



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Αυτόματη διαδραστική σχεδίαση ενδυμάτων με χρήση  
εξελικτικών αλγορίθμων - Πρώτη προσέγγιση»



Του φοιτητή  
Θεοδωρίδη Ιωάννη  
Αρ. Μητρώου: 154450

Επιβλέπων  
Αδαμίδη Παναγιώτη  
Βαθμίδα: Καθηγητής

Θεσσαλονίκη, Σεπτέμβριος 2022

Τίτλος Π.Ε. Αυτόματη διαδραστική σχεδίαση ενδυμάτων με χρήση εξελικτικών αλγορίθμων

Κωδικός Π.Ε. 21106

Όνοματεπώνυμο φοιτητή Θεοδωρίδης Ιωάννης

Όνοματεπώνυμο εισηγητή Αδαμίδης Παναγιώτης

Ημερομηνία ανάληψης Π.Ε. Φεβρουάριος 2022

Ημερομηνία περάτωσης Π.Ε. Σεπτέμβριος 2022

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Θεοδωρίδη Ιωάννη που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητα και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

## Περίληψη

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι η δημιουργία μια εφαρμογής για την επίλυση του προβλήματος της σχεδίασης ενδυμάτων με την χρήση διαδραστικού εξελικτικού αλγορίθμου για ηλεκτρονικό υπολογιστή. Για την υλοποίηση της εφαρμογής επινοήθηκε ένας καινούργιος διαδραστικός εξελικτικός αλγόριθμος, ο οποίος εφαρμόστηκε στην σχεδίαση ενδυμάτων. Επίσης χρησιμοποιήθηκε και μια γραμματική σχημάτων για την αναπαράσταση των σχεδίων της εφαρμογής. Η εφαρμογή υλοποιήθηκε με την χρήση της γλώσσας προγραμματισμού Java και η διεπαφή της με την χρήση της JavaFX που χρησιμοποιεί FXML και CSS. Η ανάπτυξη της εφαρμογής έγινε στο πρόγραμμα Eclipse IDE και η διεπαφή σε Scene Builder. Η εφαρμογή αποσκοπεί στο να δίνει στον σχεδιαστή νέα και πρωτότυπα σχέδια μέσω μια διαδραστικής διαδικασίας μαζί του, που θα αποτελούν έμπνευση για την δημιουργία νέων σχεδίων ενδυμάτων.

Στα κεφάλαια που ακολουθούν αρχικά παρουσιάζονται ο εξελικτικός υπολογισμός (evolutionary computation) και η εξελικτική σχεδίαση. Στην συνέχεια παρουσιάζονται τα προβλήματα που προκύπτουν. Έπειτα παρουσιάζεται ο τρόπος προσέγγισης που εφαρμόστηκε στην εφαρμογή και γίνεται μια ξενάγηση της διεπαφής. Ακολουθούν δύο παραδείγματα εφαρμογής και τέλος τα συμπεράσματα και προτάσεις βελτιώσης.

# Automatic interactive clothing design using evolutionary algorithms – first approach

Theodoridis Ioannis

## **Abstract**

The subject of the thesis is the creation of a desktop application of clothing design using an interactive evolutionary algorithm. To implement the application, a new interactive evolutionary algorithm was devised. A schema grammar was also used to represent the application's designs. The application was implemented using the Java programming language and its interface using JavaFX which uses FXML and CSS. The development of the application was done using the Eclipse IDE and the GUI interface using the Scene Builder. The application aims to give the designer new and original designs through an interactive process with him, which will be inspiration for creating clothes.

In the chapters that follow, evolutionary computation and evolutionary design are first presented. Problems that arise are then presented. Then the approach applied to the application and a guided tour of the interface. Then, two application examples are presented. Finally, we have the conclusions and suggestions for improvements.

## **Ευχαριστίες**

Θεωρώ υποχρέωση μου να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Παναγιώτη Αδαμίδη για την πολύτιμη καθοδήγηση του, καθώς και την υπομονή που έδειξε. Επιπρόσθετα θέλω να ευχαριστήσω την οικογένεια μου που με στήριξε σε όλα τα χρόνια της φοίτησης μου.

## Περιεχόμενα

Περίληψη	iii
Abstract	iv
Ευχαριστίες	v
Περιεχόμενα	vi
Κατάλογος Σχημάτων	ix
Συντομογραφίες	xi
Κεφάλαιο 1ο: Εισαγωγή και Στόχοι	1
1.1 Εισαγωγή	1
1.2 Στόχος της Εργασίας	1
1.3 Συνοπτική Αναφορά Κεφαλαίων	2
1.4 Επίλογος	2
Κεφάλαιο 2ο: Εξελικτικοί Αλγόριθμοι	5
2.1 Εισαγωγή	5
2.2 Βιολογικό Υπόβαθρο	5
2.3 Μεθοδολογίες Εξελικτικών Αλγορίθμων	6
2.3.1 Γενετικοί Αλγόριθμοι	7
2.3.1.1 Πως λειτουργούν οι Γενετικοί Αλγόριθμοι;	7
2.3.2 Γενετικός Προγραμματισμός	9
2.3.2.1 Πως λειτουργεί ο Γενετικός Προγραμματισμός;	10
2.3.3 Εξελικτικές Στρατηγικές	12
2.3.3.1 Πως λειτουργούν οι Εξελικτικές Στρατηγικές;	12
2.3.4 Εξελικτικός Προγραμματισμός	14
2.3.4.1 Πως λειτουργεί ο Εξελικτικός Προγραμματισμός;	15
2.4 Διαδραστικοί Εξελικτικοί Αλγόριθμοι	16
2.4.1 Διαδραστικός Γενετικός Αλγόριθμος στην Σχεδίαση Ενδυμάτων	16
2.5 Επίλογος	20
Κεφάλαιο 3ο: Εξελικτική Σχεδίαση	22
3.1 Εισαγωγή	22
3.2 Εξελικτική Βελτιστοποίηση Σχημάτων	22
3.3 Δημιουργική Εξελικτική Σχεδίαση	24
3.3.1 Θεμελιώδης Εξελικτική Σχεδίαση	24
3.3.2 Παραγωγική Εξελικτική Σχεδίαση	26
3.4 Εξελικτική Τέχνη	26

3.5	Εξελκτικές Μορφές Τεχνητής Ζωής	29
3.7	Γραμματικές Σχημάτων	32
3.7.1	Γραμματικές Σχημάτων και Εξελκτικοί Αλγόριθμοι	33
3.7.1.1	Κώδικας Σχήματος	33
3.8	Επίλογος	34
	Κεφάλαιο 4ο: Παρουσίαση Προβλήματος	35
4.1	Εισαγωγή	35
4.2	Σχεδίαση Ενδυμάτων	36
4.2.1	Βήματα Σχεδίασης	36
4.3	Εξελκτική Προσέγγιση Στην Σχεδίαση	38
4.3.1	Εξελκτική Προσέγγιση Στην Σχεδίαση Ενδυμάτων	38
4.3.1.1	Διαδραστική Αξιολόγηση	39
4.4	Επίλογος	39
	Κεφάλαιο 5ο : Μοντέλο Προσέγγισης και Εφαρμογή	41
5.1	Εισαγωγή	41
5.2	Τεχνολογίες	41
5.2.1	Java	41
5.2.1.1	Πακέτα Java	41
5.2.2	JavaFX	42
5.2.2.1	Πακέτα JavaFX	43
5.2.3	CSS	43
5.2.4	FXML	44
5.3	Γραμματική Σχημάτων	44
5.4	Γενετικό υλικό	45
5.5	Διαδραστικός Εξελκτικός Αλγόριθμος Σχεδίασης Ενδυμάτων	46
5.5.1	Παραμετροποίηση Διαδραστικού Εξελκτικού Αλγορίθμου	47
5.6	Εξοικείωση Διεπαφής	54
5.6.1	CATEGORIES MENU	54
5.6.2	TAB MENU	55
5.6.2.1	HELP TAB	55
5.6.2.2	OPTIONS TAB	55
5.6.2.3	GENERATION TAB	55
5.7	Επίλογος	61
	Κεφάλαιο 6ο : Αποτελέσματα Εφαρμογής	63
6.1	Εισαγωγή	63

6.2	Εξέλιξη Δύο Σεναρίων	63
6.2.1	Σενάριο μπλούζας	65
6.2.2	Σενάριο Παντελονιού	68
6.3	Επίλογος	72
	Κεφάλαιο 7ο : Συμπεράσματα και Περαιτέρω Βελτιώσεις	74
7.1	Εισαγωγή	74
7.2	Συμπεράσματα	74
7.3	Βελτιώσεις	75
7.4	Επίλογος	75
	ΒΙΒΛΙΟΓΡΑΦΙΑ	77
	ΠΑΡΑΡΤΗΜΑ Α : Κώδικας Εφαρμογής	80
	ΠΑΡΑΡΤΗΜΑ Β : Κώδικας Εξελικτικών Διαδικασιών	88

## Κατάλογος Σχημάτων

Σχήμα 2. 1: Χαρτογράφηση χώρου αναζήτησης και χώρου λύσεων και τα αντίστοιχα περιεχόμενα τους (γονότυποι, φαινότυποι).	8
Σχήμα 2. 2: Τρεις τεχνικές διασταύρωσης που ορίζονται από το περιβάλλον.	8
Σχήμα 2. 3: Διαδικασία μετάλλαξης: Επιλογή αλληλίου και αλλαγή της τιμής του με τυχαία τιμή.	9
Σχήμα 2. 4: Δενδροειδής δομή ατόμου στον ΓΠ.	10
Σχήμα 2. 5: Δομή ατόμου στον ΓΠ. Διαδικασία μετάλλαξης και παραγωγή απογόνου.	11
Σχήμα 2. 6: Δομή ατόμου στον ΓΠ. Διαδικασία μετάλλαξης και παραγωγή απογόνου.	14
Σχήμα 2. 7: Η διαδικασία του αθλητικού τουρνουά κατά την τυχαία επιλογή πιθανών γονέων και η σύγκριση της ποιότητας τους.	15
Σχήμα 2. 8: Γενική λεπτομέρεια ενός γυναικείου φορέματος.	18
Σχήμα 2. 9: Κωδικοποιημένα στυλ φούστας και μέσης.	18
Σχήμα 2. 10: Η μορφή της δομής του γονιδίου που έχει 3 bits για σχέδιο και 3 bits για χρώμα στην σχεδίαση ρούχων με ΔΓΑ.	18
Σχήμα 2. 11: Τελεστής διασταύρωσης του ΔΓΑ στην σχεδίαση ρούχων.	20
Σχήμα 3. 1: Εφαρμογή εξελικτικής βελτιστοποίησης σε τραπέζι.	23
Σχήμα 3. 2: Εφαρμογή θεμελιώδης εξελικτικής σχεδίασης σε τραπέζι.	25
Σχήμα 3. 3: Εξελικτική Τέχνη εφαρμοσμένη για την παραγωγή εικόνων.	27
Σχήμα 3. 4: Εξέλιξη καλλιτεχνικού τραπεζιού.	29
Σχήμα 3. 5: Τεχνητή μορφή ζωής με την εφαρμογή εξελικτικής σχεδίασης	31
Σχήμα 3. 6: Απλή εφαρμογή γραμματικών σχημάτων.	32
Σχήμα 3. 7: Το αποτέλεσμα της εφαρμογής που παράχθηκε από την εφαρμογή γραμματικών σχημάτων.	32
Σχήμα 4. 1: Η εξέλιξη των ενδυμάτων από την “αρχή” μέχρι και σήμερα	36
Σχήμα 5. 1: Κανόνες σχημάτων για την δημιουργία του προτύπου μπλούζας.	44
Σχήμα 5. 2: Γονότυπος προτύπου μπλούζας και εξήγησή αντίστοιχών αλληλίων.	45
Σχήμα 5. 3: Γραφική αναπαράσταση φαινότυπου προτύπου μπλούζας.	46
Σχήμα 5. 4: Κατηγοριοποίηση τελεστών επιλογής γονέα	49
Σχήμα 5. 5: Διασταύρωση “T-Shirt” με τελεστή ομοιόμορφης διασταύρωσης και αναπαραγωγή απογόνου.	52
Σχήμα 5. 6: Τελεστής μετάλλαξης που εφαρμόστηκε σε γονότυπου απόγονου στο πρότυπο μπλούζας.	53
Σχήμα 5. 7: Tab menu και οι επιλογές “GENERATION”, “OPTIONS” και “HELP”.	54
Σχήμα 5. 8: Μενού κατηγοριών πρότυπου ενδύματος.	55
Σχήμα 5. 9: Τα πλαίσια του OPTIONS TAB και το κουμπί GENERATE.	57
Σχήμα 5. 10: Η καρτέλα GENERATION, το πλαίσιο σχεδίων στο άνω μέρος της και το πλαίσιο ελέγχου στο κάτω μέρος της.	58
Σχήμα 5. 11: Choice box για την βαθμολόγηση του σχεδίου από τον χρήστη.	59
Σχήμα 5. 12: Τρόπος προβολής των εμπρός και πίσω πλευρών για το πρότυπο του παντελονιού.	59
Σχήμα 5. 13: Πλαίσιο ελέγχου πριν και μετά την μεγιστοποίηση του πίνακα στατιστικών.	60

Σχήμα 6. 1: Καθορισμός τιμών βέλτιστης λύσης.	65
Σχήμα 6. 2: Αρχικός Πληθυσμός	66
Σχήμα 6. 3: Μεσαίος Πληθυσμός	67
Σχήμα 6. 4: Τελικός Πληθυσμός	67
Σχήμα 6. 5: Επιλογές στο Filters panel για το πρότυπο παντελονιού.	69
Σχήμα 6. 6: Επιλογές στο Trends panel για το πρότυπο παντελονιού.	69
Σχήμα 6. 7: Αρχικός Πληθυσμός	70
Σχήμα 6. 8: Μεσαίος Πληθυσμός	70
Σχήμα 6. 9: Τελικός Πληθυσμός	71
Πίνακας 6. 1: Παράμετροι ΕΑ	64
Πίνακας 6. 1: Παράμετροι ΕΑ	64
Πίνακας 6. 2: Προκαθορισμένες τιμές παραμέτρων ΕΑ	64

## Συντομογραφίες

ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
ΕΑ	Εξελικτικός Αλγόριθμος
ΓΑ	Γενετικός Αλγόριθμος
ΓΠ	Γενετικός Προγραμματισμός
ΕΠ	Εξελικτικός Προγραμματισμός
ΔΕΑ	Διαδραστικός Εξελικτικός Αλγόριθμος
ΕΣ	Εξελικτική Σχεδίαση
ΔΕΣ	Δημιουργική Εξελικτική Σχεδίαση
ΕΒΣ	Εξελικτική Βελτιστοποίηση Σχημάτων
ΘΕΣ	Θεμελιώδης Εξελικτική Σχεδίαση
ΠΕΣ	Παραγωγική Εξελικτική Σχεδίαση
ΕΤ	Εξελικτική Τέχνη
ΤΖ	Τεχνητοί Ζωή
EMTZ	Εξελικτικές Μορφές Τεχνητής Ζωής
ΚΣ	Κώδικας Σχημάτων
ΓΣ	Στρατηγικές Σχημάτων
ΔΕΣ	Διαδραστικά Εξελικτικά Συστήματα



# Κεφάλαιο 1ο: Εισαγωγή και Στόχοι

## 1.1 Εισαγωγή

Κάποιες δεκαετίες μετά την εφεύρεση του πρώτου ηλεκτρονικού υπολογιστή ο οποίος στην αρχή είχε την δυνατότητα να πραγματοποιήσει απλές μαθηματικές πράξεις, επινοήθηκε και προτάθηκε ένας τομέας της επιστήμης των υπολογιστών ο οποίος έγινε γνωστός ως εξελικτικός υπολογισμός (evolutionary computation). Ο εξελικτικός υπολογισμός αποτελεί μια γενικότερη έννοια εύρεσης των παγκόσμιων βέλτιστων λύσεων σε προβλήματα με μεγάλο χώρο αναζήτησης, που ο άνθρωπος από μόνος του δεν θα μπορούσε με ευκολία να διανοηθεί. Αυτό το επιτυγχάνει αφού από την αρχή του είχε υιοθετήσει βασικές λειτουργίες της εξελικτικής βιολογίας.

Ο εξελικτικός υπολογισμός χωρίζεται σε τέσσερις βασικούς αλγόριθμους, όπου ο καθένας εξειδικεύεται και σε διαφορετικό χώρο αναζήτησης, ή αλλιώς σε προβλήματα διαφορετικής φύσης, γνωστοί ως εξελικτικοί αλγόριθμοι. Ο χρονοπρογραμματισμός για τη συντήρηση ενός εθνικού δικτύου ηλεκτροδότησης, η επίτευξη της στερεοσκοπικής όρασης σε ένα ανθρωποειδές ρομπότ, η πρόβλεψη για τις ενεργειακές απαιτήσεις κυκλωμάτων υψηλών ταχυτήτων, είναι κάποια από τα πιο γνωστά έργα που εφαρμόστηκε πάνω τους ο εξελικτικός υπολογισμός ανά τα χρόνια.

Μια σημαντική εφαρμογή του εξελικτικού υπολογισμού που προέκυψε με την χρόνια εφαρμογή του και τον πειραματισμό που έγινε με αυτόν αφορά την σχεδίαση. Ο υπό τομέας αυτός ονομάστηκε εξελικτική σχεδίαση και αφορά την σχεδίαση με την ευρεία έννοια του όρου αυτού. Μέχρι σήμερα η εξελικτική τέχνη έχει τέσσερις γνωστές υποκατηγορίες, η κάθε μια από αυτές με σκοπό την σχεδίαση τέχνης, την σχεδίαση τεχνητής ζωής ή την βελτιστοποίηση έτοιμων σχεδίων. Η εξελικτική σχεδίαση έχει διάφορες εφαρμογές σε συστήματα σχεδίασης όλων των ειδών υπαρκτών υλικών αντικειμένων, όπως το ένδυμα, η ανάπτυξη εξαρτημάτων για οχήματα, η ανάπτυξη διαφόρων, εικονικών πλασμάτων και η δημιουργία έργων τέχνης σε οποιαδήποτε μορφή.

## 1.2 Στόχος της Εργασίας

Ο στόχος της παρούσας Π.Ε. είναι η υλοποίηση μιας εφαρμογής για την υποβοήθηση του προβλήματος σχεδίασης ενδυμάτων την χρήση διαδραστικών εξελικτικών αλγορίθμων. Κοινώς, αποτελεί μια προσπάθεια δημιουργίας ενός εξελικτικού αλγόριθμου ο οποίος με την βοήθεια την εξελικτικής σχεδίασης θα συμβάλλει στην δημιουργία μιας εφαρμογής ικανής να δημιουργήσει πρωτότυπα σχέδια φιλικά προς τον χρήστη, που θα του προσφέρουν εμπνευση για την δημιουργία νέων σχεδίων ενδυμάτων.

Οι λύσεις που η εφαρμογή αναπαράγει, δεν βελτιστοποιούν κάποια συνάρτηση σε κάποιο παγκόσμιο βέλτιστο, καθώς δεν υπάρχουν βέλτιστα στον χώρο της μόδας. Έτσι με την βοήθεια της διάδρασης του χρήστη με το σύστημα θα βγαίνει το δικό του μοναδικό “βέλτιστο” σχέδιο. Φυσικά, για την διαδικασία της αλληλεπίδρασης ο χρήστης πρέπει να έρχεται σε επαφή με το σχέδιο και να είναι σε θέση να το παρατηρήσει και να το κρίνει. Έτσι πρέπει να δημιουργηθεί και μια διεπαφή ιδιαίτερα κατανοητή στην χρήση της και φιλική προς τον χρήστη, καθώς και ένας τρόπος αξιολόγησης των σχεδίων χωρίς να κουράζει. Η εφαρμογή υλοποιείται με την γλώσσα προγραμματισμού Java και η διεπαφή με την JavaFX. Η προσπάθεια επικεντρώνεται κυρίως στην ανάδειξη των εξελικτικών λειτουργιών.

### 1.3 Συνοπτική Αναφορά Κεφαλαίων

Σε αυτήν την ενότητα παρουσιάζονται τα κεφάλαια που ακολουθούν.

Στο δεύτερο κεφάλαιο “Εξελκτικοί Αλγόριθμοι”, μετά την εις βάθος εισαγωγή που γίνεται για τον εξελκτικό υπολογισμό, γίνεται μια εισαγωγή στο απαιτούμενο βιολογικό υπόβαθρο το οποίο θεωρείται αναγκαίο για την καλύτερη κατανόηση των εξελκτικών τελεστών, καθώς και τον όρων που χρησιμοποιούνται στους εξελκτικούς αλγόριθμους. Έπειτα αναλύονται οι τέσσερις βασικοί εξελκτικοί αλγόριθμοι, με μια ιστορική αναδρομή και με την παρουσίαση του τρόπου λειτουργίας του καθενός. Στο τέλος του κεφαλαίου, παρουσιάζεται μια υποκατηγορία των εξελκτικών αλγόριθμων, αυτή των διαδραστικών εξελκτικών αλγόριθμων όπου είναι και το ζητούμενο της Π.Ε. καθώς δίνεται και ένα παράδειγμα εφαρμογής τους.

Στο τρίτο κεφάλαιο “Εξελκτική Σχεδίαση”, γίνεται μια εισαγωγή για την κατανόηση της και έπειτα παρουσιάζονται και αναλύονται οι τέσσερις βασικές κατηγορίες της, η κάθε μια από αυτές στον κλάδο εφαρμογής της. Στο τέλος του κεφαλαίου, παρουσιάζεται και η έννοια των γραμματικών σχημάτων, η λογική της και ο τρόπος λειτουργίας της.

Στο κεφάλαιο τέσσερα με τίτλο “Παρουσίαση Προβλήματος” γίνεται μια εισαγωγή στο πρόβλημα σχεδίασης ενδυμάτων εκτός του χώρου εφαρμογής του εξελκτικού υπολογισμού, στην συνέχεια αναφέρεται ο βασικός τρόπος σχεδίασης που χρησιμοποιεί ένας σχεδιαστής του χώρου της μόδας, ενώ στην συνέχεια παρουσιάζονται κάποιοι τρόποι προσέγγισης της σχεδίασης με εξελκτικούς αλγόριθμους και παρουσιάζεται και η διαδραστική αξιολόγηση από τον χρήστη.

Στο κεφάλαιο πέντε “Μοντέλο Προσέγγισης και Εφαρμογή” αναλύονται οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής, στην συνέχεια παρουσιάζονται η γραμματική σχημάτων με τον τρόπο που εφαρμόστηκε, ο διαδραστικός εξελκτικός αλγόριθμος που επινοήθηκε και το γενετικό υλικό. Έπειτα, το κεφάλαιο κλείνει με μια ξενάγηση στην διεπαφή που αποσκοπεί στην εξοικείωση του χρήστη με αυτήν.

Στο κεφάλαιο έξι “Αποτελέσματα Εφαρμογής”, θίγονται κάποια από τα προβλήματα που προέκυψαν, κατά την διάρκεια υλοποίησης της εφαρμογής και πως αυτά αντιμετωπίστηκαν και δίνονται δύο σενάρια χρήσης της εφαρμογής με διαφορετικές ρυθμίσεις των τελεστών επιβίωσης και της τροφοδότησης αρχικοποίησης.

Στο κεφάλαιο εφτά “Συμπεράσματα και Περαιτέρω Βελτιώσεις”, το οποίο αποτελεί το τελευταίο κεφάλαιο της Π.Ε., γίνεται μια σύνοψη των συμπερασμάτων από την προσπάθεια δημιουργίας της εφαρμογής, μια κριτική σχετικά με το πόσο κοντά πλησίασε τον αρχικό της στόχο. Έπειτα γίνονται κάποιες προτάσεις για την βελτίωση, σε σημεία στα οποία δεν ικανοποιεί το αποτέλεσμα.

Τέλος και μετά τις αναφορές, δίνεται στα παραρτήματα ο κώδικας διαφόρων λειτουργιών και τελεστών του διαδραστικού εξελκτικού αλγόριθμου.

### 1.4 Επίλογος

Ανακεφαλαιώνοντας, σε αυτό το εισαγωγικό κεφάλαιο είδαμε συνοπτικά τι είναι ο εξελκτικός υπολογισμός, η εξελκτική σχεδίαση καθώς και κάποιες από τις εφαρμογές τους. Έπειτα παρουσιάσαμε τους αρχικούς στόχους που έχει η Π.Ε., ή αλλιώς τους στόχους που η εφαρμογή

καλείται να διεκπεραιώσει. Τέλος είδαμε μια σύνοψη των κεφαλαίων που ακολουθούν μετά από την εισαγωγή και τα παραρτήματα του κώδικα που ακολουθούν στο τέλος.



## Κεφάλαιο 2ο: Εξελικτικοί Αλγόριθμοι

### 2.1 Εισαγωγή

Ο εξελικτικός υπολογισμός (evolutionary computation) χρησιμοποιεί υπολογιστικά μοντέλα βασισμένα στην εξελικτική βιολογία (evolutionary biology) όπως βασικά στοιχεία για το σχεδιασμό και την εφαρμογή της επίλυσης προβλημάτων μέσω υπολογιστικών συστημάτων, με σκοπό την αναζήτηση καλύτερων - βέλτιστων λύσεων. Υπάρχει μια ποικιλία εξελικτικών υπολογιστικών μοντέλων που έχουν υπάρξει, προτείνονται και μελετώνται στους οποίους θα αναφερθούμε ως εξελικτικοί αλγόριθμοι (EA).

Αποτελούνται από μια κοινή εννοιολογική βάση προσομοίωσης της εξέλιξης των επιμέρους ατόμων ενός πληθυσμού μέσω διαδικασιών επιλογής και αναπαραγωγής. Αυτές οι διαδικασίες εξαρτώνται από τον δείκτη ποιότητας (fitness) των επιμέρους ατόμων όπως ορίζονται από ένα περιβάλλον και αφορούν την σχετική απόδοση τους σε μια μέθοδο αξιολόγησης.

Πιο συγκεκριμένα, οι EA διατηρούν έναν πληθυσμό ατόμων που εξελίσσονται σύμφωνα με κανόνες επιλογής και άλλους τελεστές, όπως ο ανασυνδυασμός και η μετάλλαξη. Κάθε άτομο-λύση στον πληθυσμό λαμβάνει μια αξιολόγηση της ποιότητας του στο περιβάλλον. Η επιλογή εστιάζει την προσοχή σε άτομα-λύσεις υψηλής ποιότητας. Οι τελεστές ανασυνδυασμού και μετάλλαξης σε αυτά τα άτομα, παρέχει γενικές ευρετικές μεθόδους για εξερεύνηση.

Ένας EA τυπικά αρχικοποιεί τον πληθυσμό του τυχαία, αν και μπορεί να δοθεί μια καθοδήγηση για το εύρος των λύσεων του προβλήματος, συνεπώς να επηρεάσει στοχευμένα τον αρχικό πληθυσμό. Πραγματοποιείτε αξιολόγηση ποιότητας για κάθε άτομο ανάλογα με την αξία του σε κάποιο περιβάλλον. Η αξιολόγηση μπορεί να είναι τόσο απλή όσο ο υπολογισμός μιας συνάρτησης ποιότητας ή τόσο περίπλοκη όσο η εκτέλεση μιας περίπλοκης προσομοίωσης. Αποφασίζει ποιό θα γίνουν γονείς και πόσα παιδιά θα έχουν οι γονείς (εύρος πληθυσμού). Η επιλογή αποφασίζει τα άτομα που θα αποτελέσουν τους γονείς για την επόμενη γενεά. Τα παιδιά δημιουργούνται μέσω του ανασυνδυασμού, ο οποίος ανταλλάσσεται πληροφορίες μεταξύ των γονέων. Τα παιδιά δεν αποτελούνται μόνο από τον ανασυνδυασμό των γονιδίων τους, καθώς, ένα μέρος περνάει από την διαδικασία της μετάλλαξης (mutation). Τέλος, έχουμε επανάληψη της διαδικασίας μέχρι έναν προκαθορισμένο αριθμό πληθυσμών, ή μέχρι την εύρεση μιας ικανοποιητικής λύσης.

Οι EA είναι αποδεδειγμένα, απόλυτα ευπροσάρμοστοι για κάθε πιθανό πρόβλημα και άκρως αποτελεσματικοί, αν εφαρμοστούν ορθά στον σωστό χώρο αναζήτησης και με την σωστή μέθοδο.

### 2.2 Βιολογικό Υπόβαθρο

Οι EA έχουν βάση το μοντέλο της βιολογικής εξέλιξης του Κάρολου Δαρβίνου. Η θεωρία της εξέλιξης εξηγεί την προσαρμοστική αλλαγή των ειδών μέσω της αρχής της φυσικής επιλογής, η οποία ευνοεί την επιβίωση και την περαιτέρω εξέλιξη εκείνων των ειδών που είναι καλύτερα προσαρμοσμένα στις περιβαλλοντικές τους συνθήκες, η λεγόμενη "επιβίωση του ικανότερου". Εκτός από την παράγοντα της επιλογής, ο άλλος σημαντικός παράγοντας για την εξέλιξη, είναι η ύπαρξη αποκλίσεων (μεταλλάξεις) ανάμεσα στους φαινοτύπους των οργανισμών, δηλαδή τα φυσικά χαρακτηριστικά όπως χρώμα ματιών, ύψος, ευφυία κλπ. τα οποία ξεχωρίζουν τους γονείς από τα

παιδιά. Οι μεταλλάξεις σε αντίθεση με τους φαινότυπους επιβιώνουν μέσω της επιλογής μόνο αν είναι ικανές να “επιβιώσουν”, διαφορετικά χάνονται.

Όλοι οι ζωντανοί οργανισμοί αποτελούνται από κύτταρα και κάθε κύτταρο περιέχει το ίδιο σύνολο από ένα ή περισσότερα χρωμοσώματα τα οποία αποτελούν ακολουθίες DNA οι οποίες λειτουργούν ως προσχέδιο ανάπτυξης του οργανισμού. Ένα χρωμόσωμά διαιρείται σε γονίδια λειτουργικά τμήματα του DNA, κάθε ένα από τα οποία κωδικοποιεί μία συγκεκριμένη πρωτεΐνη. Τα γονίδια θεωρούνται οι μονάδες μεταβίβασης του συνόλου των κληρονομικών χαρακτηριστικών. Χονδρικά, μπορεί να θεωρηθεί ότι κάθε γονίδιο κωδικοποιεί ένα χαρακτηριστικό, όπως το χρώμα των ματιών. Οι διαφορετικές πιθανές τιμές ενός γονιδίου ονομάζονται αλλήλια (alleles). Κάθε γονίδιο έχει μία συγκεκριμένη θέση μέσα στο χρωμόσωμά. Τα γονίδια μεταβάλλονται περιστασιακά μέσω μεταλλάξεων.

Η επιλογή ενεργεί επί των ατόμων τα οποία μέσω του φαινοτύπου τους εκφράζουν τις πολύπλοκες αλληλεπιδράσεις εντός του γονότυπου τους, δηλαδή την συνολική γενετική πληροφορία του οργανισμού, καθώς επίσης και την αλληλεπίδραση του γονότυπου με το περιβάλλον κατά τον καθορισμό του φαινοτύπου.

Η ποιότητα ενός ατόμου συχνά ορίζεται ως η πιθανότητα βιωσιμότητας του. Στο πλαίσιο της εξέλιξης, η ποιότητα ενός ατόμου μετρείται μόνο έμμεσα μέσω του ρυθμού αύξησής του σε σύγκριση με τα άλλα άτομά του πληθυσμού δηλαδή με την τάση του να επιβιώσει και να αναπαράγει σε ένα συγκεκριμένο περιβάλλον. Επιπλέον η φυσική επιλογή δεν είναι η ενεργή κινητήρια δύναμη. Η επιλογή είναι το όνομα που χρησιμοποιείται για να περιγράψει την ικανότητα των ατόμων τα οποία έχουν καταφέρει να επιβιώσουν και να μεταφέρουν το γενετικό υλικό τους στην επόμενη γενιά.

### 2.3 Μεθοδολογίες Εξελικτικών Αλγορίθμων

Από τις αρχές τις δεκαετίες του 60, μέχρι και σήμερα, έχουν αναπτυχθεί και παρουσιαστεί πολλές μεθοδολογίες-μοντέλα ΕΑ, το καθένα για τον δικό του χώρο αναζήτησης και εφαρμογής, με σκοπό την εύρεση βέλτιστων λύσεων. Ανά τα χρόνια λοιπόν, μετά από άφθονες ώρες εφαρμογής και πειραματισμού, έχουν καθιερωθεί οι τέσσερις επικρατέστεροι ΕΑ που μπορούν να καλύψουν ολόκληρο το φάσμα πιθανών προβλημάτων, οι γενετικοί αλγόριθμοι (ΓΑ) (genetic algorithms), ο γενετικός προγραμματισμός (ΓΠ) (genetic programming), οι εξελικτικές στρατηγικές (ΕΣ) (evolution strategies) και ο εξελικτικός προγραμματισμός (ΕΠ) (evolutionary programming). Πλέον οι πλειοψηφία των μεθοδολογιών - μοντέλα που αναπτύσσονται ή εφαρμόζονται, ακολουθούν παραλλαγές από τα τέσσερα μοντέλα ή ακόμα και αυτά καθαυτά.

Η βασική ιδέα πίσω από τους θεμέλιους λίθους των ΕΑ, είναι η ίδια με κάποιες αλλαγές είτε στην αναπαράσταση των ατόμων, είτε στην στους γενετικούς τελεστές που εφαρμόζονται πάνω στους πληθυσμούς. Άλλοι ΕΑ χρησιμοποιούν αποκωδικοποίηση για την “προβολή” των ατόμων λύσεων τους, ενώ άλλοι όχι (ο γονότυπος και ο φαινότυπος είναι ταυτόσημοι). Κάποιοι χρησιμοποιούν το δυαδικό σύστημα στις τιμές των γονιδίων τους, κάποιοι απλές συμβολοσειρές και κάποιοι δέντρα συμβολοσειρών. Η αξιολόγηση των ατόμων διαφέρει από τον έναν σε άλλων, καθώς ο καθένας χρησιμοποιεί την δικιά του μέθοδο. Το ίδιο ισχύει και για την επιλογή. Οι διαφορές είναι πάντα, παραλλαγές των βασικών γεννητικών τελεστών.

Οι παραλλαγές των τεσσάρων βασικών ΕΑ, είναι αμέτρητες και οι πλειοψηφία τους δεν είναι άξια αναφοράς για την παρούσα ΠΑ. Για αυτόν τον λόγο, λοιπόν, παρακάτω θα παρουσιάσουμε τις

βασικές μορφές των τεσσάρων ΕΑ, κάνοντας αρχικά μια ιστορική αναφορά για τον καθένα, όπως και τον κλάδο πρακτικής τους και στην συνέχεια θα αναλύσουμε τον τρόπο λειτουργίας τους.

### 2.3.1 Γενετικοί Αλγόριθμοι

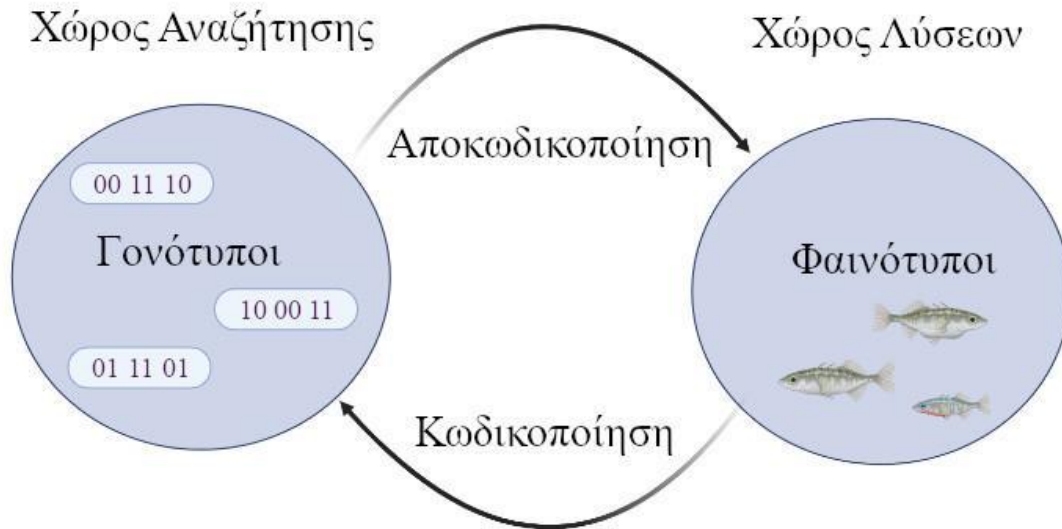
Η θεωρία των ΓΑ, αναπτύχθηκε από τον John Holland και τους συνεργάτες του τις δεκαετίες του 1960 και του 1970 [1][2], είναι ένα μοντέλο - αλγόριθμος βασισμένος στην βιολογική εξέλιξη της θεωρίας της φυσικής επιλογής του Κάρολου Δαρβίνου. Ο ΓΑ ήταν πιθανόν ο πρωτοπόρος αλγόριθμος που χρησιμοποίησε τους τελεστές ανασυνδυασμού, μετάλλαξης και της επιλογής στη μελέτη προσαρμοστικών και τεχνητών συστημάτων. Αυτοί οι γενετικοί τελεστές αποτελούν το ουσιαστικό μέρος του γενετικού αλγορίθμου ως στρατηγική επίλυσης προβλημάτων. Έκτοτε, πολλές παραλλαγές γενετικών αλγορίθμων έχουν αναπτυχθεί και εφαρμοστεί σε ένα ευρύ φάσμα προβλημάτων βελτιστοποίησης, από το χρωματισμό γραφημάτων έως την αναγνώριση σχεδίων, από τα διακριτά συστήματα (όπως το πρόβλημα του ταξιδιώτη πωλητή) έως τα συνεχή συστήματα (π.χ. τον αποτελεσματικό σχεδιασμό αεροτομής στην αεροδιαστημική μηχανική) και από τις χρηματοπιστωτικές αγορές έως τη βελτιστοποίηση μηχανικής πολλαπλών στόχων.

Ανάμεσα σε άλλους αλγόριθμους βελτιστοποίησης, ο ΓΑ ξεχωρίζει για την αποτελεσματικότητά του. Η ικανότητα του ΓΑ να αντιμετωπίζει σύνθετα προβλήματα και παραλληλισμούς, όπως και η ποικιλία του σε διάφορους τύπους βελτιστοποίησης, (π.χ. αντικειμενική συνάρτηση (καταλληλόλητα) είναι σταθερή ή μη στάσιμη (μεταβολή με το χρόνο), γραμμική ή μη γραμμική, συνεχής ή ασυνεχής ή με τυχαίο θόρυβο). Ωστόσο, η διαμόρφωση της συνάρτησης καταλληλόλητας, η χρήση του μεγέθους του πληθυσμού, η επιλογή των σημαντικών παραμέτρων όπως ο ρυθμός μετάλλαξης και η διασταύρωση, και τα κριτήρια επιλογής του νέου πληθυσμού θα πρέπει να γίνονται με περιεκτικό τρόπο.

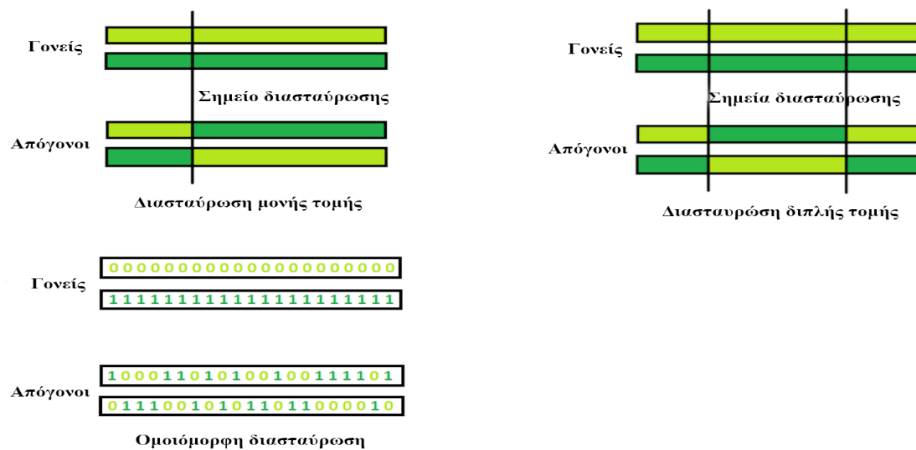
#### 2.3.1.1 Πως λειτουργούν οι Γενετικοί Αλγόριθμοι;

Οι ΓΑ διαχωρίζουν τον χώρο αναζήτησης από τον χώρο λύσεων όπως φαίνεται στο σχήμα 2.1. Ο χώρος αναζήτησης, όταν δεν είναι ταυτόσημος με τον χώρο λύσεων αποτελείται από κωδικοποιημένες δομές ατόμων - λύσεων γνωστών ως γονότυπους, οπότε για την αξιολόγηση της ποιότητας των λύσεων, ο αλγόριθμος ΓΑ αποκωδικοποιεί τους γονότυπους στον χώρο λύσεων, μετατρέποντάς τους στους αντίστοιχους φαινότυπους. Συνεπώς, ένας πληθυσμός στον ΓΑ αποτελείται από άτομα γονοτύπων και τα αντίστοιχα άτομα φαινοτύπων που αποτελούν της κωδικοποιημένες μορφές τους. Οι κωδικοποιημένες παράμετροι αποκαλούνται γονίδια και οι τιμές τους αλληλία.

