

## ΠΕΡΙΕΧΟΜΕΝΑ

<u>ΕΔΑΦΙΑ</u>	<u>ΣΕΛΙΔΑ</u>
1. Πρόλογος : Γενετικοί Αλγόριθμοι.	1
2. Εισαγωγή στους Γενετικούς Αλγορίθμους	2
3. Η έννοια της Βελτιστοποίησης.	3
4. Περιοχές Εφαρμογής των Γενετικών Αλγορίθμων.	4
5. Γλωσσάριο βιολογικών εννοιών.	4

### ΜΕΡΟΣ Ι

1. Δομή των Γενετικών Αλγορίθμων.	7
2. Η Διαδικασία εφαρμογής των Γενετικών Αλγορίθμων ή Γενετικών Εξελκτικών Αλγορίθμων.	7
2.1. Γενετικοί Εξελκτικοί Αλγόριθμοι.	8
2.2. Στόχος της επαναληπτικής διαδικασίας των Γενετικών Αλγορίθμων και το πέραςμα από την μία γενεά στην επόμενη.	8
3. Αναπαράσταση, Αποτίμηση, Καταλληλότητα.	9
3.1. Το Πρόβλημα της Κωδικοποίησης και της Βελτιστοποίησης.	9
3.1.1. Παράμετροι Ακέραιων τιμών.	10
3.1.2. Παράμετροι Πραγματικών τιμών.	11
3.1.3. Παράμετροι Αλφαβητικών τιμών.	12
3.2. Εφαρμογές Κωδικοποίησης των Δεδομένων.	13
3.2.1. Η Δυναμική κωδικοποίηση στους γενετικούς αλγορίθμους.	13
3.2.2. Η Κωδικοποίηση Gray στους γενετικούς αλγορίθμους.	13
3.2.3. Δυναμική Παραμετρική Κωδικοποίηση.	14
3.3. Αποτίμηση των μελών του πληθυσμού & η Συνάρτηση Αποτίμησης.	15
3.4. Καταλληλότητα των μελών του πληθυσμού & η Συνάρτηση Καταλληλότητας.	15
4. Αναπαράσταση του πεδίου αναζήτησης.	16
4.1. Σχήματα στους Γενετικούς Αλγορίθμους.	17
4.2. Ο Χώρος Έρευνας σαν Υπερκύβος.	17
5. Η διαδικασία της επιλογής.	18
5.1. Η Συμπεριφορά της Φύσης.	18
5.2. Η Συμπεριφορά των Γενετικών Αλγορίθμων	19
5.3. Αιτιοκρατικές μέθοδοι.	20
5.4. Στοχαστικές μέθοδοι.	20

5.4.1. Κατάλοιπο Στοχαστικής Δειγματοληψίας.	20
5.4.2. Επιλογή Ανάλογη της Καταλληλότητας.	21
5.4.2.1.	Τε
χνικές Κλιμάκωσης της τιμής Καταλληλότητας.	25
5.4.3. Επιλογή βαθμίδας.	27
5.4.4. Tournament Επιλογή.	28
5.4.5. Ελιτίστικη Επιλογή.	30
6. Τελεστές Ανασυνδυασμού Γονιδίων.	30
6.1. Οι τελεστές της Διασταύρωσης στους Γενετικούς Αλγορίθμους.	31
6.1.1. Διασταύρωση ενός σημείου.	31
6.1.2. Διασταύρωση δύο σημείων.	32
6.1.3. Μη ομοιόμορφη Διασταύρωση.	34
6.1.4. Πιθανότητα Διασταύρωσης & Αντιγραφή.	34
6.2. Ο τελεστής της Μετάλλαξης στους Γενετικούς Αλγορίθμους.	35
6.2.1. Πιθανότητα Μετάλλαξης.	37
7. Κριτήρια τερματισμού των Γενετικών Αλγορίθμων.	37

## ΜΕΡΟΣ II

1. Η βιβλιοθήκη της GALib.	41
1.1. Βασικοί τύποι.	42
1.1.1. Γενετικός Αλγόριθμος.	42
1.1.2. Ορίζοντας μία αναπαράσταση	43
1.1.3. Οι τελεστές του γονιδίου.	44
1.1.4. Το αντικείμενο του πληθυσμού.	45
1.1.5. Αντικειμενικές Διαδικασίες & κλιμάκωση της τιμής Καταλληλότητας.	45
1.2. Ένα Ιεραρχικό Διάγραμμα των κατηγοριών που μας παρέχει η βιβλιοθήκη GALib.	46
1.3. Το Περιβάλλον προγραμματισμού της GALib.	46
1.4. Τροποποιώντας την GALIB.	57
1.4.1. Αναπτύσσοντας την δική μας κατηγορία του Γονιδίου.	57
1.4.2. Συνάρτηση Αρχικοποίησης του Γονιδίου.	58
1.4.3. Συνάρτηση τελεστή Μετάλλαξης.	58
1.4.4. Συνάρτηση τελεστή Διασταύρωσης.	59
1.4.5. Συνάρτηση Σύγκρισης.	60
1.4.6. Συνάρτηση Εξέλιξης.	60
1.4.7. Συνάρτηση Πληθυσμού.	60
1.4.8. Συνάρτηση Αποτίμησης.	60
1.4.9. Συνάρτηση Κλιμάκωσης.	61
1.4.10. Συνάρτηση Επιλογής.	61

1.4.11. Γενετικός Αλγόριθμος.	62
1.4.12. Συνάρτηση Τερματισμού.	63
1.5. Παραδείγματα με την GALIB.	64
1.5.1. Ελαχιστοποίηση της $f(x_1, x_2, x_3) = x_1 * x_1 + x_2 * x_2 + x_3 * x_3$ .	64
1.5.2. Εύρεση της καλύτερης προσεγγίσεως του $\pi$ .	66
1.5.3. Το πρόβλημα του πλανόδιου πωλητή.	67

### ΜΕΡΟΣ III

1. Γενετικός Προγραμματισμός.	70
1.1. Η Μορφή των προγραμμάτων του Γενετικού Προγραμματισμού.	71
1.2. Σύνολα Συναρτήσεων & Τερματικά Σύνολα στον Γενετικό Προγραμματισμό.	71
1.3. Προβλήματα που πρέπει να αντιμετωπίσουμε.	72
1.4. Διασταύρωση στον Γενετικό Προγραμματισμό.	73
1.5. Μετάλλαξη στον Γενετικό Προγραμματισμό.	74
2. Εξελικτικός προγραμματισμός.	75
2.1. Οι βασικές διαφορές μεταξύ Γενετικών Αλγορίθμων και Εξελικτικού Προγραμματισμού.	75
3. Εξελικτικές στρατηγικές.	75
3.1. Οι βασικές ομοιότητες μεταξύ Εξελικτικών Στρατηγικών και Εξελικτικού Προγραμματισμού.	76
3.2. Οι βασικές διαφορές μεταξύ Εξελικτικών Στρατηγικών και Εξελικτικού Προγραμματισμού.	76
Αναφορές.	77

### ΣΕΛΙΔΟΔΕΙΚΤΗΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1 <sup>ο</sup> : Σχηματικό Διάγραμμα στο οποίο απεικονίζεται η διαδικασία εκτέλεσης ενός Γενετικού Αλγορίθμου.	7
Σχήμα 2 <sup>ο</sup> : Ο τροχός της τύχης.	22
Σχήμα 3 <sup>ο</sup> : Απεικόνιση τυπικής γραμμικής κατανομής μορφής $p_i = a * i + b$ , (με το $a < 0$ ). Και : Απεικόνιση τυπικής αρνητικής εκθετικής κατανομής μορφής $p_i = a * \exp(b * i + c)$ .	28
Σχήμα 4 <sup>ο</sup> : Διασταύρωση ενός σημείου.	32
Σχήμα 5 <sup>ο</sup> : Παρουσίαση των χρωμοσωμάτων σε σχήμα δακτυλιδιού.	33
Σχήμα 6 <sup>ο</sup> : Διασταύρωση δύο Σημείων.	33
Σχήμα 7 <sup>ο</sup> : Μετάλλαξη. Μέλος πριν και μετά την εφαρμογή του τελεστή μετάλλαξης.	36
Σχήμα 8 <sup>ο</sup> : Ιεραρχικό Διάγραμμα των κατηγοριών της Galib.	46
Σχήμα 9 <sup>ο</sup> : Διαγράμματα κλιμάκωσης της τιμής καταλληλότητας για γενιές 0 – 200.(Ανα ένα.)	65
Σχήμα 10 <sup>ο</sup> : Διαγράμματα κλιμάκωσης της τιμής καταλληλότητας για γενιές 0 – 1000.	65
Σχήμα 11 <sup>ο</sup> : Διαγράμματα κλιμάκωσης της τιμής καταλληλότητας για τις γενιές από 0 – 300.	67
Σχήμα 12 <sup>ο</sup> : Διαγράμματα κλιμάκωσης της τιμής καταλληλότητας για τις γενιές από 0 – 1000.	67
Σχήμα 13 <sup>ο</sup> : Διαγράμματα κλιμάκωσης της τιμής καταλληλότητας για τις γενιές από 0 – 1000.	68
Σχήμα 14 <sup>ο</sup> : Δεντροειδής απεικόνιση ενός προγράμματος στον Γενετικό Προγραμματισμό.	71
Σχήμα 15 <sup>ο</sup> : Διασταύρωση στον Γενετικό Προγραμματισμό.	73
Σχήμα 16 <sup>ο</sup> : Μετάλλαξη στον Γενετικό Προγραμματισμό.	73

### ΣΕΛΙΔΟΔΕΙΚΤΗΣ ΠΙΝΑΚΩΝ

Πίνακας 1 <sup>ος</sup> : Παράδειγμα στο οποίο φαίνεται, ο τρόπος αναπαράστασης των ψηφίων 0 -7 σε δυαδική και Gray κωδικοποίηση.	14
Πίνακας 2 <sup>ος</sup> : Παραδείγματα Σχημάτων και του πεδίου που καλύπτουν.	17
Πίνακας 3 <sup>ος</sup> : Σύνολο παραμέτρων για το πρόβλημα ελαχιστοποίησης της $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$ .	65
Πίνακας 4 <sup>ος</sup> : Σύνολο παραμέτρων για το πρόβλημα προσέγγισης του $\pi$ .	67
Πίνακας 5 <sup>ος</sup> : Σύνολο παραμέτρων για το πρόβλημα του πλανόδιου πωλητή.	68
Πίνακας 6 <sup>ος</sup> : Παραδείγματα συναρτήσεων του συνόλου $F$ .	72
Πίνακας 7 <sup>ος</sup> : Παραδείγματα τερματικών του συνόλου $T$ .	72

ΑΦΙΕΡΩΜΕΝΟ ΣΤΟΥΣ ΓΟΝΕΙΣ ΜΟΥ.

Μέσα από αυτό το έργο θα ήθελα να δείξω την ευγνωμοσύνη μου για  
την στήριξη που μου παρέχουν τόσα χρόνια.

Κυρίως με την ηθική υποστήριξη που μου παρείχαν και με την υλική  
συγκατάβαση τους, κατάφερε αυτό το έργο να πάρει υπόσταση.



## 1. Πρόλογος : Γενετικοί Αλγόριθμοι.

Για να επιλύσουμε προβλήματα που αντιμετωπίζουμε επανειλημμένα, αναπτύσσουμε διαδικασίες που περιγράφουν τα βήματα που πρέπει να ακολουθήσουμε ούτως ώστε να επιτύχουμε το επιθυμητό αποτέλεσμα. Χρησιμοποιώντας την ίδια διαδικασία μπορούμε, εύκολα και γρήγορα, να φέρνουμε εις πέρας ένα επαναληπτικά εμφανιζόμενο πρόβλημα με συγκεκριμένο ποσοστό επιτυχίας.

Η διαδικασία που ακολουθείται για την επίλυση ενός προβλήματος ονομάζεται αλγόριθμος. Οι αλγόριθμοι αποτελούνται από έναν πεπερασμένο αριθμό βημάτων σε συγκεκριμένη σειρά. Τα βήματα του αλγορίθμου περιγράφουν την ροή των ενεργειών που πρέπει να εκτελέσουμε και υπό την καθοδήγησή τους να φτάσουμε στο στόχο μας.[1]

Το αντικείμενο τριβής της παρούσης εργασίας είναι η παρουσίαση των γενετικών αλγορίθμων και η χρησιμότητά τους στην αντιμετώπιση σύνθετων και χρονοβόρων προβλημάτων. Αρχικά, θα ξεκινήσουμε κάνοντας, μία αναφορά στον εμπνευστή του όρου και την αρχική ιδέα(λίκνο) από την οποία προήλθαν οι γενετικοί αλγόριθμοι.

Η πιο ακριβής ερμηνεία του όρου 'Γενετικοί Αλγόριθμοι', αναφέρεται σε ένα μοντέλο που εισήχθη και διερευνήθηκε από τον John Holland και τους φοιτητές του, το 1975. Το έργο του ονομάζεται *Adaptation in Natural and Artificial Systems (1975)* και περιγράφει τον τρόπο με τον οποίο μία διαδικασία ανάλογη της διαδικασίας εξέλιξης, μπορεί να εφαρμοστεί στην πράξη για να λύσουμε μαθηματικά προβλήματα και προβλήματα βελτιστοποίησης. Ακόμη και σήμερα ένα μεγάλο θεωρητικό κομμάτι των γενετικών αλγορίθμων αναφέρεται στο μοντέλο που εισήγαγε ο Holland.

Κατά μία ευρύτερη ερμηνεία του όρου 'Γενετικοί Αλγόριθμοι', μπορούμε να ισχυριστούμε ότι είναι κάθε μοντέλο αλγορίθμων που βασίζεται σε έναν πληθυσμό. **Ο πληθυσμός αποτελείται από ένα σύνολο σημείων, στα οποία ορίζεται μία συνάρτηση.** Θεωρώντας σαν σημείο αναφοράς έναν αρχικό πληθυσμό επιδιώκουν να δημιουργήσουν νέα μέλη ή ακριβέστερα νέα σημεία δειγμάτων μέσα σ' ένα πεδίο έρευνας. Με αυτό ως σκοπό χρησιμοποιούν την διαδικασία της Επιλογής και τους τελεστές Ανασυνδυασμού Γονιδίων(όπως είναι οι τελεστές της Διασταύρωσης και της Μετάλλαξης) για να προσεγγίσουν νέα μέλη σε επόμενους πληθυσμούς, δηλαδή να προσεγγίσουν νέες, πιθανόν καλύτερες λύσεις μίας συνάρτησης, που δεν βρίσκονται στον υπάρχον πληθυσμό.[2],[6]

Οι μέθοδοι που χρησιμοποιούν οι γενετικοί αλγόριθμοι βασίζονται στις αρχές και στις παρατηρήσεις όπως έχουν γίνει από τους θεμελιωτές των βιολογικών μοντέλων εξέλιξης των ειδών. Για τον λόγο αυτό θα χρησιμοποιήσουμε όρους δανεισμένους από την γενετική.

## 2. Εισαγωγή στους Γενετικούς Αλγορίθμους.

Η γενική ιδέα στην οποία βασίστηκαν και από την οποία είναι κατά κύριο λόγο εμπνευσμένοι οι γενετικοί αλγόριθμοι είναι αυτή της εξέλιξης των βιολογικών ειδών. Αφομοιώνουν τις συμπεριφορές της φύσης, μιμούμενοι τις διεργασίες που ακολουθεί, όπως τις έχουν καταγράψει οι διάφοροι μελετητές της. Το αποτέλεσμα είναι να συμπεριφέρονται κατά τρόπο αληθοφανή, όπως αυτή.

Στα αρχικά στάδια εφαρμογής τους, οι Γενετικοί Αλγόριθμοι χρησιμοποιούνταν για την εύρεση των μεγίστων σημείων μίας συνάρτησης. Αργότερα, εφόσον διαπιστώθηκε η αποτελεσματικότητά τους και καθιερώθηκαν σαν μοντέλα επίλυσης προβλημάτων, η χρήση τους επεκτάθηκε και στην ελαχιστοποίηση διαφόρων συναρτήσεων. Αυτό βέβαια εξαρτάται από την υφή του εκάστοτε προβλήματος. Ουσιαστικά η εύρεση των μεγίστων ή των ελαχίστων μίας συνάρτησης είναι το ένα και το αυτό, αλλά αναφερόμαστε στην πορεία εξέλιξης τους επειδή θέλουμε να τονίσουμε το διευρυνόμενο πεδίο εφαρμογής που έχουν, με την πάροδο του χρόνου.

Οι γενετικοί αλγόριθμοι, όπως έχουν καθιερωθεί σήμερα, είναι μία οικογένεια υπολογιστικών μοντέλων που επηρεάζονται και προσαρμόζονται στις εξελίξεις και στις αλλαγές που συμβαίνουν στο περιβάλλον τους. Ως απώτερο στόχο, έχουν να καταλήξουν στην πλέον κατάλληλη λύση για ένα συγκεκριμένο πρόβλημα, από αυτές που διαθέτουν ή δημιουργούν.

Εφόσον είδαμε περιληπτικά τα στάδια εξέλιξης των γενετικών αλγορίθμων, είναι χρήσιμο σε αυτό το σημείο να γίνει μία σύντομη περιγραφή του τρόπου με τον οποίο λειτουργούν, προκειμένου να λύσουν ένα πρόβλημα. Η περιγραφή σε αυτό το σημείο έπρα από τον εισαγωγικό της χαρακτήρα, έχει και σαν σκοπό να προϊδεάσει κατάλληλα τον αναγνώστη με την ορολογία που θα χρησιμοποιήσουμε κατά την διάρκεια αυτής της μελέτης.

Οι γενετικοί αλγόριθμοι διαχειρίζονται έναν πληθυσμό από ενδεχόμενες λύσεις, τις οποίες αρχικά επιλέγουν τυχαία. Έχοντας σαν αφητηρία το σύνολο των διαθέσιμων λύσεων, εφαρμόζουν την *αρχή της επιβίωσης του πιο ισχυρού*, προκειμένου να παράγουν καλύτερες προσεγγίσεις λύσεων, μέσα από μία εξακολουθητική διαδικασία δημιουργίας καινούριων μελών σε έναν πληθυσμό.

Τα άτομα που συμμετέχουν σε ένα πληθυσμό την εκάστοτε χρονική στιγμή, τα ονομάζουμε *γενιά*. Σε κάθε γενιά ο πληθυσμός προκύπτει από τον προηγούμενο, μέσω της *επιλογής* (επιβίωση του καταλληλότερου ή καλύτερου μέλους του πληθυσμού), και με την εφαρμογή διαφόρων γενετικών τελεστών που σκοπό έχουν να παράγουν νέα μέλη. Γενετικούς τελεστές ονομάζουμε την *μετάλλαξη και την διασταύρωση*, ορμώμενοι από τον χώρο της βιολογικής εξέλιξης. Κατά την διάρκεια εφαρμογής των γενετικών αλγορίθμων μπορούμε να τους χρησιμοποιήσουμε είτε συνδυαστικά είτε μεμονωμένα, ανάλογα με τις απαιτήσεις του προβλήματος.[8]

Η παραπάνω διαδικασία έχει σαν αποτέλεσμα τα νέα μέλη του πληθυσμού της επόμενης γενεάς να είναι πιο εξελιγμένα σε σύγκριση με τους προγόνους τους, καθώς 'κληρονομούν' ορισμένα από τα χαρακτηριστικά τους και κατά κάποιο τρόπο αναπτύσσουν επιπλέον χαρακτηριστικά και ιδιότητες που οι προγενέστεροι τους δεν διέθεταν. Σε καμία περίπτωση δεν θα πρέπει να θεωρήσουμε ως δεδομένο ότι τα νέα μέλη του πληθυσμού, θα είναι καλύτερα από τους προγενέστερους τους για την επίλυση ενός προβλήματος.

Με τον όρο ‘καλύτερος’ στους γενετικούς αλγορίθμους αναφερόμαστε στον βαθμό καταλληλότητας του κάθε γονιδίου ή ενός μέλους του πληθυσμού για την επίλυση του προβλήματος μας. Η καταλληλότητα μεταβάλλεται σύμφωνα με τις απαιτήσεις του προβλήματος που αντιμετωπίζουμε, δηλαδή ένα μέλος που θεωρείται κατάλληλο για να δώσει λύση σε ένα πρόβλημα μπορεί για κάποιο παρόμοιο πρόβλημα να είναι ένα από τα χειρότερα μέλη του πληθυσμού δειγμάτων. [2]

Ορισμένα από τα προβλήματα που χρησιμοποιούμε τους γενετικούς αλγορίθμους μπορούν να επιλυθούν μέσα σε ένα εύλογο χρονικό διάστημα. Δυστυχώς όμως τα περισσότερα προβλήματα, κυρίως αυτά που απαιτούν για την σωστή περιγραφή τους συναρτήσεις με πολλές παραμέτρους, έχουν μεγάλη υπολογιστική πολυπλοκότητα, κάτι που κάνει την εύρεση των βέλτιστων λύσεων αρκετά επίπονη και δύσκολη διαδικασία. Για αυτούς τους λόγους, κρίνεται απαραίτητο στους γενετικούς αλγορίθμους να τερματίζονται μετά από κάποιο συγκεκριμένο χρονικό διάστημα, δίνοντας λύσεις ικανοποιητικής προσέγγισης. Ο βαθμός προσέγγισης της πραγματικής λύσης του προβλήματος είναι τις περισσότερες φορές ανάλογος με τον χρόνο της εκτέλεσης του αλγορίθμου.

### **3. Η έννοια της Βελτιστοποίησης.**

Πολλά από τα προβλήματα που έχουμε να αντιμετωπίσουμε μπορούμε να τα παραστήσουμε ως μία συνάρτηση. Οι παράμετροι και οι περιορισμοί που υπόκειται κάποιο πρόβλημα, μπορούν να υποκατασταθούν από τις παραμέτρους που δέχεται η συνάρτηση που το περιγράφει. Πολλοί από τους σύγχρονους ερευνητές, οι οποίοι ασχολούνται με τους γενετικούς αλγορίθμους, τους χρησιμοποιούν σαν μοντέλα βελτιστοποίησης.

Όταν αναφερόμαστε στον όρο βελτιστοποίηση, εννοούμε την διαδικασία αναζήτησης μίας λύσης για κάποια προβλήματα που εκφράζονται με μία μορφή συνάρτησης,  $f(x)$ . Η αναζήτηση αυτής της λύσης μπορεί να οδηγήσει είτε στην μεγιστοποίηση, είτε στην ελαχιστοποίηση της τιμής της συνάρτησης που περιγράφει το πρόβλημα. Συνήθως, αυτά τα προβλήματα υπόκεινται σε ορισμένους περιορισμούς. Τρεις είναι οι περιορισμοί που διαθέτουν ή εν’ μέρει πληρούν τα προβλήματα βελτιστοποίησης. Αυτοί είναι οι εξής :

1. Ένα σύνολο **αγνώστων ή μεταβλητών** του προβλήματος.
2. Ο **τύπος της συνάρτησης** που θέλουμε να βελτιστοποιήσουμε.
3. Και ένα σύνολο **περιορισμών** που θα πρέπει να πλήρη η επιστρεφόμενη βέλτιστη λύση.

Στα προβλήματα βελτιστοποίησης από τα παραπάνω χαρακτηριστικά δεν είναι απαραίτητο να υπάρχει το σύνολο των περιορισμών που θα πρέπει να πληροί η επιστρεφόμενη λύση. Γενικά, μπορούμε να ορίσουμε ένα πρόβλημα βελτιστοποίησης, ως την διαδικασία εύρεσης των τιμών που πρέπει να διαθέτει το σύνολο των παραμέτρων του προβλήματος μας, ώστε να επιτύχουμε την καλύτερη δυνατή επίλυση του. Η εύρεση αυτής της λύσης μπορεί να προέλθει τροποποιώντας το σύνολο των δεδομένων που δέχεται η συνάρτηση.

Υπάρχουν διάφοροι τύποι προβλημάτων βελτιστοποίησης. Όπως, οι μεταβλητές του προβλήματος μπορεί να λαμβάνουν συνεχείς ή διακριτές τιμές. Η συνάρτηση βελτιστοποίησης που δημιουργούμε μπορεί να είναι συνεχής ή διακριτή. Επίσης, μία συνάρτηση βελτιστοποίησης μπορεί να είναι παραμετρική είτε μη παραμετρική. Τα παραδείγματα που θα χρησιμοποιήσουμε στην παρούσα εργασία είναι παραδείγματα παραμετρικών συναρτήσεων βελτιστοποίησης.

Στις *παραμετρικές συναρτήσεις βελτιστοποίησης* αλλάζουμε λίγο το σύνολο των παραμέτρων που δέχεται η συνάρτηση που περιγράφει το πρόβλημα μας, με συγκεκριμένες μεθόδους (είτε στοχαστικές, είτε αιτιοκρατικές) προκειμένου να φτάσουμε στο αποτέλεσμα που θέλουμε. Οι γενετικοί αλγόριθμοι χρησιμοποιούνται σε τέτοιου είδους προβλήματα. Από την άλλη πλευρά, στις *μη – παραμετρικές συναρτήσεις βελτιστοποίησης* τις λύσεις ενός προβλήματος τις δώσουμε χωρίς να τροποποιούνται το σύνολο των παραμέτρων, αλλά ενεργώντας με τα ήδη υπάρχοντα δεδομένα που διαθέτουμε στον πληθυσμό. Ένα παράδειγμα μη παραμετρικής συνάρτησης βελτιστοποίησης έχουμε όταν στο πρόβλημα μας προσπαθούμε να βρούμε τη λύση μίας συνάρτησης, με την εύρεση των γειτονικών σημείων από αυτά που ήδη διαθέτουμε.

Οι περιορισμοί στα διάφορα προβλήματα βελτιστοποίησης έχουν διαφορετική μορφή και αλληλεπιδράσεις μεταξύ τους, όπως η λύση τους μπορεί να είναι είτε γραμμική είτε μη – γραμμική, ή ακόμη μπορεί να απουσιάζουν οποιοδήποτε περιορισμοί. Αυτός είναι ένας από τους τύπους προβλημάτων στους οποίους πρέπει να επιμείνουμε επειδή αναφερόμαστε σε αυτούς με παραδείγματα που αναπτύσσουμε, κατά την διάρκεια της ανάλυσης μας.

Η λύση μίας συνάρτησης είναι γραμμική όταν η συνάρτηση μας έχει την εξής υποθετική μορφή:

$$\lambda * x_1 + \mu * x_2 < k$$

Στις γραμμικές συναρτήσεις η γραφική παράσταση ορίζεται σαν μία ευθεία γραμμή που χωρίζει τον χώρο των μεταβλητών  $x_1$ ,  $x_2$  σε δύο τμήματα.

Η λύση μίας συνάρτησης είναι μη – γραμμική όταν η συνάρτηση μας έχει την εξής υποθετική μορφή :

$$x_1^2 * x_2^2 < k$$

Δίνοντας τα παραπάνω υποθετικά παραδείγματα γραμμικών και μη συναρτήσεων θέλουμε να κάνουμε την εξής διευκρίνιση, στις γραμμικές συναρτήσεις δεν υπάρχει άμεση συσχέτιση και αλληλεπιδράσεις μεταξύ των μεταβλητών της συνάρτησης μας. Οπότε μπορούμε να μελετήσουμε κάθε μεταβλητή της παραμέτρου ανεξάρτητα από τις υπόλοιπες, χωρίς να λαμβάνουμε υπόψιν μας τυχόν αλληλεπιδράσεις μεταξύ τους. Όμως από την άλλη πλευρά, στις μη – γραμμικές συναρτήσεις ο συσχετισμός και οι αλληλεπιδράσεις που έχουν οι μεταβλητές των παραμέτρων μεταξύ τους, είναι άμεσος και το τελικό αποτέλεσμα της συνάρτησης επηρεάζεται από το σύνολο των τιμών τους.

Όλες αυτές οι παράμετροι συνθέτουν μια ποικιλία στα προβλήματα της βελτιστοποίησης, γεγονός που καθιστά το πεδίο μελέτης τους, πολύ ευρύ. Σήμερα, η βελτιστοποίηση των λύσεων ενός προβλήματος αποτελεί ένα από τα πλέον σημαντικά θέματα έρευνας στον τομέα των εφαρμοσμένων μαθηματικών και της πληροφορικής.[8]

#### **4. Περιοχές Εφαρμογής των Γενετικών Αλγορίθμων.**

Οι γενετικοί αλγόριθμοι είναι μία στοχαστική διαδικασία η οποία διαχειρίζεται ένα πληθυσμό από λύσεις μίας συνάρτησης και ένα σύνολο τελεστών. Εφαρμόζουν το σύνολο των τελεστών πάνω σε έναν πληθυσμό εξελίσσοντας το σύνολο των μελών του συνεχώς, προκειμένου να βελτιώσουν την επίδοση τους σε ένα πρόβλημα. Αυτό είναι το βασικό χαρακτηριστικό που διαθέτουν οι γενετικοί αλγόριθμοι και μας επιβάλλει να τους διαφοροποιούμε από τους υπόλοιπους αλγορίθμους.

Οι λύσεις που προτείνουν οι γενετικοί αλγόριθμοι στα προβλήματα που τους χρησιμοποιούμε είναι προσεγγιστικές, όπως κάθε στοχαστική μέθοδος. Η τελική λύση που θα μας προτείνουν μετά το πέρας

της εκτέλεσης τους, δεν μπορούν να μας εγγυηθούν ότι θα είναι η πλέον βέλτιστη, χωρίς βέβαια να αποκλείεται αυτό το ενδεχόμενο. Αλλά είναι η λύση που προέκυψε μετά από αρκετά στάδια επιβίωσης και εξέλιξης των σημείων του αρχικού πληθυσμού.

Οι περιοχές εφαρμογής των γενετικών αλγορίθμων, εντοπίζονται σε σύνθετα προβλήματα, των οποίων οι λύσεις δεν βρίσκονται εύκολα με την χρήση απλών αλγορίθμων. Ο λόγος για τον οποίο υπάρχει αυτή η διάκριση τους, σε διαφορετικούς τομείς προβλημάτων είναι ότι : οι λύσεις στα προβλήματα που τους απασχολούμε δεν είναι μονοσήμαντη και μονοδιάστατη. Επίσης, την λύση ενός προβλήματος, πρέπει να την εντοπίσουνε και να την αξιολογήσουνε ως την καλύτερη, μέσα από το σύνολο των λύσεων στο πληθυσμού που χειρίζονται.

Τα προβλήματα που χρησιμοποιούνται περισσότερο οι γενετικοί αλγόριθμοι είναι προβλήματα βελτιστοποίησης. Τέτοιου είδους προβλήματα είναι :

1. Το πρόβλημα εύρεσης της ελάχιστης διαδρομής μεταξύ δύο διαφορετικών πόλεων.
2. Το πρόβλημα του βέλτιστου σχεδιασμού ενός αεροσκάφους(μηχανολογία), προκειμένου να μειώσουμε το κόστος κατασκευής του.
3. Επίσης, στην οικονομία, η βελτιστοποίηση μίας συνάρτησης κόστους μπορεί να επιφέρει σημαντικά οφέλη.

Αυτά είναι λίγα μόνο παραδείγματα των προβλημάτων και των λύσεων που προσπαθούμε να δώσουμε με την βοήθεια των γενετικών αλγορίθμων.

Πιο γενικά, οι εφαρμογές στις οποίες χρησιμοποιούνται οι γενετικοί αλγόριθμοι, ανήκουν σε τομείς όπως αυτοί : της ιατρικής, της ρομποτικής, της τεχνητής νοημοσύνης, την χημεία, της οικονομίας και των οικονομικών εφαρμογών, στον έλεγχο και την επεξεργασία σήματος, στην επεξεργασία εικόνας, και άλλοι.[10]

### **5. Γλωσσάριο βιολογικών εννοιών.**

Ορισμένοι από τους όρους που θα βασιστούμε σε αυτή την εργασία και τους έχουν σφετεριστεί οι γενετικοί αλγόριθμοι είναι οι παρακάτω :

*Gene – Γονίδιο* : είναι η βασική φυσική μονάδα κληρονομικότητας, που περιέχεται σε ένα χρωμόσωμα, αποτελείται από ένα μόριο D.N.A.

*Chromosome – Χρωμόσωμα* : νηματόμορφο τμήμα της πυρηνικής ουσίας του κυττάρου, που αποτελείται από μακρές αλυσίδες D.N.A. περιέχει όλες τις απαραίτητες πληροφορίες για την λειτουργία του κυττάρου και διαβιβάζεται από γενιά σε γενιά

*Phenotype – Φαινότυπο* : το σύνολο των χαρακτηριστικών ενός οργανισμού, τα οποία μπορούν να παρατηρηθούν εξωτερικά και προκύπτουν από την αλληλεπίδραση κληρονομικών ιδιοτήτων και του περιβάλλοντος.

*Crossover – Διασταύρωση* : η διαδικασία κατά την οποία έχουμε αμοιβαία ανταλλαγή γενετικού υλικού μεταξύ των συμμετεχόντων μελών.

*Mutation – Μετάλλαξη* : η διαδικασία τροποποίησης της κληρονομικής σύστασης ενός κυττάρου, η οποία έχει σαν αποτέλεσμα την γενετική αλλαγή κυττάρου ή πληθυσμού κυττάρων.

## ΜΕΡΟΣ Ι

Η παρούσα εργασία χωρίζεται σε τρία μέρη, σε κάθε ένα από αυτά υπάρχει μία διαχωριστική σελίδα στην οποία προλογίζουμε τον σκοπό που έχει εκείνο το μέρος της εργασίας. Περιληπτικά: στο πρώτο μέρος της εργασίας μελετήσουμε τους γενετικούς αλγορίθμους και τις μεθόδους που χρησιμοποιούν μέσα από πολλά παραδείγματα. Στο δεύτερο μέρος παρουσιάζουμε την βιβλιοθήκη της GALib, με την βοήθεια της οποίας θα αναπτύξουμε τους δικούς αλγορίθμους προκειμένου να λύσουμε κάποια παραδείγματα. Τέλος, στο τρίτο μέρος παρουσιάζουμε άλλους αλγορίθμους που ανήκουν στην κατηγορία των εξελικτικών αλγορίθμων.

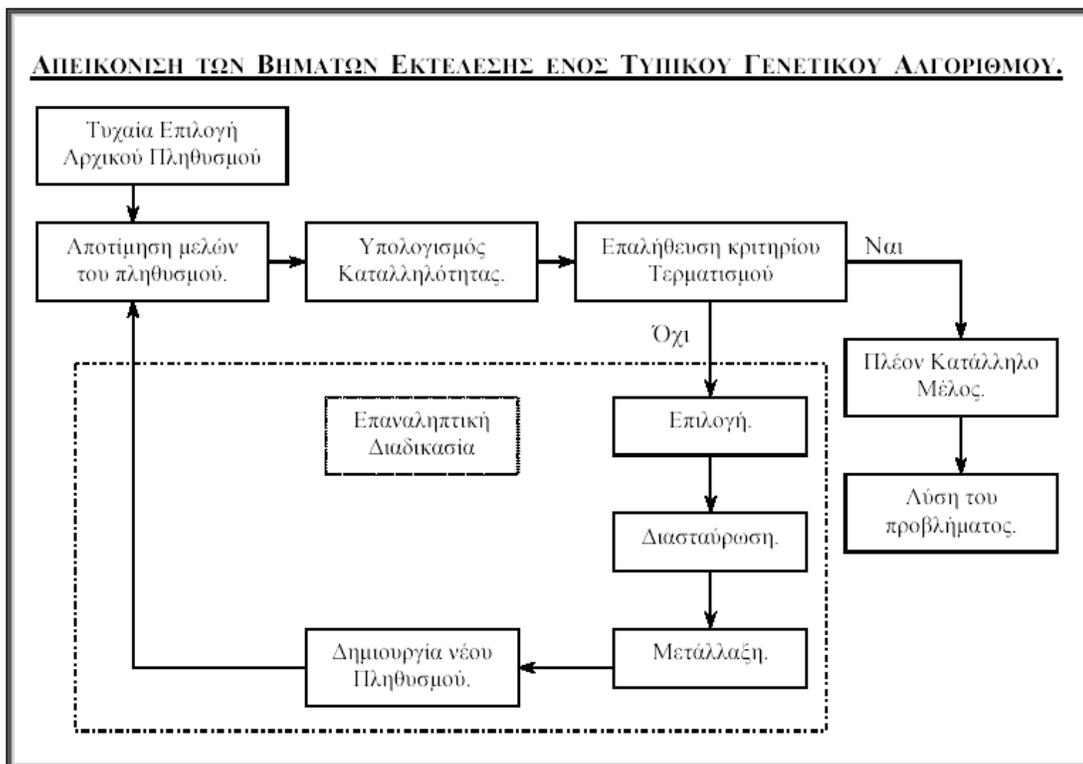
Αρχικά, θα προσεγγίσουμε τους γενετικούς αλγορίθμους από μία καθολική σκοπιά και θα εξετάσουμε τις αρχές στις οποίες βασίζονται, έπειτα θα αρχίσουμε να εμβαθύνουμε σε κάθε λειτουργία τους μία προς μία.

Μελετώντας την φιλοσοφία τους, θα προσδιορίσουμε το γενικό πλάνο που ακολουθείται κατά την διάρκεια της εκτέλεσης τους. Στην συνέχεια, θα εμβαθύνουμε στα μέρη από τα οποία αποτελούνται, με επιμέρους ανάλυση στις μεθόδους που χρησιμοποιούν για να φέρουν εις πέρας τον σκοπό τους. Κατά την διάρκεια της εργασίας θα αναφέρουμε για τροποποιήσεις του αρχικού πλάνου στο οποίο θα βασιστούμε για να εξηγήσουμε τις λειτουργίες τους. Οι τροποποιήσεις που θα αναφέρουμε βρίσκουν άμεση εφαρμογή στην πράξη.

Τα προβλήματα που αναπτύσσουμε είναι πραγματικά. Τα φέρνουμε ως παραδείγματα για να αιτιολογήσουμε και να κατανοήσουμε τις μεθόδους που χρησιμοποιούν οι γενετικοί αλγόριθμοι. Οι λύσεις που προτείνουμε σε αυτά τα προβλήματα δεν είναι ο κανόνας αλλά αποτελούν ενδεικτικές προτάσεις των τακτικών που εφαρμόζουμε στους γενετικούς αλγορίθμους.

## 1. Δομή των Γενετικών Αλγορίθμων.

Ξεκινάμε την ανάλυση μας παραθέτοντας στο παρακάτω σχήμα μία οπτική αναπαράσταση των βημάτων που εκτελούνται από έναν γενετικό αλγόριθμο. Σε αυτό το σχήμα απεικονίζουμε ένα τυπικό πλάνο που συνήθως ακολουθείται από τους γενετικούς αλγορίθμους, κατά την διάρκεια της εκτέλεσης τους. Τα βήματα που ακολουθούν περιγραφικά οι γενετικοί αλγόριθμοι είναι :



Σχήμα 1<sup>ο</sup> : Σχηματικό Διάγραμμα στο οποίο απεικονίζεται η διαδικασία εκτέλεσης ενός Γενετικού Αλγορίθμου.

Σε αυτό το σημείο είναι χρήσιμο να αναφέρουμε ότι το παραπάνω σχήμα απεικονίζει την ροή εκτέλεσης των βημάτων που τυπικά ακολουθούνται από έναν γενετικό αλγόριθμο. Αυτό το τυπικό πλάνο ακολουθείται στην πλειονότητα των περιπτώσεων εφαρμογής τους. Τυχόν τροποποιήσεις και αλλαγές που μπορεί να γίνουν είναι αλλαγές στη σειρά εκτέλεσης των βημάτων που απεικονίζονται ή παράληψη κάποιων από αυτά.

Οι συνηθέστερες αλλαγές που συμβαίνουν, είναι στην επαναληπτική διαδικασία που ακολουθούν οι αλγόριθμοι. Οι αλλαγές εξαρτώνται από το πρόβλημα που έχουμε να αντιμετωπίσουμε. Ορισμένες από τις τροποποιήσεις που μπορεί να υποστεί το παραπάνω σχήμα, θα συζητηθούν στα εδάφια της Διασταύρωσης και της Μετάλλαξης.

## 2. Η διαδικασία εφαρμογής των Γενετικών Αλγορίθμων ή Γενετικών Εξελικτικών Αλγορίθμων.

Μπορεί να ειπωθεί ότι οι Γενετικοί Αλγόριθμοι είναι ένας κλάδος των Εξελικτικών Αλγορίθμων. Οι εξελικτικοί αλγόριθμοι μπορούν να ομαδοποιηθούν στους γενετικούς αλγορίθμους, στον γενετικό προγραμματισμό, στις εξελικτικές στρατηγικές, στον εξελικτικό προγραμματισμό και στα συστήματα ταξινόμησης.

Οι εξελικτικοί αλγόριθμοι είναι διαδικασίες βελτιστοποίησης και έρευνας, εμπνευσμένες από την γενετική και από τις διαδικασίες επιλογής που χρησιμοποιεί η φύση. Η μηχανική των Εξελικτικών Αλγορίθμων έχει σαν στόχο στην βελτιστοποίηση των διαδικασιών, την συνδυαστική βελτιστοποίηση, machine learning, και την επίλυση προβλημάτων.[3]

Περιγραφικά τα στάδια εφαρμογής των εξελικτικών αλγορίθμων όπως θα δούμε αναλυτικότερα στην συνέχεια, είναι τα παρακάτω :

### **2.1. Γενετικοί Εξελικτικοί Αλγόριθμοι.**

1. Αρχικός Πληθυσμός.
2. Αποτίμηση του πληθυσμού.
3. Επαναληπτικά :
  - I. Επιλογή υπό-συνόλου του πληθυσμού για αναπαραγωγή. (*Selection*).
  - II. Ανασυνδυασμός των γενεών των επιλαχόντων γονέων. (*Recombination or Crossover*).
  - III. Μετάλλαξη των μελών του πληθυσμού στοχαστικά. (*Mutation*)
  - IV. Αποτίμηση της τιμής καταλληλότητας του νέου πληθυσμού.
  - V. Επιλογή των μελών που θα επιβιώσουν βάση της πραγματικής τιμής καταλληλότητας.[3]

### **2.2. Στόχος της επαναληπτικής διαδικασίας των Γενετικών Αλγορίθμων και το πέρασμα από την μία γενεά στην επόμενη.**

Στο σχήμα 1 της προηγούμενης σελίδας, φαίνεται ότι η διαδικασία εκτέλεσης των γενετικών αλγορίθμων από ένα σημείο και έπειτα γίνεται επαναληπτική.

Είναι σωστό προτού συνεχίσουμε την ανάλυση μας, να διευκρινίσουμε ότι, αυτή η επαναληπτική διαδικασία είναι αρμόδια για την δημιουργία νέων μελών. Το πέρας ενός κύκλου της επαναληπτικής διαδικασίας καταλήγει στην δημιουργία ενός νέου συνόλου λύσεων σε έναν πληθυσμό. Αυτό το κομμάτι της ροής εκτέλεσης των γενετικών αλγορίθμων είναι ένα πολύ βασικό συστατικό στοιχείο της λειτουργίας τους, καθώς από την αποτελεσματικότητα αυτής της διαδικασίας κρίνεται και η συνολική απόδοση του αλγορίθμου.

Περιληπτικά τα βήματα που ακολουθούμε για να περάσουμε από την μία γενεά στην επόμενη είναι τα ακόλουθα :

1. Fitness – Καταλληλότητα. Υπολογισμός της τιμής καταλληλότητας για όλα τα μέλη του πληθυσμού.
2. Selection – Επιλογή. Με τις διαδικασίες Επιλογής κάνουμε διαλογή των καλύτερων μελών του τρέχοντος πληθυσμού, με την χρήση κάποιου αλγόριθμου. Τα μέλη που επιλέγονται κάθε φορά συμμετέχουν σε έναν ενδιάμεσο πληθυσμό. Ορισμένοι αλγόριθμοι εξετάζονται στο εδάφιο που αναφέρεται στην Επιλογή.
3. Crossover – Διασταύρωση. Όσα μέλη επιλεγούν μέσα από την διαδικασία της επιλογής τα διασταυρώνουμε και παράγουμε νέα μέλη (παιδιά). Οι διασταυρώσεις συμβαίνουν μεταξύ των μελών του ενδιάμεσου πληθυσμού. Τα νέα μέλη αντικαθιστούν, είτε σκοτώνοντας τους χειρότερους του πληθυσμού, είτε σκοτώνουν τυχαία κάποια άτομα, για να πάρουν την θέση τους. Με αυτόν τον τρόπο προκύπτει ο νέος πληθυσμός. Οι διασταυρώσεις προκαλούν μεγάλες τροποποιήσεις στα μέλη και αποφέρουν ποικιλόμορφους πληθυσμούς.

