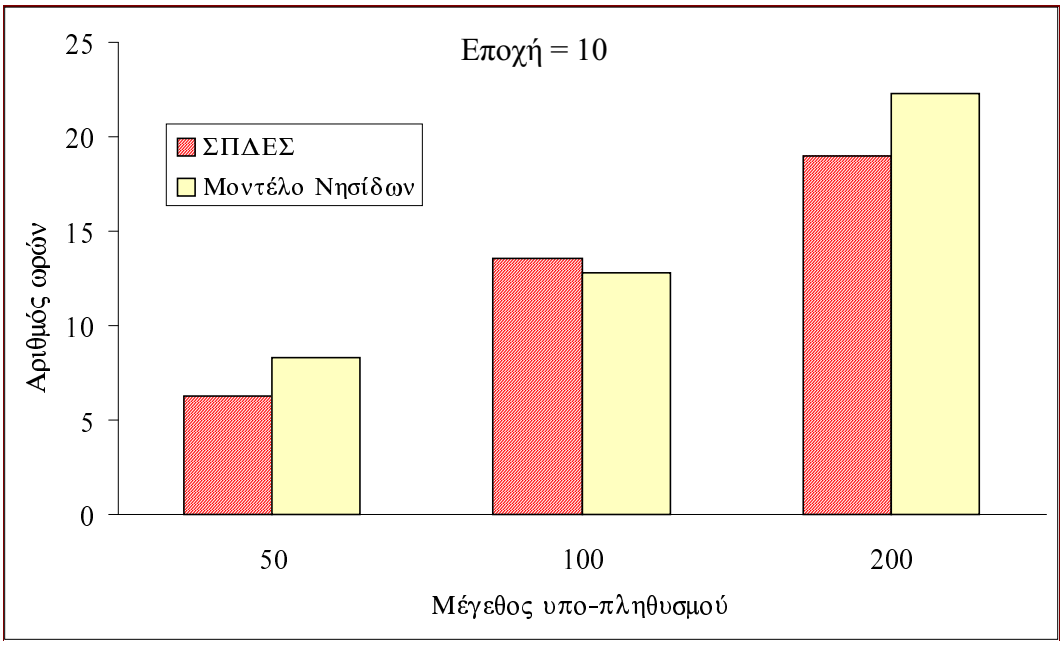
Σχήμα 6.15. Απόδοση των ΣΠΔΕΣ και του μοντέλου νησίδων για διάφορα μεγέθη υπο-πληθυσμών στο πρόβλημα TSP



Σχήμα 6.16. Μέσος αριθμός γενεών για διάφορα μεγέθη υπο-πληθυσμών στο πρόβλημα TSP



Σχήμα 6.17. Σύγκριση απόδοσης διαφορετικών μεγεθών υπο-πληθυσμού των ΣΠΔΕΣ και του μοντέλου νησίδων για διάφορες τιμές *εποχής* στο πρόβλημα TSP



Σχήμα 6.18. Μέσος αριθμός ωρών ανά πείραμα (TSP)

Τα αποτελέσματα του προβλήματος του περιοδεύοντος πωλητή συμφωνούν με τα αποτελέσματα από τα προηγούμενα προβλήματα. Δηλαδή, η επιλογή των κατάλληλων τιμών *εποχής* και μεγέθους των υπο-πληθυσμών (*ΜεΠ*), εξαρτάται από το πρόβλημα, από τον διαθέσιμο χρόνο που υπάρχει καθώς και την ποιότητα της λύσης που απαιτείται. Μεγαλύτερες τιμές *εποχής* και *ΜεΠ* οδηγούν στο ολικό ελάχιστο με μεγαλύτερη πιθανότητα από ότι μικρότερες τιμές *εποχής* και *ΜεΠ*. Αντίθετα μεγαλύτερες τιμές *εποχής* οδηγούν στην βραδύτερη σύγκλιση τόσο των ΣΠΔΕΣ όσο και του μοντέλου νησίδων.

Κεφάλαιο 7

Συμπεράσματα

Το μοντέλο των *Συνεργαζόμενων Πληθυσμών*, είναι ικανό να ορίσει οιοδήποτε τύπο εξελικτικού αλγόριθμου. Χρησιμοποιώντας αυτό το μοντέλο ορίσαμε και υλοποιήσαμε τους *ΣΠΔΕΣ*, ένα γενικότερο μοντέλο ΠΓΑ βασισμένο στην ανομοιομορφία του τρόπου εξέλιξης των υπο-πληθυσμών.

Οι ΣΠΔΕΣ είναι μία υλοποίηση του πρώτου επιπέδου των *Συνεργαζόμενων Πληθυσμών (ΣΠ ^{1})*. Η μελέτη της απόδοσης μοντέλων βασισμένων σε επίπεδα μεγαλύτερα του πρώτου είναι μία πιθανή κατεύθυνση περαιτέρω έρευνας.

Οι σημαντικότεροι παράμετροι που επηρεάζουν την σχεδίαση και την υλοποίηση των ΠΓΑ είναι η μετανάστευση γόνων και το μέγεθος των υπο-πληθυσμών (*ΜεΠ*).

Με την εισαγωγή της έννοιας της μετανάστευσης, αξιοποιούνται οι διαφορές μεταξύ των υπο-πληθυσμών, οι οποίες είναι πηγή γενετικής απόκλισης και ποικιλίας.

Η μετανάστευση γόνων είναι μια σύνθετη παράμετρος και εξαρτάται από

- την συχνότητα μετανάστευσης. Αυτό θα πρέπει να γίνει αφού έχει σχηματιστεί ο απαραίτητος αριθμός τμημάτων κατασκευής σε κάθε γόνο. Μεταναστεύοντας νωρίτερα είναι συνήθως σπατάλη πόρων. Γενικά, τα σωστά τμήματα κατασκευής επηρεάζουν την αναζήτηση προς την σωστή κατεύθυνση. Γι'αυτό η μετανάστευση δεν πρέπει να γίνεται πολύ νωρίς πριν σχηματιστούν άτομα με κάποια σωστά και απαραίτητα τμήματα στην παραπέρα εξέλιξη των πληθυσμών.
- το ποσοστό μετανάστευσης. Ο αριθμός των γόνων που μεταναστεύουν είναι στενά συνδεμένος με την συχνότητα μετανάστευσης. Εάν οι γόνοι είναι 'πλούσιοι' σε τμήματα κατασκευής, τότε μπορούμε να πάρουμε καλά αποτελέσματα με την μετανάστευση λίγων.
- την τοπολογία. Ποια είναι η καλύτερη μέθοδος σύνδεσης των υπο-πληθυσμών; Εάν επιλεγεί τοπολογία με μεγάλη διάμετρο, οι καλές λύσεις θα αργήσουν να διαδοθούν σε ολόκληρο τον πληθυσμό. Εάν η τοπολογία επιτρέπει την ταχεία διάδοση των καλών

λύσεων αυτό είναι δυνατό να οδηγήσει στην ταχεία σύγκλιση του αλγόριθμου σε κάποιο τοπικό βέλτιστο.

Ο αριθμός των ατόμων που ανταλλάσσονται μεταξύ των υπο-πληθυσμών (ποσοστό μετανάστευσης), είναι άμεσα συνδεμένος με την τιμή της *εποχής* μετανάστευσης. Μικρή τιμή *εποχής* και πολύ λίγοι μετανάστες, ή πολύ μεγάλη τιμή *εποχής* και πάρα πολλοί μετανάστες, υποβαθμίζουν την απόδοση του αλγορίθμου.

Σύμφωνα με τον D. Whitley, εάν η μετανάστευση ατόμων μεταξύ των υπο-πληθυσμών είναι πολύ σπάνια (μεγάλη τιμή *εποχής* μετανάστευσης), τότε αυτή η μετανάστευση μπορεί να μην είναι αρκετή να εμποδίσει την πρόωρη σύγκλιση του πληθυσμού [Whit91]. Η μετανάστευση ατόμων πολύ συχνά ή όχι αρκετά συχνά μειώνουν την απόδοση των ΠΓΑ [StWM90].

Τα αποτελέσματα των πειραμάτων της παρούσας διατριβής δείχνουν ότι η μη συχνή μετανάστευση βοηθά στην αποφυγή της πρόωρης σύγκλισης του συνόλου των πληθυσμών, δηλαδή του μοντέλου των ΠΓΑ, ακόμη και αν κάποιοι από αυτούς κατά την διάρκεια της περιόδου απομόνωσης έχουν συγκλίνει σε κάποιο τοπικό βέλτιστο. Τα συμπεράσματα που αναφέρονται από τους Whitley κλπ., οφείλονται στον περιορισμένο αριθμό γενεών που άφησαν τους υπο-πληθυσμούς να εξελιχθούν στα πειράματά τους.

Όταν οι υπο-πληθυσμοί επικοινωνούν αραιά, έχουν την δυνατότητα, κατά την διάρκεια της περιόδου απομόνωσης, να ερευνήσουν διαφορετικές περιοχές του χώρου λύσεων και να διαθέτουν διαφορετικές γενετικές πληροφορίες. Όταν αυτές οι πληροφορίες εισάγονται σε άλλους υπο-πληθυσμούς, οι οποίοι έχουν συγκλίνει πρόωρα σε κάποιο τοπικό βέλτιστο, τους βοηθούν να ξεφύγουν και να συνεχίσουν την αναζήτηση του ολικού βέλτιστου, βοηθώντας στην επίλυση του προβλήματος. Χωρίς καθόλου μετανάστευση, οι υπο-πληθυσμοί, και κατά συνέπεια το μοντέλο, θα είχαν συγκλίνει πολύ νωρίτερα σε κάποιο τοπικό βέλτιστο.

Όταν η μετανάστευση γίνεται πολύ συχνή, οι υπο-πληθυσμοί οδηγούνται στην διερεύνηση της ίδιας περιοχής του χώρου λύσεων. Έτσι χάνουν την ποικιλία και την απόκλιση του γενετικού τους υλικού και συνήθως συγκλίνουν πρόωρα σε κάποιο τοπικό βέλτιστο, αν και αρχικά η συχνή μετανάστευση βοηθά στην αποφυγή της πρόωρης σύγκλισης. Η απώλεια της ποικιλίας του γενετικού υλικού στην περίπτωση της πολύ συχνής μετανάστευσης, έχει ήδη αναφερθεί από τον Belding [Beld95] για τους χαΠΓΑ. Σύμφωνα με τα αποτελέσματα του προηγούμενου κεφαλαίου, το συμπέρασμα αυτό ισχύει τόσο για τους ΣΠΔΕΣ όσο και για το μοντέλο νησίδων. Αν και η πολύ συχνή επικοινωνία βοηθά στην γρήγορη διερεύνηση των ελπιδοφόρων περιοχών του χώρου λύσεων, επίσης οδηγεί όλους τους υπο-πληθυσμούς στην διερεύνηση της ίδιας περιοχής. Έτσι χάνεται γενετικό υλικό και πληροφορίες σχετικές με άλλες περιοχές του χώρου λύσεων, οδηγώντας στην πρόωρη σύγκλιση του μοντέλου σε κάποιο τοπικό βέλτιστο.

Η απομόνωση των πληθυσμών είναι ένα σημαντικό στοιχείο της εξέλιξης των ειδών. Η σημασία της είναι μεγάλη στην παρεμπόδιση, της μετανάστευσης των καλύτερα προσαρμοσμένων οργανισμών, δηλαδή των καλύτερων ατόμων, και έτσι νέες καλύτερες ποικιλίες (άτομα) στην φυσική οικονομία της περιοχής θα μείνουν ανοιχτές για να καλυφθούν από την εξέλιξη των παλαιότερων ατόμων.

Η τιμή *εποχής* που θα επιλεγεί έχει άμεση επίπτωση στον αριθμό γενεών που απαιτούνται για να συγκλίνει ο αλγόριθμος (ταχύτητα σύγκλισης), όπως επίσης και στην ποιότητα της λύσης που θα επιτευχθεί. Έτσι, ενώ η χρήση μεγάλης τιμής *εποχής* μειώνει τον χρόνο επικοινωνίας μεταξύ

των επεξεργαστών, επειδή αυξάνεται πολύ ο απαιτούμενος αριθμός γενεών, έχουμε μείωση της ταχύτητας σύγκλισης και αύξηση του συνολικού χρόνου επεξεργασίας. Αντίθετα, αν και η μικρή τιμή *εποχής* αυξάνει τις επικοινωνίες μεταξύ των επεξεργαστών, τελικά μειώνεται ο χρόνος επεξεργασίας λόγω της αύξησης της ταχύτητας σύγκλισης. Βέβαια με την χρήση μικρής τιμής *εποχής* μειώνουμε τις πιθανότητες εύρεσης του βέλτιστου.

Εάν δεν μας ενδιαφέρει η ταχύτητα σύγκλισης και ο αριθμός των γενεών που χρειάζονται για να συγκλίνει ο αλγόριθμος, η καλύτερη επιλογή είναι η χρήση μιας τιμής *εποχής* όσο το δυνατό μεγαλύτερης. Ωστόσο, οι μεγάλες τιμές εποχής αυξάνουν αισθητά τον αριθμό γενεών. Η επιλογή της κατάλληλης τιμής *εποχής* εξαρτάται από το πρόβλημα και τον διαθέσιμο χρόνο που υπάρχει για να βρεθεί μια καλή λύση. Υπάρχουν δύο διακριτές περιπτώσεις:

- Μικρή τιμή εποχής προτείνεται όταν το πρόβλημα είναι σχετικά εύκολο, επειδή σ'αυτή την περίπτωση η σύγκλιση στο ολικό βέλτιστο θα χρειαστεί μικρό αριθμό γενεών, καθώς επίσης και όταν είναι αναγκαίο να βρεθεί μία αρκετά καλή λύση (όχι το ολικό βέλτιστο) όσο το δυνατό ταχύτερα.

- Μεγάλη τιμή εποχής προτείνεται όταν επιλύεται ένα δύσκολο πρόβλημα ή δεν υπάρχει περιορισμός χρόνου καθώς επίσης και όταν είναι απαραίτητο να βρεθεί το ολικό βέλτιστο. Μεγάλη τιμή εποχής είναι απαραίτητη στις περιπτώσεις όπου η επανάληψη του πειράματος είναι δύσκολη ή αδύνατη, είτε λόγω κόστους , είτε λόγω των συνθηκών του πειράματος.

Το μέγεθος των υπο-πληθυσμών (*ΜεΠ*) είναι η άλλη σημαντική παράμετρος που επηρεάζει την απόδοση των δύο μοντέλων (ΣΠΔΕΣ και μοντέλο νησίδων), όπως και την απόδοση οιουδήποτε εξελικτικού αλγόριθμου.

Το *ΜεΠ* είναι ίσως η παράμετρος που επηρεάζει περισσότερο την απόδοση των ΣΓΑ Υπάρχει κάποια θεωρία για το μέγεθος του πληθυσμού των Σ.Γ.Α. η οποία βασίζεται στη θεωρία στατιστικής λήψης αποφάσεων και η οποία δίνει ένα κατώτερο όριο πληθυσμού για την επίλυση προβλημάτων με κάποιο συντελεστή βεβαιότητας [GoDC92]. Στους ΠΓΑ το *ΜεΠ* είναι στενά συνδεδεμένο με τις επιλογές των υπο-πληθυσμών. Οι επιλογές αυτές έχουν σχέση κυρίως με το πλήθος και το *ΜεΠ* που χρειάζεται ο ΠΓΑ για να επιλύσει ένα πρόβλημα με κάποια βεβαιότητα. Αυτό το θέμα εξετάζεται από τους Goldberg, Kargupta, Horn, και Cantu-Paz [GKHC95] και επιλύεται για την ακραία περίπτωση όπου γίνεται δεκτή τέλεια μίξη μεταξύ των υπο-πληθυσμών και για την περίπτωση των απομονωμένων υπο-πληθυσμών. Όπως σχολιάζουν, "ο βέλτιστος καθορισμός του *ΜεΠ* εξαρτάται από τις επικοινωνίες μεταξύ των υπο-πληθυσμών και από την αρχιτεκτονική της μηχανής.

Σύμφωνα με τα αποτελέσματα του προηγούμενου κεφαλαίου για τα δύο μοντέλα, η αύξηση του *ΜεΠ*, και κατά συνέπεια του μεγέθους του συνολικού πληθυσμού, βελτιώνει την μακροχρόνια απόδοση του αλγόριθμου σε βάρος της ταχύτητας σύγκλισης και των υπολογισμών της συνάρτησης ποιότητας. Έτσι, μεγάλη τιμή *ΜεΠ* μπορεί να χρησιμοποιηθεί για να βελτιώσει την απόδοση "off-line", δηλαδή όταν υπάρχει αρκετός διαθέσιμος χρόνος, ενώ μικρή τιμή (μεγαλύτερη όμως του κρίσιμου *ΜεΠ*) απαιτείται όταν ο στόχος είναι η καλή απόδοση "on-line", δηλαδή όταν τα αποτελέσματα απαιτούνται σε σύντομο χρονικό διάστημα.

