

Ατομική Διπλωματική Εργασία

**ΣΧΗΜΑΤΙΣΜΟΣ ΛΕΙΤΟΥΡΓΙΚΩΝ ΟΜΑΔΩΝ ΕΡΓΑΣΙΑΣ  
ΓΙΑ ΤΗΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΗΣ ΠΑΡΑΓΩΓΗΣ ΕΡΓΩΝ ΛΟΓΙΣΜΙΚΟΥ**

**ΔΩΡΟΣ ΓΕΩΡΓΙΟΥ**

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**



**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Μάιος 2009**

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΣΧΗΜΑΤΙΣΜΟΣ ΛΕΙΤΟΥΡΓΙΚΩΝ ΟΜΑΔΩΝ ΕΡΓΑΣΙΑΣ  
ΓΙΑ ΤΗΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΗΣ ΠΑΡΑΓΩΓΗΣ ΕΡΓΩΝ ΛΟΓΙΣΜΙΚΟΥ**

**ΔΩΡΟΣ ΓΕΩΡΓΙΟΥ**

Επιβλέπων Καθηγητής  
ΑΝΔΡΕΑΣ ΑΝΔΡΕΟΥ

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2009

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Επίκουρο Καθηγητή Δρ. Ανδρέα Ανδρέου για την καθοδήγηση που μου προσέφερε καθόλη τη διάρκεια της υλοποίησης και συγγραφής της ατομικής μου διπλωματικής εργασίας, καθώς επίσης και τον βοηθό του Κωνσταντίνο Στυλιανού. Ακόμη, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους μου για την πολύτιμη ψυχολογική, ηθική υποστήριξη, κατανόηση και όποια άλλη βοήθεια μου προσέφεραν κατά τη διάρκεια της φοίτησης μου στο Πανεπιστήμιο Κύπρου.

## Περίληψη

Η συγκεκριμένη εργασία ασχολείται με ένα πολύ σημαντικό πρόβλημα της περιοχής της Τεχνολογίας Λογισμικού, το ανεξήγητα ακόμα αυξημένο ποσοστό αποτυχίας των έργων κατά την αύξηση πολυπλοκότητας και μεγέθους των έργων αυτών. Προτείνει σαν πρόβλημα την κακή διαχείριση των ανθρώπινων πόρων και γενικότερα των έργων και στοχεύει στον σχηματισμό λειτουργικών ομάδων εργασίας για την διεκπεραίωση ενός έργου παραγωγής λογισμικού με βέλτιστη εκμετάλλευση των πόρων ή/και ελαχιστοποίηση του χρόνου διάρκειας του έργου. Περιγράφει τις μεθοδολογίες και την προτεινόμενη χρήση τους από τους εισηγητές αυτών. Οι μεθοδολογίες αυτές είναι το ψυχομετρικό ερωτηματολόγιο MBTI (Myers-Briggs Type Indicator) που καθορίζει τύπους εργαζομένων, ο τροχός επικοινωνίας του Thompson που με βάση το MBTI δίνει ένα μετρήσιμο μέγεθος δυσκολίας επικοινωνίας δύο ανθρώπων και οι Γενετικοί Αλγόριθμοι ένα σύστημα τεχνητής νοημοσύνης για επίλυση προβλημάτων αναζήτησης και βελτιστοποίησης. Στην συνέχεια περιγράφεται πως οι συγκεκριμένες μεθοδολογίες χρησιμοποιήθηκαν για την επίλυση του αρχικού προβλήματος.

## Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Το πρόβλημα και η εργασία που επιτελέστηκε	1
	1.2 Το σύστημα και οι μεθοδολογίες που επιλέχθηκαν	3
Κεφάλαιο 2	Ψυχολογικό Προφίλ Και Ψυχολογία Της Ομάδας .....	7
	2.1 Το MBTI-Τυποποίηση Προφίλ	7
	2.2 Ο κύκλος επικοινωνίας του Thompson	14
Κεφάλαιο 3	Γενετικοί Αλγόριθμοι.....	18
	3.1 Βιολογικό υπόβαθρο και ιστορικά δεδομένα	18
	3.2 Χαρακτηριστικά και Στοιχεία των Γενετικών Αλγορίθμων	20
Κεφάλαιο 4	Μοντελοποίηση Προβλήματος Και Υλοποίηση Πρωτοτύπου .....	28
	4.1 Το MBTI-Τυποποίηση Προφίλ	28
	4.2 Υλοποίηση του πρωτοτύπου	38
	4.3 Αποτελέσματα	44
Κεφάλαιο 5	Συμπεράσματα .....	52
Βιβλιογραφία.....		55
Παράρτημα Α.....		A-1

## Λίστα Εικόνων, Πινάκων και Εξισώσεων

Κεφάλαιο 1	Εισαγωγή.....	1
Κεφάλαιο 2	Ψυχολογικό Προφίλ Και Ψυχολογία Της Ομάδας .....	7
	Figure 2Error! No text of specified style in document.-1 Ο πίνακας με τους τύπους και με χαρακτηριστικές φράσεις.....	13
	Figure 2-2 Ο κύκλος του Thompson.....	16
Κεφάλαιο 3	Γενετικοί Αλγόριθμοι.....	18
	Figure 3 -1 Ένα χρωμόσωμα.....	19
	Πίνακας 3-1.....	25
	Figure 3-2 Εξαναγκασμένη ρουλέτα.....	26
	Πίνακας 3-2.....	27
Κεφάλαιο 4	Μοντελοποίηση Προβλήματος Και Υλοποίηση Πρωτοτύπου .....	28
	Πίνακας 4-1.....	31
	Πίνακας 4-2.....	32
	Equation 4-1.....	33
	Πίνακας 4-3.....	34
	Πίνακας 4-4.....	34
	Πίνακας 4-5.....	35
	Πίνακας 4-6.....	37
	Figure4-1 Οθόνη καλωσορίσματος.....	39
	Figure 4-2 Βήμα ένα: Εισαγωγή Δεδομένων.....	40
	Figure 4-3 Βήμα 2: Παρουσίαση Εισαχθέντων Δεδομένων.....	41
	Figure 4-4 Βήμα 3: Παρουσίαση αποτελεσμάτων.....	42

Πίνακας 4-7.....	45
Πίνακας 4-8.....	45
Figure 4-5 Αποτέλεσμα πρώτου σεναρίου.....	46
Πίνακας 4-9.....	47
Πίνακας 4-10.....	47
Figure 4-6 Αποτελέσματα Δεύτερου σεναρίου.....	48
Πίνακας 4-11.....	49
Figures 4-7 Υπάλληλοι του τρίτου σεναρίου.....	50
Figure 4-8 Αποτελέσματα τρίτου σεναρίου 1.....	50
Figure 4-9 Αποτελέσματα τρίτου σεναρίου 2.....	51
Κεφάλαιο 5 Συμπεράσματα .....	52
Βιβλιογραφία.....	55
Παράρτημα Α.....	A-1

# Κεφάλαιο 1

## Εισαγωγή

---

1.1 Το πρόβλημα και η εργασία που επιτελέστηκε	1
1.2 Το σύστημα και οι μεθοδολογίες που επιλέχθηκαν	3

---

### 1.1 Το πρόβλημα και η εργασία που επιτελέστηκε

Η ατομική διπλωματική εργασία ασχολείται με ένα θέμα το οποίο αν και επικεντρώνεται στον χώρο της Πληροφορικής στην δική μου περίπτωση, και πιο συγκεκριμένα στον χώρο της Τεχνολογίας Λογισμικού, είναι ένα θέμα το οποίο πιστεύω με βεβαιότητα πως απασχολεί και θα συνεχίσει να απασχολεί διευθύνοντες έργων. Με το αυξανόμενο μέγεθος των έργων καθώς και με την πολυπλοκότητα να αυξάνεται παράλληλα οι έρευνες δείχνουν ότι τα ποσοστά αποτυχίας των έργων μεγαλώνουν. Μια πιθανή εξήγηση του πιο πάνω προβλήματος είναι ότι το πρόβλημα οφείλεται στις ομάδες οι οποίες ασχολήθηκαν με την ανάπτυξη των έργων αυτών. Όμως πέρα από τις τεχνικές ικανότητες που περιγράφουν μια ομάδα είναι και τα άτομα που την σχηματίζουν. Ένα πρόσωπο δεν μπορεί να αντιμετωπίζεται σαν ένα μηχάνημα παραγωγής εφόσον έχει την δική του προσωπικότητα η οποία επηρεάζει άμεσα στην επικοινωνία του με άλλους και εν τέλει στον ρόλο του για την ανάπτυξη του έργου. Οι διευθύνοντες έργων σε μερικές περιπτώσεις αγνοούν όμως την προσωπικότητα.



Ο ρόλος ενός διευθύνοντα έργου σε ένα έργο παραγωγής λογισμικού είναι να επιβλέπει το έργο τόσο από πλευράς των υπαλλήλων όσο και από πλευράς παραγωγής κώδικα (ποιότητα και ρυθμός παραγωγής). Η ευθύνη που στοχεύει όμως να βοηθήσει το σύστημα μου είναι αυτή του στρατηγικού σχεδιασμού ανάπτυξης του έργου. Το πώς δηλαδή ανατίθενται οι εργασίες και με ποιο τρόπο βελτιστοποιείται το σχέδιο αυτό ώστε να έχουμε την μικρότερη δυνατή ανάπτυξη και την βέλτιστη εκμετάλλευση των πόρων. Είναι υπό την ευθύνη δηλαδή του διευθύνοντα έργου οι ομάδες να είναι συνεκτικές, λειτουργικές και συνάμα παραγωγικές. Λέγοντας συνεκτικές και λειτουργικές εννοείται με βάση την προσωπικότητα του κάθε ατόμου και με τον όρο παραγωγικές βάση των ικανοτήτων που έχει ο κάθε υπάλληλος. Να σημειωθεί ότι εν τέλει στην υλοποίηση, στην προσπάθεια διατήρησης αφαιρετικότητας του συστήματος μου, το οποίο αποτελεί ένα κομμάτι ενός μεγαλύτερου συστήματος που θα υλοποιηθεί σε μεταγενέστερο στάδιο, επετεύχθη να προστεθεί η δυνατότητα επίλυσης του συγκεκριμένου προβλήματος για έργα οποιουδήποτε τομέα. Ο σκοπός λοιπόν, της εργασίας, με τον οποίο ασχολήθηκα είναι ο σχηματισμός λειτουργικών ομάδων εργασίας για την διεκπεραίωση ενός έργου παραγωγής λογισμικού με βέλτιστη υλοποίηση των πόρων ή/και ελαχιστοποίηση του χρόνου διάρκειας του έργου.

Η αποπεράτωση της ατομικής διπλωματικής μου εργασίας εκτελέστηκε σε τρεις φάσεις. Η πρώτη φάση περιελάμβανε την αναγνώριση και περαιτέρω μελέτη της περιοχής του προβλήματος καθώς και της εύρεσης του τρόπου επίλυσης του προβλήματος. Σε αυτή την φάση μελετήθηκαν διάφορες μεθοδολογίες και αλγόριθμοι εν σχέση με την Διαχείριση Έργων, Τεχνολογία Λογισμικού, Ψυχολογία-Κοινωνιολογία, Ευφυή Συστήματα. Οι μεθοδολογίες αυτές αξιολογήθηκαν έτσι ώστε να εξακριβωθούν οι πιο κατάλληλες βάσει κριτηρίων που τέθηκαν με την βοήθεια και καθοδήγηση του επιβλέποντα καθηγητή. Η δουλειά αυτή έγινε κυρίως κατά την διάρκεια της περιόδου μεταξύ Σεπτεμβρίου και Δεκεμβρίου 2008 μελετώντας διάφορα επιστημονικά άρθρα και βιβλία.

Στη δεύτερη φάση της διπλωματικής εργασίας έγινε εκλέπτυνση του προβλήματος εμβαθύνοντας παράλληλα τις γνώσεις τόσο στο πρόβλημα όσο και στις τεχνικές που χρησιμοποιήθηκαν για την επίλυση του. Ακολούθως έγινε η σχεδίαση της λύσης

(μοντελοποίηση) χρησιμοποιώντας τις μεθοδολογίες αυτές. Ακολούθησαν αριθμός συναντήσεων με τον διδακτορικό φοιτητή που θα υλοποιήσει το υπερσύστημα , για να επιτευχθεί συμφωνία του ρόλου του δικού μου συστήματος. Έτσι αφού παρουσιάστηκαν αριθμός διαφορετικών προτάσεων μοντελοποίησης του προβλήματος στον επιβλέποντα καθηγητή καταλήξαμε στην τελική μορφή του συστήματος.

Στην τελευταία και ίσως την πιο ουσιαστική φάση της εργασίας που επιτελέστηκε για την αποπεράτωση της ατομικής διπλωματικής μου εργασίας έπρεπε να ασχοληθώ με την υλοποίηση του προαναφερόμενου συστήματος αλλά και στη συγγραφή της παρακείμενης εργασίας. Σε αυτή την φάση επιλέχθηκε η γλώσσα προγραμματισμού και το περιβάλλον εργασίας με τα οποία εκτελέστηκε η υλοποίηση. Και συγκεντρώνοντας τις πηγές έγινε η εκκίνηση της συγγραφής της ατομικής διπλωματικής εργασίας.

## **1.2 Το σύστημα και οι μεθοδολογίες που επιλέχθηκαν**

Το σύστημα που αναπτύχθηκε μπορεί να κατηγοριοποιηθεί ως ένα σύστημα υποστήριξης διευθυντικών αποφάσεων. Τα συστήματα αυτά μπορούν να οριστούν ως διαδραστικά συστήματα υπολογιστών τα οποία βοηθούν τους χρήστες να χρησιμοποιήσουν τα δεδομένα τους, επικοινωνίες, αρχεία ή οποιαδήποτε άλλου είδους πληροφορία για να επιλύσουν προβλήματα και να λάβουν αποφάσεις. Είναι μόνο βοηθητικά συστήματα και δεν έχουν σκοπό να αντικαταστήσουν αυτούς που παίρνουν τις αποφάσεις αλλά να τον υποστηρίξουν. Το σύστημα ανήκει στη κατηγορία του μοντέλου βελτιστοποίησης αφού προσπαθεί να βρει την καλύτερη λύση για το συγκεκριμένο πρόβλημα δεδομένων κάποιων περιορισμών και είναι βασισμένο σε τεχνολογίες Διαδικτύου. Συγκεκριμένα για να λειτουργήσει θα χρειάζεται μόνο ένα φυλλομετρητή ιστού και θα μπορεί να τρέχει είτε μέσω Διαδίκτυου είτε σε κάποιο ενδοδίκτυο (intranet). Όλοι οι υπολογισμοί θα γίνονται στον εξυπηρετητή αφήνοντας μόνο την παρουσίαση των αποτελεσμάτων στο κάθε πελάτη(χρήστη).

Το ερώτημα είναι φυσιολογικά τι διαφορετικό προσφέρει το δικό μου σύστημα από τα μέχρι τώρα υλοποιημένα συστήματα που προϋπήρχαν του δικού μου. Όπως προανέφερα λοιπόν το πρόβλημα που κλήθηκα να λύσω είναι ο σχηματισμός λειτουργικών ομάδων εργασίας για την διεκπεραίωση ενός έργου παραγωγής λογισμικού με βέλτιστη υλοποίηση των πόρων ή/και ελαχιστοποίηση του χρόνου διάρκειας του έργου.

Πολλά συστήματα έχουν ασχοληθεί και ασχολούνται με το γνωστό πρόβλημα του “job-shop” , ανάθεσης δηλαδή του έμψυχου υλικού , των υπαλλήλων για όσους αγνοούν τον όρο , σε εργασίες . Είναι ένα NP-hard πρόβλημα με το οποίο ασχολήθηκαν πολλοί ερευνητές της περιοχής επίλυσης προβλημάτων με ευφυή συστήματα ή με άλλους τρόπους, χρησιμοποιώντας διαφορετικές λύσεις. Σε αυτό το κομμάτι κλήθηκα να χρησιμοποιήσω το πολύ δυνατό «εργαλείο» των γενετικών αλγόριθμων, στο οποίο θα αναφερθώ και θα περιγράψω εκτενέστερα σε μετέπειτα κεφάλαιο επειδή είναι ένα παράλληλος αλγόριθμος αναζήτησης επειδή δίνουν την δυνατότητα επίλυσης NP προβλημάτων σε ορισμένες περιπτώσεις σχετικά γρήγορα.

Μέσα σε αυτό το πλαίσιο κατάφερα να εντάξω και τους περιορισμούς τόσο της βέλτιστης υλοποίησης των πόρων , δηλαδή των υπαλλήλων, αλλά και της ελαχιστοποίησης του χρόνου διάρκειας του έργου. Αυτό το πέτυχα κωδικοποιώντας τους δύο αυτούς περιορισμούς μέσα στο γενετικό αλγόριθμο. Με τον όρο βέλτιστη υλοποίηση των πόρων εννοώ την καλύτερη κατανομή των υπαλλήλων στις εργασίες ενός έργου έτσι ώστε να γίνετε όσο το δυνατό καλύτερη κατανομή του φόρτου εργασίας. Ενώ με τον όρο ελαχιστοποίηση του χρόνου διάρκειας του έργου εννοώ μείωση της διάρκειας ζωής του έργου λαμβάνοντας όμως υπόψη τις αλληλεξαρτήσεις που πιθανόν να προκύπτουν μεταξύ των εργασιών ενός έργου και έχοντας όμως σαν δεδομένο την διάρκεια της κάθε εργασίας σε ανθρωπομέρες. Ανθρωπομέρα ορίζεται ως οχτώ ώρες εργασίας ενός ατόμου. Έτσι αν πούμε ότι μια συγκεκριμένη εργασία χρειάζεται δύο ανθρωπομέρες για να έρθει εις πέρας , ουσιαστικά μπορούμε να πούμε ότι αν αναθέσουμε την εργασία σε δύο υπαλλήλους τότε η διάρκεια της μειώνεται σε μία μέρα.

Η καινοτομία που στοχεύει το υλοποιημένο σύστημα είναι η λειτουργικότητα των ομάδων. Ο παράγοντας άνθρωπος είναι ένας πολύ σημαντικός παράγοντας τον οποίο μερικές φορές οι διευθύνοντες έργων τείνουν να το αγνοούν. Ειδικά σε τομείς όπως η Πληροφορική όπου η παραγωγή έργου είναι εξ' ολοκλήρου ανθρωποκεντρική, ο παράγοντας άνθρωπος δεν μπορεί και δεν πρέπει να αγνοείται. Σε πολλές περιπτώσεις έργα Πληροφορικής απέτυχαν λόγω των υπαλλήλων που εργάζονταν σε αυτά. Αυτό οφειλόταν είτε λόγω προβλημάτων σε διαπροσωπικές σχέσεις είτε σε εγκατάλειψη της προσπάθειας από τους συμμετέχοντες στο έργο αυτό. Η επιλογή των ομάδων είναι συνήθως ευθύνη του εκάστοτε ηγέτη του έργου και στην κρίση του είναι το αν αυτή η ομάδα θα είναι λειτουργική ή όχι.

