



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνητή Ζωή: Θεωρία και Προσομοιώσεις

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξανδρος Χ. Παπαδόπουλος-Κορφιάτης

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής

Αθήνα, Ιούλιος 2010



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Τεχνητή Ζωή: Θεωρία και Προσομοιώσεις

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξανδρος Χ. Παπαδόπουλος-Κορφιιάτης

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 15^η Ιουλίου 2010.

.....
Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής

.....
Στέφανος Κόλλιας
Καθηγητής

.....
Γεώργιος Στάμου
Λέκτορας

Αθήνα, Ιούλιος 2010

.....
Αλέξανδρος Χ. Παπαδόπουλος-Κορφιιάτης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Αλέξανδρος Παπαδόπουλος-Κορφιιάτης, 2010

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η μελέτη της Τεχνητής Ζωής με έμφαση στην προσομοίωση της διαδικασίας της Εξέλιξης και της φυσικής επιλογής για την ανάπτυξη αναδυόμενων ευφυών συμπεριφορών. Το πρώτο μέρος της εργασίας ασχολείται με την θεωρητική μελέτη του πεδίου της Τεχνητής Ζωής. Με βάση την υπόθεση ότι η φυσική νοημοσύνη είναι αποτέλεσμα των διαδικασιών της Εξέλιξης και της φυσικής επιλογής στο οικοσύστημα του πλανήτη μας, η Τεχνητή Ζωή επιχειρεί να αναπαράγει τις διαδικασίες αυτές με υπολογιστικές προσομοιώσεις και τεχνητούς οργανισμούς με στόχο την μελέτη της Εξέλιξης αλλά και την επίτευξη τεχνητής νοημοσύνης. Η εργασία επίσης περιλαμβάνει την πρακτική μελέτη της Τεχνητής Ζωής με την σχεδίαση και υλοποίηση μιας εφαρμογής προσομοίωσης σε περιβάλλον MATLAB. Κάθε οργανισμός της προσομοίωσης χρησιμοποιεί ένα νευρωνικό δίκτυο για σύστημα αποφάσεων και έχει την δυνατότητα να εκτελέσει κάποιες βασικές ενέργειες. Σε βάθος χρόνου παρατηρούνται αναδυόμενες κινητικές συμπεριφορές. Τέλος δίνονται διάφορες προτάσεις για να ξεπεραστούν τα προβλήματα που προκύπτουν και για μελλοντική ανάπτυξη της εφαρμογής.

Λέξεις κλειδιά

Τεχνητή Ζωή, Τεχνητή Νοημοσύνη, Εξέλιξη, Φυσική επιλογή, Προσομοίωση, Νευρωνικό δίκτυο, Ανάδυση, MATLAB.

Abstract

This Diploma thesis deals with a study of Artificial Life, with a focus on modeling natural evolution and artificial selection in search of emergent intelligent behaviours. The first part of this thesis reviews the field of Artificial Life; based on the assumption that human intelligence is the product of the processes of evolution and natural selection in Earth's ecosystem, Artificial Life attempts to model evolution and natural selection in computer simulations using virtual agents in controlled environments in order to both study evolution and achieve emergent intelligent behavior. This thesis also involves a demonstration of basic intelligent behaviours emerging in a simple artificial life simulation created using MATLAB. Each agent uses a neural network-based decision system and has a basic set of hard-coded actions that allow the processes of natural selection and evolution to operate on the population. Over time, emergent (not hard-coded) behavior is observed, and further additions and changes to the simulation are proposed in order to achieve more interesting and diverse emergent intelligent behavior.

Keywords

Artificial Life, Artificial Intelligence, Evolution, Natural selection, Simulation, Neural network, Emergence, MATLAB.

Πίνακας Περιεχομένων

Πίνακας Σχεδίων.....	10
Πίνακας Πινάκων.....	10
1. Εισαγωγή – Στόχος	11
1.1. Εισαγωγή.....	11
1.2. Στόχος εργασίας.....	11
2. Τεχνητή Νοημοσύνη	13
2.1. Εισαγωγή.....	13
2.2. Ιστορία.....	13
2.3. Ανοιχτά προβλήματα	15
2.3.1. Αναζήτηση και επίλυση προβλημάτων	15
2.3.2. Αναπαράσταση γνώσης.....	16
2.3.3. Μηχανική μάθηση.....	16
2.3.4. Αντίληψη και κίνηση	17
2.3.5. Δημιουργικότητα.....	17
2.4. Γενική νοημοσύνη	18
2.5. Ερευνητικές προσεγγίσεις.....	19
2.5.1. Λογική και συμβολική Τεχνητή Νοημοσύνη	20
2.5.2. Αναπαράσταση γνώσης.....	20
2.5.3. Προσομοίωση εγκεφάλου.....	20
2.5.4. Ενσώματα Τεχνητή Νοημοσύνη.....	21
2.5.5. Υπολογιστική Νοημοσύνη	22
3. Εξέλιξη και αναδυόμενες συμπεριφορές.....	23
3.1. Η έννοια της εξέλιξης	23
3.1.1. Βασική ορολογία	24
3.1.2. Φυσική επιλογή	25
3.1.3. Γενετική διασπορά.....	26
3.2. Αναδυόμενες συμπεριφορές.....	28
3.2.1. Αναδυόμενες συμπεριφορές στην φύση.....	28
3.2.2. Αναδυόμενες συμπεριφορές στην κοινωνία και τις επιστήμες.....	30
4. Επιρροή της φύσης στην επιστήμη των Υπολογιστών.....	31
4.1. Γενετικοί αλγόριθμοι	31
4.1.1. Η έννοια των γενετικών αλγορίθμων	31
4.1.2. Περιγραφή των βημάτων ενός γενετικού αλγορίθμου	32
4.1.3. Εφαρμογές	34
4.2. Νευρωνικά δίκτυα.....	35
4.2.1. Εισαγωγή.....	35
4.2.2. Δομή νευρωνικών δικτύων.....	36

4.2.3.	Αρχιτεκτονική νευρωνικών δικτύων	37
4.2.4.	Μάθηση νευρωνικών δικτύων	38
4.2.5.	Εφαρμογές	39
4.3.	Κυτταρικά αυτόματα	40
4.4.	Συστήματα Lindenmayer	42
4.5.	Οχήματα Braitenberg.....	45
5.	Τεχνητή ζωή	49
5.1.	Εισαγωγή και ορισμοί.....	49
5.2.	Στόχοι.....	49
5.3.	Ιστορική ανασκόπηση.....	50
5.3.1.	Tierra και προγραμματιστική λογική.....	50
5.3.2.	Νευρωνικά δίκτυα & μάθηση	53
5.3.3.	Εξέλιξη και φυσική επιλογή	54
5.4.	Συμπεράσματα.....	56
6.	Σχεδιαστικές επιλογές.....	57
6.1.	Εισαγωγή.....	57
6.2.	Επιλογή προγραμματιστικού περιβάλλοντος.....	57
6.3.	Βιολογικές αντιστοιχίες	57
6.3.1.	Υλοποίηση οικοσυστήματος.....	58
6.3.2.	Υλοποίηση οργανισμών.....	58
6.3.3.	Υλοποίηση νευρωνικού δικτύου.....	59
6.4.	Επιλογές γραφικού περιβάλλοντος χρήστη	61
7.	Στοιχεία υλοποίησης	63
7.1.	Εισαγωγή και βασική δομή	63
7.2.	Ρουτίνα εκτέλεσης.....	63
7.3.	Κύρια στοιχεία	64
7.3.1.	Οργανισμοί	64
7.3.2.	Πίνακας οργανισμών	64
7.3.3.	Πίνακας οικοσυστήματος	65
7.3.4.	Πίνακας ταυτοποίησης.....	65
7.3.5.	Ουρές αναπαραγωγής και θανάτωσης	65
7.3.6.	Νευρωνικό δίκτυο.....	65
7.3.7.	Γραφικό περιβάλλον.....	65
7.4.	Συναρτήσεις	66
7.4.1.	Συνάρτηση νευρωνικού δικτύου (nnet)	66
7.4.2.	Συνάρτηση αναπαραγωγής	66
7.4.3.	Συνάρτηση θανάτου.....	66
7.4.4.	Συνάρτηση γραφικής απεικόνισης νευρωνικού δικτύου	66
7.4.5.	Βοηθητικές συναρτήσεις μετατροπών	67
7.4.6.	Βοηθητικές συναρτήσεις γραφικού περιβάλλοντος (Callback).....	68
7.5.	Διάγραμμα ροής	69

8. Λειτουργία εφαρμογής.....	71
8.1. Γραφικό περιβάλλον	71
8.1.1. Παράθυρο προσομοίωσης	71
8.1.2. Παράθυρο ελέγχου	73
8.1.3. Παράθυρο εμφάνισης νευρωνικού δικτύου	74
8.1.4. Παράθυρο ρυθμίσεων.....	76
8.2. Πειραματική διαδικασία.....	76
9. Παρατηρήσεις και συμπεράσματα.....	77
9.1. Παρατηρήσεις.....	77
9.1.1. Αλληλεπίδραση οργανισμών	77
9.1.2. Αναδύομενες συμπεριφορές.....	78
9.1.3. Πολυκυτταρικοί οργανισμοί.....	78
9.2. Συμπεράσματα και προβλήματα.....	79
9.2.1. Αργός κώδικας και ανεξέλεγκτος πολλαπλασιασμός.....	79
9.2.2. Πολύπλοκα νευρωνικά δίκτυα.....	79
9.2.3. Εύκολη διαταραχή φαινότυπου	80
10. Δυνατότητες επέκτασης.....	81
10.1. Βελτιώσεις στην ταχύτητα εκτέλεσης.....	81
10.2. Αλλαγές στο νευρωνικό δίκτυο	81
10.2.1. Μη στατικό νευρωνικό δίκτυο	81
10.2.2. Μάθηση νευρωνικού δικτύου	82
10.2.3. Περισσότερες επιλογές απεικόνισης	82
10.3. Περισσότερα στατιστικά στοιχεία	83
Παράρτημα Α – Βιβλιογραφία	85
Παράρτημα Β - Κώδικας.....	87

Πίνακας Σχεδίων

Σχέδιο 2.1, Προσομοίωση ενός νευρώνα του Blue Brain Project.....	21
Σχέδιο 3.1, Μια ιδιαίτερα απλοποιημένη μορφή του «δένδρου της ζωής»	24
Σχέδιο 3.2, Χρωμόσωμα και DNA	25
Σχέδιο 3.3, Απλή περίπτωση διασταύρωσης γενετικού υλικού	26
Σχέδιο 3.4, Μετάλλαξη σε ένα μέρος της αλυσίδας του DNA	26
Σχέδιο 3.5, Προσομοίωση της γενετικής διασποράς σε δυο πληθυσμούς.....	27
Σχέδιο 3.6, Σμήνος πτηνών	30
Σχέδιο 4.1, Συνδυασμός και μετάλλαξη σε ακολουθίες bit	32
Σχέδιο 4.2, Σχεδιάγραμμα ενός απλού τεχνητού νευρονίου	36
Σχέδιο 4.3, Συναρτήσεις ενεργοποίησης νευρονίων.....	37
Σχέδιο 4.4, Αρχιτεκτονικές νευρωνικών δικτύων	38
Σχέδιο 4.5, Στοιχειώδες κυτταρικό αυτόματο κανόνα 30.....	41
Σχέδιο 4.6, 3 πρώτες επαναλήψεις ενός L-system (σε κλίμακα).....	44
Σχέδιο 4.7, 5 πρώτες επαναλήψεις ενός L-system με χρήση στοίβας (σε κλίμακα)	45
Σχέδιο 4.8, Περίπλοκες κινήσεις οχημάτων Braitenberg.....	47
Σχέδιο 5.1, Απεικόνιση χώρου μνήμης Tierra.....	52
Σχέδιο 5.2, Γραφικό περιβάλλον της εφαρμογής Noble Ape	54
Σχέδιο 5.3, Geb & Polyworld.....	55
Σχέδιο 6.1, Πρότυπο νευρωνικό δίκτυο προσομοίωσης	60
Σχέδιο 7.1, Διάγραμμα ροής κυρίως προγράμματος	70
Σχέδιο 8.1, Παράθυρο προσομοίωσης	71
Σχέδιο 8.2, Δυο απεικονίσεις της ίδιας κατάστασης	72
Σχέδιο 8.3, Παράθυρο ελέγχου.....	72
Σχέδιο 8.4, Εμφάνιση στατιστικών στοιχείων.....	74
Σχέδιο 8.5, Γραμμή εργαλείων του Matlab.....	74
Σχέδιο 8.6, Παράθυρο εμφάνισης νευρωνικού δικτύου.....	75
Σχέδιο 8.7, Παράθυρο ρυθμίσεων	75
Σχέδιο 9.1, Μια προσομοίωση που δεν εξελίσσεται	77
Σχέδιο 9.2, Ανεξέλεγκτος πολλαπλασιασμός οργανισμών	78
Σχέδιο 9.3, Ένα ιδιαίτερα μπερδεμένο νευρωνικό δίκτυο.....	80

Πίνακας Πινάκων

Πίνακας 4.1, Ανάπτυξη βασικού συστήματος Lindenmayer	43
Πίνακας 5.1, Αντιστοιχίες προσομοίωσης Tierra.....	51
Πίνακας 6.1, Σειριακή αντιστοίχιση	60
Πίνακας 6.2, Παράλληλη αντιστοίχιση.....	61
Πίνακας 8.1, Πίνακας πλήκτρων ελέγχου.....	73
Πίνακας 8.2, Παράμετροι πειράματος	76

1. Εισαγωγή – Στόχος

1.1. Εισαγωγή

Η Τεχνητή Ζωή είναι ένα ιδιαίτερα ενδιαφέρον επιστημονικό πεδίο επηρεασμένο από την θεωρία της Εξέλιξης του Δαρβίνου και τους στόχους για ευφυία μηχανών της Τεχνητής Νοημοσύνης. Η μελέτη του πεδίου έχει λοιπόν νόημα τόσο για τους ερευνητές της Τεχνητής Νοημοσύνης όσο και για τους εξελικτικούς βιολόγους. Παρ' όλα αυτά, η Τεχνητή Ζωή είναι ένα σχετικά άγνωστο πεδίο στην επιστημονική κοινότητα καθώς η ουσιαστική έρευνα άρχισε μετά την δεκαετία του 1990 και δεν έχει επιδείξει ακόμα κάποιο σημαντικό αποτέλεσμα ή κάποια επιτυχημένη εμπορική εφαρμογή. Ιδιαίτερα στην ελληνική βιβλιογραφία δεν βρήκαμε καμία αναφορά στο αντικείμενο.

Στα πλαίσια της εργασίας αυτής λοιπόν θα προσπαθήσουμε να δώσουμε μια επισκόπηση του πεδίου της Τεχνητής Ζωής όπως έχει αναπτυχθεί έως σήμερα, καθώς και να το μελετήσουμε πρακτικά υλοποιώντας μια εφαρμογή - προσομοίωση.

1.2. Στόχος εργασίας

Ο στόχος της εργασίας αυτής λοιπόν είναι διπλός.

Στο πρώτο μέρος (Κεφάλαια 1 έως και 5) θα ασχοληθούμε με την θεωρητική μελέτη και την διερεύνηση του πεδίου της Τεχνητής Ζωής. Αφού αναλύσουμε το κοντινό επιστημονικά πεδίο της Τεχνητής Νοημοσύνης στο Κεφάλαιο 2, θα αναφερθούμε στη βασική για την Τεχνητή Ζωή έννοια της Εξέλιξης, τα χαρακτηριστικά της και τους τομείς της επιστήμης των Υπολογιστών τους οποίους έχει επηρεάσει στα Κεφάλαια 3 και 4. Τέλος στο Κεφάλαιο 5 θα αναφερθούμε στην ιστορία του πεδίου της Τεχνητής Ζωής και στην κατεύθυνση που έχει ακολουθήσει μέχρι σήμερα η επιστημονική μελέτη του.

Στο δεύτερο μέρος (Κεφάλαια 6 έως και 10) θα αναλύσουμε την σχεδίαση και την υλοποίηση μιας εφαρμογής Τεχνητής Ζωής. Αρχικά θα εξηγήσουμε τις σχεδιαστικές επιλογές της υλοποίησης στο Κεφάλαιο 6 και στη συνέχεια θα περιγράψουμε την προγραμματιστική υλοποίηση και την λειτουργία της εφαρμογής στα Κεφάλαια 7 και 8. Κλείνοντας θα αναφέρουμε τα αποτελέσματα, τις παρατηρήσεις και τις δυνατότητες μελλοντικής επέκτασης της εφαρμογής στα Κεφάλαια 9 και 10.

2. Τεχνητή Νοημοσύνη

2.1. Εισαγωγή

Σύμφωνα με τον John McCarthy ο οποίος εισήγαγε τον όρο σε ένα συνέδριο στο Dartmouth το 1956, η Τεχνητή Νοημοσύνη («Artificial Intelligence») ορίζεται ως η επιστήμη και η μηχανική της δημιουργίας ευφυών μηχανών και ειδικότερα ευφυών προγραμμάτων [1]. Αν και ενδιαφέρον για «μηχανές που σκέφτονται» και για την αποκωδικοποίηση των μηχανισμών της νόησης υπήρχε από την αρχαιότητα, η συστηματική έρευνα του τομέα ξεκίνησε στα μέσα του 20^{ου} αιώνα υποκινούμενη από την εφεύρεση των ηλεκτρονικών υπολογιστών.

Η Τεχνητή Νοημοσύνη, παρ' ότι είναι τυπικά κλάδος της επιστήμης των υπολογιστών, έχει απασχολήσει πολλούς ερευνητές από διάφορα πεδία, και κυρίως από τις φιλοσοφικές επιστήμες [2]. Θέτοντας ως στόχο την κατανόηση των μηχανισμών της ευφυΐας και την δημιουργία σκεπτόμενων μηχανών δημιουργεί ερωτήματα φιλοσοφικού ενδιαφέροντος: Μπορεί πράγματι μια μηχανή να σκέφτεται; Να είναι δημιουργική; Να έχει αισθήματα; Οι απόψεις των φιλοσόφων ως προς τις απαντήσεις στα παραπάνω ερωτήματα δίστανται, αλλά η παραδοχή των περισσότερων ερευνητών που ασχολούνται πρακτικά με την Τεχνητή Νοημοσύνη συνοψίζεται στην παρακάτω δήλωση, που προτάθηκε το 1956 στο συνέδριο του Dartmouth [3]:

«Κάθε πλευρά της μάθησης ή οποιαδήποτε άλλης ιδιότητας της ευφυΐας είναι δυνατόν να περιγραφεί με τόσο συγκεκριμένο τρόπο, ώστε να μπορεί να κατασκευαστεί μια μηχανή που να την εξομοιώνει».

Στο κεφάλαιο αυτό λοιπόν θα γίνει μια σύντομη αναφορά στην ιστορία της Τεχνητής Νοημοσύνης από την δεκαετία του 1950 μέχρι τις μέρες μας και θα αναπτυχθούν τα ανοιχτά προβλήματα του τομέα, οι εφαρμογές στην σύγχρονη τεχνολογία και οι σημερινές κατευθύνσεις της έρευνας.