4. Mutation – Μετάλλαξη. Σε αυτή την περίπτωση έχουμε τυχαία επιλογή κάποιων μελών του πληθυσμού, βάση κάποιας στοχαστικής διαδικασίας. Τα νέα μέλη που επιστρέφονται στο πληθυσμό, είναι κοντά σε σχέση με τα προηγούμενα, καθώς οι τελεστές της μετάλλαξης δεν προκαλούν μεγάλες τροποποιήσεις στα μέλη του. Γενικά, οι μέθοδοι αυτοί δεν αποφέρουν ποικιλόμορφους πληθυσμούς.

### **3. Αναπαράσταση, Αποτίμηση, Καταλληλότητα.**

Στα πρωταρχικά στάδια υλοποίησης των γενετικών αλγορίθμων, πάντα τίθεται το ερώτημα της αναπαράστασης ενός πληθυσμού λύσεων που βρίσκονται στον πραγματικό κόσμο, από ένα πεπερασμένο αριθμό συμβόλων που έχουμε την δυνατότητα να χρησιμοποιήσουμε στους ηλεκτρονικούς υπολογιστές. Οι περιορισμοί που επιβάλλονται σε τέτοιες περιπτώσεις είναι γνωστές, επομένως η περαιτέρω αναφορά σε αυτούς θεωρείται άσκοπη. Αξίζει να τονίσουμε ότι ο αριθμός των συμβόλων που θα επιλέξουμε να χρησιμοποιήσουμε, πρέπει να μας εξασφαλίζει την δυνατότητα να διατηρήσουμε το επιθυμητό επίπεδο ακρίβειας που προϋποθέτει το πρόβλημα μας. Μετά από την επιλογή της καταλληλότερης αναπαράστασης, μπορούμε να εκτελέσουμε τις διαδικασίες της αποτίμησης και της αξιολόγησης των λύσεων.

Σε αυτήν την ενότητα θα συζητήσουμε το πρόβλημα της βελτιστοποίησης, φέρνοντας δύο απτά παραδείγματα παραμετρικής βελτιστοποίησης. Σαν συνέχεια θα επεκταθούμε στο πρόβλημα της αναπαράστασης ακεραίων, πραγματικών, και αλφαβητικών χαρακτήρων. Θα σταθούμε στα προβλήματα και τις δυσκολίες με τις οποίες ερχόμαστε αντιμέτωποι και θα εξετάσουμε τις αντίστοιχες λύσεις τους.[2]

#### **3.1. Το Πρόβλημα της Κωδικοποίησης και της Βελτιστοποίησης.**

Προτού προχωρήσουμε σε μία εκτενέστερη περιγραφή των σταδίων που διαδοχικά εκτελούνται από έναν γενετικό αλγόριθμο, καλό θα ήταν να διευκρινίσουμε δύο πολύ σημαντικά προβλήματα, με τα οποία ερχόμαστε αντιμέτωποι και καλούμαστε να δώσουμε μία λύση πριν τους χρησιμοποιήσουμε. Αυτά είναι το πρόβλημα της κωδικοποίησης και αυτό της αποτίμησης των μελών του πληθυσμού.

Ας υποθέσουμε ότι έχουμε ένα πρόβλημα παραμετρικής βελτιστοποίησης, στο οποίο θα πρέπει να ρυθμίσουμε ένα σύνολο μεταβλητών μίας παραμέτρου, προκειμένου να επιτύχουμε την μεγιστοποίηση ή την ελαχιστοποίηση κάποιου στόχου. Για παράδειγμα, στόχος μας σε μία διαδικασία είναι να μεγιστοποιήσουμε το κέρδος σε μία οικονομική συνάρτηση, είτε εναλλακτικά, να ελαχιστοποιήσουμε το κόστος κατασκευής μίας γέφυρας.

Θα μπορούσαμε να δούμε αυτά τα υποθετικά προβλήματα ως ένα μαύρο κουτί, που στις εισόδους του δέχεται, σειρές από *ακολουθίες τιμών ελέγχου*. Οι τιμές αυτές αντιπροσωπεύουν διαφορετικές παραμέτρους του ίδιου προβλήματος, ενώ οι *ακολουθίες τιμών ελέγχου* είναι το σύνολο των μεταβλητών κάθε παραμέτρου. Η μοναδική έξοδος που δεχόμαστε από το μαύρο κουτί είναι μία τιμή, που δηλώνει αν και κατά πόσο καλά ένας συγκεκριμένος συνδυασμός από παραμέτρους λύνουν το πρόβλημα της βελτιστοποίησης. Η τιμή που μας επιστρέφεται είναι το αποτέλεσμα, από την εκτέλεση μίας διαδικασίας αποτίμησης των εισερχομένων τιμών. Ο τελικός μας σκοπός, είναι να ρυθμίσουμε ένα σύνολο από ποικίλες παραμέτρους, ούτως ώστε να βελτιώσουμε την έξοδο που δεχόμαστε από το μαύρο κουτί.

Στην πλειονότητα τους οι λύσεις που έχουν τα προβλήματα παραμετρικής βελτιστοποίησης είναι μη γραμμικές. Υπάρχουν βέβαια και τα προβλήματα που η λύση τους είναι γραμμική. Στην περίπτωση που η λύση ενός προβλήματος είναι μη γραμμική, αυτό δηλώνει, ότι δεν είναι δυνατό να διαπραγματευτούμε

ή να διαχειριστούμε κάθε μία μεταβλητή μίας παραμέτρου του προβλήματος ως μία ανεξάρτητη και αυτόνομη μεταβλητή, κάτι που συμβαίνει σε προβλήματα με γραμμική λύση. Μία μεταβλητή, δεν μπορεί να δώσει λύση σε ένα πρόβλημα, απομονωμένη από τις υπόλοιπες. Αυτό συμβαίνει επειδή, οι διαφορετικές μεταβλητές που συνθέτουν μία παράμετρο του προβλήματος, αλληλεπιδρούν μεταξύ τους για να διαμορφώσουν την τελική λύση.

Προκειμένου να μεγιστοποιήσουμε ή να ελαχιστοποιήσουμε την έξοδο από το μαύρο κουτί θα πρέπει οι μεταβλητές που επηρεάζουν το πρόβλημα, και οι αλληλεπιδράσεις που υπάρχουν μεταξύ τους, να μελετηθούν ενδελεχώς. Τα συνδυαστικά αποτελέσματα που δεχόμαστε στην έξοδο του πρέπει να αξιολογηθούν για να εκτιμήσουμε αν επιτύχαμε τον στόχο που είχαμε θέσει εξ' αρχής.

Στην κοινότητα που ασχολείται με την Βελτιστοποίηση, οι αλληλεπιδράσεις μεταξύ των μεταβλητών μίας παραμέτρου του προβλήματος, ονομάζεται *Epistasis*.

Το δεύτερο πρόβλημα που έχουμε να αντιμετωπίσουμε, είναι αυτό της κωδικοποίησης των δεδομένων εισόδου. Η πρώτη αποδοχή που κάνουμε σχετικά με την αναπαράσταση των χαρακτηριστικών ενός μέλους, είναι οι μεταβλητές που αναπαριστούν παραμέτρους να μπορούν να αντικατασταθούν από μία πλειάδα bit. Αυτή είναι μία σύμβαση που επιβάλλεται από το μέσο εκτέλεσης των Γενετικών Αλγορίθμων, τους ηλεκτρονικούς υπολογιστές.

Αυτό σημαίνει, ότι οι μεταβλητές διακρίνονται και κατατάσσονται βάση μίας φόρμας προτεραιότητας (priori fashion). Το εύρος της διακριτότητας των τιμών αντιστοιχεί σε κάποια δύναμη του 2, αφού οι τιμές των bit είναι 0 ή 1. Το διαθέσιμο εύρος τιμών είναι  $2^N$ . Για παράδειγμα, αν το σύνολο των μεταβλητών το αναπαριστούμε με 10 bit ανά παράμετρο, αμέσως γίνεται αντιληπτό ότι αποκτούμε ένα εύρος 1024 διαφορετικών τιμών για κάθε παράμετρο, δηλαδή  $2^{10}$ . Για να βρούμε το πλήθος των bit που θα χρησιμοποιήσουμε για να κωδικοποιήσουμε τις  $2^N$  διακριτές τιμές, αναπτύσσουμε τον τύπο  $N = \log_2(L)$ . Αν οι παράμετροι είναι πραγματικά συνεχής, τότε η διακριτότητα δεν αποτελεί πρόβλημα. Αυτό φυσικά προϋποθέτει, ότι η διακριτότητα παρέχει επαρκή ανάλυση των δεδομένων που επιθυμούμε να αναπαραστήσουμε, έτσι ώστε να μπορεί και να είναι εφικτό στην έξοδο να λάβουμε το επιθυμητό επίπεδο ακρίβειας στο αποτέλεσμα. Αυτό γίνεται πιο ευκρινές παρακάτω.[2],[3],[4],

### **3.1.1. Παράμετροι Ακεραίων Τιμών.**

Πιο γενικά, όταν πρέπει να αναπαραστήσουμε ακεραίες παραμέτρους στις μεταβλητές μας, έχουμε να κάνουμε τις εξής παρατηρήσεις : Αν μία μεταβλητή παίρνει τιμές από ένα εύρος ακεραίων αριθμών  $p \in \{0, 1, \dots, 2^N - 1\}$ , τότε μπορούμε να κωδικοποιήσουμε αυτή την τιμή χρησιμοποιώντας την δυαδική αναπαράσταση, με N ψηφία. Η τιμή του καθενός από τα N δυαδικά ψηφία είναι 0 ή 1. Το ίδιο θα ισχύει και αν η μεταβλητή μας παίρνει κάποια τιμή από το σύνολο των ακεραίων  $p \in \{M, M+1, \dots, M+2^N - 1\}$ . Τότε μπορούμε να κωδικοποιήσουμε την τιμή της μεταβλητής μας μείον την σταθερή τιμή, M.

Αν κάποια παράμετρος μπορεί να πάρει ένα συγκεκριμένο πεπερασμένο σύνολο τιμών, που το σύνολο τους δεν συμπίπτει με κάποια δύναμη του 2, τότε το θέμα της κωδικοποίησης γίνεται πιο δύσκολο. Για παράδειγμα, αν έχουμε ακριβώς 1200 διακριτές τιμές που μπορούν να αντιστοιχηθούν στη μεταβλητή  $X_i$  (Η μεταβλητή  $X_i$ , εκπροσωπεί ένα από τα χαρακτηριστικά του γονιδίου). Τότε για την μεταβλητή αυτή θα χρειαστούμε τουλάχιστον 11 bit για να καλύψουμε το απαιτούμενο εύρος τιμών. Έτσι όμως,

κωδικοποιούμε 2048 διαφορετικές τιμές, δηλαδή  $2^{11}$ . Τα 848 αχρησιμοποίητα πρότυπα τιμών μπορεί να επηρεάσουν αρνητικά την διαδικασία της αποτίμησης και σαν επακόλουθο να έχουν σε μία πιθανά χειρότερη αποτίμηση, όπως κάποιες ρυθμίσεις των παραμέτρων μπορεί να αντιπροσωπευθούν δύο φορές. Το αποτέλεσμα, είναι ότι όλα τα δυαδικά strings, να εξάγονται από ένα λάθος σύνολο τιμών για τις μεταβλητές του προβλήματος μας.

Σε παρόμοιες περιπτώσεις που η ανώτατη τιμή του πεδίου τιμών, δεν αντιστοιχεί σε κάποια τιμή η οποία να ταυτίζεται με μια τιμή του  $2^N$ , δηλαδή  $L \neq 2^N$ , τότε για να αναπαραστήσουμε όλες τις δυνατές τιμές του απαιτούμενου εύρους πρέπει να επιλέξουμε την τιμή του N μεγαλύτερη κατά ένα τουλάχιστον bit, όπως πράξαμε και στο παραπάνω παράδειγμα, δηλαδή η τιμή του N να δίνεται από την μαθηματική έκφραση  $N = \log_2(L) + 1$ . Το πρόβλημα που συναντάμε με αυτή την μέθοδο, είναι ότι συμβολίζουμε τις τιμές των μεταβλητών από 0 έως  $L - 2$  με μία μοναδική δυαδική τιμή, ενώ την τιμή  $L - 1$  την αντιπροσωπεύουν με  $2^N - L$  διαφορετικές δυαδικές τιμές (Στατιστικό σφάλμα). Ουσιαστικά, οι αρνητικές επιπτώσεις τέτοιων στατιστικών σφαλμάτων, γίνονται αντιληπτές όταν παραχθούν στοιχεία εκτός του αποδεκτού εύρους τιμών, κατά την διάρκεια της εφαρμογής των τελεστών Μετάλλαξης και σε μικρότερη κλίμακα των τελεστών της Διασταύρωσης.

Η λύση που προτείνεται για την αποφυγή τέτοιων δυσμενών περιπτώσεων είναι να κάνουμε συνεχή έλεγχο των τιμών που μπορούν να λάβουν οι μεταβλητές των μελών, βάζοντας κάποιες τιμές κατωφλίου. Έτσι, περιορίζουμε το παραγόμενο εύρος τιμών των νέων μελών του πληθυσμού, εντός κάποιων αποδεκτών ορίων που ορίζονται από το χώρο έρευνας της συνάρτησης μας.[3],[8]

### 3.1.2. Παράμετροι Πραγματικών Τιμών.

Μία ιδιάζουσα περίπτωση συνόλου τιμών είναι όταν οι μεταβλητές μας λαμβάνουν τιμές από ένα σύνολο πραγματικών αριθμών. Οι παράμετροι που δέχονται πραγματικές τιμές μπορούν να αναπαρασταθούν ως σταθεροί αριθμοί ή ακέραιοι διαμέσω της κλιμάκωσης και της ποσοτικοποίησης.

Στη γενική περίπτωση που το πεδίο ορισμού από το οποίο λαμβάνουν τιμές οι παράμετροι των πραγματικών αριθμών είναι, έστω  $p \in [p_{min}, p_{max}]$ , υπάρχει κάποιος περιορισμός όσον αναφορά τον αριθμό που μπορούμε να αναπαραστήσουμε ως τον επόμενο δεκαδικό αριθμό κάποιου δεκαδικού και δίνεται από την μαθηματική έκφραση:  $(2^N - 1) / (p_{max} - p_{min})$ .

Όπως γνωρίζουμε το πλήθος των πραγματικών αριθμών που υπάρχει στο διάστημα  $p \in [p_{min}, p_{max}]$  είναι άπειρο, εξού και ο περιορισμός που θέσαμε παραπάνω. Μπορούμε να φέρουμε ένα παράδειγμα για να γίνει πιο κατανοητός αυτό ο ισχυρισμός. Ας υποθέσουμε ότι η ελάχιστη και η μέγιστη τιμή του επιθυμητού εύρους τιμών είναι  $[-5, 5]$ , για να κωδικοποιήσουμε το σύνολο των τιμών που βρίσκονται σε αυτό το εύρος χρησιμοποιούμε 10 bit, δηλαδή  $2^{10} = 1024$  διακριτές τιμές. Έστω, ότι θέλουμε να αναπαραστήσουμε τον επόμενο δεκαδικό αριθμό του  $-5$ , σύμφωνα με την παραπάνω έκφραση θα είναι :

$$-5 + 10/1024 = -4,990234375,$$

ενώ ο μεθεπόμενος αριθμός που έχουμε την δυνατότητα να αναπαραστήσουμε είναι ο :

$$-5 + 20/1024 = -4,98046875 \text{ κ.ο.κ.}$$

Μπορούμε εύκολα να οδηγηθούμε στο συμπέρασμα ότι οι αριθμοί που έχουμε την δυνατότητα να αναπαραστήσουμε ως τον επόμενο δεκαδικό, αναμφισβήτητα εξαρτάται από το πλήθος των bit που διαθέτουμε για την κωδικοποίηση των μεταβλητών μίας παραμέτρου. Ο αριθμός των bit που θα

χρησιμοποιήσουμε για να κωδικοποιήσουμε τα δεδομένα μας, εξαρτάται από τις απαιτήσεις του προβλήματος που έχουμε να αντιμετωπίσουμε. Στην περίπτωση που απαιτείται από το πρόβλημα μας μεγαλύτερη ακρίβεια θα πρέπει να δεσμεύσουμε περισσότερα bit. Αν στο προηγούμενο παράδειγμα χρησιμοποιούσαμε 11 bit αντί για 10, τότε οι διαθέσιμες διακριτές τιμές θα ανέρχονταν σε 2048, έναντι των 1024. Παρατηρούμε ότι έχουμε διπλασιασμό του συνόλου των διακριτών τιμών, επομένως και διπλασιασμό των αριθμών που αναπαρίστανται με την δέσμευση ενός μόνο επιπλέον bit. Αυτός ο τρόπος επίλυσης πολλές φορές έχει πολύ αρνητικές συνέπειες και επιπτώσεις στην εκτέλεση του γενετικού αλγορίθμου, όπως για παράδειγμα θα έχουμε μεγαλύτερες απαιτήσεις σε μνήμη, θα έχουμε δυσκολότερες και πολυπλοκότερες αναπαραστάσεις προς διαχείριση που θα κάνουν τους τελεστές που χρησιμοποιούμε πολύ πιο αργούς.[3],[4],[5]

### **3.1.3. Παράμετροι Αλφαβητικών Τιμών.**

Στην περίπτωση που το εύρος τιμών των παραμέτρων του προβλήματος μας είναι μη – αριθμητικές, αλλά αλφαβητικές, όπως για παράδειγμα, να χρειαστεί να κωδικοποιήσουμε ονόματα πόλεων {Rome, Milan, Birmingham, London, New York, Paris}, μία απλοϊκή λύση θα ήταν να χρησιμοποιήσουμε τα αρχικά τους, δηλαδή να δημιουργήσουμε ένα σύνολο τιμών της μορφής  $e \in \{R, M, B, L, N, P\}$ .

Μία εναλλακτική λύση αναπαράστασης για τέτοιας φύσης παραμέτρους είναι να χρησιμοποιήσουμε αριθμούς. Όπως θα δείξουμε παρακάτω αυτός είναι καλύτερος τρόπος για να αναπαραστήσουμε αλφαβητικά σύμβολα σε ένα γονίδιο. Η χρήση αριθμών έναντι συμβόλων προϋποθέτει ότι θα κάνουμε μία σύμβαση και θα συνδέσουμε κάθε έναν αριθμό, ακεραίο κατά προτίμηση, με ένα αλφαβητικό σύμβολο ή λέξη.

Μπορούμε να συνεχίσουμε με το παραπάνω παράδειγμα, στο οποίο έχουμε να κωδικοποιήσουμε 6 διαφορετικές πόλεις. Ο ελάχιστος απαιτούμενος αριθμός διακριτών τιμών που μπορούμε να έχουμε, είναι  $2^3 = 8$ . Αυτό το διάστημα τιμών μας παρέχει το απαιτούμενο εύρος, ώστε να κωδικοποιήσουμε τις 6 διαφορετικές πόλεις και να τις συνδέσουμε με κάποιο μοναδικό δυαδικό αριθμό. Παρατηρούμε ότι και εδώ βρισκόμαστε αντιμέτωποι με το πρόβλημα που έχουμε ήδη συναντήσει με τους ακεραίους αριθμούς των πλεοναζόντων προτύπων bit. Ομοίως ενεργούμε και σε αυτή την περίπτωση, δηλαδή θα ορίσουμε κάποια τιμή κατωφλίου για να αποκλείσουμε ενδεχόμενη χρήση των δύο αχρησιμοποίητων πρότυπων τιμών.

Αρχικά ισχυριστήκαμε ότι η χρήση αριθμών έναντι συμβόλων είναι καλύτερη και πιο βολική στην κωδικοποίηση των δεδομένων. Πράγματι κάτι τέτοιο ισχύει, επειδή οι αριθμοί μας επιτρέπουν να κάνουμε αριθμητικές πράξεις στα γονίδια. Ενώ, για την ίδια διαδικασία αν είχαμε επιλέξει την χρήση αλφαβητικών συμβόλων θα έπρεπε να μετατρέψουμε τα αλφαβητικά σύμβολα σε αριθμητικά και μετά από την εκτέλεση της πράξης να τα επαναφέρουμε στην αρχική τους μορφή. Κάτι τέτοιο όμως είναι ασύμφορο και πολύ χρονοβόρο, ιδιαίτερα όταν οι πράξεις αυτές γίνονται επανειλημμένα. Συχνά, μαζί με το θέμα της κωδικοποίησης, δίνεται η διαδικασία της αποτίμησης σαν μέρος της περιγραφής του προβλήματος.

Σαν επίλογο στο πρόβλημα της κωδικοποίησης και της αποτίμησης πρέπει να αναφέρουμε τον παράγοντα χρόνο. Στην εισαγωγή της εργασίας αναφερθήκαμε ακροθιγώς σε έναν παράγοντα που

αποτελεί ένα από τα σημαντικότερα πλεονεκτήματα των γενετικών αλγορίθμων, μέχρι τώρα ειπώθηκαν στοιχεία για πολύ μεγάλης σημασίας διαδικασίες. Πρέπει να τονίσουμε ότι η διαδικασία της αποτίμησης πρέπει να είναι σχετικά γρήγορη, εφόσον είναι μία συνεχώς επαναλαμβανόμενη διαδικασία, επειδή, ο πληθυσμός αντικαθίσταται (ολόκληρος ή κατά ένα μέρος) σε μία γενεσιολογική βάση, καθώς τα μέλη του πληθυσμού αναπαράγονται και εξελίσσονται συνεχώς, οι απόγονοι πρέπει να αποτιμηθούν και να αξιολογηθούν σε κάθε γενιά για να συνεχίσει ο αλγόριθμος να εκτελείται. [3],[4],[5]

### 3.2. Εφαρμογές Κωδικοποίησης των δεδομένων.

Πέρα από το θεωρητικό κομμάτι των τύπων δεδομένων που μπορεί να συναντήσουμε και θα χρειαστεί να κωδικοποιήσουμε σε ένα γονίδιο, πρέπει να εξετάσουμε μερικές πρακτικές μεθόδους που μπορούμε να χρησιμοποιήσουμε.

Σε αυτήν την παράγραφο περιγράφουμε κάποιες από τις μεθόδους που χρησιμοποιούνται στους γενετικούς αλγορίθμους για την επίλυση του προβλήματος της αναπαράστασης των ακεραίων και των πραγματικών αριθμών. Συμπεριλαμβάνουμε κάποια παραδείγματα για την καλύτερη κατανόηση τους:

#### 3.2.1. Η Δυαδική Κωδικοποίηση στους Γενετικούς Αλγορίθμους

Η πιο απλή μορφή κωδικοποίησης για την αναπαράσταση των δεδομένων που μπορούμε να χρησιμοποιήσουμε στους γενετικούς αλγορίθμους, είναι η δυαδική κωδικοποίηση.

Η δυαδική κωδικοποίηση βασίζεται στο δυαδικό αριθμητικό σύστημα. Ο τρόπος με τον οποίο λειτουργεί φαίνεται στην δεύτερη στήλη του πίνακα 1. Οι αριθμοί που βρίσκονται στην αριστερή στήλη προκύπτουν από την μαθηματική έκφραση :

$$2^N * \beta + \dots + 2^1 * \beta + 2^0 * \beta$$

όπου  $\beta$  είναι η τιμή που έχει το bit στην  $n$ -οστή θέση. Για παράδειγμα, ο αριθμός 5 προκύπτει αν :

$$2^2 * 1 + 2^1 * 0 + 2^0 * 1 = 5$$

Η δυαδική κωδικοποίηση είναι εύκολη στην χρήση της, καθώς επίσης είναι η πιο γνωστή μέθοδος εξαιτίας της χρήσης της από τους ηλεκτρονικούς υπολογιστές, αυτό είναι και το μεγάλο πλεονέκτημα που διαθέτει σαν μέθοδος έναντι των υπολοίπων μεθόδων.

Όμως, σαν μέθοδος κωδικοποίησης δεδομένων καθίσταται σχεδόν ακατάλληλη για τους γενετικούς αλγορίθμους, επειδή προκαλεί μεγάλες αλλαγές στην αναπαράσταση των μελών για αριθμούς που βρίσκονται σε σειρά  $\{1, 2, \dots, 2^{N-1}\}$ . [5][6]

#### 3.2.2. Η κωδικοποίηση Gray στους Γενετικούς Αλγορίθμους.

Εναλλακτικά της δυαδικής αναπαράστασης των ακεραίων παραμέτρων στους γενετικούς αλγορίθμους συνήθως χρησιμοποιείται, η κωδικοποίηση Gray.

Μία κωδικοποίηση της μορφής Gray αναπαριστά κάθε ακεραίο αριθμό μίας ακολουθίας  $0, 1, \dots, 2^{N-1}$  σαν μία πλειάδα από bit, μήκους  $L$ . Η αναπαράσταση διαδοχικών ακεραίων αριθμών γίνεται με διαφορά ενός μόνο bit. Γενικά στους γενετικούς αλγορίθμους επιδιώκουμε να έχουμε όσο το δυνατόν λιγότερες αλλαγές και τροποποιήσεις στην αναπαράσταση των μέλη του πληθυσμού.

Υπάρχουν πολλές διαφορετικές εκδοχές της κωδικοποίησης Gray. Ένας τυπικός αλγόριθμος δυαδικής – αναπαράστασης Gray κωδικοποίησης : ξεκινάει με όλα τα bit της πρώτης πλειάδας να είναι ίσα με 0. Στην συνέχεια, για διαδοχικούς αριθμούς μετακινεί το δεξιότερο bit της πλειάδας αναπαράστασης προς τα αριστερά. Με αυτό τον τρόπο, δημιουργούμε μία καινούργια μονοσήμαντη δυαδική πλειάδα

κωδικοποίησης, αλλάζοντας μόνο ένα ή το πολύ δύο bit την φορά. Επαναλαμβάνουμε αυτή τη διαδικασία για όλο το εύρος των αριθμών που θέλουμε να καλύψουμε.

Ένα παράδειγμα φαίνεται στον πίνακα 1, όπου αναπαριστούμε τον τρόπο που περιγράψαμε πιο πάνω, κωδικοποιώντας τους ακεραίους αριθμούς από 0 – 7. Η κωδικοποίηση Gray φαίνεται στην τρίτη στήλη του πίνακα.

<i>Ακέραιοι</i>	<i>Δυαδικοί</i>	<i>Gray</i>
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Πίνακας 1<sup>ος</sup> : Τρόπος αναπαράστασης των ψηφίων 0 -7 σε δυαδική κωδικοποίηση και Gray κωδικοποίηση.

Η κωδικοποίηση Gray είναι ένας πολύ εύκολος τρόπος, κωδικοποίησης ακεραίων παραμέτρων, επίσης είναι παρόμοιος με την δυαδική κωδικοποίηση αυτά είναι ορισμένα από τα πλεονεκτήματα της μεθόδου. Το πλεονέκτημα που συγκεντρώνει αυτή η μέθοδος και την κάνει να ξεχωρίζει από την δυαδική κωδικοποίηση είναι ότι για αριθμούς που βρίσκονται σε σειρά έχουμε πολύ μικρότερης κλίμακας αλλαγές στην αναπαράσταση των μελών. Αυτό άλλωστε μπορούμε να το παρατηρήσουμε αν συγκρίνουμε τις στήλες 1 και 2 του πίνακα 1. Ένα σοβαρό μειονέκτημα της μεθόδου αυτής είναι ότι οι σπάνιες μεγάλες αλλαγές τις, είναι μεγαλύτερες από αυτές της δυαδικής κωδικοποίησης.[2],[3],[5]

### **3.2.3. Δυναμική Παραμετρική Κωδικοποίηση.**

Η Δυναμική Παραμετρική Κωδικοποίηση(ΔΠΚ) είναι μία μέθοδος που χρησιμοποιείται κυρίως για την αναπαράσταση παραμέτρων που λαμβάνουν τιμές από ένα σύνολο πραγματικών αριθμών. Αυτή η μέθοδος κωδικοποίησης επιλέγεται για τους πραγματικούς αριθμούς επειδή, έχει πολύ καλό λόγο Ακρίβειας προς Ταχύτητα, για την διαχείριση δεκαδικών αριθμών.

Στο χρονικό διάστημα που απαιτείται για την εκτέλεση της Δυναμικής Παραμετρικής Κωδικοποίησης γίνεται κάτι αντίστοιχο με τα ακόλουθα βήματα:

1. Αρχικά, η ΔΠΚ εκτελεί ένα πείραμα με μικρό αριθμό επαναλήψεων ή γενεών. Με αυτή την λειτουργία η μέθοδος αποσκοπεί στο να αποκτήσει σχετικά γρήγορα ένα αποτέλεσμα, χαμηλής – ακρίβειας. Έτσι μπορεί να ‘προσανατολίσει’ και να επικεντρώσει την έρευνα της σε κάποια υποπεριοχή του ολικού πεδίου έρευνας.
2. Στο επόμενο βήμα : Περιορίζει την έρευνα στην περιοχή που προηγουμένως έχει αναγνωρίσει ως πιθανή περιοχή όπου μπορεί να βρίσκεται η λύση. Έπειτα, εκτελεί τον γενετικό αλγόριθμο για μικρό αριθμό γενεών, έτσι αυξάνει την ακρίβεια του αποτελέσματος, καθώς επιλέγει συνεχώς μικρότερες προς έρευνα περιοχές.
3. Επαναλαμβάνει το 2<sup>ο</sup> βήμα έως ότου η ακρίβεια που παρέχει, να είναι ικανοποιητική.

Η στρατηγική που ακολουθείται από την μέθοδο ΔΠΚ, είναι να συγκεντρώνει τα στατιστικά στοιχεία των συνθηκών σύγκλισης, των δύο πρώτων bit της κάθε παραμέτρου. Με αυτό τον τρόπο υλοποιεί έναν

τελεστή εστίασης. Έτσι απομονώνει σε συνεχώς μικρότερες περιοχές την έρευνα της. Οπότε, κάθε φορά που ο πληθυσμός συγκλίνει σε μία υποπεριοχή του εύρους του χώρου έρευνας, για κάθε μία παράμετρο. Η ΔΠΚ, με τον τελεστή εστίασης μεταβάλλει την ερμηνεία των γενεών σε ερώτηση για να διπλασιάσει την ακρίβεια της έρευνας. Η έρευνα, διαδοχικά εκτελούμενη, περιορίζεται σε μικρότερα υποδιαστήματα πιο συγκεκριμενοποιημένα. Ο τελεστής εστίασης πρέπει να είναι αόρατος, δηλαδή θα πρέπει να εφαρμόζεται διακριτικά για να αποφύγουμε περιπτώσεις πρόωρου συνθήκης σύγκλισης.[5][8]

### **3.3. Αποτίμηση των μελών του πληθυσμού & η Συνάρτηση Αποτίμησης.**

Όπως έχουμε ήδη προαναφέρει οι γενετικοί αλγόριθμοι βασίζονται σε έναν σύνολο πιθανών λύσεων για ένα συγκεκριμένο πρόβλημα. Το σύνολο αυτών των πιθανών λύσεων ονομάζεται *πληθυσμός*.

Πολλές φορές ένα άτομο του πληθυσμού αναφέρεται σαν χρωμόσωμα. Όπως κάθε χρωμόσωμα φέρει κάποια χαρακτηριστικά και ιδιότητες που το κάνουν να ξεχωρίζει, ομοίως κάθε άτομο στο πληθυσμό έχει τα δικά του μοναδικά στοιχεία και πληροφορίες που το κάνουν ιδιαίτερο. Με τους όρους 'άτομο' ή 'μέλος' του πληθυσμού ονομάζουμε κάθε δομή δεδομένων που χρησιμοποιούμε για να αποθηκεύσουμε τις πληροφορίες των μεταβλητών μίας παραμέτρου. Κάθε άτομο στους γενετικούς αλγόριθμους θεωρείται σαν μία παράμετρος του προβλήματος και αντιπροσωπεύει ένα σύνολο από μεταβλητές. Η λύση του προβλήματος μας θα είναι το σύνολο των μεταβλητών που ομαδοποιούνται σε ένα άτομο του πληθυσμού. Οι γενετικοί αλγόριθμοι κωδικοποιούν μία πιθανή λύση για ένα πρόβλημα στην δομή δεδομένων ενός ατόμου και εφαρμόζουν τους τελεστές Ανασυνδυασμού Γονιδίων(τους τελεστές της Διασταύρωσης και της Μετάλλαξης) σε αυτές, έτσι ώστε να διατηρηθούν οι κρίσιμες πληροφορίες και τα χαρακτηριστικά τους κατά την διάρκεια της εξέλιξης τους. Κάθε μέλος του πληθυσμού θα αναπαρίσταται από μία πλειάδα δυαδικών ψηφίων μήκους  $L(\text{string})$ . Ο προσδιορισμός του μήκους  $L$ , με κάποια τιμή εμπίπτει στο πρόβλημα της κωδικοποίησης.

Μέσα από την διαδικασία της αποτίμησης αξιολογούμε τα χαρακτηριστικά του κάθε μέλους του πληθυσμού. Με την διαδικασία της αποτίμησης, εκτιμώνται οι δομές που έχουν παραχθεί και προσδιορίζονται οι ευκαιρίες αναπαραγωγής που έχει ένα άτομο. Κατά την διάρκεια της διαδικασίας αποτίμησης, προσπαθούμε έχοντας κάποιους κοινούς και ομότιμους κανόνες και αρχές, να χαρακτηρίσουμε όλα τα μέλη του πληθυσμού ομοιόμορφα και ισάξια. Κατά την διαδικασία αυτή θα πρέπει να τηρούνται οι κανόνες για όλα τα μέλη του πληθυσμού προκειμένου να προκύψει ένας ομοιόμορφος χαρακτηρισμός για όλες τις μονάδες του πληθυσμού, δηλαδή απαιτείται ένας ενιαίος τρόπος αποτίμησης. Ο χαρακτηρισμός ενός μέλους του πληθυσμού δεν είναι μία απλή, μονοσήμαντη διαδικασία. Η διαδικασία της αποτίμησης των μελών του πληθυσμού εκτελείτε κάθε φορά που έχουμε την δημιουργία μίας νέας γενιάς. Η συνάρτηση αποτίμησης παρέχει το μέτρο απόδοσης για το σύνολο των χαρακτηριστικών, που συγκεντρώνει ένα μέλος του πληθυσμού στο γονίδιο του. Η αποτίμηση των μελών του πληθυσμού είναι ανεξάρτητη από την αποτίμηση των υπολοίπων μελών.[2],[3]

### **3.4. Καταλληλότητα των μελών του πληθυσμού & η Συνάρτηση Καταλληλότητας.**

Η τιμή καταλληλότητας προσδιορίζει το μέτρο καταλληλότητας ενός μέλους του πληθυσμού για την επίλυση ενός προβλήματος. Το μέτρο καταλληλότητας το παίρνουμε από την εκτέλεση της συνάρτησης καταλληλότητας για κάθε μέλος του πληθυσμού. Ουσιαστικά, η συνάρτηση καταλληλότητας μετατρέπει το μέτρο απόδοσης, που λαμβάνει από την συνάρτηση Αποτίμησης, σε μία κατανομή από ευκαιρίες

αναπαραγωγής. Η τιμή καταλληλότητας ορίζεται λαμβάνοντας υπόψιν της όλα τα υπόλοιπα μέλη του τρέχοντος πληθυσμού. Η βασική διαφορά των δύο συναρτήσεων είναι ότι η συνάρτηση Αποτίμησης εκτιμά κάθε μέλος ως αυτόνομη οντότητα σε έναν πληθυσμό, βασιζόμενη στο σύνολο των χαρακτηριστικών που συγκεντρώνει κάθε μέλος. Η συνάρτηση Καταλληλότητας εκτιμά τα χαρακτηριστικά κάθε μέλους προς αυτά του συνολικού πληθυσμού. Παρακάτω μπορούμε να δούμε ορισμένα παραδείγματα συναρτήσεων που μας δίνουν κάποιο μέτρο καταλληλότητας.

Είναι πολύ πιθανό να συναντήσουμε περιπτώσεις που η διαδικασία στόχος είναι μη στατική και η μορφή της συνάρτησης καταλληλότητας αλλάζει με το χρόνο, κάτι που είναι δυνατό και που λαμβάνει χώρα στα πραγματικά βιολογικά συστήματα. Σε τέτοιες διαδικασίες υπάρχει ανάγκη για ποικιλία στην γενετική. Ένα πολύ καλό παράδειγμα όταν οι γενετικοί αλγόριθμοι εφαρμόζονται σε προβλήματα ολικής ελαχιστοποίησης, είναι να υπολογίζουμε την τιμή καταλληλότητας για κάθε μέλος του πληθυσμού δίνεται από τον τύπο :

$$\text{Τιμή καταλληλότητας για το μέλος } i = (f_{MAX} - f(x)) / (f_{MAX} - f_{MIN}). \quad \text{αν } f_{MAX} \neq f_{MIN}$$

$f_{MAX}$  : η μέγιστη τιμή της συνάρτησης των μελών του πληθυσμού.

$f_{MIN}$  : η ελάχιστη τιμή της συνάρτησης των μελών του πληθυσμού.

$f(x)$  : η τιμή της συνάρτησης για το μέλος  $x$  του πληθυσμού.

Σύμφωνα με τον παραπάνω τύπο τα μέλη εκείνα για τα οποία η τιμή της συνάρτησης είναι πολύ μικρή, έχουν μεγάλη τιμή καταλληλότητας κι αντίστροφα. Επίσης, όταν η μέγιστη και η ελάχιστη τιμή της συνάρτησης είναι ίσες, όλα τα μέλη λαμβάνουν την ίδια τιμή καταλληλότητας.

Ένα άλλο παράδειγμα για τον υπολογισμό της τιμής καταλληλότητας είναι όταν : Η τιμή καταλληλότητας κάθε μέλους του πληθυσμού, ορίζεται ως ο λόγος της αποτίμησης του κάθε μέλους, προς τον μέσο όρο αποτίμησης του τρέχοντος πληθυσμού. Δηλαδή  $f_i/f_{average}$  όπου  $f_i$  είναι η τιμή αποτίμησης που έχουμε υπολογιστεί για το μέλος  $i$ , ενώ ως  $f_{average}$  αναπαριστούμε τον μέσο όρο αποτίμησης όλων των μελών του πληθυσμού μας.

Παρακάτω στις μεθόδους Επιλογής θα συναντήσουμε ένα πρόβλημα που σχετίζεται με την τιμή της συνάρτησης Καταλληλότητας. Αυτό το πρόβλημα αναφέρεται στην πρόωρη συνθήκη σύγκλισης στους γενετικούς αλγορίθμους. Για να το αντιμετωπίσουμε είναι ανάγκη να χρησιμοποιήσουμε κάποια φόρμα για την κλιμάκωση της τιμής της συνάρτησης καταλληλότητας για τον πληθυσμό μας. Σε αυτό το πρόβλημα όσο ο μέσος όρος της αποτίμησης του πληθυσμού αυξάνει, η καταλληλότητα στον πληθυσμό μειώνεται. Ίσως μετά από αρκετές γενιές να υπάρξει μικρή διαφορά μεταξύ του καλύτερου και του χειρότερου ατόμου του πληθυσμού, και ο εξαναγκασμός για επιλογή βασιζόμενου στην τιμή της συνάρτησης καταλληλότητας μειώνεται σχετικά. [2],[3],[5]

#### **4. Αναπαράσταση του πεδίου αναζήτησης.**

Το πεδίο αναζήτησης ορίζεται ως το πεδίο ορισμού των παραμέτρων του προβλήματος. Το πεδίο αναζήτησης μπορεί σε κάποια προβλήματα να είναι πεπερασμένο ή άπειρο, γεγονός που επηρεάζει την υπολογιστική πολυπλοκότητα των αλγορίθμων αναζήτησης. Για παράδειγμα στο γνωστό πρόβλημα του *πλανόδιου πωλητή*, το πεδίο αναζήτησης είναι ένα σύνολο διακριτών τιμών κι αποτελείται από το πλήθος των πόλεων. Σε προβλήματα βελτιστοποίησης συνεχών συναρτήσεων, το πεδίο αναζήτησης μπορεί να είναι είτε άπειρο, είτε πεπερασμένο. [2][5][6]

#### 4.1. Σχήματα στους Γενετικούς Αλγορίθμους.

Ένα σχήμα ή πρότυπο ομοιότητας είναι ένα string από σύμβολα παρμένα από το αλφάβητο  $\{0,1,*\}$ . Ο χαρακτήρας ‘\*’ ονομάζεται wild card, η τιμή του οποίου μπορεί να είναι οποιαδήποτε, από το εύρος τιμών που ορίσαμε πιο πάνω, επομένως, ένα σχήμα μπορεί να αναπαραστήσει πολλά string από bit.

Ο αριθμός των μη ‘\*’ συμβόλων καλείτε τάξη – ‘order’ του σχήματος. Κάποια παραδείγματα φαίνονται στον παρακάτω πίνακα :

Σχήματα	Τάξη	Πλειάδες Αναπαράστασης
***	0	000, 001, 010, 011, 100, 101, 110, 111
*1*	1	010 011 110 111
*10	2	010 110
101	3	101

Πίνακας 2<sup>05</sup> : Παραδείγματα Σχημάτων και του πεδίου που καλύπτουν.

Μία πλειάδα από bit ταιριάζει σε ένα συγκεκριμένο σχήμα αν το string, που θέλουμε μπορεί να κατασκευαστεί από το σχήμα, αντικαθιστώντας το σύμβολο ‘\*’ με την κατάλληλη τιμή του bit.

Γενικότερα, όλα τα μέλη τα οποία ταιριάζουν σε ένα συγκεκριμένο σχήμα περιλαμβάνονται μέσα στην υπερεπίπεδη κατανομή που αντιπροσωπεύεται από το συγκεκριμένο σχήμα. Κάθε δυαδική κωδικοποίηση είναι ένα μέλος του πληθυσμού ή “χρωμόσωμα” το οποίο αντιστοιχεί στη γωνία ενός υπερκύβου και είναι ένα μέλος από τα  $2^L - 1$  διαφορετικά υπερεπίπεδα του υπερκύβου, όπου L είναι το μήκος της δυαδικής κωδικοποίησης που χρησιμοποιούμε για το εκάστοτε πρόβλημα που έχουμε να αντιμετωπίσουμε. Το string που σε όλες τις θέσεις των bit έχει το σύμβολο ‘\*’ αντιστοιχεί στον χώρο, και δεν μετράτε σαν ένα επιπλέον διαμέρισμα του χώρου. [2][5][6]

#### 4.2. Ο Χώρος Έρευνας σαν Υπερκύβος.

Στο πεδίο των γενετικών αλγορίθμων συνηθίζουμε να έχουμε την αντίληψη του πεδίου έρευνας, με την μορφή ενός υπερκύβου. Σε αυτό το σημείο οι περισσότεροι νέοι χρήστες συνηθίζουν να ρωτάνε για ποιόν λόγο να θεωρούμε κάτι τέτοιο ως χρήσιμο; Δηλαδή για ποιόν λόγο να βλέπουμε τον χώρο έρευνας στον οποίο εμπίπτει η έρευνα μας σαν έναν υπερκύβο;

Η απάντηση που ευρέως δίνεται για να εξηγήσουμε την υπολογιστική συμπεριφορά των γενετικών αλγορίθμων δίνεται από την δουλειά του Holland. Ο Holland ανέπτυξε αρκετές αντικρουόμενες σχεδιάσεις για να εξηγήσει πως είναι δυνατό ένα Γενετικό Πλάνο ή ένας Γενετικός Αλγόριθμος να επιλύσει δύσκολα προβλήματα και σύνθετες έρευνες με το να δειγματοληπτεί σε υπερεπίπεδες κατανομές ενός χώρου έρευνας. Ίσως, ο καλύτερος τρόπος για να κατανοήσουμε πώς ένας γενετικός αλγόριθμος μπορεί να ελεγχθεί με δειγματοληψία της υπερεπίπεδης κατανομής είναι να θεωρήσουμε ένα τριών διαστάσεων χώρο.