Όπως και σε άλλους ΕΑ, η διαδικασία είναι παρεμφερής, αρχικοποίηση, αξιολόγηση, ανασυνδυασμός (διασταύρωση, μετάλλαξη), επιλογή και αναπαραγωγή. Ο ΓΑ καθώς και ο πρώτος που χρησιμοποίησε τους γενετικούς τελεστές δεν διαφέρει ιδιαίτερα από τους υπόλοιπους. Η αρχικοποίηση του πληθυσμού γίνεται με την ανάθεση τυχαίων τιμών (ή αλλιώς αλληλίων) στον γονότυπο. Έπειτα, ο γονότυπος, αποκωδικοποιείται στον αντίστοιχο φαινότυπο του, με σκοπό να αξιολογηθεί και να παρέχει τον δείκτη ποιότητας. Η βασική διαφορά από τους άλλους ΕΑ είναι κατά την διαδικασία επιλογής. Ο ΓΑ εμπεριέχει έναν “χώρο”, ο οποίος ονομάζεται δεξαμενή ζευγαρώματος (mating pool). Μετά την αξιολόγηση των ατόμων, δεν υπάρχει επιλογή με βάση τον δείκτη ποιότητας. Τα άτομα - λύσης γεμίζουν την δεξαμενή ζευγαρώματος με αντίγραφα τους. Τα αντίγραφα παράγονται σε κατάλληλο αριθμό, ανάλογα με την τιμή ποιότητας των ατόμων. Η επιλογή, γίνεται με τυχαίο τρόπο, συλλέγοντας δύο γονείς από την δεξαμενή ζευγαρώματος για την αναπαραγωγή της επόμενης γενεάς.



Σχήμα 2. 1: Χαρτογράφηση χώρου αναζήτησης και χώρου λύσεων και τα αντίστοιχα περιεχόμενα τους (γονότυποι, φαινότυποι).



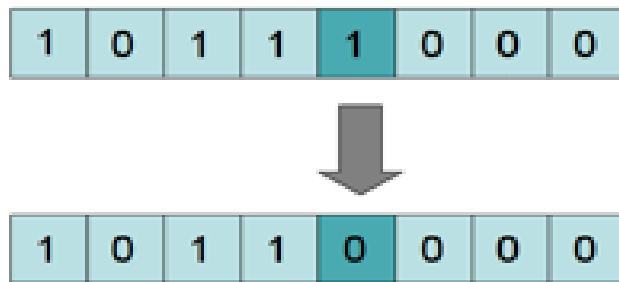
Σχήμα 2. 2: Τρεις τεχνικές διασταύρωσης που ορίζονται από το περιβάλλον.

Στην συνέχεια, η διαδικασία αναπαραγωγής, πραγματοποιείται με ανασυνδυασμό. Ο ανασυνδυασμός στον ΓΑ χρησιμοποιεί δύο γενετικούς τελεστές, τον τελεστή διασταύρωσης (crossover) και τον τελεστή μετάλλαξης (mutation). Η διασταύρωση κατανέμει τυχαία γονίδια από τους γονότυπους των γονέων σε αυτούς τον απογόνων με μία από τις μεθόδους που φαίνονται στο

σχήμα 2.2. Το μεγαλύτερο ποσοστό των γονιδίων του απογόνου καθορίζεται από την διαδικασία της διασταυρώσεως.

Ο ΓΑ χρησιμοποιεί περιστασιακά τον τελεστή μετάλλαξης. Η διαδικασία που εκτελεί ο τελεστής μετάλλαξης, είναι να επιλέγει τυχαία ένα αλληλίο δηλαδή την τιμή ενός γονιδίου όπως βλέπουμε στον σχήμα 2.3 και να το αλλάζει με μια τυχαία τιμή. Η αναπαραγωγή συνεχίζεται μέχρι την συμπλήρωση πληθυσμού της επόμενης γενεάς. Η όλη διαδικασία του ΓΑ τερματίζει όταν βρεθεί ικανοποιητική λύση, ειδικά στο πέρας προκαθορισμένου αριθμού πληθυσμών.

Ο ΓΑ που αναλύθηκε παραπάνω, ήταν η απλή μορφή του, που πρότεινε ο John Holland. Υπάρχει ένας μεγάλος αριθμός εξειδικευμένων ΓΑ, όπως ο Real-coded ΓΑ στον οποίο η μεταβλητές που λαμβάνουν πραγματικές τιμές χρησιμοποιούνται κατευθείαν (Deb and Agrawal, 1995; Deb και Kumar, 1995; Eshelman και Schaffer, 1992), ο Micro ΓΑ χρησιμοποιεί μικρό μέγεθος πληθυσμού (της τάξης των 4 ή 5) (Krishnakumar, 1989), Ο Knowledge-augmented ΓΑ, ο οποίος τροφοδοτεί το περιβάλλον με γνώσεις σχετικές της λύσεις, παρέχοντας ταχύτητα (Davidor, 1991; Deb, 1993), ο Hybrid ΓΑ που συνεργάζεται με αλγόριθμους τοπικής αναζήτησης (Powell and Skolnick, 1989· Kelly and Davis, 1991), τελευταίος και πιο σημαντικός για την εφαρμογή μας είναι ο διαδραστικός ΓΑ (Kim & Cho, 2000), ο οποίος διαφέρει στον τρόπο αξιολόγησης, που κατά βάση πραγματοποιείται από τον



Σχήμα 2. 3: Διαδικασία μετάλλαξης: Επιλογή αλληλίου και αλλαγή της τιμής του με τυχαία τιμή.

χρήστη, διότι οι φαινότυποι αποτελούν οπτικοακουστικές αναπαραστάσεις.

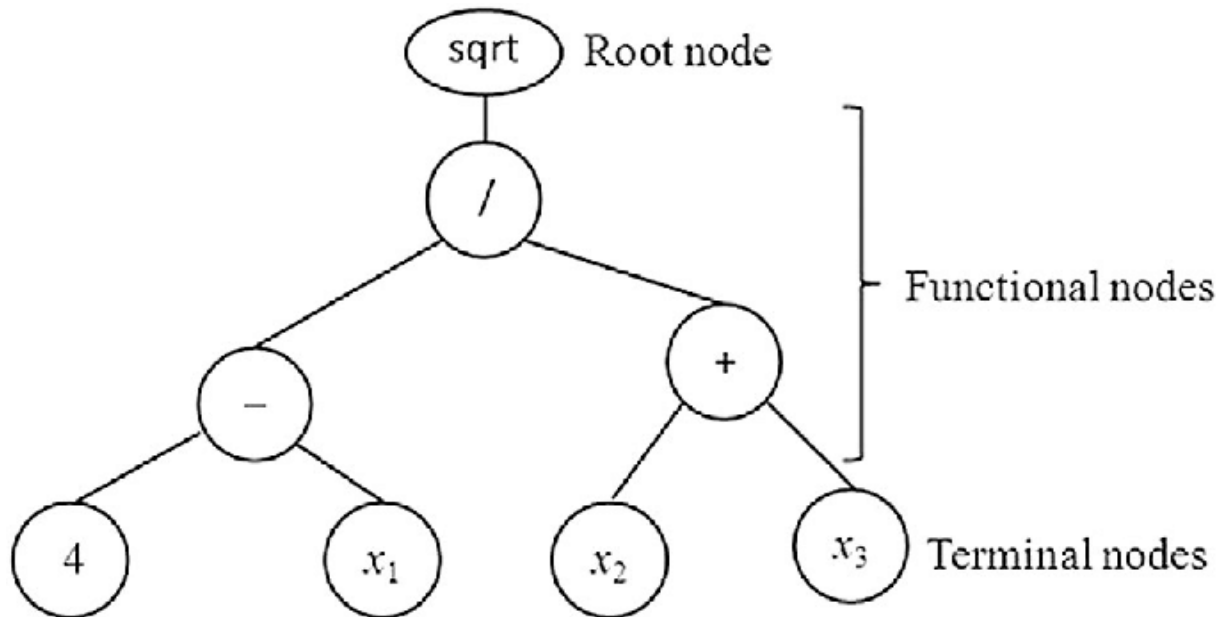
### 2.3.2 Γενετικός Προγραμματισμός

Ο ΓΠ παρουσιάστηκε από τον John Koza το 1991 [5]. Είναι μια συστηματική μέθοδος που επιτρέπει στους υπολογιστές να λύσουν αυτόματα ένα πρόβλημα ξεκινώντας από μια δήλωση υψηλού επιπέδου για το τι πρέπει να γίνει. Ο ΓΠ είναι μια μέθοδος ανεξάρτητη από το πεδίο που γενετικά δημιουργεί έναν πληθυσμό προγραμμάτων για την επίλυση ενός προβλήματος. Συγκεκριμένα, ο ΓΠ μετασχηματίζει επαναληπτικά έναν πληθυσμό προγραμμάτων σε μια νέα γενιά προγραμμάτων με την εφαρμογή ανάλογων φυσικών γενετικών επεμβάσεων.

Οι γενετικές επεμβάσεις που εφαρμόζει ο ΓΠ, αποτελούνται από τον ανασυνδυασμό, την μετάλλαξη, την αναπαραγωγή, τον γονιδιακό διπλασιασμό και την γονιδιακή διαγραφή. Ανάλογα της ανάπτυξης, μερικές φορές χρησιμοποιούνται διαδικασίες για τη μετατροπή ενός εμβρύου σε πλήρως ανεπτυγμένο οργανισμό. Ο ΓΠ είναι μια επέκταση του ΓΑ [1]. Οι δομές στον πληθυσμό δεν είναι συμβολοσειρές χαρακτήρων σταθερού μήκους που κωδικοποιούν υποψήφιες λύσεις σε ένα πρόβλημα, αλλά προγράμματα που, όταν εκτελούνται, είναι οι υποψήφιες λύσεις στο πρόβλημα. Τα

προγράμματα εκφράζονται στον γενετικό προγραμματισμό ως δέντρα σύνταξης και όχι ως γραμμές κώδικα.

Απεικονίζεται όπως φαίνεται παρακάτω στο σχήμα 2.4. Το δέντρο περιλαμβάνει κόμβους και συνδέσμους. Οι κόμβοι υποδεικνύουν τις εντολές που πρέπει να εκτελεστούν. Οι σύνδεσμοι υποδεικνύουν τα ορίσματα για κάθε εντολή. Στη συνέχεια το εσωτερικοί κόμβοι σε ένα δέντρο θα ονομάζονται συναρτήσεις (functional nodes), ενώ τα φύλλα του δέντρου θα ονομάζονται τερματικά (terminal nodes).



Σχήμα 2. 4: Δενδροειδής δομή ατόμου στον ΓΠ.

Σε πιο προηγμένες μορφές ΓΠ, τα προγράμματα μπορούν να αποτελούνται από πολλαπλά στοιχεία. Σε αυτήν την περίπτωση, η αναπαράσταση που χρησιμοποιείται στον ΓΠ είναι ένα σύνολο δέντρων ομαδοποιημένα κάτω από έναν ειδικό κόμβο που ονομάζεται ρίζα (root node). Αυτά τα υποδέντρα θα τα ονομάσουμε κλαδιά. Ο αριθμός και ο τύπος των κλαδιών σε ένα πρόγραμμα, μαζί με ορισμένα άλλα χαρακτηριστικά της δομής των κλαδιών, αποτελούν την αρχιτεκτονική του προγράμματος.

### 2.3.2.1 Πως λειτουργεί ο Γενετικός Προγραμματισμός;

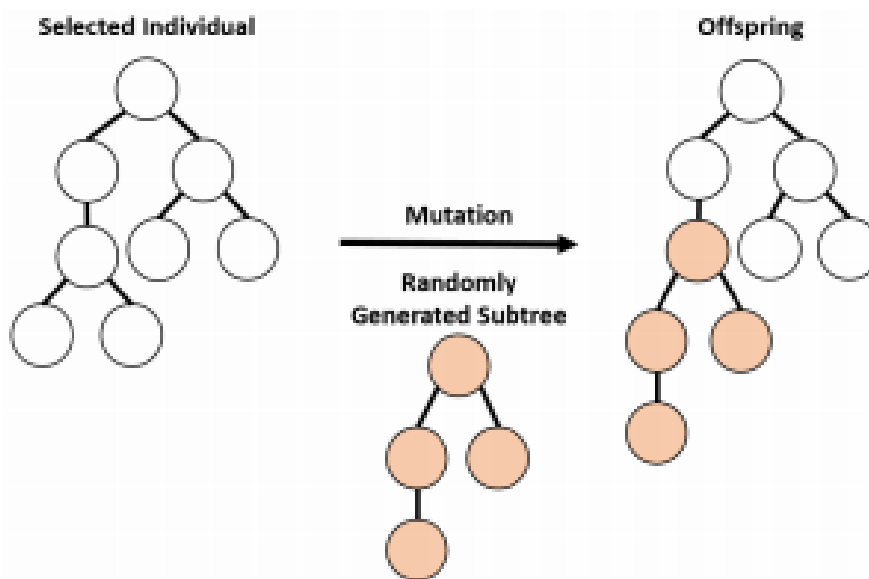
Όπως και με τα άλλα βασικά μοντέλα των ΕΑ, έτσι και με τον ΓΠ έχουμε μια τυποποιημένη διαδικασία που πρέπει να ακολουθήσει ο αλγόριθμος για να φτάσει στην λύση του αρχικού προβλήματος. Ο ΓΠ διαφέρει από τους άλλους ΕΑ, κατά κύριο λόγο, της δομής των ατόμων - λύσεων του. Αναφέρθηκε παραπάνω ότι η δομή των ατόμων αποτελείται από κώδικα (προγράμματα) και σκοπό έχει να συντάξει ένα πλήρη λειτουργικό πρόγραμμα, γι' αυτόν τον λόγο έχουμε μια δομή μεταβλητού μεγέθους.

Η αρχικοποίηση του πληθυσμού στον ΓΠ πραγματοποιείται με τυχαίο τρόπο, δίνοντας “τιμές” από έναν εύλογο αριθμό λειτουργικών προγραμμάτων που παρέχει το περιβάλλον. Ο ΓΠ δεν

διαχωρίζει τον χώρο αναζήτησης από τον χώρο λύσεων, συνεπώς γονότυποι και φαινότυποι είναι ταυτόσημοι. Ο ΓΠ δεν χρησιμοποιεί τις διαδικασίες κωδικοποιήσεις και αποκωδικοποιήσεις, διότι δεν υπάρχει η ανάγκη αναπαραστάσεις των ατόμων-λύσεων.

Όπως είναι φυσικό για την επιλογή γονέων στους ΕΑ, ακολουθεί η αξιολόγηση των ατόμων-προγραμμάτων. Επιλέγονται τα νέα άτομα-προγράμματα καθαρά με βάση του δείκτη ποιότητας τους (“επιβίωση του ικανότερου”), για την διαδικασία της αναπαραγωγής.

Η αναπαράσταση των ατόμων, λόγω της φύσης της δομής τους (μεταβλητό μέγεθος) πραγματοποιείται με ένα δενδροειδή τρόπο ο οποίος είναι ιεραρχικά δομημένος. Για να υπολογιστεί ο δείκτης ποιότητας και να εφαρμοστεί η αξιολόγηση πάνω στα άτομα-προγράμματα ο ΓΠ ακολουθεί έναν μοναδικά δικό του τρόπο. Τροφοδοτεί το περιβάλλον με ένα αόριστο αριθμό εισαγόμενων τιμών και τους αντίστοιχους επιθυμητούς εξαγόμενους. Τα άτομα - λύσεις τρέχουν από τον κώδικα τους



Σχήμα 2. 5: Δομή ατόμου στον ΓΠ. Διαδικασία μετάλλαξης και παραγωγή απογόνου.

τους αριθμούς και δίνουν τα αποτελέσματα τους. Συνεπώς, με βάση τα αποτελέσματα και την ακρίβεια τους γίνεται η αξιολόγηση και έπειτα η επιλογή των γονέων.

Κατά την διαδικασία της αναπαραγωγής - ανασυνδυασμού έχουμε την ανταλλαγή κλαδιών τυχαίας διακλάδωσης (σημεία διασταύρωσης) για την δημιουργία απογόνων, αφήνοντας ανέπαφη τη σύνταξη των προγραμμάτων. Ο τελεστής μετάλλαξης όπως φαίνεται στο σχήμα 2.5 ενεργεί πάνω στο δέντρο επιλέγοντας ένα κλαδί το οποίο τεμαχίζει και αντικαθιστά το κάτω μέρος με τυχαίους κόμβους προγραμμάτων. Στον ΓΠ ο τελεστής μετάλλαξης θεωρείται μη αναγκαίος και εφαρμόζεται σε μικρή συχνότητα. Ο αλγόριθμος του ΓΠ τερματίζει όταν βρεθεί μια ικανοποιητική λύση, ειδάλλως με το πέρασ προκαθορισμένου αριθμού γενεών.

Κατά την διάρκεια εξέλιξης των λύσεων με χρήση ΓΠ υπάρχει αύξηση στο μέσο μέγεθος και βάθος του δέντρου χωρίς την αντίστοιχη, επιθυμητή αύξηση ποιότητας, ένα φαινόμενο που συνήθως αναφέρεται ως υπερχειλίση (bloat). Η υπερχειλίση είναι ένα πολύ γνωστό φαινόμενο στον ΓΠ. Ένα μεμονωμένο πρόγραμμα στο ΓΠ θα μπορούσε να είναι οποιουδήποτε μεγέθους. Τέτοια ευελιξία αναπαράστασης παρέχει μεγαλύτερη ελευθερία στην αναζήτηση λύσεων, αλλά ταυτόχρονα προκαλεί

υπερχείλιση. Αδιαμφισβήτητα, μεγάλα άτομα συνεπάγονται υπολογιστικά, σε πιο δαπανηρή διαδικασία αξιολόγησης κατά τη διάρκεια της εξέλιξης. Αν αποτελέσουν τις τελικές λύσεις, ο χρόνος εκτέλεσής τους σε εφαρμογές θα αυξηθεί. Σαφώς αυτό δεν είναι επιθυμητό για καταστάσεις όπου η ταχύτητα είναι απαίτηση, όπως σε συστήματα πραγματικού χρόνου. Ουσιαστικά η υπερχειλίση δημιουργεί πλεονάζον κώδικα που καθυστερεί την διαδικασία της εκτέλεσης του.

Η φύση του αλγόριθμου του ΓΠ, για την επίλυση πιο εξειδικευμένων προβλημάτων όπως της υπερχειλίσης (bloat), εισάγει δικούς τους τελεστές. Ο τελεστής της ενθυλάκωσης (encapsulation), ο οποίος μετατρέπει ένα υπόδενδρο σε κόμβο, συμβάλλει ουσιαστικά στην αποφυγή διάσπασης του υπόδενδρου από τους τελεστές μετάλλαξης και διασταύρωσης. Ο τελεστής επέμβασης (editing), είναι υπεύθυνος για την βελτιστοποίηση - απλοποίηση των υπέρογκων λύσεων - προγραμμάτων. Τελευταίος ο τελεστής αντιμετάθεσης (permutation), ο οποίος πραγματοποιεί μεταφορά χαρακτήρων σε ένα δέντρο.

### 2.3.3 Εξελικτικές Στρατηγικές

Οι ρίζες των εξελικτικών στρατηγικών χρονολογούνται στα μέσα της δεκαετίας του 1960 όταν στο Τεχνικό Πανεπιστήμιο του Βερολίνου, Γερμανία, οι P. Bienert, I. Rechenberg και H.-P. Schwefel ανέπτυξαν τα πρώτα σχέδια εμπνευσμένα από τη βιονική για την εξέλιξη των βέλτιστων σχημάτων ελάχιστων σωμάτων έλξης σε μια αεροδυναμική σήραγγα χρησιμοποιώντας την αρχή της εξέλιξης του Δαρβίνου.

Οι ΕΣ μπορούν να εφαρμοστούν σε όλα τα πεδία βελτιστοποίησης, συμπεριλαμβανομένων των συνεχών, διακριτών, συνδυαστικών χώρων αναζήτησης με ή και χωρίς περιορισμούς, καθώς και μικτών χώρων αναζήτησης. Οι ΕΣ μπορούν επίσης να εφαρμοστούν σε ένα σύνολο αντικειμενικών συναρτήσεων στο πλαίσιο της βελτιστοποίησης πολλαπλών στόχων (Multiobjective Search).

#### 2.3.3.1 Πως λειτουργούν οι Εξελικτικές Στρατηγικές;

Μέχρι και σήμερα έχουν παρουσιαστεί και αναπτυχθεί τρία διαφορετικά πρότυπα αλγορίθμων ΕΣ. Ο αλγόριθμος του Schwefel [3] και Rechenberg [4], όπου παρουσιάστηκε από τον πρώτο και ύστερα αναπτύχθηκε από τον δεύτερο, έγινε γνωστός ως (1+1) - ΕΣ, διμελείς εξελικτικές στρατηγικές (two-membered evolution strategies). Στον αλγόριθμο αυτό, υπάρχουν μόλις δύο άτομα, ένας γονέας και ένα απόγονος. Ο απόγονος προκύπτει από μια διαδικασία εφαρμογής του τελεστή μετάλλαξης. Όπως στον ΓΠ, δεν υπάρχει διάκριση ανάμεσα στον χώρο αναζήτησης και στον χώρο λύσεων (γονότυπου, φαινότυπου), ενώ το κάθε άτομο αναπαρίσταται από ένα διάνυσμα σταθερού μήκους.

Ο αλγόριθμος (1+1) - ΕΣ ακολουθεί μια απλή διαδικασία τεσσάρων βημάτων:

- Ο αρχικός πληθυσμός, δηλαδή το άτομο που αποτελεί των γονέα δημιουργείται με τυχαία αρχικοποίηση (λαμβάνει τυχαίες τιμές στα γονίδιά του).
- Ο απόγονος δημιουργείται με απλή εφαρμογή μετάλλαξης στον γονέα. Για να επιτευχθεί η μετάλλαξη της μεθόδου αυτής, αθροίζεται ένα τυχαίο διάνυσμα σταθερού μήκους με κανονική κατανομή (μέσος όρος μηδέν και προκαθορισμένη τιμή απόκλισης).
- Στην συνέχεια γίνεται η σύγκριση - αξιολόγηση μεταξύ γονέα και απόγονου. Το άτομο με καλύτερη τιμή ποιότητας επιβιώνει.

- Αν το άτομο που επιβίωσε είναι ο γονέας, τότε επαναλαμβάνουμε μετάλλαξη πάνω του, ειδάλλως ο απόγονος γίνεται ο επόμενος γονέας. Στην συνέχεια επαναλαμβάνουμε τα βήματα 2,3.

Ο αλγόριθμος (1+1) - ΕΣ θα τερματίσει είτε στο πέρας προκαθορισμένου αριθμού γενεών, είτε αν κάποιο άτομο-λύση δώσει ικανοποιητικό αποτέλεσμα στο πρόβλημα.

Ο Rechenberg, μετά το πέρας δοκιμών, διατύπωσε στα αποτελέσματα εφαρμογής του αλγόριθμου, τον κανόνα επιτυχίας  $\frac{1}{5}$ . Η αναλογία των επιτυχημένων μεταλλάξεων πρέπει να είναι  $\frac{1}{5}$ . Αν ο αριθμός επιτυχημένων μεταλλάξεων είναι μεγαλύτερος του  $\frac{1}{5}$  τότε πρέπει να αυξηθεί η τυπική απόκλιση και αν είναι μικρότερος τότε πρέπει να μειωθεί. Ο αλγόριθμος (1+1) - ΕΣ δέχτηκε κριτική για την αργή διαδικασία εύρεσης λύσης και την αυξημένη πιθανότητα κινδύνου παγίδευσης σε κάποιο τοπικό βέλτιστο των αποτελεσμάτων του.

Σχεδόν μια δεκαετία μετά, στις αρχές του 1980 ο Rechenberg κατέληξε σε δύο νέους τύπους του αλγόριθμου (1+1) - ΕΣ, με σκοπό να βελτιώσει την έννοια του πληθυσμού (Να μεγαλώσει τον πληθυσμό) και να μειώσει τον κίνδυνο παγίδευσης. Οι νέοι τύποι ήταν οι  $(\mu+\lambda)$  - ΕΣ και  $(\mu,\lambda)$  - ΕΣ, όπου  $\mu$  είναι ο αριθμός των γονέων και  $\lambda$  ο αριθμός των απογόνων.

Στην γενική τους μορφή και σε σχέση με τον αλγόριθμο (1+1) - ΕΣ, ο πληθυσμός τους αποτελείται από περισσότερα του ενός άτομου - λύσης. Επίσης, έχουμε την εφαρμογή του τελεστή ανασυνδυασμού και όχι μόνο του τελεστή μετάλλαξης για την παραγωγή νέων πληθυσμών από τα βέλτιστα άτομα. Οι χώροι λύσεων και αναζήτησης δεν διαχωρίζονται, συνεπώς δεν εκτελούν διαδικασίες αποκωδικοποίησης, καθώς δρουν απευθείας στις μεταβλητές σχεδίασης του προβλήματος.

Η επιλογή των γονέων πραγματοποιείται αξιοκρατικά. Οι δύο αλγόριθμοι έχουν διαφορετικό τρόπο επιλογής γονέα. Η  $(\mu+\lambda)$  - ΕΣ μπορεί να επιλέξει τα καλύτερα άτομα των πληθυσμών απογόνων και γονέων, σε αντίθεση με τον  $(\mu,\lambda)$  - ΕΣ που περιορίζεται στον πληθυσμό των απογόνων.

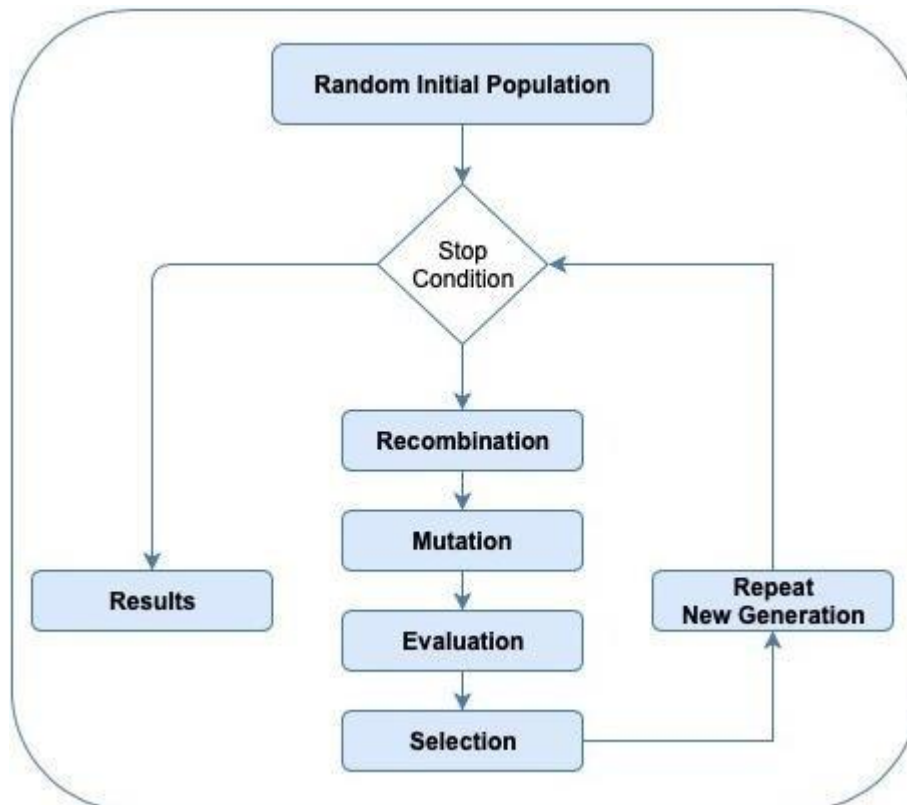
Οι αλγόριθμοι  $(\mu+\lambda)$  - ΕΣ,  $(\mu,\lambda)$  - ΕΣ ακολουθούν διαδικασία τεσσάρων βημάτων όπως φαίνεται και στον σχήμα 2.6 όπως και στον αλγόριθμο (1+1) - ΕΣ:

- Η αρχικοποίηση του πληθυσμού γίνεται με ανάθεση τυχαίων τιμών στα γονίδια των ατόμων - λύσεων ή με καθοδήγηση που παρέχει το περιβάλλον.
- Η δημιουργία πληθυσμού απογόνων, πραγματοποιείται με τυχαία δειγματοληψία από τον πληθυσμό γονέων, ακολουθώντας τις διαδικασίες των τελεστών ανασχεδιασμού και μετάλλαξης με την χρήση των στρατηγικών παραμέτρων (Υπάρχουν πέντε διαφορετικοί τύποι τυχαίας επιλογής γονέα, που συνδυάζουν έναν με δύο γονείς και πάντα ακολουθούν την διαδικασία μετάλλαξης).
- Αφού συμπληρωθεί ο πληθυσμός απογόνων, ακολουθεί η αξιολόγηση και τα  $\mu$  άτομα με μεγαλύτερο δείκτη ποιότητας επιβιώνουν και αντικαθιστούν τον πληθυσμό των γονέων.

Όπως έχει αναφερθεί, για την επιλογή στον αλγόριθμο  $(\mu+\lambda)$  - ΕΣ γίνεται αξιολόγηση στον πληθυσμό και των γονέων και των απογόνων και μετά επιλογή απ' όλους, ενώ στον  $(\mu,\lambda)$  - ΕΣ μόνο από τον πληθυσμό των απογόνων.

- Ο αλγόριθμος δεν θα τερματίσει πριν πραγματοποιηθούν αριθμός προκαθορισμένων γενεών, ή μέχρι να βρεθεί ικανοποιητική λύση στο πρόβλημα. Έως τότε έχουμε επανάληψη των βημάτων 2 και 3.

Ο κυριότερος τελεστής αναζήτησης των ΕΣ είναι αυτός της μετάλλαξης. Ακολουθεί μια κανονική κατανομή και κατευθύνει την εύρεση βέλτιστων λύσεων στον χώρο αναζήτησης. Σε αντίθεση με άλλες μεθόδους οι ΕΣ τοποθετεί στρατηγικές παραμέτρους για κάθε μεταβλητή σχεδιασμού, με αποτέλεσμα την τροποποίηση των στρατηγικών τιμών και των τιμών των μεταβλητών σε κάθε άτομο. Με αυτήν την τεχνική επιτυγχάνουν βελτιστοποίηση την κατεύθυνση της μετάλλαξης παράλληλα με τη βελτιστοποίηση των μεταβλητών. Αυτό το σημαντικό χαρακτηριστικό είναι γνωστό ως αυτό -προσαρμογή (self - adaption).



Σχήμα 2. 6: Δομή ατόμου στον ΓΠ. Διαδικασία μετάλλαξης και παραγωγή απογόνου.

### 2.3.4 Εξελικτικός Προγραμματισμός

Ο εξελικτικός προγραμματισμός εφευρέθηκε από τον Lawrence J. Fogel 1966 [6], ο οποίος δημιούργησε μια σειρά πειραμάτων στα οποία οι μηχανές πεπερασμένης κατάστασης (FSMs) αντιπροσώπευαν μεμονωμένους οργανισμούς (άτομα) σε έναν πληθυσμό λύσεων προβλημάτων. Στόχος των πειραμάτων ήταν να διατυπωθεί και να αξιολογηθεί κατά πόσο ταιριάζουν οι εξαγόμενες ακολουθίες συμβόλων, με μια επιθυμητή ακολουθία συμβόλων. Το αποτέλεσμα της αξιολόγησης αποτελεί τον γνωστό δείκτη ποιότητας. Τα FMS με μεγαλύτερο δείκτη ποιότητας διατηρούνταν για να γίνουν γονείς της επόμενης γενεάς. Η τεχνική του ΕΠ παρόλο που αποτέλεσε θέμα ποικίλων διατριβών, παρέμεινε στάσιμος για περίπου δύο δεκαετίες και ξανά πήρε ζωή με την ανασχεδίαση του από τον γιο του Lawrence Fogel, David Fogel στα τέλη της δεκαετίας του 1980.

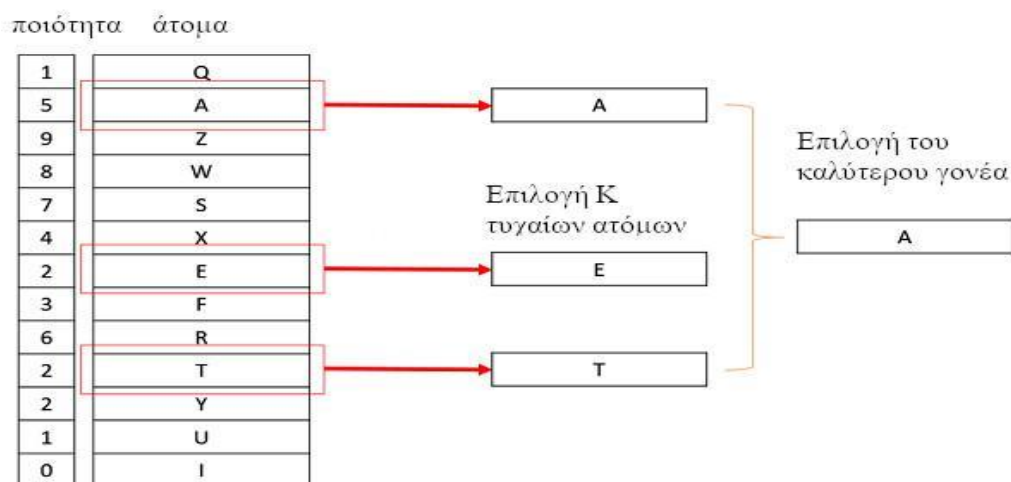
### 2.3.4.1 Πως λειτουργεί ο Εξελκτικός Προγραμματισμός;

Ο ΕΠ, πριν την ανάπτυξη του από τον David Fogel ήταν σε θέση να εξελίξει μόνο διακριτές παραμετροποιήσεις. Ο David Fogel του έδωσε την δυνατότητα να πραγματοποιήσει βελτιστοποιήσεις συνεχών παραμέτρων (ανύσματα πραγματικών τιμών, σταθερού μήκους). Επιπλέον εισήγαγε το χαρακτηριστικό της αυτό - προσαρμογής όπως είδαμε και στους τύπους (μ+λ) - ΕΣ και (μ,λ) - ΕΣ του ΕΣ, δηλαδή τη χρήση εξελισσόμενων στρατηγικών παραμέτρων για την καθοδήγηση της μετάλλαξης.

Σε άλλους ΕΑ όπως για παράδειγμα στους ΓΑ ο πληθυσμός διαχωρίζεται σε δύο διακριτούς χώρους, τον χώρο αναζήτησης και το χώρο λύσεων. Ο χώρος αναζήτησης περιέχει τους γονότυπους, ενώ ο χώρος λύσεων τους φαινότυπους. Πρακτικά, ο χώρος αναζήτησης αποτελείται από κωδικοποιημένες τις λύσεις του προβλήματος, οι οποίες αποκωδικοποιούνται στον χώρο λύσεων για να πραγματοποιηθεί η διαδικασία αξιολόγησής, να αποτυπωθεί η ποιότητα τους και να γίνει η επιλογή γονέα.

Ο ΕΠ μοιράζεται κοινό χώρο αναζήτησης και χώρο λύσεων, οπότε δεν υπάρχει κάποια διαδικασία αποκωδικοποίησης και αξιολόγηση γίνεται άμεσα επάνω στις μεταβλητές των γονότυπων. Η αρχικοποίηση του πληθυσμού γίνεται με λήψη τυχαίων τιμών στις μεταβλητές, ίσως μια κάποια καθοδήγηση από προκαθορισμένους τελεστές που παρέχει το περιβάλλον. Μετά την διαδικασία αξιολόγησης, καθώς όλα τα άτομα έχουν αποκτήσει μια τιμή ποιότητας, γίνεται η επιλογή γονέα, μέσα από μια διαδικασία γνωστή ως αθλητικό τουρνουά (tournament selection). Κατά την διαδικασία του αθλητικού τουρνουά, για κάθε άτομο επιλέγεται τυχαία μια ομάδα (Με προκαθορισμένο από τον αλγόριθμο πλήθος) άλλων ατόμων και ακολουθεί μια σειρά “αγώνων”. Η επιλογή για αναπαραγωγή δίνει περισσότερες πιθανότητες στα άτομα που κατάφεραν να νικήσουν την πλειοψηφία των αγώνων τους όπως φαίνεται στο σχήμα 2.7. Η διαδικασία του αθλητικού τουρνουά συναντάται συχνά και σε παραλλαγές των ΓΑ.

Ο ΕΠ σε αντίθεση με τους άλλους ΕΑ δε χρησιμοποιεί τελεστή ανασυνδυασμού, βασίζεται αποκλειστικά στον τελεστή μετάλλαξης. Η αναπαραγωγή γίνεται αποκλειστικά με την αντιγραφή του γονέα και την εφαρμογή μετάλλαξης επάνω στο αντίγραφο. Τα αντίγραφα τοποθετούνται στον ήδη υπάρχον πληθυσμό μέχρι το νούμερο του πληθυσμού να διπλασιαστεί από αυτό του πρώτου, όλα τα άτομα αξιολογούνται και ο μισός πληθυσμός διαγράφεται. Ο αλγόριθμος τερματίζει ύστερα από το πέρας προκαθορισμένου αριθμού γενεών.



Σχήμα 2. 7: Η διαδικασία του αθλητικού τουρνουά κατά την τυχαία επιλογή πιθανών γονέων και η σύγκριση της ποιότητας τους.

## 2.4 Διαδραστικοί Εξελικτικοί Αλγόριθμοι

Η ιδέα του ΔΕΑ, προτάθηκε στα μέσα της δεκαετίας του 1980, αποτελείται από αποτελεσματικές μεθόδους για την επίλυση ενός προβλήματος βελτιστοποίησης με ασαφείς δείκτες αξιολόγησης. Για την ακρίβεια, εφαρμόζονται πάνω σε προβλήματα που αναζητούν λύσεις καλαισθησίας, με βάση το ανθρώπινο γούστο, το οποίο “δυστυχώς” αλλάζει. Αυτοί οι αλγόριθμοι συνδυάζουν τον παραδοσιακό μηχανισμό εξέλιξης (διασταύρωση, μετάλλαξη, επιλογή, αναπαραγωγή), με την έξυπνη αξιολόγηση ενός χρήστη και ο χρήστης εκχωρεί την ποιότητα ενός ατόμου και όχι μια συνάρτηση που είναι δύσκολο ή ακόμα και αδύνατο να εκφραστεί ρητά. Μέχρι σήμερα έχουν εφαρμοστεί με επιτυχία σε πολλούς τομείς, π.χ. αναγνώριση προσώπου (Caldwell & Johnston, 1991), σχεδίαση ενδυμάτων (Kim & Cho, 2000), μουσική σύνθεση (Tokui & Iba, 2000), εφαρμογή ακουστικών βαρηκοΐας (Takagi & Ohsaki, 2007).

Μεγάλο ρόλο για την εφαρμογή ενός ΔΕΑ παίζει ο τελεστής αναπαράστασης των ατόμων - λύσεων, καθώς πρέπει να διεγείρουν τις αισθήσεις του χρήστη, είτε είναι η ακοή είτε η όραση του, για να είναι σε θέση να τα αξιολογήσει. Ένα μεγάλο ελάττωμα του ΔΕΑ είναι, το κομμάτι της αξιολόγησης και συγκεκριμένα η κούραση που μπορεί να αποφέρει στον χρήστη, για αυτό και οι ΔΕΑ διατηρούν μικρούς πληθυσμούς ατόμων (συνήθως μέχρι 10 άτομα ανά πληθυσμό). Για την σωστή κατανόηση του ΔΕΑ θα αναπτυχθεί παρακάτω λεπτομερώς ο διαδραστικός γενετικός αλγόριθμος και η εφαρμογή του σύμφωνα με την σχεδίαση ενδυμάτων [7].