Η συνεκτικότητα της ομάδας είναι ένα θέμα που τα τελευταία χρόνια αναπτύσσεται ραγδαία και η κατανόηση της συμπεριφοράς των ατόμων μέσα σε μια ομάδα αυξάνεται. Οι διευθύνοντες έργων πλέον δεν προσπαθούν να σχηματίσουν μια ομάδα ατόμων που ίσως να τους διεκπεραιώσει μια συγκεκριμένη εργασία. Θέλουν μια λειτουργική ομάδα που θα την φέρει εις πέρας άριστα, χωρίς, αν αυτό είναι ποτέ δυνατό, ρήξεις στις διαπροσωπικές σχέσεις τους. Εμφανίζονται συνεχώς νέα μοντέλα τυποποίησης των ομάδων και ειδικότερα των ατόμων. Έτσι μπορούμε να συμπεράνουμε με βεβαιότητα ότι μια τέτοια μεθοδολογία μπορεί να βοηθήσει πολύ ένα διευθύνοντα έργου. Ιδανικά μια τέτοια μεθοδολογία θα πρέπει να με ένα απλό ερωτηματολόγιο να σχηματίζει κάποιο ψυχολογικό προφίλ του υπαλλήλου και έχοντας πλέον το προφίλ να δίνει την δυνατότητα να αποφασίσει ο χρήστης αυτού, με ποια άλλα προφίλ μπορεί ο συγκεκριμένος υπάλληλος να εργαστεί και γενικότερα να συνυπάρξει. Στην δική μου εργασία χρησιμοποίησα τον κύκλο επικοινωνίας του Thompson στον οποία αναφέρομαι εκτενέστερα στο επόμενο κεφάλαιο.

Τα επόμενα κεφάλαια θα ασχοληθούν με την εκτενέστερη ανάλυση των μεθοδολογιών που χρησιμοποίησα. Πιο συγκεκριμένα στο δεύτερο κεφάλαιο την περιγραφή του MBTI και του Thompson's Communication Wheel ενώ στο τρίτο την λειτουργία των γενετικών αλγορίθμων. Μετέπειτα στο τέταρτο κεφάλαιο θα περιγράφει πώς χρησιμοποίησα τις συγκεκριμένες μεθοδολογίες για να δημιουργήσω τον δικό μου γενετικό αλγόριθμο που επιλύει το ήδη αναφερόμενο πρόβλημα. Έχοντας παρουσιάσει αυτά θα δείξω επίσης και το πρωτότυπο που υλοποίησα το οποίο χρησιμοποιεί τον αλγόριθμο μου καθώς και κάποια

αποτελέσματα που προέκυψαν από δοκιμές του πρωτότυπου. Στο τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα και η μελλοντική δουλειά που στοχεύετε να γίνει.

# Κεφάλαιο 2

## Ψυχολογικό Προφίλ Και Ψυχολογία Της Ομάδας

---

2.1 Το MBTI-Τυποποίηση Προφίλ	7
2.2 Ο κύκλος επικοινωνίας του Thompson	14

---

### 2.1 Το MBTI-Τυποποίηση Προφίλ

Το Myers-Briggs Type Indicator[3] είναι ένα ψυχομετρικό ερωτηματολόγιο , το οποίο σχεδιάστηκε για να μετρά τις ψυχολογικές προτιμήσεις των ανθρώπων ως προς το πως αντιλαμβάνονται τον κόσμο και πως παίρνουν αποφάσεις. Η θεωρία αυτή είχε ως βάση ένα βιβλίο , το οποίο έγραψε ένας από τους θεωρούμενους πατέρες της ψυχολογίας ο Carl Gustav Jung, το οποίο ονομάζεται «Ψυχολογικοί Τύποι», αν και εν τέλει υπάρχουν αρκετές διαφορές μεταξύ των δύο θεωριών. Αρχικοί σχεδιαστές του Myers-Briggs Type Indicator είναι οι Katharine Cook Briggs και η κόρη της , Isabel Briggs Myers . Η ιδέα ξεκίνησε κατά την διάρκεια του Β' Παγκοσμίου Πολέμου για βοήθεια των νεοεισερχόμενων τότε γυναικών στην αμερικανική βιομηχανία, ώστε να μπορέσουν πιο εύκολα να κατευθυνθούν σε κάποιο τομέα. Από αυτή την πολύ αρχική μορφή αναπτύχθηκε στο Myers-Briggs Type Indicator το οποίο πρωτοεκδόθηκε το 1962.

Υπάρχουν 16 ψυχολογικοί τύποι σύμφωνα με το MBTI οι οποίοι παράγονται με δύο επιλογές τεσσάρων τύπων. Οι τέσσερις τύποι που χρησιμοποιούνται χωρίζονται σε τρεις ομάδες. Η πρώτη ομάδα ασχολείται με τη συμπεριφορά – κατεύθυνση ενέργειας και

προσοχής, η δεύτερη με τις λειτουργίες – διαδικασία αντίληψης και κρίσης και η Τρίτη με την αντιμετώπιση ζωής – προσανατολισμός προς τον εξωτερικό κόσμο.

Πρώτη ομάδα (Συμπεριφορά – Κατεύθυνση ενέργειας και προσοχής):

Εσωστρεφής ή Εξωστρεφής (Introvert ή Extrovert): Ένας άνθρωπος μπορεί να αντιλαμβάνεται διαφορετικά πληροφορίες ως προς το πως τις επεξεργάζεται. Αν ο άνθρωπος αυτός είναι Introvert τότε προτιμούν να αντιλαμβάνονται πράγματα με το να τα σκεφτούν ενώ ένας Extrovert θα προτιμήσει να συζητήσει για κάποιο θέμα ώστε να το αντιληφθεί. Γενικά οι Extrovert είναι πιο κοινωνικοί και πιο ομιλητικοί και μοιράζονται εύκολα τις ιδέες τους ενώ οι Introvert είναι πιο κλειστοί χαρακτήρες και μοιράζονται τις ιδέες τους μόνο αφού τις τελειοποιήσουν στο μυαλό τους. Οι Extrovert ταυτίζονται με το δυναμισμό ενώ οι Introvert είναι πιο αμυντικοί.

Δεύτερη ομάδα(Λειτουργίες – Διαδικασία αντίληψης και κρίσης):

Αίσθηση ή Διαίσθηση (Sensing ή iNtuiting): Η συγκεκριμένη μεθοδολογία χωρίζει το πως αντιλαμβάνονται ή καλύτερα πως δέχονται τις νέες πληροφορίες τα άτομα. Όσα άτομα προτιμούν το Sensing αρέσκονται στο να μαθαίνουν τα γεγονότα και τις λεπτομέρειες ενώ στα iNtuiting άτομα αρέσει να ακούν την γενική ιδέα. Γενικά στα Sensing άτομα αρέσουν οι δοκιμασμένες μέθοδοι, και οι απτές αποδείξεις, θέλουν να λαμβάνουν τα αναμενόμενα αποτελέσματα και να αναπαράγουν την γνώση τους. Από την άλλη πλευρά όσοι είναι iNtuiting θέλουν να δοκιμάζουν καινούριες ιδέες, τους αρέσουν οι αφηρημένες ιδέες και οι θεωρίες, θέλουν αλλαγή και τους αρέσει να δουλεύουν με βάση την έμπνευση.

Σκεπτικισμός ή Συναίσθημα (Thinking ή Feeling): Ο τύπος αυτός αναφέρεται στο πως λαμβάνουν αποφάσεις τα άτομα. Αν κάποιος προτιμά να λαμβάνει αποφάσεις με την κατηγορία Thinking τότε θα προσπαθήσει να έχει μια διαύγεια που αντλείτε από την λογική, θα ρωτήσει πριν αποφασίσει, θα ενδιαφερθεί για τα δεδομένα, θα προσπαθήσει να είναι αντικειμενικός και θα ζυγίσει τα αρνητικά και τα θετικά της απόφασης του παραμένοντας αμερόληπτος. Προτιμούν την δικαιοσύνη και ξέρουν πότε πρέπει να αντιδράσουν λογικά. Στην άλλη κατηγορία Feeling τα άτομα λαμβάνουν αποφάσεις προσπαθώντας να έχουν συναισθηματική διαύγεια. Δέχονται πιο εύκολα καταστάσεις,

Ενδιαφέρονται για τους ανθρώπους και ξέρουν πότε να τους υποστηρίξουν. Προτιμούν τα πράγματα να είναι σε προσωπικό επίπεδο. Λαμβάνουν αποφάσεις έχοντας προσωπική θέση την οποία παίρνουν αφού ζύγισαν τις προσωπικές τους αξίες. Προσπαθούν να είναι συμπονετικοί.

Όλοι οι άνθρωποι χρησιμοποιούν και τα τέσσερα χαρακτηριστικά αυτής της ομάδας. Όμως δεν έχουν την ίδια προτεραιότητα αφού υπάρχει κάποιο κυρίαρχο χαρακτηριστικό ενώ τα άλλα ακολουθούν. Αυτό γίνεται πάντα συναρτήσει και του χαρακτηριστικού της πρώτης ομάδας.

### Τρίτη Ομάδα (Αντιμετώπιση Ζωής – Προσανατολισμός προς τον εξωτερικό κόσμο):

Κρίση ή Παρατήρηση (Judging ή Perceiving): Αυτός ο τύπος είναι μια προσθήκη στην αρχική θεωρία του Jung από τους Myers-Briggs. Αναφέρεται στο πώς τα άτομα συσχετίζονται με τον υπόλοιπο κόσμο. Γενικά τα άτομα που προτιμούν το Judging δεν τους αρέσει να αφήνουν θέματα σε εκκρεμή κατάσταση. Θέλουν να διευθετούν τα θέματα τους, να τελειώνουν πριν τις προθεσμίες. Θέλουν να πετυχαίνουν τους στόχους τους και να έχουν καλά αποτελέσματα χωρίς εκπλήξεις. Θέλουν τα πάντα σχεδιασμένα, και τους αρέσει να ακολουθούν ρουτίνες. Η κατηγορία των Perceiving ατόμων είναι πιο προσαρμόσιμοι σε νέες καταστάσεις. Τους αρέσει να είναι «ευλύγιστοι», δεν έχουν πρόβλημα να αφήνουν πράγματα μέχρι την τελευταία στιγμή που πρέπει να γίνουν και αντιμετωπίζουν καταστάσεις με εκπλήξεις και νέες καταστάσεις που μπορεί να παρουσιαστούν. Αρέσκονται να αλλάζουν τα σχέδια δυναμικά ανάλογα με την κατάσταση.

Ο τύπος ενός ατόμου είναι μια συνάθροιση των τεσσάρων πιο πάνω κατηγοριών αλλά με κάποια από τα συγκεκριμένα χαρακτηριστικά να υπερισχύουν κάθε φορά κάνοντας έτσι αναγκαία την περιγραφή του κάθε τύπου ξεχωριστά. Για την κωδικοποίηση των τύπων χρησιμοποιούνται τα εξής γράμματα:

E για Extrovert ή I για Introvert,

S για Sensing ή N για iNtuiting,



T για Thinking ή F για Feeling,  
J για Judging ή P για Perceiving.

Οι 16 Τύποι:

ISTJ- Εξάρτηση: Σχολαστικοί, δουλεύουν σκληρά και είναι υπεύθυνοι. Μπορούν να δουλέψουν καλά σε ρουτίνα με καθορισμένες διαδικασίες κρατώντας ημερολόγιο των πράξεων τους. Τους αρέσει να ξεκαθαρίζουν τον ρόλο τους και τις ευθυνότητές τους και προσπαθούν να διατηρούν την αποδοτικότητα τους και την χρησιμότητα τους και να ακολουθούν τις δεσμεύσεις τους.

ISFJ- Δέσμευση: Φιλόντιμοι, πιστοί, και αφοσιωμένοι. Δουλεύουν καλά όταν έχουν ξεκαθαρισμένο ρόλο και ευθύνη. Προσέχουν τις ειδικές ανάγκες των ανθρώπων και στηρίζονται σε καθιερωμένες μεθόδους. Ψάχνουν την σταθερότητα και την αρμονία και περιμένουν από τους άλλους να είναι επιμελής. Μπορούν να κάνουν θυσίες για να αποπερατώσουν μια δουλειά.

INFJ- Ακεραιότητα: Διορατικοί, εμπνευστικοί και δημιουργικοί. Δουλεύουν καλά όταν ο στόχος είναι το ενδιαφέρον των ανθρώπων, ασκούν ήσυχα επιρροή και είναι πρότυπα ακεραιότητας. Οραματίζονται τρόπους να πετύχουν μακροπρόθεσμους στόχους, αρέσκονται να δημιουργούν ατμόσφαιρα αμοιβαίας εμπιστοσύνης και να οργανώνουν άτομα και διεργασίες προς όφελος όλων.

INTJ- Διορατικότητα: Ανεξάρτητοι, ατομιστές και διορατικοί. Δουλεύουν καλά όταν μπορούν να αναπτύξουν στρατηγικές, να χρησιμοποιήσουν πρόβλεψη, να υλοποιήσουν τις ιδέες τους και να δημιουργήσουν έξυπνα σχήματα για να φτάσουν προθεσμίες. Δεν τους φοβίζουν οι δυσκολίες, αναλύουν αντικειμενικά παράγοντες και θέματα για να αντιμετωπίσουν πολύπλοκες προκλήσεις που βλέπουν στο μέλλον.

ISTP- Ιδιοφυΐα: Πραγματιστές, ρεαλιστές και προσαρμόσιμοι. Δουλεύουν καλά όταν δεν υπάρχουν πολλοί κανόνες και όταν μπορούν να βάλουν σε εφαρμογή τις ικανότητές τους με ένα πρακτικό και λογικό τρόπο. Μπορούν να διακρίνουν ασυνέπειες σε μεθοδολογίες

και διαδικασίες, αντιδρούν γρήγορα σε επείγοντες καταστάσεις και επινοούν ευθυτενή σχέδια για να αντιμετωπίσουν προβλήματα που μπορεί να συναντήσουν.

ISFP- Ευαισθησία: Χαμηλών τόνων, ελαστικοί, ενδιαφέρονται. Δουλεύουν καλά όταν αντιμετωπίζουν προσωπικές ανάγκες ατόμων με ένα ευθύ και προσωπικό τρόπο. Εκτιμούν την αρμονία και την ανεκτικότητα. Είναι γνήσιοι, ειλικρινείς και ανοικτόμυαλοι. Βελτιώνουν τον χώρο εργασίας τους αφού βεβαιώνονται ότι σε όλους φαίνονται με καλοσύνη.

INFP- Ιδεαλισμός: Αυθεντικοί, βασισμένοι σε αρχές, ενδιαφέρονται για τους άλλους. Δουλεύουν καλά όταν μπορούν να παρέχουν την ευεξία των άλλων και να βοηθούν τους άλλους να φτάσουν τις πλήρεις δυνατότητες τους. Ψάχνουν το νόημα και τον λόγο για οτιδήποτε ασχοληθούν. Προσφέρουν αυθεντικότητα και ένα πνεύμα αρμονίας, πίστης και συμπόνιας στα άτομα και στους θεσμούς που υπηρετούν.

INTP- Λογική: Αναλυτικοί, διανοούμενοι, και ιδιοφυείς. Δουλεύουν καλά όταν μπορούν να λειτουργήσουν ανεξάρτητα, όταν ψάχνουν για την αλήθεια και όταν χρησιμοποιούν προσεγγίσεις βασισμένες στη λογική για να λύσουν πολύπλοκα προβλήματα. Η περιέργεια τους, τους οδηγεί να αναζητήσουν θεωρίες, να παρατηρήσουν το λόγο που κάνουν τα πράγματα να δουλεύουν και να ανακαλύψουν τις μακροπρόθεσμες συνέπειες ενός δεδομένου σχεδίου ή στρατηγικής.

ESTP- Ενέργεια: Οδηγούνται από την δράση, ενεργητικοί, ρεαλιστικοί. Δουλεύουν καλά όταν είναι πλήρως αναμειγμένοι, τους αρέσουν οι προκλήσεις και όταν προσπαθούν να εξαλείψουν εμπόδια μέσα από μία λογική, πραγματική, ουσιώδη προσέγγιση. Τους αρέσει να λύνουν προβλήματα, να δουλεύουν για να πετύχουν άμεσα αποτελέσματα και να είναι εκεί που είναι η δράση.

ESFP- Ενθουσιασμός: Φιλικοί, ευκοινωνήτοι και ενθουσιώδεις. Δουλεύουν καλά όταν μπορούν να χρησιμοποιήσουν την ζωντάνια τους και το χιούμορ τους για να κάνουν κάποια δουλειά. Κάνουν συλλογικές προσπάθειες διασκεδαστικές με το να εφαρμόζουν

κοινή λογική και την αυθόρμητη προσέγγιση τους για να αντιμετωπίσουν προκλήσεις. Τους αρέσει να χρησιμοποιούν την ζεστασιά και την γενναιοδωρία τους για να βοηθήσουν ανθρώπους.

ENFP- Φαντασία: Ζωντανοί, χαρισματικοί και ενθαρρυντικοί. Δουλεύουν καλά όταν μπορούν να καινοτομήσουν και να είναι δημιουργικοί, όταν μπορούν να πείσουν τους άλλους να δράσουν και να στιγματίσουν θετική αλλαγή. Παράγουν ενθουσιασμό για νέες δραστηριότητες, είναι ακούραστοι στην ενασχόληση καινούργιων ενδιαφερόντων και προβλέπουν τις ανάγκες ανθρώπων και οργανισμών.

ENTP- Πρωτοβουλία: Οξυδερκείς, προσαρμόσιμοι και έξυπνοι. Δουλεύουν καλά όταν μπορούν να αναλάβουν προκλήσεις, να αυτοσχεδιάσουν αφηρημένα πλαίσια και να ενθαρρύνει άλλους να καταλάβουν στόχους στρατηγικά. Έχουν ένα εθελοντικό πνεύμα όσο αφορά επιχειρηματικές προσπάθειες, προσαρμόζονται στις αλλαγές και στις καινοτομίες και βρίσκουν τρόπους να κάνουν κάτι που δεν είναι άμεσα φανερό.

ESTJ- Αποφασιστικότητα: Λογικοί, ρυθμιστικοί και οργανωμένοι. Δουλεύουν καλά όταν μπορούν να διοικούν και να διαχειρίζονται πόρους, όταν υλοποιούν σχέδια και όταν επιτυγχάνουν καθήκοντα. Βρίσκουν και διορθώνουν ελαττώματα, παρατηρούν γεγονότα και κρατούν τους πάντες υπόλογους. Δουλεύουν σκληρά, είναι υπεύθυνοι και ψάχνουν πρακτικές και ρεαλιστικές λύσεις σε προβλήματα.

ESFJ- Συμφιλίωση: Βοηθητικοί, ζεστοί και συνεργάσιμοι. Δουλεύουν καλά όταν εξυπηρετούν ανάγκες ανθρώπων με ένα δομημένο, έγκαιρο και πρακτικό τρόπο. Προσπαθούν να επιβεβαιώνουν ότι άνθρωποι και καθήκοντα βρίσκονται οργανωμένα σε αρμονία. Ασκούν αποφασιστικότητα και συνεχή παρακολούθηση, και δουλεύουν για να επιτύχουν αποτελέσματα που βελτιώνουν την κατάσταση για όσους ενδιαφέρονται.