Θεωρούμε ότι έχουμε ένα πρόβλημα που για την επίλυση του επαρκή η κωδικοποίηση των μελών με ακριβώς τρία Bit. Αυτό μπορεί να αναπαρασταθεί σαν ένας κύβος, με το string 000 στην αρχή. Οι γωνίες του κύβου αριθμούνται με πλειάδες από bit. Κάθε μία από τις διπλανές γωνίες θα έχουμε μία επιγραφή και θα διαφέρουν με ένα bit. Στο μπροστινό επίπεδο υπάρχουν όλα τα σημεία που ξεκινάνε από 0. Αν υπάρχει το σύμβολο ‘\*’ που ονομάζεται wild card και ο ρόλος του είναι να ταιριάζει σύμβολα, τότε αυτό το επίπεδο μπορεί να αντιπροσωπευθεί από ειδικά string που έχουν την μορφή 0\*\*. Τα string που

περιέχουν τον χαρακτήρα '\*' ονομάζονται ως schemata. Κάθε schema αντιστοιχεί σε ένα υπερεπίπεδο στον χώρο έρευνας. Ο όρος 'order' ενός υπερεπιπέδου αναφέρεται στην πραγματική τιμή που έχουν τα bit σε κάθε schema, έτσι το 1\*\* λέγεται και order – 1.

Αμα θέλαμε να απεικονίσουμε έναν χώρο τεσσάρων διαστάσεων θα μπορούσαμε να τον αναπαραστήσουμε ως έναν κύβο 'κρεμασμένο' στο εσωτερικό ενός άλλου κύβου. Τα σημεία σε αυτήν την περίπτωση θα μαρκάρονται ως εξής : Στον εσωτερικό όσο και στον εξωτερικό κύβο θα ακολουθήσουμε την ίδια στρατηγική που ακολουθήσαμε στον τριών διαστάσεων κύβο. Το πρόθεμα για την κάθε κορυφή του εσωτερικού κύβου η τιμή του bit θα είναι 1, ενώ για τον εξωτερικό θα είναι 0. Αυτό δημιουργεί μία καταχώρηση στα σημεία του υπερδιαστήματος και δίνει την κατάλληλη ένδειξη ότι τα σημεία είναι γειτονικά μεταξύ τους, εφόσον η τιμή των strings που τα χαρακτηρίζουν διαφέρει μόνο κατά ένα bit. Ο εσωτερικός κύβος τώρα αντιστοιχεί με το υπερεπίπεδο διάστημα 1\*\*\*, ενώ του εξωτερικού κύβου αντιστοιχούν στο 0\*\*\*. Με αυτό τον τρόπο αναπαράστασης των σημείων του υπερκύβου μπορούμε να βγάλουμε συμπεράσματα για τα σημεία που ελέγχουμε. Μερικά παραδείγματα πολύ ενδεικτικά είναι τα ακόλουθα : Έστω ότι έχουμε το σημείο \*0\*\* αντιστοιχεί στο υποσύνολο των σημείων που αντιστοιχούν στο μπροστινό μέρος των δύο κύβων. Ένα order – 2 υπερεπίπεδο αντιστοιχεί στο μπροστινό μέρος του εσωτερικού και εξωτερικού κύβου. [2][5][6]

## **5. Η διαδικασία της Επιλογής.**

Όπως έχουμε ήδη προαναφέρει οι γενετικοί αλγόριθμοι είναι εμπνευσμένοι από την συμπεριφορά της φύσης. Με γνώμονα λοιπόν την φύση, αφομοιώνουμε τις διαδικασίες και τις λειτουργίες που ακολουθεί κατά τρόπο αληθοφανή, στους γενετικούς αλγορίθμους. Σε αυτό το σημείο θα ήταν ενδιαφέρον να δούμε τον τρόπο με τον οποίο η φύση λειτουργεί και πώς πράττει τις διαδικασίες της επιλογής για να τις συγκρίνουμε με αυτές που εκτελούν οι γενετικοί αλγόριθμοι.

### **5.1. Η συμπεριφορά της φύσης.**

Μία ενδιαφέρουσα προσέγγιση από την οποία μπορούμε να δούμε την συμπεριφορά της φύσης και των διεργασιών της, δίνεται από την μελέτη του Lamarck.

Στην μελέτη του, ο Lamarck μεταξύ άλλων αναφέρει ότι κατά το χρονικό διάστημα που στο περιβάλλον γίνεται μία μεταβολή, αυτό δηλώνει ότι εκείνη την χρονική στιγμή συμβαίνει μία εξέλιξη. Οι εξελίξεις και οι αλλαγές στο περιβάλλον γίνονται κατά την διάρκεια πολλών γενεών. Τα είδη που συμμετέχουν στις γενιές υφίστανται τις αλλαγές του περιβάλλοντος και προσαρμόζονται σε αυτές. Οι μονάδες ενός πληθυσμού που ταιριάζουν και προσαρμόζονται καλύτερα στο περιβάλλον και στις αλλαγές του, τείνουν να επιβιώνουν και να παράγονται για μεγαλύτερο χρονικό διάστημα.

Επίσης, οι αλλαγές στα είδη προάγονται από την κληρονομικότητα. Τα χαρακτηριστικά των προγόνων κωδικοποιημένα διαμέσω των γενεών, μεταφέρονται στους απογόνους τους και διαδίδονται στις καινούργιες γενεές. Στην αναπαραγωγή, τα χρωμοσώματα των απογόνων είναι η μίξη αυτών που διαθέτουν οι γονείς. Τα χαρακτηριστικά των απογόνων μερικώς κληρονομούνται από τους γονείς και μερικά είναι αποτέλεσμα της καινούργιας γενεάς που συμμετέχουν, αυτά δημιουργούνται κατά την διάρκεια της διαδικασίας της αναπαραγωγής.

Συνοπτικά, μπορούμε να πούμε ότι οι απόγονοι διαμορφώνουν το σύνολο των χαρακτηριστικών τους λαμβάνοντας κάποιο μέρος των επίκτητων χαρακτηριστικών των προγόνων τους, και οι νέες ικανότητες

που τα ίδια τα άτομα αναπτύσσουν προκειμένου να προσαρμοστούν και να επιβιώσουν στο μεταβαλλόμενο περιβάλλον τους.[3]

## 5.2. Συμπεριφορά των Γενετικών Αλγορίθμων

Η διαδικασία της επιλογής, είναι η λειτουργία κατά την οποία οι μονάδες που βρίσκονται στον τρέχον πληθυσμό επιλέγονται είτε για «ζευγάρωμα», είτε για να συμμετάσχουν απευθείας στην επόμενη γενιά. Στη πλειονότητα των περιπτώσεων εφαρμογής των γενετικών αλγορίθμων τα επιλεγμένα μέλη δημιουργούν έναν ενδιάμεσο πληθυσμό και περνάνε από την διαδικασία του Ανασυνδυασμού Γονιδίων με τους τελεστές της Διασταύρωσης και της Μετάλλαξης.

Για να εξομοιώσουμε την διαδικασία της επιλογής, όπως ακριβώς την εκτελεί η ίδια η φύση, χρησιμοποιούμε την τιμή καταλληλότητας, όπως έχει υπολογιστεί για κάθε ένα μέλος από την συνάρτηση καταλληλότητας. Με την βοήθεια της τιμής καταλληλότητας μπορούμε να επιλέξουμε τις μονάδες με την καλύτερη πιθανότητα επιλογής.

Τα περισσότερα σχήματα επιλογής που χρησιμοποιούνται από τους Γενετικούς Αλγορίθμους, ακολουθούν την παρακάτω διαδικασία :

1. Αρχικά, με την βοήθεια της διαδικασίας της Επιλογής δημιουργούμε έναν ενδιάμεσο πληθυσμό, συνήθως τον αποκαλούμε mating pool.
2. Έπειτα, επιλέγουμε τυχαία ζεύγη μελών από τον ενδιάμεσο πληθυσμό και εφαρμόζουμε τους τελεστές της Διασταύρωσης.

Στην παραπάνω διαδικασία που περιγράψαμε πρέπει να γίνουν κάποιες θεωρήσεις προκειμένου να είναι αντιπροσωπευτική των διαδικασιών της φύσης, όπως :

- I. Πρέπει να έχουμε αναπτύξει μία αντικειμενική διαδικασία, η οποία να μας παρέχει ένα μέτρο ποιότητας, έστω Q, για να αξιολογήσουμε και να κρίνουμε τις λύσεις του προβλήματος μας.
- II. Με την βοήθεια του μέτρου ποιότητας, γίνεται πολύ πιθανή η επιλογή του καταλληλότερου μέλους για να συνεχίσει να υπάρχει σε επόμενες γενεές. Μπορούμε να λάβουμε το μέτρο ποιότητας ως την τιμή καταλληλότητας, του κάθε μέλους του πληθυσμού. Το μέτρο ποιότητας πρέπει είτε να μεγιστοποιηθεί, είτε να ελαχιστοποιηθεί ανάλογα με την εκδοχή του γενετικού αλγορίθμου που έχουμε επιλέξει να χρησιμοποιήσουμε. Σε διαφορετικές εκδοχές τους οι γενετικοί αλγόριθμοι μπορεί να υποστηρίζουν να κρατάνε τους καλύτερους ενός πληθυσμού σε επόμενες γενιές, είτε να τους αντικαθιστούνε.

Στους γενετικούς αλγορίθμους η πιθανότητα τα μέλη του τρέχοντος πληθυσμού να αντιγραφούν(Duplicate) και να τοποθετηθούν στον ενδιάμεσο πληθυσμό είναι ανάλογη με την τιμή της συνάρτησης καταλληλότητας. Όσο καλύτερη είναι η τιμή που έχει ένα μέλος σε σχέση με την τιμή των υπολοίπων μελών, τόσο αυξάνεται η πιθανότητα να επιλεγεί το συγκεκριμένο μέλος. Ένα μέλος μπορεί να επιλεγεί περισσότερες από μια φορές για να συμμετάσχει σε επόμενους πληθυσμούς.

Έχουν αναπτυχθεί διαφορετικές μέθοδοι για να εκτελέσουμε την λειτουργία της Επιλογής – Selection. Μία αρχική διάκριση των μεθόδων αυτών είναι σε αιτιοκρατικές και σε στοχαστικές μεθόδους. Για την ολοκληρωμένη κάλυψη του αντικειμένου θα αναφέρουμε λίγα στοιχεία για τις αιτιοκρατικές μεθόδους χωρίς να μπούμε σε πολλές λεπτομέρειες καθώς δεν συσχετίζεται άμεσα με τους γενετικούς αλγορίθμους.[3],[7]

### 5.3. Οι αιτιοκρατικές μέθοδοι.

Οι αιτιοκρατικές μέθοδοι εφαρμόζουν μια εξαντλητική αναζήτηση σε όλο το πεδίο ορισμού των μεταβλητών της συνάρτησης που θέλουμε να μελετήσουμε. Τα αποτελέσματα που μας επιστρέφουν αυτές οι μέθοδοι είναι εγγυημένα τα βέλτιστα για να δώσουμε λύση σε ένα πρόβλημα. Αυτό είναι και το μεγαλύτερο πλεονέκτημα τους έναντι των στοχαστικών μεθόδων.

Ένα από τα σημαντικότερα πρόβλημα που τίθενται όταν επιλέξουμε να χρησιμοποιήσουμε κάποια από τις αιτιοκρατικές μεθόδους, είναι να διατυπώσουμε το καταλληλότερο κριτήριο τερματισμού της διαδικασίας. Όταν όμως, επιλέξουμε να εφαρμόσουμε μία από αυτές τις μέθοδοι θα πρέπει η συνάρτηση  $f$  να πληρή κάποιες προϋποθέσεις. Πρέπει επίσης να τονίσουμε ότι στην επιλογή του ορθότερου κριτηρίου τερματισμού οφείλεται κατά ένα πολύ μεγάλο ποσοστό το πλεονέκτημα της μεθόδου.

Το σημαντικότερο μειονέκτημα που έχουν οι αιτιοκρατικές μέθοδοι είναι το γεγονός ότι απαιτείται μεγάλη υπολογιστική ισχύς προκειμένου να φτάσουμε στο κριτήριο τερματισμού. Ένας επιπλέον αποτρεπτικός παράγοντας είναι ο χρόνος που απαιτείται για να εξετάσουμε ολόκληρο το πεδίο ορισμού της συνάρτησης, καθώς έχει διαπιστωθεί ότι ο χρόνος αυξάνει εκθετικά όσο αυξάνει το πλήθος των μεταβλητών του προβλήματος μας.

### 5.4. Στοχαστικές μέθοδοι.

Οι Στοχαστικές μέθοδοι είναι μία αρκετά δημοφιλής κατηγορία μεθόδων κυρίως για προβλήματα πολλών μεταβλητών.

Οι στοχαστικές μέθοδοι σε αντιπαράθεση με της αιτιοκρατικές δεν εφαρμόζουν εξαντλητική αναζήτηση σε όλο το πεδίο ορισμού των μεταβλητών της συνάρτησης που μας ενδιαφέρει. Ο τρόπος με τον οποίο εξετάζουν το πεδίο έρευνας τους, είναι να λαμβάνουν δείγματα από αυτό και να επικεντρώνουν την έρευνα τους στις ενδεικνυόμενες υποπεριοχές του, ανάλογα με τα αποτελέσματα που λαμβάνουν κατά την διάρκεια εκτέλεσης τους. Επομένως, τα αποτελέσματα που μας επιστρέφουν ως λύση ενός προβλήματος δεν είναι απολύτως εγγυημένα. Οι μέθοδοι αυτοί γίνονται περισσότερο αξιόπιστες όσο περισσότερο αυξάνεται το πλήθος των σημείων του αρχικού δείγματος.

Μία επίσης σημαντική διαφορά που έχουν οι στοχαστικές μέθοδοι με τις αιτιοκρατικές είναι οι συναρτήσεις τερματισμού που χρησιμοποιούν. Σε αυτές τις μεθόδους σαν κριτήριο τερματισμού χρησιμοποιούμε κυρίως μία συνθήκη που να πιστοποιεί σε μεγάλο βαθμό ότι η πιθανότητα για την ορθότητα των αποτελεσμάτων τους, πλησιάζει σε έναν ικανοποιητικό βαθμό.

#### 5.4.1. Κατάλοιπο Στοχαστικής Δειγματοληψίας.

Η μέθοδος του *Κατάλοιπου Στοχαστικής Δειγματοληψίας* χρησιμοποιεί μία μέθοδο γνωστή ως Stochastic Universal Sampling. Είναι μία διαδικασία επιλογής που πλησιάζει πιο κοντά στην αναμενόμενη τιμή καταλληλότητας. Η διαδικασία αυτή έχει ως εξής :

Για κάθε μέλος  $i$ , η τιμή καταλληλότητας δίνεται από την έκφραση  $f_i / f_{average}$ . Όταν η τιμή είναι μεγαλύτερη της μονάδας, το ακέραιο μέρος του αριθμού δηλώνει πόσα αντίγραφα του μέλους θα τοποθετηθούν απευθείας στον ενδιάμεσο πληθυσμό. Έπειτα σε αυτό το μέλος δίνεται η δυνατότητα να τοποθετήσει επιπλέον μέλη στον ενδιάμεσο πληθυσμό ανάλογα όμως με το δεκαδικό μέρος της τιμής καταλληλότητας. Όλα τα μέλη συμπεριλαμβανομένων και αυτών με μικρότερη τιμή από 1.0 , να

τοποθετήσουν επιπρόσθετα αντίγραφα στον ενδιάμεσο πληθυσμό, ανάλογα με την πιθανότητα που αντιστοιχεί στο κλασματικό μέρος της τιμής καταλληλότητας.

Για παράδειγμα, ένα μέλος με τιμή καταλληλότητας ίση με 1,36, τοποθετείται 1 αντίγραφο στον ενδιάμεσο πληθυσμό, και μετά λαμβάνει 0,36 ευκαιρίες για να τοποθετήσει ένα δεύτερο αντίγραφο στον ενδιάμεσο πληθυσμό. Ένα string με συνάρτηση καταλληλότητας ίση με 0,54 έχει 0,54 ευκαιρία για να τοποθετήσει ένα string στον ενδιάμεσο πληθυσμό.[2] , [5]

**Stochastic Universal Sampling** η διαδικασία αυτής της μεθόδου είναι η εξής: Θεωρούμε ότι ο πληθυσμός είναι εξαπλωμένος τυχαία σε ένα διάγραμμα πίτας, όπου για κάθε μέλος του πληθυσμού εκχωρείται χώρος αντίστοιχος με την τιμή της συνάρτησης καταλληλότητας του κάθε μέλους. Δίπλα από τον αρχικό τροχό της τύχης τοποθετείται ένας εξωτερικός τροχός με  $N$  ίσα μεταξύ τους χωρισμένα κομμάτια που το κάθε ένα έχει και ένα δείκτη. Ένα μόνο γύρισμα του τροχού της τύχης θα μας δώσει ταυτόχρονα τα  $N$  μέλη που απαιτούνται για να γεμίσουμε τον ενδιάμεσο πληθυσμό.

#### 5.4.2. Επιλογή Ανάλογη της Καταλληλότητας.

Η επιλογή ανάλογη της καταλληλότητας είναι το πιο ευρέως χρησιμοποιούμενο είδος Επιλογής και βρίσκει άμεση εφαρμογή στους γενετικούς αλγορίθμους με την ανάπτυξη δύο πρακτικών εφαρμογών τις οποίες περιγράφουμε σε αυτό το εδάφιο. Αυτή η μέθοδος Επιλογής βασίζεται στην τιμή καταλληλότητας που εκχωρείται σε κάθε μέλος του πληθυσμού. Η τιμή καταλληλότητας μας παρέχει το μέτρο ποιότητας που χρειαζόμαστε για την επιλογή ή όχι ενός μέλους.

Ας υποθέσουμε ότι  $f_i$  είναι η τιμή καταλληλότητας του μέλους  $i$  , και η μαθηματική έκφραση  $f_{average} = 1 / N * \sum_i f_i$  μας δίνει το μέσο όρο καταλληλότητας του συνολικού πληθυσμού. Σε αυτούς τους αλγορίθμους επιλογής η πιθανότητα που έχει το μέλος  $i$  να επιλεγεί για αναπαραγωγή είναι ίση με τον λόγο  $p_i = f_i / \sum_i f_i = f_i / f_{average} N$  .

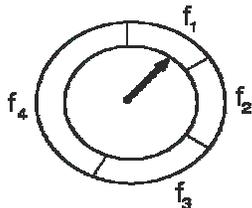
Μία πρακτική εφαρμογή της μεθόδου επιλογής ανάλογης της τιμής καταλληλότητας μπορεί να υλοποιηθεί με την “Μέθοδο του τροχού της τύχης”. Η φιλοσοφία που ακολουθεί αυτός ο στοχαστικός αλγόριθμος είναι ο ακόλουθος :

Αναπαριστούμε τον τρέχον πληθυσμό, ως χαρτογράφηση πάνω στον τροχό της τύχης. Κάθε μέλος του πληθυσμού αντιπροσωπεύεται από ένα τμήμα του τροχού. Ο χώρος που θα καταλαμβάνει κάθε ένα μέλος στον τροχό, είναι αντίστοιχος με την τιμή καταλληλότητας που του έχει εκχωρηθεί, από την εκτέλεση της συνάρτησης καταλληλότητας (Fitness Function).

Είναι φανερό ότι όσο μεγαλύτερη είναι η τιμή καταλληλότητας για κάποιο μέλος του πληθυσμού, αυτό συνεπάγεται ότι θα εκπροσωπείται από μεγαλύτερο τμήμα του συνολικού χαρτογραφημένου χώρου του τροχού της τύχης. Έτσι η πιθανότητα να επιλεγεί αυτό το μέλος είναι μεγαλύτερη, συγκριτικά με άλλα μέλη που η τιμή καταλληλότητας τους είναι μικρότερη.

Αυτός ο στοχαστικός αλγόριθμος είναι μία δίκαιη διαδικασία Επιλογής καθώς όλα τα μέλη που συμμετέχουν σε έναν πληθυσμό έχουν την ευκαιρία να επιλεγούν και να συνεχίσει η πορεία εξέλιξης τους. Σε αυτή την μέθοδο δεν αποκλείουμε τους χειρότερους του πληθυσμού, αλλά τους δίνουμε ευκαιρίες ανάλογες της τιμής καταλληλότητας τους, όπως όλα τα μέλη.

Για παράδειγμα, αν το πλήθος των μελών του πληθυσμού είναι  $N = 4$ , οι τιμές καταλληλότητας των μελών είναι  $f_1 = f_2 = 10$ ,  $f_3 = 15$  και  $f_4 = 25$ . Η απεικόνιση των παραπάνω τιμών στον τροχό της τύχης θα είναι η εξής:



Σχήμα 2<sup>ο</sup> : Ο τροχός της τύχης.

Η διαδικασία με την οποία γίνεται η επιλογή των μονάδων που θα συνεχίσουν στην επόμενη γενιά είναι η ακόλουθη : Επαναληπτικά γυρίζουμε τον τροχό της τύχης. Τα μέλη επιλέγονται όταν ο τροχός σταματήσει και υποδεικνύονται από έναν δείκτη. Ο αριθμός των επαναλήψεων εξαρτάται από το πλήθος των μελών που χρειαζόμαστε, για να γεμίσουμε τον ενδιαμέσο πληθυσμό.

Η διαδικασία γίνεται χρησιμοποιώντας την Στοχαστική Δειγματοληψία με Αντικατάσταση, ο απώτερος σκοπός που έχει είναι να γεμίσουμε τον ενδιαμέσο πληθυσμό.

Για λόγους ευνόητους ο τροχός της τύχης δεν μπορεί να πραγματοποιηθεί άμεσα από τους υπολογιστές.

Παρακάτω θα περιγράψουμε δύο διαφορετικές πρακτικές εφαρμογές του τροχού της τύχης.

Στην πράξη ο τροχός της τύχης μπορεί να αναπαρασταθεί από ένα διάνυσμα  $v$  αποτελούμενο από  $M$  στοιχεία, σε ένα εύρος  $\{1, \dots, N\}$  και από έναν τυχαίο δείκτη  $\theta$ . Την τιμή του  $M$  την ορίζουμε εμείς, κατά αυτόν τον τρόπο προσδιορίζουμε το πλήθος των ευκαιριών που δίνουμε σε κάθε μέλος να επιλεγεί. Έτσι όσο μεγαλύτερη είναι η τιμή του δείκτη  $M$  δίνουμε περισσότερες ευκαιρίες στα μέλη να επιλεγούν. Αν αυτή η τιμή είναι μικρή, τότε υπάρχει ο κίνδυνος να μην είναι αντικειμενική η όλη διαδικασία προς τα στοιχεία με μικρή τιμή καταλληλότητας. Επομένως, θα πρέπει να δείξουμε ιδιαίτερη προσοχή στην εκχώρηση κάποιας τιμής στο πλήθος των στοιχείων του διανύσματος,  $M$ .

Ο τρόπος με τον οποίο θα τοποθετηθούν τα στοιχεία στο διάνυσμα δεν είναι τυχαίος. Τα υποψήφια στοιχεία τοποθετούνται σύμφωνα με την τιμή καταλληλότητας τους και η ταξινόμηση που θα πρέπει να έχουν στο διάνυσμα είναι αύξουσα. Δηλαδή θα ξεκινήσουμε να τοποθετούμε πρώτα τα μέλη με την μικρότερη τιμή καταλληλότητας και θα καταλήξουμε με τα μέλη που έχουν την μεγαλύτερη τιμή καταλληλότητας.

Επίσης, ο αριθμός των εκπροσώπων του κάθε μέλους, που θα τοποθετηθούν στο πλήθος των στοιχείων  $M$  του διανύσματος πρέπει να είναι ανάλογο με την τιμή καταλληλότητας του μέλους, στον πληθυσμό. Έτσι ένα μέλος με μεγαλύτερη τιμή καταλληλότητας θα εκπροσωπείται με περισσότερα αντίγραφα του στο διάνυσμα. Με αυτόν τον τρόπο δίνουμε περισσότερες ευκαιρίες επιλογής στα καταλληλότερα μέλη του πληθυσμού και τις αντίστοιχες ελπίδες στα μέλη με την μικρότερη πιθανότητα επιβίωσης, χωρίς όμως να τα απορρίπτουμε πλήρως. Έτσι γίνεται πιο δίκαιη ολόκληρη η διαδικασία και ταυτίζεται περισσότερο με την διαδικασία που ακολουθεί η φύση.

Την 'κυκλική' κίνηση του τροχού της τύχης την υποκαθιστούμε με έναν τυχαίο δείκτη  $\theta$ . Ο δείκτης  $\theta$  λαμβάνει συνεχώς τυχαίες τιμές, και μας υποδεικνύει κάθε φορά το στοιχείο που θα επιλεγεί για να

συνεχίσει είτε για ζευγάρι, είτε για να εισαχθεί απευθείας σε νέο πληθυσμό. Για την επιλογή της κάθε μονάδας, έχουμε  $v(\theta)$ .

Στο προηγούμενο παράδειγμα, αν το στοιχείο  $f_1$  είναι το 1,  $f_2$  το 2,  $f_3$  το 3 και  $f_4$  το 4. Έστω ότι ο αριθμός των ευκαιριών επιλογής για κάθε μέλος είναι  $M = 12$ , τότε το διάνυσμα  $v$  θα έχει την μορφή :

1	1	2	2	3	3	3	4	4	4	4
---	---	---	---	---	---	---	---	---	---	---

Αν ο δείκτης  $\theta = 6$  τότε επιλέγηκε το  $v(6) = 3$ , δηλαδή το στοιχείο  $f_3$  με τιμή καταλληλότητας 15.

Το πρόβλημα που αντιμετωπίζουμε με αυτή την μέθοδο είναι η ποσοτική αναπαράσταση της πιθανότητας  $p_i$  στο διάνυσμα : Τι πολιτική θα ακολουθήσουμε για να αντιστοιχίσουμε την τιμή καταλληλότητας του μέλους  $i$ , με το πλήθος των αντιγράφων του στο διάνυσμα. Η ορθότητα αυτής της μεθόδου εξαρτάται άμεσα από το πλήθος των ευκαιριών επιλογής που δίνουμε στα μέλη, δηλαδή από το πλήθος των στοιχείων  $M$  του διανύσματος  $v$ .

Εναλλακτικά ένας διαφορετικός τρόπος για να εξομοιώσουμε την εφαρμογή του τροχού της τύχης, είναι ο παρακάτω : Ομοίως και σε αυτή την εφαρμογή, τοποθετούμε τα στοιχεία ταξινομημένα ανάλογα με την τιμή καταλληλότητας τους σε ένα διάνυσμα, κατά αύξουσα σειρά. Η διαφορά με την προηγούμενη εφαρμογή είναι ότι τα στοιχεία τοποθετούνται μόνο μία φορά στο διάνυσμα και όχι τόσες φορές σε αναλογία με την τιμή καταλληλότητας που τους έχει υπολογιστεί. Έτσι κατά αυτόν τον τρόπο λύνουμε το πρόβλημα της ποσοτικής αναπαράστασης που αντιμετωπίζαμε στην προηγούμενη μέθοδο.

Έπειτα, αθροίζουμε τις τιμές καταλληλότητας όλων των μελών, και δημιουργούμε ένα σύνολο τιμών. Το άθροισμα που υπολογίζουμε μας δίνει το ανώτερο όριο του συνόλου. Από αυτό το διάστημα τιμών μπορεί να πάρει τιμή ένας δείκτης  $r$ , τον οποίο χρησιμοποιούμε για να μας υποδείξει το επιλαχόν στοιχείο. Η τιμή του δείκτη είναι τυχαία και πρέπει να ανήκει στο διάστημα του αθροίσματος των τιμών καταλληλότητας, δηλαδή  $r \in [0, \sum_{j=1}^N f_j]$  σύνολο τιμών.

Δανειζόμαστε τα δεδομένα που χρησιμοποιήσαμε στο προηγούμενο παράδειγμα για να αναπτύξουμε στην διαδικασία που περιγράψαμε νωρίτερα. Τοποθετούμε σε αύξουσα σειρά τις τιμές καταλληλότητας των μελών του πληθυσμού. Η μορφή του διανύσματος θα είναι :

$f_1$	$f_2$	$f_3$	$f_4$
10	10	15	25

Το επόμενο βήμα είναι να υπολογίσουμε το άθροισμα των τιμών καταλληλότητας των μελών. Το άθροισμα τους είναι 50. Επομένως, δημιουργείται το σύνολο τιμών  $[0,50)$ . Από το διάστημα αυτό μπορεί να πάρει τιμή ο δείκτης  $r$ . Έστω ότι η τιμή που λαμβάνει τυχαία ο δείκτης  $r$  είναι 19. Για να βρούμε ποιο στοιχείο θα επιλεγεί για αναπαραγωγή, ο αλγόριθμος ενεργεί ως εξής:

Αθροίζει τις τιμές καταλληλότητας, από την μικρότερη προς την μεγαλύτερη και συγκρίνει αν το άθροισμα τους είναι μεγαλύτερο από την τυχαία τιμή του δείκτη  $r$ . Το στοιχείο που θα επιλεγεί είναι αυτό που το άθροισμα των τιμών καταλληλότητας των μελών είναι μεγαλύτερο του δείκτη  $r$ . Το επιλαχόν στοιχείο είναι το τελευταίο που αθροίσαμε. Η διαδικασία αυτή επαναλαμβάνεται για όσα στοιχεία θέλουμε να τοποθετήσουμε στον ενδιάμεσο πληθυσμό. Στο παράδειγμα μας πάλι, το στοιχείο που επιλέγεται με  $r = 19$  είναι το δεύτερο μέλος, δηλαδή το  $f_2 = 10$ .

Το μειονέκτημα αυτής της μεθόδου είναι η ταχύτητα. Αν για παράδειγμα θέλουμε να επιλέξουμε 10 στοιχεία για τον ενδιάμεσο πληθυσμό, τότε πρέπει να υπολογίζουμε 10 διαφορετικά αθροίσματα.

Το κοινό σημείο της προηγούμενης μεθόδου με αυτή που περιγράψαμε μόλις είναι ότι τα στοιχεία θα πρέπει να είναι διατεταγμένα σε αύξουσα σειρά, από την μικρότερη προς την μεγαλύτερη τιμή καταλληλότητας.

Στις μεθόδους Επιλογής Ανάλογης της Καταλληλότητας τα προβλήματα που αντιμετωπίζουμε είναι αυτό της Πρόωρης Σύγκλισης και της Στασιμότητας.

**Πρόωρη σύγκλιση** έχουμε όταν δεν μπορούμε να συνεχίσουμε την περαιτέρω εξέλιξη του πληθυσμού λύσεων που διαθέτουμε. Κάτι τέτοιο μπορεί να προέλθει όταν ο πληθυσμός μας κατακλυστεί από πολλαπλά αντίγραφα του ίδιου μέλους σε έναν πληθυσμό. Κάτι τέτοιο είναι δυνατό να γίνει όταν σε έναν πληθυσμό εμφανιστεί κάποιο στοιχείο σε σχετικά πρώιμες γενεές, που η τιμή καταλληλότητας του είναι κατά πολύ μεγαλύτερη από τον μέσο όρο καταλληλότητας του συνολικού πληθυσμού, δηλαδή  $f_i \gg f_{average}$ , και απέχει πολύ από την λύση που επιζητούμε στο πρόβλημα μας. Αν για παράδειγμα, επιθυμούσαμε να μεγιστοποιήσουμε μία συνάρτηση, που η τιμή καταλληλότητας αυτού του στοιχείου απέχει πολύ από το πραγματικό μέγιστο της συνάρτησης που αναζητούμε για το πρόβλημα, δηλαδή  $f_i \ll f_{max}$ .

Ο κίνδυνος που υπάρχει σε ένα τέτοιο ενδεχόμενο όταν έχουμε την εμφάνιση περιπτώσεων πρόωρου σύγκλισης είναι το στοιχείο αυτό να κατακλείσει τον διαθέσιμο πληθυσμό με αντίγραφα του, τότε οι γενετικοί αλγόριθμοι δεν μπορούν να παράγουν νέες λύσεις με τους τελεστές ανασυνδυασμού γονιδίων. Μόνο με τον τελεστή της Μετάλλαξης είναι εφικτό κάτι τέτοιο.

Για να αποδείξουμε τον τελευταίο ισχυρισμό, φέρνουμε ένα ακραίο παράδειγμα το οποίο όμως φανερώνει το μέγεθος του προβλήματος. Υποθέτουμε ότι έχουμε έναν πληθυσμό  $N = 5$  μελών. Οι τιμές καταλληλότητας των πρώτων τεσσάρων μελών είναι  $f_1 = f_2 = f_3 = f_4 = 1$ , ενώ η τιμή καταλληλότητας του πέμπτου στοιχείου του πληθυσμού  $f_5 = 10$ . Η αναπαράσταση των τιμών καταλληλότητας των μελών του πληθυσμού σε ένα διάγραμμα θα είναι η εξής :

1	1	1	1	10
---	---	---	---	----

Το άθροισμα των τιμών καταλληλότητας του πληθυσμού είναι 14. Η πιθανότητα που έχει το πέμπτο στοιχείο να επιλεγεί είναι  $10 / 14 = 0,71$  ή 71%. Αν ο ενδιάμεσος πληθυσμός που θέλουμε να δημιουργήσουμε είναι  $N = 100$  μέλη, τότε τα 71 στοιχεία του πληθυσμού θα είναι αντίγραφα του 5<sup>ου</sup> στοιχείου.

Ένα άλλο πρόβλημα που έχουμε να αντιμετωπίσουμε είναι το πρόβλημα της Στασιμότητας. **Στασιμότητα** έχουμε όταν η τιμή καταλληλότητας των μελών δεν μας παρέχει ένα επαρκές μέτρο ποιότητας για την ασφαλή και αξιόπιστη πληροφόρηση του αλγορίθμου ως κριτήριο επιλογής. Μία εκδοχή αυτού του προβλήματος εμφανίζεται όταν προς το πέρασ της εκτέλεσης του γενετικού αλγορίθμου όλες οι μονάδες του πληθυσμού, τείνουν να έχουν σχετικά υψηλές και παρόμοιες τιμές καταλληλότητας, δηλαδή  $\forall i \ f_i \approx f_{max}$ .

Αν για παράδειγμα έχουμε έναν πληθυσμό 1000 στοιχείων, με σχεδόν παρόμοιες τιμές καταλληλότητας, δεν είναι πολύ εύκολο να γίνει η επιλογή του καταλληλότερου στοιχείου. Να επιλέξουμε ένα μέλος

τυχαία είναι σαν να κάνουμε τυχαία επιλογή, αυτή η πολιτική δεν είναι στα πλαίσια των γενετικών αλγορίθμων. Μία εναλλακτική εκδοχή του ίδιου προβλήματος είναι όταν στον ενδιάμεσο πληθυσμό έχουμε από ένα μόνο αντίγραφο του κάθε μέλους, δηλαδή  $N_I \approx \dots \approx N_N \approx I$ .

Και στις δύο εναλλακτικές εκδοχές του προβλήματος της Στασιμότητας που αναφέραμε αντιμετωπίζουμε το ίδιο δίλημμα, επειδή δεν είναι εμφανές πιο μέλος του πληθυσμού είναι το καταλληλότερο για να επιλεγεί. Αυτό οφείλεται στο γεγονός ότι οι τιμές καταλληλότητας των μελών δεν μας παρέχουν ξεκάθαρα την πληροφορία που χρειαζόμαστε για να διαχωρίσουμε ποία είναι τα καλύτερα και ποία τα χειρότερα μέλη σε έναν πληθυσμό.

Στις μεθόδους Επιλογής Ανάλογης της τιμής Καταλληλότητας δίδεται ιδιαίτερη βαρύτητα στην λεπτομέρεια υπολογισμού των τιμών καταλληλότητας των μελών. Αυτό είναι ένα αρνητικό σημείο αυτών των μεθόδων επιλογής.

Η αντιμετώπιση των δύο παραπάνω προβλημάτων γίνεται με την εφαρμογή τεχνικών Κλιμάκωσης στην τιμή Καταλληλότητας. [3],[5],[6]

#### 5.4.2.1. Τεχνικές Κλιμάκωσης της τιμής Καταλληλότητας.

Τα προβλήματα της προώρου Σύγκλισης και της Στασιμότητας που εμφανίζονται στις μεθόδους επιλογής ανάλογης της καταλληλότητας τα αντιμετωπίζουμε με διάφορους μεθόδους κλιμάκωσης των τιμών καταλληλότητας των μελών του πληθυσμού. Τα προβλήματα που είδαμε οφείλονται στο γεγονός ότι δεν υπάρχει διάκριση στις τιμές καταλληλότητας των μελών ή ότι υπάρχουν πολύ μεγάλες διαφορές μεταξύ τους.

Σε αυτήν την ενότητα θα δούμε ορισμένα παραδείγματα ίσως λίγο ακραίων προβλημάτων με τις αντίστοιχες ενδεικτικές λύσεις τις οποίες μπορούμε να δώσουμε. Μέσα από αυτά τα παραδείγματα θα προσπαθήσουμε να τονίσουμε το μέγεθος των προβλημάτων που περιγράψαμε πιο πάνω και να δείξουμε τις αρνητικές επιπτώσεις που μπορεί να έχουν στην εξέλιξη ενός γενετικού αλγορίθμου.

Ένα παράδειγμα που ταυτίζεται με την περιγραφή της πρώτης εκδοχής του προβλήματος της Στασιμότητας είναι το παρακάτω :

Έστω ότι έχουμε έναν πληθυσμό  $N = 4$  μελών, με τις αντίστοιχες τιμές καταλληλότητας τους να είναι τοποθετημένες και ταξινομημένες σε αύξουσα σειρά, σε ένα διάνυσμα. Η μορφή του είναι:

9,8	10,0	10,2	11,0
-----	------	------	------

Σε αυτήν την περίπτωση οι διαφορές στις τιμές καταλληλότητας τους είναι πολύ μικρές για να μπορέσουμε να επιλέξουμε ένα μέλος ως το καλύτερο του πληθυσμού ή να απορρίψουμε κάποιο άλλο ως το χειρότερο.

Μία λύση που μπορούμε να δώσουμε σε μία αντίστοιχη περίπτωση είναι να πριμοδοτήσουμε το λίγο καλύτερο μέλος του πληθυσμού με κάποια αρκετά μεγάλη τιμή καταλληλότητας, έστω  $M$ , το αμέσως επόμενο καλύτερο μέλος του πληθυσμού με μία μικρότερη τιμή έστω  $M / 2$ , κ.ο.κ. για όλα τα μέλη του διανύσματος. Στο παράδειγμα που αναφέραμε πιο πάνω αν η τιμή του  $M = 100$  την εκχωρούμε στο μέλος με την λίγο καλύτερη καταλληλότητα, στο αμέσως επόμενο μέλος δίνουμε την τιμή  $M / 2$ . Η τελική μορφή που θα πάρει το διάνυσμα θα είναι :

25	33,3	50	100
----	------	----	-----

Με αυτόν τον τρόπο αντιμετώπισης επιτυγχάνουμε να υπάρχει κάποιος βαθμός διάκρισης στις τιμές καταλληλότητας των μελών του πληθυσμού. Έτσι το κριτήριο ποιότητας μας παρέχει μία πιο σαφή εικόνα για την καταλληλότητα επιλογής και συμμετοχής των μελών σε επόμενους πληθυσμούς.

Ένα διαφορετικό παράδειγμα ακραίου προβλήματος που είναι πιθανό να βρεθούμε αντιμετώπιζοί είναι όταν έχουμε να διαχειρισθούμε έναν πληθυσμό με πολύ μεγάλες τιμές καταλληλότητας.

Μία ενδεικτική λύση κλιμάκωσης των τιμών καταλληλότητας είναι να αντικαταστήσουμε αυτές τις τιμές με την τιμή του λογαρίθμου τους. Έστω ότι διαθέτουμε έναν πληθυσμό  $N = 4$  μελών, με τις αντίστοιχες τιμές καταλληλότητας τοποθετημένες σε ένα διάνυσμα, είναι :

$10^{10}$	$10^{20}$	$10^{40}$	$10^{45}$
-----------	-----------	-----------	-----------

Ο μέσος όρος των τιμών καταλληλότητας είναι  $10^{29}$ . Όπως παρατηρούμε οι τιμές καταλληλότητας των μελών είναι πολύ μεγάλες και επομένως είναι πολύ δύσκολο να τις διαχειρισθούμε καθώς αυξάνονται οι απαιτήσεις σε μνήμη και ο χρόνος εκτέλεσης απλών πράξεων. Σε παρόμοιες περιπτώσεις μία λύση είναι να πάρουμε τους λογαρίθμους των μελών και να τις αντικαταστήσουμε με την τιμή του λογαρίθμου. Δηλαδή, στο παραπάνω παράδειγμα, οι λογαριθμικές τιμές είναι :

$$\log(10^{10}) = 10, \log(10^{20}) = 20, \log(10^{40}) = 40, \log(10^{45}) = 45$$

Η τελική μορφή του διανύσματος θα είναι η παρακάτω :

10	20	40	45
----	----	----	----

Με αυτό τον τρόπο κάνουμε τα μέλη του πληθυσμού πιο όμοια μεταξύ τους και τους δίνουμε περισσότερες ευκαιρίες να επιλεγούν. Επίσης κάνουμε τα μέλη του πληθυσμού πιο εύκολα διαχειρίσιμα, έτσι μπορούμε να εκτελέσουμε πράξεις όπως διαιρέσεις κ.α. πιο εύκολα. Καθώς επίσης, λύνουμε ένα πρόβλημα που είχαμε συναρτήσει και στην αναπαράσταση των μελών του πληθυσμού, το οποίο αναφερόταν στην δέσμευση υπερβάλλουσας μνήμης, την αύξηση της πολυπλοκότητας του αλγορίθμου και των τελεστών του.

Με αυτό το παραπάνω παράδειγμα δεν μας δίνεται η δυνατότητα να καλύψουμε όλες τις πτυχές που μπορεί να εμφανιστούν σε ένα παρόμοιο πρόβλημα, για αυτόν τον λόγο θα αναφέρουμε ένα παρόμοιο παράδειγμα. Έστω, ότι έχουμε έναν πληθυσμό  $N = 5$ , οι αντίστοιχες τιμές καταλληλότητας των μελών είναι τοποθετημένες σε ένα διάνυσμα ως :

1,5	1,6	50	100	100
-----	-----	----	-----	-----

Σε αυτό το παράδειγμα παρατηρούμε ότι η διαφορά που υπάρχει στις τιμές καταλληλότητας των μελών είναι πολύ μεγάλες. Το πρώτο και το πιο εμφανές πρόβλημα που έχουμε να αντιμετωπίσουμε είναι αυτό της Προώρου Σύγκλισης, δηλαδή τα δύο τελευταία μέλη με καταλληλότητα 100 να αντιγραφούν πολλές φορές στο πληθυσμό. Ένα δεύτερο πρόβλημα που μπορεί να οφείλεται στη μεγάλη διαφορά των καταλληλότητων, είναι ο κίνδυνος τα μέλη με πολύ μικρές καταλληλότητες να έχουν ουσιαστικά ελάχιστες ή ανύπαρκτη πιθανότητα να επιλεγούν για να συνεχίσουν σε επόμενους πληθυσμούς.

Στους γενετικούς αλγορίθμους δεν επιθυμούμε να συμβεί κάτι τέτοιο, αντιθέτως θέλουμε να δημιουργήσουμε τις διαδικασίες της επιλογής δίκαιες, όπως αυτές της φύσης. Σε έναν πληθυσμό

επιδιώκουμε να υπάρχουν κάποια μέλη με χαμηλές τιμές καταλληλότητας. Έτσι αφήνουμε μία πιθανότητα από τυχόν συμμετοχή τους σε μία διαδικασία διασταύρωσης για παράδειγμα, να αντιγραφεί ένα τμήμα της πληροφορίας τους και να μας επιστρέψουν πολύ καλά μέλη. Γενικά στους γενετικούς αλγορίθμους επιδιώκουμε να υπάρχει πλουραλισμός και ποικιλία χαρακτηριστικών στα μέλη του πληθυσμού που διατηρούμε.