### 2.4.1 Διαδραστικός Γενετικός Αλγόριθμος στην Σχεδίαση Ενδυμάτων

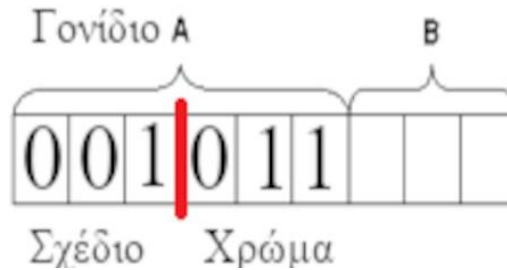
Η σχεδίαση ενδυμάτων αποτελείται από τρία μέρη σχεδίου. Τη σιλουέτα (Silhouette) που αποτελεί το περίγραμμα των ρούχων ή του ενδύματος και αναφέρεται στη γραμμή του ντυσίματος. Το τριμάρισμα (Trimming), το οποίο είναι ένας γενικός όρος όλων των στολιδιών φινιρίσματος και τέλος η λεπτομέρεια (Design) η οποία αποτελείται από υποδιαιρούμενα μέρη της σιλουέτας. Πρακτικά η σιλουέτα αποτελεί την γενική εικόνα - αναπαράσταση ενός ρούχου - ατόμου και η λεπτομέρεια το σύνολο των γονιδίων (λαιμόκοψη, γιακάς, μανίκι, μανσέτες, γραμμή μέσης, φούστα, τσέπη, κλπ.) του όπως φαίνεται στο σχήμα 2.8.

Ο ΔΓΑ, καθώς αποτελεί ένα εξειδικευμένο ΓΑ χρησιμοποιεί δύο διακριτούς χώρους, τον χώρο αναζήτησης και τον χώρο λύσεων. Ο χώρος αναζήτησης αποτελείται από κωδικοποιημένα άτομα - λύσεις γνωστά ως γονότυπα. Κάθε γονότυπος, γνωρίζουμε ότι περιέχει αριθμό γονιδίων που το καθένα αντιστοιχεί σε ένα χαρακτηριστικό του. Όποτε αν ο γονότυπος αποτελεί μια μπλούζα, τότε το γονίδιο πιθανόν να είναι μανίκια και οι τιμές που μπορεί να δεχτεί το γονίδιο είναι μια γκάμα σχεδίων από μανίκια. Τα γονίδια στον γονότυπο αποτελούνται από μια συμβολοσειρά μεταβλητού μήκους (στο παράδειγμα που θα δούμε από 0 και 1 χαρακτήρες). Ο χώρος λύσεων, εμπεριέχει τους αντίστοιχους φαινότυπους των γονότυπων. Γνωρίζουμε πως για να οι φαινότυποι αποτελούν την αποκωδικοποιημένη μορφή των γονότυπων. Στην περίπτωση της σχεδίασης ενδυμάτων με ΔΓΑ των Kim και Cho, υπάρχει μια βάση για το κάθε γονίδιο που αποτελείται από τρισδιάστατα μοντέλα (τα χαρακτηριστικά που μπορεί να πάρει), όπου κάθε μοντέλο έχει και το αντίστοιχο γονίδιο στον γονότυπο του όπως φαίνεται στο σχήμα 2.8.

Έστω όπως φαίνεται στο σχήμα 2.9 πώς το σχέδιο της φούστας αντιστοιχεί σε 4 bit ενός γονιδίου, υπάρχουν κάποια εξτρά bits τα οποία αντιστοιχούν στην απόχρωση που θα μπορούσε να πάρει. Η δομή του γονιδίου ενός γονότυπου φαίνεται στο σχήμα 2.10 . Η αρχικοποίηση του πληθυσμού στον ΔΓΑ γίνεται με τυχαίο τρόπο με επιλογή ανάμεσα στα είδη υπάρχοντα σχέδια και χρώματα που δομούν τα άτομα - λύσεις. Όπως έχει αναφερθεί ο πληθυσμός παραμένει σταθερός και δεν ξεπερνά συνήθως τα 8 άτομα για ευνότητους λόγους. Ο χρήστης μέσω μια διαδραστικής πλατφόρμας, αξιολογεί τα άτομα - λύσεις σύμφωνα με το γούστο του. Ο ΔΓΑ όπως και ο απλώς ΓΑ την δεξαμενή ζευγαρώματος για να επιλέξει την επόμενη γενεά γονέων. Με βάση τον δείκτη

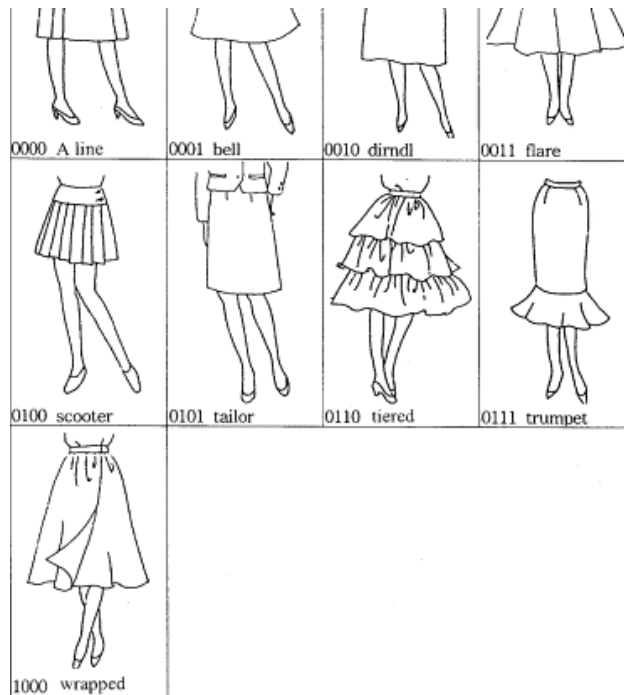


ποιότητας που έχει ορίσει ο χρήστης δημιουργείται ανάλογος αριθμός αντιγράφων. Επιλέγονται μέσα από την δεξαμενή ζευγαρώματος δυο τυχαίοι γονείς για να αναπτυχθεί η επόμενη γενιά. Η αξιολόγηση με διαδραστικό τρόπο τείνει τους δείκτες ποιότητας να μεγαλώνουν ανά γενεά. Ακολουθεί η διαδικασία ανασχηματισμού, με δύο γενετικούς τελεστές. Ο τελεστής διασταύρωσης, που εφαρμόζεται με έναν από τους τρόπους που φαίνονται στο σχήμα 2.2 (Ο τρόπος αυτός είναι προκαθορισμένος από το περιβάλλον και δεν επιλέγεται με τυχαίο τρόπο), πραγματοποιεί μία ή περισσότερες τομές στους γονότυπο των γονέων, για να συνθέσει απογόνους όπως φαίνεται στο σχήμα 2.11.



Σχήμα 2. 8: Γενική λεπτομέρεια ενός γυναικείου φορέματος.

Σχήμα 2. 10: Η μορφή της δομής του γονιδίου που έχει 3 bits για σχέδιο και 3 bits για χρώμα στην σχεδίαση ρούχων με ΔΓΑ.

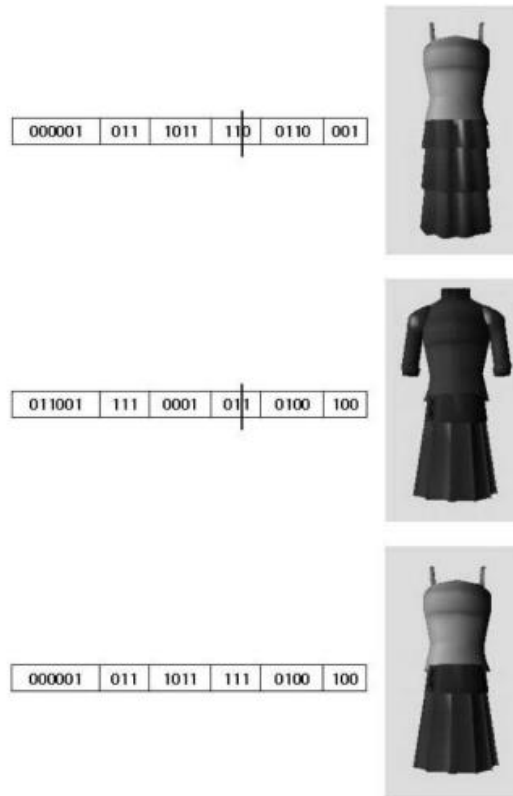


Σχήμα 2. 9: Κωδικοποιημένα στυλ φούστας και μέσης.

Ο τελεστής μετάλλαξης, επηρεάζει μόλις ένα γονίδιο του απογόνου, δίνοντας του μια τυχαία τιμή αλλάζοντας πρακτικά το σχέδιο ενός γονιδίου. Στην σχεδίαση ενδυμάτων, ο τελεστής μετάλλαξης θεωρείται σημαντικός τελεστής, καθώς δίνει κάτι καινούργιο που ο χρήστης δεν είχε προσέξει για να του διεγείρει το ενδιαφέρον. Ο αλγόριθμος ΔΓΑ τερματίζει όταν βρεθεί βέλτιστη λύση, συνεπώς την τερματίζει ο χρήστης, ή μετά το πέρας προκαθορισμένου αριθμού γενεών.

Εν κατακλείδι, η προσέγγιση σχεδίασης ενδυμάτων με την χρήση του ΔΓΑ έχει θετικό πρόσημο. Τα αποτελέσματα που επιφέρει, σύμφωνα και με πειράματα [7], είναι πάντα τα επιθυμητά. Επίσης μπορεί κανείς να σκεφτεί, πως η χρήση του δεν είναι ανάγκη να γίνει από κάποιον ειδικό στην σχεδίαση ενδυμάτων, καθώς γούστο έχουν όλοι οι άνθρωποι. Από την άλλη, σαν όλους τους άλλους ΕΑ δεν παύει να έχει και τα αρνητικά του.

Πρώτο και πιο σημαντικό, η ταλαιπωρία του χρήστη που χρειάζεται, μέχρι να βρεθεί το βέλτιστο σχέδιο. Παρόλο που η “λύση” σε αυτό το πρόβλημα έχει βρεθεί με την διατήρηση μικρού πληθυσμού, δεν αποτελεί και την πιο αποτελεσματική, καθώς μικρός πληθυσμός σημαίνει πιο αργή και με περιορισμένα σχέδια εξέλιξη. Επίσης, για να καλυφθεί όλοι η γκάμα, ενδυμάτων, σχεδίων και χρωμάτων για την περίπτωση αυτού του παραδείγματος απαιτούνται υπολογιστική πόροι, καθώς έχουμε να κάνουμε ίσως με χιλιάδες τρισδιάστατα σχέδια αποθηκευμένα σε μία βάση.



Σχήμα 2. 11: Τελεστής διασταύρωσης του ΔΓΑ στην σχεδίαση ρούχων.

## 2.5 Επίλογος

Συνοψίζοντας, το κεφάλαιο “Εξελικτικοί Αλγόριθμοι”, παρουσιάστηκε και πιθανόν κατανοήθηκε η χρησιμότητα των ΕΑ για την εύρεση βέλτιστων λύσεων σε ποικίλα προβλήματα βασισμένα στην καθημερινή ζωή ενός ανθρώπου και όχι μόνο. Πως η ΕΑ εμπνεύστηκαν από τους εξελικτικούς μηχανισμούς που παρέχει η φύση σε κάθε ζωντανό ον και αυτοί από την πλευρά τους οδήγησαν στο σήμερα. Από όλους τους ΕΑ αναλύθηκαν οι τέσσερις που αποτελούν τους θεμέλιους λίθους, καθώς κάθε άλλη μέθοδος προσέγγισης, πλέον, αναπτύσσεται γύρω από την λογική τους αποτελώντας παραλλαγές τους. Αναλύσαμε για τον καθένα ξεχωριστά, τους γενετικούς τελεστές τους, όπως και τον λόγο εφαρμογής των συγκεκριμένων με βάση της φύσης του προβλήματος τους. Επίσης, είδαμε και κάποια από τα βασικά προβλήματα που παρουσιάζουν κάποιοι από αυτούς, καθώς και τους πιο διαδεδομένους τρόπους που μπορούν να αντιμετωπιστούν. Τέλος, παρουσιάστηκε και μια παραλλαγή του ΓΑ, του ΔΓΑ, με παράδειγμα στην σχεδίαση ενδυμάτων που αποτελεί το στόχο της τρέχουσας Π.Ε. και που θα αποτελέσει κατά ένα βαθμό λύση για την δημιουργία της εφαρμογής.



## Κεφάλαιο 3ο: Εξελικτική Σχεδίαση

### 3.1 Εισαγωγή

Τα υπολογιστικά συστήματα, αποτελούν υποχείριο του ανθρώπου από την ημέρα της εφευρέσεως τους μέχρι σήμερα, απόλυτα εξαρτημένα από τις δυνατότητες που τους παρέχονται από το λογισμικό τους και τα προγράμματα που εμείς οι ίδιοι τα τροφοδοτούμε. Πρακτικά δεν έχουν δική τους βούληση. Τι γίνεται όμως όταν τους παρέχεται έστω και σε ένα βαθμό η ικανότητα να γίνουν “δημιουργικά”;

Ένας από τους πλέον, αρκετούς κλάδους της επιστήμης των υπολογιστών, που δίνει στα υπολογιστικά συστήματα την ελευθερία και την δυνατότητα να επιλέγουν αυτά αντί για εμάς, είναι η εξελικτική σχεδίαση (evolutionary design). Η ΕΣ απαρτίζεται από έναν συνδυασμό, από την σχεδίαση και την βιολογική εξέλιξη που πλέον θεωρείται οικεία από το κεφάλαιο “Εξελικτικοί Αλγόριθμοι”. Στην πράξη, τα λογισμικά που χρησιμοποιούν την ΕΣ, είναι παρόμοια με τα λογισμικά σχεδίασης με την βοήθεια υπολογιστή (computer aided design / CAD), με την διαφορά ότι δανείζονται τις διαδικασίες και την λογική της βιολογικής εξέλιξης και έχουν στόχο την σχεδίαση για παράδειγμα αισθητικά αλλά και τεχνικά σωστών σχεδίων, όπως είναι τα ρούχα, τα αρχιτεκτονικά σχέδια κτιρίων, εξαρτήματα οχημάτων κ.α. .

Η ΕΣ διαθέτει τέσσερις βασικές κατηγορίες ή τύπους με πολλά κοινά στοιχεία στις οποίες διαιρείται. Η εξελικτική βελτιστοποίηση σχεδίων (evolutionary design optimisation), η δημιουργική εξελικτική σχεδίαση (creative evolutionary design), η εξελικτική τέχνη (evolutionary art) και οι εξελικτικές μορφές τεχνητής ζωής (evolutionary artificial life forms). Παρακάτω, θα παρουσιαστούν και οι τέσσερις κατηγορίες της ΕΣ, όπως επίσης και η θεωρία και ένα παράδειγμα της γραμματικής σχημάτων.

### 3.2 Εξελικτική Βελτιστοποίηση Σχημάτων

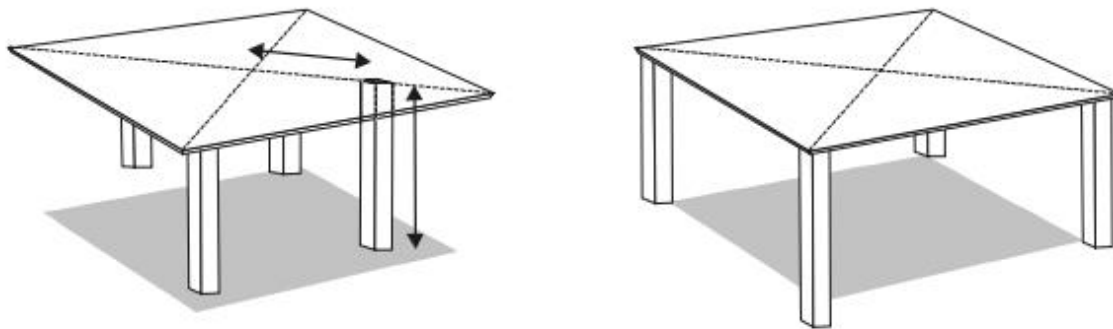
Η χρήση της εξελικτικής βελτιστοποίησης σχεδίων εφαρμόζεται είτε με εκτέλεση λεπτομερούς σχεδιασμού ή με παραμετρικό σχεδιασμό. Ήταν η πρώτη κατηγορία του ΕΣ που επινοήθηκε. Τα τελευταία τριάντα χρόνια, μια τεράστια ποικιλία διαφορετικών μηχανολογικών σχεδίων έχει βελτιστοποιηθεί με την επιτυχημένη εφαρμογή της ΕΒΣ (Holland, 1992; Gen and Cheng, 1997), από τους σφονδύλους (Eby et al., 1997) έως τις γεωμετρίες αεροσκαφών (Husbands et al., 1996). Λιγότερο εφαρμοσμένες παραλλαγές της ΕΒΣ περιλαμβάνουν την βελτιστοποίηση αξιοπιστίας και τεχνικές για την επίλυση του προβλήματος της αναδίπλωσης των πρωτεϊνών (Canal et al., 1998).

Η προσέγγιση που ακολουθεί ένα λογισμικό που εφαρμόζει την ΕΒΣ, μπορεί να διαφέρει σε μεγάλο βαθμό η μία από την άλλη. Αν και υπάρχει όγκος προσεγγίσεων για την επίτευξη αλγορίθμου που αποφέρει αποτέλεσμα, ο τύπος της ΕΒΣ δεν αποτελεί έναν δημιουργικό, ή καινοτόμο ως προς τα σχέδια και την ποικιλία τύπο της ΕΣ. Ένα λογισμικό ΕΒΣ σχεδιάζεται με στόχο της βελτιστοποίηση επιμέρους σημείων ενός υπάρχοντος σχεδίου που προσφέρεται από τον χρήστη. Συνεπώς, επιμέρους σημείο που επιλέγεται για βελτιστοποίηση αποτελεί το γονίδιο και τα χαρακτηριστικά από τα οποία απαρτίζεται θεωρούνται τα αλληλία του γονιδίου και στην συνέχεια εξελίσσονται με την εφαρμογή κάποιου ΕΑ, ωστόσο να βρεθεί η βέλτιστη λύση. Τα σχέδια - άτομα συχνά αξιολογούνται με την

διασύνδεση του λογισμικού ΕΒΣ με ένα άλλο λογισμικό ανάλυσης, το οποίο κρίνει και βαθμολογεί, παρέχοντας για το καθένα ένα δείκτη ποιότητας.

Οι αναπαράσταση των φαινοτύπων για αυτά τα προβλήματα ΕΒΣ αφορούν συγκεκριμένες εφαρμογές, με αποτέλεσμα να αποτελούνται από ήδη υπάρχοντα σχέδια, με τις εξελισσόμενες τιμές των αλληλίων τους να εισάγονται στα αντίστοιχα παραμετροποιημένα σημεία του σχεδίου. Οι τεχνική εμβρυογένεσης (mapping) [10], η οποία αποτελεί μια τεχνική κωδικοποίησης - αποκωδικοποίησης του γονότυπου σε φαινότυπο και το αντίστροφο, συνήθως δεν αποτελεί κομμάτι της εφαρμογής της ΕΒΣ, απλά γιατί δεν είναι απαραίτητη για αυτόν τον τύπο ΕΣ. Κατά συνέπεια, η αναπαράσταση των γονότυπων και των φαινοτύπων, ταιριάζει αρκετά, συχνά πραγματοποιείται με one-to-one mapping μεταξύ γονιδίων και αλληλίων και αποφεύγονται οι διαδικασίες προσθήκης ή διαγραφής γονιδίων από γονότυπους και αλληλίων από φαινότυπους, έτσι δεν εκτελούνται από τον ΕΑ για την ΕΒΣ.

Για την καλύτερη κατανόηση, του τύπου ΕΒΣ, ένα παράδειγμα είναι η βελτιστοποίηση ενός τραπέζιου με τέσσερα πόδια. Η τυπική προσέγγιση για την βελτιστοποίηση, είναι να η παραμετροποίηση των ποδιών του τραπέζιου, με αλληλία το ύψος τους καθώς και την θέση τους στην βάση του τραπέζιου από το κέντρο του όπως φαίνεται στο σχήμα 3.1. Ο ΕΒΣ είναι ικανός να παρέχει λύσης αν όχι βέλτιστες, τουλάχιστον ικανοποιητικές, αλλά δεν είναι ικανός να παρέχει καινούργια σχέδια (όπως να αλλάξει την επιφάνεια του τραπέζιου σε κάποια άλλη).



#### Phenotype:

Table consisting of fixed top and four legs defined by:

*Length of leg 1, Distance of leg 1 from centre*

*Length of leg 2, Distance of leg 2 from centre*

*Length of leg 3, Distance of leg 3 from centre*

*Length of leg 4, Distance of leg 4 from centre*

#### Genotype:

11010110	10101101	10101110	10011010	01101010	10001010	11110010	00101110
----------	----------	----------	----------	----------	----------	----------	----------

*Length 1 Distance 1 Length 2 Distance 2 Length 3 Distance 3 Length 4 Distance 4*

Σχήμα 3. 1: Εφαρμογή εξελικτικής βελτιστοποίησης σε τραπέζι.

Ο τύπος της ΕΒΣ δίνει μεγάλη έμφαση στην εύρεση βέλτιστης λύσης που να πλησιάζει το αναγνωρισμένο παγκόσμιο βέλτιστο, περισσότερο από κάθε άλλο τύπο της ΕΣ. Παρουσιάζει μεγάλη δυσκολία στην βελτιστοποίηση σχεδίων καλής ποιότητας, που συχνά μπορεί και να μην τα καταφέρει. Λόγο αυτής της δυσκολίας, που προσφέρουν στην ΕΒΣ τα παγκόσμια βέλτιστα, οι ερευνητές πλέον,

αναζητούν μεθόδους εξελικτικής αναζήτησης που δεν επικεντρώνεται στα παγκόσμια βέλτιστα. Επιπλέον, τα λογισμικά ανάλυσης που προσφέρουν τους δείκτες ποιότητας τείνουν να μεγαλώνουν τις απαιτήσεις τους ως προς τους υπολογιστικούς πόρους, με αποτέλεσμα να αναζητούνται σε μεγάλο βαθμό λογισμικά με μικρότερο αριθμό αξιολογήσεων πριν παρέχουν το τελικό σχέδιο.

### 3.3 Δημιουργική Εξελικτική Σχεδίαση

Είναι γνωστό και κοινός αποδεκτό ότι οι άνθρωποι μπορούν να είναι δημιουργικοί ως προς την σχεδίαση, αλλά πως ένα υπολογιστικό σύστημα μπορεί να έχει αυτήν την δυνατότητα να σχεδιάσει και παράλληλα να θεωρηθεί από τον ίδιο τον άνθρωπο “δημιουργικό”;

Οι απαντήσεις που μπορούν να δοθούν έχουν αποτυπωθεί στο χαρτί από την Boden και Gero αμφότερους. Ο Gero με την εργασία του «Computers and Creative Design» [8] αναφέρει “Οι έννοιες της δημιουργικότητας βρίσκονται τόσο στις κοινωνικές όσο και στις γνωστικές απόψεις του κόσμου”, έτσι και κάνει την διάκριση του ανάμεσα τους, παραθέτοντας το γεγονός ότι ένας άνθρωπος μπορεί να επιδείξει δημιουργικότητα σχεδιάζοντας, παράλληλα όμως και ένα σχέδιο μπορεί να έχει δημιουργικά χαρακτηριστικά. Εν τέλει, με βάση τον παραπάνω ορισμό καταλήγει στο συμπέρασμα ότι, ένα υπολογιστικό σύστημα μπορεί να γίνει “δημιουργικό”, όταν διερευνά το σύνολο των πιθανών χώρων λύσεων σχεδίασης, εκτός από την εξερεύνηση των παραμέτρων σε επιμέρους χώρους σχεδιασμού. Η Boden με παρόμοια προσέγγιση επί του θέματος [9] καταλήγει στο συμπέρασμα ότι η δημιουργικότητα επιτυγχάνεται ξεπερνώντας τα όρια της αναπαράστασης, βρίσκοντας την λύση που δεν μπορεί να δοθεί μέσω της αναπαράστασης. Η Boden βέβαια δεν θεωρεί πως ένα υπολογιστικό σύστημα είναι ικανό να φτάσει σε αυτά τα επίπεδα και συνεπώς δεν το θεωρεί τόσο “δημιουργικό”.

Τα λογισμικά που εφαρμόζουν την ΔΕΣ έχουν σαν κύριο κοινό χαρακτηριστικό και γνώρισμα, την δυνατότητα σχεδίασης καινούργιων λύσεων είτε με την τροφοδότηση κάποιας καθοδήγησης από τον χρήστη, είτε από το τίποτα (δηλαδή με τυχαία αρχικοποίηση πληθυσμών). Φυσικά για την εύρεση λύσεων και την εξέλιξη παραμέτρων, εξαρτώνται αυστηρά από κριτήρια λειτουργικής απόδοσης. Για την επίτευξη της “δημιουργικής” σχεδίασης και με στόχο την επίτευξη καινοτόμων αλλά και λειτουργικών λύσεων, διαφοροποιούν τον αριθμό των μεταβλητών απόφασης κατά τη διάρκεια της εξέλιξης [11][12].

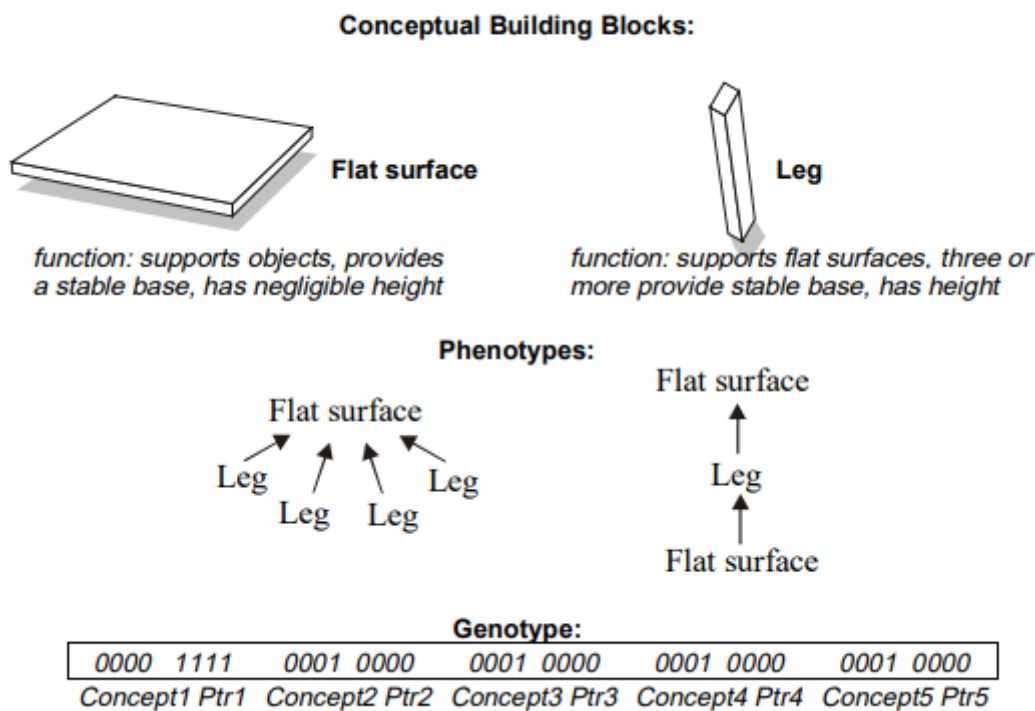
Η έρευνα στον τομέα του ΔΕΣ διατύπωσε δύο κύριες προσεγγίσεις. Και οι δύο περιλαμβάνουν τη χρήση του εξελικτικού υπολογισμού για τη δημιουργία εντελώς νέων καθώς και καινοτόμων σχεδίων από το μηδέν. Η θεμελιώδης εξελικτική σχεδίαση (conceptual evolutionary design) παράγει υψηλού επιπέδου πλαίσια εργασίας και η παραγωγική εξελικτική σχεδίαση (generative evolutionary design) που έχει άμεση αναπαράσταση των σχεδίων. Παρακάτω θα παρουσιαστούν και οι δύο προσεγγίσεις.

#### 3.3.1 Θεμελιώδης Εξελικτική Σχεδίαση

Η προσέγγιση της θεμελιώδης εξελικτικής σχεδίασης για την επίτευξη σχεδιασμού υψηλού επιπέδου και την παραγωγή καινοτόμων λύσεων, εξελίσσει σχέσεις και ρυθμίζει μεταξύ τους. Σε αυτόν τον τύπο εξελικτικού σχεδιασμού, οι σχέσεις και οι ρυθμίσεις του σχεδιασμού υψηλού επιπέδου. Οι έννοιες εξελίσσονται σε μια προσπάθεια δημιουργίας νέων προκαταρκτικών σχεδίων. Ένα καλό παράδειγμα του αυτό είναι το έργο του Pham, ο οποίος περιγράφει το προκαταρκτικό του σύστημα, γνωστό ως TRADES (TRAnsmiSSion DESigner) [13]. Η TRADES χρησιμοποιεί έναν ΓΑ για την εξέλιξη της οργάνωσης ενός συνόλου εννοιολογικών δομικών στοιχείων (building blocks). Όταν

δίνεται ο τύπος εισόδου και η επιθυμητή έξοδος, το λογισμικό δημιουργεί ένα κατάλληλο εννοιολογικό σύστημα μετάδοσης για να μετατρέψετε την είσοδο σε έξοδο. Σε αυτό το σύστημα η εξέλιξη αποσκοπεί στην εύρεση λύσεων μέσω των πιθανών δικτύων (networks) των διασυνδεδεμένων δομικών λίθων.

Γυρνώντας στο παράδειγμα σχεδίασης του τραπέζιου, η προσέγγιση της ΘΕΣ, θα ήταν αρχικά να βρεθεί ένας αριθμός δομικών λίθων με διαφορετική συμπεριφορά. Με την χρήση της εξέλιξης αναζητά έναν τρόπο που θα προσφέρει την κατάλληλη οργάνωση των δομικών λίθων, ώστε να πάρει μια λύση λειτουργική που θα αποτελεί το τραπέζι. Έστω ότι το τραπέζι αποτελείται από δύο δομικούς λίθους. Ένας είναι η βάση του τραπέζιου, και δεύτερος είναι τα πόδια του όπως φαίνεται στο σχήμα 3.2. Οι φαινότυποι αποτελούνται από δίκτυα αυτών των δομικών λίθων. Με λίγα λόγια, το λογισμικό ορίζει τους δομικούς λίθους και δοκιμάζει διαφορετικούς τρόπους, με τους οποίους ο συνδυασμός τους θα μας παρέχει ένα λειτουργικό σχέδιο (π.χ. στην αρχή μπορεί να τοποθετήσει τα πόδια στην πλάγια επιφάνεια του τραπέζιου και όχι από την κάτω πλευρά. Μέσω της εξέλιξης, θα τοποθετήσει τα πόδια σε σημεία πάνω στους άξονες και μακριά από το κέντρο του τραπέζιου). Ανάλογα με την χρήση διαφορετικών ΕΑ, το κάθε σχέδιο θα εξελιχθεί με διαφορετικό τρόπο και διαφορετική ταχύτητα. Οι



Σχήμα 3. 2: Εφαρμογή θεμελιώδους εξελικτικής σχεδίασης σε τραπέζι.

βασικοί ΕΑ αλγόριθμοι είναι σχεδόν πάντα αποτελεσματική πάνω στα λογισμικά ΘΕΣ.

Σε αντίθεση με την ΒΕΣ, η ΘΕΣ είναι ικανή για να δημιουργήσει καινοτόμα εξελικτικά σχέδια, παρόλο την ταχύτητα της και την ανάγκη για καθοδήγηση σε μεγάλο βαθμό σε σχέση με την ΒΕΣ. Τα λογισμικά που εφαρμόζουν την θες δεν είναι σε θέση να ξεφύγουν από τους δομικούς τους λίθους και από τις παρεχόμενες λειτουργίες τους.

### 3.3.2 Παραγωγική Εξελικτική Σχεδίαση

Η προσέγγιση της παραγωγικής εξελικτικής σχεδίασης διαφέρει αρκετά από την προσέγγιση της θεμελιώδης εξελικτικής σχεδίασης, με βασικό στόχο την δημιουργία πάντα καινοτόμων σχεδίων, παρά απλά λειτουργικών. Η βασική διαφορά λοιπόν, μεταξύ των δύο τύπων της ΔΕΣ, είναι ως προς τον τρόπο που η καθεμία πλαισιώνει το αποτέλεσμα. Η ΘΕΣ ως γνωστόν αναζητά δομικούς λίθους για να παράγει ένα υψηλού επιπέδου σύνολο από μεμονωμένα σχέδια και κάνει δοκιμές για να βρει σχέσεις και ρυθμίσεις επάνω τους, ενώ η ΠΕΣ παράγει απευθείας ένα αποτέλεσμα δίνοντας ζωή, σε μια τυχαία μορφή σχεδίων. Ένα υπολογιστικό σύστημα, δέχεται μεγαλύτερη “ελευθερία” σχεδίασης με την εφαρμογή της εν λόγω προσεγγίσεις, πλεονεκτεί στην ποικιλία αναπαράστασης οποιαδήποτε μορφής είναι δυνατόν να αναπαρασταθεί, καθώς και με την βοήθεια της εξέλιξης, μπορεί κάλλιστα να έχει ως αποτέλεσμα απεριόριστα μοτίβα σχεδίων. Ωστόσο, κάποιες φορές, απαιτείται καθορισμένη δυναμική αναπαράσταση και περίπλοκες διαδικασίες αξιολόγησης, που καθιστούν την προσέγγιση ΠΕΣ, δαπανηρή σε υπολογιστική δύναμη και χρόνο.

Επειδή, έμφαση δίνεται στην δημιουργία νέων μορφών σχεδίων, οι αναπαραστάσεις ποικίλουν και είναι αρκετά γενικές, καθεμία ικανή να αντιπροσωπεύσει την μορφολογία που επρόκειτο να αναπαραστήσει. Οι αναπαραστάσεις κυμαίνονται από την άμεση, ή ένα-προς-ένα (αντιστοιχία γονιδίων γονότυπων και φαινότυπων), σε έμμεσες αναπαραστάσεις που χρησιμοποιούν γραμματικές σχημάτων ή κυτταρικά αυτόματα, με ορισμένη προηγμένη εμβρυογένεση για την χαρτογράφηση των γονότυπων σε φαινότυπους [14], [15], [16].

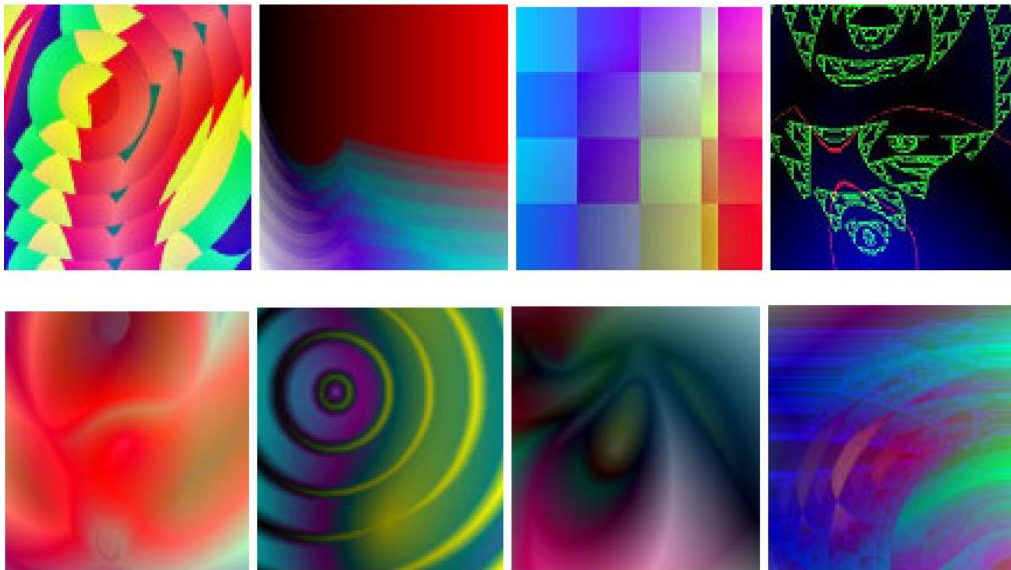
Αυτή την φορά, παίρνοντας για παράδειγμα την κατασκευή μιας μορφής καρέκλας. Η πρώτη αναπαράσταση που θα δοθεί μέσω ενός λογισμικού που χρησιμοποιεί την προσέγγιση της ΠΕΣ, θα αποτελείται από την μια τυχαία τοποθέτηση διάφορων γεωμετρικών σχημάτων με τρόπο που να είναι ικανά να τροποποιηθούν. Τα ίδια τα σχήματα αποτελούν τα γονίδια και οι παραμετρικές τους τιμές (θέση σε τρισδιάστατο χώρο, μέγεθος) τα αλληλία, τα οποία μπορούν να αλλάξουν ελάχιστα. Στόχος φυσικά είναι, μέσω των διαδικασιών αξιολόγησης, να παραμετροποιηθούν οι θέσεις και το μέγεθος σχεδόν κάθε σχήματος, ώστε κάποια στιγμή να δώσουν ως αποτέλεσμα μια μορφή σχεδίου που θα μπορούσε να αποτελέσει μια καρέκλα.

### 3.4 Εξελικτική Τέχνη

Ως η εμπορικά πιο διαδεδομένη προσέγγιση της ΕΣ η εξελικτική τέχνη οφείλεται για την υπάρξει μεγάλου αριθμού έργων τέχνης, που μπορούμε να δούμε και να αγοράσουμε. Παρόλο της φήμης της στον χώρο του εμπορίου, η ΕΤ δεν περνάει πολύ στον ακαδημαϊκό χώρο καθώς προσφέρει μονάχα καλαισθησία παρά λειτουργικότητα στα σχέδιά της, τα οποία αποτελούνται είτε από, απεριόριστες μορφές σχεδίων είτε από εικόνες όπως φαίνεται στο σχήμα 3.3. Βασικό χαρακτηριστικό είναι η ανθρώπινη αξιολόγηση για την επίτευξη της υψηλής αισθητικής, που αποσκοπεί να προσφέρει η ΕΣ, κυρίως γιατί ένα υπολογιστικό σύστημα δεν μπορεί να κατανοήσει την αισθητική έλξη.

Το μέγεθος του πληθυσμού κάθε γενιάς που παράγεται, από ένα λογισμικό που ακολουθεί την προσέγγιση της ΕΤ, είναι επιθυμητά μικρό συνήθως μέχρι 10 άτομα, με σκοπό να επιτρέπει στον χρήστη να παρατηρεί και να αξιολογεί το κάθε άτομο ξεχωριστά, χωρίς να τον κουράζει ένας τεράστιος όγκος σχεδίων ή και χρωμάτων που το λογισμικό μπορεί να αναπαραστήσει. Οι διεπαφές συνήθως είναι παρόμοιες μεταξύ τους, αναπαριστώντας τα σχέδια σε κελιά στοιχισμένα το ένα δίπλα στο άλλο και η αξιολόγηση γίνεται μέσω κάποια φόρμας που έχουν πάνω τους ή πατώντας πάνω τους, έτσι ώστε να οριστεί ο δείκτης ποιότητας τους. Με παρόμοιο τρόπο κατασκευάστηκε και η διεπαφή,

της εφαρμογής της ΠΕ, διότι μια από τις διαδικασίες αξιολόγησης προέρχεται από ανθρώπινη παρέμβαση λόγω καλαισθησίας.



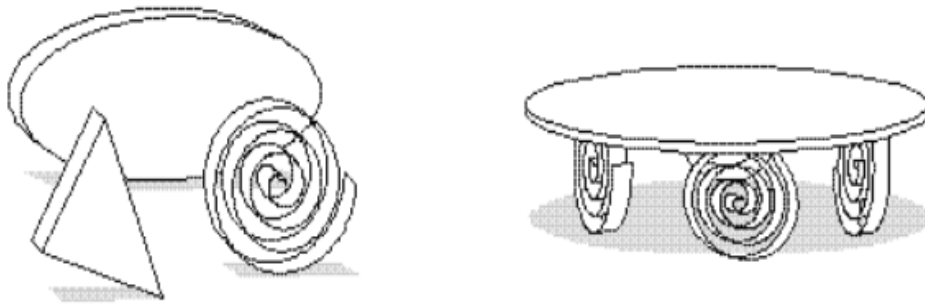
Σχήμα 3. 3: Εξελικτική Τέχνη εφαρμοσμένη για την παραγωγή εικόνων.