ENFJ- Υπευθυνότητα: Ζεστοί, υποστηρικτικοί και φιλικοί. Δουλεύουν καλά όταν μπορούν να συγκεντρωθούν σε ανθρώπινες προσδοκίες, όταν αναπτύσσουν οργανωμένα σχέδια για να φτάσουν στόχους και διατηρούν την ακεραιότητα τους κάνοντας αυτά. Συντονίζονται με

τους άλλους, μαθαίνοντας τις ελπίδες και τα σχέδια τους, προωθούν την συνεργασία και αγωνίζονται για το κοινό καλό.

ENTJ- Οδηγός: Ενεργητικοί, σθεναροί και με αυτοπεποίθηση. Δουλεύουν καλά όταν μπορούν να διευθύνουν έργα, όταν φέρνουν τάξη και λογική στα καθήκοντα, χρησιμοποιούν καλά θεωρητικά μοντέλα για να καθοδηγήσουν πράξεις και δημιουργούν μακροπρόθεσμα σχέδια. Επινοούν σχέδια και λύσεις για να διορθώσουν προβλήματα, παίρνουν αποφασιστικές στάσεις και επωμίζονται την ευθύνη για την επίτευξη ομαδικών στόχων.

<b>ISTJ</b> "Doing What Should Be Done"	<b>ISFJ</b> "A High Sense of Duty"	<b>INFJ</b> "An Inspiration to Others"	<b>INTJ</b> "Everything Has Room For Improvement"
<b>ISTP</b> "Ready to Try Anything Once"	<b>ISFP</b> "Sees Much But Shares Little"	<b>INFP</b> "Noble Service to Aid Society"	<b>INTP</b> "A Love of Problem Solving"
<b>ESTP</b> "The Ultimate Realists"	<b>ESFP</b> "You Only Go Around Once in Life"	<b>ENFP</b> "Giving Life an Extra Squeeze"	<b>ENTP</b> "One Exciting Challenge After Another"
<b>ESTJ</b> "Life's Administrators"	<b>ESFJ</b> "Hosts and Hostesses of the World"	<b>ENFJ</b> "Smooth Talking Persuaders"	<b>ENTJ</b> "Natural Leaders"

Figure 2Error! No text of specified style in document.-1 Ο πίνακας με τους τύπους και με χαρακτηριστικές φράσεις

Κάθε υπάλληλος απαντώντας σε ένα ερωτηματολόγιο μπορεί να καταταχθεί σε ένα από τους πιο πάνω τύπους και έτσι ο διευθύνοντας μαθαίνει αμέσως μερικές πολύ σημαντικές πληροφορίες για τον υπάλληλο του. Ο διευθύνοντας όμως πρέπει να είναι προσεκτικός στο

ότι το συγκεκριμένο test δεικνύει τις προτιμήσεις του υπάλληλου και όχι τις ικανότητες και τις γνώσεις του. Επίσης δεν ξεκαθαρίζει ποια είναι η πιο ισχυρή προτίμηση απλά τις διακρίνει. Κανένας τύπος δεν είναι καλύτερος ή χειρότερος από άλλον, ο κάθε ένας έχει τα θετικά και τα αρνητικά του και η ποικιλομορφία είναι πολλές φορές το πιο θεμιτό. Το ερωτηματολόγιο πρέπει να γίνεται εθελοντικά και όχι κάτω από πίεση. Εννοείται ακόμα ότι τα αποτελέσματα πρέπει να είναι εμπιστευτικά εφόσον αυτός που απάντησε το ερωτηματολόγιο δεν θέλει να το αποκαλύψει. Τέλος δεν αποτελεί κριτήριο επιλογής αλλά μάλλον οδηγό για την καλύτερη χρήση ενός υπαλλήλου.

## **2.2 Ο κύκλος επικοινωνίας του Thompson**

Αν και το MBTI είναι ένα πολύ δυνατό εργαλείο, παραμένει ένα ερωτηματολόγιο εύρεσης του τύπου προσωπικότητας. Πέρα από τα συμπεράσματα που μπορεί να προσφέρει, δεν μπορεί να προσφέρει ένα αυστηρό και μετρήσιμο τρόπο σχηματισμού ομάδας στον διευθύνοντα. Έτσι μετά από περισσότερη έρευνα βρέθηκε ένα μοντέλο που βασίζεται στο MBTI αλλά προσφέρει ένα μετρήσιμο μέγεθος που θα μπορεί να χρησιμοποιηθεί στην ανάπτυξη της μοντελοποίησης. Η μεθοδολογία ονομάζεται κύκλος επικοινωνίας και αναπτύχθηκε προς τα τέλη της δεκαετίας του 1990 από τον Henry L. Thompson.

Η επικοινωνία είναι αναμφισβήτητα ένας από τους πιο σημαντικούς παράγοντες σε μια διαπροσωπική σχέση. Μπορεί να οριστεί ως μια κυκλική διαδικασία μεταφοράς πληροφορίας από ένα άνθρωπο σε ένα άλλο άνθρωπο ή μια ομάδα ανθρώπων. Η επικοινωνία επηρεάζεται από πάρα πολλούς παράγοντες όπως η ηλικία, το φύλο, την καταγωγή καθώς και άλλους παράγοντες. Η συγκεκριμένη μεθοδολογία βρίσκει ως ένα πολύ σημαντικό παράγοντα και τον τύπο του ατόμου σύμφωνα με το MBTI. Έγιναν και άλλες προσπάθειες[4] που να συνδυάζουν τον τύπο με την επικοινωνία που όμως απέτυχαν. Αυτό έγινε επειδή επικεντρώνονταν σε μια μόνο κατηγορία ή σε ένα συνδυασμό των δύο. Η συγκεκριμένη θεωρία όμως ασχολείται με τον συνδυασμό όλων των παραγόντων που σχηματίζουν ένα τύπο του MBTI. Πιο συγκεκριμένα, καθορίζει ότι ένα

άτομο μπορεί να μιλά μια από τέσσερις «γλώσσες», μια από τις οκτώ «διαλέκτους» με ένα από τους 16 «τρόπους».

Οι τέσσερις γλώσσες της μεθοδολογίας του τροχού επικοινωνίας είναι οι 4 κατηγορίες της δεύτερης ομάδας από τις τρεις που καθόρισα στο προηγούμενο υποκεφάλαιο του παρόντος κεφαλαίου, Sensing, iNtuiting, Thinking και Feeling. Οι τέσσερις γλώσσες τοποθετούνται στο εξωτερικό του τροχού με τις αντίθετες να τοποθετούνται απέναντι. Οι οκτώ διάλεκτοι αναφέρονται στον συνδυασμό των τεσσάρων γλωσσών με τον εξής τρόπο:

#### Sensing Διάλεκτοι:

Sensing-Thinking: Με κυρίαρχη την Sensing κατηγορία και όλα τα παρεπόμενα χρησιμοποιεί ως δεύτερο χαρακτηριστικό την κατηγορία Thinking με αποτέλεσμα μια προσγειωμένη διάλεκτο με στόχο την άμεση αντιμετώπιση του προβλήματος.

Sensing-Feeling: Έχοντας όλα τα χαρακτηριστικά του Sensing ως πρωτεύουσα γλώσσα όπως το μάζεμα των δεδομένων συνδυάζεται με τον ανθρωποκεντρισμό της κατηγορίας Feeling.

#### iNtuiting Διάλεκτοι:

iNtuiting-Thinking: Η NT διάλεκτος είναι μια διάλεκτος που γεννά ιδέες που κυλά ελεύθερα σε συνδυασμό με την ακρίβεια και την λογική ανάλυση της κατηγορίας Thinking.

iNtuiting-Feeling: Αυτή η διάλεκτος έχει γρήγορο ρυθμό παράγει ιδέες και είναι ανθρωποκεντρική λόγω της Feeling κατηγορίας.

#### Thinking Διάλεκτοι:

Thinking-Sensing: Η συγκεκριμένη διάλεκτος οδηγεί με οργάνωση, δομή και λογική και δεν σταματά αν δεν πετύχει τον σκοπό της.

Thinking-iNtuiting: Εφόσον κυρίαρχη κατηγορία είναι η Thinking και αυτή η διάλεκτος είναι πάνω από όλα λογική και διανοούμενη με βλέψεις όμως στο μέλλον και την βελτίωση καταστάσεων

### Feeling Διάλεκτοι:

Feeling- Sensing: Είναι μια ανθρωποκεντρική διάλεκτος που επικεντρώνεται στις εμπειρίες των ανθρώπων από το παρελθόν και τα γεγονότα που τους στιγμάτισαν.

Feeling- iNtuiting: Μια διάλεκτος που κινείται με γοργό ρυθμό αλλά είναι παράλληλα μια ζεστή και φιλική γλώσσα όπως και δημιουργική. Παρόλα αυτά η iNtuiting κατηγορία κατευθύνει την ομιλία στο μέλλον.

Οι διάλεκτοι τοποθετούνται στον κύκλο στον αμέσως πιο εσωτερικό δακτύλιο μετά τις γλώσσες μέσα από τη γλώσσα που είναι το κυρίαρχο τους χαρακτηριστικό, με τα αντίθετα χαρακτηριστικά σε αντίθετη θέση. Στον πιο εσωτερικό δακτύλιο μπαίνουν οι δεκαέξι τύποι του MBTI με τα ίδια χαρακτηριστικά που ανάλυσα στο προηγούμενο υποκεφάλαιο. Ο κύκλος εν τέλει μοιάζει όπως στο σχήμα 2.2.

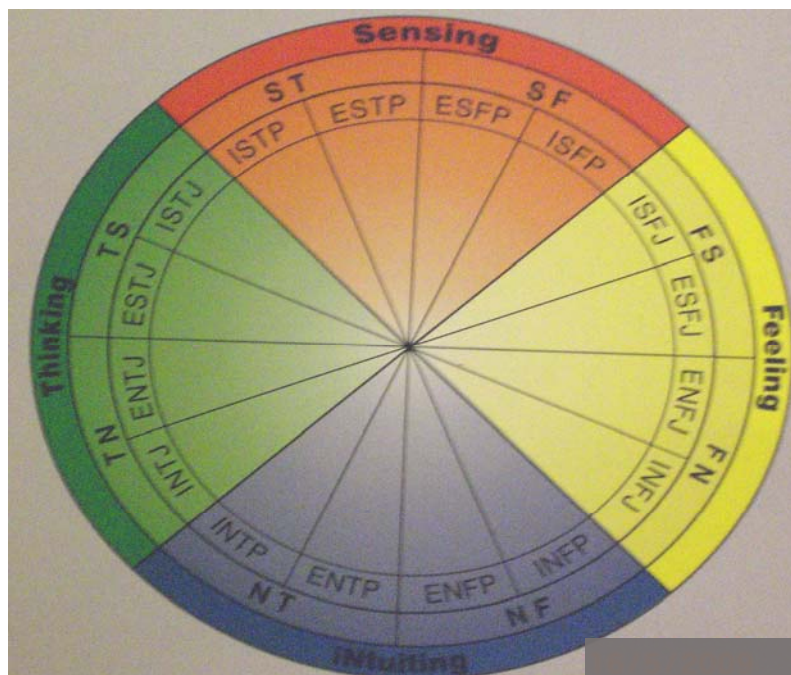


Figure 2-2 Ο κύκλος του Thompson

Αυτό όμως που προσδίδει δύναμη στη συγκεκριμένη μεθοδολογία είναι ότι κάνει τη δυσκολία της επικοινωνίας μεταξύ δύο ανθρώπων ένα μετρήσιμο μέγεθος. Αυτό επιτυγχάνεται με το να τοποθετήσουμε τα δύο άτομα πάνω στον κύκλο του Thompson στην θέση κάτω από τον τύπο τον οποίο ανήκει το κάθε άτομο και να μετρήσουμε την ελάχιστη απόσταση αμφίδρομα. Για παράδειγμα ένα άτομο με τύπο ISTJ έχει απόσταση επτά και όχι εννέα από ένα INFP αφού η μέγιστη δυνατή απόσταση είναι οχτώ εφόσον υπάρχουν συνολικά 16 τύποι. Εννοείται ότι όσο πιο μεγάλη είναι η απόσταση μεταξύ δύο τύπων τόσο μεγαλύτερο και το χάσμα επικοινωνίας μεταξύ τους.



# Κεφάλαιο 3

## Γενετικοί Αλγόριθμοι

---

3.1 Υπόβαθρο και ιστορικά δεδομένα	18
4.1 Χαρακτηριστικά και Στοιχεία των Γενετικών Αλγορίθμων	20

---

### 4.1 Υπόβαθρο και ιστορικά δεδομένα

Οι γενετικοί αλγόριθμοι[1] έχουν ως υπόβαθρο την βιολογική εξέλιξη των ειδών. Η θεωρία αυτή αναπτύχθηκε από τον Δαρβίνο στα μέσα του περασμένου αιώνα. Έγινε αποδεκτή λόγω του ότι έδωσε απαντήσεις σε ερωτήματα αιώνων, όπως το φαινόμενο της ζωής τη προέλευση της και τις βασικές λειτουργίες της. Οι γενετικοί αλγόριθμοι βασίστηκαν σε κάποιες βασικές αρχές αυτής της θεωρίας. Όλοι οι ζωντανοί οργανισμοί από τους μονοκύτταρους μέχρι τους πολυκύτταρους και πολύπλοκους οργανισμούς όπως ο άνθρωπος αναπαράγονται. Κάποιοι οργανισμοί παράγουν περισσότερους απογόνους από άλλους του ίδιου είδους με αποτέλεσμα τα κληρονομικά χαρακτηριστικά τους να είναι περισσότερα στην επόμενη γενιά. Εξωγενείς και μη παράγοντες καθορίζουν κατά πόσο ένας οργανισμός θα επιβιώσει και θα πολλαπλασιαστεί. Αλλάζοντας το περιβάλλον αλλάζουν και τα κυρίαρχα χαρακτηριστικά που είναι αναγκαία για να επιβιώσει ένας οργανισμός προσπαθώντας να προσαρμοστούν κάθε φορά με στόχο την εξασφάλιση της επιβίωσης.

Η αλλαγή αυτή στα χαρακτηριστικά των οργανισμών είναι ουσιαστικά αλλαγή στα χρωμοσώματα τους, που περιέχουν το γνωστό DNA, τα οποία είναι πολύπλοκα οργανικά μόρια τα οποία κωδικοποιούν τη δομή και τα χαρακτηριστικά τους. Οι κυρίαρχες λειτουργίες του φαινομένου της εξέλιξης είναι η αναπαραγωγή και η μετάλλαξη. Στην μετάλλαξη έχουμε με τυχαίο τρόπο αλλαγή της δομής των χρωμοσωμάτων κυρίως λόγω εξωγενών παραγόντων όπως παραδείγματος χάρη της ακτινοβολίας. Στην αναπαραγωγή έχουμε την παραγωγή ενός νέου ατόμου του οποίου τα χρωμοσώματα αποτελούνται κατά το ήμισυ από γενετικό υλικό της μητέρας και κατά το ήμισυ από γενετικό υλικό του πατέρα.



**Figure 3 -1 Ένα χρωμόσωμα**

Η έξυπνη αυτή στρατηγική βελτιστοποίησης της ζωής δεν θα μπορούσε να ξεφύγει από τα μάτια των ανθρώπων ώστε να την χρησιμοποιήσουν προς όφελος τους. Συγκεκριμένα πατέρας των γενετικών αλγορίθμων είναι ο John Holland ο οποίος συνέλαβε την συγκεκριμένη ιδέα στις αρχές της δεκαετίας του '70. Ήταν ο πρώτος που προσπάθησε να βάλει τη συγκεκριμένη φυσική διεργασία σε πλαίσια και να την μεταφέρει στους ηλεκτρονικούς υπολογιστές ως αλγόριθμους επίλυσης πολύπλοκων προβλημάτων. Το αποτέλεσμα αυτής της ιδέας ήταν οι Γενετικοί αλγόριθμοι μια υποσχόμενη τότε τεχνική αναζήτησης και βελτιστοποίησης.

Για να εξηγήσω την διεργασία που ακολουθούν οι Γενετικοί Αλγόριθμοι θα παρουσιάσω ένα ανάλογο βιολογικό παράδειγμα. Έστω ένας πληθυσμός από αντιλόπες. Κάποιες από αυτές θα είναι πιο γρήγορες από τις άλλες. Με το να είναι πιο γρήγορες έχουν περισσότερες πιθανότητες να επιβιώσουν από τα άγρια ζώα της ζούγκλας. Εφόσον συνεχίσουν να επιβιώνουν σε κάποια φάση της ζωής τους θα αναπαραχθούν. Πιθανόν από τις αντιλόπες που επιβίωσαν να είναι και κάποιες πιο αργές αντιλόπες που ήταν αρκετά τυχερές ή για κάποιο άλλο λόγο επιβίωσαν. Έτσι οι επιβιώσασες αντιλόπες θα αναμειχθούν και θα αναπαραχθούν μεταξύ τους με τυχαίο τρόπο. Όπως και να έχει λόγω του ξεκαθαρίσματος ακριβώς των αργών αντιλόπων έστω και αν μερικές επιβίωσαν, η επόμενη γενιά θα είναι γενικά γρηγορότερη. Φυσικά αυτή η διεργασία στη φύση παίρνει από εκατοντάδες μέχρι και χιλιάδες χρόνια για να υπάρξει φανερή αλλαγή στις περισσότερες περιπτώσεις. Για άλλα πιο απλά χαρακτηριστικά όπως το χρώμα τα αποτελέσματα μπορεί να είναι φανερά και μετά από μια ή δύο γενιές.

### **3.2 Χαρακτηριστικά και Στοιχεία των Γενετικών Αλγορίθμων**

Οι γενετικοί αλγόριθμοι έχουν κάποια χαρακτηριστικά τα οποία τους διαφοροποιούν από άλλους αντίστοιχους αλγόριθμους αναζήτησης και βελτιστοποίησης. Κατά πρώτους οι Γενετικοί αλγόριθμοι είναι από τη φύση τους ένας παράλληλος αλγόριθμος που εκτελεί την αναζήτηση σε πολλά σημεία του πεδίου ορισμού του προβλήματος, τόσα όσα και ο πληθυσμός των ατόμων του Γενετικού Αλγόριθμου. Ακόμα ένα πολύ σημαντικό και επωφελές χαρακτηριστικό είναι ότι οι αλγόριθμοι αυτοί ασχολούνται με τις τιμές που παίρνει ως παραμέτρους το πρόβλημα μας αλλά όχι με τις ίδιες τις μεταβλητές του προβλήματος. Με την σωστή κωδικοποίηση των τιμών μπορούμε να κινούμαστε μέσα στο πεδίο ορισμού των μεταβλητών, όσο πολύπλοκος και αν είναι αυτός, αλλάζοντας απλά την δυαδική ακολουθία που κωδικοποιεί ένα άτομο. Το κυρίαρχο όμως χαρακτηριστικό των γενετικών αλγορίθμων είναι ίσως η αντικειμενική συνάρτηση αξιολόγησης που έχει τον ρόλο του περιβάλλοντος. Χωρίς καμιά άλλη πληροφορία μπορούμε να αξιολογούμε τις λύσεις που παράγονται από τον γενετικό αλγόριθμο μας, κομμάτι πολύ σημαντικό για την λειτουργία του αλγόριθμου. Οι γενετικοί αλγόριθμοι παίρνουν την τύχη με το μέρος τους.