Η απομόνωση των πληθυσμών θα δώσει τον χρόνο σε κάποια νέα ποικιλία να βελτιωθεί με αργό ρυθμό, και αυτό μερικές φορές είναι μεγάλης σημασίας. Οι νεο-σχηματιζόμενες ποικιλίες, γενικά, αρχικά είναι τοπικές, που είναι και ο συνήθης κανόνας με τις ποικιλίες στη φύση, έτσι

ώστε όμοια τροποποιημένα άτομα συχνά αναπαράγονται μεταξύ τους. Εάν η νέα ποικιλία είναι επιτυχής στον αγώνα για επιβίωση, θα διαδοθεί αργά, ανταγωνιζόμενη τις υπόλοιπες και κατακτώντας άτομα στα όρια ενός διαρκώς αυξανόμενου κύκλου. Όταν όμως, ο συνολικός αριθμός των ατόμων είναι πολύ μικρός, αυτό θα επιβραδύνει την δημιουργία νέων ειδών, μειώνοντας τις πιθανότητες εμφάνισης ωφέλιμων ποικιλιών.

Η επιλογή μιας ενδιάμεσης τιμής *ΜεΠ* είναι μία γενική επιλογή που μπορεί να χρησιμοποιηθεί σε όλες τις περιπτώσεις.

Ο συνδυασμός της τιμής *εποχής* μετανάστευσης και του μεγέθους υπο-πληθυσμών που θα χρησιμοποιηθούν, εξαρτάται από την δυσκολία του προβλήματος, τον διαθέσιμο χρόνο, την επιθυμητή ποιότητα λύσης και τις ιδιαιτερότητες του πειράματος (π.χ. δυνατότητα επανάληψης, διαθέσιμος εξοπλισμός).

Σε ένα δύσκολο πρόβλημα ή όταν είναι δύσκολο να επαναληφθεί το πείραμα, είναι απαραίτητη η χρήση μεγάλης τιμής εποχής και μεγάλου *ΜεΠ*. Οι μεγάλες τιμές *εποχής* και *ΜεΠ* έχουν αρνητική επίδραση στην ταχύτητα και στο χρόνο σύγκλισης.. Εάν είναι απαραίτητο να βελτιωθεί ο χρόνος σύγκλισης, τότε είναι προτιμότερο να ελαττωθεί το μέγεθος των υπο-πληθυσμών παρά η τιμή της εποχής μετανάστευσης.

Σε ένα εύκολο πρόβλημα ή όταν επιθυμούμε ταχεία εύρεση κάποιας λύσης (όχι της βέλτιστης) χρησιμοποιούμε μικρές τιμές εποχής και *ΜεΠ*. Η εύρεση του βέλτιστου ενός δύσκολου προβλήματος, απαιτεί μεγάλης τιμής εποχής και μεγάλου *ΜεΠ*.

Τα αποτελέσματα που παρουσιάστηκαν φανερώνουν την ικανότητα των ΣΠΔΕΣ να βελτιώνουν την απόδοση των ΠΓΑ, επιταχύνοντας την σύγκλιση του αλγόριθμου, αντιμετωπίζοντας το πρόβλημα της πρόωρης σύγκλισης και παρέχοντας δυνατότητες καλύτερης διερεύνησης και εκμετάλλευσης του χώρου λύσεων. Επίσης βοηθούν στην αντιμετώπιση του προβλήματος επιλογής των καλύτερων τελεστών, μηχανισμών και παραμέτρων.

Η συνεργασία πληθυσμών με διαφορετικές εξελικτικές συμπεριφορές βοηθά την απόδοση του αλγόριθμου. Η ομοιομορφία της εξέλιξης των πληθυσμών τους καθιστά λιγότερο ανταγωνιστικούς.

Ερευνώντας ακόμη περισσότερο τους ΣΠΔΕΣ, είναι δυνατό να ελεγχθούν άλλες υλοποιήσεις του μοντέλου μέσω της επίδρασης διαφόρων άλλων στοιχείων στην απόδοσή τους όπως ρυθμός μετανάστευσης, διαφορετικοί μηχανισμοί επιλογής ανά υπο-πληθυσμό, διαφορετικές κωδικοποιήσεις ανά υπο-πληθυσμό, προσαρμογή του μεγέθους των υπο-πληθυσμών, διαφορετικά μεγέθη υπο-πληθυσμών, διαφορετικές στρατηγικές μετανάστευσης, διαφορετικές τοπολογίες σύνδεσης κ.α.

Μία περαιτέρω κατεύθυνση έρευνας είναι η σύγκριση της απόδοσης μεταξύ συνεργαζόμενων πληθυσμών και ανταγωνιζόμενων πληθυσμών. Η συνεργασία των πληθυσμών ορίζεται όπως στο τρίτο κεφάλαιο της παρούσας διατριβής. Ο ανταγωνισμός των πληθυσμών μπορεί να γίνει με αρκετούς τρόπους όπως:

- αντικατάσταση του χειρότερου ή ενός χαμηλής ποιότητας υπο-πληθυσμού από άλλον (τον καλύτερο ή κάποιον τυχαίο)
- δημιουργία νέου υπο-πληθυσμού με ποιοτικούς γόνους από άλλους υπο-πληθυσμούς και αντικατάσταση ενός κάτω του μετρίου υπο-πληθυσμού

Στόχος είναι ο ορισμός ενός μοντέλου με στοιχεία συνεργασίας αλλά και ανταγωνισμού μεταξύ των υπο-πληθυσμών ή η επιλογή μεταξύ συνεργασίας και ανταγωνισμού.

Βιβλιογραφία

[Abra92] Abramson D. (1992). A parallel genetic algorithm for solving the school timetabling problem. In *Proceedings of the 15th Australian Computer Science Conf.*, Hobart.

[AbMP92] Abramson D., Mills G., & Perkins S. (1993). Parallelisation of a genetic algorithm for the computation of efficient train schedules. In *Proceedings of 1993 Parallel Computing and Transputers Conf.*, Brisbane.

[Adam94] Adamidis, P. (1994). Review of parallel genetic algorithms bibliography, Internal Technical Report, Automation and Robotics Lab., Dept. of Electrical and Computer Eng., Aristotle University of Thessaloniki, Greece (http://www.control.ee.auth.gr/~panos/papers/pga_review.ps.gz).

[AdPe96] Adamidis, P., & Petridis, V. (1996). Co-operating populations with different evolution behavior. *Proceedings of the Second IEEE Conference on Evolutionary Computation.*

[ATBM92] Ahuactzin J.M., Talbi A-G., Bessiere P., & Mazer E. (1992). Using genetic algorithms for robot motion planning. In *Proceedings of European Conference on Artificial Intelligence.*

[Bäck93] Bäck, T. (1993) Oprimal mutation rates in genetic search. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp.2-8). San Mateo, CA: Morgan Kaufmann.

[Bäck94a] Bäck, T. (1994a). Parallel optimization of evolutionary algorithms. In Y. Davidor, H-P. Schwefel and R. Manner (Eds.), *Proceedings of the Third Conference on Parallel Problem Solving from Nature* (pp.418-427), vol 866 of Lecture Notes in Computer Science, Berlin: Springer-Verlag.

[Bäck94b] Bäck, T. (1994b). Selective pressure in evolutionary algorithms: A Characterization of Selection Mechanisms, *Proceedings of the First IEEE Conf. on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, vol I, (pp.57-62)

[BäHo91] Bäck, T., & Hoffmeister, F. (1991). Extended selection mechanisms in genetic algorithms. In R-K. Belew and L. B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp.92-99). San Mateo, CA: Morgan Kaufmann.

[BaPK94] Bakirtzis, A., Petridis, V., & Kazarlis, S. (1994). A genetic algorithm solution to the economic dispatch problem. *IEE Proceedings - Gener. Transm. Distrib.,* vol 141 (4), 377-382.

[Balu92] Baluja S. (1992). A massively distributed parallel genetic algorithm (mdpGA). Tech. Report CMU-CS-92-196, Carnegie Mellon University, School of Computer Science.

[Balu93] Baluja S. (1993). Structure and performance of fine-grain parallelism in genetic search. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp.155-162). San Mateo, CA: Morgan Kaufmann.

[Beld95] Belding, T.C. (1995). The distributed genetic algorithm revisited. In D. Eshelman (Ed.) *Procceedings of the Sixth International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufman.

[BATM93] Bessiere P., Ahuactzin J-M., Talbi E-G., & Mazer E. (1993). The Ariadne's clew algorithm: Global planning with local methods. In *Proceedings of the IEEE-IROS'93 Conference on Intelligent Robots and Systems*, Yokohama, Japan.

[Beye93] Beyer, H-G. (1993). Optimization of large-scale order problems by the Evolution Strategy. In R. Schumacher (Ed.), *One Year KSR1 at the University of Mannheim - Results & Experiences*. Technical Report RUM 35/93, University of Mannheim, December.

[BiBr93] Bianchini, R., & Brown, C., (1993). Parallel genetic algorithms on distributed-memory architectures. In S.Atkins and A.S. Wagner(Eds.), *Transputer Research and Applications, Proceedings of the Sixth Conference of the North American Transputer Users Group* (pp. 67-82). Berlin: IOS Press. (ftp://ftp.cs.rochester.edu/pub/systems_papers/93.tr436.parallel_genetic_ algorithms.ps.Z)

[Bram91] Bramlette M.F. (1991). Initialization, mutation and selection methods in genetic algorithms for function optimization. In R-K. Belew and L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp.100-107). San Mateo, CA: Morgan Kaufmann

[Brau90] Braun, H.. (1990). On travelling salesman problems by genetic algorithms. In H.-P.Schwefel and R.Manner (Eds.), *Proceedings of the First Workshop on Parallel Problem Solving from Nature*, vol. 496 of Lecture Notes in Computer Science (pp.129-133). Berlin : Springer-Verlag.

[BrHS89] Brown D.E., Huntley C.L., & Spillane A.R. (1989). A parallel genetic heuristic for solving the quadratic assignment problem. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp.406-415). San Mateo, CA: Morgan Kaufmann Publishers

[BuMo94] Bui, T., & Moon, B. (1994). A new genetic approach for the travelling salesman problem. *Proceedings of the IEEE World Congress on Computational Intelligence, First IEEE Conference on Evolutionary Computation*, vol I, (pp. 7-12).

[Cant95] Cantú-Paz, E. (1995). A summary of research on parallel genetic algorithms. IlliGAL Report No. 95007, Illinois Genetic Algorithms Lab., Dept. of General Eng., University of Illinois at Urbana-Champaign, July. (ftp://gal4.ge.uiuc.edu:pub/papers/IlliGALs/95007.ps.Z)

[CaMe94] Cantú-Paz, E., & Mejía-Olvera, M., (1994). Experimental results in distributed genetic algorithms. In *International Symposium on Applied Corporate Computing* (pp.99-108). Montery, Mexico.

[CHMR87] Cohoon, J.P., Hedge S., Martin, W.N., & Richards, D.S. (1990). Punctuated Equilibria: A parallel genetic algorithm. In J. Grefenstette (Ed.), *Proceedings of the Second International Conference on Genetic Algorithms* (pp.148-154). Cambridge, MA: Lawrence Erlbaum Associates.

[CoMR90] Cohoon, J.P., Martin, W.N., & Richards, D.S. (1990). Genetic algorithms and punctuated equilibria in VLSI. In H.-P.Schwefel and R.Manner (Eds.), *Proceedings of the First Workshop on Parallel Problem Solving from Nature*, vol. 496 of Lecture Notes in Computer Science (pp.134-144). Berlin : Springer-Verlag.

[CoMR91] Cohoon, J.P., Martin, W.N., & Richards, D.S. (1991). A multi-population genetic algorithm for solving the K-partition problem on hyper-cubes. In R-K. Belew and L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp.244-248). San Mateo, CA: Morgan Kaufmann

[CoJe91] Collins R.J., & Jefferson D.R. (1991). Selection in massively parallel genetic algorithms. In R-K. Belew, L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp.249-256). San Mateo, CA: Morgan Kaufmann.

[CuFo92] Cui, J., & Fogarty, T.C. (1992). Optimization by using a parallel genetic algorithm on a transputer computing surface. In M. Varelo, E. Onate, M. Jane, J.L. Larriba, & B. Suarez (Eds.), *Parallel Computing and Transputer Applications, Transputer and Occam Engineering Series*, vol. 28 (pp.246-254). Amsterdam: IOS Press.

[Darw59] Darwin, C. (1859). On the origin of species by means of natural selection. Penguin.

[Davi91] Davidor, Y. (1991). A naturally occuring niche & species phenomenon: The model and first results. In R-K. Belew and L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp.257-263). San Mateo, CA: Morgan Kaufmann.

[DaYN] Davidor, Y., Yamada, T., & Nakano, R. (1993). The ECOlogical framework II: Improving GA performance at virtually zero cost. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp.171-176). San Mateo, CA: Morgan Kaufmann.

[Davi89] Davis, L. (1989). Adapting operator probabilities in genetic algorithms. In J.D. Schaffer(Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 61-69) San Mateo, CA: Morgan Kaufmann.

[Davi91] Davis, L. (Ed.) (1991). *Handbook of genetic algorithms*. New York: Van Norstrand.

[DJSp92] De Jong K.A., & Spears W.M. (1992). A formal analysis of the role of multi-point crossover in Genetic Algorithms. *Annals of Mathematics and Artificial Intelligence*, 5(1), 1-26.

[dlMT93] de la Maza, M., & Tidor B. (1993). An analysis of selection procedures with particular attention paid to proportional and Boltzmann selection. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp.124-131). San Mateo, CA: Morgan Kaufmann.

[DeRi93] Dekker, L., & Ribeiro Filho, J.L. (1993). The GAME virtual machine architecture. In J. Stender (Ed.), *Parallel Genetic Algorithms: Theory and Applications* (pp.93-110). Brainware GmbH, Berlin: IOS Press.

[Dori92] Dorigo M. (1992) Using transputers to increase speed and flexibility of genetics-based machine learning systems. *Microprocessing and Microprogramming* 34 (1-5), 147-152.

[DoMa93] Dorigo, M., & Maniezzo, V. (1993). Parallel genetic algorithms: Introduction and overview of current research. In J. Stender (Ed.), *Parallel Genetic Algorithms: Theory and Applications* (pp.5-42). Brainware GmbH, Berlin: IOS Press.

[DzWh94] Dzubera, J., & Whitley, D. (1994). Advanced correlation analysis of operators for the traveling salesman problem. In Y.Davidor, H-P. Schwefel, and R. Manner (Eds.), *Proceedings of the Third Conference on Parallel Problem Solving from Nature* (pp.68-77). Berlin: Springer-Verlag.

[EaMa93] East, I., & Macfarlane, D. (1993). Implementation in Occam of parallel genetic algorithms on transputer networks. In J. Stender (Ed.), *Parallel Genetic Algorithms: Theory and Applications* (pp.43-64). Brainware GmbH, Berlin: IOS Press.

[Foga89] Fogarty, T.C. (1989). Varying the probability of mutation in the genetic algorithm. In J.D. Schaffer(Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 104-109) San Mateo, CA: Morgan Kaufmann.

[FoHu90] Fogarty, T.C., & Huang, R. (1990). Implementing the genetic algorithm on transputer based parallel processing systems. In H.-P.Schwefel and R.Manner (Eds.), *Proceedings of the First Workshop on Parallel Problem Solving from Nature*, vol. 496 of Lecture Notes in Computer Science (pp.145-149). Berlin : Springer-Verlag.

[FoOW66] Fogel, L.J., Owens A.J., & Walsh M.J. (1966) *Artificial Intelligence through simulated evolution*. New York: Wiley Publishing.

[Gold89a] Goldberg, D.E. (1989a). *Genetic algorithms in search optimization and machine learning*. Reading, MA: Addison-Wesley.

[Gold89b] Goldberg, D.E. (1989b). Sizing populations for serial and parallel genetic algorithms. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp.70-79). San Mateo, CA: Morgan Kaufmann.

[GoDe91] Goldberg, D.E., & Deb K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In G. Rawlins (Ed.), *Foundations of Genetic Algorithms* (pp.69-93). San Mateo, CA: Morgan Kaufmann.

[GoDC92] Goldberg, D.E., Deb K., & Clark J. (1992). Genetic algorithms, noise and the sizing of population. *Complex Systems* 6, 333-362.

[GKHC95] Goldberg, D.E., Kargupta, H., Horn, J., & Cantu-Paz, E. (1995). Critical deme size for serial and parallel genetic algorithms. IlliGAL Report No. 95002, Illinois Genetic Algorithms Lab., Dept. of General Eng., University of Illinois at Urbana-Champaign, January. (ftp://gal4.ge.uiuc.edu:pub/ papers/IlliGALs/95002.ps.Z)

[GoWh93] Gordon, V., & Whitley, D. (1993). Serial and parallel genetic algorithms as function optimizers. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp.177-183). San Mateo, CA: Morgan Kaufmann.

[Gorg89] Gorges-Schleuter, M. (1989). ASPARAGOS An asynchronous parallel genetic optimization strategy. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp.422-427). San Mateo, CA: Morgan Kaufmann.