Οι μεγαλύτερες διαφορές μεταξύ της ΕΤ και των άλλων τύπων ΕΣ έγκεινται στις αναπαραστάσεις του φαινότυπου τους. Η γκάμα διαφόρων μεθόδων που έχουν εφαρμοστεί για την αναπαράσταση, στην ΕΤ ποικίλει ανάμεσα σε αμέτρητες εναλλακτικές, από αναδρομικούς γραμματικούς κανόνες που χρησιμοποιούν εποικοδομητική συμπαγή γεωμετρία (Todd και Latham, 1992), έως και σε εξισώσεις φράκταλ, όπου η καθεμία εφαρμόζεται με στόχο διαφορετικό αποτέλεσμα.

Στα λογισμικά που εφαρμόζουν την ΕΤ, συνήθως υπάρχουν σταθερές δομές (γεωμετρικά σχήματα), από τις οποίες αρχικοποιείται ο πρώτος πληθυσμός σχεδίων και μέσω της εξέλιξης μεταβάλλονται, ο αριθμός των σχέσεων που δημιουργούνται μεταξύ τους καθώς και τα ίδια τα σχήματα (θέση και μέγεθος). Είναι αρκετά συνηθισμένο ο ίδιος ο χρήστης, να ορίζει τις δομές των σχημάτων που θα εξελιχθούν, πρακτικά καθορίζουν το γονιδίωμα και την εμβρυογένεση που θα εφαρμόσει το λογισμικό ΕΤ. Φυσικά μετά από την τροφοδότηση αρκετών κανόνων του χρήστη στο σύστημα, τα σχέδια θα είναι αρκετά περιορισμένα και συνεπώς θα μοιάζουν αρκετά μεταξύ τους. Επειδή η αξιολόγηση πραγματοποιείται από τον χρήστη, δεν είναι αναγκαστικό η διαδικασία αξιολόγησης να είναι πολύπλοκη και χρονοβόρα. Η εξέλιξη δεν παίρνει μέρος ως γεννήτρια βελτίωσης, αλλά καινοτομίας και για τον λόγο αυτό η εξελικτικοί υπολογισμοί που χρησιμοποιούνται έχουν βασικό χαρακτηριστικό τη μη σύγκλιση (για να αποφεύγεται η οποιαδήποτε επανάληψη στοιχείων μεταξύ των σχεδίων). Εξαιτίας αυτού, η ΕΑ που χρησιμοποιούνται στα λογισμικά ΕΤ δεν εφαρμόζουν την λειτουργία την διασταύρωσης, αντιθέτως πραγματοποιούν μετάλλαξη επάνω στους γονείς (συνήθως είναι μόνο ένας) και μέσου αυτής παράγεται η επόμενη γενεά.

Επιστρέφοντας στο παράδειγμα του τραπεζιού, στο σχήμα 3.4 βλέπουμε αρχικά στο αριστερό πάνω κομμάτι, κάποια από τα σχήματα τα οποία θα μπορούσε πιθανόν να παρέχει ένα λογισμικό ΕΤ, και δεξιά ένα τελικό “καλλιτεχνικό” σχέδιο πλήρως αποτελούμενο από αυτά.

Φυσικά τα σχήματα στο πάνω αριστερά άκρο είναι αυτά που έχει θέση ο χρήστης στο λογισμικό ώστε να αποτελέσουν την σταθερή δομή από την οποία θα αποτελείται το τελικό σχέδιο. Στο παράδειγμα, κάθε τραπέζι, πρέπει να είναι κατασκευασμένο από μια έλλειψη και έναν μεταβλητό αριθμό στροβιλισμών με διαφορετική θέση και περιστροφή. Αυτή η δομή ορίζει στη συνέχεια πόσα γονίδια θα αναπτυχθούν από το σύστημα και πώς θα χρησιμοποιηθούν οι τιμές των γονιδίων για τη δημιουργία των φαινοτύπων, δηλαδή ορίζει το γονιδίωμα και την εμβρυολογία. Με την εξέλιξη των τιμών των γονιδίων, θα προκύψουν μια σειρά από ασυνήθιστα και πιθανότατα καλαισθητικά σχέδια ,αλλά όχι και συχνά λειτουργικά.



### Phenotype:

*Ellipse, width 80, height 50, depth 4, rotated by 90 degrees*

*Spiral, radius 40, curliness 4, depth 6, shifted horizontally by 50, vertically by -20*

*Spiral, radius 40, curliness 4, depth 6, shifted horizontally by 50, vertically by -20, rotated 90 degrees*

*Spiral, radius 40, curliness 4, depth 6, shifted horizontally by 50, vertically by -20, rotated 180 degrees*

*Spiral, radius 40, curliness 4, depth 6, shifted horizontally by 50, vertically by -20, rotated 270 degrees*

### Fixed structure (embryology) using Shape Description Language:

```
Table = { Ellipse ( width, height, depth )
          YZ_Rotate ( angle )
          { Leg
            X_Shift ( distance )
            Y_Shift ( distance )
            Rotate&Duplicate ( angle, #duplicates ) }
          Leg = Spiral ( radius, curliness, depth )
```

### Genotype:

80	50	4	90	40	4	6	50	-20	90	4
<i>width</i>	<i>height</i>	<i>depth</i>	<i>angle</i>	<i>radius</i>	<i>curliness</i>	<i>depth</i>	<i>distance</i>	<i>distance</i>	<i>angle</i>	<i>#duplicates</i>

Σχήμα 3. 4: Εξέλιξη καλλιτεχνικού τραπεζιού.

## 3.5 Εξελικτικές Μορφές Τεχνητής Ζωής

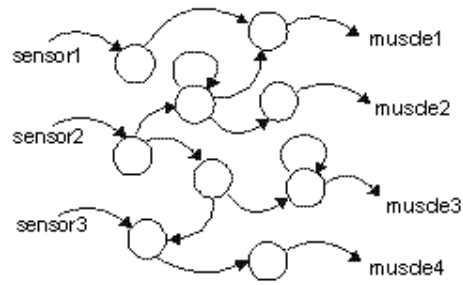
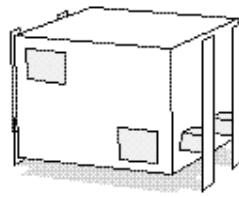
Ο εξελικτικός υπολογισμός διαδραματίζει σημαντικό ρόλο σε πολλές πτυχές του πεδίου της επιστήμης των υπολογιστών της τεχνητής ζωής (artificial life). Τεχνητοί «εγκέφαλοι», στρατηγικές

συμπεριφοράς, μέθοδοι επικοινωνίας, κατανεμημένη επίλυση προβλημάτων και πολλά άλλα θέματα διερευνώνται συνήθως χρησιμοποιώντας ΕΑ και άλλες τεχνικές εξελικτικής αναζήτησης. Αν και όλοι οι τύποι εξελιγμένης ΤΖ, θα μπορούσαν να περιγράφουν σαν πτυχές του εξελικτικού σχεδιασμού, είναι σαφές ότι ορισμένα θέματα εντός της ΤΖ εμπίπτουν στην κατηγορία του ΕΣ με μεγαλύτερη άνεση από ότι σε άλλα. Συνήθως τα κίνητρα πίσω από την δημιουργία ΤΖ είναι σε θεωρητικό επίπεδο και οι έρευνες που πραγματοποιούνται αποσκοπούν στην ανακάλυψη νέων, ανεξερεύνητων μηχανισμών της φυσικής εξέλιξης, ή στην εκμετάλλευση και αντιγραφή των μορφών που ζουν στην φύση.

Το κάθε λογισμικό που προσεγγίζει με την εν λόγω ΕΣ, χρησιμοποιεί συνήθως δικό του, “μοναδικό” τρόπο αναπαράστασης, καθώς υπάρχουν πολλά παραδείγματα, όπως με “πίνακες κανόνων” στα χρωμοσώματα τους για αυτό - αναπαραγωγή (Lohn and Reggia, 1995) και ο Sims χρησιμοποιεί μια ιεραρχική δομή χρωμοσώματος για να ορίσει τόσο τον «εγκέφαλο» όσο και το «σώμα» (Sims, 1994a,b). Πολλές από αυτές τις αναπαραστάσεις είναι εμπνευσμένες από τη δομή του γονότυπου φυσικών οργανισμών και ορισμένοι ερευνητές έχουν προσπαθήσει να εξελίσσουν μορφές ΤΖ με πολύπλοκες εμβρυολογίες. Άλλοι ερευνητές επινοούν τα δικά τους περίπλοκα σχήματα κωδικοποίησης. Οι αναπαραστάσεις είναι εξαιρετικά ευέλικτες και ποικίλου μήκους, απαιτώντας σύνθετους γενετικούς τελεστές από τους ΕΑ που χρησιμοποιούν.

Παραδόξως, η εφαρμογή της ΕΣ για την δημιουργία ΤΖ, παρουσιάζει αρκετά κοινά ως προς αυτήν της ΕΤ, διότι κατά κύριο λόγο χρησιμοποιείται ως εργαλείο εξερεύνησης. Επίσης, μπορούν να χρησιμοποιηθούν για τη δημιουργία πολλαπλών λύσεων, ενσωματώνοντας την εύρεση, την ειδογένεση, τον παρασιτισμό, τον ανταγωνισμό, τη συνεργασία και άλλες προηγμένες μεθόδους. Η αξιολόγηση πραγματοποιείται πάνω στις συμπεριφορές των λύσεων και είτε προσομοιώνεται σε εικονικούς κόσμους, είτε σε με την χρήση ρομπότ (φυσικά η χρήση ρομπότ αποτελεί δαπανηρή μέθοδο σε χρόνο αλλά και χρήμα). Η αρχικοποίηση του πληθυσμού μπορεί να πραγματοποιηθεί χωρίς καθοδήγηση, άλλα και με καθοδήγηση για την μείωση κυρίως του χρόνου. Συνήθως, χρησιμοποιούνται τροποποιημένοι μέθοδοι ΓΑ για την επιτάχυνση της εξέλιξης. Όπως επίσης, σε κάποια λογισμικά υπάρχουν και διαδικασίες αξιολόγησης που μεταβάλλονται με την χρήση εξειδικευμένων τεχνικών γενετικής αναζήτησης.

Επιστρέφοντας λοιπόν στο παράδειγμα του τραπέζιού. Φυσικά σε αυτήν την περίπτωση στόχος δεν είναι να σχεδιαστεί ένα “άψυχο” τραπέζι, άλλα ένα τραπέζι ρομπότ το οποίο θα έχει την όποια χρήση θα μπορούσαμε να του ορίσουμε, όπως να κουβαλάει πράγματα πάνω του. Το σχήμα 3.5 δείχνει τη διπλή φύση αυτών των σχεδίων: Οι EMTZ συνήθως περιλαμβάνουν την εξέλιξη της μορφής του αλλά και του εγκέφαλου. Σε αυτό το παράδειγμα, η μορφή ορίζεται από μια συλλογή γεωμετρικών σχημάτων μεταβλητού μεγέθους, ο «εγκέφαλος» ορίζεται από ένα δίκτυο νευρώνων που λαμβάνουν είσοδο από τα κάποια σχήματα που αποτελούν τους αισθητήρες και παράγουν έξοδο στα μυϊκά σχήματα. Κάθε τμήμα του φαινοτύπου κωδικοποιείται ως χρωμόσωμα μεταβλητού μήκους στον γονότυπο. Η διαδικασία αξιολόγησης κρίνει τους φαινοτύπους σχετικά με την ικανότητά τους να κινούνται διατηρώντας παράλληλα την επίπεδη επάνω επιφάνεια ως έχει Με την πάροδο του χρόνου, η εξέλιξη θα εξελίξει και τα δύο χρωμοσώματα σε ένα για να δημιουργήσει ένα εικονικό πλάσμα ικανό να υποστηρίξει αντικείμενα και κίνηση σε έναν εικονικό κόσμο.



**Phenotype:**

**body**

*type, x, y, z, width, height, depth of block 1    type, x, y, z, width, height, depth of block 2*  
*type, x, y, z, width, height, depth of block 3    .... type, x, y, z, width, height, depth of block 9*

**brain**

*sensor1 excites neuron 1, weight 5    neuron2 inhibits neuron 5, weight 0*  
*neuron1 excites neuron 6, weight 6    sensor3 excites neuron 3, weight 8*  
*sensor2 excites neuron 2, weight 4    .... neuron 3 excites neuron 9, weight 4,*  
*neuron2 excites neuron 4, weight 3    output of neuron 9 to muscle4*

**Genotype:**

**Chromosome 1**

11	11010110	10101101	10101110	10011010	01101010	01101010	...	10001010	10001010	10001010
<i>type1</i>	<i>χρος 1</i>	<i>γρος 1</i>	<i>zρος 1</i>	<i>width 1</i>	<i>height 1</i>	<i>depth 1</i>	...	<i>width 9</i>	<i>height 9</i>	<i>depth 9</i>

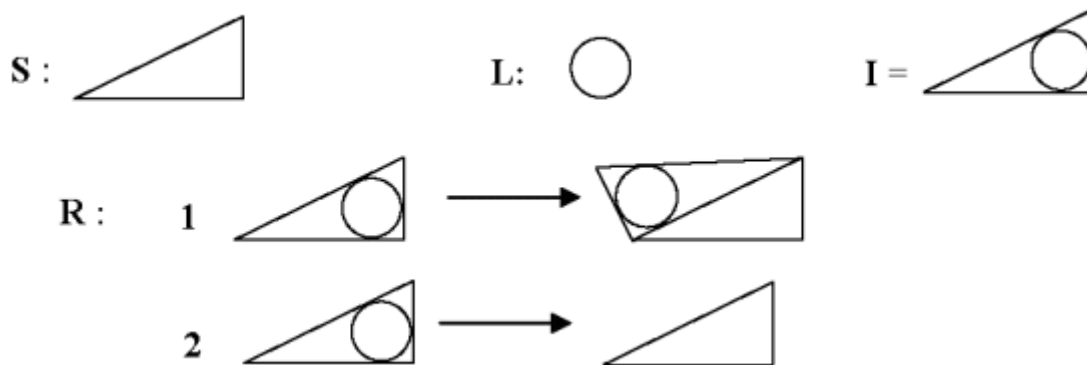
**Chromosome 2**

0000	10	1001	1	0110	1	...	1110
<i>neuron1 marker</i>	<i>neuron1 type (in/out)</i>	<i>neuron1 link1</i>	<i>neuron1 link1 type (ex/inhib)</i>	<i>neuron1 link2</i>	<i>neuron1 link1 type (ex/inhib)</i>	...	<i>neuron9 link2</i>

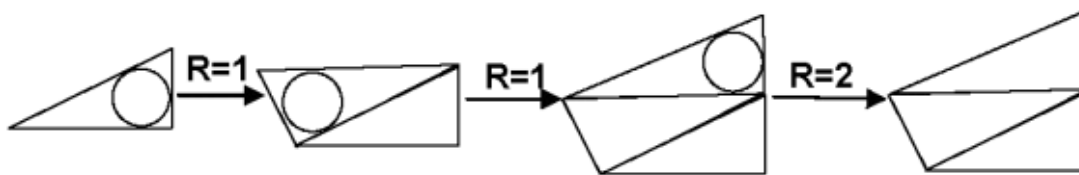
Σχήμα 3. 5: Τεχνητή μορφή ζωής με την εφαρμογή εξελικτικής σχεδίασης

### 3.7 Γραμματικές Σχημάτων

Η θεωρία των γραμματικών σχημάτων (shapes grammars) πρωτοπαρουσιάστηκε το έτος 1972 από τους Stiny και Gips [17], ως μια μέθοδος σχεδίασης, που χρησιμοποιεί γεωμετρικά σχήματα και κανόνες που αναπτύσσονται μεταξύ τους. Αρκετά αργότερα χρησιμοποιήθηκε από την ΕΣ. Μια ΓΣ αποτελείται από ένα σύνολο πεπερασμένου αριθμού σχημάτων (shapes), ένα σύνολο ετικετών (labels), ένα αρχικό σχήμα (initial shape) καθώς και τους κανόνες των σχημάτων που ορίζουν τις χωρικές σχέσεις (rules) μεταξύ των σχημάτων [18]. Θεωρείται μια επίσημη μέθοδος δημιουργίας σχημάτων, μέσω μιας ακολουθίας εφαρμογής κανόνων (σχήμα 1  $\rightarrow$  σχήμα 2), που ξεκινούν από ένα αρχικό σχήμα. Ρόλος του κανόνα είναι να καταφέρει να εναρμονίσει το σχήμα της αριστερής πλευράς σε αυτό του σχήματος της δεξιάς, για αυτό του δίνεται η δυνατότητα φυσικά, να επεξεργαστεί χωρικά το σχήμα. Μπορεί να μετασχηματίσει, να το επιστρέψει, να το μεταβάλλει ή να το ανάκλαση. Παρακάτω στο σχήμα 3.6 μπορούμε να δούμε την εφαρμογή του κανόνα 1 δύο φορές και τον κανόνα 2 μία φορά στο αρχικό σχήμα. Το σχήμα που προκύπτει φαίνεται στο σχήμα 3.7 βήμα προς βήμα.



Σχήμα 3. 6: Απλή εφαρμογή γραμματικών σχημάτων.



Σχήμα 3. 7: Το αποτέλεσμα της εφαρμογής που παράχθηκε από την εφαρμογή γραμματικών σχημάτων.

Από την αρχή θεωρήθηκε χρήσιμη η θεωρία των ΓΣ και διάφορες εφαρμόστηκαν με επιτυχία τόσο σε λογισμικά ανάλυσης όσο και σε σύνθεση. Ως προς την ανάλυση, μια ΓΣ, αποτελεί το βασικό μέσο περιγραφής του στυλ κυρίως σε ότι αφορά οικοδομικά σχέδια, όπως σχέδια από βίλες, εκκλησίες, σπίτια και εξαρτήματα κτιρίων, όπως παράθυρα, όπου εφαρμόστηκε με τεράστια αποτελεσματικότητα, επάνω σε ιστορικό στυλ κτιρίων.

Πέρα από την ικανότητα της σχεδίασης, αποτελούν μεθόδους δημιουργίας καινούργιων γλωσσών σχεδίασης. Βασικό τους όπλο, είναι δημιουργία μιας ποικιλίας σχημάτων και ενός συνόλου κανόνων από την αρχή, γεγονός που τις καθιστά ικανές. Η χρήση μιας ΓΣ σε ένα γενεσιουργό λογισμικό δείχνει να έχει ιδιαίτερο και μεγαλύτερο ενδιαφέρον από ότι σε ένα λογισμικό ανάλυσης, καθώς με την χρήση μόλις μια μικρής ποσότητας σχημάτων και ελάχιστους κανόνες μπορεί να προσφέρει απρόβλεπτα και συνάμα καινοτόμα σχήματα.

### 3.7.1 Γραμματικές Σχημάτων και Εξελικτικοί Αλγόριθμοι

Οι ΕΑ όπως είναι γνωστό αποτελούν αλγόριθμους αναζήτησης, ικανούς να βρουν αν όχι απλά ικανοποιητικές, τότε βέλτιστες λύσεις σε προβλήματα με καθορισμένες συνθήκες. Οι ΓΣ μπορούν να παρέχουν αξιοπιστία προς την δημιουργία σχεδίων καλαισθητών αλλά και εξαιρετικά οργανωμένων. Ο συνδυασμός των ΓΣ και των ΕΑ παράγει ένα αποτέλεσμα όπου ο χώρος αναπαράστασης καθορίζεται από μια ΓΣ την οποία στην συνέχεια εξελίσσει κάποιος ΕΑ (συνήθως ΓΑ). Ο ΓΣ με ΕΑ έχει προταθεί από πολλούς. Η καινοτομία στην τρέχουσα προσέγγιση έγκειται στον τρόπο με τον οποίο η ΓΣ που χρησιμοποιείται, συνδέεται με τον ΕΑ, μέσω της χρήσης κώδικα σχήματος (shape code).

#### 3.7.1.1 Κώδικας Σχήματος

Ο συνδυασμός των δύο παραπάνω μεθόδων καλείται “Εξέλιξη Σχημάτων”. Η διαδικασία παραγωγής περιλαμβάνει μια αλυσίδα κανόνων ΓΣ που ξεκινούν με ένα αρχικό σχήμα. Πάντα, ορίζεται μικρός αριθμός σχημάτων, καθώς και κανόνων. Κατά συνέπεια, καθιστά εύκολη την αναπαράσταση ενός σχεδίου του οποίου ο γονότυπος θα αποτελείται από μια μικρή συμβολοσειρά (string), στην οποία επίσης φαίνεται η εφαρμογή των κανόνων στο σχήμα. Αυτή λοιπόν η συμβολοσειρά καλείται “κώδικας σχήματος”.

Χρησιμοποιώντας τον ΚΣ ως γονότυπο προσφέρει τρία πολύ σημαντικά οφέλη. Πρώτον, αυτό διασφαλίζει ότι όλα τα σχέδια που δημιουργούνται κατά τη διάρκεια της εξελικτικής διαδικασίας είναι έγκυρα στη γλώσσα. Επίσης σημαίνει ότι οι λειτουργίες του γενετικού αλγορίθμου που εφαρμόστηκαν στον γονότυπο, όπως η διασταурώση και η μετάλλαξη, αλλάζουν την επιλογή και τη σειρά των κανόνων γραμματικής σχήματος που χρησιμοποιούνται για τη δημιουργία μιας γενεάς σχεδίων, τα οποία είναι δυναμικά πολύ σημαντικές αλλαγές. Επιπλέον, η διαδικασία μετατροπής του ΚΣ σε τρισδιάστατη μορφή είναι απλώς θέμα εφαρμογής της ακολουθίας του ΚΣ στη γραμματική του σχήματος, παράγοντας έτσι κατάλληλη γεωμετρία.

Ωστόσο, χρησιμοποιώντας τον ΚΣ ως τον γονότυπο μπορεί να εισάγει επιπλοκή στο λογισμικό. Η χωρική επικάλυψη των αφηρημένων σχημάτων συνήθως δεν αποτελεί κάποιο πρόβλημα, αλλά όταν τα σχήματα αντιπροσωπεύουν κάποια φυσική οντότητα μπορούν με ιδιαίτερη ευκολία να δημιουργηθούν επιπλοκές. Η λύση στο πρόβλημα που μπορεί να προκληθεί είναι η απομάκρυνση των ΚΣ που το δημιουργούν πριν την διαδικασία αξιολόγησης. Υπό την προϋπόθεση, ότι τα άτομα του πρώτου πληθυσμού αποτελούνται από έγκυρα και μόνο άτομα, τότε ο μόνος τρόπος δημιουργίας μη έγκυρων ΚΣ παράγεται μέσω των λειτουργιών διασταύρωσης και μετάλλαξης. Συνεπώς, απαιτείται προσαρμογή μηχανισμών ελέγχου μετά τις παραπάνω λειτουργίες, ώστε να διασφαλιστεί ότι, τα νέα στοιχεία που προστίθενται, θα τοποθετηθούν σε άδειες περιοχές.

### 3.8 Επίλογος

Εν κατακλείδι, το κεφάλαιο “Εξελικτική Σχεδίαση” επιχειρεί να εξηγήσει τα θεμελιώδη ζητήματα που αφορούν τον εξελικτικό σχεδιασμό από υπολογιστές. Στην αρχή του κεφαλαίου δίνει το ερώτημα αν τα υπολογιστικά συστήματα είναι ικανά να δημιουργήσουν, καθώς στην συνέχεια ξεκινά να αιτιολογεί την έννοια του ρήματος “δημιουργώ”. Καθώς, το παραπάνω ερώτημα μένει να αιωρείται, παρουσιάζει την έννοια του ΕΣ, γιατί η ΕΣ και γενικά οποιαδήποτε μέθοδος σχεδίασης θα μπορούσε να εμπνευστεί από την εξέλιξη και συγκεκριμένα τους ΕΑ και τις τέσσερις βασικές κατηγορίες στις οποίες διαιρείται. Για την κάθε μια από αυτές επεξηγεί τον τομέα εφαρμογής, δηλαδή τον στόχο στον οποίο προσπαθούν να φτάσουν και γιατί αυτός ο στόχος μπορεί να επιτευχθεί μέσω της εφαρμογής τους. Παρουσιάζει τον τρόπο λειτουργίας τους και πως αυτός επιστρατεύει την εξέλιξη με παραδείγματα, εύκολα και κατανοητά. Τέλος, παρουσιάζει την ΓΣ, που αποτελεί μια διαδοδομένη μέθοδο σχεδίασης, όσον αφορά την δημιουργία με γεωμετρικά σχέδια και την αποτελεσματικότητα της σε αυτήν.

## Κεφάλαιο 4ο: Παρουσίαση Προβλήματος

### 4.1 Εισαγωγή

Είναι γνωστό ότι η ενδυμασία ή αλλιώς τα ρούχα, μας πλαισιώνουν σχεδόν από την ώρα μηδέν της ανθρωπότητας πάνω στον πλανήτη. Στην αρχή υπήρχε η ανάγκη του τότε ανθρώπου να προστατευτεί από τις θερμοκρασίες και τις καιρικές συνθήκες, άλλα με το πέρασ του χρόνου, περισσότερες ανάγκες δημιουργήθηκαν στο ανθρώπινο γένος, όπως αυτή του να κρύψει την γύμνια του και να καλύψει τα ευαίσθητα σημεία του, ή η ανάγκη του να προστατέψει το σώμα του από πιθανούς κινδύνους. Φυσικά, από τότε που ο άνθρωπος ξεκίνησε να αντιλαμβάνεται ότι έχει φυσική εικόνα, ήρθε και η ανάγκη του να δείχνει περιποιημένος και καλαίσθητος. Έτσι ξεκίνησε ο ίδιος να σχεδιάζει και να διακοσμεί τα ρούχα του, έχοντας στο μυαλό του να ικανοποιεί τις παραπάνω ανάγκες, εν συνεχεία η σχεδίαση ενδυμάτων και δημιουργία έγινε επάγγελμα. Η σχεδίαση, αποτέλεσε την αρχή και το βασικό πρόβλημα που απασχόλησε καθώς έπρεπε να συνδυάζει, την σιλουέτα, το υλικό, το χρώμα την πρακτικότητα ως προς τον λόγο που εξυπηρετεί, καθώς και την καλαισθησία που δείχνει διαφορετική από μάτι σε μάτι. Συνεπώς, δημιουργήθηκαν διάφορες τεχνικές και βήματα που ο καθένας χρησιμοποιούσε με δικό του τρόπο, χωρίς πάντα αυτές να είναι οι βέλτιστες.

Ο κλάδος της σχεδίασης μόδας, υφίσταται από πολύ παλιά και εν τέλει ενώθηκε ως μια “οντότητα” σε παγκόσμια κλίμακα. Έτσι δημιουργήθηκαν και η κατάλληλη βιομηχανία, η οποία, από τότε άρχισε να ακολουθεί κοινό δρόμο λογικής. Σήμερα, η ανάγκες έχουν μεγαλώσει. Στην εξίσωση έχουν μπει, ο χρόνος καθώς πρέπει να παράγεται κάθε εποχή και διαφορετική κολλεκτιόν, καθώς και η ανάγκη ελάττωσης του κόστους με την χρήση φθηνών και ευρετών υλικών στην αγορά. Με την ανάπτυξη της τεχνολογίας, η βιομηχανία μόδας δεν άργησε να την προσαρμόσει στις τάξεις της και να αρχίσει να σχεδιάζει και να υπολογίζει μέσω αυτής.

Evolution of clothing	Periods in time	Secondary skins
Realization of need for primary body protection		
Discovery of fabric-making processes		
Development of clothing from function to fashion		
Refinement of clothing due to technological and material evolution		

Σχήμα 4. 1: Η εξέλιξη των ενδυμάτων από την “αρχή” μέχρι και σήμερα

## 4.2 Σχεδίαση Ενδυμάτων

Η σημερινή σχεδίαση ενδυμάτων (clothing design) της βιομηχανίας της μόδας, αποτελεί ένα συνδυασμό γνωστών βημάτων, που ακολουθούν οι σχεδιαστές, ο καθένας με την δική του ξεχωριστή προσέγγιση, από την αρχική φάση της σύλληψης μιας γενικής ή πιο ιδιαίτερης εικόνας, μέχρι και την ολοκλήρωση ενός φορητού και υπαρκτού ενδύματος. Πλέον, φυσικά, με την βοήθεια της τεχνολογίας, συγκεκριμένα των συστημάτων CAD και των EA, κάποια από τα βήματα αυτά έχουν γίνει λιγότερο χρονοβόρα.

### 4.2.1 Βήματα Σχεδίασης

Τα βήματα λοιπόν αυτά δεν ισχύουν σε κάθε περίπτωση ΣΕ. Συγκεκριμένα υπάρχουν εννιά γνωστά βήματα που παρουσιάζονται παρακάτω κάποια από τα πιο σημαντικά.

- **Βοήθεια περιγραφής :** Οι σχεδιαστές δέχονται συχνά παραγγελίες από υπεύθυνους τους ή από πελάτες που επιθυμούν να σχεδιάσουν συγκεκριμένα σχέδια μόδας ή να δημιουργήσουν μια νέα συλλογή. Συνεπώς, είναι δουλειά του σχεδιαστή, να ακολουθήσει τις απαιτήσεις που του έχουν δοθεί και έτσι προσανατολίζει την δουλειά του για να τις ικανοποιήσει. Αυτό μπορεί να περιλαμβάνει ένα συγκεκριμένο στυλ, η μια σιλουέτα, ίσως να καθορίσει το υλικό που θα χρησιμοποιηθεί για να βρισκεται εντός του προϋπολογισμού. Συνήθως αποτελεί ένα πολύ βοηθητικό βήμα, φυσικά δεν μπορεί να ισχύει σε όλες τις περιπτώσεις.

- **Αναζήτηση έμπνευσης :** Η έμπνευση είναι κάτι που δεν έρχεται πάντα στο κεφάλι του σχεδιαστή από μόνο του και έτσι, έτσι μπορεί να αναζητήσει την έμπνευση από την καθημερινή ζωή. Συνήθως, συμβουλευεται αρκετά και τις τάσεις της μόδας ώστε να μην υπάρχει μεγάλη απόκλιση από την γνώμη του αγοραστικού κοινού.
- **Δημιουργία σκίτσων :** Με την δημιουργία σκίτσων ο σχεδιαστής πρακτικά οπτικοποιεί την έμπνευσή του στο χαρτί. Τα σκίτσα αποτελούν φυσικά τον βασικό λίθο της σχεδίασης ενδύματων καθώς δίνουν με τα φυσικά τους στοιχεία (ραφές, μήκος μανικιών, λαιμόκοψη κ.α.), μια διαφορετική οπτική γωνία από μια απλή ιδέα και αποτελούν το προσχέδιο της δημιουργίας του ενδύματος.
- **Επιλογή υλικού κατασκευής :** Η επιλογή υλικού, προαναφέρθηκε για λόγους οικονομίας, αλλά φυσικά δεν ικανοποιεί μονάχα αυτή την ανάγκη ιδιαίτερα στο μυαλό ενός σχεδιαστή. Για τον σχεδιαστή το υλικό που θα χρησιμοποιηθεί για να δώσει ζωή στο ένδυμα, αποτελεί ένα μεγάλο κομμάτι του στυλ που επιθυμεί να του δώσει, καθώς επηρεάζει σε μεγάλο βαθμό το τελικό αποτέλεσμα της αίσθησης που δίνει.
- **Επιλογή χρώματος :** Το χρώμα που επιλέγει ο σχεδιαστής, παίζει ένα μεγάλο ρόλο ως προς την διάθεση που προσπαθεί να περάσει στον ένδυμα. Μπορεί να καθορίσει την επισημότητα του ενδύματος. Είναι σημαντικό για το αγοραστικό κοινό, να μπορεί να το αποδεχτεί, για αυτό συνήθως ο σχεδιαστής ακολουθεί χρώματα που υπάρχουν στην τάση της μόδας, ή κοινώς αποδεκτά χρώματα, με εξαιρέσεις.
- **Θεώρηση σιλουέτας :** Η θεώρηση σιλουέτας αποτελεί ένα μεγάλο βήμα, στο οποίο ο σχεδιαστής πρέπει να σκεφτεί προσεκτικά ποια προσέγγιση θα ακολουθήσει καθώς υπάρχουν πολλές. Ο σχεδιαστής πρέπει να λάβει υπόψη τον στόχο - αγοραστή, κοινώς την ποικιλία σωματότυπων, τις αναλογίες κ.α. Αποτελεί ένα βήμα που είναι αρκετά εύκολο να παραληφθεί στο πέρας των βημάτων κι αυτό μπορεί να αποτελέσει μεγάλο πρόβλημα.
- **Δημιουργία πρωτοτύπου :** Φτάνοντας κοντά στην ολοκλήρωση του ενδύματος, είναι αναγκαίο να κατασκευαστεί ένα η περισσότερα δείγματα με την χρήση κοινού υλικού κατασκευής και όχι με αυτό που επιλέχθηκε να χρησιμοποιηθεί τα οποία θα φορεθούν και θα υποστούν αξιολόγηση σχετικά με το αν είναι όντως άνετα, τηρούν τον σκοπό τους και φυσικά αν δίνουν την ίδια αίσθηση σε απτή μορφή με αυτή του σχεδίου.

Τα υπόλοιπα βήματα αποτελούνται από την δημιουργία του ενδύματος καθώς και τον έλεγχο του με το υλικό που έχει επιλεγεί για κατασκευή. Αυτά τα δύο βήματα είναι φυσικά σημαντικά για τον σχεδιαστή, αλλά δεν αποτελούν θέμα ανάπτυξης για την τρέχουσα Π.Ε. καθώς δεν σχετίζονται με την εφαρμογή.

### 4.3 Εξελικτική Προσέγγιση Στην Σχεδίαση

Με την εφαρμογή εξελικτικών τεχνικών στον τομέα της σχεδίαση κατάφεραν να ξεπεραστούν μεγάλοι όγκοι προβλημάτων, που περιόριζαν την σχεδίαση μέχρι πρότινος. Οι επιλογή λύσεων με τυχαίο και πιθανολογικό τρόπο, όπως αυτός εφαρμόζεται στους ΕΑ δεν εγγυάται την εύρεση μιας βέλτιστης λύσης, παρόλα αυτά οι λειτουργίες των γενετικών τελεστών που εφαρμόζονται, εντείνουν τον αρχικό πληθυσμό, να πλησιάζει σε αυτό με επιτυχία. Σημαντικό μέρος είναι αυτό της γονοτυπικής αναπαραστάσεις των λύσεων, που μέσω της αλληλουχίας της βοηθάει τους τελεστές μετασχηματισμού όταν εφαρμόζονται πάνω σε ένα σχέδιο χωρίς να απαιτεί συχνά ιδιαίτερα πολύπλοκες μαθηματικές διατυπώσεις.

Για την αντιμετώπιση του προβλήματος πολλαπλών αντικειμενικών στόχων, οι ΕΑ τα αντιμετωπίζουν ως μαθηματικά προγραμματιστικά μοντέλα τα οποία διαχωρίζουν σε τρεις κατηγορίες, τις απλές συναθροιστικές προσεγγίσεις (plain aggregating approaches), τις προσεγγίσεις που βασίζονται στο κατά Παρέτο κριτήριο (Paetro-based approaches) και τις μια συναθροιστικές προσεγγίσεις που βασίζονται σε πληθυσμούς (non aggregating non-Paetro population based approaches).

#### 4.3.1 Εξελικτική Προσέγγιση Στην Σχεδίαση Ενδυμάτων

Αρχικά η πρώτη προσέγγιση που έγινε για την δημιουργία συστημάτων σχεδίασης ενδυμάτων, πραγματοποιήθηκε με απλά CAD λογισμικά αλλά πολλές φορές ήταν δύσκολα στην χρήση από τους σχεδιαστές. Στις αρχές της χιλιετίας ο Kim και ο Cho [7] πρότειναν το πρώτο λογισμικό σχεδίασης ενδυμάτων με την χρήση διαδραστικού γενετικού αλγορίθμου, έτσι η δημιουργία διαδραστικών εξελικτικών συστημάτων έγινε η βασική ιδέα.

Τα διαδραστικά εξελικτικά συστήματα είναι σε θέση να παράγουν και εξελίσσουν τεράστια ποικιλία εναλλακτικών σχεδίων. Χρησιμοποιώντας τέτοια συστήματα, οι χρήστες πρέπει να αλληλοεπιδρούν με το σύστημα συνεχώς στην δημιουργία κάθε γενεάς από πραγματοποιώντας αξιολογήσεις και επιλογές των σχεδίων που παράγονται και εξελίσσονται. Βασικό χαρακτηριστικό των ΔΕΣ είναι ότι η αξιολόγηση των σχεδίων είναι υποκειμενική καθώς πραγματοποιείται εξολοκλήρου με ανθρώπινη παρέμβαση, είτε έχει μεγάλο ποσοστό της τελικής απόφασης, καθώς είναι πιθανό να υπάρχουν διαδικασίες αξιολόγησης που ήδη παρέχονται από το ίδιο το λογισμικό. Έχει υπάρξει ένας αριθμός ΔΕΣ που υλοποιεί γραφικά, τέχνη μέσω και βιομηχανικό σχεδιασμό. Ο σχεδιασμός ενδυμάτων είναι ένας σημαντικός πρόβλημα στα ΔΕΣ γιατί είναι δύσκολο να διατηρηθεί μια λειτουργία αξιολόγησης για σχέδια. Από τότε μέχρι και σήμερα έχουν προταθεί αρκετά συστήματα που χρησιμοποιούν ΔΓΑ.

Ο ΔΓΑ «αλληλοεπιδρά» με τους χρήστες και ενσωματώνει την προτίμηση του χρήστη στη διαδικασία της εξέλιξης. Είναι προφανές ότι είναι ανώτερο από τον απλό ΓΑ όταν η λειτουργία αξιολόγησης για την εύρεση του δείκτη ποιότητας είναι δύσκολο να σχεδιαστεί. Η υποκειμενική αξιολόγηση του χρήστη καθορίζει την καταλληλότητα της λύσης. Ο ΔΓΑ ως τυπικό ΔΕΣ εφαρμόστηκε με επιτυχία στη βελτιστοποίηση, το σχέδιο και την τέχνη, τη μουσική σύνθεση, χειρισμός γλώσσας και έλεγχο ομοιοκαταληξίας, γνώση απόκτηση και εξόρυξη δεδομένων. Ωστόσο, αποτελεί μια επέκταση του ΓΑ, η σύγκλιση και η ταχύτητα του είναι χαμηλή. Επιπλέον, απαιτούνται πολλές φορές αξιολόγηση, που αποτρέπει τις εφαρμογές του από χρήση γιατί κουράζουν τον σχεδιαστή.

#### 4.3.1.1 Διαδραστική Αξιολόγηση

Όσον αφορά, συστήματα εξελικτικής σχεδίασης οι μηχανισμοί αξιολόγησης που παρέχονται είναι υπεύθυνη για την καταλληλότητα των λύσεων σχετικά με τους ζητούμενους στόχους του προβλήματος. Συνήθως αυτοί παρέχονται, είτε με καθοδήγηση από τον χρήστη, είτε είναι ήδη προσαρμοσμένοι. Στην περίπτωση της σχεδίασης με ένα πρόβλημα που δεν έχει αντικειμενικό βέλτιστο, καθώς ο πιο σημαντικός στόχος είναι να δείχνει καλαισθητό. Στην συγκεκριμένη περίπτωση απαιτείται η ανθρώπινη παρέμβαση.

Σε περιπτώσεις μεγάλων οίκων μόδας, καλούνται ειδικές ομάδες αξιολόγησης που αποτελούνται από ειδικούς σχεδιαστές. Αυτοί, μέσω μιας φιλικής διεπαφής παρατηρούν τα σχέδια και βαθμολογούν σύμφωνα με την όψη και άλλους παράγοντες στους οποίους εξειδικεύονται. Ο μέσος όρος της βαθμολογίας αποτελεί και τον δείκτη ποιότητας. Σε άλλες περιπτώσεις αυτή η διαδικασία γίνεται αποκλειστικά από έναν και μόνο σχεδιαστή, ενώ υπάρχουν και οι περιπτώσεις στις οποίες υπάρχει και κάποια άλλη διαδικασία αξιολόγησης, που παρέχεται από το σύστημα ή την τροφοδοτεί ο χρήστης.