Χωρίς να γίνονται ένα τυχαίο ψάξιμο χρησιμοποιούν την τύχη για να κατευθυνθούν στην υποτιθέμενη καλύτερη περιοχή του χώρου αναζήτησης και δεν αποκλείουν όμως λύσεις που αν και δεν έχουν τόσο καλή απόδοση στο μέλλον ίσως οδηγήσουν στη βέλτιστη λύση.

Οι Γενετικοί Αλγόριθμοι απευθύνονται σε ένα ευρύ φάσμα προβλημάτων. Για να χρησιμοποιηθούν όμως προς όφελος μας θα πρέπει να γνωρίζουμε τα βασικά στοιχεία που τον αποτελούν και την σωστή διεργασία η οποία πρέπει να διέπει τον υλοποιημένο αλγόριθμο. Τα τρία βασικά δομικά στοιχεία που επιτελούν ένα Γενετικό Αλγόριθμο είναι η κωδικοποίηση, η αντικειμενική συνάρτηση αξιολόγησης και οι γενετικές διαδικασίες.

### Κωδικοποίηση:

Για να συνδέσουμε ένα πρόβλημα με ένα γενετικό αλγόριθμο πρέπει να κωδικοποιήσουμε τις πιθανές λύσεις του προβλήματος με ένα τρόπο που θα επιτρέπει σε ένα ηλεκτρονικό υπολογιστή να τις επεξεργαστεί ώστε να μπορεί να κινηθεί σε όλο το εύρος των τιμών των λύσεων. Με λίγα λόγια η κωδικοποίηση προσπαθεί να αναπαραστήσει όλα τα χαρακτηριστικά με τον καλύτερο πιθανό τρόπο μιας πιθανής λύσης ώστε να διευκολύνει την μετέπειτα λειτουργία του αλγόριθμου.

Είναι πιστεύω φανερό ότι η κωδικοποίηση συνήθως διαφέρει ανάλογα με την φύση του προβλήματος. Η πιο συνήθης κωδικοποίηση στη χρήση είναι με την χρήση δυαδικών συμβολοσειρών πεπερασμένου μήκους, όπου οι λύσεις αναπαριστώνται με δυαδικά ψηφία. Πολλές κωδικοποιήσεις έχουν προταθεί και χρησιμοποιηθεί κατά καιρούς με γνώμονα φυσικά πάντα το πρόβλημα το οποίο ήθελαν να κωδικοποιήσουν οι εκάστοτε ερευνητές. Λανθασμένη κωδικοποίηση στη χειρότερη περίπτωση ίσως δεν αφήσει τον αλγόριθμο να λειτουργήσει σωστά, ενώ στην καλύτερη θα αφαιρέσει τιμές από το πεδίο ορισμού του προβλήματος.

### Συνάρτηση Αξιολόγησης:

Το πλέον σημαντικό στοιχείο του γενετικού αλγόριθμου είναι η αντικειμενική συνάρτηση αξιολόγησης. Εδώ βλέπουμε το πόσο ικανά λύνει το πρόβλημα μια συγκεκριμένη λύση. Παίρνοντας ως είσοδο τα αποκωδικοποιημένα μέρη του ατόμου επιστρέφει μια τιμή συνήθως στο πεδίο των πραγματικών αριθμών η οποία δεικνύει πόσο ικανοποιητικά λύει το πρόβλημα το συγκεκριμένο άτομο. Η τιμή αυτή θα παίζει καθοριστικό ρόλο στην διαδικασία της επιλογής. Θα πρέπει να γίνεται προσπάθεια η συνάρτηση αυτή να είναι όσο πιο απλή γίνεται για να μην επιβραδύνει πολύ τον αλγόριθμο.

### Γενετικές Διαδικασίες:

Ένας απλός ψευδοκώδικας για τον γενετικό αλγόριθμο είναι η εξής:

1. Αρχικοποίηση
2. Αποκωδικοποίηση
3. Υπολογισμός Ικανότητας
4. Αναπαραγωγή
  - a. Επιλογή
  - b. Διασταύρωση
  - c. Μετάλλαξη
5. Επανάληψη από το βήμα (2) μέχρι να ικανοποιηθεί το κριτήριο τερματισμού.

Η αρχικοποίηση, το πρώτο βήμα του γενετικού αλγόριθμου, είναι το σημείο που ορίζεται ο αρχικός πληθυσμός από τον ξεκινά η αναζήτηση μας. Η επιλογή του πληθυσμού αυτού γίνεται με τυχαίο τρόπο μέσα από το πεδίο ορισμού και κωδικοποιείται σύμφωνα με την επιλεγμένη κωδικοποίηση. Το μέγεθος του πληθυσμού επιλέγεται από τον χρήστη και εξαρτάται και πάλι από τον χρήστη και το εύρος τους πεδίου ορισμού του προβλήματος.

Έχοντας σχηματίσει την πρώτη γενιά ο Γενετικός αλγόριθμος μπορεί να ξεκινήσει τον κύκλο επαναλήψεων του. Το πρώτο βήμα είναι η αξιολόγηση όλων των ατόμων του

πληθυσμού του γενετικού αλγόριθμου ξεχωριστά για το κάθε ένα. Για να επιτευχθεί αυτό κάθε άτομο πρέπει να αποκωδικοποιηθεί και να εισαχθεί στην συνάρτηση αξιολόγησης. Όπως προανέφερα η συνάρτηση αξιολόγησης αποτελεί το περιβάλλον του τεχνητού μοντέλου και όπως στο περιβάλλον μεγαλύτερη πιθανότητα να επιβιώσουν έχουν τα πιο αξιόλογα ζώα ως προς κάποια ικανότητα έτσι και στον αλγόριθμο η συνάρτηση αξιολόγησης πρέπει να επιστρέφει μεγαλύτερη τιμή για τα πιο αξιόλογα άτομα.

Μετά ακολουθεί η αναπαραγωγή όπου βρίσκεται ο μεγαλύτερος φόρτος εργασίας και μέσω αυτής της διεργασίας επιτυγχάνεται και η βελτιστοποίηση συνεπώς και το τελικό αποτέλεσμα. Χωρίζεται σε τρία μέρη την επιλογή, την διασταύρωση και την μετάλλαξη.

Η επιλογή είναι η μεταφορά του νόμου επιβίωσης του ισχυρότερου στα πλαίσια του αλγορίθμου. Μέσω αυτής της διαδικασίας επιλέγονται ποια άτομα από την παρούσα γενιά θα επιβιώσουν ώστε να αναπαραχθούν και να μεταβιβάσουν έτσι στην επόμενη γενιά τα χαρακτηριστικά τους ολόκληρα ή ένα κομμάτι εάν υπάρξει διασταύρωση η μετάλλαξη. Σκοπός τη επιλογής είναι φανερά η βαθμιαία αύξηση των ικανοτέρων στον πληθυσμό χωρίς την άμεση απόρριψη των λιγότερο ικανών, για να έχουμε μια πιο σφαιρική αναζήτηση, μέχρι να επιτευχθεί η πλήρης αντικατάσταση του πληθυσμού από τα ικανότερα άτομα. Και πάλι οι ερευνητές έχουν χρησιμοποιήσει διάφορους τρόπους επιλογής όμως ο πιο ευρέως διαδεδομένος και χρησιμοποιημένος αλγόριθμος είναι αυτός της εξαναγκασμένης ρουλέτας. Στην εξαναγκασμένη ρουλέτα αντιπροσωπεύεται κάθε άτομο του πληθυσμού με μερίδα τον λόγο της απόδοσης του ιδίου του ατόμου δια την συνολική καταλληλότητα. Για παράδειγμα ένα άτομο με υποτιθέμενη ικανότητα 4 σε ένα πληθυσμό συνολικής ικανότητας 20 θα είχε πιθανότητα επιλογής 20%. Για να επιλέξουμε ένα άτομο τοποθετούμε αθροιστικά τις πιθανότητες επιλογής, συνεχίζοντας το προηγούμενο παράδειγμα το άτομο αυτό ίσως να καταλάμβανε τον χώρο από 65 μέχρι 85. Το άτομο θα επιλεγόταν αν η παραγωγή ενός τυχαίου αριθμού στο χώρο αυτό των πιθανοτήτων έπεφτε στο πεδίο που καλύπτει. Οπότε για το πιο πάνω παράδειγμα αν ο τυχαίος αριθμός ήταν 67 το άτομο θα επιλεγόταν αφού ανήκει στο πεδίο 65-85, αν ο τυχαίος αριθμός ήταν 23 τότε θα επιλεγόταν το άτομο του οποίου το πεδίο πιθανότητας καλύπτει τον συγκεκριμένο αριθμό. Πιθανώς πιο βατή εξήγηση να υπάρχει στο παράδειγμα που ακολουθεί στο τέλος του

κεφαλαίου. Η διαδικασία αυτή επαναλαμβάνεται τόσες φορές όσες ο αριθμός των ατόμων του πληθυσμού. Ο παραγόμενος προσωρινός πληθυσμός είναι αυτός που θα συνεχίσει την αναπαραγωγή και ονομάζεται δεξαμενή ζευγαρώματος (mating pool).

Με τα άτομα που θα αναπαραχθούν να έχουν επιλεγθεί ήδη ο νέος πληθυσμός χωρίζεται με τυχαίο τρόπο σε ομάδες των δύο για να επιτελεστεί η πράξη της διασταύρωσης. Η διασταύρωση είναι μια απαραίτητη λειτουργία που συμβάλλει αποφασιστικά στην λύση του προβλήματος με το οποίο ασχολείται ο γενετικός αλγόριθμος. Η διασταύρωση έχει σκοπό να δημιουργήσει μια νέα διαφορετική γενιά που περιέχουν αναμειγμένο κληρονομικό υλικό από τους ικανότερους της προηγούμενης γενιάς με στόχο να παραχθούν άτομα με ακόμα μεγαλύτερη ικανότητα. Αυτό εξάλλου δείχνει και τη σημαντικότητα της διασταύρωσης αφού οδηγεί στην δημιουργία νέων λύσεων που με μεγάλη πιθανότητα θα μας οδηγήσουν στο βέλτιστο. Υπάρχει και το ενδεχόμενο μερικά από τα άτομα που θα παραχθούν να είναι χειρότερα αλλά αυτά θα εξαλειφθούν στον επόμενο κύκλο από την διαδικασία της επιλογής. Η διασταύρωση λαμβάνει χώρα με τυχαίο τρόπο. Υπάρχει μια πιθανότητα που ονομάζεται πιθανότητα διασταύρωσης και καθορίζει κατά πόσον ένα άτομο θα αναπαραχθεί. Και πάλιν πιο βατή εξήγηση στο παράδειγμα που θα ακολουθήσει.

Η μετάλλαξη είναι μια λιγότερο σημαντική πράξη και μάλιστα μερικές φορές είναι αμελητέα. Χρησιμοποιείται για διαφυγή από τοπικά ακρότατα του χώρου λύσεων. Καθώς αντιγράφονται τα άτομα της προηγούμενης γενιάς στα άτομα της επόμενης γενιάς επιλέγεται τυχαία μια θέση στο άτομο και αντιστρέφεται το ψηφίο αυτό από 0 σε 1 ή αντίστροφα. Αυτό γίνεται με πολύ μικρή πιθανότητα που ονομάζεται πιθανότητα μετάλλαξης. Η συγκεκριμένη πιθανότητα πρέπει να είναι πολύ μικρή για να μην μετατρέπεται ο αλγόριθμος μας σε τυχαίο ψάξιμο.

#### ΠΑΡΑΔΕΙΓΜΑ:

Έστω η συνάρτηση προς μεγιστοποίηση  $F=x^2+y$  με πεδία ορισμού  $0 \leq x, y \leq 15$  .

Φυσικά αυτό είναι ένα τετριμμένο πρόβλημα του οποίου γνωρίζουμε τη λύση αλλά το χρησιμοποιώ χάριν του παραδείγματος.

Κατ' αρχήν πρέπει να κωδικοποιήσουμε τα άτομα. Η κωδικοποίηση των ατόμων πρέπει να κωδικοποιεί τις λύσεις των ατόμων. Έχουμε δύο μεταβλητές των οποίων οι τιμές κυμαίνονται από 0 έως 15. Σε δυαδική μορφή χρειαζόμαστε 4 ψηφία για να κωδικοποιήσουμε αυτούς τους αριθμούς. Οπότε το άτομο θα αποτελείται από 8 ψηφία όπου τα 4 πρώτα αντιστοιχούν στην τιμή του  $x$  ενώ τα επόμενα 4 στην τιμή του  $y$ . Το άτομο 00101010 κωδικοποιεί την τιμή 2 για το  $x$  και 10 για το  $y$ .

Τώρα έστω ο αρχικός πληθυσμός  $\alpha_1=01000101$ ,  $\alpha_2=01100010$ ,  $\alpha_3=01000101$ ,  $\alpha_4=01001001$ .

Η συνάρτηση αξιολόγησης σε αυτή την περίπτωση είναι το ίδιο το πρόβλημα. Έτσι αποκωδικοποιώντας τις συμβολοσειρές παίρνουμε τα αποτελέσματα του Πίνακα 3.1.

**Πίνακας 3-1**

Συμβολοσειρά	Απόδοση	Λόγος Απόδοσης/ Συνολικής Απόδοσης.
01000101 $\rightarrow x=4, y=5$	31	27%
01100010 $\rightarrow x=6, y=2$	38	33%
01000101 $\rightarrow x=4, y=5$	21	18%
01001001 $\rightarrow x=4, y=9$	25	22%
Συνολική απόδοση	115	

Μετά ακολουθεί η διαδικασία της επιλογής. Η καταναγκασμένη ρουλέτα που σχηματίζεται θα ήταν κάπως έτσι.



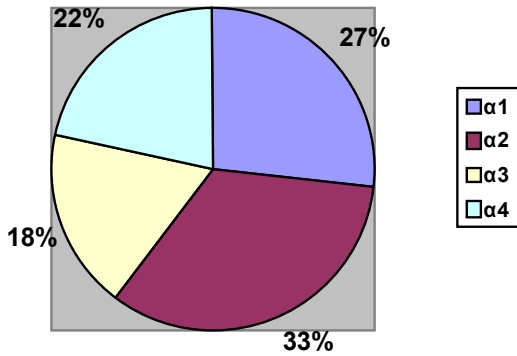


Figure 3-2 Εξαναγκασμένη ρουλέτα

Έτσι θα μπορούσαμε να παραγάγουμε 4 τυχαίους αριθμούς από το 0 ως το 99 για να σχηματίσουμε τη δεξαμενή ζευγαρώματος. Αν ο αριθμός ήταν από 0 ως 26 τότε επιλέγεται το α1 , από 27 έως 59 επιλέγεται το α2, από 60 έως 77 το α3 και 78 μέχρι το 99 το α4.

Μια πιθανή ακολουθία αριθμών είναι 29,56,8,90. Αυτό θα μας έδινε την δεξαμενή α1'=α2, α2'=α2, α3'=α1, α4'=α4.

Έστω ότι η πιθανότητα διασταύρωσης είναι 20%. Και έστω η ακολουθία 34,15,14,76. Αυτό θα σήμαινε ότι τα α2' και α3' θα διασταυρωθούν ενώ τα άλλα όχι.

Έτσι πρέπει να βρούμε ένα τυχαίο αριθμό μεταξύ 1 και 8 όπου θα γίνει η διασταύρωση. Έστω ότι ο αριθμός είναι το 6.

$$\alpha2' = 011000|10 \rightarrow 01100001$$

$$\alpha3' = 010001|01 \rightarrow 01000110$$

Η μετάλλαξη που γίνεται με πιο μικρή πιθανότητα έστω ίση με 2% ας υποθέσουμε ότι εφαρμόζεται στο 1 bit του α4' για χάριν του παραδείγματος.

$$\alpha4' = 01001001 \rightarrow 11001001$$

Η νέα γενιά θα είναι τώρα η εξής  $\alpha_1=01100010$ ,  $\alpha_2=01100001$ ,  $\alpha_3=01000110$ ,  $\alpha_4=11001001$ .

**Πίνακας 3-2**

Συμβολοσειρά	Απόδοση	Λόγος Απόδοσης/ Συνολικής Απόδοσης.
01100010 $\rightarrow x=6, y=2$	38	15%
01100001 $\rightarrow x=6, y=1$	37	15%
01000110 $\rightarrow x=4, y=6$	22	9%
11001001 $\rightarrow x=12, y=9$	153	61%
Συνολική απόδοση	250	

Βλέπουμε ότι μετά από μόλις μια επανάληψη η συνολική απόδοση έχει αυξηθεί κατά πολύ.

# Κεφάλαιο 4

## Μοντελοποίηση Προβλήματος και Υλοποίηση Πρωτοτύπου

---

4.1 Ανάλυση του προβλήματος και μοντελοποίηση του	28
4.2 Υλοποίηση του πρωτοτύπου	38
4.3 Αποτελέσματα	44

---

### 4.1 Ανάλυση του προβλήματος και μοντελοποίηση του

Το πρόβλημα όπως ορίστηκε και στο πρόβλημα το κεφάλαιο είναι ο σχηματισμός λειτουργικών ομάδων εργασίας για την διεκπεραίωση ενός έργου παραγωγής λογισμικού με βέλτιστη υλοποίηση των πόρων ή/και ελαχιστοποίηση του χρόνου διάρκειας του έργου. Όσοι υπήρξαν μέρος ενός έργου παραγωγής λογισμικού ξέρουν ότι αποτελεί μια πολύ ιδιαίτερη περίπτωση. Είναι ένα συνεχές τρέξιμο ενάντια στον χρόνο με πολλές αναπάντεχες ανατροπές που οφείλονται είτε σε ανθρώπους είτε σε υλικό. Ένα πρόβλημα στις διαπροσωπικές σχέσεις των υπαλλήλων που εργάζονται σε κάποιο έργο μπορεί να καθυστερήσει ή ακόμα να γίνει και αιτία αποτυχίας. Οι περιορισμένες ικανότητες ή ακόμα και απουσία αυτών από κάποιους υπαλλήλους σε κάποιο συγκεκριμένο θέμα πιθανόν να δημιουργήσουν ανυπολόγιστες ζημιές. Ακόμα ο λανθασμένος προγραμματισμός των εργασιών πιθανόν να καθυστερήσουν ένα έργο με αποτέλεσμα να χαθεί η εργασία. Για όλα αυτά υπεύθυνος και υπόλογος είναι μόνο ένας ο διευθύνων του έργου.