2.2. Ιστορία

Οι απαρχές της σύγχρονης Τεχνητής Νοημοσύνης ανάγονται στην δεκαετία του 1940. Η πρόσβαση των ερευνητών στους ηλεκτρονικούς υπολογιστές και η εισαγωγή νέων ιδεών όπως η θεωρία των παιγνίων του John von Neumann το 1944 και τα τεχνητά νευρωνικά δίκτυα (τα οποία θα περιγράψουμε αναλυτικότερα στο Κεφ. 3.2) οδήγησαν ερευνητές από διάφορους τομείς στο να εξετάσουν την πιθανότητα δημιουργίας ενός τεχνητού εγκεφάλου ή μιας σκεπτόμενης μηχανής. Στις αρχές της δεκαετίας του 1950 γράφτηκαν τα πρώτα προγράμματα που έπαιζαν παιχνίδια, και ο Alan Turing δημοσίευσε ένα άρθρο-σταθμό για την μετέπειτα

εξέλιξη του κλάδου, στο οποίο διαπραγματευόταν την πιθανότητα δημιουργίας ευφυών μηχανών, και εισήγαγε μια διαδικασία δοκιμής της ευφυΐας μιας μηχανής – το περίφημο Turing test [4].

Το 1956, ο John McCarthy και ο Marvin Minsky, με δυο άλλους επιστήμονες, οργάνωσαν ένα συνέδριο στο κολέγιο του Dartmouth στις Ηνωμένες Πολιτείες της Αμερικής [3]. Ανάμεσα στις προτάσεις του συνεδρίου ήταν και αυτή που αναφέραμε παραπάνω:

«Κάθε πλευρά της μάθησης ή οποιαδήποτε άλλης ιδιότητας της ευφυΐας είναι δυνατόν να περιγραφεί με τόσο συγκεκριμένο τρόπο, ώστε να μπορεί να κατασκευαστεί μια μηχανή που να την εξομοιώνει».

Στο συνέδριο αυτό, που θεωρείται η γέννηση της Τεχνητής Νοημοσύνης, ο McCarthy έδωσε στον αναπτυσσόμενο κλάδο την ονομασία του (Artificial Intelligence) και οι συμμετέχοντες έθεσαν τις πρώτες βάσεις και τους πρώτους στόχους. Οι περισσότεροι από αυτούς που παρακολούθησαν το συνέδριο έγιναν οι πρώτοι ερευνητές της νέας επιστήμης, και ίδρυσαν εργαστήρια στα μεγαλύτερα πανεπιστήμια των Ηνωμένων Πολιτειών.

Από τη στιγμή αυτή μέχρι και τα μέσα της δεκαετίας του 1970, η έρευνα στον τομέα της Τεχνητής Νοημοσύνης άνθησε. Τα αποτελέσματα ήταν εντυπωσιακά: οι ηλεκτρονικοί υπολογιστές μάθαιναν να λύνουν προβλήματα άλγεβρας, να καταλαβαίνουν και να εκτελούν απλές εντολές σε φυσική γλώσσα. Η χρηματοδότηση ήταν γενναιόδωρη, σταθερή και χωρίς περιορισμούς και οι προσδοκίες των ερευνητών ήταν υψηλές: ο Marvin Minsky το 1967 δήλωσε ότι μέσα σε μια γενιά, το πρόβλημα της δημιουργίας τεχνητής νοημοσύνης θα είχε λυθεί [5].

Παρ' όλα αυτά, οι ελπίδες αυτές δεν επαληθεύτηκαν. Η έρευνα σκόνταψε σε κάποια προβλήματα τα οποία δεν λύνονταν εύκολα, και τα πρώτα χρόνια της δεκαετίας του 70 έφτασαν χωρίς τα αναμενόμενα αποτελέσματα. Η έλλειψη αποτελεσμάτων οδήγησε σταδιακά στην διακοπή της χρηματοδότησης για γενική, μη κατευθυνόμενη έρευνα στον τομέα και άρχισε ο πρώτος «χειμώνας» της Τεχνητής Νοημοσύνης.

Στις αρχές της δεκαετίας του 1980, το ενδιαφέρον για την Τεχνητή Νοημοσύνη άρχισε να ανακάμπτει με την δημιουργία ενός νέου είδους προγραμμάτων τεχνητής νοημοσύνης, των «έμπειρων συστημάτων» [6]. Τα έμπειρα συστήματα είναι προγράμματα που απαντάνε σε ερωτήσεις ή βγάζουν χρήσιμα συμπεράσματα για έναν πολύ συγκεκριμένο τομέα, χρησιμοποιώντας ένα σύνολο από κανόνες και μια βάση δεδομένων που περιέχει γνώσεις σχετικές με τον τομέα. Οι κανόνες και οι γνώσεις εισάγονται στο σύστημα από ειδικούς με εμπειρία στον συγκεκριμένο τομέα. Τα έμπειρα συστήματα άρχισαν να χρησιμοποιούνται με

επιτυχία από επιχειρήσεις παγκοσμίως, με αποτέλεσμα να δημιουργηθεί μια αγορά δισεκατομμυρίων από την πώληση των εξειδικευμένων μηχανημάτων που έτρεχαν τα συστήματα. Παράλληλα, τα χειροπιαστά αποτελέσματα και οι ευκαιρίες που δημιούργησε η νέα αγορά αυτή οδήγησαν σε ένα νέο κύμα κρατικών χρηματοδοτήσεων για έρευνα στην Τεχνητή Νοημοσύνη. Οι ερευνητές την δεκαετία αυτή επικεντρώθηκαν σε συστήματα βασισμένα στην γνώση και την αναπαράστασή της.

Το ενδιαφέρον αυτό όμως δεν κράτησε για πολύ: το 1987 η αγορά των εξειδικευμένων μηχανημάτων για έμπειρα συστήματα κατέρρευσε, κυρίως λόγω της αυξανόμενης ταχύτητας των ηλεκτρονικών υπολογιστών, οι οποίοι ξεπέρασαν τα μηχανήματα έμπειρων συστημάτων σε υπολογιστική ισχύ. Παράλληλα, στο τέλος της δεκαετίας του 1980 η κρατική χρηματοδότηση για έρευνα ξαφνικά κόπηκε.

Από τις αρχές της δεκαετίας του 1990 μέχρι και σήμερα, ο τομέας της Τεχνητής Νοημοσύνης έχει σταδιακά ανακάμψει και έχει πετύχει αρκετούς από τους αρχικούς στόχους που είχαν θέσει οι πρώτοι ερευνητές. Τα αποτελέσματα της έρευνας αυτών των χρόνων όλο και περισσότερο χρησιμοποιούνται σε πρακτικές εφαρμογές. Για παράδειγμα, η οπτική αναγνώριση χαρακτήρων, η αναγνώριση φωνής, η αναζήτηση σε βάσεις δεδομένων, οι μηχανές αναζήτησης στο διαδίκτυο (όπως η Google) και τα ρομποτικά συστήματα παραγωγής βασίζονται σε συστήματα τεχνητής νοημοσύνης [7, 8]. Αρκετά προβλήματα όμως που προέκυψαν στα πρώτα χρόνια της έρευνας είναι ακόμα ανοιχτά, και η στροφή των ερευνητών σε ζητήματα με πιο πρακτικά και άμεσα εφαρμόσιμα αποτελέσματα οδήγησε στο να μείνει πίσω η έρευνα για την δημιουργία νόησης.

2.3. Ανοιχτά προβλήματα

Στο κεφάλαιο αυτό θα αναφέρουμε κάποια από τα ανοιχτά προβλήματα με τα οποία ασχολείται η σύγχρονη έρευνα στην Τεχνητή Νοημοσύνη [9].

2.3.1. Αναζήτηση και επίλυση προβλημάτων

Μια από τις αρχικές προσεγγίσεις της Τεχνητής Νοημοσύνης την δεκαετία του 1960 ήταν η κατασκευή αλγορίθμων που επιχειρούσαν να επιλύσουν ένα δεδομένο πρόβλημα ακολουθώντας την λογική σκέψη και την συλλογιστική διαδικασία που θα χρησιμοποιούσε ένας άνθρωπος. Για δύσκολα προβλήματα, η πολυπλοκότητα των αλγορίθμων αυτών αυξάνει απότομα, με αποτέλεσμα να συμβαίνει μια «έκρηξη συνδυασμών» και η λύση του προβλήματος να μην είναι τεχνικά εφικτή.

Από τα πρώτα λοιπόν αποτελέσματα της έρευνας στην Τεχνητή Νοημοσύνη ήταν αποδοτικοί αλγόριθμοι αναζήτησης, οι οποίοι χρησιμοποιούνται πλέον σε πολλές τεχνολογικές εφαρμογές. Η εύρεση όλο και πιο γρήγορων αλγορίθμων αναζήτησης αποτελεί ακόμα και σήμερα σημαντικό μέρος της έρευνας στην Τεχνητή Νοημοσύνη. Παράλληλα, η ταχεία αύξηση της υπολογιστικής δύναμης των σύγχρονων μηχανημάτων δίνει συνεχώς περισσότερες δυνατότητες στους ερευνητές [10].

2.3.2. Αναπαράσταση γνώσης

Ένα ακόμα πρόβλημα με το οποίο ασχολείται η έρευνα είναι η αναπαράσταση και η κωδικοποίηση της γνώσης. Η ανθρώπινη ευφυΐα και ικανότητα επίλυσης προβλημάτων βασίζεται εν μέρει σε έναν μεγάλο αριθμό γνώσεων για τον κόσμο, τα αντικείμενα που υπάρχουν σε αυτόν και τις σχέσεις μεταξύ των αντικειμένων. Την δεκαετία του 1980 η έρευνα επικεντρώθηκε στον τομέα αυτόν. Ένα δημοφιλές σύστημα αναπαράστασης γνώσης ήταν τα έμπειρα συστήματα.

Ένας από τους στόχους σήμερα είναι η αναπαράσταση της «κοινότυπης γνώσης», δηλαδή του συνόλου των κοινών γνώσεων, δεδομένων και πληροφοριών που κατέχουν οι περισσότεροι άνθρωποι. Το εγχείρημα αυτό είναι αρκετά δύσκολο λόγω του τεράστιου αριθμού των γνώσεων αυτών. Πολλές από τις κοινότυπες γνώσεις δεν είναι συνειδητές ή δεν μπορούν να διατυπωθούν υπό την μορφή λογικών προτάσεων οπότε γίνεται προσπάθεια να αναπαρασταθούν με υπό-συμβολική μορφή, για παράδειγμα με την χρήση νευρωνικών δικτύων. Συστήματα βάσεων γνώσεων που συνεχίζουν να εμπλουτίζονται και να έχουν ερευνητική δραστηριότητα είναι για παράδειγμα το σύστημα «Open Mind Common Sense» του Media Lab του MIT [11] και το σύστημα Cyc της εταιρίας Cycorp [12].

2.3.3. Μηχανική μάθηση

Πριν ακόμα αρχίσει η έρευνα στον τομέα της Τεχνητής Νοημοσύνης, από την δεκαετία του 1940, είχε προταθεί η ιδέα να κατασκευαστεί μια «μηχανή-παιδί» η οποία θα έχει την δυνατότητα να μάθει [1]. Από τότε, στα πλαίσια της Τεχνητής Νοημοσύνης έχουν γίνει πολλές προσπάθειες να κατασκευαστούν συστήματα που να έχουν την δυνατότητα να μαθαίνουν από μόνα τους να αναγνωρίζουν πρότυπα και να παίρνουν αποφάσεις ανάλογα με τα δεδομένα εισόδου τους.

Υπάρχουν διάφοροι τύποι αλγορίθμων μηχανικής μάθησης [13]. Οι αλγόριθμοι επιβλεπόμενης μάθησης εκπαιδεύονται αρχικά με κάποια δεδομένα εκπαίδευσης ώστε να συσχετίσουν τα δεδομένα εισόδου με τις επιθυμητές εξόδους και στη συνέχεια εφαρμόζουν την συσχέτιση αυτή

σε οποιοδήποτε σύνολο εισόδων. Οι αλγόριθμοι μη επιβλεπόμενης μάθησης αναγνωρίζουν μόνοι τους πρότυπα στα δεδομένα εισόδου, τα οποία και ταξινομούν αντίστοιχα. Οι αλγόριθμοι ενισχυτικής μάθησης μαθαίνουν μέσω ανάδρασης. Κάθε απόφαση του αλγορίθμου έχει είτε θετική είτε αρνητική ανάδραση από το περιβάλλον, οπότε ο αλγόριθμος μάθησης οδηγείται στις αποφάσεις που μεγιστοποιούν την ανάδραση.

Για την κατασκευή συστημάτων μηχανικής μάθησης προσφέρονται ιδιαίτερα τα νευρωνικά δίκτυα, τα οποία θα εξετάσουμε στο Κεφ. 4.2.

2.3.4. Αντίληψη και κίνηση

Μια από τις σύγχρονες απόψεις ανάμεσα στους ερευνητές είναι αυτή της «ενσώματης νοημοσύνης», σύμφωνα με την οποία η ύπαρξη σώματος και η αλληλεπίδραση με το περιβάλλον είναι απαραίτητες προϋποθέσεις για την εκδήλωση νοημοσύνης [14]. Για να υπάρχει αλληλεπίδραση με το περιβάλλον πρέπει να υπάρχει δυνατότητα κίνησης στο χώρο και δυνατότητα αντίληψης του περιβάλλοντος. Η αντίληψη περιλαμβάνει για παράδειγμα την όραση, την ακοή και την αφή.

Με την αυτόματη κίνηση των μηχανών ασχολείται η Ρομποτική, επιστήμη κοντινή στην Τεχνητή Νοημοσύνη. Πολλά από τα αυτόνομα μηχανήματα που κατασκευάζονται χρησιμοποιούν συστήματα τεχνητής νοημοσύνης για την λήψη αποφάσεων και για την κατάστρωση ενός σχεδίου κίνησης. Η επεξεργασία φωνητικών εντολών και η μηχανική όραση, κλάδοι της επεξεργασίας σημάτων, δίνουν την δυνατότητα ακοής και όρασης αντίστοιχα.

2.3.5. Δημιουργικότητα

Ένα σημαντικό γνώρισμα της ανθρώπινης νοημοσύνης είναι η φαντασία και η δημιουργικότητα. Υπάρχει λοιπόν ερευνητικό ενδιαφέρον τόσο για την θεωρητική μελέτη της μηχανικής δημιουργικότητας, όσο και για την πρακτική μελέτη και την δημιουργία μηχανημάτων ή προγραμμάτων τα οποία εμφανίζουν δημιουργικότητα. Ένα παράδειγμα δημιουργικού προγράμματος είναι το AARON του Harold Cohen, το οποίο σχεδιάζει πίνακες ζωγραφικής [15]. Αναπτύσσεται από το 1973 μέχρι και σήμερα, και έργα που παρήγαγε έχουν προβληθεί σε εκθέσεις τέχνης παγκοσμίως.

2.4. Γενική νοημοσύνη

Ο αρχικός στόχος της Τεχνητής Νοημοσύνης ήταν η δημιουργία μιας ευφυούς μηχανής, και η προσπάθεια της επιστημονικής κοινότητας τα πρώτα χρόνια ήταν να κατανοήσει και να πετύχει τον στόχο αυτόν. Όπως όμως είδαμε στο προηγούμενο κεφάλαιο, η έρευνα σήμερα έχει επικεντρωθεί σε συγκεκριμένα και εξειδικευμένα υπό-προβλήματα, τα οποία έχουν άμεσα, ελέγξιμα αποτελέσματα και εμπορικό ενδιαφέρον, όπως τα νευρωνικά δίκτυα, οι αλγόριθμοι αναζήτησης ή η αναγνώριση προτύπων. Η διαφοροποίηση αυτή οδήγησε στον διαχωρισμό των εννοιών της «γενικής» ή «δυνατής» Τεχνητής Νοημοσύνης («general / strong AI») και της «εφαρμοσμένης» ή «αδύναμης» Τεχνητής Νοημοσύνης («applied / weak AI»). Η γενική Τεχνητή Νοημοσύνη αναφέρεται στην δημιουργία μιας μηχανής με ευφυΐα αντίστοιχη ή εκτενέστερη της ανθρώπινης, ενώ η εφαρμοσμένη Τεχνητή Νοημοσύνη αναφέρεται στην χρήση προγραμμάτων για την επίτευξη συγκεκριμένων στόχων που αποτελούν υποσύνολο της νοητικής δυνατότητας των ανθρώπων [16].

Οι περισσότεροι ερευνητές ασχολούνται με την εφαρμοσμένη Τεχνητή Νοημοσύνη για διάφορους λόγους. Λόγω έλλειψης άμεσων και εφαρμόσιμων αποτελεσμάτων η γενική Τεχνητή Νοημοσύνη δεν έχει εμπορικό ενδιαφέρον οπότε δεν υπάρχουν και πηγές χρηματοδότησης. Επίσης, οι γνώσεις μας σε ζητήματα τα οποία ενδεχομένως να είναι κρίσιμα για την σχεδίαση ενός πραγματικά ευφυούς συστήματος είναι ακόμα ελλιπείς. Για παράδειγμα η λειτουργία του ανθρώπινου εγκεφάλου και οι διαδικασίες της νόησης (και των χαρακτηριστικών της, όπως η μάθηση και η λογική σκέψη) δεν έχουν ακόμα εξηγηθεί ικανοποιητικά. Τέλος, σημαντικό ρόλο έπαιξε το γεγονός ότι τόσο η πρώτη γενιά ερευνητών όσο και οι ερευνητές την δεκαετία του 1980 είχαν υποτιμήσει την δυσκολία του εγχειρήματος σχεδίασης ενός ευφυούς συστήματος. Έθεσαν μεγαλεπήβολους στόχους και ανακοίνωσαν αποτελέσματα τα οποία δεν κατάφεραν να πετύχουν, με αποτέλεσμα οι σημερινοί ερευνητές να διστάζουν να δηλώσουν αντίστοιχους στόχους και να διατυπώσουν οποιουδήποτε είδους πρόβλεψη, φοβούμενοι τον χαρακτηρισμό του «οραματιστή».

Οι περισσότεροι ερευνητές ελπίζουν πάντως ότι μελλοντικά τα αποτελέσματα που θα έχουν προκύψει από την μελέτη των επιμέρους προβλημάτων θα συνδυαστούν σε ένα σύστημα γενικής νοημοσύνης. Παρ' όλα αυτά, κάποιες μεμονωμένες ερευνητικές ομάδες (όπως το Singularity Institute for Artificial Intelligence, [17]) ασχολούνται αποκλειστικά με την γενική Τεχνητή Νοημοσύνη, με τους τρόπους με τους οποίους μπορεί να επιτευχθεί και με τους κινδύνους ή τις ευκαιρίες που θα δημιουργήσει η ανάπτυξη μιας τεχνητής νοημοσύνης εξίσου ή περισσότερο έξυπνης από τον άνθρωπο. Ο Raymond Kurzweil, ένας γνωστός εφευρέτης και επιστήμονας

υπολογιστών, υποστηρίζει ότι η εκθετική πρόοδος της υπολογιστικής ισχύος σημαίνει ότι μέσα στα επόμενα 20 χρόνια θα έχουμε αναπτύξει μια γενική νοημοσύνη [10].