Μπορούμε σε αυτήν την περίπτωση να κινηθούμε υπό την ίδια λογική με αυτή που ακολουθήσαμε προηγουμένως. Δηλαδή, να αντικαταστήσουμε τις τιμές καταλληλότητας με την τιμή του λογαρίθμου των τιμών καταλληλότητας. Οι πράξεις είναι :

$$\log(1,5) = 0,18 \text{ , } \log(1,6) = 0,2 \text{ , } \log(50) = 1,7 \text{ , } \log(100) = 2$$

Το διάνυσμα καταλληλοτήτων θα διαμορφωθεί ως εξής :

0,18	0,2	1,7	2	2
------	-----	-----	---	---

Με αυτό τον τρόπο μετατρέπουμε τις τιμές καταλληλότητας των μελών σε πιο όμοιες μεταξύ τους. Έτσι, μειώνουμε την πιθανότητα να επιλεγεί ο καλύτερος πάρα πολλές φορές και δίνουμε κάποιες ευκαιρίες επιλογής, σε μέλη με μικρότερες τιμές καταλληλότητας. Κατά αυτόν τον τρόπο κάνουμε την διαδικασία πιο δίκαιη επειδή δίνουμε την δυνατότητα και σε χειρότερα μέλη του πληθυσμού, να συνεχίσουν να συμμετάσχουν σε επόμενους πληθυσμούς.

Τέλος θα αναφέρουμε μία τελείως διαφορετική μέθοδο κλιμάκωσης. Με την μέθοδο αυτή επιτυγχάνουμε να φέρουμε τις τιμές καταλληλότητας των μελών του πληθυσμού στο διάστημα  $[0, 1]$ .

Έστω για παράδειγμα ότι έχουμε έναν πληθυσμό  $N = 5$  μελών. Οι τιμές καταλληλότητας των μελών βρίσκονται τοποθετημένες σε αύξουσα σειρά σε ένα διάνυσμα :

2	3	4	5	6
---	---	---	---	---

Το άθροισμα του συνόλου των καταλληλοτήτων είναι 20. Το 5<sup>ο</sup> στοιχείο έχει την μεγαλύτερη τιμή καταλληλότητας στο διάνυσμα με τιμή  $f_5 = f_{\max} = 6$ . Ενώ το στοιχείο με την μικρότερη καταλληλότητα είναι το 1<sup>ο</sup> στοιχείο με  $f_1 = f_{\min} = 2$ . Ο τύπος που θα χρησιμοποιήσουμε για την μεταφορά των καταλληλοτήτων στο διάστημα  $[0, 1]$ , είναι :  $f_i - f_{\min} / f_{\max} - f_{\min}$

Αν εφαρμόσουμε τον παραπάνω τύπο τότε οι τιμές των καταλληλοτήτων στο διάνυσμα θα είναι :

0	1/4	1/2	3/4	1
---	-----	-----	-----	---

Με αυτόν τον τρόπο η τιμή του καταλληλότερου μέλους αντικαθίσταται με μονάδα(1), ενώ του χειρότερου μέλους με το 0 .[3],[5],[6]

### 5.4.3. Επιλογή Βαθμίδας.

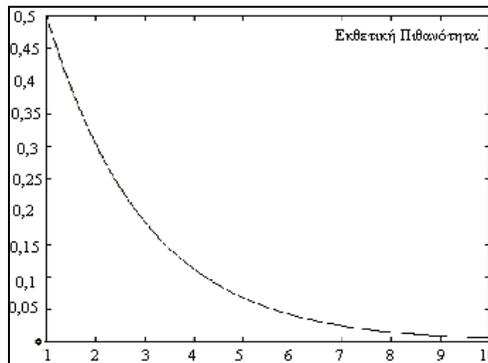
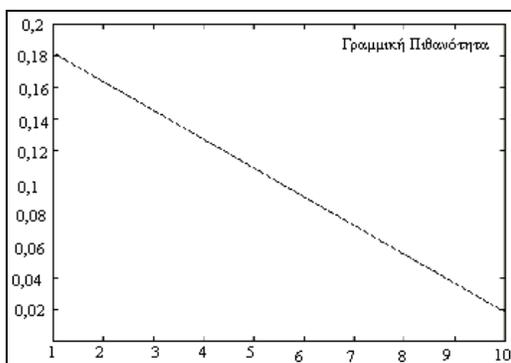
Μία διαφορετική μέθοδος επιλογής που μπορούμε να ακολουθήσουμε στους γενετικούς αλγορίθμους για να εκτελέσουμε την διαδικασία της Επιλογής, είναι η *Επιλογή Βαθμίδας*. Όπως η προηγούμενη μέθοδος επιλογής έτσι και αυτή είναι μία μέθοδος βιολογικά αληθοφανής.

Σε κάθε μονάδα του πληθυσμού εκχωρείται μία τιμή που εκφράζει την πιθανότητα που έχει το μέλος για να επιλεγεί και να συνεχίσει σε επόμενες γενεές. Η τιμή της πιθανότητας λαμβάνεται από μία δοθείσα **διακριτή κατανομή** ή αλλιώς, από την **συνάρτηση πυκνότητας πιθανότητας**. Ο ορισμός μίας διακριτής κατανομής που διαθέτει  $T$  στοιχεία, είναι για κάθε στοιχείο η πιθανότητα ( $p_i$ ) να επιλεγεί βρίσκεται

μεταξύ του διαστήματος τιμών  $[0, 1]$ . Το άθροισμα όλων των τιμών των πιθανοτήτων που δίνονται στα μέλη του πληθυσμού είναι ίσες με την μονάδα(1), δηλαδή,  $\sum_{i=1}^N f_i = 1$ , όπου  $f_i$  η τιμή καταλληλότητας του  $i$  μέλους του πληθυσμού.

Χρησιμοποιούμε την κατανομή για να μετατρέψουμε την τιμή καταλληλότητας του κάθε μέλους στην πιθανότητα που έχει για να επιλεγεί αυτό το μέλος. Η τιμή της πιθανότητας επιλογής, υπολογίζεται συναρτήσει της τιμής καταλληλότητας που διέθεταν τα μέλη. Σε αυτή την μέθοδο επιλογής τα μέλη του πληθυσμού ταξινομούνται σε ένα διάνυσμα ανάλογα με την τιμή της πιθανότητας. Η ταξινόμηση που ακολουθούμε στο διάνυσμα, ανάμοια με τις προηγούμενες μεθόδους που περιγράψαμε, είναι φθίνουσα, έτσι  $f_i \geq f_j$  για  $i < j$ . Δηλαδή το καταλληλότερο μέλος για επιλογή είναι το πρώτο μέλος ενώ το χειρότερο είναι το τελευταίο.

Παραδείγματα Τυπικών Κατανομών είναι η γραμμική, η αρνητική εκθετική, η τριγωνική κ.α. Η Γραμμική κατανομή έχει τη μορφή:  $p_i = a * i + b$ , (με το  $a < 0$ ), τυπικής αρνητική εκθετικής κατανομής μορφής  $p_i = a * \exp(b * i + c)$ .



Σχήμα 3<sup>ο</sup>: Απεικόνιση τυπικής γραμμικής κατανομής μορφής  $p_i = a * i + b$ , (με το  $a < 0$ ). Και: Απεικόνιση τυπικής αρνητική εκθετικής κατανομής μορφής  $p_i = a * \exp(b * i + c)$ .

Αυτό είναι ισοδύναμο με την επιλογή της πρώτης μονάδας με πιθανότητα  $p$  της δεύτερης με  $p^2$  της τρίτης με  $p^3$  κ.ο.κ. (σημείωση  $p^N < p^{N-1}$  για  $p < 1$ )

Η πλέον απλούστερη περίπτωση κατανομής την οποία και θα εξετάσουμε είναι ο μετασχηματισμός της τιμής καταλληλότητας των μελών ενός πληθυσμού σε πιθανότητα επιλογής τους, με την διαίρεση κάθε τιμής καταλληλότητας, προς το άθροισμα όλων των καταλληλοτήτων. Δηλαδή,  $f_i / \sum_{i=1}^N f_i$ .

Έστω για παράδειγμα ότι έχουμε έναν πληθυσμό  $N = 5$  μελών. Οι τιμές καταλληλότητας των μελών βρίσκονται τοποθετημένες σε ένα διάνυσμα:

6	5	4	3	2
---	---	---	---	---

Το άθροισμα του συνόλου των καταλληλοτήτων των μελών είναι  $\sum_{i=1}^N f_i = 20$ . Για την μετατροπή της καταλληλότητας σε πιθανότητα επιλογής, θα χρησιμοποιήσουμε τον τύπο που περιγράψαμε προηγουμένως  $f_i / \sum_{i=1}^N f_i$ . Μετά από την εφαρμογή του τύπου σε όλα τα στοιχεία του πληθυσμού, η μορφή που θα έχουν στο διάνυσμα είναι:

0,3	0,25	0,2	0,15	0,1
-----	------	-----	------	-----

Τα αποτελέσματα που δεχθήκαμε από την εφαρμογή του παραπάνω τύπου αποδεικνύουν τον ισχυρισμό που κάναμε παραπάνω για τις ιδιότητες των κατανομών. Πράγματι όλα τα αποτελέσματα που υπολογίσαμε βρίσκονται μέσα στο διάστημα τιμών  $[0, 1]$ , και το άθροισμα τους είναι μονάδα.

Τα πλεονεκτήματα που διαθέτει η μέθοδος της Βαθμίδας Επιλογής είναι ότι δεν αντιμετωπίζουμε τα προβλήματα που συναντήσαμε στην μέθοδο επιλογής ανάλογης της καταλληλότητας, όπως είχαμε εξηγήσει το πρόβλημα της Προώρου Σύγκλισης και το πρόβλημα της Στασιμότητας. Ένα επιπλέον σημαντικό πλεονέκτημα που διαθέτει αυτή η μέθοδος επιλογής είναι ότι δεν χρειάζεται να δώσουμε μεγάλη βαρύτητα στον υπολογισμό της τιμής καταλληλότητας των μελών του πληθυσμού. Σε αυτή την μέθοδο επιλογής χρησιμοποιούμε την καταλληλότητα μόνο για να συγκρίνουμε και να ταξινομήσουμε σωστά τα μέλη. Έτσι μπορούμε να δημιουργήσουμε πιο γρήγορες και ευέλικτες τις διαδικασίες της Κωδικοποίησης και της Αποτίμησης.

Τα μειονεκτήματα που αντιμετωπίζουμε σε αυτή τη μέθοδο επιλογής είναι δύο. Το πρώτο είναι ότι ο πληθυσμός μας χρειάζεται συνεχείς ταξινομήσεις. Η διαδικασία αυτή παίρνει αρκετή ώρα για την σωστή ταξινόμηση των σημείων του πληθυσμού μας. Ένα δεύτερο μειονέκτημα της μεθόδου είναι ότι δεν μπορούμε να κάνουμε θεωρητική ανάλυση της συνθήκης σύγκλισης. Γενικά, δεν αποδεικνύεται ο λόγος για τον οποίο έχουμε κάποια σύγκλιση.[3],[4],[5]

#### **5.4.4. Tournament επιλογή.**

Η τρίτη μέθοδος επιλογής που θα δούμε στην παρούσα ανάλυση είναι η μέθοδος της Tournament επιλογής. Ένα κοινό χαρακτηριστικό που έχουν όλες οι μέθοδοι επιλογής είναι ότι όλες τους είναι εμπνευσμένες από τις διαδικασίες που ακολουθεί η ίδια η φύση. Αυτή η μέθοδος επιλογής μπορεί να θεωρηθεί σαν μία 'θορυβώδης' εκδοχή της μεθόδου Επιλογής Βαθμίδας.

Για να επιλέξουμε κάποιο μέλος με αυτή την μέθοδο, δημιουργούμε ομάδες από  $N$  άτομα (το  $N$  – πλήθος των μελών κάθε ομάδας είναι μεγαλύτερο ίσο του 2). Οι ομάδες δεν επιλέγονται με κάποιο συγκεκριμένο κριτήριο αξιολόγησης. Κάθε ομάδα δημιουργείται τυχαία, αρκεί να τηρείται ο περιορισμός που αναφέραμε προηγουμένως, δηλαδή το πλήθος των μελών κάθε ομάδας πρέπει να είναι περισσότερα ή το ελάχιστο δύο. Από κάθε ομάδα που δημιουργούμε επιλέγουμε μόνο ένα στοιχείο, πάντα επιλέγεται αυτό με την μεγαλύτερη τιμή καταλληλότητας. Το στοιχείο που επιλέγεται τοποθετείται στον ενδιάμεσο πληθυσμό. Τα υπόλοιπα μέλη που συμμετείχαν στην ομάδα αφαιρούνται (tournament). Το πλήθος των ομάδων που θα δημιουργήσουμε εξαρτάται από τον αριθμό των μελών που θέλουμε να βάλουμε στον ενδιάμεσο πληθυσμό. Σε αυτή την μέθοδο μπορούν να συμμετέχουν στον ενδιάμεσο πληθυσμό τα ίδια άτομα περισσότερες από μία φορές.

Τα πλεονεκτήματα που έχει αυτή η μέθοδος επιλογής είναι ότι δεν ερχόμαστε αντιμέτωποι με το πρόβλημα της πρόωρης σύγκλισης ούτε της στασιμότητας που αντιμετωπίζαμε με την επιλογή ανάλογης της καταλληλότητας. Επίσης, όπως και στην μέθοδο επιλογής Βαθμίδας, δεν είμαστε υποχρεωμένοι να υπολογίσουμε με μεγάλη ακρίβεια την τιμή καταλληλότητας των μελών. Σε αυτή την μέθοδο επιλογής χρησιμοποιούμε την καταλληλότητα για να διακρίνουμε αν κάποιο μέλος είναι καλύτερο από κάποιο άλλο, δεν μας απασχολεί κατά πόσο καλύτερο είναι για να του δώσουμε περισσότερες ή λιγότερες ευκαιρίες επιλογής.

Τα πλεονεκτήματα που αναφέραμε παραπάνω έχουν αναφερθεί και στην μέθοδο επιλογής Βαθμίδας. Το πλεονέκτημα που διαφοροποιεί αυτή την μέθοδο από την προηγούμενη και υπερέχει έναντι αυτής είναι ότι δεν απαιτείται να επαναταξινομήσουμε τα μέλη. Για μεγαλύτερη ακρίβεια, δεν απαιτείται καθόλου ταξινόμηση των μελών για να συμμετάσχουν σε κάποια ομάδα. Η μέθοδος αυτή διαλέγει τους καλύτερους από κάθε ομάδα και είναι η γρηγορότερη μέθοδος επιλογής που έχει αναπτυχθεί. [3],[4],[5]

#### **5.4.5. Ελιτίστικη Επιλογή.**

Η τελευταία μέθοδος που θα εξετάσουμε είναι η Ελιτίστικη Επιλογή. Αυτή η μέθοδος είναι τροποποίηση όλων των υπολοίπων μεθόδων, δεν είναι μία μέθοδος επιλογής η οποία να είναι εμπνευσμένη από την φύση. Η Ελιτίστικη Επιλογή δεν ακολουθεί τα πρότυπα της φύσης, επειδή κατά κάποιον τρόπο επιβάλλει στον μηχανισμό Επιλογής του γενετικού αλγορίθμου να επιλέξει κάποιο στοιχείο του πληθυσμού που ίσως να μην επιλεγόταν. Συνήθως η ελιτίστικη επιλογή χρησιμοποιείται σαν προσθήκη στις παραπάνω μεθόδους επιλογής. Η Ελιτίστικη Επιλογή μας εξασφαλίζει ότι το ελάχιστο ένα αντίγραφο από τα καλύτερα μέλη του πληθυσμού πάντοτε περνάνε στην καινούργια γενιά.

Γενικά, επιλέγουμε να χρησιμοποιούμε αυτή την μέθοδο σαν συμπληρωματική στις υπόλοιπες, επειδή μας εξασφαλίζει ότι θα επιβιώσουν οι καλύτεροι, σε ένα ή περισσότερα αντίγραφα, που εμφανίζονται σε κάθε γενιά

Το πλεονέκτημα της μεθόδου της ελιτίστικης επιλογής είναι ότι η συνθήκη σύγκλισης είναι πάντοτε εγγυημένη. Αν ανακαλυφθεί το ολικό μέγιστο ή ελάχιστο σημείο της συνάρτησης που μελετάμε, ο γενετικός αλγόριθμος συγκλίνει σε αυτό. Επομένως, δεν τίθεται περίπτωση να χαθεί η καλύτερη λύση σε κάποιο ενδιάμεσο βήμα.

Το μειονέκτημα που έχει αυτή η μέθοδος είναι ο υπαρκτός κίνδυνος να παγιδευτούμε σε τοπικά μέγιστα ή ελάχιστα της συνάρτησης την οποία εξετάζουμε. Ένας τρόπος αντιμετώπισης είναι να κρατήσουμε την καλύτερη δομή που προέκυψε κατά την διάρκεια εκτέλεσης του αλγορίθμου σε κάποιο μητρώο. Στο τέλος της εκτέλεσης του, καθορίζεται ποια δομή είναι η καλύτερη για την επίλυση του προβλήματος, εν' αντιθέσει να χρησιμοποιήσουμε τα καταλληλότερα μέλη από την τελευταία γενιά. [3],[4],[5]

### **6. Τελεστές Ανασυνδυασμού Γονιδίων.**

Σε όλα τα στάδια που περιγράψαμε μέχρι στιγμής, αναφερθήκαμε για αλλαγές και για τροποποιήσεις που μπορεί να υποστεί ένα μέλος του πληθυσμού, χωρίς όμως να μπορούμε σε περαιτέρω ανάλυση του είδους και της μορφής που μπορούν να έχουν αυτές οι τροποποιήσεις.

Σε αυτή την ενότητα θα κατονομάσουμε τους Γενετικούς Τελεστές Ανασυνδυασμού Γονιδίων παρουσιάζοντας τις ιδιότητες τους και τις διεργασίες που εκτελούν. Πέρα από την μορφή που μπορεί να έχουν οι τροποποιήσεις ενός γονιδίου θα εξετάσουμε ορισμένες διαφοροποιήσεις που μπορεί να υποστεί ένας γενετικός αλγόριθμος κατά την διάρκεια της εξέλιξης του, στα βήματα που τον αποτελούν.

Οι γενετικοί αλγόριθμοι χρησιμοποιούν τους Γενετικούς Τελεστές Ανασυνδυασμού Γονιδίων, για να τροποποιήσουν τα άτομα του τρέχοντος πληθυσμού με σκοπό να τα μετεξελέξουν σε νέα μέλη πιθανόν καταλληλότερα για την επίλυση του προβλήματος μας. Οι τελεστές για τους οποίους γίνεται λόγος είναι ο τελεστής της Διασταύρωσης και ο τελεστής της Μετάλλαξης. Οι τελεστές της Διασταύρωσης και της Μετάλλαξης μπορεί να εφαρμόζονται σε έναν γενετικό αλγόριθμο συνδυαστικά, όπως απεικονίσαμε στο σχήμα 1<sup>ο</sup>. Η σειρά με την οποία θα επιλέξουμε να τους εφαρμόσουμε κατά την διάρκεια της εκτέλεσης

του αλγορίθμου εξαρτάται από εμάς και από τον σκοπό που έχουμε να επιτύχουμε. Μπορούμε όμως να τους χρησιμοποιήσουμε και απομονωμένα, δηλαδή σε ένα πρόβλημα να επιλέξουμε να χρησιμοποιήσουμε μόνο τους τελεστές της μετάλλαξης ή της διασταύρωσης.

Με αυτούς τους τελεστές επιτυγχάνουμε διαφορετικά πράγματα. Για παράδειγμα με την εφαρμογή των τελεστών της Διασταύρωσης επιφέρουμε μεγάλες αλλαγές στον πληθυσμό που διαθέτουμε. Σε αυτούς τους τελεστές χρησιμοποιούμε περισσότερα του ενός μέλη του πληθυσμού. Ένα μέλος για να συμμετάσχει στην διαδικασία της διασταύρωσης πρέπει πρώτα να έχει Επιλεγεί από τον συνολικό πληθυσμό. Από την άλλη πλευρά ο τελεστής της Μετάλλαξης επιφέρει μικρότερης κλίμακας αλλαγές στον πληθυσμό μας. Ο τελεστής αυτός εφαρμόζεται σε τυχαία μέλη του πληθυσμού και δεν συμμετέχουν περισσότερα του ενός μέλη στην διαδικασία. Επίσης, την Μετάλλαξη την χρησιμοποιούμε για να μπορέσουμε να ανακτήσουμε κάποια χαμένη πληροφορία ή κάποια χαρακτηριστικά προγενέστερων πληθυσμών.

Πολλά από τα παραπάνω στοιχεία που αναφέρουμε για τους τελεστές ανασυνδυασμού γονιδίων έχουμε αναφέρει καθ' όλη την διάρκεια της ανάλυσης μας, όπου κρινόταν απαραίτητο για την σφαιρικότερη κάλυψη των θεμάτων που αναλύσαμε. Σε αυτό το σημείο επιδιώκουμε να τα συνοψίσουμε σε λίγες γραμμές και ίσως να τονίσουμε κάποια που τυχόν να πέρασαν απαρατήρητα.

### **6.1. Οι τελεστές της Διασταύρωσης στους Γενετικούς Αλγορίθμους.**

Η διαδικασία της Διασταύρωσης ταυτίζεται με την διαδικασία της φυσικής αναπαραγωγής. Στην αναπαραγωγή, όπως γίνεται στην φύση, τελείται αμοιβαία ανταλλαγή μέρους του DNA μεταξύ μη συγγενών ζευγαριών χρωμοσωμάτων. Παρόμοια, στους γενετικούς αλγορίθμους κατά την διάρκεια της εφαρμογής των τελεστών της διασταύρωσης εκτελείται μία διαδικασία ανάλογη της αντιγραφής, μέρους του γενετικού υλικού μεταξύ των συμμετεχόντων μελών. Έτσι έχουμε την διάδοση χαρακτηριστικών και ιδιοτήτων που διέθεταν οι προγενέστεροι προς τους απόγονους οι οποίοι φαίνεται να κληρονομούν ορισμένα στοιχεία τους από προηγούμενες γενεές, όπως συμβαίνει και στα βιολογικά συστήματα.

Με τους τελεστές της Διασταύρωσης γίνεται η πράξη του ζευγαρώματος των μελών που έχουν επιλεγεί να συμμετάσχουν στον ενδιάμεσο πληθυσμό ή όπως εναλλακτικά τον έχουμε αναφέρει ως 'mating pool'. Οι τελεστές της Διασταύρωσης εφαρμόζεται πάντα μεταξύ δύο μελών, τους γονείς. Κατά την διάρκεια της εφαρμογής του τελεστή της διασταύρωσης έχουμε την αντιγραφή κάποιου τυχαίου μέρους της πληροφορίας και των χαρακτηριστικών μεταξύ των συμμετεχόντων μελών. Τους τελεστές της Διασταύρωσης τους χρησιμοποιούμε για να επιφέρουμε μεγάλης κλίμακας αλλαγές στο πληθυσμό μας. Αυτό το επιτυγχάνουν με την ανταλλαγή των τιμών των μεταβλητών, ανάμεσα στα διάφορα μέλη που διασταυρώνονται.

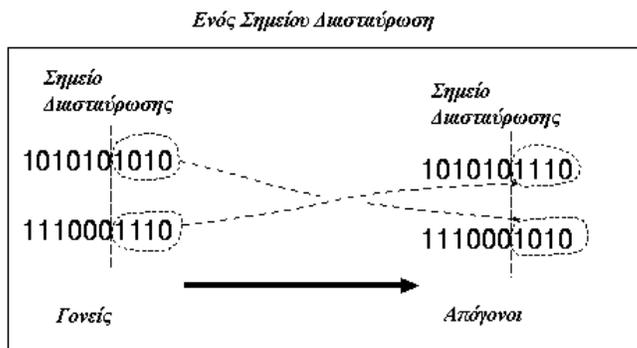
Όταν αναφερόμασταν στη διασταύρωση, μιλούσαμε για τελεστές Διασταύρωσης αντί για τελεστή Διασταύρωσης. Αυτή η διαφοροποίηση οφείλεται στο γεγονός ότι υπάρχουν διαφορετικοί τρόποι για να εκτελέσουμε αυτό τον τελεστή. Παρακάτω αναφέρουμε αυτούς τους διαφορετικούς τρόπους κάνοντας μία αρχική διάκριση που υπάρχει μεταξύ τους σε *ομοιόμορφους* και *μη*. Σε πρώτη φάση θα μελετήσουμε δύο τρόπους ομοιόμορφης Διασταύρωσης :

### 6.1.1. Διασταύρωση ενός σημείου.

Στην διασταύρωση ενός σημείου έχουμε την τυχαία επιλογή ενός σημείου διασταύρωσης στα χρωμοσώματα των μελών που συμμετάσχουν. Τυπικά αυτά τα μέλη τα ονομάζουμε γονείς, ενώ το νέο ή τα νέα μέλη που θα προκύψουν τα αποκαλούμε απογόνους ή παιδιά. Η διαδικασία γίνεται ως εξής : Στην πιο συνήθη περίπτωση διαλέγουμε μέλη που έχουν περάσει από την διαδικασία της Επιλογής και έχουν καταχωρηθεί στον ενδιάμεσο πληθυσμό. Μπορεί όμως σε διαφορετικές εκδοχές των γενετικών αλγορίθμων να μην δημιουργούμε κάποιον ενδιάμεσο πληθυσμό. Σε αυτές τις περιπτώσεις διασταυρώνουμε τα μέλη που επιλέγουμε και τα αντικαθιστούμε ή προσθέτουμε τα μέλη που προέκυψαν από την διαδικασία απευθείας στον υπάρχον πληθυσμό.

Από τα μέλη που έχουν επιλεγεί να συμμετάσχουν στην Διασταύρωση, αποκοκόπτουμε το δεξιότερο τμήμα των χρωμοσωμάτων τους. Το σημείο από το οποίο θα ξεκινήσουμε να αποκόπτουμε είναι κάποιο σημείο που επιλέγεται τυχαία, μέχρι το τέλος της δομής δεδομένων τους.

Παρακάτω στο σχήμα 5<sup>ο</sup> απεικονίζουμε τον τρόπο διασταύρωσης μελών που έχουν δυαδική αναπαράσταση.



Σχήμα 4<sup>ο</sup> : Διασταύρωση ενός σημείου.

Ένα άλλο παράδειγμα με την βοήθεια του οποίου θα περιγράψουμε σταδιακά την ίδια διαδικασία διασταύρωσης ενός σημείου δίνεται παρακάτω : Έστω ότι έχουμε δύο γονίδια που τα 'χρωμοσώματα' τους αποτελούνται από τρεις ακέραιες μεταβλητές :

$$\begin{aligned} \text{Γονίδιο 1}^\circ &: 2, 3, \uparrow 5 \\ \text{Γονίδιο 2}^\circ &: 5, 9, \downarrow 7 \end{aligned}$$

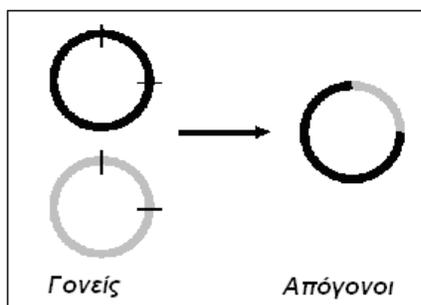
Έστω ότι η θέση διασταύρωσης που επιλέγεται τυχαία και στα δύο μέλη είναι η θέση που υποδεικνύει το βέλος. Όπως έχουμε ήδη εξηγήσει και απεικονίσει την διαδικασία, το δεξιότερο μέρος των χρωμοσωμάτων αποκόπτεται και ανταλλάσσεται μεταξύ τους. Το αποτέλεσμα στο παράδειγμα που αναφέραμε θα είναι τα νέα μέλη να φέρουν την πληροφορία :

$$\begin{aligned} \text{Γονίδιο 1}^\circ &: 2, 3, 7 \\ \text{Γονίδιο 2}^\circ &: 5, 9, 5 \end{aligned}$$

Η Διασταύρωση ενός σημείου είναι η πλέον απλούστερη και πιο ευρέως εφαρμοζόμενη διαφοροποίηση εκτέλεσης του τελεστή της Διασταύρωσης. Έτσι επιτυγχάνουμε την αμοιβαία ανταλλαγή γενετικού υλικού μεταξύ των γονέων μέσα από την απλή αντιμετάθεση μέρους τις πληροφορίας που διαθέτουν. Παρακάτω θα μελετήσουμε έναν διαφορετικό τρόπο εκτέλεσης της διαδικασίας της Διασταύρωσης, την Διασταύρωση δύο σημείων.

### 6.1.2. Διασταύρωση δύο σημείων.

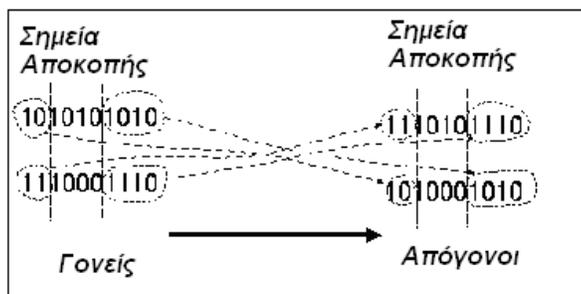
Στην διασταύρωση δύο σημείων μας βοηθάει να φανταζόμαστε τα χρωμοσώματα των μελών κυκλικά σαν δακτυλίδι. Στην διασταύρωση δύο σημείων ορίζουμε δύο τυχαία σημεία αποκοπής στα χρωμοσώματα των συμμετεχόντων μελών. Παρόμοια με την διασταύρωση ενός σημείου, με αυτόν τον τρόπο τα υπό – τμήματα των μελών αποκόπτονται και ανταλλάσσονται μεταξύ τους. Σε αυτήν την διαφοροποίηση του τελεστή της Διασταύρωσης ορίζουμε το σημείο Αρχής και Τέλους του τμήματος προς ανταλλαγή. Σε αυτό το σημείο η οπτική απεικόνιση της Διασταύρωσης δύο σημείων σαν ένα δακτυλίδι θα μας βοηθούσε να εξηγήσουμε στον αναγνώστη.



Σχήμα 5<sup>ο</sup> : Παρουσίαση των χρωμοσωμάτων σε σχήμα δακτυλιδιού.

ή παρόμοια με την προηγούμενη αναπαράσταση :

#### Διασταύρωση δύο Σημείων.



Σχήμα 6<sup>ο</sup> : Διασταύρωση δύο Σημείων.

Στο σχήμα 6<sup>ο</sup> απεικονίζουμε την διαδικασία της Διασταύρωσης με δύο σημεία αποκοπής, τα μέλη που συμμετάσχουν έχουν δυαδική αναπαράσταση. Η διαφορά που έχουν οι δύο τρόποι Διασταύρωσης μεταξύ τους είναι ευκρινέστερη, δηλαδή σε αυτόν τον τρόπο τα δύο τυχαία σημεία αποκοπής ορίζουν το αρχικό και το τελικό σημείο διασταύρωσης, σε αντίθεση με την διασταύρωση ενός σημείου. Με αυτόν τον τρόπο ορίζουμε ολόκληρο το τμήμα που θα αντιγραφεί μεταξύ των μελών.

Θα συνεχίσουμε με το προηγούμενο παράδειγμα για να περιγράψουμε σταδιακά την διαδικασία διασταύρωσης δύο σημείων. Έστω ότι έχουμε δύο γονίδια που τα 'χρωμοσώματα' τους αποτελούνται από τρεις ακέραιες μεταβλητές :

$$\begin{aligned} \text{Γονίδιο 1}^{\circ} &: 2 \uparrow^1, 3 \uparrow^2, 5 \\ \text{Γονίδιο 2}^{\circ} &: 5 \downarrow, 9 \downarrow, 7 \end{aligned}$$

Οι θέσεις που επιλέχθηκαν για να γίνει η διασταύρωση υποδεικνύονται από τα δύο βέλη με τον αντίστοιχο δείκτη αρχής και τέλους του επιλεγμένου προς διασταύρωση τμήματος. Όπως είναι αναμενόμενο οι αλλαγές που μπορεί να γίνουν στα γονίδια, είναι περισσότερες επομένως και ο αριθμός

των πιθανών νέων μελών είναι αυξημένος. Ορισμένα από τα αποτελέσματα στο παράδειγμα που αναφέραμε θα είναι :

Γονίδιο 1° : 5 , 3 , 7

Γονίδιο 2° : 2 , 9 , 5

Γονίδιο 3° : 2 , 9 , 5

Γονίδιο 4° : 5 , 3 , 7

Οι δύο παραπάνω τρόποι που περιγράψαμε μας επιστρέφουν ισομεγέθη απογόνους, αυτός είναι ο λόγος για τον οποίο της ονομάζουμε **Ομοιόμορφες Μεθόδους Διασταύρωσης**. Υπάρχει μία διαφορετική μέθοδος διασταύρωσης που μας δίνει διαφορετικού μεγέθους απογόνους.

### 6.1.3. Μη Ομοιόμορφη Διασταύρωση.

Παρατηρούμε ότι οι υποψήφιοι που συμμετείχαν σε όλα τα παραδείγματα που αναφέραμε, τα χρωμοσώματα τους αποκόπτονταν στο ίδιο σημείο. Με αυτόν τον τρόπο επιλογής των σημείων αποκοπής δημιουργούμε συνεχώς μέλη με το ίδιο μήκος χρωμοσωμάτων. Γενικά με τους δύο παραπάνω τρόπους Διασταύρωσης έχουμε 'Ομοιόμορφη Διασταύρωση'. Μέσα από αυτή την διαδικασία προκύπτουν μέλη με χρωμοσώματα σταθερού μήκους με αυτά των γονέων τους. Μπορεί όμως να συναντήσουμε και περιπτώσεις που τα χρωμοσώματα των γονέων έχουν διαφορετικό σημείο αποκοπής. Ας εξετάσουμε αυτό το ενδεχόμενο μέσα από ένα παράδειγμα.

Έστω ότι έχουμε τα παρακάτω γονίδια που θα συμμετέχουν στην Διασταύρωση :

Γονίδιο 1° : 2 ,  $\uparrow$  3 , 5 (Σημείο Διασταύρωσης το πρώτο στοιχείο.)

Γονίδιο 2° : 5 , 9 ,  $\downarrow$  7 (Σημείο Διασταύρωσης το δεύτερο στοιχείο.)

Τα μέλη που θα προκύψουν θα είναι μέλη με διαφορετικού μήκους χρωμοσώματα, εφόσον αποκόπτουμε τους γονείς σε διαφορετικά σημεία διασταύρωσης και ανταλλάσσουμε διαφορετικού μεγέθους τμήμα πληροφορίας. Τα νέα μέλη που εξάγονται από την διαδικασία αυτή θα είναι :

Γονίδιο 1° : 2 , 7

Γονίδιο 2° : 5 , 9 , 3 , 5

Συνήθως οι απόγονοι που παράγονται μέσα από την μη ομοιόμορφη Διασταύρωση είναι μέλη με διαφορετικού μήκους χρωμοσώματα, χωρίς να αποκλείεται το ενδεχόμενο να επιλεγούν τυχαία κοινά σημεία Αποκοπής.

Χρησιμοποιούμε αυτήν την μέθοδο διασταύρωσης επειδή υπάρχει η πιθανότητα οι απόγονοι που θα προκύψουν μέσα από αυτή την διασταύρωση να είναι πιο δυνατοί από αυτούς που θα προέκυπταν με την ομοιόμορφη διασταύρωση. Αυτός ο τρόπος Διασταύρωσης δεν είναι βιολογικά αποδεκτός και συμβατός με αυτούς που επιτάσσει η φύση. Παρόλα αυτά είναι δυνατόν να συμβούν και στην ίδια την φύση, όμως τα νέα μέλη που θα προκύψουν μέσα από τις διαδικασίες αυτές στην βιολογία θα τις χαρακτηρίζαμε ως τερατογενέσεις. [3],[5],[7],[6]

### 6.1.4. Πιθανότητα Διασταύρωσης & Αντιγραφή.

Το μέτρο με το οποίο θα χρησιμοποιηθούν οι τελεστές της Διασταύρωσης στον πληθυσμό ενός γενετικού αλγορίθμου δίνεται από τον παράγοντα **Πιθανότητα Διασταύρωσης**. Η πιθανότητα διασταύρωσης

συμβολίζεται με το γράμμα  $p_c$  και ορίζεται να ανήκει σε κάποιο διάστημα για παράδειγμα της μορφής  $p_c \in [0,5, 0,8]$ .

Με την Πιθανότητα Διασταύρωσης δηλώνουμε σε έναν γενετικό αλγόριθμο το ποσοστό των μελών του πληθυσμού της παρούσης γενιάς που θέλουμε να αντικαταστήσουμε, με στοιχεία που προήλθαν από την διαδικασία της διασταύρωσης. Τα νέα μέλη που προκύπτουν από την εφαρμογή του τελεστή είτε θα αντικαταστήσουν, είτε θα προστεθούν στον υπάρχον πληθυσμό, αυτό εξαρτάται από την πολιτική που ακολουθείται από τον αλγόριθμο.

Έστω για παράδειγμα ότι διαθέτουμε έναν πληθυσμό  $N = 3$  μέλη, η δομή δεδομένων τους φαίνεται παρακάτω. Εφαρμόζουμε τον τελεστή της διασταύρωσης ενός σημείου, όπως ακριβώς στο παράδειγμα που αναφέραμε σε εκείνο το κομμάτι της ανάλυσης μας και θα δανειστούμε τα αποτελέσματα του, επίσης προσθέτουμε ένα επιπλέον γονίδιο. Ο νέος πληθυσμός που θα προκύψει είναι ο ακόλουθος :

*Αρχικός Πληθυσμός :*  
Γονίδιο 1° : 2 , 3 , 5  
Γονίδιο 2° : 5 , 9 , 7  
Γονίδιο 3° : 1 , 9 , 2

Μετά από την εφαρμογή της διαδικασίας της Διασταύρωσης και με πιθανότητα διασταύρωσης δοσμένη ίση με  $p_c = 2/3$  . Σύμφωνα με τον λόγο της πιθανότητας  $p_c$  τα δύο τρίτα του τρέχοντος πληθυσμού θα πρέπει είτε να τα αντικαταστήσουμε είτε να τα προσθέσουμε στον ήδη υπάρχον πληθυσμό. Εφόσον η πολιτική που ακολουθούμε για τον καινούργιο πληθυσμό είναι αυτή της αντικατάστασης των γονέων από τα νέα παραγόμενα μέλη, ο νέος πληθυσμός θα διαμορφωθεί ως :

*Επόμενος Πληθυσμός :*  
Γονίδιο 1° : 2 , 3 , 7  
Γονίδιο 2° : 5 , 9 , 5  
Γονίδιο 3° : 1 , 9 , 2

Παρατηρούμε ότι το τρίτο μέλος του τρέχοντος πληθυσμού, περνάει στον επόμενο πληθυσμό χωρίς να υποστεί καμία τροποποίηση από την διαδικασία της δημιουργίας ενός νέου πληθυσμού. Το γεγονός αυτό το ονομάζουμε Cloning – Αντιγραφή, και δηλώνει το πέρασμα ορισμένων μελών του πληθυσμού από μία γενιά σε επόμενες χωρίς να υποστούν καμία απολύτως αλλαγή στο γενετικό τους υλικό.

Το πλήθος των μελών του πληθυσμού που θα περάσουν από την μία γενιά στην επόμενη, χωρίς να περάσουν την διαδικασία της διασταύρωσης δίνεται από την πιθανότητα  $1 - p_c$  . Για αυτό το πλήθος των μελών που θα περάσουν με απευθείας αντιγραφή από τον έναν πληθυσμό σε κάποιον επόμενο, μπορούμε να κάνουμε Ελιτισμό. Για το ποσοστό  $1 - p_c$  μπορούμε να επιλέξουμε τους καλύτερους του τρέχοντος πληθυσμού.

Αν για παράδειγμα στον αρχικό πληθυσμό είχαμε υπολογίσει τις τιμές καταλληλότητας 10 , 5 , 2 για το πρώτο, δεύτερο, τρίτο γονίδιο αντίστοιχα, και για το ποσοστό  $1 - (2/3)$  κάναμε Ελιτισμό, τότε ο επόμενος πληθυσμός θα είχε την εξής μορφή :

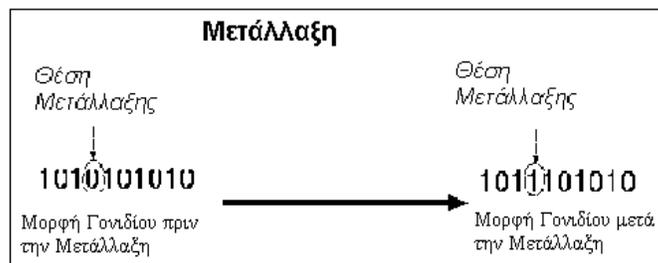
*Επόμενος Πληθυσμός :*  
Γονίδιο 1° : 2 , 3 , 7  
Γονίδιο 2° : 5 , 9 , 5

## 6.2. Οι τελεστές της Μετάλλαξης στους Γενετικούς Αλγορίθμους.

Στην γενετική όταν αναφερόμαστε στον όρο της ‘Μετάλλαξης’, αναφερόμαστε σε κάποια γενετική ανωμαλία. Η μετάλλαξη στην φύση μπορεί να συμβεί κατά την διάρκεια της φάσης της αντιγραφής του γενετικού υλικού. Οι γενετικές ανωμαλίες οφείλονται στην μικρή τροποποίηση που μπορεί να υποστεί το γενετικό υλικό. Αυτή είναι η λειτουργία της Μετάλλαξης στους γενετικούς αλγορίθμους. Ο τελεστής της Μετάλλαξης προκαλεί συνήθως μικρές αλλαγές στην τιμή ενός ή περισσότερων μεταβλητών του μέλους ή του γονιδίου του πληθυσμού. Τον τελεστή της Μετάλλαξης τον χρησιμοποιούμε στους γενετικούς αλγορίθμους προκειμένου να ανακτήσουμε τυχόν χαμένη πληροφορία σε κάποιο γονίδιο.

Αυτός ο τρόπος είναι ο καλύτερος και ευκολότερος από την αναζήτηση των καταλληλότερων λύσεων σε κάποιο πρόβλημα που αντιμετωπίζουμε. Επειδή, δεν απαιτείται να προηγηθούν οι πολύπλοκες και χρονοβόρες διαδικασίες της Επιλογής και της δημιουργίας ενός ενδιάμεσου πληθυσμού. Ο τελεστής της Μετάλλαξης εφαρμόζεται τυχαία στα μέλη του πληθυσμού. Ένας επιπλέον λόγος που επιλέγουμε να χρησιμοποιήσουμε την Μετάλλαξη στους γενετικούς αλγορίθμους είναι για να επιφέρουμε μικρές αλλαγές στο πληθυσμό.

Στην διάρκεια εκτέλεσης ενός γενετικού αλγορίθμου μπορούμε να τον χρησιμοποιήσουμε είτε μαζί με τους τελεστές της Διασταύρωσης συνδυαστικά, είτε αποκλειστικά και μόνο αυτόν, ή ακόμη και καθόλου. Ένα εύκολο παράδειγμα που μπορούμε να χρησιμοποιήσουμε προκειμένου να μας εισάγει στην φιλοσοφία του για να δούμε το είδος των τροποποιήσεων που προκαλεί, είναι στην περίπτωση που η δομή δεδομένων των χρωμοσωμάτων των μελών είναι δυαδικά κωδικοποιημένα. Για παράδειγμα :



Σχήμα 7<sup>ο</sup> : Μετάλλαξη. Μέλος πριν και μετά την εφαρμογή του τελεστή μετάλλαξης.

Σε αυτή την περίπτωση ο τελεστής της Μετάλλαξης μπορεί να επιφέρει αντιστροφή της τιμής του bit, και από 0 κάποιο bit να γίνει 1. Πιο συγκεκριμένα, στους γενετικούς αλγορίθμους η Μετάλλαξη σημαίνει : αλλαγή κάποιας μεταβλητής σε κάποιο μέλος του πληθυσμού μας, στοχαστικά.