Με άλλα λόγια ένας διευθύνων έργου δεν μπορεί να κοιμάται ποτέ ήσυχος, ειδικά αν όλος ο προγραμματισμός που αφορά το έργο γίνεται ad-hoc. Η όλη οργάνωση του έργου πρέπει να γίνεται μεθοδικά και διεξοδικά μελετώντας όλες τις διαθέσιμες παραμέτρους. Όμως αυτό δεν μπορεί να γίνεται συνέχεια από τον διευθύνοντα έργο, όντας και αυτός άνθρωπος αν αφηθεί μονάχος πιθανόν να του διαφύγουν μερικά σημεία που ίσως να είναι σημαντικά ίσως και όχι. Για αυτό τον σκοπό δημιουργούνται και τα εργαλεία χρονοπρογραμματισμού. Όπως προανέφερα όμως τα έργα παραγωγής λογισμικού αποτελούν μια ιδιάζουσα περίπτωση. Εφόσον ένα έργο ασχολείται κυρίως με ανθρώπινες σχέσεις, δεν μπορεί να παρατηρείται με την ψυχρή λογική των μέχρι τώρα υπάρχοντων εργαλείων και παράλληλα ο διευθύνων έργου με την δική του εμπειρία δεν μπορεί να παραγκωνίζεται εντελώς από αυτή τη διεργασία.

Ο ρόλος που είχε η διπλωματική μου εργασία ήταν να καλύψω αυτό το κενό. Μέσα από τη μοντελοποίηση του αλγορίθμου και εν τέλει και την υλοποίηση του πρωτοτύπου να έχω ένα υποσύστημα που να βλέπει και την ανθρώπινη πτυχή αλλά και να μπορεί ο διευθύνων έργου να προσθέσει την υποκειμενικότητα του όπως και να μπορεί να διαλέξει μέσα από εναλλακτικές λύσεις. Και φυσικά να υλοποιεί τις δύο αρχικές προϋποθέσεις να έχει βέλτιστη υλοποίηση των πόρων ή/και ελαχιστοποίηση του χρόνου διάρκειας του έργου.

Αρχικά έπρεπε να κάμω κάποιες υποθέσεις. Οι υποθέσεις αυτές ήταν οι εξής:

1. Οι υπάλληλοι χωρίζονται σε πέντε κατηγορίες και κάθε υπάλληλος μπορεί να ανήκει μόνο σε μια εξ αυτών
  - a. Προγραμματιστές
  - b. Σχεδιαστές
  - c. Αναλυτές
  - d. Ελεγκτές
  - e. Και χειριστές βάσεων δεδομένων
2. Οι εργασίες είναι εντελώς απλοποιημένες στο επίπεδο που εργάζονται υπάλληλοι μόνο από μια κατηγορία. Άρα μια εργασία μπορεί να είναι είτε προγραμματιστική, είτε σχεδιαστική και ούτω καθ' εξής.

3. Για κάθε υπάλληλο έχω τον ψυχολογικό του τύπο MBTI και τις βαθμολογίες του για καθορισμένο αριθμό ικανοτήτων. Εδώ είναι που μπαίνει και η υποκειμενικότητα του διευθύνοντα αφού αυτός βαθμολογεί τον κάθε υπάλληλο.
4. Για κάθε εργασία γνωρίζω πέρα από το είδος της το βάρος που έχει κάθε αναγκαία ικανότητα για την συγκεκριμένη εργασία και ακόμα την διάρκεια σε ανθρωπομέρες όπως και τις άλλες εργασίες από τις οποίες εξαρτάται η συγκεκριμένη εργασία.
5. Σαν είσοδο έχω όλες τις εργασίες ενός συγκεκριμένου έργου και όλους τους διαθέσιμους υπάλληλους.
6. Τέλος μια τελευταία υπόθεση είναι ότι ο αριθμός των υπαλλήλων κάθε κατηγορίας είναι αρκετός για να καλύψει με τουλάχιστον ένα ανά εργασία της ίδιας κατηγορίας.

Έχοντας κάνει αυτές τις υποθέσεις και έχοντας ήδη επιλέξει τις μεθοδολογίες τις οποίες θα χρησιμοποιούσα έπρεπε πλέον να δημιουργήσω το μοντέλο του γενετικού μου αλγόριθμου. Για να μπορέσω να το πράξω αυτό μελέτησα έναν αριθμό από μελέτες και επιστημονικά άρθρα που ασχολούνται με παρόμοια θέματα ώστε να πάρω μερικές ιδέες. Εν τέλει κατέληξα ότι οι ήδη προτεινόμενες μοντελοποιήσεις δεν μπορούσαν να εξυπηρετήσουν τον σκοπό μου έτσι κατέληξα σε μια καινούρια δική μου μοντελοποίηση. Φυσικά αυτό έγινε με την καθοδήγηση του επιβλέποντα μου καθηγητή και του βοηθού του.

Στις μοντελοποιήσεις τις οποίες είχα την ευκαιρία να μελετήσω σε καμιά περίπτωση δεν συνδύαζαν την έννοια του χρονοπρογραμματισμού μαζί με την δέσμευση πόρων, ή αν αυτό γινόταν ήταν μέσα από τον συνδυασμό δύο γενετικών αλγορίθμων. Στον δικό μου αλγόριθμο αντιλήφθηκα ότι ο δεύτερος ΓΑ δεν είναι αναγκαίος αφού ενσωμάτωσα την έννοια της διάρκειας μέσα στη συνάρτηση αξιολόγησης του πρώτου ΓΑ που ασχολείται με την δέσμευση των πόρων. Έτσι πλέον ο γενικός σκοπός του γενετικού αλγόριθμου είναι να σχηματίσει τις ομάδες από πρόσωπα για την κάθε εργασία που αποτελεί ένα έργο συναρτήσει του ρόλου, των ικανοτήτων και της συνοχής της ομάδας, έχοντας παράλληλα την μικρότερη δυνατή διάρκεια τιμωρώντας παραβιάσεις των αλληλεξαρτήσεων των εργασιών.

## ΚΩΔΙΚΟΠΟΙΗΣΗ:

Έχοντας υπόψη μου τις υποθέσεις που διατύπωσα πιο πάνω κατέληξα στην εξής κωδικοποίηση. Ένα έργο μπορεί να διαφέρει κατά πολύ από ένα άλλο στον αριθμό και το είδος των εργασιών που το απαρτίζουν. Αποφάσισα λοιπόν να καθορίσω μια δομή ατόμου κάπως ελαστική έτσι ώστε να μπορεί να αντεπεξέλθει σε κάθε δομή έργου. Το άτομο είναι μια δυαδική ακολουθία. Το μήκος του εξαρτάται από τρεις παράγοντες. Ο πρώτος παράγοντας είναι ο αριθμός από εργασίες που αποτελούν το έργο. Οπότε μια αφαιρετική μορφή του ατόμου λαμβάνοντας υπόψη μόνο αυτό τον πρώτο παράγοντα θα ήταν αυτή.

Π.χ. σε ένα έργο που αποτελείται από 7 εργασίες το άτομο του Γενετικού Αλγόριθμου θα έμοιαζε κάπως έτσι :

**Πίνακας 4-1**

Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7
--------	--------	--------	--------	--------	--------	--------

Ας επικεντρωθούμε σε ένα κομμάτι του πιο πάνω χρωμοσώματος. Κάθε εργασία δύναται να περιέχει  $n$  ισομεγέθη δυαδικές συμβολοσειρές μήκους  $m$ . Όπου  $n$  είναι ο αριθμός των υπαλλήλων που επανδρώνουν την εργασία συν ένα λόγω της θέσης της εργασίας στο έργο. Και  $m$  είναι η οροφή του λογάριθμου με βάση το 2, εφόσον μιλάμε για δυαδική ακολουθία, του μέγιστου αριθμού εκ των μεγεθών του πλήθους των εργασιών, του πλήθους των προγραμματιστών, του πλήθους των σχεδιαστών, του πλήθους των αναλυτών, του πλήθους των δοκιμαστών και του πλήθους των χειριστών βάσεων δεδομένων. Να σημειώσω εδώ ότι μπροστά βρίσκεται η θέση της εργασίας στο χρονοπρόγραμμα και μετά ακολουθούν οι δεσμευμένοι υπάλληλοι. Οπότε παίρνοντας μια εργασία του πιο πάνω παραδείγματος π.χ. αν η εργασία 1 χρειαζόταν 6 άτομα για να επανδρωθεί, έστω προγραμματιστές, θα είχε τη εξής μορφή:

**Πίνακας 4-2**

POSITION	Programmer	Programmer	Programmer	Programmer	Programmer	Programmer
	1	2	3	4	5	6

Έτσι αν για παράδειγμα το μήκος που όρισα ως  $m$  πιο πάνω είναι 3 και το σύνολο των προγραμματιστών ήταν επτά ενώ το πλήθος των εργασιών ήταν 5 τότε αυτό το κομμάτι του χρωματοσώματος με τυχαία νούμερα θα μπορούσε να έχει την εξής μορφή: 010 – 011-010- 001- 000-110-101. Η σημασία αυτής της συμβολοσειράς θα ήταν ότι η εργασία ένα θα εκτελεστεί δεύτερη στη σειρά από όλες τις εργασίες και μέσα σε αυτήν θα εργάζονται ο τέταρτος, ο τρίτος, ο δεύτερος, ο πρώτος, ο έβδομος και ο έκτος προγραμματιστές. Όπως είναι φανερό τόσο στους υπάλληλους, οποιασδήποτε κατηγορίας, όσο και στις εργασίες η αναφορά σε αυτούς γίνεται μέσω αριθμού δείκτη. Ο δείκτης αυτός αναφέρεται στην δομή μέσα στην οποία θα βρίσκεται αποθηκευμένο το κάθε αντικείμενο.

Εννοείται ότι η δομή του χρωματοσώματος πρέπει να είναι γνωστή για να μπορεί να γίνεται η σωστή αποκωδικοποίηση του. Για τον λόγο αυτό παράλληλα με κάθε συμβολοσειρά θα υπάρχει ένας πίνακας πινάκων ακεραίων μεγέθους ίσου με τον αριθμό των εργασιών που θα κρατά τις θέσεις στο συμβολοσειρά που τελειώνει κάθε εργασία για πιο εύκολη αποκωδικοποίηση. Έτσι για να αποκωδικοποιήσουμε μια συγκεκριμένη εργασία βρίσκουμε τη θέση της στον πίνακα αυτών των ακεραίων και εφόσον ξέρουμε την κατηγορία της (προγραμματιστική, σχεδιαστική κτλ), το μήκος  $m$  και την θέση που τελειώνει η συγκεκριμένη εργασία πάμε στην δομή δεδομένων που ανταποκρίνεται στους υπαλλήλους με αυτό τον ρόλο και αφού διαβάσαμε το πρώτο κομμάτι που μας δίνει την θέση της εργασίας αποκωδικοποιούμε τους υπαλλήλους που βρίσκονται μέσα. Για παράδειγμα έστω η συμβολοσειρά 001-000-010-000-010. Ξέροντας ότι το δομικό στοιχείο  $m$  είναι 3 και ότι η εργασία1 είναι στην θέση 0 ενώ η εργασία2 στην θέση 9 μπορούμε να πούμε ότι η εργασία1 θα διεξαχθεί δεύτερη στη σειρά με τους υπάλληλους 1 και 3 της κατηγορίας με την οποία ασχολείται. Ενώ η εργασία2 εκτελείται πρώτη στη σειρά με τον υπάλληλο 3 της κατηγορίας με την οποία ασχολείται.

## ΣΥΝΑΡΤΗΣΗ ΑΞΙΟΛΟΓΗΣΗΣ:

Βεβαίως όπως τόνισα και στο προηγούμενο κεφάλαιο η αντικειμενική συνάρτηση αξιολόγησης είναι το πιο σημαντικό κομμάτι του Γενετικού Αλγόριθμου. Στην δική μου περίπτωση συνδυάζει πολλά διαφορετικά μεγέθη εφόσον ασχολούμαι με τόσους πολλούς διαφορετικούς παράγοντες. Εδώ βρίσκεται η γενική μορφή της αντικειμενική συνάρτηση αξιολόγησης και παρακάτω θα εξηγήσω τα επιμέρους κομμάτια της συνάρτησης:

### Equation 4-1

$$T = \sum_{i=0}^N \left( \frac{\text{Ικανότητα}}{\text{Επικοινωνία} + 1} \right) \times \frac{1}{(M + 1) * \text{Διάρκεια}}$$

T: ο βαθμός ικανότητας του ατόμου του Γενετικού αλγόριθμου

N: ο αριθμός των εργασιών του έργου

Ικανότητα: Το άθροισμα των ικανοτήτων των υπαλλήλων σταθμισμένα με το βάρος τους

Επικοινωνία: Η δυσκολία επικοινωνίας της ομάδας με βάση τον τροχό επικοινωνίας Thompson

M: Ο αριθμός των παραβάσεων των εξαρτήσεων

Διάρκεια: Η διάρκεια σε ανθρωπομέρες της συγκεκριμένης εργασίας

Άθροισμα ικανοτήτων:

Για τη συνάρτηση αξιολόγησης θα λαμβάνονται υπόψη οι αναγκαίες ικανότητες από το task καθώς και η συνοχή της ομάδας. Για να γίνει αυτό θα πρέπει να χρησιμοποιηθεί η δομή δεδομένων που ανταποκρίνεται στην σωστή κατηγορία των υπαλλήλων σε σχέση με την εργασία που ελέγχουμε με τις ικανότητες που έχουν. Για παράδειγμα ένας προγραμματιστής, λόγω χάρη ο 0010, θα έχει βαθμολογία σε όλες τις ικανότητες (π.χ ικανότητα προγραμματισμού σε Java) σε μια κλίμακα από 0-10 όπου 0 σημαίνει δεν κατέχει την ικανότητα αυτή ενώ 10 σημαίνει ότι είναι άριστος γνώστης αυτής. Έτσι οι βαθμολογίες για τις αναγκαίες ικανότητες θα πολλαπλασιάζονται με κάποιο βάρος που θα αντιστοιχεί στην σημαντικότητα της ικανότητας αυτής για την εργασία. Τόσο το βάρος όσο και οι αναγκαίες ικανότητες θα καθορίζονται από τον χρήστη του συστήματος. Τέλος θα



αθροίζονται για κάθε εργασία ξεχωριστά τα αποτελέσματα και ξανά άθροισμα θα γίνεται για όλα τις εργασίες ατόμου του Γ.Α.

Π.χ. έστω ένα Task x

**Πίνακας 4-3**

Task X	Προγραμματισμός σε C	Documentation	Οργάνωση Κώδικα
Βάρος Ικανότητας	6/10	3/10	1/10

που περιέχει τρεις προγραμματιστές

**Πίνακας 4-4**

	Προγραμματισμός σε C	Documentation	Οργάνωση Κώδικα
Ικανότητα Α Προγραμματιστή	7/10	5/10	9/10
Ικανότητα Β Προγραμματιστή	5/10	9/10	8/10
Ικανότητα Γ Προγραμματιστή	8/10	3/10	4/10

Τότε θα έχουμε ως αποτέλεσμα αξιολόγησης του γονιδίου

$$7/10*6/10+5/10*3/10+9/10*1/10+5/10*6/10+9/10*3/10+8/10*1/10+8/10*6/10+3/10*3/10+4/10*1/10=2.28.$$

Μπορούμε να υποθέσουμε οποιεσδήποτε ικανότητες χωρίς αυτές να επηρεάζουν το σύστημα όπως ούτε και αριθμός τους μας επηρεάζει. Αυτό είναι που δίνει την δυνατότητα στον αλγόριθμο μου να χρησιμοποιηθεί για έργα οποιασδήποτε φύσης. Ακόμα το γεγονός ότι οι βαθμολογίες των ικανοτήτων καθώς και το βάρος τους για κάθε εργασία εισάγεται από τον χειριστή του συστήματος άρα από τον διευθύνων του έργου προσθέτει την θεμιτή υποκειμενικότητα που ήθελα να προσθέσω στο σύστημα μου.

### Δυσκολία επικοινωνίας ομάδας:

Είδαμε στο δεύτερο κεφάλαιο πως μπορούμε να βρούμε την δυσκολία δύο προσώπων εφόσον φυσικά γνωρίζουμε τον τύπο του σύμφωνα με την μεθοδολογία MBTI. Για ευκολία της υλοποίησης του αλγόριθμου έχω κωδικοποιήσει τους 16 τύπους του MBTI με ακέραιους αριθμούς από ένα μέχρι 16. Θεωρώ τον τύπο ESFP ως ένα και συνεχίζω την μέτρηση δεξιόστροφα με το ESTP να παίρνει την τιμή 16.(βλέπε σχήμα 2-2) Για να υπολογίσω τη συνολική δυσκολία επικοινωνίας σε μια ομάδα έχω αυτοματοποιήσει τη διαδικασία που περιέγραψα στο τέλος του δευτέρου κεφαλαίου. Δημιουργώ ένα τετραγωνικό πίνακα μέσα στον οποίο τοποθετώ τη δυσκολία επικοινωνίας μεταξύ κάθε ζευγαριού της ομάδας. Σε ένα παράδειγμα με τρεις υπαλλήλους έχουμε τον παρακάτω πίνακα:

**Πίνακας 4.5**

	Υπάλληλος 1	Υπάλληλος 2	Υπάλληλος 3
Υπάλληλος 1	X	3/8	1/8
Υπάλληλος 2	3/8	X	6/8
Υπάλληλος 3	1/8	6/8	X

$$\text{Άρα ως αποτέλεσμα έχουμε } (3/8+1/8+6/8)/3=11/24$$

Παρατηρούμε ότι οι δυσκολίες επικοινωνίας επαναλαμβάνονται δύο φορές στον πίνακα, έτσι πρέπει να ασχοληθούμε μόνο με το πάνω μισό τριγωνικό πίνακα, εκεί όπου ο αριθμός των στηλών είναι μεγαλύτερος από τον αριθμό των γραμμών. Όπως προανέφερα η μέγιστη πιθανή δυσκολία επικοινωνίας είναι 8. Τέλος για να λάβουμε τη συνολική δυσκολία της ομάδας αθροίζουμε όλες τις θέσεις του πίνακα που βρίσκονται στο τριγωνικό κομμάτι που περιέγραψα και βρίσκουμε το μέσο όρο αυτών διαιρώντας δηλαδή το άθροισμα με τον αριθμό των υπαλλήλων.

### Αλληλεξαρτήσεις μεταξύ των εργασιών:

Μέσα στα άτομα του γενετικού αλγορίθμου, όπως τον περιέγραψα πιο πάνω υπάρχει κωδικοποιημένη η θέση κάθε εργασίας σε σχέση με τις υπόλοιπες εργασίες. Επίσης μέσα στις υποθέσεις τις οποίες έκανα ήταν ότι οι αλληλεξαρτήσεις των εργασιών μου ήταν γνωστές εκ των προτέρων. Έτσι το μόνο που πρέπει να κάνω για να βρω κατά πόσο κάποια αλληλεξάρτηση μιας συγκεκριμένης εργασίας παραβιάζεται είναι να κοιτάξω σε όλες τις εργασίες από τις οποίες εξαρτάται η συγκεκριμένη εργασία και να δω αν αυτές οι εργασίες έχουν μεγαλύτερη θέση σε σχέση με αυτή την εργασία. Για κάθε αλληλεξάρτηση που παραβιάζεται αυξάνω το μετρητή M. Με αυτό το τρόπο πετυχαίνω να μην αποκλείω εντελώς λύσεις με τέτοιες παραβιάσεις αλλά παράλληλα να τις τιμωρώ και να μειώνω την πιθανότητα να αναπαραχθούν στην επόμενη γενιά. Αυτό μου δίνει τη δυνατότητα να κινούμε σε ένα ευρύτερο χώρο λύσεων.