2.5. Ερευνητικές προσεγγίσεις

Κλείνοντας την επισκόπηση της Τεχνητής Νοημοσύνης, θα αναπτύξουμε σύντομα κάποιες από τις κατευθύνσεις που ακολουθεί σήμερα η έρευνα. Οι ερευνητές της Τεχνητής Νοημοσύνης δεν έχουν συμφωνήσει ως τώρα σε κάποιον συγκεκριμένο τρόπο να πετύχουν τον κοινό στόχο της δημιουργίας ενός ευφυούς συστήματος, με αποτέλεσμα να υπάρχουν διάφορες προσεγγίσεις και «σχολές» που εργάζονται παράλληλα.

Γενικά θα μπορούσαμε να χωρίσουμε τις προσεγγίσεις σε δύο κατηγορίες [18]. Η πρώτη κατηγορία προσπαθεί να δημιουργήσει μια τεχνητή ευφυΐα αναλύοντας και κωδικοποιώντας την διαδικασία ή τα χαρακτηριστικά της ανθρώπινης νόησης και μεταφέροντάς τα σε κάποια μορφή σε ένα μηχάνημα ή ένα πρόγραμμα. Ακολουθεί δηλαδή μια προσέγγιση «από πάνω προς τα κάτω» (top-down approach). Στην κατηγορία αυτή περιλαμβάνονται οι προσπάθειες που βασίζονται στην «λογική» ή «συμβολική» Τεχνητή Νοημοσύνη και οι προσπάθειες συσσώρευσης γνώσεων σε βάσεις δεδομένων.

Η δεύτερη κατηγορία ακολουθεί την προσέγγιση «από κάτω προς τα πάνω» (bottom-up approach). Αντί να προσπαθήσει να κατανοήσει την ανθρώπινη νόηση, επιχειρεί να την συνθέσει σε υπό-συμβολική μορφή και να ερευνήσει την πολυπλοκότητα από την οποία αναδύεται το φαινόμενο της νοημοσύνης. Στην κατηγορία αυτή περιλαμβάνονται οι προσπάθειες προσομοίωσης του ανθρώπινου εγκεφάλου, η έρευνα σύνθετων συστημάτων όπως τα νευρωνικά δίκτυα, ο κλάδος της «ενσώματης» Τεχνητής Νοημοσύνης και κυρίως η υπολογιστική νοημοσύνη.

Από την δεκαετία του 1950 μέχρι και τα τέλη της δεκαετίας του 1980, οι προσπάθειες των ερευνητών είχαν επικεντρωθεί αποκλειστικά στην πρώτη κατηγορία, αρχικά στην συμβολική Τεχνητή Νοημοσύνη και αργότερα στις βάσεις γνώσεων. Από την δεκαετία του 1990, μετά τις διαδοχικές αποτυχίες των αρχικών προσεγγίσεων να παράγουν χρήσιμα αποτελέσματα, η έρευνα στράφηκε στην κατηγορία της υπό-συμβολικής Τεχνητής Νοημοσύνης, μελετώντας την υπολογιστική νοημοσύνη και τα νευρωνικά δίκτυα. Σήμερα οι δύο κατηγορίες προσεγγίσεων συνυπάρχουν και ερευνώνται παράλληλα.

2.5.1. Λογική και συμβολική Τεχνητή Νοημοσύνη

Η προσέγγιση αυτή ονομάζεται και «Παλιά Καλή Τεχνητή Νοημοσύνη» («Good Old Fashioned AI»). Βασίζεται στην υπόθεση ότι η νοημοσύνη μπορεί να εξηγηθεί ως μια διαδικασία επεξεργασίας συμβόλων και ήταν δημοφιλής τις δεκαετίες του 1960 – 1970 [19]. Κάποιοι από τους επιστήμονες προσπάθησαν να κατανοήσουν την ανθρώπινη συλλογιστική πορεία επίλυσης ενός προβλήματος και να την προσομοιώσουν, μεταφέροντάς την με την χρήση αλγορίθμων στα μηχανήματα. Άλλοι υποστήριζαν ότι ένα ευφυές πρόγραμμα δεν χρειάζεται αναγκαστικά να ακολουθεί την πορεία της ανθρώπινης σκέψης και επικεντρώθηκαν στην επίλυση προβλημάτων χρησιμοποιώντας αρχικά συστήματα τυπικής λογικής για την λήψη αποφάσεων. Γρήγορα έγινε σαφές ότι η «δυαδική» λογική επαρκούσε μόνο για ορισμένες εφαρμογές, και δόθηκε έμφαση στην ανάπτυξη συστημάτων μη-μονοτονικής λογικής.

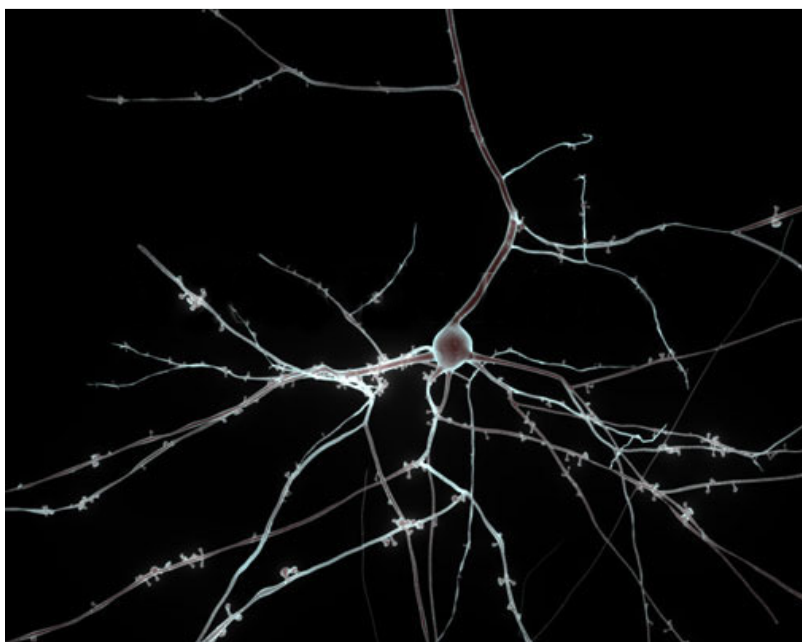
2.5.2. Αναπαράσταση γνώσης

Όπως αναφέραμε και στο Κεφ. 2.3.2, την δεκαετία του 1980 δόθηκε ιδιαίτερη σημασία στην προσέγγιση της αναπαράστασης γνώσης, αφού οι ερευνητές συνειδητοποίησαν ότι τα προγράμματα τεχνητής νοημοσύνης θα απαιτούσαν πλήθος γνώσεων για να μπορούν να λύσουν προβλήματα. Η στροφή στην αναπαράσταση γνώσης οδήγησε στα πρώτα εμπορικά επιτυχημένα συστήματα Τεχνητής Νοημοσύνης, τα έμπειρα συστήματα. Σήμερα γίνεται προσπάθεια από διάφορες ερευνητικές ομάδες να συσσωρευτούν όλο και περισσότερες κοινότητες γνώσεις σε βάσεις δεδομένων. Το έργο αυτό είναι όμως ιδιαίτερα απαιτητικό και οι αντίστοιχες προσπάθειες δέχονται πολλές κριτικές. Οι περισσότερες προσπάθειες, όπως για παράδειγμα το σύστημα Cyc, ελπίζουν η πρόοδος στην κατανόηση φυσικής γλώσσας να επιτρέψει στα συστήματά τους να αντλούν αυτόνομα πληροφορίες «διαβάζοντας» από το Internet [12].

2.5.3. Προσομοίωση εγκεφάλου

Μια κατεύθυνση προς την οποία εργάζονται οι ερευνητές είναι η ανάπτυξη υλικού και λογισμικού που να προσομοιώνουν την λειτουργία του ανθρώπινου εγκεφάλου. Η προσπάθεια αυτή ερευνάται από την δεκαετία του 1950 και είναι ιδιαίτερα δημοφιλής σήμερα ανάμεσα στους ερευνητές γενικής τεχνητής νοημοσύνης. Δεν προϋποθέτει την πλήρη κατανόηση της λειτουργίας της νόησης, οπότε σε συνδυασμό με την εκθετικά αυξανόμενη υπολογιστική δύναμη που έχουμε στην διάθεσή μας είναι ίσως ένας από τους πιο πιθανούς τρόπους να αναπτυχθεί ένα ευφυές σύστημα στο άμεσο μέλλον.

Από τις πιο φιλόδοξες σύγχρονες προσπάθειες προσομοίωσης του ανθρώπινου εγκεφάλου είναι το «Blue Brain Project» της Πολυτεχνικής Σχολής της Λωζάννης στη Σουηδία. Χρησιμοποιώντας έναν υπερυπολογιστή της IBM, τον Blue Gene, στοχεύει στην προσομοίωση σε βιολογικό επίπεδο ενός συμπλέγματος νευρώνων (Σχ. 2.1). Το 2006 προσομοιώθηκε στα πλαίσια του προγράμματος ένα μικρό μέρος του εγκεφάλου ενός ποντικιού. Στόχος των ερευνητών είναι η μελέτη της διαδικασίας της νόησης και πιο μακροπρόθεσμα η προσομοίωση του ανθρώπινου εγκεφάλου [20].



Σχέδιο 2.1, Προσομοίωση ενός νευρώνα του Blue Brain Project

2.5.4. Ενσώματη Τεχνητή Νοημοσύνη

Κάποιοι ερευνητές υποστηρίζουν ότι ένας απομονωμένος εγκέφαλος, χωρίς αντίστοιχο σώμα, δεν έχει την δυνατότητα να εμφανίσει νοημοσύνη. Σύμφωνα με την άποψη αυτή η ύπαρξη σώματος και η αλληλεπίδραση με το περιβάλλον είναι απαραίτητα χαρακτηριστικά για την ευφυΐα, οπότε παράλληλα με τον προγραμματισμό ενός συστήματος λήψης αποφάσεων, κατασκευάζεται και ένα ρομποτικό σώμα που έχει δυνατότητα αυτό-κίνησης και αίσθηση του περιβάλλοντος.

Πηγαίνοντας ένα βήμα παραπέρα, ένα μέρος των ερευνητών απορρίπτουν εντελώς την συμβολική φύση της νοημοσύνης και αποφεύγουν την χρήση συμβολικής λογικής για τον τρόπο λήψης αποφάσεων του συστήματος, ενώ δίνουν έμφαση στις βασικές ανάγκες αυτό-κίνησης και αυτό-προστασίας του ρομποτικού σώματος. Παραδείγματα της προσέγγισης αυτής είναι τα ρομπότ BEAM («Biology, Electronics, Aesthetics, Mechanics») [21] τα οποία χρησιμοποιούν

αποκλειστικά αναλογικά κυκλώματα και όχι μικροεπεξεργαστές για την λήψη αποφάσεων, και τα οχήματα Braitenberg του ιταλό-αυστριακού κυβερνητιστή Valentino Braitenberg, τα οποία περιγράφουμε στο Κεφ. 4.3.

2.5.5. Υπολογιστική Νοημοσύνη

Σε αντίθεση με την κλασική, συμβολική προσέγγιση της «Παλιάς Καλής Τεχνητής Νοημοσύνης», η Υπολογιστική Νοημοσύνη επιχειρεί την δημιουργία τεχνητής νόησης σε υπό-συμβολικό επίπεδο, μελετώντας τις δυνατότητες αναγνώρισης προτύπων διάφορων σύνθετων συστημάτων και τις συμπεριφορές που αναδύονται από τα συστήματα αυτά. Η Υπολογιστική Νοημοσύνη χρησιμοποιεί κυρίως τρία «εργαλεία»: τα ασαφή συστήματα, τα νευρωνικά δίκτυα και τους εξελικτικούς ή γενετικούς αλγορίθμους, αλλά δανείζεται στοιχεία από πολλές επιστήμες, όπως την θεωρία του χάους ή την ευφυΐα των αποκεντρωμένων συστημάτων (για παράδειγμα ευφυΐα σμηγών) [22].

Το εξελικτικό μέρος της υπολογιστικής νοημοσύνης περιλαμβάνει και ένα σχετικά νεοσύστατο επιστημονικό πεδίο που ονομάζεται «Τεχνητή Ζωή» («Artificial Life»). Το πεδίο αυτό αντλεί τις επιρροές του από την φύση, την εξελικτική θεωρία και τις αναδυόμενες συμπεριφορές και ένας από τους στόχους του είναι κοινός με την Τεχνητή Νοημοσύνη: η ανάπτυξη ενός ευφυούς συστήματος.

Στα πλαίσια της εργασίας αυτής, θα αναπτύξουμε αναλυτικότερα το πεδίο της Τεχνητής Ζωής. Πρώτα όμως θα παρουσιάσουμε τις απαραίτητες βιολογικές έννοιες και θα αναφέρουμε διάφορους τομείς της υπολογιστικής επιστήμης που έχουν επηρεαστεί και έχουν δανειστεί στοιχεία από την παρατήρηση της φύσης και την επιστήμη της βιολογίας.

3. Εξέλιξη και αναδυόμενες συμπεριφορές

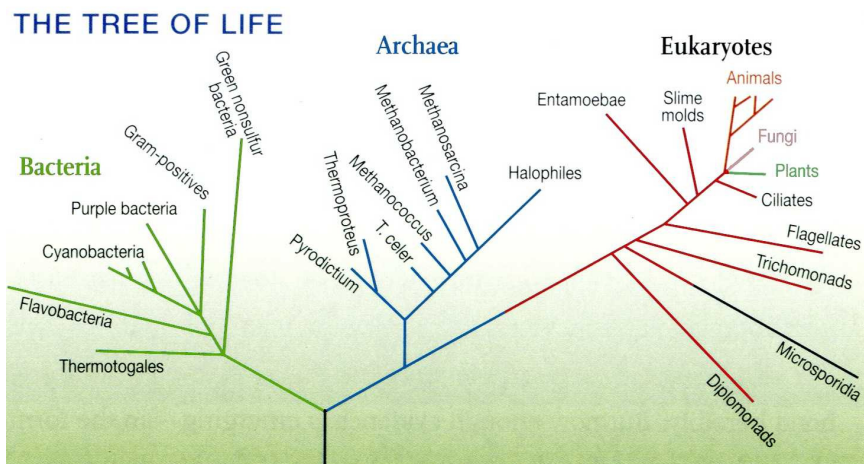
Στο κεφάλαιο αυτό θα αναφερθούμε σύντομα σε κάποιες έννοιες που θα φανούν χρήσιμες στη συνέχεια της εργασίας: τις έννοιες της βιολογικής εξέλιξης και των αναδυόμενων συμπεριφορών.

3.1. Η έννοια της εξέλιξης

Η εξέλιξη στην Βιολογία είναι η αλλαγή στο γενετικό υλικό μιας οικογένειας οργανισμών σε βάθος γενεών [23]. Ανάμεσα σε διαδοχικές γενιές οι αλλαγές του γενετικού υλικού είναι μικρές, αλλά σταδιακά συσσωρεύονται και οδηγούν σε σημαντικές διαφοροποιήσεις στην οικογένεια των οργανισμών και τελικά στην διάσπαση της οικογένειας σε διαφορετικά είδη. Τα δομικά στοιχεία που επιτρέπουν την εξέλιξη είναι τα γονίδια, οι μονάδες που μεταφέρουν τα κληρονομήσιμα χαρακτηριστικά ανάμεσα στις γενιές των οργανισμών.

Η εξέλιξη των ειδών ως ιδέα υπήρχε από πολύ παλιά, αλλά η επιστημονική εξέταση της στα πλαίσια της βιολογίας άρχισε τον 18^ο αιώνα. Μελετώντας τα απολιθώματα και την ποικιλομορφία των ζωντανών οργανισμών, οι περισσότεροι επιστήμονες της εποχής είχαν φτάσει στο συμπέρασμα ότι οι οργανισμοί αλλάζουν με τον χρόνο, ο μηχανισμός της διαδικασίας αυτής όμως δεν ήταν γνωστός. Στα μέσα του αιώνα ο Κάρολος Δαρβίνος (Charles Darwin) και ο Alfred Wallace διατύπωσαν ανεξάρτητα ο ένας από τον άλλον παρόμοιες θεωρίες που εξηγούσαν την εξέλιξη των ειδών ως αποτέλεσμα μιας διεργασίας που ο Δαρβίνος ονόμασε «φυσική επιλογή». Το 1859 με την δημοσίευση της εργασίας του «On the Origin of Species» [24], η θεωρία του Δαρβίνου έγινε ευρέως γνωστή και αποδεκτή από το σύνολο σχεδόν της επιστημονικής κοινότητας.

Αυτό που δεν κατάφερε να εξηγήσει ο Δαρβίνος ήταν ο μηχανισμός σε επίπεδο πλασμάτων με τον οποίο δούλευε η φυσική επιλογή. Ο μηχανισμός αυτός (τα γονίδια) εξηγήθηκε την δεκαετία του 1930 με την θεωρία της κληρονομικότητας του Gregor Mendel, και έτσι συμπληρώθηκε το κυρίως μέρος της σημερινής θεωρίας της εξέλιξης [25]. Οι επιστήμονες σήμερα, μετά από την ανακάλυψη του τρόπου μετάλλαξης των γονιδίων έχουν μια αρκετά ολοκληρωμένη εικόνα της διαδικασίας της εξέλιξης και της διαφοροποίησης των ειδών. Αν και δεν έχει εξηγηθεί πλήρως ακόμα η εμφάνιση της ζωής, τα κοινά στοιχεία μεταξύ των ζωντανών οργανισμών δείχνουν ότι όλοι προέρχονται από έναν κοινό πρόγονο (Σχ. 3.1).



Σχέδιο 3.1, Μια ιδιαίτερα απλοποιημένη μορφή του «δένδρου της ζωής»

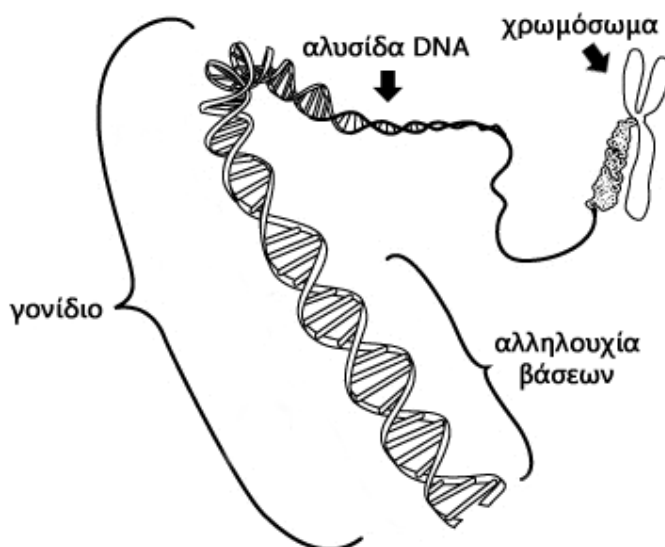
Η σύγχρονη θεωρία της εξέλιξης υποστηρίζει ότι οι μηχανισμοί που προκαλούν την εξέλιξη είναι δύο: η φυσική επιλογή και η γενετική διασπορά. Η φυσική επιλογή είναι αποτέλεσμα του γεγονότος ότι τα γονίδια των οργανισμών που αναπαράγονται πιο αποτελεσματικά (δηλαδή που καταφέρνουν για παράδειγμα να επιβιώσουν περισσότερο ή να κάνουν περισσότερους απογόνους) διατηρούνται και πολλαπλασιάζονται στον πληθυσμό σε βάθος χρόνου. Η γενετική διασπορά επιτρέπει την εμφάνιση νέων γονιδίων και την ποικιλομορφία που είναι απαραίτητη για την δημιουργία και την εξέλιξη των ειδών. Παρακάτω θα αναφερθούμε πιο λεπτομερώς στους μηχανισμούς που δημιουργούν τα φαινόμενα της φυσικής επιλογής και της γενετικής διασποράς.