Ας φέρουμε ένα πιο συγκεκριμένο παράδειγμα για να δείξουμε την στοχαστικότητα της διαδικασίας. Για κάθε στοιχείο του πληθυσμού ρίχνουμε ένα ζάρι. Έστω ότι από το συνολικό πληθυσμό επιλέχθηκε τυχαία με την μέθοδο του ζαριού, ένα γονίδιο :

Έχουμε το γονίδιο με τις μεταβλητές : 1 , 7 , 9 , 3

Αν θα μεταλλαχθεί ή όχι το γονίδιο εξαρτάται από κάποιο κατώτερο όριο. Για κάθε μία από τις μεταβλητές του γονιδίου ρίχνουμε ένα ζάρι. Αν η τιμή του ζαριού είναι πιο κάτω από κάποιο όριο, το οποίο ορίζουμε εμείς, τότε η μεταβλητή του γονιδίου θα υποστεί κάποια αλλαγή.

Έστω ότι το κατώτερο όριο που θέτουμε είναι μία πιθανότητα 10 %. Για το πρώτο στοιχείο του γονιδίου ρίχνουμε ένα ζάρι. Αν για παράδειγμα φέρουμε 0,2 τιμή μεγαλύτερη από αυτή που καθορίσαμε ως την κατώτερη, τότε το πρώτο στοιχείο δεν θα υποστεί κάποια μετάλλαξη. Ακολουθούμε την ίδια διαδικασία για το δεύτερο στοιχείο του γονιδίου, έστω ότι φέρνουμε 0,01 τιμή μικρότερη από αυτή που καθορίσαμε ως την κατώτερη. Αυτό σημαίνει ότι η δεύτερη μεταβλητή θα υποστεί κάποιου είδους μετάλλαξη.

Ο τρόπος με τον οποίο θα γίνει η αλλαγή στην μεταβλητή του γονιδίου πρέπει να είναι αντίστοιχος με το περιεχόμενο της μεταβλητής. Στο πρώτο παράδειγμα, η τιμή της μεταβλητής είναι δυαδική τότε θα υποστεί μία δυαδική μετάλλαξη όπως δείξαμε παραπάνω θα έχουμε αντιστροφή της τιμής του bit, αν μεταβλητή είναι πραγματική τότε θα έχουμε πραγματική μετάλλαξη στην μεταβλητή.

Στο παράδειγμα που αναφέραμε η μετάλλαξη που μπορεί να υποστεί το δεύτερο στοιχείο του μέλους, μπορεί να είναι να αντικαταστήσουμε την τιμή 7 με μία τυχαία ακέραια τιμή που θα παράγουμε χρησιμοποιώντας κάποια συνάρτηση που μας επιστρέφει τυχαίες τιμές. Επίσης, η αλλαγή μπορεί να είναι της μορφής να προσθέσουμε κάποιο θόρυβο στην τιμή της μεταβλητής. Έστω ότι θα επιλέξουμε να προσθέσουμε στην αρχική τιμή 10% θόρυβο, τότε η νέα τιμή που θα εισάγουμε προς αντικατάσταση της παλιάς θα είναι 7,7.

### 6.2.1. Πιθανότητα Μετάλλαξης.

Στους γενετικούς αλγορίθμους ο τελεστής της μετάλλαξης εφαρμόζεται στα μέλη ενός πληθυσμού με πολύ χαμηλή πιθανότητα. Το μέτρο εφαρμογής του τελεστή ονομάζεται Πιθανότητα Μετάλλαξης συμβολίζεται ως  $p_m$ , στις περισσότερες περιπτώσεις η τιμή της πιθανότητας ορίζεται να ανήκει σε ένα κλειστό διάστημα  $p_m \in [0,001, 0,01]$ .

Η επιλογή της τιμής που θα επιλέξουμε για να χρησιμοποιηθεί ο τελεστής της μετάλλαξης είναι πολύ κρίσιμη, καθώς μπορεί να επηρεάσει την ροή της εξέλιξης του γενετικού αλγορίθμου και υπάρχει ο κίνδυνος να μην πάρουμε τα αποτελέσματα που θέλουμε.

Όταν δεν γνωρίζουμε ποια θα πρέπει να είναι αυτή η τιμή ένας καλός κανόνας που μας παρέχει το κατώτερο όριο χρήσης του τελεστή είναι ο παρακάτω :

$$p_m = \frac{1}{\text{bits ανά χρωμόσωμα}}$$

Ο κανόνας αυτός μας εξασφαλίζει ότι το γονίδιο που έχει επιλεγεί για μετάλλαξη, θα υποστεί τουλάχιστον μία τροποποίηση σε ένα από τα στοιχεία που το αποτελούν. Αυτό είναι το κατώτερο επιθυμητό αποτέλεσμα μετάλλαξης σε κάποιο χρωμόσωμα. Στο παράδειγμα που αναλύσαμε παραπάνω η τιμή που θα μας δώσει ο παραπάνω τύπος είναι :  $p_m = 1 / 4 = 0,25$

Δηλαδή σε όλες τις μεταβλητές του παραδείγματος μας δίνουμε την πιθανότητα  $p_m$  να υποστούν κάποια αλλαγή. Μπορούμε αυτή την τιμή να την χρησιμοποιήσουμε ως την ανώτερη ή την κατώτερη τιμή του ορίου. [3],[7],[5]

## 7. Κριτήρια τερματισμού των Γενετικών Αλγορίθμων.

Κάθε αλγόριθμος πρέπει να έχει κάποιο καλά προσδιορισμένο κριτήριο τερματισμού, που να πιστοποιεί ότι ο λόγος για τον οποίο έχει δημιουργηθεί μετά από την εκτέλεση του έχει ικανοποιηθεί και έχει επιφέρει την λύση για το πρόβλημα που τον εφαρμόζουμε.

Τυπικά οι γενετικοί αλγόριθμοι μπορεί να εκτελούνται για πάντα. Συνήθως το πεδίο ορισμού της συνάρτησης που εξετάζουν είναι άπειρο, όπως επίσης και ο αριθμός των παραμέτρων που επηρεάζουν το πρόβλημα μας μπορεί να είναι πολύ μεγάλος. Αυτά τα δύο χαρακτηριστικά που ενδεχομένως να διαθέτει ένα πρόβλημα στο οποίο απασχολούμε έναν γενετικό αλγόριθμο φανερώνουν ως έναν βαθμό την δυσκολία που αντιμετωπίζουμε προκειμένου να ορίσουμε ένα σαφές κριτήριο τερματισμού του αλγορίθμου.

Ένας σημαντικός παράγοντας που επηρεάζει όλους τους αλγορίθμους είναι ο χρόνος που απαιτείται για την εκτέλεση τους. Σε αυτόν τον παράγοντα αναφερθήκαμε ευκαιριακά κατά την διάρκεια της ανάλυσης μας, πράγμα που δεν δείχνει την ιθύνουσα σημασία του. Το μεγάλο πλεονέκτημα των στοχαστικών μεθόδων και σαν προέκταση τους των γενετικών αλγορίθμων είναι ότι μπορούν να εξετάζουν το πεδίο ορισμού μίας συνάρτησης δειγματοληπτώντας σε αυτό και σε ένα εύλογο χρονικό διάστημα να μας επιστρέψουν αποτελέσματα ικανοποιητικής προσέγγισης της πραγματικής λύσης. Σε αντίθεση με άλλες μεθόδους που απαιτούν να εξετάσουν ολόκληρο το πεδίο ορισμού μίας συνάρτησης.

Παρακάτω θα δούμε ορισμένα από τα πιο συνηθισμένα κριτήρια τερματισμού που χρησιμοποιούνται στους γενετικούς αλγορίθμους :

Το πιο απλό κριτήριο τερματισμού ενός γενετικού αλγορίθμου είναι, όταν τα μέλη του πληθυσμού στην τρέχουσα γενιά έχουν παρόμοιες τιμές. Ένα άλλο κριτήριο, είναι να ελέγχουμε την βελτίωση στην συνάρτηση καταλληλότητας, όταν η βελτίωση της είναι κάτω από ένα όριο κατωφλίου, τότε τερματίζεται ο αλγόριθμος.

Τα κριτήρια τερματισμού είναι κάποιες φορές εξειδικευμένα και κατάλληλα μόνο για τον συγκεκριμένο αλγόριθμο, όπου τα χρησιμοποιούμε. Ωστόσο τα γενικότερα κριτήρια τερματισμού είναι : τέλος του αλγορίθμου μετά από ένα προκαθορισμένο πλήθος αποτιμήσεων της συνάρτησης (function's evaluations) ή τερματισμός όταν ο βαθμός βελτίωσης της τρέχουσας βέλτιστης τιμής παραμένει σταθερός για κάποιο προκαθορισμένο αριθμό βημάτων του αλγορίθμου.[2],[3]

Κάθε ένα από όσα κριτήρια τερματισμού αναφέραμε έχει τα δικά του μειονεκτήματα και πλεονεκτήματα έναντι των υπολοίπων. Για παράδειγμα, όταν ορίζουμε τον αλγόριθμο να σταματήσει μετά από ένα συγκεκριμένο αριθμό γενεών, αυτό μας εγγυάται ότι ο αλγόριθμος θα σταματήσει την εκτέλεση του, σε ένα θεμιτό χρονικό διάστημα. Αυτό το κριτήριο τερματισμού μας βολεύει να το χρησιμοποιήσουμε σε προβλήματα που γνωρίζουμε εξ' αρχής την λύση τους, όπως ακριβώς το χρησιμοποιούμε στα δύο πρώτα παραδείγματα που αναλύουμε στο δεύτερο μέρος. Επίσης, με αυτό το μέτρο τερματισμού μπορούμε να δούμε τον βαθμό βελτίωσης του αποτελέσματος που μας επιστρέφει ο αλγόριθμος, αν τον εκτελέσουμε επανειλημμένα για διαφορετικό αριθμό γενεών. Ένα μειονέκτημα που έχει αυτό το κριτήριο είναι ότι ο αριθμός επαναλήψεων κρίνεται προτού ο αλγόριθμος ξεκινήσει να εκτελείται, επομένως για διαφορετικές παραμέτρους θα πρέπει να μεταβάλλουμε τον αριθμό επαναλήψεων. Αν επιλέξουμε έναν πολύ μεγάλο αριθμό επαναλήψεων ενώ ο αλγόριθμος καταλήξει στην βέλτιστη λύση του προβλήματος από τις πρώτες γενιές, τότε θα εκτελείται για τις υπόλοιπες γενιές χωρίς να υπάρχει κάποια ουσιαστική αλλαγή στο αποτέλεσμα.

Αυτό το μειονέκτημα μπορούμε να το αντιμετωπίσουμε θέτοντας σαν κριτήριο τερματισμού τον βαθμό βελτίωσης της τιμής καταλληλότητας. Όμως με αυτό το κριτήριο τερματισμού θα πρέπει να προσέξουμε

τον αριθμό των γενεών για τον οποίο θα πρέπει να μένει ο βαθμός καταλληλότητας σταθερός. Σε αυτό το κριτήριο παραμένει ο φόβος να μην έχουμε για μεγάλα χρονικά διαστήματα σταθερή την καταλληλότητα και ο αλγόριθμος να εκτελείται για πάντα ή να εγκλωβισθεί σε τοπικά μέγιστα ή ελάχιστα και να λάβουμε λανθασμένα και πολύ πρόωρα αποτελέσματα.

Το καλύτερο κριτήριο τερματισμού για την αντιμετώπιση ενός προβλήματος που αντιμετωπίζουμε με τους γενετικούς αλγορίθμους είναι ο συνδυασμός των παραπάνω κριτηρίων τερματισμού επειδή το κάθε ένα από αυτά κατά κάποιο τρόπο καλύπτει τις αδυναμίες του άλλου.

Επίσης η ανάπτυξη ενός κριτηρίου που να ταιριάζει στο πρόβλημα μάς εγγυάται ότι μπορούμε να πάρουμε τα αποτελέσματα που θέλουμε από την εκτέλεση του. Για παράδειγμα, στο πρόβλημα της ελαχιστοποίησης

## ΜΕΡΟΣ ΙΙ

Στο δεύτερο μέρος της παρούσης εργασίας έχουμε ως αντικειμενικό στόχο να εκτελέσουμε μία σειρά πειραμάτων χρησιμοποιώντας τους γενετικούς αλγόριθμους.

Η εκτέλεση των πειραμάτων θα περατωθεί με την βοήθεια της βιβλιοθήκης GALib. Η GALib είναι μία C++ βιβλιοθήκη με αντικείμενα (κατηγορίες) που προορίζονται για να τα χρησιμοποιήσουμε και να αναπτύξουμε κάποιον γενετικό αλγόριθμο. Η βιβλιοθήκη της GALib μας παρέχει αυτό το σύνολο των εργαλείων τα οποία μπορούμε να χρησιμοποιήσουμε σε οποιοδήποτε πρόγραμμα C++, για να βελτιστοποιήσουμε μία διαδικασία.

Σε πρώτη φάση θα παρουσιάσουμε την βιβλιοθήκη της GALib και το σύνολο των εργαλείων που μας παρέχει, παραθέτοντας τις ιδιότητες και τις απαιτήσεις που έχουν προκειμένου να μπορέσουμε να τα χρησιμοποιήσουμε σωστά και αποδοτικά. Ολόκληρο το δεύτερο μέρος έχει γραφτεί με κύριο οδηγό το εγχειρίδιο της βιβλιοθήκης της GALib[8] που βρίσκεται στον δικτυακό τόπο : *lancet.mit.edu/ga/dist/galibdoc.pdf*. Για την περάτωση των πειραμάτων μας θα χρησιμοποιήσουμε ορισμένα από τα εργαλεία της βιβλιοθήκης, που είναι η δεύτερη φάση αυτού του μέρους.

Η βιβλιοθήκη της GALib έχει αναπτυχθεί στο Τεχνολογικό Ινστιτούτο της Massachusetts(U.S.A). Η βιβλιοθήκη δεν είναι δημοσίας κυριότητας, αλλά είναι διαθέσιμη χωρίς κόστος για μη – κερδοσκοπικούς σκοπούς στον δικτυακό τόπο :

[http://lancet.mit.edu/ga/  
galib-request@mit.edu](http://lancet.mit.edu/ga/galib-request@mit.edu)

Η χρήση της για κερδοσκοπικούς σκοπούς απαιτεί ειδική άδεια. Όλα τα στοιχεία της GALib προστατεύονται από την συνθήκη της Berne.

## 1. Η βιβλιοθήκη της GALib.

Η βιβλιοθήκη GALib συμπεριλαμβάνει εργαλεία που μπορούμε να τα χρησιμοποιήσουμε σε όλα τα προγράμματα που είναι υλοποιημένα στη γλώσσα προγραμματισμού C++, καθώς επίσης χρησιμοποιούν κάποια αναπαράσταση ή κάποιον γενετικό τελεστή. Για να χρησιμοποιήσουμε την βιβλιοθήκη της GALib, πρέπει κατά κύριο λόγο να εργαστούμε με δύο κατηγορίες αντικειμένων : την κατηγορία του γονιδίου και την κατηγορία του γενετικού αλγορίθμου.

Η *κατηγορία του γονιδίου* μας επιτρέπει να αναπαραστήσουμε μια λύση του προβλήματος που αντιμετωπίζουμε. Σε κάθε κατηγορία που περιγράφει ένα γονίδιο είναι προσδιορισμένη η δομή δεδομένων του γονιδίου, όπως επίσης και το σύνολο των τελεστών του. Τα χαρακτηριστικά που επιθυμούμε να έχουν οι λύσεις του προβλήματος μας, πρέπει να είναι συμβατά με το σύνολο των πληροφοριών που έχουμε την δυνατότητα να αποθηκεύσουμε στην δομή δεδομένων. Οι τελεστές του γονιδίου είναι αυτοί που μπορούν να χρησιμοποιηθούν από την κατηγορία του γενετικού αλγορίθμου για να εκτελεστούν κάποιες πράξεις, όπως για παράδειγμα αυτή της Διασταύρωσης.

Η *κατηγορία του γενετικού αλγορίθμου* περιγράφει τον τρόπο με τον οποίο πρέπει να εκτελεστεί η διαδικασία εξέλιξης των γονιδίων διαμέσω της ανάπτυξης νέων γενεών. Ο γενετικός αλγόριθμος χρησιμοποιεί μία αντικειμενική διαδικασία, την οποία προσδιορίζουμε εμείς προκειμένου να ορίσουμε ποια γονίδια(περιγραφές λύσεων) ταιριάζουν καλύτερα στην επίλυση του προβλήματος μας και πρέπει να επιβιώσουν στην επόμενη γενιά. Επίσης, χρησιμοποιεί τους τελεστές που βρίσκονται ορισμένοι στην κατηγορία του γονιδίου και περιγράφει στρατηγικές επιλογής και αντικατάστασης μελών ενός πληθυσμού για την δημιουργία ενός νέου. Οι στρατηγικές Επιλογής και Αντικατάστασης, είναι προσδιορισμένες στην κατηγορία του γενετικού αλγορίθμου, σαν σκοπό έχουν να δημιουργήσουν καινούργια σύνολα δειγμάτων στο πεδίο αναζήτησης της λύσης.

Για να χρησιμοποιήσουμε τους γενετικούς αλγορίθμους στην επίλυση προβλημάτων πρέπει πρωταρχικά να αποσαφηνίσουμε τρία αντικείμενα. Αυτά είναι :

- I. Να ορίσουμε την αναπαράσταση.
- II. Να ορίσουμε τους γενετικούς τελεστές.
- III. Και τέλος, να ορίσουμε μία αντικειμενική διαδικασία.

Η GALib μας βοηθάει να ορίσουμε τα δύο πρώτα αντικείμενα, παρέχοντας μας παραδείγματα και κομμάτια από τα οποία μπορούμε να δημιουργήσουμε τις αναπαραστάσεις και τους τελεστές που επιθυμούμε για το πρόβλημα που έχουμε να φέρουμε εις πέρας. Αυτό σε πολλές περιπτώσεις γίνεται με λίγες ή ακόμη και χωρίς καμία τροποποίηση τους. Στο τρίτο πρόβλημα για τον σαφή καθορισμό μίας αντικειμενικής διαδικασίας αποτίμησης των μελών πρέπει να δώσουμε λύση εμείς. Η διαδικασία αυτή εξαρτάται και ποικίλη από πρόβλημα σε πρόβλημα. Εφόσον, διαθέτουμε την αναπαράσταση, τους τελεστές, και το κριτήριο εκτίμησης, μπορούμε να εφαρμόσουμε οποιονδήποτε γενετικό αλγόριθμο για να βρούμε την βέλτιστη λύση σε κάποιο πρόβλημα.

Όταν θέλουμε να χρησιμοποιήσουμε τους γενετικούς αλγορίθμους για να επιλύσουμε κάποιο πρόβλημα βελτιστοποίησης, θα πρέπει να είμαστε ικανοί να αναπαραστήσουμε μία λύση του προβλήματος, σε μία μόνο δομή δεδομένων. Ο γενετικός αλγόριθμος θα δημιουργήσει έναν πληθυσμό από λύσεις βασισμένος

στην δομή δεδομένων που του παρέχουμε στην κατηγορία Γονίδιο. Στην συνέχεια ο αλγόριθμος δουλεύει πάνω στον πληθυσμό και τον εξελίσσει προκειμένου να βρει την καλύτερη λύση.

Στην GALib μία δομή δεδομένων ονομάζεται GAGenome (ορισμένοι το αναφέρουν σαν χρωμόσωμα). Η βιβλιοθήκη περιέχει τέσσερις τύπους γονιδιωμάτων ή τέσσερα πρότυπα δομών δεδομένων:

- |     |              |      |                      |
|-----|--------------|------|----------------------|
| I.  | GAListGenome | III. | GAArrayGenome        |
| II. | GATreeGenome | IV.  | GABinaryStringGenome |

Αυτές οι τέσσερις βασικές κατηγορίες παράγονται από την πρωταρχική κατηγορία GAGenome. Η δομή δεδομένων που ακολουθεί κάθε κατηγορία αναγνωρίζεται από το όνομα της. Στις εφαρμογές που αναπτύσσουμε πρέπει να χρησιμοποιούμε εκείνη την δομή δεδομένων που ταιριάζει με τον ορισμό του προβλήματος μας. Για παράδειγμα, αν έχουμε να βελτιστοποιήσουμε μία διαδικασία που εξαρτάται από 5 πραγματικούς αριθμούς, τότε θα ήταν καλύτερο να χρησιμοποιήσουμε μία δομή μονοδιάστατου πίνακα πραγματικών αριθμών, 5 στοιχείων.

Η GALib για την επέκταση των δυνατοτήτων της στην ανάπτυξη αντικειμένων παρέχει δύο κύριους μηχανισμούς. Από τις συστάσεις της GALib ο προτεινόμενος μηχανισμός είναι να παράγουμε τις δικές μας κατηγορίες, να ορίσουμε νέα μέλη και νέες διαδικασίες. Ο δεύτερος μηχανισμός είναι να τροποποιήσουμε τις ήδη υπάρχουσες συναρτήσεις που μας παρέχει η ίδια η βιβλιοθήκη.

Όταν η GALib είναι κατάλληλα ανεπτυγμένη είναι ικανή να διεξάγει έρευνα σε ένα ευρύ πεδίο τιμών αναζήτησης, επίσης μπορεί το ίδιο καλά να διεξάγει μία έρευνα σε κάποιο τοπικού μεγέθους πεδίου τιμών. Ο τρόπος συμπεριφοράς της εξαρτάται από τον τρόπο με τον οποίο λειτουργούν οι τελεστές που χρησιμοποιεί και από το σχήμα του χώρου έρευνας της συνάρτησης μας.

### **1.1. Βασικοί τύποι.**

Από τα εισαγωγικά στοιχεία που αναφέραμε για την βιβλιοθήκη της GALib είναι ότι μας παρέχει κάποιες κατηγορίες και τύπους δεδομένων προκειμένου να μας βοηθήσει να αναπτύξουμε τα δικά μας προγράμματα και να χτίσουμε τις δικές μας εφαρμογές. Παραπάνω μπήκαμε σε μερική περιγραφή των δυο βασικών κατηγοριών και των τύπων δεδομένων, στην συνέχεια θα επεκταθούμε σε αυτά, στα περιεχόμενα τους και τις αρμοδιότητες που έχουν στην βιβλιοθήκη.

#### **1.1.1. Γενετικός Αλγόριθμος.**

Ο βασικότερος σκοπός του αντικείμενου του γενετικού αλγορίθμου είναι να αποφασίζει ποιες μονάδες πρέπει να επιβιώσουν, ποιες θα πρέπει να αναπαραχθούν, και ποιες θα πρέπει να φύγουν από έναν πληθυσμό. Επίσης, κρατάει στατιστικά στοιχεία και αποφασίζει για πόσο χρονικό διάστημα θα πρέπει να διαρκέσει η εξέλιξη ενός αλγορίθμου.

Η βασική κατηγορία των γενετικών αλγορίθμων, περιέχει τελεστές και συναρτήσεις μέλη που είναι κοινά για όλες τις κατηγορίες των γενετικών αλγορίθμων που παρέχει η GALib. Όταν δημιουργήσουμε κάποιον γενετικό αλγόριθμο μπορούμε να χρησιμοποιήσουμε αυτά τα δεδομένα και τις συναρτήσεις. Συνοπτικά μία τυπική κατηγορία που περιγράφει έναν γενετικό αλγόριθμο περιέχει το αντικείμενο του πληθυσμού, την στρατηγική αντικατάστασης των μελών του πληθυσμού, τις παραμέτρους για την εκτέλεση του αλγορίθμου, την συλλογή των μεταβλητών του γονιδίου και τέλος περιέχει κάποια στατιστικά στοιχεία για τους τελεστές της Επιλογής και τους τελεστές Κλιμάκωσης.

Ένας τυπικός γενετικός αλγόριθμός θα εκτελείται για πάντα. Γενικά στους γενετικούς αλγορίθμους δεν υπάρχει κάποιο αδιαμφισβήτητο κριτήριο τερματισμού της διαδικασίας εκτέλεσης τους. Το βάρος της ευθύνης το φέρει ο δημιουργός που θα πρέπει να αποφασίσει πότε να σταματήσει η εκτέλεση του αλγορίθμου. Συνήθως, σαν μέτρο λήξης του αλγορίθμου χρησιμοποιούμε τον αριθμό των γενεών. Βέβαια, μπορούμε να χρησιμοποιήσουμε διάφορα κριτήρια όπως την καταλληλότητα της λύσης, κάποια συνθήκη σύγκλισης του πληθυσμού, ή οποιοδήποτε άλλο κριτήριο που να ταιριάζει με το πρόβλημα μας. Η βιβλιοθήκη περιλαμβάνει έτοιμες, ποικίλες συναρτήσεις για τον προσδιορισμό της συνθήκης τερματισμού, υπάρχουν συναρτήσεις που μας επιτρέπουν να προσδιορίζουμε το μέγιστο αριθμό επαναλήψεων εκτέλεσης του αλγορίθμου, συναρτήσεις που να προσδιορίζουμε την τιμή της καλύτερης γενιάς που πρέπει να συγκλίνει κ.α.

Υπάρχουν τέσσερις διαφορετικοί τύποι γενετικών αλγορίθμων. Ο πρώτος είναι ο βασικός αλγόριθμός και ονομάζεται, ‘απλός – *simple* γενετικός αλγόριθμος’. Αυτός ο αλγόριθμος χρησιμοποιεί : μη – υπερθετικούς πληθυσμούς και ελιτιστική επιλογή. Σε κάθε γενιά αντικαθίσταται ολόκληρος ο πληθυσμός εκ’ νέου από νέα μέλη.

Η δεύτερη κατηγορία αλγορίθμων είναι ο ‘κανονικός – *steady-state* γενετικός αλγόριθμος’ που χρησιμοποιεί υπερθετικούς πληθυσμούς. Σε αυτού του τύπου τους αλγορίθμους μπορούμε να καθορίσουμε πόσα μέλη του πληθυσμού πρέπει να αντικατασταθούν σε κάθε γενιά.

Η τρίτη κατηγορία ονομάζεται ‘αυξανόμενοι – *incremental* γενετικοί αλγόριθμοι’ στους οποίους κάθε γενιά αποτελείται από ένα ή δύο παιδιά. Αυτή η κατηγορία αλγορίθμων μας επιτρέπει να προσδιορίσουμε μία μέθοδο αντικατάστασης έτσι ώστε να προσδιορίσουμε τον τρόπο ολοκλήρωσης της νέας γενεάς μέσα στον υπάρχον πληθυσμό. Έτσι για παράδειγμα, ένα νέο μέλος μπορεί να αντικαταστήσει τους γονείς του, ή κάποιο μέλος τυχαία, ή κάποιο μέλος που του ταιριάζει περισσότερο βάση της τιμής καταλληλότητας.

Ο τέταρτος τύπος ονομάζεται ‘*deme*’ γενετικός αλγόριθμος. Αυτού του τύπου οι αλγόριθμοι συμπεριλαμβάνονται πολλαπλοί παράλληλα εξελισσόμενοι πληθυσμοί χρησιμοποιώντας τον *steady-state* αλγόριθμο. Κάθε γενιά του αλγορίθμου αποδημεί / μεταναστεύει κάποια μέλη του πληθυσμού προς άλλους πληθυσμούς.

Η βασική διαφορά των τύπων που περιγράψαμε παραπάνω και τους διακρίνουμε μεταξύ τους είναι ως προς τον τρόπο με τον οποίο δημιουργούν καινούργιες μονάδες σε έναν πληθυσμό και ο τρόπος που αντικαθιστούν τις παλιές κατά την διάρκεια της εκτέλεσης τους.

### **1.1.2. Ορίζοντας μία αναπαράσταση.**

Για να αντιμετωπίσουμε ένα πρόβλημα ορθά, πρέπει να χρησιμοποιήσουμε την καταλληλότερη δομή δεδομένων η οποία μας δίνει την δυνατότητα να αναπαραστήσουμε τα δεδομένα του προβλήματος, σε ένα γονίδιο. Για παράδειγμα, αν θέλουμε να βελτιστοποιήσουμε μία συνάρτηση πραγματικών αριθμών, θα πρέπει να δημιουργήσουμε ένα γονίδιο από πραγματικούς αριθμούς. Αν η λύση του προβλήματος αποτελείται από μιγαδικούς και ακεραίους αριθμούς, τότε θα πρέπει να ορίσουμε ένα γονίδιο με αυτά τα χαρακτηριστικά.

Ο καθορισμός μίας αναπαράστασης είναι η τέχνη της χρησιμοποίησης των γενετικών αλγορίθμων. Πρέπει να χρησιμοποιήσουμε μία αναπαράσταση όσο πιο μικρή γίνεται, αλλά απολύτως εκφραστική,

δηλαδή πρέπει να είναι ικανή να αναπαραστήσει όποια λύση θέλουμε για το πρόβλημα μας. Ο αριθμός των δυνατών αναπαραστάσεων είναι ατελείωτες, εμείς από την πλευρά μας πρέπει να επιλέξουμε την φτωχότερη αναπαράσταση όπως έναν πίνακα από πραγματικούς αριθμούς. Με αυτόν τον τρόπο κάνουμε τον αλγόριθμο πιο γρήγορο και αποδοτικό. Επίσης, μπορούμε να ορίσουμε την αναπαράσταση έτσι ώστε να μην επιτρέπουμε να αναπαραστήσει μη πρακτικές λύσεις για το πρόβλημα μας, για παράδειγμα να ορίσουμε κάποιες τιμές κατωφλίου. Αν το γονίδιο μπορεί να αναπαραστήσει μη πρακτικές λύσεις, όπως να επιτρέπουμε να πάρει μία τιμή εκτός του αποδεκτού εύρους τιμών, τότε η συνάρτηση καταλληλότητας πρέπει να σχεδιαστεί ανάλογα για να δίνει μεροληπτική εκτίμηση σε αυτά τα μέλη.

Τυπικά μία αναπαράσταση δεν θα πρέπει να είναι δυνατόν να πάρει τιμές πέρα από αυτές που χρειαζόμαστε για να δώσουμε λύση σε ένα πρόβλημα. Ορισμένες φορές όμως αξίζει να χρησιμοποιήσουμε αναπαραστάσεις που να επιτρέπουν την λήψη επιπλέον γενετικού υλικού, σύμφωνα πάντα με την αντικειμενική διαδικασία και με τον τρόπο μελέτης του τύπου και των χαρακτηριστικών του πεδίου έρευνας. Αυτό τείνει να αυξήσει το μέγεθος του πεδίου αναζήτησης και σαν επακόλουθο έχει να ελαττώσει την απόδοση του αλγορίθμου.

### **1.1.3. Οι τελεστές του γονιδίου.**

Κάθε κατηγορία που περιγράφει ένα γονίδιο διαθέτει ορισμένους τελεστές, χρησιμοποιώντας τους μπορούμε να προκαταλάβουμε τον αρχικό πληθυσμό, επίσης μπορούμε να ορίσουμε τους τελεστές της Διασταύρωσης και της Μετάλλαξης και να τους προσαρμόσουμε ειδικά στην αναπαράσταση του προβλήματος, ή να αναπτύξουμε μέρη του αλγορίθμου καθώς ο πληθυσμός μας εξελίσσεται.

Η GALib έρχεται με τους τελεστές αυτούς ορισμένους για κάθε τύπο γονιδίου, αλλά μπορούμε να τους αλλάξουμε σύμφωνα με τις προτιμήσεις μας. Οι τρεις βασικοί τελεστές είναι οι εξής : ο τελεστής της Αρχικοποίησης, ο τελεστής της Διασταύρωσης και ο τελεστής της Μετάλλαξης.

Ο *τελεστής αρχικοποίησης* ορίζει τον τρόπο με τον οποίο αρχικοποιείτε κάθε γονίδιο, δίνοντας του αρχικές τιμές. Η κατηγορία καλείται κάθε φορά που αρχικοποιείται ένας πληθυσμός ή ο γενετικός αλγόριθμος. Στην πράξη ο τελεστής αρχικοποίησης δεν δημιουργεί καινούργια γονίδια, αλλά γεμίζει τα νέα γονίδια με το βασικό ‘γενετικό υλικό’ στο οποίο μπορούν να βασιστούν και να αναπτυχθούν όλες οι λύσεις του προβλήματος μας. Το αντικείμενο πληθυσμός διαθέτει τον δικό του τελεστή αρχικοποίησης, αλλά εξ’ ορισμού καλεί τον τελεστή αρχικοποίησης του γονιδίου.

Ο *τελεστής της Μετάλλαξης* ορίζει την διαδικασία μετάλλαξης κάθε γονιδίου. Η Μετάλλαξη σημαίνει διαφορετικά πράγματα για διαφορετικούς τύπους δεδομένων. Για παράδειγμα, ένας τυπικός μεταλλάκτης όταν ενεργοποιείται σε μία δυαδική πλειάδα από bit, αναστρέφει τα bits από 0 σε 1 και το αντίστροφο, σύμφωνα με την δοσμένη πιθανότητα. Ένας τυπικός μεταλλάκτης δέντρου, μεταθέσει τα υποδέντρα με την δοσμένη πιθανότητα. Γενικά θα πρέπει να ορίσουμε έναν μεταλλάκτη που να μπορεί να κάνει εξερεύνηση και αξιολόγηση των μελών που δημιουργεί, για να εξαλείψουμε το ενδεχόμενο να παράγουμε μέλη που έχουν τιμές εκτός του αποδεκτού πεδίου τιμών.

Ο *τελεστής της Διασταύρωσης* ορίζει την διαδικασία για την δημιουργία ενός παιδιού από δύο γονίδια. Όπως ο τελεστής της μετάλλαξης, έτσι και ο τελεστής της διασταύρωσης είναι σύμφωνος προς τον τύπο δεδομένων του γονιδίου, μία διαφορά τους είναι ότι ο τελεστής της Διασταύρωσης περιλαμβάνει πολλαπλά γονίδια. Στην GALib, κάθε αντικείμενο του γονιδίου ‘γνωρίζει’ την μέθοδο ζευγαρώματος,

αλλά είναι ανήμπορη να την εκτελέσει από μόνη της. Κάθε κατηγορία του γενετικού αλγορίθμου 'μαθαίνει' πως να εκτελέσει την μέθοδο Διασταύρωσης από το αντικείμενο του γονιδίου και έπειτα χρησιμοποιεί την μέθοδο ζευγαρώματος. Με αυτό το μοντέλο είναι πολύ πιθανό να παράγουμε καινούργιες κατηγορίες του γενετικού αλγορίθμου που να χρησιμοποιούν την μέθοδο ζευγαρώματος να είναι όμως διαφορετικές από τις προεπιλεγμένες που ορίζονται στο γονίδιο.

Κάθε μία από τις τρεις μεθόδους μπορούνε να τροποποιηθούνε και να γίνουν πιο συγκεκριμένες και κατάλληλες για τον τύπο δεδομένων, αλλά και για τον τύπο του προβλήματος που αντιμετωπίζουμε.

Επιπρόσθετα στο αντικείμενο του γονιδίου θα πρέπει να περιέχεται μία αντικειμενική διαδικασία και ίσως κάποια διαδικασία σύγκρισης. Η *αντικειμενική διαδικασία* χρησιμοποιείται για να αξιολογήσουμε τα γονίδια. Η *διαδικασία σύγκρισης* χρησιμοποιείται για να προσδιορίσουμε πόσο διαφορετικό είναι ένα γονίδιο από κάποιο άλλο. Η τελευταία διαδικασία συνήθως αναφέρεται και σαν διαδικασία 'Απόστασης'. Κάθε αντικείμενο γενετικού αλγορίθμου απαιτεί για την σωστή λειτουργία του τα αποτελέσματα που προέρχονται από την αντικειμενική διαδικασία. Μέσω αυτού ένας γενετικός αλγόριθμος αποφασίζει ποια μονάδα του πληθυσμού είναι καλύτερη από τις υπόλοιπες. Κάποιοι αλγόριθμοι απαιτούν την ύπαρξη της διαδικασίας σύγκρισης.

Γενικά, ένας γενετικός αλγόριθμος δεν χρειάζεται να γνωρίζει τίποτε για την δομή και τα περιεχόμενα των δεδομένων των γονιδίων στα οποία τον εφαρμόζουμε. Μπορούμε να αναμιξούμε και να ταιριάξουμε τα γονίδια στην διάρκεια της εκτέλεσης του.

#### **1.1.4. Το αντικείμενο του πληθυσμού.**

Το αντικείμενο του πληθυσμού είναι μία συλλογή γονιδίων, έχει την δική της συνάρτηση αρχικοποίησης και αποτίμησης(η προεπιλογή είναι το κάλεσμα των συναρτήσεων για κάθε μονάδα του πληθυσμού). Επίσης διατηρεί εγγραφές των καταλληλότερων μελών του πληθυσμού, εγγραφές μέσου όρου και απόκλισης για κάθε γενιά. Η μέθοδος επιλογής είναι επίσης ορισμένη στο αντικείμενο του πληθυσμού. Αυτή η μέθοδος χρησιμοποιείται για να επιλέξουμε ποιες μονάδες θα ζευγαρώσουν.

Σε αυτήν την κατηγορία υπάρχουν επεκτάσεις, όμως οι πρόσθετη υπολογισμοί που απαιτούνται αυξάνουν τις απαιτήσεις στην υπολογιστική ισχύ, δεν είναι προεπιλεγμένη η διατήρηση τέτοιων πληροφοριών.

#### **1.1.5. Αντικειμενικές Διαδικασίες & κλιμάκωση της τιμής Καταλληλότητας.**

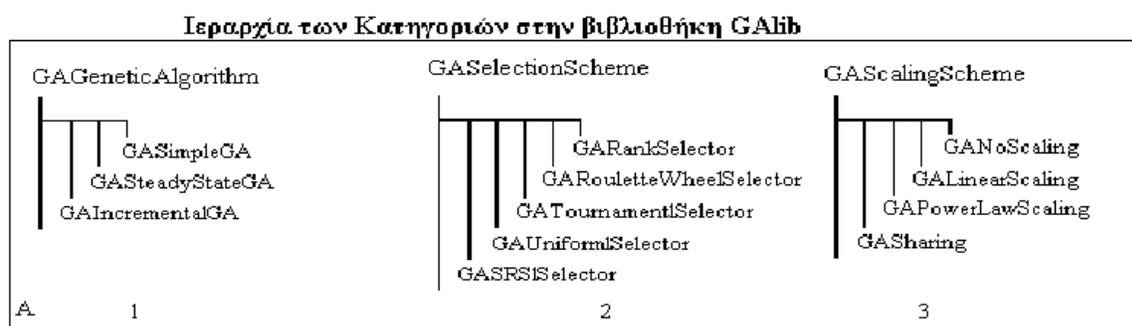
Οι γενετικοί αλγόριθμοι είναι πιο ελκυστικοί από την βαθμωτή έρευνα επειδή χειρίζονται περίπλοκες και δύσκολες εξισώσεις ή μικρούς χώρους έρευνας. Οι γενετικοί αλγόριθμοι χρειάζονται ένα απλό μέτρο σύγκρισης των μονάδων που διαχειρίζονται, αυτό το μέτρο μας το παρέχει η αντικειμενική διαδικασία, έτσι δίνουμε μία απλή και εύκολη λύση σε ένα πολύ δύσκολο και χρονοβόρο πρόβλημα.

Είναι πολύ σημαντικό να μπορούμε να διακρίνουμε την διαφορά που υπάρχει μεταξύ των τιμών που μας επιστρέφουν οι διαδικασίες υπολογισμού της καταλληλότητας και αυτών που υπολογίζουν την αντικειμενική τιμή ή τιμή εκτίμησης. Η τιμή εκτίμησης είναι μία 'ακατέργαστη' τιμή που μας επιστρέφεται από την αντικειμενική διαδικασία και μας δείχνει την απόδοσή του γονιδίου πρωταρχικά. Η τιμή καταλληλότητας είναι αυτή που χρησιμοποιείται από το αντικείμενο του γενετικού αλγορίθμου, για να υποδείξει ποια γονίδια θα συμμετάσχουν για ζευγάριμα στην διασταύρωση. Τυπικά αυτή η τιμή αποκτεείται από την γραμμική κλιμάκωση της αντικειμενικής τιμής των γονιδίων. Μπορούμε να

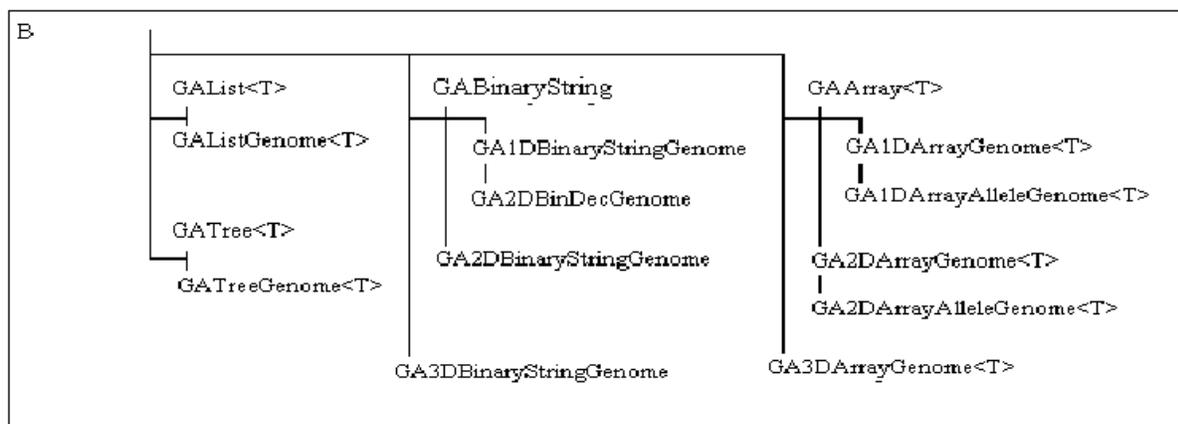
αναπτύξουμε μία διαδικασία εκτίμησης του κάθε γονιδίου, όταν αυτό δημιουργείται, επίσης μπορούμε να αναπτύξουμε μία διαδικασία εκτίμησης ολόκληρου του πληθυσμού. Στην πρώτη περίπτωση θα πρέπει να ορίσουμε την αντικειμενική συνάρτηση στο αντικείμενο του γονιδίου, αλλιώς θα πρέπει να την τοποθετήσουμε στο αντικείμενο του πληθυσμού. Και στις δύο περιπτώσεις την συνάρτηση αυτή πρέπει να οριστεί από εμάς.

## 1.2. Ένα Ιεραρχικό Διάγραμμα των κατηγοριών που μας παρέχει η βιβλιοθήκη GALib.

Στο παρακάτω σχήμα απεικονίζουμε το σύνολο των κατηγοριών που μας παρέχει η βιβλιοθήκη της GALib σε μία ιεραρχική διάταξη ανάλογα με το είδος τους. Οι κατηγορίες που υπάρχουν στο κουτί A με τον δείκτη 1 αναφέρονται στους τύπους του γενετικού αλγορίθμου. Με τον δείκτη 2 αναφέρονται στα σχήματα επιλογής των μελών του νέου πληθυσμού. Με τον δείκτη 3 είναι τα σχήματα κλιμάκωσης της τιμής καταλληλότητας των μελών.



Στο κουτί B δείχνουμε ιεραρχικά τις Δομές Δεδομένων που ορίζονται στους γενετικούς αλγορίθμους :



Σχήμα 8<sup>ο</sup> : Ιεραρχικό Διάγραμμα των κατηγοριών της GALib.

## 1.3. Το Περιβάλλον Προγραμματισμού της GALib.

### GAGeneticAlgorithm

Αυτή είναι η βασική κατηγορία που περιγράφει έναν γενετικό αλγόριθμο και τους προεπιλεγμένους τελεστές της, επίσης σε αυτή την κατηγορία κρατάμε στοιχεία και στατιστικά για την εξέλιξη του αλγορίθμου. Για παράδειγμα, στοιχεία για τον τελεστή της Μετάλλαξης, της Διασταύρωσης, τον αριθμό των γενεών εξέλιξης, τους καλύτερους – ενδιάμεσους – χειρότερους σε κάθε γενιά και στοιχεία για τον αρχικό και τον τρέχον πληθυσμό. Επίσης, ορίζουμε μία διαδικασία που προσδιορίζει το κριτήριο τερματισμού του αλγορίθμου, τον Εξολοθρευτή.