[Gorg90] Gorges-Schleuter M. (1990). Explicit parallelism of genetic algorithms through population structures. In H.-P.Schwefel and R.Manner (Eds.), *Proceedings of the First Workshop on Parallel Problem Solving from Nature*, vol. 496 of Lecture Notes in Computer Science (pp. 150-159). Berlin: Springer-Verlag.

[Gorg92] Gorges-Schleuter M. (1992). Comparison of local mating strategies in massively parallel genetic algorithms. In R.Manner and B.Manderick (Eds.), *Proceedings of the Second Workshop on Parallel Problem Solving from Nature* (pp. 553-562). Amsterdam: Elsevier Science Publishers.

[Gref86] Grefenstette, J.J. (1986) Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cubernetics*, 16, 122-128.

[Gros85] Grosso, P. (1985). *Computer simulations of genetic adaptation: Parallel subcomponenet interaction in multilocus model.* PhD. Thesis, University of Michigan.

[Grua94] Gruau F. (1994*). Neural network synthesis using cellular encoding and the genetic algorithm.* PhD. Thesis, Ecole Normale Superieure de Lyon, Laboratoire de l'Informatique du Parallelisme (LIP-IMAG), January.

[HaMa94] Hauser, R., & Manner, R. (1994). Implementation of standard genetic algorithm on MIMD machines. In Y. Davidor, H-P. Schwefel, and R. Manner (Eds.), *Proceedings of the Third Conference on Parallel Problem Solving from Nature,* vol 866 of Lecture Notes in Computer Science (pp.504-513). Berlin: Springer-Verlag.

[HoBä90] Hoffmeister, F., & Bäck, T. (1990). Genetic algorithms and evolution strategies: similarities and differences. In H.-P.Schwefel and R.Manner (Eds.), *Proceedings of the First*

Workshop on Parallel Problem Solving from Nature, vol. 496 of Lecture Notes in Computer Science (pp. 455-471). Berlin: Springer-Verlag.

[Holl75] Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, Michigan: The University Press.

[HoGL93] Homaifar, A., Guan, S., & Liepins, G.E. (1993). A new approach on the traveling salesman problem by genetic algorithms. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp.460-466). San Mateo: Morgan Kaufmann.

[HuMi91] Husbands, P., & Mill, F. (1991). Simulated co-evolution as the mechanism for emergent planning and scheduling. In R-K. Belew, L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp.264-270). San Mateo, CA: Morgan Kaufmann.

[JoSV89] Jog, P., Suh, J.Y., & Van Gucht, D. (1989). The effects of population size, heuristic crossover and local improvement on a genetic algorithm for the travelling salesman problem. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp.104-109). San Mateo, CA: Morgan Kaufmann

[KaBP94] Kazarlis, S., Bakirtzis, A., & Petridis, V. (1994). A genetic algorithm solution to the unit commitment problem.

[KiRT93] Kingdon J., Ribeiro Filho J.L., & Treleaven P. (1993). The GAME programming environment architecture. In J. Stender (Ed.), *Parallel Genetic Algorithms: Theory and Applications* (pp.85-92). Brainware GmbH, Berlin: IOS Press

[KiSH91] Kitano H., Smith S.F., & Higuchi T. (1991). GA-1: A parallel associative memory processor for rule learning with genetic algorithms. In R-K. Belew, L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp.311-317). San Mateo, CA:Morgan Kaufmann.

[KoAn95] Koza, J., & Andre D. (1995). Parallel genetic programming on a network of transputers. Technical Report CS-TR-95-1542, Stanford University.

[KrSV90] Kroger B., Schwenderling P., & Vornberger O. (1990). Parallel genetic packing of rectangles. In H.-P.Schwefel and R.Manner (Eds.), *Proceedings of the First Workshop on Parallel Problem Solving from Nature*, vol. 496 of Lecture Notes in Computer Science (pp. 160-164). Berlin: Springer-Verlag.

[KrSV93] Kroger B., Schwenderling P., & Vornberger O. (1993). Parallel genetic packing on transputers. In J. Stender (Ed.), *Parallel Genetic Algorithms: Theory and Applications* (pp.151-186), Brainware GmbH, Berlin: IOS Press.

[Levi94] Levine, D. (1994). *A parallel genetic algorithm for the set partitioning problem.* Ph.D. Thesis, Mathematics and Computer Science Division, Argonne National Laboratory, May. (info.mcs.anl.gov: pub/tech_reports/reports/ANL9423.ps.Z).

[LiMa90] Li, T., & Mashford, J. (1990). A parallel genetic algorithm for quadratic assignment. In R.A. Ammar (Ed.), *Proceedings of the ISMM International Conference. Parallel and Distributed Computing and Systems* (pp.391-394). New York, NY: Acta Press.

[LiBa91] Liepens, G., & Baluja, S. (1991). apGA: An adaptive parallel genetic algorithm. Technical Report, Oak Ridge National Laboratoty.

[LiPG94] Lin, S.-C., Punch III, W.F., & Goodman, E.D. (1994). Coarse-grain parallel genetic algorithms: Categorization and new approach, *Parallel & Distributed Processing 94*, Dallas TX.

[MaEa89] Macfarlane, D., & East, I. (1989). An investigation of several parallel genetic algorithms. In S.J. Turner (Ed), *Tools and Techniques for Transputer Applications, Proceedings of the 12th Occam User Group Technical Meeting* (pp.60-67). Amsterdam : IOS Press

[MaSp89] Manderick B., & Spiessens P. (1989). Fine-grained parallel genetic algorithms. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp.428-433). San Mateo, CA: Morgan Kaufmann.

[Mare94] Maresky, J.G. (1994). On *effective communication in distributed genetic algorithms.* MSc. Thesis, Institute of Computer Science, Hebrew Univ. of Jerusalem, Israel. (ftp://ftp.huji.ac.il/pub/misc/jonathan/msc-maresky.ps.gz)

[MaTS94] Marin, F.J., Trelles-Salazar, O., & Sandoval, F. (1994). Genetic algorithms on LAN-message passing architectures using PVM: Application to the routing problem. In Y. Davidor, H-P. Schwefel, and R. Manner (Eds.), *Proceedings of the Third Conference on Parallel Problem Solving from Nature*, vol 866 of Lecture Notes in Computer Science (pp.534-543). Berlin: Springer-Verlag.

[MaWh92] Mathias, K., & Whitley, D. (1992). Genetic operators, the fitness landscape and the traveling salesman problem. In R.Manner and B.Manderick (Eds.), *Proceedings of the Second Workshop on Parallel Problem Solving from Nature* (pp.219-228). Amsterdam: Elsevier Science Publishers.

[McGD90] McClurkin, G.D., Geary, R.A., & Durrani T.S. (1990). An investigation into the parallelising of genetic algorithms (Extended Abstract). In D.J. Prichard and C.J. Scott (Eds), *Applications of Transputers 2, Proceedings of the Second International Conference on Applications of Transputers* (pp.581-7). Amsterdam: IOS Press.

[Mitc96] Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.

[Mühl89] Mühlenbein, H. (1989). Parallel genetic algorithms, population genetics, and combinatorial optimization. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 416-421). San Mateo, CA: Morgan Kaufmann.

[Mühl91] Mühlenbein, H. (1991). Evolution in time and space - The parallel genetic algorithm. In G.E. Rawlins (Ed.), *Foundations of Genetic Algorithms* (pp.316-337). San Mateo, CA: Morgan Kaufmann.

[Mühl92a] Mühlenbein H. (1992). Parallel genetic algorithms and combinatorial optimization. In O. Balchi, R. Sharda, S. Zenios (Eds.), *Computer Science and Operations Research* (pp.441-456), Pergamon Press

[Mühl92b] Mühlenbein, H. (1992). How genetic algorithms really work: I. mutation and hillclimbing. In R.Manner and B.Manderick (Eds.), *Proceedings of the Second Workshop on Parallel Problem Solving from Nature* (pp.15-25). Amsterdam: Elsevier Science Publishers.

[MuGK87] Mühlenbein H., Gorges-Schleuter M., & Kramer O. (1987). New solutions to the mapping problem of parallel systems: The evolution approach. *Parallel Computing* 4, 269-279.

[MuGK88] Mühlenbein, H., Gorges-Schleuter, M., & Kramer, O. (1988). Evolution algorithms in combinatorial optimization. *Parallel Computing*, 7, 65-85.

[MuSB91a] Mühlenbein, H., Schomisch, M., & Born, J., (1991a). The parallel genetic algorithm as function optimizer. *Parallel Computing*, 17, 619-632.

[MuSB91b] Mühlenbein, H., Schomisch, M., & Born, J. (1991b). The parallel genetic algorithm as function optimizer. In R-K. Belew and L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp. 271-278). San Mateo, CA: Morgan Kaufmann.

[MuTS94] Munetomo, M., Takai, Y., & Sato, Y. (1994). A genetic approach to dynamic load balancing in a distributed computing system. *Proceedings of the IEEE World Congress on Computational Intelligence, First IEEE Conference on Evolutionary Computation*, vol I, (pp. 418-421).

[MuTa91] Muntean, T., & Talbi, E.G. (1991). A parallel genetic algorithm for process-processors maping, in M.Durand, F.El. Dabaghi (Eds.), *High Performance Computing II, Proceedings of the Second Symposium* (pp.71-82). Amsterdam: North-holland.

[Norm88] Norman, M.G. (1988). A genetic approach to topology optimization for multiprocessor architectures. Technical Report ECSP-TR-7, Dept. of Physics, Univ. of Edinburgh.

[Pete90] Peterson, C. (1990). Parallel distributed approaches to combinatorial optimization: Benchmark studies on travelling salesman problem. *Neural Computation* 2, 261-269.

[PeKa94] Petridis, V., & Kazarlis, S. (1994). Varyiing quality function in genetic algorithms and the cutting problem. *Proceedings of the IEEE World Congress on Computational Intelligence, First IEEE Conference on Evolutionary Computation*, vol I, (pp. 166-169).

[PeKP93] Petridis, V., Kazarlis, S., & Papaikonomou, A. (1993). A Genetic Algorithm for Training Recurrent Neural Networks. *Proceedings of International Joint Conference on Neural Networks* (pp. 2706-2709). IEEE

[PePa94] Petridis, V., & Papaikonomou, A. (1994). Recurrent Neural Networks as Pattern Generators. *Proceedings of the IEEE International Conference on Neural Networks, IEEE World Congress on Computational Intelligence,* vol II, (pp. 872-875).

[PeLe89] Pettey, C.C., & Leuze, M.R. (1989). A Theoretical Investigation of a Parallel Genetic Algorithm. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 398-405). San Mateo, CA: Morgan Kaufmann.

[PeLG87] Pettey, C.C., Leuze, M.R., & Grefenstette, J. (1987). A Parallel Genetic Algorithm. In J. Grefenstette (Ed.), *Proceedings of the Second International Conference on Genetic Algorithms* (pp.155-161). Cambridge, MA: Lawrence Erlbaum Associates.

[Pham94] Pham, Q.T. (1994). Competitive evolution: A natural approach to operator selection, School of Chemical Eng. and Industrial Chemistry, Univ. of New South Wales, Sydney, Australia

[PoGY94] Potts, J.C., Giddens, T.D., & Yadav, S.B. (1994). The development and evaluation of an improved genetic algorithm based on migration and artificial selection, *IEEE Transactions on Systems, Man, and Cubernetics*, 24 (1), 73-86.

[Rech73] Rechenberg, I. (1973) *Evolutionsstrategie: Oprimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog, Stuttgart.

[RiAT93] Ribeiro Filho J.L., Alippi C., & Treleaven P. (1993). Genetic algorithm programming environments. In J. Stender (Ed.), *Parallel Genetic Algorithms: Theory and Applications* (pp.65-84). Brainware GmbH, Berlin: IOS Press

[SCED89] Schaffer, J.D., Caruana, R.A., Eshelman, L.J., & Das, R. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. In

J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp.51-60). San Mateo, CA: Morgan Kaufmann.

[Schw81] Schwefel, H.-P. (1981) *Numerical Optimization of Computer Models.* New York: John Wiley & Sons.

[Schw92] Schwem, M. (1992). Implementation of genetic algorithms on various interconnection networks. In M. Varelo, E. Onate, M. Jane, J.L. Larriba, & B. Suarez (Eds.), *Parallel Computing and Transputer Applications, Transputer and Occam Engineering Series*, vol. 28 (pp.195-203). Amsterdam: IOS Press.

[ScMü94] Schlierkamp-Voosen, D., & Mühlenbein, H. (1994). Strategy adaptation by competing subpopulations. In Y. Davidor, H-P. Schwefel, and R. Manner (Eds.), *Proceedings of the Third Conference on Parallel Problem Solving from Nature.,* vol 866 of Lecture Notes in Computer Science (pp.199-208). Berlin: Springer-Verlag.

[SWLD94] Sepehri N., Wan F.L.K., Lawrence P.D., & Dumont G.A. (1994). Hydraulic compliance identification using a parallel genetic algorithm. *Mechatronics*, 4(6), 617-633.

[Sere94a] Seredynski, F., (1994a). Dynamic Mapping and Load Balancing with Parallel Genetic Algorithms. *Proceedings of the IEEE World Congress on Computational Intelligence, First IEEE Conference on Evolutionary Computation*, vol II, (pp. 834-839).

[Sere94b] Seredynski, F., (1994b). Loosely Coupled Distributed Genetic Algorithms. In Y.Davidor, H-P. Schwefel and R. Manner (Eds.), *Proceedings of the Third Conference on Parallel Problem Solving from Nature* (pp.514-523), vol 866 of Lecture Notes in Computer Science. Berlin: Springer-Verlag.

[Shon93] Shonkwiler, R. (1993). Parallel Genetic Algorithms. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp.199-205). San Mateo, CA: Morgan Kaufmann.

[Spea92] Spears, W.M. (1992). Crossover or mutation. In L.D. Whitley (Ed.), *Foundations of Genetic Algorithms 2* (pp.221-238). San Mateo, CA: Morgan Kaufmann

[SpDJ91] Spears, W. M., & De Jong, K.A. (1991). . An analysis of multi-point crossover. In G. Rawlins (Ed.), *Foundations of Genetic Algorithms* (pp.301-315). San Mateo, CA: Morgan Kaufmann

[SpMa91] Spiesens, P., & Manderick, B. (1991). A massively parallel genetic algorithm. Implementation and first analysis. In R-K. Belew, L.B. Booker (Eds.), *Proceedings of the Fourth International Conference on Genetic Algorithms* (pp.264-270). San Mateo, CA: Morgan Kaufmann.

[SrPa94] Srinivas, M., & Patnaik, L.M. (1994). Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 24 (4).

[StWM90] Starkweather, T., Whitley, D., & Mathias, K. (1990). Optimization using distributed genetic algorithms. In H.-P.Schwefel and R.Manner (Eds.), *Proceedings of the First Workshop on Parallel Problem Solving from Nature*, vol. 496 of Lecture Notes in Computer Science (pp. 176-185). Berlin : Springer-Verlag.

[SuRa94] Surry, P.D., & Radcliffe, N.J. (1994). RPL2: A language and parallel framework for evolutionary computing, Tech Report EPCC-TR94-10, University of Edinburgh, Edinburgh Parallel Computing Centre.

[Sysw89] Syswerda, G. (1989). Uniform crossover in gentic algorithms. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 2-9). San Mateo, CA: Morgan Kaufmann.

[TaBe91] Talbi, E-G., & Bessiere, P. (1991). A Parallel Genetic Algorithm for the Graph Partitioning Problem. In *Proceedings of the 1991 International Conference on Supercomputing* (pp.312-320), Cologne, Germany. ACM Press.

[TaMu93] Talbi, E-G., & Muntean, T. (1993). General Heuristics for the Mapping Problem. *Proceedings of the World Transputer Conference.*

[TKSM94] Tamaki, H., Kita, H., Shimizu, N., Maekawa, K., & Nishikawa, Y. (1994). A comparison study of genetic codings for the travelling salesman problem. *Proceedings of the IEEE World Congress on Computational Intelligence, First IEEE Conference on Evolutionary Computation*, vol I (pp. 1-6).

[TaNi92] Tamaki, H., & Nishikawa, Y. (1992). A Paralleled Genetic Algorithm based on a Neighborhood Model and its Application to the Jobshop Scheduling. In R.Manner and B.Manderick (Eds.), *Proceedings of the Second Workshop on Parallel Problem Solving from Nature* (pp.573-582). Amsterdam: Elsevier Science Publishers.

[Tane87] Tanese, R. (1987). Parallel Genetic Algorithms for a hypercube. In J.J Grefenstette (Ed.), *Proceedings of the First International Conference on Genetic Algorithms and their Applications* (pp.177-183). Cambridge, MA: Lawrence Erlbaum Associates

[Tane89] Tanese, R. (1989). Distributed Genetic Algorithms. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp.434-439). San Mateo, CA: Morgan Kaufmann.

[TaLe94] Tang, A. Y.-C., & Leung, K.-S. (1994). A modified edge recombination operator for the travelling salesman problem. In Y. Davidor, H-P. Schwefel, and R. Manner (Eds.), *Proceedings of the Third Conference on Parallel Problem Solving from Nature* (pp.180-188). Berlin: Springer-Verlag.