#### 4.4 Επίλογος

Συνοψίζοντας, ένα τελικό μοντέλο για την δημιουργία ενός συστήματος σχεδίασης ενδυμάτων και για να καλύψουν σχεδόν όλο το φάσμα των βημάτων που παρουσιάστηκαν, καθώς κάποια από τα βήματα, ξεφεύγουν από τις δυνατότητες ενός υπολογιστικού συστήματος είναι η χρήση διαδραστικού εξελικτικού αλγορίθμου χωρίς να αποτελέσει απαραίτητα τον ΔΓΑ καθώς υπάρχουν παραλλαγές στις λειτουργίες του. Καθώς και περιορισμοί σε μεγάλο μέρος των γονιδίων των λύσεων ώστε να παραμένουν προσγειωμένα στην πραγματικότητα.



## Κεφάλαιο 5ο : Μοντέλο Προσέγγισης και Εφαρμογή

### 5.1 Εισαγωγή

Στην παρούσα Π.Ε. πραγματοποιήθηκε μια προσπάθεια αντιμετώπισης του προβλήματος της σχεδίασης ενδυμάτων, με την χρήση διαδραστικού εξελικτικού αλγορίθμου. Δόθηκε βάση, στην δημιουργία ενός διαδραστικού εξελικτικού αλγορίθμου, εμπνευσμένου από αυτόν των Kim και Cho [7]. Οι βασικές διαφορές που πραγματοποιήθηκαν, είναι πάνω στους μηχανισμούς επιλογής και αναπαραστάσης. Οι αναπαραστάσεις για κάθε πρότυπο ένδυμα (μπλούζα, παντελόνι), πραγματοποιήθηκε με τρόπο εμπνευσμένο της γραμματικής σχημάτων χωρίς να ακολουθούν ρητά την θεωρία της. Αν και στην αρχή της δημιουργίας της εφαρμογής, υπήρξε η φιλοδοξία να συμπεριφέρεται ως ικανή ώστε να σχεδιάζει πραγματικά σχέδια ενδυμάτων, μετά από αρκετή ερευνά αποδείχθηκε πως ο συγκεκριμένος στόχος ήθελε μεγάλη σχεδιαστική ικανότητα, καθώς και πραγματικές διαστάσεις τις οποίες μπορούσαν να παρέχουν μόνο ειδικοί του χώρου. Έτσι η εφαρμογή επικεντρώθηκε περισσότερο στις εξελικτικές τεχνικές της.

Για την υλοποίηση των παραπάνω δημιουργήθηκε μια εφαρμογή σε Java και μια διεπαφή που αναπτύχθηκε με JavaFX στα στοιχεία του περιβάλλοντος του SceneBuilder που παρέχει η ίδια καθώς υπήρξε και χρήση CSS. Εδώ πρέπει να αναφερθεί πως η διεπαφή είναι λειτουργική μόνο σε πλήρη οθόνη και δεν αναπτύχθηκε σε responsive design. Σε επίπεδο ανάπτυξης και ανάλυσης η εφαρμογή φτιάχτηκε εξ ολοκλήρου από την αρχή. Δεν υπήρξε η χρήση κάποια βάσης για να μην επιβαρύνει το πρόγραμμα κατά την λειτουργία του, καθώς δεν υπάρχει η ανάγκη, συνεπώς χάνει όλα τα δεδομένα κατά τον τερματισμό του.

### 5.2 Τεχνολογίες

Στην υποενότητα “Τεχνολογίες” γίνεται ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής καθώς και γιατί προτιμήθηκαν από άλλες. Αναλύονται τα πακέτα της Java και της JavaFX, καθώς και γιατί επιλέχθηκε η JavaFX για την ανάπτυξη της διεπαφής.

#### 5.2.1 Java

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού. Δημιουργήθηκε μέσω της C++ με τέτοιο τρόπο ώστε να μην είναι δύσκολη ως προς την κατανόηση της. Αρχικά η Java είχε στόχο κινητές συσκευές που θα ανταλλάσσαν δεδομένα μέσω δικτύων, κατασκευάστηκε για να περιβάλλει υψηλό επίπεδο ασφαλείας. Επίσης είναι ανεξάρτητη πλατφόρμας. Η Java επιλέχθηκε για την υλοποίηση της εφαρμογής γιατί εκτός από ότι προσφέρει όλα τα παραπάνω, υπάρχει και μεγάλη εξοικείωση με την γλώσσα, παρόλο που δεν χρησιμοποιήθηκε σε μεγάλο βάθος.

##### 5.2.1.1 Πακέτα Java

Παρακάτω παρουσιάζονται τα πακέτα της Java που χρησιμοποιήθηκαν για την υλοποίηση μεθόδων και λειτουργιών της εφαρμογής.

- **java.util.Arrays** : Το πακέτο Arrays της Java περιέχει ένα σύνολο μεθόδων για την διαχείριση πινάκων, καθώς και τρόπους αναπαραστάσης των πινάκων σε λίστες.

- **java.util.Random** : Το πακέτο Random της Java είναι υπεύθυνο για την καταχώρηση τυχαίων τιμών σε παραμέτρους. Εμπεριέχει μεθόδους που αναθέτουν στην τύχη τιμές είτε αυτές είναι αριθμοί είτε συμβολοσειρές.
- **java.awt.Toolkit** : Το πακέτο Toolkit της Java είναι η αφηρημένη υπερκλάση όλων των πραγματικών υλοποιήσεων του Abstract Window Toolkit. Οι υποκλάσεις της κλάσης Toolkit χρησιμοποιούνται για τη σύνδεση των διαφόρων στοιχείων σε συγκεκριμένες υλοποιήσεις εγγενών kit εργαλείων.
- **java.awt.Rectangle** : Το πακέτο Rectangle της Java περιέχει μεθόδους δημιουργίας ενός rectangle. Το πάνω αριστερό άκρο του τοποθετείτε στους άξονες X και Y καθώς ορίζεται το ύψος και το πλάτος του. Επίσης υπάρχουν και μέθοδοι που ορίζουν το στυλ του rectangle.
- **java.awt.Robot** : Το πακέτο Robot της Java χρησιμοποιείται για την δημιουργία εγγενών συμβάντων εισόδου στο σύστημα για σκοπούς αυτοματοποίησης δοκιμών, διαδικασιών αυτόματης εκτέλεσης καθώς και άλλων εφαρμογών όπου απαιτείται η χρήση του ποντικιού ή του πληκτρολογίου (Events).
- **java.awt.image.BufferedImage** : Το πακέτο της Java BufferedImage είναι μια υποκλάση της κλάσης Image. Χρησιμοποιείται για το χειρισμό των δεδομένων εικόνας. Ένα BufferedImage είναι κατασκευασμένο από ColorModel δεδομένων εικόνας. Όλα τα αντικείμενα BufferedImage έχουν συντεταγμένη πάνω αριστερή γωνία (0, 0).
- **java.io.File** : Το πακέτο File είναι η αναπαράσταση της Java ενός ονόματος διαδρομής αρχείου ή καταλόγου (path). Η File περιέχει πολλές μεθόδους επεξεργασίας για το όνομα διαδρομής, διαγραφή και μετονομασία αρχείων, δημιουργία νέων καταλόγων, λίστα με τα περιεχόμενα ενός καταλόγου και προσδιορισμό πολλών κοινών χαρακτηριστικών αρχείων και καταλόγων.
- **javax.imageio.ImageIO** : Το πακέτο ImageIO της Java περιέχει μεθόδους στατικής ευκολίας για τον εντοπισμό των ImageReaders και ImageWriters και την εκτέλεση απλής κωδικοποίησης και αποκωδικοποίησης.
- **javax.swing.filechooser.FileSystemView** : Το πακέτο FileSystemView της Java είναι η πύλη του JFileChooser στο σύστημα αρχείων. Δεδομένου ότι το JDK1.1 File API δεν επιτρέπει την πρόσβαση σε πληροφορίες όπως κατατιμήσεις ρίζας, πληροφορίες τύπου αρχείου ή κρυφά bit αρχείου, αυτή η κλάση έχει σχεδιαστεί για να κατανοεί όσο το δυνατόν περισσότερες πληροφορίες συστήματος αρχείων για το λειτουργικό σύστημα.

### 5.2.2 JavaFX

Η JavaFX αποτελεί ένα σύνολο πακέτων γραφικών και πολυμέσων που επιτρέπει στους προγραμματιστές να σχεδιάζουν, να δημιουργούν, να δοκιμάζουν, να διορθώνουν και να αναπτύσσουν πλούσιες εφαρμογές πελατών που λειτουργούν με συνέπεια σε διάφορες πλατφόρμες (cross - platform). Η εμφάνιση και η αίσθηση που δείχνουν εφαρμογές υλοποιημένες με JavaFX μπορούν να τροποποιηθούν. Η CSS διαχωρίζει την εμφάνιση και το στυλ από την υλοποίηση, έτσι ώστε ο προγραμματιστής να μπορεί να επικεντρωθεί στην κωδικοποίηση. Για τους γνώστες του backend, της HTML, η JavaFX δίνει την δυνατότητα ανάπτυξης της διεπαφής με την FXML ή οποία

διατηρεί παρόμοια λογική με την HTML. Η JavaFX διαθέτει την δική της πλατφόρμα ανάπτυξης της διεπαφής που ουσιαστικά με αυτήν παραλείπεις τον κώδικα της FXML και σχεδιάζεις εύκολα με το χέρι. Το Scene Builder δημιουργεί το δικό του αρχείο, το οποίο είναι τροποποιήσιμο.

Η JavaFX επιλέχθηκε αρχικά γιατί υπάρχει εξοικείωση με την Java καθώς και με την HTML και την CSS. Θεωρείται ιδανική για την δημιουργία διεπαφής μέσω του Scene Builder καθώς είναι εύκολο και εύχρηστο. Μπορεί να φιλοξενήσει την δημιουργία τρισδιάστατων και δισδιάστατων σχημάτων που ήταν ένα από τα ζητούμενα για την λειτουργία της αναπαραγωγής.

### 5.2.2.1 Πακέτα JavaFX

Τα πακέτα της JavaFX που χρησιμοποιεί η java είναι κατά κύριο λόγο πακέτα παραμετροποιήσεις η δημιουργίας στοιχείων καθώς και διαχείρισης γεγονότων (event). Παρακάτω αναφέρονται ανά κατηγορία, καθώς εξηγούνται και οι λειτουργίες που προσφέρουν.

- **javafx.scene(...)** : Αυτό το πακέτο της JavaFX αφορά την δημιουργία στοιχείων μέσω της java, την διαμόρφωσή τους και την εισαγωγή τους στην διεπαφή. Τα στοιχεία αυτά μπορεί να είναι τα κλασικά όπως button, label, text κ.α.
- **javafx.fxml.FXML** : Το πακέτο της JavaFX για την FXML βοήθεια για την δήλωση των στοιχείων του FXML αρχείου στην Java.
- **javafx.geometry** : Το πακέτο αυτό παρέχει ένα σύνολο κλάσεων για τον ορισμό και την εκτέλεση πράξεων σε αντικείμενα που σχετίζονται με τη δισδιάστατη γεωμετρία.
- **javafx.event.Handler & javafx.event.ActionEvent** : Το πακέτο αυτό εξειδικεύεται στην διαχείριση γεγονότων που μπορούν να συμβούν με την αλληλεπίδραση κάποιου στοιχείου ή στοιχείων με ένα σύνολο ενεργειών. Ένα απλό παράδειγμα είναι ένα απλό κλικ του ποντικιού.
- **import javafx.stage.Stage** : Το πακέτο JavaFX Stage είναι το κοντέινερ που τοποθετείται στην κορυφή της πυραμίδας της JavaFX. Η κύρια σκηνή κατασκευάζεται από την πλατφόρμα. Όλα τα στοιχεία που εισέρχονται στην διεπαφή μέσω του αρχείου της FXML τοποθετούνται πάνω της.
- **javafx.scene.shape(...)** : Το πακέτο της JavaFX shape είναι υπεύθυνο για την δημιουργία σχημάτων, όπως Line, Quadcurve κ.α.. Το συγκεκριμένο σύνολο σχημάτων μπορεί να προστεθεί είτε στην scene είτε σε οποιαδήποτε υποστοίχιο της και η τοποθέτηση των σχημάτων αυτό δεν περιλαμβάνει ύψος και πλάτος, άλλα ένα σύνολο συντεταγμένων X και Y που αφορούν την αρχή, το τέλος του σχήματος καθώς κι άλλα όπως για παράδειγμα στο Quadcurve όπου απαιτεί και ένα τρίτο σημείο συντεταγμένων που είναι το σημείο της καμπύλης.

### 5.2.3 CSS

Η CSS (Cascading Style Sheets) είναι μια stylesheet γλώσσα υπολογιστή (γλώσσα σχεδίασης παρουσιαστικού) που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης όπως η HTML, η XML και στην δική μας περίπτωση FXML. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML, XML και FXML. Συνήθως είναι υπεύθυνη για την εμφάνιση μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την HTML. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα

η χρήση της CSS κρίνεται ως απαραίτητη. Η JavaFX την έχει προσαρμόσει με δικές της παραμετροποιημένες εντολές που δεν καλύπτουν ολόκληρο το φάσμα των εντολών της κοινής CSS.

### 5.2.4 FXML

Το JavaFX FXML είναι μια μορφή XML που δίνει τη δυνατότητα της σύνθεσης JavaFX διεπαφής με τρόπο παρόμοιο με τον τρόπο που χρησιμοποιούν γραφικά περιβάλλοντα ιστού σε HTML. Το FXML δίνει έτσι τη δυνατότητα να διαχωριστεί ο κώδικα διάταξης JavaFX από τον υπόλοιπο κώδικα της εφαρμογής. Αυτό καθαρίζει τόσο τον κώδικα διάταξης όσο και τον υπόλοιπο κώδικα εφαρμογής.

## 5.3 Γραμματική Σχημάτων

Δύο κύριες λειτουργίες ενοποίησης για την εξέλιξη σχεδίων ενδυμάτων είναι αυτές τις κωδικοποίησης και αποκωδικοποίησης. Η διεπαφή κωδικοποίησης χρησιμοποιεί ΓΣ κανόνες, αρχικά σχήματα (αναλόγως με το πρότυπο που εξελίξει), παράμετροι και περιορισμοί για την παροχή του σχεδίου για τον γονότυπο που θα χρησιμοποιηθεί από τον εξελικτικό αλγόριθμο. Κάθε κανόνας σχήματος έχει και τα σχετικά σχήματα, τις παραμέτρους και τους περιορισμούς του. Η λειτουργία αποκωδικοποίησης επιτρέπει στον εξελικτικό αλγόριθμο να αποκωδικοποιήσει τον γονότυπο σε φαινότυπο. Ο φαινότυπος είναι η πραγματική αναπαράσταση κανόνων σχήματος και τις παραμέτρους που απαιτούνται για τη δημιουργία των πραγματικών σχημάτων σχεδίασης. Η διαδικασία της κωδικοποίησης είναι απαραίτητη κατά το πρώιμο στάδιο σχεδιασμού των εξελικτικών αλγορίθμων. Ο γονότυπος κωδικοποιείται σε μια δισδιάστατη διάταξη δομής δεδομένων και στο πλαίσιο κανόνων και παραμέτρων γραμματικής σχήματος.

Δημιουργία σώματος	}	$\cdot I$	→		βασικό σχήμα
Δημιουργία χαμηλού μέρους σώματος		$\underline{\cdot I}$	→		κανόνας 1
Δημιουργία λαιμόκοψης		$\underline{\cdot I}$	→		κανόνας 2
Δημιουργία μανικιών	}	$\underline{\cdot U}$	→		κανόνας 3
Δημιουργία τελειώματος μανικιών		$\underline{\cdot U}$	→		κανόνας 4
Δημιουργία τσέπης/ων		$\underline{\cdot D}$	→		κανόνας 5

Σχήμα 5. 1: Κανόνες σχημάτων για την δημιουργία του προτύπου μπλούζας.

Οι διαδικασίες κωδικοποίησης και αποκωδικοποίησης στην τελική εφαρμογή γίνεται μέσω μιας σειράς παραμέτρων σταθερού μήκους. Αυτό αποτελεί τον γονότυπο όπως φαίνεται στο σχήμα

5.2. Ο φαινότυπος αποτελεί την γραφική αναπαράσταση του γονότυπου (σχήμα 5.3). Η διαδικασία αποκωδικοποίησης στην ουσία είναι η ανάθεση των τιμών των παραμέτρων σε μια μέθοδο σχεδίασης η οποία σχεδιάζει και τοποθετεί ανάλογα τα σχήματα.

Για παράδειγμα στην δημιουργία του πρότυπου της μπλούζας που φαίνεται στο σχήμα 5.1, μετά την τοποθέτηση του βασικού σχήματος που αποτελεί το σώμα της μπλούζας αμετάβλητου ύψους και πλάτους. Γίνεται η τοποθέτηση στο άνω μέρος τους σώματος που προκύπτει από τον περιορισμό αν είναι επαρκές το ύψος των μανικιών ( $> 0$ ). Σε κάθε περίπτωση ο κανόνας δίνει και το ανάλογο σχήμα. Πάνω στο άνω μέρος του σώματος τοποθετείται η λαιμόκοψη η οποία τοποθετείται σε σταθερό σημείο και ποικίλει σε σχήματα ή συνδυασμό αυτών. Η τοποθέτηση των μανικιών γίνεται επίσης σε σταθερό σημείο και έχει παραμέτρους ύψους και πλάτους. Από τις παραπάνω παραμέτρους προκύπτει ο περιορισμός να μην υπάρχει τελείωμα μανικιών αν το ύψος αυτών δεν υπερβαίνει ένα κατώτατο άκρο. Τέλος οι τσέπες, έχουν παραμέτρους ύψους και πλάτους, ένα σχήμα που προκύπτει από τις παραμέτρους, αλλά παραπάνω από ένα σημεία και τοποθετούνται μέσα στο αρχικά σχήμα.

#### 5.4 Γενετικό υλικό

Για την δημιουργία του γονότυπου δεν χρησιμοποιήθηκε μια ακολουθία από 0 και 1, τουλάχιστον όχι για όλα τα αλληλία του. Σε κάποια αλληλία δόθηκαν οι αριθμοί των παραμέτρων ύψους και πλάτους, σε κάποια άλλα δόθηκαν τιμές που αντιστοιχούν σε κανόνες (στην περίπτωση της λαιμόκοψης κάθε αριθμός αντιστοιχεί και σε έναν κανόνα), ενώ κάποια άλλα αποτελούν πύλες και εκεί οι τιμές μπορούν να είναι μόνο 0 και 1. Πύλες εννοούμε για παράδειγμα αν θα υπάρχουν τσέπες η όχι και με παρόμοιο τρόπο και το τελείωμα των μανικιών. Τέλος υπάρχουν και τα αλληλία του χρώματος τα οποία παίρνουν τιμές από έτοιμο πίνακα που αντιστοιχεί σε γνωστά χρώματα rgb.

Κατά την αρχικοποίηση του πληθυσμού, είτε αυτός πραγματοποιηθεί με τροφοδότηση από κατάλληλο panel φίλτρων από τον χρήστη είτε όχι. Οι τιμές των αλληλίων των σχημάτων προκύπτει με τυχαία τρόπο, άλλα καθώς μιλάμε για δημιουργία ενδυμάτων υπάρχει εύρος αριθμών για την σωστή αναπαράσταση ύψους πλάτους καθώς και για την ορθή επιλογή των κατάλληλων κανόνων για εφαρμογή. Στο σχήμα 5.2 μπορούμε να δούμε το πρότυπο της μπλούζας καθώς και σε τι αντιστοιχεί το κάθε γονίδιο, ενώ στο σχήμα 5.3 την αναπαράσταση ενός φαινοτύπου, που όπως προαναφέρθηκε αποτελούν ταυτόσημα στην εφαρμογή μας.

Γονότυπος :  $G = (S1|S2, LP, SLLP1|SLLP2, C, P1|P2|P3|P4, CL1|CL2|CL3|CL4)$

SL : Αφορά τα μανικία. Το 1 είναι το ύψος και το 2 το πλάτος

LP: Αφορά το κάτω μέρος του σώματος και είναι το ύψος

SLLP: Αφορά το τελείωμα των μανικιών. Το 1 είναι το ύψος το 2 το πλάτος

C: Αφορά την λαιμόκοψη.

P: Αφορά τις τσέπες. Το 1 είναι η ύπαρξη τους, το 2 το στυλ τους, το 3 το ύψος και το 4 το πλάτος

CL: Αφορά τα χρώματα που στο πρότυπο της μπλούζας είναι 4.



Σχήμα 5. 3: Γραφική αναπαράσταση φαινότυπου προτύπου μπλούζας.

### 5.5 Διαδραστικός Εξελικτικός Αλγόριθμος Σχεδίασης Ενδυμάτων

Η εξελικτική διαδικασία που εφαρμόζεται για την επίλυση του προβλήματος σχεδίασης ενδυμάτων, όπως προαναφέρθηκε στηρίζεται στον ΔΕΑ των Kim και Cho [7], ο οποίος προσαρμόστηκε έτσι ώστε να φιλοξενήσει και την παραλλαγμένη ΓΣ που παρουσιάστηκε παραπάνω. Ο βασικός λόγος που υιοθετήθηκε μια τέτοια παραλλαγή του ΓΣ και δεν εφαρμόστηκε η ιδέα αναπαραστάσεις του ΔΓΑ των Kim και Cho, ήταν η ελαττώσει των απαιτούμενων πόρων καθώς πρέπει να υπάρχει μια βάση δεδομένων διασδιάστατων φωτογραφιών, από την οποία η αποκωδικοποίηση του γονότυπου δείχνει την αντίστοιχη φωτογραφία για κάθε γονίδιο. Έτσι προτιμήθηκε μια μέθοδος δημιουργίας γεωμετρικών σχημάτων οι οποία υλοποιήθηκε σε ελάχιστες γραμμές κώδικα.

Η μέθοδος αξιολόγησης διατηρήθηκε όπως είναι. Ο χρήστης παρατηρεί τα σχέδια που του εμφανίζει η διεπαφή και τα βαθμολογεί και αυτή η βαθμολογία αποτελεί τον δείκτη ποιότητας. Αυτό φυσικά προκύπτει από την ανάγκη της διάδρασης στην σχεδίαση ενδυμάτων. Η επιλογή γονέων παραμένει η ίδια με αυτή του ΓΑ η ΔΓΑ, δηλαδή με την εφαρμογή της δεξαμενής ζευγαρώματος. Συνεπώς παρόλο που ο χρήστης μπορεί να επιθυμεί οι γονείς να είναι τα δυο μεγαλύτερα βαθμολογούμενα σχέδια, δεν είναι σαφές πως αυτά θα τους αποτελέσουν. Η διαδικασία διασταύρωσης διαφοροποιείται από αυτήν του ΔΓΑ. Επειδή με την εφαρμογή της διασταυρώσεως μονής τομής, σε συνδυασμό με την έλλειψη ποικιλίας χαρακτηριστικών, τα σχέδια που παράγονται είναι αυτόματα ταυτόσημα μεταξύ τους, επιλέχθηκε η ομοιόμορφη διασταύρωση που επιλέγει για κάθε αλληλίο τυχαία ανάμεσα από τους δύο γονείς. Η μετάλλαξη δεν διαφοροποιήθηκε και επηρεάζει μόνο ένα αλληλίο με μικρή πιθανότητα να εφαρμοστεί σε κάποιο άτομο. Η αναπαραγόμενη γενιά περνάει από τον τελεστή επιβίωσης όπου επιλέγονται τα  $n$  επικρατέστερα άτομα. Τέλος ο πληθυσμός της νέας γενιάς αναπαρίσταται ολοκληρώνοντας τον βρόγχο, ο οποίος ξεκινάει και πάλι μετά την αξιολόγηση με εντολή του χρήστη.

### 5.5.1 Παραμετροποίηση Διαδραστικού Εξελικτικού Αλγορίθμου

Παρακάτω παρουσιάζεται το σύνολο των ρυθμίσεων της παραμετροποίησης του ΔΕΑ, οι οποίες είναι ομαδοποιημένες βάσει διακριτών βημάτων του αλγόριθμου με τυχαία σειρά.

#### Κατηγορίες Πληθυσμών

- Αρχικός Πληθυσμός :** Η πρώτη ενέργεια που εκτελεί ο ΔΕΑ με την εκτέλεση του βρόχου είναι η αρχικοποίηση του πληθυσμού, ο οποίος είτε δημιουργείται αναθέτοντας τυχαίες εντελώς τιμές στα αλληλία των γονότυπων των ατόμων, είτε με την βοήθεια του πίνακα φίλτρων που συναντά στην καρτέλα επιλογών “OPTIONS” και επιλέγοντας τα φίλτρα που θέλει μπορεί να καθοδηγήσει τις τιμές. Υπάρχει μια πιθανότητα 50% να μην εφαρμόσει τα φίλτρα στην αρχικοποίηση για να μην επηρεάσει τελείως μονόπλευρα το σχέδιο και να μπορεί να επιλέξει δύο εντελώς διαφορετικά σχέδια για γονείς. Το μέγεθος του πληθυσμού μπορεί να οριστεί από τον πίνακα “Generation Options” στην επιλογή “number of designs per generation”. Αυτό επηρεάζει τον αριθμό των σχεδίων που επιλέγονται για προβολή στον χρήστη καθώς και το μέγεθος της αρχικοποιημένης γενεάς. Μπορεί να πάρει αριθμούς του εύρους δύο μέχρι δέκα. Η εφαρμογή έχει αρχικό αριθμό το δέκα, για να υπάρχει ικανοποιητική ποικιλία σχεδίων
- Αναπαραγόμενος Πληθυσμός :** Αφού ολοκληρωθεί η αξιολόγηση της αρχικοποιημένης γενιάς και εκτελεσθεί η διαδικασία επιλογής γονέων, παράγεται η γενιά των απογόνων μέσω των τελεστών εξέλιξης. Το μέγεθος αυτής της γενιάς δεν ορίζεται από τον ίδιο τον χρήστη. Έχει σταθερό αριθμό διακόσια και προκύπτουν από τις διαδικασίες διασταύρωσης και μετάλλαξης. Από αυτούς επιβιώνουν τα άτομα που επιλέγονται είτε τυχαία είτε με την τροφοδότηση του χρήστη στον πίνακα “Trends”, όπου από εκεί ο χρήστης δίνει πρακτικά τις προτιμήσεις του ως προς κάποια χαρακτηριστικά του επιλεγμένου ενδύματος είτε σε όλα.
- Επιλεγμένος Πληθυσμός :** Ο επιλεγμένος πληθυσμός προκύπτει από την επιβίωση των ατόμων - σχεδίων του αναπαραγόμενου πληθυσμού. Είναι αυτός που δίνεται από τον ΔΕΑ για αναπαράσταση και προβολή στον χρήστη. Η διαδικασία που ακολουθεί στην συνέχεια είναι γνωστή. Το μέγεθος του επιλεγμένου πληθυσμού είναι ταυτόσημο με αυτό του αρχικού πληθυσμού.
- Τελικός Πληθυσμός :** Με το πέρας των επαναλήψεων του βρόχου, είτε με την απόφαση του χρήστη, μετά την εύρεση της βέλτιστης λύσης, είτε με την ολοκλήρωση ενός ανώτατου ορίου  $n$  επαναλήψεων (γενεών), προκύπτει ο τελικός πληθυσμός. Αυτός ίσως, αποτελείται από τα πραγματικά βέλτιστα σχέδια που επιθυμούσε ο χρήστης, ή έστω ικανοποιητικά. Πάντα υπάρχει και η περίπτωση να μην έμεινε ικανοποιημένος καθώς μπορεί να μην βρήκε αυτό που ήθελε μετά το πέρας των  $n$  επαναλήψεων, ή να κουράστηκε. Το μέγεθος του τελικό πληθυσμού είναι ταυτόσημο με αυτό του επιλεγμένου πληθυσμού.

### Σενάρια Τερματισμού

Τα σενάρια τερματισμού είναι δύο και επειδή πρόκειται για ένα σύστημα αλληλεξάρτησης με τον χρήστη, αυτός είναι υπεύθυνος να τα ορίσει.

- **Μέγιστο Πλήθος Γενεών :** Σε άλλες περιπτώσεις ΔΕΑ το να υπάρχει ένας αριθμός μέγιστων γενεών μπορεί να μην ήταν αναγκαίος, άλλα όπως και να έχει ακόμα και έτσι μπορεί να βρεθεί σε τέλμα. Η εφαρμογή δίνει την δυνατότητα στον χρήστη να ορίσει αυτών τον αριθμό. Μέσω της επιλογής “number of maximum generations” προτείνει έναν χαμηλό αριθμό που είναι δέκα γενιές και είναι ο κατώτερος, Μπορεί να φτάσει μέχρι και τις τριάντα.
- **Εύρεση Βέλτιστης Λύσης / Τερματισμός Βρόχου :** Όταν ο χρήστης βρει μια ικανοποιητική ή μια βέλτιστη λύση ανάμεσα σε ολόκληρη την γενιά ή κάποιο σχέδιο, έχει την δυνατότητα να τερματίσει την εφαρμογή πατώντας το κουμπί “final generation”, όπου έτσι παίρνει και τον τελικό πληθυσμό. Έτσι τερματίζεται ο βρόχος, αρχικοποιούνται σχεδόν όλοι οι παράμετροι του προγράμματος και δημιουργείται μια φωτογραφία στην αρχική οθόνη του χρήστη.
- **Στάσιμη Ποιότητα Σχεδίων :** Όταν και αν ο χρήστης παρατηρήσει την επανάληψη η την λοξοδρόμηση των επιλεγμένων πληθυσμών που του προβάλλονται, ή έχει μετανιώσει ο ίδιος για τις επιλογές που έχει κάνει, του δίνεται η επιλογή της επανεκκίνησης μέσω του κουμπιού “reset”. Μεταφέρεται απευθείας στην καρτέλα επιλογών “OPTIONS” και από εκεί πρέπει να ξανά επιλέξει ότι επιθυμεί από τις επιλογές από την αρχή.

### Αναπαραγωγή

Για την αναπαραγωγή πληθυσμού κάθε γενεάς εφαρμόζονται δύο διακριτές διαδικασίες γνωστές ως επιλογή και ανασυνδυασμός. Κατά την διαδικασία επιλογής, ο ΔΓΑ πρέπει να επιλέξει δύο γονείς από το σύνολο του πληθυσμού λύσεων της τωρινής γενεάς και συμβουλευμένος από τον μηχανισμό αξιολόγησης που παρέχει ο χρήστης, να τους τοποθετήσει στην διαδικασία ανασυνδυασμού. Εκεί οι δύο γονείς ζευγαρώνουν και αναπαράγουν τις νέες λύσεις. Είναι ευρέως διαδεδομένο, ότι οι δύο παραπάνω διαδικασίες επιτυγχάνονται με τελεστές. Για την διαδικασία επιλογής υπάρχει και ένας αντίστοιχος τελεστής επιλογής, ενώ για την διαδικασία του ανασυνδυασμού, υπάρχουν οι τελεστές διασταύρωσης και μετάλλαξης.

- **Τελεστής Επιλογής :** Για την υλοποίηση του τελεστή επιλογής, έχουν προταθεί αρκετές παραλλαγές, για τροποποίηση και βελτιστοποίηση και πιθανόν θα προταθούν άλλες τόσες, καθώς πρόκειται για διαδικασία που βρίσκεται περισσότερο στην κρίση του προγραμματιστή να βρει ποια είναι η κατάλληλη. Υπάρχουν τέσσερις πολύ γνωστοί τελεστές επιλογής, η τυχαία επιλογή (random), η επιλογή κατάταξης (random), η επιλογή αθλητικού τουρνουά (tournament) και η επιλογή περιστρεφόμενης ρουλέτας (roulette wheel) που θα μπορούσαν να εφαρμοστούν. Παρακάτω παρουσιάζονται με τελευταίο τον τελεστή που έχει χρησιμοποιηθεί.

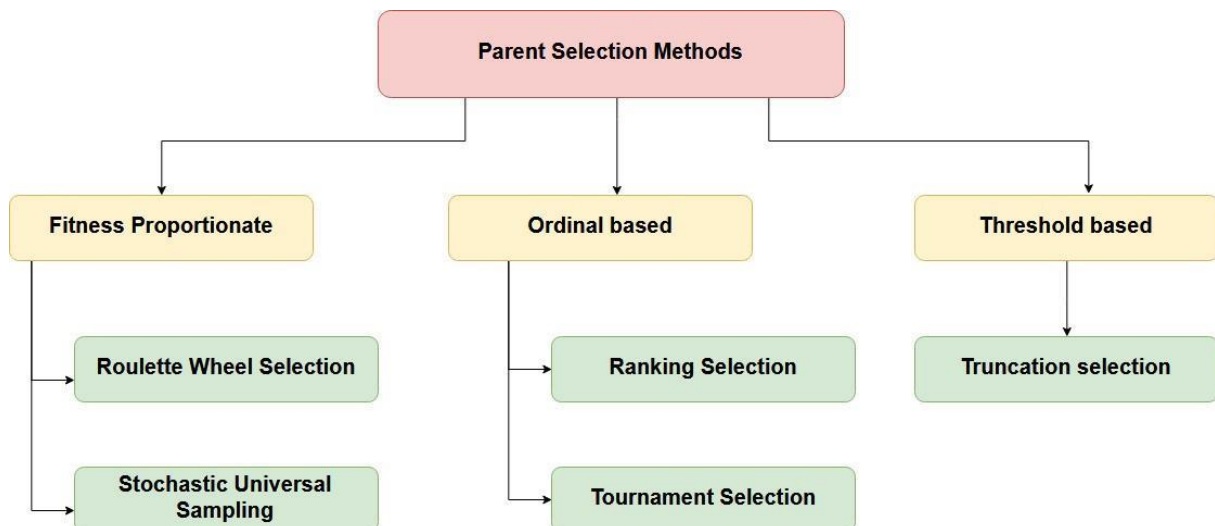
Ο τελεστής τυχαίας επιλογής αποτελεί τον πιο παράδοξο τελεστή από τους παραπάνω, καθώς δεν υπολογίζει καθόλου τον δείκτη ποιότητας κάθε ατόμου. Ουσιαστικά ο τελεστής αυτός εφαρμόζεται με την χρήση μιας μεθόδου, η οποία έχοντας υπόψη τον μέγιστο

αριθμό ατόμων του πληθυσμού, βγάζει τυχαία ένα αριθμό που αντιστοιχεί σε κάποια λύση και έτσι επιλέγει για τον ρόλο του γονέα. Για την επιλογή και δευτέρου γονέα απλά αφαιρεί τον πρώτο που επιλέχθηκε και επαναλαμβάνει την διαδικασία.

Ο δίκαιος τελεστής κατάταξης, ταξινομεί τα άτομα σύμφωνα με τον δείκτη ποιότητας τους. Ορίζει αριθμούς σε κάθε άτομο με φθίνουσα ταξινόμηση του πληθυσμού (π.χ. στο άτομο που έχει τον μεγαλύτερο δείκτη ποιότητας δίνει τον αριθμό 1, ενώ σε αυτόν που έχει το χαμηλότερο δίνει τον αριθμό  $n$  όπου  $n$  είναι ο μέγιστος αριθμός της γενιάς). Δημιουργώντας λοιπόν μια λίστα με αριθμημένα άτομα, κατά την επιλογή ατόμου για γονέα, δίνει αυτό που βρίσκεται στην λίστα πρώτο και προχωράει στο επόμενο. Σε περίπτωση που η λίστα τερματίσει, ξεκινάει από την αρχή.

Ο τελεστής αθλητικού τουρνουά, απαιτεί από τον χρήστη δύο παραμέτρους για την υλοποίηση της μεθόδου του. Πρέπει να εισάγει έναν αριθμό που αντιστοιχεί στο σύνολο των διαγωνιζόμενων (δεν μπορεί να ξεπερνά τον αριθμό του αναπαραγόμενου πληθυσμού), που διαγωνίζονται για την θέση του γονέα, δηλαδή των ατόμων που συμμετέχουν στην διαδικασία της επιλογής. Η δεύτερη παράμετρος αφορά ένα ποσοστό το οποίο αφορά κατά πόσο της εκατό το άτομο με μεγαλύτερο δείκτη ποιότητας θα αποτελέσει και τον νικητή, συνεπώς θα επιλεγεί για γονέας.

Τέλος η επιλογή περιστρεφόμενης ρουλέτας η οποία αποτελεί και τον εφαρμοσμένο τελεστή της εφαρμογής. Στην επιλογή με περιστρεφόμενη ρουλέτα, δημιουργείται ένας “χώρος” στον οποίο δημιουργούνται τόσο αντίγραφα κάθε ατόμου, όσα είναι ο δείκτης ποιότητας που έχει το καθένα. Στην συνέχεια όπως και σε μια πραγματική ρουλέτα υπάρχει ένα μηχανισμός οποίος σκανάρει και τυχαία σταματάει σε κάποια θέση. Η θέση αυτή αντιστοιχεί σε κάποιο άτομο, το οποίο επιλέγεται για γονέας. Σε αυτήν την προσέγγιση επιλογής, το ίδιο άτομο μπορεί να αποτελέσει και τον δεύτερο γονέα.



Σχήμα 5. 4: Κατηγοριοποίηση τελεστών επιλογής γονέα



Ο τελεστής περιστρεφόμενης ρουλέτας επιλέχθηκε διότι αποτελεί έναν από τους τελεστές, που είναι από αυτούς που βασίζονται περισσότερο στον δείκτη ποιότητας των ατόμων και όχι σε κάποιον άλλο παράγοντα ή περισσότερο στην τύχη. Αυτό εξυπηρετεί, διότι το ζητούμενο ζευγάρι γονέων που θα επιλεγεί, κατά πάσα πιθανότητα θα είναι αυτό που θα έκανε μεγαλύτερη εντύπωση στον χρήστη. Στο σχήμα 5.4 φαίνονται όλες οι κατηγορίες τελεστών επιλογής και οι κατηγορίες τους.

- **Τελεστής Διασταύρωσης :** Κατά την διαδικασία της διασταυρώσεως η επιλεγμένοι γονείς - άτομα και πιο συγκεκριμένα ο γονότυπος τους ορίζει κοινό ή κοινά σημεία τομείς και ανταλλάσσει τα δύο σημεία μεταξύ τους. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να γεμίσει καινούργιος πληθυσμός απογόνων - ατόμων της επόμενης γενιάς. Υπάρχουν τριών ειδών τελεστές διασταύρωσης. Η διασταύρωση μονής τομής (one-point crossover), η διασταύρωση διπλής τομής (two-point crossover) και η ομοιόμορφη διασταύρωση (uniform crossover) όπως φαίνεται στο σχήμα 2.2 στο κεφάλαιο 2 “Εξελκτικοί Αλγόριθμοι”. Θεωρείται απαραίτητο οι γονότυποι των δύο γονέων που διασταυρώνονται να έχουν το ίδιο μήκος και τον ίδιο αριθμό αλληλίων. Αυτό ισχύει για κάθε έναν από τους τελεστές που εκτελούν αυτή την διαδικασία.

Στον τελεστή μονής τομής, σε έναν από τους δύο γονείς επιλέγεται ένα σημείο τομής στην τύχη, αυτό το σημείο τεμαχίζει στα δύο και τους δύο γονείς και με την επιλογή του αριστερού και του δεξιού κομματιού από κάθε γονέα ξεχωριστά δημιουργείται ένας καινούργιος απόγονος. Η διαδικασία μπορεί να επαναλαμβάνεται ξεχωριστά για την δημιουργία κάθε απογόνου - ατόμου.

Στον τελεστή διπλής τομής η διαδικασία είναι παρόμοια με αυτή του τελεστή μονής τομής. Στην περίπτωση αυτή, ο γονότυπος του γονέα που επιλέχθηκε στην τύχη διαιρείται σε τρία κομμάτια. Έτσι τα με τυχαίο τρόπο, τοποθετούνται τα δυο ακριανά του ενός γονέα και το μεσαίο του άλλου για την δημιουργία απογόνου. Η διαδικασία μπορεί να επαναλαμβάνεται ξεχωριστά για την δημιουργία κάθε απογόνου - ατόμου.