3.1.1. Βασική ορολογία

Η εξέλιξη βασίζεται στην ιδιότητα των κληρονομήσιμων χαρακτηριστικών να μεταφέρονται από γενιά σε γενιά μεταξύ των οργανισμών. Τα χαρακτηριστικά αυτά ελέγχονται από τα γονίδια των οργανισμών. Το σύνολο των γονιδίων του οργανισμού ονομάζεται *γονότυπος*, ενώ το σύνολο των εξωτερικών χαρακτηριστικών του οργανισμού ονομάζεται *φαινότυπος*. Ο φαινότυπος εξαρτάται από τον γονότυπο του κάθε οργανισμού και το περιβάλλον στο οποίο αναπτύσσεται.

Η μεταφορά του γενετικού υλικού ανάμεσα στις γενιές γίνεται μέσω του *DNA* (Σχ. 3.2). Το DNA είναι ένα πολυμερές αποτελούμενο από τέσσερα είδη βάσεων, η αλληλουχία των οποίων στο μόριο του DNA κωδικοποιεί την γενετική πληροφορία. Μέσα στα κύτταρα των οργανισμών, οι αλυσίδες του DNA συσπειρώνονται και ονομάζονται *χρωμοσώματα*. Μια συγκεκριμένη θέση ενός χρωμοσώματος ονομάζεται *γενετικός τόπος*. Αν η αλληλουχία του DNA σε έναν συγκεκριμένο γενετικό τόπο διαφέρει ανάμεσα σε συγκεκριμένους οργανισμούς, οι διάφορες

παραλλαγές ονομάζονται *αλληλόμορφα*. Η αλληλουχία του DNA μπορεί να μεταβληθεί μέσω *μεταλλάξεων*, δημιουργώντας νέα αλληλόμορφα.



Σχέδιο 3.2, Χρωμόσωμα και DNA

3.1.2. Φυσική επιλογή

Η φυσική επιλογή είναι ένας από τους δύο κύριους μηχανισμούς της εξέλιξης. Φυσική επιλογή ονομάζουμε την διαδικασία κατά την οποία τα κληρονομήσιμα χαρακτηριστικά ενός πληθυσμού οργανισμών που αυξάνουν τις πιθανότητες του οργανισμού που τα φέρει να επιβιώσει και να αναπαραχθεί επιτυχώς, πληθαίνουν σε αριθμό σε βάθος γενεών του πληθυσμού αυτού.

Η φυσική επιλογή ως μηχανισμός προκύπτει από τα εξής δεδομένα που ισχύουν σε κάθε πληθυσμό οργανισμών [26].

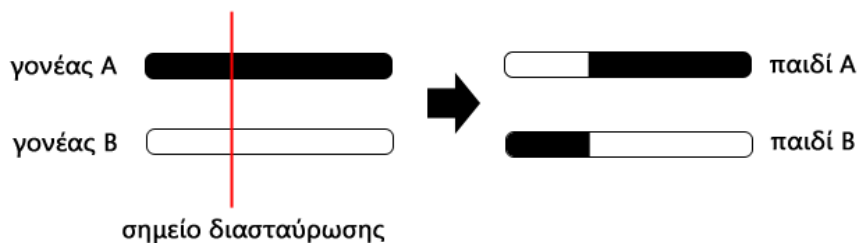
- *Κληρονομικότητα*: Ένα μέρος των χαρακτηριστικών του κάθε οργανισμού προέρχεται από τον γενετικό του κώδικα, ο οποίος είναι κληρονομήσιμος.
- *Θνησιμότητα*: Οι φυσικοί πόροι του περιβάλλοντος διαβίωσης του πληθυσμού δεν είναι αρκετοί για να επιβιώσουν όλοι οι απόγονοι του κάθε οργανισμού, οπότε ένα μέρος των οργανισμών αναγκαστικά πεθαίνουν.
- *Ποικιλομορφία*: Οι οργανισμοί διαφέρουν μεταξύ τους, τόσο στα εξωτερικά χαρακτηριστικά τους (φαινότυπος) όσο και στον γενετικό κώδικα (γονότυπος).

Αναγκαστικά λοιπόν, εφόσον δεν μπορούν να επιβιώσουν όλοι οι οργανισμοί και η επιβίωση δεν εξαρτάται μόνο από την τύχη αλλά και από τα χαρακτηριστικά του οργανισμού τα οποία

κληρονομούνται, τα γονίδια που δίνουν σε κάποιον οργανισμό χαρακτηριστικά που τον βοηθάνε να επιβιώσει και να αναπαραχθεί θα κληρονομούνται από οργανισμό σε οργανισμό και από γενιά σε γενιά και θα αυξηθεί η σχετική τους συχνότητα στον πληθυσμό.

Ο μηχανισμός της φυσικής επιλογής λοιπόν εξαρτάται από τα γονίδια, τα οποία είναι η βασική μονάδα της κληρονομικότητας σε έναν οργανισμό. Όταν δυο οργανισμοί αναπαράγονται μεταξύ τους και παράγουν απογόνους, το γενετικό υλικό των απογόνων είναι μια μίξη του γενετικού υλικού των γονέων. Με τον τρόπο αυτό τα γονίδια κληρονομούνται και οι οργανισμοί διαφοροποιούνται μεταξύ τους.

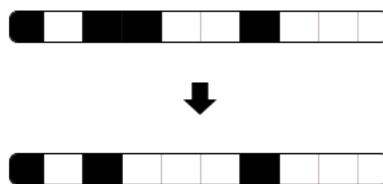
Ένας από τους τρόπους μίξης των γενετικών υλικών κατά την αναπαραγωγή δύο οργανισμών είναι η μέθοδος της *διασταύρωσης* («crossover»), στην οποία επιλέγεται ένα τυχαίο σημείο διασταύρωσης και συνδυάζεται το γενετικό υλικό του ενός γονέα μέχρι το σημείο αυτό και του άλλου γονέα από το σημείο αυτό και μετά [27]. Μια απλή περίπτωση διασταύρωσης φαίνεται στο Σχ. 3.3.



Σχέδιο 3.3, Απλή περίπτωση διασταύρωσης γενετικού υλικού

3.1.3. Γενετική διασπορά

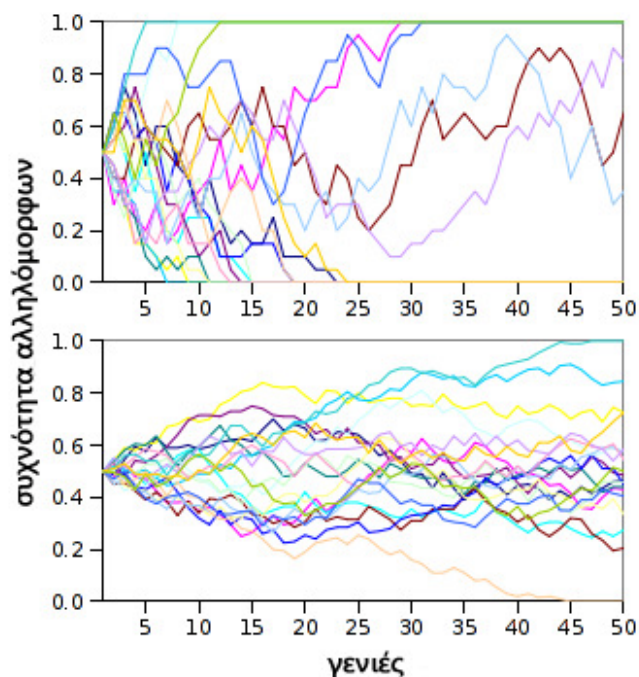
Κατά την διαδικασία της αναπαραγωγής, εκτός από την διασταύρωση του γενετικού υλικού, υπάρχει πιθανότητα να συμβούν και *μεταλλάξεις*. Μετάλλαξη ονομάζεται η τυχαία αλλαγή στην ακολουθία των βάσεων στο DNA [27]. Αν η μετάλλαξη συμβεί σε περιοχή του DNA που κωδικοποιεί κάποιο γονίδιο, πιθανόν η αλλαγή αυτή του γονότυπου να επηρεάσει και τα εξωτερικά χαρακτηριστικά του οργανισμού.



Σχέδιο 3.4, Μετάλλαξη σε ένα μέρος της αλυσίδας του DNA

Η πιθανότητα μετάλλαξης εισάγει την τύχη στην διαδικασία της εξέλιξης. Η τύχη υπάρχει και σε άλλες πτυχές της εξέλιξης: η επιλογή των αλληλόμορφων του κάθε γονέα που μεταφέρονται στους απογόνους στην αναπαραγωγή είναι τυχαία, και σίγουρα η τύχη παίζει ρόλο στην επιβίωση του κάθε οργανισμού. Η αλλαγή που παρατηρείται στις συχνότητες των αλληλόμορφων σε έναν πληθυσμό οργανισμών λόγω των τυχαίων αυτών φαινομένων ονομάζεται *γενετική διασπορά*, και είναι μαζί με την φυσική επιλογή οι δύο κύριοι μηχανισμοί της εξέλιξης [28]. Η διασπορά σε έναν πληθυσμό συνεχίζεται μέχρι ένα αλληλόμορφο να βρίσκεται στο σύνολο του πληθυσμού, φαινόμενο που ονομάζεται *σταθεροποίηση* του αλληλόμορφου.

Η σχετική σημασία της φυσικής επιλογής και της γενετικής διασποράς ως μηχανισμοί που προωθούν την εξέλιξη είναι ένα ζήτημα που ερευνάται ακόμα στην σύγχρονη βιολογία, με κάποιους επιστήμονες να υποστηρίζουν ότι η διασπορά παίζει πολύ μικρό ρόλο και άλλους να πιστεύουν ότι οι περισσότερες αλλαγές στο γενετικό υλικό οφείλονται στην γενετική διασπορά [29]. Γενικά η επιρροή της διασποράς είναι μεγαλύτερη σε μικρούς πληθυσμούς οργανισμών, ενώ σε μεγαλύτερους πληθυσμούς ευνοείται ο μηχανισμός της φυσικής επιλογής. Μια προσομοίωση της γενετικής διασποράς σε μικρό πληθυσμό (10 οργανισμοί) και μεγαλύτερο (100 οργανισμοί) φαίνεται στο Σχ. 3.5.



Σχέδιο 3.5, Προσομοίωση της γενετικής διασποράς σε δυο πληθυσμούς

3.2. Αναδυόμενες συμπεριφορές

Οι *αναδυόμενες συμπεριφορές* (emergent behaviors) είναι ένα από τα χαρακτηριστικά που εμφανίζονται στα σύνθετα συστήματα. Πρόκειται για νέες και συνεπείς δομές και πρότυπα που προκύπτουν από την αλληλεπίδραση μεταξύ των στοιχείων του συστήματος. Χαρακτηριστικό των αναδυόμενων συμπεριφορών είναι η εξαιρετικά σύνθετη δομή που προκύπτει από μικρό αριθμό απλών κανόνων αλληλεπίδρασης [30, 31]. Αναδυόμενες συμπεριφορές παρατηρούνται τόσο σε συστήματα των οποίων οι κανόνες αλληλεπίδρασης είναι πλήρως ορισμένοι όσο και σε συστήματα με πιο ασαφείς κανόνες όπως τα οικονομικά συστήματα ή τα συστήματα ηθικής.

Ακόμα και για απλούς κανόνες και μικρό αριθμό αλληλεπιδρώντων στοιχείων, οι αναδυόμενες συμπεριφορές είναι δύσκολο να προβλεφθούν εκ των προτέρων γιατί ο αριθμός των αλληλεπιδράσεων μεταξύ των στοιχείων αυξάνεται με συνδυαστικό ρυθμό και όχι γραμμικό. Για παράδειγμα, για ένα σύστημα 20 μορίων, ένας σύγχρονος υπολογιστής δεν μπορεί καν να απαριθμήσει τους δυνατούς συνδυασμούς μεταξύ των μορίων, πόσο μάλλον να προβλέψει τα αποτελέσματα των αλληλεπιδράσεων σε βάθος χρόνου [32].

Το φαινόμενο της παρουσίασης των αναδυόμενων συμπεριφορών σε σύνθετα συστήματα ονομάζεται *ανάδυση* (emergence). Η έννοια της ανάδυσης δεν είναι νέα - αντίστοιχες έννοιες μελετούνται τουλάχιστον από την εποχή του Αριστοτέλη. Σήμερα με την έννοια της ανάδυσης ασχολούνται διάφορες επιστήμες – από την Φιλοσοφία και την Θρησκευσιολογία μέχρι την Αρχιτεκτονική και την επιστήμη των υπολογιστών. Στη συνέχεια θα δούμε κάποια παραδείγματα αναδυόμενων συμπεριφορών που παρατηρούνται στην φύση, την κοινωνία και τις επιστήμες.

3.2.1. Αναδυόμενες συμπεριφορές στην φύση

Μπορούμε να διαχωρίσουμε τις αναδυόμενες συμπεριφορές στη φύση σε δυο κατηγορίες: σε αυτές που παρατηρούνται σε μη ζωντανά συστήματα, όπως η αλληλεπίδραση των ατόμων με βάση τους κανόνες της κβαντομηχανικής και σε αυτές που παρατηρούνται σε συστήματα ζωντανών οργανισμών.

Η πρώτη κατηγορία αναφέρεται σε συστήματα Φυσικής, και χρησιμοποιείται κυρίως για να περιγράψει διαδικασίες που συμβαίνουν σε μικροσκοπικό ή μακροσκοπικό επίπεδο. Κάποια παραδείγματα αναδυόμενων συμπεριφορών στην Φυσική είναι:

- Οι κανόνες της Κλασικής Μηχανικής, οι οποίοι προκύπτουν αναδυόμενοι από τις αλληλεπιδράσεις των ατόμων και των μορίων και τους κανόνες που ισχύουν για τις αλληλεπιδράσεις αυτές, όπως η κβαντομηχανική [32].
- Ο καιρός, ο οποίος είναι ένα ιδιαίτερα σύνθετο σύστημα που προκύπτει από τις αλληλεπιδράσεις της κατάστασης διάφορων μεγεθών (όπως η ατμοσφαιρική πίεση και η υγρασία) σε διαφορετικά σημεία της Γης.
- Τα χρώματα. Τα βασικά σωματίδια δεν απορροφούν συγκεκριμένες κυματομορφές του φωτός, οπότε δεν έχουν χρώμα. Μόνο όταν διατάσσονται σε μορφή ατόμων απορροφούν ή εκπέμπουν συγκεκριμένες κυματομορφές οπότε μπορούμε να τους αποδώσουμε την ιδιότητα του χρώματος.
- Η τριβή, η οποία δεν εμφανίζεται μεταξύ των βασικών σωματιδίων, αλλά μόνο σε πιο σύνθετες μορφές ύλης μέσω των αλληλεπιδράσεων των σωματιδίων.

Η δεύτερη κατηγορία περιλαμβάνει τα ζωντανά και βιολογικά συστήματα. Χαρακτηριστικό παράδειγμα της ανάδυσης στην κατηγορία αυτή είναι οι φωλιές των μυρμηγκιών. Το κάθε μυρμήγκι δεν δέχεται εντολές από κάποια κεντρική αρχή, αλλά αντιδράει ανάλογα με το τοπικό του περιβάλλον, για παράδειγμα τις χημικές οσμές που αφήνουν τα άλλα μυρμηγκία και οι εχθροί της φωλιάς, την οσμή του φαγητού ή των απορριμμάτων. Παράλληλα, το κάθε μυρμήγκι αφήνει και αυτό ένα ίχνος χημικής οσμής. Παρά τους απλούς κανόνες που ακολουθεί κάθε μυρμήγκι, το οποίο δεν έχει επίγνωση της συνολικής φωλιάς, οι κοινωνίες των μυρμηγκιών ως σύνολο εμφανίζουν σύνθετη συμπεριφορά και έχει αποδειχθεί ότι λύνουν ακόμα και προβλήματα γεωμετρίας. Οι κοινωνίες μυρμηγκιών είναι ένας δημοφιλής τρόπος μελέτης και προσομοίωσης των αναδυόμενων συμπεριφορών [33].

Άλλο χαρακτηριστικό παράδειγμα αναδυόμενης συμπεριφοράς στην φύση είναι τα σμήνη των πτηνών (Σχ. 3.6). Η πτήση του κάθε πτηνού που ανήκει στο σμήνος εξαρτάται από λίγους, πολύ απλούς κανόνες της μορφής «κράτα απόσταση X από το διπλανό σου πτηνό» και «ακολούθα σε απόσταση Y το μπροστινό σου πτηνό». Η πολύπλοκη μορφολογία του συνολικού σμήνους είναι ένα αναδυόμενο χαρακτηριστικό.



Σχέδιο 3.6, Σμήνος πτηνών

3.2.2. Αναδυόμενες συμπεριφορές στην κοινωνία και τις επιστήμες

Αναδυόμενα χαρακτηριστικά παρατηρούνται και σε διάφορους τομείς της κοινωνίας και των επιστημών. Η αλληλεπίδραση μεταξύ των ανθρώπων που ανήκουν σε μια κοινωνία είναι ιδιαίτερα σύνθετη, και οι κανόνες που ακολουθούν οι αλληλεπιδράσεις αυτές είναι ασαφείς και διαρκώς αλλάζουν. Τα ηθικά συστήματα, η γλώσσα και η θρησκεία μπορούν όλα να χαρακτηριστούν αναδυόμενα χαρακτηριστικά του συστήματος της ανθρώπινης κοινωνίας. Άλλα παραδείγματα της ανάδυσης στην κοινωνία είναι τα παρακάτω.

- Σε πόλεις στις οποίες δεν υπάρχει προκαθορισμένο σχέδιο κτίσεως, η αρχιτεκτονική των κτιρίων και της δομής της πόλης μπορεί να θεωρηθεί ως αναδυόμενο χαρακτηριστικό ενός αυτό-οργανούμενου συστήματος.
- Το Internet είναι ένα άλλο παράδειγμα ανάδυσης [34]. Ενώ δεν υπάρχει κάποια κεντρική δομή που να το ελέγχει, η αλληλεπίδραση μεταξύ των συστημάτων υπολογιστών που αποτελούν το internet οδηγεί στην δημιουργία σύνθετων συμπεριφορών ως προς την κίνηση του δικτύου ή τον αριθμό των συνδέσμων.
- Οι αναδυόμενες συμπεριφορές είναι σημαντικό στοιχείο της υπολογιστικής πολυπλοκότητας ως προσέγγιση της Τεχνητής Νοημοσύνης.

