

# Σύγκριση Νέων Εξελικτικών Αλγορίθμων

Σιλιαχλή Σουλτάνα

Πτυχιακή Εργασία

Πανεπιστήμιο Στερεάς Ελλάδας  
Τμήμα Πληροφορικής με Εφαρμογές  
στη Βιοϊατρική  
Λαμία

Επιβλέπων: Καθηγητής Βασίλειος Π. Πλαγιανάκος

(Οκτώβριος 2010)

ΣΤΟ ΣΗΜΕΙΟ ΑΥΤΟ ΘΑ ΗΘΕΛΑ ΝΑ ΕΚΦΡΑΣΩ ΤΙΣ ΕΥΧΑΡΙΣΤΙΕΣ ΜΟΥ ΣΤΟΝ ΕΠΙΒΛΕΠΟΝΤΑ ΚΑΘΗΓΗΤΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΜΟΥ ΕΡΓΑΣΙΑΣ Κ.ΒΑΣΙΛΕΙΟ Π. ΠΛΑΓΙΑΝΑΚΟ, ΕΠΙΚΟΥΡΟ ΚΑΘΗΓΗΤΗ ΤΟΥ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΣΤΕΡΕΑΣ ΕΛΛΑΔΑΣ, ΓΙΑ ΤΗΝ ΥΠΟΜΟΝΗ ΚΑΙ ΤΗ ΔΙΑΘΕΣΗ ΠΟΥ ΕΠΕΔΕΙΞΕ ΟΛΟ ΤΟ ΔΙΑΣΤΗΜΑ ΠΟΥ ΜΕ ΚΑΘΟΔΗΓΟΥΣΕ ΚΑΙ ΜΕ ΕΠΕΒΛΕΠΕ.



<b>I</b>	<b>Εισαγωγή και Βασικές Έννοιες</b>	<b>1</b>
<b>1</b>	<b>Εισαγωγή</b>	<b>3</b>
1.1	Εισαγωγή . . . . .	3
1.1.1	Συστατικά των Εξελικτικών Αλγορίθμων . . . . .	4
1.1.2	Αναπαράσταση (προσδιορισμός των ατόμων) . . . . .	5
1.1.3	Συνάρτηση καταλληλότητας . . . . .	5
1.1.4	Πληθυσμός . . . . .	5
1.1.5	Μηχανισμός επιλογής γονέων . . . . .	6
1.1.6	Τελεστές μετάλλαξης . . . . .	6
1.1.7	Μηχανισμός επιλογής επιβίωσης (αντικατάσταση) . . . . .	7
<b>II</b>	<b>Κατηγορίες Εξελικτικών Αλγορίθμων</b>	<b>9</b>
<b>2</b>	<b>Γενετικοί Αλγόριθμοι (Genetic Algorithms)</b>	<b>11</b>
2.1	Σκοπός του Κεφαλαίου . . . . .	11
2.2	Παράδειγμα εισαγωγής . . . . .	11
2.3	Αναπαράσταση των ατόμων . . . . .	12
2.3.1	Δυαδικές αναπαραστάσεις . . . . .	13
2.3.2	Αναπαραστάσεις ακεραίων . . . . .	14
2.3.3	Πραγματικές μεταβλητές ή κινητής υποδιαστολής σημεία αναπαράστασης	14
2.3.4	Αναπαραστάσεις αντιμετάθεσης . . . . .	14
2.4	Μετάλλαξη . . . . .	15
2.4.1	Μετάλλαξη για δυαδικές αναπαραστάσεις . . . . .	16
2.4.2	Τελεστές μετάλλαξης για ακέραιες αναπαραστάσεις . . . . .	16
2.4.3	Τελεστές μετάλλαξης για κινητής υποδιαστολής αναπαραστάσεις . . . . .	17
2.4.4	Τελεστές μετάλλαξης για αναπαραστάσεις αντιμετάθεσης . . . . .	17
2.5	Ανασυνδιασμός . . . . .	18
2.5.1	Τελεστές ανασυνδιασμού για δυαδικές αναπαραστάσεις . . . . .	19
2.5.2	Τελεστές ανασυνδιασμού για ακέραιες αναπαραστάσεις . . . . .	20
2.5.3	Τελεστές ανασυνδιασμού για κινητής υποδιαστολής αναπαραστάσεις . . . . .	20
2.5.4	Τελεστές ανασυνδιασμού για μετάλλαξη αναπαραστάσεων . . . . .	22
2.5.5	Ανασυνδιασμός πολλαπλών γονέων . . . . .	26
2.6	Μοντέλα πληθυσμού . . . . .	27
2.7	Επιλογή γονέων . . . . .	28
2.7.1	Επιλογή αναλογικής καταλληλότητας . . . . .	28
2.7.2	Υλοποίηση των πιθανοτήτων επιλογής . . . . .	29
2.7.3	Επιλογή διαγωνισμού . . . . .	30
2.8	Επιλογή επιβίωσης . . . . .	31
2.8.1	Αντικατάσταση που βασίζεται στην ηλικία . . . . .	32
2.8.2	Αντικατάσταση βασισμένη στην καταλληλότητα . . . . .	32
2.9	Παράδειγμα εφαρμογής: Επίλυση του προβλήματος προγραμματισμός εργασίας . . . . .	33

<b>3</b>	<b>Εξελικτικές Στρατηγικές (Evolution Strategies)</b>	<b>35</b>
3.1	Σκοπός αυτού του κεφαλαίου . . . . .	35
3.2	Παράδειγμα εισαγωγής . . . . .	35
3.3	Αναπαράσταση . . . . .	36
3.4	Μετάλλαξη . . . . .	37
3.4.1	Ασυσχέτιστες μεταλλάξεις με βήμα μετάλλαξης ένα . . . . .	38
3.4.2	Ασυσχέτιστες μεταλλάξεις με βήμα μετάλλαξης $n$ . . . . .	38
3.4.3	Συσχετισμένες μεταλλάξεις . . . . .	40
3.5	Ανασυνδιασμός . . . . .	40
3.6	Επιλογή γονέων . . . . .	41
3.7	Επιλογή επιβίωσης . . . . .	42
3.8	Αυτοπροσαρμογή . . . . .	42
<b>4</b>	<b>Εξελικτικός Προγραμματισμός (Evolutionary Programming)</b>	<b>45</b>
4.1	Σκοπός αυτού του κεφαλαίου . . . . .	45
4.2	Παράδειγμα εισαγωγής . . . . .	45
4.3	Αναπαράσταση . . . . .	46
4.4	Μετάλλαξη . . . . .	46
4.5	Ανασυνδιασμός . . . . .	47
4.6	Επιλογή γονέων . . . . .	48
4.7	Επιλογή επιβίωσης . . . . .	48
<b>5</b>	<b>Γενετικός Προγραμματισμός (Genetic Programming)</b>	<b>49</b>
5.1	Σκοπός αυτού του κεφαλαίου . . . . .	49
5.2	Παράδειγμα εισαγωγής . . . . .	49
5.3	Αναπαράσταση . . . . .	50
5.4	Μετάλλαξη . . . . .	52
5.5	Ανασυνδιασμός . . . . .	52
5.6	Επιλογή γονέων . . . . .	52
5.7	Επιλογή επιβίωσης . . . . .	53
5.8	Αρχικοποίηση . . . . .	53
<b>III</b>	<b>Αποτελέσματα</b>	<b>55</b>
<b>6</b>	<b>Διαφοροεξελικτικοί Αλγόριθμοι (Differential Evolution)</b>	<b>57</b>
6.1	Εισαγωγικά . . . . .	57
6.2	Ο αλγόριθμος . . . . .	57
6.3	Ρύθμιση παραμέτρων . . . . .	58
6.4	Μεταβλητές . . . . .	58
6.5	Αποτελέσματα . . . . .	58
<b>7</b>	<b>Βελτιστοποίηση με σμήνος σωματιδίων (Particle Swarm Optimization)</b>	<b>69</b>
7.1	Εισαγωγικά . . . . .	69
7.2	Ο αλγόριθμος . . . . .	69
7.3	Ρύθμιση παραμέτρων . . . . .	70
7.4	Μεταβλητές . . . . .	70
7.5	Αποτελέσματα . . . . .	70

<b>8</b>	<b>Εξελικτική στρατηγική προσαρμογής πινάκων συνδιακύμανσης (Covariance Matrix Adaptation Evolution Strategy (CMA-ES))</b>	<b>77</b>
8.1	Εισαγωγικά . . . . .	77
8.2	Αρχές . . . . .	77
8.3	Αποτελέσματα . . . . .	77
<b>IV</b>	<b>Συμπεράσματα – Βιβλιογραφία</b>	<b>85</b>
<b>9</b>	<b>Συμπεράσματα</b>	<b>87</b>
9.1	Εισαγωγικά . . . . .	87
9.2	Σύγκριση αποτελεσμάτων . . . . .	88
9.3	Συμπεράσματα . . . . .	93
	<b>Βιβλιογραφία</b>	<b>95</b>



---

## Κατάλογος Σχημάτων

---

2.1	Λογική μετάλλαξη για δυαδικές κωδικοποιήσεις . . . . .	16
2.2	Μετάλλαξη αντιμετάθεσης . . . . .	18
2.3	Μετάλλαξη εισαγωγής . . . . .	18
2.4	Μετάλλαξη ανάμειξης . . . . .	18
2.5	Μετάλλαξη αναστροφής . . . . .	19
2.6	Διασταύρωση ενός σημείου . . . . .	19
2.7	Διασταύρωση $n$ -σημείων: $n = 2$ . . . . .	19
2.8	Ομοιόμορφη διασταύρωση: Σε αυτό το παράδειγμα ο πίνακας $[0.35, 0.62, 0.18, 0.42, 0.83, 0.76, 0.39, 0.51, 0.36]$ χρησιμοποιήθηκε για να αποφασιστεί η κληρονομικότητα των τυχαίων μεταβλητών που σχεδιάστηκαν ομοιόμορφα από το $[0, 1)$ .	20
2.9	Απλός αριθμητικός ανασυνδιασμός: $k = 8, a = 1/2$ . . . . .	21
2.10	Μονός αριθμητικός ανασυνδιασμός: $k = 3, a = 1/2$ . . . . .	21
2.11	Ολόκληρος αριθμητικός ανασυνδιασμός: $a = 1/2$ . . . . .	22
2.12	ΜΧΔ, βήμα 1: αντιγράφονται τα τυχαίως επιλεγμένα σημεία από τον πρώτο γονέα στον απόγονο . . . . .	23
2.13	ΜΧΔ, βήμα 2: εξετάζει με τη σειρά του την τοποθέτηση των στοιχείων που παρουσιάζονται στο μεσαίο τμήμα του γονέα 2 και όχι του γονέα 1. Η θέση όπου το 8 έχει στον $P2$ καταλαμβάνεται από το 4 στον απόγονο, έτσι ώστε να μπορούμε να τοποθετήσουμε το 8 στη θέση που εκκενώθηκε από το 4 στον $P2$ . Η θέση του 2 στον $P2$ καταλαμβάνεται από το 5 στον απόγονο, έτσι κοιτάμε πρώτα την θέση που κατέχει το 5 στον $P2$ , όπου είναι η θέση 7. Αυτή είναι ήδη κατειλημμένη από την μεταβλητή 7, έτσι ώστε να δούμε το πού θα πάει στον $P2$ και τελικά να βρεθεί ένα άνοιγμα στον απόγονο το οποίο είναι κενό-το τρίτο. Τέλος, να σημειωθεί ότι οι μεταβλητές 6 και 5 καταλαμβάνονται στα μεσαία τμήματα και των δύο γονέων. . . . .	23
2.14	ΜΧΔ, βήμα 3: αντιγράφονται τα υπόλοιπα στοιχεία από τον δεύτερο γονέα στις ίδιες θέσεις στον απόγονο . . . . .	23
2.15	Σειρά διασταύρωσης, βήμα 1: αντιγράφονται τα τυχαία επιλεγμένα τμήματα από τον πρώτο γονέα στον απόγονο . . . . .	25
2.16	Σειρά διασταύρωσης, βήμα 2: αντιγράφονται τα υπόλοιπα των αλληλουχιών με τη σειρά που εμφανίζονται στο δεύτερο γονέα, επεξεργάζοντας την ακολουθία ως σπειροειδή . . . . .	25
2.17	Κυκλική διασταύρωση, βήμα 1: αναγνώριση των κύκλων . . . . .	26
2.18	Κυκλική διασταύρωση, βήμα 2: κατασκευή του απογόνου . . . . .	26
2.19	Ο ψευδοκώδικας για τον αλγόριθμο της ρουλέτας . . . . .	30
2.20	Ο ψευδοκώδικας για τον στοχαστικό αλγόριθμο γενικού δείγματος . . . . .	30
2.21	Ο ψευδοκώδικας για τον αλγόριθμο επιλογής τουρνουά . . . . .	32
3.1	Περιγραφή μίας απλής διμελούς εξελικτικής στρατηγικής . . . . .	36



3.2	Μετάλλαξη με $n = 2$ , $n_\sigma = 1$ , $n_\alpha = 0$ . Κομμάτι από ένα μέρος της καταλληλότητας με ένα κωνικό σχήμα να φαίνεται. Το μαύρο δεν υποδεικνύει ένα άτομο. Τα σημεία όπου οι απόγονοι μπορούν να διατεθούν με μία δοσμένη πιθανότητα σχηματίζουν ένα κύκλο. Η πιθανότητα μετακίνησης κατά μήκος του άξονα $y$ (μικρή επίδραση της καταλληλότητας) είναι η ίδια με αυτή της μετακίνησης κατά μήκος του άξονα $x$ (μεγάλη επίδραση της καταλληλότητας).	39
3.3	Μετάλλαξη με $n = 2$ , $n_\sigma = 2$ , $n_\alpha = 0$ . Κομμάτι από ένα μέρος της καταλληλότητας με ένα κωνικό σχήμα να φαίνεται. Το μαύρο δεν υποδεικνύει ένα άτομο. Τα σημεία όπου οι απόγονοι μπορούν να διατεθούν με μία δοσμένη πιθανότητα σχηματίζουν μία έλλειψη. Η πιθανότητα μετακίνησης κατά μήκος του άξονα $x$ (μεγάλη επίδραση της καταλληλότητας) είναι μεγαλύτερη της μετακίνησης κατά μήκος του άξονα $y$ (μικρή επίδραση της καταλληλότητας).	39
3.4	Μετάλλαξη με $n = 2$ , $n_\sigma = 2$ , $n_\alpha = 1$ . Κομμάτι από ένα μέρος της καταλληλότητας με ένα κωνικό σχήμα να φαίνεται. Το μαύρο δεν υποδεικνύει ένα άτομο. Τα σημεία όπου οι απόγονοι μπορούν να διατεθούν με μία δοσμένη πιθανότητα σχηματίζουν μία περιστρεφόμενη έλλειψη. Η πιθανότητα παραγωγής μίας κίνησης προς την διεύθυνση μίας απότομης ανάβασης (μεγαλύτερη επίδραση της καταλληλότητας) είναι μεγαλύτερη από αυτές άλλων διευθύνσεων. . . . .	41
4.1	Παράδειγμα πεπερασμένων μηχανισμών κατάστασης που αποτελείται από τρεις καταστάσεις: A,B,C. Το αλφάβητο εισόδου είναι $I = 0, 1$ και το αλφάβητο εξόδου είναι $O = a, b, c$ . Η συνάρτηση μετάβασης ΠΜΚ $d : S \times I \rightarrow S \times O$ μετατρέπει το ρεύμα εισόδου στο ρεύμα εξόδου που προσδιορίζεται από τα βέλη και τις ετικέτες υποδεικνύοντας την είσοδο/έξοδο από την δοσμένη μετάβαση.	46
5.1	Αναλυτικό δένδρο . . . . .	50
5.2	Αναλυτικό δένδρο που ανήκει στις σχέσεις (5.2) (αριστερά) και (5.3) (δεξιά) .	51
5.3	Αναλυτικό δένδρο που ανήκει στο παραπάνω πρόγραμμα . . . . .	51
5.4	Η μετάλλαξη ΓΠ υλοποιείται: Ο κόμβος που σχεδιάστηκε από ένα κύκλο στο αριστερό δένδρο επιλέχθηκε για μετάλλαξη. Το υποδένδρο που ξεκινάει από αυτόν τον κόμβο αντικαθιστάται από ένα τυχαία παραγόμενο δένδρο, το οποίο είναι ένα φύλλο εδώ. . . . .	52
5.5	Η διασταύρωση του ΓΠ υλοποιείται: οι κόμβοι που σχεδιάζονται από έναν κύκλο στα γονικά δένδρα επιλέγονται ως σημεία διασταύρωσης. Τα υποδένδρα που ξεκινούν από αυτούς τους κόμβους αντιστρέφονται, με αποτέλεσμα να έχουμε δύο νέα δένδρα, τα οποία είναι τα παιδιά. . . . .	53
6.1	Γραφική παράσταση της συνάρτησης DeJong . . . . .	59
6.2	Γραφική παράσταση της συνάρτησης Griewank . . . . .	59
6.3	Γραφική παράσταση της συνάρτησης Rastrigrin . . . . .	60
6.4	Γραφική παράσταση της συνάρτησης Rosenbrock . . . . .	60

---

## Κατάλογος Πινάκων

---

1.1	Η βασική μεταφορά εξελικτικού υπολογισμού συνδέει την φυσική εξέλιξη της επίλυσης του προβλήματος . . . . .	4
2.1	Πίνακας απλού ΓΑ . . . . .	12
2.2	Παράδειγμα $x^2$ , 1:αρχικοποίηση, εκτίμηση και επιλογή γονέα . . . . .	12
2.3	Παράδειγμα $x^2$ , 2:διασταύρωση και αποτίμηση απογόνου . . . . .	13
2.4	Παράδειγμα $x^2$ , 3:μετάλλαξη και αποτίμηση απογόνου . . . . .	13
2.5	Διασταύρωση στην άκρη: παράδειγμα με πίνακα άκρων . . . . .	24
2.6	Διασταύρωση στην άκρη: παράδειγμα της κατασκευής μετάλλαξης . . . . .	25
2.7	Αναλογική καταλληλότητα (FP) έναντι επιλογής γραμμικής σειράς (LR) . . . . .	29
3.1	Σκίτσο της ΕΣ . . . . .	35
3.2	Μερικές πιθανές τοποθετήσεις του $n_\sigma$ και του $n_\alpha$ για διάφορους τελεστές μετάλλαξης . . . . .	41
4.1	Σκίτσο του ΕΠ . . . . .	45
5.1	Σκίτσο του ΓΠ . . . . .	49
5.2	Δεδομένα για το πρόβλημα της τράπεζας . . . . .	50
5.3	Το σει συνάρτησης και το τελικό σει της έκφρασης της εξίσωσης (5.2) . . . . .	51
5.4	Αναλογία του ταξινομημένου πληθυσμού σε ‘καταλληλότερο’ υποπληθυσμό από τον οποίο επιλέχθηκε η πλειονότητα των γονέων . . . . .	53
6.1	Πίνακας συνάρτησης DeJong στην κατηγορία DE . . . . .	64
6.2	Πίνακας συνάρτησης Rosenbrock στην κατηγορία DE . . . . .	65
6.3	Πίνακας συνάρτησης Griewank στην κατηγορία DE . . . . .	66
6.4	Πίνακας συνάρτησης Rastrigrin στην κατηγορία DE . . . . .	67
7.1	Πίνακας συναρτήσεων DeJong και Rosenbrock στην κατηγορία PSO . . . . .	74
7.2	Πίνακας συναρτήσεων Griewnak και Rastrigrin στην κατηγορία PSO . . . . .	75
8.1	Πίνακας συναρτήσεων DeJong και Rosenbrock στην κατηγορία CMA-ES . . . . .	83
8.2	Πίνακας συναρτήσεων Griewnak και Rastrigrin στην κατηγορία CMA-ES . . . . .	84

**Μέρος Ι**

**Εισαγωγή και Βασικές Έννοιες**



---

## Εισαγωγή

---

### 1.1 Εισαγωγή

Οι *Εξελικτικοί Αλγόριθμοι* (Evolutionary Algorithms) σχεδιάστηκαν για να μπορούν να αντιμετωπίσουν δύσκολα προβλήματα Βελτιστοποίησης, όπου η αντικειμενική συνάρτηση είναι μη γραμμική, πιθανά μη παραγωγίσιμη και έχει πολλά τοπικά, αλλά και ολικά ελάχιστα. Για να επιτευχθεί αυτός ο στόχος, οι Εξελικτικοί Αλγόριθμοι είναι μία κλάση στοχαστικών παράλληλων μεθόδων που δανείζεται ιδέες από την φύση και την εξέλιξη των ειδών με σκοπό να επιλύσουν ένα μαθηματικό πρόβλημα. Οι Εξελικτικοί Αλγόριθμοι έχουν καλές ιδιότητες σύγκλισης (ταχύτητα, ποσοστό επιτυχίας, εύρεση του ολικού ελαχίστου) και ξεπερνούν σε επίδοση τους κλασικούς αλγόριθμους Βελτιστοποίησης.

Ο *Εξελικτικός Υπολογισμός* (Evolutionary Computing) είναι μια περιοχή έρευνας της επιστήμης των ηλεκτρονικών υπολογιστών. Όπως υποδηλώνει και το όνομα του, είναι μια ιδιαίτερη αίσθηση του υπολογισμού, που εμπνέεται από την πρόοδο της φυσικής εξέλιξης. Γι' αυτό ορισμένοι επιστήμονες των ηλεκτρονικών υπολογιστών έχουν επιλέξει την φυσική εξέλιξη ως πηγή έμπνευσης, διότι η δύναμη της εξέλιξης στην φύση είναι φανερή σε διάφορα είδη, με το κάθε είδος να επιβιώνει καλά από την δική του θέση. Η θεμελιώδης μεταφορά του εξελικτικού υπολογισμού συσχετίζει αυτή την δυνατή φυσική εξέλιξη με το ιδιαίτερο στυλ επίλυσης προβλημάτων - αυτό της δοκιμής και του λάθους.

Περιγραφές σχετικών αναφορών της εξελικτικής θεωρίας και γενετικής θα δοθούν αργότερα. Προς το παρόν μας επιτράπηκε να μελετήσουμε την φυσική εξέλιξη τόσο απλά όσο παρακάτω. Ένα δοσμένο περιβάλλον είναι γεμάτο με έναν πληθυσμό λύσεων που αντιμάχεται για την επιβίωση και την αναπαραγωγή [1]. Η καταλληλότητα αυτών των πιθανών λύσεων - που αποφασίζεται από το περιβάλλον - συσχετίζει το πόσο καλά αυτές καταφέρνουν να πετύχουν τους στόχους τους, π.χ. παρουσιάζει τις ευκαιρίες επιβίωσης και πολλαπλασιασμού. Στο πλαίσιο μίας διαδικασίας επίλυσης ενός προβλήματος με ένα στοχαστικό στυλ δοκιμής και λάθους (γνωστά επίσης και ως παράγω και έλεγχω), έχουμε μία συλλογή από προτεινόμενες λύσεις. Η ποιότητα τους (τι είναι, πόσο καλά επιλύουν το πρόβλημα) καθορίζει την ευκαιρία να κρατηθούν και να χρησιμοποιηθούν ως βάσεις για τη δημιουργία περισσότερων υποψήφιας λύσεων.

Ο *Εξελικτικός Αλγόριθμος* (Evolutionary Algorithm) είναι ένα υποσύνολο του εξελικτικού υπολογισμού και από την ιστορία φαίνεται ότι υπάρχουν διάφορες μεταβλητές των εξελικτικών αλγορίθμων. Η βασική κοινή ιδέα πίσω από όλες τις τεχνικές είναι η ίδια: δοσμένου των πιθανών λύσεων η πίεση του περιβάλλοντος προκαλεί φυσική επιλογή, η οποία προκαλεί μία αύξηση στην καταλληλότητα του πληθυσμού. Για να βρεθεί το μέγιστο μίας δοσμένης ποιοτικής συνάρτησης, μπορούμε σύντομα να δημιουργήσουμε μία σειρά από προτεινόμενες λύσεις, π.χ. στα στοιχεία των συναρτήσεων με πεδίο ορισμού να εφαρμόσουμε την ποιοτική συνάρτηση ως ένα διαχωρισμό στο μέτρο της μορφής-όσο πιο ψηλά τόσο πιο καλά. Βασισμένο σε αυτή τη μορφή, μερικές από τις καλύτερες προτεινόμενες λύσεις που επιλέχθηκαν για να τροφοδοτήσουν την επόμενη γενιά εφαρμόζουν ανασυνδιασμό γονιδίων και/ή μετάλλαξη αυτών. Ο ανασυνδιασμός γονιδίων είναι ένας τελεστής που εφαρμόζεται

Εξέλιξη	Επίλυση προβλήματος
Περιβάλλον	↔ Πρόβλημα
Λύση	↔ Υποψήφια λύση
Καταλληλότητα	↔ Ποιότητα

**Πίνακας 1.1:** Η βασική μεταφορά εξελικτικού υπολογισμού συνδέει την φυσική εξέλιξη της επίλυσης του προβλήματος

σε δύο ή περισσότερους υποψηφίους πληθυσμούς (ονομάζονται και γονείς) και προκύπτουν ένας ή περισσότεροι νέοι υποψήφιοι (τα παιδιά). Η μετάλλαξη εφαρμόζεται σε έναν υποψήφιο πληθυσμό γονιό και προκύπτει ένας νέος υποψήφιος πληθυσμός [2]. Εκτελώντας τον ανασυνδιασμό και την μετάλλαξη οδηγούμαστε σε ένα σετ νέων υποψήφιας απογόνων το οποίο συναγωνίζεται - βασισμένο στη καταλληλότητά τους (και πιθανώς την ηλικία) - με τα παλιότερα αυτών, για μία θέση στην επόμενη παραγωγή. Αυτή η διαδικασία μπορεί να επαναλαμβάνεται συνεχώς μέχρι οι υποψήφιοι με επαρκή ποιότητα (μία λύση) να βρεθούν ή ένα προηγούμενο σετ υπολογιστικών ορίων να ξαναεμφανιστεί.

Σε αυτή τη διαδικασία υπάρχουν δύο θεμελιώδεις δυνάμεις οι οποίες αναπτύσσουν την βάση για τα εξελικτικά συστήματα :

1. Οι τελεστές των μεταβλητών (ανασυνδιασμός γονιδίων και μετάλλαξη) δημιουργούν την απαραίτητη ποικιλία και έτσι διευκολύνεται η καινοτομία.
2. Η επιλογή δρα ως μία επιθετική δύναμη ποιότητας.

Η συνδυασμένη εφαρμογή της απόκλισης και γενικά της επιλογής οδηγεί στην βελτίωση των μεταβλητών της καταλληλότητας στους συνεχή πληθυσμούς. Είναι εύκολο (αν και κάπως παραπλανητικό) να δούμε μία τέτοια διαδικασία σαν να ήταν η εξέλιξη της βελτίωσης, ή τουλάχιστον 'προσεγγιστικά', προσεγγίζοντας οπτικές μεταβλητές [3]. Εναλλακτικά, η εξέλιξη συχνά θεωρείται ως μία διαδικασία προσαρμογής. Από αυτή την άποψη, η καταλληλότητα δεν θεωρείται ως μία αντικειμενική συνάρτηση που θα βελτιστοποιηθεί, αλλά ως μία έκφραση των απαιτήσεων του περιβάλλοντος. Η πιο κοντινή αντιστοίχιση αυτών των απαιτήσεων συνεπάγεται μία αυξημένη βιωσιμότητα, και αντανακλάται σε έναν υψηλότερο αριθμό απογόνων. Η εξελικτική διαδικασία κάνει τον πληθυσμό όλο και καλύτερο στο να είναι προσαρμοσμένος στο περιβάλλον.

### 1.1.1 Συστατικά των Εξελικτικών Αλγορίθμων

Σε αυτό το σημείο θα αναφερθούμε στους εξελικτικούς αλγορίθμους. Οι εξελικτικοί αλγόριθμοι έχουν έναν αριθμό συστατικών, διαδικασιών ή τελεστών που πρέπει να καθοριστούν προκειμένου να προσδιοριστούν με λεπτομέρεια οι εξελικτικοί αλγόριθμοι. Τα πιο σημαντικά συστατικά είναι :

1. Αναπαράσταση (προσδιορισμός των ατόμων)
2. Συνάρτηση καταλληλότητας
3. Πληθυσμός
4. Μηχανισμός επιλογής γονέων
5. Τελεστές μεταβλητών, ανασυνδιασμός γονιδίων και μετάλλαξη
6. Μηχανισμός επιλογής επιβίωσης (αντικατάσταση)

Καθένα από αυτά τα συστατικά πρέπει να προσδιοριστούν προκειμένου να καθοριστούν με λεπτομέρεια οι εξελικτικοί αλγόριθμοι [4]. Επιπλέον, για να εξασφαλιστεί η διαδικασία ενός τρέχοντος αλγορίθμου πρέπει να διευκρινιστούν η διαδικασία της αρχικοποίησης και του τερματισμού.

### 1.1.2 Αναπαράσταση (προσδιορισμός των ατόμων)

Το πρώτο βήμα για να προσδιοριστεί ένας εξελικτικός αλγόριθμος είναι να συνδεθεί ο πραγματικός κόσμος με τον κόσμο των εξελικτικών αλγορίθμων, και αυτό γίνεται αν δημιουργηθεί μία γέφυρα μεταξύ του αυθεντικού πλαισίου του προβλήματος και του χώρου επίλυσης του προβλήματος όπου η εξέλιξη λαμβάνει μέρος. Τα αντικείμενα που παρουσιάζουν πιθανές λύσεις με το αυθεντικό πλαίσιο του προβλήματος αναφέρονται ως φαινότυποι, ενώ όταν αυτοί κωδικοποιούνται, δηλαδή οι πιθανές λύσεις μέσα στους εξελικτικούς αλγόριθμους, ονομάζονται γονότυποι [5]. Το πρώτο σχεδιαστικό βήμα είναι η κοινή αναπαράσταση, καθώς αυτή μετρά τις προδιαγραφές για την χαρτογράφηση των φαινοτύπων πάνω σε ένα σετ γονότυπων οι οποίοι αναπαριστούν αυτούς τους φαινότυπους. Για παράδειγμα, δίνοντας σε ένα πρόβλημα βελτιστοποίησης ακέραιους αριθμούς, τα δοσμένα σετ ακέραιων αριθμών θα σχηματίσουν το σετ των δεδομένων των φαινοτύπων. Έπειτα ένα σετ θα μπορούσε να αποφασιστεί για να τα αναπαραστήσει όλα από τον δυαδικό κωδικό τους, γι' αυτό το λόγο το 18 θα μπορούσε να φαίνεται ως ένας φαινότυπος και το 10010 ως ένας γονότυπος που το αντιπροσωπεύει. Είναι σημαντικό να γίνει κατανοητό ότι ο χώρος των φαινοτύπων μπορεί να είναι πολύ διαφορετικός από τον χώρο των γονοτύπων και ότι ολόκληρη η εξελικτική αναζήτηση πραγματοποιείται στον χώρο των γονοτύπων. Μια λύση - ένας καλός φαινότυπος - αποκτάται αποκωδικοποιώντας τον καλύτερο γονότυπο μετά τη λήξη της διαδικασίας. Τελειώνοντας, πρέπει να κρατηθεί ότι η καταλληλότερη επίλυση του προβλήματος - ενός φαινοτύπου - αντιπροσωπεύεται από τον δοσμένο χώρο του γονότυπου.

### 1.1.3 Συνάρτηση καταλληλότητας

Ο ρόλος της συνάρτησης επαναλήψεων είναι να αναπαριστώνται οι απαιτήσεις που προσαρμόζονται σε αυτή. Διευθετεί την βάση για την επιλογή και με αυτό τον τρόπο διευκολύνει τις βελτιώσεις. Επακριβώς, διευκρινίζει τι σημαίνει κάθε βελτίωση. Από την πλευρά της επίλυσης του προβλήματος, αντιπροσωπεύεται ο στόχος της επίλυσης του εξελικτικού πλαισίου. Τεχνικά, είναι μία συνάρτηση ή μία διαδικασία που προσδιορίζει ένα μέτρο ποιότητας στους γονότυπους. Τυπικά, αυτή η συνάρτηση αποτελείται από ένα μέτρο ποιότητας στο χώρο των φαινοτύπων και το αντίστροφο στην αναπαράσταση. Για να παραμείνουμε στο παραπάνω πείραμα, εάν θέλαμε να μεγιστοποιήσουμε το  $x^2$  στους ακεραίους, η κατάσταση των γονότυπων 10010 θα μπορούσε να καθοριστεί ως το τετράγωνο των αντίστοιχων φαινοτύπων:  $18^2 = 324$ .

### 1.1.4 Πληθυσμός

Ο ρόλος του πληθυσμού είναι να διατηρεί τις πιθανές λύσεις. Ο πληθυσμός είναι ένα σετ γονότυπων (στο οποίο ο πολλαπλασιασμός κάποιου στοιχείου είναι πιθανός). Ο πληθυσμός είναι ένα σετ γονότυπων όπου πολλαπλά αντίγραφα ενός στοιχείου είναι πιθανά [6]. Ο πληθυσμός σχηματίζει την ομάδα εξέλιξης. Οι πιθανές λύσεις είναι στατικά αντικείμενα τα οποία δεν αλλάζουν ή δεν προσαρμόζονται, ο πληθυσμός είναι αυτός που το κάνει. Δίνοντας μία αναπαράσταση, το να προσδιορίσεις ένα πληθυσμό μπορεί να είναι τόσο απλό όσο το να προσδιοριστούν οι πιθανές λύσεις σε αυτό, δηλαδή, να καθοριστεί το μέγεθος του πληθυσμού. Σε μερικούς εξεζητημένους εξελικτικούς αλγορίθμους ένας πληθυσμός έχει ένα επιπλέον χώρο δομής, με ένα μέτρο απόστασης ή μία γειτονική σχέση. Σε αυτές τις περιπτώσεις η επιπλέον δομή πρέπει να καθοριστεί εξίσου καλά με το πλήρη καθορισμό ενός πληθυσμού.

Καθώς αντικρούονται με την ποικιλία των τελεστών οι οποίοι δρουν σε έναν ή δύο γονείς μεμονωμένα, η επιλογή των τελεστών (επιλογή γονέων και επιλογή επιβίωσης) λειτουργεί στο επίπεδο του πληθυσμού. Γενικά, για τον υπολογισμό παίρνουν ολόκληρο τον τρέχοντα πληθυσμό και οι επιλογές είναι πάντα σχετικές με το τι έχουμε. Για παράδειγμα, το καλύτερο άτομο από τους δοσμένους πληθυσμούς επιλέγεται για να ψάξει την επόμενη γενιά, ή το χειρότερο άτομο από τους δοσμένους επιλέγεται για να αντικατασταθεί από μία καινούρια. Σχεδόν σε όλους τους εξελικτικούς αλγορίθμους οι εφαρμογές στο μέγεθος του πληθυσμού είναι σταθερές και δεν αλλάζουν κατά τη διάρκεια της εξελικτικής αναζήτησης.

Η ποικιλία ενός πληθυσμού είναι ένα μέτρο των αριθμών των διαφορετικών λύσεων που παρουσιάζονται. Δεν υπάρχει ένα μονό μέτρο για την ποικιλία. Τυπικά οι άνθρωποι μπορεί να προτιμούν τον αριθμό των διαφορετικών μεταβλητών καταλληλότητας που παρουσιάζονται, τον αριθμό των διαφορετικών φαινότυπων που παρουσιάζονται, ή τον αριθμό των διαφορετικών γονότυπων. Άλλοι χρησιμοποιούν στατιστικά μέτρα όπως η εντροπία. Σημειώνουμε ότι μόνο μία μεταβλητή καταλληλότητας δεν συνεπάγεται απαραίτητα μόνο ένα φαινότυπο που παρουσιάζεται και με την σειρά του μόνο ένας φαινότυπος δεν συνεπάγεται απαραίτητα μόνο ένα γονότυπο. Το αντίθετο, ωστόσο, δεν είναι αληθές: ένας γονότυπος συνεπάγεται μόνο ένα φαινότυπο και μία τιμή της καταλληλότητας.

### 1.1.5 Μηχανισμός επιλογής γονέων

Ο ρόλος της επιλογής γονέων ή επιλογή ζευγαριού είναι να διευκρινιστεί πέρα από τα άτομα που βασίζονται στην ποιότητά τους, να επιτραπούν τα καλύτερα άτομα να γίνουν γονείς της επόμενης γενιάς. Ένα άτομο είναι γονιός εάν έχει επιλεγεί να υποβληθεί σε μία ποικιλία προκειμένου να δημιουργηθεί απόγονος. Μαζί με τον μηχανισμό επιλογής επιβίωσης, η επιλογή γονέων είναι υπεύθυνη για την ώθηση της βελτιώσης της καταλληλότητας. Στον εξελικτικό υπολογισμό, η επιλογή γονέων είναι τυπικά πιθανή. Επιπλέον, υψηλής ποιότητας άτομα έχουν μεγαλύτερη πιθανότητα να γίνουν γονείς από αυτές με χαμηλή ποιότητα. Παρ' όλα αυτά, η χαμηλής ποιότητας άτομα δίνουν συχνά μία μικρή πιθανότητα αλλά θετική, διαφορετικά ολόκληρη η αναζήτηση θα μπορούσε να γίνει πολύ άπληστη και να κολλήσει σε ένα τοπικό βέλτιστο.

### 1.1.6 Τελεστές μετάλλαξης

Ο ρόλος των τελεστών μετάλλαξης είναι να δημιουργήσει νέα άτομα από τα παλιά. Στον αντίστοιχο χώρο των φαινότυπων αυτές οι ποσότητες είναι για την παραγωγή νέων υποψηφίων λύσεων. Από την πλευρά της παραγωγής-και-του-ελέγχου αναζήτησης, οι τελεστές μετάλλαξης παρουσιάζουν το βήμα της 'παραγωγής'. Οι τελεστές μετάλλαξης στον εξελικτικό υπολογισμό χωρίζονται σε δύο τύπους που βασίζονται στον αριθμό των παραμέτρων που παίρνουν ως όρισμα.

#### Μετάλλαξη

Ένα μοναδιαίο σύμβολο τελεστή καλείται συνήθως μετάλλαξη. Εφαρμόζεται σε έναν γονότυπο και παραδίδει ένα (ελαφρώς) τροποποιημένο μεταλλαγμένο, το παιδί ή τον απόγονο από αυτό. Ένας τελεστής μετάλλαξης είναι πάντα στοχαστικός: οι έξοδοι του -το παιδί- εξαρτώνται από τα αποτελέσματα μίας σειράς τυχαίων επιλογών. Θα πρέπει να σημειωθεί ότι ένας αυθαίρετος μοναδιαίος τελεστής δεν αναζητείται απαραίτητα σαν μετάλλαξη. Ένας τελεστής που είναι ικανός για λύση σε ένα συγκεκριμένο πρόβλημα που δρα μόνο σε ένα άτομο θα μπορούσε να εκφραστεί ως μετάλλαξη όντας μοναδιαίος [24]. Ωστόσο, γενικά στη μετάλλαξη υποτίθεται ότι η πιθανότητα προκαλείται τυχαία και αμερόληπτα. Γι' αυτό το λόγο θα ήταν πιο κατάλληλο να μην απαιτούμαι ευριστικούς μοναδιαίους τελεστές στη μετάλλαξη. Ο ρόλος της μετάλλαξης στον εξελικτικό υπολογισμό είναι διαφορετικός σε διάφορες διαλέκτους του εξελικτικού υπολογισμού. Για παράδειγμα, στο γενετικό προγραμματισμό συχνά



δε χρησιμοποιείται καθόλου, ενώ στους γενετικούς αλγορίθμους παραδοσιακά φαίνεται ως τελεστής-φόντο για να γεμίσει τον χώρο των γονιδίων 'με φρέσκο αίμα' και στον εξελικτικό προγραμματισμό είναι ο ένας και μοναδικός τελεστής μετάλλαξης που κάνει όλη τη δουλειά της αναζήτησης [8].

### **Ανασυνδιασμός γονιδίων**

Ένας δυαδικός τελεστής μετάλλαξης ονομάζεται ανασυνδιασμός ή διασταύρωση. Καθώς τα ονόματα ποικίλουν, ένας τέτοιος τελεστής ενώνει την πληροφορία από τους γονότυπους των δύο γονέων σε ένα ή δύο γονότυπους των απογόνων. Όμοια με τη μετάλλαξη, ο ανασυνδιασμός γονιδίων είναι ένας στοχαστικός τελεστής: η επιλογή των κομματιών από κάθε γονέα που θα συνδιαστούν και ο τρόπος που αυτά τα κομμάτια συνδιάζονται, εξαρτώνται από τυχαία τραβήγματα. Ξανά, ο ρόλος του ανασυνδιασμού γονιδίων είναι διαφορετικός στις διαλέκτους του εξελικτικού υπολογισμού: στο γενετικό προγραμματισμό είναι συχνά ο μόνος τελεστής μετάλλαξης, ενώ στους γενετικούς αλγορίθμους είναι ο κύριος τελεστής αναζήτησης και στον εξελικτικό προγραμματισμό δεν χρησιμοποιείται ποτέ. Οι μεταβλητές ανασυνδιασμού με ένα υψηλό αριθμό παραμέτρων οι οποίοι παίρνονται ως όρισμα (χρησιμοποιώντας παραπάνω από δύο γονείς) είναι μαθηματικά πιθανό και εύκολο να εφαρμοστούν, αλλά δεν έχουν βιολογικά ισοδύναμα. Ίσως γι' αυτό δεν χρησιμοποιούνται συχνά, αν και αρκετές υποψήφιας μελέτες έχουν θετικά αποτελέσματα στην εξέλιξη.