Τα στατιστικά στοιχεία μπορούμε να τα κρατάμε σε κάποιο αρχείο ανά γενιά ή περιοδικά ορίζοντας την συχνότητα που θα γίνεται η καταγραφή τους. Οι παράμετροι όπως ο αριθμός των γενεών περάτωσης του αλγορίθμου, ή η πιθανότητα Διασταύρωσης και Μετάλλαξης μπορούν να οριστούν από κάποια συνάρτηση μέλος της κατηγορίας, είτε από κάποιο αρχείο, είτε από την γραμμή εντολών.

Περιγραφή του αντικειμένου: Η διαδικασία εξέλιξης(evolne) είναι μία συνάρτηση μέλος του γενετικού αλγορίθμου, η οποία αρχικά καλεί την συνάρτηση Αρχικοποίησης. Στην συνέχεια καλεί την συνάρτηση step επαναληπτικά, η επανάληψη σταματάει όταν ικανοποιηθεί το κριτήριο τερματισμού του αλγορίθμου. Η συνάρτηση done ελέγχει το κριτήριο τερματισμού και επιστρέφει την τιμή gaTrue. Τέλος καλείται η συνάρτηση flushScores, προκειμένου να τραβήξει όσα στοιχεία βρίσκονται στον Buffer.

Αν δεν χρησιμοποιήσουμε στον γενετικό αλγόριθμο την συνάρτηση εξέλιξης, τότε θα πρέπει να καλέσουμε την συνάρτηση Αρχικοποίησης, πριν ξεκινήσουμε την εξέλιξη του πληθυσμού με την συνάρτηση step. Σε κάθε επανάληψη της η συνάρτηση step εξελίσσει μία μόνο γενιά του πληθυσμού. Μετά το πέρας της εκτέλεσης του αλγορίθμου πρέπει να καλέσουμε την συνάρτηση flushScores

#### **GASimpleGA** (Μη – Υπερθετικοί πληθυσμοί)

Ο απλός – simple γενετικός αλγόριθμος χρησιμοποιεί σταθερούς πληθυσμούς. Όταν επιλέξουμε να τον εφαρμόσουμε για την επίλυση του προβλήματος, θα πρέπει να προσδιορίσουμε είτε κάποια μονάδα, είτε κάποιον πληθυσμό. Ο νέος αλγόριθμός θα αντιγράψει τις μονάδες που προσδιορίσαμε για να δημιουργήσει τον δικό του καινούργιο πληθυσμό. Μπορούμε να αλλάξουμε τις περισσότερες από τις ιδιοτροπίες του αλγορίθμου μετά από την δημιουργία κατά την διάρκεια της εξέλιξης του.

Ο simple γενετικός αλγόριθμος δημιουργεί τον αρχικό πληθυσμό με την αντιγραφή της μονάδας ή του πληθυσμού που του παρέχουμε. Σε κάθε νέα γενιά ο αλγόριθμος δημιουργεί τον πληθυσμό εξ' ολοκλήρου με μονάδες που προέρχονται από το προγενέστερο πληθυσμό, ορισμένες από αυτές 'ζευγαρώνουν' και παράγουν νέους απογόνους, που συμπληρώνουν τον πληθυσμό. Αυτή η διαδικασία ακολουθείται επαναληπτικά έως ότου τα κριτήρια τερματισμού ικανοποιηθούν. Τα κριτήρια τερματισμού ορίζονται από τον εξολοθρευτή.

Η διαδικασία του Ελιτισμού είναι η προεπιλογή και χρησιμοποιείται από τον αλγόριθμο. Με αυτή την επιλογή επιβεβαιώνουμε ότι τα καταλληλότερα μέλη που βρίσκονται σε έναν πληθυσμό θα επιβιώσουν. Για να απενεργοποιήσουμε την χρήση του θα πρέπει να περάσουμε στην αρμόδια συνάρτηση μέλους την τιμή gaFalse. Η προεπιλεγμένη μέθοδος κλιμάκωσης της τιμής καταλληλότητας είναι η Γραμμική. Η μέθοδος επιλογής που χρησιμοποιείται είναι η μέθοδος του Τροχού της Τύχης.

#### **GASteadyStateGA** (Υπερθετικοί πληθυσμοί).

Αυτός ο τύπος γενετικού αλγορίθμου χρησιμοποιεί υπερθετικούς πληθυσμούς, το ποσοστό επικάλυψης ενός πληθυσμού προσδιορίζεται από τον χρήστη. Ο αλγόριθμος δημιουργεί μονάδες με την αντιγραφή των γονιδίων ή του πληθυσμού, που επιλέγουν και δημιουργούν οι χρήστες. Σε κάθε γενιά ο αλγόριθμος δημιουργεί έναν ενδιάμεσο πληθυσμό από μονάδες, έπειτα τις προσθέτει στον προηγούμενο πληθυσμό και τέλος απορρίπτει τις μονάδες με την χειρότερη τιμή καταλληλότητας προκειμένου να επαναφέρει τον πληθυσμό στο κανονικό μέγεθος.

Το ποσοστό επικάλυψης των μελών ενός πληθυσμού μεταξύ των γενεών δηλώνεται με την παράμετρο pReplacement. Αυτή η παράμετρος δηλώνει το ποσοστό του πληθυσμού που θα πρέπει να

αντικατασταθεί σε κάθε γενιά. Οι απόγονοι που παρήχθησαν από την εκτέλεση της διαδικασίας προστίθενται στον πληθυσμό και στην συνέχεια οι χειρότεροι του πληθυσμού καταστρέφονται. Με την ακολουθία αυτών των βημάτων τα καινούργια μέλη του πληθυσμού μπορούν να επιζήσουν ή να απορριφθούν από τον πληθυσμό, αυτό εξαρτάται αν είναι κατάλληλοι ή όχι για τον πληθυσμό. Εναλλακτικά αν δεν θελήσουμε να ορίσουμε κάποιο ποσοστό, μας δίνεται η δυνατότητα να ορίσουμε έναν αριθμό (μικρότερο από το μέγεθος του πληθυσμού) που να δηλώνει τον αριθμό των μελών που θέλουμε να αντικαταστήσουμε. Δεν μπορούμε να ορίσουμε και τις δύο παραμέτρους σε έναν αλγόριθμο, συνήθως χρησιμοποιείται ο αριθμός. Η προεπιλεγμένη μέθοδος κλιμάκωσης της τιμής καταλληλότητας είναι η Γραμμική Κλιμάκωση. Η μέθοδος επιλογής που χρησιμοποιείται και σε αυτό τον αλγόριθμο είναι η μέθοδος του Τροχού της Τύχης.

**GAIncrementalGA** (*Υπερθετικοί πληθυσμοί με 1 ή 2 παιδιά ανά γενιά*).

Αυτός ο αλγόριθμος βασίζεται σε κάποιο μοντέλο γεννήτορα. Χρησιμοποιούμε υπερθετικούς πληθυσμούς, αλλά με μικρό βαθμό επικάλυψης(μόνο ένα ή το πολύ δύο μονάδες αντικαθίστανται σε κάθε γενιά). Το προεπιλεγμένο σχήμα αντικατάστασης υποστηρίζει ότι πρέπει να αντικαταστήσουμε τα χειρότερα μέλη του πληθυσμού. Μία διαδικασία αντικατάστασης απαιτείται μόνο αν επιλέξουμε να χρησιμοποιήσουμε σαν σχήματα αντικατάστασης Custom ή Crowding σχήματα.

Μπορούμε να ορίσουμε το πλήθος των παιδιών που δημιουργούνται σε κάθε γενιά χρησιμοποιώντας την συνάρτηση nOffspring. Εφόσον ο αριθμός των γονέων είναι δύο γονείς, τα παιδιά μπορούν να είναι από μία διαδικασία 1 ή 2. Η προεπιλογή είναι 2. Για να ορίσουμε τον τρόπο με τον οποίο κάποιο παιδί θα εισαχθεί στον πληθυσμό, θα πρέπει να χρησιμοποιήσουμε την συνάρτηση αντικατάστασης. Σε αυτή την συνάρτηση θα προσδιορίσουμε την στρατηγική αντικατάστασης που θα ακολουθήσουμε στον γενετικό αλγόριθμο. Για παράδειγμα, αν θέλουμε το παραγόμενο παιδί να αντικαταστήσει τον γονέα του θα πρέπει να χρησιμοποιήσουμε την στρατηγική αντικατάστασης γονέων. Άλλες στρατηγικές που μπορούμε να χρησιμοποιήσουμε είναι η στρατηγική τυχαίας αντικατάστασης και αυτή του χειρότερου μέλους στον πληθυσμό.

**GADemeGA** (*Παράλληλοι πληθυσμοί με μετανάστευση*)

Αυτό ο τύπος γενετικού αλγορίθμου έχει πολλαπλά, εξελισσόμενους πληθυσμούς. Δημιουργεί άλλους πληθυσμούς με αντιγραφή του γονιδίου ή του πληθυσμού που δίνουμε στον αλγόριθμο όταν δημιουργείται.

Κάθε πληθυσμός εξελίσσεται χρησιμοποιώντας τον steady-state γενετικό αλγόριθμο, αλλά κάποιες μονάδες του πληθυσμού μετακινούνται από τον έναν πληθυσμό σε κάποιον άλλο. Ο αλγόριθμος μετακίνησης είναι ένας αιτιοκρατικός αλγόριθμος. Κάθε πληθυσμός μετακινεί τις καλύτερες μονάδες του σε γειτονικούς πληθυσμούς. Ο βασικός πληθυσμός ανανεώνεται σε κάθε γενιά με τις καλύτερες μονάδες από κάθε πληθυσμό. Αν θέλουμε να πειραματιστούμε με κάποιες άλλες μεθόδους μετακίνησης, μπορούμε να αναπτύξουμε δικές κατηγορίες και να ορίσουμε έναν καινούργιο τελεστή μετακίνησης.

**Συνάρτηση τερματισμού – Εξολοθρευτής.**

Η συνάρτηση περάτωσης – completion χρησιμοποιείται για να προσδιορίσουμε πότε ο γενετικός αλγόριθμος θα σταματήσει να εκτελείται. Η συνάρτηση completion καλείται από την συνάρτηση μέλος done προκειμένου να αποφανθεί αν ο αλγόριθμος θα σταματήσει να εκτελείται ή όχι.

Η προκαθορισμένη μορφή της συνάρτησης περάτωσης – completion χρησιμοποιεί τις συναρτήσεις Generation(Γενιά) και Convergence(Συνθήκη Σύγκλισης) για να μας επιστρέψει την τιμή τερματισμού. Οι επιστρεφόμενες τιμές της συνάρτησης περάτωσης είναι gaTrue για να συνεχίσει να εκτελείται ο αλγόριθμος και gaFalse για να σταματήσει την εκτέλεση του.

### **Σχήματα Αντικατάστασης.**

Τα σχήματα αντικατάστασης χρησιμοποιούνται από τους αυξητικούς γενετικούς αλγορίθμους προκειμένου να προσδιορίσουν πώς οι καινούργιες μονάδες του πληθυσμού θα πρέπει να εισαχθούν στον πληθυσμό. Αποδεκτά σχήματα αντικατάστασης είναι τα :

GAIncrementalGA::RANDOM	GAIncrementalGA::CUSTOM
GAIncrementalGA::BEST	GAIncrementalGA::CROWDING
GAIncrementalGA::WORST	GAIncrementalGA::PARENT

Γενικά, η αντικατάσταση των χειρότερων μονάδων φέρνει τα καλύτερα αποτελέσματα. Η αντικατάσταση των γονέων είναι χρήσιμη για την βασική διαφοροποίηση του είδους, τέλος η custom αντικατάσταση μπορεί να χρησιμοποιηθεί όταν θέλουμε να ορίσουμε το δικό μας είδος.

Αν επιλέξουμε το custom ή το crowding σχήμα αντικατάστασης τότε θα πρέπει να ορίσουμε μία συνάρτηση αντικατάστασης. Η συνάρτηση αντικατάστασης θα λαμβάνει παραμέτρους τα μέλη του πληθυσμού και τον πληθυσμό στον οποίο θα πρέπει να τοποθετηθούν τα μέλη. Η συνάρτηση αυτή θα μας επιστρέφει μία αναφορά στο γονίδιο το οποίο θα πρέπει να αντικατασταθεί, σε διαφορετική περίπτωση που το μέλος δεν πρέπει να εισαχθεί στον πληθυσμό, τότε η συνάρτηση θα πρέπει να επιστρέφει μία αναφορά στην μονάδα. Κάθε συνάρτηση αντικατάστασης πρέπει να έχει ένα πρότυπο :

```
typedef GAgenome& (*GAIncrementalGA::ReplacementFunction)(GAgenome &, GApopulation &);
```

Η πρώτη παράμετρος είναι το γονίδιο το οποίο είναι υποψήφιο να εισαχθεί στον πληθυσμό, η δεύτερη παράμετρος είναι ο πληθυσμός στον οποίο θέλουμε να εισαχθεί.

### **GAgenome.**

Η γενική κατηγορία που περιγράφει ένα γονίδιο είναι το GAgenome. Η κατηγορία αυτή είναι μία εικονική κατηγορία που σκοπό έχει να περιγράψει τον αριθμό των σταθερών και των προτύπων συναρτήσεων που αρμόζουν σε οποιοδήποτε τύπο γονιδίου.

Η παράμετρος διάσταση στα γονίδια χρησιμοποιείται για να ορίσουμε σε ποια διάσταση της δομής δεδομένων του γονιδίου αναφερόμαστε όταν έχουμε επιλέξει να χρησιμοποιήσουμε πολυδιάστατα γονίδια. Η μέθοδος αντιγραφής – clone καθορίζει πότε θα αντιγράψουμε ολόκληρο το γονίδιο, δηλαδή θα δημιουργήσουμε ένα ταυτόσημο γονίδιο με το κανονικό ή θα αντιγράψει τις ιδιότητες του γονιδίου, δηλαδή θα δημιουργήσει ένα νέο γονίδιο με απaráλλαχτα χαρακτηριστικά. Στις δύο περιπτώσεις αντιγραφής, η συνάρτηση που καλεί την μέθοδο αντιγραφής είναι υπεύθυνη για να αποδεσμεύσει την μνήμη που κατανέμεται στην συνάρτηση μέλος clone.

Οι γενετικοί τελεστές των γονιδίων είναι συναρτήσεις που δέχονται σαν ορίσματα τους τύπους γονιδίων. Έτσι μας δίνεται η δυνατότητα να αναπτύξουμε καινούργιες συμπεριφορές στα γονίδια μας χωρίς να αναπτύξουμε καινούργιες κατηγορίες.

### ***GA1DArrayGenome<T>***

Αυτός ο τύπος δεδομένων χρησιμοποιείται όταν η καταλληλότερη δομή δεδομένων για την περιγραφή των λύσεων σε κάποιο πρόβλημα είναι ένα μονοδιάστατος πίνακας. Ο πίνακας 1 διάστασης μπορεί να περιέχει πολλά στοιχεία στην δομή του. Κάθε στοιχείο στον πίνακα είναι ένα γονίδιο του γονιδιώματος, η τιμή που θα περιέχει το γονίδιο εξαρτάται από τον τύπο του γονιδίου. Για παράδειγμα, η τιμή ενός πίνακα ακεραίων είναι ακέραιες τιμές, ενώ ένας πίνακας από δεκαδικούς θα έχει αριθμούς κινητής υποδιαστολής.

### ***GA1DArrayAlleleGenome<T>***

Το μίας διάστασης πίνακας allele γονιδίωμα αναπτύσσεται από το γονίδιο μονής διάστασης πίνακα που περιγράψαμε παραπάνω. Έχει την ίδια συμπεριφορά με το προηγούμενο αλλά προσθέτει στις ιδιότητες του και το σύνολο allele set. Η τιμή που εκχωρείται σε κάθε στοιχείο του πίνακα εξαρτάται από το σύνολο των επιτρεπτών τιμών του allele set. Η πλέον απλή περίπτωση είναι να δημιουργήσουμε απλό σύνολο τιμών – allele set στο οποίο να ορίζουμε τις επιτρεπόμενες τιμές.

### ***GA2DArrayGenome<T>***

Ο τύπος αυτός περιγράφει το γονιδίωμα που απαιτεί έναν δύο διαστάσεων πίνακα. Το μέγεθος τους αλλάζει και εξαρτάται από τα αντικείμενα του πίνακα. Είναι μία κατηγορία πρότυπο και μπορεί να αναπτυχθεί από τις κατηγορίες του Genome ή του GAArray<>. Κάθε στοιχείο στον πίνακα είναι η ένα γονίδιο του γονιδιώματος, η τιμή που θα περιέχει το γονίδιο εξαρτάται από τον τύπο του γονιδίου.

### ***GA2DArrayAlleleGenome<T>***

Το δύο διαστάσεων πίνακας allele γονιδίωμα αναπτύσσεται από το γονίδιο δύο διαστάσεων πίνακα που περιγράψαμε παραπάνω. Έχει την ίδια συμπεριφορά με το προηγούμενο αλλά προσθέτει στις ιδιότητες του και το σύνολο allele set. Η τιμή που εκχωρείται σε κάθε στοιχείο του πίνακα εξαρτάται από το σύνολο των επιτρεπτών τιμών του allele set. Η πλέον απλή περίπτωση είναι να δημιουργήσουμε απλό σύνολο τιμών – allele set στο οποίο να ορίζουμε τις επιτρεπόμενες τιμές. Σε περισσότερο πολύπλοκα παραδείγματα μπορούμε για κάθε στοιχείο του πίνακα να ορίσουμε ένα σύνολο από allele set.

### ***GA3DArrayGenome<T>***

Ο τύπος αυτός περιγράφει το γονιδίωμα που απαιτεί έναν τριών διαστάσεων πίνακα. Το μέγεθος τους αλλάζει και εξαρτάται από τα αντικείμενα του πίνακα. Είναι μία κατηγορία πρότυπο και μπορεί να αναπτυχθεί από τις κατηγορίες του Genome ή του GAArray<>. Κάθε στοιχείο στον πίνακα είναι η ένα γονίδιο του γονιδιώματος, η τιμή που θα περιέχει το γονίδιο εξαρτάται από τον τύπο του γονιδίου.

### ***GA3DArrayAlleleGenome<T>***

Το τριών διαστάσεων πίνακας allele γονιδίωμα αναπτύσσεται από το γονίδιο τριών διαστάσεων πίνακα που περιγράψαμε παραπάνω. Έχει την ίδια συμπεριφορά με το προηγούμενο αλλά προσθέτει στις ιδιότητες του και το σύνολο allele set. Η τιμή που εκχωρείται σε κάθε στοιχείο του πίνακα εξαρτάται από το σύνολο των επιτρεπτών τιμών του allele set. Η πλέον απλή περίπτωση είναι να δημιουργήσουμε απλό σύνολο τιμών – allele set στο οποίο να ορίζουμε τις επιτρεπόμενες τιμές.

### **GA1DBinaryStringGenome**

Το γονίδιο που αποτελείται από μία δυαδική πλειάδα παράγεται από τις κατηγορίες του GABinaryString και του GAGenome. Αυτή η δομή δεδομένων είναι μία πλειάδα 0 και 1 το μήκος των οποίων μπορεί να είναι σταθερό ή μεταβλητό. Τα γονίδια στο γονιδίωμα είναι bits και το πεδίο τιμών των bit είναι 0 ή 1.

### **GA2DBinaryStringGenome**

Ομοίως με την δυαδική πλειάδα μονής διάστασης και η δύο διαστάσεων δυαδική πλειάδα παράγεται από τις ίδιες κατηγορίες. Είναι ένας πίνακας από 0, 1 και το μήκος τους μπορεί να είναι σταθερό ή μη.

### **GA3DBinaryStringGenome**

Το γονιδίωμα τριπλής πλειάδας παράγεται από τις κατηγορίες του GABinaryString και του GAGenome. Είναι ένα τριών διαστάσεων τμήμα από 0 και 1 των οποίων το μήκος και το πλάτος είναι σταθερό ή μεταβλητό. Τα γονίδια στο γονιδίωμα είναι bits και το πεδίο τιμών των bit είναι 0 ή 1.

### **GABin2DecGenome**

Αυτό το γονιδίωμα είναι μία υλοποίηση της παραδοσιακής μεθόδου για μετατροπή των δυαδικών πλειάδων σε δεκαδικές τιμές. Περιέχει έναν μηχανισμό για την προσαρμοσμένη κωδικοποίηση δυαδικής πλειάδας σε δεκαδική βάση μίας φόρμας για κωδικοποίηση Gray. Και τα δύο χαρακτηριστικά βρίσκονται στην βιβλιοθήκη. Ο προεπιλεγμένος μηχανισμός είναι η σχεδίαση μίας δυαδικής πλειάδας σε δεκαδική(με την αρίθμηση βασισμένη στο 2).

Προκειμένου να χρησιμοποιήσουμε αυτό το γονιδίωμα θα πρέπει να δημιουργήσουμε μία σχεδίαση με την οποία θα ορίζουμε τον αριθμό των bits που θα χρησιμοποιούνται για την αναπαράσταση των δεκαδικών αριθμών. Ο μηχανισμός μετατροπής των δυαδικών σε δεκαδικούς αριθμούς παράγεται από την κατηγορία 1DBinaryStringGenome. Για να γίνει άμεσα εφαρμόσιμη αυτή η κατηγορία, θα πρέπει να δημιουργήσουμε κάποιο φαινότυπο\*. Το φαινότυπο ορίζει πόσα bits χρειάζονται για την χαρτογράφηση των δεκαδικών τιμών και αντιστρόφως. Ένα απλό φαινότυπο περιέχει τον αριθμό των bits που απαιτούνται για να αναπαραστήσουμε την δεκαδική τιμή και την ελάχιστη και μέγιστη δεκαδική τιμή του συνόλου των bits που θα χαρτογραφηθούν.

### **GABin2DecPhenotype**

Το φαινότυπο δυαδικό προς δεκαδικό προσδιορίζει την σχεδίαση μετατροπής της δυαδικής πλειάδας σε δεκαδικές τιμές. Μία σχεδίαση μετατροπής από δυαδική σε δεκαδική αποτελείται από έναν εύρος δεκαδικών τιμών και από έναν αριθμό bits. Για παράδειγμα, αν στην σχεδίαση μας παρέχουμε 8 bits τότε το διαθέσιμο εύρος τιμών θα είναι από 0 έως 255 μοναδικές τιμές.

### **GAListGenome<T>**

Το γονιδίωμα *λίστα* είναι ένα πρότυπο κατηγορίας. Αναπτύξουμε από τις κατηγορίες GAGenome και GAList<>. Μπορεί να χρησιμοποιηθεί για σταθερές αναπαραστάσεις ή για μεταβλητού μήκους σειρές όπως είναι οι παραδοσιακές εφαρμογές με λίστες. Σε αυτή την κατηγορία θα πρέπει να ορίσουμε τον αρχικό τελεστή της κατηγορίας. Ο προεπιλεγμένος τελεστής αρχικοποίησης είναι ο NoInitializer. Σε περίπτωση που δεν επιλέξουμε κάποιον τελεστή αρχικοποίησης θα δεχθούμε λάθη όποτε επιχειρούμε να αρχικοποιήσουμε κάποιο γονιδίωμα.

## **GARealGenome**

Το γονιδίωμα πραγματικών αριθμών είχε σχεδιαστεί με σκοπό να χρησιμοποιηθεί σε εφαρμογές που οι αναπαραστάσεις τους απαιτούσαν κάποιον πίνακα (από πιθανότατα περιορισμένο αριθμό) πραγματικών παραμέτρων. Τα στοιχεία του πίνακα μπορεί να θεωρηθούν ως ένα σύνολο από απαιτούμενες τιμές, ένα σύνολο από περιορισμένες διακριτές τιμές, ή κάποιο σύνολο από απαριθμημένες τιμές, αυτό εξαρτάται από τον τύπο του πεδίου ορισμού που χρησιμοποιήθηκε για να δημιουργήσουμε το γονίδιο. Μπορούμε να προσμίξουμε το σύνολο των πεδίων τιμών σε ένα γονίδιο προσδιορίζοντας έναν πίνακα με τα σύνολα των πεδίων τιμών. Το πεδίο τιμών ορίζει το σύνολο των τιμών που το κάθε στοιχείο στο γονιδίωμα μπορεί να αναπαραστήσει. Ένα γονιδίωμα από πραγματικούς αριθμούς είναι μία δομική οργανική προσαρμογή ενός πίνακα γονιδιώματος με πεδία τιμών. Η δομική οργανική προσαρμογή είναι τύπου κινητής υποδιαστολής.

Θα πρέπει να δημιουργήσουμε ένα πεδίο ορισμού ή έναν πίνακα από πεδία ορισμού πριν εφαρμόσουμε αυτόν τον τύπο γονιδιώματος. Σε περίπτωση που δημιουργήσουμε ένα γονιδίωμα που λαμβάνει τιμές από ένα μόνο πεδίο ορισμού, τότε κάθε στοιχείο του γονιδιώματος θα χρησιμοποιεί αυτό το πεδίο του ορισμού για να προσδιορίσει την τιμή του. Σε περίπτωση που δημιουργήσουμε ένα γονιδίωμα πραγματικών αριθμών χρησιμοποιώντας ένα πίνακα τότε το γονίδιο θα έχει μήκος ίσο με τον αριθμό των στοιχείων του πίνακα και κάθε στοιχείο πραγματικών αριθμών θα καλύπτεται το αντίστοιχο πεδίο τιμών του πίνακα ανάλογα με την θέση του στο γονιδίωμα.

Για να χρησιμοποιήσουμε αυτό τον τύπο γονιδίου στον κώδικα που αναπτύσσουμε, θα πρέπει να συμπεριλάβουμε τα αρχεία επικεφαλίδων – header σε κάθε ένα από τα αρχεία που δημιουργούμε και χρησιμοποιούν το γονιδίωμα. Επίσης θα πρέπει να συμπεριλάβουμε το αρχείο με τον πηγαίο κώδικα σε μόνο ένα από τα αρχεία κώδικα. Αν δεν συμπεριλάβουμε ορισμένες μέθοδοι των πεδίων ορισμού δεν θα λειτουργούν σωστά.

## **GAStringGenome**

Το γονιδίωμα String μπορεί να χρησιμοποιηθεί σε εφαρμογές με σταθερού και μεταβλητού μήκους String ή σε πεδία τιμών με αλφαβητικά στοιχεία. Το allele set ορίζει το σύνολο των πιθανών τιμών που μπορεί να πάρει ένα String στην διάρκεια εκτέλεσης του αλγορίθμου.

Το γονιδίωμα String είναι μία δομική προσαρμογή του πίνακα του γονιδιώματος με τα πεδία ορισμού. Η προσαρμογή είναι τύπου χαρακτήρα, ομοίως και με τα προηγούμενα θα πρέπει να δημιουργήσουμε ένα πεδίο τιμών ή έναν πίνακα από τα σύνολα των πεδίων τιμών προτού χρησιμοποιήσουμε αυτών τον τύπο σε κάποιο πρόγραμμα. Σε περίπτωση που δημιουργήσουμε ένα γονιδίωμα με ένα απλό σύνολο τιμών, τότε κάθε στοιχείο του γονιδίου θα έχει ως σημείο αναφοράς αυτό το πεδίο ορισμού και το σύνολο των τιμών που ορίζει. Αν δημιουργήσουμε ένα γονιδίωμα και χρησιμοποιήσουμε έναν πίνακα από πεδία τιμών, τότε το string θα έχει το ίδιο μήκος με τον αριθμό των στοιχείων που βρίσκονται στον πίνακα, και κάθε στοιχείο του string θα καλύπτεται από το αντίστοιχο σύνολο τιμών – allele set σε αντιστοιχία με αυτό που βρίσκεται στον πίνακα.

Προκειμένου να χρησιμοποιήσουμε το γονιδίωμα string, θα πρέπει πρώτα να συμπεριλάβουμε το header αρχείο σε όλα τα αρχεία που θα δημιουργήσουμε και που χρησιμοποιούν το string. Επίσης θα πρέπει να

συμπεριλάβουμε το αρχείο που περιέχει τον πηγαίο κώδικα του string. Σε περίπτωση που δεν συμπεριλάβουμε τον πηγαίο κώδικα του string κάποιες από τις μεθόδους των allele set δεν θα λειτουργούν.

### **GATreeGenome<T>**

Το γονιδίωμα που περιγράφει την δεντροειδή δομή δεδομένων είναι μία κατηγορία πρότυπο. Μπορεί να αναπτυχθεί από την κατηγορία GAGenome και από την κατηγορία GATree<>. Η δεντροειδή δομή δεδομένων μας παρέχει ευκολότερο χειρισμό ενός δεντροειδούς αντικειμένου.

Για να χρησιμοποιήσουμε αυτού του τύπου τις κατηγορίες θα πρέπει να ορίσουμε έναν τελεστή αρχικοποίησης. Σε περίπτωση που δεν ορίσουμε κάποιο τελεστή αρχικοποίησης θα δεχόμαστε λάθη αρχικοποίησης όποτε ο αλγόριθμος προσπαθεί να αρχικοποιήσει κάποιο αντικείμενο.

### **GAEvalData**

Ένα από τα κυριότερα αντικείμενα στα οποία βασίζεται η βασική κατηγορία GAGenome είναι το αντικείμενο της αποτίμησης των δεδομένων. Παρόλο που τα userData ενός γονιδιώματος είναι κοινά για όλα τα γονιδιώματα σε έναν πληθυσμό, τα evalData είναι μοναδικά για τα γονιδιώματα. Η βασική κατηγορία προσδιορίζει την μέθοδο αντιγραφής και clone για το αντικείμενο αποτίμησης των δεδομένων. Οι κατηγορίες που αναπτύσσουμε θα πρέπει να χρησιμοποιούν αυτό τον μηχανισμό, σε κάθε κατηγορία που αναπτύσσουμε θα πρέπει να ορίζουμε συναρτήσεις μέλη αντιγραφής και clone. Αυτές οι συναρτήσεις θα πρέπει να καλούνται από την βασική κατηγορία όταν συμβαίνει μία εξέλιξη και τα δεδομένα του γονιδιώματος ή του πληθυσμού αντιγράφονται ή κλωνοποιούνται.

### **GAAlleleSet<T>**

Η κατηγορία που περιγράφει ένα allele set είναι μία συλλογή από διαφορετικές τιμές που μπορεί να πάρει ένα γονίδιο. Σε αυτήν την κατηγορία μπορούμε να συμπεριλάβουμε αντικείμενα κάθε τύπου όπως τους τελεστές =, ==, !=, και άλλους.

Στα σύνολα Allele sets μπορεί να υπάρχει ένας απαριθμητός αριθμός δεδομένων, ένας δεσμευτικός αριθμός δεδομένων, ή δεσμευτικός αλλά διακριτός αριθμός δεδομένων. Για παράδειγμα ένα Allele set από ακέραιες τιμές μπορεί να οριστεί ως {1,3,5,2,99,-53} (Απαριθμητός αριθμός δεδομένων). Ένας δεσμευτικός πραγματικός αριθμός μπορεί να οριστεί ως [2,743), δηλαδή το σύνολο των αριθμών από το δύο συμπεριλαμβανομένων μέχρι το 743 αποκλειστικά. Ένα δεσμευτικό αλλά διακριτό αριθμό δεδομένων μπορεί να οριστεί ως [4.5,7.05](0.05), δηλαδή το σύνολο των αριθμών από 4,5 μέχρι 7,5 να συμπεριληφθούν με αύξηση 0,05.

Σε περίπτωση που καλέσουμε την συνάρτηση μέλος allele με κανένα όρισμα, τότε το allele set σύνολο επιλέγεται τυχαία από τα σύνολα που διαθέτει και επιστρέφει κάποιο στην τύχη.

### **GAAlleleSetArray<T>**

Τα GAAlleleSetArray είναι ένα σύνολο αντικειμένων με έναν πίνακα από αντικείμενα από allele set.

### **GAParameter and GAParameterList**

Το αντικείμενο λίστας παραμέτρων περιέχει πληροφορία σχετικά με τον τρόπο θα πρέπει να συμπεριφέρεται ο γενετικός αλγόριθμος. Κάθε λίστα παραμέτρων περιέχει έναν πίνακα από παραμέτρους.

Κάθε παράμετρος είναι ένα ζευγάρι ονόματος – τιμής, όπου το όνομα είναι ένα string και η τιμή είναι ένας ακέραιος, κινητής υποδιαστολής, χαρακτήρας, Boolean, ακόμη και δείκτης. Κάθε παράμετρος είναι μοναδικά αναγνωρισμένη από το ζεύγος ονομάτων : ένα πλήρες όνομα και ένα βραχύ όνομα. Μαζί με το κάθε όνομα υπάρχει και μία τιμή. Κάθε παράμετρος έχει επίσης μία λίστα από απαριθμητούς τύπους δεδομένων. Το αντικείμενο GParameter

### **GAStatistics**

Το αντικείμενο statistics κρατάει πληροφορίες σχετικά με την τρέχουσα κατάσταση του αντικειμένου του γενετικού αλγορίθμου. Όλα τα αντικείμενα του γενετικού αλγορίθμου περιέχουν αντικείμενα αρμόδια για τα στατιστικά στοιχεία.

Σε αυτά τα αντικείμενα ορίζουμε κάποιους σταθερούς απαριθμητούς τύπους δεδομένων προκειμένου να χρησιμοποιηθούν από το μέλος selectScores. Μπορεί να περάσουν από την bitwise – OR για να προσδιορίσουν τον επιθυμητό συνδυασμό στοιχείων. Μπορούμε επίσης να χρησιμοποιήσουμε την κατηγορία name για να αναφερθούμε στις τιμές αυτές για παράδειγμα : GAStatistics::Mean | GAStatistics::Deviation

### **GAPopulation**

Το αντικείμενο του πληθυσμού είναι μία συλλογή για τα αντικείμενα των γονιδιωμάτων, επίσης κρατάει κάποια στατιστικά στοιχεία για τον εκάστοτε πληθυσμό, όπως για παράδειγμα τον μέσο όρο, την μέγιστη και την ελάχιστη αντικειμενική τιμή του γονιδίου. Κάθε πληθυσμός περιέχει κάποιο αντικείμενο υπεύθυνο για την κλιμάκωση της τιμής καταλληλότητας, το οποίο χρησιμοποιείται για να ορίσει την τιμή καταλληλότητας των γονιδίων. Επίσης περιέχει κάποια συνάρτηση για την επιλογή ατόμων από τον πληθυσμό.

Μπορούμε να προσαρμόσουμε τις μεθόδους της αρχικοποίησης του πληθυσμού, της αποτίμησης καθώς επίσης και τις μεθόδους ταξινόμησης. Οι δικές μας κατηγορίες αντικειμένων θα πρέπει να χρησιμοποιούν το κατάλληλο δέσιμο μεταξύ τους. Η προεπιλεγμένη μέθοδος κλιμάκωσης είναι η Γραμμική Κλιμάκωση. Επίσης η προεπιλεγμένη συνάρτησης αποτίμησης επιδρά σε όλα τα γονίδια ξεχωριστά. Τέλος η προεπιλεγμένη μέθοδος Επιλογής είναι η Μέθοδος του Τροχού της Τύχης και χρησιμοποιεί τις τιμές που προέρχονται από την κλιμάκωσης των τιμών καταλληλότητας.

### **GAScalingScheme**

Το αντικείμενο της κλιμάκωσης εμπεριέχεται στην κατηγορία του πληθυσμού. Το βασικό αντικείμενο κλιμάκωσης δεν είναι άμεσα εφαρμόσιμο σε έναν γενετικό αλγόριθμο. Αυτό το αντικείμενο κρατάει στοιχεία για την τιμή των τιμών καταλληλότητας για κάθε μονάδα του πληθυσμού. Τα γονίδια που επιστρέφει είναι γονίδια από τον πληθυσμό στον οποίο είναι συνδεδεμένα, δεν κρατάει δικά του αντίγραφα.

### **GASelectionScheme**

Τα σχήματα Επιλογής χρησιμοποιούνται για να επιλέγουν γονίδια από τον πληθυσμό για ζευγάρωμα. Το αντικείμενο GASelectionScheme καθορίζει την βασική συμπεριφορά που θα έχει μία συνάρτηση Επιλογής. Κάθε αντικείμενο Επιλογής μπορεί να συνδεθεί με κάποιον πληθυσμό από τον οποίο θα πρέπει να κάνει τις επιλογές του. Όταν κάποιο μέλος επιλεγεί θα επιστραφεί μία αναφορά σε εκείνο το γονίδιο.



Στην βασική κατηγορία καθορίζει της σταθερές με τις οποίες προσδιορίζουμε πού θα γίνονται οι παρεμβολές νέων κόμβων. Αυτό είναι σχετικό με την τρέχουσα κατάσταση στον κόμβο όπου δείχνει η επαναληπτική διαδικασία.

Οι κόμβοι στην λίστα αριθμούνται από το 0 στο 1, δηλαδή αύξουσα και έχουν μέγεθος μικρότερο της λίστας. Ο αρχικός κόμβος είναι ο κόμβος 0.

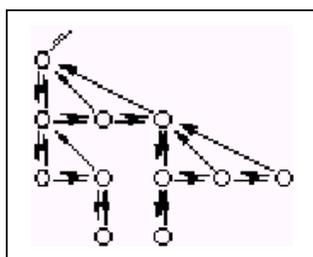
Όταν κάνουμε κάποια εισαγωγή η λίστα κάνει ένα αντίγραφο του αντικειμένου που ορίσαμε, κατανέμοντας χώρο στην διαδικασία για το νέο αντικείμενο. Το αντίγραφο του αντικειμένου δημιουργείται χρησιμοποιώντας την συνάρτηση δημιουργίας αντιγράφου. Η επαναληπτική διαδικασία θα δείχνει προς τα αριστερά το νέο μέλος. Το μέλος εισάγεται σύμφωνα με την τρέχουσα θέση της λίστας της εσωτερικής επαναληπτικής λίστας. Μπορούμε να χρησιμοποιήσουμε την σημαία `where` προκειμένου να ορίσουμε την θέση που θα εισαχθεί το νέο μέλος πριν ή μετά το τρέχον κόμβο ή αν αυτός ο κόμβος γίνει ο πρώτος κόμβος της λίστας.

Η αφαίρεση κάποιου μέλους από την λίστα γίνεται με την υπόδειξη του με κάποιον δείκτη στον κόμβο όπου βρίσκεται το στοιχείο. Είμαστε υπεύθυνοι για τον αναδιανομή του χώρου μνήμης που καταλάμβανε το αντικείμενο. Το μέλος Εξολόθρευσης του τρέχοντος κόμβου είναι υπεύθυνο για την αναδιανομή της μνήμης.

### **GATree<T> and GATreeIter<T>**

Το αντικείμενο `GATree<T>` ορίζεται για δική μας άνεση, έτσι δεν είναι απαραίτητο να ορίσουμε εμείς το δικό μας αντικείμενο *δέντρου*. Αυτή η κατηγορία είναι μία συλλογή προτύπων του οποίου οι κόμβοι μπορούν να συμπεριλάβουν αντικείμενα οποιουδήποτε τύπου. Κάθε επίπεδο του δέντρου είναι κυκλικό και διπλά συνδεδεμένο. Το μεγαλύτερο παιδί ενός επιπέδου είναι η κεφαλή της λίστας διασυνδέσεων. Κάθε παιδί σε κάποιο επίπεδο αναφέρεται στον πατέρα του, ο πατέρας αυτών των παιδιών αναφέρεται στον αμέσως ανώτερο, τον πατέρα του.

Κάθε δέντρο μπορεί να έχει μόνον έναν κόμβο ρίζα. Κάθε κόμβος μπορεί να έχει οποιονδήποτε αριθμό παιδιών. Σε ένα δέντρο ένας επαναλήπτης ορίζεται προκειμένου να κινούμαστε ανάμεσα στην λίστα και να κρατάμε στοιχεία του τρέχοντος, του προηγούμενου, του επόμενου κόμβου. Ένας επαναλήπτης δεν



πρέπει να αλλάξει την κατάσταση του δέντρου.

Οι κύκλοι είναι οι κόμβοι του δέντρου. Κάθε κόμβος περιέχει ένα αντικείμενο. Η μέθοδος αρχικοποίησης ορίζει την τοπολογία και τα περιεχόμενα κάθε κόμβου του δέντρου. Κάθε δέντρο έχει μόνον μία ρίζα.

Κάθε επίπεδο σε ένα δέντρο είναι κυκλικά και προς τις δύο κατευθύνσεις συνδεδεμένα. Η κεφαλίδα κάθε λίστας ονομάζεται 'eldest' παιδί, κάθε

κόμβος στην κορυφή έχει μία σύνδεση προς τον πατέρα του. Ο πατέρας έχει τόσες συνδέσεις όσες χρειάζεται προς τα παιδιά του.

Το πρότυπο `GATree<T>` αναπτύσσεται από την γενετική βασική συνάρτηση που περιγράφει ένα δέντρο και ονομάζεται `GATreeBASE`. Η συνάρτηση που περιγράφει το πρότυπο δέντρο ορίζεται στο αρχείο `treetmpl.h`, η βασική κατηγορία δέντρο ορίζεται στο αρχείο `treebase.h`. Όποιο αντικείμενο χρησιμοποιηθεί στους κόμβους θα πρέπει να έχει προσδιορισμένες τις ακόλουθες μεθόδους και θα πρέπει να είναι δημόσια διαθέσιμες :

copy constructor

operator ==

operator =

operator !=

Κάθε αντικείμενο του δέντρου περιέχει έναν επαναλήπτη. Το μέλος της συνάρτησης της λίστας `traversal(next, prev, etc)` καλεί την συνάρτηση μέλος της εσωτερικής επανάληψης. Μπορούμε να ορίσουμε έναν εξωτερικό επαναλήπτη στο αντικείμενο του δέντρου έτσι μπορούμε να διερχόμαστε των μελών χωρίς να τροποποιούμε την κατάσταση του.

Όταν επιθυμούμε να κάνουμε κάποια εισαγωγή στο δέντρο, τότε δημιουργούμε ένα αντίγραφο του αντικειμένου που ορίσαμε, κατανέμοντας χώρο στην διαδικασία για το νέο αντικείμενο. Το αντίγραφο του αντικειμένου δημιουργείται χρησιμοποιώντας την συνάρτηση δημιουργίας αντιγράφου. Η επαναληπτική διαδικασία θα δείχνει προς τα αριστερά το νέο μέλος. Το μέλος εισάγεται σύμφωνα με την τρέχουσα θέση της εσωτερικής επαναληπτικής λίστας. Μπορούμε να χρησιμοποιήσουμε την σημαία `where` προκειμένου να ορίσουμε την θέση που θα εισαχθεί το νέο μέλος πριν ή μετά το τρέχον κόμβο ή αν αυτός ο κόμβος γίνει ο πρώτος κόμβος της λίστας.

Η αφαίρεση κάποιου μέλους από την λίστα γίνεται με την υπόδειξη του με κάποιον δείκτη στον κόμβο όπου βρίσκεται το στοιχείο. Είμαστε υπεύθυνοι για τον αναδιανομή του χώρου μνήμης που καταλάμβανε το αντικείμενο. Το μέλος Εξολόθρευσης του τρέχοντος κόμβου είναι υπεύθυνο για την αναδιανομή της μνήμης.

#### 1.4. Προσαρμογή της βιβλιοθήκης GAlib.

Σε αυτό το σημείο του κειμένου θα αναφερθούμε στον τρόπο με τον οποίο μπορούμε να προεκτείνουμε της δυνατότητες της βιβλιοθήκης ορίζοντας τους δικούς μας τελεστές, τα δικά μας γονιδιώματα. Ο καλύτερος τρόπος για να επιτύχουμε αυτό είναι να δημιουργήσουμε καινούργιες κατηγορίες. Μπορούμε όμως να τροποποιήσουμε την συμπεριφορά των ήδη υφιστάμενων συναρτήσεων που διαθέτει η βιβλιοθήκη.