4. Επιρροή της φύσης στην επιστήμη των Υπολογιστών

Στο κεφάλαιο αυτό θα μελετήσουμε κάποιους τομείς της επιστήμης υπολογιστών που αντλούν επιρροές από την φύση και την βιολογία. Δυο από αυτούς είναι ιδιαίτερα σημαντικοί για την έρευνα Τεχνητής Νοημοσύνης: οι γενετικοί ή εξελικτικοί αλγόριθμοι και τα τεχνητά νευρωνικά δίκτυα. Δυο άλλοι τομείς, τα κυτταρικά αυτόματα και τα συστήματα Lindenmayer αποτελούν χρήσιμα εφόδια για την εξελικτική προσέγγιση στην Τεχνητή Νοημοσύνη η οποία μας απασχολεί στην εργασία αυτή. Τέλος, θα εξετάσουμε σύντομα τα οχήματα Braitenberg τα οποία αποτελούν ένα ενδιαφέρον παράδειγμα «ενσώματης νοημοσύνης».

4.1. Γενετικοί αλγόριθμοι

Μια από τις πιο δημοφιλείς εφαρμογές φυσικών εννοιών στην επιστήμη των Υπολογιστών είναι οι γενετικοί αλγόριθμοι, τους οποίους θα μελετήσουμε συνοπτικά παρακάτω.

4.1.1. Η έννοια των γενετικών αλγορίθμων

Οι *γενετικοί* (ή *εξελικτικοί*) αλγόριθμοι ανήκουν στην κατηγορία των στοχαστικών ή μη-ντετερμινιστικών μεθόδων αναζήτησης. Η μέθοδος αναζήτησης που ακολουθούν είναι επηρεασμένη από το μοντέλο της εξέλιξης μέσω της φυσικής επιλογής, το οποίο εξετάσαμε στο Κεφ. 3.1. Πράγματι, αντί να λειτουργεί με μια μόνο λύση όπως οι περισσότεροι στοχαστικοί αλγόριθμοι αναζήτησης, δημιουργεί έναν πληθυσμό λύσεων τον οποίο βελτιστοποιεί για το συγκεκριμένο πρόβλημα [35].

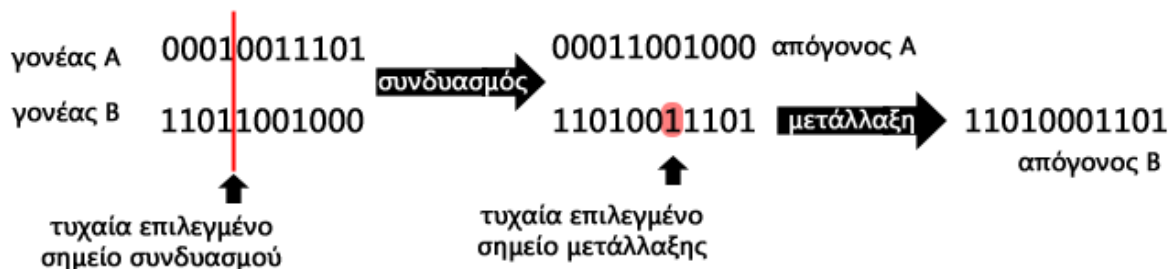
Για να εφαρμόσουμε έναν γενετικό αλγόριθμο σε ένα πρόβλημα πρέπει να είναι δυνατή η κωδικοποίηση των λύσεων του προβλήματος σε μορφή αποθηκεύσιμη σε υπολογιστή, την οποία ονομάζουμε *χρωμόσωμα* ή *γονίδιο* ή *γονότυπο*, ενώ η λύση του προβλήματος που αντιστοιχεί σε κάποια κωδικοποίηση είναι ο *φαινότυπος* της. Για να βρεθεί η βέλτιστη λύση, οι πιθανές λύσεις αναπαράγονται μεταξύ τους, συνδυάζονται και μεταλλάσσονται, όπως τα γονίδια στην φύση. Απαραίτητη έννοια για έναν γενετικό αλγόριθμο είναι επίσης η *συνάρτηση προσαρμογής*. Αυτή είναι μια συνάρτηση που αποτιμά πόσο κοντά βρίσκεται μια λύση στην επιθυμητή λύση του προβλήματός μας, και πρέπει να επιλεγθεί προσεκτικά κατά την σχεδίαση του γενετικού αλγορίθμου.

4.1.2. Περιγραφή των βημάτων ενός γενετικού αλγορίθμου

Τα βήματα ενός γενετικού αλγορίθμου είναι τα εξής:

- 1) Αναπαράσταση των λύσεων του προβλήματος και επιλογή της συνάρτησης προσαρμογής u .
- 2) Δημιουργία ενός τυχαίου αρχικού πληθυσμού $M(t)$, με $t = 0$.
- 3) Υπολογισμός της τιμής της συνάρτησης προσαρμογής $u(m)$ για κάθε μέλος m του πληθυσμού $M(t)$.
- 4) Επιλογή δυο μελών του πληθυσμού ως γονείς με βάση την τιμή της συνάρτησης προσαρμογής $u(t)$.
- 5) Αναπαραγωγή των γονέων που επιλέχθηκαν, και αντικατάσταση δυο μελών του πληθυσμού με τους δυο απογόνους που προέκυψαν, έτσι ώστε να προκύψει η επόμενη γενιά πληθυσμού $M(t+1)$.
- 6) Αν δεν έχουν επιτευχθεί οι συνθήκες τερματισμού της αναζήτησης, επανάληψη της διαδικασίας από το βήμα 3.

Η αναπαράσταση των λύσεων του προβλήματος γίνεται συνήθως σε μορφή συμβολοσειράς δυαδικών ψηφίων (bit), δηλαδή ακολουθίας 0 και 1. Παρ' όλα αυτά, είναι δυνατή οποιασδήποτε μορφής αναπαράσταση (για παράδειγμα σε μορφή λίστας, πίνακα ή δένδρου) αρκεί να μπορεί να αναπαρασταθεί πλήρως μια λύση του προβλήματος στην επιλεγμένη μορφή και να ορίσουμε τις πράξεις του συνδυασμού και της μετάλλαξης για την μορφή αυτή. Ο συνδυασμός και η μετάλλαξη ακολουθιών bit φαίνεται στο παρακάτω σχήμα (Σχ. 4.1).



Σχέδιο 4.1, Συνδυασμός και μετάλλαξη σε ακολουθίες bit

Ένα σημαντικό ζήτημα στους γενετικούς αλγορίθμους είναι ο τρόπος με τον οποίο γίνεται η επιλογή των γονέων που θα αναπαραχθούν στο βήμα 4. Διάφοροι πιθανοί τρόποι επιλογής είναι:

- Επιλογή των καλύτερων λύσεων
- Επιλογή με την μέθοδο της «ρουλέτας»
- Επιλογή με την μέθοδο του «τουρνουά»
- Επιλογή με κάποια στοχαστική συνάρτηση

Συνήθως χρησιμοποιείται η μέθοδος της «ρουλέτας», κατά την οποία σε κάθε μέλος του πληθυσμού αντιστοιχίζεται μια πιθανότητα επιλογής p ανάλογη με την τιμή της συνάρτησης προσαρμογής u του μέλους αυτού (και έτσι ώστε το άθροισμα όλων των p να είναι μοναδιαίο), και οι γονείς που αναπαράγονται επιλέγονται με βάση την πιθανότητα p . Στην μέθοδο του «τουρνουά», επιλέγεται ένας αριθμός ατόμων με την μέθοδο της «ρουλέτας» και οι καλύτεροι από αυτούς (δηλαδή αυτοί που έχουν την μεγαλύτερη τιμή συνάρτησης προσαρμογής) είναι αυτοί που επιλέγονται τελικά για αναπαραγωγή [36].

Η μέθοδος της ρουλέτας δημιουργεί δυο προβλήματα [37]. Πρώτον, αν υπάρχει στον αρχικό πληθυσμό ένα μέλος με πολύ υψηλή αρχική τιμή συνάρτησης προσαρμογής, η πιθανότητα επιλογής του είναι πολύ μεγάλη και αυτό μπορεί να οδηγήσει στην γρήγορη σύγκλιση του συνόλου του πληθυσμού στα γονίδια του μέλους αυτού. Αυτό δεν είναι επιθυμητό γιατί δεν διατηρείται η ποικιλομορφία του χώρου αναζήτησης με αποτέλεσμα να μην μπορεί ο γενετικός αλγόριθμος να φτάσει στην ζητούμενη βέλτιστη λύση. Επίσης, όσο προχωράει ο αλγόριθμος μικραίνουν οι διαφορές ανάμεσα στις τιμές της συνάρτησης προσαρμογής, οπότε σε πολλές περιπτώσεις υπάρχουν αρκετά μέλη με την ίδια ή πολύ κοντινή πιθανότητα επιλογής και ο αλγόριθμος δεν μπορεί να συνεχίσει αποτελεσματικά.

Ένας τρόπος αντιμετώπισης των προβλημάτων αυτών είναι η αφαίρεση της τιμής της συνάρτησης προσαρμογής $u(w)$, όπου w είναι το χειρότερο πλάσμα του πληθυσμού, από όλες τις τιμές της συνάρτησης u , με αποτέλεσμα να μειώνεται η διαφορά ανάμεσα στις πιθανότητες επιλογής των πιο επιτυχημένων ατόμων και επιπλέον να έχουμε μια κατανομή που αρχίζει από 0. Άλλος τρόπος είναι η ανάθεση πιθανοτήτων p σταθερής διαφοράς ανάλογα με την κατάταξη των μελών του πληθυσμού (το πιο επιτυχημένο μέλος έχει πιθανότητα επιλογής 0.6, το επόμενο 0.55, 0.50 και λοιπά).

Ένα δεύτερο ζήτημα είναι η επιλογή των ατόμων που θα αντικατασταθούν από τους απογόνους που δημιουργούνται στο βήμα 5 [36]. Διάφοροι πιθανοί τρόποι επιλογής είναι:

- Αντικατάσταση των χειρότερων
- Αντικατάσταση των καλύτερων
- Αντικατάσταση των γονέων
- Αντικατάσταση τυχαίων ατόμων
- Αντικατάσταση των πιο ομοίων

Συνήθως χρησιμοποιείται η αντικατάσταση των χειρότερων μελών του πληθυσμού, αλλά αυτό μπορεί πάλι να οδηγήσει στο πρόβλημα της γρήγορης σύγκλισης. Ένας καλύτερος τρόπος επιλογής είναι η αντικατάσταση των ατόμων που έχουν τα γονίδια με την μεγαλύτερη ομοιότητα με τους νέους απογόνους, μέθοδος που ονομάζεται και «δημιουργία πλήθους τύπου DeJong», από τον Kenneth DeJong που ανέλυσε πρώτος το πρόβλημα αυτό στην διδακτορική του διατριβή [36].

4.1.3. Εφαρμογές

Οι γενετικοί αλγόριθμοι αναζήτησης είναι ιδιαίτερα χρήσιμοι σε αρκετές περιπτώσεις. Είναι απλοί και μπορούν να εφαρμοστούν σε πολλά προβλήματα, αρκεί να μπορεί να βρεθεί μια κωδικοποίηση της λύσης του προβλήματος και να μπορεί να διατυπωθεί μια συνάρτηση. Είναι όμως γενικά πιο αργοί από τους κλασικούς, μη-στοχαστικούς αλγορίθμους αναζήτησης, οπότε δεν ενδείκνυται η χρήση τους σε όλες τις περιπτώσεις. Γενικά οι γενετικοί αλγόριθμοι δίνουν καλύτερα αποτελέσματα από τους κλασικούς αλγορίθμους όταν:

- Ο χώρος αναζήτησης είναι εκτενής και όχι πλήρως κατανοητός,
- Το πρόβλημα είναι δύσκολο να μοντελοποιηθεί μαθηματικά,
- Ο χώρος αναζήτησης έχει πολλά τοπικά μέγιστα και τοπικά ελάχιστα και οι κλασικοί αλγόριθμοι «κολλάνε» σε κάποιο από αυτά,
- Όταν οι κλασικοί αλγόριθμοι αποτυγχάνουν.

Τέλος ένα πλεονέκτημα των γενετικών αλγορίθμων είναι ότι μπορούν εύκολα να αξιοποιήσουν τις δυνατότητες παραλληλισμού των σύγχρονων υπολογιστών [36].

Οι γενετικοί αλγόριθμοι χρησιμοποιούνται σήμερα σε πολλές εφαρμογές, όπως η μοντελοποίηση και η μελέτη οικονομικών και κοινωνικών συστημάτων, η λήψη αποφάσεων (κυρίως οικονομικών) σε επιχειρήσεις και η βελτιστοποίηση λύσεων σε διάφορα σύνθετα προβλήματα, όπως το γνωστό πρόβλημα των πλανόδιων πωλητών (TSP, «Travelling Salesman

Problem»). Τέλος, οι γενετικοί αλγόριθμοι είναι χρήσιμοι στην Τεχνητή Νοημοσύνη και χρησιμοποιούνται εκτενώς σε συστήματα μηχανικής μάθησης.

4.2. Νευρωνικά δίκτυα

4.2.1. Εισαγωγή

Τα βιολογικά νευρωνικά δίκτυα είναι ομάδες χημικά διασυνδεδεμένων μονάδων που ονομάζονται *νευρώνες*. Οι συνδέσεις μεταξύ των νευρώνων ονομάζονται *συνάψεις*, και μέσω των συνάψεων μεταφέρονται τα σήματα επικοινωνίας των νευρώνων, που είναι κυρίως ηλεκτρικά. Οι εγκέφαλοι των ζωντανών πλασμάτων αποτελούνται από έναν τεράστιο αριθμό νευρώνων (περίπου 10.000.000.000 για τον ανθρώπινο εγκέφαλο), διασυνδεδεμένων σε μεγάλο βαθμό [38]. Ως εκ τούτου, πρόκειται για ένα ιδιαίτερα σύνθετο σύστημα του οποίου τα χαρακτηριστικά ακόμα μελετώνται από τις αντίστοιχες επιστήμες.

Όταν αναφερόμαστε όμως στα «νευρωνικά δίκτυα» συνήθως εννοούμε τα τεχνητά νευρωνικά δίκτυα. Τα νευρωνικά δίκτυα αυτά είναι ένα μαθηματικό μοντέλο που μιμείται κάποια από τα χαρακτηριστικά της δομής και της λειτουργίας των βιολογικών νευρωνικών δικτύων. Η ενασχόληση με τα τεχνητά νευρωνικά δίκτυα (από εδώ και πέρα θα τα αναφέρουμε απλά ως «νευρωνικά δίκτυα») άρχισε με την έρευνα των McCulloch και Pitts το 1943, και αρκετοί γνωστοί επιστήμονες (όπως ο Marvin Minsky) ασχολήθηκαν με τον τομέα αυτό. Το 1969 ο Minsky και ο Papert δημοσίευσαν ένα άρθρο που αμφισβητούσε την ικανότητα των νευρωνικών δικτύων στην επίλυση προβλημάτων και το επιστημονικό ενδιαφέρον σχεδόν εξαφανίστηκε. Από τα μέσα της δεκαετίας του 1980 όμως όλο και περισσότεροι επιστήμονες Τεχνητής Νοημοσύνης άρχισαν να χρησιμοποιούν τα νευρωνικά δίκτυα με σημαντικά αποτελέσματα και η ερευνητική προσπάθεια στον τομέα σήμερα είναι ισχυρή [39].

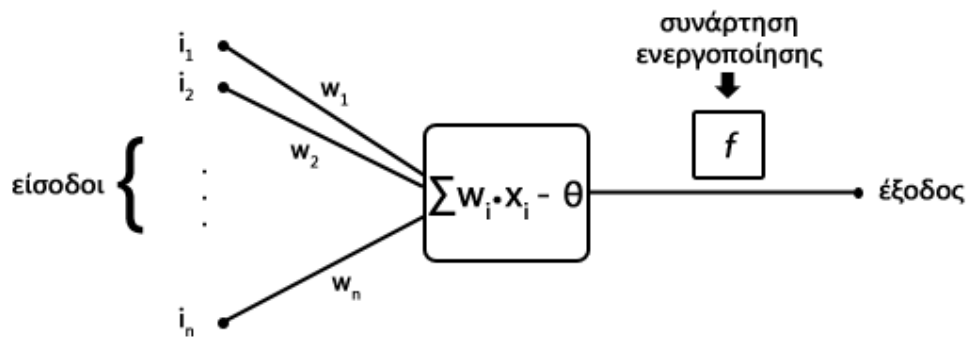
Τα τεχνητά νευρωνικά δίκτυα είναι ιδιαίτερα χρήσιμα στην υπό-συμβολική επεξεργασία πληροφοριών, και χρησιμοποιούνται σήμερα με μεγάλη επιτυχία σε διάφορους τομείς, κυρίως συναφείς με την Τεχνητή Νοημοσύνη. Η όραση υπολογιστών, η οπτική αναγνώριση χαρακτήρων, η φωνητική αναγνώριση και οι μηχανές αναζήτησης στο Internet είναι όλες τεχνολογίες που βασίζονται στα νευρωνικά δίκτυα.

4.2.2. Δομή νευρωνικών δικτύων

Τα δομικά στοιχεία των τεχνητών νευρωνικών δικτύων είναι τα *νευρόνια* (Σχ. 4.2) [40]. Το κάθε νευρόνιο έχει:

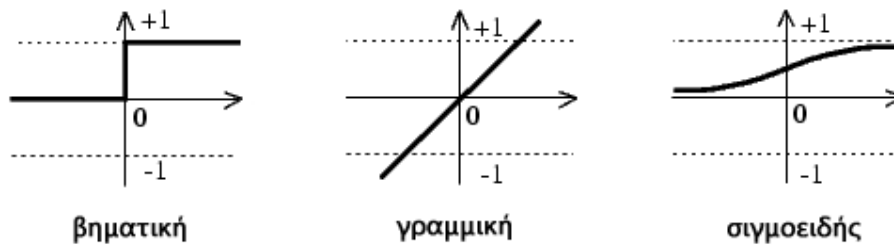
- Έναν αριθμό εισόδων x (είτε από δεδομένα, είτε από άλλα νευρόνια). Η κάθε είσοδος προέρχεται από μια σύνδεση η οποία έχει ένα ορισμένο *βάρος* w , θετικό ή και αρνητικό.
- Μια τιμή *κατωφλίου* φ . Η τιμή αυτή αφαιρείται από το άθροισμα όλων των εισόδων πολλαπλασιασμένων με τα αντίστοιχα βάρη, και προκύπτει η *ενεργοποίηση* ε του νευρονίου ($\varepsilon = \sum_i x_i w_i - \varphi$).
- Τέλος, μια *συνάρτηση ενεργοποίησης*, από την οποία περνάει η ενεργοποίηση ε του νευρονίου και προκύπτει η τιμή εξόδου.

Αν επιλέξουμε ως συνάρτηση ενεργοποίησης την βηματική (η έξοδος της οποίας είναι 0 αν έχει αρνητική είσοδο και 1 αν έχει μη αρνητική), το τεχνητό νευρόνιο θα έχει συμπεριφορά παρόμοια με τους βιολογικούς νευρώνες: όσο το άθροισμα των εισόδων παραμένει κάτω από την τιμή του κατωφλίου το νευρόνιο έχει μηδενική έξοδο, και όταν το άθροισμα υπερβεί το κατώφλι έχει μοναδιαία έξοδο.