[ThGo93] Thierens, D., & Goldberg, D.E. (1993). Mixing in genetic algorithms. *Proceedings of the Fifth International Conference on Genetic Algorithms* (pp.38-45). San Mateo, CA: Morgan Kaufmann.

[TöZi89] Törn, A., & Zilinskas, A. (1989). Global optimization. *Lecture Notes in Computer Science*, vol 350, Berlin: Springer-Verlag.

[ToLo93] Toth, G.J., & Lorincz, A. (1993). Genetic algorithm with migration on topology conserving maps. In *Proceedings of the International Conference on Artificial Neural Networks* (pp.605-8). Amsterdam.

[VaJo93] Valenzuela, C.L., & Jones, A.J. (1993). Evolutionary divide and conquer (I): A novel genetic approach to the TSP. *Evolutionary Computation* 1(4), 313-333

[VoSB92] Voigt, H-M., Santibanez-Koref, I., & Born, J. (1992). Hierarchically Structured Distributed Genetic Algorithms. In R.Manner and B.Manderick (Eds.), *Proceedings of the Second Conference on Parallel Problem Solving from Nature* (pp.155-164). Amsterdam: Elsevier Science Publishers.

[vLaM90] von Laszewsski, G., & Mühlenbein, H. (1990) Partitioning a graph with a Parallel Genetic Algorithm. In H.-P.Schwefel and R.Manner (Eds.), *Proceedings of the First Workshop on Parallel Problem Solving from Nature*, vol. 496 of Lecture Notes in Computer Science (pp.165-169). Berlin: Springer-Verlag.

[Whit93] Whitley, D. (1993). A Genetic Algorithm Tutorial, Technical Report CS-93-1-3, Dept. of Computer Science, Colorado State University.

[WhSB90] Whitley, D., Starkweather, T., & Bogart, C. (1990). Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity. *Parallel Computing*, 14 (3), 347

[WhSF89] Whitley, D., Starkweather, T., & Fuquay D'Ann. (1989). Scheduling problems and traveling salesmen: The genetic edge recombination operator. In J.D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp.133-140). San Mateo, CA: Morgan Kaufmann

[WoMa95] Wolpert, D.H., & Macready W.C. (1995). No free lunch theorems for search. Technical Report SFI-TR-95-02-010, The Santa Fe Institute.

Παράρτημα Α

Απόδοση Όρων

Artificial Life	Τεχνητή Ζωή
Binary strings	Αλυσίδα δυαδικών ψηφίων
Bottleneck	Συμφόρηση
Bridged network	Δίκτυο γεφύρωσης
Building block	Τμήμα κατασκευής
Chromosome	Χρωμόσωμα
Coarse grained	Χαμηλή ανάλυση
Co-operating populations	Συνεργαζόμενοι πληθυσμοί
Crossover	Ανασυνδιασμός
Cycle crossover	Κυκλικός ανασυνδυασμός
Co-operating populations with different	Συνεργαζόμενοι πληθυσμοί με διαφορετικές
Evolution behavior	Εξελικτική συμπεριφορά
Decomposition approach	Προσέγγιση διαχωρισμού
Defining length	Μήκος ορισμού
Deletion operator	Τελεστής διαγραφής
Deme	Υπο-πληθυσμός
Different evolution behavior	Διαφορετική εξελικτική συμπεριφορά
Diffusion model	Μοντέλο διάχυσης
Distributed memory	Κατανεμημένη μνήμη
Edge rcombination	Ανασυνδιασμός ακμών
Epoch	Εποχή
Evolutionary Algorithms	Εξελικτικοί Αλγόριθμοι
Evolution Strategies	Στρατηγικές Εξέλιξης
Evolutionary Programming	Εξελικτικός Προγραμματισμός
Exploitation	Εκμετάλευση
Fine grained	Υψηλή ανάλυση
Fit individual	Ικανό άτομο
Fitness function	Συνάρτηση ποιότητας
Fitness scaling	Κλιμάκωση της ποιότητας
Gene	Γονίδιο
Gene doubling operator	Τελεστής διπλασιασμού γονιδίων
Generational replacement	Γενεαλογική αντικατάστασης
Global parallelization	Τυπική παράλληλη προσέγγιση
Hill-climbing operator	Τελεστής αναρρίχησης
Inversion operator	Τελεστής αναστροφής
Island model	Μοντέλο νησίδων
Isolation-by-distance model	Μοντέλο απομόνωσης-μέσω-απόστασης

Local search	Τοπική αναζήτηση
Massively parallel computers	Μαζικά παράλληλοι Η/Υ
Master	Συντονιστής
Master processor	Συντονιστής επεξεργαστής
Matrix crossover	Ανασυνδιασμός μήτρας
Mesh topology	Τοπολογία πλέγματος
Migration	Μετανάστευση
Migration frequency	Συχνότητα μετανάστευσης
Migration interval	Διάτημα μετανάστευσης
Migration model	Μοντέλο μετανάστευσης
Migration rate	Ποσοστό μετανάστευσης
Migration strategy	Μέθοδος μετανάστευσης
Mutation	Μετάλλαξη
Mutation rate	Ρυθμός μετάλλαξης
Neighborhood model	Μοντέλο γειτνίασης
Node	Κόμβος
Order crossover	Ανασυνδιασμός σειράς
Parallel genetic algorithms (PGAs)	Παράλληλοι γενετικοί αλγόριθμοι (ΠΓΑ)
Partially matched crossover (PMX)	Ανασυνδιασμός μερικής ταύτισης
Phenotype mutation	Φαινοτυπική μετάλλαξη
Pipeline network	Δίκτυο διασωλήνωσης
Population structures	Δομές πληθυσμού
Position based crossover	Ανασυνδυσμός βασισμένος στην θέση
Proportionate selection	Αναλογική επιλογή
Quality function	Συνάρτηση ποιότητας
Ranking selection	Επιλογή κατάταξης
Recombination	Ανασυνδιασμός
Redundant genetic coding	Πλεονάζουσα γενετική κωδικοποίηση
Reproductive trial	Αναπαραγωγική δοκιμή
Search space	Χώρος αναζήτησης
Selection method	Μέθοδος επιλογής
Shared Memory	Διαμοιραζόμενη μνήμη
Sigma truncation	Αποκοπή σίγμα
Solution space	Χώρος λύσεων
Standard parallelization	Τυπική παράλληλη προσέγγιση
Steady state replacement	Αντικατάσταση σταθερής κατάστασης
Stepping-stone model	Βηματικό μοντέλο
Sub-population	Υπο-πληθυσμός
Tournament selection	Επιλογή κατόπιν αγώνων
Two-dimensional (2-D) array	Διδιάστατος πίνακας
Worker	Εργαζόμενος
Worker processor	Επεξεργαστής εργασίας

Παράρτημα Β

Πρόβλημα Περιοδεύοντος Πωλητή

NAME : bays29
COMMENT : 29_cities_in_Bavaria_(Street_distances)
TYPE : TSP
DIMENSION : 29
EDGE_WEIGHT_TYPE : EXPLICIT
EDGE_WEIGHT_FORMAT : FULL_MATRIX
DISPLAY_DATA_TYPE : TWOD_DISPLAY
EDGE_WEIGHT_SECTION

0 107 241 190 124 80 316 76 152 157 283 133 113 297 228 129 348 276 188 150 65 341 184 67 221 169 108 45 167
107 0 148 137 88 127 336 183 134 95 254 180 101 234 175 176 265 199 182 67 42 278 271 146 251 105 191 139 79
241 148 0 374 171 259 509 317 217 232 491 312 280 391 412 349 422 356 355 204 182 435 417 292 424 116 337 273 77
190 137 374 0 202 234 222 192 248 42 117 287 79 107 38 121 152 86 68 70 137 151 239 135 137 242 165 228 205
124 88 171 202 0 61 392 202 46 160 319 112 163 322 240 232 314 287 238 155 65 366 300 175 307 57 220 121 97
80 127 259 234 61 0 386 141 72 167 351 55 157 331 272 226 362 296 232 164 85 375 249 147 301 118 188 60 185
316 336 509 222 392 386 0 233 438 254 202 439 235 254 210 187 313 266 154 282 321 298 168 249 95 437 190 314 435
76 183 317 192 202 141 233 0 213 188 272 193 131 302 233 98 344 289 177 216 141 346 108 57 190 245 43 81 243
152 134 217 248 46 72 438 213 0 206 365 89 209 368 286 278 360 333 284 201 111 412 321 221 353 72 266 132 111
157 95 232 42 160 167 254 188 206 0 159 220 57 149 80 132 193 127 100 28 95 193 241 131 169 200 161 189 163
283 254 491 117 319 351 202 272 365 159 0 404 176 106 79 161 165 141 95 187 254 103 279 215 117 359 216 308 322
133 180 312 287 112 55 439 193 89 220 404 0 210 384 325 279 415 349 285 217 138 428 310 200 354 169 241 112 238
113 101 280 79 163 157 235 131 209 57 176 210 0 186 117 75 231 165 81 85 92 230 184 74 150 208 104 158 206
297 234 391 107 322 331 254 302 368 149 106 384 186 0 69 191 59 35 125 167 255 44 309 245 169 327 246 335 288
228 175 412 38 240 272 210 233 286 80 79 325 117 69 0 122 122 56 56 108 175 113 240 176 125 280 177 266 243
129 176 349 121 232 226 187 98 278 132 161 279 75 191 122 0 244 178 66 160 161 235 118 62 92 277 55 155 275
348 265 422 152 314 362 313 344 360 193 165 415 231 59 122 244 0 66 178 198 286 77 362 287 228 358 299 380 319
276 199 356 86 287 296 266 289 333 127 141 349 165 35 56 178 66 0 112 132 220 79 296 232 181 292 233 314 253
188 182 355 68 238 232 154 177 284 100 95 285 81 125 56 66 178 112 0 128 167 169 179 120 69 283 121 213 281
150 67 204 70 155 164 282 216 201 28 187 217 85 167 108 160 198 132 128 0 88 211 269 159 197 172 189 182 135
65 42 182 137 65 85 321 141 111 95 254 138 92 255 175 161 286 220 167 88 0 299 229 104 236 110 149 97 108
341 278 435 151 366 375 298 346 412 193 103 428 230 44 113 235 77 79 169 211 299 0 353 289 213 371 290 379 332
184 271 417 239 300 249 168 108 321 241 279 310 184 309 240 118 362 296 179 269 229 353 0 121 162 345 80 189 342
67 146 292 135 175 147 249 57 221 131 215 200 74 245 176 62 287 232 120 159 104 289 121 0 154 220 41 93 218
221 251 424 137 307 301 95 190 353 169 117 354 150 169 125 92 228 181 69 197 236 213 162 154 0 352 147 247 350
169 105 116 242 57 118 437 245 72 200 359 169 208 327 280 277 358 292 283 172 110 371 345 220 352 0 265 178 39
108 191 337 165 220 188 190 43 266 161 216 241 104 246 177 55 299 233 121 189 149 290 80 41 147 265 0 124 263
45 139 273 228 121 60 314 81 132 189 308 112 158 335 266 155 380 314 213 182 97 379 189 93 247 178 124 0 199
167 79 77 205 97 185 435 243 111 163 322 238 206 288 243 275 319 253 281 135 108 332 342 218 350 39 263 199 0

```
DISPLAY_DATA_SECTION
  1   1150.0  1760.0
  2    630.0  1660.0
  3     40.0  2090.0
  4    750.0  1100.0
  5    750.0  2030.0
  6   1030.0  2070.0
  7   1650.0   650.0
  8   1490.0  1630.0
  9    790.0  2260.0
 10    710.0  1310.0
 11    840.0   550.0
 12   1170.0  2300.0
 13    970.0  1340.0
 14    510.0   700.0
 15    750.0   900.0
 16   1280.0  1200.0
 17    230.0   590.0
 18    460.0   860.0
 19   1040.0   950.0
 20    590.0  1390.0
 21    830.0  1770.0
 22    490.0   500.0
 23   1840.0  1240.0
 24   1260.0  1500.0
 25   1280.0   790.0
 26    490.0  2130.0
 27   1460.0  1420.0
 28   1260.0  1910.0
 29    360.0  1980.0
EOF
```

Παράρτημα Γ

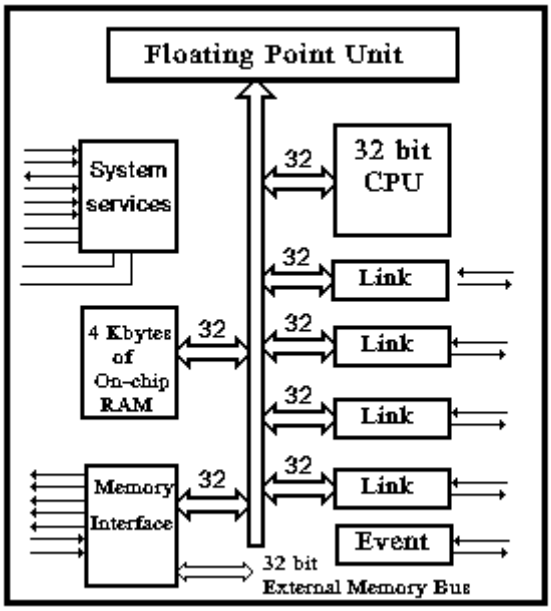
Περιγραφή του Transputer IMS T800

Ο στόχος της σχεδίασης του transputer της INMOS ήταν μία συσκευή που θα μπορούσε να χρησιμοποιηθεί σε συστήματα μεταβίβασης μηνυμάτων (message passing) πολλών επεξεργαστών, όπου κάθε επεξεργαστής θα έχει την δική του φυσική μνήμη, αλλά με υποστήριξη για πολλαπλές διεργασίες διαμοιραζόμενης μνήμης (shared memory) σε κάθε επεξεργαστή.

Οι transputers διαφέρουν από άλλους επεξεργαστές στα εξής :

- έχουν στην ψηφίδα τους (on-chip) τη δική τους τοπική μνήμη (RAM).
- έχουν την δυνατότητα να επικοινωνούν με τέσσερις άλλους transputers, μέσω σειριακών συνδέσμων (links), προσφέροντας έτσι υποστήριξη για συστήματα πολλαπλών επεξεργαστών.
- η αρχιτεκτονική τους υποστηρίζει το μοίρασμα του χρόνου του επεξεργαστή (time shearing), επιτρέποντας έτσι την ταυτόχρονη (concurrent) εκτέλεση διαδικασιών.

Οι transputers της κάρτας Fast9 είναι οι IMS T800 της INMOS, με μήκος λέξης 32-bit και τεχνολογία CMOS. Κάθε transputer έχει τοπική μνήμη 4 Kbytes και μονάδα κινητής υποδιαστολής για αριθμούς 32 bit. Κάθε ένας από αυτούς δουλεύει ταυτόχρονα και ανεξάρτητα από τους άλλους. Στο σχήμα Γ1 απεικονίζεται η εσωτερική δομή του transputer και οι βασικές λειτουργικές μονάδες του.



Σχήμα Γ1. Διάγραμμα του INMOS T800 Transputer

Αυτές είναι οι εξής :

- Η Κεντρική Μονάδα Επεξεργασίας (Central Processing Unit) των 32-bits. Επιτρέπει την ψευδο-παράλληλη (concurrent) εκτέλεση διεργασιών οι οποίες μοιράζονται το χρόνο επεξεργασίας με τη σειρά. Το υλικό (hardware), έχει ενσωματωμένο ένα διαχειριστή (scheduler), ο οποίος πραγματοποιεί την εναλλαγή τους (task switching), κατά τρόπο ώστε ανενεργές διαδικασίες να μην καταναλώνουν χρόνο επεξεργαστή. Οι διεργασίες αυτές οι οποίες μπορεί να είναι ολλές σε αριθμό, έχουν τη δυνατότητα να εκτελούνται με δύο προτεραιότητες. Υπάρχουν δε γι' αυτό το λόγο δύο χρονιστές (timers). Ο ένας, με συχνότητα 1 χτύπο/μsec, είναι προσπελάσιμος μόνο από τις διεργασίες με υψηλή προτεραιότητα. Ο άλλος (1 χτύπος/0.64 μsec) είναι προσπελάσιμος από διεργασίες με χαμηλή προτεραιότητα.

- Η μονάδα κινητής υποδιαστολής (Floating Point Unit), υποστηρίζει αριθμητικές πράξεις κινητής υποδιαστολής, αριθμών απλών (32-bits) ή διπλού μήκους (64-bits) σε πολύ μεγάλες ταχύτητες, σύμφωνα με το πρότυπο IEEE-754. Τυπική απόδοση της μονάδας κινητής υποδιαστολής 2.25 Mflops με ταχύτητα ρολογιού 30 Mhz. Η FPU μπορεί να λειτουργεί ανεξάρτητα από την KME και να εκτελεί εντολές επεξεργασίας αριθμητικών δεδομένων παράλληλα με αυτήν. Όμως οι δύο αυτές μονάδες μοιράζονται τον ίδιο εσωτερικό διάδρομο δεδομένων και έτσι δεν είνια δυνατή η ταυτόχρονη προσπέλαση της μνήμης.