Η αρχή πίσω από τον ανασυνδιασμό είναι απλή, ταιριάζοντας δύο άτομα με διαφορετικά αλλά επιθυμητά χαρακτηριστικά, μπορούμε να παράγουμε έναν απόγονο που συνδιάζει και τα δύο χαρακτηριστικά.

### **1.1.7 Μηχανισμός επιλογής επιβίωσης (αντικατάσταση)**

Ο ρόλος της **επιλογής επιβίωσης ή περιβαλλοντικής επιλογής** είναι να ξεχωρίσει τα άτομα που βασίζονται στη ποιότητα τους. Υπό την έννοια ότι είναι παρόμοιο με την επιλογή γονέων, αλλά χρησιμοποιείται σε ένα διαφορετικό στάδιο του εξελικτικού κύκλου. Ο μηχανισμός επιλογής επιβίωσης καλείται εφόσον έχει δημιουργηθεί ο απόγονος των επιλεγμένων γονέων. Όπως αναφέρθηκε και παραπάνω, στον εξελικτικό υπολογισμό το μέγεθος του πληθυσμού είναι σχεδόν πάντα σταθερό και επιπλέον μία επιλογή πρέπει να γίνει για το ποια άτομα θα επιτραπούν στην επόμενη γενιά. Αυτή η απόφαση συνήθως βασίζεται στις τιμές καταλληλότητάς τους, ευνοώντας αυτές με την υψηλότερη ποιότητα, αν και συχνά χρησιμοποιείται το σκεπτικό της ηλικίας. Αντίθετα με την επιλογή γονέων, που είναι τυπικά στοχαστική, η επιλογή επιβίωσης είναι συχνά ντετερμινιστική, για παράδειγμα, πρώτιστα συνενώνονται τα σετ των γονέων και των απογόνων και επιλέγεται το κορυφαίο τμήμα (που βασίζεται στη καταλληλότητα), ή επιλέγοντας μόνο από τους απογόνους (που βασίζονται στη ηλικία).

Η επιλογή επιβίωσης επίσης συχνά ονομάζεται και αντικατάσταση ή στρατηγική αντικατάστασης. Σε πολλές περιπτώσεις οι δύο όροι μπορούν να χρησιμοποιηθούν εναλλακτικά.



**Μέρος II**

**Κατηγορίες Εξελικτικών  
Αλγορίθμων**



---

## Γενετικοί Αλγόριθμοι (Genetic Algorithms)

---

### 2.1 Σκοπός του Κεφαλαίου

Σε αυτό το κεφάλαιο θα περιγράψουμε τους πιο γνωστούς τύπους των εξελικτικών αλγορίθμων: τους γενετικούς αλγόριθμους. Μετά την παρουσίαση ενός απλού παραδείγματος για την εισαγωγή των βασικών ιδεών, θα ξεκινήσουμε με το ποια είναι συνήθως η πιο κρίσιμη απόφαση σε μια οποιαδήποτε εφαρμογή, δηλαδή πως θα παρουσιαστεί καλύτερα μια υποψήφια λύση του αλγορίθμου. Θα παρουσιαστούν τέσσερις πιθανές λύσεις, δηλαδή, τέσσερις εκτενώς χρησιμοποιούμενες αναπαραστάσεις. Έπειτα ακολουθώντας τα παραπάνω θα περιγραφεί η ποικιλία των τελεστών (μετάλλαξη και γενετική διασταύρωση) που είναι ιδανικοί για διαφορετικούς τύπους αναπαραστάσης, προτού στρέψουμε την προσοχή μας στην επιλογή και στους μηχανισμούς αντικατάστασης που συνήθως διαχειρίζονται τους πληθυσμούς των λύσεων.

Όπως θα δούμε από αυτή τη χρήση, δεν υπάρχει μόνο ένας καθορισμένος γενετικός αλγόριθμος, αντιθέτως εμείς δημιουργούμε αλγορίθμους από μια ακολουθία τελεστών για να ικανοποιήσουμε προσεκτικά τις εφαρμογές. Αυτό το κεφάλαιο είναι αναγκαία σχετικά μεγάλο εξαιτίας της αφθονίας των πιθανών αναπαραστάσεων, της ποικιλίας και της επιλογής τελεστών. Συγκεκριμένα είναι μεγαλύτερο από τα κεφάλαια που ακολουθούν σε άλλες παραλλαγές εξελικτικών αλγορίθμων. Αυτή η διαφορά στο μέγεθος δεν έχει την τάση να υποδηλώσει ότι μία μέθοδος είναι περισσότερο ή λιγότερο σημαντική από κάποια άλλη, μερικά από αυτά συνέβησαν για να αντιμετωπιστούν πρώτα οι γενετικοί αλγόριθμοι και έτσι πολλά θέματα που είναι σχετικά με άλλα παραδείγματα βρίσκονται για πρώτη φορά και περιγράφονται εδώ. Τελικά, το κεφάλαιο αυτό καταλήγει με ένα παράδειγμα εφαρμογής, δείχνοντας περισσότερες λεπτομέρειες του μονοπατιού από ένα πρόβλημα σε μία εξελικτική λύση του προβλήματος.

### 2.2 Παράδειγμα εισαγωγής

Ο γενετικός αλγόριθμος αρχικά ξεκίνησε από τον Holland καθώς μελετούσε προσαρμοσμένες συμπεριφορές, όπως προτείνεται από τον τίτλο του βιβλίου του στο οποίο συσσωρεύεται μαζί η πρόωρη έρευνά του: *Προσαρμογή σε φυσικά και τεχνητά συστήματα*. Ωστόσο, έχει συζητηθεί εκτενώς ως μια μέθοδος βελτιστοποίησης συναρτήσεων και ξεκινάει σκιαγραφώντας την εφαρμογή από τί θα συζητηθεί πιθανά σε έναν κλασσικό γενετικό αλγόριθμο. Αυτό έχει μια δυαδική αναπαραστάση, επιλογή αναλογικής καταλληλότητας, μια χαμηλή πιθανότητα μετάλλαξης και μια έμφαση σε ένα γενετικά εμνευσμένο ανασυνδιασμό γονιδίων δίνοντας σημασία σε γενετικά νέες υποψήφιες λύσεις. Αυτό αναφέρεται ως 'απλός ΓΑ' ή ως 'κανονικός ΓΑ' και αυτό συνοψίζεται στον πίνακα 2.1.

Για να απεικονιστεί αυτό, θα δειχθούν οι λεπτομέρειες από ένα κύκλο επιλογής - αναπαραγωγής σε ένα απλό πρόβλημα (συνεπώς ανιχνεύσιμο) κατά τον Goldberg [11], δηλαδή αυτό που μεγιστοποιεί τις μεταβλητές του  $x^2$  για κάθε  $x$  από 0 – 31. Χρησιμοποιώντας έναν απλό 5-bit δυαδικό αριθμό κωδικοποίησης, ο πίνακας 2.2 δείχνει έναν τυχαίο αρχικό πληθ-

Αναπαράσταση	ακολουθίες bit
Ανασυνδιασμός	διασταύρωση ενός σημείου
Μετάλλαξη	αναστροφή bit
Επιλογή γονέων	αναλογική καταλληλότητα
Επιλογή επιβίωσης	γενιές

**Πίνακας 2.1:** Πίνακας απλού ΓΑ

θυσμό τεσσάρων γονότυπων, τους αντίστοιχους φαινότυπους και τις τιμές καταλληλότητάς τους. Η στήλη  $Prob_i$  δείχνει την πιθανότητα ενός ατόμου  $i \in (1, 2, 3, 4)$  που έχει επιλεγεί να είναι ένας γονιός, ο οποίος για την καταλληλότητα είναι  $Prob_i = f_i / \sum f_j$ . Ο αριθμός των γονέων είναι ο ίδιος με το μέγεθος του πληθυσμού, επομένως ο αναμενόμενος αριθμός αντίγραφων από κάθε πληθυσμό μετά την επιλογή είναι  $f_i / \bar{f}$ . Όπως μπορούμε να δούμε, αυτοί οι αριθμοί δεν είναι ακέραιοι, αντιθέτως αναπαριστούν μία πιθανότητα κατανομής και η πιθανότητα ζευγαρώματος δημιουργείται παίρνοντας αριθμούς με τυχαίες επιλογές για να δειγματοστεί από αυτή την κατανομή. Η στήλη 'πραγματική μέτρηση' αντιπροσωπεύει τον αριθμό των αντίγραφων στη πιθανότητα ζευγαρώματος, π.χ. δείχνει ένα πιθανό αποτέλεσμα.

Τα επιλεγμένα άτομα συνδιάζονται τυχαία και για κάθε ζευγάρι έχει επιλεγεί ένα τυχαίο σημείο κατά μήκος της ακολουθίας. Ο πίνακας 2.3 δείχνει τα αποτελέσματα της διασταύρωσης στη δοθείσα πιθανότητα ζευγαρώματος για τα σημεία διασταύρωσης μετά το τέταρτο και δεύτερο γονίδιο, αντίστοιχα, μαζί με τις αντίστοιχες τιμές καταλληλότητας.

Στον απλό ΓΑ η μετάλλαξη λειτουργεί παράγοντας ένα τυχαίο αριθμό (από μία ομοιόμορφη κατανομή από το εύρος  $[0, 1]$ ) σε κάθε θέση bit και συγκρίνοντάς τον με μία σταθερή χαμηλή μεταβλητή (π.χ. 0.001), που συνήθως ονομάζεται *ρυθμός μετάλλαξης*. Αν ο τυχαίος αριθμός είναι μεταξύ αυτού του ρυθμού, η μεταβλητή του γονιδίου στην αντίστοιχη θέση αντιστρέφεται. Στο παράδειγμα μας έχουμε  $4 * 5 = 20$  γονίδια και ο πίνακας 2.4 δείχνει τα αποτελέσματα της μετάλλαξης όταν η πρώτη και η δέκατη όγδοση μεταβλητή στην ακολουθία των τυχαίων αριθμών είναι μεταξύ της πιθανότητας της μετάλλαξης. Σε αυτή την περίπτωση, οι μεταλλάξεις δείχνουν ότι μπορεί να προξενήσουν θετικές αλλαγές στην καταλληλότητα, αλλά θα πρέπει να δώσουμε έμφαση ότι σε επόμενες γενιές η μετάλλαξη μπορεί να είναι επιβλαβής. Αυτό το παράδειγμα δείχνει μία τυπική εξέλιξη: ο μέσος όρος της καταλληλότητας αυξάνεται από 293 σε 588.5 και η καλύτερη καταλληλότητα του πληθυσμού από 576 σε 729 μετά τη διασταύρωση και την μετάλλαξη.

### 2.3 Αναπαράσταση των ατόμων

Όπως έχει εξηγηθεί, το πρώτο βήμα για να δημιουργηθεί οποιοσδήποτε εξελικτικός αλγόριθμος είναι να αποφασιστεί μια γενετική αναπαράσταση μιας υποψήφιας λύσης του προ-

Ακολουθία	Αρχικός πληθυσμός	Μεταβλητή $x$	Καταλληλότητα $f(x) = x^2$	$Prob_i$	Αναμενόμενη μέτρηση	Πραγματική μέτρηση
1	01101	13	169	0.14	0.58	1
2	11000	24	576	0.49	1.97	2
3	01000	8	64	0.06	0.22	0
4	10011	19	361	0.31	1.23	1
Σύνολο			1170	1.00	4.00	4
Μέσος όρος			293	0.25	1.00	1
Μέγιστο			576	0.49	1.97	2

**Πίνακας 2.2:** Παράδειγμα  $x^2$ , 1:αρχικοποίηση, εκτίμηση και επιλογή γονέα

Ακολουθία	Πιθανότητα ζευγαρώματος	Σημείο διασταύρωσης	Απόγονος μετά την διασταύρωση	Μεταβλητή $x$	Καταλληλότητα $f(x) = x^2$
1	0110 1	4	01100	12	144
2	1100 0	4	11001	25	625
3	11 000	2	11011	27	729
4	10 011	2	10000	16	256
Σύνολο					1754
Μέσος όρος					439
Μέγιστο					729

**Πίνακας 2.3:** Παράδειγμα  $x^2$ , 2:διασταύρωση και αποτίμηση απογόνου

Ακολουθία	Απόγονος μετά την διασταύρωση	Απόγονος μετά την μετάλλαξη	Μετάλλαξη $x^2$	Καταλληλότητα $f(x) = x^2$
1	01100	11100	26	676
2	11001	11001	25	625
3	11011	11011	27	729
4	10000	10100	18	324
Σύνολο				2354
Μέσος όρος				588.5
Μέγιστο				729

**Πίνακας 2.4:** Παράδειγμα  $x^2$ , 3:μετάλλαξη και αποτίμηση απογόνου

βλήματος. Αυτό περιλαμβάνει τον καθορισμό των γονότυπων και τον σχεδιασμό από τους γονότυπους στους φαινότυπους.

Όταν επιλεγεί η αναπαράσταση, είναι σημαντικό να επιλεγεί η 'σωστή' αναπαράσταση για να λυθεί το πρόβλημα. Το να δημιουργηθεί η αναπαράσταση σωστά είναι ένα από τα πιο δύσκολα κομμάτια για τον σχεδιασμό ενός καλού εξελικτικού αλγορίθμου. Συχνά αυτό γίνεται από εξάσκηση και καλή γνώση του πεδίου ορισμού της εφαρμογής. Στα επόμενα κεφάλαια, θα κοιτάξουμε προσεκτικά σε μερικές συχνά χρησιμοποιούμενες αναπαραστάσεις και τους γενετικούς τελεστές που μπορεί να εφαρμόζονται σε αυτές. Είναι πολύ σημαντικό να τονίσουμε ωστόσο, ότι καθώς οι αναπαραστάσεις περιγράφονται εδώ που χρησιμοποιούνται συχνά, μπορεί να μην είναι οι καλύτερες αναπαραστάσεις για την εφαρμογή. Εξίσου, παρόλο που παρουσιάζουμε τις αναπαραστάσεις και τους συνδεδεμένους τελεστές ξεχωριστά, συχνά αποδεικνύεται στην πράξη ότι χρησιμοποιούνται για να αναμειξουμε τις αναπαραστάσεις με ένα πιο φυσικό και κατάλληλο τρόπο περιγραφής και διαχείρισης μιας λύσης από το να προσπαθούμε με δυσκολία να ενσωματώσουμε διαφορετικές απόψεις ενός προβλήματος σε μια κοινή μορφή.

### 2.3.1 Δυαδικές αναπαραστάσεις

Η πρώτη αναπαράσταση που παρατηρούμε είναι μια από τις απλούστερες- αυτή με την δυαδική χρησιμοποιείται στο παράδειγμα παραπάνω. Αυτή είναι μια από τις πρώτες αναπαραστάσεις και ιστορικά πολλοί ΓΑ έχουν λανθασμένα χρησιμοποιηθεί σε αυτή την αναπαράσταση ανεξάρτητα από το πρόβλημα που προσπαθούμε να λύσουμε. Εδώ ο γονότυπος υπάρχει απλά σε μια ακολουθία δυαδικών ψηφίων- μία ακολουθία bit.

Για μία συγκεκριμένη εφαρμογή έχουμε να αποφασίσουμε πόσο μεγάλη θα είναι η ακολουθία και πώς θα μπορέσουμε να την ερμηνεύσουμε για να παραχθεί ένας φαινότυπος. Για την επιλογή του σχεδιασμού γονότυπου-φαινότυπου για ένα ειδικό πρόβλημα, ένα πράγμα πρέπει να είναι σίγουρο ότι η κωδικοποίηση επιτρέπει όλα τα πιθανά bit ακολουθιών να δηλώσουν μία αποτελεσματική λύση στο δοθέν πρόβλημα και όλες οι πιθανές λύσεις μπορούν

να εκπροσωπηθούν.

Για μερικά προβλήματα, ειδικά αυτά που αφορούν τις αποφάσεις για τις μεταβλητές Boolean, ο σχεδιασμός γονότυπου-φαινότυπου είναι φυσικός, αλλά συχνά (όπως στο παράδειγμά μας) οι ακολουθίες bit χρησιμοποιούνται για να κωδικοποιήσουν άλλες μη δυαδικές πληροφορίες. Για παράδειγμα, μπορεί να ερμηνευτεί μία ακολουθία bit μήκους 80 ως δέκα 8-bit ακέραιους, ή πέντε 16-bit πραγματικούς αριθμούς. Συνήθως αυτό είναι λάθος και καλύτερα αποτελέσματα μπορούν να εξασφαλιστούν χρησιμοποιώντας τους ακέραιους ή απευθείας τις πραγματικές-μεταβλητές στις αναπαραστάσεις.

Ένα από τα προβλήματα της κωδικοποίησης αριθμών σε δυαδικούς είναι ότι τα διαφορετικά bits έχουν διαφορετική υπογραφή. Αυτό μπορεί να βοηθηθεί χρησιμοποιώντας κωδικοποίηση Gray, η οποία είναι μια παραλλαγή του τρόπου με τον οποίο οι ακέραιοι αριθμοί σχεδιάζονται σε ακολουθίες bit. Η καθιερωμένη μέθοδος έχει το μειονέκτημα της απόστασης Hamming μεταξύ συνεχόμενων ακεραίων που συχνά δεν είναι ίση με το ένα. Εάν ο στόχος είναι να αναπτυχθεί ένας ακέραιος αριθμός, θα θέλαμε να είχαμε την ευκαιρία της αλλαγής από 7 σε 8 που ισοδυναμεί με την αλλαγή αυτού σε 6. Η ευκαιρία αλλαγής του 0111 σε 1000 από ανεξάρτητες αναστροφές bit δεν είναι η ίδια, ωστόσο, γι' αυτό το λόγο αυτό αλλάζει σε 0110. Η κωδικοποίηση Gray είναι μια αναπαράσταση που εξασφαλίζει ότι συνεχής ακέραιοι πάντα θα έχουν απόσταση Hamming ένα.

### 2.3.2 Αναπαραστάσεις ακεραίων

Καθώς υποδείξαμε προηγουμένως, οι αναπαραστάσεις δυαδικών δεν είναι πάντα οι πιο κατάλληλες εάν το πρόβλημά μας σχεδιάζεται φυσικά, σε μια αναπαράσταση όπου τα διαφορετικά γονίδια μπορούν να έχουν ένα σει μεταβλητών. Ένα προφανές παράδειγμα από το πότε αυτό μπορεί να συμβεί είναι το πρόβλημα εύρεσης καταλληλότερων μεταβλητών για ένα σει μεταβλητών που όλο δέχονται ακέραιες μεταβλητές. Αυτές οι μεταβλητές μπορεί να είναι απεριόριστες (π.χ. οποιαδήποτε ακέραια μεταβλητή επιτρέπεται), ή μπορεί να είναι περιορισμένη σε ένα πεπερασμένο σει: για παράδειγμα, εάν προσπαθούμε να αναπτύξουμε ένα μονοπάτι σε ένα τετράγωνο πλέγμα, τότε μπορεί να περιορίσουμε τις μεταβλητές σε ένα σει (0, 1, 2, 3) αντιπροσωπεύοντας (Βορρά, Ανατολή, Νότο, Δύση). Σε οποιαδήποτε περίπτωση ένας ακέραιος αριθμός κωδικοποίησης είναι πιθανά πιο κατάλληλος από έναν δυαδικό. Όταν σχεδιάζουμε την κωδικοποίηση και τους τελεστές μεταβλητών, αξίζει να σκεφτούμε κατά πόσο υπάρχουν φυσικές σχέσεις μεταξύ των πιθανών μεταβλητών όπου ένα χαρακτηριστικό μπορεί να πάρει. Αυτό μπορεί να είναι προφανές για ταξινομημένα χαρακτηριστικά όπως οι ακέραιοι (το 2 μοιάζει πιο πολύ με το 3 παρά με το 389), αλλά για πρωτεύοντα χαρακτηριστικά όπως τα σημεία πυξίδας παραπάνω, μπορεί να μην έχουν μια φυσική διάταξη.

### 2.3.3 Πραγματικές μεταβλητές ή κινητής υποδιαστολής σημεία αναπαράστασης

Συχνά ο πιο σημαντικός τρόπος για να αναπαραστήσουμε μία υποψήφια λύση σε ένα πρόβλημα είναι να έχουμε μία ακολουθία πραγματικών μεταβλητών. Αυτό αναπτύσσεται όταν οι μεταβλητές που θέλουμε να αναπαραστήσουμε σαν γονίδια προήλθαν από μία συνεχή παρά από μία διακριτή κατανομή. Φυσικά, σε έναν υπολογιστή η ακρίβεια προσέγγισης αυτών των πραγματικών μεταβλητών είναι στην πραγματικότητα περιορισμένη σε σχέση με την υλοποίηση, γι' αυτό το λόγο τις προτιμάμε ως αριθμούς κινητής υποδιαστολής. Ο γονότυπος ως λύση  $k$  γονιδίων είναι τώρα ένα διάνυσμα  $\langle x_1, \dots, x_k \rangle$  με  $x_i \in \mathbb{R}$ .

### 2.3.4 Αναπαραστάσεις αντιμετάθεσης

Πολλά προβλήματα παίρνουν την μορφή της απόφασης με την σειρά όπου μία ακολουθία γεγονότων πρέπει να αναπτύσσεται. Καθώς άλλες μορφές αναπτύσσονται (για παράδειγμα



συναρτήσεις αποκωδικοποίησης βασισμένες σε απεριόριστες αναπαραστάσεις ακεραίων [23] ή 'μεταβλητά κλειδιά' βασισμένα σε αναπαραστάσεις πραγματικών ακεραίων [21], η πιο φυσική αναπαράσταση αυτών των προβλημάτων είναι η αντιμετάθεση ενός σετ ακεραίων. Μία άμεση συνέπεια είναι ότι καθώς μία ταξινομημένη ακολουθία ΓΑ επιτρέπει τους αριθμούς να αναπτύσσονται παραπάνω από μία φορά, τέτοιες ακολουθίες ακεραίων δεν θα αναπαριστούν αποτελεσματικές αντιμεταθέσεις. Είναι ξεκάθαρο πως χρειαζόμαστε νέα μεταβολή τελεστών για να διατηρήσουμε σωστά την αντιμετάθεση όπου κάθε πιθανή αλληλουχία μεταβλητών αναπτύσσεται ακριβώς μία φορά στην λύση.

Όταν επιλέγουμε κατάλληλους τελεστές μεταβλητών αξίζει επίσης να κατανοήσουμε ότι υπάρχουν πραγματικά δύο κλάσεις προβλημάτων όπου αναπαριστώνται από αντιμεταθέσεις. Στη πρώτη περίπτωση, η σειρά με την οποία τα γεγονότα συμβαίνουν είναι σημαντική. Αυτό μπορεί να συμβαίνει όταν τα γεγονότα χρησιμοποιούν περιορισμένους πόρους ή χρόνο και ένα τυπικό παράδειγμα αυτού του είδους προβλημάτων είναι το πρόβλημα 'προγραμματισμός εργασίας' που περιγράφεται παρακάτω. Για παράδειγμα, μπορεί να είναι καλύτερο για τον παράγοντα 1 να παραχθεί πριν από τους παράγοντες 2 και 3, οι οποίοι με την σειρά τους μπορεί να είναι προτιμότερο να παραχθούν από τον παράγοντα 4, άσχετα με το πόσο μακριά στην εξέλιξη μπορεί να προκύψει. Σε αυτήν την περίπτωση μπορεί εύκολα να υποθεί ότι οι ακολουθίες [1, 2, 3, 4] και [1, 3, 2, 4] έχουν όμοια καταλληλότητα και είναι πολύ καλύτερες για παράδειγμα από το [4, 3, 2, 1].

Ένας εναλλακτικός τύπος για ταξινομημένα προβλήματα εξαρτάται από την *γεινίαση* και συμβολίζεται με το πρόβλημα του πλανώδιου πωλητή. Το πρόβλημα είναι να βρεθεί μια ολοκληρωμένη περιοδεία από  $n$  δοσμένες πόλεις ελαχίστου μήκους. Ο χώρος αναζήτησης για αυτό το πρόβλημα είναι πολύ μεγάλος: υπάρχουν  $(n - 1)!$  διαφορετικές πιθανές διαδρομές για  $n$  δοσμένες πόλεις (για την ασύμμετρη περίπτωση που η μέτρηση γίνεται και πίσω και εμπρός όπως οι δύο διαδρομές). Για  $n = 30$  υπάρχουν περίπου  $10^{32}$  διαφορετικές διαδρομές. Προσδιορίζουμε τις πόλεις 1, 2, ...,  $n$ . Μία ολοκληρωμένη διαδρομή είναι μία αντιμετάθεση των πόλεων, έτσι ώστε για  $n = 4$ , οι διαδρομές [1, 2, 3, 4] και [3, 4, 2, 1] να είναι και οι δύο αποτελεσματικές. Η διαφορά από τα ταξινομημένα προβλήματα μπορεί να φανεί ξεκάθαρα εάν σκεφτούμε ότι το σημείο έναρξης της διαδρομής δεν είναι σημαντικό και επιπλέον τα [1, 2, 3, 4], [2, 3, 4, 1], [3, 4, 1, 2] και [4, 1, 2, 3] είναι όλα ισοδύναμα. Πολλά παραδείγματα αυτού του είδους είναι επίσης συμμετρικά, έτσι ώστε το [4, 3, 2, 1] και τα υπόλοιπα να είναι επίσης ισοδύναμα.

Τελικά, πρέπει να αναφέρουμε ότι υπάρχουν δύο πιθανοί τρόποι για να κωδικοποιήσουμε μία αντιμετάθεση. Για τον πρώτο από αυτούς (τον πιο συχνά χρησιμοποιούμενο) το  $i$ -οστό στοιχείο της αναπαράστασης δηλώνει το τι συμβαίνει σε αυτόν τον χώρο της ακολουθίας (ή ο  $i$ -οστός προορισμός που επισκέφτηκε). Στον δεύτερο, η μεταβλητή του  $i$ -οστού στοιχείου δηλώνει την θέση της ακολουθίας στην οποία συμβαίνει το  $i$ -οστό γεγονός. Επιπλέον για τις τέσσερις πόλεις [A,B,C,D] και την αντιμετάθεση [3, 1, 2, 4], η πρώτη κωδικοποίηση δηλώνει την διαδρομή [C,A,B,D] και η δεύτερη την διαδρομή [B,C,A,D].

## 2.4 Μετάλλαξη

Μετάλλαξη είναι το γενικό όνομα που δόθηκε στους τελεστές μεταβλητών που χρησιμοποιούν μόνο ένα γονέα και δημιουργούν ένα παιδί εφαρμόζοντας κάποιο είδος τυχαίας επιλογής για την αναπαράσταση (γονότυπου). Η μορφή που παίρνει εξαρτάται από την επιλογή της κωδικοποίησης που χρησιμοποιείται, καθώς αυτό σημαίνει τις παραμέτρους που συνδέονται, η οποία συχνά αναφέρεται ως ρυθμός μετάλλαξης. Στις λεπτομέρειες παρακάτω συγκεντρωνόμαστε στην επιλογή των τελεστών αντί των παραμέτρων. Ωστόσο, το τελευταίο μπορεί να έχει σημαντική διαφορά στη συμπεριφορά του γενετικού αλγορίθμου.

### 2.4.1 Μετάλλαξη για δυαδικές αναπαράστασεις

Αν και μερικά άλλα σχέδια έχουν χρησιμοποιηθεί περιστασιακά, ο πιο γνωστός τελεστής μετάλλαξης που χρησιμοποιείται για δυαδική κωδικοποίηση μελετά κάθε γονίδιο χωριστά και επιτρέπει κάθε bit να αναστρέφεται (π.χ. από 1 σε 0 ή από 0 σε 1) με μία μικρή πιθανότητα  $p_m$ . Ο πραγματικός αριθμός των μεταβλητών που αλλάζουν δεν καθορίζεται, αλλά εξαρτάται από την ακολουθία των τυχαίων αριθμών που σχεδιάζονται, έτσι ώστε για την κωδικοποίηση μήκους  $L$  να έχουμε μέσο όρο  $L * p_m$  μεταβλητών που θα αλλάξουν. Στο σχήμα 2.1 αυτό διευκρινίζεται για την περίπτωση όπου η τρίτη, τέταρτη και όγδοη τυχαία μεταβλητή που παράγονται είναι λιγότερο λογικός ο ρυθμός μετάλλαξης από  $p_m$ .



**Σχήμα 2.1:** Λογική μετάλλαξη για δυαδικές κωδικοποιήσεις

Ένας αριθμός ερευνών και αναπαραστάσεων έχουν επιλεγεί από τις κατάλληλες μεταβλητές του λογικού ρυθμού μετάλλαξης και αξίζει να σημειώσουμε ότι στην έναρξη η πιο κατάλληλη επιλογή που είναι να χρησιμοποιηθεί εξαρτάται από το επιθυμητό αποτέλεσμα. Για παράδειγμα, όντως η εφαρμογή απαιτεί έναν πληθυσμό στον οποίο όλα τα μέλη έχουν υψηλή καταλληλότητα, ή απλά βρέθηκε αυτή η μία υψηλή κατάσταση ατόμων; Ωστόσο, οι περισσότεροι δυαδικοί κωδικοποιούν ΓΑ που χρησιμοποιούν ρυθμό μετάλλαξης από μία ποικιλία τέτοια ώστε ο μέσος όρος μεταξύ ενός γονιδίου ανά γενιά και ενός γονιδίου ανά απόγονο να μεταλλάσσεται.

### 2.4.2 Τελεστές μετάλλαξης για ακέραιες αναπαραστάσεις

Για ακέραιες κωδικοποιήσεις υπάρχουν δύο βασικές μορφές μετάλλαξης που χρησιμοποιούνται, όπου και οι δύο μεταλλάσσουν κάθε γονίδιο ξεχωριστά με καθορισμένο από τον χρήστη πιθανότητα  $p_m$ .

#### Τυχαία επανατοποθέτηση

Εδώ το 'bit που αντιστρέφεται' στη μετάλλαξη της δυαδικής κωδικοποίησης επεκτείνεται στη 'τυχαία επανατοποθέτηση', ούτως ώστε με την πιθανότητα  $p_m$  μία νέα μεταβλητή να επιλέγεται τυχαία από το σει των επιτρεπόμενων μεταβλητών σε κάθε θέση. Αυτός είναι ο πιο κατάλληλος τελεστής για να χρησιμοποιηθεί όταν τα γονίδια κωδικοποιούνται για τα βασικά χαρακτηριστικά, από τότε που όλες οι άλλες μεταβλητές γονιδίων είναι ισοδύναμα κατάλληλες για να επιλεγούν.

#### Αλλαγή μετάλλαξης

Αυτό το σχήμα σχεδιάστηκε για ταξινομημένα χαρακτηριστικά και λειτουργεί προσθέτοντας μία μικρή μεταβλητή (θετική ή αρνητική) για κάθε γονίδιο με πιθανότητα  $p$ . Συνήθως αυτές οι μεταβλητές δειγματίζονται τυχαία για κάθε θέση, από μία κατανομή όπου είναι συμμετρική περίπου στο μηδέν και είναι πιο κατάλληλες για να παράγουν μικρές αλλαγές από ότι μεγάλες. Θα πρέπει να σημειωθεί ότι η αλλαγή μετάλλαξης απαιτεί έναν αριθμό παραμέτρων που ελέγχουν την κατανομή από την οποία τα τυχαία νούμερα έχουν σχεδιαστεί και ως εκ τούτου το μέγεθος των βημάτων που η μετάλλαξη λαμβάνει μέρος στο χώρο αναζήτησης. Η εύρεση των κατάλληλων σει για αυτές τις παραμέτρους μπορεί να μην είναι εύκολη και μερικές φορές είναι κοινό να χρησιμοποιηθεί παραπάνω από ένας τελεστής μετάλλαξης σε μία σειρά από προβλήματα που βασίζονται σε ακέραιους. Εναλλακτικά, η τυχαία επανατοποθέτηση μπορεί να χρησιμοποιηθεί με χαμηλή πιθανότητα, σε συνδυασμό με έναν τελεστή αλλαγής ο οποίος τείνει να κάνει μικρές αλλαγές που σχετίζονται με το πεδίο των επιτρεπόμενων μεταβλητών.

### 2.4.3 Τελεστές μετάλλαξης για κινητής υποδιαστολής αναπαραστάσεις

Για κινητής υποδιαστολής αναπαραστάσεις, είναι φυσικό να αγνοήσουμε την διακριτοποίηση που επιβάλλεται από το υλικό και να μελετήσουμε τις αλληλουχίες των μεταβλητών οι οποίες προέρχονται από μία συνεχή αντί από μία διακριτή κατανομή, έτσι οι μορφές της μετάλλαξης που περιγράφονται παραπάνω δεν είναι πλέον εφαρμόσιμες. Αντιθέτως είναι κοινό να αλλάξουν οι αλληλουχίες των μεταβλητών για κάθε γονίδιο τυχαία εντός του πεδίου ορισμού που δίνεται από ένα κάτω όριο  $L_i$  και ένα άνω  $U_i$ , απορρέοντας την επακόλουθη μετατροπή:  $\langle x_1, \dots, x_n \rangle \rightarrow \langle x'_1, \dots, x'_n \rangle$  όπου τα  $x_i, x'_i \in [L_i, U_i]$ . Δύο τύπους μπορούμε να διακρίνουμε σύμφωνα με την πιθανότητα κατανομής από την οποία οι μεταβλητές των νέων γονιδίων σχεδιάστηκαν: την ομοιόμορφη και την ανομοιόμορφη μετάλλαξη.

#### Ομοιόμορφη μετάλλαξη

Γι' αυτούς τους τελεστές οι μεταβλητές  $x'_i$  έχουν σχεδιαστεί ομοιόμορφα τυχαία από το  $[L_i, U_i]$ . Αυτή είναι η πιο ακριβής άποψη, ανάλογα με τα bit που αντιστρέφονται για τις δυαδικές κωδικοποιήσεις και τις τυχαίες επανατοποθετήσεις που σχεδιάστηκαν παραπάνω για τις ακέραιες κωδικοποιήσεις. Κανονικά χρησιμοποιείται με μία πιθανότητα μετάλλαξης σύμφωνα με την θέση.

#### Ανομοιόμορφη μετάλλαξη με μία σταθερή κατανομή

Ίσως η πιο κοινή μορφή της ανομοιόμορφης μετάλλαξης χρησιμοποιείται με κινητής υποδιαστολής αναπαραστάσεις που παίρνουν μία μορφή ανάλογα με την μετάλλαξη για ακεραίους. Έχει σχεδιαστεί έτσι ώστε συνήθως, αλλά όχι πάντα, το ποσοστό των αλλαγών που έχει εισαχθεί να είναι μικρό. Αυτό επιτυγχάνεται προσθέτοντας στη τωρινή μεταβλητή γονιδίου ένα ποσό που σχεδιάστηκε τυχαία από την γκαουσιανή κατανομή με αριθμητικό μέσο το μηδέν και έπειτα περικόπτουμε την μεταβλητή του αποτελέσματος στο εύρος  $[L_i, U_i]$  εάν είναι απαραίτητο. Η γκαουσιανή (ή κανονική) κατανομή έχει την ιδιότητα ότι περίπου τα  $2/3$  των δειγμάτων που σχεδιάστηκαν παρέμειναν με μία σταθερή απόκλιση. Αυτό σημαίνει ότι οι περισσότερες από τις αλλαγές που γίνονται θα είναι μικρές, αλλά υπάρχει μη μηδενική πιθανότητα παραγωγής μεγάλων αλλαγών καθώς η ουρά της κατανομής ποτέ δεν φτάνει το μηδέν. Είναι μία συνήθης πρακτική να εφαρμόζεται αυτός ο τελεστής με πιθανότητα ένα για κάθε γονίδιο και αντί της παραμέτρου της μετάλλαξης να χρησιμοποιείται για τον έλεγχο της τυπικής απόκλισης της γκαουσιανής και γι' αυτό το λόγο καταγράφεται η πιθανότητα κατανομής των μεγεθών του βήματος.

Μία εναλλακτική στην γκαουσιανή κατανομή είναι η χρήση της κατανομής Cauchy, η οποία έχει πιο 'παχιά' ουρά. Δηλαδή, οι πιθανότητες παραγωγής μεγαλύτερων μεταβλητών είναι λίγο πιο ψηλά από της γκαουσιανής κατανομής με την ίδια τυπική απόκλιση.

### 2.4.4 Τελεστές μετάλλαξης για αναπαραστάσεις αντιμετάθεσης

Για αναπαραστάσεις αντιμετάθεσης, δεν είναι πια λογικό να μελετάμε κάθε γονίδιο ανεξάρτητα, αντιθέτως η εύρεση 'νόμιμων' μεταλλάξεων είναι ένα θέμα μετακίνησης αλληλουχιών γύρω από ένα γονιδίωμα. Αυτό έχει την άμμεση συνέπεια ότι η παράμετρος μετάλλαξης ερμηνεύεται ως η πιθανότητα της ακολουθίας που υφίσταται μετάλλαξη, και όχι ότι ένα γονίδιο της ακολουθίας μεταβάλλεται. Οι τρεις πιο κοινές μορφές μετάλλαξης χρησιμοποιούνται για ταξινομημένα προβλήματα που περιγράφηκαν πρώτα. Ενώ οι τρεις πρώτοι τελεστές παρακάτω, (συγκεκριμένα η μετάλλαξη εισαγωγής) λειτουργούν δημιουργώντας μικρές αλλαγές στη σειρά όπου υπάρχουν αλληλουχίες μεταβλητών [9], για τα προβλήματα που βασίζονται στη γεινίαση αυτό μπορεί να προκαλέσει τεράστιους αριθμούς συνδέσεων και έτσι η αντιστροφή χρησιμοποιείται πιο συχνά.

#### Μετάλλαξη αντιμετάθεσης

Αυτός ο τελεστής λειτουργεί διαλέγοντας τυχαία δύο θέσεις (γονίδια) της ακολουθίας και αντιμεταθέτει τις αλληλουχίες μεταβλητών. Αυτό υλοποιείται στο σχήμα 2.2 όπου οι μεταβλητές των θέσεων δύο και πέντε έχουν αντιμετατεθεί.



**Σχήμα 2.2:** Μετάλλαξη αντιμετάθεσης

### Μετάλλαξη Εισαγωγής

Αυτός ο τελεστής λειτουργεί διαλέγοντας δύο αλληλουχίες τυχαία και μετακινώντας τη μία έτσι ώστε να είναι δίπλα στην άλλη, ανακατεύοντας τις άλλες κατά μήκος για να κάνουν χώρο. Αυτό υλοποιείται στο σχήμα 2.3 όπου οι μεταβλητές δύο και πέντε έχουν αλλάξει.



**Σχήμα 2.3:** Μετάλλαξη εισαγωγής

### Μετάλλαξη ανάμειξης

Εδώ σε ολόκληρη την ακολουθία, ή σε κάποιο τυχαίο υποσύνολο μεταβλητών που έχει επιλεγεί από το εσωτερικό της, έχουν ανακατευτεί οι θέσεις. Αυτό υλοποιείται στο σχήμα 2.4, όπου οι μεταβλητές από δύο έως πέντε έχουν επιλεγεί.



**Σχήμα 2.4:** Μετάλλαξη ανάμειξης

### Μετάλλαξη αναστροφής

Η μετάλλαξη αναστροφής λειτουργεί επιλέγοντας τυχαία δύο θέσεις στην ακολουθία και αντιστρέφοντας την σειρά στην οποία οι μεταβλητές εμφανίζονται μεταξύ αυτών των θέσεων. Αποτελεσματικά διαχωρίζεται η ακολουθία σε τρία κομμάτια με όλες τις συνδέσεις εσωτερικά του ενός κομματιού να διατηρούνται και μόνο οι δύο συνδέσεις μεταξύ των κομματιών να σπάνε. Η τυχαία αναστροφή της υπο-ακολουθίας είναι η μικρότερη αλλαγή που μπορεί να γίνει για τα προβλήματα που βασίζονται στη γειτνίαση [14]. Η προτεραιότητα του χώρου αναζήτησης προτρέπει αυτόν τον τελεστή και συνεπώς τις μορφές με μία φυσική βάση για να μελετηθεί αυτό το είδος προβλημάτων, ισοδύναμα με τον χώρο Hamming για τις αναπαραστάσεις των δυαδικών προβλημάτων. Είναι η βασική κίνηση πίσω από την  $2 - opt$  ευριστική αναζήτηση για το πρόβλημα του πλανόδιου πωλητή και κατ' επέκταση  $k - opt$ . Αυτός ο τελεστής υλοποιείται στο σχήμα 2.5, όπου η υπο-ακολουθία μεταξύ των θέσεων δύο και πέντε έχει επιλεγεί για αναστροφή.

## 2.5 Ανασυνδιασμός

Ο ανασυνδιασμός είναι μία διαδικασία μέσω της οποίας μία νέα υποψήφια λύση έχει δημιουργηθεί από την πληροφορία που συγκροτείται εντός των δύο (ή περισσότερων) γονικών λύσεων και είναι το ένα από τα πιο σημαντικά χαρακτηριστικά στους γενετικούς αλγόριθμους. Μεγάλη δραστηριότητα αναζήτησης εστιάστηκε στον πρωταρχικό μηχανισμό για την δημιουργία ποικιλίας, με την μετάλλαξη να μελετάται ως υπόβαθρο τελεστή αναζήτησης.