Σχέδιο 4.2, Σχεδιάγραμμα ενός απλού τεχνητού νευρονίου

Γενικά η βηματική συνάρτηση ενεργοποίησης χρησιμοποιείται μόνο στα απλούστερα νευρωνικά δίκτυα. Κάποιες πιο συνηθισμένες συναρτήσεις ενεργοποίησης που χρησιμοποιούνται στην πράξη είναι η γραμμική και η σιγμοειδής συνάρτηση (Σχ. 4.3).



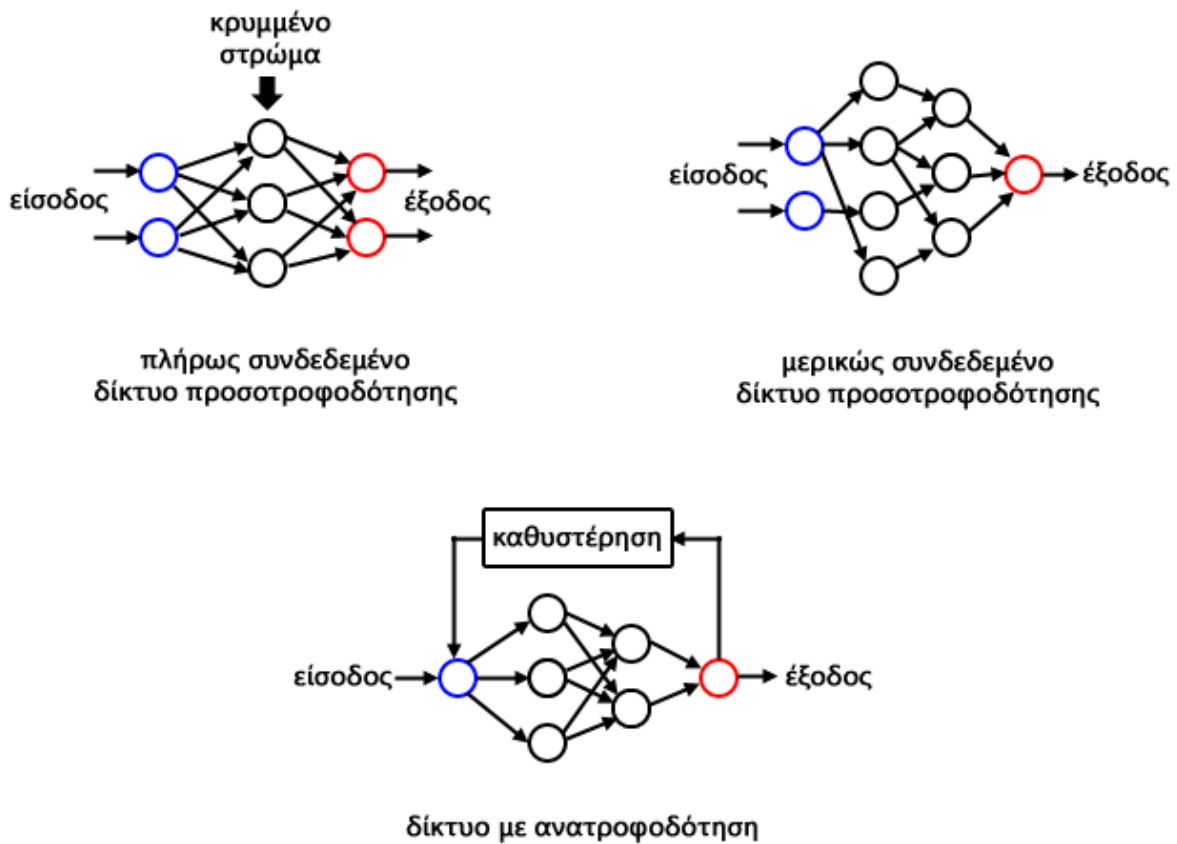
Σχέδιο 4.3, Συναρτήσεις ενεργοποίησης νευρονίων

4.2.3. Αρχιτεκτονική νευρωνικών δικτύων

Τα τεχνητά νευρωνικά δίκτυα αποτελούνται λοιπόν από μια ομάδα διασυνδεδεμένων τεχνητών νευρονίων. Το κύριο χαρακτηριστικό των νευρωνικών δικτύων, εκτός από την συνάρτηση ενεργοποίησης των νευρονίων που χρησιμοποιείται, είναι η τοπολογία του δικτύου και οι συνδέσεις ανάμεσα στα νευρόνια της τοπολογίας αυτής. Τα νευρόνια κάθε δικτύου είναι οργανωμένα σε *στρώματα*. Έχουμε το στρώμα εισόδου το οποίο δέχεται τις τιμές του από τα δεδομένα εισόδου και το στρώμα εξόδου που βγάζει στην έξοδό του την απόκριση του συστήματος. Ενδιάμεσα υπάρχουν ένα ή περισσότερα «κρυμμένα» στρώματα νευρονίων που παίρνουν την είσοδό τους από το προηγούμενο στρώμα και στέλνουν την έξοδό τους στο επόμενο.

Μια διάκριση της τοπολογίας των νευρωνικών δικτύων μπορεί να γίνει με βάση τις διασυνδέσεις των στρωμάτων [41]. Ένα δίκτυο στο οποίο κάθε κόμβος ενός στρώματος είναι διασυνδεδεμένος με όλους τους κόμβους του επόμενου στρώματος ονομάζεται *πλήρως διασυνδεδεμένο*. Αντίστοιχα ένα δίκτυο στο οποίο λείπουν κάποιες από τις διασυνδέσεις μεταξύ διαδοχικών στρωμάτων ονομάζεται *μερικώς διασυνδεδεμένο*. Συνήθως χρησιμοποιούνται πλήρως διασυνδεδεμένα νευρωνικά δίκτυα.

Μια άλλη διάκριση που μπορεί να γίνει είναι ανάμεσα στα νευρωνικά δίκτυα *προσοτροφοδότησης* και τα νευρωνικά δίκτυα *ανατροφοδότησης*. Τα τελευταία έχουν το στοιχείο της αναδρομής: περιέχουν έναν αναδρομικό βρόχο ο οποίος ανακυκλώνει την έξοδο του νευρωνικού δικτύου στην είσοδό του με κάποια καθυστέρηση (Σχ. 4.4).



Σχέδιο 4.4, Αρχιτεκτονικές νευρωνικών δικτύων

4.2.4. Μάθηση νευρωνικών δικτύων

Μια σημαντική ιδιότητα των νευρωνικών δικτύων είναι η δυνατότητα που έχουν για *μάθηση*. Η μάθηση (ή εκπαίδευση) των νευρωνικών δικτύων είναι η διαδικασία σταδιακής προσαρμογής των βαρών των συνδέσεων του δικτύου ώστε να επιτευχθεί η επιθυμητή συμπεριφορά του συστήματος (για παράδειγμα, η έξοδος του νευρωνικού δικτύου να δίνει τις αναμενόμενες τιμές). Ανάλογα με τις απαιτήσεις του προβλήματος που θέλουμε να λύσουμε και την διαδικασία που ακολουθούμε, η μάθηση των νευρωνικών δικτύων μπορεί να είναι *επιβλεπόμενη*, *μη επιβλεπόμενη* ή *ενισχυτική* [42].

Στην επιβλεπόμενη (ή αλλιώς *ενεργή*) μάθηση έχουμε ένα σύνολο εκπαίδευσης το οποίο αποτελείται από ζευγάρια τιμών $(x, y), x \in X, y \in Y$. Το ζητούμενο είναι να βρεθεί μια συνάρτηση $f: X \rightarrow Y$ που να επαληθεύει τα παραδείγματα. Υπάρχουν διάφορες τεχνικές επιβλεπόμενης μάθησης, η πιο γνωστή από τις οποίες είναι η μάθηση *ανάστροφης διάδοσης* (back propagation). Ο αλγόριθμος μάθησης ανάστροφης διάδοσης ανανεώνει σε κάθε βήμα τα βάρη των συνδέσεων έτσι ώστε να ελαχιστοποιείται το μέσο τετραγωνικό σφάλμα $e = \sum (\Delta y)^2$, όπου:

$$\Delta y = y - y_{out} ,$$

y είναι η επιθυμητή έξοδος για είσοδο x και

y_{out} είναι η έξοδος του νευρωνικού δικτύου για είσοδο x .

Η μετάδοση του σφάλματος και η αντίστοιχη ανανέωση των βαρών γίνεται διαδοχικά από στρώμα σε στρώμα, αρχίζοντας από το στρώμα εξόδου και προχωρώντας ανάποδα μέχρι το στρώμα εισόδου. Από αυτή την αναστροφή διάδοση παίρνει την ονομασία του ο αλγόριθμος. Η επιβλεπόμενη μάθηση χρησιμοποιείται κυρίως σε εφαρμογές αναγνώρισης προτύπων, όπως για παράδειγμα αναγνώριση χαρακτήρων ή ταξινόμηση δεδομένων. Το δίκτυο εκπαιδεύεται με ένα σύνολο δεδομένων εκπαίδευσης και στη συνέχεια μπορεί να χρησιμοποιηθεί για το ζητούμενο πρόβλημα αναγνώρισης.

Στην ενισχυτική μάθηση το σύστημα δέχεται όπως και στην επιβλεπόμενη μάθηση ένα σύνολο δεδομένων x . Αντί για την επιθυμητή απόκριση y δέχεται όμως από το περιβάλλον μια απόκριση a (που παίζει το ρόλο της επιβράβευσης ή της τιμωρίας). Ο στόχος του συστήματος είναι να μάθει να αντιδρά με τέτοιο τρόπο ώστε να μεγιστοποιήσει την επιβράβευση που δέχεται [43]. Για να το πετύχει αυτό ο αλγόριθμος μάθησης αυξάνει τις τιμές των βαρών του δικτύου που οδήγησαν σε επιβράβευση μειώνοντας παράλληλα τις τιμές των βαρών που οδήγησαν σε τιμωρία.

Τέλος στην μη επιβλεπόμενη (ή *αυτό-οργανούμενη*) μάθηση το νευρωνικό δίκτυο δέχεται ένα σύνολο δεδομένων x χωρίς να του δίνεται η επιθυμητή απόκριση ή κάποιου είδους ανατροφοδότηση από το περιβάλλον [43]. Ο στόχος είναι να δημιουργήσει το σύστημα εσωτερικές αναπαραστάσεις και κατηγοριοποιήσεις των δεδομένων εισόδου ώστε να μπορέσει για παράδειγμα να πάρει αποφάσεις ή να μαντέψει τις επόμενες εισόδους.

4.2.5. Εφαρμογές

Τα νευρωνικά δίκτυα βρίσκουν εφαρμογές σε προβλήματα όπου υπάρχει κάποια σχεσιακή σύνδεση ανάμεσα στα δεδομένα εισόδου και τα δεδομένα εξόδου. Ακόμα και αν η σχέση αυτή είναι ιδιαίτερα σύνθετη και δεν μπορεί να μοντελοποιηθεί τα νευρωνικά δίκτυα έχουν την δυνατότητα να την αναπαραστήσουν. Οι τομείς στους οποίους χρησιμοποιούνται κυρίως τα νευρωνικά δίκτυα είναι οι εξής:

- Αναζήτηση πληροφοριών σε εκτενείς βάσεις δεδομένων. Ένα γνωστό παράδειγμα είναι η μηχανή αναζήτησης της Google η οποία βασίζεται στα νευρωνικά δίκτυα.

- Πρόβλεψη μελλοντικών μεγεθών με βάση ένα ιστορικό προηγούμενων δεδομένων. Νευρωνικά δίκτυα χρησιμοποιούνται σήμερα εκτενώς για εκτίμηση του καιρού ή της πορείας του χρηματιστηρίου.
- Αναγνώριση προτύπων. Τα νευρωνικά δίκτυα είναι ιδιαίτερα χρήσιμα σε εφαρμογές αναγνώρισης προτύπων όπως για παράδειγμα η οπτική αναγνώριση χαρακτήρων (OCR), η όραση υπολογιστών, η φωνητική αναγνώριση ή η αναγνώριση προσώπων σε φωτογραφίες.
- Τέλος, τα νευρωνικά δίκτυα χρησιμοποιούνται εκτενώς σε ιατρικές εφαρμογές, όπου αναλύουν τα ιατρικά δεδομένα και συμπτώματα των ασθενών ώστε να βοηθήσουν τον ιατρό να καταλήξει σε κάποια διάγνωση.

4.3. Κυτταρικά αυτόματα

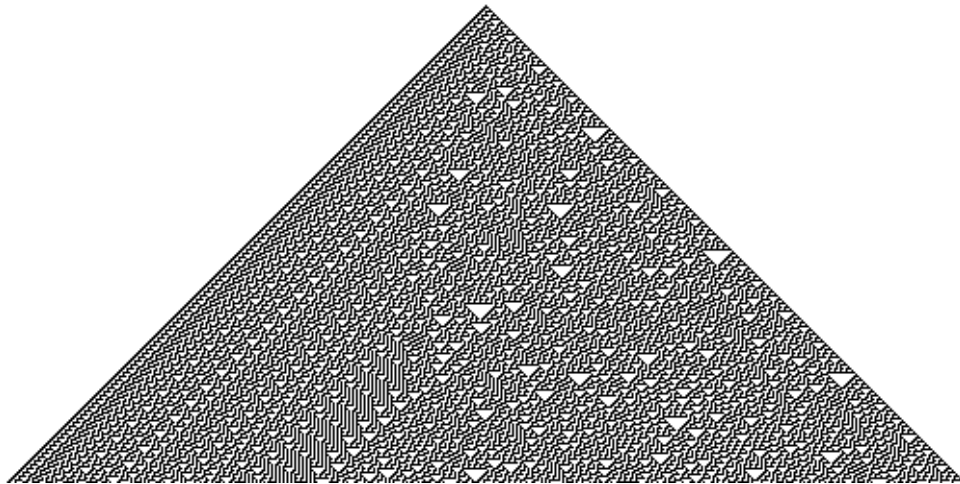
Τα κυτταρικά αυτόματα (Cellular Automata) είναι δυναμικά συστήματα που αποτελούνται από ένα πλέγμα κελιών. Κάθε κελί βρίσκεται σε μια διακριτή κατάσταση, επιλεγμένη από έναν πεπερασμένο αριθμό καταστάσεων. Η κατάσταση του κάθε κελιού αλλάζει ντετερμινιστικά με τον χρόνο (ο οποίος είναι διακριτός) σύμφωνα με ένα σύνολο κανόνων που περιλαμβάνει τις καταστάσεις των γειτονικών κελιών. Τα κυτταρικά αυτόματα μελετήθηκαν αρχικά την δεκαετία του 1940 από τον Stanislaw Ulam και τον John von Neumann ως απλά μοντέλα για την μελέτη βιολογικών λειτουργιών, όπως η αναπαραγωγή συστημάτων (την οποία ερευνούσε ο von Neumann) και η ανάπτυξη των κρυστάλλων (με την οποία είχε ασχοληθεί ο Ulam) [44].

Την δεκαετία του 1970, ο John Conway περιέγραψε ένα κυτταρικό αυτόματο που έγινε γρήγορα δημοφιλές με το όνομα «Παιχνίδι της Ζωής» («Game of Life»). Το παιχνίδι αυτό είναι ένα δισδιάστατο, δυαδικό κυτταρικό αυτόματο, το οποίο χρησιμοποιεί τους εξής κανόνες: το κάθε κελί έχει δύο δυνατές καταστάσεις (0 και 1, ή «άσπρο» και «μαύρο»). Αν ένα μαύρο κελί έχει 2 ή 3 γειτονικά μαύρα κελιά, παραμένει μαύρο. Αν έχει λιγότερα από 2 ή περισσότερα από 3 γειτονικά μαύρα κελιά, γίνεται άσπρο. Τέλος, αν ένα άσπρο κελί έχει ακριβώς 3 μαύρα γειτονικά κελιά, γίνεται μαύρο. Αυτοί οι απλοί κανόνες δημιουργούν ένα σύστημα μεγάλης πολυπλοκότητας, με συστοιχίες κελιών που αναπαράγουν τον εαυτό τους ή μετακινούνται στον διακριτό χώρο. Έχει αποδειχθεί ότι το παιχνίδι της ζωής είναι υπολογιστικά αντίστοιχο μιας οικουμενικής μηχανής Turing [44], άρα τα κυτταρικά αυτόματα μπορούν να χαρακτηριστούν παράλληλοι υπολογιστές, με την αρχική κατάσταση να αντιστοιχεί στο πρόγραμμα που τρέχουμε και τα δεδομένα εισόδου ενώ η έξοδος δίνεται από την κατάσταση του συστήματος.

Πιο πρόσφατα, ο Stephen Wolfram [45, 46] ασχολήθηκε με τα «στοιχειώδη κυτταρικά αυτόματα», μια απλή κατηγορία κυτταρικών αυτομάτων. Αυτά είναι μονοδιάστατα δυαδικά κυτταρικά αυτόματα, με τους γείτονες κάθε κελιού να είναι τα δύο κελιά αριστερά και δεξιά του.

Το κάθε κελί με τους γείτονές του αντιστοιχούν σε $2^3 = 8$ καταστάσεις, οπότε το δυνατό σύνολο κανόνων είναι $2^8 = 256$ διαφορετικοί κανόνες. Κάποιοι από αυτούς, όπως ο κανόνας 30 (αντιστοιχεί στον δυαδικό 00011110) και ο κανόνας 110 (01101110) δημιουργούν περίπλοκα κυτταρικά αυτόματα με ιδιαίτερο ενδιαφέρον (Σχ. 4.5). Ο κανόνας 110 αποδείχθηκε μάλιστα ότι αντιστοιχεί σε έναν οικουμενικό υπολογιστή, αποτέλεσμα ιδιαίτερα ενδιαφέρον για τόσο απλό σύστημα. Με βάση τον κανόνα 110 έχουν κατασκευαστεί κάποιες από τις μικρότερες οικουμενικές μηχανές Turing.

Τα κυτταρικά αυτόματα έχουν χρησιμοποιηθεί αποτελεσματικά για την μοντελοποίηση περίπλοκων φυσικών και βιολογικών διεργασιών, όπως η δυναμική υγρών και αερίων και κάποιες χημικές αντιδράσεις [46]. Έχουν επίσης εφαρμογές στην κρυπτογραφία και σε μηχανισμούς αποσφαλμάτωσης.



Σχέδιο 4.5, Στοιχειώδες κυτταρικό αυτόματο κανόνα 30

4.4. Συστήματα Lindenmayer

Τα συστήματα Lindenmayer («L-Systems») είναι ένας μαθηματικός φορμαλισμός που προτάθηκε από τον βιολόγο Aristid Lindenmayer το 1968 [47]. Πρόκειται για συστήματα επανεγγραφής, παρόμοια με τις γραμματικές του Chomsky, με την διαφορά ότι οι κανόνες παραγωγής εφαρμόζονται παράλληλα και όχι σειριακά. Ο Lindenmayer χρησιμοποίησε τα L-systems για να μελετήσει και να περιγράψει την ανάπτυξη απλών φυκιών και γενικά βιολογικών οργανισμών. Την δεκαετία του 1990 τα L-systems άρχισαν να χρησιμοποιούνται στα γραφικά υπολογιστών, κυρίως για την σχεδίαση αληθοφανών φυτών [48] και για την παραγωγή fractals [49].