- Η μονάδα εξυπηρέτησης του συστήματος (System Services). Δέχεται όλα τα σήματα ελέγχου για να αρχικοποιήσει και να διατηρήσει τη λειτουργία του transputer. Δέχεται τάση λειτουργίας 5V, σήμα εξωτερικού ρολογιού 5 Mhz, σήματα λαθών και σήματα αρχικοποίησης (reset). Επίσης δέχεται τρεις γραμμές επιλογής της ταχύτητας του επεξεργαστή. Οι δυνατές ταχύτητες λειτουργίας είναι 17.5, 20, 25, 30 και 35 Mhz. Στις υλοποιήσεις της παρούσας εφαρμογής, οι transputers χρησιμοποιούνται στην ταχύτητα των 30 Mhz (33nsec χρόνος κύκλου μηχανής).

- Η στατική μνήμη (Static RAM). Έχει 4 Kbytes γρήγορη εσωτερική στατική μνήμη στο chip του. Ο ρυθμός προσπέλασης δεδομένων απ' αυτή τη μνήμη είναι 120 Mbytes/sec. Η προσπέλασή της ολοκληρώνεται σε έναν κύκλο μηχανής. Ο transputer μπορεί επίσης να έχει προσπέλαση σε εξωτερική μνήμη 4 Gbytes με ρυθμό προσπέλασης 40 Mbytes/sec. Εσωτερική και εξωτερική μνήμη απεικονίζονται στον ίδιο γραμμικό χώρο διευθύνσεων. Η μνήμη είναι προσπελάσιμη ανά byte με το λιγότερο σημαντικό byte μιας λέξης να τοποθετείται στη μικρότερη διεύθυνση. Η εσωτερική μνήμη μπορεί να απενεργοποιηθεί με κατάλληλο σήμα στη γραμμή "DisableIntRam".

- Το προσαρμοστικό εξωτερικής μνήμης (External Memory Interface), που επιτρέπει προσπέλαση σε μνήμη με διευθύνσεις των 32 bits. Υποστηρίζει δε δυναμική και στατική RAM καθώς και ROM και EPROM. Ο χρονισμός της EMI μπορεί να ελεγχθεί με εσωτερικά σήματα ελέγχου για συνεργασία με τους περισσότερους τύπους μνήμης .

- Η μονάδα συγχρονισμού διεργασιών (Event), προσαρμόζει με τη μέθοδο της χειραψίας (hand-shaking) τις εξωτερικές ασύγχρονες αιτήσεις διακοπών με τις εσωτερικές διεργασίες

- Οι τέσσερις σύνδεσμοι (links) και τα προσαρμοστικά τους (link inerfases), τα οποία επιτρέπουν τη σειριακή επικοινωνία με άλλους transputers. Ο transputer T800 διαθέτει τέσσερις πλήρως αμφίδρομους (full duplex) εξωτερικούς συνδέσμους επικοινωνίας. Η φυσική σύνδεση μεταξύ δύο transputers πραγματοποιείται με την ηλεκτρική διασύνδεση των προσαρμοστικών των

συνδέσμων τους. Κάθε σύνδεσμος (ζεύγος αγωγών) υλοποιείται με ένα κανάλι εισόδου και ένα εξόδου. Κάθε byte δεδομένων στέλνεται σειριακά στο ένα κανάλι, μαζί με ένα bit αρχής και ένα bit τέλους. Η διεργασία που στέλνει δεδομένα στο ένα κανάλι περιμένει από το άλλο επιβεβαίωση λήψης. Έτσι και η ρουτίνα εκπομπής στον έναν transputer και η ρουτίνα λήψης στον άλλον, βεβαιώνονται για τη σωστή μετάδοση του μηνύματος και ή συνεχίζουν με άλλο μήνυμα ή τελειώνουν την εργασία τους.

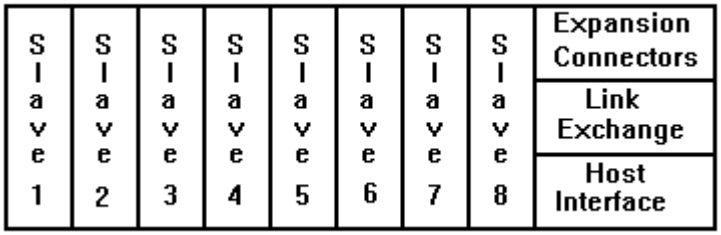
Οι σύνδεσμοι του transputer IMS T800 υποστηρίζουν την καθιερωμένη INMOS ταχύτητα επικοινωνίας των 10 Mbps. Η ταχύτητα αυτή μπορεί να αλλάξει με ειδικά σήματα στις γραμμές "LinkSpesial", "Link0Spesial" και "Link123Spesial", οι οποίες αποτελούν σήματα εισόδου στη μονάδα ελέγχου συνδέσμων (link services). Ετσι η ταχύτητα επικοινωνίας μπορεί να είναι 5 ή 10 ή 20 Mbps.

Παράρτημα Δ

Περιγραφή της κάρτας Fast9

Η κάρτα Fast9 της Quintek μπορεί να προσαρμοστεί σε έναν απλό IBM PC, AT ή συμβατό, καταλαμβάνοντας μια υποδοχή (slot) των 16 bits του διαδρόμου συστήματος. Αποτελείται από εννιά transputers των 32 bits με ένα Mbyte τοπική RAM για τον καθένα. Ένα προγραμματιζόμενο ραβδεπαφικό διακόπτη (crossbar switch) των συνδέσμων, καθώς και κύκλωμα διασύνδεσης (host interface) ανάμεσα στην αρχιτεκτονική της INMOS και του διαδρόμου συστήματος του PC.

Το σχήμα Δ1 δείχνει το συνοπτικό διάγραμμα της κάρτας όπου έχουν σχεδιαστεί οι ανεξάρτητες λειτουργικές της μονάδες. Ο επεξεργαστής του PC αναφέρεται σαν **Host** επεξεργαστής. Ο transputer που συνδέεται απ' ευθείας με τον Host ονομάζεται **Master**, ενώ οι άλλοι οκτώ ονομάζονται **Slaves.** Η παρακάτω ανάπτυξη γίνεται σύμφωνα μ' αυτά τα ονόματα.



Σχήμα Δ1. Κάρτα Fast9

Η διασύνδεση του Host περιλαμβάνει δύο ομάδες σημάτων, την ομάδα Host Link και την ομάδα Host System (σχ. Δ2) Το Host Link διακινεί τα δεδομένα από την κάρτα προς το PC και αντίστροφα, ενώ το Host System παρέχει τα αναγκαία σήματα για τη συνεργασία των δύο διαφορετικών αρχιτεκτονικών.

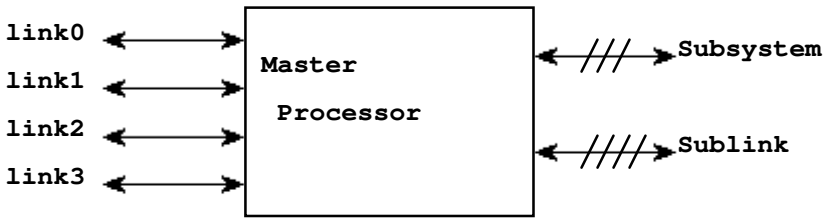
Σχήμα.Δ2. Host Interface

Το Host Interface από τη μεριά των transputer συμπεριφέρεται σαν ένας Inmos σύνδεσμος και από τη μεριά του PC σαν μια ομάδα θυρών εισόδου-εξόδου (I/O Ports) του 80x86. Όσον αφορά την υλοποίηση του, αυτή γίνεται με ένα chip IMS C0012, το οποίο είναι ένας προσαρμογέας (adaptor) από σειριακό σύνδεσμο σε παράλληλο διάδρομο δεδομένων. Τα σήματα Host Link και Host System εκτός από τη σύνδεσή τους με το διάδρομο συστήματος του PC, παρέχονται και στο πίσω μέρος της κάρτας.

Επίσης τα σήματα αυτά συνδέονται στο Master μέσω του διακόπτη SW2 (Πίνακας Δ1), ο οποίος επιτρέπει την απενεργοποίηση του Host Interface σε περίπτωση που χρησιμοποιούνται περισσότερες της μίας κάρτας.

Ο Master αποτελείται από έναν Transputer με 1 Mbyte εσωτερική μνήμη και δύο εξωτερικά κυκλώματα ελέγχου (subsystem logic και sublink logic) και επιτρέπει έλεγχο πάνω στην τοπολογία των Slaves.

Συνδέεται με τους άλλους Transputers μέσω των τεσσάρων Inmos συνδέσμων (links), καθώς και μέσω των γραμμών ελέγχου συστήματος (sublink και subsystem) (σχ. Δ3).



Σχήμα Δ3. Master

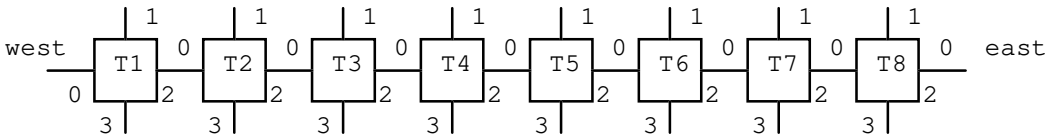
Το "Sublink" υλοποιείται με ένα IMS C012, το οποίο ο transputer το βλέπει σαν μνήμη και οδηγεί το "Configuration link " του ραβδεπαφικού δικτύου διασύνδεσης, επιτυγχάνοντας έτσι τον έλεγχο της τοπολογίας του δικτύου.

Το "Subsystem", αποτελείται από καταχωρητές προσπελάσιμους σαν μνήμη και συντονίζει τη συνολική λειτουργία του δικτύου.

Οι τέσσερις σύνδεσμοι (links0-3) και οι τρεις γραμμές ελέγχου που απαρτίζουν τη λογική (Subsystem). Είναι διαθέσιμες και στο πίσω μέρος της κάρτας (rear connector) ώστε να υπάρχει δυνατότητα σύνδεσης με άλλη κάρτα.

Η συγκεκριμένη υλοποίηση των δύο κυκλωμάτων ελέγχου με χαρτογράφηση μνήμης (memory mapped), αφήνει ελεύθερους για χρήση τους τέσσερις συνδέσμους του Master.

Οι οκτώ Slaves έχουν από 1 Mbyte RAM ο καθένας. Ενώνονται μεταξύ τους μέσω Inmos συνδέσμων και γραμμών ελέγχου συστήματος όπως φαίνεται στο σχήμα Δ4.

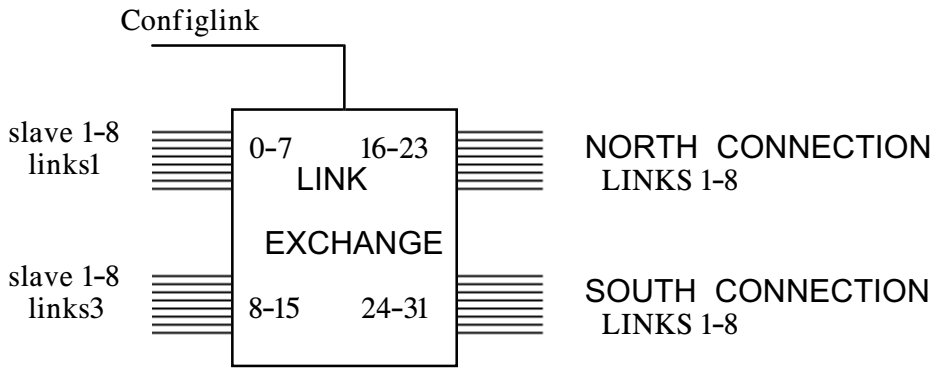


Σχήμα Δ4. Slaves

Οι σύνδεσμοι 0 και 2 κάθε Slave αποτελούν αλυσίδα μόνιμα συνδεδεμένη ηλεκτρικά μέσω αγωγών. Οι σύνδεσμοι 1 και 3 συνδέονται στο chip IMS C004 για να προγραμμα-τιστούν κατάλληλα. Ο σύνδεσμος 0 του Slave-1 και σύνδεσμος 2 του Slave-8 ονομάζονται "West link" και "East link" αντίστοιχα και μπορούν να προσπελασθούν μέσω του rear connector. Εναλλακτικά

μπορούν να συνδεθούν στους συνδέσμου 1 και 3 του Master μέσω διακοπτών των SW 2.4 και SW 2.5 (Πίνακας Δ2).

Το "Link Exchange" είναι ένας ραβδεπαφικός διακόπτης (crossbar switch) και χρησιμοποιείται για να συνδέσει τους ελεύθερους συνδέσμους των Slaves και τους δύο "8-link connectors" που παρέχονται για επικοινωνία με άλλες κάρτες. Είναι ένας ηλεκτρονικά προγραμματιζόμενος διακόπτης με δυνατότητα πλήρους διασύνδεσης 32 συνδέσμων εισόδου με 32 συνδέσμων εξόδου (σχ. Δ5).



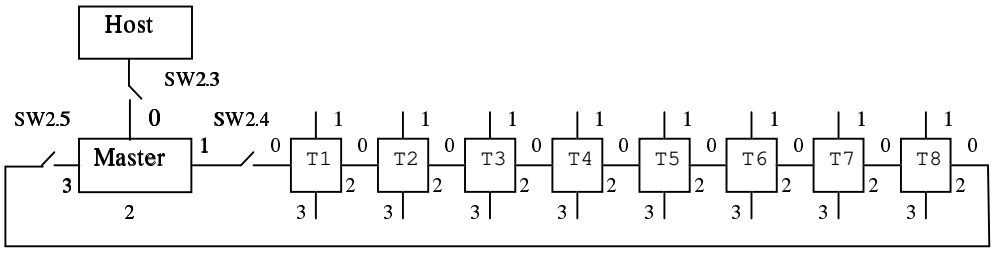
Σχήμα Δ5. Link Exchange

Ο προγραμματισμός του γίνεται μέσω μηνυμάτων που στέλνονται στις γραμμές "Config link", οι οποίες συνδέονται μόνιμα με τις γραμμές "Sublink" του Master.

Οι ταχύτητες των συνδέσμων διαμέσου του "Link exchange" δεν είναι ακριβώς ίδιες μ' αυτές των μόνιμα συνδεδεμένων συνδέσμων που υπάρχουν και καθορίζονται από το διακόπτη SW 1.10 (Πίνακας Δ1). Το "Link exchange" επιφέρει μια μικρή καθυστέρηση στη μεταφορά των δεδομένων χωρίς να προκαλεί εξασθένιση των σημάτων.

Οι Inmos σύνδεσμοι στην κάρτα είναι μεγάλης ταχύτητας και αποτελούνται από δύο σειριακές γραμμές επικοινωνίας μιας κατεύθυνσης. Η κάρτα υποστηρίζει ταχύτητες μεταφοράς δεδομένων: 10 Mbps και 20 Mbps, που διαλέγονται από τους διακόπτες SW 1.4, SW 1.5, SW 1.10. Πρέπει όμως για τις δύο άκρες να έχουν επιλεγεί οι ίδιες ταχύτητες.

Για κανονική χρήση της κάρτας Fast9 ο Master πρέπει να συνδέεται με τον "Host" και οι "Slaves" σε αλυσίδα με το Master. Οι διακόπτες SW 2.3, SW 2.4 και SW2.5 ενώνουν τους συνδέσμους του Master, όπως φαίνεται στο σχήμα Δ6.



Σχήμα Δ6. Μόνιμη τοπολογία των Transputers στην κάρτα Fast9

Για μερικές ασυνήθιστες εφαρμογές ιδίως για τοπολογίες που απαιτούν πολλές κάρτες οι τρεις σύνδεσμοι του Master μπορούν να χρησιμοποιηθούν και με άλλους τρόπους. Σ' αυτή την περίπτωση ο χρήστης έχει τη δυνατότητα να κάνει τις απαιτούμενες συνδέσεις εύκολα, στο "rear connector".

Πίνακας Δ1. Διακόπτης SW1

Αριθμός του διακόπτη Χρώμα του διακόπτη	SW1.1 καφέ	SW1.2 κόκκινο	SW1.3 πορτοκ.	SW1.4 κίτρινο	SW1.5 πράσινο
Switches not used	-	-	-	-	-
Master link 0& Host στα 10Mbits/sec	-	-	-	OFF	-
Master link 0& Host στα 20Mbits/sec	-	-	-	ON	-
Master link 1-3 στα 10Mbits/sec	-	-	-	-	OFF
Master link 1-3 στα 20Mbits/sec	-	-	-	-	ON

Αριθμός του διακόπτη Χρώμα του διακόπτη	SW1.6 μπλε	SW1.7 μοβ	SW1.8 γκρι	SW1.9 λευκό	SW1.10 μαύρο
Internal subsystem disabled	OFF	-	-	-	-
Internal subsystem enabled	ON	-	-	-	-
Switches not used	-	-	-	-	-
Slave links στα 10Mbits/sec	-	-	-	-	OFF
Slave links στα 20Mbits/sec	-	-	-	-	ON

Πίνακας Δ2. Διακόπτης SW2

Αριθμός του διακόπτη Χρώμα του διακόπτη	SW2.1 καφέ	SW2.2 κόκκινο	SW2.3 πορτοκ.	SW2.4 κίτρινο	SW2.5 πράσινο
Host ανενεργός Host system δε συνδέεται στο master	OFF	OFF	-	-	-
Host ενεργός Host system συνδέεται στο master	ON	ON	-	-	-
Host link δε συνδ. στο master link0	-	-	OFF		-
Host link συνδέεται στο master link 0	-	-	ON		-
West link δε συνδ. στο master link 1	-	-	-	OFF	-
West link συνδέεται στο master link 1	-	-	-	ON	-
East link δε συνδ. στο master link 3	-	-	-	-	OFF
East link συνδέεται στο master link 3	-	-	-	-	ON

Για τον έξω κόσμο η Fast9 έχει τέσσερα γκρουπ από γραμμές συστήματος ελέγχου: τις **UP**, **DOWN**, **Subsystem** και **Host**, που είναι διαθέσιμες και από το "Rear connector".