Αν και η λέξη ανασυνδιασμός προήλθε επειδή χρησιμοποιήθηκε για τις περισσότερες γενικές περιπτώσεις, νωρίς οι συγγραφείς χρησιμοποιήσαν τον όρο διασταύρωση (επίσης από την βιολογική αναλογία της μείωσης) αν και ο όρος διασταύρωση τείνει να προτιμάται στις πιο πολλές περιπτώσεις των δύο γονέων. Οι τελεστές του ανασυνδιασμού συνήθως εφαρμόζονται πιθανοκρατικά σύμφωνα με το ρυθμό διασταύρωσης  $p_c$ , ο οποίος είναι τυπικά στο εύρος  $[0.5, 1.0]$ . Συνήθως δύο γονείς έχουν επιλεγεί και έπειτα μια τυχαία μεταβλητή σχεδιάζεται από  $[0, 1]$  και συγκρίνεται με το  $p_c$ . Εάν η μεταβλητή είναι πιο μικρή, δύο απόγονοι



**Σχήμα 2.5:** Μετάλλαξη αναστροφής

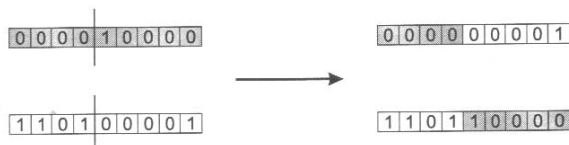
δημιουργούνται μέσω του ανασυνδιασμού των δύο γονέων, διαφορετικά δημιουργούνται χωρίς κάποιον αναπαραγωγικό τρόπο, π.χ. αντιγράφοντας τους γονείς. Το τελικό αποτέλεσμα είναι ότι γενικά το σετ των αποτελεσμάτων των απογόνων αποτελείται από κάποιες αντιγραφές των γονέων και άλλων ατόμων που αναπαριστούσαν νέες λύσεις. Συνεπώς, σε σύγκριση με την πιθανότητα μετάλλαξης  $p_m$ , η οποία ελέγχει πώς τα κομμάτια από τα χρωμοσώματα διαταράσσονται ανεξάρτητα, η πιθανότητα διασταύρωσης καθορίζει την ευκαιρία όπου ένα επιλεγμένο ζευγάρι γονέων υφίσταται αυτόν τον τελεστή.

### 2.5.1 Τελεστές ανασυνδιασμού για δυαδικές αναπαραστάσεις

Τρεις σταθερές μορφές ανασυνδιασμού γενικά χρησιμοποιούνται για δυαδικές αναπαραστάσεις. Όλα ξεκινάνε από τους δύο γονείς και δημιουργούνται δύο παιδιά, αν και όλα αυτά έχουν επεκταθεί στη πιο γενική περίπτωση όπου ένας αριθμός γονέων μπορεί να χρησιμοποιηθεί [20] και υπάρχουν επίσης καταστάσεις στις οποίες μόνο ένας απόγονος μπορεί να μελετηθεί.

#### Διασταύρωση σε ένα σημείο

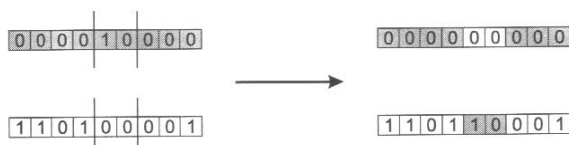
Η διασταύρωση σε ένα σημείο είναι ο αυθεντικός τελεστής ανασυνδιασμού που προτάθηκε [14] και εξετάστηκε [13]. Λειτουργεί διαλέγοντας έναν τυχαίο αριθμό από το εύρος  $[0, l - 1]$  (όπου  $l$  το μήκος της κωδικοποίησης) και μετά διαχωρίζονται και οι δύο γονείς σε αυτό το σημείο και δημιουργούν δύο παιδιά ανταλλάσσοντας τις ουρές (σχήμα 2.6).



**Σχήμα 2.6:** Διασταύρωση ενός σημείου

#### Διασταύρωση σε $N$ -σημεία

Η διασταύρωση σε ένα σημείο μπορεί εύκολα να γενικευτεί σε διασταύρωση  $n$ -σημείων, όπου η αναπαράσταση σπάει σε περισσότερα από δύο κομμάτια του συνεχούς γονιδίου και μετά ο απόγονος δημιουργείται παίρνοντας εναλλακτικά κομμάτια από τους δύο γονείς. Στη πράξη αυτό σημαίνει πως επιλέγονται  $n$  τυχαία σημεία διασταύρωσης από  $[0, l - 1]$ , το οποίο υλοποιείται στο σχήμα 2.7 για  $n = 2$ .



**Σχήμα 2.7:** Διασταύρωση  $n$ -σημείων:  $n = 2$

#### Ομοιόμορφη διασταύρωση

Οι προηγούμενοι δύο τελεστές λειτουργούν διαιρώντας τους γονείς σε έναν αριθμό τμημάτων συνεχούς γονιδίου και συναθροίζοντάς τα ξανά για να δημιουργήσουν απόγονο. Σε αντίθεση με αυτό, η ομοιόμορφη διασταύρωση, (uniform crossover), λειτουργεί εκθέτοντας κάθε γονίδιο ανεξάρτητα και κάνοντας μία τυχαία επιλογή σύμφωνα με την οποία οι γονείς

θα έπρεπε να κληρονομηθούν. Αυτό υλοποιείται παράγοντας μία ακολουθία από  $L$  τυχαίες μεταβλητές από μία ενιαία κατανομή από  $[0, 1]$ . Σε κάθε θέση, εάν μία μεταβλητή είναι κάτω από μία παράμετρο  $p$  (συνήθως 0.5), το γονίδιο κληρονομείται από τον πρώτο γονέα, αλλιώς από τον δεύτερο [16]. Ο δεύτερος απόγονος δημιουργήθηκε χρησιμοποιώντας την αντίστροφη σχεδίαση. Αυτό υλοποιείται στο σχήμα 2.8.



**Σχήμα 2.8:** Ομοιόμορφη διασταύρωση: Σε αυτό το παράδειγμα ο πίνακας  $[0.35, 0.62, 0.18, 0.42, 0.83, 0.76, 0.39, 0.51, 0.36]$  χρησιμοποιήθηκε για να αποφασιστεί η κληρονομικότητα των τυχαίων μεταβλητών που σχεδιάστηκαν ομοιόμορφα από το  $[0, 1]$

Από τα σχήματα 2.6 και 2.7 θα πρέπει αμέσως να φανεί ότι ενώ προτείνεται η έλλειψη της προηγούμενης πληροφορίας, ο ανασυνδιασμός λειτουργεί με την τυχαία ανάμειξη των κομματιών από τους γονείς, η διασταύρωση  $n$ -σημείων έχει μία κληρονομική προκατάληψη η οποία τείνει να κρατήσει τα γονίδια μαζί, τα οποία τοποθετούνται στην αναπαράσταση κοντά το ένα στο άλλο. Επιπλέον, όταν το  $n$  προστίθεται (π.χ. διασταύρωση ενός σημείου) υπάρχει μία δυνατή προκατάληψη έναντι του να κρατηθούν ενωμένοι οι ανασυνδιασμοί των γονιδίων όπου βρίσκονται στα απέναντι άκρα των αναπαραστάσεων. Αυτά τα αποτελέσματα είναι γνωστά ως προκατάληψη θέσης (positional bias) και έχουν μελετηθεί σε μεγάλο βαθμό και από την θεωρητική και από την πειραματική πλευρά. Σε αντίθεση, η ομοιόμορφη κατανομή δεν παρουσιάζει οποιαδήποτε προκατάληψη θέσης, αλλά έχει μια δυνατή τάση για μεταβίβαση 50% του γονιδίου από κάθε γονέα και για μεταβίβαση ενός απογόνου με γονίδια υψηλής καταλληλότητας από τον ένα γονέα.

Η γενική φύση αυτών των αλγορίθμων κάνουν αδύνατο να δηλώσουμε ότι ένας ή περισσότεροι τελεστές θα αποδώσουν καλύτερα σε οποιοδήποτε πρόβλημα. Ωστόσο, μία κατανόηση για τους τύπους των προκαταλήψεων παρουσιάζεται με διαφορετικούς τελεστές ανασυνδιασμού που μπορεί να είναι ανεκτίμητοι όταν σχεδιάζεται ένας αλγόριθμος για ένα συγκεκριμένο πρόβλημα, ειδικά εάν υπάρχουν γνωστά πρότυπα ή εξαρτήσεις στην επιλεγμένη αναπαράσταση που μπορεί να επιτευχθεί.

### 2.5.2 Τελεστές ανασυνδιασμού για ακέραιες αναπαραστάσεις

Για αναπαραστάσεις όπου κάθε γονίδιο έχει έναν υψηλό αριθμό πιθανών μεταβλητών αλληλουχίας (όπως οι ακέραιοι) είναι φυσικό να χρησιμοποιηθεί το ίδιο σετ τελεστών για δυαδικές αναπαραστάσεις. Αυτό συμβαίνει επειδή συνήθως δεν κάνει καμία αίσθηση να μελετηθούν 'αναμειγμένες' αλληλουχίες μεταβλητών αυτού του τρόπου. Για παράδειγμα, ακόμη και αν τα γονίδια αναπαριστούν ακέραιες μεταβλητές, ο μέσος όρος ενός οριζόντιου και ενός πρόσθετου ακέραιου αποδίδει ένα μη ολοκληρωμένο αποτέλεσμα.

### 2.5.3 Τελεστές ανασυνδιασμού για κινητής υποδιαστολής αναπαραστάσεις

Έχουμε δύο δυνατότητες για ανασυνδιασμό δύο ακολουθιών με σημεία κινητής υποδιαστολής:

- Χρησιμοποιώντας έναν αναλογικό τελεστή από αυτούς που χρησιμοποιούνται για ακολουθίες bit, αλλά τώρα διαχωρίζεται μεταξύ των υποδιαστολών. Με άλλα λόγια, μία αλληλουχία είναι μία μεταβλητή σε ένα σημείο κινητής υποδιαστολής αντί ενός bit. Αυτό έχει το μειονέκτημα (από κοινού με όλους τους τελεστές ανασυνδιασμού που περιγράφηκαν παραπάνω) ότι μόνο η μετάλλαξη μπορεί να εισάγει νέες μεταβλητές

σε έναν πληθυσμό, καθώς ο ανασυνδιασμός μας δίνει μόνο νέους συνδιασμούς των υποδιαστολών που υπάρχουν. Οι τελεστές ανασυνδιασμού αυτού του τύπου για αναπαράστασεις κινητής υποδιαστολής είναι γνωστοί ως διακριτοί ανασυνδιασμοί και έχει την ιδιότητα ότι εάν δημιουργούμε έναν απόγονο  $z$  από γονείς  $x$  και  $y$ , έπειτα η μεταβλητή αλληλουχίας για το γονίδιο  $i$  δίνεται από  $z_i = x_i$  ή  $y_i$  με ίση πιθανότητα.

- Χρησιμοποιώντας έναν τελεστή όπου σε κάθε θέση γονιδίου δημιουργεί μία νέα αλληλουχία μεταβλητής στον απόγονο που βρίσκεται μεταξύ αυτών των γονέων. Χρησιμοποιώντας την ορολογία παραπάνω, έχουμε  $z_i = ax_i + (1 - a)y_i$  για μερικά  $a$  στο  $[0, 1]$ . Με αυτό τον τρόπο, ο ανασυνδιασμός είναι τώρα ικανός για να δημιουργήσει νέο κατάλληλο γονίδιο, αλλά έχει το μειονέκτημα ότι ως αποτέλεσμα της διαδικασίας του μέσου όρου ο ρυθμός της αλληλουχίας μεταβλητών του πληθυσμού του κάθε γονιδίου ελαττώνεται. Τελεστές αυτού του τύπου είναι γνωστοί ως *ενδιάμεσος ή αριθμητικός ανασυνδιασμός*.

### Αριθμητικός ανασυνδιασμός

Τρεις τύποι αριθμητικού ανασυνδιασμού περιγράφονται. Σε όλους αυτούς, η επιλογή της παραμέτρου  $a$  φτιάχνεται μερικές φορές τυχαία, αλλά στην πράξη συνήθως να χρησιμοποιούμε μία σταθερή μεταβλητή, συχνά 0.5 (στην οποία περίπτωση έχουμε ομοιόμορφο αριθμητικό ανασυνδιασμό).

#### Απλός ανασυνδιασμός

Πρώτη επιλογή ένα σημείο ανασυνδιασμού  $k$ . Έπειτα, για το πρώτο παιδί, παίρνουμε τα πρώτα  $k$  σημεία του γονέα 1 και τα τοποθετούμε στο παιδί. Το υπόλοιπο είναι ο αριθμητικός μέσος των γονέων 1 και 2:

Παιδί 1:  $\langle x_1, \dots, x_k, a \cdot y_{k+1} + (1 - a) \cdot x_{k+1}, \dots, a \cdot y_n + (1 - a) \cdot x_n \rangle$ .

Το παιδί 2 είναι ανάλογο με τα  $x$  και  $y$  να είναι αντίστροφα. (σχήμα 2.9)

#### Μονός αριθμητικός ανασυνδιασμός

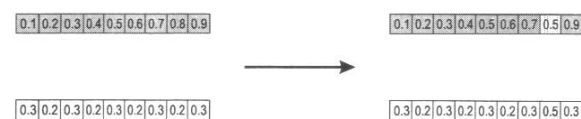
Διαλέγουμε μία τυχαία αλληλουχία  $k$ . Σε αυτή την θέση, παίρνουμε τον αριθμητικό μέσο των δύο γονέων. Τα άλλα σημεία είναι τα σημεία από τους γονείς, π.χ.:

Παιδί 1:  $\langle x_1, \dots, x_{k-1}, a \cdot y_k + (1 - a) \cdot x_k, x_{k+1}, \dots, x_n \rangle$ .

Το δεύτερο παιδί δημιουργείται με τον ίδιο τρόπο με τα  $x$  και  $y$  να είναι αντίστροφα. (σχήμα 2.10)



Σχήμα 2.9: Απλός αριθμητικός ανασυνδιασμός:  $k = 8, a = 1/2$



Σχήμα 2.10: Μονός αριθμητικός ανασυνδιασμός:  $k = 3, a = 1/2$

### Ολόκληρος αριθμητικός ανασυνδιασμός

Αυτός είναι ο πιο συχνά χρησιμοποιούμενος τελεστής και λειτουργεί παίρνοντας το σταθμικό άθροισμα των δύο γονικών αλληλουχιών για κάθε γονίδιο, π.χ.:

$$\text{Παιδί 1} = a \cdot \bar{x} + (1 - a) \cdot \bar{y},$$

$$\text{Παιδί 2} = a \cdot \bar{y} + (1 - a) \cdot \bar{x}$$

Αυτό υλοποιείται στο σχήμα 2.11. Όπως δείχνει το παράδειγμα, εάν  $a = 1/2$  οι δύο απόγονοι θα είναι πανομοιότυποι για αυτόν τον τελεστή.



**Σχήμα 2.11:** Ολόκληρος αριθμητικός ανασυνδιασμός:  $a = 1/2$

#### 2.5.4 Τελεστές ανασυνδιασμού για μετάλλαξη αναπαραστάσεων

Με την πρώτη ματιά, οι αναπαραστάσεις που βασίζονται στην μετάλλαξη παρουσιάζουν συγκεκριμένες δυσκολίες για τον σχεδιασμό των τελεστών ανασυνδιασμού, καθώς δεν είναι γενικά πιθανό να γίνει μετάλλαξη των υπο-ακολουθιών μεταξύ των γονέων και να διατηρείται ακόμη η ιδιότητα της μετάλλαξης. Ωστόσο, αυτή η κατάσταση είναι καταπραϋντική όταν μελετούμε τι είναι αυτό που πραγματικά οι λύσεις αναπαριστούν, π.χ., είτε μία σειρά στην οποία βρίσκονται τα στοιχεία, είτε ένα σετ κινήσεων που συνδέονται με τα ζευγάρια των στοιχείων. Ένας αριθμός ειδικευμένων τελεστών αναπαραστάσεως έχει σχεδιαστεί για μεταλλάξεις, οι οποίες έχουν σκοπό στη μετάδοση να βρεθούν οι περισσότερες πληροφορίες από τις οποίες συγκροτούνται οι γονείς, ειδικά αυτές που επιδρούν κοινά [1]. Θα πρέπει να εστιάσουμε εδώ στην περιγραφή δύο πιο γνωστών και κοινά χρησιμοποιούμενων τελεστών για κάθε υποκλάση των προβλημάτων μετάλλαξης.

##### Μερικώς χαρτογραφημένη διασταύρωση

Η μερικώς χαρτογραφημένη διασταύρωση προτάθηκε πρώτα από τους Goldberg και Linde, ως τελεστής ανασυνδιασμού για το πρόβλημα του πλανόδιου πωλητή και έχει γίνει ένας από τους ευρέως χρησιμοποιούμενους τελεστές για τα προβλήματα γεινίασης [12]. Κατά την διάρκεια των χρόνων πολλές αμελητέες μεταβλητές καθορισμού της μερικώς χαρτογραφημένης διασταύρωσης βρέθηκαν σε έντυπο υλικό, εδώ εμείς θα χρησιμοποιήσουμε του Whitley το οποίο λειτουργεί όπως τα ακόλουθα (σχήματα 2.12-2.14).

1. Διαλέγουμε δύο τυχαία σημεία διασταύρωσης και αντιγράφουμε το σημείο μεταξύ αυτών από τον πρώτο γονέα  $P1$  στον πρώτο απόγονο.
2. Ξεκινώντας από το πρώτο σημείο διασταύρωσης ψάχνουμε για στοιχεία σε εκείνο το σημείο του δεύτερου γονέα  $P2$  που δεν έχουν αντιγραφεί.
3. Για καθένα από αυτά (ας πούμε  $i$ ), ψάχνουμε στον απόγονο να δούμε ποια στοιχεία (ας πούμε  $j$ ) έχουν αντιγραφεί στο δικό του μέρος από τον  $P1$ .
4. Το μέρος των  $i$  καταλαμβάνει την θέση των  $j$  στον  $P2$ , καθώς γνωρίζουμε ότι δεν θα τοποθετούσαμε το  $j$  εκεί (εφόσον το έχουμε ήδη στην ακολουθία μας).
5. Εάν το μέρος καταλαμβάνεται από  $j$  ο  $P2$  έχει ήδη γεμίσει τον απόγονο με ένα στοιχείο  $k$  και τοποθετεί το  $i$  στην θέση που καταλαμβάνει το  $k$  στον  $P2$ .
6. Αντιμετωπίζοντας τα στοιχεία από το σημείο διασταύρωσης, το υπόλοιπο του απόγονου μπορεί να συμπληρωθεί από τον  $P2$  και το δεύτερο παιδί δημιουργήθηκε αναλογικά, με τους γονικούς ρόλους να έχουν αντιστραφεί.



Επιθεωρώντας τον απόγονο που δημιουργήθηκε φαίνεται ότι σε αυτή την περίπτωση οι έξι από τις εννέα συνδέσεις που παρουσιάζονται στον απόγονο παρουσιάζονται μία ή περισσότερες φορές στους γονείς. Ωστόσο, από τις δύο κοινές άκρες 5 – 6 και 7 – 8 και στους δύο γονείς, μόνο η πρώτη παρουσιάζεται στον απόγονο. Ο Radcliffe προτείνει ότι η επιθυμητή ιδιότητα οποιουδήποτε τελεστή ανασυνδιασμού είναι αυτή του *σεβασμού*, για παράδειγμα, οποιαδήποτε μεταφέρεται μία πληροφορία από τους δύο γονείς πρέπει επίσης να παρουσιάζεται και στον απόγονο. Μία στιγμή αναδίπλωσης μας υποδεικνύει ότι αυτό είναι αληθές για όλους τους τελεστές ανασυνδιασμού που περιγράφηκαν παραπάνω για δυαδικές και ακέραιες αναπαραστάσεις, για διακριτό ανασυνδιασμό και για αναπαραστάσεις κινητής υποδιαστολής, αλλά όπως το παράδειγμα παραπάνω δείχνει, δεν είναι απαραίτητα αληθές η μερικώς χαρτογραφημένη διασταύρωση. Έχοντας αυτό το θέμα στο μυαλό μας, πολλοί άλλοι τελεστές έχουν σχεδιαστεί για προβλήματα που βασίζονται στις μεταλλάξεις γειννίας, όπου περιγράφονται καλύτερα παρακάτω.



**Σχήμα 2.12:** ΜΧΔ, βήμα 1: αντιγράφονται τα τυχαίως επιλεγμένα σημεία από τον πρώτο γονέα στον απόγονο



**Σχήμα 2.13:** ΜΧΔ, βήμα 2: εξετάζει με τη σειρά του την τοποθέτηση των στοιχείων που παρουσιάζονται στο μεσαίο τμήμα του γονέα 2 και όχι του γονέα 1. Η θέση όπου το 8 έχει στον  $P2$  καταλαμβάνεται από το 4 στον απόγονο, έτσι ώστε να μπορούμε να τοποθετήσουμε το 8 στη θέση που εκκενώθηκε από το 4 στον  $P2$ . Η θέση του 2 στον  $P2$  καταλαμβάνεται από το 5 στον απόγονο, έτσι κοιτάμε πρώτα την θέση που κατέχει το 5 στον  $P2$ , όπου είναι η θέση 7. Αυτή είναι ήδη κατειλημμένη από την μεταβλητή 7, έτσι ώστε να δούμε το πού θα πάει στον  $P2$  και τελικά να βρεθεί ένα άνοιγμα στον απόγονο το οποίο είναι κενό-το τρίτο. Τέλος, να σημειωθεί ότι οι μεταβλητές 6 και 5 καταλαμβάνονται στα μεσαία τμήματα και των δύο γονέων.



**Σχήμα 2.14:** ΜΧΔ, βήμα 3: αντιγράφονται τα υπόλοιπα στοιχεία από τον δεύτερο γονέα στις ίδιες θέσεις στον απόγονο

### Διασταύρωση στην άκρη

Η διασταύρωση στην άκρη βασίζεται στην ιδέα ότι ένας απόγονος πρέπει να δημιουργείται, όσο είναι δυνατό, χρησιμοποιώντας μόνο άκρες που παρουσιάζονται σε έναν ή περισσότερους γονείς. Έχει υποβληθεί σε έναν αριθμό διορθώσεων κατά το πέρασμα των χρόνων. Εδώ θα περιγράψουμε την πιο συχνά χρησιμοποιούμενη εκδοχή: διασταύρωση στην άκρη-3 κατά Whitley, όπου σχεδιάστηκε για να εξασφαλιστεί ότι οι συχνές άκρες διατηρούνται.

Προκειμένου να επιτευχθεί αυτό κατασκευάστηκε ένας πίνακας άκρων (γνωστός επίσης ως λίστες γειτνίασης), όπου για κάθε στοιχείο, καταγράφονται τα άλλα στοιχεία που είναι συνδεδεμένα σε αυτό στους δύο γονείς. Ένα '+' στον πίνακα υποδεικνύει ότι η άκρη παρουσιάζεται και στους δύο γονείς. Ο τελεστής λειτουργεί όπως παρακάτω :

1. Κατασκευάζουμε τον πίνακα άκρων
2. Διαλέγουμε ένα αρχικό στοιχείο τυχαία και το τοποθετούμε στον απόγονο
3. Αναθέτουμε την μεταβλητή τωρινό\_στοιχείο=είσοδος
4. Απαλείφουμε όλες τις αναφορές του τωρινού\_στοιχείου από τον πίνακα
5. Εξετάζουμε την λίστα για το τωρινό\_στοιχείο
  - Εάν υπάρχει μία συχνή άκρη, διαλέγουμε αυτή να είναι το επόμενο στοιχείο
  - Αλλιώς διαλέγουμε την είσοδο της λίστας η οποία από μόνη της έχει την μικρότερη λίστα
  - Τα δομικά στοιχεία διαχωρίζονται τυχαία
6. Στην περίπτωση αναζήτησης μίας κενής λίστας, το άλλο άκρο του απόγονου εξετάζεται για επέκταση, αλλιώς ένα νέο στοιχείο επιλέγεται τυχαία

Προφανώς μόνο στην τελευταία περίπτωση θα αποκαλέσουμε ξένες άκρες (foreign edges), αυτές που θα εισαχθούν. Ο ανασυνδιασμός στην άκρη-3 υλοποιείται από το παρακάτω παράδειγμα όπου οι γονείς χρησιμοποιούν τις δύο ίδιες μεταλλάξεις όπως το παράδειγμα της μερικώς χαρτογραφημένης διασταύρωσης [123456789] και [937826514], δίνοντας τον πίνακα των άκρων που φαίνεται στον πίνακα 2.5 και η κατασκευή υλοποιείται στον πίνακα 2.6. Σημειώνουμε ότι μόνο ένα παιδί ανά ανασυνδιασμό δημιουργείται από αυτόν τον τελεστή.

#### Σειρά διασταύρωσης

Ο τελεστής της σειράς διασταύρωσης σχεδιάστηκε από τον Davis για προβλήματα που βασίζονται στη σειρά μετάλλαξης. Ξεκινά με έναν αντίστοιχο τρόπο με τη μερικώς χαρτογραφημένη διασταύρωση, αντιγράφοντας ένα τυχαία επιλεγμένο σημείο του πρώτου γονέα μέσα στον απόγονο. Ωστόσο, αυτό προέρχεται διαφορετικά επειδή ο σκοπός είναι να μεταδοθεί η πληροφορία σχετικά με τη *συγγενική σειρά* από τον δεύτερο γονέα.

1. Επιλέγουμε τυχαία δύο σημεία διασταύρωσης και αντιγράφουμε το σημείο μεταξύ αυτών από τον πρώτο γονέα  $P1$  στον απόγονο.
2. Ξεκινώντας από το δεύτερο σημείο διασταύρωσης στον δεύτερο γονέα, αντιγράφουμε τους υπολοίπους μη χρησιμοποιούμενους αριθμούς στο πρώτο παιδί με τη σειρά όπου αυτοί εμφανίζονται στον δεύτερο γονέα, περιβάλλοντας το τέλος κάθε λίστας.
3. Δημιουργούμε τον δεύτερο απόγονο με ανάλογο τρόπο, με τους αντίστροφους ρόλους γονέων.

Στοιχείο	Άκρες	Στοιχείο	Άκρες
1	2, 5, 4, 9	6	2, 5+, 7
2	1, 3, 6, 8	7	3, 6, 8+
3	2, 4, 7, 9	8	2, 7+, 9
4	1, 3, 5, 9	9	1, 3, 4, 8
5	1, 4, 6+		

**Πίνακας 2.5:** Διασταύρωση στην άκρη: παράδειγμα με πίνακα άκρων

Επιλογές	Στοιχείο που επιλέχθηκε	Λόγος	Μερικό αποτέλεσμα
Όλα	1	Τυχαία	[1]
2, 5, 4, 9	5	Μικρότερη λίστα	[15]
4, 6	6	Κοινή άκρη	[156]
2, 7	2	Τυχαία επιλογή (και οι δύο έχουν δύο αντικείμενα στη λίστα)	[1562]
3, 8	8	Μικρότερη λίστα	[15628]
7, 9	7	Κοινή άκρη	[156287]
3	3	Μοναδικό αντικείμενο στη λίστα	[1562873]
4, 9	9	Τυχαία επιλογή	[15628739]
4	4	Τελευταίο στοιχείο	[156287394]

**Πίνακας 2.6:** Διασταύρωση στην άκρη: παράδειγμα της κατασκευής μετάλλαξης



**Σχήμα 2.15:** Σειρά διασταύρωσης, βήμα 1: αντιγράφονται τα τυχαία επιλεγμένα τμήματα από τον πρώτο γονέα στον απόγονο



**Σχήμα 2.16:** Σειρά διασταύρωσης, βήμα 2: αντιγράφονται τα υπόλοιπα των αλληλουχιών με τη σειρά που εμφανίζονται στο δεύτερο γονέα, επεξεργάζοντας την ακολουθία ως σπειροειδή

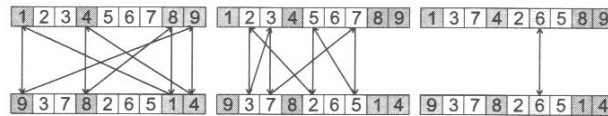
Αυτό υλοποιείται στα σχήματα 2.15 και 2.16.

### Κυκλική διασταύρωση

Ο τελευταίος τελεστής που θα μελετήσουμε σε αυτό το κεφάλαιο είναι η κυκλική διασταύρωση, η οποία μπλέκει διατηρώντας όσο το δυνατό περισσότερο πληροφορία, σχετικά με την απόλυτη θέση που βρίσκονται τα στοιχεία. Ο τελεστής λειτουργεί διαιρώντας τα στοιχεία σε *κύκλους*. Ο κύκλος είναι ένα υποσύνολο των στοιχείων ο οποίος έχει την ιδιότητα ότι κάθε στοιχείο πάντα είναι ζευγάρι με ένα άλλο στοιχείο του ίδιου κύκλου όταν οι δύο γονείς είναι ευθυγραμμισμένοι. Διαιρώντας τις μεταλλάξεις σε κύκλους, ο απόγονος δημιουργείται επιλέγοντας εναλλασσόμενους κύκλους από κάθε γονέα. Η διαδικασία για την κατασκευή κύκλων είναι η ακόλουθη:

1. Ξεκινάμε με την πρώτη μη χρησιμοποιημένη θέση και την αλληλουχία του  $P1$
2. Κοιτάμε την αλληλουχία στην ίδια θέση στον  $P2$
3. Πάμε στην θέση με την ίδια αλληλουχία στον  $P1$
4. Προσθέτουμε αυτή την αλληλουχία στον κύκλο
5. Επαναλαμβάνουμε τα βήματα 2 έως 4 μέχρι να φτάσουμε στην πρώτη αλληλουχία του  $P1$

Η ολοκληρωμένη εφαρμογή του τελεστή υλοποιείται από ένα παράδειγμα στα σχήματα 2.17 και 2.18.



**Σχήμα 2.17:** Κυκλική διασταύρωση, βήμα 1: αναγνώριση των κύκλων



**Σχήμα 2.18:** Κυκλική διασταύρωση, βήμα 2: κατασκευή του απογόνου

### 2.5.5 Ανασυνδιασμός πολλαπλών γονέων

Κοιτώντας την ποικιλία των τελεστών, από την προοπτική του αριθμού των παραμέτρων  $\alpha$  που δέχεται, κάνει μία άμεση ιδέα να πάει πέρα από τα συνηθισμένα  $\alpha = 1$  (μετάλλαξη) και  $\alpha = 2$  (διασταύρωση). Οι αποτελεσματικοί τελεστές του ανασυνδιασμού πολλαπλών γονέων για  $\alpha = 3, 4, \dots$  είναι απλό να καθοριστούν και να υλοποιηθούν. Αυτό εξασφαλίζει την ευκαιρία να πειραματιστούμε με εξελικτικές διαδικασίες που χρησιμοποιούν αντίγραφο σχεδίου που δεν υπάρχει στην βιολογία. Από τεχνικής πλευράς προσφέρει ένα εργαλείο που ενισχύει τα αποτελέσματα του ανασυνδιασμού. Αν και τέτοιοι τελεστές δεν χρησιμοποιούνται ευρέως στον εξελικτικό προγραμματισμό, υπάρχουν πολλά παραδείγματα που έχουν προταθεί κατά τη διάρκεια της ανάπτυξης του πεδίου, ακόμη από τις αρχές του 1966. Αυτοί οι τελεστές μπορούν να κατηγοριοποιηθούν από τον βασικό μηχανισμό που χρησιμοποιεί τον συνδιασμό της πληροφορίας για τα άτομα [1]. Αυτός ο μηχανισμός μπορεί να είναι:

- Βασιζόμαστε στις συχνότητες αλληλουχιών, π.χ., η αναγνώριση του p-sexual γενικεύει ομοιόμορφες διασταύρωσεις
- Βασιζόμαστε στην κατάτμηση και τον ανασυνδιασμό γονέων, π.χ. τη διαγώνια διασταύρωση, η οποία γενικεύει την διασταύρωση  $n$ -σημείων
- Βασιζόμαστε στους αριθμητικούς τελεστές σε αλληλουχίες με πραγματικές μεταβλητές, π.χ., το κέντρο της συνολικής διασταύρωσης το οποίο γενικεύει τους τελεστές του αριθμητικού ανασυνδιασμού

Γενικά, δεν μπορεί να απαιτηθεί η αύξηση του αριθμού των παραμέτρων του ανασυνδιασμού να έχει μία θετική επίδραση στην επίδοση των ΕΑ διότι αυτό εξαρτάται πάρα πολύ από τον τύπο του ανασυνδιασμού και το πρόβλημα.

## 2.6 Μοντέλα πληθυσμού

Μέχρι τώρα στη συζήτησή μας για τους γενετικούς αλγορίθμους, έχουμε επικεντρωθεί στον τρόπο με τον οποίο δυναμικές λύσεις έχουν παρουσιαστεί να δώσουν πληθυσμό για διάφορα άτομα και με τον τρόπο που λειτουργούν οι τελεστές παραλλαγής (ανασυνδιασμού και μετάλλαξης) σε αυτά τα άτομα για να παράγουν απόγονο. Αυτοί οι απόγονοι γενικά θα κληρονομήσουν κάποιες ιδιότητες από τους γονείς τους αλλά επίσης και κάποιες μικρές διαφορές από αυτούς, εξασφαλίζοντας την αποτίμηση νέων δυναμικών λύσεων. Τώρα θα στρέψουμε την προσοχή μας στο δεύτερο πιο σημαντικό στοιχείο της εξελικτικής διαδικασίας-όπου οι διαφορικές επιβίωσης (differential survival) των ατόμων για να ολοκληρωθούν οι πόροι και να συμμετέχουν και στην αναπαραγωγή, βασίζονται στην σχετική καταλληλότητα.

Δύο διαφορετικά μοντέλα ΓΑ διακρίνονται στην βιβλιογραφία: το *γενικό μοντέλο* και το *σταθερής κατάστασης μοντέλο*. Το γενικό μοντέλο περιγράφεται στο παράδειγμα παραπάνω (κεφάλαιο 2.2). Σε κάθε γενιά ξεκινάμε με έναν πληθυσμό μεγέθους  $\mu$ , από τον οποίο επιλέγεται μία πιθανότητα ζευγαρώματος από  $\mu$  γονείς. Ο επόμενος  $\lambda (= \mu)$  απόγονος δημιουργείται από την πιθανότητα ζευγαρώματος από την εφαρμογή των τελεστών παραλλαγής, και αποτιμάται. Μετά από κάθε γενιά, ολόκληρος ο πληθυσμός αντικαθίσταται από τον απόγονό του, ο οποίος αποκαλείται 'επόμενη γενιά'.

Στο σταθερής κατάστασης μοντέλο, ολόκληρος ο πληθυσμός δεν αλλάζεται ταυτόχρονα, αλλά κατά κάποιο τρόπο αλλάζουν κομμάτια από αυτό. Σε αυτή την περίπτωση,  $\lambda (< \mu)$  παλιά άτομα αντικαθίσταται από  $\lambda$  νέα. Η ποσοστιαία αναλογία του πληθυσμού που αντικαθίσταται ονομάζεται *κενό γενιάς* (generational gap) και είναι ίσο με  $\lambda/\mu$ . Εφόσον η εισαγωγή του αλγορίθμου GENITOR, στο σταθερής κατάστασης μοντέλο έχει μελετηθεί και εφαρμοστεί ευρέως συνήθως με  $\lambda = 1$  και ένα αντίστοιχο κενό παραγωγής  $1/\mu$ .

Σε αυτό το σημείο αξίζει να σημειωθεί ότι οι τελεστές, επιλογή και αντικατάσταση, που είναι υπεύθυνοι για αυτό το ανταγωνιστικό στοιχείο της διαχείρισης του πληθυσμού, λειτουργούν στη βάση της καταλληλότητας των ατόμων (που αποτιμήθηκαν ή εκτιμήθηκαν). Αυτό σημαίνει ότι η έμφασή μας αλλάζει τώρα κατεύθυνση από τις μερικές λύσεις στις ολικές. Μία άμεση συνέπεια αυτού είναι ότι αυτοί οι τελεστές λειτουργούν ανεξάρτητα από την αναπαράσταση του προβλήματος που επιλέχθηκε.

Υπάρχουν δύο σημεία στον εξελικτικό κύκλο στον οποίο ο ανταγωνισμός βασίζεται στην καταλληλότητα που μπορεί να βρισκεται: κατά τη διάρκεια της επιλογής για να συμμετέχει στην αντιστοίχιση και κατά τη διάρκεια της επιλογής των ατόμων για να επιβιώσει στην επόμενη γενιά. Ξεκινάμε περιγράφοντας τις πιο συχνά χρησιμοποιούμενες μεθόδους για την επιλογή γονέα, αλλά να σημειώσουμε ότι τα περισσότερα από αυτά μπορούν να εφαρμοστούν κατά τη διάρκεια της φάσης της επιλογής επιβίωσης [11]. Να σημειωθεί ότι προσπαθούμε να μεγιστοποιήσουμε την καταλληλότητα και ότι οι τιμές καταλληλότητας δεν είναι αρνητικές. Συχνά προβλήματα που ελαχιστοποιούνται εκφράζονται με ορολογία μίας αντικειμενικής

συνάρτησης και μερικές φορές βρίσκονται αρνητικές τιμές καταλληλότητας. Ωστόσο, σε όλες τις περιπτώσεις αυτό μπορεί να σχεδιαστεί στην επιθυμητή μορφή χρησιμοποιώντας ένα κατάλληλο μετασχηματισμό.

## 2.7 Επιλογή γονέων

### 2.7.1 Επιλογή αναλογικής καταλληλότητας

Οι αρχές της επιλογής αναλογικής καταλληλότητας περιγράφονται στο απλό παράδειγμα στο κεφάλαιο 2.2. Ανακαλώντας αυτό για κάθε επιλογή, η πιθανότητα όπου ένα άτομο  $f_i$  επιλέχθηκε για αντιστοίχιση είναι  $f_i / \sum_{j=1}^{\mu} f_j$ . Με άλλα λόγια η πιθανότητα επιλογής εξαρτάται από την τιμή της απόλυτης καταλληλότητας του ατόμου που συγκρίνεται με τις τιμές της απόλυτης καταλληλότητας του υπόλοιπου πληθυσμού.

Αυτός ο μηχανισμός επιλογής εισήχθη από τον Holland. Ωστόσο, έχει αναγνωρισθεί ότι υπάρχουν μερικά προβλήματα με αυτόν το μηχανισμό:

- Διακεκριμένα άτομα, π.χ. άτομα που είναι πολύ καλύτερα από τα υπόλοιπα, αναλαμβάνουν ολόκληρο τον πληθυσμό πολύ γρήγορα. Αυτό είναι γνωστό ως πρόωρη σύγκλιση (premature convergence).
- Όταν οι τιμές καταλληλότητας είναι όλες πολύ κοντά μεταξύ τους, δεν υπάρχει σχεδόν καμία πίεση επιλογής, καθώς τα κομμάτια της ρουλέτας (roulette wheel) που προσδιορίζουν τα άτομα είναι περίπου (περισσότερο ή λιγότερο) το ίδιο μέγεθος, έτσι η επιλογή είναι σχεδόν τυχαία ομοιόμορφη και έχοντας μία λίγο καλύτερη καταλληλότητα δεν είναι πολύ 'χρήσιμη' σε ένα άτομο. Επομένως, όταν υπάρχει κάποια σύγκλιση και τα χειρότερα άτομα έχουν φύγει, η απόδοση αυξάνεται πολύ σιγά.
- Ο μηχανισμός συμπεριφέρεται διαφορετικά σε μετατοπισμένες εκδοχές της ίδιας συνάρτησης καταλληλότητας.

Αυτό το τελευταίο σημείο υλοποιείται σε ένα σχήμα, όπου δείχνει τις αλλαγές στις πιθανότητες επιλογής για τρία σημεία που ανέρχονται όταν μία τυχαία συνάρτηση καταλληλότητας  $y = f(x)$  μετατοπίζεται προσθέτοντας 10.0 σε όλες τις τιμές καταλληλότητας. Όπως φαίνεται το επιλεγμένο πλεονέκτημα του καταλληλότερου σημείου (B) μειώνεται.

Για να αποφύγουμε το δεύτερο, δύο προβλήματα με επιλογή αναλογικής καταλληλότητας, μία διαδικασία γνωστή ως windowing χρησιμοποιείται συχνά. Υπό αυτό το σχέδιο, οι διαφορικές καταλληλότητας διατηρούνται αποσπώντας από την ακατέργαστη καταλληλότητα  $f(x)$  μία μεταβλητή  $\beta^t$ , η οποία εξαρτάται κατά κάποιο τρόπο από την πρόσφατη αναζήτηση ιστορίας. Η πιο απλή προσέγγιση είναι απλά να θέσουμε  $\beta = \min_{y \in P^t} f(y)$ , π.χ., για να αποσπάσουμε την μεταβλητή από τα λιγότερο κατάλληλα μέλη του τωρινού πληθυσμού  $P^t$ . Αυτή η μεταβλητή μπορεί να διακυμανθεί γρήγορα, έτσι ώστε μία εναλλακτική είναι να χρησιμοποιήσουμε έναν τρέχοντα μέσο όρο μέσω των τελευταίων γενεών.

Μία άλλη ευρέως γνωστή προσέγγιση είναι η *σίγμα αναλογία* του Goldberg's (Goldberg's sigma scaling), η οποία ενσωματώνει την πληροφορία σχετικά με τον αριθμητικό μέσο  $f$  και την τυπική απόκλιση  $\sigma_f$  της καταλληλότητας στον πληθυσμό:

$$f'(x) = \max(f(x) - (\bar{f} - c \cdot \sigma_f), 0.0)$$

όπου  $c$  μία σταθερή μεταβλητή, που συνήθως τοποθετείται στο 2.

Στον πίνακα 2.7 δείχνουμε ένα παράδειγμα του πώς οι πιθανότητες επιλογής διαφέρουν για ένα πληθυσμό τριών διαφορετικών ατόμων με αναλογική καταλληλότητα και την επιλογή που βασίζεται στη σειρά με διαφορετικές μεταβλητές του  $s$ .