Ο τελεστής ομοιόμορφης διασταύρωσης κρίθηκε ο κατάλληλος για την εφαρμογή. Η λογική του είναι διαφορετική από τους δύο παραπάνω. Δεν δημιουργεί τομές στους γονότυπους των γονέων - ατόμων. Πρακτικά, επιλέγει στην τύχη γονίδια και τα τοποθετεί σε αυτά του απογόνου του, καλύπτοντα μία μία τις “τρύπες”. Η διαδικασία αυτού του τελεστή διασταύρωσης πραγματοποιήθηκε με την δημιουργία ενός δισδιάστατου πίνακα που δίνει η τιμές 0 και 1, όπου 0 ο πρώτος γονέας και 1 ο δεύτερος και n ο σταθερός αριθμός των αλληλίων στο κατάλληλο πρότυπο. Στην συνέχεια αφού φτιάξει τον πίνακα 200 x n όπου διακόσια ο αριθμός του αναπαραγόμενου πληθυσμού, τον δημιουργεί αναθέτοντας τις κατάλληλες τιμές σε κάθε απόγονο. Στο σχήμα 5.5 φαίνεται η διασταύρωση δύο ατόμων του πρότυπου της μπλούζας και ο απόγονος, παράγεται με την ομοιόμορφη διασταύρωση. Το αποτέλεσμα της διασταύρωσης των δύο γονέων είναι της μορφής (διαστάσεις μανικιών → a, μήκος μπλούζας → b, τελειώμα μανικιών → a, λαιμόκοψη → a, τσέπες → b, παλέτα χρωμάτων → a).

Γονέας a : (SLa1|SLa2, LPa, SLLPa1|SLLPa2, Ca, Pa1|Pa2|Pa3|Pa4, CLa1|CLa2|CLa3|CLa4)

Γονέας b : (SLb1|SLb2, LPb, SLLPb1|SLLPb2, Cb, Pb1|Pb2|Pb3|Pb4, CLb1|CLb2|CLb3|CLb4)

Μετά την εφαρμογή ομοιόμορφης διασταυρώσης

Απόγονος : (SLa1|SLa2, LPb, SLLPb1|SLLPb2, Ca, Pb1|Pb2|Pb3|Pb4, CLa1|CLa2|CLa3|CLa4)

Σχήμα 5. 5: Διασταύρωση “T-Shirt” με τελεστή ομοιόμορφης διασταύρωσης και αναπαραγωγή απογόνου.

- **Τελεστής μετάλλαξης :** Ο τελεστής μετάλλαξης είναι αυτός που ακολουθεί τον τελεστή διασταύρωσης και εφαρμόζεται πάνω στον ήδη αναπαραγόμενο πληθυσμό που προκύπτει από το ζευγάρι των γονέων - ατόμων. Οι πιθανότητες εφαρμογής του τελεστή μετάλλαξης αλλάζουν σε κάθε ΕΑ. Υπάρχουν ΕΑ που δίνουν μεγάλη βάση σε αυτόν για την εξέλιξη και άλλη που δεν δίνουν καθόλου.

Στην εφαρμογή ο τελεστής μετάλλαξης εφαρμόζεται στην αναπαραγόμενη γενιά με πιθανότητα 20%. Οι αναπαραγόμενοι απόγονοι εισέρχονται ένας ένας στον τελεστή μετάλλαξης και εκεί κρίνεται αν θα μεταλλαχτούν, φυσικά με τον παράγοντα της τύχης. Η διαδικασία της μετάλλαξης πραγματοποιείται και πάλι με τον παράγοντα της τύχης, όπου επιλέγεται τυχαία ένα γονίδιο και από εκεί με παρόμοιο τρόπο και ένα αλληλίο το οποίο αλλάζει τιμή όπως και όταν παίρνει τιμή στην διαδικασία αρχικοποίησης πληθυσμού. Η επίδραση του τελεστή μετάλλαξης πάνω σε γονότυπο απόγονου του αναπαραγόμενου πληθυσμού στο πρότυπο της μπλούζας φαίνεται παρακάτω στο σχήμα 5.6. Στο παράδειγμα βλέπουμε το άτομο που επιλέχθηκε για μετάλλαξη, στο οποίο ο τελεστής διαλέγει ένα γονίδιο στην τύχη, όπου στην συγκεκριμένη περίπτωση είναι το γονίδιο των τσεπών και στην συνέχεια με παρόμοιο τρόπο επιλέγει το τρίτο αλληλίο κατά σειρά, το οποίο αποτελεί το ύψος κάθε τσέπης που είναι κοινό, είτε υπάρχει μία τσέπη είτε δυο.

## Επιβίωση

Έπειτα της ολοκλήρωσης των τελεστών και μετάλλαξης ο νέος αναπαραγόμενος πληθυσμός ατόμων - σχεδίων πρέπει να προσαρμοστεί στον πληθυσμό της καινούργιας γενιάς. Επειδή ο αριθμός αυτού του πληθυσμού υπερβαίνει τον ζητούμενο αριθμό που είτε έχει ορίσει ο χρήστης είτε έχει οριστεί με τον προτεινόμενο αριθμό επιλεγμένου πληθυσμού, επιβάλλεται να εφαρμοστεί ένας τελεστής επιλογής, ο οποίος καθορίζει ποια από τα αναπαραγόμενα άτομα επιβιώνουν και καταφέρνουν να μπου σε αυτόν.

- **Επιλογή απογόνων με βάση παγκόσμιου βέλτιστου:** Ο τελεστής επιβίωσης που επιλέχθηκε για την εφαρμογή, είναι σαφώς σχετικός με την σχεδίαση ενδυμάτων. Όπως γνωρίζουμε, κάθε ένα τακτό χρονικό διάστημα, η βιομηχανία της μόδας ορίζει

τάσεις. Αυτές θα μπορούσαν για παράδειγμα να είναι ως προς τον τύπο μιας μπλούζας ή ενός παντελονιού, ως προς την φόρμα και το πιο γνωστό όλων αποτελεί το χρώμα.

Επιλογή απόγονου για μετάλλαξη :

(SL1|SI2, LP, SLLP1|SLLP2, C, P1|P2|P3|P4, CL1|CL2|CL3|CL4)

Επιλογή τυχαίου γονιδίου για μετάλλαξη :

(SL1|SI2, LP, SLLP1|SLLP2, C, P1|P2|P3|P4, CL1|CL2|CL3|CL4)

Επιλογή τυχαίου αλληλίου γονιδίου για μετάλλαξη :

(SL1|SI2, LP, SLLP1|SLLP2, C, P1|P2|P3|P4, CL1|CL2|CL3|CL4)

*Σχήμα 5. 6: Τελεστής μετάλλαξης που εφαρμόστηκε σε γονότυπου απόγονου στο πρότυπο μπλούζας.*

Έτσι για την προσαρμογή ενός τελεστής επιβίωσης που σχετίζεται με τις τάσεις μόδας, δημιουργήθηκε μια μέθοδος που συγκρίνει τα γονίδια κάθε απόγονου με αυτά τις κάθε τάσης του πρότυπου που εξελίσσεται εκείνη την στιγμή. Αυτό όμως, θα είχε ως αποτέλεσμα να περιορίσει αρκετά τα σχέδια - άτομα στην εξέλιξή τους και σε πολλές περιπτώσεις θα κατέληγε να παρουσιάζει παρόμοια σχέδια. Κοινός δεν θα υπάρχει ποικιλία στα τελικά σχέδια. Την λύση στο πρόβλημα αυτό έδωσε ένα panel “Trends”, στο οποίο ο χρήστης επιλέγει όλα, κάποια ή και κανένα από τις επιλογές για το πρότυπο που θέλει να εξέλξει. Φυσικά, όταν δεν επιλέγεται κανένα, ο τελεστής επιβίωσης εφαρμόζεται, ως επιβίωση με τυχαίο τρόπο.

## Αναπαράσταση

Ο τελευταίος σε σειρά τελεστής που εφαρμόζεται πριν την ολοκλήρωση του βρόχου του ΕΑ είναι αυτός της αναπαράστασης των ατόμων του πληθυσμού της νέας γενιάς. Για να υλοποιηθεί ο τελεστής αναπαράστασης, πρέπει να απαντηθεί το ερώτημα ποια μέθοδος που παρέχουν οι ΕΑ αποδίδει καλύτερα. Για την απάντηση αυτού του ερωτήματος πρέπει να ληφθεί υπόψη η φύση του προβλήματος στο οποίο εφαρμόζεται η εξέλιξη. Υπάρχουν δύο γενικές προσεγγίσεις για την επίτευξη της αναπαράστασης. Παρακάτω παρουσιάζονται, καθώς δικαιολογείται και ποια από τις δύο επιλέχθηκε για την υλοποίηση του τελεστή αναπαράστασης της εφαρμογής στο πρόβλημα σχεδίασης ενδυμάτων.

- **Άμεση αναπαράσταση :** Η πρώτη προσέγγιση, η άμεση αναπαράσταση, κωδικοποιεί λύσεις του προβλήματος στο «φυσικό» χώρο του και σχεδιάζει τους τελεστές αναζήτησης για να λειτουργήσουν σε αυτήν την αναπαράσταση.
- **Έμμεση αναπαράσταση :** Κατά την προσέγγιση της έμμεσης αναπαράστασης, μια λύση - άτομο κωδικοποιείται σε μια τυπική δομή δεδομένων, όπως συμβολοσειρές, ή διανύσματα, και σε αυτούς τους γονότυπους εφαρμόζονται οι γνωστοί τελεστές αναζήτησης. Αυτό συμβαίνει για παράδειγμα σε τυπικούς ΓΑ, ΕΣ (εξελικτικές

Στρατηγικές) και ορισμένες προσεγγίσεις ΓΠ όπως η γραμματική εξέλιξη ή ο καρτεσιανός γενετικός προγραμματισμός. Για να αξιολογηθεί η λύση, ο γονότυπος πρέπει να χαρτογραφηθεί στο χώρο του φαινοτύπου. Η σωστή επιλογή αυτής της χαρτογράφησης γονοτύπου - φαινοτύπου είναι σημαντική για την απόδοση της διαδικασίας αναζήτησης ΕΑ.

Η έμμεση αναπαράσταση επιλέχθηκε για την υλοποίηση του τελεστή αναπαράστασης, διότι σε προβλήματα σχεδίασης ενδυμάτων τα άτομα - σχέδια πρέπει να έρχονται σε άμεση επαφή με τον χρήστη για να αξιολογηθούν.

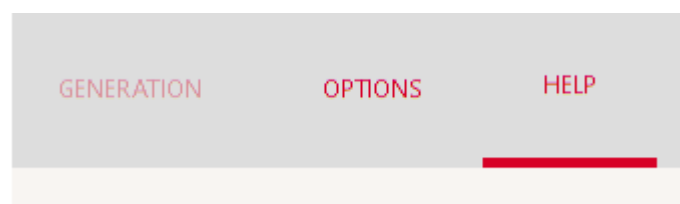
Για τον τελεστή αναπαράστασης κάθε προτύπου φτιάχτηκαν οι κατάλληλοι μέθοδοι. Κάθε μέθοδος στην αρχή βλέπει αν το άτομο που θα αναπαρασταθεί ανήκει στον πληθυσμό αρχικοποίησης ή όχι. Αν ναι, τότε υπάρχουν πενήντα πενήντα πιθανότητες να τροφοδοτηθεί από τα φίλτρα ή όχι και έτσι του δίνει τυχαίες η καθοδηγούμενες τιμές. Όπως και να έχει παίρνει τα αλλήλια των γονιδίων του γονότυπου και τις αποθηκεύει σε παραμέτρους. Στην συνέχεια για την σχεδίαση της αναπαράστασης, περνάει ένα ένα σε κάποια γονίδια. Κάποια γονίδια λειτουργούν με πύλες και κάποια σχεδιάζονται απευθείας με ανάθεση τιμών. Για παράδειγμα στην δημιουργία του πρότυπου της μπλούζας τα μανίκια σχεδιάζονται με την ανάθεση τιμών ύψους και πλάτους ενώ η τσέπες πρέπει πρώτα να περάσουν την πύλη ύπαρξης. Αν υπάρχουν θα συνεχίσουν στην πύλη στυλ που ορίζει αν θα υπάρχουν δύο (ένα σε κάθε πλευρά), ή αν θα είναι ένα και σε ποια πλευρά θα σχεδιαστεί. Τέλος σχεδιάζεται στο κατάλληλο σημείο με τις τιμές ύψους και πλάτους.

## 5.6 Εξοικείωση Διεπαφής

Για την ορθή χρήση και κατανόηση του τρόπου λειτουργίας της εφαρμογής γίνεται μια “ξενάγηση” της διεπαφής που στοχεύει στην εξοικείωση του χρήστη με αυτήν.

### 5.6.1 CATEGORIES MENU

Το categories menu βρίσκεται στα αριστερά της διεπαφής, κάτω από το logo και χρωματίζεται με ένα χαρακτηριστικό κόκκινο χρώμα. Προσφέρει τρεις κατηγορίες για επιλογή, “MAN”, “WOMAN”, “ALL”, που με την επιλογή τους εμφανίζουν τις κατάλληλες επιλογές προτύπου ανάλογο με το φύλλο (Τα πρότυπα που επιλέχθηκαν για την εξέλιξη είναι ανεξαρτήτως φύλλου). Συγκεκριμένα εμφανίζουν δύο κατηγορίες επιλογών “T-SHIRTS” και “TROUSERS”. Με το πάτημα ενός από των δύο επιλογών ενεργοποιείται και το κατάλληλο Filter. Στο σχήμα 5.7 φαίνεται το μενού κατηγοριών

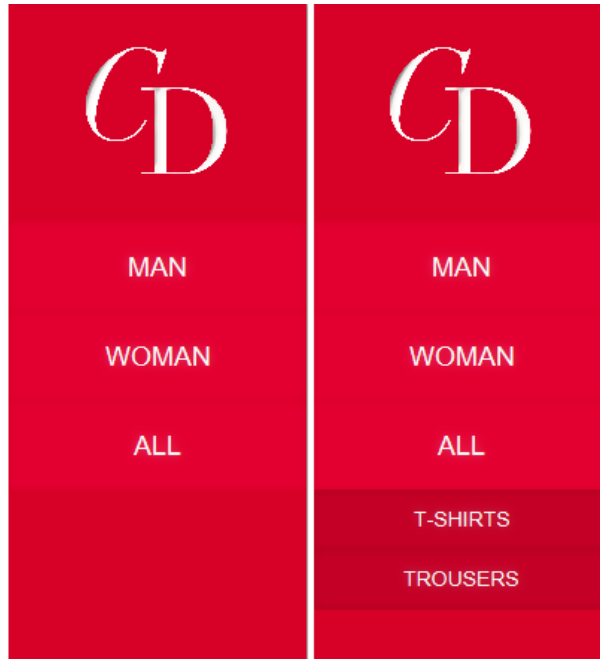


Σχήμα 5. 7: Tab menu και οι επιλογές “GENERATION”, “OPTIONS” και “HELP”.

πριν και μετά την επιλογή κατηγορίας φύλλου.

### 5.6.2 TAB MENU

Το tab menu βρίσκεται στο δεξί μέρος της διεπαφής όπως φαίνεται στο σχήμα 5.8 και περιέχει τρεις καρτέλες για επιλογή :



Σχήμα 5. 8: Μενού κατηγοριών πρότυπου ενδύματος.

#### 5.6.2.1 HELP TAB

Η καρτέλα βοήθειας (help tab) αποτελεί τον μπούσουλα του χρήστη καθώς εκεί εξηγεί πως λειτουργεί η εφαρμογή βήμα προς βήμα. Σε αντίθεση με τις άλλες δύο καρτέλες, είναι πάντα ενεργοποιημένη για τον χρήστη για να καλύψει τις απορίες τους.

#### 5.6.2.2 OPTIONS TAB

Η καρτέλα επιλογών (options tab) είναι η καρτέλα όπου ο χρήστης μπορεί να συμπληρώσει κάποιες επιλογές όπως τις επιλογές γενιάς (generation options), τα φίλτρα (filters), τις τάσεις (trends), να δει τις φωτογραφίες των προηγούμενων βέλτιστων γενεών και το πιο σημαντικό να ξεκινήσει την αρχικοποίηση του πληθυσμού. Όλα αυτά παρουσιάζονται αναλυτικά παρακάτω. Η καρτέλα επιλογών φαίνεται στο σχήμα 5.9.

#### 5.6.2.3 GENERATION TAB

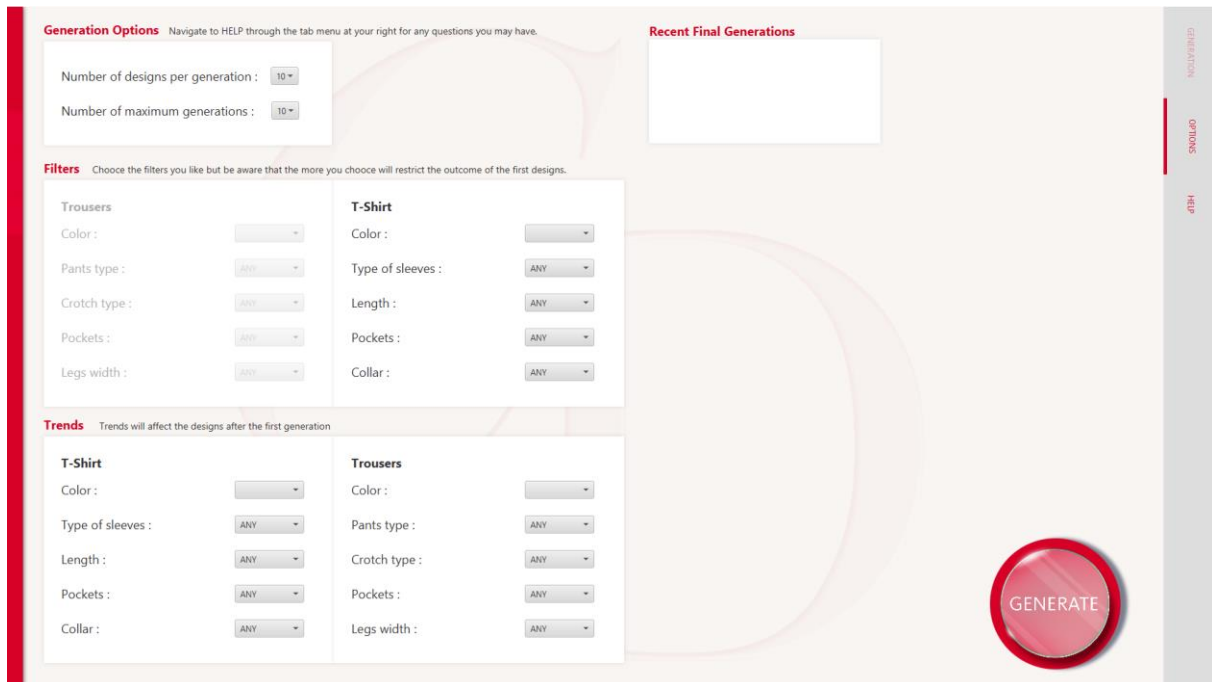
Η καρτέλα γενιάς (generation tab) είναι το περιβάλλον στο οποίο ο χρήστης έρχεται σε επαφή και αλληλεπιδρά με το σχέδιο άτομο. Η καρτέλα γενιάς ενεργοποιείται από το GENERATE κουμπί στην καρτέλα OPTIONS. Η καρτέλα γενιάς φαίνεται στο σχήμα 5.10. Αποτελείται από το πλαίσιο σχεδίων που βρίσκεται πάντα στο άνω μέρος της καρτέλας και το πλαίσιο ελέγχου. Το πλαίσιο σχεδίων περιέχει τα κελιά σχεδίων και πάνω στα κελιά πέρα από το σχέδιο υπάρχει και ένα αντίστοιχο

## Κεφάλαιο 5

choice box που από εκεί βαθμολογείται. Το πλαίσιο το οποίο περιέχει τρία κουμπιά και ένα πίνακα στατιστικών. Τα κουμπιά είναι το “NEW GENERATION”, το “FINAL GENERATION” και το “RESET”.

## GENERATION OPTIONS

Στο πλαίσιο επιλογές γενεάς (generation options) υπάρχουν δύο επιλογές. Η πρώτη είναι το “Number of designs per generation”, δηλαδή το μέγεθος του πληθυσμού ανά γενιά. Από εκεί ο χρήστης μπορεί να επιλέξει τον αριθμό που θέλει (προσφέρονται αριθμοί από το δύο μέχρι το δέκα). Η επιλογή αυτή επηρεάζει την καρτέλα γεννών (generations tab) και φυσικά το σύνολο των σχεδίων που προβάλλονται ανά γενιά. Η δεύτερη επιλογή “Number of maximum generations” καθορίζει τον έναν από τους τρεις τρόπους που τερματίζει ο βρόχος εξέλιξης. Αποτελεί λοιπόν τον μέγιστο αριθμό γενεών. Αυτός μπορεί να πάρει να πάρει αριθμό από το εύρος έξι μέχρι τριάντα γενιών.



Σχήμα 5. 9: Τα πλαίσια του OPTIONS TAB και το κουμπί GENERATE.

## FILTERS

Τα φίλτρα (filters) επηρεάζουν την αρχικοποίηση του πρώτου πληθυσμού. Υπάρχουν επιλογές για όλα τα χαρακτηριστικά που έχουν τα αντίστοιχα πρότυπα ενδυμάτων. Τα φίλτρα επηρεάζουν με πιθανότητα 50% κάθε άτομο ξεχωριστά, δηλαδή κάθε ένα κατά την αρχικοποίηση του έχει 50% πιθανότητα να σχεδιαστεί σύμφωνα με τις επιλογές του χρήστη. Αυτό γίνεται για να υπάρχει ποικιλία ανάμεσα στα σχέδια. Δίνει επίσης την επιλογή στον χρήστη να συνδυάσει αυτό που θέλει με κάτι καινούργιο. Τα φίλτρα για κάθε διαφορετικό πρότυπο, αρχικά είναι απενεργοποιημένα και ενεργοποιούνται όταν ο χρήστης επιλέξει το πρότυπο ενδύματος μέσω του categories menu.

## TRENDS

Οι τάσεις (trends), οπτικά είναι το ίδιο πλαίσιο με τα φίλτρα. Οι επιλογές όμως που πραγματοποιούνται στις τάσεις, επηρεάζουν τον τελεστή επιβίωσης. Συνεπώς, δίνεται η δυνατότητα

στον χρήστη να ορίσει τις δικές του τάσεις και τα σχέδια γενιά με γενιά, έχουν μεγάλες πιθανότητες να συγκλίνουν προς τα εκεί. Τα φίλτρα αυτά είναι πάντα ενεργοποιημένα.

## RECENT FINAL GENERATIONS

Το πλαίσιο “Recent Final Generations” φιλοξενεί τις φωτογραφίες των βέλτιστων λύσεων. Πριν την ολοκλήρωση μιας πλήρης εξέλιξης παραμένει άδειο, καθώς όπως έχει ήδη αναφερθεί η εφαρμογή δεν συνδέεται με βάση δεδομένων. Κατά την ολοκλήρωση όμως κάθε εξέλιξης, η φωτογραφία κατεβαίνει αυτόματα στην επιφάνεια εργασίας του χρήστη. Δεν μπορεί να δεχτεί παραπάνω από δύο φωτογραφίες και στην περίπτωση τρίτης η παλαιότερη χρονικά χάνεται.

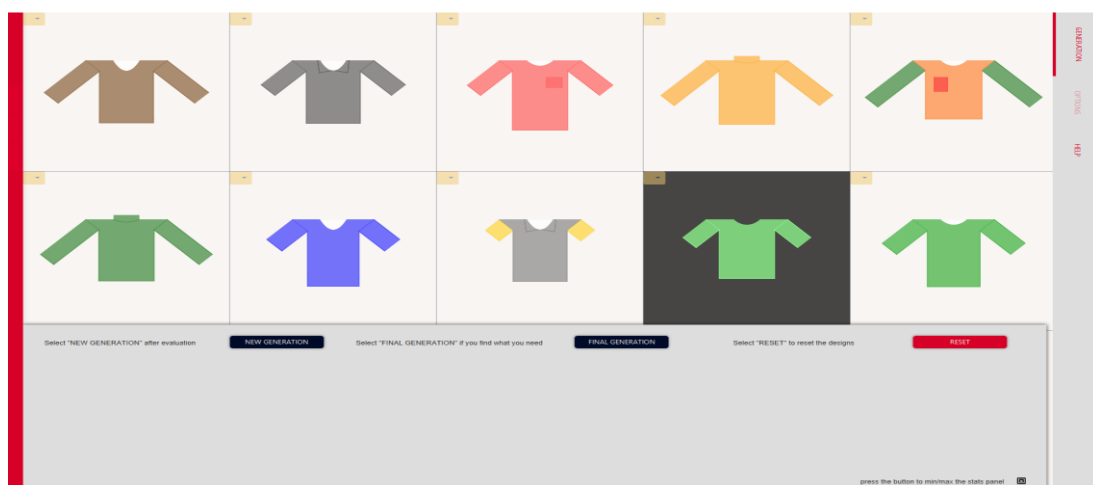
## GENERATE BUTTON

Το κουμπί “GENERATE”, είναι αυτό που πατάει ο χρήστης για να γίνει η αρχικοποίηση του πληθυσμού. Μετά το πάτημα του κουμπιού ο χρήστης μεταφέρεται αυτόματα στο generation tab και η καρτέλα OPTIONS απενεργοποιείται καθώς δεν μπορούν να γίνουν περαιτέρω επιλογές από τον χρήστη ή να τροποποιηθούν κατά την διάρκεια της εξέλιξης.

## ΠΛΑΙΣΙΟ ΣΧΕΔΙΩΝ

Το πλαίσιο σχεδίων αποτελεί ουσιαστικά την φαινοτυπική μορφή των ατόμων του πληθυσμού που αναπαρίσταται. Λόγω της χρήσης ΔΕΑ επιβάλλεται να είναι φιλικό ως προς τον χρήστη, διότι αυτός είναι υπεύθυνος για την αξιολόγηση των σχεδίων. Συνεπώς πρέπει να είναι σε θέση να τα παρατηρήσει και να κρίνει.

Στο πλαίσιο σχεδίων δημιουργούνται η κελιά για κάθε σχέδιο - άτομο του πληθυσμού (το n είναι ο αριθμός που ορίζει ο χρήστης στο Generation Options, με την επιλογή αριθμού σχεδίων για προβολή). Μέσα στα αυτά τα κελιά πέρα από το ίδιο το σχέδιο δημιουργείται ένα choice box (σχήμα 5.11) με αριθμούς από το μηδέν μέχρι το δέκα. Μέσω των choice boxes ο χρήστης βαθμολογεί όλα τα σχέδια. Δεν μπορεί να αφήσει κανένα κενό αν θέλει να προχωρήσει την διαδικασία. Για την περίπτωση εξέλιξης του προτύπου παντελονιού, επειδή ένα παντελόνι πρέπει να εξετάζεται και από

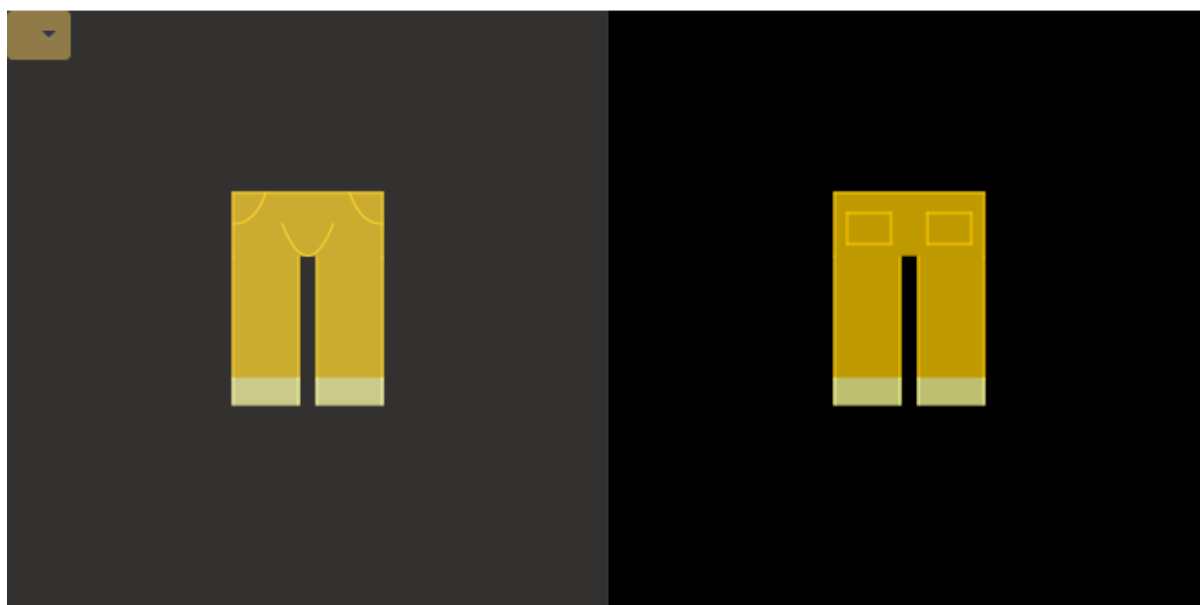


Σχήμα 5. 10: Η καρτέλα GENERATION, το πλαίσιο σχεδίων στο άνω μέρος της και το πλαίσιο ελέγχου στο κάτω μέρος της.

τις δύο πλευρές



Σχήμα 5. 11: Choice box για την βαθμολόγηση του σχεδίου από τον χρήστη.



Σχήμα 5. 12: Τρόπος προβολής των εμπρός και πίσω πλευρών για το πρότυπο του παντελονιού.

## ΠΛΑΙΣΙΟ ΕΛΕΓΧΟΥ

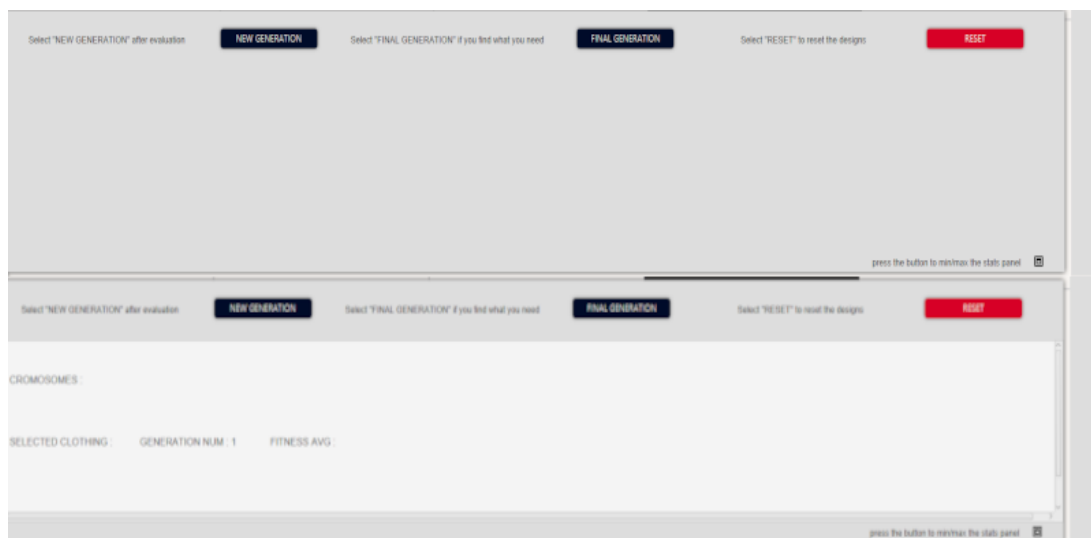
Το πλαίσιο ελέγχου είναι αυτό που δίνει τον έλεγχο στον χρήστη σχετικά με το ποια είναι η ενεργεία που γίνεται στην συνέχεια. Πέρα από αυτό όμως έχει και έναν πίνακα στατιστικών ο οποίος είναι κρυφός αρχικά γιατί πιθανόν να μην απασχολεί τον χρήστη που σκοπός του είναι να βρει βέλτιστη λύση. Παρακάτω αναλύονται τα τρία κουμπιά και ο πίνακας ελέγχου αναλυτικά.

Το κουμπί “NEW GENERATION” είναι αυτό καλείτε να πατήσει ο χρήστης μετά την αξιολόγηση έτσι ώστε να ξεκινήσει ο βρόχος του EA μέχρι να αναπαραστήσει τον πληθυσμό της επόμενης γενιάς. Αν όλα τα σχέδια δεν έχουν βαθμολογηθεί εμφανίζεται ένα πλαίσιο με το κατάλληλο μήνυμα που ειδοποιεί τον χρήστη.

Το κουμπί “FINAL GENERATION”, είναι αυτό που τερματίζει την εξέλιξη του προτύπου. Με την επιλογή του η εξέλιξη σταματάει και δεν μπορεί να συνεχίσει. Επίσης δημιουργεί μια εικόνα στην οποία απεικονίζεται το πλαίσιο σχεδίων, την οποία αποθηκεύει στην επιφάνεια εργασίας καθώς και στο πλαίσιο Recent Final Generations του OPTIONS TAB.

Τελευταίο από τα κουμπιά είναι το “RESET”. Το κουμπί αυτό σταματάει την εξέλιξη του προτύπου δίχως να βρεθεί η βέλτιστη λύση. Ο χρήστης πρέπει να το πατήσει σε κάθε περίπτωση, ακόμα και μετά την επιλογή του “FINAL GENERATION”. Με το πάτημα του η καρτέλα GENERATION απενεργοποιείται και μεταφέρεται στην καρτέλα OPTIONS όπου έχουν αρχικοποιηθεί όλες η επιλογές που έχει κάνει ο χρήστης.

Ο πίνακας στατιστικών αφορά αποκλειστικά τον χρήστη που μελετά το πρόγραμμα από πλευράς EA. Από μόνος του είναι κρυφός και γίνεται φανερός με την επιλογή ενός κουμπιού μεγιστοποίησης (maximize button) στο κάτω δεξί άκρο του πλαισίου ελέγχου. Στο σχήμα 5.13 φαίνεται το πλαίσιο ελέγχου πριν και μετά την μεγιστοποίηση του πίνακα στατιστικών.



Σχήμα 5. 13: Πλαίσιο ελέγχου πριν και μετά την μεγιστοποίηση του πίνακα στατιστικών.

Ο πίνακας στατιστικών δείχνει το γονότυπο του “επιλεγμένου” ατόμου - σχεδίου (με ενέργειες το ποντικιού πάνω στο σχέδιο) δίπλα από την ετικέτα CHROMOSOME και τον αριθμό του

κελιού στην ετικέτα “SELECTED CLOTHING”. Επίσης δείχνει με την ετικέτα “GENERATION NUM” σε ποια γενιά βρίσκεται και τέλος με την ετικέτα “FITNESS AVG” δείχνει τον μέσο όρο του δείκτη ποιότητας όλων των ατόμων που έχουν βαθμολογηθεί.

## **5.7 Επίλογος**

Συνοψίζοντας στο κεφάλαιο αυτό είδαμε τον τρόπο προσέγγισης για την δημιουργία του ΔΕΑ με την χρήση της λογικής μιας ΓΣ που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής μας. Στην αρχή είδαμε τις τεχνολογίες που χρησιμοποιήθηκαν για την καλύτερη κατανόηση των τελεστών. Μετά είδαμε τον ρόλο των ΓΣ στην εφαρμογή. Για τον ΔΕΑ, εξετάσουμε τις μεθόδους δημιουργίας τελεστών που μπορούσαν να έχουν χρησιμοποιηθεί καθώς κι αυτούς που χρησιμοποιήθηκαν. Τέλος, αφού παρουσιάστηκαν τα “υλικά” για την κατασκευή των λειτουργιών που εκτελεί ο αλγόριθμος της εφαρμογής, παρουσιάστηκε η διεπαφή και ο τρόπος λειτουργίας της.



## Κεφάλαιο 6ο : Αποτελέσματα Εφαρμογής

### 6.1 Εισαγωγή

Η εφαρμογή που υλοποιήθηκε για την αυτόματη σχεδίαση ενδυμάτων με ΔΕΑ στα πλαίσια αυτής της Π.Ε., ακολουθεί το μοντέλο προσέγγισης εφαρμογής της ΔΕΑ με χρήση της ΓΣ. Το τελικό αποτέλεσμα διαχειρίζεται με επιτυχία τους εξελικτικούς τελεστές και καταφέρνει να αποφέρει αποτελέσματα προτείνοντας στον χρήστη ικανοποιητικές λύσεις, καθώς δεν είναι δυνατόν να χαρακτηριστούν βέλτιστες, αφού το βέλτιστο στην περίπτωση της σχεδίασης ενδυμάτων είναι υποκειμενικό.

Ένα τρόπος αντιμετώπισης για την εύρεση ενός “βέλτιστου” σχεδίου στην σχεδίαση ενδυμάτων είναι η αντιστοίχιση σιλουέτας με την κατάλληλη υποκατηγορία ενδύματος (σωστές αναλογίες). Η προσαρμογή όμως κατάλληλης σιλουέτας ακόμα και σε ένα διδιάστατο σχέδιο, είναι ιδιαίτερα χρονοβόρα διαδικασία, θα απαιτούσε μεγάλη σχεδιαστική δεινότητα, καθώς και τις σωστές αναλογίες. Έτσι, η προσοχή στράφηκε περισσότερο στις εξελικτικές διαδικασίες της εφαρμογής, δίνοντας στον χρήστη την δυνατότητα να ορίσει αυτός την βέλτιστη λύση.

Η χρήση Γραμματικής Σχημάτων (ΓΣ) στο πρόβλημα της σχεδίασης ενδυμάτων, αποδείχθηκε αρκετά περιορισμένη σε ένα σύστημα διάδρασης με τον χρήστη. Η δομή ενός ρούχου δεν μπορεί να επηρεαστεί εύκολα έως και καθόλου. Έτσι δημιουργήθηκαν ομάδες κανόνων που αφορούν τον διαχωρισμό ενός ρούχου σε κομμάτια και κάθε κανόνας είναι πρακτικά ένα γονίδιο του γονότυπου του ατόμου. Για παράδειγμα στο πρότυπο της μπλούζας έχουμε το μανίκι που αποτελεί μια ομάδα κανόνων. Το μανίκι από μόνο του δεν έχει αρχικό σχήμα καθώς μπορεί να έχουμε τη περίπτωση αμάνικης μπλούζας, αυτό φυσικά αποτελεί έναν περιορισμό κατά τον οποίο δεν θα σχεδιαστεί κανένας από τους κανόνες της ομάδας αυτής. Οι δύο υποκανόνες του μανικιού είναι το ίδιο το μανίκι και το τελείωμα του.

Άλλο ένα πρόβλημα που παρουσιάστηκε μετά την δημιουργία των εξελικτικών τελεστών παρατηρήθηκε στην προβολή των αποτελεσμάτων - σχεδίων. Η προσέγγιση της σχεδίασης ενδυμάτων με ΔΕΑ και με ΓΣ αποδείχθηκε ιδιαίτερα πεινασμένη από την άποψη χαρακτηριστικών. Η έλλειψη ελευθερίας που επιβάλλεται στον αλγόριθμο για να διατηρηθεί η άμεση και σωστή δομή των ενδυμάτων, έτσι ώστε η εφαρμογή να μην χρειάζεται να κουράζει τον χρήστη με πολύ μεγάλους αριθμούς γενεών, έφερε την άμεση επανάληψη μοτίβων των σχεδίων. Η εύρεση όμως χαρακτηριστικών στα δουλεμένα πρότυπα ήταν δύσκολη. Έτσι, για την αντιμετώπιση αυτού του προβλήματος δόθηκε μεγαλύτερη πιθανότητα στον τελεστή μετάλλαξης, αλλά αυτό κάποιες φορές επηρεάζει αρνητικά τα αποτελέσματα.

Μετά την παράθεση κάποιων προβλημάτων τα οποία παρουσιάστηκαν κατά την διαρκεία δημιουργίας της εφαρμογής δίνεται ένα παράδειγμα για το πρότυπο της μπλούζας και αναλύεται η ορθή λειτουργία της εξέλιξης σε αυτό.

### 6.2 Εξέλιξη Δύο Σεναρίων

Στην ενότητα αυτό παρουσιάζονται δύο σενάρια εφαρμογής του προγράμματος ένα για το πρότυπο της μπλούζα και ένα για αυτό του παντελονιού, πριν όμως παρουσιαστούν τα δύο σενάρια,

## Κεφάλαιο 6

στον πίνακα 6.1 παρουσιάζονται αριθμημένες όλες οι παράμετροι που χρησιμοποιεί ο ΔΕΑ για την διευκόλυνση των αναφορών, μέσω των αντίστοιχων αριθμών.

Ο πίνακας 6.2 μας δείχνει τις τιμές των παραμέτρων του ΔΕΑ, οι οποίες είναι κοινές στα δύο σενάρια που αναπτύσσονται. Οι τιμές που χρησιμοποιήθηκαν για την ανάπτυξη των δύο σεναρίων, παραμένουν οι ίδιες και στο τελική εφαρμογή με την διαφορά πως οι τιμές (1), (3) και (4) μπορούν να αλλάζουν από την παρέμβαση του χρήστη από το panel Generation Options.