Οι διακόπτες SW 2.1 και SW 2.2 ενώνουν το Host system με το Up system και ο διακόπτης SW 1.6 καθορίζει την εσωτερική οργάνωση λειτουργίας των επεξεργαστών.

Για τον προγραμματισμό του link exchange και την υλοποίηση μιας συγκεκριμένης τοπολογίας του δικτύου των transputers, προσφέρεται μαζί με την κάρτα το πρόγραμμα Fast9config καθώς και διάφορες έτοιμες τοπολογίες. Το πρόγραμμα αυτό αρχικοποιεί το "Link exchange" ώστε να αναιρεθούν οι τυχόν υπάρχουσες προγραμματισμένες συνδέσεις και μετά προγραμματίζει τις επιθυμητές.

Οι επιθυμητές συνδέσεις ορίζονται σε ένα αρχείο (hardware configuration file), το οποίο χρησιμοποιείται σαν παράμετρος στο Fast9config. Δηλαδή:

C:\> F9config <filename.cfg>

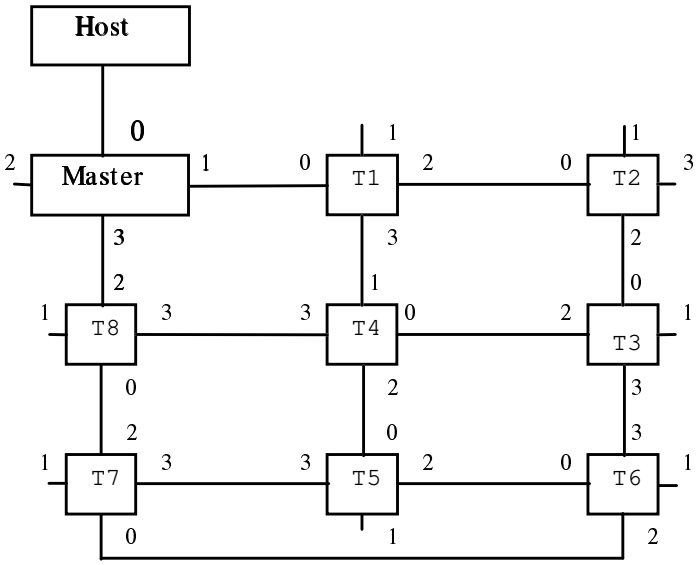
Δίνεται παρακάτω ένα παράδειγμα ενός τέτοιου αρχείου για την τοπολογία πλέγματος (mesh) (σχ. Δ7).

Slave1_link3 Slave4_link1

Slave3_link3 Slave6_link3

Slave4_link3 Slave8_link3

Slave5_link3 Slave7_link3



Σχήμα Δ7. Τοπολογία πλέγματος

Παράρτημα Ε

Περιληπτική περιγραφή της 3L Parallel C

Η γλώσσα προγραμματισμού Parallel C της 3L αναπτύχθηκε ειδικά για προγραμματισμό σε περιβάλλον πολυεπεξεργασίας με transputers. Παρέχει τη δυνατότητα διαχωρισμού μιας εφαρμογής σε επιμέρους διεργασίες, οι οποίες είτε θα εκτελούνται σε έναν transputer (ψευδο-παράλληλα), είτε σε διαφορετικούς (παράλληλα).

Η Parallel C της 3L βασίζεται κυρίως στη διάλεκτο της ANSI C. Η εκμετάλευση των ιδιαιτέρων χαρακτηρισστικών των transputer, όπως η επικοινωνία μεταξύ τους, η ταυτόχρονη εκτέλεση κλπ., γίνεται με την χρήση ορισμένων επιπλέον συναρτήσεων.

Οι διεργασίες που αποτελούν την εφαρμογή χρειάζεται να επικοινωνούν μεταξύ τους και η επικοινωνία αυτή απαιτεί να ορίσουμε τον τρόπο με τον οποίο κατανέμονται αυτές. Η κατανομή αυτή των διεργασιών υλοποιείται από προγράμματα διαμόρφωσης (configurers) που παρέχει η 3L Parallel C. Υπάρχουν δύο προγράμματα διαμόρφωσης (config και fconfig). Η επιλογή μεταξύ τους γίνεται ανάλογα με τον τρόπο παραλληλοποίησης του αλγόριθμου.

Το config επιλέγεται όταν ο αλγόριθμος παραλληλοποιείται σε διεργασίες οι οποίες εκτελούν διαφορετικές εντολές και οι οποίες, ανά διαστήματα, ανταλλάσσουν δεδομένα. Σε κάθε κόμβο του δικτύου των transputers ανατίθεται μια τέτοια διεργασία. Για να γίνει δυνατή η ανταλλαγή δεδομένων πρέπει να είναι γνωστή η τοπολογία του δικτύου. Η τοπολογία περιγράφεται σε ένα αρχείο διαμόρφωσης (configuration file).

Το fconfig επιλέγεται όταν οι διάφορες διεργασίες εκτελούν τις ίδιες εντολές, σε διαφορετικά δεδομένα. Τότε δεν ενδιαφέρει η σύνδεση των διεργασιών αυτών, αφού δεν χρειάζεται να ανταλάσουν δεδομένα μεταξύ τους. Σε τέτοιου είδους εφαρμογές απαιτείται η ύπαρξη μιας επιπλέον διεργασίας, η οποία θα αναλαμβάνει τον έλεγχο των υπολοίπων.

Παρακάτω αναφέρονται τα βασικά βήματα κατά την ανάπτυξη και εκτέλεση μιας εφαρμογής με την 3L Parallel C.

i. Δημιουργία πηγαίου κώδικα με την χρήση κάποιου editor.

ii. Μεταγλώτιση του πηγαίου κώδικα και δημιουργία του αντικειμενικού κώδικα (object code) με εντολές του T800 transputer. T8 transputer. Αυτό γίνεται με την χρήση του compiler t8c

C:\> t8c <filename>

Αν δεν υπάρχουν λάθη θα παραχθεί ο αντικειμενικός κώδικας filename.bin.

iii. Ο αντικειμενικός κώδικας, πρέπει να συνδεθεί (linking) με όλες τις εξωτερικές συναρτήσεις και/ή βιβλιοθήκες που απαιτούνται. Αυτό γίνεται με την εντολή:

C:\> t8clink <filename>

Το t8clink είναι ένα batch αρχείο, το οποίο παράγει το εκτελέσιμο πρόγραμμα filename.b4.

iv. Το εκτελέσιμο πρόγραμμα, φορτώνεται στην κάρτα των transputers και εκτελείται χρησιμοποιώντας το πρόγραμμα afserver. Αυτό είναι ένα συνήθες πρόγραμμα που τρέχει στον επεξεργαστή του PC (Host) καθ'όλη τη διάρκεια που το παράλληλο πρόγραμμα εκτελείται στη κάρτα των transputers.

```
C:\> afserver -:b hello.b4
```

Η παράλληλη επεξεργασία σε συστήματα transputers βασίζεται στην ιδέα της συγχρονισμένης επικοινωνίας μηνυμάτων μεταξύ σειριακών διαδικασιών (communicating sequential processes).

Σ'αυτό το μοντέλο, το "υπολογιστικό σύστημα" είναι ένα σύνολο διεργασιών οι οποίες εκτελούνται ταυτόχρονα και μπορούν να επικοινωνούν μεταξύ τους μόνο μέσω "καναλιών" (channels). Ένα τέτοιο κανάλι συνδέει μόνο μια συγκεκριμένη διεργασία με μια άλλη. Μεταφέρει δε, μηνύματα μόνο προς μια κατεύθυνση. Για αμφίδρομη επικοινωνία μεταξύ δύο διεργασιών πρέπει να χρησιμοποιηθούν δύο κανάλια. Βασική αρχή που διέπει την επικοινωνία αυτή είναι ο συγχρονισμός των καναλιών. Κάθε διαδικασία μπορεί να έχει οποιοδήποτε αριθμό καναλιών, αλλά δεν μπορούν να προστεθούν καινούρια κατά τη διάρκεια της εκτέλεσης.

Κάθε ένας από τους τέσσερις συνδέσμους (links) κάθε transputer έχει δύο κανάλια, ένα για κάθε κατεύθυνση. Τα κανάλια του υλικού, τα "φυσικά κανάλια", συμπεριφέρονται ακριβώς όπως τα θεωρητικά κανάλια που είδαμε πιο πάνω και παρέχουν αμφίδρομη επικοινωνία. Με την χρήση των συνδέσμων είναι δυνατόν να δημιουγηθούν δίκτυα από transputers. Ο μόνος περιορισμός είναι ότι κάθε transputer δεν μπορεί να συνδεθεί απευθείας με περισσότερους από τέσσερις άλλους.

Σε έναν transputer είναι δυνατόν να εκτελούνται ταυτόχρονα πολλές διαδικασίες. Οποιαδήποτε λέξη (word) της μνήμης του transputer μπορεί να χρησιμοποιηθεί σαν κανάλι για τη σύνδεση δύο εσωτερικών διαδικασιών. Οι εντολές της Parallel C που χρησιμοποιούνται για αποστολή και λήψη μηνυμάτων (συναρτήσεις), χρησιμοποιούν τη διεύθυνση αυτής της λέξης-κανάλι για να πραγματοποιήσουν την επικοινωνία. Το περιεχόμενο της λέξης αυτής χρησιμοποιείται από τον tranputer για τον συγχρονισμό των διαδικασιών που στέλνουν και λαμβάνουν μηνύματα.

Από προγραμματιστικής άποψης αυτά τα εσωτερικά κανάλια, "λογικά κανάλια", είναι ακριβώς ίδια με τα φυσικά κανάλια. Οι ίδιες εντολές (συναρτήσεις) χρησιμοποιούνται για τον χειρισμό και των δύο τύπων. Τα φυσικά κανάλια έχουν ειδικές σταθερές διευθύνσεις, ενώ τα λογικά έχουν διευθύνσεις που ορίζονται από το πρόγραμμα. Η εξίσωση των λογικών καναλιών με τα φυσικά, σημαίνει ότι είναι δυνατό να αναπτυχθεί μια παράλληλη εφαρμογή σε ένα μόνο transputer και αργότερα να μεταφερθούν κάποιες από τις διεργασίες της σε άλλους transputers.

Μια εφαρμογή αποτελείται από ένα σύνολο επιμέρους διεργασιών (tasks), οι οποίες εκτελούνται ταυτόχρονα. Κάθε διεργασία έχει τη δική της περιοχή μνήμης για κώδικα και δεδομένα, ένα διάνυσμα με τις θύρες εισόδου (input ports) και ένα με τις θύρες εξόδου (output ports). Κάθε στοιχείο των διανυσμάτων αυτών είναι μια ειδικού τύπου μεταβλητή, δείκτης σε

λέξη-κανάλι (CHAN *). Αυτά τα διανύσματα περνάνε στις διεργασίες μέσω των παραμέτρων του κυρίως προγράμματος (main).

Οι διεργασίες είναι δυνατό να θεωρηθούν "μαύρα κουτιά" που συνδέονται μεταξύ τους μέσω των θυρών όπως φαίνεται στο σχήμα Ε1, ανεξάρτητα μέρη ενός παράλληλου συστήματος συνδεδεμένα με καλώδια, σαν να ήταν ηλεκτρονικά στοιχεία

Σχήμα Ε1. Διεργασία (Task)

Το ειδικό πρόγραμμα που κάνει τη διαμόρφωση (configuration) της τοπολογίας, παρέχει τρόπους για να ορίσουμε σε ποιον transputer θα ανατεθεί κάθε διεργασία. Κάθε επεξεργαστής μπορεί να υποστηρίξει πολλές διεργασίες, με μοναδικό περιορισμό τη διαθέσιμη μνήμη. Διακρίνουμε δύο περιπτώσεις:

1) Διεργασίες που βρίσκονται στον ίδιο transputer μπορούν να έχουν οποιοδήποτε αριθμό λογικών καναλιών για την μεταξύ τους επικοινωνία.

2) Διεργασίες που βρίσκονται σε διαφορετικούς transputers μπορούν να επικοινωνούν μόνο μέσω των συνδέσμων των συγκεκριμένων transputers.

Κάθε σύνδεση μεταξύ δύο διεργασιών που βρίσκονται σε διαφορετικούς transputers χρησιμοποιεί κατά αποκλειστικότητα μια από τις φυσικές συνδέσεις μεταξύ των transputers. Έτσι οι λογικές συνδέσεις αυτών των διεργασιών είναι περιορισμένες από τον αριθμό των φυσικών συνδέσεων που διαθέτουν οι συγκεκριμένοι transputers. Ειδικές διεργασίες, “πολυπλέκτες” (multiplexers), χρησιμοποιούνται όταν χρειάζονται περισσότερες από τέσσερις λογικές συνδέσεις σε κάθε κατεύθυνση (είσοδο-έξοδο), μεταξύ ενός transputer και των γειτονικών του.

Η παραλληλοποίηση μιας εφαρμογής, είναι ο διαχωρισμός της σε ένα σύνολο διεργασιών, που επικοινωνούν μεταξύ τους και λειτουργούν ταυτόχρονα. Η επικοινωνία απαιτεί, λόγω των περιορισμών που αναφέρθηκαν, μια συγκεκριμένη διάταξη των διεργασιών (διάφορες λογικές συνδέσεις μεταξύ τους). Όταν είναι σε διαφορετικούς transputers τότε απαιτείται και μια συγκεκριμένη διαμόρφωση του συστήματος των transputers (διάφορες φυσικές συνδέσεις μεταξύ τους). Οι φυσικές και λογικές συνδέσεις μεταξύ των διεργασιών και των transputers, καθώς και οι αναθέσεις των διεργασιών στους transputers δηλώνονται σε ένα αρχείο (configuration file) το οποίο ορίζει:

- τους επεξεργαστές και τις φυσικές συνδέσεις που απαιτεί η εφαρμογή.
- τα ονόματα των εκτελέσιμων αρχείων (.b4) των διαφόρων διεργασιών.
- τις λογικές συνδέσεις των διεργασιών μέσω των διαφόρων ports.
- την τοποθέτηση συγκεκριμένων διεργασιών σε συγκεκριμένους transputers.

Η επικοινωνία μεταξύ των διεργασιών γίνεται με την ανταλλαγή μηνυμάτων οποιουδήποτε μήκους μέσα από τις λογικές συνδέσεις (channels) που συνδέουν τα input-output ports των

αντιστοίχων tasks. Το αρχείο της βιβλιοθήκης συναρτήσεων <chan.h> χρησιμοποιείται γι' αυτό το σκοπό.

Οι συναρτήσεις που χρησιμοποιούνται πιο πολύ και είναι η γενικευμένη περίπτωση για όλες τις άλλες είναι:

- chan_out_message(int len, char * b, CHAN * chan)
- chan_in_message(int len, char *b, CHAN *chan)

Τα τρία ορίσματα είναι:

- ένας ακέραιος ίσος με τον αριθμό των bytes που θέλουμε να στείλουμε.
- η διεύθυνση από την οποία αρχίζει το μήνυμα που θα σταλεί.
- το output port μέσω του οποίο στέλνουμε το μήνυμα. Είναι μια μεταβλητή τύπου CHAN δηλαδή δείκτης σε λέξη.

Οι εντολές αυτές αποτελούν ζεύγος και πρέπει να χρησιμοποιηθούν από τις διεργασίες εκπομπής και λήψης αντίστοιχα. Τα ορίσματα του ζεύγους των εντολών chan_in_message και chan_out_message πρέπει να αναφέρονται στο ίδιο μήνυμα (ίδιου μήκους) και τα input-output ports πρέπει να είναι αυτά που ορίζονται από τη λογική σύνδεση των δύο διεργασιών. Οι εντολές αυτές επιτρέπουν τον συγχρονισμό των διεργασιών, αφού για να συνεχιστεί η εκτέλεση μιας διεργασίας μετά την εντολή chan_in_message, το άλλο πρέπει να έχει εκτελέσει την chan_out_message.