Η κύρια ιδέα των L-systems είναι η παράλληλη επανεγγραφή. Τα L-systems αποτελούνται από ένα αρχικό αξίωμα και ένα σύνολο κανόνων παραγωγής, οι οποίοι εφαρμόζονται παράλληλα κατ' επανάληψιν. Θα μπορούσαμε να ορίσουμε τυπικά ένα σύστημα L ως μια τετράδα:

$$L = \{V, S, a, K\}$$

Όπου:

- V είναι η αλφάβητος, δηλαδή ένα σύνολο συμβόλων που περιέχει στοιχεία που μπορούν να αντικατασταθούν (μεταβλητές)
- S είναι ένα σύνολο συμβόλων που περιέχει στοιχεία που δεν μεταβάλλονται (σταθερές)
- a είναι μια συμβολοσειρά στοιχείων του V , η οποία ορίζει την αρχική κατάσταση του συστήματος (αξίωμα)
- K είναι ένα σύνολο κανόνων. Οι κανόνες αποτελούνται από δυο συμβολοσειρές, την αριστερή και την δεξιά, και περιγράφουν τρόπους με τους οποίους μια μεταβλητή (αριστερή συμβολοσειρά) μπορεί να αντικατασταθεί από μια συμβολοσειρά σταθερών και μεταβλητών (δεξιά συμβολοσειρά).

Για παράδειγμα, το σύστημα που μελέτησε αρχικά ο Lindenmayer για την ανάπτυξη των φυκιών χρησιμοποιούσε τις εξής τιμές. Η ανάπτυξή του φαίνεται στον Πίνακα 4.1.

Μεταβλητές: A, B

Σταθερές: κενό

Αρχικό αξίωμα: A

Κανόνες παραγωγής: (A > AB), (B > A)

Συμβολοσειρά	
A	(αξίωμα)
AB	(αντικατάσταση A με AB)
ABA	(A με AB, B με A)
ABAAB	κλπ.
ABAABABABA	
ABAABABAABAAB	
ABAABABAABAABABAABAABABA	
ABAABABAABAABABAABAABABAABAABABAABAAB	

Πίνακας 4.1, Ανάπτυξη βασικού συστήματος Lindenmayer

Τα συστήματα Lindenmayer επινοήθηκαν ως μαθηματικός φορμαλισμός για την περιγραφή της θεωρίας της ανάπτυξης, οπότε δεν δόθηκε σημασία στην γεωμετρία και την δυνατότητα σχεδίασης των συστημάτων. Το 1989 ο Przemyslaw Prusinkiewicz και ο Aristid Lindenmayer δημοσίευσαν το άρθρο «Η αλγοριθμική ομορφιά των φυτών», προκαλώντας ένα κύμα ενδιαφέροντος για τα L-systems ως συστήματα παραγωγής γραφικών υπολογιστή και fractals [48]. Για να πετύχει την γραφική απεικόνιση των L-systems, ο Prusinkiewicz χρησιμοποίησε μια αντιστοιχία συμβόλων και κανόνων σχεδίασης στο προγραμματιστικό – γραφικό περιβάλλον LOGO. Χαρακτηριστικό γνώρισμα του LOGO είναι μια χελώνα, η οποία έχει μια θέση στο χώρο, και υπακούει σε εντολές όπως «προχώρα A βήματα», «στρίψε B μοίρες», «άρχισε να σχεδιάζεις» και «σταμάτα την σχεδίαση».

Η αντιστοιχία την οποία χρησιμοποίησε ο Prusinkiewicz είναι η παρακάτω.

- F: «προχώρα A βήματα, ζωγραφίζοντας»
- f: «προχώρα A βήματα χωρίς να ζωγραφίζεις»
- +: «στρίψε δεξιά B μοίρες»
- -: «στρίψε αριστερά B μοίρες»

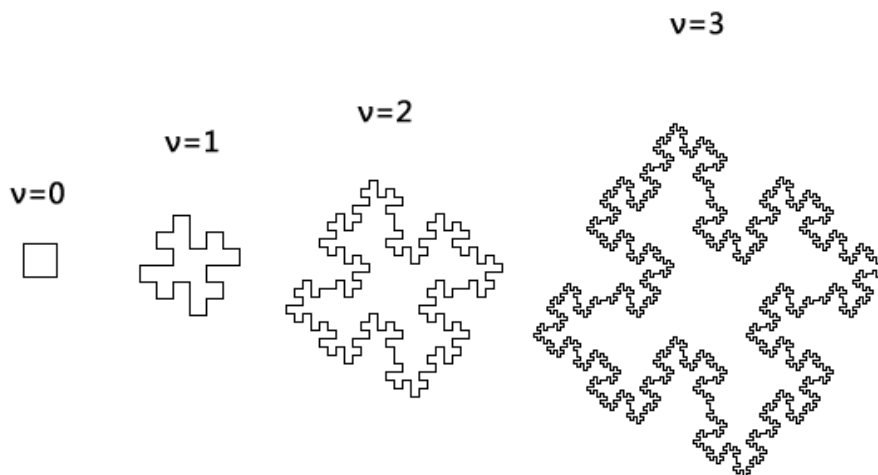
Τα A και B είναι σταθερές. Αν υποθέσουμε τώρα ότι έχουμε το L-system:

Μεταβλητές: F, +, -

Αξίωμα: F+F+F+F

Κανόνες παραγωγής: (F > F+F-F-FF+F+F-F)

Οι 3 πρώτες επαναλήψεις της γραφικής ανάπτυξης του συστήματος με $A = 5$ και $B = 90^\circ$ φαίνονται στο Σχ. 4.6.



Σχέδιο 4.6, 3 πρώτες επαναλήψεις ενός L-system (σε κλίμακα)

Οι κανόνες σχεδίασης που περιγράψαμε παραπάνω μπορούν να οδηγήσουν σε σχέδια μεγάλης πολυπλοκότητας, αλλά οδηγούν στον περιορισμό ότι το σχέδιο είναι μόνο μια γραμμή. Προσθέτοντας δυο επιπλέον κανόνες έχουμε ένα σύστημα με περισσότερες δυνατότητες, με το οποίο μπορούμε να μοντελοποιήσουμε για παράδειγμα την αρχιτεκτονική φυτών και φύλλων. Οι κανόνες που θα προσθέσουμε θα είναι οι εξής:

-]: «πρόσθεσε (push) την κατάσταση της χελώνας σε μια στοίβα»
- [: «ανάκτησε (pop) το πάνω στοιχείο της στοίβας και κάνε το την τρέχουσα κατάσταση της χελώνας»

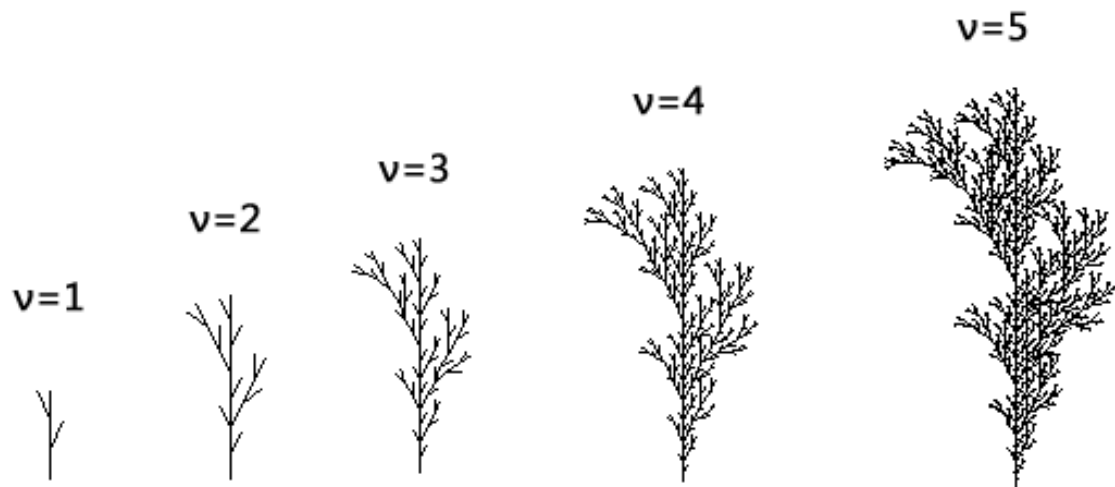
Ένα παράδειγμα συστήματος που χρησιμοποιεί και τους κανόνες αυτούς είναι το παρακάτω:

Μεταβλητές: F, +, -, [,]

Αξίωμα: F

Κανόνες παραγωγής: (F > F[-F]F[+F][F])

Στο Σχ. 4.7 φαίνονται οι 5 πρώτες επαναλήψεις του συστήματος αυτού, το οποίο αν και απλό, αρχίζει να θυμίζει την ανάπτυξη ενός φυτού.



Σχέδιο 4.7, 5 πρώτες επαναλήψεις ενός L-system με χρήση στοιβάς (σε κλίμακα)

Η ομοιότητα των L-systems με τα συστήματα ανάπτυξης των οργανισμών στη φύση, που οφείλεται στην παράλληλη επανεγγραφή των κανόνων, τα έχει κάνει τα τελευταία χρόνια δημοφιλή σε υπολογιστικά συστήματα προσομοίωσης βιολογικών διαδικασιών και στον τομέα της Τεχνητής Ζωής.

4.5. Οχήματα Braitenberg

Τα οχήματα Braitenberg («Braitenberg vehicles») είναι ένα νοητικό πείραμα του ιταλό-αυστριακού επιστήμονα Valentino Braitenberg [50]. Πρόκειται για μια σειρά δεκατεσσάρων μηχανημάτων, σταδιακά αυξανόμενης πολυπλοκότητας, τα οποία χρησιμοποιώντας αποκλειστικά συνδεδεμένους κινητήρες και αισθητήρες εμφανίζουν σύνθετες συμπεριφορές που θα παρατηρούσαμε κανονικά σε ζωντανά πλάσματα ή τουλάχιστον σε προγραμματιζόμενα ρομπότ.

Τα πρώτα οχήματα είναι ιδιαίτερα απλά. Για παράδειγμα, το πρώτο όχημα που περιγράφει ο Braitenberg αποτελείται από έναν κινητήρα και έναν αισθητήρα (για παράδειγμα αισθητήρα φωτός ή θερμότητας – θα υποθέσουμε στη συνέχεια ότι χρησιμοποιούμε σε όλες τις περιπτώσεις αισθητήρες φωτός). Ο αισθητήρας συνδέεται με τον κινητήρα με τέτοιο τρόπο ώστε όσο περισσότερο φως υπάρχει στον χώρο που βρίσκεται το όχημα, τόσο πιο έντονα θα λειτουργεί ο κινητήρας. Αυτό έχει ως αποτέλεσμα το όχημα να φαίνεται ότι «φοβάται» το φως και να προσπαθεί να απομακρυνθεί από αυτό, σταματώντας να κινείται όταν βρει κάποια εστία σκοταδιού. Το τρίτο όχημα αποτελείται από δύο κινητήρες, έναν σε κάθε μεριά του οχήματος, και δύο αντίστοιχα τοποθετημένους αισθητήρες φωτός. Οι αισθητήρες είναι συνδεδεμένοι είτε με τους κινητήρες της πλευράς τους, είτε «σταυρωτά», και όσο περισσότερο φως υπάρχει τόσο λιγότερο δουλεύουν οι κινητήρες. Το όχημα αυτό παρουσιάζει πιο σύνθετη συμπεριφορά, την οποία ο Braitenberg περιγράφει ως «αγαπητική»: το όχημα «αγαπάει» το φως και κινείται στον χώρο μέχρι να βρει μια έντονη εστία φωτός, οπότε σταματάει κοιτάζοντάς το. Αν οι αισθητήρες είναι συνδεδεμένοι σταυρωτά, μπορεί εύκολα κάποιος περισπασμός να το αποσπάσει και να απομακρυνθεί.

Τα οχήματα που περιγράφει ο Braitenberg γίνονται όλο και πιο σύνθετα [51], καθώς προστίθενται συνδέσεις που «θυμούνται» και ενισχύονται όταν τα στοιχεία που συνδέουν ενεργοποιούνται ταυτόχρονα («mnemotrix»), συνδέσεις που ενισχύονται όταν τα στοιχεία που συνδέουν ενεργοποιούνται στη σειρά («ergotrix»), πολλαπλά είδη αισθητήρων, στοιχεία κατωφλιού (παρόμοια σε λειτουργία με τεχνητά νευρόνια) και άλλα στοιχεία. Ακόμα και τα πιο απλά οχήματα όταν βρίσκονται σε ένα σύνθετο περιβάλλον με πολλαπλές πηγές φωτός και σκοταδιού παρουσιάζουν μια σύνθετη και δυναμική στόχο-κεντρική συμπεριφορά και κινούνται με απρόβλεπτες τροχιές, εξερευνώντας τον χώρο. Η κίνηση ενός προσομοιωμένου οχήματος Braitenberg φαίνεται στο Σχ. 4.8.



Σχέδιο 4.8, Περίπλοκες κινήσεις οχημάτων Braitenberg

5. Τεχνητή Ζωή

5.1. Εισαγωγή και ορισμοί

Ο πρώτος τυπικός ορισμός του τομέα της Τεχνητής Ζωής δόθηκε από τον Christopher Langton, σε ένα συνέδριο που οργάνωσε στο Los Alamos το 1987 [52]:

«Η Τεχνητή Ζωή (Artificial Life) είναι η μελέτη των τεχνητών συστημάτων που επιδεικνύουν συμπεριφορές που χαρακτηρίζουν τα έμβια φυσικά συστήματα. Είναι η προσπάθεια να εξηγήσουμε την ζωή σε κάθε πιθανή της εκδήλωση, χωρίς να περιοριζόμαστε στα συγκεκριμένα παραδείγματα που εξελίχθηκαν στον πλανήτη μας. Η προσπάθεια αυτή περιλαμβάνει βιολογικά και χημικά πειράματα, υπολογιστικές προσομοιώσεις και καθαρά θεωρητικές προσεγγίσεις. [...] Ο τελικός στόχος είναι η εξώρυξη της λογικής μορφής των ζωντανών συστημάτων».

Πριν από τον Christopher Langton, αρκετοί επιστήμονες είχαν ασχοληθεί με την προσπάθεια να μιμηθούν σε διάφορα επίπεδα έναν ζωντανό οργανισμό, από τα μηχανικά αυτόματα του 18^{ου} αιώνα μέχρι τα κυτταρικά αυτόματα του von Neumann τα οποία μελετήσαμε στο κεφάλαιο 4.3. Η ουσιαστική ενασχόληση με την Τεχνητή Ζωή ως επιστήμη όμως άρχισε με την εξάπλωση των ηλεκτρονικών υπολογιστών και τις δυνατότητες προσομοίωσης που προσέφεραν στους ερευνητές.

5.2. Στόχοι

Ο ορισμός που δώσαμε παραπάνω υπονοεί ως στόχο του πεδίου της Τεχνητής Ζωής την δημιουργία ζωντανών πλασμάτων. Ο Langton υποστηρίζει μάλιστα ότι ένας προσομοιωμένος οργανισμός που εμφανίζει τις ίδιες θεμελιώδεις συμπεριφορές που χαρακτηρίζουν τους έμβιους οργανισμούς είναι «ζωντανός» με τον ίδιο τρόπο που κάθε φυσικός οργανισμός είναι ζωντανός [53]. Παρ' όλα αυτά, ένα μεγάλο μέρος των ερευνητών του πεδίου δεν συμμερίζονται την θέση αυτή οπότε έχουν και διαφορετικούς στόχους. Αυτή η διαφορά απόψεων διαχωρίζει την Τεχνητή Ζωή σε δυο κατηγορίες:

- Η «δυνατή» Τεχνητή Ζωή (“Strong ALife”, σε αντιστοιχία με το “Strong AI”) αναφέρεται στην άποψη ότι η «ζωή» είναι μια διαδικασία η οποία δεν περιορίζεται σε κάποιο συγκεκριμένο φυσικό μέσο. Στόχος των ερευνητών της δυνατής Τεχνητής Ζωής είναι η δημιουργία ζωντανών και ευφύων οργανισμών μέσω της προσομοίωσης των διαδικασιών της εξέλιξης και της φυσικής επιλογής.

- Η «αδύναμη» Τεχνητή Ζωή (“Weak ALife”, σε αντιστοιχία με το “Weak AI”) απορρίπτει την παραπάνω άποψη και υποστηρίζει ότι δεν είναι δυνατή η σύνθεση «ζωής» χωρίς τις αντίστοιχες χημικές και βιολογικές διαδικασίες. Ως εκ τούτου, στόχος των ερευνητών είναι η χρήση προσομοιώσεων για την μελέτη και την καλύτερη κατανόηση των αντίστοιχων βιολογικών φαινομένων και εννοιών.

Γενικά, θα μπορούσε κανείς να υποστηρίξει ότι τα επιστημονικά πεδία της Τεχνητής Ζωής και της Τεχνητής Νοημοσύνης έχουν παραπλήσιο στόχο – την σύνθεση νοήμων οργανισμών. Η Τεχνητή Νοημοσύνη προσεγγίζει τον στόχο αυτό αρχίζοντας από έννοιες υψηλού επιπέδου που χαρακτηρίζουν την νοημοσύνη και υλοποιώντας τις («top-down approach») ενώ η Τεχνητή Ζωή τον προσεγγίζει συνθέτοντας τα βασικά χαρακτηριστικά που οδήγησαν στην εκδήλωση της και δουλεύοντας «προς τα πάνω» («bottom-up approach»).

5.3. Ιστορική ανασκόπηση

Στο σημείο αυτό θα κάνουμε μια αναφορά σε ορισμένες κύριες προσπάθειες και προσομοιώσεις που συνθέτουν το πεδίο της Τεχνητής Ζωής. Οι προσομοιώσεις αυτές μπορούν να ταξινομηθούν σε τρεις κατηγορίες, οι οποίες αντιπροσωπεύουν σε γενικές γραμμές την χρονολογική εξέλιξη του πεδίου από την δεκαετία του 1990 μέχρι και σήμερα.

5.3.1. Tierra και προγραμματιστική λογική

Στις αρχές της δεκαετίας του 1990 ο Thomas Ray, τροπικός βιολόγος στα δάση της Costa Rica, άρχισε να ασχολείται με την επιστήμη των υπολογιστών – και πιο συγκεκριμένα με την μελέτη της εξέλιξης σε ψηφιακά μέσα. Η μελέτη αυτή άρχισε με την δημιουργία του Tierra, το οποίο είναι μια από τις πρώτες προσομοιώσεις Τεχνητής Ζωής [54].