Πίνακας 6. 1: Παράμετροι ΕΑ

Πίνακας 6. 2: Παράμετροι ΕΑ

1	Μέγεθος Αρχικού Πληθυσμού
2	Μέγεθος Πλήθους Αναπαραγώμενου Πληθυσμού
3	Μέγεθος Τελικού Πληθυσμού
4	Μέγιστο Πλήθος Γεννεών
5	Τελεστής Επιλογής
6	Τελεστής Μετάλλαξης: Πιθανότητα

Πίνακας 6. 3: Προκαθορισμένες τιμές παραμέτρων ΕΑ

Αριθμηση	Τιμή	Πρότυπο Μπλούζας
1		10
2		200
3		10
4		10
5		Roulette Wheel
6		20%

Το μέγεθος του αρχικού πληθυσμού και το μέγεθος του τελικού πληθυσμού και κάθε άλλου ενδιάμεσου πληθυσμού που αξιολογεί ο χρήστης είναι πάντα σταθερό ((1) = (3)). Το κριτήριο τερματισμού που χρησιμοποιεί είναι το μέγιστο πλήθος γενεών. Ο αριθμός αυτός είναι ο προκαθορισμένος αριθμός που προσφέρει το πρόβλημα καθώς και ο μικρότερος που μπορεί να δεχτεί.

Η αξιολόγηση πραγματοποιείται από τον χρήστη. Για την επιλογή γονέων έχει επιλεγθεί ο τελεστής περιστρεφόμενης ρουλέτας (Roulette Wheel), ο οποίος είναι ο μοναδικός που χρησιμοποιεί η εφαρμογή. Οπότε άσχετα με την βαθμολογία του χρήστη, οι γονείς προκύπτουν από μια διαδικασία τυχαίας επιλογής στην οποία απλά ο χρήστης ορίζει τις πιθανότητες. Για τον τελεστή επιβίωσης, ο χρήστης μπορεί να θέσει μέσω του Trends panel κάποια από τα χαρακτηριστικά που επιθυμεί να υπάρχουν στα σχέδια του. Έτσι όσα περισσότερα από αυτά τα χαρακτηριστικά έχει το σχέδιο του αναπαραγόμενου πληθυσμού τόσο πιο ψηλά είναι στην ιεραρχία για να επιβιώσει. Για το στάδιο του

ανασυνδυασμού, ο τελεστής διασταύρωσης εκτελείται πάντα και παράγει τον αναπαραγόμενο πληθυσμό, ενώ ο τελεστής μετάλλαξης έχει πιθανότητα 20%. Οι τιμές αυτές έχουν ήδη αναφερθεί και δεν έχουν αλλάξει για την εκτέλεση των σεναρίων.

Τα δύο σεναρία που ακολουθούν ασχολούνται το καθένα για διαφορετικό πρότυπο. Για το καθένα υπάρχουν διαφορετικά ζητούμενα Trends. Το ένα γίνεται χωρίς την τροφοδοτήσει από τον χρήστη, δηλαδή με τυχαία αρχικοποίηση, ενώ στο άλλο έχει δοθεί μερική τροφοδότηση. Και για τα δύο σεναρία ακολουθούν σχήματα από τον αρχικό, ενδιάμεσο και τελικό πληθυσμό.

### 6.2.1 Σενάριο μπλούζας

Για το σενάριο εξέλιξης του προτύπου της μπλούζας, δόθηκαν στο panel των Trends οι παρακάτω επιλογές χαρακτηριστικών που φαίνονται στο σχήμα 6.1. Όπως είπαμε αυτές οι επιλογές παίζουν ρόλο στον τελεστή επιβίωσης. Επειδή η εύρεση βέλτιστης λύσης σε προβλήματα σχεδίασης ενδυμάτων με την χρήση ΔΕΑ είναι υποκειμενική, το σενάριο αναπτύσσεται μέχρι να βρεθεί το άτομο σχέδιο με τα παρακάτω χαρακτηριστικά. Υπάρχει η περίπτωση που δεν βρέθηκε το ζητούμενο βέλτιστο και έτσι παρουσιάζονται απλά τα αποτελέσματα του τελευταίου πληθυσμού και εξετάζονται ποια από τα ζητούμενα χαρακτηριστικά υπάρχουν στο τελικό σχέδιο.

**Trends** Trends will affect the designs after the first generation

**T-Shirt**

Color : RED

Type of sleeves : LONG SLEEVES

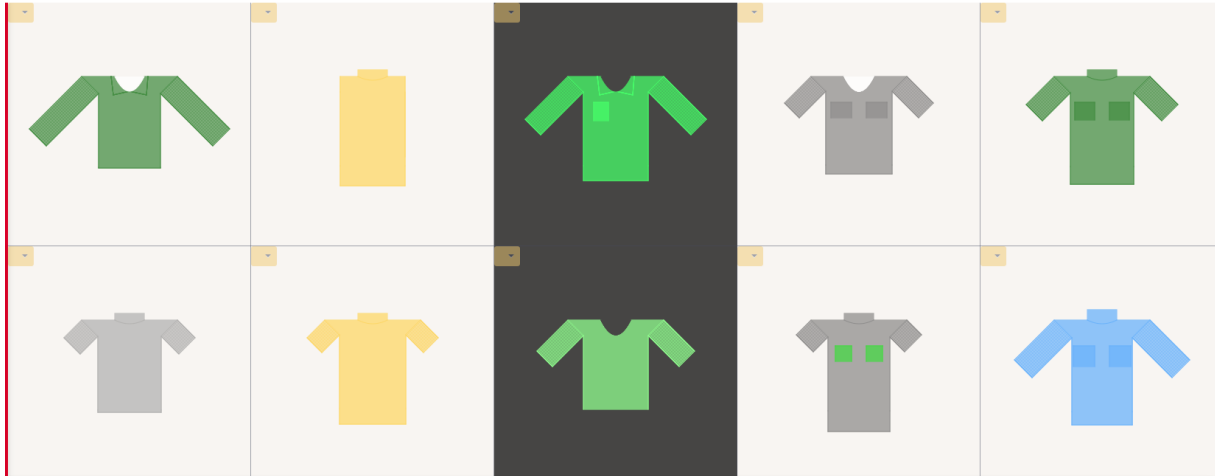
Length : SHORT

Pockets : YES

Collar : ROLL

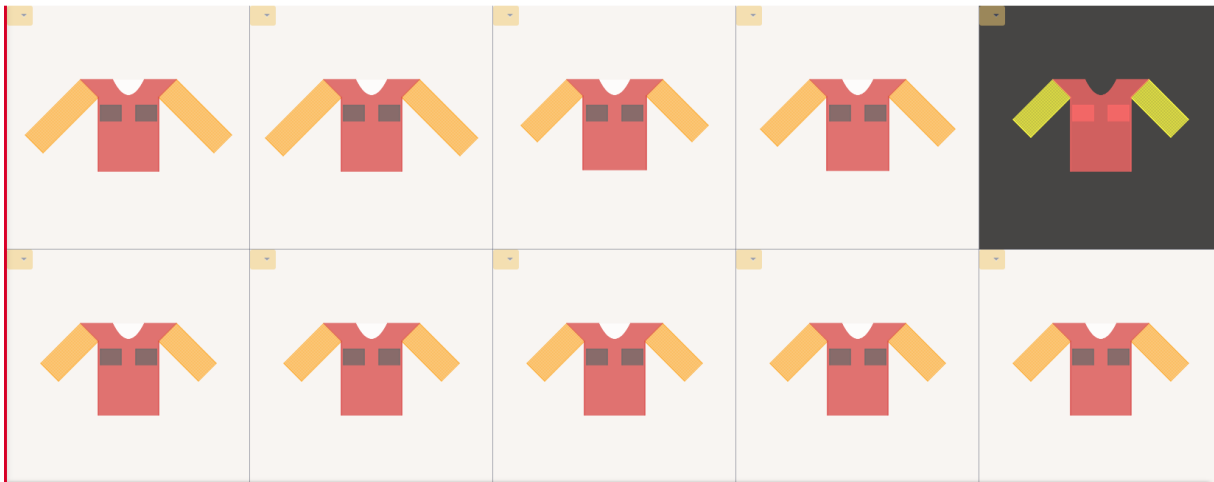
Σχήμα 6. 1: Καθορισμός τιμών βέλτιστης λύσης.

Η αρχικοποίηση στο σενάριο της μπλούζας γίνεται με τυχαία αρχικοποίηση δηλαδή χωρίς την τροφοδότηση από την χρήστη με την βοήθεια των Filters. Ο αρχικός πληθυσμός που προέκυψε φαίνεται στο σχήμα 6.2, όπου φαίνεται ότι υπάρχουν κάποια από τα ζητούμενα χαρακτηριστικά σε κάποια σχέδια (τα μανίκια, οι τσέπες κ.α.), αλλά όχι όλα πάνω σε ένα σχέδιο. Αυτό φυσικά θα ήταν πολύ δύσκολο να συμβεί με τυχαία αρχικοποίηση. Μην ξεχνάμε πως το νόημα της εφαρμογής είναι ο χρήστης να δει νέα σχέδια και ιδέες και όχι να σχεδιάσει απευθείας αυτό που θέλει.



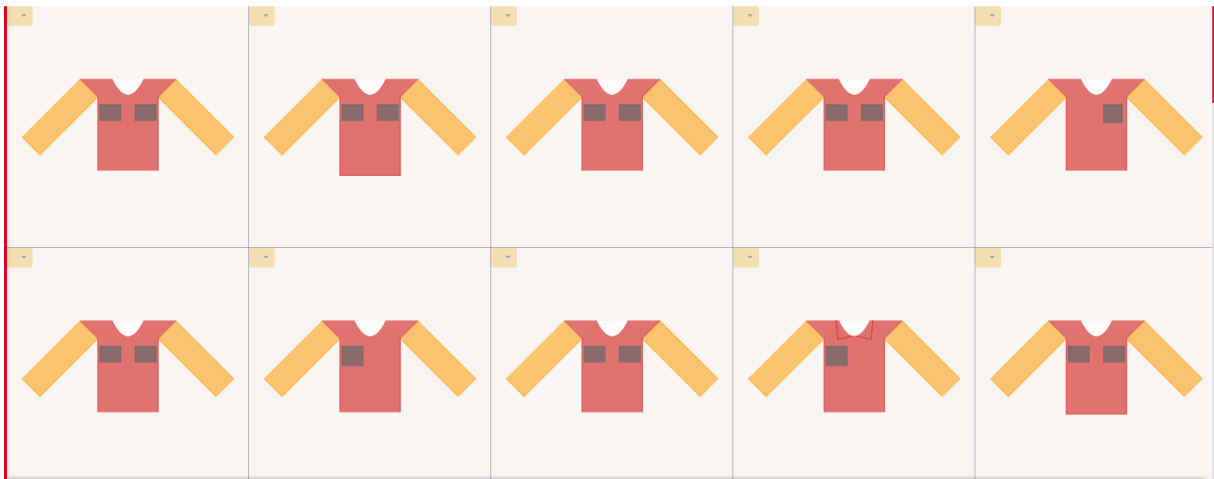
Σχήμα 6. 2: Αρχικός Πληθυσμός

Στο σχήμα 6.3 παρατηρούμε ότι υπάρχει μεγάλη επανάληψη των σχεδίων καθώς μόνο λίγα χαρακτηριστικά αλλάζουν μεταξύ τους και αυτό γίνεται κατά κύριο λόγο με την εφαρμογή της μετάλλαξης. Οι επιλογές των Trends του τελεστή επιβίωσης έχουν τεράστια επιρροή στα τελικά σχέδια ειδικά όταν επιλέγονται όλα όπως στο παράδειγμα αυτό. Αν παρατηρήσουμε καλά τα σχέδια, μόνο το χαρακτηριστικό της λαιμόκοψης roll λείπει από αυτά. Φυσικά, ο μέσος όρος ποιότητας ανεβαίνει από γενιά σε γενιά καθώς τα σχέδια έχουν τα χαρακτηριστικά που έχει επιλέξει ο χρήστης.



Σχήμα 6. 3: Μεσαίος Πληθυσμός

Στο σχήμα 6.4 βλέπουμε τον τελικό πληθυσμό. Όπως φαίνεται ο τελικός πληθυσμός με αυτόν του μεσαίου πληθυσμού δεν διαφέρει πολύ με την διαφορά πως τα περισσότερα σχέδια πλέον διαφέρουν ακόμα λιγότερο μεταξύ τους. Αυτό φυσικά προκύπτει από τα Trends καθώς τα αναπαραγόμενα σχέδια συνέχισαν να παράγουν σχέδια με τα επιθυμητά χαρακτηριστικά και έτσι επιλέχθηκαν από τον τελεστή επιβίωσης.



Σχήμα 6. 4: Τελικός Πληθυσμός

Στο τελικό αποτέλεσμα που προέκυψε από το πέρας των δέκα γενεών. Ο ΔΕΑ δεν κατάφερε να βρει την βέλτιστη για τον χρήστη λύση. Συγκεκριμένα δεν κατάφερε να βρει το χαρακτηριστικό της λαιμόκοψης. Παρόλα αυτά όπως είναι φανερό ο μέσος όρος ποιότητας μονάχα ανέβαινε. Αυτό, θα μπορούσε να αλλάξει αν το τελεστής μετάλλαξης είχε μεγαλύτερη πιθανότητα εφαρμογής, αλλά

δεν ξέρουμε πως θα μπορούσε να επηρεάσει και τα υπόλοιπα χαρακτηριστικά. Άλλη μια λύση θα ήταν να αλλάξει το μέγιστο πλήθος γενεών σε μεγαλύτερο αριθμό.

Συμπερασματικά, ο ΔΕΑ στο σενάριο της μπλούζας λειτούργησε αλλά με την τροφοδότηση του κριτηρίου επιλογής περιορίσε άμεσα την ποικιλομορφία των πληθυσμών. Η έλλειψη χαρακτηριστικών στο πρότυπο της μπλούζας και ειδικότερα με την χρήση των Trends δημιουργεί μια άμεση εξέλιξη που καταλήγει σύντομα σε ένα ικανοποιητικό σχέδιο αλλά όχι πάντα στο βέλτιστο.

### 6.2.2 Σενάριο Παντελονιού

Στο σενάριο παντελονιού, μπαίνει σε χρήση το panel Filters όπου υπάρχουν πιθανότητες να επηρεάσει την αρχικοποίηση του πληθυσμού, καθώς και ελαττώνονται οι επιλογές στο panel Trends με σκοπό της προβολή μεγαλύτερης ποικιλίας σχεδίων κατά την διάρκεια της εξέλιξης. Όπως φαίνεται στο σχήμα 6.5 για τα Filters επιλέχθηκαν τα τρία πρώτα χαρακτηριστικά, ενώ για το τα Trends που φαίνονται στο σχήμα 6.6 τα δύο τελευταία. Ο στόχος είναι για την αρχικοποίηση με χαρακτηριστικά που θέλουμε να έχουμε στην αρχή και με άλλα που αποτελούν τα κριτήρια επιβίωσης, να δούμε αν θα καταφέρουμε να έχουμε ένα σχέδιο που τα συνδυάζει όλα.

Ο αρχικός πληθυσμός που προέκυψε έχει σαφέστατα κάποια από τα χαρακτηριστικά με τα οποία τροφοδοτήθηκε από το panel filters (σχήμα 6.7). Κατά κύριο λόγο φαίνεται το μπλε χρώμα το οποίο υπάρχει σε 7 από τα 10 σχέδια, κάποια είναι ψηλοκάβαλα και μόλις ένα δεν είναι μακρύ.

Στο σχήμα 6.8 φαίνεται το μεσαίος πληθυσμός δηλαδή η γενιά νούμερο πέντε κατά σειρά. Μπορούμε να παρατηρήσουμε ότι τα σχέδια σε σχέση με το σενάριο μπλούζας παρουσιάζουν μεγαλύτερη ποικιλία και θα παρουσιάζουν ακόμα περισσότερη αν ο στόχος τον αξιολογήσεων που έγιναν μέχρι τότε δεν είχε στόχο τον συνδυασμό των επιθυμητών χαρακτηριστικών.

The image shows a 'Filters' panel with a red header. Under the heading 'Trousers', there are five filter options, each with a dropdown menu:

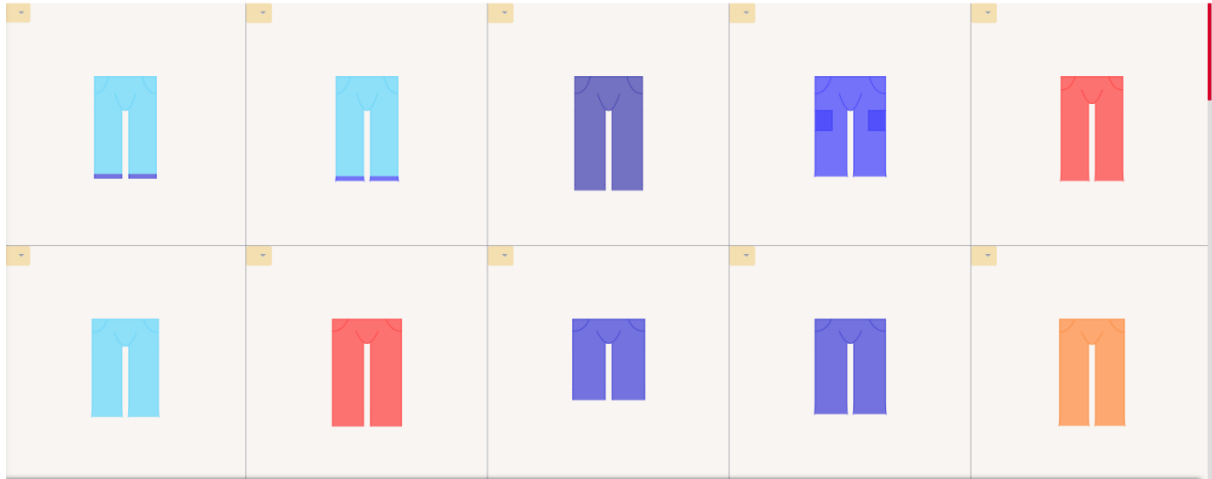
- Color : BLUE
- Pants type : LONG
- Crotch type : HIGH
- Pockets : ANY
- Legs width : ANY

Σχήμα 6. 5: Επιλογές στο Filters panel για το πρότυπο παντελονιού.

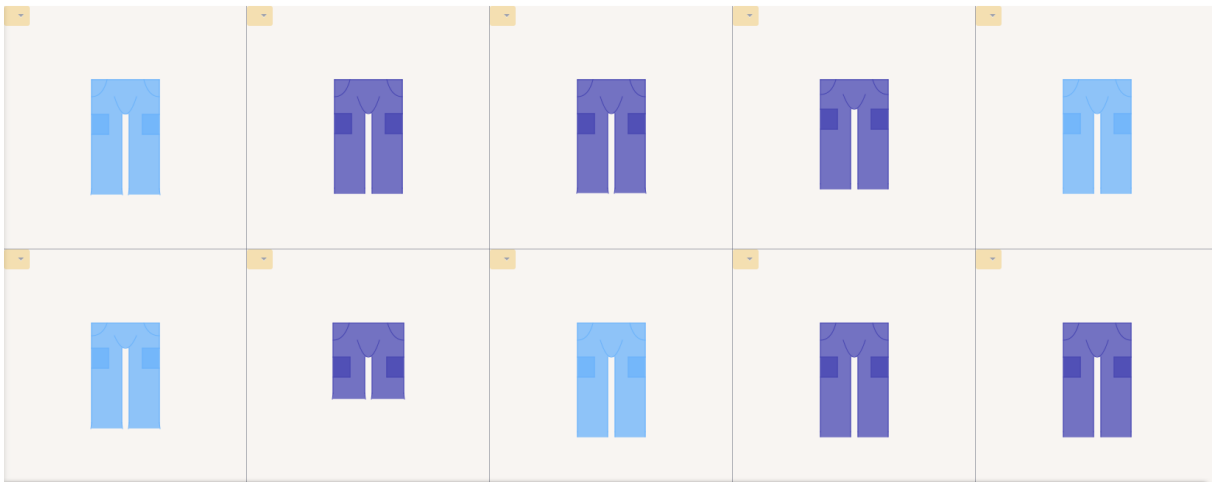
The image shows a 'Trends' panel with a red header. Under the heading 'Trousers', there are five filter options, each with a dropdown menu:

- Color : (empty)
- Pants type : ANY
- Crotch type : ANY
- Pockets : YES
- Legs width : REGULAR

Κεφάλαιο 6



Σχήμα 6. 7: Αρχικός Πληθυσμός



Σχήμα 6. 8: Μεσαίος Πληθυσμός



Σχήμα 6. 9: Τελικός Πληθυσμός

Στο σχήμα 6.9 βλέπουμε τον τελικό πληθυσμό, στον οποίο έχουμε 9 στα 10 σχέδια αυτό που ψάχναμε εξ αρχής. Όλα τα παντελόνια είναι μπλε και έχουν τσέπες (οι μεσαίες τσέπες είναι οι ζητούμενες, καθώς οι πάνω τσέπες τοποθετούνται σε όλα τα παντελόνια). Σχεδόν όλα είναι μακριά και έχουν ψηλό καβάλο.

Το σενάριο του παντελονιού μας δείχνει ότι η χρήση των Trends δηλαδή των κριτηρίων επιβίωσης επιβαρύνει αρκετά την ποικιλία των σχεδίων αν και έχει άμεσα αποτελέσματα δεν εγγυάται την εύρεση βέλτιστων λύσεων. Ο συνδυασμός Filters και Trends στο συγκεκριμένο παράδειγμα έδειξε μεγαλύτερη αποτελεσματικότητα αλλά με μικρότερη ταχύτητα. Φυσικά στο τέλος το αποτέλεσμα ήταν σχεδόν το ίδιο, αλλά αν κάθε πρότυπο είχε ακόμα περισσότερο χαρακτηριστικά τότε θα ήταν πιο ξεκάθαρη η διαφορά στους χρόνους.

### 6.3 Επίλογος

Ανακεφαλαιώνοντας, σε αυτό το κεφάλαιο παρουσιάστηκαν τα προβλήματα που εντοπίστηκαν κατά την διάρκεια υλοποίησης της εφαρμογής για την επίλυση στο πρόβλημα σχεδίαση ενδυμάτων με ΔΕΑ, με την χρήση ΓΣ. Στην συνέχεια αναφέρθηκαν οι παράμετροι και οι τελεστές που επιλέχθηκαν για την υλοποίηση των λειτουργιών του ΔΕΑ. Τέλος δόθηκαν δύο σενάρια με τις πραγματικές παραμέτρους των ΔΕΑ και με την μόνη διαφορά το κριτήριο επιβίωσης, καθώς και σε ένα από τα δύο σενάρια πραγματοποιήθηκε αρχικοποίηση με κατεύθυνση. Συμπερασματικά το κάθε σενάριο έχει την δική του χρήση, κάπου υστερεί και κάπου όχι.



## Κεφάλαιο 7ο : Συμπεράσματα και Περαιτέρω Βελτιώσεις

### 7.1 Εισαγωγή

Στο τελευταίο κεφάλαιο αυτής της Π.Ε. παρατίθενται τα συμπεράσματα καθώς και προτάσεις για την βελτίωση της εφαρμογής. Είναι πολύ σημαντικό μετά την ολοκλήρωση ενός έργου γενικότερα η μιας εφαρμογής ειδικότερα, να πραγματοποιείται μια σύνοψη στο αν έχει πετύχει το σκοπό για τον οποίο δημιουργήθηκε η όχι, όπως επίσης τι θα μπορούσε να την βοηθήσει για να αναπτυχθεί περαιτέρω.

### 7.2 Συμπεράσματα

Η ανάληψη του θέματος της εργασίας και η ενασχόληση με το θέμα εφαρμογής βιολογικών διαδικασιών για την σχεδίαση υπήρξε κάτι πρωτόγνωρο για μένα. Κατά την διάρκεια της εξοκείωσης μέσω της έρευνας, ακόμα και τον πιο απλών πτυχών της επιστήμης των ΕΑ, μου έγινε σαφές ότι πρόκειται για μια εξαιρετικά ενδιαφέρουσα επιστήμη.

Μετά την κατανόηση των βασικών εννοιών και των γενικών κατηγοριών των ΕΑ, ακολουθούσε η ερευνά της εφαρμογής τους για την σχεδίαση ενδυμάτων. Εκεί, παρουσιάστηκαν αρκετές επιλογές προσέγγισης, άλλες εφικτές και άλλες απλησίαστες λόγω του όγκου πόρων και χρόνου που απαιτούνται για την υλοποίησή τους. Έτσι υπήρξε η απόφαση ανάπτυξης ενός ΔΕΑ βασισμένου πάνω σε κάποιον ήδη έτοιμο αλλά με κάποιες διαφορές για να δημιουργηθεί κάτι μοναδικό και με όσο λιγότερες απαιτήσεις θα μπορούσε να έχει. Όπως να αποφευχθεί η δημιουργία εικόνων για την αναπαράσταση των επιμέρους κομματιών που χωρίζεται μια μπλούζα. Η ανάγκη δημιουργίας μιας βάσης που με άπειρες φωτογραφίες, καθώς και ο χρόνος σχεδίασης αυτών, θα έπαιρνε το μεγαλύτερο μέρος του χρόνου υλοποίησης της εφαρμογής, καθώς και θα ήταν μεγάλο βάρος για το πρόγραμμα. Έτσι με βάση μιας ήδη γνωστής προσέγγισης και με πειραματικές τροποποιήσεις σε τελεστές και λειτουργίες, δημιουργήθηκε η εφαρμογή.

Το τελικό αποτέλεσμα, κρίθηκε εν μέρη πετυχημένο ως προς τις λειτουργίες των ΕΑ που χρησιμοποιεί. Η εφαρμογή των τελεστών, επιλογής, διασταύρωσης και μετάλλαξης λειτουργούν ακριβώς έτσι όπως οραματίστηκαν από την αρχή. Ο τελεστής επιλογής όμως έγινε θέμα συζήτησης, καθώς ο σκοπός της επιλογής, σε προβλήματα σχεδίασης με ανθρώπινη διάδραση, καλό είναι να μην αφορά την εύρεση βέλτιστων λύσεων που αφορούν απλά τα χαρακτηριστικά αυτών. Η χρήση ΓΣ δεν ήταν στα αρχικά πλάνα, αλλά η σχεδίαση με γεωμετρικά σχέδια η οποία άρχισε να υλοποιείται έκανε απαραίτητη την χρήση της. Αν και δεν χρησιμοποιείται ακριβώς όπως ορίζεται, ο ΔΕΑ που υλοποιήθηκε δανείζεται κάποια στοιχεία της.

Η διεπαφή που δημιουργήθηκε, είναι αρκετά ευχάριστη ως προς τον χρήστη, όσον αφορά τον σχεδιασμό τα χρώματα και γενικότερα τον γραφικό περιβάλλον, αλλά με την δημιουργία του δευτέρου panel Trends ίσως προβληματίσει τον χρήστη ως προς το τι σκοπό έχει. Θεωρούμε ότι η διαδικασία αξιολόγησης από τον χρήστη είναι αρκετά εύκολη και δεν τον κουράζει. Η αναπαράσταση των σχεδίων, δεν κρίθηκε ιδιαίτερα ρεαλιστικοί για έναν σχεδιαστή, λόγω της έλλειψης χαρακτηριστικών και αριθμών αναλογιών που θα έδιναν περισσότερη ζωή στα ενδύματα.

Η εφαρμογή δουλεύει σχεδόν άψογα κατά την εξελικτική διαδικασία αλλά δεν ανταποκρίνεται στις απαιτήσεις που έχει ένας σχεδιαστής, ο οποίος επιζητά κάτι πιο κοντά σε πραγματικά σκίτσα ρούχων ή απλά καλύτερα σχεδιασμένα ρούχα.

### 7.3 Βελτιώσεις

Οι βελτιώσεις που προτείνονται δεν έχουν να κάνουν με την βελτίωση του κώδικα, αλλά την βελτίωση των εξελικτικών μεθόδων και τις αναπαραστάσεις. Η βελτιστοποίηση του κώδικα, απαιτεί άλλου είδους μελέτη και θεωρούμαι ότι, έχει ήδη εφαρμοστεί.

Για την βελτίωση του τελεστή επιβίωσης, το κριτήριο αξιολόγησης θα μπορούσε να αφορά την φόρμα κάθε ενδύματος με βάση κάποιον ζητούμενο σωματότυπο, όπου με βάση των συνδυασμό χαρακτηριστικών και αναλογιών του ενδύματος θα κρίνεται αν και πόσο κατάλληλο είναι το ένδυμα του αναπαραγόμενου πληθυσμού για να επιβιώσει. Άλλη μια πρόταση βελτίωσης για το κριτήριό αξιολόγησης του τελεστή επιβίωσης, αφορά την αξιολόγηση των ενδυμάτων με βάση τον συνδυασμό χαρακτηριστικών όπως το είδος της κατάλληλης λαιμόκοψης για τα μανίκια . Αυτό θα μπορούσε να υλοποιηθεί, αλλά με τις κατάλληλες τροποποιήσεις, εισάγοντας κανόνες όπου ο συνδυασμός χαρακτηριστικών με τις κατάλληλες τιμές θα έδιναν μεγαλύτερη βαθμολογία και θα είχαν περισσότερες πιθανότητες για να επιβιώσουν.

Για την αναπαράσταση των σχεδίων με τρόπο που θα ανταποκρινόταν στις ανάγκες των σχεδιαστών, το σχέδιο θα μπορούσε να είναι κάποιο σκίτσο η κάποια ένδυμα μεγαλύτερης λεπτομέρειας αντί για την αναπαράσταση με γεωμετρικά σχέδια. Αυτό φυσικά απαιτεί βοήθεια από άτομα σχετικά του χώρου τα οποία και θα έδιναν περισσότερα σημεία ή χαρακτηριστικά καθώς και τις σωστές αναλογίες. Τέλος, ακόμα και η οπτική αναπαράσταση με μια βάση φωτογραφιών, όπου κάθε τιμή χαρακτηριστικού θα είχε την αντίστοιχη φωτογραφία στην βάση, θα έδινε σίγουρα πιο ρεαλιστικό αποτέλεσμα.

### 7.4 Επίλογος

Η εφαρμογή η οποία αναπτύχθηκε στην παρούσα ΠΕ για την επίλυση του προβλήματος της αυτόματης σχεδίασης ενδυμάτων, μπορεί να μην βρίσκεται σε επίπεδο να σταθεί ανταγωνιστικά απέναντι σε άλλες που χρησιμοποιούνται στον χώρο της μόδας, αλλά αυτή η πρώτη προσέγγιση έχει θέσει τις απαραίτητες βάσεις για μια ρεαλιστική και πραγματικά χρήσιμη διαδραστική εφαρμογή η οποία θα προτείνει κάποιες ιδέες σχεδίασης ενδυμάτων σε έναν επαγγελματία του χώρου.



## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

- [1] Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.
- [2] De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. (Doctoral dissertation, University of Michigan). *Dissertation Abstracts International* 36(10), 5140B.
- [3] H.-P. Schwefel, *Kybernetische Evolution als Strategie der experimentellen Forschung in der Strömungstechnik*, Dipl.-Ing. Thesis, Technical University of Berlin, Hermann F [ottinger-Institute for Hydrodynamics, March 1965.
- [4] Rechenberg, I. (1973) *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog, Stuttgart
- [5] Koza, J. R. (1991) *Evolving a computer program to generate random numbers using the genetic programming paradigm*. *Proceedings of the Fourth International Conference on Genetic Algorithms*, 37-44. La Jolla, CA: Morgan Kaufmann.
- [6] Fogel, L.J., Owens, A.J., Walsh, M.J. (1966), *Artificial Intelligence through Simulated Evolution*, John Wiley.
- [7] Kim H-S, Cho S-B. *Application of interactive genetic algorithm to fashion design*. *Engineering Applications of Artificial Intelligence* 2000;13(6):635–44.
- [8] Gero, J.M. 1996. *Archaeological practice and gendered encounters with field data*, in R. Wright (ed.) *Gender and archaeology*: 251-80. Philadelphia: University of Pennsylvania Press.
- [9] Boden, M. A. (1992). *The Creative Mind: Myths and Mechanisms*. Basic Books
- [10] Dawkins, R. (1989). *The Evolution of Evolvability*. In Langton, C. G. (ed.), *Artificial Life. The Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*, vol. VI, September,

- [11] Bentley, P. J. and Wakefield, J. P. (1997b). Generic Evolutionary Design. In Chawdhry, P. K., Roy, R. and Pant, R. K. (eds), *Soft Computing in Engineering Design and Manufacturing*, Springer Verlag, London, Part 6, pp. 289–298.
- [12] Rosenman, M. (1997). The Generation of Form Using an Evolutionary Approach. In Dasgupta, D. and Michalewicz, Z. (eds), *Evolutionary Algorithms in Engineering Applications*, Springer-Verlag, pp. 69–86.
- [13] Pham, D. T. and Yang, Y. (1993). A Genetic Algorithm Based Preliminary Design System. *Journal of Automobile Engineers*, 207:D2, 127–133.
- [14] Frazer, J. (1995). *An Evolutionary Architecture*. Architectural Association, London
- [15] Coates, P. (1997). Using Genetic Programming and L-Systems to explore 3D design worlds. In Junge, R. (ed.), *CAAD Futures '97*, Kluwer Academic Publishers, Munich.
- [16] Rosenman, M. (1997). The Generation of Form Using an Evolutionary Approach. In Dasgupta, D. and Michalewicz, Z. (eds), *Evolutionary Algorithms in Engineering Applications*, Springer-Verlag, pp. 69–86.
- [17] STINY, G. & GIPS, J. 1972, "Shape Grammars and the Generative Specification of Painting and Sculpture", in C.V.Freiman (Ed.) *Proceedings of IFIP Congress 71*, North Holland, Amsterdam, Holland, pp. 1460-1465.
- [18] STINY, G. & MITCHELL, W J. 1980, "The Grammar of Paradise: on the Generation of Mughul Gardens", *Environment and Planning B: Planning and Design*, 7, pp. 209-226.



## ΠΑΡΑΡΤΗΜΑ Α : Κώδικας Εφαρμογής

Στο παράρτημα αυτό θα αναλυθεί ο κώδικας που δημιουργήθηκε για τις επιμέρους λειτουργίες της διεπαφής και όχι οι εξελικτικού μέθοδοι των τελεστών. Εδώ πρέπει να αναφερθεί πως δεν είναι δυνατόν να δοθεί ο κώδικας της FXML καθώς πρόκειται για χίλιες σχεδόν γραμμές και που δημιουργεί αυτόματα ο Scene Builder όπως και επίσης η CSS που δόθηκε για την τροποποίηση των επιμέρους σχεδίων που δημιουργήθηκε από τον χρήστη αλλά δεν θεωρείται σημαντική.