Ο transputer που επικοινωνεί με τον host Η/Υ ονομάζεται root. Σ' αυτόν τοποθετείται και η διεργασία "φίλτρο" (filter) που εξασφαλίζει την επικοινωνία του afserver με την κάρτα των transputers και τρέχει παράλληλα με τις άλλες διεργασίες. Μόνο ο root transputer μπορεί να εκτελεί εντολές εισόδου-εξόδου, όπως διάβασμα από το πληκτρολόγιο, εμφάνιση στην οθόνη κ.λ.π. Υπάρχει βέβαια μια τεχνική με πολυπλέκτες που χρησιμοποιείται για να υπερβούμε αυτόν τον περιορισμό (εκτενέστερη περιγραφή στο manual της Parallel C).

Στο επίπεδο των επεξεργαστών, φυσικές συνδέσεις συνδέουν τον επεξεργαστή host με τον root transputer και τον root transputer με τους transputers T1 και T8. Στο επίπεδο των διεργασιών, αμφίδρομες λογικές συνδέσεις συνδέουν τον afserver με τον filter, τον filter με την κύρια διεργασία που τρέχει στον root και αυτή με άλλες διεργασίες μέσω των συγκεκριμένων ports. Μόνο μία διεργασία μπορεί να συνδεθεί με τον afserver. Έτσι οι εφαρμογές πρέπει να αναπτύσσονται με τέτοιο τρόπο ώστε μία μόνο διεργασία να χρησιμοποιεί εντολές εισόδου-εξόδου. Η διεργασία αυτή συνδέεται με τον afserver και με την C run-time-library. Όλες οι άλλες διεργασίες συνδέονται με μια ειδική έκδοση της C run-time-library, την stand-alone, η οποία δεν απαιτεί επικοινωνία με τον afserver. Επίσης οι διεργασίες αυτές δεν χρησιμοποιούν τις συνηθισμένες εντολές της C για είσοδο-έξοδο αλλά τις συναρτήσεις επικοινωνίας της Parallel C για σύνδεση με το παραπάνω task εισόδου-εξόδου. Η stand-alone C run-time-library περιέχει όλες τις συναρτήσεις εκτός από αυτές που χρειάζονται υποστήριξη από τον afserver (I/O, date, DOS calls κ.λ.π).

Όταν η εφαρμογή αποτελείται από ανεξάρτητες όμοιες διεργασίες χωρίς να απαιτείται η μεταξύ τους επικοινωνία, είναι δυνατή η χρήση ενός άλλου προγράμματος διευθέτησης (configurer), τον flood fill configurer. Αυτό συνδέει τους κώδικες του master task και του worker task σε ένα εκτελέσιμο αρχείο, το οποίο μπορεί να φορτώνεται αυτόματα και να τρέχει σε οποιοδήποτε δίκτυο transputer, με τον afserver.

Μία διεργασία μπορεί να περιέχει έναν αριθμό από ταυτόχρονα (concurrent) threads, τα οποία μοιράζονται το χρόνο επεξεργαστή με τη σειρά. Όλα τα treads κάποιου task μοιράζονται τα ίδια δεδομένα αλλά μπορούν να λειτουργούν ανεξάρτητα, επειδή καθένα έχει το δικό του σωρό (stack) για τις τοπικές μεταβλητές (auto variables).

Καινούρια threads μπορούν να δημιουργηθούν μέσα σε μία διεργασία καλώντας τη συνάρτηση βιβλιοθήκης:

tread_create(funct, stacksize, nargc, a, b,....)

όπου funct ένα thread που απαιτεί σωρό μεγέθους ίσο με τον ακέραιο stacksize, και έχει nargc ορίσματα, τα a,b, ...

Αν σε μία διεργασία, λειτουργούν πάνω στα ίδια δεδομένα πολλά threads, τότε πρέπει να υπάρχει ένας συγχρονισμός των προσβάσεων σ' αυτά, για να αποφευχθούν συγκρούσεις και αδιέξοδα. Τα threads μπορούν να συγχρονίσουν τις λειτουργίες τους χρησιμοποιώντας τα κανάλια (channels) ή σηματοφόρους (semaphors). Πρέπει να επισημανθεί ότι ένας transputer έχει μία μόνο CPU και συνεπώς δεν μπορεί να εκτελέσει δύο πράξεις συγχρόνως. Έτσι διαφορετικά threads χρησιμοποιούνται συνήθως, στην υλοποιήση διαδικασιών επικοινωνίας, οι οποίες από τη φύση τους είναι ασύγχρονες.

Παράρτημα ΣΤ

Αρχεία Διαμόρφωσης (Configuration Files)

Α. Περιεχόμενα του αρχείου διαμόρφωσης της τοπολογίας σύνδεσης των transputers

```
# MAX_CON9
Slave1_Link1    Slave4_Link1
Slave1_Link3    Slave6_Link3
Slave2_Link1    Slave7_Link1
Slave2_Link3    Slave5_Link3
Slave3_Link1    Slave6_Link1
Slave3_Link3    Slave8_Link3
Slave4_Link3    Slave7_Link3
Slave5_Link1    Slave8_Link1
```

Οι παραπάνω συνδέσεις χρησιμοποιούνται για τον προγραμματισμό του Link Exchange δημιουργώντας την τοπολογία σύνδεσης του σχήματος 5.2

Β. Αρχεία διαμόρφωσης των παράλληλων υλοποιήσεων.

Συνδέσεις μεταξύ των διεργασιών και αναθέσεις των διεργασιών στους επεξεργαστές.

Β1. Αρχεία διαμόρφωσης για το πρόβλημα της εκπαίδευσης του αναδρομικού τεχνητού νευρωνικού δικτύου

```
!
! GNN.CFG      configuration file for fixed hardware configuration of the feed-forward neural
               network simulation, using all the transputers on the board

!              It is NECESSARY to set up the link exchange with the configuration max_con9.cfg,
                using the command:
!              C:\FASTSOFT\FASTSOFT>f9config max_con.cfg
!
! Hardware
processor host
processor root
processor worker1
processor worker2
processor worker3
processor worker4
processor worker5
processor worker6
processor worker7
processor worker8

wire ? host[0] root[0]          !anonymous wire connecting PC to root transputer
wire ? root[1] worker1[0]       !anonymous wire connecting root to worker1
wire ? root[3] worker8[2]       !anonymous wire connecting root to worker8
wire ? worker1[1] worker2[1]    !anonymous wire connecting worker1 to worker2
wire ? worker1[2] worker4[0]    !anonymous wire connecting worker1 to worker4
wire ? worker1[3] worker6[3]    !anonymous wire connecting worker1 to worker6
wire ? worker8[0] worker5[2]    !anonymous wire connecting worker8 to worker5
wire ? worker8[1] worker7[1]    !anonymous wire connecting worker8 to worker7
wire ? worker8[3] worker3[3]    !anonymous wire connecting worker8 to worker3
wire ? worker2[0] worker3[2]    !anonymous wire connecting worker2 to worker3
wire ? worker2[2] worker7[0]    !anonymous wire connecting worker2 to worker7
wire ? worker2[3] worker5[3]    !anonymous wire connecting worker2 to worker5
wire ? worker3[0] worker4[2]    !anonymous wire connecting worker3 to worker4
wire ? worker3[1] worker6[1]    !anonymous wire connecting worker3 to worker6
wire ? worker4[1] worker5[1]    !anonymous wire connecting worker4 to worker5
```

```
wire ? worker4[3] worker7[3]    !anonymous wire connecting worker4 to worker5
wire ? worker5[0] worker6[2]    !anonymous wire connecting worker1 to worker2
wire ? worker6[0] worker7[2]    !anonymous wire connecting worker1 to worker4
!
! Task declarations indicating channel I/O ports and memory requirements
task  afserver   ins=1   outs=1
task  filter     ins=2   outs=2  data=10K
task  gnnm     ins=4   outs=4
task  gnnw1    ins=4   outs=4
task  gnnw2    ins=4   outs=4
task  gnnw3    ins=4   outs=4
task  gnnw4    ins=4   outs=4
task  gnnw5    ins=4   outs=4
task  gnnw6    ins=4   outs=4
task  gnnw7    ins=4   outs=4
task  gnnw8    ins=4   outs=4
!
! Assign software tasks to physical processors
place  afserver   host
place  gnnm     root
place  filter     root
place  gnnw1    worker1
place  gnnw2    worker2
place  gnnw3    worker3
place  gnnw4    worker4
place  gnnw5    worker5
place  gnnw6    worker6
place  gnnw7    worker7
place  gnnw8    worker8
!
! Set up the connections between the tasks.
connect ? afserver[0] filter[0]
connect ? filter[0] afserver[0]
connect ? filter[1] gnnm[1]
connect ? gnnm[1] filter[1]
connect ? gnnm[2] gnnw1[0]
connect ? gnnw1[0] gnnm[2]
connect ? gnnm[3] gnnw8[0]
connect ? gnnw8[0] gnnm[3]
connect ? gnnw1[1] gnnw2[0]
connect ? gnnw2[0] gnnw1[1]
connect ? gnnw1[2] gnnw4[0]
connect ? gnnw4[0] gnnw1[2]
connect ? gnnw1[3] gnnw6[0]
```

```
connect ? gnnw6[0] gnnw1[3]
connect ? gnnw8[1] gnnw3[0]
connect ? gnnw3[0] gnnw8[1]
connect ? gnnw8[2] gnnw5[0]
connect ? gnnw5[0] gnnw8[2]
connect ? gnnw8[3] gnnw7[0]
connect ? gnnw7[0] gnnw8[3]
connect ? gnnw2[1] gnnw3[1]
connect ? gnnw3[1] gnnw2[1]
connect ? gnnw2[2] gnnw5[1]
connect ? gnnw5[1] gnnw2[2]
connect ? gnnw2[3] gnnw7[1]
connect ? gnnw7[1] gnnw2[3]
connect ? gnnw3[2] gnnw4[1]
connect ? gnnw4[1] gnnw3[2]
connect ? gnnw3[3] gnnw6[1]
connect ? gnnw6[1] gnnw3[3]
connect ? gnnw4[2] gnnw5[2]
connect ? gnnw5[2] gnnw4[2]
connect ? gnnw4[3] gnnw7[2]
connect ? gnnw7[2] gnnw4[3]
connect ? gnnw5[3] gnnw6[2]
connect ? gnnw6[2] gnnw5[3]
connect ? gnnw6[3] gnnw7[3]
connect ? gnnw7[3] gnnw6[3]
```

Β2. Αρχεία διαμόρφωσης για το πρόβλημα της βελιστοποίησης της συνάρτησης Rastrigin

```
!
! RASTR.CFG      configuration file for fixed hardware configuration of the optimization of the
                  Rastrigin function, using all the transputers on the board
!
!           It is NECESSARY to set up the link exchange with the configuration file
!           max_con9.cfg, using the command:          C:\ >f9config max_con.cfg
!
! Hardware
processor host
processor root
processor worker1
processor worker2
processor worker3
processor worker4
```

```
processor worker5
processor worker6
processor worker7
processor worker8

wire ? host[0] root[0]              !anonymous wire connecting PC to root transputer
wire ? root[1] worker1[0]           !anonymous wire connecting root to worker1
wire ? root[3] worker8[2]           !anonymous wire connecting root to worker8
wire ? worker1[1] worker2[1]   !anonymous wire connecting worker1 to worker2
wire ? worker1[2] worker4[0]   !anonymous wire connecting worker1 to worker4
wire ? worker1[3] worker6[3]   !anonymous wire connecting worker1 to worker6
wire ? worker8[0] worker5[2]   !anonymous wire connecting worker8 to worker5
wire ? worker8[1] worker7[1]   !anonymous wire connecting worker8 to worker7
wire ? worker8[3] worker3[3]   !anonymous wire connecting worker8 to worker3
wire ? worker2[0] worker3[2]   !anonymous wire connecting worker2 to worker3
wire ? worker2[2] worker7[0]   !anonymous wire connecting worker2 to worker7
wire ? worker2[3] worker5[3]   !anonymous wire connecting worker2 to worker5
wire ? worker3[0] worker4[2]   !anonymous wire connecting worker3 to worker4
wire ? worker3[1] worker6[1]   !anonymous wire connecting worker3 to worker6
wire ? worker4[1] worker5[1]   !anonymous wire connecting worker4 to worker5
wire ? worker4[3] worker7[3]   !anonymous wire connecting worker4 to worker5
wire ? worker5[0] worker6[2]   !anonymous wire connecting worker1 to worker2
wire ? worker6[0] worker7[2]   !anonymous wire connecting worker1 to worker4
!
! Task declarations indicating channel I/O ports and memory requirements
task  afserver   ins=1   outs=1
task  filter     ins=2   outs=2  data=10K
task  rastrm     ins=4   outs=4
task  rastrw1    ins=4   outs=4
task  rastrw2    ins=4   outs=4
task  rastrw3    ins=4   outs=4
task  rastrw4    ins=4   outs=4
task  rastrw5    ins=4   outs=4
task  rastrw6    ins=4   outs=4
task  rastrw7    ins=4   outs=4
task  rastrw8    ins=4   outs=4
!
! Assign software tasks to physical processors
place  afserver   host
place  rastrm     root
place  filter     root
place  rastrw1    worker1
place  rastrw2    worker2
place  rastrw3    worker3
```

```
place  rastrw4    worker4
place  rastrw5    worker5
place  rastrw6    worker6
place  rastrw7    worker7
place  rastrw8    worker8
!
! Set up the connections between the tasks.
connect ? afserver[0] filter[0]
connect ? filter[0] afserver[0]
connect ? filter[1] rastrm[1]
connect ? rastrm[1] filter[1]
connect ? rastrm[2] rastrw1[0]
connect ? rastrw1[0] rastrm[2]
connect ? rastrm[3] rastrw8[0]
connect ? rastrw8[0] rastrm[3]
connect ? rastrw1[1] rastrw2[0]
connect ? rastrw2[0] rastrw1[1]
connect ? rastrw1[2] rastrw4[0]
connect ? rastrw4[0] rastrw1[2]
connect ? rastrw1[3] rastrw6[0]
connect ? rastrw6[0] rastrw1[3]
connect ? rastrw8[1] rastrw3[0]
connect ? rastrw3[0] rastrw8[1]
connect ? rastrw8[2] rastrw5[0]
connect ? rastrw5[0] rastrw8[2]
connect ? rastrw8[3] rastrw7[0]
connect ? rastrw7[0] rastrw8[3]
connect ? rastrw2[1] rastrw3[1]
connect ? rastrw3[1] rastrw2[1]
connect ? rastrw2[2] rastrw5[1]
connect ? rastrw5[1] rastrw2[2]
connect ? rastrw2[3] rastrw7[1]
connect ? rastrw7[1] rastrw2[3]
connect ? rastrw3[2] rastrw4[1]
connect ? rastrw4[1] rastrw3[2]
connect ? rastrw3[3] rastrw6[1]
connect ? rastrw6[1] rastrw3[3]
connect ? rastrw4[2] rastrw5[2]
connect ? rastrw5[2] rastrw4[2]
connect ? rastrw4[3] rastrw7[2]
connect ? rastrw7[2] rastrw4[3]
connect ? rastrw5[3] rastrw6[2]
connect ? rastrw6[2] rastrw5[3]
connect ? rastrw6[3] rastrw7[3]
```

```
connect ? rastrw7[3] rastrw6[3]
```