Όταν ένας γραμμικός σχεδιασμός χρησιμοποιείται από τη σειρά για την επιλογή πιθανοτήτων, το ποσοστό της πίεσης της επιλογής όπου μπορεί να εφαρμοστεί είναι περιορισμένο. Αυτό προέρχεται από την υπόθεση ότι, κατά μέσο όρο, ένα άτομο μεσαίας καταλληλότητας

	Καταλληλότητα	Σειρά	$P_{selFP}$	$P_{selLR}(s = 2)$	$P_{selLR}(s = 1.5)$
A	1	1	0.1	0	0.167
B	5	2	0.5	0.67	0.5
Σ	4	2	0.4	0.33	0.33
Σύνολο	10		1.0	1.0	1.0

**Πίνακας 2.7:** Αναλογική καταλληλότητα (FP) έναντι επιλογής γραμμικής σειράς (LR)

θα πρέπει να έχει μία ευκαιρία να αναπαραχθεί, όπου διαδοχικά επιβάλλει μία μέγιστη μεταβλητή  $s = 2.0$ . Εάν απαιτείται μία υψηλότερη πίεση επιλογής, π.χ. περισσότερη έμφαση στην επιλογή ατόμων των παραπάνω, από τον μέσο όρο καταλληλότητας, ένα εκθετικής θέσης σχέδιο χρησιμοποιείται συχνά, το οποίο είναι της μορφής:

$$P_{\text{exp-rank}}(i) = \frac{1-e^{-i}}{c}$$

### 2.7.2 Υλοποίηση των πιθανοτήτων επιλογής

Παραπάνω περιγράψαμε δύο εναλλακτικά σχέδια για να αποφασιστεί μία πιθανότητα κατανομής όπου καθορίζει την πιθανότητα κάθε ατόμων στον πληθυσμό που επιλέχθηκε για αναπαραγωγή. Σε έναν ιδανικό κόσμο, η πιθανότητα ζευγαρώματος των γονέων συμμετέχει στον ανασυνδιασμό όπου θα έχει ακριβώς τις ίδιες αναλογίες με αυτές της κατανομής της επιλογής πιθανοτήτων. Ωστόσο, στην πράξη αυτό είναι αδύνατο επειδή στο πεπερασμένο μέγεθος του πληθυσμού, (π.χ. όταν πολλαπλασιάζουμε την πιθανότητα επιλογής από τον συνολικό αριθμό των γονέων στη πιθανότητα ζευγαρώματος), τυπικά βρίσκουμε ότι τα άτομα έχουν έναν αναμενόμενο αριθμό αντίγραφων ο οποίος δεν είναι ακέραιος. Με άλλα λόγια, η πιθανότητα ζευγαρώματος των γονέων δειγματίζεται από την κατανομή πιθανότητας επιλογής, αλλά γενικά δεν απεικονίζεται επακριβώς, όπως είδαμε στο παράδειγμα στο κεφάλαιο 2.2.

Ο πιο απλός τρόπος για να πετύχουμε αυτό το δείγμα είναι γνωστός ως τον αλγόριθμο της ρουλέτας (roulette wheel). Θεμελιωδώς αυτό είναι το ίδιο καθώς γυρίζουμε μία φορά την ρουλέτα, όπου τα μεγέθη των οπών απεικονίζουν την επιλογή πιθανοτήτων. Εάν υποθέσουμε ότι ο αλγόριθμος εφαρμόζεται στα επιλεγμένα  $\lambda$  μέλη από το σετ των  $\mu$  γονέων σε μία πιθανότητα ζευγαρώματος, συνήθως υλοποιείται όπως παρακάτω. Υποθέτοντας κάποια σειρά στον πληθυσμό (με προτεραιότητα ή τυχαία) από το 1 μέχρι το  $\mu$ , υπολογίζουμε μία λίστα μεταβλητών  $[\alpha_1, \alpha_2, \dots, \alpha_\mu]$  έτσι ώστε  $\alpha_i = \sum_1^i P_{sel}(i)$ , όπου  $P_{sel}(i)$  καθορίζεται από την κατανομή επιλογής - αναλογικά με την καταλληλότητα ή με την προτεραιότητα. Να σημειώσουμε ότι αυτό σημαίνει  $a_m = 1.0$ . Τα κύρια σημεία του αλγορίθμου δίνονται στο σχήμα 2.19.

Παρά την απλότητα του, έχει αναγνωρισθεί ότι ο αλγόριθμος της ρουλέτας δεν δίνει στην πραγματικότητα ένα συγκεκριμένο καλό δείγμα της απαιτούμενης κατανομής. Κάθε φορά που περισσότερα από ένα δείγματα είναι να σχεδιασθούν από την κατανομή, προτιμάται η χρήση του αλγορίθμου του γενικού στοχαστικού δείγματος (stochastic universal sampling). Αυτό είναι ισοδύναμο με το να γυρίσει μία φορά η ρουλέτα με  $\mu$  ίσα διαστήματα, αντί  $\mu$  περιστροφές με ένα διάστημα και υπολογίζεται η λίστα των αθροιστικών πιθανοτήτων επιλογής  $[a_1, a_2, \dots, a_m]$  όπως περιγράφεται στο σχήμα 2.20.

Καθώς η τιμή της μεταβλητής  $r$  κυμαίνεται στο εύρος  $[0, 1/\mu]$  και αυξάνεται από ένα ποσοστό  $1/\mu$  κάθε φορά γίνεται μία επιλογή και εγγυάται ότι ο αριθμός των αντίγραφων που έγινε για κάθε γονέα  $i$  είναι τουλάχιστον ο ακέραιος που συμμετέχει  $\mu \cdot P_{sel}(i)$  και δεν είναι μεγαλύτερος του ένα. Τελικά, θα πρέπει να σημειωθεί ότι με ασήμαντες αλλαγές στον κώδικα, το γενικό στοχαστικό δείγμα μπορεί να χρησιμοποιηθεί για έναν οποιοδήποτε αριθμό επιλογής από τους γονείς και στην περίπτωση όπου γίνει μόνο μία επιλογή, είναι το ίδιο με την ρουλέτα.

## ΜΟΝΤΕΛΟ ΑΠΛΟΥ ΓΕΝΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ

```

{
  BEGIN
  set (current_member= 1);
  WHILE (current_member ≤ μ) DO
  Pick a random value r uniformly from [0, 1];
  set i = 1;
  WHILE (αi < r) DO
  set i = i + 1;
  OD
  set mating_pool[current_member]=parents[i];
  set current_member=current_member+1;
  OD
  END
}

```

**Σχήμα 2.19:** Ο ψευδοκώδικας για τον αλγόριθμο της ρουλέτας

## ΜΟΝΤΕΛΟ ΑΠΛΟΥ ΓΕΝΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ

```

{
  BEGIN
  set (current_member=i=1);
  Pick a random value r uniformly from [0, 1/μ];
  WHILE (current_member ≤ μ) DO
  set mating_pool[current_member]=parents[i];
  set r = r + 1/μ
  set current_member=current_member+1;
  OD
  set i = i + 1
  OD
  END
}

```

**Σχήμα 2.20:** Ο ψευδοκώδικας για τον στοχαστικό αλγόριθμο γενικού δείγματος

### 2.7.3 Επιλογή διαγωνισμού

Οι προηγούμενοι δύο μέθοδοι επιλογής και οι αλγόριθμοι χρησιμοποιούνται για να δειγματούσουν από την πιθανότητα κατανομής τους που βασίζεται στην γνώση ολόκληρου του πληθυσμού. Σε ορισμένες καταστάσεις, για παράδειγμα, εάν το μέγεθος του πληθυσμού είναι πολύ μεγάλο, ή εάν ο πληθυσμός έχει καταμεμηθεί με κάποιον τρόπο (ίσως σε ένα παράλληλο σύστημα), αποκτώντας αυτή την γνώση είναι είτε μεγάλος ο χρόνος κατανάλωσης ή στη χειρότερη περίπτωση αδύνατο. Στις άλλες περιπτώσεις μπορεί να μην υπάρχει γενικός καθορισμός καταλληλότητας. Για παράδειγμα σε μία εφαρμογή που εξελίσσονται παιχνίδια στρατηγικής μπορεί να μην είμαστε ικανοί να ποσοτικοποιήσουμε την καταλληλότητα των δοσμένων ατόμων, δηλαδή μία συγκεκριμένη στρατηγική, αλλά μπορούμε να συγκρίνουμε οποιεσδήποτε δύο από αυτές προσομοιώνοντας ένα παιχνίδι που παίζεται από αυτές τις στρατηγικές ως αντίπαλοι. Παρόμοιες καταστάσεις βρίσκονται στον εξελικτικό σχεδιασμό και στις εξελικτικές εφαρμογές τέχνης. Σε τέτοιες εφαρμογές είναι κοινό ο χρήστης υποκειμενικά να επιλέγει μεταξύ των ατόμων που αναπαριστούν σχέδια ή κομμάτια τέχνης συγκρίνοντάς τα,



χωρίς να χρησιμοποιηθεί κάποιο ποσοτικό μέτρο που προσδιορίζει μία τιμή καταλληλότητας σε κάθε μέλος του πληθυσμού.

Η επιλογή του διαγωνισμού είναι ένας τελεστής με την χρήσιμη ιδιότητα ότι δεν απαιτεί μία γενική γνώση του πληθυσμού. Αντίθετα βασίζεται σε μια συσχέτιση σειράς που μπορεί να κατατάσσει οποιαδήποτε δύο άτομα. Επομένως είναι απλό και γρήγορο να υλοποιηθεί και να εφαρμοστεί. Η εφαρμογή της επιλογής του διαγωνισμού για να επιλέξει  $\mu$  γονείς λειτουργεί σύμφωνα με την διαδικασία που φαίνεται στο σχήμα 2.21.

Επειδή η επιλογή του διαγωνισμού εξετάζει συσχετισμένη αντί απόλυτη καταλληλότητα, έχει τις ίδιες ιδιότητες όπως τα σχέδια προτεραιότητας όσον αφορά την σταθερότητα στη μετατροπή και την μετατόπιση της συνάρτησης της καταλληλότητας. Η πιθανότητα ένα άτομο να επιλεγεί ως αποτέλεσμα ενός διαγωνισμού εξαρτάται από τέσσερις παράγοντες:

- Την προτεραιότητα στον πληθυσμό. Αποτελεσματικά αυτό υπολογίζεται χωρίς την ανάγκη ταξινόμησης για ολόκληρο τον πληθυσμό.
- Τον διαγωνισμό μεγέθους  $k$ . Όσο μεγαλύτερος ο διαγωνισμός, τόσο μεγαλύτερες πιθανότητες θα περιέχουν τα μέλη παραπάνω από το μέσο όρο καταλληλότητας και το λιγότερο όπου θα υπάρχουν όλα τα χαμηλής καταλληλότητας μέλη.
- Η πιθανότητα  $p$  είναι η πιθανότητα όπου τα πιο κατάλληλα μέλη του τουρνουά επιλέγονται. Συνήθως αυτό είναι 1.0 (ντετερμινιστικά τουρνουά), αλλά οι στοχαστικές εκδοχές χρησιμοποιούνται επίσης με  $p < 1.0$ . Προφανώς σε αυτή τη περίπτωση υπάρχει χαμηλότερη πίεση επιλογής
- Εάν επιλέγονται άτομα με ή χωρίς αντικατάσταση. Στη δεύτερη περίπτωση, με τους ντετερμινιστικούς διαγωνισμούς, τα  $k - 1$  λιγότερο κατάλληλα μέλη του πληθυσμού δε μπορούν ποτέ να συλληχθούν, ενώ εάν οι υποψήφιοι του διαγωνισμού διαλεχθούν για αντικατάσταση, θα είναι πάντα πιθανό ακόμα και για το λιγότερο κατάλληλο μέλος του πληθυσμού να επιλεγεί ως αποτέλεσμα τυχερού σχεδιασμού.

Αυτές οι ιδιότητες της επιλογής διαγωνισμού χαρακτηρίζουν και παρουσιάζουν ότι για δυαδικό διαγωνισμό με ( $k = 2$ ) και με παράμετρο  $p$  ο αναμενόμενος χρόνος για ένα απλό άτομο υψηλής καταλληλότητας που αναλαμβάνει τον πληθυσμό είναι ο ίδιος με αυτόν της γραμμικής προτεραιότητας με  $s = 2p$ . Ωστόσο, καθώς απαιτούνται  $\lambda$  διαγωνισμοί για να παραχθούν  $\lambda$  επιλογές, υφίστανται από τα ίδια τα προβλήματα όπως ο αλγόριθμος της ρουλέτας, όπου τα αποτελέσματα μπορούν να δείξουν μία μεγάλη διαφορά από την θεωρητική πιθανότητα κατανομής. Πέρα από αυτό το μειονέκτημα, η επιλογή διαγωνισμού είναι ίσως ο πιο ευρέως χρησιμοποιούμενος τελεστής επιλογής σε σύγχρονες εφαρμογές ΓΑ, εξαιτίας της ακραίας απλότητας και το γεγονός ότι η πίεση επιλογής είναι εύκολο να ελεγχθεί αλλάζοντας το μέγεθος του  $k$  του τουρνουά.

## 2.8 Επιλογή επιβίωσης

Ο μηχανισμός της επιλογής επιβίωσης είναι υπεύθυνος για την διαχείριση της διαδικασίας διά της οποίας η επαρκής μνήμη του ΓΑ μειώνεται από ένα σετ  $\mu$  γονέων και  $\lambda$  απογόνων για να παράγουν το σετ των  $\mu$  ατόμων για την επόμενη γενιά. Όπως έχει εξηγηθεί, αυτό το βήμα στον κύριο εξελικτικό κύκλο ονομάζεται αντικατάσταση. Στο τωρινό τμήμα συχνά χρησιμοποιείται ο τελικός όρος για να είναι σύμφωνο με τη βιβλιογραφία. Μέσω της ιστορίας των ΓΑ ένας αριθμός στρατηγικής αντικαταστάσεων έχει προταθεί και χρησιμοποιηθεί ευρέως, ο οποίος μπορεί να κατηγοροποιηθεί σύμφωνα είτε με την διάκριση της καταλληλότητας είτε με την ηλικία των ατόμων.

---

**ΜΟΝΤΕΛΟ ΑΠΛΟΥ ΓΕΝΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ**

```

{
  BEGIN
  set (current_member= 1);
  WHILE (current_member ≤ μ) DO
  Pick k individuals randomly, with or without replacement;
  Select the best of these k comparing their fitness values;
  Denote this individual as i;
  set mating_pool[current_member]=i;
  set current_member=current_member+1;
  OD
  END
}

```

---

**Σχήμα 2.21:** Ο ψευδοκώδικας για τον αλγόριθμο επιλογής τουρνουά

### 2.8.1 Αντικατάσταση που βασίζεται στην ηλικία

Η βάση αυτών των θεμάτων είναι ότι η καταλληλότητα των ατόμων δεν λαμβάνεται υπόψιν κατά τη διάρκεια της επιλογής όπου τα άτομα αντικαθιστούν τον πληθυσμό, αντί να σχεδιάσθουν έτσι ώστε κάθε άτομο να υπάρχει στον πληθυσμό για τον ίδιο αριθμό επαναλήψεων των ΓΑ. Αυτό εμποδίζει την συνεχή παρουσία υψηλής καταλληλότητας στις λύσεις του πληθυσμού, αλλά αυτό εξαρτάται από την επιλογή τους τουλάχιστον μία φορά στην φάση της επιλογής και έπειτα τα στάδια της επιβίωσης του ανασυνδιασμού και της μετάλλαξης.

Αυτή είναι η στρατηγική που χρησιμοποιείται στον απλό ΓΑ. Καθώς ο αριθμός των απογόνων παράγεται είναι ο ίδιος με τον αριθμό των γονέων ( $\mu = \lambda$ ), κάθε άτομο υπάρχει μόνο για ένα κύκλο και οι γονείς απλά απορρίπτονται και αντικαθιστούν από ολόκληρο το σετ των απογόνων. Αυτή η στρατηγική αντικατάστασης μπορεί να υλοποιηθεί απλά σε έναν ΓΑ με υπέρθετους πληθυσμούς ( $\lambda < \mu$ ) και στο άλλο ακραίο όπου ένας μονός απόγονος δημιουργείται και εισάγεται στον πληθυσμό σε κάθε κύκλο. Σε αυτή την περίπτωση η στρατηγική παίρνει την μορφή της ουράς (FIFO).

Μία εναλλακτική μέθοδος της αντικατάστασης για σταθερή κατάσταση γενετικών αλγορίθμων είναι να επιλεγεί τυχαία ένας γονέας για αντικατάσταση, όπου έχει το ίδιο μέσο αποτέλεσμα. Οι DeJong και Sarma ερεύνησαν αυτή τη στρατηγική πειραματικά και βρήκαν ότι ο αλγόριθμος έδειξε μεγαλύτερη διακύμανση στην παρουσίαση από έναν ισοδύναμο γενικό ΓΑ. Οι Smith και Vavak έδειξαν ότι αυτό ίσχυε επειδή η τυχαία στρατηγική είναι πολύ πιο πιθανό να χάσει τα καλύτερα μέλη του πληθυσμού από την στρατηγική του να διαγράφονται τα παλαιότερα FIFO. Γι' αυτούς τους λόγους δεν συνιστάται η στρατηγική της τυχαίας αντικατάστασης.

### 2.8.2 Αντικατάσταση βασισμένη στην καταλληλότητα

Ένας μεγάλος αριθμός στρατηγικών προτάθηκαν για να επιλεγεί ποιο  $\mu$  από τους  $\mu + \lambda$  γονείς, θα είναι αυτό όπου οι απόγονοι θα έπρεπε να προχωρήσουν στην επόμενη επανάληψη ΓΑ. Πολλές από αυτές επίσης συμπεριλαμβάνουν μερικά στοιχεία ηλικίας, έτσι ώστε όλοι οι απόγονοι να προχωρήσουν, π.χ., χρησιμοποιούν την καταλληλότητα για να αποφασίσουν ποιο  $\lambda$  από τους  $\mu > \lambda$  γονείς θα έπρεπε να αντικατασταθεί από τους απογόνους.

Ήδη έχει αναφερθεί η αναλογική καταλληλότητα, η επιλογή του διαγωνισμού και η στοχαστική εκδοχή που βασίζεται στην προτεραιότητα επιλογής. Επομένως θα περιορίσουμε αυτή τη συζήτηση για να αναφέρουμε ότι είναι επίσης πιθανό τα σχέδια αντικατάστασης (βασισμένα στην αντίστροφη καταλληλότητα ή προτεραιότητα) και σύντομα περιγράφονται

δύο άλλοι κοινοί μηχανισμοί.

### **Αντικατάσταση χειρότερου γονιού**

Σε αυτό το σχέδιο τα χειρότερα  $\lambda$  μέλη του πληθυσμού επιλέγονται για αντικατάσταση. Αν και αυτό μπορεί να οδηγήσει σε πολύ γρήγορες βελτιώσεις στο μέσο πληθυσμό καταλληλότητας, μπορεί επίσης να οδηγήσει σε πρόωρη σύγκληση καθώς ο πληθυσμός τείνει γρήγορα να επικεντρωθεί στα καταλληλότερα μέλη που παρουσιάζονται συγχρόνως. Γι' αυτό το λόγο χρησιμοποιείται συχνά για την σύνδεση μεγάλων πληθυσμών και/ή η τακτική 'όχι αντίγραφα'.

### **Ελιτισμός**

Αυτό το σχέδιο χρησιμοποιείται για τη σύνδεση που βασίζεται στην ηλικία και τα στοχαστικά σχέδια βασισμένα στην καταλληλότητα αντικατάστασης, σε μία προσπάθεια να εμποδιστεί η απώλεια του πρόσφατα καταλληλότερου μέλους του πληθυσμού. Στην ουσία ένα ίχνος κρατείται στο πρόσφατα καταλληλότερο μέλος και πάντα διατηρείται στον πληθυσμό. Επιπλέον εάν αυτό επιλεγεί στο γκρουπ που θα αντικατασταθεί και κανένας από τους απόγονους που εισήχθηκε στον πληθυσμό δεν έχει ισοδύναμη ή καλύτερη καταλληλότητα, τότε διατηρείται και ένας από τους απόγονους απορρίπτεται.

## **2.9 Παράδειγμα εφαρμογής: Επίλυση του προβλήματος προγραμματισμός εργασίας**

Δύο παραδείγματα εφαρμογής των γενετικών αλγορίθμων είναι το πρόβλημα των οχτώ-βασίλισσών και το πρόβλημα του σακιδίου. Αυτά τα παραδείγματα χρησιμοποίησαν άμεσες αναπαραστάσεις, όπου ο γονότυπος και ο φαινότυπος έχουν ένα αρκετά απλό σχεδιασμό. Μία άλλη μέθοδος (μερικές φορές πολύ χρήσιμη) είναι η χρήση έμμεσης αναπαράστασης. Εδώ υπάρχει μία συνάρτηση ανάπτυξης που χτίζει τον φαινότυπο και χρησιμοποιεί τον γονότυπο ως μία παράμετρο εισόδου. Στο παράδειγμα που ακολουθεί, το πρόβλημα του προγραμματισμού εργασίας επιλύεται έχοντας ένα ευριστικό πρόγραμμα οικοδομής και μία αναπαράσταση που βασίζεται στην προτεραιότητα που καθορίζει με ποια σειρά θα προσπαθήσουμε να τοποθετήσουμε τις δουλειές στο πρόγραμμα. Καθορίζουμε το πρόγραμμα από:

- $J$ , ένα σει εργασιών
- $O$ , ένα σει τελεστών
- $M$ , ένα σει μηχανισμών
- Ικανό:  $O \rightarrow M$ , μία συνάρτηση καθορισμού όπου ο μηχανισμός μπορεί να παρουσιάσει τον δοσμένο τελεστή
- $Pre \subseteq O \times O$ , μία συσχέτιση καθορισμού όπου ο τελεστής πρέπει να ξεπερνά άλλους τελεστές
- $d : O \times M \rightarrow \mathbb{R}$ , μία συνάρτηση καθορισμού της διάρκειας του  $o \in O$  στο  $m \in M$

Γενικά, θα μπορούσαν να υπάρχουν περισσότεροι μηχανισμοί που θα ήταν ικανοί να παρουσιάσουν ένα δοσμένο τελεστή. Σε αυτή την περίπτωση το Ικανό  $\subseteq d : O \times M$  είναι μία συσχέτιση, αντί μία συνάρτηση. Για να διατηρηθούν απλά τα πράγματα για παράδειγμα, θα αποβάλλουμε την διάσταση και θα υποθέσουμε ότι για κάθε τελεστή υπάρχει ακριβώς ένας μηχανισμός Ικανός ( $o$ )  $\in M$  που μπορεί να παρουσιαστεί. Προγραμματίζοντας έναν τελεστή σημαίνει να προσδιοριστεί ένας χρόνος έναρξης σε αυτόν (με τον μόνο μηχανισμό που μπορεί να τον παρουσιάσει) και ένα πρόγραμμα είναι μία συλλογή τέτοιων εκχωρήσεων που περιλαμβάνονται σε έναν τελεστή παραπάνω από μία φορά. Ο στόχος τώρα είναι να βρεθεί ένα πρόγραμμα δηλαδή:

- Ολοκληρωμένο: όλες οι εργασίες προγραμματίστηκαν
- Σωστό: όλες οι συνθήκες που καθορίστηκαν από το Pre ικανοποιούνται
- Καταλληλότερο: η συνολική διάρκεια του προγράμματος είναι μηδαμινή

Μπορούμε να θέσουμε έναν ΓΑ γι' αυτό το πρόβλημα έχοντας έναν πληθυσμό ατόμων που είναι όλες οι μεταλλαγές των σειρών των πιθανών τελεστών. Τέτοια μετάλλαξη δημιουργεί ένα πρόγραμμα για οποιαδήποτε προτεραιότητα προγραμματίζοντας κάθε τελεστή με τη σειρά εμφάνισης της δοσμένης μετάλλαξης. Το πρόγραμμα οικοδομής λειτουργεί με τον παρακάτω τρόπο:

- Παίρνουμε τον επόμενο τελεστή (ξεκινώντας από τον πρώτο) ως καθορισμένο στη μετάλλαξη.
- Αναζητούμε τον μηχανισμό του.
- Προσδιορίζουμε τον πιο πρόσφατο πιθανό χρόνο έναρξης σε αυτό τον μηχανισμό, υπό τον όρο της απασχόλησης του μηχανισμού και τις συσχετίσεις διαδικασίας που διατηρείται αυτός ο τελεστής στο πρόγραμμα μέχρι τώρα.

Μέχρι τώρα έχουμε καθορίσει την αναπαράσταση, δηλαδή καθορίσαμε ποιό είναι ο γονότυπος μας (μεταλλάξεις), φαινότυπους (προγράμματα) και καθορίσαμε λεπτομερώς πώς ένας γονότυπος σχεδιάστηκε σε φαινότυπο (με το πρόγραμμα οικοδομής). Είναι σημαντικό να σημειώσουμε ότι ο σχεδιασμός εγγυάται την πληρότητα και την ορθότητα των απαιτήσεων. Η πληρότητα είναι απλά μία συνέπεια της χρήσης μεταλλάξεων που περιλαμβάνουν όλους τους τελεστές. Η συνθήκη της ορθότητας ικανοποιείται από το πρόγραμμα οικοδομής που προσδιορίζει χρόνους έναρξης, παίρνοντας την προτεραιότητα που περιορίζεται σε ποσοστά. Όσο για την καταλληλότερη εφαρμογή, το πρόγραμμα οικοδομής χρησιμοποιεί έναν κατάλληλο τοπικό ευριστικό, που πάντα προσδιορίζει το πιο πρόσφατο πιθανό χρόνο έναρξης στο δοσμένο τελεστή. Αυτό ωστόσο, δεν συνεπάγεται ότι το ολόκληρο πρόγραμμα θα είναι κατάλληλο. Για να επιτευχθεί αυτός ο στόχος καθορίζουμε την καταλληλότητα ενός ατόμου, δηλαδή, έναν γονότυπο, ως τη διάρκεια του αντίστοιχου φαινότυπου (πρόγραμμα). Προφανώς αυτή η καταλληλότητα πρέπει να ελαττωθεί. Για να ολοκληρωθεί ο σχεδιασμός ενός ΓΑ γι' αυτό το πρόβλημα πρέπει να προσδιορίσουμε τα συστατικά του υπόλοιπου αλγόριθμου: τελεστές επιλογής (για τους γονείς και την επιλογή επιβίωσης) και τελεστές μεταβολής (για μετάλλαξη και ανασυνδιασμό). Τέλος, πρέπει να καθορίσουμε την διαδικασία υλοποίησης και την συνθήκη τερματισμού. Ένα πολύ ελκυστικό χαρακτηριστικό των εξελικτικών αλγορίθμων είναι ότι υπάρχουν πολλοί ευρέως εφαρμόσιμοι τελεστές που μπορούν απλά να γίνουν εύκολα. Τελεστές μεταβολής είναι στενά συσχετισμένοι με την αναπαράσταση, δηλαδή στις χρησιμοποιημένες μορφές γονότυπων. Στην περίπτωση μας μπορούμε να σχεδιάσουμε από την συλλογή της μετάλλαξης που βασίζεται στην προτεραιότητα και τους τελεστές διασταύρωσης. Τυπικά οποιοσδήποτε από αυτούς το κάνει, με την έννοια ότι είναι εφαρμόσιμοι δηλαδή συντακτικά σωστοί γονείς (μεταλλάξεις) πάντα θα έχουν ως αποτέλεσμα συντακτικά ορθά παιδιά. Η διαφορά μεταξύ τους μπορεί να είναι η παρουσίαση τους όσον αφορά το τέλος της λύσης που παραδίδεται από τους ΓΑ που τους χρησιμοποιούν. Όσο για τους τελεστές επιλογής, το θέμα είναι ακόμα πιο απλό καθώς δεν εξαρτώνται από συγκεκριμένη αναπαράσταση, έτσι δεν θα πρέπει απαραίτητως να ταιριάζουν με τους γονότυπους. Εδώ μπορούμε να χρησιμοποιήσουμε οποιοδήποτε από τους μηχανισμούς που αναφέρθηκαν σε αυτό το κεφάλαιο. Το μόνο πράγμα για το οποίο πρέπει να είμαστε ενήμεροι είναι ότι εμείς τώρα ελαχιστοποιούμε την διάρκεια των προγραμμάτων. Λαμβάνοντας υπόψη τους δύο τελευταίους παράγοντες, την υλοποίηση και την συνθήκη τερματισμού συνήθως επαρκούν για να χρησιμοποιήσεις τυχαία υλοποίηση και για παράδειγμα, να σου επιτραπεί ένας μέγιστος αριθμός της αποτίμησης καταλληλότητας.

## Εξελικτικές Στρατηγικές (Evolution Strategies)

### 3.1 Σκοπός αυτού του κεφαλαίου

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τις *Εξελικτικές Στρατηγικές* (Evolution Strategies) (ΕΣ), ένα ακόμη μέλος της οικογένειας των εξελικτικών αλγορίθμων. Χρησιμοποιούμε αυτούς τους αλγορίθμους για να υλοποιήσουμε ένα πολύ χρήσιμο χαρακτηριστικό του εξελικτικού υπολογισμού: την αυτοπροσαρμογή (self-adaptation) των παραμέτρων της στρατηγικής. Γενικά, η αυτοπροσαρμοστικότητα σημαίνει ότι μερικοί παράμετροι των ΕΑ αλλάζουν κατά τη διάρκεια μίας εκτέλεσης με ένα ειδικό τρόπο όπου οι παράμετροι συμπεριλαμβάνονται στα χρωμοσώματα και εξελίσσονται με τις λύσεις. Αυτό το χαρακτηριστικό είναι κληρονομικό στις σύγχρονες εξελικτικές στρατηγικές. Μία περίληψη των ΕΣ δίνεται στον πίνακα 3.1.

### 3.2 Παράδειγμα εισαγωγής

Τώρα θα περιγράψουμε τον βασικό αλγόριθμο που εκφράζει τα δύο μέλη της εξελικτικής στρατηγικής για το πρόβλημα της μείωσης μίας  $n$ -διαστάσεων συνάρτησης στο  $\mathbb{R}^n \rightarrow \mathbb{R}$ . Μία περιγραφή μιας απλής εξελικτικής στρατηγικής δύο μελών δίνεται στο σχήμα 3.1.

Δίνοντας μία λύση που ισχύει  $\bar{x}^t$  στη μορφή ενός διανύσματος μήκους  $n$ , δημιουργείται μία νέα υποψήφια λύση  $\bar{x}^{t+1}$  προσθέτοντας έναν τυχαίο αριθμό  $z_i$  για  $i \in \{1, \dots, n\}$  για κάθε ένα από τους  $n$  παράγοντες. Μία γκαουσιανή ή κανονική κατανομή χρησιμοποιήθηκε με αριθμητικό μέσο μηδέν και τυπική απόκλιση  $\sigma$  για την επιλογή τυχαίων αριθμών. Αυτή η κατανομή είναι συμμετρική γύρω στο μηδέν και έχει το χαρακτηριστικό ότι η πιθανότητα επιλογής ενός τυχαίου αριθμού με οποιοδήποτε δοσμένο μέγεθος είναι μία ελαττούμενη συνάρτηση της τυπικής απόκλισης  $\sigma$ . Επιπλέον η μεταβλητή  $\sigma$  είναι μία παράμετρος του αλγορίθμου που προσδιορίζει την έκταση στην οποία οι δοσμένες μεταβλητές  $x_i$  αλλάζουν από τον τελεστή μετάλλαξης. Γι' αυτό το λόγο το  $\sigma$  συχνά ονομάζεται βήμα μετάλλαξης (mutation step size). Θεωρητικές μελέτες ωθούν μία εφαρμογή των βημάτων μετάλλαξης από τον γνωστό  $1/5$  κανόνα επιτυχίας του Rechenberg [7]. Αυτός ο κανόνας δηλώνει ότι η αναλογία των επιτυχημένων μεταλλάξεων (εκείνων των μεταλλάξεων όπου τα παιδιά είναι

Αναπαράσταση	διανύσματα πραγματικών μεταβλητών
Ανασυνδιασμός	διακριτός ή ενδιάμεσος
Μετάλλαξη	γκαουσιανή διατάραξη
Επιλογή γονέα	ομοιόμορφα τυχαία
Επιλογή επιβίωσης	$(\mu, \lambda)$ ή $(\mu + \lambda)$
Ειδικότητα	αυτοπροσαρμογή των βημάτων μετάλλαξης

Πίνακας 3.1: Σκίτσο της ΕΣ

## ΜΟΝΤΕΛΟ ΑΠΛΗΣ ΕΞΕΛΙΚΤΙΚΗΣ ΣΤΡΑΤΗΓΙΚΗΣ

```

{
  BEGIN
  set  $t = 0$ ;
  Create an initial point  $\langle x_1^t, \dots, x_n^t \rangle \in \mathbb{R}^2$ ;
  REPEAT UNTIL (TERMINATION CONDITION is satisfied) DO
  draw  $z_i$  from a normal distr. for all  $i \in 1, \dots, n$ ;
   $y_i^t = x_i^t + z_i$  for all  $i \in 1, \dots, n$ ;
  IF ( $f(\bar{x}^2) \leq f(\bar{y}^2)$ ) THEN
   $\bar{x}^{t+1} = \bar{x}^t$ ;
  ELSE
   $\bar{x}^{t+1} = \bar{y}^t$ ;
  FI
  set  $t = t + 1$ ;
  OD
  END
}

```

**Σχήμα 3.1:** Περιγραφή μίας απλής διμελής εξελικτικής στρατηγικής

καταλληλότερα από τους γονείς) προς όλες τις μεταλλάξεις θα έπρεπε να είναι  $1/5$ . Γι' αυτό το λόγο αν η αναλογία είναι μεγαλύτερη από  $1/5$  το βήμα μετάλλαξης θα πρέπει να αυξηθεί για να γίνει μεγαλύτερο το διάστημα αναζήτησης στο χώρο και αν η αναλογία είναι μικρότερη του  $1/5$  τότε θα πρέπει να μειωθεί για να συγκεντρωθεί η αναζήτηση πιο πολύ γύρω από την λύση που ισχύει. Ο κανόνας εκτελείται κατά περιοδικά διαστήματα, για παράδειγμα, μετά από  $k$  επαναλήψεις κάθε  $\sigma$  επαναπροσδιορίζεται από

$$\sigma = \begin{cases} \sigma/c & \text{εάν } p_s > 1/5, \\ \sigma \cdot c & \text{εάν } p_s < 1/5, \\ \sigma & \text{εάν } p_s = 1/5 \end{cases}$$

όπου  $p_s$  είναι η σχετική συχνότητα των επιτυχημένων μεταλλάξεων που έχει μετρηθεί πάνω από κάποιο αριθμό προσπαθειών και η παράμετρος  $c$  ανήκει στο διάστημα  $0.817 \leq c \leq 1$ . Χρησιμοποιώντας αυτό το μηχανισμό το βήμα μετάλλαξης αλλάζει βασισμένο στις πληροφορίες από την διαδικασία αναζήτησης.

Αυτό το παράδειγμα διευκρινίζει μερικά χαρακτηριστικά των εξελικτικών στρατηγικών:

1. Οι εξελικτικές στρατηγικές χρησιμοποιούνται για συνεχής παραμέτρους βελτιστοποίησης.
2. Υπάρχει ισχυρή χρήση στη μετάλλαξη για την δημιουργία απογόνων.
3. Η μετάλλαξη υλοποιείται προσθέτοντας κάποιον τυχαίο θόρυβο που προέρχεται από μία γκαουσιανή κατανομή.
4. Οι παράμετροι της μετάλλαξης αλλάζουν κατά τη διάρκεια της εκτέλεσης του αλγορίθμου.

### 3.3 Αναπαράσταση

Οι εξελικτικές στρατηγικές χρησιμοποιούνται για συνεχής παραμέτρους βελτιστοποίησης, που σημαίνει ότι το πρόβλημα μπορεί να δοθεί ως μία αντικειμενική συνάρτηση

$\mathfrak{R}^n \rightarrow \mathfrak{R}$ . Η καθιερωμένη αναπαράσταση των αντικειμενικών μεταβλητών  $x_1, \dots, x_n$  είναι πολύ ακριβής, όπου κάθε  $x_i$  αναπαριστάται από μία μεταβλητή κινητής υποδιαστολής. Αγνοώντας την αυτοπροσαρμοστικότητα για την ώρα, σημαίνει ότι ο χώρος του γονότυπου είναι ταυτόσημος με τον χώρο του φαινότυπου και γι' αυτό το λόγο δεν χρειάζεται κανένα ειδικό βήμα κωδικοποίησης. Ωστόσο καθώς στις μέρες μας οι εξελικτικές στρατηγικές σχεδόν πάντα χρησιμοποιούν την αυτοπροσαρμογή (αντί τον 1/5 κανόνα επιτυχίας της προσαρμογής), το διάνυσμα  $\bar{x} = \langle x_1, \dots, x_n \rangle$  μορφοποιεί μόνο ένα μέρος από τον γονότυπο των ΕΣ. Τα άτομα περικλείουν μερικές παραμέτρους στρατηγικής, συγκεκριμένα, παραμέτρους από τον τελεστή μετάλλαξης.

Οι παράμετροι στρατηγικής μπορούν να χωριστούν σε δύο σεί, τις  $\sigma$  και τις  $\alpha$  μεταβλητές. Οι  $\sigma$  μεταβλητές αναπαριστούν το βήμα μετάλλαξης και το  $n_\sigma$  είναι συνήθως είτε 1 είτε  $n$ . Για οποιοδήποτε μηχανισμό αυτοπροσαρμογής πρέπει να παρουσιαστεί τουλάχιστο ένα  $\sigma$ . Οι  $\alpha$  μεταβλητές αναπαριστούν τις αλληλεπιδράσεις μεταξύ των βημάτων μετάλλαξης και χρησιμοποιούνται για διάφορες μεταβλητές. Στις πιο γενικές περιπτώσεις έχουμε  $n_\alpha = (n - \frac{n_\sigma}{2})(n_\sigma - 1)$ . Αν τα τοποθετήσουμε όλα αυτά μαζί, παίρνουμε

$$\left\langle \underbrace{x_1, \dots, x_n}_{\bar{x}}, \underbrace{\sigma_1, \dots, \sigma_{n_\sigma}}_{\bar{\sigma}}, \underbrace{\alpha_1, \dots, \alpha_{n_\alpha}}_{\bar{\alpha}} \right\rangle$$

ως την γενική μορφή των ατόμων των ΕΣ.

### 3.4 Μετάλλαξη

Ο τελεστής μετάλλαξης στις ΕΣ βασίζεται σε μια κανονική (γκουσιανή) κατανομή και απαιτεί δύο παραμέτρους: τον αριθμητικό μέσο  $\xi$  και την τυπική απόκλιση  $\sigma$ . Μεταλλάξεις υλοποιούνται προσθέτοντας  $\Delta x_i$  σε κάθε  $x_i$ , όπου οι μεταβλητές  $\Delta x_i$  επιλέχθηκαν από την δοσμένη γκαουσιανή κατανομή  $N(\xi, \sigma)$ , με την αντίστοιχη συνάρτηση πυκνότητας πιθανότητας

$$p(\Delta x_i) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(\Delta x_i - \xi)^2}{2\sigma^2}}. \quad (3.1)$$

Στην πράξη, ο αριθμητικός μέσος  $\xi$  πάντα τοποθετείται στο μηδέν και το διάνυσμα  $\bar{x}$  αλλάζει αντικαθιστώντας τις μεταβλητές  $x_i$  με  $x'_i = x_i + N(0, \sigma')$ , όπου το  $N(0, \sigma)$  δηλώνει έναν τυχαίο αριθμό που προήλθε από μία γκαουσιανή κατανομή με αριθμητικό μέσο μηδέν και τυπική απόκλιση  $\sigma$ . Χρησιμοποιώντας μία γκαουσιανή κατανομή, οι μικρές μεταλλάξεις είναι πιο πιθανές από τις μεγάλες. Το ιδιαίτερο χαρακτηριστικό της μετάλλαξης στις ΕΣ -και η βάση της αυτοπροσαρμογής- είναι ότι το βήμα μετάλλαξης συμπεριλαμβάνεται στα χρωμοσώματα και τα ίδια υφίστανται μετάλλαξη και επιλογή. Στην πιο απλή περίπτωση θα είχαμε ένα βήμα μετάλλαξης το οποίο εφαρμόζεται σε όλους τους παράγοντες  $x_i$  και οι υποψήφιοι λύσεις είναι της μορφής  $\langle x_1, \dots, x_n, \sigma \rangle$ . Στη συνέχεια υλοποιούνται μεταλλάξεις αντικαθιστώντας το  $\langle x_1, \dots, x_n, \sigma \rangle$  με το  $\langle x'_1, \dots, x'_n, \sigma' \rangle$ , όπου  $\sigma'$  είναι η μεταβλητή  $\sigma$  που άλλαξε και  $x'_i = x_i + N(0, \sigma')$ .