### Εμφάνιση των κατάλληλων κουμπιών

Επιλέγοντας στο αριστερό κόκκινο μενού μια εκ των επιλογών εμφανίζεται και το κατάλληλο submenu το οποίο είναι πάντα κρυμμένο. Παρακάτω φαίνονται τρεις μέθοδοι που δημιουργήθηκαν για την εμφάνιση των submenu, μια για κάθε ένα από αυτά που εμφανίζει ένα και κρύβει τα υπόλοιπα.

```
//loads the options of the man menu
public void LoadManSub() {

    sp.setStyle("-fx-focus-color: #D70026;");
    womansub.setVisible(false);
    all.setVisible(false);

    if(mansub.isVisible() == false) {

        mansub.setVisible(true);
    }else {
        mansub.setVisible(false);
    }

}

//loads the options of the woman menu
public void LoadWomanSub() {

    sp.setStyle("-fx-focus-color: #D70026;");
    mansub.setVisible(false);
    all.setVisible(false);

    if(womansub.isVisible() == false) {

        womansub.setVisible(true);
    }else {
        womansub.setVisible(false);
    }

}

//loads all the options of the menu
public void LoadAllSub() {

    sp.setStyle("-fx-focus-color: #D70026;");
    mansub.setVisible(false);
    womansub.setVisible(false);

    if(all.isVisible() == false) {

        all.setVisible(true);
    }else {
        all.setVisible(false);
    }

}
```

## Ενεργοποίηση / Απενεργοποίηση Filters

Εδώ βλέπουμε δύο μεθόδους οι οποίες έχουν τα submenu του μενού κατηγοριών. Μια για κάθε πρότυπο που αναπτύχθηκε (μπλούζα, παντελόني). Όταν ο χρήστης πατήσει ένα από τα δύο απευθείας μεταφέρεται στο tab OPTIONS έπειτά ενεργοποιείται το κατάλληλο panel φίλτρων και ενεργοποιείται το κουμπί GENERATE που εκτελεί την αρχικοποίηση.

```
//enable option tab options for SHIRT
public void optionsEnabledShirt() {

    SingleSelectionModel<Tab> selectionModel = tabM.getSelectionModel();
    selectionModel.select(1);
    filter2.setDisable(true);
    filter1.setDisable(false);
    generateB.setDisable(false);

}

//enable option tab options for SHIRT
public void optionsEnabledTrousers() {

    SingleSelectionModel<Tab> selectionModel = tabM.getSelectionModel();
    selectionModel.select(1);
    filter1.setDisable(true);
    filter2.setDisable(false);
    generateB.setDisable(false);

}
```

## Filters και Trends

Στην μέθοδο fillgenn που καλείται με την ενεργοποίηση του OPTIONS tab, γεμίζουν τα κατάλληλα choice box με όλες τις πιθανές τιμές τους και δίνεται μια default value στο καθένα.

```
//fills the choice boxes of generation size, max generation number, filters and trends
public void fillgenn() {

    if(generationflag == false) {

        // fill siblings per generation choice box and set default
        for(int i = 2; i <=10; i ++) {

            gennumber.getItems().add(i);
            generationflag = true;

        }

        gennumber.setValue(10);

        // fill max generation choice box and set default
        for(int i = 6; i <= 30; i ++) {

            maxgens.getItems().add(i);
            generationflag = true;

        }

    }

}
```

```

maxgens.setValue(10);

// fill color of choice boxes and set default values
for(int i = 0; i < colorpalet.length; i++) {

    colorpal.getItems().add(colorpalet[i]);
    colorpalT.getItems().add(colorpalet[i]);
    colorpal1.getItems().add(colorpalet[i]);
    colorpalT1.getItems().add(colorpalet[i]);
}

colorpal.getSelectionModel().selectedItemProperty().addListener((v, oldValue, newValue) ->
colorpal.setStyle("-fx-background-color: "+ colorpal.getValue() + ";"));

colorpal.setValue("");

colorpalT.getSelectionModel().selectedItemProperty().addListener((v, oldValue, newValue) ->
colorpalT.setStyle("-fx-background-color: "+ colorpalT.getValue() + ";"));

colorpalT.setValue("");

colorpal1.getSelectionModel().selectedItemProperty().addListener((v, oldValue, newValue) ->
colorpal1.setStyle("-fx-background-color: "+ colorpal1.getValue() + ";"));

colorpal1.setValue("");

colorpalT1.getSelectionModel().selectedItemProperty().addListener((v, oldValue, newValue) ->
colorpalT1.setStyle("-fx-background-color: "+ colorpalT1.getValue() + ";"));

colorpalT1.setValue("");

// fill sleeves type choice box and set default
for(int i = 0; i < sleevestype.length; i++) {

    sleevesheight.getItems().add(sleevestype[i]);
    sleevesheight1.getItems().add(sleevestype[i]);
}

sleevesheight.setValue("ANY");
sleevesheight1.setValue("ANY");

// fill pockets choice box and set default
for(int i = 0; i < pocketsex.length; i++) {

    pocketsch.getItems().add(pocketsex[i]);
    pocketschT.getItems().add(pocketsex[i]);
    pocketsch1.getItems().add(pocketsex[i]);
    pocketschT1.getItems().add(pocketsex[i]);
}

```

```

pocketsch.setValue("ANY");
pocketschT.setValue("ANY");
pocketsch1.setValue("ANY");
pocketschT1.setValue("ANY");

// fill shirts height choice box and set default
for(int i = 0; i < shirth.length; i++) {

    shirtsheight.getItems().add(shirth[i]);
    panttype.getItems().add(shirth[i]);
    shirtsheight1.getItems().add(shirth[i]);
    panttype1.getItems().add(shirth[i]);

}

shirtsheight.setValue("ANY");
panttype.setValue("ANY");
shirtsheight1.setValue("ANY");
panttype1.setValue("ANY");

//fill crotch choice box and set default
for(int i = 0; i < crotchtype.length; i++) {

    crotch.getItems().add(crotchtype[i]);
    crotch1.getItems().add(crotchtype[i]);

}

crotch.setValue("ANY");
crotch1.setValue("ANY");

//fill pants lower part choice box
for(int i = 0; i < pantslp.length; i++) {

    pantslowerpart.getItems().add(pantslp[i]);
    pantslowerpart1.getItems().add(pantslp[i]);

}

pantlowerpart.setValue("ANY");
pantlowerpart1.setValue("ANY");

// fill polo neck choice box and set default
for(int i = 0; i < necktype.length; i++) {

    poloneck.getItems().add(necktype[i]);
    poloneck1.getItems().add(necktype[i]);

}

poloneck.setValue("ANY");
poloneck1.setValue("ANY");

```

## Δημιουργία panels για το Generation tab

Στην παρακάτω μέθοδο η οποία καλείται για την αρχικοποίηση του πληθυσμού μέσω κλικ στο κουμπί GENERATE, καθαρίζει αρχικά το panel atome που φιλοξενεί τα κελιά με τα σχέδια πιθανόν από την προηγούμενη εξέλιξη κάποιου προτύπου. Στην συνέχεια αν δεν έχει γίνει αρχικοποίηση, δημιουργείται ο κατάλληλος αριθμός panels, choice boxes και απλών int arrays για την δημιουργία του panel των σχεδίων. Ύστερα γίνεται ο έλεγχος για το επιλεγμένο πρότυπο και δημιουργείται ένας πίνακας αντικειμένων. Αν δεν έχουν οριστεί τιμές από τον χρήστη βάζει τις default τιμές με μια try...catch. Στην συνέχεια δημιουργεί κι άλλα panes όπου τους δίνει css class και μια mouseevent να εμφανίζουν τα στατιστικά τους στο stats panel. Για την περίπτωση του προτύπου του παντελονιού δημιουργεί κρυφά panes που εμφανίζονται on hover και δείχνουν το πίσω μέρος τους. Τέλος προσθέτει choice box σε κάθε panel σχεδίου.

```
//creates the panels of the generations and set the number for max generations
public void createGen() {

    atome.getChildren().clear();

    if(initializeflag == false) {

        try {

            len = gennumber.getValue().intValue();
            chb = new ChoiceBox[len];
            ng = new Pane[len];
            ev = new int[len];

            if(filter1.isDisable() == false) {

                nt = new TShirtC[len];

            }else if(filter2.isDisable() == false) {

                ntr = new TrousersC[len];

            }

        }catch(Exception e) {

            len = 10;
            chb = new ChoiceBox[len];
            ng = new Pane[len];
            ev = new int[len];

            if(filter1.isDisable() == false) {

                nt = new TShirtC[len];

            }else if(filter2.isDisable() == false) {

                ntr = new TrousersC[len];

            }

        }

    }

}
```



```

pane.addEventHandler(MouseEvent.MOUSE_EXITED, new EventHandler<MouseEvent>() {

    public void handle(MouseEvent e) {

        chromosomes.setText("");
        selcetedcloth.setText("SELECTED CLOTHING :");

        if(filter1.isDisable() == true) {

            pantsbackside.setVisible(false);
            pantsbackside.getChildren().clear();

        }
    }
});
atome.getChildren().add(pane);
ng[i] = pane;

ChoiceBox<Object> cb = new ChoiceBox<Object>();
cb.setId("ch" + i);
cb.getStyleClass().add("rating");
pane.getChildren().add(cb);
chb[i] = cb;

} catch (Exception e) {

    System.out.println(e);

}
}
}
}

```

## Δημιουργία Αντικειμένων Κατάλληλου Προτύπου

Παρακάτω φαίνονται δύο μέθοδοι ένας για κάθε πρότυπο που παίρνοντας τον κατάλληλο αριθμό που έχει θέσει ο χρήστης δημιουργούν και αρχικοποιούν αντικείμενα.

```

//creates the TShirtC for generation
public void createGenT() {

    try {

        len = gennumber.getValue().intValue();

    } catch (Exception e) {

        len = 10;

    }

    for(int i = 0; i < len; i ++) {

        try {

            TShirtC nb = new TShirtC(null, 0, null, null, 0, null);
            nt[i] = nb;

        } catch (Exception e) {

            System.out.println(e);

        }

    }

}
}
}

```

```

//creates the TrousersC for generation
public void createGenTr() {

    try {

        len = gennumber.getValue().intValue();

    }catch(Exception e) {

        len = 10;

    }

    for(int i = 0; i < len; i ++ ) {

        try {

            TrousersC nb = new TrousersC(null, null, null, 0, null);
            ntr[i] = nb;

        }catch(Exception e) {

            System.out.println(e);

        }

    }

}

```

### Εμφάνιση / Κρύψιμο Stats Panel

Στο tab GENERATION και συγκεκριμένα κάτω από τα κουμπιά ελέγχου υπάρχει ένα κουμπί το οποίο παραπέμπει στην παρακάτω μέθοδο. Πρακτικά εμφανίζει ή κρύβει το stats panel όπως φαίνεται στο σχήμα 32 στο κεφάλαιο πέντε.

```

//min max button for the statistics panel
public void minmaxStatsPane() {

    if(StatsPane.isVisible() == false) {

        StatsPane.setVisible(true);
        sp.setVisible(true);
    }
    else {
        StatsPane.setVisible(false);
        sp.setVisible(false);
    }

}

```

## ΠΑΡΑΡΤΗΜΑ Β : Κώδικας Εξελικτικών Διαδικασιών

Σε αυτό το παράρτημα θα δούμε τον αρχικοποιείται ενός πρότυπου από τα δύο, θα δούμε την σειρά με την οποία εκτελούνται η εξελικτικού τελεστές, το πως αρχικοποιούνται και αναπαρίστανται.

### Constructors

Παρακάτω θα δούμε μόνο έναν από τους δυο constructors των αντικειμένων προτύπων. Πιο συγκεκριμένα για το πρότυπο την μπλούζας. Όπως φαίνεται για την μπλούζα έχουμε τα χαρακτηριστικά της που αποτελούνται από τρεις μονοδιάστατους πίνακες int έναν διδιάστατο πίνακα int και δύο int.

Έπειτα φαίνονται κάποιες από τις set και get μεθόδους, όπως και κάποιες μεθόδους που δίνουν τυχαίες τιμές ανάμεσα σε κάποιο εύρος αριθμών.

```
public class TShirtC {  
  
    private int[] sleevehw;  
    private int[] lowerparthw;  
    private int[] pockets;  
    private int[][] colors;  
    private int lowerpartheight;  
    private int poloneck;  
  
    public TShirtC(int[] i, int j, int[] a, int[] b, int f, int[][] c) {  
  
        this.sleevehw = i;  
        this.lowerpartheight = j;  
        this.lowerparthw = a;  
        this.pockets = b;  
        this.poloneck = f;  
        this.colors = c;  
  
    }  
  
    //get and set for parameters  
    public int[] getLowerparthw() {  
        return lowerparthw;  
    }  
  
    public void setLowerparthw(int[] lowerparthw) {  
        this.lowerparthw = lowerparthw;  
    }  
  
    public int[] getSleevehw() {  
        return sleevehw;  
    }  
  
    public void setSleevehw(int[] sleevehw) {  
        this.sleevehw = sleevehw;  
    }  
}
```

```

//random pockets width
public int Randompocketsw() {

    Random r = new Random();
    int lowphw = 20;
    int highphw = 30;
    int pwidth = r.nextInt(highphw-lowphw) + lowphw;

    return pwidth;

}

// random pockets style
public int Randompockettype() {

    Random a = new Random();
    int pocketstype = a.nextInt(3);

    return pocketstype;

}

```

Για την μέθοδο τυχαίας ανάθεσης χρωμάτων έχουν δημιουργηθεί μια σειρά από μεθόδους αρχικοποίησης που διαφέρουν από τις άλλες. Η πρώτη μέθοδος RandomColor δέχεται σαν παράμετρο το χρώμα προτιμήσεις του χρήστη και μέσω ενός πίνακα με κάποια rgb χρώματα, ανάλογο με το χρώμα αυτό δίνει μια απόχρωση του στην τύχη και έπειτα επιστρέφει το χρώμα στην μορφή rgb μέσω ενός πίνακα colors.

```

public int[] RandomColor( String pcolor) {

    int[][] colorpal = {
        { 255, 102, 102}, { 255, 51, 51}, { 255, 0, 0}, { 204, 0, 0}, { 153, 0, 0},
        { 51, 204, 255}, { 51, 153, 255}, { 0, 0, 255}, { 0, 0, 204}, { 0, 0, 153},
        { 102, 255, 102}, { 0, 255, 51}, { 0, 204, 0}, { 0, 153, 0}, { 0, 102, 0},
        { 255, 255, 204}, { 255, 255, 153}, { 255, 255, 0}, { 255, 204, 0},
        { 255, 153, 0}, { 255, 102, 0}, { 255, 204, 51},
        { 204, 204, 204}, { 153, 153, 153}, { 102, 102, 102}, { 51, 51, 51},
        { 153, 102, 0}, { 102, 51, 0}, { 51, 0, 0},
        { 102, 0, 153},
        { 0, 0, 0},
        { 255, 255, 255}
    };

    int[] colors = new int[3];
    Random r = new Random();

```

```

if( prefcolor == "RED") {

    colors = colorpal[r.nextInt(4)];

}else if( prefcolor == "BLUE") {

    int max = 9;
    int min = 5;
    colors = colorpal[(int) ((Math.random() * ((max - min) + 1)) + min)];

}else if( prefcolor == "GREEN") {

    int max = 14;
    int min = 10;
    colors = colorpal[(int) ((Math.random() * ((max - min) + 1)) + min)];
}

```

Η παραπάνω μέθοδος καλείτε από την μέθοδο RandomColorCombination η οποία με τυχαίο τρόπο δίνει χρώμα σε τέσσερα σημεία στα οποία χωρίζεται μια μπλούζα, δηλαδή δημιουργεί ένα μοτίβο και το χρωματίζει με την RandomColor

```

// random color combination
public int[][] RandomColorCombination(TShirtC sib, String prefcolor){

    int[][] rcc = new int[4][3];
    Random a = new Random();
    int chances = a.nextInt(3);

    //Shirt will have same color everywhere
    switch(chances) {

    case 1:

        int[] pattern = new int[4];

        for(int i = 0; i < 4; i++) {

            pattern[i] = a.nextInt(3);

        }

        int[] color1, color2, color3, color4 = new int[3];

        color1 = sib.RandomColor(prefcolor);
        color2 = sib.RandomColor(prefcolor);
        color3 = sib.RandomColor(prefcolor);
        color4 = sib.RandomColor(prefcolor);
    }
}

```

```

        for(int i = 0; i < 4; i++) {
            switch(pattern[i]) {
                case 0:
                    rcc[i] = color1;
                    break;
                case 1:
                    rcc[i] = color2;
                    break;
                case 2:
                    rcc[i] = color3;
                    break;
                case 3:
                    rcc[i] = color4;
                    break;
            }
        }
        break;
    default:
        Arrays.fill(rcc, sib.RandomColor(prefcolor));
    }
    return rcc;
}

```

---

Έπειτα έχουμε την evaluate, η οποία χρησιμοποιείται από τον τελεστή επιβίωσης και βαθμολογεί κάθε άτομο ανάλογα με το αν έχουν το ζητούμενο χρώμα τον TRENDS.

```

//Evaluate color
public static int evaluate(TShirtC a, String prefcolor) {
    int rating = 0;
    int check = 0;
    int[][] colorpal = {
        { 255, 102, 102}, { 255, 51, 51}, { 255, 0, 0}, { 204, 0, 0}, { 153, 0, 0},
        { 51, 204, 255}, { 51, 153, 255}, { 0, 0, 255}, { 0, 0, 204}, { 0, 0, 153},
        { 102, 255, 102}, { 0, 255, 51}, { 0, 204, 0}, { 0, 153, 0}, { 0, 102, 0},
        { 255, 255, 204}, { 255, 255, 153}, { 255, 255, 0}, { 255, 204, 0},
        { 255, 153, 0}, { 255, 102, 0}, { 255, 204, 51},
        { 204, 204, 204}, { 153, 153, 153}, { 102, 102, 102}, { 51, 51, 51},
        { 153, 102, 0}, { 102, 51, 0}, { 51, 0, 0},
        { 102, 0, 153},
        { 0, 0, 0},
        { 255, 255, 255}
    };
    int[][] arrayColor = a.getColors();
    int[][] acolors = arrayColor;
}

```

```

}else if( pcolor == "BROWN" ) {
    for(int i = 0; i < acolors.length; i++) {
        for(int x = 27; x < 29; x++) {
            for(int j = 0; j < acolors[0].length; j++) {
                if(acolors[i][j] == colorpal[x][j]){
                    check ++;
                }
            }
        }
        if (check == 3) {
            rating ++;
        }
        check = 0;
    }
}

```

Τέλος στους constructors έχουν δημιουργηθεί δύο μέθοδοι. Η μέθοδος atome εξυπηρετεί τον προγραμματιστή να εμφανίσει αναλυτικά τον γονότυπο κάποιου αντικειμένου ατόμου. Η μέθοδος statspane επιστρέφει ένα string με όλα τα στοιχεία του ατόμου.

```

//print atom
public void atome(TShirtC atome, String text) {

    int[] array3 = atome.getSleevehw();
    int[] array1 = atome.getLowerparthw();
    int[] array2 = atome.getPockets();
    int[][] arrayColor = atome.getColors();
    System.out.println(text + ": " + array3[0] + " . " + array3[1] +

}

//statspane atome
public String statspane(TShirtC atome) {

    int[] array3 = atome.getSleevehw();
    int[] array1 = atome.getLowerparthw();
    int[] array2 = atome.getPockets();
    int[][] arrayColor = atome.getColors();
    String string = "chromosome 1(sleves h/w): " + array3[0] + " . "

    return string;
}

```

## Αρχικοποίηση Πληθυσμού

Κατά την αρχικοποίηση η οποία καλείται με το κουμπί GENERATE το OPTIONS TAB απενεργοποιείται, έπειτα με τον έλεγχο του προτύπου που επιλέχθηκε γίνεται η κατάλληλη δημιουργία είτε T Shirt είτε Trouser, επίσης ακολουθεί μια σειρά από disable και visible σε κουμπιά και panel και ο χρήστης μεταφέρεται αυτόματα στο GENERATION TAB, όπου έχουν δημιουργηθεί τα πρώτα σχέδια.

```
//starts the initialisation
public void loadPopulation(ActionEvent event){

    options.setDisable(true);

    if(filter1.isDisable() == false) {

        createGen();
        createGenT();
        cgen ++;
        gennum.setText("GENERATION NUM : " + cgen);

        if(this.initializeflag == false) {

            for(int i = 0; i < len; i++) {

                tshirt.createTshirt(ng[i], nt[i], chflag, colorpal.getValue(), sleevesheight.getValue(),
                    shirtsheight.getValue(), poloneck.getValue(), pocketsch.getValue());
                nt[i].atome(nt[i], "atom " + i);
            }

            this.initializeflag = true;

        }

    }else if(filter2.isDisable() == false){

        createGen();
        createGenTr();
        cgen ++;
        gennum.setText("GENERATION NUM : " + cgen);

        if(this.initializeflag == false) {

            for(int i = 0; i < len; i++) {

                trouser.createTrouser(ng[i], ntr[i], chflag, colorpalT.getValue(), panttype.getValue(),
                    pantslowerpart.getValue(), crotch.getValue(), pocketschT.getValue());
                ntr[i].atome(ntr[i], "atom " + i);
            }

            this.initializeflag = true;

        }

    }

}
```

```

SingleSelectionModel<Tab> selectionModel = tabM.getSelectionModel();

if(generation.isDisabled() == true) {

    mansub.setVisible(false);
    womansub.setVisible(false);
    all.setVisible(false);
    checkB();
    selectionModel.select(0);
    generation.setDisable(false);
    buttons.setDisable(true);
    buttons1.setDisable(true);
    buttons2.setDisable(true);
    help.setText("Generation Board: Activated");
    pb.setProgress(1F);
    pb.setStyle("-fx-accent: #EDB83D;");

} else {

    for(int i = 0; i < len; i++) {

        removeAll(ng[i]);

    }
    selectionModel.select(1);
    generation.setDisable(true);
    buttons.setDisable(false);
    buttons1.setDisable(false);
    buttons2.setDisable(false);
    help.setText("Generation Board: Deactivated");
    pb.setProgress(0F);

}

}

```

### Αναπαράσταση και Αρχικοποίηση

Στην μέθοδο αυτή σκοπός είναι κυρίως η αναπαράσταση του ατόμου που έχει σταλεί αλλά εδώ γίνεται και η αρχικοποίηση του. Δημιουργείται ένας τελεστής double ο οποίος θα πάρει τιμές μεταξύ 0 και 1. Η μέθοδος δέχεται σαν παραμέτρους πέρα από το αντικείμενο και τις προτιμήσεις του χρήστη από το panel Filters σε μορφή string. Αν το flag που δείχνει αν έχει γίνει αρχικοποίηση δείξει false τότε εκτελείται η ανάθεση τιμών χαρακτηριστικών. Οι πιθανότητες να ισχύουν στο σχέδιο οι προτιμήσεις του χρήστη είναι 50%, σε περίπτωση που ισχύουν θα δοθεί ένας ανάλογος αριθμός ή θα πάρει τιμή μέσω της κατάλληλης random μέχρι να ικανοποιήσει μια συνθήκη. Αυτή η διαδικασία επαναλαμβάνεται για κάθε σχέδιο ξεχωριστά. Έπειτα γίνεται ανάθεση των χαρακτηριστικών με set και σε παραμέτρους που θα χρησιμοποιήσει η μέθοδος για την σχεδίαση. Σε περίπτωση που έχει γίνει η αρχικοποίηση απλά εκτελεί τις get και δίνει τις κατάλληλες τιμές στις παραμέτρους του.

Αφού έχει πάρει τις τιμές δηλαδή τα άλληλια των δονιδιών εκτελεί έναν έλεγχο στα χρώματα του σχεδίου για να δώσει το κατάλληλο background color ώστε να κάνει σωστή αντίθεση με αυτά. Ο σχεδιασμός ξεκινάει με την εκτέλεση μιας try...catch χρησιμοποιούνται lines και curves και με την κατάλληλη τροποποίηση τους δίνουν τις κατάλληλες τιμές X και Y για να τοποθετηθούν κατάλληλα. Η σχεδίαση γίνεται από μέρος σε μέρος, δηλαδή πρώτα το σώμα μετά τα μανίκια κ.τ.λ.. Ο χρωματισμός επίσης των σημείων του προτύπου γίνεται με lines. Τέλος τα curve κατά κύριο λόγο δίνουν λεπτομέρεια στο σχήμα.

Επειδή ο κώδικας στο κομμάτι αυτό είναι κοντά στις 700 γραμμές για κάθε πρότυπο δόθηκαν κομμάτια για το πρότυπο της μπλούζας.

```

        int arrayColor[][] = new int[4][3];
        Random r = new Random();
        double chances = r.nextInt();

    if(flag == false) {

        //random number for body lower part
        if(prefsh == "ANY" || prefchances > chances) {

            lpheight = a.Randomlowerpart();

        }else if(prefsh == "SHORT") {

            lpheight = 0;

        }else {

            do {

                lpheight = a.Randomlowerpart();

            }while(lpheight > 0);

        }

        //set values
        int[] array = {resultx, resulty};
        int[] array1 = {slheight, slwidth};
        int[] array2 = {pocketsflag , pocketstype, pheight, pwidth};

        if( prefchances > chances ) {

            arrayColor = a.RandomColorCombination( a, "");

        }else {

            arrayColor = a.RandomColorCombination( a, prefcolor);

        }

        a.setSleevehw(array);
        a.setLowerparthw(array1);
        a.setPockets(array2);
        a.setLowerpartheight(lpheight);
        a.setPoloneck(colar);
        a.setColors(arrayColor);
    }
}

```

```

}else {

    // get sleeves height and width
    int[] array = a.getSleevehw();
    resultx = array[0];
    resulty = array[1];

    endX = 100 - resultx;
    endY = 100 + resultx;

    //body lower part height
    lpheight = a.getLowerpartheight();

    // lower sleeves height and width
    int[] array1 = a.getLowerparthw();
    slheight = array1[0];
    slwidth = array1[1];

    //neck type
    colar = a.getPoloneck();

    //get pockets existence type height and width
    int[] array2 = a.getPockets();
    pocketsflag = array2[0];
    pocketstype = array2[1];
    pheight = array2[2];
    pwidth = array2[3];

    //get colors
    arrayColor = a.getColors();
}

//check for lighter colors and change the background if needed
for(int i = 0; i < 3; i ++) {
    if(Arrays.equals(arrayColor[i], white)
        || Arrays.equals(arrayColor[i], lyellow)
        || Arrays.equals(arrayColor[i], vlyellow)
        || Arrays.equals(arrayColor[i], yellow)
        || Arrays.equals(arrayColor[i], vlgreen)
        || Arrays.equals(arrayColor[i], lgreen)) {

        cpane.setStyle("-fx-background-color: #000000");
    }
}
}

```

```

//if we have no sleeves then
if(resultx >= 20) {

//create left sleeve
Line line = new Line(100, 100, endX, endY);
Line line3 = new Line( resulty + 110, resulty + 110, endX + 10 + resulty, endY + 10 + resulty);
Line line5 = new Line( endX, endY, endX + 10 + resulty, endY + 10 + resulty );
line.setStroke(Color.rgb(arrayColor[1][0], arrayColor[1][1], arrayColor[1][2]));
line3.setStroke(Color.rgb(arrayColor[1][0], arrayColor[1][1], arrayColor[1][2]));
line5.setStroke(Color.rgb(arrayColor[1][0], arrayColor[1][1], arrayColor[1][2]));
cpane.getChildren().add(line);
cpane.getChildren().add(line3);
cpane.getChildren().add(line5);

for(double i = 0; i < resulty + 10;) {

    i = i + 1;

    Line inline = new Line( endX + i, endY + i, 100 + i, 100 + i);
    inline.setStroke(Color.rgb(arrayColor[1][0], arrayColor[1][1], arrayColor[1][2]));
    cpane.getChildren().add(inline);

}

//create right sleeve

Line line2 = new Line( 230, 100, 330 - endX, endY);
Line line4 = new Line( 220 - resulty, 110 + resulty, 320 - endX - resulty, endY + 10 + resulty );
Line line6 = new Line( 330 - endX, endY, 320 - endX - resulty, endY + 10 + resulty );
line2.setStroke(Color.rgb(arrayColor[1][0], arrayColor[1][1], arrayColor[1][2]));
line4.setStroke(Color.rgb(arrayColor[1][0], arrayColor[1][1], arrayColor[1][2]));
line6.setStroke(Color.rgb(arrayColor[1][0], arrayColor[1][1], arrayColor[1][2]));
cpane.getChildren().add(line2);
cpane.getChildren().add(line4);
cpane.getChildren().add(line6);

for(double i = 0; i < resultx;) {

    i = i + 1;

    Line inline = new Line( 330 - endX - i, endY - i, 320 - endX - resulty - i, endY + 10 + resulty - i);
    inline.setStroke(Color.rgb(arrayColor[1][0], arrayColor[1][1], arrayColor[1][2]));
    cpane.getChildren().add(inline);

}
}

```

## Εξελικτική Διαδικασία

Η παρακάτω μέθοδος εκτελείται όταν ο χρήστης επιλέγει και πατήσει το κουμπί Next Generation στο GENERATION tab. Αρχικά μια μεταβλητή cgen που μετράει τις γενιές μεγαλώνει κατά ένα έπειτα μπαίνει σε μια switch case όπου ελέγχει αν η γενιά που ακολουθεί ξεπερνάει τον μέγιστο αριθμό γενεών. Αυτό γίνεται με την διαφορά του mgen και cgen όπου mgen η μέγιστη γενιά. Έπειτα και στις δύο περιπτώσεις ακολουθεί την ίδια διαδικασία με ελάχιστες διαφορές. Παίρνει τη βαθμολογία του χρήστη για κάθε σχέδιο σε ένα πίνακα int και αθροίζει την βαθμολογία για να βρει ύστερα τον μέσο όρο ποιότητας, τον οποίο στο τέλος τοποθετεί στο κατάλληλο text στο stats panel. Στην περίπτωση που η βαθμολογίες είναι μηδενικές θέτει για γονείς τα άτομα στα κελιά 0 και 1. Αλλιώς τα στέλνει στην select όπου γίνεται η επιλογή με roulette wheel. Η μέθοδος της select θα παρουσιαστεί αργότερα. Υστερα αφαιρεί τα παλιά κελιά με την createGen και δημιουργεί καινούργια. Αφού κάνει έλεγχο για να δει ποιο πρότυπο εξελίσει ο αλγόριθμος ορίζει τους γονείς και τους στέλνει στον τελεστή διασταύρωσης crossover όπου παίρνει μια καινούργια γενιά 200 ατόμων η οποία στέλνεται στον τελεστή επιβίωσης και μόνο τόσο όσο είναι ο μέγιστος αριθμός πληθυσμού ανά γενιά θα συνεχίσει για προβολή με της create μεθόδους.

```

//next generation
public void loadNewPopulation(ActionEvent event){

    cgen++;
    gennum.setText("GENERATION NUM : " + cgen);
    int dgen = mgen - cgen;
    float sumfa = 0;

    switch (dgen) {
    case 0:
        try {

            //get and send ratings for evaluation
            for(int i = 0; i < len; i++) {

                ev[i] = (int) chb[i].getValue();
                sumfa = sumfa + (int) chb[i].getValue();

            }

            System.out.println("sumfa :" + sumfa);

            fitnessavg = (fitnessavg + (sumfa/len)) / (cgen - 1);
            fitavg.setText("FITNESS AVG : " + fitnessavg);

            if(sumfa == 0) {

                rating[0] = 0;
                rating[1] = 1;

            }else {

                rating = eva.select(ev);

            }

            help.setText("Generation Board: Evaluation completed");
            pb.setProgress(0.25F);

            //remove old generation
            createGen();
            chflag = false;
            checkB();
            help.setText("Generation Board: Crossover and mutation in the process");
            pb.setProgress(0.75F);

```

```

if(filter1.isDisable() == false) {

    parents[0] = nt[rating[0]];
    parents[1] = nt[rating[1]];

    newgen = cr.CrossoverTshirts(parents[0], parents[1]);

    for(int i = 0; i < len; i ++ ) {

        nt[i] = newgen[i];
        nt[i].atome(nt[i], "sibling" + i);
        tshirt.createTshirt(ng[i], nt[i], initializeflag, colorpal.getValue(), sleeve);
        nt[i].atome(nt[i], "atom " + i);

    }
}else if(filter2.isDisable() == false) {

    parentsT[0] = ntr[rating[0]];
    parentsT[1] = ntr[rating[1]];

    newgenT = cr.CrossoverTrousers(parentsT[0], parentsT[1]);

    for(int i = 0; i < len; i ++ ) {

        ntr[i] = newgenT[i];
        ntr[i].atome(ntr[i], "sibling" + i);
        trouser.createTrousers(ng[i], ntr[i], initializeflag, colorpalT.getValue(), pi);
        nt[i].atome(nt[i], "atom " + i);

    }

}

}catch(Exception e) {

    System.out.println("ProblemOnLoadGen" + e);
    help.setText("Generation Board: Please evaluate the solutions");
    notev.setVisible(true);
}

help.setText("Generation Board: Maximum Generation Number");
pb.setProgress(1.0F);
newGen.setDisable(true);
notevp.setText("PLEASE SELECT FINAL GENERATION TO SAVE THE DESIGNS");
notev.setVisible(true);
break;

```

```

default:
    try {

        sumfa = 0;

        //get and send ratings for evaluation
        for(int i = 0; i < len; i++) {

            ev[i] = (int) chb[i].getValue();
            sumfa = sumfa + (int) chb[i].getValue();
        }

        System.out.println("sumfa :" + sumfa);

        fitnessavg = (fitnessavg + (sumfa/len)) / (cgen - 1);
        fitavg.setText("FITNESS AVG : " + fitnessavg);

        if(sumfa == 0) {

            rating[0] = 0;
            rating[1] = 1;

        }else {

            rating = eva.select(ev);

        }

        help.setText("Generation Board: Evaluation completed");
        pb.setProgress(0.25F);

        //remove old generation
        createGen();
        chflag = false;
        checkB();
        help.setText("Generation Board: Crossover and mutation in the process");
        pb.setProgress(0.75F);
    }
}

```

```

if(filter1.isDisable() == false) {

    parents[0] = nt[rating[0]];
    parents[1] = nt[rating[1]];

    newgen = cr.CrossoverTshirts(parents[0], parents[1]);
    int[] urpass = new int[len];

    urpass = userPrefEvaluationShirt(newgen);

    for(int i = 0; i < len; i++) {

        System.out.println("top rated: " + urpass[i]);
        nt[i] = newgen[urpass[i]];
        nt[i].atome(nt[i], "sibling" + i);
        tshirt.createTshirt(ng[i], nt[i], initializeflag, colorpal.getValue(), sleevesheigh
        nt[i].atome(nt[i], "atom " + i);
    }
}
else if(filter2.isDisable() == false) {

    parentsT[0] = ntr[rating[0]];
    parentsT[1] = ntr[rating[1]];

    newgenT = cr.CrossoverTrousers(parentsT[0], parentsT[1]);
    int[] urpass = new int[len];

    urpass = userPrefEvaluationTrousers(newgenT);

    for(int i = 0; i < len; i++) {

        System.out.println("top rated: " + urpass[i]);
        ntr[i] = newgenT[urpass[i]];
        ntr[i].atome(ntr[i], "sibling" + i);
        trouser.createTrousers(ng[i], ntr[i], initializeflag, colorpalT.getValue(), panttype
        ntr[i].atome(ntr[i], "atom " + i);
    }
}
}
}
catch(Exception e) {

    System.out.println("ProblemOnLoadGen" + e);
    help.setText("Generation Board: Please evaluate the solutions");
    notev.setVisible(true);
}

}

break;

```

## Τελεστής Επιλογής

Ο τελεστής επιλογής δέχεται τις βαθμολογίες του χρήστη και επιλέγει με την γνωστή μέθοδο επιλογής roulette wheel.

```
// mating pool selection
public int[] select(int[] ev) {

    a = 0;

    //sum the fitness of every individual
    for(int i = 0; i < ev.length; i++) {

        sum = sum + ev[i];

    }

    //create a pool with copies equal to the fitness for each individual
    mp = new int[sum];

    try {
        do {

            for(int i = 0; i < ev[a]; i++) {

                mp[sum - 1] = a;
                sum--;

            }

            a++;

        }while(sum > 0);

    }catch(Exception e) {

        System.out.println("EvaluationCatch" + e);

    }

    //pick first parent at random from the pool
    rnd = new Random().nextInt(mp.length - 1);
    parents[0] = mp[rnd];

    //check if all other parents have zero fitness
    if( a > 1 ) {

        //pick one at random
        do {

            rnd = new Random().nextInt(mp.length - 1);
            parents[1] = mp[rnd];

        }while(parents[0] == parents[1]);

    }else {

        //pick one at random
        do {

            parents[1] = new Random().nextInt(ev.length - 1);

        }while(parents[0] == parents[1]);

    }

    //print parents number of cell
    System.out.println("parents: " + parents[0] + "/" + parents[1]);

    return parents;
}
```

## Τελεστής Διασταύρωσης

Ο τελεστής διασταύρωσης πραγματοποιείται με ομοιόμορφη διασταύρωση. Δημιουργεί ένα κατάλληλο πίνακα με μηδέν και ένα και έπειτα δημιουργεί 200 άτομα με χαρακτηριστικά όπου 0 από τον πρώτο γονέα και όπου 1 τον δεύτερο γονέα. Έπειτα τα στέλνει στον τελεστή μετάλλαξης και τέλος πάλι πίσω με την return.

```
public TShirtC[] CrossoverTshirts(TShirtC pb, TShirtC pa) {  
  
    comb1.clear();  
    TShirtC[] newgen = new TShirtC[200];  
    temp1 = null;  
    temp = 0;  
    pb.atome(pb, "parent 1");  
    pa.atome(pa, "parent 0");  
  
    //fill a 2d array with different combinations  
    for(int i = 0; i < 200; i++) {  
  
        int[] comb = new int[6];  
  
        for(int j = 0; j < 6; j++) {  
  
            chances = Math.random();  
  
            if(chances <= 0.5) {  
  
                chances = 0;  
                comb[j] = (int) chances;  
  
            }else {  
  
                chances = 1;  
                comb[j] = (int) chances;  
  
            }  
        }  
  
        comb1.add(comb);  
  
    }  
}
```

```

//create the new generation with the random numbers we received above
for(int i = 0; i < 200; i++) {
    TShirtC nw = new TShirtC(null, 0, null, null, 0, null);
    temp2 = comb1.get(i);

    if(temp2[0] == 0 ) {
        temp1 = pa.getSleevehw();
        nw.setSleevehw(temp1);
    }else if(temp2[0] == 1 ) {
        temp1 = pb.getSleevehw();
        nw.setSleevehw(temp1);
    }
    if(temp2[1] == 0 ) {
        temp = pa.getLowerpartheight();
        nw.setLowerpartheight(temp);
    }else if(temp2[1] == 1 ) {
        temp = pb.getLowerpartheight();
        nw.setLowerpartheight(temp);
    }
    if(temp2[2] == 0 ) {
        temp1 = pa.getLowerparthw();
        nw.setLowerparthw(temp1);
    }else if(temp2[2] == 1 ) {
        temp1 = pb.getLowerparthw();
        nw.setLowerparthw(temp1);
    }
    if(temp2[3] == 0 ) {
        temp1 = pa.getPockets();
        nw.setPockets(temp1);
    }else if(temp2[3] == 1 ) {
        temp1 = pb.getPockets();
        nw.setPockets(temp1);
    }
    if(temp2[4] == 0 ) {
        temp = pa.getPoloneck();
        nw.setPoloneck(temp);
    }else if(temp2[4] == 1 ) {
        temp = pb.getPoloneck();
        nw.setPoloneck(temp);
    }
    if(temp2[5] == 0 ) {
        temp3 = pa.getColors();
        nw.setColors(temp3);
    }else if(temp2[5] == 1 ) {
        temp3 = pb.getColors();
        nw.setColors(temp3);
    }
    newgen[i] = nw;
}

```

```

//send the new generation for mutation

try {

    for(int i = 0; i < newgen.length; i++) {

        newgen[i].atome(newgen[i], "pre mutated newgen " + i + " ");
        TShirtC w = new TShirtC(null, 0, null, null, 0, null);
        w = mt.MutationTshirt(newgen[i]);
        newgen[i] = w;
        newgen[i].atome(newgen[i], "mutated newgen " + i + " ");
        System.out.println("-----");

    }

} catch(Exception e) {

    System.out.println("CrossoverGen" + e);

}

return newgen;

```

### Τελεστής Μετάλλαξης

Ο τελεστής μετάλλαξης δέχεται ένα ένα τα άτομα της καινούργιας γενιάς. Με μια random γίνεται ο έλεγχος εάν θα μεταλλαχθεί. Αν ναι τότε με μια νέα random δίνεται το χαρακτηριστικό που θα υποστεί την μετάλλαξη. Έπειτα το επιστρέφει πίσω.

```

//TROUSERS MUTATION
public TrousersC MutationTrousers(TrousersC mutant) {

    chances = Math.random();

    if(mutationrate >= chances) {

        //mutation will happen randomly between the 5 chromosomes and will
        max = 5;
        min = 1;
        chances1 = (int) ((Math.random() * ((max - min) + 1)) + min);
        System.out.println("Mutation happend in crhromosome: " + chances1);

        if(chances1 == 1) {

            int[] b = new int[2];
            b[0] = mutant.randomPatnsh();
            b[1] = mutant.randomPatnsw();
            mutant.setPantshw(b);

        } else if(chances1 == 2) {

            temp = mutant.randomCrotch();
            mutant.setCrotch(temp);

        } else if(chances1 == 3) {

            int[] b = new int[2];
            b[0] = mutant.randomLowerPantsTypeT();
            b[1] = mutant.randomLowerPantsHeightT();
            mutant.setPantslp(b);

        }

    }

}

```

```

}else if(chances1 == 4) {

    temp = mutant.randomPocketTEX();
    System.out.println("pockets ex mut: " + temp);

    //pockets can only take other random values if they
    if(temp == 0) {

        int[] b = new int[5];
        b[0] = temp;
        b[1] = mutant.randomPocketTType();
        b[2] = mutant.randomPocketTBackType();
        b[3] = mutant.randomLowerPocketsTH();
        b[4] = mutant.randomLowerPocketsTW();
        mutant.setPocketsT(b);

    }else {

        int[] b = {1, 0, 0, 0, 0};
        mutant.setPocketsT(b);

    }

}else if(chances1 == 5) {

    int[][] b = new int[3][3];
    b = mutant.RandomColorCombinationT( mutant, "");
    mutant.setColorsT(b);

}
return mutant;

```

## Τελεστής Επιβίωσης

Ο τελεστής επιβίωσης δέχεται την καινούργια γενιά των 200 ατόμων και την βαθμολογεί σύμφωνα με τις προτιμήσεις του χρήστη. Έπειτα αφού έχει δώσει βαθμολογίες σε όλα τα στέλνει στην `universalSurviveNextGen` όπου τα ταξινομεί και στέλνει πίσω τον κατάλληλο αριθμό που θα επιβιώσουν από την γενιά.

```
public int[] universalSurviveNextGen(int[] gen) {

    int[][] ngen = new int[gen.length][2];
    int[] high = new int[gen.length];
    int cnt = 0;

    for(int i = 0; i < gen.length; i++) {

        ngen[i][1] = i;
        ngen[i][0] = gen[i];

    }

    Arrays.sort(ngen, Comparator.comparingInt(o -> o[0]));

    for(int i = gen.length - 1; i >= 0; i--) {

        if (ngen[i][0] > 0) {

            System.out.println(ngen[i][1] + "rating :" + ngen[i][0]);

        }

        if(cnt < len) {

            high[cnt] = ngen[i][1];
            cnt++;

        }

    }

    return high;
}
```

```

public int[] userPrefEvaluationShirt(TShirtC[] evgen) {

    int colortemp = 0;
    int[] ur = new int[200];
    int[] urpass = new int[len];
    String option1 = poloneck1.getValue();
    String option2 = sleevesheight1.getValue();
    String option3 = shirtsheight1.getValue();
    String option4 = pocketsch1.getValue();
    String option5 = colorpal1.getValue();

    for(int i = 0; i < ur.length; i ++) {

        // check neck users preference
        // "ANY", "REGULAR", "POLO", "CREW", "FUNNEL", "HIGH", "ROLL"
        int poloneck = evgen[i].getPoloneck();

        if(option1 == "REGULAR" && poloneck == 0) {

            ur[i]++;

        }else if(option1 == "POLO" && poloneck == 1) {

            ur[i]++;

        }else if(option1 == "CREW" && poloneck == 2) {

            ur[i]++;

        }else if(option1 == "FUNNEL" && poloneck == 3) {

            ur[i]++;

        }else if(option1 == "HIGH" && poloneck == 4) {

            ur[i]++;

        }else if(option1 == "ROLL" && poloneck == 5) {

            ur[i]++;

        }

    }

}

```

```

// check sleeves user preference
//"ANY", "NO SLEEVES", "SHORT SLEEVES", "LONG SLEEVES", "3/4"
int[] slhw = evgen[i].getSleevehw();

if(option2 == "NO SLEEVES" && slhw[0] == 0) {
    ur[i]++;
}
else if(option2 == "SHORT SLEEVES" && slhw[0] < 60 && slhw[0] > 0) {
    ur[i]++;
}
else if(option2 == "LONG SLEEVES" && slhw[0] > 60) {
    ur[i]++;
}
else if(option2 == "3/4" && slhw[0] == 60) {
    ur[i]++;
}

//check users preference in shirts height
//"LONG", "SHORT"
if(option3 == "SHORT" && evgen[i].getLowerpartheight() == 0) {
    ur[i]++;
}
else if(option3 == "LONG" && evgen[i].getLowerpartheight() > 0) {
    ur[i]++;
}

// check pockets user preference
//"YES", "NO"
int[] pex = evgen[i].getPockets();

if(option4 == "YES" && pex[0] == 0) {
    ur[i]++;
}
else if(option4 == "NO" && pex[0] == 1) {
    ur[i]++;
}

urpass = universalSurviveNextGen(ur);

return urpass;
}

```