### Διάρκεια εργασίας:

Εφόσον έχω ως δεδομένο τη διάρκεια μιας εργασίας σε ανθρωπομέρες, όπως προανέφερα στις υποθέσεις μου, μπορώ αποκωδικοποιώντας το γονίδιο που αναφέρεται σε μια συγκεκριμένη εργασία να ξέρω πόσοι είναι οι υπάλληλοι που εργάζονται στη συγκεκριμένη εργασία. Με τον απλό τρόπο διαίρεσης των ανθρωπομέρων δια του αριθμού των υπαλλήλων και την υπόθεση ότι η εργασία μπορεί να διαιρεθεί και να ανατεθεί σε όσα άτομα αναλάβουν τη εκτέλεση της, μπορεί να γίνει γνωστό ποια θα είναι η διάρκεια της εργασίας αυτής. Για παράδειγμα σε μια εργασία αρχικής διάρκειας 9 ανθρωπομέρων στην οποία εργάζονται τρεις υπάλληλοι θα έχουμε αποτέλεσμα διάρκεια 3 μέρες.

### Υλοποίηση έμψυχου υλικού:

Για να μεγιστοποιήσω και συνεπώς να βελτιστοποιήσω την υλοποίηση του έμψυχου υλικού δεν επιτρέπω σε ένα υπάλληλο σε οποιαδήποτε κατηγορία και να ανήκει αυτός να δουλεύει σε περισσότερες από μία εργασίες στο ίδιο έργο. Γι' αυτό εξάλλου υπάρχει και η

υπόθεσή μου ότι ο αριθμός των υπαλλήλων κάθε κατηγορίας είναι αρκετός για να καλύψει με τουλάχιστον ένα ανά εργασία της ίδιας κατηγορίας.

ΔΙΑΣΤΑΥΡΩΣΗ:

Η διασταύρωση στον Γενετικό μου αλγόριθμο γίνεται με τον τρόπο που περιέγραψα και στο τρίτο κεφάλαιο. Εφόσον γίνει η επιλογή των ικανότερων ατόμων από τον πληθυσμό κάθε γενιάς και έτσι δημιουργηθεί η δεξαμενή ζευγαρώματος μπορούμε να προχωρήσουμε στην παραγωγή μιας ακολουθίας τυχαίων αριθμών μεγέθους ίσου με το γενετικό πληθυσμό. Η πιθανότητα διασταύρωσης ορίζεται ως 0.2. Έτσι για όλους τους τυχαίους αριθμούς, που εννοείται ότι έχουν εύρος από 0 έως 1, συγκρίνουμε με την πιθανότητα διασταύρωσης και εφόσον ο τυχαίος αριθμός είναι μικρότερος τότε το αντίστοιχο άτομο του πληθυσμού της δεξαμενής διασταύρωσης θα συμμετέχει στο γενετικό τελεστή ως το ένα εκ των δύο του ζεύγους. Η διασταύρωση και πιο συγκεκριμένα η αποκοπή μπορεί να λάβει χώρα σε οποιοδήποτε σημείο της συμβολοσειράς του ατόμου. Μια πιθανότητα θα ήταν να αποκοπεί χωρίς να επηρεάσει εσωτερικά μια από τις συμβολοσειρές όπως φαίνεται στον Πίνακα 4-2.

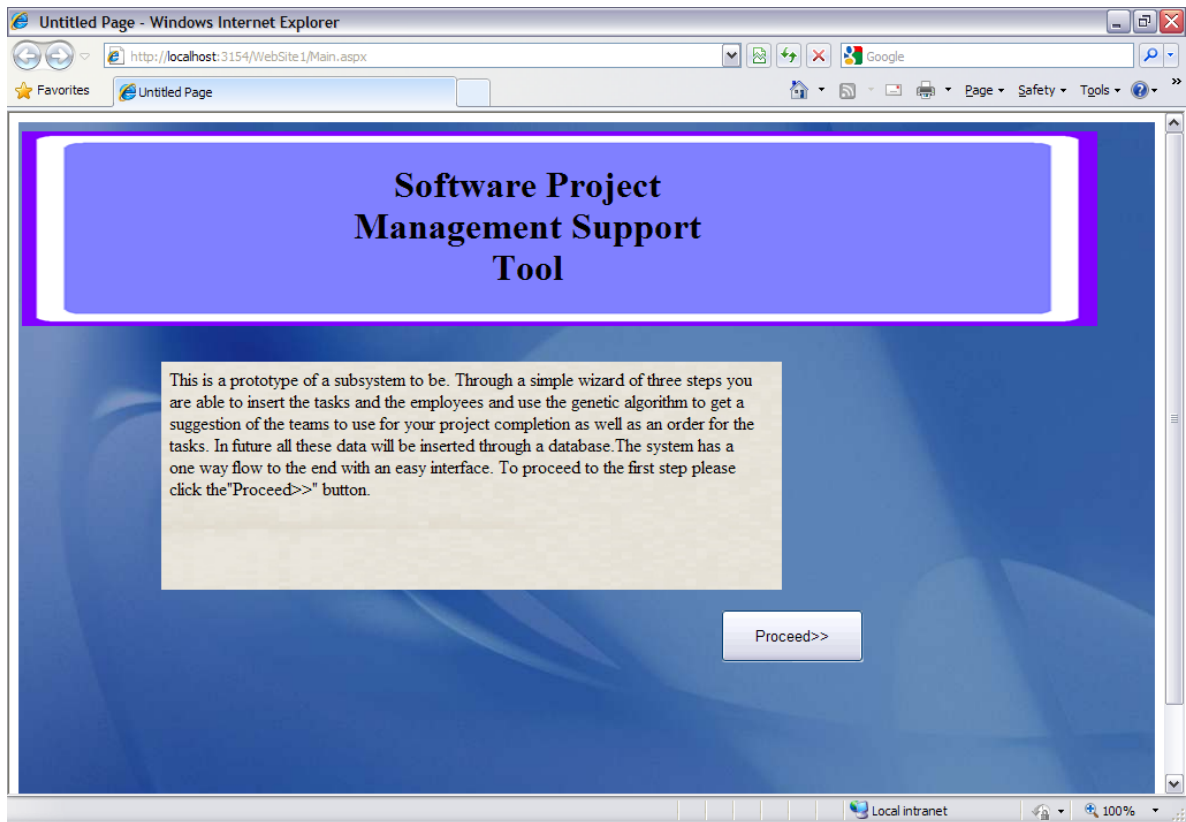
**Πίνακας 4-6**

TASK 1		TASK 2			TASK 3	
POSITION	PROGRAMMER 1	PROGRAMMER X	POSITION	DESIGNER 1	DESIGNER Y	ANALYSTS 1
TASK 1'		TASK 2'			TASK 3'	
POSITION	PROGRAMMER 1	PROGRAMMER R	POSITION	DESIGNER 1	DESIGNER T	ANALYSTS 1
TASK 1'		TASK 2'			TASK 3	
POSITION	PROGRAMMER 1	PROGRAMMER R	POSITION	DESIGNER 1	DESIGNER Y	ANALYSTS 1
TASK 1		TASK 2			TASK 3'	
POSITION	PROGRAMMER 1	PROGRAMMER X	POSITION	DESIGNER 1	DESIGNER T	ANALYSTS 1

Μια άλλη πιθανότητα θα ήταν να αλλοιωθεί μια συμβολοσειρά όπως παραδείγματος χάρη το Position του Task2 και εφόσον οι συμβολοσειρές δεν είναι ισομεγέθεις, ο προγραμματιστής X του Task 1. Φυσικά σε αυτή την περίπτωση πρέπει να ελέγξουμε κατά πόσον η αλλαγή αυτή έχει παραβιάσει κάποιο από τους κανόνες όπως να διαφύγει από το πεδίο ορισμού της συγκεκριμένης κατηγορίας (π.χ. προγραμματιστές) ή αν επαναλαμβάνεται κάπου αλλού. Σε αυτή την περίπτωση το νεοπαραχθέν αφήνεται στην καινούρια του τιμή και αλλάζεται το ήδη υπάρχον με κάποια άλλη διαθέσιμη τυχαία τιμή. Να σημειώσω εδώ ότι όπου παράγεται κάποια τυχαία τιμή αυτό γίνεται μέσα στο επιτρεπτό πεδίο ελέγχοντας κατά πόσο είναι διαθέσιμη. Λόγω της τόσο συχνής χρήσης τυχαίων τιμών απέφυγα να χρησιμοποιήσω την μετάλλαξη η οποία εξάλλου δεν είναι και τόσο σημαντική λειτουργία.

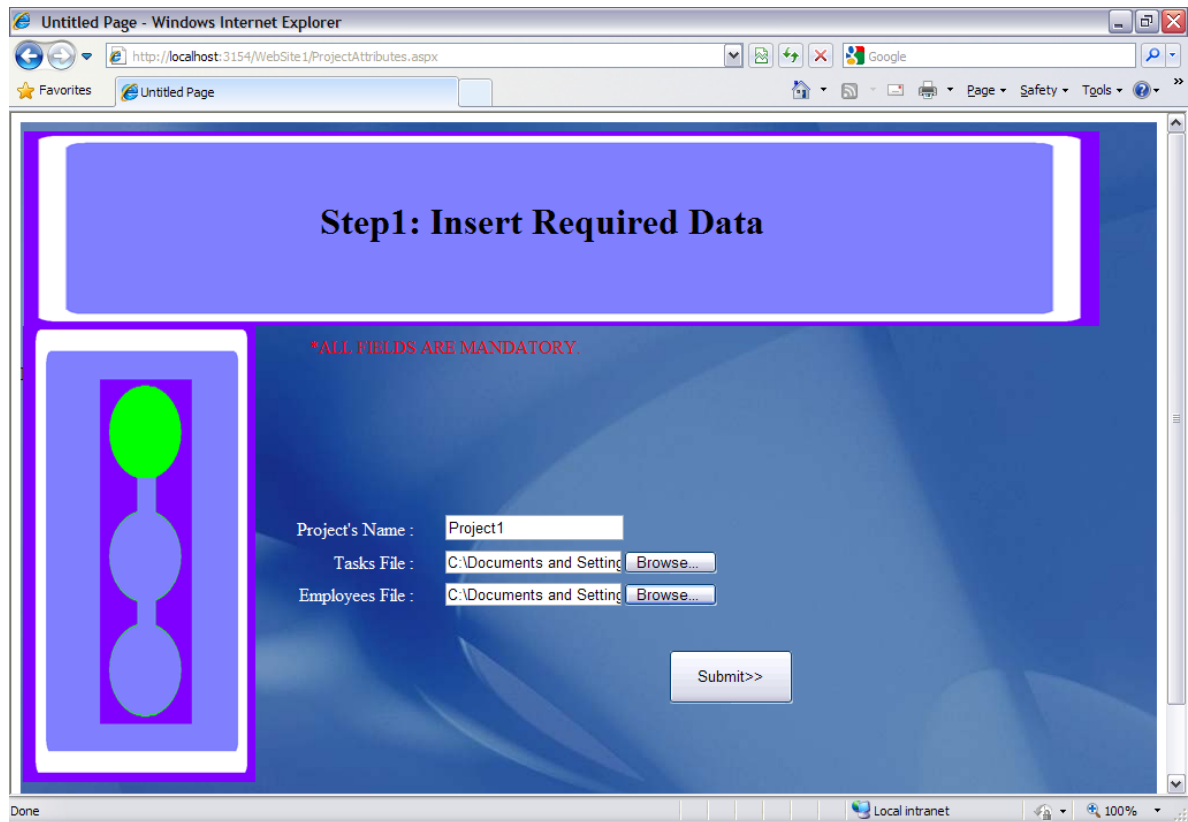
#### **4.2 Υλοποίηση του πρωτοτύπου**

Το πρωτότυπο υλοποιήθηκε σε περιβάλλον Microsoft Visual Studio 2005. Είναι μια εφαρμογή Διαδικτύου γραμμένη σε C# πάνω από ASP.NET. Είναι μια πολύ απλή εφαρμογή τεσσάρων ακολουθιακών σελίδων. Η πρώτη σελίδα είναι ένα απλό καλωσόρισμα του χρήστη στο σύστημα. Για να προχωρήσει ο χρήστης στην επόμενη σελίδα πρέπει να πατήσει το πλήκτρο “Proceed>>”.



**Figure4-1** Οθόνη καλωσορίσματος

Η δεύτερη η οποία δεν θα υπάρχει στο πλήρες σύστημα είναι μια φόρμα για εισαγωγή του ονόματος του έργου, του αρχείου που περιέχει τις εργασίες με όλα τα σχετικά στοιχεία και το αρχείο με τους διαθέσιμους υπαλλήλους και όλα τα σχετικά στοιχεία τους. Στο ολοκληρωμένο σύστημα αυτά τα δεδομένα θα εισάγονται μέσα στο υποσύστημα μέσα από μια βάση δεδομένων που θα περιέχονται όλα αυτά τα στοιχεία. Για αυτό το λόγο δεν υπάρχουν έλεγχοι στα αρχεία αφού η ακεραιότητα των τιμών θα είναι υπ' ευθύνη της βάσης. Έτσι αφού συμπληρώσει τα απαραίτητα πεδία πατώντας "Submit>>" ο χρήστης μπορεί να προχωρήσει στην επόμενη σελίδα.



**Figure 4-2 Βήμα ένα: Εισαγωγή Δεδομένων**

Στην τρίτη οθόνη του συστήματος γίνεται παρουσίαση των δεδομένων που διαβάστηκαν από το πρόγραμμα μέσα από τα αρχεία που εισήχθησαν από τη προηγούμενη σελίδα. Η όλη δομή των σελίδων θυμίζει ένα wizard που οδηγεί στην δημιουργία ομάδων. Στην περίπτωση που συμφωνεί με τα δεδομένα που εισήχθησαν στο σύστημα μπορεί να προχωρήσει στην παραγωγή των ομάδων με την χρήση του γενετικού αλγόριθμου. Ο χρήστης δεν χρειάζεται να εισαγάγει καμιά περαιτέρω παράμετρο στο σύστημα πατώντας "Generate Suggestion>>".

The screenshot shows a web browser window titled 'Untitled Page - Windows Internet Explorer'. The address bar shows 'http://localhost:3154/WebSite1/ChosenData.aspx'. The main content area has a blue background with a white header box containing the text 'Step2: Project1 Contents'. Below the header, there is a vertical navigation bar on the left with a red circle icon. The main content area contains two tables and a button.

Name	ID	Role	Man Days
Task1	1	Programmer	10
Task2	2	Designer	3
Task3	3	Analyst	5
Task4	4	Tester	2
Task5	5	Database	4

Name	Surname	Role	Type
George	Georgiou	Programmer	1
Prokopis	Prokopiou	Designer	5
Christos	Christou	Programmer	6
Giannis	Ioannou	Designer	13
Demetris	Demetriou	Analyst	10
Maria	Mpakodimou	Analyst	2
Eleni	Andreou	Tester	6
Kostas	Kostantinou	Tester	7
Marios	Andreou	Database	9
Mihalis	Mihail	Database	14

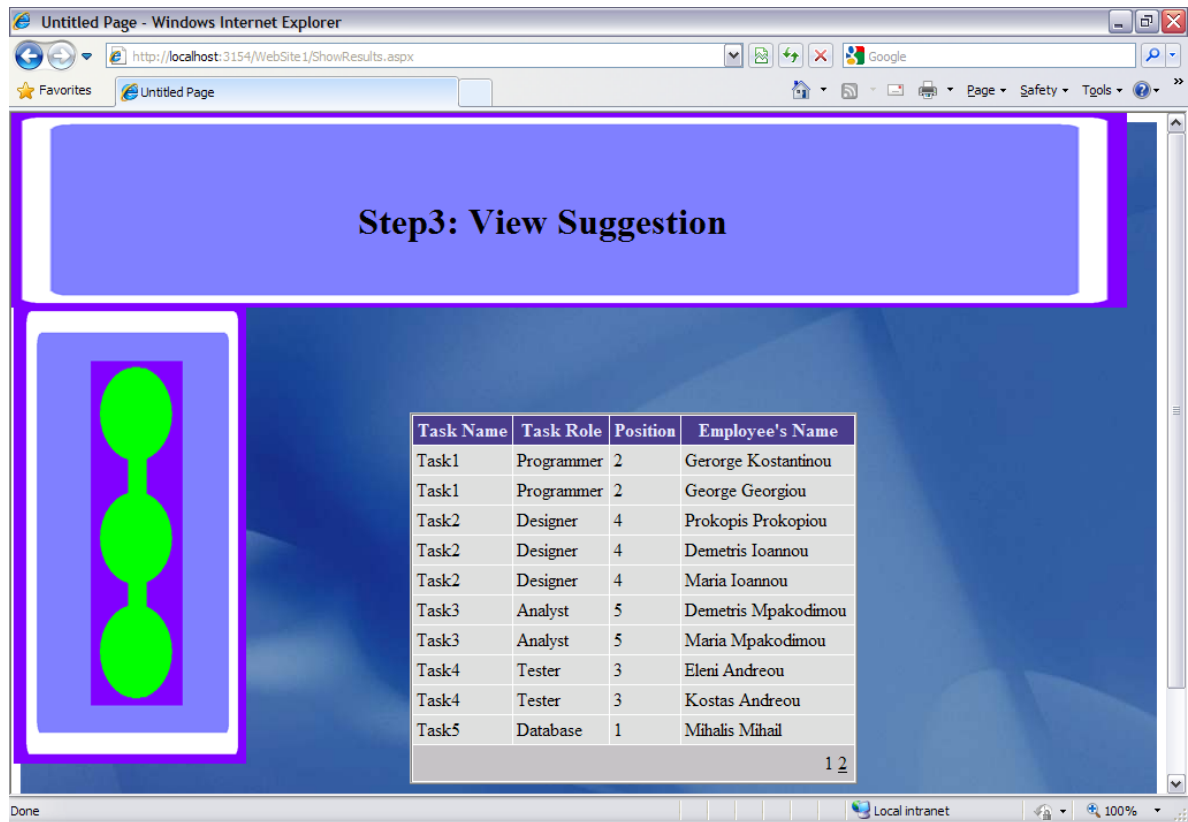
1 2 3

Generate Suggestion>>

**Figure 4-3 Βήμα 2: Παρουσίαση Εισαχθέντων Δεδομένων**

Έτσι με το τέλος της εκτέλεσης του αλγόριθμου ο χρήστης οδηγείται στην τελευταία σελίδα όπου του παρουσιάζεται το αποτέλεσμα της εκτέλεσης του Γενετικού Αλγόριθμου.