Β3. Αρχεία διαμόρφωσης για το πρόβλημα του περιοδεύοντος πωλητή

```
!
! TSP.CFG      configuration file for fixed hardware configuration of the TSP problem, using all
                the transputers on the board

!             It is NECESSARY to set up the link exchange with the configuration file max_con9.cfg,
               using the command:
!             C:\>f9config max_con.cfg
!
! Hardware
processor host
processor root
processor worker1
processor worker2
processor worker3
processor worker4
processor worker5
processor worker6
processor worker7
processor worker8

wire ? host[0] root[0]        !anonymous wire connecting PC to root transputer
wire ? root[1] worker1[0]     !anonymous wire connecting root to worker1
wire ? root[3] worker8[2]     !anonymous wire connecting root to worker8
wire ? worker1[1] worker2[1]  !anonymous wire connecting worker1 to worker2
wire ? worker1[2] worker4[0]  !anonymous wire connecting worker1 to worker4
wire ? worker1[3] worker6[3]  !anonymous wire connecting worker1 to worker6
wire ? worker8[0] worker5[2]  !anonymous wire connecting worker8 to worker5
wire ? worker8[1] worker7[1]  !anonymous wire connecting worker8 to worker7
wire ? worker8[3] worker3[3]  !anonymous wire connecting worker8 to worker3
wire ? worker2[0] worker3[2]  !anonymous wire connecting worker2 to worker3
wire ? worker2[2] worker7[0]  !anonymous wire connecting worker2 to worker7
wire ? worker2[3] worker5[3]  !anonymous wire connecting worker2 to worker5
wire ? worker3[0] worker4[2]  !anonymous wire connecting worker3 to worker4
wire ? worker3[1] worker6[1]  !anonymous wire connecting worker3 to worker6
wire ? worker4[1] worker5[1]  !anonymous wire connecting worker4 to worker5
wire ? worker4[3] worker7[3]  !anonymous wire connecting worker4 to worker5
wire ? worker5[0] worker6[2]  !anonymous wire connecting worker1 to worker2
wire ? worker6[0] worker7[2]  !anonymous wire connecting worker1 to worker4
```

```
!
! Task declarations indicating channel I/O ports and memory requirements
!
task  afserver   ins=1   outs=1
task  filter     ins=2   outs=2  data=10K
task  tspm     ins=4   outs=4
task  tspw1    ins=4   outs=4
task  tspw2    ins=4   outs=4
task  tspw3    ins=4   outs=4
task  tspw4    ins=4   outs=4
task  tspw5    ins=4   outs=4
task  tspw6    ins=4   outs=4
task  tspw7    ins=4   outs=4
task  tspw8    ins=4   outs=4
!
! Assign software tasks to physical processors
!
place  afserver   host
place  tspm     root
place  filter     root
place  tspw1    worker1
place  tspw2    worker2
place  tspw3    worker3
place  tspw4    worker4
place  tspw5    worker5
place  tspw6    worker6
place  tspw7    worker7
place  tspw8    worker8
!
! Set up the connections between the tasks.
!
connect ? afserver[0] filter[0]
connect ? filter[0] afserver[0]
connect ? filter[1] tspm[1]
connect ? tspm[1] filter[1]
connect ? tspm[2] tspw1[0]
connect ? tspw1[0] tspm[2]
connect ? tspm[3] tspw8[0]
connect ? tspw8[0] tspm[3]
connect ? tspw1[1] tspw2[0]
connect ? tspw2[0] tspw1[1]
connect ? tspw1[2] tspw4[0]
connect ? tspw4[0] tspw1[2]
connect ? tspw1[3] tspw6[0]
```

connect ? tspw6[0] tspw1[3]
connect ? tspw8[1] tspw3[0]
connect ? tspw3[0] tspw8[1]
connect ? tspw8[2] tspw5[0]
connect ? tspw5[0] tspw8[2]
connect ? tspw8[3] tspw7[0]
connect ? tspw7[0] tspw8[3]
connect ? tspw2[1] tspw3[1]
connect ? tspw3[1] tspw2[1]
connect ? tspw2[2] tspw5[1]
connect ? tspw5[1] tspw2[2]
connect ? tspw2[3] tspw7[1]
connect ? tspw7[1] tspw2[3]
connect ? tspw3[2] tspw4[1]
connect ? tspw4[1] tspw3[2]
connect ? tspw3[3] tspw6[1]
connect ? tspw6[1] tspw3[3]
connect ? tspw4[2] tspw5[2]
connect ? tspw5[2] tspw4[2]
connect ? tspw4[3] tspw7[2]
connect ? tspw7[2] tspw4[3]
connect ? tspw5[3] tspw6[2]
connect ? tspw6[2] tspw5[3]
connect ? tspw6[3] tspw7[3]
connect ? tspw7[3] tspw6[3]

Παράρτημα Ζ

Οδηγίες Χρήσης Προγραμμάτων

Σε αυτό το παράρτημα παρατίθενται οδηγίες χρήσεως και περιγραφή της παράλληλης προσομοίωσης, τόσο των *Συνεργαζόμενων Πληθυσμών με Διαφορετικές Εξελικτικές Συμπεριφορές (CoPDEB - Co-operating Populations with Different Evolution Behavior)*, όσο και του κλασσικού ΠΓΑ με την ίδια Εξελικτική Συμπεριφορά σε κάθε Πληθυσμό (*Μοντέλο Νησίδων*). Ο τρόπος χρήσης και η περιγραφή των δύο προσομοιώσεων αναλύονται μέσω της επίλυσης των εξής προβλημάτων:

1. Εκπαίδευση Αναδρομικού Τεχνητού Δικτύου
2. Βελτιστοποίηση της συνάρτησης Rastrigin
3. Πρόβλημα Περιοδεύοντος Πωλητή

Οι προσομοιώσεις έγιναν στην κάρτα Fast9, η οποία διαθέτει εννιά (9) transputers και είναι τοποθετημένη σε ένα PC. Ο transputer ο οποίος επικοινωνεί με τον επεξεργαστή του PC, θα αναφέρεται ως **συντονιστής (master)** και έργο του είναι να συντονίζει την λειτουργία των υπόλοιπων οι οποίοι θα αναφέρονται ως **εργαζόμενοι (workers)**.

Η γλώσσα Parallel C της 3L, χρησιμοποιείται για τον προγραμματισμό των προσομοιώσεων. Κάθε εργαζόμενος transputer εκτελεί ένα ΓΑ ανεξάρτητα από τους υπόλοιπους. Οι εργαζόμενοι transputers επικοινωνούν μεταξύ τους στέλνοντας τον καλύτερο γόνο του πληθυσμού τους σε όλους τους άλλους και τον συντονιστή ο οποίος είναι υπεύθυνος για την έναρξη και λήξη της προσομοίωσης. Ο συντονιστής στέλνει τις αρχικές απαραίτητες πληροφορίες σε όλους τους εργαζόμενους και παρακολουθεί την εξέλιξη των πληθυσμών για να κρίνει τον τερματισμό της υλοποίησης.

Α. Περιγραφή των αρχείων

Η διεργασία εξέλιξης κάθε ΓΑ, και η λειτουργία του συντονιστή ορίζονται σε ένα διαφορετικό πρόγραμμα για κάθε transputer. Συνολικά εννιά προγράμματα για κάθε προσομοίωση. Εκτός των εννιά προγραμμάτων σε γλώσσα C, πρέπει να οριστούν οι τοποθετήσεις των διεργασιών στους επεξεργαστές, όπως επίσης και οι συνδέσεις των επεξεργαστών μεταξύ τους στα αντίστοιχα αρχεία διαμόρφωσης (configuration files - Παράρτημα ΣΤ). Κάθε προσομοίωση αποτελείται από τα εξής αρχεία:

ΑΤΝΔ	**Rastrigin**	**TSP**	**Περιγραφή**
gnnm.c	rastrm.c	tspm.c	Κώδικας συντονιστή
gnnw1.c	rastrw1.c	tspw1.c	Κώδικας εργαζόμενου 1
gnnw2.c	rastrw2.c	tspw2.c	Κώδικας εργαζόμενου 2
gnnw3.c	rastrw3.c	tspw3.c	Κώδικας εργαζόμενου 3
gnnw4.c	rastrw4.c	tspw4.c	Κώδικας εργαζόμενου 4
gnnw5.c	rastrw5.c	tspw5.c	Κώδικας εργαζόμενου 5
gnnw6.c	rastrw6.c	tspw6.c	Κώδικας εργαζόμενου 6
gnnw7.c	rastrw7.c	tspw7.c	Κώδικας εργαζόμενου 7
gnnw8.c	rastrw8.c	tspw8.c	Κώδικας εργαζόμενου 8
gnn8.def	rastr8.def	tsp8.def	Ορισμός δομών και σταθερών
gnn.tsk	rastr.tsk	tsp.tsk	Παράμετροι και συνδέσεις
gnn8.cfg	rastr8.cfg	tsp8.cfg	Αρχείο διαμόρφωσης

Τα αρχεία "*m.c" περιέχουν τον κώδικα του συντονιστή. Ο κώδικας είναι ίδιος και για τις δύο προσομοιώσεις (*ΣΠΔΕΣ* και μοντέλο νησίδων). Διαφορές υπάρχουν μεταξύ των τριών προβλημάτων, κυρίως στην δομή των αρχικών πληροφοριών τις οποίες μεταβιβάζουν στους εργαζόμενους.

Ο συντονιστής επικοινωνεί με τους εργαζόμενους 1 και 8 (gnnw1.c, gnnw8.c), μέσω των οποίων επικοινωνεί και με τους υπόλοιπους. Παρακάτω δίνεται ο ψευδο-κώδικας του συντονιστή.

```
Εισαγωγή των στοιχείων του αρχείου παραμέτρων και συνδέσεων (*.tsk)
Αρχικές τιμές μεταβλητών
Αποστολή αρχικών πληροφοριών στους 8 υπο-πληθυσμούς
Επανέλαβε {
    Αύξηση του αριθμού εποχής
    Λήψη της ποιότητας των καλύτερων γόνων και των οκτώ υπο-πληθυσμών
    Εύρεση της καλύτερης ποιότητας
    Έλεγχος εάν η καλύτερη ποιότητα έχει βελτιωθεί
    Εμφάνιση μερικών αποτελεσμάτων
    Ενημέρωση τρέχοντος αριθμού γενεών
} μέχρι να βρεθεί το βέλτιστο ή να ξεπεραστεί το όριο γενεών ή να
    σταματήσει η βελτίωση
Εμφάνιση και αποθήκευση τελικών αποτελεσμάτων
```

Τα αρχεία "*w#.c" περιέχουν τον κώδικα του συντονιστή. Στην περίπτωση της προσομοίωσης του μοντέλου νησίδων, οι κώδικες των εργαζόμενων διαφέρουν στον τρόπο επικοινωνίας με τους υπόλοιπους εργαζόμενους. Ο εργαζόμενος 1 επικοινωνεί με τον συντονιστή και επίσης με τους εργαζόμενους 2, 3 και 5. Ο εργαζόμενος 8 επικοινωνεί με τον συντονιστή και επίσης με τους εργαζόμενους 4, 6 και 7.

Στην προσομοίωση των *ΣΠΔΕΣ* οι κώδικες διαφέρουν επίσης στον χρησιμοποιούμενο τελεστή ανασυνδυασμού και στις πιθανότητες ανασυνδυασμού και μετάλλαξης. Ο εργαζόμενος 1 χρησιμοποιεί τους ίδιους τελεστές και παραμέτρους με τους εργαζόμενους του μοντέλου νησίδων. Ο εργαζόμενος 2 χρησιμοποιεί τον τελεστή ανασυνδυασμού MSP_LSP (σελ. 49), ενώ ο 3 τον τελεστή LSP_ MSP (σελ. 50). Ο εργαζόμενος 4 χρησιμοποιεί ανασυνδυασμό δύο σημείων, ενώ ο 5 ανασυνδυασμό τόσων σημείων ίσο με το μισό των γονιδίων. Ο εργαζόμενος 6 χρησιμοποιεί

ομοιόμορφο (uniform) ανασυνδυασμό. Ο 7 ξεκινά με ίση πιθανότητα ανασυνδυασμού και μετάλλαξης (0.5) και υπάρχει μεγαλύτερη πιθανότητα αύξησης της πιθανότητας μετάλλαξης παρά του ανασυνδυασμού. Ο εργαζόμενος 8 χρησιμοποιεί σταθερές πιθανότητες μετάλλαξης (0.8) και ανασυνδυασμού (0.2).

Μεταξύ των τριών προβλημάτων οι κώδικες διαφέρουν κυρίως στην συνάρτηση ποιότητας.

Τα αρχεία "*.def" περιέχουν τον ορισμό των δομών δεδομένων και των σταθερών. Ενδεικτικά παρατίθεται το αρχείο που χρησιμοποιείται στο πρόβλημα του περιοδεύοντος πωλητή

```
/*************************************************************
**                TSP8.DEF
**
**    Implementation of a Genetic Algorithm simulation on a transputer board.
**    This file contains the structure definitions
*************************************************************/

#define pi 3.141592654
#define word unsigned long int
#define No_of_workers 8
#define gen_limit 10000

#define FALSE 0
#define TRUE 1
#define pper 10
#define SAME 300

/* =============== Global Declarations ============== */
typedef struct initial_info {
      int   m;        /* Population size */
      int   Gene_no;     /* No of genes */
      int   epoch;
} INITIAL_INFO;

typedef struct worker_performance {
      int   same;
      float  qual;
      float  fitness;
} PERFORMANCE;

typedef struct what_to_do_info {
      int   cont;
      int   rec_port,send_port[4];
} WHAT_TO_DO;

struct genotype {
      float  qual;
      float  qual2;
      word  *gene;
};

typedef struct gene_quality {
      float  qual;
      float  qual2;
} QUALITY;
```

Τα αρχεία "*.tsk" είναι αρχεία πληροφοριών τα οποία περιέχουν το όνομα του αρχείου όπου θα γραφούν τα αποτελέσματα, απαραίτητες γενικές παραμέτρους (π.χ. μέγεθος πληθυσμού και εποχής), παράμετροι που έχουν σχέση με το συγκεκριμένο πρόβλημα και τις συνδέσεις μεταξύ των διεργασιών. Ενδεικτικά παρατίθεται το αρχείο για το πρόβλημα βελτιστοποίησης της συνάρτησης Rastrigin με πληθυσμό 100 και τιμή εποχής 50 (rastr.tsk), .

```
REPORT_FILE:      pop100.e50


T_R_A_I_N___P_A_R_A_M_E_T_E_R_S

Population       100
No_fo_genes       20
Epoch             50
Max_error        0.001


T_A_S_K___C_O_N_N_E_C_T_I_O_N_S

Root      10   10   1   8
Worker1    0    2   4   6
Worker2    1    3   5   7
Worker3    8    2   4   6
Worker4    1    3   5   7
Worker5    8    2   4   6
Worker6    1    3   5   7
Worker7    8    2   4   6
Worker8    0    3   5   7
```

Τα αρχεία "*.cfg" καθορίζουν:

- την διαμόρφωση του υλικού (επεξεργαστές και τις συνδέσεις μεταξύ τους) στο οποίο θα εκτελεστεί η προσομοίωση.
- τα αρχεία **.b4** τα οποία έχουν ήδη περάσει την διαδικασία σύνδεσης (link)
- τις συνδέσεις μεταξύ των διεργασιών
- την τοποθέτηση των διεργασιών σε συγκεκριμένους επεξεργαστές.

Το Παράρτημα Z περιέχει τα αντίστοιχα αρχεία.

Β. Εκτέλεση των προσομοιώσεων

Το πρώτο βήμα είναι η μεταγλώττιση του πηγαίου κώδικα με την χρήση του compiler t8c. Με την μεταγλώττιση δημιουργούνται τα αντίστοιχα δυαδικά αρχεία με επέκταση "bin". Για παράδειγμα η μεταγλώττιση των αρχείων πηγαίου κώδικα στο πρόβλημα βελτιστοποίησης της συνάρτησης Rastrigin γίνεται με τις εξής εντολές:

```
C:\> t8c rastrm.c
C:\> t8c rastrw1.c
C:\> t8c rastrw2.c
```

```
C:\> t8c rastrw3.c
C:\> t8c rastrw4.c
C:\> t8c rastrw5.c
C:\> t8c rastrw6.c
C:\> t8c rastrw7.c
C:\> t8c rastrw8.c
```

Τα δυαδικά αρχεία (.bin) που έχουν δημιουργηθεί συνδέονται με την βιβλιοθήκη της Parallel C και με ειδικά προγράμματα σύνδεσης μέσω των linker t8ctask και t8cstask και δημιουργούνται τα αρχεία (image files) με επέκταση "b4". Μέσω του t8cstask, γίνεται σύνδεση με την *stand-alone* έκδοση της βιβλιοθήκης της Parallel C, η οποία δεν περιέχει συναρτήσεις οι οποίες απαιτούν υποστήριξη από τον afserver (I/O, date, DOS calls, κλπ.). Αφού μόνο μία διεργασία μπορεί να επικοινωνεί με τον afserver, τότε μία μόνο διεργασία μπορεί να χρησιμοποιήσει τον linker t8ctask. Οι υπόλοιπες πρέπει να χρησιμοποιήσουν τον linker t8cstask και να συνδεθούν με την *stand-alone* έκδοση της βιβλιοθήκης. Η χρήση των εντολών στο πρόβλημα βελτιστοποίησης της συνάρτησης Rastrigin γίνεται ως εξής:

```
C:\> t8ctask rastrm
C:\> t8cstask rastrw1
C:\> t8cstask rastrw2
C:\> t8cstask rastrw3
C:\> t8cstask rastrw4
C:\> t8cstask rastrw5
C:\> t8cstask rastrw6
C:\> t8cstask rastrw7
C:\> t8cstask rastrw8
```

Μετά από την δημιουργία των image files (.b4), χρησιμοποιείται ένα ειδικό πρόγραμμα (*configurer*) το οποίο δημιουργεί το τελικό εκτελέσιμο αρχείο το οποίο μπορεί να εκκινήσει την εφαρμογή. Το πρόγραμμα *configurer* χρησιμοποιεί το κατάλληλο αρχείο διευθέτησης. Χρήση:

```
C:\> config rastr8.cfg ras50.app
```

Η έξοδος του *configurer* είναι ένα αρχείο εφαρμογής το οποίο μπορεί να τοποθετηθεί στο δίκτυο των transputer και να αρχίσει την εκτέλεση χρησιμοποιώντας τον afserver.

```
C:\> afserver -:b ras50.app rastr.tsk
```

Πρίν την έναρξη του προγράμματος είναι απαραίτητος ο προγραμματισμός του link exchange της κάρτας Fast9. Εδώ χρησιμοποιούμε το αρχείο διαμόρφωσης max_con9.cfg (Παράρτημα Z):

```
C:\> f9config max_con9.cfg
```