Λεπτομέρειες για το πώς θα αλλάξει η μεταβλητή  $\sigma$  δίνονται παρακάτω. Το βήμα μετάλλαξης δεν ορίζεται από τον χρήστη. Προκειμένου να επιτευχθεί αυτό είναι απαραίτητο να τροποποιηθεί πρώτα η μεταβλητή  $\sigma$  και μετά να αλλάξουν οι μεταβλητές  $x_i$  με την νέα μεταβλητή  $\sigma$ . Το σκεπτικό πίσω από αυτό είναι ότι ένα νέο άτομο  $\langle \bar{x}', \sigma' \rangle$  αποτιμάται δύο φορές. Αρχικά, αποτιμάται για την βιωσιμότητά του κατά τη διάρκεια της επιλογής επιβίωσης που βασίζεται στο  $f(\bar{x}')$ . Έπειτα, αποτιμάται για την ικανότητά του να δημιουργηθεί ένας καλός απόγονος. Αυτό συμβαίνει έμμεσα καθώς ένα δοσμένο βήμα μετάλλαξης αποτιμάται ευνοϊκά αν ο απόγονος παράγεται επειδή αποδεικνύεται βιώσιμος. Επιπλέον, ένα άτομο  $\langle \bar{x}', \sigma' \rangle$  αναπαριστά και ένα καλό  $x'$  που επιβίωσε από την επιλογή και ένα καλό  $\sigma'$  το οποίο είναι επιτυχημένο επειδή παρήγαγε αυτό το καλό  $\bar{x}'$  από το  $\bar{x}$ .

### 3.4.1 Αουσχέτιστες μεταλλάξεις με βήμα μετάλλαξης ένα

Σε αυτή την περίπτωση αουσχέτιστων μεταλλάξεων με βήμα μετάλλαξης ένα, χρησιμοποιείται η ίδια κατανομή για να αλλαχθεί κάθε  $x_i$ , επομένως έχουμε μόνο ένα  $\sigma$  για κάθε άτομο. Αυτό το  $\sigma$  αλλάζει κάθε βήμα πολλαπλασιάζοντάς το με τον όρο  $e^\Gamma$ , με  $\Gamma$  να είναι μία τυχαία μεταβλητή που επιλέχθηκε από την κανονική κατανομή με αριθμητικό μέσο 0 και τυπική απόκλιση  $\tau$ . Καθώς  $N(0, \tau) = \tau \cdot N(0, 1)$ , ο μηχανισμός μετάλλαξης καθορίζεται από τους παρακάτω τύπους:

$$\sigma' = \sigma \cdot e^{\tau \cdot N(0,1)}, \quad (3.2)$$

$$x'_i = x_i + \sigma' N_i(0, 1). \quad (3.3)$$

Εφόσον οι τυπικές αποκλίσεις είναι ανεπιθύμητες πολύ κοντά στο μηδέν (θα έχουν για μέσο όρο ένα αμελητέο αποτέλεσμα), ο ακόλουθος κανόνας ορίου χρησιμοποιείται για να αναγκαστούν τα βήματα μετάλλαξης να είναι μεγαλύτερα από ένα κατώφλι:

$$\sigma' < e_0 \Rightarrow \sigma' = e_0.$$

Σε αυτούς τους τύπους το  $N(0, 1)$  δηλώνει μία επιλογή από την κανονική κατανομή, ενώ το  $N_i(0, 1)$  δηλώνει μία ξεχωριστή επιλογή από την κανονική κατανομή για κάθε μεταβλητή  $i$ . Η αναλογικότητα της σταθεράς  $\tau$  είναι μία εξωτερική παράμετρος που θα οριστεί από τον χρήστη. Συνήθως είναι αντίστροφα ανάλογο με την τετραγωνική ρίζα του μεγέθους του προβλήματος:  $\tau \propto 1/\sqrt{n}$ . Η παράμετρος  $\tau$  μπορεί να ερμηνευτεί ως ένα είδος **ρυθμού μάθησης**, όπως τα νευρωνικά δίκτυα. Ο Back εξηγεί παρακάτω τους λόγους που αλλάζουμε το  $\sigma$  πολλαπλασιάζοντάς το με μία μεταβλητή μίας λογαριθμικής κατανομής:

- Οι μικρές τροποποιήσεις θα πρέπει να απατώνται πιο συχνά από τις μεγάλες.
- Οι τυπικές αποκλίσεις θα πρέπει να είναι μεγαλύτερες του 0.
- Ο αριθμητικός μέσος θα πρέπει να είναι 1, καθώς θέλουμε να πολλαπλασιάσουμε το  $\sigma$ .
- Η μετάλλαξη θα πρέπει να είναι ουδέτερη στον μέσο όρο.

Η λογαριθμική κατανομή ικανοποιεί όλες αυτές τις απαιτήσεις.

Στο σχήμα 3.2 φαίνονται τα αποτελέσματα της μετάλλαξης σε δύο διαστάσεις. Δηλαδή, έχουμε μία αντικειμενική συνάρτηση  $\mathbb{R}^2 \rightarrow \mathbb{R}$  και τα άτομα είναι της μορφής  $\langle x, y, \sigma \rangle$ . Εφόσον υπάρχει μόνο ένα  $\sigma$ , το βήμα μετάλλαξης είναι το ίδιο για κάθε διεύθυνση ( $x$  και  $y$ ).

### 3.4.2 Αουσχέτιστες μεταλλάξεις με βήμα μετάλλαξης $n$

Ο λόγος που χρησιμοποιούμε  $n$  βήματα μετάλλαξης είναι γιατί θέλουμε να επεξεργαστούμε διαφορετικές διαστάσεις. Θέλουμε να είμαστε ικανοί να χρησιμοποιούμε διάφορα βήματα μετάλλαξης για διάφορες διαστάσεις  $i \in 1, \dots, n$ . Η λύση είναι ακριβής: κάθε χρωμόσωμα  $\langle x_1, \dots, x_n \rangle$  επεκτείνεται με  $n$  βήματα μετάλλαξης, ένα για κάθε διάσταση, καταλήγοντας σε  $\langle x_1, \dots, x_n, \sigma_1, \dots, \sigma_n \rangle$ . Ο μηχανισμός μετάλλαξης τώρα γίνεται:

$$\sigma'_i = \sigma_i \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)}, \quad (3.4)$$

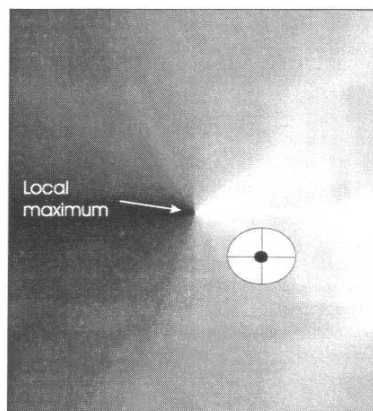
$$x'_i = x_i + \sigma_i \cdot N_i(0, 1), \quad (3.5)$$

όπου  $\tau' \propto 1/\sqrt{2n}$  και  $\tau \propto 1/\sqrt{2\sqrt{n}}$ . Μία ακόμη φορά ένας κανόνας ορίου εφαρμόζεται για να προλάβει τις τυπικές αποκλίσεις που είναι πολύ κοντά στο μηδέν.

$$\sigma'_i < e_0 \Rightarrow \sigma'_i = e_0.$$

Να σημειωθεί ότι ο τύπος της μετάλλαξης για το  $\sigma$  είναι διαφορετικός από αυτόν της εξίσωσης (3.2). Ο τωρινός μηχανισμός μετάλλαξης βασίζεται σε μικρότερα τμήματα (finer granularity). Αντί για το επίπεδο του ατόμου (κάθε άτομο  $\bar{x}$  έχει το δικό του  $\sigma$ ) λειτουργεί στο ισοδύναμο επίπεδο (ένα  $\sigma_i$  για κάθε  $x_i$  στο  $\bar{x}$ ). Η αντίστοιχη άμεση τροποποίηση της εξίσωσης (3.2) είναι

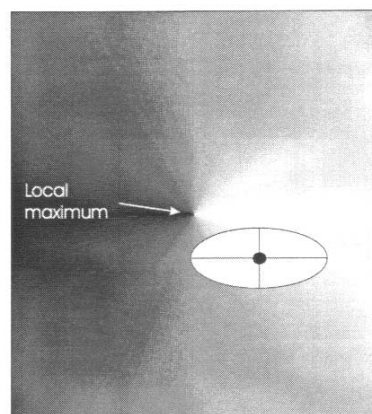




**Σχήμα 3.2:** Μετάλλαξη με  $n = 2$ ,  $n_\sigma = 1$ ,  $n_\alpha = 0$ . Κομμάτι από ένα μέρος της καταλληλότητας με ένα κωνικό σχήμα να φαίνεται. Το μαύρο δεν υποδεικνύει ένα άτομο. Τα σημεία όπου οι απόγονοι μπορούν να διατεθούν με μία δοσμένη πιθανότητα σχηματίζουν ένα κύκλο. Η πιθανότητα μετακίνησης κατά μήκος του άξονα  $y$  (μικρή επίδραση της καταλληλότητας) είναι η ίδια με αυτή της μετακίνησης κατά μήκος του άξονα  $x$  (μεγάλη επίδραση της καταλληλότητας).

$$\sigma'_i = \sigma_i \cdot e^{\tau \cdot N_i(0,1)},$$

αλλά οι ΕΣ χρησιμοποιούν την εξίσωση (3.4). Αυτό είναι σωστό καθώς το άθροισμα των δύο κανονικά κατανομημένων μεταβλητών είναι κανονικά κατανομημένο, γι' αυτό το λόγο η κατανομή που προκύπτει είναι ακόμη λογαριθμική.



**Σχήμα 3.3:** Μετάλλαξη με  $n = 2$ ,  $n_\sigma = 2$ ,  $n_\alpha = 0$ . Κομμάτι από ένα μέρος της καταλληλότητας με ένα κωνικό σχήμα να φαίνεται. Το μαύρο δεν υποδεικνύει ένα άτομο. Τα σημεία όπου οι απόγονοι μπορούν να διατεθούν με μία δοσμένη πιθανότητα σχηματίζουν μία έλλειψη. Η πιθανότητα μετακίνησης κατά μήκος του άξονα  $x$  (μεγάλη επίδραση της καταλληλότητας) είναι μεγαλύτερη της μετακίνησης κατά μήκος του άξονα  $y$  (μικρή επίδραση της καταλληλότητας).

Στο σχήμα 3.3 τα αποτελέσματα των μεταλλάξεων φαίνονται σε δύο διαστάσεις. Εάν έχουμε μία αντικειμενική συνάρτηση  $\mathbb{R}^2 \rightarrow \mathbb{R}$ , αλλά τα άτομα τώρα έχουν τον τύπο  $\langle x, y, \sigma_x, \sigma_y \rangle$ . Καθώς τα βήματα μετάλλαξης μπορούν να διαφέρουν σε κάθε διεύθυνση ( $x$  και  $y$ ), τα σημεία σε αυτό τον χώρο αναζήτησης όπου οι απόγονοι μπορούν να τοποθετηθούν με μία δοσμένη πιθανότητα σχηματίζουν μία έλλειψη. Οι άξονες μίας τέτοιας έλλειψης είναι παράλληλοι με τους άξονες συντεταγμένων, με το μήκος του άξονα  $i$  να είναι ανάλογο με την μεταβλητή  $\sigma_i$ .

### 3.4.3 Συσχετισμένες μεταλλάξεις

Η δεύτερη εκδοχή της μετάλλαξης αναφέρεται στο προηγούμενο κομμάτι που εισήγαγε διάφορες τυπικές αποκλίσεις για κάθε άξονα, αλλά αυτό επιτρέπει μόνο ελλείψεις ορθογώνιες με τους άξονες. Το σκεπτικό που κρύβεται πίσω από τις συσχετισμένες μεταλλάξεις είναι να επιτραπεί στις ελλείψεις να έχουν οποιονδήποτε προσανατολισμό περιστρέφοντάς τους με μία περιστροφή του πίνακα  $C$ .

Η συνάρτηση πυκνότητας πιθανότητας για  $\bar{\Delta x}$  που αντικαθιστάται στην εξίσωση (3.1) τώρα γίνεται

$$p(\bar{\Delta x}) = \frac{e^{-\frac{1}{2}\bar{\Delta x}^2 \cdot C^{-1} \cdot \bar{\Delta x}}}{(\det C \cdot (2\pi)^n)^{1/2}}$$

με  $C$  τον πίνακα συνδιακύμανσης με εισόδους

$$c_{ii} = \sigma_i^2, \quad (3.6)$$

$$c_{ij, i \neq j} = \begin{cases} 0 & \text{όχι συσχετίσεις,} \\ \frac{1}{2}(\sigma_i^2 - \sigma_j^2)\tan(2\alpha_{ij}) & \text{συσχετίσεις.} \end{cases} \quad (3.7)$$

Η σχέση μεταξύ της συνδιακύμανσης και της γωνίας περιστροφής είναι η παρακάτω:

$$\tan(2\alpha_{ij}) = \frac{2c_{ij}}{\sigma_i^2 - \sigma_j^2}$$

η οποία εξηγεί την εξίσωση (3.7). Μία περιστροφή σε δύο διαστάσεις είναι ένας πολλαπλασιασμός με τον πίνακα

$$\begin{pmatrix} \cos(\alpha_{ij}) & -\sin(\alpha_{ij}) \\ \sin(\alpha_{ij}) & \cos(\alpha_{ij}) \end{pmatrix}$$

Ο ολοκληρωμένος μηχανισμός μετάλλαξης περιγράφεται από τις παρακάτω εξισώσεις:

$$\sigma'_i = \sigma_i \cdot e^{\tau' \cdot N(0,1) + \tau \cdot N(0,1)}, \quad \alpha'_j = \alpha_j + \beta \cdot N(0,1), \quad \bar{x}' = \bar{x} + \bar{N}(\bar{0}, C')$$

όπου  $n_a = \frac{n(n-1)}{2}, j \in 1, \dots, n_a$ . Τις άλλες σταθερές τις δεχόμαστε ως:  $\tau\alpha 1/\sqrt{2\sqrt{n}}$ ,  $\tau'\alpha 1/\sqrt{2n}$  και  $\beta \approx 5^\circ$ .

Οι αντικειμενικές μεταβλητές  $\bar{x}$  τώρα αλλάζουν προσθέτοντας το  $\bar{\Delta x}$  που επιλέχθηκε από μία  $n$ -διάστατη κανονική κατανομή με πίνακα συνδιακύμανσης  $C'$ . Το  $C'$  στον τύπο είναι το παλιό  $C$  μετά τη μετάλλαξη των  $\alpha$  μεταβλητών. Το  $\sigma_i$  αλλάζει με τον ίδιο τρόπο όπως πριν: με τον πολλαπλασιασμό μίας λογαριθμική μεταβλητή. Οι  $\alpha_j$  αλλάζουν με μία πρόσθετη κανονικά κατανεμημένη μεταβλητή, όμοια με την μετάλλαξη των αντικειμενικών μεταβλητών.

Επίσης έχουμε έναν κανόνα ορίου για τις  $\alpha_j$  μεταβλητές. Οι γωνίες περιστροφής θα πρέπει να είναι στο διάστημα  $[-\pi, \pi]$ , έτσι ώστε η νέα μεταβλητή να σχεδιάζεται κυκλικά μέσα στο εφικτό διάστημα:

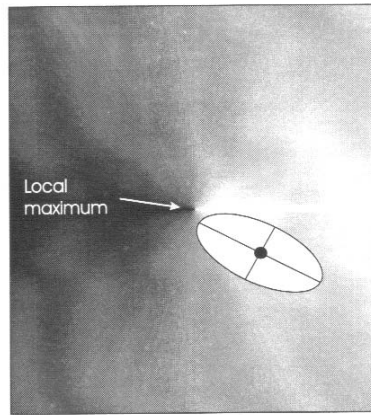
$$|\alpha'_j| > \pi \Rightarrow |\alpha'_j| = |\alpha'_j| - 2\pi \text{sign}(\alpha'_j).$$

Το σχήμα 3.4 δείχνει τα αποτελέσματα των συσχετισμένων μεταλλάξεων σε δύο διαστάσεις. Τα άτομα τώρα έχουν την μορφή  $\langle x, y, \sigma_x, \sigma_y, \alpha_{x,y} \rangle$  και τα σημεία στο χώρο αναζήτησης που οι απόγονοι μπορούν να τοποθετηθούν με μία δοσμένη πιθανότητα σχηματίζουν μία περιστρεφόμενη έλλειψη, όπου ξανά τα μήκη των αξόνων είναι ανάλογα με τις  $\sigma$  μεταβλητές.

Ο πίνακας 3.2 συνοψίζει τρεις πιθανές τοποθετήσεις για τη μετάλλαξη στις εξελικτικές στρατηγικές σύμφωνα με το μήκος και την δομή των ατόμων.

## 3.5 Ανασυνδιασμός

Ο ανασυνδιασμός στις εξελικτικές στρατηγικές περιλαμβάνει δύο γονείς που δημιουργούν ένα παιδί. Για να εξασφαλιστούν  $\lambda$  απόγονοι ο ανασυνδιασμός εκτελείται  $\lambda$  φορές. Υπάρχουν



**Σχήμα 3.4:** Μετάλλαξη με  $n = 2$ ,  $n_\sigma = 2$ ,  $n_\alpha = 1$ . Κομμάτι από ένα μέρος της καταλληλότητας με ένα κωνικό σχήμα να φαίνεται. Το μαύρο δεν υποδεικνύει ένα άτομο. Τα σημεία όπου οι απόγονοι μπορούν να διατεθούν με μία δοσμένη πιθανότητα σχηματίζουν μία περιστρεφόμενη έλλειψη. Η πιθανότητα παραγωγής μίας κίνησης προς την διεύθυνση μίας απότομης ανάβασης (μεγαλύτερη επίδραση της καταλληλότητας) είναι μεγαλύτερη από αυτές άλλων διευθύνσεων.

$n_\sigma$	$n_\alpha$	Δομή ατόμων	Παρατήρηση
1	0	$\langle x_1, \dots, x_n, \sigma \rangle$	Σταθερή μετάλλαξη
$n$	0	$\langle x_1, \dots, x_n, \sigma_1, \dots, \sigma_n \rangle$	Σταθερές μεταλλάξεις
$n$	$n(n-1)/2$	$\langle x_1, \dots, x_n, \sigma_1, \dots, \sigma_n, \alpha_1, \dots, \alpha_{n(n-1)/2} \rangle$	Συσχετισμένες μεταλλάξεις

**Πίνακας 3.2:** Μερικές πιθανές τοποθετήσεις του  $n_\sigma$  και του  $n_\alpha$  για διάφορους τελεστές μεταλλάξης

δύο είδη ανασυνδιασμού που διακρίνονται από το είδος των ανασυνδιασμένων αλληλουχιών στους γονείς. Χρησιμοποιώντας τον **διακριτό ανασυνδιασμό** επιλέγεται τυχαία μία από τις αλληλουχίες των γονέων με ίσες πιθανότητες για καθένα από τους δύο γονείς. Στον **ενδιάμεσο ανασυνδιασμό** οι μεταβλητές των αλληλουχιών των γονέων έχουν υπολογιστεί. Δίνοντας δύο διανύσματα γονέων  $\bar{x}$  και  $\bar{y}$ , δημιουργείται ένα παιδί  $\bar{z}$ , όπου

$$z_i = \begin{cases} (x_i + y_i)/2 & \text{ενδιάμεσος ανασυνδιασμός} \\ x_i \text{ ή } y_i & \text{επιλέγεται τυχαία, διακριτός ανασυνδιασμός} \end{cases}$$

για όλα τα  $i \in 1, \dots, n$ .

Μία προέκταση αυτού επιτρέπει την χρήση περισσότερων από δύο ανασυνδιασμών, επειδή οι δύο γονείς  $\bar{x}$  και  $\bar{y}$  επιλέχθηκαν τυχαία για κάθε θέση  $i \in 1, \dots, n$  στον εκ νέου απόγονο. Αυτές οι επιλογές λαμβάνουν όλο το πληθυσμό των  $\mu$  ατόμων και το αποτέλεσμα είναι ένας τελεστής ανασυνδιασμού με παραπάνω από δύο άτομα να συνεισφέρουν στον απόγονο. Ωστόσο ο ακριβής αριθμός γονέων δεν μπορεί να καθοριστεί. Αυτή η μεταβλητή πολυ-γονέων ονομάζεται **γενικός ανασυνδιασμός**. Για να γίνει σαφής η ορολογία, η αρχική μεταβλητή ονομάζεται **τοπικός ανασυνδιασμός**.

Οι εξελικτικές στρατηγικές χρησιμοποιούν γενικό ανασυνδιασμό. Διάφοροι ανασυνδιασμοί χρησιμοποιούνται για το κομμάτι των αντικειμενικών μεταβλητών (ο διακριτός ανασυνδιασμός συνιστάται) και για το κομμάτι των παραμέτρων στρατηγικής (ο ενδιάμεσος ανασυνδιασμός συνιστάται).

### 3.6 Επιλογή γονέων

Η επιλογή γονέων στις εξελικτικές στρατηγικές δεν βασίζεται στις τιμές καταλληλότητας. Όποτε ένας τελεστής ανασυνδιασμού απαιτεί ένα γονέα, επιλέγεται τυχαία με ομοιόμορφη

κατανομή από τον πληθυσμό των  $\mu$  ατόμων. Θα πρέπει να σημειωθεί ότι η ορολογία των ΕΣ αποκλίνει από την ορολογία των ΓΑ στην χρήση της λέξης 'γονέας'. Στις ΕΣ ολόκληρος ο πληθυσμός φαίνεται ως γονέας. Αντίθετα, στους ΓΑ ο όρος 'γονέας' δηλώνει ένα μέλος του πληθυσμού που έχει επιλεγεί για να υποβληθεί στην μεταβολή (διασταύρωση ή μετάλλαξη).

### 3.7 Επιλογή επιβίωσης

Μετά την δημιουργία  $\lambda$  απογόνων και τον υπολογισμό της καταλληλότητας τους, οι καλύτεροι  $\mu$  από αυτούς επιλέχθηκαν ντετερμινιστικά, είτε μόνο από τον απόγονο και ονομάζεται  $(\mu, \lambda)$  επιλογή, είτε από την ένωση των γονέων και του απόγονου και ονομάζεται  $(\mu + \lambda)$  επιλογή. Και οι δύο επιλογές  $(\mu, \lambda)$  και  $(\mu + \lambda)$  είναι ντετερμινιστικές και βασίζονται στη σειρά αντί στην τιμή καταλληλότητας.

Η επιλογή που χρησιμοποιείται γενικά στις εξελικτικές στρατηγικές είναι η  $(\mu, \lambda)$  επιλογή, η οποία προτιμάται από την  $(\mu + \lambda)$  επιλογή για τους παρακάτω λόγους:

- Η  $(\mu, \lambda)$  απορρίπτει όλους τους γονείς και συνεπώς είναι ικανή να αφήσει ένα (μικρό) τοπικό βέλτιστο.
- Αν η συνάρτηση καταλληλότητας δεν είναι σταθερή, αλλά αλλάζει με τον χρόνο, η επιλογή  $(\mu + \lambda)$  διατηρεί απαραιτωμένους λύσεις και έτσι δεν είναι ικανή να ακολουθήσει το βέλτιστο που μετακινείται.
- Η επιλογή  $(\mu + \lambda)$  παρεμποδίζει τον μηχανισμό αυτοπροσαρμογής σε σχέση με τις παραμέτρους στρατηγικής να λειτουργήσει αποτελεσματικά, επειδή οι παράμετροι στρατηγικής που δεν είναι σωστά προσαρμοσμένοι μπορεί να επιζήσουν από έναν σχετικά μεγάλο αριθμό γενεών όταν τα άτομα έχουν σχετικά καλές αντικειμενικές μεταβλητές και κακή στρατηγική παραμέτρων. Σε αυτή τη περίπτωση συχνά όλα τα παιδιά θα είναι κακά και οι κακοί παράμετροι της στρατηγικής θα επιβιώσουν.

Η πίεση επιλογής στις εξελικτικές στρατηγικές είναι πολύ υψηλή επειδή το  $\lambda$  είναι μεγαλύτερο από το  $\mu$ . Ο **εξαγοράσιμος χρόνος** (takeover time),  $\tau^*$  του δοσμένου μηχανισμού επιλογής καθορίζεται ως ο αριθμός των γενεών που απαιτείται μέχρι η εφαρμογή της επιλογής να γεμίσει εντελώς τον πληθυσμό με αντίγραφα των καλύτερων ατόμων, δίνοντας αρχικά ένα αντίγραφο. Οι Goldberg και Deb μας έδειξαν ότι  $\tau^* = \frac{\ln \lambda}{\ln(\lambda/\mu)}$ .

Για μία εξελικτική στρατηγική με  $\mu = 15$  και  $\lambda = 100$  αυτό έχει αποτέλεσμα  $\tau^* \approx 2$ . Η ανάλογη επιλογή σε έναν γενετικό αλγόριθμο είναι  $\tau^* = \lambda \ln \lambda$ , καταλήγοντας στο  $\tau^* = 460$  για πληθυσμό μεγέθους  $\lambda = 100$ .

### 3.8 Αυτοπροσαρμογή

Μία από τις κύριες συνεισφορές των εξελικτικών στρατηγικών είναι η αυτοπροσαρμογή. Αυτό το χαρακτηριστικό εισήχθηκε πρώτα στις ΕΣ και τα πλεονεκτήματά του έχουν φανεί από την αναζήτηση των ΕΣ, όχι μόνο για πραγματικές-μεταβλητές αλλά και για δυαδικούς και ακεραίους χώρους αναζήτησης.

Το κεντρικό αίτημα των ΕΣ είναι να λειτουργεί η αυτοπροσαρμογή. Φαίνεται ότι μία ΕΣ με αυτοπροσαρμογή υπερτερεί της ΕΣ χωρίς αυτοπροσαρμογή και υπάρχουν και θεωρητικά αποτελέσματα που το επιβεβαιώνουν [18]. Θεωρητικά και πειραματικά αποτελέσματα μπορούν απλά να συμπληρώσουν το ένα τ' άλλο εάν για μία (απλή) αντικειμενική συνάρτηση  $f : \mathcal{R}^n \rightarrow \mathcal{R}$  μπορεί να υπολογιστεί θεωρητικά το καταλληλότερο βήμα μετάλλασης. Εάν τα δεδομένα που εξασφαλίζονται πειραματικά δείξουν μία καλή αντιστοιχία με τις θεωρητικά παραγόμενες μεταβλητές, τότε μπορούμε να καταλήξουμε ότι η αυτοπροσαρμογή λειτουργεί για να βρει τα πιο κοντινά βέλτιστα βήματα μετάλλασης.

Οι αναγκαίες συνθήκες για την αυτοπροσαρμογή είναι :

1.  $\mu > 1$  έτσι ώστε να παρουσιάζονται διάφορες στρατηγικές
2. Η γενιά ενός απογόνου:  $\lambda > \mu$
3. Μία όχι τόσο ισχυρή πίεση επιλογής (ευριστικά:  $\lambda/\mu = 7$ , π.χ., (15, 100) )
4.  $(\mu, \lambda)$ -επιλογή (για να εγγυηθεί η εξάλειψη των μισο-προσαρμοσμένων ατόμων)
5. Ο ανασυνδιασμός στις παραμέτρους της στρατηγικής (ειδικά ο ενδιάμεσος ανασυνδιασμός)



## Εξελικτικός Προγραμματισμός (Evolutionary Programming)

### 4.1 Σκοπός αυτού του κεφαλαίου

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τον Εξελικτικό Προγραμματισμό (Evolutionary Programming) (ΕΠ), ένα ακόμη μέλος των εξελικτικών υπολογισμών. Ορισμένοι εξελικτικοί υπολογισμοί έχουν έναν αλγόριθμο που μπορεί να αναγνωρισθεί ως 'καθιερωμένος'. Για τον ΕΠ τέτοια καθιερωμένη εκδοχή είναι δύσκολο να καθορισθεί για λόγους που θα αναφερθούν αργότερα σε αυτό το κεφάλαιο. Έτσι η περίληψη του ΕΠ παρουσιάζεται στον πίνακα 4.1.

### 4.2 Παράδειγμα εισαγωγής

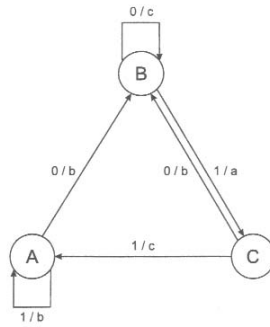
Ο εξελικτικός προγραμματισμός αρχικά αναπτύχθηκε για να προσομοιώσει την εξέλιξη ως μία διαδικασία μάθησης με σκοπό τη παραγωγή τεχνητής νοημοσύνης (artificial intelligence) [5, 37]. Στο κλασικό παράδειγμα του ΕΠ, αναπτύσσονται προβλέψεις στη μορφή των πεπερασμένων μηχανισμών κατάστασης. Ένας πεπερασμένος μηχανισμός καταστάσεων (ΠΜΚ) είναι μία διάταξη μετατροπής που μπορεί να δημιουργηθεί από ένα πεπερασμένο αλφάβητο συμβόλων εισόδου και μπορεί να ανταποκριθεί σε ένα πεπερασμένο αλφάβητο συμβόλων εξόδου. Αποτελείται από έναν αριθμό καταστάσεων  $S$  και έναν αριθμό μεταβάσεων. Οι μεταβάσεις καθορίζουν την δουλειά των ΠΜΚ: ανάλογα με την κατάσταση και το σύμβολο εισόδου, προσδιορίζουν ένα σύμβολο εξόδου και την επόμενη κατάσταση στην οποία θα πάει. Ένα παράδειγμα τριών-καταστάσεων ΠΜΚ φαίνεται στο σχήμα 4.1.

Ο Fogel et al. περιέγραψε ένα πείραμα όπου οι προβλέψεις αναπτύσσονται για να δούμε αν η επόμενη είσοδος (όντας ακέραιος) σε μία ακολουθία είναι κύρια ή όχι. Γι' αυτό το λόγο οι ΠΜΚ χρησιμοποιήθηκαν ως άτομα με αλφάβητο εισόδου  $I = \mathbb{N}$  και αλφάβητο εξόδου  $O = (0, 1)$ . Η καταλληλότητα των ΠΜΚ προσδιορίζεται ως η ακρίβεια πρόβλεψης στην ακολουθία εισόδου των ακεραίων  $1, 2, 3, \dots$ . Η επιλογή γονέων δεν λαμβάνει μέρος, αλλά κάθε ΠΜΚ στο πληθυσμό αλλάζει μία φορά για να παράγει έναν απόγονο. Γενικά υπάρχουν πέντε χρήσιμοι τελεστές μετάλλαξης για να παραχθούν νέοι ΠΜΚ:

- Αλλάζοντας ένα σύμβολο εξόδου

Αναπαράσταση	διανύσματα πραγματικών μεταβλητών
Ανασυνδιασμός	κανένας
Μετάλλαξη	γκουουσιανή διατάραξη
Επιλογή γονέα	ντετερμινιστικά (κάθε γονέας δημιουργεί έναν απόγονο μέσω μετάλλαξης)
Επιλογή επιβίωσης	$(\mu + \mu)$
Ειδικότητα	αυτοπροσαρμογή των βημάτων μετάλλαξης (σε μετά-ΕΠ)

Πίνακας 4.1: Σκίτσο του ΕΠ



**Σχήμα 4.1:** Παράδειγμα πεπερασμένων μηχανισμών κατάστασης που αποτελείται από τρεις καταστάσεις: A, B, C. Το αλφάβητο εισόδου είναι  $I = 0, 1$  και το αλφάβητο εξόδου είναι  $O = a, b, c$ . Η συνάρτηση μετάβασης ΠΜΚ  $d : S \times I \rightarrow S \times O$  μετατρέπει το ρεύμα εισόδου στο ρεύμα εξόδου που προσδιορίζεται από τα βέλη και τις ετικέτες υποδεικνύοντας την είσοδο/έξοδο από την δοσμένη μετάβαση.

- Αλλάζοντας μία μετάβαση (π.χ. αλλάζοντας την επόμενη κατάσταση)
- Προσθέτοντας μία κατάσταση
- Διαγράφοντας μία κατάσταση
- Αλλάζοντας τις αρχικές καταστάσεις

Η επιλογή αυτών των τελεστών μετάλλαξης γίνεται τυχαία από μία ομοιόμορφη κατανομή. Ο ανασυνδιασμός (διασταύρωση) δεν χρησιμοποιείται και αφού έχουμε δημιουργήσει  $\mu$  απόγονους από έναν πληθυσμό των  $\mu$  ΠΜΚ, το πρώτο 50% της ένωσης τους αποθηκεύεται ως την επόμενη γενιά. Αυτή η στρατηγική είναι αρκετά καλή με ακρίβεια περίπου 81%. Ο ΕΠ έχει συνδεθεί με σκοπούς πρόβλεψης και η χρήση των πεπερασμένων μηχανισμών καταστάσεων ως η αναπαράστασή του.

### 4.3 Αναπαράσταση

Ο ΕΠ χρησιμοποιείται για διάφορες εφαρμογές και έχει μία πολύ καλή προσέγγιση επιλέγοντας την σωστή αναπαράσταση που βασίζεται στα χαρακτηριστικά του προβλήματος. Ωστόσο, χρησιμοποιείται πιο συχνά για συναρτήσεις βελτιστοποίησης της μορφής  $f : \mathcal{R}^n \rightarrow \mathcal{R}$  και σε αυτή τη περίπτωση ο εξελικτικός προγραμματισμός χρησιμοποιεί μία αναπαράσταση κινητής υποδιαστολής όπου  $\langle x_1, \dots, x_n \rangle \in \mathcal{R}^n$  είναι η μορφή των ατόμων. Αν και η αυτοπροσαρμογή των παραμέτρων μετάλλαξης αναγνωρίστηκε ως ένα επιπλέον χαρακτηριστικό του αποκαλούμενου μετα-ΕΠ, πλέον είναι μία επιλογή που χρησιμοποιείται συχνά. Επομένως, την συμπεριλαμβάνουμε ως ένα 'καθιερωμένο' χαρακτηριστικό, ισοδυναμώντας τις παραμέτρους στρατηγικής με τα άτομα. Παρομοίως με τις ΕΣ, αυτό οδηγεί στη σχέση

$$\left\langle \underbrace{x_1, \dots, x_n}_{\bar{x}} \underbrace{\sigma_1, \dots, \sigma_n}_{\bar{\sigma}} \right\rangle$$

ως την γενική μορφή των ατόμων στο ΕΠ.

### 4.4 Μετάλλαξη

Εδώ θα υλοποιήσουμε τον τελεστή μετάλλαξης που συνδέεται ευρέως με ότι πάει να γίνει γνωστό ως 'μετά-ΕΠ' - η μεταβλητή της αυτοπροσαρμογής των παραμέτρων της στρατηγικής και η αναπαράσταση των πραγματικών μεταβλητών.



Η μετάλλαξη αλλάζει ένα χρωμόσωμα  $\langle x_1, \dots, x_n, \sigma_1, \dots, \sigma_n \rangle$  σε  $\langle x'_1, \dots, x'_n, \sigma'_1, \dots, \sigma'_n \rangle$ , όπου

$$\begin{aligned} \sigma'_i &= \sigma_i \cdot (1 + \alpha \cdot N(0, 1)), (4.1) \\ x'_i &= x_i + \sigma'_i \cdot N_i(0, 1). \end{aligned}$$

Εδώ το  $N(0, 1)$  δηλώνει το αποτέλεσμα μίας τυχαίας επιλογής από μία γκαουσιανή κανονική κατανομή με αριθμητικό μέσο μηδέν και τυπική απόκλιση 1 και το  $\alpha \approx 0.2$ . Ορίζεται ένας κανόνας ορίου για να παρουσιάσει τις τυπικές αποκλίσεις πολύ κοντά στο μηδέν και ως εκ τούτου να επιτραπούν αποτελεσματικές μεταλλάξεις:

$$\sigma'_i < e_0 \Rightarrow \sigma'_i := e_0. (4.2)$$

Να σημειωθεί ότι έχει προταθεί ένας αριθμός μεταλλάξεων όπου τα βήματα μετάλλαξης είναι αντίστροφα συσχετισμένα με την καταλληλότητα των ατόμων. Από τότε που έχει καθιερωθεί η αυτοπροσαρμογή των βημάτων μετάλλαξης πολλοί τύποι διαφέρουν από αυτόν της εξίσωσης (4.1). Αυτές οι διαφορές αφορούν:

- Τον τύπο για τη μορφοποίηση των βημάτων μετάλλαξης
- Την ενσωμάτωση των *μεταβλητών* αντί των *τυπικών αποκλίσεων* ως τις παραμέτρους στρατηγικής.
- Την σειρά με την οποία οι  $\sigma$  και οι  $x$  μεταβλητές αλλάζουν.

Από αυτά, το πρώτο και το δεύτερο οδηγούν σε προβλήματα στα οποία συχνά παράγονται αρνητικές μεταβλητές για την μεταβλητή των απογόνων. Γι' αυτό το λόγο ο κανόνας ορίου της εξίσωσης (4.2) καλείται συχνά να εκτελεστεί. Εφόσον η γκαουσιανή κατανομή είναι συμμετρική κοντά στον αριθμητικό της μέσο, τα σημάδια των αρνητικών τυπικών αποκλίσεων μπορούν να αντιστραφούν.

Η τρίτη κουκίδα είναι η πιο αξιόλογη, καθώς αλλάζοντας αντίστροφα τις  $\sigma$  και τις  $x$  μεταβλητές θα καταστρεφόταν το σκεπτικό που εξηγήθηκε σε προηγούμενο κεφάλαιο (3.4) μετά τις εξισώσεις (3.2) και (3.4). Οι συγγραφείς Gehlhaar και Fogel συγκρίνουν τις 'πρώτες σίγμα' και τις 'τελευταίες σίγμα' στρατηγικές και καταλήγουν ότι το πρώτο προσφέρει ένα γενικό πλεονέκτημα στο δεύτερο [40].

Όπως αναφέραμε στο κεφάλαιο 3.8 η αυτοπροσαρμογή των παραμέτρων της στρατηγικής βασίζεται στη σχέση μεταξύ καλών στρατηγικών και καλών λύσεων προκειμένου να αναπτυχθούν κατάλληλοι τελεστές παραμέτρων. Μία προοπτική χρησιμοποιείται για να εξηγήσει το αποτέλεσμα της  $(\mu, \lambda)$  επιλογής, όπου  $\lambda \gg \mu$ , είναι αυτό που οι  $\lambda/\mu$  στρατηγικές αναζήτησης προσπαθούν να ξεκινήσουν από κάθε γονέα. Επιπλέον ο Yao υποστηρίζει ότι βελτιώθηκαν γρήγορα οι αλγόριθμοι εξελικτικού προγραμματισμού καθώς δύο απόγονοι δημιουργούνται από κάθε γονέα, ο ένας χρησιμοποιεί την γκαουσιανή κατανομή για να παράγει τυχαίες μεταλλάξεις και ο άλλος χρησιμοποιεί την κατανομή Cauchy [17]. Το τελευταίο έχει πλούσια ουρά το οποίο δίνει στον γενικό αλγόριθμο την ευκαιρία να ξεφύγει από τοπικά ελάχιστα.

## 4.5 Ανασυνδιασμός

Ο ανασυνδιασμός στον ΕΠ δεν συνηθίζεται και δεν έχει νόημα αφού δεν μπορεί να εφαρμοστεί σε διάφορα είδη. Τεχνικά είναι πιθανό να σχεδιαστεί και να εφαρμοστεί σε τελεστές μετάλλαξης ενώνοντας πληροφορίες από δύο ή περισσότερα άτομα [10]. Οι Fogel και Atmar σύγκριναν τα αποτελέσματα των αλγορίθμων του ΕΠ με και χωρίς ανασυνδιασμό με μία σειρά από γραμμικές συναρτήσεις με επαναλήψεις μεταξύ των γονιδίων [5]. Καταλήξαν στο ότι η απόδοση βελτιώθηκε χωρίς ανασυνδιασμό.

Τα τελευταία αποτελέσματα επιβεβαιώνουν ότι η ικανότητα της διασταύρωσης ή της μετάλλαξης Gaussian για να παραχθούν νέοι απόγονοι υψηλότερης καταλληλότητας από τους γονείς τους εξαρτώνται σημαντικά από την διαδικασία αναζήτησης, με την μετάλλαξη να είναι καλύτερη αρχικά αλλά τη διασταύρωση να κερδίζει όσο προχωράει η διαδικασία.

## 4.6 Επιλογή γονέων

Στον ΕΠ κάθε μέλος του πληθυσμού δημιουργεί ακριβώς ένα απόγονο μέσω μεταλλάξεων. Με αυτό τον τρόπο διαφέρει από τους ΓΑ και ΓΠ, όπου η επιλογή βασίζεται στην καταλληλότητα που εφαρμόζεται σε αυτό το στάδιο [15]. Επίσης διαφέρει από τις ΕΣ, καθώς η επιλογή γονέων στον ΕΠ είναι ντετερμινιστική, ενώ στις ΕΣ είναι στοχαστική.

## 4.7 Επιλογή επιβίωσης

Ο τελεστής επιλογής είναι γενικά η επιλογή  $(\mu + \mu)$ . Διαγωνισμοί ζευγαριών διατηρούνται σε σχήμα κυκλικής επαναφοράς (round-robin) περιλαμβάνοντας και τους γονείς και τους πληθυσμούς απογόνων. Κάθε λύση  $\bar{a} \in P(t) \cup P'(t)$  αξιολογείται έναντι άλλων τυχαίων λύσεων  $q$  που επιλέχθηκαν. Για κάθε σύγκριση, μία 'νίκη' αποδίδεται εάν το  $\bar{a}$  είναι καλύτερο. Οι  $\mu$  λύσεις με τον μεγαλύτερο αριθμό νικών εξασφαλίζονται να είναι οι γονείς της επόμενης γενιάς. Συνιστάται  $q = 10$ . Αξίζει να σημειώσουμε ότι αυτή η μεταβλητή επιλογής επιτρέπει τις λιγότερο κατάλληλες λύσεις να επιβιώσουν στη γενιά εάν είναι τυχερές όσον αφορά την επιλογή αντιπάλων.