**Figure 4-4 Βήμα 3: Παρουσίαση αποτελεσμάτων**

Πέραν των κλάσεων που παράχθηκαν αυτόματα με την δημιουργία των σελίδων. Αφού κάθε σελίδα αντιπροσωπεύει μια κλάση. Για την πιο εύκολη αλλά παράλληλα και πιο δομημένη υλοποίηση του συστήματος δημιούργησα τρεις νέες κλάσεις, οι οποίες είναι οι εξής:

Public Class Employee:

Είναι μια κλάση που είναι υπεύθυνη για την δημιουργία αντικειμένων με τα χαρακτηριστικά του υπαλλήλου όπως ήδη τα όρισα στο προηγούμενο υποκεφάλαιο συν ακόμα μερικά επιπλέον. Συνολικά τα χαρακτηριστικά της κλάσης αυτής είναι το όνομα και το επίθετο του υπαλλήλου, ο ρόλος του στην εταιρεία, ο ψυχολογικός του τύπος μεταποιημένος σε αριθμό όπως ορίστηκε ήδη, και όλες οι βαθμολογίες των ικανοτήτων τις οποίες ήταν θεμιτό να εισαχθούν.

### Public Class Task:

Αυτή η κλάση σκοπό έχει να παράγει αντικείμενα τύπου εργασίας και πάλι όπως ορίστηκε στο προηγούμενο υποκεφάλαιο. Τα χαρακτηριστικά που περιγράφουν ένα αντικείμενο αυτής τη κλάσης είναι το όνομα της εργασίας, το είδος της (σχεδιαστική, προγραμματιστική κτλ), η διάρκεια της σε ανθρωπομέρες, μια μοναδική ταυτότητα που βασίζεται στην σειρά με την οποία εισήχθηκε μια συγκεκριμένη εργασία στο σύστημα, οι εξαρτήσεις που έχει η εργασία αυτή και το βάρος κάθε ικανότητας που χρειάζεται για να υλοποιηθεί η συγκεκριμένη εργασία.

### Public Class Bintransformer:

Αυτή η κλάση σκοπό δεν έχει την δημιουργία αντικειμένων, αλλά την παροχή των δύο στατικών μεθόδων που εμπεριέχει. Η πρώτη μέθοδος `bincode` με παραμέτρους δύο ακέραιους παίρνει τον πρώτο αριθμό και τον μετατρέπει σε μια δυαδική συμβολοσειρά μήκους ίσου με τον δεύτερο αριθμό την οποία και επιστρέφει. Η δεύτερη `bindcode` παίρνει σαν παράμετρο μια δυαδική συμβολοσειρά και επιστρέφει τον ακέραιο που κωδικοποιεί η συγκεκριμένη συμβολοσειρά.

Όλες οι διεργασίες σε σχέση με την υλοποίηση του γενετικού αλγόριθμου περιέχονται στην κλάση της τρίτης σελίδας. Ότι αφορά την αρχικοποίηση του αρχικού πληθυσμού και τον κύριο βρόχο γίνεται στην μέθοδο του πατήματος του κουμπιού `Generate Teams`. Όμως για την πιο δομημένη μορφή του κώδικα μου επέκτεινα την διαπροσωπεία της συγκεκριμένης κλάσης με τρεις μεθόδους σχετικές με τον γενετικό αλγόριθμο.

### fitnessFunction:

Η μέθοδος αυτή παίρνοντας ως παραμέτρους μια δυαδική συμβολοσειρά, ένα πίνακα με τις θέσεις των εργασιών πάνω στη συμβολοσειρά, το μέγεθος του δομικού στοιχείου της συμβολοσειράς και τις δομές δεδομένων με τους υπαλλήλους κάθε ρόλου και τις εργασίες

Αποκωδικοποιεί και αξιολογεί εφαρμόζοντας την συνάρτηση αξιολόγησης πιο πάνω και επιστρέφει ένα πραγματικό αριθμό που δεικνύει αυτή την ικανότητα.

#### Crossover:

Εδώ εκτελείται η άλλη σημαντική διεργασία του γενετικού αλγορίθμου η διασταύρωση. Παίρνοντας την δομή δεδομένων του πληθυσμού της δεξαμενής ζευγαρώματος, τον πίνακα με τις θέσεις των εργασιών για κάθε ένα από τα άτομα του πληθυσμού, το μέγεθος του δομικού στοιχείου της συμβολοσειράς, την δομή δεδομένων των εργασιών και το μέγεθος των δομών κάθε κατηγορίας των υπαλλήλων υπολογίζει ποιες συμβολοσειρές θα εκτελέσουν διασταύρωση και για κάθε ζευγάρι εκτελεί την διασταύρωση και καλεί την μέθοδο ελέγχου για κάθε νέα συμβολοσειρά για να βεβαιωθεί ότι είναι έγκυρη. Επιστρέφει τον πληθυσμό της νέας γενιάς.

#### Check:

Ο πολύ σημαντικός έλεγχος εγκυρότητας μιας συμβολοσειράς εκτελείται εδώ. Παίρνοντας μια συμβολοσειρά, , ένα πίνακα με τις θέσεις των εργασιών πάνω στη συμβολοσειρά, το μέγεθος του δομικού στοιχείου της συμβολοσειράς την δομή δεδομένων των εργασιών και το μέγεθος των δομών κάθε κατηγορίας των υπαλλήλων ελέγχει αν η συγκεκριμένη συμβολοσειρά είναι έγκυρη, αν ναι επιστρέφεται η ίδια συμβολοσειρά σε διαφορετική περίπτωση διορθώνονται όλες οι παραβιάσεις με τον τρόπο που ήδη περιέγραψα.

### **4.3 Αποτελέσματα**

Θέλοντας επαλήθευση των αποτελεσμάτων το πρωτότυπο μετά την υλοποίηση του δοκιμάστηκε με απλά σενάρια δοκιμής για έλεγχο των αποτελεσμάτων. Ακολουθούν τρία σενάρια δοκιμής:

#### Σενάριο 1:

Το σενάριο αυτό αναφέρεται σε ένα έργο αποτελείται από μια εργασία με 4 υπαλλήλους.

**Πίνακας 4-7**

Εργασίες	Τύπος Εργασίας	Διάρκεια σε ανθρωπομέρες	Βάρος Ικανότητας 1	Βάρος Ικανότητας 2	Βάρος Ικανότητας 3
Εργασία 1	Προγραμματιστική	10	2/10	5/10	3/10

**Πίνακας 4-8**

Υπάλληλοι	Ρόλος Υπαλλήλου	Τύπος MBTI	Βαθμολογία Ικανότητας 1	Βαθμολογία Ικανότητας 2	Βαθμολογία Ικανότητας 3
George Georgiou	Προγραμματιστής	1	9/10	9/10	9/10
Prokopis Prokopiou	Προγραμματιστής	1	9/10	9/10	9/10
Christos Christou	Προγραμματιστής	8	1/10	1/10	1/10
Giannis Ioannou	Προγραμματιστής	8	1/10	1/10	1/10

Θα περιμέναμε να τοποθετηθούν όλοι οι υπάλληλοι για την μείωση της διάρκειας της εργασίας, όμως λόγω ακριβώς του ότι η διάρκεια συνυπάρχει με τους άλλους παράγοντες όπως η αξιολόγηση στις ικανότητες και ο MBTI τύπος των υπαλλήλων παίρνουμε το εξής αποτέλεσμα.



**Figure 4-5** Αποτέλεσμα πρώτου σεναρίου

### Σενάριο 2:

Το σενάριο αυτό δοκιμάζει και την ικανότητα του συστήματος στην σωστή τοποθέτηση των εργασιών πέρα από την ανάθεση των εργασιών στους υπαλλήλους. Αποτελείται από 2 εργασίες δύο διαφορετικών τύπων και 2 υπαλλήλους για κάθε κατηγορία. Επίσης χάριν του παραδείγματος η πρώτη εργασία εξαρτάται από την δεύτερη.

**Πίνακας 4-9**

Εργασίες	Τύπος Εργασίας	Διάρκεια σε ανθρωπομέρες	Βάρος Ικανότητας 1	Βάρος Ικανότητας 2	Βάρος Ικανότητας 3
Εργασία 1	Προγραμματιστική	10	2	5	3
Εργασία 2	Σχεδιαστική	3	6	3	1

**Πίνακας 4-10**

Υπάλληλοι	Ρόλος Υπαλλήλου	Τύπος MBTI	Βαθμολογία Ικανότητας 1	Βαθμολογία Ικανότητας 2	Βαθμολογία Ικανότητας 3
George Georgiou	Προγραμματιστής	1	9/10	9/10	9/10
Prokopis Prokopiou	Προγραμματιστής	1	9/10	9/10	9/10
Maria Ioannou	Σχεδιαστής	16	9/10	9/10	9/10
Demetris Ioannou	Σχεδιαστής	16	9/10	9/10	9/10

Εφόσον έχουμε πανομοιότυπους υπαλλήλους τόσο στους MBTI τύπους όσο και στην αξιολόγηση ικανοτήτων θα περιμέναμε να γίνει μια πλήρης χρήση των υπαλλήλων με τον περιορισμό όμως ότι η δεύτερη εργασία θα προηγείται της πρώτης.



**Figure 4-6** Αποτελέσματα Δεύτερου σεναρίου

### Σενάριο 3:

Το συγκεκριμένο σενάριο είναι πιο πολύπλοκο από τα προηγούμενα δύο και σκοπό έχει να δείξει ότι το σύστημα μπορεί να χρησιμοποιηθεί και σε πιο πολύπλοκες περιπτώσεις.

**Πίνακας 4-11**

Εργασίες	Τύπος Εργασίας	Διάρκεια σε ανθρωπομέρες	Βάρος Ικανότητας 1	Βάρος Ικανότητας 2	Βάρος Ικανότητας 3	Εξαρτήσεις
Εργασία 1	Προγραμματιστική	10	2	5	3	2,3
Εργασία 2	Σχεδιαστική	3	6	3	1	
Εργασία 3	Ανάλυσης Συστημάτων	5	3	5	2	2
Εργασία 4	Ελεγκτική	2	6	2	2	1
Εργασία 5	Σχετική με Βάσεις Δεδομένων	4	4	4	2	2

Οι υπάλληλοι λόγω αυξημένου αριθμού παρουσιάζονται στις εικόνες που ακολουθούν.



Name	Surname	Role	Type
George	Georgiou	Programmer	1
Prokopis	Prokopiou	Designer	5
Christos	Christou	Programmer	6
Giannis	Ioannou	Designer	13
Demetris	Demetriou	Analyst	10
Maria	Mpakodimou	Analyst	2
Eleni	Andreou	Tester	6
Kostas	Kostantinou	Tester	7
Marios	Andreou	Database	9
Mihalis	Mihail	Database	14
			1 2 3

Name	Surname	Role	Type
Maria	Ioannou	Designer	16
Eleni	Demetriou	Analyst	12
Giannis	Mpakodimou	Programmer	1
Demetris	Andreou	Tester	5
Mihalis	Kostantinou	Programmer	8
Kostas	Andreou	Database	13
Marios	Mihail	Database	2
Maria	Demetriou	Analyst	8
Demetris	Mpakodimou	Analyst	6
Kostas	Andreou	Tester	9
			1 2 3

Name	Surname	Role	Type
Eleni	Kostantinou	Tester	7
Mihalis	Andreou	Database	9
Marios	Mihail	Database	14
Demetris	Ioannou	Designer	16
Giannis	Demetriou	Analyst	11
Eleni	Mpakodimou	Programmer	15
Maria	Andreou	Tester	5
Gerorge	Kostantinou	Programmer	8
			1 2 3

Figures 4-7 Υπάλληλοι του τρίτου σεναρίου

Σαν αποτέλεσμα θα περιμέναμε να διατηρούνται φυσικά οι εξαρτήσεις ενώ παράλληλα να γίνεται σωστή δημιουργία των ομάδων. Βλέπουμε όμως κάποιες από τις εξαρτήσεις να παραβιάζονται λόγω ακριβώς της έλλειψης των βαρών από την συνάρτηση αξιολόγησης που θα πολλαπλασιάζονται με κάθε μέγεθος δίνοντας τους την ανάλογη σημασία.

The screenshot shows a web browser window titled 'Untitled Page - Windows Internet Explorer'. The address bar shows 'http://localhost:3154/WebSite1/ShowResults.aspx'. The main content area has a blue header with the text 'Step3: View Suggestion'. Below the header, there is a vertical stack of three red circles connected by lines, representing task dependencies. To the right of this stack is a table with the following data:

Task Name	Task Role	Position	Employee's Name
Task1	Programmer	3	George Kostantinou
Task1	Programmer	3	Mihalis Kostantinou
Task1	Programmer	3	Christos Christou
Task2	Designer	2	Demetris Ioannou
Task2	Designer	2	Maria Ioannou
Task3	Analyst	5	Maria Mpakodimou
Task3	Analyst	5	Demetris Mpakodimou
Task4	Tester	4	Eleni Kostantinou
Task4	Tester	4	Kostas Andreou
Task4	Tester	4	Eleni Andreou

The browser's status bar at the bottom shows 'Local intranet' and '100%' zoom level.

Figure 4-8 Αποτελέσματα τρίτου σεναρίου 1



**Figure 4-9** Αποτελέσματα τρίτου σεναρίου 2

Το όλο σύστημα επιδέχεται βελτιώσεων και ολοκλήρωσης πριν ενσωματωθεί στο τελικό ολοκληρωμένο σύστημα. Για να γίνουν όμως αυτές οι βελτιώσεις χρειάζεται πειραματισμός σε ελεγχόμενο περιβάλλον για στάθμιση των μεγεθών που χρησιμοποιούνται αλλά και επαλήθευση των αποτελεσμάτων. Μετατροπές μπορούν να γίνουν σε παράλληλα συστήματα για την χρήση παρόμοιων μεθοδολογιών και αλγορίθμων όπως π.χ., του Particle Swarm Optimization αλγόριθμου, για σύγκριση και εύρεση της πιο χρήσιμης μεθοδολογίας.

# Κεφάλαιο 5

## Συμπεράσματα και Μελλοντική Εργασία

---

---

Η εργασία αυτή πραγματεύεται ένα πολύ σημαντικό πρόβλημα της περιοχής της Τεχνολογίας Λογισμικού, το ανεξήγητα ακόμα αυξημένο ποσοστό αποτυχίας των έργων κατά την αύξηση πολυπλοκότητας και μεγέθους των έργων αυτών. Μελετώντας ανακαλύφθηκε ότι ένας από τους σημαντικούς παράγοντες που επηρεάζουν το ποσοστό αποτυχίας των έργων παραγωγής λογισμικού είναι η κακή διαχείριση των έργων όσον αφορά την ανάθεση εργασιών, τον σχηματισμό ομάδων και τον στρατηγικό σχεδιασμό ανάπτυξης του έργου. Έτσι προτείνεται ένα σύστημα σχηματισμού λειτουργικών ομάδων εργασίας για την βελτιστοποίηση της παραγωγής έργων λογισμικού. Στα πρώτα κεφάλαια παρουσιάζεται το MBTI, ο κύκλος επικοινωνίας του Thompson και οι γενετικοί αλγόριθμοι, τα συστατικά στοιχεία τα οποία περιέχει το σύστημα. Έπειτα παρουσιάζεται η μοντελοποίηση του συστήματος, ένα υλοποιημένο πρωτότυπο και κάποια αρχικά αποτελέσματα.

Η διπλωματική εργασία πέτυχε τον αρχικό της στόχο να δημιουργήσει ένα πρωτότυπο σύστημα που να λαμβάνει υπόψη της προσωπικότητες και τις ικανότητες των ατόμων για τον σχηματισμό ομάδων και την ανάθεση εργασιών ενώ παράλληλα να γίνεται ο βέλτιστος στρατηγικός σχεδιασμός ανάπτυξης του έργου. Φυσικά αυτά σε ένα πολύ αρχικό στάδιο

αφού δεν έχει γίνει ο αναγκαίος πειραματισμός τόσο σε ελεγχόμενο ούτε σε ανεξέλεγκτο περιβάλλον για σύγκριση αποτελεσμάτων, καθώς επίσης το σύστημα μπορεί να χρήζει μετατροπές ή και βελτιώσεις για να φτάσει την τελική του μορφή.

Αναμφισβήτητα έχουν αποκομιστεί πολλά πράγματα από την συγκεκριμένη εργασία. Επετεύχθη να συμπεριληφθεί η ανθρώπινη πτυχή στον χώρο της δέσμευσης ανθρωπίνων πόρων που μέχρι τώρα αντιμετωπιζόνταν όπως και τους άψυχους πόρους. Εδείχθη έτσι ότι η Ψυχολογία και η Κοινωνιολογία μπορεί και πρέπει να παίζει πολύ σημαντικό ρόλο στο μέλλον τόσο του τομέα των Συστημάτων Υποστήριξης Διευθυντικών Αποφάσεων αλλά και γενικότερα στον χώρο της Πληροφορικής αφού πολλές είναι οι εφαρμογές που θα μπορούσαν να εισαγάγουν θεωρίες από αυτές τις Επιστήμες.

Συνεχίζοντας αναγνωρίστηκε πόσο πολύπλευρη είναι η δουλειά ενός Διευθύνοντα έργου και με πόσο φόρτο Δεδομένων μπορεί να έρχεται αντιμέτωπος ένας άνθρωπος με αυτό το ρόλο. Για αυτό το λόγο άλλωστε γίνεται και η τόσο μεγάλη προσπάθεια υποστήριξης τους με τις συνεχείς μεθοδεύσεις των εργασιών που επιτελούν αλλά και με την συνεχή ανάπτυξη συστημάτων που πρέπει να χρησιμοποιούν για να αντεπεξέλθουν από τα συνεχή εμπόδια που παραθέτει η φύση της δουλειάς τους.

Ακόμα με την Διπλωματική εργασία χρησιμοποιήθηκε ένα πολύ δυνατό εργαλείο επίλυσης προβλημάτων οι γενετικοί αλγόριθμοι καθώς και άλλες παρόμοιες θεωρίες. Οι απεριόριστες δυνατότητες που παρέχουν στα χέρια ενός επιστήμονα Πληροφορικής τα καθιστούν ένα πολύτιμο εφόδιο για τις μετέπειτα μελέτες . Έγινε δυνατό να μοντελοποιηθεί ένα πρόβλημα από ένα τελείως αρχικό σημείο με την καθοδήγηση φυσικά του επιβλέποντα καθηγητή και του βοηθού του. Αυτή η συνεχής τριβή έδωσε την αναγκαία αυτοπεποίθηση τόσο στην επίλυση προβλημάτων αλλά και στην χρήση τέτοιων συστημάτων τεχνητής νοημοσύνης. Πιστεύω ότι αυτά τα συστήματα καθώς και άλλα παρόμοια που ίσως ανακαλυφθούν θα αποτελέσουν το μέλλον της Πληροφορικής.