##### 1.4.1. Αναπτύσσοντας την δική μας κατηγορία του Γονιδίου.

Μπορούμε να δημιουργήσουμε την δική μας κατηγορία περιγραφής ενός Γονιδίου κληρονομώντας κάποια στοιχεία από την βασική κατηγορία του `genome` και τον τύπο δομής δεδομένων που έχουμε επιλέξει. Ένα παράδειγμα, έστω ότι έχουμε ορίσει κάποιο αντικείμενο `MyObject` και έπειτα αναπτύσσουμε μία καινούργια κατηγορία με όνομα `MyGenome` της οποίας η δομή είναι η παρακάτω :

```
class MyGenome : public MyObject, public GAGenome {
public:
    GADefineIdentity("MyGenome", 201);
    static void Init(GAGenome&);
    static int Mutate(GAGenome&, float);
    static float Compare(const GAGenome&, const GAGenome&);
    static float Evaluate(GAGenome&);
    static int Cross(const GAGenome&, const GAGenome&, GAGenome*, GAGenome*);
public:
    MyGenome() : GAGenome(Init, Mutate, Compare) {
        evaluator(Evaluate);
        crossover(Cross);
    }
    MyGenome(const MyGenome& orig) { copy(orig); }
    virtual ~MyGenome() {}
    MyGenome& operator=(const GAGenome& orig){
        if(&orig != this) copy(orig);    return *this;    }
    virtual GAGenome* clone(CloneMethod) const {return new MyGenome(*this);}
```

```

virtual void copy(const GAGenome& orig) {
GAGenome::copy(orig); // this copies all of the base genome parts
// copy any parts of MyObject here      copy any parts of MyGenome here
// any data/member functions specific to this new class };
void MyGenome::Init(GAGenome&) { // your initializer here
} int MyGenome::Mutate(GAGenome&, float){ // your mutator here
} float MyGenome::Compare(const GAGenome&, const GAGenome&){
// Customizing GAlib: Deriving your own genome class your comparison here
} float MyGenome::Evaluate(GAGenome&){ // your evaluation here
} int MyGenome::Cross(const GAGenome& mom, const GAGenome& dad, GAGenome* sis, GAGenome*
bro){ // your crossover here }

```

Συνηθίζεται σαν ένα από τα ορίσματα σε κάποιο γονίδιο που αναπτύσσουμε να είναι η αντικειμενική διαδικασία. Εναλλακτικά, μπορούμε να ορίσουμε μία προεπιλεγμένη διαδικασία μέσα στο γονίδιο. Αφού αναπτύξουμε την κατηγορία του γονιδίου θα πρέπει έπειτα να ορίσουμε την συνάρτηση Αρχικοποίησης, την συνάρτηση Μετάλλαξης, την συνάρτηση Σύγκρισης και τους τελεστές Διασταύρωσης για την κατηγορία. Ο τελεστής Σύγκρισης είναι προαιρετικός, αλλά αν δεν τον ορίσουμε δεν θα μπορούμε να έχουμε ποικίλα μέτρα ποιότητας στον αλγόριθμο.

Πρέπει να τονίσουμε ότι οι γενετικοί τελεστές σύγκρισης δεν είναι όμοιοι με τους Boolean τελεστές == και != . Οι γενετικοί τελεστές σύγκρισης επιστρέφουν 0 αν τα μέλη είναι ίδια, -1 αν η σύγκριση αποτύχει για οποιονδήποτε λόγο, και κάποιον ακέραιο αριθμό μεγαλύτερο του μηδενός (0) που θα υποδηλώνει τον βαθμό που θα διαφέρουν τα μέλη.

Προκειμένου να εργαστούμε σωστά με την GAlib πρέπει να ορίσουμε τα εξής :

```

MyGenome( -default-args-for-your-genome-constructor )
MyGenome(const MyGenome&)
virtual GAGenome* clone(GAGenome::CloneMethod) const

```

Όταν αναπτύσσουμε το δικό μας γονίδιο πρέπει να χρησιμοποιούμε την σημαία \_evaluated οι οποία δείχνει ότι η κατάσταση του γονιδίου έχει αλλάξει χρειάζεται εξέλιξη. Αν η συνάρτηση μέλος αλλάξει την κατάσταση του γονιδίου, η συνάρτηση μέλος θα πρέπει να θέσει την σημαία \_evaluated με τιμή gaFalse. Αν η συνάρτηση μέλος δεν ρυθμίσει την σημαία και αλλάξουν τα περιεχόμενα του, τότε το γονίδιο δεν θα υποστεί την διαδικασία της εξέλιξης.

Μπορούμε να χρησιμοποιήσουμε τους προεπιλεγμένους τελεστές Διασταύρωσης, Μετάλλαξης, Αρχικοποίησης και Σύγκρισης .

#### 1.4.2. Συνάρτηση Αρχικοποίησης του Γονιδίου.

Η Συνάρτηση Αρχικοποίησης δέχεται μόνο ένα όρισμα : το γονίδιο που θέλουμε να αρχικοποιήσουμε. Το γονίδιο έχει ήδη καταναμηθεί, ο αρχικοποιητής χρειάζεται για να γεμίσει το γονίδιο με το απαραίτητο γενετικό υλικό. Παρακάτω δίνουμε ένα παράδειγμα μίας συνάρτησης Αρχικοποίησης για την δομή δεδομένων GATreeGenome<int> :

```

void TreeInitializer(GAGenome & c) { GATreeGenome<int> &child=(GATreeGenome<int> &)c;
// destroy any pre-existing tree
child.root(); child.destroy(); // Create a new tree with depth of 'depth' and each eldest node containing 'n'
// children (the other siblings have none).
int depth=2, n=3, count=0;
child.insert(count++,GATreeBASE::ROOT);
for(int i=0; i<depth; i++){ child.eldest(); child.insert(count++);
for(int j=0; j<n; j++) child.insert(count++,GATreeBASE::AFTER); }}

```

#### 1.4.3. Συνάρτηση τελεστή Μετάλλαξης.

Η συνάρτηση Μετάλλαξης δέχεται δύο ορίσματα : το γονίδιο που θα υποστεί την Μετάλλαξη, και την πιθανότητα Μετάλλαξης. Το ακριβές νόημα του Μεταλλάκτη υπόκειται στον δημιουργό της συνάρτησης. Η συνάρτηση μας επιστρέφει τον αριθμό των Μεταλλάξεων που συνέβησαν στο γονίδιο.

Οι περισσότεροι γενετικοί αλγόριθμοι εφαρμόζουν τον τελεστή Μετάλλαξης στους καινούργιους απογόνους. Οι τελεστές αυτοί βασίζονται στις ενέργειες τους στην τιμή της πιθανότητας μετάλλαξης. Για παράδειγμα, αν έχουμε έναν πίνακα με δεκαδικούς με δοσμένη την πιθανότητα Μετάλλαξης (pmut) να λαμβάνει τιμές σύμφωνα με μία στοχαστική διαδικασία, ανάλογη του ριζιματος ενός νομίσματος, για όλες τις θέσεις του πίνακα.

Παρακάτω δίνουμε το παράδειγμα μετάλλαξης ενός στοιχείου GA1DBinaryString. Αυτός ο Μεταλλάκτης αναστρέφει την τιμή του bit σε ένα string ανάλογα με το γύρισμα ενός νομίσματος.

```
int GA1DBinStrFlipMutator(GAGenome & c, float pmut) {
    GA1DBinaryStringGenome &child=(GA1DBinaryStringGenome &)c;
    if(pmut <= 0.0) return(0);
    int nMut=0;
    for(int i=child.length()-1; i>=0; i--){
        if(GAFlipCoin(pmut)){ child.gene(i, ((child.gene(i) == 0) ? 1 : 0));
            nMut++; } }
    return nMut; }
```

#### 1.4.4. Συνάρτηση τελεστή Διασταύρωσης.

Η μέθοδος Διασταύρωσης χρησιμοποιείται από το αντικείμενο του γενετικού αλγορίθμου προκειμένου να ζευγαρώσουν μέλη από κάποιον πληθυσμό. Σε κάθε γονίδιο θα πρέπει να βρίσκεται ορισμένη μία διαδικασία Διασταύρωσης που να μπορεί να χρησιμοποιήσει ο γενετικός αλγόριθμος. Σε περίπτωση που στη κατηγορία μας δεν ορίζουμε μία μέθοδο Διασταύρωσης τότε θα λαμβάνουμε μηνύματα λάθους όταν επιχειρείται διασταύρωση.

Συνήθως, η συνάρτηση διασταύρωσης αλλάζει σύμφωνα με την επιθυμητή μέθοδο ζευγαρώματος, στα γονίδια η συνάρτηση μέλος δεν υλοποιεί την διασταύρωση, μόνο ο γενετικός αλγόριθμος είναι αρμόδιος για την εκτέλεση της Διασταύρωσης.

Μερικοί αλγόριθμοι χρησιμοποιούν sexual ζευγάρι, ενώ κάποιοι άλλοι asexual. Μπορούμε να ορίσουμε και τους δύο τρόπους. Στην Sexual Διασταύρωση η συνάρτηση δέχεται τέσσερα ορίσματα : τους δύο γονείς και τα δύο παιδιά. Αν ορίσουμε ένα παιδί τότε ο τελεστής θα επιστρέψει έναν απόγονο. Έτσι η συνάρτηση θα πρέπει να μπορεί να δουλέψει σε ένα ή δύο απογόνους. Η συνάρτηση Διασταύρωσης θα πρέπει να επιστρέφει τον αριθμό των διασταυρώσεων που έγιναν.

Παρακάτω βλέπουμε μία εφαρμογή στην οποία συμμετέχουν δύο γονείς. Χρησιμοποιούμε την μέθοδο Διασταύρωσης ενός σημείου και παράγουμε σταθερού μήκους απογόνους. Τα γονίδια είναι τύπου GA1DBinaryStringGenome.

```
int SinglePointCrossover(const GAGenome& p1, const GAGenome& p2, GAGenome* c1, GAGenome* c2){
    GA1DBinaryStringGenome &mom=(GA1DBinaryStringGenome &)p1;
    GA1DBinaryStringGenome &dad=(GA1DBinaryStringGenome &)p2;
    int n=0;
    unsigned int site = GARandomInt(0, mom.length());
    unsigned int len = mom.length() - site;
    if(c1){ GA1DBinaryStringGenome &sis=(GA1DBinaryStringGenome &)*c1;
        sis.copy(mom, 0, 0, site); sis.copy(dad, site, site, len);
        n++; }
    if(c2){ GA1DBinaryStringGenome &bro=(GA1DBinaryStringGenome &)*c2;
```

```

        bro.copy(dad, 0, 0, site);        bro.copy(mom, site, site, len);
        n++; }
return n; }

```

#### 1.4.5. Συνάρτηση Σύγκρισης.

Η συνάρτηση Σύγκρισης χρησιμοποιείται για μία ποικιλία υπολογισμών. Συγκρίνει δύο γονίδια και επιστρέφει κάποιον αριθμό μεγαλύτερο ή είναι ίσο του μηδενός. Αν η τιμή είναι 0 τότε τα γονίδια είναι ταυτόσημα. Δεν υπάρχει κάποια μέγιστη τιμή η οποία να ορίζεται στην συνάρτηση Σύγκρισης. Σε περίπτωση που η τιμή είναι -1 αυτό δηλώνει ότι για κάποιον λόγο δεν μπόρεσε η συνάρτηση να συγκρίνει τα δύο γονίδια. Παρακάτω δείχνουμε την συνάρτηση Σύγκρισης για γονιδιώματα Δυαδικής πλειάδας :

```

float GAIDBinStrComparator(const GAGenome& a, const GAGenome& b){ GAIDBinaryStringGenome
&sis=(GAIDBinaryStringGenome &a); GAIDBinaryStringGenome
&bro=(GAIDBinaryStringGenome &b);
if(sis.length() != bro.length()) return -1;
float count = 0.0;
Customizing GALib: Genome Evaluation
for(int i=sis.length()-1; i>=0; i--) count += ((sis.gene(i) == bro.gene(i)) ? 0 : 1);
return count/sis.length(); }

```

Σε αυτό το παράδειγμα ο κώδικας απλά μετράει τα δυαδικά bit που έχουν στις δομές τους τα δύο γονιδιώματα. Η συνάρτηση επιστρέφει -1 όταν τα γονιδιώματα δεν έχουν το ίδιο μήκος.

#### 1.4.6. Συνάρτηση Εξέλιξης.

Η συνάρτηση εξέλιξης είναι η αντικειμενική διαδικασία για τον πληθυσμό. Δέχεται κάποιο γονίδιο σαν παράμετρο της. Η συνάρτηση επιστρέφει έναν αριθμό που δηλώνει αν είναι καλό ή κακό το γονίδιο. Θα πρέπει να προβλέψουμε ένα γενικό γονιδίωμα για τον τύπο του γονιδίου που θα χρησιμοποιήσουμε. Αν η συνάρτηση αποτίμησης δουλεύει μα διαφορετικά γονιδιώματα τότε θα πρέπει να χρησιμοποιήσουμε αντικείμενα γονιδιωμάτων με όνομα className και / ή classID μέλη συναρτήσεων που θα ορίζουν την κατηγορία του γονιδίου προτού κάνουμε την πρόβλεψη.

Παρακάτω βλέπουμε μία συνάρτηση που περιγράφει την απλή εξέλιξη ενός γονιδιώματος πραγματικών αριθμών. Αυτή η συνάρτηση προσπαθεί να μεγιστοποιήσει το ημίτονο.

```

float Objective(GAGenome& g){
    GARealGenome& genome = (GARealGenome &g);
    return 1 + sin(genome.gene(0)*2*M_PI); }

```

#### 1.4.7. Συνάρτηση Πληθυσμού.

Αυτή η συνάρτηση εφαρμόζεται όταν ο πληθυσμός αρχικοποιείται. Παρακάτω δείχνουμε μία εφαρμογή που εφαρμόζει την συνάρτηση αρχικοποίησης σε κάθε γονιδίωμα του πληθυσμού.

```

void PopInitializer(GAPopulation & p){
    for(int i=0; i<p.size(); i++)
        p.individual(i).initialize(); }

```

#### 1.4.8. Συνάρτηση Αποτίμησης.

Αυτή η συνάρτηση εφαρμόζεται όταν ο πληθυσμός αποτιμάται. Αν ο αντικειμενικός σκοπός είναι βασισμένος στον πληθυσμό τότε μπορούμε να χρησιμοποιήσουμε αυτή την μέθοδο ώστε να θέσουμε μία τιμή για κάθε γονιδίωμα από το να αναπτύξουμε μία συνάρτηση αποτίμησης για κάθε γονίδιο. Παρακάτω αναπτύσσουμε μία υλοποίηση για κάθε γονίδιο σε κάποιον πληθυσμό.

```

void PopEvaluator(GAPopulation & p){    for(int i=0; i<p.size(); i++)
    p.individual(i).evaluate(); }

```

### 1.4.9. Συνάρτηση Κλιμάκωσης.

Το αντικείμενο την κλιμάκωσης είναι υπεύθυνο για τον μετασχηματισμό της αντικειμενικής τιμής ενός μέλους σε τιμή καταλληλότητας. Η περισσότερο σημαντική συνάρτηση μέλος που θα πρέπει να ορίσει το νέο αντικείμενο κλιμάκωσης είναι η συνάρτηση μέλος Αποτίμησης. Αυτή η συνάρτηση υπολογίζει την τιμή καταλληλότητας του κάθε μέλους, βασίζεται στην τιμή αποτίμησης που έχουν όλα τα μέλη ενός πληθυσμού.

Η κατηγορία GAScalingScheme είναι μία εικονική κατηγορία και δεν μπορεί να εφαρμοστεί άμεσα σε κάποιο πρόγραμμα. Προκειμένου να χρησιμοποιηθεί θα πρέπει να ορίσουμε μία συνάρτηση clone και κάποια συνάρτηση Αποτίμησης. Πρέπει επίσης να ορίσουμε μία συνάρτηση αντιγραφής – copy αν η αναπτυσσόμενη μέθοδος συνιστά κάποια επιπρόσθετη όχι εικονική μέθοδο αντιγραφής.

Προσαρμόζοντας την βιβλιοθήκη της GAlib : Σχήμα Επιλογής. Η συνάρτηση κλιμάκωσης είναι πολυμορφική, έτσι θα πρέπει να ορίσουμε κάποιο αντικείμενο ομοιότητας χρησιμοποιώντας την μακροεντολή GADefineIdentity. Αυτή καταχωρεί έναν αριθμό ID και το όνομα που θα χρησιμοποιηθεί σε μηνύματα λάθους για την κατηγορία. Μπορούμε να χρησιμοποιήσουμε οποιονδήποτε αριθμό κάτω από 200, αλλά θα πρέπει να είμαστε σίγουροι ότι είναι διαφορετικοί για κάθε ένα από τα αντικείμενα.

Παρακάτω βρίσκεται μία εφαρμογή της μεθόδου κλιμάκωσης sigma truncation :

```
class SigmaTruncationScaling : public GAScalingScheme {
public:
    GADefineIdentity("SigmaTruncationScaling", 286);
    SigmaTruncationScaling(float m=gaDefSigmaTruncationMultiplier) : c(m) {}
    SigmaTruncationScaling(const SigmaTruncationScaling & arg) {copy(arg);}
    SigmaTruncationScaling & operator=(const GAScalingScheme & arg) {copy(arg); return *this; }
    virtual ~SigmaTruncationScaling() {}
    virtual GAScalingScheme * clone() const { return new SigmaTruncationScaling(*this); }
    virtual void evaluate(const GAPopulation & p);
    virtual void copy(const GAScalingScheme & arg) { if(&arg != this && sameClass(arg)) {
        GAScalingScheme::copy(arg); c=((SigmaTruncationScaling&)arg).c; } }
    float multiplier(float fm) { return c=fm; }
    float multiplier() const { return c; }
protected:
    float c; }; // std deviation multiplier
void SigmaTruncationScaling::evaluate(const GAPopulation & p) {
    float f;
    for(int i=0; i<p.size(); i++){
        f = p.individual(i).score() - p.ave() + c * p.dev();
        if(f < 0) f = 0;    p.individual(i).fitness(f); } }
```

### 1.4.10. Συνάρτηση Επιλογής.

Η συνάρτηση Επιλογής χρησιμοποιείται για να επιλέξει εκείνες τις μονάδες από τον πληθυσμό προκειμένου να ζευγαρώσουν. Πριν από την Επιλογή καλείται η μέθοδος update. Μπορούμε να χρησιμοποιήσουμε αυτήν την μέθοδο για να κάνουμε όποιον πριν την επιλογής μετασχηματισμό θεωρούμε ότι είναι καλύτερος για το σχήμα Επιλογής που χρησιμοποιούμε

Όταν απαιτείται κάποια επιλογή τότε η μέθοδος καλείται. Η συνάρτηση θα πρέπει να επιστρέψει μία αναφορά στην μονάδα του πληθυσμού. Η μέθοδος Επιλογής κάνει τις επιλογές τις βασίζομενη είτε στην τιμή καταλληλότητας των μονάδων, είτε στην αντικειμενική τιμή που αυτά έχουν στον πληθυσμό.

Πρέπει να σημειώσουμε ότι ο πληθυσμός μπορεί να ταξινομηθεί και από την μικρότερη προς την μεγαλύτερη τιμή και το αντίστροφο, ανάλογα με την μέθοδο επιλογής που χρησιμοποιείται. Η

συνάρτηση επιλογής θα πρέπει να είναι έτοιμος να κάνει τις επιλογές του και στις δύο δυνατές περιπτώσεις.

Η συνάρτηση επιλογής είναι πολυμορφική, έτσι θα πρέπει να ορίσουμε κάποιο αντικείμενο ομοιότητας χρησιμοποιώντας την μακροεντολή GADefineIdentity. Αυτή καταχωρεί έναν αριθμό ID και το όνομα που θα χρησιμοποιηθεί σε μηνύματα λάθους για την κατηγορία. Μπορούμε να χρησιμοποιήσουμε οποιονδήποτε αριθμό κάτω από 200, αλλά θα πρέπει να είμαστε σίγουροι ότι είναι διαφορετικοί για κάθε ένα από τα αντικείμενα.

Παρακάτω αναπτύσσουμε μία εφαρμογή της tournament Επιλογής. Βασίζεται στην μέθοδο του Τροχού της Τύχης. Πιο συγκεκριμένα η μέθοδος επιλογής, επιλέγει μονάδες ανάλογα με την τιμή καταλληλότητας τους και επιστρέφει μέλι με καλύτερες τιμές.

```
class TournamentSelector : public GARouletteWheelSelector {
public:
    GADefineIdentity("TournamentSelector", 255);
    TournamentSelector(int w=GASelectionScheme::FITNESS) : GARouletteWheelSelector(w) {}
    TournamentSelector(const TournamentSelector& orig) { copy(orig); }
    TournamentSelector& operator=(const GASelectionScheme& orig) {
        if(&orig != this) copy(orig); return *this; }
    virtual ~TournamentSelector() {}
    virtual GASelectionScheme* clone() const { return new TournamentSelector; }
    virtual GAGenome& select() const; };
GAGenome & TournamentSelector::select() const {
    int picked=0;          float cutoff;      int i, upper, lower;
    cutoff = GARandomFloat();
    lower = 0; upper = pop->size()-1;
    while(upper >= lower) {
        i = lower + (upper-lower)/2;
        if(psum[i] > cutoff)
            upper = i-1;
        else lower = i+1; }
    lower = Min(pop->size()-1, lower);
    lower = Max(0, lower);
    picked = lower;
    cutoff = GARandomFloat();
    lower = 0; upper = pop->size()-1;
    while(upper >= lower) {
        i = lower + (upper-lower)/2;
        if(psum[i] > cutoff) upper = i-1;
        else lower = i+1; }
    lower = Min(pop->size()-1, lower); lower = Max(0, lower);
    GAPopulation::SortBasis basis = (which == FITNESS ? GAPopulation::SCALED : GAPopulation::RAW);
    if(pop->order() == GAPopulation::LOW_IS_BEST){
        if(pop->individual(lower,basis).score() < pop->individual(picked,basis).score())
            picked = lower;}
    else { if(pop->individual(lower,basis).score() >
        pop->individual(picked,basis).score())
            picked = lower; }
    return pop->individual(picked,basis); }
```

#### 1.4.11. Γενετικός Αλγόριθμος.

Παρακάτω βλέπουμε ένα παράδειγμα στο οποίο ένας από τους γονείς επιλέγεται όπως γίνεται συνήθως, η επιλογή όμως του δεύτερου μέλους γίνεται μόνον αν είναι όμοιο με το πρώτο, αν δεν είναι τότε προχωράμε σε επιλογή άλλης μονάδας. Αν γίνουν πολλές επιλογές και αποτύχουν τότε επιλέγεται κάποιο τυχαία.

```
class RestrictedMatingGA : public GASteadyStateGA {
```

```

public:
    GADefineIdentity("RestrictedMatingGA", 288);
    RestrictedMatingGA(const GAGenome& g) : GASteadyStateGA(g) {}
    virtual ~RestrictedMatingGA() {}
    virtual void step();
    RestrictedMatingGA & operator++() { step(); return *this; } };
void RestrictedMatingGA::step() {
    int i, k;
    for(i=0; i<tmpPop->size(); i++){        mom = &(pop->select());
        k=0;
        do {        k++; dad = &(pop->select());
            } while(mom->compare(*dad) < THRESHOLD && k<pop->size());
    stats.numsel += 2;
    if(GAFlipCoin(pCrossover()))
        stats.numcro += (*scross)(*mom, *dad, &tmpPop->individual(i), 0);
    else
        tmpPop->individual(i).copy(*mom);
        stats.nummut += tmpPop->individual(i).mutate(pMutation()); }
    for(i=0; i<tmpPop->size(); i++)
        pop->add(tmpPop->individual(i));
    pop->evaluate(); // get info about current pop for next
    time pop->scale(); // remind the population to do its scaling
    for(i=0; i<tmpPop->size(); i++)        pop->destroy(GAPopulation::WORST, GAPopulation::SCALED);
    stats.update(*pop); // update the statistics by one generation        }

```

#### 1.4.12. Συνάρτηση Τερματισμού.

Η συνάρτηση τερματισμού ορίζει πότε ο αλγόριθμός θα πρέπει να σταματήσει την εξέλιξη του. Αυτή η συνάρτηση δέχεται έναν γενετικό αλγόριθμο σαν μία παράμετρο της και επιστρέφει την τιμή `gaTrue` αν ο αλγόριθμος πρέπει να σταματήσει την εκτέλεση του, ή `gaFalse` αν ο γενετικός αλγόριθμος πρέπει συνεχίσει την εκτέλεση του.

Παρακάτω θα δούμε τρία παραδείγματα τερματικών συναρτήσεων. Η πρώτη συγκρίνει τον τρέχον πληθυσμό με το επιθυμητό αριθμό γενεών. Αν ο τρέχον αριθμός γενεών είναι λιγότερες από τον επιθυμητό τότε η συνάρτηση επιστρέφει `gaFalse`, έτσι η εκτέλεση του αλγορίθμου συνεχίζεται.

```

GABoolean GATerminateUponGeneration(GAGeneticAlgorithm & ga){
    return(ga.generation() < ga.nGenerations() ? gaFalse : gaTrue); }

```

Στο δεύτερο παράδειγμα συγκρίνουμε τον μέσο όρο του τρέχοντος πληθυσμού με την τιμή της καλύτερης μονάδος του τρέχοντος πληθυσμού. Αν η αναλογία αυτών υπερβαίνει κάποια τιμή κατωφλίου τότε η συνάρτηση επιστρέφει `gaTrue` προκειμένου να σταματήσει την εκτέλεση του γενετικού αλγορίθμου. Ουσιαστικά, αυτό σημαίνει ότι ολόκληρος ο πληθυσμός τείνει σε κάποια καλή τιμή σύγκλισης.

```

// σταμάτα την εκτέλεση όταν ο μέσος όρος του πληθυσμού είναι 95% του καταλληλότερου μέλους
const float desiredRatio = 0.95;
Customizing GALib: Termination Function
GABoolean GATerminateUponScoreConvergence(GAGeneticAlgorithm & ga){
    if(ga.statistics().current(GAStatistics::Mean) /
        ga.statistics().current(GAStatistics::Maximum) > desiredRatio)
        return gaTrue;
    else return gaFalse; }

```

Στο τρίτο παράδειγμα χρησιμοποιεί την ποικιλία σε κάποιον πληθυσμό σαν κριτήριο τερματισμού. Αν η ποικιλία πέσει κάτω από κάποια τιμή κατωφλίου τότε θα σταματήσει η εκτέλεση του αλγορίθμου.

```

// σταμάτα την εκτέλεση όταν η ποικιλία πέσει κάτω από αυτήν την τιμή
const float thresh = 0.01;
GABoolean StopWhenNoDiversity(GAGeneticAlgorithm & ga){
    if(ga.statistics().current(GAStatistics::Diversity) < thresh)        return gaTrue;
    else return gaFalse; }

```

Μία γρηγορότερη μέθοδος σύγκλισης είναι όταν ο πληθυσμός παρεκκλίνει από κάποια σταθερή τιμή, αυτή η μέθοδος δεν χρειάζεται σύγκριση των μονάδων του πληθυσμού.

```
// σταμάτα την εκτέλεση όταν η σύγκλιση φτάσει κάτω από αυτήν την τιμή
const float thresh = 0.01;
GABoolean StopWhenNoDeviation(GAGeneticAlgorithm & ga){
    if(ga.statistics().current(GAStatistics::Deviation) < thresh)    return gaTrue;
    else    return gaFalse; }
```

### 1.5. Παραδείγματα με την GALib.

Στην δεύτερη φάση αυτού του μέρους είχαμε θέσει ως στόχο να χρησιμοποιήσουμε την βιβλιοθήκη της GALib για να λύσουμε τρία προβλήματα. Τα προβλήματα και οι λύσεις τους βρίσκονται παρακάτω.

#### 12.5.1. Ελαχιστοποίηση της συνάρτησης $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$ .

Το πρώτο παράδειγμα που θα εκτελέσουμε με την βοήθεια της βιβλιοθήκης GALib είναι να προσπαθήσουμε να ελαχιστοποιήσουμε την τιμή της συνάρτησης :  $x_1^2 + x_2^2 + x_3^2$  . Σε αυτό το παράδειγμα ο αλγόριθμος θα πρέπει να εντοπίσει την τιμή που θα πρέπει να έχουν οι παράμετροι της συνάρτησης μας για να ελαχιστοποιηθεί η τιμή που επιστρέφει. Το ελάχιστο σημείο αυτής της συνάρτησης είναι γνωστό και εντοπίζεται όταν οι συντεταγμένες των παραμέτρων είναι  $f(0, 0, 0) = 0$ .

Για να εκτελέσουμε αυτό το παράδειγμα θα χρησιμοποιήσουμε τον simple γενετικό αλγόριθμο και τους τελεστές αυτής της κατηγορίας. Η αναπαράσταση που θα εφαρμόσουμε για να κωδικοποιήσουμε τα δεδομένα είναι η GABin2DecGenome. Η αναπαράσταση αυτή απαιτεί να ορίσουμε κάποιο φαινότυπο, όπως εξηγήσαμε νωρίτερα, έτσι ορίζουμε ότι χρησιμοποιούμε 16 bit προκειμένου να κωδικοποιήσουμε δυαδικά τις τιμές που θα λαμβάνονται από το διάστημα των δεκαδικών τιμών -10 έως 10. Το κριτήριο τερματισμού που χρησιμοποιούμε είναι ο αριθμός των γενεών που θα εξελιχθούν από τον αλγόριθμο. Έτσι απαντάμε στις τρεις προϋποθέσεις όπως τις κατονομάσαμε στα εισαγωγικά στοιχεία για την GALib προκειμένου να χρησιμοποιήσουμε τους γενετικούς αλγόριθμους. Παρακάτω παραθέτουμε τον κώδικα :

```
#include <stdio.h>                                #include <iostream.h>
#include <fstream.h>                               #include <ga/ga.h>
float objective(GAGenome &);
int main(int argc, char **argv){
    int popsize = 100; // Μέγεθος Πληθυσμού.    float pmut = 0.01; // Πιθανότητα Μετάλλαξης.
    int ngen = 1000; // Αριθμός Γενεών.          float pcross = 0.6; // Πιθανότητα Διασταύρωσης.
    GABin2DecPhenotype map; // ορίζουμε το φαινότυπο όπως απαιτεί ο τύπος δεδομένων.
    map.add(16, -10, 10);    map.add(16, -10, 10);    map.add(16, -10, 10); //
    GABin2DecGenome genome(map, objective); // Δημιουργούμε το πρότυπο γονίδιο χρησιμοποιώντας το
    //φαινότυπο.
    GASimpleGA ga(genome);                        GASigmaTruncationScaling scaling;
    ga.populationSize(popsize);                    ga.nGenerations(ngen);
    ga.pMutation(pmut);                            ga.pCrossover(pcross);
    ga.scaling(scaling);                           ga.scoreFilename("bog.dat");
    ga.scoreFrequency(10);                          ga.flushFrequency(50);
    ga.evolve(seed);
    genome = ga.statistics().bestIndividual();
    cout << "the ga found a minimum at the point (";
    cout << genome.phenotype(0) << ", " << genome.phenotype(1) << ", " << genome.phenotype(2) <<")\n\n";
    cout << "best of generation data are in " << ga.scoreFilename() << "\n";
    return 0;
}
float objective(GAGenome & c) { // Αντικειμενική Διαδικασία προσπαθεί να βρει το ελάχιστο της συνάρτησης.
    GABin2DecGenome & genome = (GABin2DecGenome &);
    float y;
    y = genome.phenotype(0) * genome.phenotype(0);
    y += genome.phenotype(1) * genome.phenotype(1);
```

```

y += genome.phenotype(2) * genome.phenotype(2);
return -y;
}

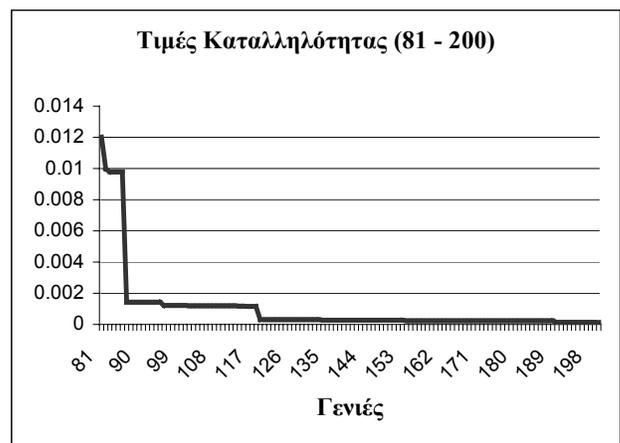
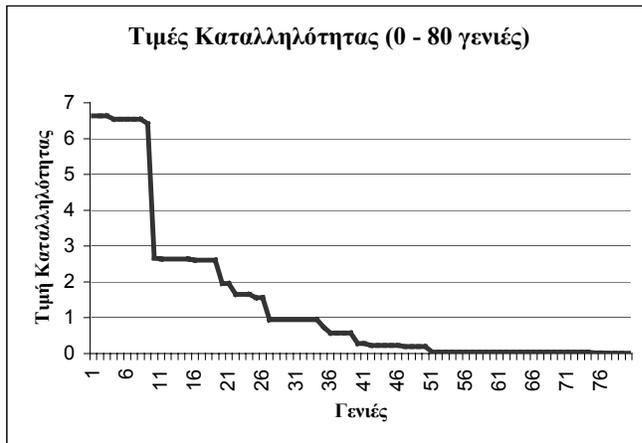
```

Η συνάρτηση που μας δίνει την ελάχιστη τιμή της παραπάνω συνάρτησης είναι η  $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$ . Σε αυτό το παράδειγμα ο υπολογισμός της ελάχιστης τιμής γίνεται από την συνάρτηση Αποτίμησης. Οι παράμετροι που δίνουμε στον αλγόριθμο βρίσκονται στον παρακάτω πίνακα. Στην πρώτη στήλη δηλώνουμε την παράμετρο που θέλουμε να εισάγουμε, ενώ στην δεύτερη στήλη δείχνουμε την εντολή με την οποία το επιτυγχάνουμε :

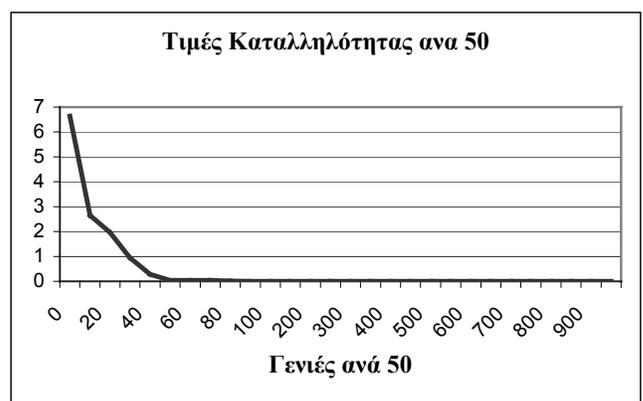
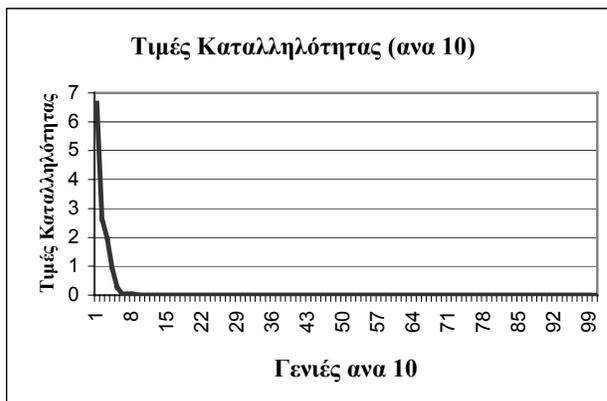
Παράμετροι εκτέλεσης του Γενετικού Αλγορίθμου.	Εντολή
Διασταύρωση ενός σημείου με πιθανότητα $p_c = 0.6$	ga.pCrossover(pcross);
Μετάλλαξη με πιθανότητα $p_m = 0.01$	ga.pMutation(pmut);
Χρωμοσώματα με 16 bits	map.add(16, -10, 10);
Συνάρτηση Καταλληλότητας = $-F(x_1, x_2, x_3)$	
Αριθμός μελών γενιάς = 50	ga.populationSize(popsize);
Αριθμός Γενεών (Επαναλήψεων) = 1000	ga.nGenerations(nngen);

Πίνακας 3<sup>ος</sup> : Σύνολο παραμέτρων για το πρόβλημα ελαχιστοποίησης της  $f(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2$ .

Τα αποτελέσματα που λάβαμε από την εκτέλεση του αλγορίθμου τα απεικονίζουμε σε δύο διαφορετικά διαγράμματα. Μπορούμε εύκολα να παρατηρήσουμε την πτώση που έχει η τιμή καταλληλότητας από τον αρχικό πληθυσμό(Γενιά 0) έως τον τελευταίο πληθυσμό(γενιά 1000). Στα αρχικά διαγράμματα 9<sup>ο</sup> παρατηρούμε για μικρό αριθμό γενεών την μεγάλη πτώση που έχει η τιμή καταλληλότητας.



Σχήμα 9<sup>ο</sup> : Διαγράμματα κλιμάκωσης της τιμής καταλληλότητας για τις γενιές 0 – 200.(Ανά ένα.)



Σχήμα 10<sup>ο</sup> : Διαγράμματα κλιμάκωσης της τιμής καταλληλότητας για τις γενιές 0 – 1000.(Ανά 10 και 50.)

Στα παραπάνω διαγράμματα βλέπουμε την πτώση της τιμής καταλληλότητας για διαφορετικές διαβαθμίσεις των γενεών. Αυτό δηλώνει ότι ο αλγόριθμος σε κάθε νέα γενιά του πληθυσμού εντοπίζει και καλύτερους συνδυασμούς αριθμών για την ελαχιστοποίηση της συνάρτησης. Το σημείο που μας υπέδειξε ο αλγόριθμος ως το ελάχιστο για την συνάρτηση που μελετάμε είναι το σημείο  $(-0.0001525, 0.0001525, 0.0001525)$ . Πρέπει να σημειώσουμε ότι όταν αυξάνει ο αριθμός επαναλήψεων αυξάνει περισσότερο η προσέγγιση της πραγματικά ελάχιστης τιμής της συνάρτησης.

### 1.5.2. Εύρεση της καλύτερης προσέγγισης του $\pi$ .

Το δεύτερο παράδειγμα που θα εκτελέσουμε με την βοήθεια της βιβλιοθήκης είναι να αναπτύξουμε έναν αλγόριθμο για την καλύτερη προσέγγιση του  $\pi$ . Ο σκοπός του αλγορίθμου είναι να εντοπίσει σε κάθε γενιά τον καλύτερο συνδυασμό αριθμών που να τείνουν σε μία ικανοποιητική προσέγγιση όσο το δυνατόν περισσότερων δεκαδικών ψηφίων του  $\pi$ . Η συνάρτηση που είναι επιφορτισμένη με αυτήν την αρμοδιότητα είναι η συνάρτηση Αποτίμησης.

Για να φέρουμε εις πέρας το δεύτερο παράδειγμα θα χρησιμοποιήσουμε τον simple γενετικό αλγόριθμο και τους τελεστές αυτής της κατηγορίας. Η αναπαράσταση που θα χρησιμοποιήσουμε για να κωδικοποιήσουμε τα δεδομένα είναι η GABin2DecGenome. Για την αναπαράσταση των δεδομένων χρησιμοποιούμε 30 bit και το εύρος των δεκαδικών τιμών που θα καλύψουμε είναι από 0 έως 1023. Το κριτήριο τερματισμού που χρησιμοποιούμε είναι ο αριθμός των γενεών που θα εξελιχθούν από τον αλγόριθμο. Παρακάτω παραθέτουμε μέρος του κώδικα :

```
float Objective(GAGenome& g) { // Αντικειμενική Διαδικασία
    GABin2DecGenome & genome = (GABin2DecGenome &)g;
    float z;
    z = abs((genome.phenotype(0) / genome.phenotype(1)) - M_PI);
    return z;
}
```

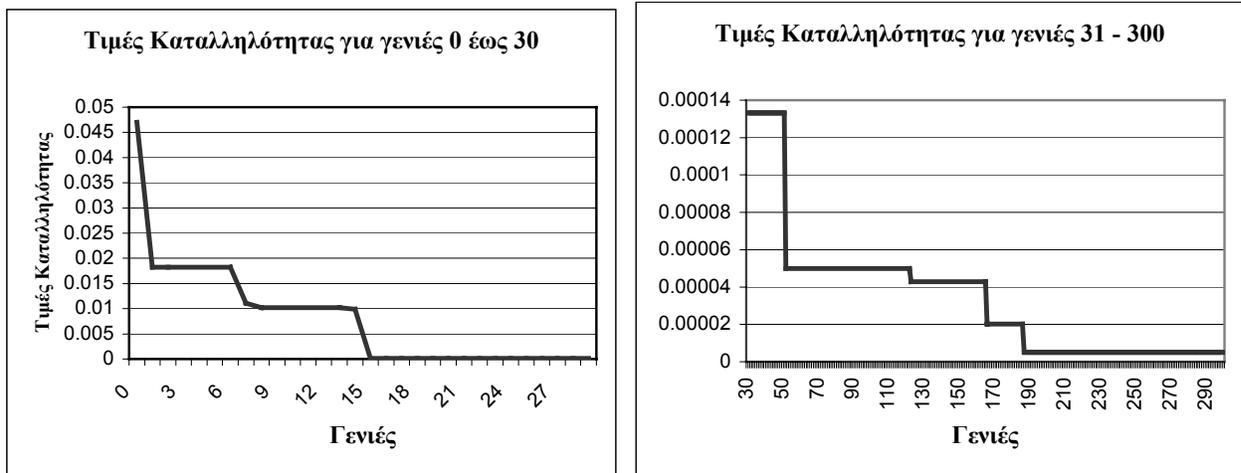
Σε αυτό το παράδειγμα ο υπολογισμός της βέλτιστης τιμής γίνεται από την συνάρτηση Αποτίμησης, την οποία και παραθέτουμε παραπάνω. Από μαθηματική σκοπιά η συνάρτηση που μας δίνει την καλύτερη προσέγγιση του  $\pi$  έχοντας ως παραμέτρους δύο ακεραίους αριθμούς είναι ή  $F(x_1, x_2) = |(x_1 / x_2) - \pi|$ . Ο συνδυασμός των ακεραίων που μας δίνει την βέλτιστη τιμή προσέγγισης είναι 355/113 ( $U < 0.00001$ ) (Μία επίσης καλή τιμή προσέγγισης είναι η 22/7). Οι παράμετροι που δίνουμε στον αλγόριθμο είναι οι :

Παράμετροι εκτέλεσης του Γενετικού Αλγορίθμου.	Εντολή
Διασταύρωση ενός σημείου με πιθανότητα $p_c = 0.6$	ga.pCrossover(0,7);
Μετάλλαξη με πιθανότητα $p_e = 0.01$	ga.pMutation(0,01);
Χρωμοσώματα με 16 bits	map.add(16, 0, 1023);
Συνάρτηση Καταλληλότητας = $-F(x_1, x_2)$	$ (x_1 / x_2) - \pi $
Αριθμός μελών γενιάς = 100	ga.populationSize(100);
Αριθμός Γενεών (Επαναλήψεων) = 1000	ga.nGenerations(1000);

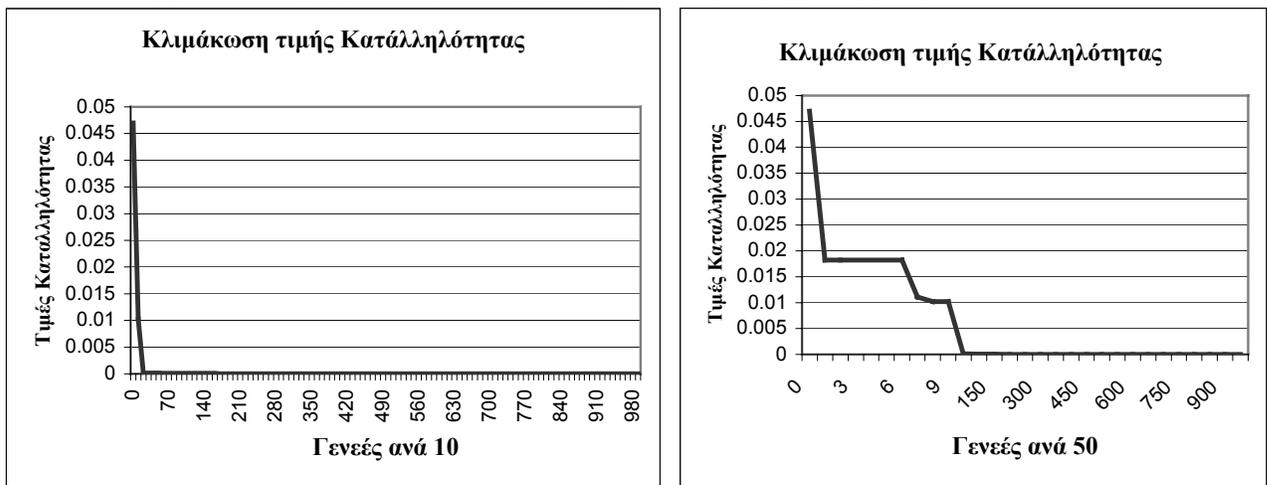
Πίνακας 4<sup>95</sup> : Σύνολο παραμέτρων για το πρόβλημα προσέγγισης του  $\pi$ .