## Γενετικός Προγραμματισμός (Genetic Programming)

### 5.1 Σκοπός αυτού του κεφαλαίου

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τον Γενετικό Προγραμματισμό (Genetic Programming) (ΓΠ), το πιο νέο μέλος της οικογένειας των εξελικτικών αλγορίθμων. Εκτός από την αναπαράσταση (καθώς χρησιμοποιεί δένδρα ως χρωμοσώματα), διαφέρει από τους άλλους εξελικτικούς αλγορίθμους και στην εφαρμογή. Ενώ οι εξελικτικοί αλγόριθμοι που αναφέρθηκαν μέχρι τώρα εφαρμόζονται σε προβλήματα βελτιστοποίησης, ο ΓΠ μπορεί να εφαρμοστεί σε μηχανή μάθησης. Ο ΓΠ ψάχνει πρότυπα με υψηλή καταλληλότητα. Όμως τα πρότυπα αυτά μπορούμε να τα θεωρήσουμε ως περιπτώσεις βελτιστοποίησης. Αυτό είναι και η βάση για τον ΓΠ καθώς τα πρότυπα τα θεωρούμε ως άτομα και την καταλληλότητά τους ως την ποιότητα του πρότυπου που πρέπει να μεγιστοποιηθεί. Η περίληψη του ΓΠ δίνεται στον πίνακα 5.1.

### 5.2 Παράδειγμα εισαγωγής

Ως παράδειγμα θα αναφερθεί το πρόβλημα μίας τράπεζας η οποία παρατηρεί πώς οι πελάτες πληρώνουν τα δάνειά τους. Μπορούν να χρησιμοποιηθούν πληροφορίες σχετικά με τους πελάτες για να αναπτυχθεί ένα πρότυπο το οποίο περιγράφει καλούς και κακούς πελάτες. Αυτό το πρότυπο μπορεί να χρησιμοποιηθεί για να προβλεφθεί η συμπεριφορά των πελατών. Το πρότυπο δημιουργείται βασισμένο σε (ιστορικά) δεδομένα που προέρχονται από προσωπικές πληροφορίες των πελατών. Χρησιμοποιεί αυτά τα προσωπικά δεδομένα ως είσοδο και παράγει μία δυαδική έξοδο, αναπαριστώντας την προβλεπόμενη αξία του αντίστοιχου προσώπου. Για παράδειγμα, ο ετήσιος μισθός, η οικογενειακή κατάσταση και ο αριθμός των παιδιών μπορούν να χρησιμοποιηθούν ως είσοδοι. Ο πίνακας 5.2 δείχνει ένα μικρό σετ δεδομένων. Ένα πιθανό πρότυπο που χρησιμοποιεί αυτά τα δεδομένα μπορεί να είναι το ακόλουθο:

ΕΑΝ (ΑΡΙΘΜΟΣ.ΠΑΙΔΙΩΝ= 2) ΚΑΙ (ΜΙΣΘΟΣ> 80000) ΤΟΤΕ καλό ΑΛΛΙΩΣ κακό (5.1)

Γενικά, το πρότυπο θα δείχνει κάπως έτσι:

ΕΑΝ τύπος ΤΟΤΕ καλό ΑΛΛΙΩΣ κακό

Σημειώνουμε ότι ο *τύπος* είναι ο μόνος άγνωστος και όλα τα άλλα στοιχεία είναι σταθερά. Στόχος μας είναι να βρούμε τον βέλτιστο τύπο.

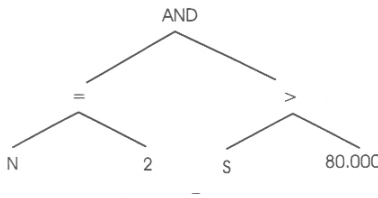
Αναπαράσταση	δομές δένδρων
Ανασυνδιασμός	ανταλλαγή υποδένδρων
Μετάλλαξη	τυχαία αλλαγή στα δένδρα
Επιλογή γονέα	αναλογική καταλληλότητα
Επιλογή επιβίωσης	αντικατάσταση γενεών

Πίνακας 5.1: Σκίτσο του ΓΠ

Id Πελάτη	Αριθμός παιδιών	Μισθός	Οικογενειακή κατάσταση	Πιστοληπτική ικανότητα
$Id - 1$	2	45.000	Παντρεμένος	0
$Id - 2$	0	30.000	Ανύπαντρος	1
$Id - 3$	1	40.000	Παντρεμένος	1
$Id - 4$	2	60.000	Διαζευγμένος	1
...	...	...	...	...
$Id - 10000$	2	50.000	Παντρεμένος	1

**Πίνακας 5.2:** Δεδομένα για το πρόβλημα της τράπεζας

Σε αυτό το σημείο έχουμε μορφοποιήσει το πρόβλημά μας ως ένα πρόβλημα αναζήτησης στον χώρο των πιθανών τύπων, όπου η ποιότητα ενός τύπου  $F$  μπορεί να προσδιοριστεί ως το ποσοστό των σωστών πελατών που ταξινομούνται από το πρότυπο ΕΑΝ  $F$  ΤΟΤΕ καλό ΑΛΛΙΩΣ κακό. Έτσι έχουμε προσδιορίσει τους φαινότυπους (τύποι) και την καταλληλότητα (ακρίβεια της ταξινόμησης). Σύμφωνα με τον ΓΠ χρησιμοποιούμε αναλυτικά δένδρα ως γονότυπους οι οποίοι αναπαριστώνται από τους τύπους. Το σχήμα 5.1 δείχνει το αναλυτικό δένδρο του τύπου της εξίσωσης (5.1).



**Σχήμα 5.1:** Αναλυτικό δένδρο

Αυτή η αναπαράσταση διαφέρει από αυτές που χρησιμοποιούνται στους ΓΑ ή στις ΕΣ για δύο σημαντικούς λόγους:

- Τα χρωμοσώματα είναι μη γραμμικές δομές, ενώ στους ΓΑ και στις ΕΣ είναι γραμμικά διανύσματα του τύπου  $\bar{v} \in D_1 \times \dots \times D_n$ , όπου  $D_i$  είναι το πεδίο ορισμού του  $v_i$ .
- Τα χρωμοσώματα μπορεί να διαφέρουν στο μέγεθος, μετρούνται από τον αριθμό των κόμβων στο δένδρο, ενώ στους ΓΑ και στις ΕΣ το μέγεθος τους, δηλαδή το μήκος των χρωμοσωμάτων  $n$ , είναι συνήθως σταθερό.

Αυτός ο νέος τύπος επιβάλλει νέους τελεστές κατάλληλους για δένδρα. Τέτοιοι τελεστές διασαύρωσης και μετάλλαξης αναφέρονται παρακάτω (τμήματα 5.4, 5.5). Όσον αφορά την επιλογή, να σημειώσουμε ότι βασίζεται μόνο στην καταλληλότητα και επομένως είναι ανεξάρτητη από την δομή χρωμοσωμάτων. Γι' αυτό το λόγο, μπορεί να εφαρμοστεί οποιαδήποτε γνωστή επιλογή από τους άλλους ΕΑ, π.χ. η επιλογή  $(\mu, \lambda)$  ή η αναλογική καταλληλότητα.

### 5.3 Αναπαράσταση

Η γενική ιδέα του ΓΠ είναι να χρησιμοποιούμε αναλυτικά δένδρα ως χρωμοσώματα. Τέτοια αναλυτικά δένδρα δημιουργούν εκφράσεις από ένα τύπο που δίνεται. Ανάλογα με το πρόβλημα και την αντίληψη του χρήστη οι εκφράσεις μπορεί να είναι αριθμητικές, λογικές ή κάποιο πρόγραμμα. Ένα παράδειγμα αυτών των εκφράσεων δίνεται παρακάτω:

- Ένας αριθμητικός τύπος:

$$2p + ((x + 3) - \frac{y}{5+1}), (5.2)$$

- Ένας λογικός τύπος:

$$(x \wedge true) \rightarrow ((x \vee y) \vee (z \leftrightarrow (x \wedge y))), (5.3),$$

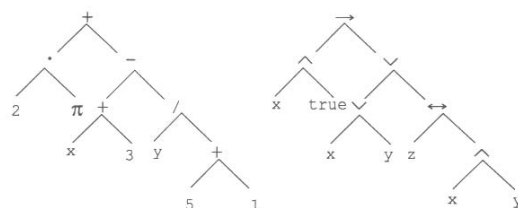
- Το παρακάτω πρόγραμμα:

```

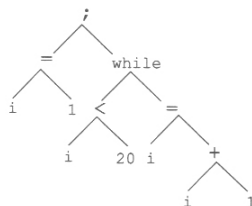
i = 1;
while (i < 20)
{
  i = i + 1
}

```

Τα σχήματα 5.2 και 5.3 δείχνουν τα αναλυτικά δένδρα που ανήκουν σε αυτές τις εκφράσεις.



**Σχήμα 5.2:** Αναλυτικό δένδρο που ανήκει στις σχέσεις (5.2) (αριστερά) και (5.3) (δεξιά)



**Σχήμα 5.3:** Αναλυτικό δένδρο που ανήκει στο παραπάνω πρόγραμμα

Αυτά τα παραδείγματα υλοποιούν γενικά πως τα αναλυτικά δένδρα μπορούν να χρησιμοποιηθούν και να ερμηνευθούν. Ανάλογα με την ερμηνεία ο ΓΠ μπορεί να οριστεί με διάφορους τρόπους. Ο ΓΠ απλά λειτουργεί με διαφορετικά δεδομένα δομής: τα χρωμοσώματα είναι δένδρα.

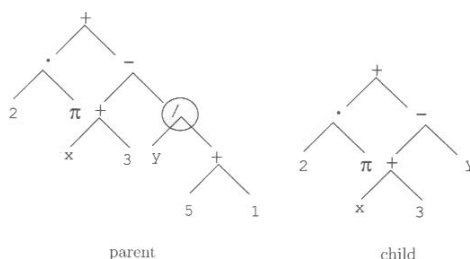
Το πώς θα αναπαραστήσουμε τα άτομα στους ΓΑ καθορίζεται από τη σύνταξη των δένδρων ή την σύνταξη των εκφράσεων (*s*-εκφράσεις). Αυτό γίνεται συχνά από τον προσδιορισμό ενός **σει συνάρτησης** και ενός **τελικού σει**. Τα στοιχεία του τελικού σει τα θεωρούμε ως τα φύλλα, ενώ τα σύμβολα από το σει συνάρτησης είναι οι εσωτερικοί κόμβοι [1]. Για παράδειγμα, μία κατάλληλη συνάρτηση και ένα τελικό σει της έκφρασης της εξίσωσης (5.2) δίνεται στον πίνακα 5.3.

Σει συνάρτησης	$+, -, \cdot, /$
Τελικό σει	$\mathbb{R} \cup x, y$

**Πίνακας 5.3:** Το σει συνάρτησης και το τελικό σει της έκφρασης της εξίσωσης (5.2)

## 5.4 Μετάλλαξη

Η μετάλλαξη στο ΓΠ είναι η ίδια όπως σε όλους τους αλγορίθμους του εξελικτικού υπολογισμού, δηλαδή, δημιουργείται ένα νέο άτομο από ένα παλιό μέσω μικρών τυχαίων μεταβολών. Η πιο συχνή υλοποίηση λειτουργεί αντικαθιστώντας το υποδένδρο ξεκινώντας σε ένα τυχαίο κόμβο από ένα τυχαία παραγόμενο δένδρο. Το καινούριο δένδρο που δημιουργήθηκε συνήθως παράγεται με τον ίδιο τρόπο όπως ο αρχικός πληθυσμός [30]. Το σχήμα 5.4 υλοποιεί πώς τα αναλυτικά δένδρα που ανήκουν στην εξίσωση (5.2) (αριστερά) και μεταλλάσσονται σε ένα αναλυτικό δένδρο που αναπαριστάται από  $2 \cdot p + ((x + 3) - y)$ . Να σημειώσουμε ότι το μέγεθος (βάθος δένδρου) του παιδιού μπορεί να υπερβαίνει αυτό του δένδρου του γονέα.



**Σχήμα 5.4:** Η μετάλλαξη ΓΠ υλοποιείται: Ο κόμβος που σχεδιάστηκε από ένα κύκλο στο αριστερό δένδρο επιλέχθηκε για μετάλλαξη. Το υποδένδρο που ξεκινάει από αυτόν τον κόμβο αντικαθιστάται από ένα τυχαία παραγόμενο δένδρο, το οποίο είναι ένα φύλλο εδώ.

Η μετάλλαξη στο ΓΠ έχει δύο παραμέτρους:

- Τη πιθανότητα να επιλεγεί για μετάλλαξη ο κόμβος με τον ανασυνδιασμό
- Τη πιθανότητα να επιλεγεί ένα εσωτερικό σημείο, με τον γονέα να είναι ρίζα του υποδένδρου που αντικαταστάθηκε

## 5.5 Ανασυνδιασμός

Ο ανασυνδιασμός στο ΓΠ δημιουργεί απόγονο ανταλλάσσοντας γενετικό υλικό μεταξύ των επιλεγμένων γονέων. Δηλαδή είναι ένας δυαδικός τελεστής που δημιουργεί δύο δένδρα παιδιά από δύο δένδρα γονείς. Η πιο συχνή υλοποίηση είναι το **υποδένδρο διασταύρωσης**, το οποίο λειτουργεί ανταλλάζοντας τα υποδένδρα ξεκινώντας από δύο τυχαία επιλεγμένους κόμβους από τους γονείς που δίνονται. Να σημειώσουμε ότι το μέγεθος (βάθος δένδρου) των παιδιών μπορεί να υπερβεί αυτό των δένδρων των γονέων [35].

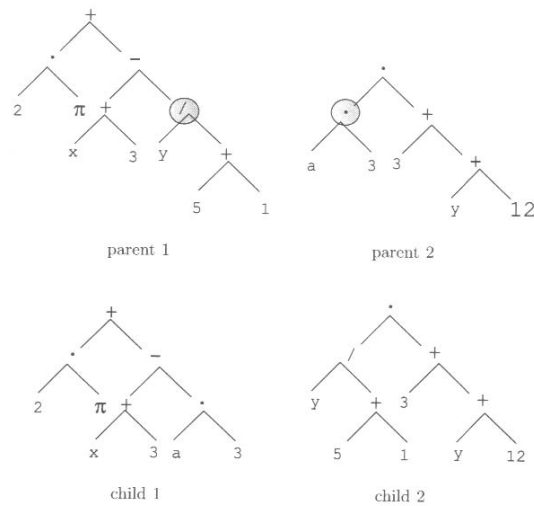
Κατά τη διάρκεια της εξέλιξης του ΓΠ δημιουργούνται πολλοί τελεστές διασταύρωσης [28]. Ο πιο συχνά χρησιμοποιούμενος, το υποδένδρο διασταύρωσης, υλοποιείται στο σχήμα 5.5.

Ο ανασυνδιασμός στο ΓΠ έχει δύο παραμέτρους:

- Τη πιθανότητα να επιλεγεί ανασυνδιασμός στον κόμβο με μετάλλαξη
- Τη πιθανότητα να επιλεγεί ένα εσωτερικό σημείο με κάθε γονέα ως σημείο διασταύρωσης

## 5.6 Επιλογή γονέων

Ο ΓΠ χρησιμοποιεί την επιλογή αναλογικής καταλληλότητας. Ωστόσο, επειδή συχνά έχουμε μεγάλο μέγεθος του πληθυσμού, χρησιμοποιείται μία μέθοδος που ονομάζεται **υπέρ-επιλογή** για μεγέθη πληθυσμού από 1000 και πάνω.



**Σχήμα 5.5:** Η διασταύρωση του ΓΠ υλοποιείται: οι κόμβοι που σχεδιάζονται από έναν κύκλο στα γονικά δένδρα επιλέγονται ως σημεία διασταύρωσης. Τα υποδένδρα που ξεκινούν από αυτούς τους κόμβους αντιστρέφονται, με αποτέλεσμα να έχουμε δύο νέα δένδρα, τα οποία είναι τα παιδιά.

Σε αυτή τη μέθοδο, ταξινομείται πρώτα ο πληθυσμός από την καταλληλότητα και μετά χωρίζεται σε δύο ομάδες, η μία περιλαμβάνει το πρώτο  $x\%$  και η άλλη περιλαμβάνει το άλλο  $(100 - x)\%$ . Όταν επιλέγονται οι γονείς, το 80% της επιλογής πρόερχεται από τη πρώτη ομάδα και το άλλο 20% από την δεύτερη [29]. Οι μεταβλητές  $x$  που χρησιμοποιούνται βρίσκονται εμπειρικά από τον 'κανόνα των φυλλομετρητών' και εξαρτώνται από το μέγεθος του πληθυσμού (πίνακας 5.4).

## 5.7 Επιλογή επιβίωσης

Ο ΓΠ χρησιμοποιεί μία γενική στρατηγική χωρίς ελιτισμό, π.χ. ο αριθμός των απογόνων που δημιουργήθηκαν είναι ο ίδιος με το μέγεθος του πληθυσμού και όλα τα άτομα έχουν διάστημα ζωής μίας γενιάς. Στο βιβλίο του το 1998, ο Banzhaf et al. έδωσε ισοδύναμη χρήση των γενεών και της σταθερής-κατάστασης του ΓΠ και πλέον φαίνεται να χρησιμοποιείται η σταθερή-κατάσταση.

## 5.8 Αρχικοποίηση

Η πιο συχνή μέθοδος αρχικοποίησης για το ΓΠ είναι αυτή που ονομάζεται μέθοδος ramped half-and-half. Σε αυτή τη μέθοδο επιλέγεται ένα μέγιστο αρχικό βάθος  $D_{max}$  των δένδρων και έπειτα κάθε μέλος του αρχικού πληθυσμού δημιουργείται από τα σετ των συναρτήσεων  $F$  και των τερματικών  $T$  χρησιμοποιώντας μία από τις παρακάτω δύο μεθόδους με ίση πιθανότητα:

Μέγεθος πληθυσμού	Αναλογία του πληθυσμού σε καταλληλότερα γκρουπ ( $x$ )
1000	32%
2000	16%
4000	8%
8000	4%

**Πίνακας 5.4:** Αναλογία του ταξινομημένου πληθυσμού σε 'καταλληλότερο' υποπληθυσμό από τον οποίο επιλέχθηκε η πλειονότητα των γονέων

- *Ολοκληρωμένη μέθοδος*: εδώ κάθε κλαδί από το δένδρο έχει βάθος  $D_{max}$ . Τα περιεχόμενα των κόμβων σε βάθος  $d$  επιλέχθηκαν από το  $F$  εάν  $d < D_{max}$  ή από το  $T$  εάν  $d = D_{max}$ .
- *Αναπτυσσόμενη μέθοδος*: εδώ τα κλαδιά των δένδρων μπορεί να έχουν διαφορετικά βάθη, μέχρι το  $D_{max}$ . Το δένδρο κατασκευάστηκε αρχικά από την ρίζα, με τα περιεχόμενα ενός κόμβου να επιλέγονται στοχαστικά από το  $F \cup T$  εάν  $d < D_{max}$ .



**Μέρος ΙΙΙ**

**Αποτελέσματα**



## Διαφοροεξελικτικοί Αλγόριθμοι (Differential Evolution)

### 6.1 Εισαγωγικά

Οι Διαφοροεξελικτικοί Αλγόριθμοι (Differential Evolution) (ΔΑ) είναι μία μέθοδος για αριθμητική βελτιστοποίηση χωρίς τη γνώση του διανύσματος των μερικών παραγών της συνάρτησης που θα βελτιστοποιηθεί. Η μέθοδος των ΔΑ λειτουργεί σε πολυ-διάστατες συναρτήσεις πραγματικών μεταβλητών που δεν είναι απαραίτητα συνεχής ή διαφορίσιμες. Οι ΔΑ βελτιστοποιούν ένα πρόβλημα διατηρώντας ένα πληθυσμό από τις υποψήφιες λύσεις και δημιουργεί νέες υποψήφιες λύσεις συνδυάζοντας αυτές που υπάρχουν ήδη σύμφωνα με απλούς τύπους διασταύρωσης και μετάλλαξης και έπειτα κρατώντας οποιαδήποτε υποψήφια λύση που έχει την καλύτερη καταλληλότητα. Με αυτό τον τρόπο το πρόβλημα βελτιστοποίησης αντιμετωπίζεται ως ένα μαύρο κουτί το οποίο απλώς παρέχει ένα μέτρο ποιότητας που δίνει μία υποψήφια λύση και επομένως η κλίση δεν χρειάζεται [45].

### 6.2 Ο αλγόριθμος

Η συνάρτηση καταλληλότητας ή κόστους η οποία πρέπει να ελαχιστοποιηθεί είναι:  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . Το  $x \in \mathbb{R}^n$  προσδιορίζει μία υποψήφια λύση (και από εδώ και πέρα ονομάζεται άτομο) στον πληθυσμό. Ο βασικός αλγόριθμος των ΔΑ μπορεί να περιγραφεί ως εξής:

- Αρχικοποιούμε όλα τα άτομα  $x$  με τυχαίες θέσεις στο χώρο αναζήτησης.
- Μέχρι να βρεθεί ένα κριτήριο τερματισμού (π.χ. ο αριθμός των επαναλήψεων που παρουσιάστηκαν ή να επιτευχθεί η επαρκή καταλληλότητα), επανέλαβε τα ακόλουθα:
- Για κάθε άτομο  $x$  κάνε:
  - Διάλεξε τρία άτομα  $a, b, c$  τυχαία, διαφορετικά μεταξύ τους όπως επίσης και από το άτομο  $x$ .
  - Διάλεξε ένα τυχαίο ακέραιο  $r \in 1, \dots, n$  όπου  $n$  είναι η διαστατικότητα του προβλήματος που θα βελτιστοποιηθεί.
  - Υπολόγισε τα άτομα στη νέα θέση  $y = [y_1, \dots, y_n]$  επαναλαμβάνοντας για κάθε  $i \in 1, \dots, n$  τα ακόλουθα:
    1. Διάλεξε  $r_i \in \bigcup(0, 1)$  ομοιόμορφα από το ανοιχτό διάστημα  $(0, 1)$
    2. Εάν  $(i = r)$  ή  $r_i < Cr$  άφησε  $y_i = \alpha_i + F \cdot (b_i - c_i)$ , αλλιώς άφησε  $y_i = x_i$
    3. Εάν  $(f(y) < f(x))$  τότε αντικατέστησε το άτομο στον πληθυσμό με την βελτιωμένη υποψήφια λύση, δηλαδή, θέσε  $x = y$  στον πληθυσμό.

- Διάλεξε το άτομο από τον πληθυσμό το οποίο έχει τη καλύτερη καταλληλότητα και επέστρεψε το ως την καλύτερη υποψήφια λύση που βρέθηκε.

Να σημειωθεί ότι  $F \in [0, 2]$  ονομάζεται πιθανότητα μετάλλαξης και το  $Cr \in [0, 1]$  ονομάζεται η πιθανότητα διασταύρωσης. Και οι δύο αυτοί παράμετροι επιλέγονται από τους χρήστες μαζί με το μέγεθος του πληθυσμού  $NP > 3$ .

### 6.3 Ρύθμιση παραμέτρων

Ένας απλός και αποτελεσματικός τρόπος της ρύθμισης των παραμέτρων των ΔΑ  $F$ ,  $Cr$  και  $NP$  παρουσιάστηκε από τον Pedersen ο οποίος έκανε πολυάριθμα πειράματα με διαφορετικά προβλήματα βελτιστοποίησης και ρυθμίσεις. Η τεχνική ονομάζεται *μετά-βελτιστοποίηση* διότι μία άλλη μέθοδος βελτιστοποίησης χρησιμοποιείται για να ρυθμίσει τις παραμέτρους των ΔΑ. Μία υλοποίηση μπορεί να βρεθεί στη βιβλιοθήκη του κώδικα πηγής SwarmOps [46].

### 6.4 Μεταβλητές

Οι μεταβλητές στον βασικό αλγόριθμο των ΔΑ συνεχώς εξελίσσονται σε μία προσπάθεια βελτίωσης του πληθυσμού. Μία δημοφιλής τάση αναζήτησης ήταν να επινοήσουν σχήματα για την διατάραξη ή την προσαρμογή των παραμέτρων των ΔΕ κατά τη διάρκεια της βελτιστοποίησης. Μεγάλο μέρος αυτής της αναζήτησης αμφισβητήθηκε από τον Pedersen ο οποίος βρήκε ότι δεν φαίνεται να υπάρχει γενικό και σταθερό πλεονέκτημα τέτοιων μεταβλητών ΔΑ εάν οι παράμετροι των βασικών ΔΑ ρυθμίζονται κατάλληλα.

### 6.5 Αποτελέσματα

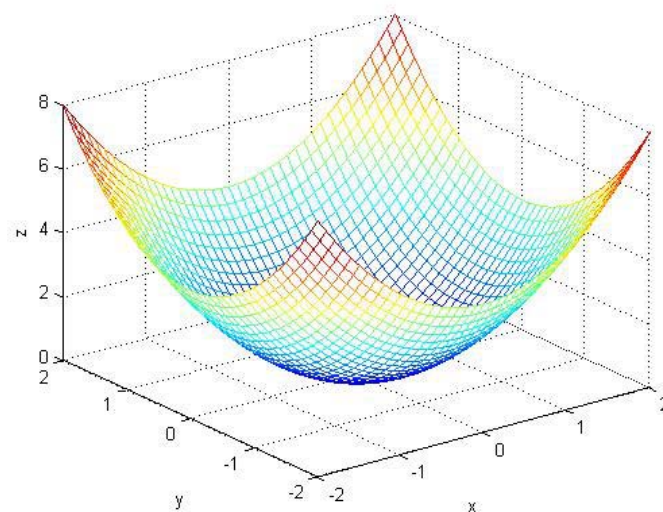
Σε αυτό το σημείο θα αναφερθούμε στα αποτελέσματα που λάβαμε από την κατηγορία των αλγορίθμων των Διαφοροεξελικτικών Αλγορίθμων. Χρησιμοποιήθηκε το περιβάλλον του MATLAB για να εκτελεστεί ο αλγόριθμος, τον οποίο αντήσαμε από το διαδίκτυο [47] και τον χρησιμοποιήσαμε για συγκεκριμένες συναρτήσεις. Οι συναρτήσεις που θα ασχοληθούμε και θα συγκρίνουμε με τις κατηγορίες των αλγορίθμων που ακολουθούν στα επόμενα κεφάλαια είναι η συνάρτηση DeJong, η συνάρτηση Griewank, η συνάρτηση Rastrigrin και η συνάρτηση Rosenbrock. Δηλαδή θα χρησιμοποιήσουμε τον αλγόριθμο των Διαφοροεξελικτικών Αλγορίθμων και κάθε φορά θα ενσωματώνουμε διαφορετική συνάρτηση και διαφορετικές παραμέτρους. Οι συναρτήσεις που χρησιμοποιούμε έχουν ελάχιστο στο 0. Για κάθε συνάρτηση χρησιμοποιήσαμε διάφορους συνδιασμούς των παραμέτρων: μέγεθος του πληθυσμού, διάσταση του προβλήματος, τις επαναλήψεις που έγιναν και το σφάλμα. Επιπλέον σε αυτή τη κατηγορία θα ασχοληθούμε με δύο στρατηγικές, την στρατηγική DE/best/1/exp όπου αυτός ο αλγόριθμος χρησιμοποιεί το καλύτερο άτομο του πληθυσμού και για το λόγο αυτό συγκλίνει γρηγορότερα σε μια λύση του προβλήματος. Και τον αλγόριθμο της στρατηγικής DE/rand/1/exp που χρησιμοποιεί μόνο τυχαία επιλεγμένα άτομα του πληθυσμού, με αποτέλεσμα να ανιχνεύει καλύτερα το χώρο αναζήτησης και να συγκλίνει συνήθως πιο αργά αλλά με καλύτερες λύσεις.

Η μαθηματική συνάρτηση της DeJong δίνεται από τον παρακάτω τύπο και ακολουθεί και η γραφική παράστασή της:  $\sum x_i^2$

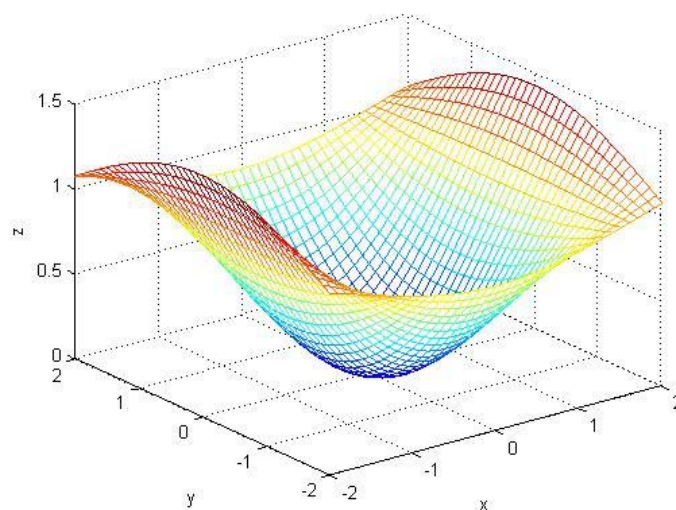
Η μαθηματική συνάρτηση της Griewank δίνεται από τον παρακάτω τύπο και ακολουθεί και η γραφική παράστασή της:  $\sum((x_i^2)/4000) - \text{prod}(\cos(x_i/\text{sqrt}(i))) + 1$

Η μαθηματική συνάρτηση της Rastrigrin δίνεται από τον παρακάτω τύπο και ακολουθεί και η γραφική παράστασή της:  $\sum(x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i) + 10)$

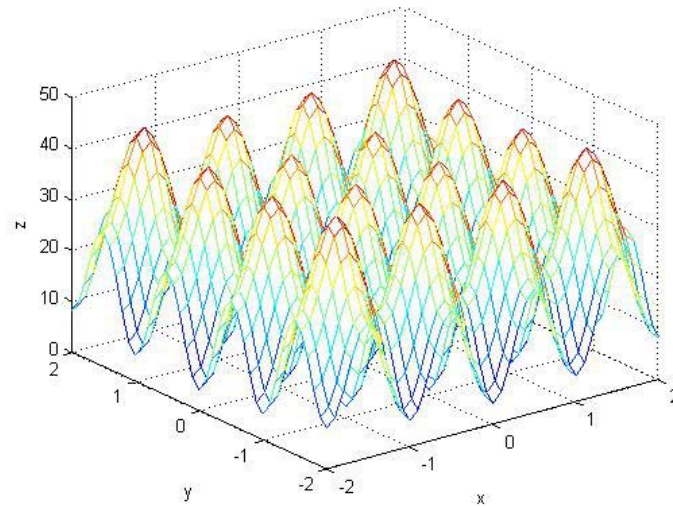
Η μαθηματική συνάρτηση της Rosenbrock δίνεται από τον παρακάτω τύπο και ακολουθεί και η γραφική παράστασή της:  $\sum(100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$



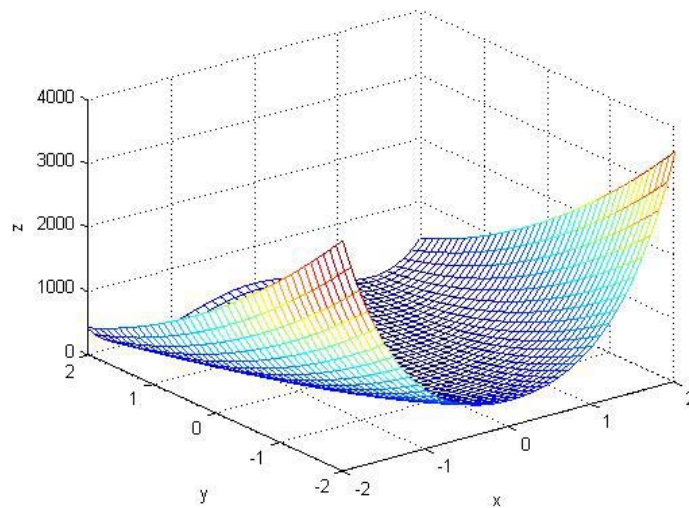
**Σχήμα 6.1:** Γραφική παράσταση της συνάρτησης DeJong



**Σχήμα 6.2:** Γραφική παράσταση της συνάρτησης Griewank



**Σχήμα 6.3:** Γραφική παράσταση της συνάρτησης Rastrigin



**Σχήμα 6.4:** Γραφική παράσταση της συνάρτησης Rosenbrock

Τα αποτελέσματα που παίρνουμε από το περιβάλλον του MATLAB κάθε φορά που εκτελούμε μία συνάρτηση έχουν την παρακάτω μορφή:

Simulation: #1: Best value found:  $5.099481e - 006$  after 420 function evaluations.  
Simulation: #2: Best value found:  $6.399434e - 006$  after 540 function evaluations.  
Simulation: #3: Best value found:  $8.251356e - 007$  after 400 function evaluations.  
Simulation: #4: Best value found:  $9.887138e - 006$  after 440 function evaluations.  
Simulation: #5: Best value found:  $9.505872e - 006$  after 460 function evaluations.  
Simulation: #6: Best value found:  $9.633329e - 006$  after 380 function evaluations.  
Simulation: #7: Best value found:  $3.162240e - 006$  after 520 function evaluations.  
Simulation: #8: Best value found:  $2.654852e - 006$  after 680 function evaluations.  
Simulation: #9: Best value found:  $2.881930e - 006$  after 520 function evaluations.  
Simulation: #10: Best value found:  $2.967709e - 006$  after 480 function evaluations.  
Simulation: #11: Best value found:  $9.272939e - 006$  after 420 function evaluations.  
Simulation: #12: Best value found:  $1.390465e - 006$  after 580 function evaluations.  
Simulation: #13: Best value found:  $3.423762e - 006$  after 540 function evaluations.  
Simulation: #14: Best value found:  $4.009633e - 006$  after 480 function evaluations.  
Simulation: #15: Best value found:  $9.806983e - 006$  after 400 function evaluations.  
Simulation: #16: Best value found:  $6.015351e - 006$  after 400 function evaluations.  
Simulation: #17: Best value found:  $9.194897e - 007$  after 480 function evaluations.  
Simulation: #18: Best value found:  $2.483201e - 006$  after 580 function evaluations.  
Simulation: #19: Best value found:  $7.506094e - 006$  after 600 function evaluations.  
Simulation: #20: Best value found:  $7.753968e - 007$  after 420 function evaluations.  
Simulation: #21: Best value found:  $2.478855e - 006$  after 360 function evaluations.  
Simulation: #22: Best value found:  $8.463201e - 006$  after 500 function evaluations.  
Simulation: #23: Best value found:  $9.443614e - 006$  after 420 function evaluations.  
Simulation: #24: Best value found:  $9.804653e - 006$  after 500 function evaluations.  
Simulation: #25: Best value found:  $9.829469e - 006$  after 540 function evaluations.  
Simulation: #26: Best value found:  $1.230855e - 006$  after 580 function evaluations.  
Simulation: #27: Best value found:  $9.387645e - 006$  after 520 function evaluations.  
Simulation: #28: Best value found:  $5.015829e - 007$  after 540 function evaluations.  
Simulation: #29: Best value found:  $4.156108e - 006$  after 560 function evaluations.  
Simulation: #30: Best value found:  $1.997768e - 006$  after 520 function evaluations.  
Simulation: #31: Best value found:  $5.353549e - 006$  after 540 function evaluations.  
Simulation: #32: Best value found:  $1.265064e - 006$  after 400 function evaluations.  
Simulation: #33: Best value found:  $7.544923e - 006$  after 640 function evaluations.  
Simulation: #34: Best value found:  $7.847610e - 006$  after 520 function evaluations.  
Simulation: #35: Best value found:  $9.899128e - 007$  after 500 function evaluations.  
Simulation: #36: Best value found:  $9.296117e - 006$  after 4520 function evaluations.  
Simulation: #37: Best value found:  $4.661413e - 006$  after 740 function evaluations.  
Simulation: #38: Best value found:  $2.032810e - 006$  after 400 function evaluations.  
Simulation: #39: Best value found:  $5.154117e - 006$  after 400 function evaluations.  
Simulation: #40: Best value found:  $9.686235e - 006$  after 320 function evaluations.  
Simulation: #41: Best value found:  $9.234521e - 006$  after 340 function evaluations.  
Simulation: #42: Best value found:  $8.888019e - 006$  after 500 function evaluations.  
Simulation: #43: Best value found:  $9.414314e - 006$  after 640 function evaluations.  
Simulation: #44: Best value found:  $1.233265e - 006$  after 480 function evaluations.  
Simulation: #45: Best value found:  $2.658232e - 006$  after 400 function evaluations.  
Simulation: #46: Best value found:  $6.489858e - 006$  after 680 function evaluations.  
Simulation: #47: Best value found:  $6.691219e - 006$  after 500 function evaluations.  
Simulation: #48: Best value found:  $4.474191e - 006$  after 540 function evaluations.  
Simulation: #49: Best value found:  $4.020162e - 006$  after 440 function evaluations.

Simulation: #50: Best value found:  $4.148734e - 006$  after 440 function evaluations.  
Simulation: #51: Best value found:  $6.399465e - 006$  after 480 function evaluations.  
Simulation: #52: Best value found:  $3.549906e - 007$  after 560 function evaluations.  
Simulation: #53: Best value found:  $6.068357e - 006$  after 560 function evaluations.  
Simulation: #54: Best value found:  $3.809349e - 006$  after 520 function evaluations.  
Simulation: #55: Best value found:  $2.936714e - 006$  after 420 function evaluations.  
Simulation: #56: Best value found:  $3.619759e - 006$  after 560 function evaluations.  
Simulation: #57: Best value found:  $7.184599e - 006$  after 500 function evaluations.  
Simulation: #58: Best value found:  $3.874230e - 007$  after 380 function evaluations.  
Simulation: #59: Best value found:  $9.351437e - 009$  after 500 function evaluations.  
Simulation: #60: Best value found:  $8.998610e - 006$  after 540 function evaluations.  
Simulation: #61: Best value found:  $7.095511e - 006$  after 680 function evaluations.  
Simulation: #62: Best value found:  $9.987391e - 007$  after 480 function evaluations.  
Simulation: #63: Best value found:  $5.218304e - 006$  after 500 function evaluations.  
Simulation: #64: Best value found:  $9.401707e - 006$  after 560 function evaluations.  
Simulation: #65: Best value found:  $4.941923e - 006$  after 380 function evaluations.  
Simulation: #66: Best value found:  $1.170208e - 006$  after 500 function evaluations.  
Simulation: #67: Best value found:  $9.213281e - 006$  after 400 function evaluations.  
Simulation: #68: Best value found:  $2.801646e - 006$  after 420 function evaluations.  
Simulation: #69: Best value found:  $5.608104e - 006$  after 620 function evaluations.  
Simulation: #70: Best value found:  $7.726182e - 006$  after 520 function evaluations.  
Simulation: #71: Best value found:  $4.052182e - 006$  after 660 function evaluations.  
Simulation: #72: Best value found:  $1.588979e - 006$  after 500 function evaluations.  
Simulation: #73: Best value found:  $3.499468e - 006$  after 560 function evaluations.  
Simulation: #74: Best value found:  $7.624967e - 006$  after 360 function evaluations.  
Simulation: #75: Best value found:  $4.643500e - 006$  after 500 function evaluations.  
Simulation: #76: Best value found:  $6.798376e - 007$  after 480 function evaluations.  
Simulation: #77: Best value found:  $8.211945e - 006$  after 460 function evaluations.  
Simulation: #78: Best value found:  $3.820603e - 006$  after 420 function evaluations.  
Simulation: #79: Best value found:  $3.922379e - 007$  after 520 function evaluations.  
Simulation: #80: Best value found:  $2.412385e - 006$  after 540 function evaluations.  
Simulation: #81: Best value found:  $6.994869e - 006$  after 620 function evaluations.  
Simulation: #82: Best value found:  $4.364680e - 006$  after 500 function evaluations.  
Simulation: #83: Best value found:  $7.619095e - 006$  after 460 function evaluations.  
Simulation: #84: Best value found:  $3.563511e - 006$  after 560 function evaluations.  
Simulation: #85: Best value found:  $8.657929e - 006$  after 460 function evaluations.  
Simulation: #86: Best value found:  $5.608431e - 006$  after 500 function evaluations.  
Simulation: #87: Best value found:  $3.442135e - 006$  after 560 function evaluations.  
Simulation: #88: Best value found:  $2.993896e - 006$  after 380 function evaluations.  
Simulation: #89: Best value found:  $1.499479e - 006$  after 480 function evaluations.  
Simulation: #90: Best value found:  $5.543397e - 006$  after 420 function evaluations.  
Simulation: #91: Best value found:  $8.934256e - 006$  after 520 function evaluations.  
Simulation: #92: Best value found:  $3.517597e - 006$  after 480 function evaluations.  
Simulation: #93: Best value found:  $4.156746e - 006$  after 580 function evaluations.  
Simulation: #94: Best value found:  $4.454925e - 006$  after 540 function evaluations.  
Simulation: #95: Best value found:  $5.200081e - 006$  after 480 function evaluations.  
Simulation: #96: Best value found:  $6.602585e - 006$  after 440 function evaluations.  
Simulation: #97: Best value found:  $3.213922e - 006$  after 580 function evaluations.  
Simulation: #98: Best value found:  $4.154378e - 006$  after 380 function evaluations.  
Simulation: #99: Best value found:  $3.161680e - 006$  after 600 function evaluations.  
Simulation: #100: Best value found:  $3.489854e - 006$  after 740 function evaluations.



=====FINAL RESULTS=====

Average best value found:  $4.950422e - 006$ .

Average function evaluations needed: 500.