Αναπολώντας την διάρκεια της Διπλωματικής μου εργασίας βλέπω πόσο με έχει βοηθήσει σαν επικείμενο επιστήμονα Πληροφορικής αλλά και σαν σωστό επαγγελματία. Πέρα από τις γνώσεις που μου προσέφερε η μελέτη τόσων επιστημονικών άρθρων και βιβλίων απέκτησα μια νέα νοοτροπία απέναντι στην τρόπο μελέτης αλλά και στην αναπαραγωγή των όσων έχω μάθει. Βλέπω τώρα με μεγαλύτερο δέος τις ερευνητικές επιτυχίες των εκάστοτε ερευνητών. Ήταν ένα πολύτιμο εφόδιο πέρα από αυτά που μου έχει ήδη προσφέρει το Πανεπιστήμιο Κύπρου το οποίο διαμόρφωσε βαθιά τον χαρακτήρα και τον τρόπο σκέψης μου. Πέτυχε να ανοίξει το μυαλό μου κάνοντας με να κατανοήσω πως και εγώ μπορώ να είμαι ένας από τους ερευνητές που με το έργο μου μπορώ να προσφέρω στον χώρο της Πληροφορικής και συνεπώς στην ανθρωπότητα.

Σαν μελλοντική δουλειά που θα γίνει, προβλέπεται η ενσωμάτωση του συστήματος στο ευρύτερο ολοκληρωμένο σύστημα που θα υλοποιηθεί. Πριν όμως γίνει αυτό καλό θα ήταν ακόμα να χρησιμοποιηθούν κάποιες εναλλακτικές θεωρίες και συστατικά στοιχεία όπως το Particle Swarm Optimization καθώς και άλλα για σύγκριση των αποτελεσμάτων, ή και να γίνει κάποια βελτιστοποίηση σε αυτές που ήδη χρησιμοποιούνται όπως π.χ., κάποιοι πιο εξελιγμένοι τρόποι διασταύρωσης και μετάλλαξης στους γενετικούς αλγόριθμους όπως uniform ή non-uniform καθώς και άλλοι προτεινόμενοι τρόποι. Ακόμα θα έπρεπε να γίνει κάποια επαλήθευση των παραγομένων αποτελεσμάτων μέσα σε κάποιο ελεγχόμενο περιβάλλον όπως το Πανεπιστήμιο Κύπρου και πιο συγκεκριμένα το μάθημα Τεχνολογίας Λογισμικού. Μέσα στα βραχυπρόθεσμα σχέδια της δουλειάς αυτής είναι η δημοσίευση της Ατομικής Διπλωματικής Εργασίας σε κάποιο επιστημονικό περιοδικό.

## Βιβλιογραφία

- [1] Σ. Λυκοθανάσης, “Γενετικοί Αλγόριθμοι και Εφαρμογές -Τεχνητή Νοημοσύνη Εφαρμογές,” Τομος Γ', Πάτρα,2001.
- [2] L.F. Capretz, “Personality types in software engineering,” International Journal of Human – Computer Studies., vol. 58, pp. 207-214, 2003.
- [3] E. Hirsh, K.W. Hirsh and S.K Hirsh, “Introduction to Type and Teams”, CPP, Inc., California, 2003.
- [4] H.L. Thompson, “Introduction to the CommunicationWheel,“ Georgia , 2000.
- [5] Sönke Hartmann, A Competitive Genetic Algorithm for Resource-Constrained Project Scheduling
- [6] E. Alba, F. Chicano , Software Project Management with Gas, Information Sciences, Vol.177, pp 2380-2401, 2007.
- [7] S. T. Acuna, N. Juristo, A. M. Moreno, Emphasizing Human Capabilities in Software Development
- [8] J.S. Karn, S.S Abdullah, A.J Cowling and M. Holcombe “A Study into the effects of personality type and methodology on cohesion in software engineering teams”, Vol.26, No2, pp. 99-111, 2007
- [9] E. W. Connie, “Teaching Teamwork in Information Systems”, Idea Group Publishing. USA 2002.

- [10] N. Gorla and Y.W. Lam, "Who Should Work with Whom? BUILDING EFFECTIVE SOFTWARE PROJECT TEAMS" Communications of the acm, Vol. 47, No. 6,pp 79-82, 2004
- [11] J. Tucker "Type & Project Management: Putting Type on the Critical Path" Otto Kroeger Associates.
- [12] A. Howard, "Software Engineering Project Management", COMMUNICATIONS OF THE ACM, May 2001/Vol. 44, No. 5, pp. 23-24.
- [13] E. Kaluzniacky "Managing Psychological Factors in Information Systems Work: An Orientation to Emotional Intelligence" University of Winnipeg, Canada, 2004.
- [14] J.Tucker, H. Rutlege "Transforming Cultures: A New Approach to Assessing and Improving Technical Programs", **CROSSTALK** The Journal of Defense Software Engineering, January 2006.
- [15] A.R. Peslak, "The Impact of Personality on Information Technology", ACM 2006 California pp. 273-279.
- [16] V. PIETERSE, D.G. KOURIE and I.P. SONNEKUS," Software Engineering Team Diversity and Performance", Proceedings of SAICSIT 2006, pp. 180 –186.
- [17] A.B. Pyster, Richard H.Thayer "Software Engineering Project Management 20 Years Later", 07 4 0 - 7 4 5 9 October 2005 I E E E Computer society.
- [18] W.S. Humphrey, "The Team Software ProcessSM (TSPSM)", Carnegie Mellon University, November 2000.

## Παράρτημα Α

### Αρχείο Εισαγωγής Εργασιών:

Μια γραμμή κωδικοποιεί μια εργασία με την σειρά όνομα εργασίας, ταυτότητα εργασίας, διάρκεια εργασίας σε ανθρωπομέρες, βάρος αναγκαίων ικανοτήτων και εξαρτήσεις από εργασίες εφόσον υπάρχουν. Τα διαφορετικά στοιχεία διαχωρίζονται με τον χαρακτήρα '\*' ενώ τα όμοια με τον χαρακτήρα '|'. Ακολουθεί παράδειγμα αρχείου.

Task1\*1\*Programmer\*10\*2|5|3\*3|2

Task2\*2\*Designer\*3\*6|3|1\*

Task3\*3\*Analyst\*5\*3|5|2\*2

Task4\*4\*Tester\*2\*6|2|2\*1

Task5\*5\*Database\*4\*4|4|2\*2

### Αρχείο Εισαγωγής Υπαλλήλων:

Μια γραμμή κωδικοποιεί μια εργασία με την σειρά Όνομα, Επώνυμο, ρόλος στην εταιρεία, τύπος στο MBTI κωδικοποιημένος σε αριθμό από 1-16 με 1 τον τύπο ESFP και κινούμενοι δεξιόστροφα στον κύκλο επικοινωνίας του Thompson και μετά οι βαθμολογίες για τις ικανότητες. Τα διαφορετικά στοιχεία διαχωρίζονται με τον χαρακτήρα '\*' ενώ τα όμοια με τον χαρακτήρα '|'. Ακολουθεί παράδειγμα αρχείου.

Marios\*Andreou\*Database\*9\*8|7|8

Mihalis\*Mihail\*Database\*14\*7|9|2

Maria\*Ioannou\*Designer\*16\*9|2|6

Eleni\*Demetriou\*Analyst\*12\*2|9|7

Giannis\*Mpakodimou\*Programmer\*1\*3|8|4



## Κώδικας της συνάρτησης fitnessFunction:

```
/// <summary>
/// //////////////ipologismos a3iologismos sigkwkrimenou atomou
/// </summary>
/// <param name="s"></param>
/// <param name="pos"></param>
/// <param name="size"></param>
/// <returns> fitness number</returns>

private double fitnessFunction(string s, int[] pos, int
size, List<Tasks> jobs, List<Employee> programmers, List<Employee>
designers, List<Employee> analysts, List<Employee> testers,
List<Employee> database)
{

    char[] med;
    int sumskills=0;
    int sumcommunication = 0;
    char[] piece = new char[size];
    int[] order = new int[jobs.Count];
    double durationsum=0;
    int[][] task = new int[jobs.Count][];
    double[] preresult = new double[jobs.Count];
    double result=0;

    ///apokwdikopoiisi tou atomou
    for (int i = 0; i < pos.Length; i++)
    {
        if (i < pos.Length - 1)
        {
            med = new char[pos[i + 1] - pos[i]];
            s.CopyTo(pos[i], med, 0, pos[i + 1] - pos[i]);
        }
        else
        {
            med = new char[s.Length - pos[i]];
            s.CopyTo(pos[i], med, 0, s.Length - pos[i]);
        }
        task [i] = new int[med.Length / size];
        int j = 0;
        for (int k = 0; k < med.Length; k++)
        {
            piece[k % size] = med[k];
            if ((k % size) == (size - 1))
            {
                task[i][j] = Bintransformer.decode(new
string(piece));
                j++;
            }
        }
    }
}
```

```

    }

    //////////////////////////////////////////////////pros8etw ta omoia skills ka8e ipallilou sto gonidio
    kai meta pollaplasiazw
    //////////////////////////////////////////////////ka8e 8esi tou pinaka autou me to baros apo ti lista
    jobs kai briskw sinoliko a8roisma
    for (int i = 0; i < jobs.Count; i++)
    {

        int[] skillsum = new int[jobs[i].Skills.Length];
        for (int k = 0; k < skillsum.Length; k++)
            skillsum[k] = 0;

        sumskills = 0;

        for (int k = 1; k < task[i].Length; k++)
        {
            for (int j = 0; j < skillsum.Length; j++)
                if (jobs[i].Role == "Programmer")
                    skillsum[j] = skillsum[j] +
programmers[task[i][k]].Skills[j];
                else if (jobs[i].Role == "Designer")
                    skillsum[j] = skillsum[j] +
designers[task[i][k]].Skills[j];
                else if (jobs[i].Role == "Analyst")
                    skillsum[j] = skillsum[j] +
analysts[task[i][k]].Skills[j];
                else if (jobs[i].Role == "Tester")
                    skillsum[j] = skillsum[j] +
testers[task[i][k]].Skills[j];
                else if (jobs[i].Role == "Database" && task[i][k]>0
&& database.Count>task[i][k])
                    skillsum[j] = skillsum[j] +
database[task[i][k]].Skills[j];
            }

            for (int k = 0; k < skillsum.Length; k++)
                sumskills = sumskills + skillsum[k] * jobs[i].Skills[k];

            //////epikoinwnia
            sumcommunication = 0;
            if (task[i].Length > 2)
            {

                int temp;

                for (int k = 1; k < task[i].Length; k++)
                    for (int j = 1; j < task[i].Length; j++)
                        if (k > j)
                        {
                            if (jobs[i].Role == "Programmer")
                            {
                                temp =
Math.Abs(programmers[task[i][k]].MBTI - programmers[task[i][j]].MBTI);

```

```

        if (temp < 8)
            sumcommunication += temp;
        else
        {
            if (programmers[task[i][k]].MBTI <
programmers[task[i][j]].MBTI)
                sumcommunication += 16 +
programmers[task[i][k]].MBTI - programmers[task[i][j]].MBTI;
            else
                sumcommunication += 16 +
programmers[task[i][j]].MBTI - programmers[task[i][k]].MBTI;
        }
        }
        else if (jobs[i].Role == "Designer")
        {
            temp =
Math.Abs(designers[task[i][k]].MBTI - designers[task[i][j]].MBTI);
            if (temp < 8)
                sumcommunication += temp;
            else
            {
                if (designers[task[i][k]].MBTI <
designers[task[i][j]].MBTI)
                    sumcommunication += 16 +
designers[task[i][k]].MBTI - designers[task[i][j]].MBTI;
                else
                    sumcommunication += 16 +
designers[task[i][j]].MBTI - designers[task[i][k]].MBTI;
            }
        }
        else if (jobs[i].Role == "Analyst")
        {
            temp = Math.Abs(analysts[task[i][k]].MBTI
- analysts[task[i][j]].MBTI);
            if (temp < 8)
                sumcommunication += temp;
            else
            {
                if (analysts[task[i][k]].MBTI <
analysts[task[i][j]].MBTI)
                    sumcommunication += 16 +
analysts[task[i][k]].MBTI - analysts[task[i][j]].MBTI;
                else

```

```

sumcommunication += 16 +
analysts[task[i][j]].MBTI - analysts[task[i][k]].MBTI;
    }
}

else if (jobs[i].Role == "Tester")
{
temp = Math.Abs(testers[task[i][k]].MBTI
- testers[task[i][j]].MBTI);
if (temp < 8)

sumcommunication += temp;
else
{

if (testers[task[i][k]].MBTI <
testers[task[i][j]].MBTI)

sumcommunication += 16 +
testers[task[i][k]].MBTI - testers[task[i][j]].MBTI;
else

sumcommunication += 16 +
testers[task[i][j]].MBTI - testers[task[i][k]].MBTI;
}
}
else if(task[i][k]>=0&&
task[i][k]<database.Count && task[i][j]>=0 && task[i][j]<database.Count)
{

temp = Math.Abs(database[task[i][k]].MBTI
- database[task[i][j]].MBTI);
if (temp < 8)

sumcommunication += temp;
else
{

if (database[task[i][k]].MBTI <
database[task[i][j]].MBTI)

sumcommunication += 16 +
database[task[i][k]].MBTI - database[task[i][j]].MBTI;
else

sumcommunication += 16 +
database[task[i][j]].MBTI - database[task[i][k]].MBTI;
}
}
}
}
}

```

```

    }
    else
        sumcommunication = 0;
    //ipologismos diarkeias an8rwpomeres/no. ipallilwn
    if (task[i].Length - 1 > 0)
        durationsum += jobs[i].Mandays / (task[i].Length - 1);
    else
        durationsum += jobs[i].Mandays *3;

    //ipologismos ari8mou parabasis e3arthsewn
    int count = 0;
    if (jobs[i].Depend != null)
    {
        for (int k = 0; k < jobs[i].Depend.Length; k++)
            if (task[i][0] < task[jobs[i].Depend[k] - 1][0])
                count++;
    }
    //ipologismos ikanotitas gonidiou
    preresult[i] = sumskills / ((sumcommunication + 1) *
durationsum * ((count*10) + 1));

}

//ipologismos ikanotitas xrwmatoswmatos
for (int i = 0; i < preresult.Length; i++)
    result += preresult[i];

return result;
}

```

### Κώδικας της συνάρτησης crossover:

```

/// <summary>
/// diastaurwsi twn atomwn efoson i pi8anotita to epetrepse
/// </summary>
/// <param name="pool"></param>
/// <param name="pos"></param>
/// <param name="size"></param>
/// <param name="progs"></param>
/// <param name="analysts"></param>
/// <param name="testers"></param>
/// <param name="designers"></param>
/// <param name="databases"></param>
/// <returns></returns>

private string[] crossover(string[] pool, int[][] pos, int
size,List<Tasks> jobs, int progs, int anal, int tests, int design, int
databases)

```

```

{
    Random rand = new Random();
    double[] possibility = new double[pool.Length];
    int counter=0;

    //kathorismos pithanotitas diastaurwsis
    for (int i = 0; i < possibility.Length; i++) {

        possibility[i] = rand.NextDouble();
        if (possibility[i] < 0.2)
            counter++;
    }
    //se periptwsi pou den exoume diastaurwsi
    if (counter <= 1)
        return pool;
    // an exoume diastaurwsi
    else{
        // euresi diadoxikwn zeugariwn
        for (int i = 0; i < possibility.Length; i++)
        {
            if (possibility[i] < 0.2 && i < possibility.Length - 1)
            {
                int j = i + 1;

                while (j < possibility.Length && possibility[j] >
0.2)
                    j++;
                if (j >= possibility.Length)
                    break;
                //euresi simeiou diastaurwsis
                int temp = rand.Next(Math.Min(pool[i].Length,
pool[j].Length) -1)+1;

                //diastaurwsi twn dio string
                char[] buff = new char[temp];

                pool[i].CopyTo(0, buff, 0, temp);

                char[] one=pool[i].ToCharArray();
                char[] two = pool[j].ToCharArray();

                for (int k = 0; k < temp; k++)
                {
                    one[k] = two[k];
                    two[k] = buff[k];
                }

                //elegxos twn paragomenwn string kai dior8wsi tous
                pool[i] = check(new string(one), pos[i], size,jobs,
progs, analis, tests, design, databases);
                pool[j] = check(new string(two), pos[i], size,jobs,
progs, analis, tests, design, databases);
            }
        }
    }
}

```

```

    }
}
}
return pool;
}

```

### Κώδικας αρχικοποίησης του γενετικού πληθυσμού:

```

/////arxikopoiisi tou arxikou pli8ismou
for (int i = 0; i < 15; i++)
{
    atoms[i] = "";
    List<int> order = new List<int>();
    // kratw auta p xrisimopoiisa se listes g na min ta 3anabalw
    kai tin 8esi ka8e kainouriou job
    for (int j = 0; j < jobs.Count; j++)
    {

        List<int> usedProg = new List<int>();
        List<int> usedDes = new List<int>();
        List<int> usedAnal = new List<int>();
        List<int> usedTest = new List<int>();
        List<int> usedData = new List<int>();
        int temp = RandNum.Next(jobs.Count);

        while(order.Contains(temp))
            temp = RandNum.Next(jobs.Count);
        order.Add(temp);

        taskpos[i][j] = atoms[i].Length;

        atoms[i] = atoms[i] +
Bintransformer.bincodetemp, geneSize );

        if (jobs[j].Role == "Programmer")
        {

            for (int k = RandNum.Next(programmers.Count); k <
programmers.Count; k++)
            {

                int a= RandNum.Next(programmers.Count);

                if(!usedProg.Contains(a)){
                    atoms[i] = atoms[i] +
Bintransformer.bincodetemp, geneSize);

```

```

        usedProg.Add(a);
    }
}
else if (jobs[j].Role == "Designer")
{
    for (int k = RandNum.Next(designers.Count-1); k <
designers.Count; k++)
    {
        int a = RandNum.Next(designers.Count);
        if (!usedDes.Contains(a))
        {
            atoms[i] = atoms[i] +
Bintransformer.bincode(a, geneSize);
            usedDes.Add(a);
        }
    }
    else if (jobs[j].Role == "Analyst")
    {
        for (int k = RandNum.Next(analysts.Count-1); k <
analysts.Count; k++)
        {
            int a = RandNum.Next(analysts.Count);
            if (!usedAnal.Contains(a))
            {
                atoms[i] = atoms[i] +
Bintransformer.bincode(a, geneSize);
                usedAnal.Add(a);
            }
        }
    }
    else if (jobs[j].Role == "Tester")
    {
        for (int k = RandNum.Next(testers.Count-1); k <
testers.Count; k++)
        {
            int a = RandNum.Next(testers.Count);
            if (!usedTest.Contains(a))
            {
                atoms[i] = atoms[i] +
Bintransformer.bincode(a, geneSize);
                usedTest.Add(a);
            }
        }
    }
    else
    {
        for (int k = RandNum.Next(database.Count-1); k <
database.Count; k++)
        {
            int a = RandNum.Next(database.Count);

```



```
        if (!usedData.Contains(a))
        {
            atoms[i] = atoms[i] +
Bintransformer.bincode(a, geneSize);
            usedData.Add(a);
        }
    }
}
```