Τα αποτελέσματα που λάβαμε από την εκτέλεση του αλγορίθμου τα απεικονίζουμε σε δύο διαφορετικά διαγράμματα. Μπορούμε εύκολα να παρατηρήσουμε την πτώση που έχει η τιμή καταλληλότητας από τον αρχικό πληθυσμό(Γενιά 0) έως τον τελευταίο πληθυσμό που απεικονίζουμε (γενιά 300). Η πτώση στην τιμή καταλληλότητας που παρατηρείται στα σχεδιαγράμματα δηλώνει ότι καθώς οι γενιές των

πληθυσμών αυξάνουν ο αλγόριθμος εντοπίζει καλύτερες προσεγγίσεις αριθμών για την ακριβέστερη προσέγγιση του  $\pi$ .



Σχήμα 11° : Διαγράμματα κλιμάκωσης της τιμής καταλληλότητας για τις γενιές από 0 – 300.



Σχήμα 12° : Διαγράμματα κλιμάκωσης της τιμής καταλληλότητας για τις γενιές από 0 – 1000.

Ο αλγόριθμος μετά την εκτέλεση του για 1000 γενιές μας εμφανίζει ότι ο καλύτερος συνδυασμός αριθμών για την προσέγγιση του  $\pi$  είναι : **448.65, 142.81**.

### 1.5.3. Το πρόβλημα του πλανόδιου πωλητή.

Στο πρόβλημα του πλανόδιου πωλητή, ο αλγόριθμος θα πρέπει να μας συστήσει μία μοναδική διαδρομή στην οποία κάθε κόμβος θα συμμετέχει το πολύ μία φορά. Στο πρόβλημα αυτό ο πωλητής θέλει να ταξιδέψει από κάποιες πόλεις,  $N$  – κόμβους, και να τις περάσει μόνο μία φορά από κάθε κόμβο.

Αυτό το παράδειγμα μπορεί να εφαρμοστεί το πολύ για 50 πόλεις. Για να εκτελέσουμε αυτό το παράδειγμα θα χρησιμοποιούμε τον SteadyState γενετικό αλγόριθμο και ορισμένους από τους τελεστές αυτής της κατηγορίας. Η αναπαράσταση που θα χρησιμοποιήσουμε για να κωδικοποιήσουμε τα δεδομένα είναι η GAListGenome, σταθερού μήκους. Αυτός ο τύπος δεδομένων απαιτεί να ορίσουμε την συνάρτηση Αρχικοποίησης, όπως εξηγήσαμε νωρίτερα. Η συνάρτηση Αρχικοποίησης είναι :

```
void Initializer(GAGenome& g) {
    GAListGenome<int> &child=(GAListGenome<int> &)g;
    while(child.head()) child.destroy(); // Καταστρέφει όλες τις προϋπάρχουσες λίστες
    int i,town;          static int visit[MAX_TOWNS];
}
```

```

memset(visit, 0, MAX_TOWNS*sizeof(int));
town=GARandomInt(0,ntowns-1);
visit[town]=1;
child.insert(town,GAListBASE::HEAD); // Αρχικός Κόμβος.
for( i=1; i<ntowns; i++) { do { town=GARandomInt(0,ntowns-1);
                                } while (visit[town]);
                            visit[town]=1; child.insert(town);
}} // Κόμβοι που προστίθενται.

```

Το κριτήριο τερματισμού που χρησιμοποιούμε είναι ο αριθμός των γενεών που θα εξελιχθούν από τον αλγόριθμο.

Είχαμε αναφερθεί στην αντικειμενική συνάρτηση που μπορεί να εφαρμόσουμε σε αυτό το πρόβλημα, σαν παραδείγματα είχαμε φέρει την ταχύτητα κ.α. Η καταλληλότητα μίας λύσεις(συνολικής διαδρομής) σε αυτό το παράδειγμα, κρίνεται από την συνολική απόσταση που χαρακτηρίζει σε μία λίστα. Η αντικειμενική διαδικασία είναι επιφορτισμένη να βρίσκει την απόσταση και έχει την παρακάτω μορφή :

```

float Objective(GAGenome& g) { GAListGenome<int> & genome = (GAListGenome<int> &)g;
float dist = 0;
if(genome.head()) {
for(int i=0; i<ntowns; i++) {
int xx = *genome.current();
int yy = *genome.next();
dist += DISTANCE[xx][yy]; }
} return dist; }

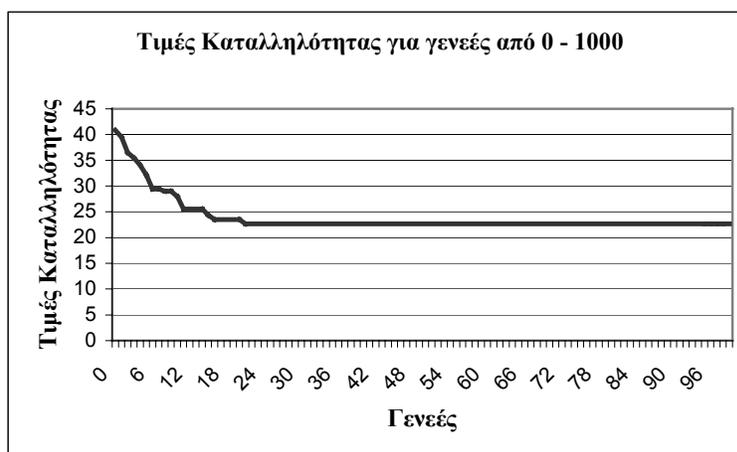
```

Τα δεδομένα που εισάγουμε στον αλγόριθμο :

Παράμετροι εκτέλεσης του Γενετικού Αλγορίθμου.	Εντολή
Διασταύρωση ενός σημείου με πιθανότητα $p_c = 0.6$	ga.pCrossover(1,0);
Μετάλλαξη με πιθανότητα $p_m = 0.01$	ga.pMutation(0,1);
Αριθμός μελών γενιάς = 100	ga.populationSize(100);
Αριθμός Γενεών (Επαναλήψεων) = 1000	ga.nGenerations(1000);

Πίνακας 5<sup>ος</sup> : Σύνολο παραμέτρων για το πρόβλημα του πλανόδιου πωλητή.

Τα αποτελέσματα που δεχόμαστε από τον αλγόριθμο σε διάγραμμα φαίνεται παρακάτω. Μπορούμε να παρατηρήσουμε ότι από ένα σημείο και έπειτα ο αλγόριθμος καταλήγει στην βέλτιστη διαδρομή.



Σχήμα 13<sup>ο</sup> : Διαγράμματα κλιμάκωσης της τιμής καταλληλότητας για τις γενιές από 0 – 1000.

Το αποτέλεσμα που μας επιστρέφει ο αλγόριθμος ως την βέλτιστη διαδρομή έχει τιμή καταλληλότητας ίση με : **22.6503** και η διαδρομή είναι η : 18 19 15 11 7 3 2 6 5 1 20 0 4 8 12 16 17 13 9 10 14. Σε αυτό το αποτέλεσμα ο αλγόριθμος κατέληξε από την 25 γενιά.

### ΜΕΡΟΣ ΙΙΙ

Στο τρίτο μέρος της παρούσης εργασίας θα αναφέρουμε κάποια επιπλέον στοιχεία σχετικά με την ευρύτερη κατηγορία των Εξελικτικών Αλγορίθμων. Αυτός ο όρος χρησιμοποιείται για να περιγράψουμε εκείνη την κατηγορία συστημάτων επίλυσης προβλημάτων, που βασίζονται σε ηλεκτρονικούς υπολογιστές και χρησιμοποιούν σαν βασικό στοιχείο στον σχεδιασμό τους υπολογιστικά μοντέλα εξελικτικών διαδικασιών.

Έχουν προταθεί ποικίλα μοντέλα εξελικτικών αλγορίθμων. Οι βασικές κατηγορίες είναι οι παρακάτω :

- |                            |                                 |
|----------------------------|---------------------------------|
| 1) Γενετικοί Αλγόριθμοι.   | 3) Γενετικός Προγραμματισμός.   |
| 2) Εξελικτικοί Αλγόριθμοι. | 4) Εξελικτικός Προγραμματισμός. |

Οι κατηγορίες αυτές στο σύνολο τους μέσα από τις διαδικασίες που ακολουθούν μοιράζονται την κοινή θεμελιώδη αρχή με την οποία εξομοιώνουν την εξέλιξη των δομών δεδομένων που χρησιμοποιούν για να αναπαραστήσουν τις μονάδες του πληθυσμού που διαχειρίζονται, διαμέσω των διαδικασιών της Επιλογής, της Αναπαραγωγής και της Μετάλλαξης. Οι διαδικασίες αυτές εκτελούνται σύμφωνα με την λαμβανόμενη επίδοση τους στο περιβάλλον.

Μία πιο ακριβής περιγραφή είναι : Οι Εξελικτικοί Αλγόριθμοι διατηρούν έναν πληθυσμό από δομές δεδομένων τις οποίες εξελίσσουν σύμφωνα με τους κανόνες της Επιλογής και άλλων τελεστών οι οποίοι αναφέρονται ως 'Τελεστές Έρευνας' και είναι οι τελεστές της Διασταύρωσης και της Μετάλλαξης. Κάθε μονάδα του πληθυσμού λαμβάνει ένα μέτρο που εκφράζει την καταλληλότητα του στο περιβάλλον του. Οι τελεστές της Διασταύρωσης και της Μετάλλαξης έχουν σαν κύριο σκοπό να παράγουν καλύτερα μέλη, εκμεταλλευόμενοι τις υπάρχουσες πληροφορίες.

Χρησιμοποιώντας τους αλγορίθμους αυτούς μπορούμε να φέρουμε εις πέρας πολύ δύσκολες και χρονοβόρες έρευνες και να δώσουμε λύσεις σε μία μεγάλη ποικιλία προβλημάτων, αναπτύσσοντας πολύ εύελικτους μηχανισμούς αναζήτησης και έρευνας.

Τους αλγορίθμους που αναφέρουμε σε αυτό το μέρος θα τους προσεγγίσουμε μένοντας στις κυριότερες λειτουργίες και διαφορές που έχουν με τους γενετικούς αλγορίθμους χωρίς να επιμείνουμε σε όλες τις τεχνικές που ακολουθούν.

## 1. Γενετικός Προγραμματισμός

Ο Γενετικός Προγραμματισμός είναι ένας ανεξάρτητος – τομέας επίλυσης προβλημάτων. Ουσιαστικά μπορεί να θεωρηθεί ότι είναι μία προέκταση των Γενετικών Αλγορίθμων. Οι βασικές αρχές στις οποίες βασίζεται ο γενετικός προγραμματισμός είναι αυτές της αναπαραγωγής και της επιβίωσης, που διατύπωσε ο Darwinian του καταλληλότερου μέλους. Αυτές βρίσκονται σε αναλογία της φυσιολογικής γενετικής διαδικασίας με τους τελεστές της διασταύρωσης και της μετάλλαξης.

Πιο γενικά, ο γενετικός προγραμματισμός συνθέτει την έκφραση υψηλού επιπέδου συμβολικών αναπαραστάσεων των προγραμμάτων H/Y με την βέλτιστη απόδοση της εκμάθησης του μοντέλου των γενετικών αλγορίθμων του Holland.

Οι δομές δεδομένων που αποτελούν τον πληθυσμό λύσεων, δεν είναι συμβολοσειρές χαρακτήρων σταθερού μεγέθους που κωδικοποιούν μία πιθανή λύση σε ένα πρόβλημα, όπως είχαμε συναντήσει στους γενετικούς αλγορίθμους. Σε αυτούς τους αλγορίθμους χρησιμοποιούμε τους γενετικούς αλγόριθμους για να διαχειρίζονται τις δομές δεδομένων του πληθυσμού τους. Ένας πληθυσμός αποτελείται από προγράμματα διαφορετικού μεγέθους τα οποία όταν εκτελούνται, μας επιστρέφουν τις πιθανές λύσεις σε ένα πρόβλημα. Αυτά τα προγράμματα υπολογιστή αναπτύσσονται για να λύσουν, ή να προσεγγίσουν λύσεις διαφόρων προβλημάτων.

Ο βασικός αλγόριθμος του Γενετικού Προγραμματισμού είναι ένας κλασικός γενετικός αλγόριθμος που χρησιμοποιεί εξειδικευμένους τελεστές Μετάλλαξης και Διασταύρωσης στις σύνθετες και πολύπλοκες δομές δεδομένων του πληθυσμού τους.

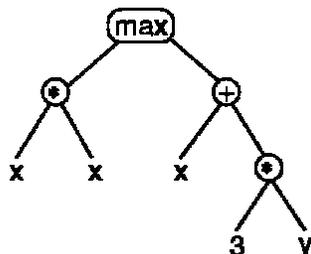
Ο Γενετικός Προγραμματισμός ανατρέπει προγράμματα υπολογιστών με σκοπό να επιλύσει προβλήματα ακολουθώντας τα παρακάτω τρία βήματα :

1. Δημιουργεί έναν αρχικό πληθυσμό από τυχαίες συνθέσεις διαδικασιών και τερματικών του προβλήματος μας.
2. Επαναληπτικά εκτελεί τα παρακάτω υπό-βήματα έως ότου ικανοποιηθούν τα κριτήρια τερματισμού:
  - 2.1. Εκτέλεση κάθε προγράμματος του πληθυσμού και εκχώρηση σε αυτό μίας τιμής καταλληλότητας χρησιμοποιώντας ένα ενιαίο μέτρο καταλληλότητας.
  - 2.2. Δημιουργία ενός νέου πληθυσμού από προγράμματα εφαρμόζοντας τους ακόλουθους τελεστές. Οι τελεστές εφαρμόζονται στα προγράμματα που έχουν επιλεγεί από τον πληθυσμό με κάποια πιθανότητα ανάλογη της τιμής καταλληλότητας τους.
    - 2.2.1. Αναπαραγωγή Darwinian : Αναπαραγωγή ενός προγράμματος αντιγράφοντας' το σε έναν νέο πληθυσμό.
    - 2.2.2. Διασταύρωση : Δημιουργία δύο νέων προγραμμάτων από δύο ήδη υπάρχοντα προγράμματα. Τα νέα προγράμματα θα προκύψουν συνδυάζοντας γενεσιολογικά τυχαία τμήματα των επιλεγμένων προγραμμάτων. Η επιλογή των σημείων αποκοπής γίνεται τυχαία όπως περιγράψαμε στους τελεστές διασταύρωσης των γενετικών αλγορίθμων. Έτσι επιλέγουμε τμήματα των προγραμμάτων που θα διασταυρωθούν.
    - 2.2.3. Μετάλλαξη : Δημιουργία ενός νέου προγράμματος από κάποιο ήδη υπάρχον. Το νέο πρόγραμμα προκύπτει μεταλλάσσοντας κάποιο τυχαίο τμήμα του αρχικού προγράμματος.

3. Το πρόγραμμα που υποδεικνύεται από την μέθοδο του επιστρεφόμενου αποτελέσματος του. Το αποτέλεσμα μπορεί να είναι η λύση ή κάποια προσέγγιση στο πρόβλημα μας.

### 1.1. Η Μορφή των προγραμμάτων του Γενετικού Προγραμματισμού.

Τα προγράμματα στους γενετικούς αλγορίθμους συνήθως παρίστανται ως τεχνολογικά δέντρα, παρά σαν γραμμές κώδικα. Για παράδειγμα, μία απλή μαθηματική έκφραση έστω  $\max(x * x, x + 3 * y)$ , μπορούμε να την απεικονίσουμε ως :



Σχήμα 14° : Δεντροειδής απεικόνιση ενός προγράμματος στον Γενετικό Προγραμματισμό.

Στο παραπάνω σχήμα απεικονίζουμε μία έκφραση σε μορφή δέντρου. Η παραπάνω μαθηματική έκφραση είναι κάποιο πρόγραμμα που έχει σαν σκοπό να επιλύσει κάποιο πρόβλημα. Παρατηρούμε ότι στα άκρα του δέντρου υπάρχουν οι μεταβλητές ή οι σταθερές του προγράμματος μας. Το σύνολο των στοιχείων που βρίσκονται στα άκρα του δέντρου ονομάζονται τερματισμοί. Οι τερματισμοί τοποθετούνται κάτω από κάποιον κόμβο ή μαθηματικό τελεστή. Οι μαθηματικοί τελεστές του σχήματος ονομάζονται συναρτήσεις.

Πιο γενικά στην δεντροειδή μορφή που μπορούμε να δώσουμε για να αναπαραστήσουμε κάποιο πρόγραμμα, πρέπει να σημειώσουμε τα εξής : Το σύνολο των εσωτερικών κόμβων που χρησιμοποιούνται στα τεχνολογικά δέντρα (σαν αναπαράσταση των Γενετικών Αλγορίθμων), ονομάζεται σύνολα συναρτήσεων (*function set*), και τα ορίζουμε ως  $F = \{f_1, \dots, f_N\}$ , όπως στο προηγούμενο παράδειγμα το σύνολο συναρτήσεων  $F$  απαρτίζεται από  $F = \{\max, *, +\}$ . Όλες οι συναρτήσεις  $F$  έχουν περιττό πλήθος ορισμάτων, δηλαδή  $z(f_i) \geq 1$ .

Το σύνολο των τερματικών κόμβων στα τεχνολογικά δέντρα, αντιπροσωπεύουν προγράμματα στον γενετικό προγραμματισμό. Το σύνολο αυτών των προγραμμάτων ονομάζουμε τερματικά σύνολα και τα συμβολίζονται ως  $T = \{t_1, \dots, t_N\}$ , το σύνολο των τερματικών κόμβων  $T$  που χρησιμοποιήσαμε στο προηγούμενο παράδειγμα είναι  $T = \{x, y, 3\}$ . Τα σύνολα των συναρτήσεων  $F$  και  $T$  μπορούν να ενωθούν σε ένα ενιαίο σύνολο  $C$ , που συμβολίζουμε ως  $C = F \cup T$ . Η ένωση των συνόλων μας δίνει το σύνολο :

$$C = F \cup T = \{\max, *, +\} \cup \{x, y, 3\} = \{\max, *, +, x, y, 3\}$$

Το πεδίο αναζήτησης στον γενετικό προγραμματισμό είναι το σύνολο όλων των πιθανών (αναδρομικών) συνθέσεων των συναρτήσεων που περιέχονται στο σύνολο  $C$ .

### 1.2. Σύνολα Συναρτήσεων & Τερματικά Σύνολα στον Γενετικό Προγραμματισμό.

Μπορούμε να συνοψίσουμε ορισμένες από τις συναρτήσεις που μπορούν να συμπεριληφθούν στο σύνολο  $F$  στον παρακάτω πίνακα :

Είδος	Παραδείγματα
Μαθηματικές	sin, cos, exp
Αριθμητικές	+, *, /

Boolean	AND, OR, NOT
Ελέγχου Συνθήκης	IF-THEN-ELSE
Επαναληπτικές	FOR, REPEAT

Πίνακας 3<sup>ος</sup> : Παραδείγματα συναρτήσεων του συνόλου  $F$ .

Τα Τερματικά σύνολα που να συμπεριληφθούν στο σύνολο  $T$  είναι οι παρακάτω:

Είδος	Παραδείγματα
Μεταβλητές	x, y
Σταθερές Τιμές	3, 0.45
0-arity functions	rand, go_left

Πίνακας 4<sup>ος</sup> : Παραδείγματα τερματικών του συνόλου  $T$ .

Προκειμένου ο Γενετικός Προγραμματισμός να λειτουργήσει κανονικά, τα σύνολα  $F$  και  $T$  πρέπει να ικανοποιούν τις ακόλουθες δύο προϋποθέσεις:

1. Το κλείσιμο απαιτεί ότι κάθε συνάρτηση στο  $F$  είναι ικανή να δεχθεί ως όρισμά της οποιαδήποτε τιμή και τύπο δεδομένων που μπορεί να επιστραφεί από οποιαδήποτε συνάρτηση του συνόλου  $C = F \cup T$ . Αυτό εμποδίζει τα σφάλματα κατά την διάρκεια εκτέλεσής τους.
2. Η αποτελεσματικότητα απαιτεί ότι οι συναρτήσεις του συνόλου  $C$  είναι ικανές να εκφράσουν τις λύσεις του προβλήματος, για παράδειγμα ότι τουλάχιστον μία λύση ανήκει στο σύνολο όλων των πιθανών (αναδρομικών) συνθέσεων των συναρτήσεων του συνόλου  $C$ .

### 1.3. Προβλήματα που πρέπει να αντιμετωπίσουμε.

Για να είναι δυνατή η εφαρμογή του γενετικού προγραμματισμού σε κάποιο πρόβλημα, οι χρήστες πρέπει να επιλύσουν πέντε βασικά πρωταρχικά βήματα. Πρέπει να ορίσουν :

1. ένα σύνολο τερματισμών.
2. ένα σύνολο από πρωταρχικές διαδικασίες.
3. ένα μέτρο καταλληλότητας.
4. τις παραμέτρους για τον έλεγχο της εκτέλεσής τους.
5. μία μέθοδο για την αναγνώριση κάποιου αποτελέσματος και κάποιο κριτήριο για τον τερματισμό της εκτέλεσής τους.

Το πρώτο βασικό βήμα που απαιτείται να κάνουμε για την προετοιμασία χρήσης του γενετικού προγραμματισμού είναι να ορίσουμε ένα σύνολο τερματισμών. Αυτούς μπορούμε να τους φανταστούμε ως τις εισόδους για κάποιο αναζητούμενο πρόγραμμα. Αυτό το σύνολο των τερματισμών είναι τα συστατικά που βασίζεται ο γενετικός προγραμματισμός για να κατασκευαστεί ένα πρόγραμμα που θα λύσει ή θα προσεγγίσει την λύση ενός προβλήματος.

Το δεύτερο βασικό κριτήριο για την προετοιμασία χρήσης τους είναι να αναγνωρίσουμε ένα σύνολο συναρτήσεων οι οποίες μπορούν να χρησιμοποιηθούν έτσι ώστε να είναι δυνατόν να δημιουργηθεί η μαθηματική έκφραση που να ταιριάζει στο τελικό δείγμα δεδομένων. Κάθε πρόγραμμα είναι η σύνθεση των διαδικασιών από το σύνολο συναρτήσεων  $F$  και τα τερματικά από το σύνολο των τερματικών  $T$ . Επίσης, κάθε διαδικασία σε ένα σύνολο διαδικασιών πρέπει να είναι ικανή να δέχεται σαν ορίσματα της κάθε τιμή και τύπο δεδομένων που είναι δυνατόν να επιστραφεί ως αποτέλεσμα από μία διαδικασία στο σύνολο των διαδικασιών και οποιαδήποτε τιμή ή τύπο δεδομένων που είναι δυνατόν να εκχωρηθούν σε

ένα τερματικό σε ένα σύνολο τερματικών. Αυτό σημαίνει ότι το σύνολο των συναρτήσεων και των τερματικών που θα επιλέξαμε πρέπει να έχουν στενές ιδιότητες έτσι ώστε να υπάρχει η δυνατότητα σύνθεσης των συναρτήσεων και των τερματικών που να παράγουν επιτρεπτά εκτελέσιμα προγράμματα.

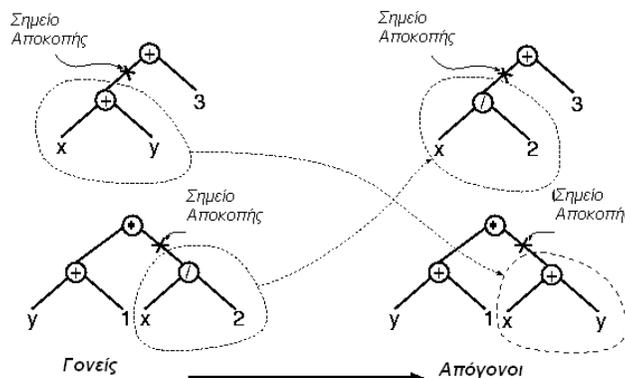
Η εξελικτική διαδικασία οδηγείται από την τιμή του μέτρου καταλληλότητας. Κάθε πρόγραμμα μέλος του πληθυσμού εκτελείται και αποτιμάται, χρησιμοποιώντας το μέτρο καταλληλότητας. Με την βοήθεια του μέτρου καταλληλότητας προσδιορίζουμε το πόσο καλά εκτελεί μία διαδικασία ένα συγκεκριμένο πρόγραμμα στο περιβάλλον που δραστηριοποιείται. Η καταλληλότητα διαφέρει ανάλογα με το πρόβλημα που έχουμε να αντιμετωπίσουμε. Σε αρκετά προβλήματα η καταλληλότητα μετράται από την διαφορά μεταξύ του επιστρεφόμενου αποτελέσματος της εκτέλεσης ενός υποψήφιου προγράμματος και της τιμής του επιθυμητού αποτελέσματος. Σε αυτές τις περιπτώσεις όσο η τιμή τείνει στο 0 τόσο καλύτερο θεωρείται ένα πρόγραμμα. Σε ένα πρόβλημα ελέγχου βελτίωσης, η τιμή καταλληλότητας ενός προγράμματος είναι το ποσοστό της ώρας που απαιτείται για να φτάσει το σύστημα στην επιθυμητή κατάσταση. Σε τέτοιες περιπτώσεις όσο μικρότερο είναι αυτό το ποσοστό τόσο καλύτερο είναι ένα πρόγραμμα.

Οι κυριότεροι παράμετροι για τον έλεγχο της εκτέλεσης του γενετικού προγραμματισμού είναι το μέγεθος του πληθυσμού,  $M$ , και ο μέγιστος αριθμός γενεών που θα εκτελεστεί ο αλγόριθμος,  $G$ .

Η εκτέλεση ενός γενετικού προγραμματισμού απαιτεί τον προσδιορισμό των κριτηρίων τερματισμού για τον τερματισμό της εκτέλεσης του και μία μέθοδο αναγνώρισης του αποτελέσματος. Η συνηθέστερη μέθοδος που χρησιμοποιείται για την αναγνώριση του επιθυμητού αποτελέσματος είναι να κατονομάσουμε την καλύτερη λύση που αποκτάται σε κάθε γενιά του πληθυσμού κατά την διάρκεια της εκτέλεσης τους.

#### 1.4. Διασταύρωση στον Γενετικό Προγραμματισμό.

Ο τελεστής της Διασταύρωσης στον Γενετικό Προγραμματισμό δημιουργεί καινούργια προγράμματα απογόνους από δύο μητρικά προγράμματα. Η επιλογή των προγονικών προγραμμάτων γίνεται βάση της τιμής καταλληλότητας που έχουν στο πληθυσμό. Συνήθως τα μητρικά προγράμματα είναι διαφορετικού μεγέθους και σχήματος. Τα προγράμματα απόγονοι συντίθενται λαμβάνοντας από τα μητρικά προγράμματα υπό – δέντρα, υπό – προγράμματα, ή υπό – ρουτίνες.



Σχήμα 15<sup>ο</sup> : Διασταύρωση στον Γενετικό Προγραμματισμό.

Το παραπάνω σχήμα απεικονίζει τον τελεστή Διασταύρωσης στον Γενετικό προγραμματισμό. Η Διασταύρωση λειτουργεί ως εξής : αποκόπτει και αντιγράφει τμήματα προγραμμάτων. Στην διαδικασία

της Διασταύρωσης συμμετέχουν δύο επιλεγμένα μέλη του πληθυσμού προγραμμάτων. Το σημείο αποκοπής, όμοια με τους γενετικούς αλγορίθμους, επιλέγεται τυχαία. Τα δύο γονίδια, είχαν τα προγράμματα που υπολόγιζαν τις παραστάσεις :

$$\text{Γονίδιο 1}^\circ : (x + y) + 3.$$

$$\text{Γονίδιο 2}^\circ : (y + 1) * (x / 2).$$

Μετά από την εφαρμογή του τελεστή της Διασταύρωσης τα προγράμματα που θα προκύψουν θα υπολογίζουν τις παραστάσεις :

$$\text{Γονίδιο 1}^\circ : (x / 2) + 3.$$

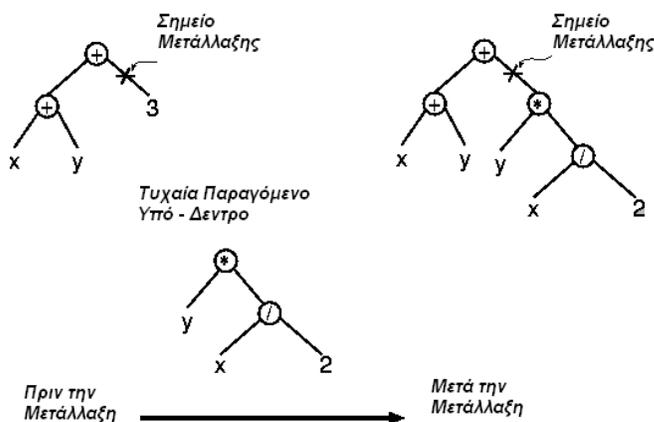
$$\text{Γονίδιο 2}^\circ : (y + 1) * (x + y).$$

Μία διαφορά που υπάρχει με τον τελεστή Διασταύρωσης που χρησιμοποιούμε στους γενετικούς αλγορίθμους είναι ότι εδώ αποκόπτουμε και ανταλλάσσουμε τμήματα κώδικα μεταξύ των συμμετεχόντων μελών. Η γενικότερη φιλοσοφία που ακολουθείτε από τον τελεστή παραμένει η ίδια.

### 1.5. Μετάλλαξη στον Γενετικό Προγραμματισμό.

Ο τελεστής της Μετάλλαξης στον Γενετικό Προγραμματισμό δημιουργεί νέα προγράμματα από κάποιο μητρικό πρόγραμμα. Η επιλογή του πατρικού προγράμματος βασίζεται στην τιμή καταλληλότητας του προγράμματος στον πληθυσμό.

Ο τελεστής της Μετάλλαξης λειτουργεί ως εξής : επιλέγει τυχαία κάποιο σημείο αποκοπής του προγράμματος και μετά έχουμε την δημιουργία ενός τυχαίου μέρους προγράμματος το οποίο επισυνάπτεται έπειτα του αποκομμένου σημείου.



Σχήμα 16<sup>ο</sup> : Μετάλλαξη στον Γενετικό Προγραμματισμό.

Στο παραπάνω σχήμα απεικονίζουμε την εφαρμογή του τελεστή της Μετάλλαξης. Μπορούμε να διακρίνουμε ότι το αρχικό πρόγραμμα έχει το σύνολο τερματικών  $T = \{x, y, 3\}$ . Το σύνολο των συναρτήσεων είναι  $F = \{+, *\}$ . Η ένωση των δύο συνόλων μας δίνει το σύνολο  $C = \{x, y, 3, +, *\}$ . Η τυχαία παραγόμενη παράσταση που προστίθεται έπειτα του σημείου αποκοπής είναι η  $y*(x / 2)$ . Το παραγόμενο πρόγραμμα έχει την μορφή που φαίνεται στο δέντρο μετά την μετάλλαξη.

$$\text{Αρχική κατάσταση Γονιδίου} : (x + y) + 3.$$

$$\text{Τελική κατάσταση Γονιδίου} : (x + y) + (y * (x / 2)).[9][10]$$

## **2. Εξελικτικός Προγραμματισμός.**

Ο εξελικτικός προγραμματισμός είναι μία στοχαστική στρατηγική βελτιστοποίησης παρόμοια με τους γενετικούς αλγόριθμους. Αντίθετα όμως με αυτούς δίνει περισσότερη έμφαση στην συμπεριφορά διασύνδεσης μεταξύ των γονέων με τους απογόνους τους, παρά στην έρευνα εξειδικευμένων γενετικών τελεστών όπως παρατηρείται στην φύση. Ο Εξελικτικός Προγραμματισμός είναι παρόμοιος με τις Εξελικτικές Στρατηγικές.

Στον Εξελικτικό Προγραμματισμό όπως και στους γενετικούς αλγόριθμους υπάρχει η βασική αποδοχή ότι η καταλληλότητα μίας λύσης μπορεί να χαρακτηριστεί σε σχέση με το σύνολο των μεταβλητών της, καθώς επίσης ότι υπάρχει μία βέλτιστη λύση ή ένα σύνολο τέτοιων σε σχέση με το σύνολο των μεταβλητών μίας λύσης. Για παράδειγμα στο 'πρόβλημα του Πλανόδιου Πωλητή' κάθε μία λύση είναι ένα μονοπάτι που πρέπει να ακολουθήσει ο πωλητής. Το μήκος κάθε μονοπατιού θα μπορούσε να είναι η τιμή καταλληλότητας της λύσης. Πολύ επιπόλαια μπορούμε να πούμε ότι η καταλληλότητα κάθε λύσης είναι ανάλογη της απόστασης, θα μπορούσαμε όμως να έχουμε άλλα κριτήρια αξιολόγησης μίας λύσης όπως για παράδειγμα την μικρότερη απόσταση ή την γρηγορότερη λύση.

Η βασική μέθοδος του, αναπτύσσεται με τρία βήματα. Αυτά τα βήματα εκτελούνται επαναληπτικά έως ότου υπερβούν κάποιο όριο κατοφλίου επαναληπτικής εκτέλεσης ή βρεθεί κάποια ικανοποιητική λύση από τον αλγόριθμο.

1. Επιλέγουμε έναν αρχικό πληθυσμό από ενδεχόμενες λύσεις, τυχαία. Ο αριθμός των λύσεων στον πληθυσμό είναι ανάλογος με την ταχύτητα της βελτιστοποίησης. Καμία οριστική απάντηση δεν είναι διαθέσιμη, όπως για παράδειγμα πόσες λύσεις είναι κατάλληλες και ποιες από αυτές είναι αχρείαστες για την επίλυση του προβλήματος.

2. Κάθε λύση αντιγράφεται σε έναν καινούργιο πληθυσμό. Κάθε ένας από τους απογόνους ζευγαρώνεται σύμφωνα με μία κατανομή από τύπους Μετάλλαξης, ξεκινώντας κλιμακωτά από τους κατώτερης έως του πιο οριακούς τύπους μετάλλαξης. Η δριμύτητα της Μετάλλαξης κρίνεται με βάση των λειτουργικών αλλαγών που απαιτούνται στους γονείς.

3. Για κάθε έναν απόγονο υπολογίζεται η τιμή καταλληλότητας του. Τυπικά, μια στοχαστική tournament διατηρείται προκειμένου να προσδιορίσουμε N λύσεις για τον πληθυσμό των λύσεων.

### **2.1. Οι βασικές διαφορές μεταξύ Γενετικών Αλγορίθμων και Εξελικτικού Προγραμματισμού.**

Είναι ότι στον εξελικτικό προγραμματισμό δεν είμαστε υποχρεωμένοι να ορίσουμε μία αναπαράσταση για τις λύσεις του προβλήματος, η αναπαράσταση μιμείται το πρόβλημα. Μία δεύτερη διαφορά είναι ότι ο τελεστής της Μετάλλαξης αλλάζει όψη στην λύση σύμφωνα με στατιστικά. Όσο το βέλτιστο αποτέλεσμα προσεγγίζεται τόσο η αυστηρότητα του τελεστή μειώνεται. Έχουν αναπτυχθεί ποικίλες τεχνικές, για να αντιμετωπίσουμε την περίπτωση που δεν γνωρίζουμε το γενικό βέλτιστο. Η πιο γνωστή είναι η Μετα – Εξελικτικές τεχνικές, με τις οποίες η μεταβλητή της Μετάλλαξης εξελίσσεται μαζί με τον αλγόριθμο.[11],[9]

## **3. Εξελικτικές Στρατηγικές.**

Οι Εξελικτικές Στρατηγικές αρχικά αναπτύχθηκαν προκειμένου να δώσουν λύσεις σε Τεχνικά Προβλήματα Βελτιστοποίησης(TOPs), όπως για παράδειγμα στην κατασκευή του βέλτιστου επιταχυντήρα ανάφλεξης καυσίμων σε κινητήρα.

Αργότερα ο Schewfer γενίκευσε αυτές τις στρατηγικές σε πολλών μελών Εξελικτικές Στρατηγικές, δηλώνονται από  $(m+1)$  και  $(m,l)$  αυτά απομεινώνται τις ακόλουθες βασικές αρχές τις οργανικής εξέλιξης : ένας πληθυσμός, στον οποίο δίνεται μία πιθανότητα για ανασυνδυασμό γονιδίων με κάποιο τυχαίο δείκτη ζευγάρισμα, Μετάλλαξη και Επιλογή. Αυτές τις στρατηγικές τις αποκαλούμε PLUS STRATEGIES και COMMA STRATEGIES. Στην περίπτωση των PLUS στρατηγικών ο μητρικός πληθυσμός συμμετέχει στην διαδικασία Επιλογής, ενώ στην περίπτωση της COMMA στρατηγικής μόνο οι απόγονοι υποβάλλονται στην διαδικασία της Επιλογής, ενώ ο μητρικός πληθυσμός πεθαίνει. Τα γράμματα  $m$  και  $l$  αντίστοιχα δηλώνουν το μέγεθος του πληθυσμού και τον αριθμό των απογόνων που θέλουμε να έχουμε σε κάθε μία γενεά.

Στον πληθυσμό των εξελικτικών στρατηγικών μία μονάδα αναπαριστά ένα σημείο στον χώρο έρευνας και αποτελείται από το γονότυπο: *Αντικειμενική Μεταβλητή*, είναι μία μεταβλητή( $X$ ) που πρέπει να προσαρμόζεται με τους τελεστές ανασυνδυασμού γονιδίων όπως η αντικειμενική διαδικασία φτάνει στο βέλτιστο αποτέλεσμα. *Μεταβλητή Στρατηγικής*, που δηλώνει την μεταβλητότητα της μεταβλητής( $X$ ).

Σε μία PLUS STRATEGY, οι  $m$  καλύτερες μονάδες από όλες  $(m + 1)$ , θα επιβιώσουν σε επόμενες γενεές. Με την μεταβλητή comma μόνο οι  $l$  μονάδες του πληθυσμού θα επιλεγούν για να περάσουν από τους τελεστές ανασυνδυασμού. Στο δεύτερο σχήμα, έχουμε πιο ρεαλιστική συμπεριφορά του αλγορίθμου, επειδή καμία μονάδα δεν θα επιβιώσει για πάντα, που θεωρητικά μπορεί να συμβεί με την PLUS μεταβλητή.

### **3.1. Οι βασικές ομοιότητες μεταξύ Εξελικτικών Στρατηγικών και Εξελικτικού Προγραμματισμού.**

Παρόλο που η πρώτη επαφή μεταξύ των ομάδων που εργάζονταν για την ανάπτυξη των δύο μοντέλων έγινε το 1992 πριν από το πρώτο ετήσιο συνέδριο για τον Εξελικτικό προγραμματισμό και την πλήρως ανεξάρτητη ανάπτυξη τους, τα δύο μοντέλα έχουν μοιράζονται πολλές ομοιότητες. Για παράδειγμα όταν υλοποιούνται και έχουν να επιλύσουν προβλήματα πραγματικών μεταβλητών, και οι δύο τυπικά εργάζονται στις τιμές παρά την κωδικοποίηση τους που συναντήσαμε στους γενετικούς αλγορίθμους. Επίσης ο μηχανισμός της Επιλογής εφαρμόζεται για να μας υποδείξει ποιες μονάδες του πληθυσμού θα απομακρυνθούν από τον πληθυσμό. Οι ομοιότητες τους επεκτείνονται και στην χρήση μεθόδων αυτό – προσαρμογής των μονάδων προκειμένου να προσδιορίσουν τις καλύτερες μεταλλάξεις τους, οι μέθοδοι κατά τις οποίες οι γονείς δεν δείχνουν μόνο μία πιθανή λύση στο πρόβλημα, αλλά και πληροφορίες για το πώς να διανεμηθούν οι επόμενες δοκιμές(απόγονοι). Και στα δύο μοντέλα αναπτύσσονται θεωρητικά καταλήξεις σύγκλισης όπως το ασυμπτωτικό και της ταχύτητας.

### **3.2. Οι βασικές διαφορές μεταξύ Εξελικτικών Στρατηγικών και Εξελικτικού Προγραμματισμού.**

Μία από τις βασικές διαφορές τους είναι ότι στον Εξελικτικό Προγραμματισμό τυπικά χρησιμοποιούμε την στοχαστική Επιλογή αντί της απόρριψης. Κάθε δοκιμαστική λύση του πληθυσμού αντιμετωπίζει τον συναγωνισμό εναντίον ενός αριθμού προεπιλεγμένων στοιχείων και παίρνει τον χαρακτηρισμό ‘win’ αν τουλάχιστον κάποιο από τα συστατικά του είναι καλό από την σύγκριση. Στη διαδικασία της Επιλογής αποβάλλονται οι λύσεις με τα λιγότερα ‘win’. Αντίθετα οι Εξελικτικές Στρατηγικές χρησιμοποιούν αιτιοκρατικές μεθόδους Επιλογής. Σε αυτές οι χειρότερες λύσεις απορρίπτονται από τον πληθυσμό, βασίζονται στην συνάρτηση Αποτίμησης. [11], [12]

### Αναφορές :

- [1] Σύγχρονο λεξικό Πληροφορικής. Εκδόσεις 'Διάυλος'.
- [2] A Genetic Algorithm Tutorial. Editor 'Darrell Whitley'. *University of Colorado. Computer Science Department*. [www.cs.colostate.edu/TechReports/ Reports/1993/tr-103.pdf](http://www.cs.colostate.edu/TechReports/Reports/1993/tr-103.pdf)
- [3] [http://www.cs.bham.ac.uk/~rmp/slide\\_book/slide\\_book.html](http://www.cs.bham.ac.uk/~rmp/slide_book/slide_book.html) Editor 'Dr Riccardo Poli'. *University of Birmingham*. Computer Science Department.
- [4] <http://cs.felk.cvut.cz/~xobitko/ga/> . Editor 'by Marek Obitko'. *Czech Technical University*.
- [5] [http://www.systemtechnik.tu-ilmenau.de/~pohlheim/GA\\_Toolbox/](http://www.systemtechnik.tu-ilmenau.de/~pohlheim/GA_Toolbox/) .
- [6] <http://www.miosearch.com>
- [7] Adapting Crossover in a Genetic Algorithm. Editor 'by **William M. Spears**'. [www.aic.nrl.navy.mil/~spears/papers/adapt.crossover.pdf](http://www.aic.nrl.navy.mil/~spears/papers/adapt.crossover.pdf)
- [8] Μεταπτυχιακή Εργασία του Φώτη Θεοδώρου στα πλαίσια του μεταπτυχιακού προγράμματος του Πανεπιστημίου Ιωαννίνων. Έτος 2000 –2001.
- [9] GALib A C++ Library of Genetic Algorithm Components. Editor 'Matthew Wall'. *University of Massachusetts(U.S.A)*. [lancet.mit.edu/ga/dist/galibdoc.pdf](http://lancet.mit.edu/ga/dist/galibdoc.pdf)
- [10] Genetic Programming. Editor 'John R. Koza'. *University of Stanford(California)*. *Computer Science Department*.
- [11] Genetic Algorithm and neural Networks. Editor 'Periaux and G Winter'
- [12] <http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/ai-repository/ai/html/faqs/ai/genetic>. Editor 'Joerg Heitkoetter'. *University of Carnegie Mellon*.
- [13] Comparison of Parallel Genetic and Memetic Algorithms. Editor 'Jason Digalakis'. <http://macedonia.uom.gr/~jason/bookjason.pdf>.