=====

Για να χαρακτηρίσουμε ένα σημείο ως ‘καλό’ θα πρέπει να έχει τιμή κοντά στο μηδέν και όσο το δυνατό λιγότερους συναρτησιακούς υπολογισμούς. Για να βρούμε τον συναρτησιακό υπολογισμό κάθε καλύτερου σημείου πολλαπλασιάζουμε τις επαναλήψεις που γίνονται μέχρι να συγκλίνει η μέθοδος επί τον πληθυσμό που είχαμε για εκείνο το σημείο. Τα αποτελέσματα που πήραμε παρουσιάζονται στους παρακάτω πίνακες:

Συνάρτηση	Μέσος Όρος	Συναρτησιακοί Υπολογισμοί
Συνάρτηση DeJong		
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 1	$2.394454e - 006$	567
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 1	$2.999260e - 006$	226
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 1	$3.129899e - 004$	85
Μέγεθος πληθυσμού 200 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 600 Στρατηγική 1	$1.493068e - 004$	212
Μέγεθος πληθυσμού 200 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 3000 Στρατηγική 1	$1.666405e - 004$	212
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 2	$2.383316e - 006$	861
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 2	$3.014600e - 006$	315
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 2	$2.729563e - 004$	85
Μέγεθος πληθυσμού 200 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 600 Στρατηγική 2	$1.395781e - 004$	206
Μέγεθος πληθυσμού 200 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 3000 Στρατηγική 2	$1.099195e - 004$	204

Πίνακας 6.1: Πίνακας συνάρτησης DeJong στην κατηγορία DE

Συνάρτηση	Μέσος Όρος	Συναρτησιακοί Υπολογισμοί
Συνάρτηση Rosenbrock		
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 600 Στρατηγική 1	$5.091328e - 004$	694
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 4000 Στρατηγική 1	$4.930214e - 004$	1219
Μέγεθος πληθυσμού 150 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 500 Στρατηγική 1	$5.078323e - 006$	6159
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 1	$4.175005e - 006$	5502
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 1	$4.594791e - 006$	1118
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 600 Στρατηγική 2	$4.086808e - 004$	1377
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000 Στρατηγική 2	$4.494383e - 004$	2619
Μέγεθος πληθυσμού 150 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 500 Στρατηγική 2	$4.358751e - 006$	6006
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 2	$4.389651e - 006$	11448
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 2	$4.628171e - 006$	2230

**Πίνακας 6.2:** Πίνακας συνάρτησης Rosenbrock στην κατηγορία DE

Συνάρτηση	Μέσος Όρος	Συναρτησιακοί Υπολογισμοί
Συνάρτηση Griewank		
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 3000 Στρατηγική 1	$1.472158e - 004$	115
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 700 Στρατηγική 1	$1.615814e - 004$	109
Μέγεθος πληθυσμού 80 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 1	$1.962018e - 004$	944
Μέγεθος πληθυσμού 150 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500 Στρατηγική 1	$9.859441e - 005$	156
Μέγεθος πληθυσμού 400 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500 Στρατηγική 1	$2.591628e - 005$	400
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 3000 Στρατηγική 2	$1.705759e - 004$	105
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 700 Στρατηγική 2	$1.797543e - 004$	115
Μέγεθος πληθυσμού 80 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 2000 Στρατηγική 2	$2.188412e - 004$	100
Μέγεθος πληθυσμού 150 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500 Στρατηγική 2	$1.458265e - 004$	156
Μέγεθος πληθυσμού 400 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 500 Στρατηγική 2	$1.832673e - 005$	400

**Πίνακας 6.3:** Πίνακας συνάρτησης Griewank στην κατηγορία DE

Συνάρτηση	Μέσος Όρος	Συναρτησιακοί Υπολογισμοί
Συνάρτηση Rastrigrin		
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500 Στρατηγική 1	$2.835317e - 004$	432
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000 Στρατηγική 1	$2.973424e - 004$	500
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000 Στρατηγική 1	$2.971368e - 004$	762
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500 Στρατηγική 1	$2.869727e - 004$	801
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 5000 Στρατηγική 1	$2.848023e - 004$	294
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500 Στρατηγική 2	$3.368246e - 004$	819
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000 Στρατηγική 2	$3.157905e - 004$	750
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000 Στρατηγική 2	$3.453582e - 004$	1173
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500 Στρατηγική 2	$3.200093e - 004$	1023
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 5000 Στρατηγική 2	$3.236548e - 004$	522

**Πίνακας 6.4:** Πίνακας συνάρτησης Rastrigrin στην κατηγορία DE



## Βελτιστοποίηση με σμήνος σωματιδίων (Particle Swarm Optimization)

### 7.1 Εισαγωγικά

**Η** Βελτιστοποίηση με σμήνος σωματιδίων (Particle Swarm Optimization) (ΒΣΣ) είναι μία μέθοδος για την βελτιστοποίηση χωρίς τη σαφή γνώση της κλίσης του προβλήματος. Η ΒΣΣ αποδίδεται στον Kennedy, Eberhart και Shi και αρχικά ο αλγόριθμος προοριζόταν για την προσομοίωση της *κοινωνικής συμπεριφοράς*. Το βιβλίο από τους Kennedy και Eberhart περιγράφει πολλές φιλοσοφικές πτυχές της ΒΣΣ και πλήθος πληροφοριών. Η ΒΣΣ βελτιστοποιεί μία συνάρτηση διατηρώντας έναν πληθυσμό υποψήφιων λύσεων που ονομάζονται σωματίδια και κινούνται αυτά τα ελάχιστα γύρω από το χώρο αναζήτησης σύμφωνα με τον απλό τύπο. Οι κινήσεις των σωματιδίων καθοδηγούνται από τις καλύτερες θέσεις που βρέθηκαν στο χώρο αναζήτησης, ο οποίος συνεχώς ενημερώνεται καθώς οι καλύτερες θέσεις βρίσκονται από τα σωματίδια [45].

### 7.2 Ο αλγόριθμος

Η συνάρτηση καταλληλότητας ή κόστους η οποία πρέπει να ελαχιστοποιηθεί είναι:  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  [45]. Το  $S$  είναι ο αριθμός των σωματιδίων στο σμήνος, καθένα από τα οποία έχει μία θέση  $x_i \in \mathbb{R}^n$  στο χώρο αναζήτησης και ταχύτητα  $v_i \in \mathbb{R}^n$ . Ο βασικός αλγόριθμος της ΒΣΣ μπορεί να περιγραφεί ως εξής:

- Για κάθε πλήθος  $i = 1, \dots, S$  κάνε:
  - Αρχικοποίησε την θέση του σμήνους:  $x_i \sim U(b_{lo}, b_{up})$ , όπου  $b_{lo}, b_{up}$  είναι τα κατώτερα και ανώτερα όρια του χώρου αναζήτησης και το  $U(.,.)$  είναι ένα τυχαία ομοιόμορφα καταναμημένο διάνυσμα.
  - Αρχικοποίησε την καλύτερη θέση που είναι γνωστή:  $p_i \leftarrow x_i$ .
  - Αρχικοποίησε την ταχύτητα:  $v_i \sim U(-(b_{up} - b_{lo}), (b_{up} - b_{lo}))$
  - Αρχικοποίησε την καλύτερη θέση των σωματιδίων που είναι γνωστή:  $g \leftarrow \operatorname{argmin}_{p_i} f(p_i)$
  - Μέχρι να βρεθεί ένα κριτήριο τερματισμού (π.χ. τον αριθμό των επαναλήψεων που παρουσιάστηκαν ή να βρεθεί επαρκής καταλληλότητα), επανέλαβε:
    - Για κάθε σμήνος  $i = 1, \dots, S$  κάνε:
      1. Δημιουργούμε τυχαία διανύσματα  $r_p, r_g \sim U(0, 1)$
      2. Ενημέρωσε την ταχύτητα του σμήνους:  $v_i \leftarrow \omega v_i + \Phi_p r_p \circ (p_i - x_i) + \Phi_g r_g \circ (g - x_i)$ , όπου το  $\circ$  υποδεικνύει το πολλαπλασιασμό στοιχείου με στοιχείο (π.χ. τον τελεστή Hadamard για τον πολλαπλασιασμό πινάκων)
      3. Ενημέρωσε την θέση του σμήνους:  $x_i \leftarrow x_i + v_i$ , να σημειωθεί ότι αυτό είναι ανεξάρτητο από την βελτίωση της καταλληλότητας.

4. Εάν ( $f(x_i) < f(p_i)$ ) κάνε:

- Ενημέρωσε την καλύτερη θέση του σμήνους που είναι γνωστή:  $p_i \leftarrow x_i$
- Εάν ( $f(p_i) < f(g)$ ) ενημέρωσε την καλύτερη θέση των σωματιδίων:  $g \leftarrow p_i$

- Τώρα το  $g$  διατηρεί την καλύτερη θέση.

Οι παράμετροι  $\omega, \phi_p, \phi_g$  ελέγχουν την συμπεριφορά και την αποτελεσματικότητα της μεθόδου της ΒΣΣ.

### 7.3 Ρύθμιση παραμέτρων

Η επιλογή καλών μεταβλητών ΒΣΣ αποτέλεσαν αντικείμενο πολλών ερευνών, μπορούμε να δούμε π.χ. Shi και Eberhart, Carlisle και Dozier, van den Bergh Clerc και Kennedy, Trelea και Bratton και Blackwell. Ένας απλός και αποτελεσματικός τρόπος ρύθμισης των παραμέτρων ΒΣΣ παρουσιάστηκε από τον Pederson et al. ο οποίος επίσης έκανε πολυάριθμα πειράματα με διαφορετικά προβλήματα βελτιστοποίησης και ρυθμίσεις. Η τεχνική ονομάζεται *μετά-βελτιστοποίηση* διότι μία άλλη μέθοδος βελτιστοποίησης χρησιμοποιείται με έναν υπερθετικό τρόπο για να ρυθμίσει τις παραμέτρους της ΒΣΣ. Η καλύτερη παρουσίαση των παραμέτρων της ΒΣΣ βρέθηκε να είναι αντίθετη από τις κατευθυντήριες γραμμές της βιβλιοθήκης και συχνά των ικανοποιητικών αποδόσεων της παρουσίασης της βελτιστοποίησης για τις απλές μεταβλητές της ΒΣΣ. Μία υλοποίηση μπορεί να βρεθεί στη βιβλιοθήκη του κώδικα πηγής SwarmOps.

### 7.4 Μεταβλητές

Νέες μεταβλητές ΒΣΣ συνεχώς εισάγονται σε μία προσπάθεια βελτίωσής της. Υπάρχουν ορισμένες τάσεις σε αυτή την αναζήτηση. Μία είναι να κάνει την μέθοδο υβριδικής βελτιστοποίησης χρησιμοποιώντας ΒΣΣ σε συνδιασμό με έναν ή περισσότερους άλλους βελτιστοποιητές. Μία άλλη τάση είναι να δοκιμάσουμε να αυξήσουμε την εξερεύνηση της βελτιστοποίησης, π.χ. αντιστρέφοντας ή διαταράσσοντας την κίνηση του σμήνους της ΒΣΣ. Και τέλος υπάρχουν προσπάθειες για την προσαρμογή των παραμέτρων συμπεριφοράς της ΒΣΣ κατά τη διάρκεια της βελτιστοποίησης.

### 7.5 Αποτελέσματα

Και εδώ θα αναφέρουμε τα αποτελέσματα που λάβαμε από το περιβάλλον του MATLAB για την κατηγορία αυτή. Όπως και στο προηγούμενο κεφάλαιο, τον αλγόριθμο που θα χρησιμοποιήσουμε τον αντλήσαμε από το διαδίκτυο [48] και θα τον χρησιμοποιήσουμε για τις ίδιες συναρτήσεις με τους Διαφοροεξελικτικούς Αλγορίθμους, δηλαδή τις συναρτήσεις DeJong, Griewank, Rastrigrin και Rosenbrock. Οι μαθηματικοί τύποι αυτών των συναρτήσεων καθώς και οι γραφικές τους παραστάσεις παρουσιάστηκαν στο προηγούμενο κεφάλαιο. Θα χρησιμοποιήσουμε τον αλγόριθμο Βελτιστοποίησης με σμήνος σωματιδίων και θα ενσωματώσουμε και πάλι τις συναρτήσεις που αναφέραμε παραπάνω. Οι συναρτήσεις που χρησιμοποιούμε έχουν ελάχιστο στο μηδέν. Για κάθε συνάρτηση χρησιμοποιήσαμε διάφορους συνδιασμούς των παραμέτρων όπως το μέγεθος του πληθυσμού, τις επαναλήψεις που έγιναν και το σφάλμα.

Τα αποτελέσματα που παίρνουμε από το περιβάλλον του MATLAB κάθε φορά που εκτελούμε μία συνάρτηση έχουν την παρακάτω μορφή:

Simulation: #1: Best value found: 8.836851e - 004 after 94300 function evaluations.  
Simulation: #2: Best value found: 6.695024e - 004 after 91800 function evaluations.



Simulation: #3: Best value found:  $9.213190e - 004$  after 94500 function evaluations.  
Simulation: #4: Best value found:  $9.622927e - 004$  after 98100 function evaluations.  
Simulation: #5: Best value found:  $3.490885e - 004$  after 96200 function evaluations.  
Simulation: #6: Best value found:  $6.850612e - 004$  after 92900 function evaluations.  
Simulation: #7: Best value found:  $6.820975e - 004$  after 90300 function evaluations.  
Simulation: #8: Best value found:  $9.136834e - 004$  after 73500 function evaluations.  
Simulation: #9: Best value found:  $8.728707e - 004$  after 91200 function evaluations.  
Simulation: #10: Best value found:  $2.577108e - 004$  after 83800 function evaluations.  
Simulation: #11: Best value found:  $5.014565e - 004$  after 88100 function evaluations.  
Simulation: #12: Best value found:  $8.102938e - 004$  after 117500 function evaluations.  
Simulation: #13: Best value found:  $6.868900e - 004$  after 107400 function evaluations.  
Simulation: #14: Best value found:  $6.068984e - 004$  after 88300 function evaluations.  
Simulation: #15: Best value found:  $1.917026e - 004$  after 93400 function evaluations.  
Simulation: #16: Best value found:  $2.268861e - 004$  after 103200 function evaluations.  
Simulation: #17: Best value found:  $4.619743e - 004$  after 85700 function evaluations.  
Simulation: #18: Best value found:  $9.669017e - 004$  after 111400 function evaluations.  
Simulation: #19: Best value found:  $3.134341e - 004$  after 82700 function evaluations.  
Simulation: #20: Best value found:  $6.634501e - 004$  after 102600 function evaluations.  
Simulation: #21: Best value found:  $1.365196e - 004$  after 75100 function evaluations.  
Simulation: #22: Best value found:  $9.940182e - 005$  after 105800 function evaluations.  
Simulation: #23: Best value found:  $4.381107e - 004$  after 87100 function evaluations.  
Simulation: #24: Best value found:  $4.988842e - 004$  after 86600 function evaluations.  
Simulation: #25: Best value found:  $7.261465e - 004$  after 91600 function evaluations.  
Simulation: #26: Best value found:  $1.357033e - 004$  after 71700 function evaluations.  
Simulation: #27: Best value found:  $8.758865e - 004$  after 102100 function evaluations.  
Simulation: #28: Best value found:  $5.649953e - 004$  after 98400 function evaluations.  
Simulation: #29: Best value found:  $4.058401e - 004$  after 96500 function evaluations.  
Simulation: #30: Best value found:  $1.502141e - 004$  after 91300 function evaluations.  
Simulation: #31: Best value found:  $6.804634e - 004$  after 95800 function evaluations.  
Simulation: #32: Best value found:  $6.054921e - 004$  after 101600 function evaluations.  
Simulation: #33: Best value found:  $7.425857e - 004$  after 114000 function evaluations.  
Simulation: #34: Best value found:  $7.035740e - 004$  after 99800 function evaluations.  
Simulation: #35: Best value found:  $8.172011e - 004$  after 101600 function evaluations.  
Simulation: #36: Best value found:  $7.088206e - 004$  after 78100 function evaluations.  
Simulation: #37: Best value found:  $2.388020e - 004$  after 90100 function evaluations.  
Simulation: #38: Best value found:  $5.651329e - 004$  after 99800 function evaluations.  
Simulation: #39: Best value found:  $8.716457e - 004$  after 94200 function evaluations.  
Simulation: #40: Best value found:  $2.179323e - 004$  after 61400 function evaluations.  
Simulation: #41: Best value found:  $9.938872e - 004$  after 88900 function evaluations.  
Simulation: #42: Best value found:  $5.816539e - 004$  after 88800 function evaluations.  
Simulation: #43: Best value found:  $3.772506e - 004$  after 82500 function evaluations.  
Simulation: #44: Best value found:  $5.641012e - 004$  after 101500 function evaluations.  
Simulation: #45: Best value found:  $4.339085e - 004$  after 86400 function evaluations.  
Simulation: #46: Best value found:  $3.175468e - 004$  after 87800 function evaluations.  
Simulation: #47: Best value found:  $1.960578e - 004$  after 85900 function evaluations.  
Simulation: #48: Best value found:  $8.802399e - 004$  after 87400 function evaluations.  
Simulation: #49: Best value found:  $4.785155e - 004$  after 103100 function evaluations.  
Simulation: #50: Best value found:  $9.178583e - 004$  after 104100 function evaluations.  
Simulation: #51: Best value found:  $9.923583e - 004$  after 78400 function evaluations.  
Simulation: #52: Best value found:  $2.302009e - 004$  after 91200 function evaluations.  
Simulation: #53: Best value found:  $4.819023e - 004$  after 98500 function evaluations.

Simulation: #54: Best value found:  $8.616350e - 005$  after 100500 function evaluations.  
Simulation: #55: Best value found:  $5.095671e - 004$  after 104700 function evaluations.  
Simulation: #56: Best value found:  $4.836775e - 005$  after 97400 function evaluations.  
Simulation: #57: Best value found:  $4.687986e - 004$  after 99000 function evaluations.  
Simulation: #58: Best value found:  $9.478368e - 004$  after 97000 function evaluations.  
Simulation: #59: Best value found:  $7.515187e - 004$  after 101000 function evaluations.  
Simulation: #60: Best value found:  $9.955792e - 004$  after 81400 function evaluations.  
Simulation: #61: Best value found:  $9.266661e - 004$  after 97700 function evaluations.  
Simulation: #62: Best value found:  $6.446359e - 004$  after 107000 function evaluations.  
Simulation: #63: Best value found:  $8.527497e - 004$  after 94000 function evaluations.  
Simulation: #64: Best value found:  $3.946887e - 004$  after 99500 function evaluations.  
Simulation: #65: Best value found:  $9.086575e - 004$  after 99900 function evaluations.  
Simulation: #66: Best value found:  $7.183144e - 004$  after 91200 function evaluations.  
Simulation: #67: Best value found:  $2.566376e - 004$  after 86600 function evaluations.  
Simulation: #68: Best value found:  $1.441569e - 004$  after 112000 function evaluations.  
Simulation: #69: Best value found:  $8.450945e - 004$  after 99000 function evaluations.  
Simulation: #70: Best value found:  $6.701597e - 004$  after 98400 function evaluations.  
Simulation: #71: Best value found:  $7.008601e - 004$  after 91000 function evaluations.  
Simulation: #72: Best value found:  $3.831426e - 004$  after 76500 function evaluations.  
Simulation: #73: Best value found:  $6.418787e - 005$  after 99100 function evaluations.  
Simulation: #74: Best value found:  $9.729305e - 004$  after 75300 function evaluations.  
Simulation: #75: Best value found:  $9.493814e - 004$  after 103200 function evaluations.  
Simulation: #76: Best value found:  $8.703510e - 004$  after 89800 function evaluations.  
Simulation: #77: Best value found:  $8.893674e - 004$  after 100700 function evaluations.  
Simulation: #78: Best value found:  $5.650430e - 004$  after 83900 function evaluations.  
Simulation: #79: Best value found:  $8.264727e - 004$  after 98900 function evaluations.  
Simulation: #80: Best value found:  $1.002985e - 005$  after 106500 function evaluations.  
Simulation: #81: Best value found:  $2.582018e - 004$  after 63200 function evaluations.  
Simulation: #82: Best value found:  $9.007677e - 004$  after 97200 function evaluations.  
Simulation: #83: Best value found:  $6.217129e - 004$  after 98200 function evaluations.  
Simulation: #84: Best value found:  $9.380691e - 004$  after 105000 function evaluations.  
Simulation: #85: Best value found:  $8.088923e - 004$  after 99700 function evaluations.  
Simulation: #86: Best value found:  $8.242145e - 004$  after 78000 function evaluations.  
Simulation: #87: Best value found:  $6.040356e - 004$  after 110000 function evaluations.  
Simulation: #88: Best value found:  $6.238117e - 005$  after 92100 function evaluations.  
Simulation: #89: Best value found:  $4.350992e - 004$  after 85600 function evaluations.  
Simulation: #90: Best value found:  $2.819141e - 004$  after 108700 function evaluations.  
Simulation: #91: Best value found:  $6.547485e - 004$  after 106800 function evaluations.  
Simulation: #92: Best value found:  $6.804538e - 004$  after 94800 function evaluations.  
Simulation: #93: Best value found:  $9.752850e - 004$  after 107500 function evaluations.  
Simulation: #94: Best value found:  $9.869080e - 004$  after 94100 function evaluations.  
Simulation: #95: Best value found:  $9.768503e - 004$  after 105900 function evaluations.  
Simulation: #96: Best value found:  $4.167403e - 004$  after 107600 function evaluations.  
Simulation: #97: Best value found:  $5.506660e - 004$  after 90600 function evaluations.  
Simulation: #98: Best value found:  $2.492327e - 004$  after 93200 function evaluations.  
Simulation: #99: Best value found:  $8.140211e - 004$  after 92200 function evaluations.  
Simulation: #100: Best value found:  $8.685768e - 004$  after 89000 function evaluations.

=====FINAL RESULTS=====

Average best value found:  $5.956616e - 004$ .

Average function evaluations needed: 94246.

=====

Για να χαρακτηρίσουμε ένα σημείο ως ‘καλό’ θα πρέπει να έχει τιμή κοντά στο μηδέν και όσο το δυνατό λιγότερους συναρτησιακούς υπολογισμούς. Για να βρούμε τον συναρτησιακό υπολογισμό κάθε καλύτερου σημείου πολλαπλασιάζουμε τις επαναλήψεις που γίνονται μέχρι να συγκλίνει η μέθοδος επί τον πληθυσμό που είχαμε για εκείνο το σημείο. Τα αποτελέσματα παρουσιάζονται στους παρακάτω πίνακες:

Συνάρτηση	Μέσος Όρος	Συναρτησιακοί Υπολογισμοί
Συνάρτηση DeJong		
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000	$5.269961e - 006$	151827
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000	$4.942182e - 004$	28292
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 2000	$5.476773e - 004$	28575
Μέγεθος πληθυσμού 200 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 600	$4.720834e - 004$	39750
Μέγεθος πληθυσμού 200 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 3000	$5.105108e - 004$	116638
Συνάρτηση Rosenbrock		
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 600	0.01663703	15533
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000	$5.956616e - 004$	94246
Μέγεθος πληθυσμού 150 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 500	$5.422027e - 004$	36337
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000	$5.225810e - 004$	163998
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000	$5.079806e - 004$	31164

**Πίνακας 7.1:** Πίνακας συναρτήσεων DeJong και Rosenbrock στην κατηγορία PSO

Συνάρτηση	Μέσος Όρος	Συναρτησιακοί Υπολογισμοί
Συνάρτηση Griewank		
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 3000	$5.724630e - 004$	69405
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 700	$7.398747e - 004$	31288
Μέγεθος πληθυσμού 80 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 2000	$5.933147e - 004$	46620
Μέγεθος πληθυσμού 150 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500	$4.532035e - 004$	35133
Μέγεθος πληθυσμού 400 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500	$4.499530e - 004$	77700
Συνάρτηση Rastrigrin		
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500	$4.821615e - 004$	23220
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000	$5.386886e - 004$	88574
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000	$4.999626e - 004$	256881
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500	$4.328268e - 004$	67851
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 5000	$5.404701e - 004$	52799

**Πίνακας 7.2:** Πίνακας συναρτήσεων Griewnak και Rastrigrin στην κατηγορία PSO



---

## Εξελικτική στρατηγική προσαρμογής πινάκων συνδιακύμανσης (Covariance Matrix Adaptation Evolution Strategy (CMA-ES))

---

### 8.1 Εισαγωγικά

**Μ**ία Εξελικτική Στρατηγική είναι μία στοχαστική μέθοδος βελτιστοποίησης για μη γραμμικά προβλήματα. Σε μία Εξελικτική Στρατηγική οι νέες υποψήφιες λύσεις προκύπτουν από μία κανονική κατανομή. Οι εξαρτήσεις μεταξύ των μεταβλητών σε αυτή την κατανομή περιγράφονται από έναν πίνακα συνδιακύμανσης. Η προσαρμογή πινάκων συνδιακύμανσης (CMA) είναι μία μέθοδος που ενημερώνει τον πίνακα συνδιακύμανσης σε αυτή τη κατανομή. Σε αντίθεση με τις κλασικές μεθόδους, μόνο η σειρά μεταξύ των υποψήφιων λύσεων αξιοποιείται για μάθηση [45].

### 8.2 Αρχές

Ο αλγόριθμος CMA-ES εκμεταλλεύεται δύο αρχές για την προσαρμογή παραμέτρων. Πρώτον, η αρχή μέγιστης πιθανοφάνειας, βασίζεται στην ιδέα της αύξησης της πιθανότητας ενός επιτυχημένου βήματος αναζήτησης. Ο πίνακας συνδιακύμανσης ενημερώνεται έτσι ώστε η προηγούμενη πιθανότητα να πραγματοποιήσει επιτυχημένα βήματα που εμφανίζονται ξανά αυξημένα. Στην πραγματικότητα, η CMA οδήγησε σε επανειλημμένη ανάλυση των επιτυχημένων βημάτων αναζήτησης. Ενώ η επανάληψη των αλγορίθμων κατανομής βασίζεται σε πολύ όμοιες ιδέες, εδώ επαναλαμβάνεται ο πίνακας συνδιακύμανσης στη μέγιστη πιθανοφάνεια των επιτυχημένων σημείων λύσης αντί των επιτυχημένων βημάτων αναζήτησης.

Δεύτερον, υπάρχουν δύο μονοπάτια που προκύπτουν από την κατανομή και ονομάζονται εξελικτικά μονοπάτια. Αυτά τα μονοπάτια περιέχουν σημαντικές πληροφορίες για τη συσχέτιση μεταξύ των διαδοχικών βημάτων. Αν τα διαδοχικά βήματα είναι προς μία διεύθυνση το εξελικτικό μονοπάτι γίνεται μακρύ. Τα εξελικτικά μονοπάτια αξιοποιούνται με δύο τρόπους. Το ένα μονοπάτι χρησιμοποιείται για την διαδικασία της προσαρμογής των πινάκων συνδιακύμανσης και το άλλο μονοπάτι χρησιμοποιείται για να ελέγξει ένα επιπλέον βήμα μετάλλαξης.

### 8.3 Αποτελέσματα

Όπως και στα δύο προηγούμενα κεφάλαια θα αναφερθούμε στα αποτελέσματα που πήραμε από το περιβάλλον του MATLAB. Με την ίδια διαδικασία όπως προηγουμένως αντήσαμε τον αλγόριθμο από το διαδίκτυο [46] και θα τον χρησιμοποιήσουμε για τις ίδιες συναρτήσεις. Δηλαδή τις συναρτήσεις DeJong, Griewank, Rastrigrin και Rosenbrock. Οι μαθηματικοί τύποι αυτών των συναρτήσεων καθώς και οι γραφικές τους παραστάσεις παρουσιάστηκαν σε προηγούμενο κεφάλαιο. Ακολουθώντας την ίδια διαδικασία, χρησιμοποιούμε

τον αλγόριθμο της Εξελικτικής στρατηγικής προσαρμογής πινάκων συνδιακύμανσης και ενσωματώνουμε τις συναρτήσεις παραπάνω. Οι συναρτήσεις που χρησιμοποιούμε έχουν ελάχιστο στο μηδέν. Για κάθε συνάρτηση χρησιμοποιήσαμε διάφορους συνδιασμούς των παραμέτρων όπως το μέγεθος του πληθυσμού, τη διάσταση του προβλήματος, τις επαναλήψεις που έγιναν και το σφάλμα.

Τα αποτελέσματα που παίρνουμε από το περιβάλλον του MATLAB κάθε φορά που εκτελούμε μία συνάρτηση έχουν την παρακάτω μορφή:

Simulation: #1: Best value found:  $2.107295e - 003$  after  $1.373242e - 001$  function evaluations.

Simulation: #2: Best value found:  $2.040239e - 003$  after  $3.436912e - 001$  function evaluations.

Simulation: #3: Best value found:  $1.117191e - 003$  after  $3.543528e - 001$  function evaluations.

Simulation: #4: Best value found:  $4.387237e - 004$  after  $1.233036e + 000$  function evaluations.

Simulation: #5: Best value found:  $4.009499e - 004$  after  $9.425140e - 001$  function evaluations.

Simulation: #6: Best value found:  $8.294599e - 004$  after  $1.223592e + 000$  function evaluations.

Simulation: #7: Best value found:  $7.914444e - 004$  after  $5.149921e - 001$  function evaluations.

Simulation: #8: Best value found:  $9.949586e - 001$  after  $7.395001e - 002$  function evaluations.

Simulation: #9: Best value found:  $9.728429e - 004$  after  $6.794172e - 002$  function evaluations.

Simulation: #10: Best value found:  $6.580768e - 004$  after  $1.236685e + 000$  function evaluations.

Simulation: #12: Best value found:  $1.642982e - 003$  after  $1.661042e - 001$  function evaluations.

Simulation: #13: Best value found:  $7.776591e - 004$  after  $5.014386e - 001$  function evaluations.

Simulation: #14: Best value found:  $7.802355e - 005$  after  $9.210336e - 001$  function evaluations.

Simulation: #15: Best value found:  $4.304739e - 005$  after  $1.035924e + 000$  function evaluations.

Simulation: #16: Best value found:  $1.367983e - 003$  after  $1.627107e + 000$  function evaluations.

Simulation: #17: Best value found:  $1.191612e - 003$  after  $5.285021e - 001$  function evaluations.

Simulation: #18: Best value found:  $1.601626e - 003$  after  $8.565924e - 001$  function evaluations.

Simulation: #19: Best value found:  $1.137950e - 003$  after  $1.451983e + 000$  function evaluations.

Simulation: #20: Best value found:  $3.902634e - 005$  after  $1.216003e - 001$  function evaluations.

Simulation: #21: Best value found:  $2.150902e - 004$  after  $2.235190e - 001$  function evaluations.

Simulation: #22: Best value found:  $3.163550e - 005$  after  $1.454272e + 000$  function evaluations.

Simulation: #23: Best value found:  $2.125997e - 003$  after  $8.707150e - 001$  function evaluations.



Simulation: #24: Best value found:  $1.359072e - 003$  after  $9.633147e - 001$  function evaluations.

Simulation: #25: Best value found:  $9.676295e - 004$  after  $1.002454e + 000$  function evaluations.

Simulation: #26: Best value found:  $1.456032e - 003$  after  $5.748992e - 001$  function evaluations.

Simulation: #27: Best value found:  $2.253688e - 004$  after  $1.116981e + 000$  function evaluations.

Simulation: #28: Best value found:  $1.214202e - 004$  after  $7.586524e - 001$  function evaluations.

Simulation: #29: Best value found:  $7.383974e - 004$  after  $1.141569e + 000$  function evaluations.

Simulation: #30: Best value found:  $8.820755e - 005$  after  $9.766725e - 001$  function evaluations.

Simulation: #31: Best value found:  $3.186196e - 004$  after  $2.734740e - 001$  function evaluations.

Simulation: #32: Best value found:  $2.975038e - 004$  after  $3.383738e - 001$  function evaluations.

Simulation: #33: Best value found:  $1.743945e - 004$  after  $6.027368e - 001$  function evaluations.

Simulation: #34: Best value found:  $2.011496e - 003$  after  $2.249410e - 001$  function evaluations.

Simulation: #35: Best value found:  $8.252336e - 004$  after  $2.004261e + 000$  function evaluations.

Simulation: #36: Best value found:  $1.792904e - 003$  after  $1.672100e + 000$  function evaluations.

Simulation: #37: Best value found:  $5.776716e - 004$  after  $9.185718e - 001$  function evaluations.

Simulation: #38: Best value found:  $5.488066e - 004$  after  $3.182503e - 001$  function evaluations.

Simulation: #39: Best value found:  $9.841197e - 004$  after  $1.416843e - 002$  function evaluations.

Simulation: #40: Best value found:  $9.949586e - 001$  after  $1.031294e + 000$  function evaluations.

Simulation: #41: Best value found:  $1.813573e - 003$  after  $1.477393e + 000$  function evaluations.

Simulation: #42: Best value found:  $2.213686e - 003$  after  $6.464341e - 001$  function evaluations.

Simulation: #43: Best value found:  $1.259452e - 004$  after  $9.721395e - 001$  function evaluations.

Simulation: #44: Best value found:  $1.613480e - 003$  after  $9.549635e - 001$  function evaluations.

Simulation: #45: Best value found:  $1.959180e - 004$  after  $5.091322e - 001$  function evaluations.

Simulation: #46: Best value found:  $5.079521e - 004$  after  $7.433758e - 001$  function evaluations.

Simulation: #47: Best value found:  $5.640055e - 004$  after  $1.045000e + 000$  function evaluations.

Simulation: #48: Best value found:  $5.423208e - 004$  after  $1.915511e - 001$  function evaluations.

Simulation: #49: Best value found:  $2.284582e - 004$  after  $1.016964e + 000$  function evaluations.

luations.

Simulation: #50: Best value found:  $1.811392e - 003$  after  $1.431053e + 000$  function evaluations.

Simulation: #51: Best value found:  $2.925997e - 004$  after  $6.615673e - 001$  function evaluations.

Simulation: #52: Best value found:  $1.381468e - 003$  after  $1.453043e - 001$  function evaluations.

Simulation: #53: Best value found:  $1.660196e - 004$  after  $3.632391e - 001$  function evaluations.

Simulation: #54: Best value found:  $1.238378e - 003$  after  $6.111288e - 001$  function evaluations.

Simulation: #55: Best value found:  $1.060780e - 003$  after  $8.666073e - 002$  function evaluations.

Simulation: #56: Best value found:  $1.351305e - 003$  after  $9.032485e - 001$  function evaluations.

Simulation: #57: Best value found:  $9.949586e - 001$  after  $7.265394e - 002$  function evaluations.

Simulation: #58: Best value found:  $1.225783e - 003$  after  $1.297869e + 000$  function evaluations.

Simulation: #59: Best value found:  $1.167757e - 004$  after  $2.564275e + 000$  function evaluations.

Simulation: #60: Best value found:  $2.035977e - 003$  after  $1.154555e + 000$  function evaluations.

Simulation: #61: Best value found:  $9.949586e - 001$  after  $3.776439e - 001$  function evaluations.

Simulation: #62: Best value found:  $2.675303e - 004$  after  $1.461423e + 000$  function evaluations.

Simulation: #63: Best value found:  $9.949586e - 001$  after  $5.349052e - 001$  function evaluations.

Simulation: #64: Best value found:  $2.126722e - 003$  after  $8.887863e - 001$  function evaluations.

Simulation: #65: Best value found:  $1.752016e - 003$  after  $6.904400e - 001$  function evaluations.

Simulation: #66: Best value found:  $2.093755e - 004$  after  $2.056922e - 001$  function evaluations.

Simulation: #67: Best value found:  $1.166969e - 003$  after  $5.939508e - 001$  function evaluations.

Simulation: #68: Best value found:  $1.152028e - 004$  after  $3.813450e - 001$  function evaluations.

Simulation: #69: Best value found:  $1.484321e - 003$  after  $8.757246e - 001$  function evaluations.

Simulation: #70: Best value found:  $1.245542e - 003$  after  $1.380620e + 000$  function evaluations.

Simulation: #71: Best value found:  $7.654082e - 004$  after  $7.080610e - 001$  function evaluations.

Simulation: #72: Best value found:  $8.521559e - 004$  after  $6.773352e - 001$  function evaluations.

Simulation: #73: Best value found:  $6.441819e - 004$  after  $7.254900e - 001$  function evaluations.

Simulation: #74: Best value found:  $1.016032e - 003$  after  $1.728280e + 000$  function evaluations.

Simulation: #75: Best value found:  $3.634426e - 004$  after  $6.605447e - 001$  function evaluations.

Simulation: #76: Best value found:  $9.949586e - 001$  after  $7.773907e - 001$  function evaluations.

Simulation: #77: Best value found:  $6.259516e - 004$  after  $3.323412e - 001$  function evaluations.

Simulation: #78: Best value found:  $3.199713e - 004$  after  $7.203027e - 002$  function evaluations.

Simulation: #79: Best value found:  $1.895069e - 003$  after  $3.420676e - 001$  function evaluations.

Simulation: #80: Best value found:  $2.117569e - 003$  after  $1.682084e - 001$  function evaluations.

Simulation: #81: Best value found:  $2.230723e - 003$  after  $8.272008e - 001$  function evaluations.

Simulation: #82: Best value found:  $6.543784e - 004$  after  $6.704193e - 001$  function evaluations.

Simulation: #83: Best value found:  $8.213165e - 004$  after  $9.275997e - 001$  function evaluations.

Simulation: #84: Best value found:  $2.021261e - 003$  after  $8.194193e - 001$  function evaluations.

Simulation: #85: Best value found:  $9.530696e - 004$  after  $9.930798e - 001$  function evaluations.

Simulation: #86: Best value found:  $9.949586e - 001$  after  $4.645927e - 001$  function evaluations.

Simulation: #87: Best value found:  $2.129112e - 003$  after  $1.258032e + 000$  function evaluations.

Simulation: #88: Best value found:  $9.949586e - 001$  after  $2.276436e - 001$  function evaluations.

Simulation: #89: Best value found:  $1.578443e - 003$  after  $6.536402e - 001$  function evaluations.

Simulation: #90: Best value found:  $1.710334e - 003$  after  $1.572892e + 000$  function evaluations.

Simulation: #91: Best value found:  $2.033226e - 004$  after  $5.888759e - 002$  function evaluations.

Simulation: #92: Best value found:  $1.548751e - 003$  after  $4.302681e - 001$  function evaluations.

Simulation: #93: Best value found:  $9.413019e - 004$  after  $2.434338e - 001$  function evaluations.

Simulation: #94: Best value found:  $1.191561e - 003$  after  $4.900891e - 001$  function evaluations.

Simulation: #95: Best value found:  $8.201643e - 004$  after  $1.486998e - 001$  function evaluations.

Simulation: #96: Best value found:  $6.659443e - 004$  after  $8.614522e - 001$  function evaluations.

Simulation: #97: Best value found:  $6.074796e - 004$  after  $1.000786e + 000$  function evaluations.

Simulation: #98: Best value found:  $1.732551e - 003$  after  $1.315384e + 000$  function evaluations.

Simulation: #99: Best value found:  $1.870118e - 003$  after  $8.787904e - 002$  function evaluations.

Simulation: #100: Best value found:  $9.218765e - 005$  after  $3.833897e - 001$  function evaluations.

luations.

=====FINAL RESULTS=====

Average best value found:  $5.933924e - 001$ .

Average function evaluations needed: 336.

=====

Για να χαρακτηρίσουμε ένα σημείο ως ‘καλό’ θα πρέπει να έχει τιμή κοντά στο μηδέν και όσο το δυνατό λιγότερους συναρτησιακούς υπολογισμούς. Για να βρούμε τον συναρτησιακό υπολογισμό κάθε καλύτερου σημείου πολλαπλασιάζουμε τις επαναλήψεις που γίνονται μέχρι να συγκλίνει η μέθοδος επί τον πληθυσμό που είχαμε για εκείνο το σημείο. Τα αποτελέσματα παρουσιάζονται στους παρακάτω πίνακες:

Συνάρτηση	Μέσος Όρος	Συναρτησιακοί Υπολογισμοί
Συνάρτηση DeJong		
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000	$4.975323e - 001$	456
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000	$4.550071e - 001$	159
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 2000	$5.849052e - 001$	655
Μέγεθος πληθυσμού 200 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 600	$5.996983e - 001$	202
Μέγεθος πληθυσμού 200 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 3000	$4.620695e - 001$	206
Συνάρτηση Rosenbrock		
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 600	$8.854009e - 001$	549
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000	$8.310328e - 001$	499
Μέγεθος πληθυσμού 150 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 500	$7.798164e - 001$	274
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000	$8.295158e - 001$	462
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 5$ Διάσταση 2 Επαναλήψεις 2000	0.1000023	895

**Πίνακας 8.1:** Πίνακας συναρτήσεων DeJong και Rosenbrock στην κατηγορία CMA-ES

Συνάρτηση	Μέσος Όρος	Συναρτησιακοί Υπολογισμοί
Συνάρτηση Griewank		
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 3000	$4.774541e - 001$	106
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 700	$3.362377e - 001$	108
Μέγεθος πληθυσμού 80 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 2000	$5.528184e - 001$	872
Μέγεθος πληθυσμού 150 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500	$4.898526e - 001$	157
Μέγεθος πληθυσμού 400 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500	$5.332831e - 001$	400
Συνάρτηση Rastrigrin		
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500	$4.917862e - 001$	337
Μέγεθος πληθυσμού 100 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000	$5.000313e - 001$	443
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 4000	$4.778799e - 001$	600
Μέγεθος πληθυσμού 300 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 500	$5.625892e - 001$	474
Μέγεθος πληθυσμού 50 Σφάλμα $1.e - 3$ Διάσταση 2 Επαναλήψεις 5000	$5.933924e - 001$	336

**Πίνακας 8.2:** Πίνακας συναρτήσεων Griewnak και Rastrigrin στην κατηγορία CMA-ES

## **Μέρος IV**

# **Συμπεράσματα – Βιβλιογραφία**





---

## Συμπεράσματα

---

### 9.1 Εισαγωγικά

Ως εξέλιξη, χαρακτηρίζεται η διαδικασία προσαρμογής ενός συστήματος στο περιβάλλον του. Οι εξελικτικοί αλγόριθμοι είναι υπολογιστικά μοντέλα που βασίζονται στη θεωρία της εξέλιξης. Σύμφωνα με τη θεωρία αυτή τα άτομα που επιβιώνουν και έχουν τις περισσότερες πιθανότητες αναπαραγωγής είναι αυτά που προσαρμόζονται καλύτερα στο περιβάλλον τους. Στους εξελικτικούς αλγορίθμους τα άτομα (υποψήφια λύσεις) που είναι καλύτερα προσαρμοσμένα στο περιβάλλον είναι αυτά που εμφανίζουν τη μικρότερη τιμή της συνάρτησης στόχου. Η λογική αυτή είναι βασισμένη στη θεωρία του Δαρβίνου σχετικά με τις αρχές εξέλιξης των ειδών, που διατυπώθηκε περί το 1960 [49]. Οι εξελικτικοί αλγόριθμοι άρχισαν να βρίσκουν ευρεία εφαρμογή από τη δεκαετία του 1990 και μετά. Ο λόγος που οι εξελικτικοί αλγόριθμοι έχουν κερδίσει έδαφος τα τελευταία χρόνια συγκριτικά με άλλες μεθόδους βελτιστοποίησης είναι η κατακόρυφη αύξηση της υπολογιστικής ισχύος των ηλεκτρονικών υπολογιστών. Τα βασικότερα χαρακτηριστικά των εξελικτικών αλγορίθμων που έχουν επικεντρώσει το ενδιαφέρον, είναι το μη-μαθηματικό τους υπόβαθρο, η ευκολία με την οποία προσαρμόζονται σε κάθε νέο πρόβλημα (αρκεί να υπάρχει συνάρτηση αξιολόγησης κάθε υποψήφιας λύσης) και κυρίως η δυνατότητά τους να μην εγκλωβίζονται σε τοπικά ακρότατα (ως στοχαστική μέθοδος). Οι εξελικτικοί αλγόριθμοι χειρίζονται πληθυσμούς υποψήφιας λύσεων και όχι μία μόνο λύση σε κάθε γενιά (κύκλο βελτιστοποίησης). Αυτό το χαρακτηριστικό τους δίνει τη δυνατότητα να επιλύσουν προβλήματα βελτιστοποίησης με πολλά ακρότατα. Με τον τρόπο αυτό, ως αποτέλεσμα της βελτιστοποίησης προκύπτει ένα σύνολο βέλτιστων λύσεων και η επιλογή της λύσης που τελικά θα χρησιμοποιηθεί εξαρτάται από τον χρήστη και την εφαρμογή.

#### **Βασικά στοιχεία των Εξελικτικών Αλγορίθμων (EA)**

1. Χρησιμοποιούν πληθυσμούς (populations) σημείων αναζήτησης, οι οποίοι μεταβάλλονται ταυτόχρονα. Αυτό τους καθιστά ιδιαίτερα ελκυστικούς για εφαρμογές σε παράλληλα και κατανεμημένα συστήματα.
2. Οι πληθυσμοί εξελίσσονται με χρήση τελεστών που προσομοιώνουν τις βασικές διαδικασίες της Δαρβινικής Θεωρίας Εξέλιξης των Ειδών. Τέτοιοι τελεστές είναι οι ακόλουθοι:
  - Επιλογή (Selection)
  - Ανασυνδιασμός (Recombination) ή Διασταύρωση (Crossover)
  - Μετάλλαξη (Mutation)

#### **Πλεονεκτήματα Εξελικτικών Αλγορίθμων (EA)**

1. Για την λειτουργία τους απαιτούν μόνο την συναρτησιακή τιμή. Στην πραγματικότητα, πολλοί EA μπορούν να λειτουργήσουν ακόμα και αν δεν είναι διαθέσιμη η συναρτησιακή τιμή, αλλά απλά μια σχέση ολικής διάταξης στα σημεία του χώρου αναζήτησης.

2. Γίνεται αναζήτηση από διαφορετικά σημεία σε πολλές περιοχές του χώρου αναζήτησης ταυτόχρονα και η πληροφορία διαχέεται στον πληθυσμό.
3. Υλοποιούνται εύκολα και είναι ιδανικοί για παράλληλα συστήματα. [50]

## 9.2 Σύγκριση αποτελεσμάτων

Τώρα θα αναφερθούμε στο πειραματικό μέρος που έγινε σε αυτή την πτυχιακή εργασία. Όπως είδαμε και σε προηγούμενα κεφάλαια χρησιμοποιήσαμε το περιβάλλον του Matlab για να πάρουμε τα αποτελέσματα για τους αλγόριθμους των Διαφοροεξελικτικών Αλγορίθμων (DE), της Βελτιστοποίησης με σμήνος σωματιδίων (PSO) και της Εξελικτικής Στρατηγικής Προσαρμογής Πινάκων Συνδιακύμανσης (CMA-ES). Στους Διαφοροεξελικτικούς Αλγορίθμους (DE) έχουν λάβει μέρος δύο στρατηγικές. Η πρώτη στρατηγική χρησιμοποιεί το καλύτερο άτομο του πληθυσμού ενώ η δεύτερη χρησιμοποιεί μόνο τυχαία επιλεγμένα άτομα του πληθυσμού. Σε κάθε αλγόριθμο ασχοληθήκαμε με τέσσερις συνάρτησεις, την συνάρτηση DeJong, την συνάρτηση Griewank, τη συνάρτηση Rastrigrin και τη συνάρτηση Rosenbrock. Σε κάθε συνάρτηση λαμβάνουν μέρος πέντε διαφορετικοί συνδιασμοί παραμέτρων όπως ο πληθυσμός, οι επαναλήψεις και το σφάλμα. Και στις τρεις μεθόδους χρησιμοποιήσαμε τους ίδιους πιθανούς συνδιασμούς για κάθε συνάρτηση έτσι ώστε στη συνέχεια να μπορούμε να τις συγκρίνουμε μεταξύ τους και να δούμε ποια μέθοδος είναι καλύτερη. Από αυτές τις συναρτήσεις παίρνουμε το καλύτερο σημείο του πληθυσμού και τον συναρτησιακό υπολογισμό (για να βρούμε τον συναρτησιακό υπολογισμό κάθε καλύτερου σημείου πολλαπλασιάζουμε τις επαναλήψεις που γίνονται μέχρι να συγκλίνει η μέθοδος ή να τερματίσει επί τον πληθυσμό που είχαμε για εκείνο το σημείο). Για να πάρουμε μία πιο σφαιρική άποψη για τα αποτελέσματα 'τρέξαμε' την κάθε συνάρτηση 100 φορές και βρήκαμε τον μέσο όρο των καλύτερων σημείων και των συναρτησιακών υπολογισμών. Για να είναι ένα σημείο 'καλό' θα πρέπει να έχει τιμή κοντά στο μηδέν και όσο το δυνατό λιγότερους συναρτησιακούς υπολογισμούς.

Σε αυτό το σημείο θα συγκρίνουμε κάθε συνάρτηση από αυτές που αναφέραμε παραπάνω (με τις ίδιες παραμέτρους) για τις τρεις αλγορίθμους που ασχοληθήκαμε. Ας ξεκινήσουμε να συζητάμε την συνάρτηση DeJong με σφάλμα=  $1e - 5$ , διάσταση= 2, πληθυσμό= 300 και επαναλήψεις= 2000. Τα αποτελέσματα που πήραμε από την μέθοδο PSO ήταν  $5.269961e - 006$  ο μέσος όρος για το καλύτερο σημείο και 151827 ο μέσος όρος των συναρτησιακών υπολογισμών που έγιναν. Από την μέθοδο DE για τη πρώτη στρατηγική πήραμε μέσο όρο για το καλύτερο σημείο  $2.394454e - 006$  και μέσο όρο των συναρτησιακών υπολογισμών που έγιναν 567. Για τη δεύτερη στρατηγική ο μέσος όρος των καλύτερων σημείων είναι  $2.383316e - 006$  και 861 ο μέσος όρος των συναρτησιακών υπολογισμών. Τέλος για τον αλγόριθμο CMA-ES ο μέσος όρος για το καλύτερο σημείο είναι  $4.975323e - 001$  και ο μέσος όρος για τους συναρτησιακούς υπολογισμούς είναι 456. Παρατηρούμε πως και στις δύο στρατηγικές της μεθόδου DE το καλύτερο σημείο είναι περίπου το ίδιο. Οι μέθοδοι DE και PSO έχουν μέσο όρο για καλύτερο σημείο πολύ κοντά στο μηδέν (όπως θέλουμε), αλλά στη μέθοδο PSO χρειάζεται να γίνουν πολλοί συναρτησιακοί υπολογισμοί για να βρεθεί. Αντιθέτως στον αλγόριθμο CMA-ES έχουμε λίγους συναρτησιακούς υπολογισμούς αλλά το καλύτερο σημείο δεν είναι πολύ κοντά στο μηδέν. Επομένως το καλύτερο σημείο το έχουμε στην μέθοδο DE με την στρατηγική που ο αλγόριθμος χρησιμοποιεί το καλύτερο σημείο του πληθυσμού.

Η συνάρτηση με σφάλμα=  $1e - 5$ , διάσταση= 2, πληθυσμό= 50 και επαναλήψεις= 2000 μας δίνει για τον αλγόριθμο PSO μέσο όρο καλύτερου σημείου  $4.942182e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 28292. Από τον αλγόριθμο DE για τη πρώτη στρατηγική πήραμε μέσο όρο για το καλύτερο σημείο  $2.999260e - 006$  και μέσο όρο των συναρτησιακών υπολογισμών που έγιναν 226. Για τη δεύτερη στρατηγική ο μέσος όρος των καλύτερων σημείων είναι  $3.014600e - 006$  και 315 ο μέσος όρος των συναρτησιακών υπολογισμών. Τέλος

για τη μέθοδο CMA-ES ο μέσος όρος για το καλύτερο σημείο είναι  $4.550071e-001$  και ο μέσος όρος για τους συναρτησιακούς υπολογισμούς είναι 159. Τα καλύτερα σημεία προκύπτουν από τις μεθόδους PSO και DE, αλλά στη μέθοδο PSO γίνονται περισσότεροι συναρτησιακοί υπολογισμοί. Στη μέθοδο CMA-ES μπορεί να βρισκεται το καλύτερο σημείο με λίγους συναρτησιακούς υπολογισμούς αλλά δεν βρίσκεται πολύ κοντά στο μηδέν. Επομένως το καλύτερο σημείο το βρίσκει η μέθοδος DE με την πρώτη στρατηγική.

Από τη συνάρτηση με σφάλμα=  $1e-3$ , διάσταση= 2, πληθυσμό= 50 και επαναλήψεις= 2000 πήραμε από τον αλγόριθμο PSO μέσο όρο καλύτερου σημείου  $5.476773e-004$  και μέσο όρο συναρτησιακών υπολογισμών 28575. Από τον αλγόριθμο DE με την στρατηγική που χρησιμοποιεί το καλύτερο σημείο του πληθυσμού πήραμε μέσο όρο για το καλύτερο σημείο  $3.129899e-004$  και μέσο όρο των συναρτησιακών υπολογισμών 85. Για τη δεύτερη στρατηγική ο μέσος όρος των καλύτερων σημείων είναι  $2.729563e-004$  και 85 ο μέσος όρος των συναρτησιακών υπολογισμών. Τέλος για τον αλγόριθμο CMA-ES ο μέσος όρος για το καλύτερο σημείο είναι  $5.849052e-001$  και ο μέσος όρος για τους συναρτησιακούς υπολογισμούς είναι 655. Τα καλύτερα σημεία προκύπτουν από τους αλγόριθμους PSO και DE, αλλά και πάλι στον αλγόριθμο PSO γίνονται περισσότεροι συναρτησιακοί υπολογισμοί. Στον αλγόριθμο CMA-ES μπορεί να βρισκεται το καλύτερο σημείο με λίγους συναρτησιακούς υπολογισμούς σε σχέση με τον αλγόριθμο PSO αλλά δεν βρίσκεται πολύ κοντά στο μηδέν. Επομένως το καλύτερο σημείο το βρίσκει ο αλγόριθμος DE με την στρατηγική όπου ο αλγόριθμος χρησιμοποιεί μόνο τυχαία άτομα του πληθυσμού.

Για τη συνάρτηση DeJong με σφάλμα=  $1e-3$ , διάσταση= 2, πληθυσμό= 200 και επαναλήψεις= 600 πήραμε από την μέθοδο PSO μέσο όρο καλύτερου σημείου  $4.720834e-004$  και μέσο όρο συναρτησιακών υπολογισμών 39750. Από την μέθοδο DE με την πρώτη στρατηγική πήραμε μέσο όρο για το καλύτερο σημείο  $1.493068e-004$  και μέσο όρο των συναρτησιακών υπολογισμών 212. Για τη στρατηγική όπου ο αλγόριθμος χρησιμοποιεί μόνο τυχαία άτομα του πληθυσμού ο μέσος όρος των καλύτερων σημείων είναι  $1.395781e-004$  και 206 ο μέσος όρος των συναρτησιακών υπολογισμών. Για την μέθοδο CMA-ES ο μέσος όρος για το καλύτερο σημείο είναι  $5.996983e-001$  και ο μέσος όρος για τους συναρτησιακούς υπολογισμούς είναι 202. Και με αυτές τις παραμέτρους τα καλύτερα σημεία προκύπτουν από τις μεθόδους PSO και DE, αλλά και πάλι στη μέθοδο PSO γίνονται περισσότεροι συναρτησιακοί υπολογισμοί. Στη μέθοδο CMA-ES μπορεί να βρισκεται το καλύτερο σημείο με λίγους συναρτησιακούς υπολογισμούς σε σχέση με την μέθοδο PSO αλλά δεν βρίσκεται πολύ κοντά στο μηδέν. Επομένως το καλύτερο σημείο το βρίσκει η μέθοδος DE με την πρώτη στρατηγική.

Για τη συνάρτηση DeJong με σφάλμα=  $1e-3$ , διάσταση= 2, πληθυσμό= 200 και επαναλήψεις= 3000 πήραμε από την μέθοδο PSO μέσο όρο καλύτερου σημείου  $5.105108e-004$  και μέσο όρο συναρτησιακών υπολογισμών 116638. Από την μέθοδο DE για τη στρατηγική με το καλύτερο εκθετικό σημείο πήραμε μέσο όρο για το καλύτερο σημείο  $1.666405e-004$  και μέσο όρο των συναρτησιακών υπολογισμών 212. Για τη στρατηγική με τυχαίο εκθετικό σημείο ο μέσος όρος των καλύτερων σημείων είναι  $1.099195e-004$  και 204 ο μέσος όρος των συναρτησιακών υπολογισμών. Για την μέθοδο CMA-ES ο μέσος όρος για το καλύτερο σημείο είναι  $4.620695e-001$  και ο μέσος όρος για τους συναρτησιακούς υπολογισμούς είναι 206. Και με αυτές τις παραμέτρους τα καλύτερα σημεία προκύπτουν από τις μεθόδους PSO και DE, αλλά και πάλι στη μέθοδο PSO γίνονται περισσότεροι συναρτησιακοί υπολογισμοί. Ομοίως με προηγουμένως, στη μέθοδο CMA-ES μπορεί να βρισκεται το καλύτερο σημείο με λίγους συναρτησιακούς υπολογισμούς σε σχέση με την μέθοδο PSO αλλά δεν βρίσκεται πολύ κοντά στο μηδέν. Επομένως το καλύτερο σημείο το βρίσκει η μέθοδος DE με την δεύτερη στρατηγική.

Στη συνέχεια θα ασχοληθούμε με την συνάρτηση Rosenbrock, όπου και εδώ θα αλλάζουμε τις παραμέτρους και θα συγκρίνουμε τα αποτελέσματα. Η συνάρτηση Rosenbrock με σφάλμα=  $1e-3$ , διάσταση= 2, πληθυσμό= 50 και επαναλήψεις= 600 έχει μέσο όρο

καλύτερου σημείου για την μέθοδο PSO 0.01663703 και μέσο όρο συναρτησιακών υπολογισμών 15533. Η μέθοδος DE όπου ο αλγόριθμος χρησιμοποιεί το καλύτερο άτομο του πληθυσμού έχει μέσο όρο καλύτερου σημείου  $5.091328e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 694. Η ίδια μέθοδος αλλά με τη στρατηγική όπου ο αλγόριθμος χρησιμοποιεί μόνο τυχαία άτομα του πληθυσμού δίνει μέσο όρο καλύτερου σημείου  $4.086808e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 1377. Η τρίτη μέθοδος, η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $8.854009e - 001$  και μέσο όρο συναρτησιακών υπολογισμών 549. Όπως και στην προηγούμενη συνάρτηση παρατηρούμε πως η μέθοδος CMA-ES έχει λίγους συναρτησιακούς υπολογισμούς αλλά το καλύτερο σημείο δεν βρίσκεται πολύ κοντά στο μηδέν. Με αυτές τις παραμέτρους παρατηρούμε πως και η μέθοδος PSO δεν έχει πολύ καλό σημείο. Αντιθέτως η μέθοδος DE βρίσκει το καλύτερο σημείο με λίγους συναρτησιακούς υπολογισμούς. Επομένως το καλύτερο σημείο το βρίσκει η μέθοδος DE με την πρώτη στρατηγική.

Όταν οι παράμετροι στην ίδια συνάρτηση είναι: το σφάλμα=  $1e - 3$ , η διάσταση= 2, ο πληθυσμός= 100 και οι επαναλήψεις= 4000 η μέθοδος PSO εμφανίζει μέσο όρο καλύτερου σημείου  $5.956616e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 94246. Η μέθοδος DE με την πρώτη στρατηγική έχει μέσο όρο καλύτερου σημείου  $4.930214e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 1219. Με τη δεύτερη στρατηγική έχουμε μέσο όρο καλύτερου σημείου  $4.494383e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 2619. Τέλος η μέθοδος CMA-ES εμφανίζει μέσο όρο καλύτερου σημείου  $8.310328e - 001$  και μέσο όρο συναρτησιακών υπολογισμών 499. Για άλλη μια φορά παρατηρούμε πως η μέθοδος CMA-ES δεν έχει πολύ καλό σημείο ενώ οι άλλες δύο μέθοδοι έχουν. Συνεπώς η μέθοδος με το καλύτερο σημείο είναι αυτή των Διαφοροεξελικτικών Αλγορίθμων με την πρώτη στρατηγική.

Η συνάρτηση Rosenbrock με σφάλμα=  $1e - 5$ , διάσταση= 2, πληθυσμός= 150 και επαναλήψεις= 500 έχει μέσο όρο καλύτερου σημείου για την μέθοδο PSO  $5.422027e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 36337. Η μέθοδος DE με την πρώτη στρατηγική έχει μέσο όρο καλύτερου σημείου  $5.078323e - 006$  και μέσο όρο συναρτησιακών υπολογισμών 6159. Η ίδια μέθοδος αλλά με την στρατηγική όπου ο αλγόριθμος χρησιμοποιεί μόνο τυχαία άτομα του πληθυσμού δίνει μέσο όρο καλύτερου σημείου  $4.358751e - 006$  και μέσο όρο συναρτησιακών υπολογισμών 6006. Η τελευταία μέθοδος, η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $7.798164e - 001$  και μέσο όρο συναρτησιακών υπολογισμών 274. Με τον ίδιο τρόπο παρατηρούμε πως καλύτερο σημείο βρίσκει η μέθοδος DE με τη δεύτερη στρατηγική.

Όταν στην ίδια συνάρτηση έχουμε σφάλμα=  $1e - 5$ , διάσταση= 2, πληθυσμός= 300 και επαναλήψεις= 2000, η μέθοδος PSO έχει μέσο όρο καλύτερου σημείου  $5.225810e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 163998. Η μέθοδος DE με την πρώτη στρατηγική έχει μέσο όρο καλύτερου σημείου  $4.175005e - 006$  και μέσο όρο συναρτησιακών υπολογισμών 5502. Η ίδια μέθοδος αλλά με την δεύτερη στρατηγική δίνει μέσο όρο καλύτερου σημείου  $4.389651e - 006$  και μέσο όρο συναρτησιακών υπολογισμών 11448. Η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $8.295158e - 001$  και μέσο όρο συναρτησιακών υπολογισμών 462. Παρατηρούμε πως και με αυτές τις παραμέτρους η μέθοδος DE με την στρατηγική 1 βρίσκει το καλύτερο σημείο.

Η συνάρτηση Rosenbrock με σφάλμα=  $1e - 5$ , διάσταση= 2, πληθυσμός= 50 και επαναλήψεις= 2000 έχει μέσο όρο καλύτερου σημείου για την μέθοδο PSO  $5.079806e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 31164. Η μέθοδος DE με την στρατηγική 1 έχει μέσο όρο καλύτερου σημείου  $4.594791e - 006$  και μέσο όρο συναρτησιακών υπολογισμών 1118. Η ίδια μέθοδος αλλά με την στρατηγική 2 δίνει μέσο όρο καλύτερου σημείου  $4.628171e - 006$  και μέσο όρο συναρτησιακών υπολογισμών 2230. Η τελευταία μέθοδος, η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου 0.1000023 και μέσο όρο συναρτησιακών υπολογισμών 895. Επομένως η μέθοδος με το καλύτερο σημείο είναι η μέθοδος DE με την πρώτη στρατηγική.

Έπειτα θα αναφερθούμε στη συνάρτηση Griewank και θα συγκρίνουμε τα αποτελέσματα των συνδιασμών που έγιναν. Η συνάρτηση αυτή με σφάλμα=  $1e - 3$ , διάσταση= 2, πληθυσμός= 100 και επαναλήψεις= 3000 έχει μέσο όρο καλύτερου σημείου για την μέθοδο PSO  $5.724630e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 69405. Η μέθοδος DE με τον αλγόριθμο που χρησιμοποιεί το καλύτερο σημείο του πληθυσμού έχει μέσο όρο καλύτερου σημείου  $1.472158e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 115. Η ίδια μέθοδος αλλά με την στρατηγική όπου ο αλγόριθμος χρησιμοποιεί μόνο τυχαία άτομα του πληθυσμού δίνει μέσο όρο καλύτερου σημείου  $1.705759e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 105. Η τελευταία μέθοδος, η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $4.774541e - 001$  και μέσο όρο συναρτησιακών υπολογισμών 106. Με τον ίδιο τρόπο όπως προηγουμένως παρατηρούμε πως η μέθοδος CMA-ES μπορεί να βρίσκει το καλύτερο σημείο της με λίγους συναρτησιακούς υπολογισμούς αλλά δεν είναι πολύ κοντά στο μηδέν. Από τις άλλες δύο μεθόδους η μέθοδος PSO κάνει αρκετούς συναρτησιακούς υπολογισμούς για να βρει το καλύτερο σημείο της. Επομένως η μέθοδος DE με την πρώτη στρατηγική είναι η μέθοδος με το καλύτερο σημείο.

Για τη συνάρτηση με σφάλμα=  $1e - 3$ , διάσταση= 2, πληθυσμός= 100 και επαναλήψεις= 700 η μέθοδος PSO έχει μέσο όρο καλύτερου σημείου  $7.398747e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 31288. Η μέθοδος DE με την πρώτη στρατηγική έχει μέσο όρο καλύτερου σημείου  $1.615814e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 109. Η ίδια μέθοδος αλλά με τη δεύτερη στρατηγική δίνει μέσο όρο καλύτερου σημείου  $1.797543e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 115. Η τελευταία μέθοδος, η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $3.362377e - 001$  και μέσο όρο συναρτησιακών υπολογισμών 108. Άρα για άλλη μία φορά η μέθοδος DE με την στρατηγική 1 είναι η μέθοδος με το καλύτερο σημείο καθώς έχει το καλύτερο σημείο της πιο κοντά στο μηδέν και το βρίσκει με λίγους συναρτησιακούς υπολογισμούς.

Στην ίδια συνάρτηση με σφάλμα=  $1e - 3$ , διάσταση= 2, πληθυσμός= 80 και επαναλήψεις= 2000 η μέθοδος PSO έχει μέσο όρο καλύτερου σημείου  $5.933147e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 46620. Η μέθοδος DE με την πρώτη στρατηγική έχει μέσο όρο καλύτερου σημείου  $1.962018e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 944. Η ίδια μέθοδος αλλά με τη δεύτερη στρατηγική δίνει μέσο όρο καλύτερου σημείου  $2.188412e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 100. Η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $5.528184e - 001$  και μέσο όρο συναρτησιακών υπολογισμών 872. Η μέθοδος που προκύπτει με το καλύτερο σημείο είναι η DE με την δεύτερη στρατηγική διότι έχει το καλύτερο σημείο της πιο κοντά στο μηδέν και το βρίσκει με λίγους συναρτησιακούς υπολογισμούς.

Όταν αλλάζουμε τις παραμέτρους στην συνάρτηση Griewank και έχουμε σφάλμα=  $1e - 3$ , διάσταση= 2, πληθυσμός= 150 και επαναλήψεις= 500 η μέθοδος PSO έχει μέσο όρο καλύτερου σημείου  $4.532035e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 35133. Η μέθοδος DE με τον αλγόριθμο που χρησιμοποιεί το καλύτερο σημείο του πληθυσμού έχει μέσο όρο καλύτερου σημείου  $9.859441e - 005$  και μέσο όρο συναρτησιακών υπολογισμών 156. Η ίδια μέθοδος αλλά με τον αλγόριθμο που χρησιμοποιεί μόνο τυχαία άτομα του πληθυσμού δίνει μέσο όρο καλύτερου σημείου  $1.458265e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 156. Η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $4.898526e - 001$  και μέσο όρο συναρτησιακών υπολογισμών 157. Παρατηρούμε και πάλι πως η μέθοδος DE με την πρώτη στρατηγική είναι αυτή με το καλύτερο σημείο.

Ο τελευταίος συνδιασμός παραμέτρων που πήραμε γι' αυτή τη συνάρτηση είναι σφάλμα=  $1e - 3$ , διάσταση= 2, πληθυσμός= 400 και επαναλήψεις= 500. Από την μέθοδο PSO πήραμε μέσο όρο καλύτερου σημείου  $4.499530e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 77700. Από την μέθοδο DE με στρατηγική 1 πήραμε μέσο όρο καλύτερου σημείου  $2.591628e - 005$  και μέσο όρο συναρτησιακών υπολογισμών 400. Η ίδια μέθοδος αλλά με στρατηγική 2 δίνει μέσο όρο καλύτερου σημείου  $1.832673e - 005$  και μέσο όρο συναρτησια-

κών υπολογισμών 400. Η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $5.332831e-001$  και μέσο όρο συναρτησιακών υπολογισμών 400. Βλέπουμε πως οι δύο τελευταίες μέθοδοι έχουν τους ίδιους συναρτησιακούς υπολογισμούς ενώ η πρώτη μέθοδος έχει πολύ περισσότερους παρόλο που το καλύτερο σημείο της πλησιάζει το μηδέν. Επομένως η μέθοδος DE με τη δεύτερη στρατηγική είναι αυτή με το καλύτερο σημείο.

Η τελευταία συνάρτηση με την οποία ασχοληθήκαμε είναι η Rastrigrin. Και σε αυτή τη συνάρτηση έχουμε πέντε συνδιασμούς παραμέτρων τους οποίους θα συγκρίνουμε. Πρώτα θα δούμε τη συνάρτηση αυτή με σφάλμα=  $1e-3$ , διάσταση= 2, πληθυσμό= 100 και επαναλήψεις= 500. Με αυτό τον συνδιασμό παραμέτρων η μέθοδος PSO έχει μέσο όρο καλύτερου σημείου  $4.821615e-004$  και μέσο όρο συναρτησιακών υπολογισμών 23220. Η μέθοδος DE με την πρώτη στρατηγική έχει μέσο όρο καλύτερου σημείου  $2.835317e-004$  και μέσο όρο συναρτησιακών υπολογισμών 432. Η ίδια μέθοδος αλλά με τη δεύτερη στρατηγική δίνει μέσο όρο καλύτερου σημείου  $3.368246e-004$  και μέσο όρο συναρτησιακών υπολογισμών 819. Η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $4.917862e-001$  και μέσο όρο συναρτησιακών υπολογισμών 337. Παρατηρούμε πως η πρώτη μέθοδος κάνει πολλούς συναρτησιακούς υπολογισμούς για να βρει το καλύτερο σημείο ενώ η μέθοδος CMA-ES κάνει λίγους συναρτησιακούς υπολογισμούς αλλά το καλύτερο σημείο της δεν είναι πολύ κοντά στο μηδέν. Άρα η DE με τον αλγόριθμο που χρησιμοποιεί το καλύτερο άτομο του πληθυσμού είναι αυτή που βρίσκει το καλύτερο σημείο.

Η συνάρτηση αυτή με σφάλμα=  $1e-3$ , διάσταση= 2, πληθυσμό= 100 και επαναλήψεις= 4000 μας δίνει μέσο όρο καλύτερου σημείου για την μέθοδο PSO  $5.386886e-004$  και μέσο όρο συναρτησιακών υπολογισμών 88574. Η μέθοδος DE με την πρώτη στρατηγική έχει μέσο όρο καλύτερου σημείου  $2.973424e-004$  και μέσο όρο συναρτησιακών υπολογισμών 500. Η ίδια μέθοδος αλλά με τη δεύτερη στρατηγική δίνει μέσο όρο καλύτερου σημείου  $3.157905e-004$  και μέσο όρο συναρτησιακών υπολογισμών 750. Η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $5.000313e-001$  και μέσο όρο συναρτησιακών υπολογισμών 443. Όπως και παραπάνω προκύπτει πως η μέθοδος με το καλύτερο σημείο είναι η DE με τη πρώτη στρατηγική εφόσον το καλύτερο σημείο που βρίσκει είναι κοντά στο μηδέν και με λίγους συναρτησιακούς υπολογισμούς.

Έχοντας τώρα στην ίδια συνάρτηση σφάλμα=  $1e-3$ , διάσταση= 2, πληθυσμό= 300 και επαναλήψεις= 4000 η PSO μας δίνει μέσο όρο καλύτερου σημείου  $4.999626e-004$  και μέσο όρο συναρτησιακών υπολογισμών 256881. Από την μέθοδο DE με την πρώτη στρατηγική έχουμε μέσο όρο καλύτερου σημείου  $2.971368e-004$  και μέσο όρο συναρτησιακών υπολογισμών 762. Η ίδια μέθοδος αλλά με τη δεύτερη στρατηγική δίνει μέσο όρο καλύτερου σημείου  $3.453582e-004$  και μέσο όρο συναρτησιακών υπολογισμών 1173. Η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $4.778799e-001$  και μέσο όρο συναρτησιακών υπολογισμών 600. Παρατηρούμε πως και πάλι βγάζουμε τα ίδια συμπεράσματα όπως προηγουμένως. Δηλαδή η μέθοδος DE με την πρώτη στρατηγική βρίσκει το καλύτερο σημείο.

Αλλάζοντας τις παραμέτρους και έχοντας τώρα σφάλμα=  $1e-3$ , διάσταση= 2, πληθυσμό= 300 και επαναλήψεις= 500 προκύπτουν τα παρακάτω αποτελέσματα. Για την μέθοδο PSO έχουμε μέσο όρο καλύτερου σημείου  $4.328268e-004$  και μέσο όρο συναρτησιακών υπολογισμών 67851. Για την μέθοδο DE με την πρώτη στρατηγική έχουμε μέσο όρο καλύτερου σημείου  $2.869727e-004$  και μέσο όρο συναρτησιακών υπολογισμών 801. Η ίδια μέθοδος αλλά με τη δεύτερη στρατηγική δίνει μέσο όρο καλύτερου σημείου  $3.200093e-004$  και μέσο όρο συναρτησιακών υπολογισμών 1023. Η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $5.625892e-001$  και μέσο όρο συναρτησιακών υπολογισμών 474. Επομένως και εδώ έχουμε την μέθοδο DE με τον αλγόριθμο που χρησιμοποιεί το καλύτερο σημείο του πληθυσμού να βρίσκει το καλύτερο σημείο.

Τέλος, για τον τελευταίο συνδιασμό των παραμέτρων (σφάλμα=  $1e-3$ , διάσταση= 2, πληθυσμό= 50 και επαναλήψεις= 5000) έχουμε στην PSO μέσο όρο καλύτερου σημείου  $5.404701e-004$  και μέσο όρο συναρτησιακών υπολογισμών 52799. Από την μέθοδο DE

με την πρώτη στρατηγική έχουμε μέσο όρο καλύτερου σημείου  $2.848023e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 294. Η ίδια μέθοδος αλλά με τη δεύτερη στρατηγική δίνει μέσο όρο καλύτερου σημείου  $3.236548e - 004$  και μέσο όρο συναρτησιακών υπολογισμών 522. Η μέθοδος CMA-ES έχει μέσο όρο καλύτερου σημείου  $5.933924e - 001$  και μέσο όρο συναρτησιακών υπολογισμών 336. Παρατηρούμε πως η μέθοδος CMA-ES ενώ έχει λίγους συναρτησιακούς υπολογισμούς το καλύτερο σημείο της δεν είναι κοντά στο μηδέν. Αντίθετα η μέθοδος PSO έχει το καλύτερο σημείο της να είναι κοντά στο μηδέν αλλά χρειάζεται να κάνει πολλούς συναρτησιακούς υπολογισμούς για να το βρει. Γι' αυτό προκύπτει πως η μέθοδος που βρίσκει το καλύτερο σημείο και με λίγους συναρτησιακούς υπολογισμούς είναι η DE με τον αλγόριθμο που χρησιμοποιεί το καλύτερο σημείο του πληθυσμού.

### 9.3 Συμπεράσματα

Από τις παραπάνω συγκρίσεις που κάναμε προκύπτει πως μεταξύ των τριών αλγορίθμων που ασχοληθήκαμε (PSO, CMA-ES και DE) καλύτερος αλγόριθμος είναι αυτός των Διαφοροεξελικτικών Αλγορίθμων (DE) και μάλιστα με την στρατηγική που χρησιμοποιεί το καλύτερο άτομο του πληθυσμού. Αυτή η στρατηγική είναι η καλύτερη διότι έχει το καλύτερο σημείο και το βρίσκει και με λίγους συναρτησιακούς υπολογισμούς. Σε σύγκριση με την άλλη στρατηγική των Διαφοροεξελικτικών Αλγορίθμων προκύπτουν παρόμοια αποτελέσματα απλά στη στρατηγική που χρησιμοποιεί μόνο τυχαία άτομα του πληθυσμού θα έχουμε ή περισσότερους συναρτησιακούς υπολογισμούς ή λίγο χειρότερο καλύτερο σημείο. Η μέθοδος CMA-ES από την άλλη ενώ έχει λίγους συναρτησιακούς υπολογισμούς, το καλύτερο σημείο που βρίσκει δεν είναι κοντά στο μηδέν όπως θα έπρεπε να είναι. Τέλος, η μέθοδος PSO βρίσκει το καλύτερο σημείο κοντά στο μηδέν αλλά για να το βρει χρειάζεται να κάνει πολλούς συναρτησιακούς υπολογισμούς. Επομένως μετά από όλα αυτά τα δεδομένα καταλήγουμε ότι καλύτερος αλγόριθμος είναι αυτός των Διαφοροεξελικτικών Αλγορίθμων.





---

## Βιβλιογραφία

---

- [1] A.E. Eiben-J.E. Smith, Introduction to Evolutionary Computing, Springer (2003).
- [2] T. Back, Evolutionary Algorithms in Theory and Practice, Oxford University Press, (New York, 1996)
- [3] T. Back, H.P. Schwefel, An overview of evolutionary algorithms for parameter optimization. Evolutionary Computation, pp. 1-23, (1993)
- [4] A.E. Eiben, Evolutionary computing: the most powerful problem solver in the universe?, Dutch Mathematical Archive (Nederlands Archief voor Wiskunde), pp.126-131, (2002)
- [5] D.B. Fogel, Evolution Computation, IEEE Press, (1995)
- [6] M.S. Hillier, F.S. Hillier, Conventional optimization techniques, pp.3-25 in Sarker et. al., Eds.
- [7] I. Rechenberg, Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien des Biologischen Evolution. Fromman-Hozlboog Verlag, Stuttgart, (1973).
- [8] X. Yao, Evolutionary computation: A gentle introduction, pp. 27-53 in Sarker et. al., Eds.
- [9] Kenneth De Jong, Genetic algorithms are NOT function optimizers, In Whitley, pages 5-18.
- [10] K. Chellapilla, D.B. Fogel, Evolving an expert checkers playing program without human expertise. IEEE Transactions on Evolutionary Computation, pp.422-428, (2001).
- [11] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, (1989).
- [12] D.E. Goldberg, R. Lingle, Alleles, loci and the traveling salesman problem, pp.154-159.
- [13] K.A. De Jong, An Analysis of the Behaviour of a Class of Genetic Adaptive Systems, University of Michigan, (1975).
- [14] J.H. Holland, Adaption in natural and artificial systems, MIT Press, (1992).
- [15] A. Jain, D.B. Fogel, Case studies in applying fitness distributions in evolutionary algorithms. II. comparing the improvements from crossover and gaussian mutation on simple neural networks. IEEE Press, Piscataway, NJ, pp.91-97, (2000).
- [16] Z. Michalewicz, Genetic Algorithms + Data structures = Evolution programs, (Berlin, 1996).
- [17] X. Yao, Y. Liu, K.H. Liang, G. Lin, Fast evolutionary algorithms. In: Ghosh, Tsutsui, pp. 45-94.

- [18] H.G. Beyer, *The Theory of Evolution Strategies*. Springer, Berlin, Heidelberg, New York, (2001).
- [19] F. Hoffmeister, T. Back, Genetic self-learning. In Varela, Bourgine, pp.227-235.
- [20] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, (1996).
- [21] T. Back, D.B. Fogel, and Z. Michalewicz, *Evolutionary Computation 1: Basic Algorithms and Operators*, Institute of Physics Publishing, (2000).
- [22] H.P. Schwefel, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, (Basel/Stuttgart 1977)
- [23] T. Back, D.B. Fogel, and Z. Michalewicz, *Evolutionary Computation 2: Advanced Algorithms and Operators*, Institute of Physics Publishing, (2000).
- [24] A.E. Eiben, E.H.L. Aarts, K.M. Van Hee, Global convergence of genetic algorithms: a Markov chain analysis, In: Schwefel, Manner, pp. 4-12.
- [25] Hans-Georg Beyer, *The theory of Evolution Strategies*, (Berlin, 2001).
- [26] H.G. Beyer and H.P. Schwefel, Evolution strategies: A comprehensive introduction, *Natural Computing*, pp. 3-52, (2002).
- [27] H.P. Schwefel, *Evolution and Optimum Seeking*, Wiley, (New York, 1995).
- [28] P.J. Angeline, Subtree crossover: Building block engine or macromutation? In: Koza et al., pp.9-17
- [29] S. Luke, L. Spector, A comparison of crossover and mutation in genetic programming. In: Koza et al., pp.240-248.
- [30] J.R. Koza, *Genetic Programming*, MIT Press, (1992).
- [31] T. Soule, J.A. Foster, J. Dickinson, Code growth in genetic programming. In: Koza et al., pp.215-223.
- [32] J.R. Koza, *Genetic Programming II*, MIT Press, (1994).
- [33] T. Soule, J.A. Foster, Effects of code growth and parsimony pressure on populations in genetic programming. *Evolutionary Computation*, pp.293-309, (1998).
- [34] H. Iba, H. de Garis, T. Sato, Genetic programming using a minimum description length principle. In: Kinnear, pp.265-284
- [35] W. Banzhaf, P. Norbin, R.E. Keller, and F.D. Francone, *Genetic Programming: An Introduction*, Morgan Kaufmann, (1998).
- [36] W.B. Langdon, *Genetic Programming + Data Structures = Automatic Programming!* Kluwer, (1998).
- [37] L.J. Fogel, A.J. Owens, M.J. Walsh, *Artificial Intelligence through Simulated Evolution*, John Wiley, (1966).
- [38] W.B. Langdon and R. Poli, *Foundations of Genetic Programming*, Springer-Verlag, (2001).
- [39] T. Back, G. Rudolph, H.P. Schwefel, Evolutionary programming and evolution strategies: Similarities and differences, In Fogel, Atmar Eds., pp. 11-22.

- [40] D.K. Gehlhaar, D.B. Fogel, Tuning evolutionary programming for conformationally flexible molecular docking, In Fogel et al., pp. 419-429.
- [41] D.B. Fogel, *Blondie24: Playing at the Edge of AI*, Morgan Kaufmann, (San Francisco, 2002).
- [42] David B. Fogel and J.W. Atmar, Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems, *Biological Cybernetics*, pp. 111-114, (1990).
- [43] W.M. Spears, Crossover or mutation, In Whitley, pp. 220-237.
- [44] Larry J. Eshelman and J. David Schaffer, Crossover's niche, In Forrest, pp.9-14.
- [45] [en.wikipedia.org](http://en.wikipedia.org)
- [46] [www.icsi.berkeley.edu/storn/code.html](http://www.icsi.berkeley.edu/storn/code.html)
- [47] [www.lri.fr/hansen/cmaes\\_inmatlab.html](http://www.lri.fr/hansen/cmaes_inmatlab.html)
- [48] [psotoolbox.sourceforge.net/](http://psotoolbox.sourceforge.net/)
- [49] <http://www.worldlingo.com>
- [50] <http://www.cs.uoi.gr>