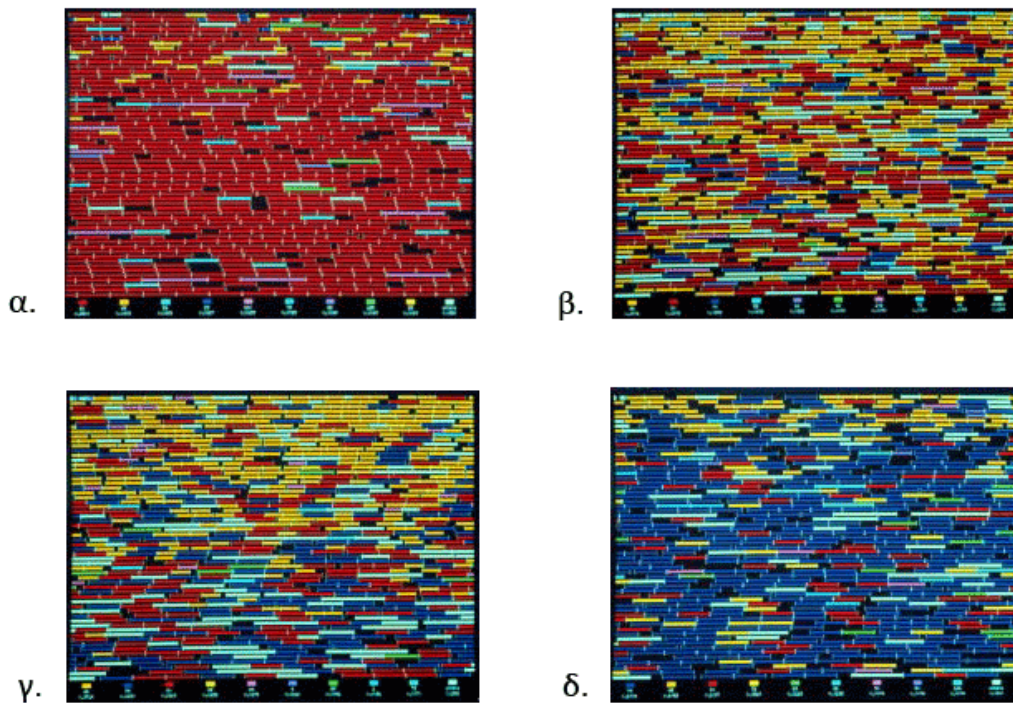
Το Tierra χρησιμοποιεί προγραμματιστική λογική: το περιβάλλον είναι ένα εικονικό υπολογιστικό σύστημα, του οποίου αγαθά είναι η μνήμη και η επεξεργαστική ισχύς. Για τα αγαθά αυτά αντιμάχονται οργανισμοί - διεργασίες των οποίων το γενετικό υλικό είναι κώδικας μηχανής. Οι διεργασίες μπορούν να αντιγράψουν τον εαυτό τους, και το σύστημα δίνει την δυνατότητα μεταλλάξεων και συνδυασμού του γενετικού υλικού ώστε να μπορέσουν να ενεργήσουν οι διεργασίες της εξέλιξης και της φυσικής επιλογής. Στον Πίνακα 5.1. φαίνονται οι αντιστοιχίες μεταξύ της προσομοίωσης του Tierra και του οικοσυστήματος της Γης:

Οικοσύστημα	Γη	Tierra
Περιβάλλον	Γήινο οικοσύστημα	Υπολογιστικό σύστημα
Αγαθά	Χώρος	Μνήμη
	Ενέργεια	Υπολογιστική ισχύς
Οργανισμοί	Μορφές ζωής βασισμένες στον άνθρακα	Διεργασίες
Γενετικό υλικό	Αλληλουχία DNA	Αλληλουχία εντολών κώδικα μηχανής

Πίνακας 5.1, Αντιστοιχίες προσομοίωσης Tierra

Η προσομοίωση αρχίζει με προκατασκευασμένες διεργασίες των οποίων ο γενετικός κώδικας ευνοεί την αναπαραγωγή. Οι τυχαίες μεταλλάξεις δημιουργούν νέες διεργασίες – παράσιτα, οι οποίες αν και δεν έχουν τον αντίστοιχο κώδικα, χρησιμοποιούν τον κώδικα αναπαραγωγής των γύρω διεργασιών ώστε να πολλαπλασιαστούν (Σχ. 5.1α, οι αρχικές διεργασίες φαίνονται με κόκκινο ενώ τα παράσιτα με κίτρινο). Οι διεργασίες αυτές είναι πιο αποδοτικές, οπότε γρήγορα καταλαμβάνουν ένα μεγάλο μέρος του υπολογιστικού χώρου (Σχ. 5.1β).

Στο Σχ. 5.1β παρατηρούμε ότι έχει εμφανισθεί κωδικοποιημένο με μπλέ χρώμα ένα νέο είδος διεργασιών. Οι διεργασίες αυτές έχουν ανοσία στα παράσιτα, τα οποία δεν μπορούν πια να τις χρησιμοποιήσουν για να αναπαράγουν τον εαυτό τους. Σύντομα τα παράσιτα περιορίζονται σε συγκεκριμένα μέρη της μνήμης (Σχ. 5.1γ) και σχεδόν εξαλείφονται, με το νέο είδος διεργασιών να καταλαμβάνουν πλέον το μεγαλύτερο μέρος της μνήμης (Σχ. 5.1δ).



Σχέδιο 5.1, Απεικόνιση χώρου μνήμης Tierra

Τα αποτελέσματα αυτά ήταν ιδιαίτερα σημαντικά για την εποχή και ενίσχυσαν τις ελπίδες των πρώτων ερευνητών της Τεχνητής Ζωής για ανάδυση ευφών συμπεριφορών. Το Tierra και οι βασικές αρχές του αποτέλεσαν έμπνευση για πολλούς άλλους ερευνητές και προγράμματα με πιο γνωστό το Avida του Michigan State University, το οποίο παραμένει μέχρι σήμερα ερευνητικά ενεργό. Παράλληλα αναπτύχθηκε μια έκδοση του Tierra με δυνατότητες δικτύωσης ώστε ξεφεύγοντας από τις δυνατότητες του κάθε υπολογιστή να δημιουργηθεί ένα διαμοιρασμένο και μεγαλύτερης ποικιλομορφίας οικοσύστημα. Παρ' όλα αυτά οι ελπίδες του Ray για αναδυόμενες συμπεριφορές υψηλότερου επιπέδου διαψεύστηκαν. Το σύστημα μετά από κάποιο σημείο σταματάει να αναπτύσσει νέα είδη διεργασιών, δηλαδή η *εξελιξιμότητα* του είναι περιορισμένη.

Η περιορισμένη εξελιξιμότητα του Tierra και των αντίστοιχων συστημάτων μπορεί να οφείλεται σε κάποιους από τους παρακάτω λόγους [55].

- Η χρήση εντολών μηχανής ως γενετικό κώδικα δεν προσφέρει αρκετή ευελιξία άρα δεν ευνοεί πολύπλοκες συμπεριφορές. Για παράδειγμα, μεταγενέστερες μελέτες έδειξαν ότι αρκούν απλές μεταλλάξεις του ενός ή δύο ψηφίων του γενετικού κώδικα για να εξηγήσουν τις περισσότερες από τις συμπεριφορές που παρατήρησε ο Ray.
- Στο σύστημα δεν υλοποιούνται κάποια σημαντικά χαρακτηριστικά των οργανισμών που εξελίχθηκαν στο γήινο οικοσύστημα όπως η δυνατότητα κίνησης και η δυνατότητα αναπαραγωγής ανάμεσα σε οργανισμούς.

- Δεν υπάρχει σαφής διαχωρισμός μεταξύ των εννοιών του *χώρου* στον οποίο υπάρχουν οι οργανισμοί, του *γενότυπου* των οργανισμών και του *φαινότυπού* τους. Ο γενότυπος και ο φαινότυπος ταυτίζονται και είναι οι εντολές του προγράμματος – οργανισμού, ενώ και ο χώρος που αντιλαμβάνεται ο κάθε οργανισμός είναι η γειτονική του μνήμη, στην οποία περιέχεται ο κώδικας άλλων οργανισμών.
- Δεν υπάρχει καμία δυνατότητα μάθησης ή προσαρμογής στην διάρκεια ζωής του κάθε οργανισμού.

5.3.2. Νευρωνικά δίκτυα & μάθηση

Όπως είδαμε στο Κεφάλαιο 4.2, ένα από τα πιο χρήσιμα εργαλεία της Τεχνητής Νοημοσύνης είναι τα νευρωνικά δίκτυα. Η δυνατότητά τους να προσομοιώσουν σε συμβολικό επίπεδο την λειτουργία του ανθρώπινου εγκεφάλου τα κάνει ιδανικό εργαλείο για την λήψη αποφάσεων σε εφαρμογές Τεχνητής Ζωής. Πράγματι, ένα μεγάλο μέρος της σημερινής ερευνητικής προσπάθειας στον τομέα βασίζεται στα νευρωνικά δίκτυα.

Μια από τις πρώτες και πιο γνωστές προσομοιώσεις που χρησιμοποιούν εκτενώς νευρωνικά δίκτυα είναι το Noble Aple του Tom Barbalet το οποίο είναι ανοικτού κώδικα και βρίσκεται σε ανάπτυξη από το 1996 [56]. Το Noble Ape εκτελεί παράλληλα δύο προσομοιώσεις (Σχ. 5.2). Η μια είναι η προσομοίωση του περιβάλλοντος το οποίο αποτελείται από θάλασσα και ξηρά και περιλαμβάνει εναλλαγές καιρικών φαινομένων, ώρας και νύχτας – μέρας. Η δεύτερη είναι η προσομοίωση του εγκεφάλου ενός αριθμού οργανισμών τα οποία μετακινούνται, τρώνε, κοιμούνται και πολλαπλασιάζονται στο περιβάλλον αυτό. Η χρήση νευρωνικών δικτύων εξαλείφει τους περισσότερους από τους λόγους που αναφέραμε παραπάνω ως αποτρεπτικούς για την εξελισσιμότητα της προσομοίωσης.

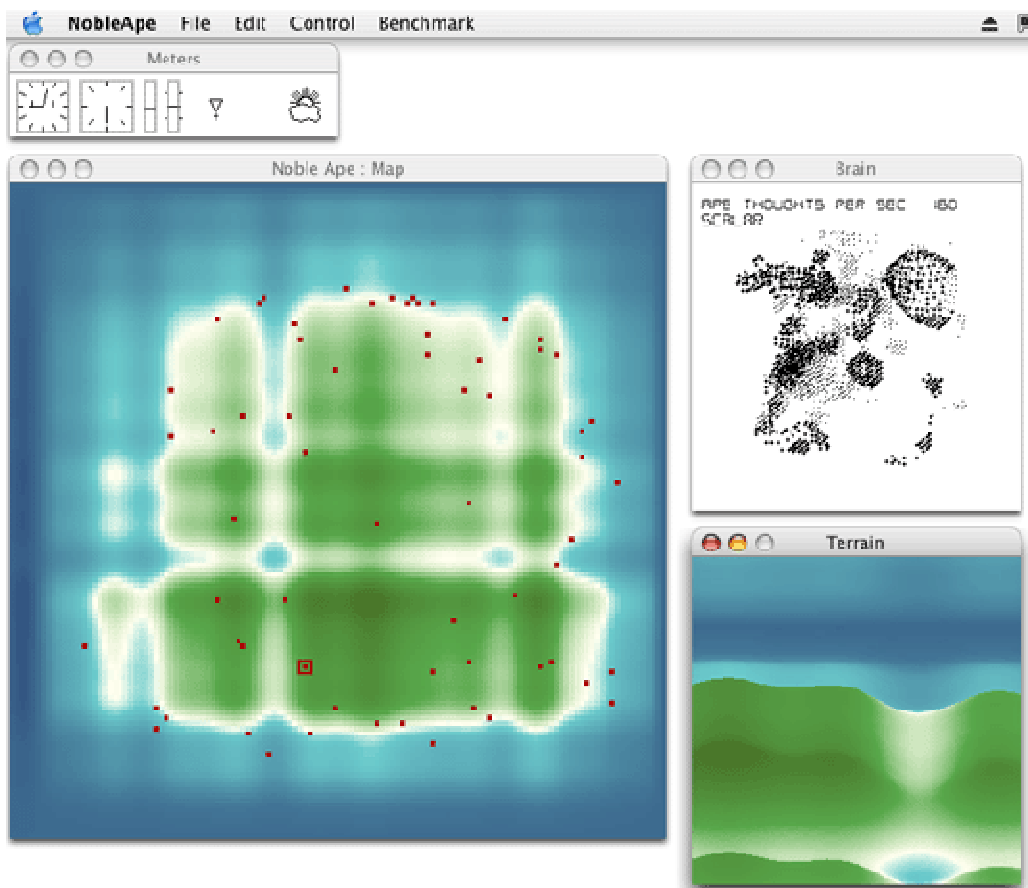
- Η προσομοίωση είναι βιολογικά πιο ακριβής. Υπάρχει μεταξύ άλλων δυνατότητα κίνησης και αναπαραγωγής ανάμεσα στους οργανισμούς.
- Η έννοια του χώρου είναι σαφώς ορισμένη.
- Η χρήση νευρωνικών δικτύων ως μηχανισμό αποφάσεων δίνει σημαντική ευελιξία και πολυπλοκότητα στο σύστημα, όπως και δυνατότητα μάθησης.

Παρ' όλα αυτά, το Noble Ape ως προσομοίωση δεν εστιάζει στην εξελισσιμότητα. Είναι περισσότερο ένα γνωσιακό πείραμα αφού δίνεται μεγαλύτερη βάση στην διαβίωση μάθηση του κάθε οργανισμού και στον προγραμματισμό αισθήσεων και αισθημάτων και όχι στην διαδικασία της φυσικής επιλογής και στην μεταφορά γονιδίων ανάμεσα σε διαδοχικές γενιές.

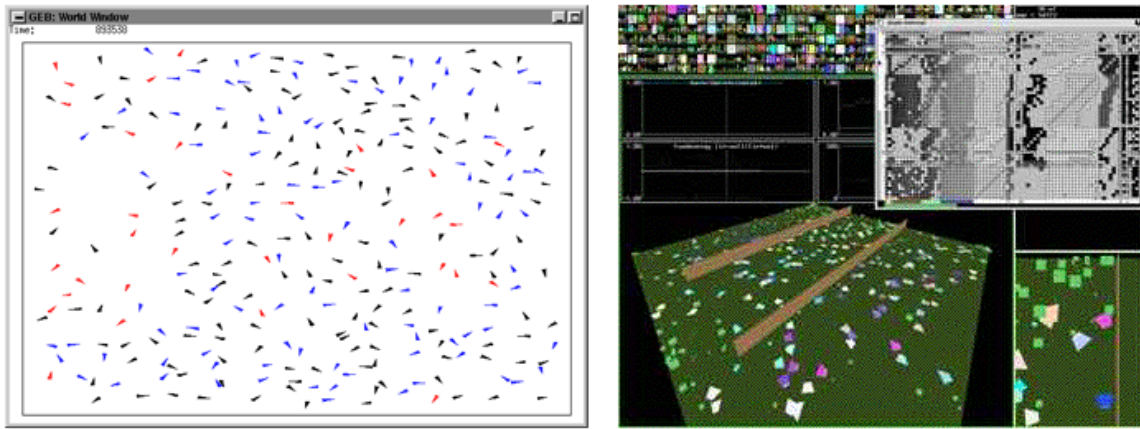
5.3.3. Εξέλιξη και φυσική επιλογή

Η τελευταία κατηγορία προσομοιώσεων που θα εξετάσουμε χρησιμοποιεί και αυτή εκτενώς νευρωνικά δίκτυα κυρίως ως μέσο λήψης αποφάσεων. Δίνεται έμφαση λιγότερο στην μάθηση και κυρίως στις βιολογικές έννοιες της εξέλιξης και της φυσικής επιλογής και στην προσπάθεια να ξεπεραστεί η πεπερασμένη εξελικτικότητα των προηγούμενων μοντέλων. Η κατηγορία αυτή θα μας απασχολήσει και στο ερευνητικό μέρος της εργασίας το οποίο θα περιγραφεί στα επόμενα κεφάλαια.

Παραδείγματα αντίστοιχων προσομοιώσεων είναι το Geb του Alastair Channon και το Polyworld του Larry Yaeger (Σχ. 5.3).



Σχέδιο 5.2, Γραφικό περιβάλλον της εφαρμογής Noble Ape



Σχέδιο 5.3, *Geb & Polyworld*

Η έμφαση των προσομοιώσεων αυτών στην διαδικασία της εξέλιξης επιτυγχάνεται με την χρήση των παρακάτω στοιχείων.

- Υπάρχει κάποιος μηχανισμός θανάτωσης, είτε περιβαλλοντικός είτε μεταξύ των οργανισμών. Ο μηχανισμός αυτός σε συνδυασμό με τις δυνατότητες αναπαραγωγής και κίνησης που έχουν οι οργανισμοί ενεργοποιεί την διαδικασία της επιλογής.
- Η έλλειψη συναρτήσεων αποτίμησης σημαίνει ότι η διαδικασία της επιλογής είναι φυσική και όχι τεχνητή όπως για παράδειγμα στους γενετικούς αλγόριθμους [57].
- Οι έννοιες του γονότυπου και του φαινότυπου είναι διαχωρισμένες. Ο γονότυπος του κάθε οργανισμού είναι ένα γονίδιο ενώ ο φαινότυπος ένα νευρωνικό δίκτυο.
- Τα γονίδια αποτελούνται από μια αλληλουχία ψηφίων ώστε να υπάρχει δυνατότητα μετάλλαξης και συνδυασμού τους από το περιβάλλον. Οι διαδικασίες αυτές αυξάνουν την ποικιλομορφία του πληθυσμού και είναι απαραίτητες για την εξέλιξη.
- Κατά την διαδικασία της δημιουργίας του κάθε οργανισμού, ο γονότυπος μετατρέπεται μέσω κάποιου μηχανισμού στον φαινότυπο, δηλαδή σε ένα νευρωνικό δίκτυο. Ο μηχανισμός αυτός συνήθως είναι ένα L-System (Κεφ. 4.4) ή ένα κυτταρικό αυτόματο (Κεφ. 4.3).
- Το νευρωνικό δίκτυο ως μηχανισμός αποφάσεων έχει το πλεονέκτημα ότι είναι εύπλαστο, ευέλικτο και δίνει την δυνατότητα υψηλής πολυπλοκότητας και μάθησης στον οργανισμό.

Πράγματι, οι προσομοιώσεις που αναφέραμε παραπάνω επέδειξαν κάποια καινούρια αποτελέσματα. Ένα από τα αποτελέσματα αυτά ήταν η ύπαρξη διαφόρων ειδών πλασμάτων και η κυκλική αυξομείωση του πληθυσμού του κάθε είδους, η οποία περιγράφεται από συγκεκριμένες εξισώσεις. Οι εξισώσεις αυτές ονομάζονται Lotka–Volterra και ισχύουν και στις κατανομές του πληθυσμού κυνηγών και θηραμάτων στα φυσικά οικοσυστήματα.

Ένα ακόμα σημαντικό συμπέρασμα που προέκυψε ήταν ότι όταν οι προσομοιώσεις ξεπεράσουν μια ορισμένη πολυπλοκότητα η αντίληψη του ανθρώπινου ερευνητή δεν είναι αρκετή για να αξιολογήσει την συμπεριφορά του συστήματος. Απο το γεγονός αυτό έγινε αισθητή η ανάγκη για στατιστική ανάλυση των προσομοιώσεων και των αποτελεσμάτων. Πράγματι, ένα μεγάλο μέρος της προσπάθειας για την ανάπτυξη του Geb έχει αφιερωθεί στην στατιστική μελέτη του μοντέλου και στην απόδειξη ότι ο μέσος όρος νέων αναδυόμενων συμπεριφορών του συστήματος είναι αυξανόμενος σε βάθος χρόνου.

5.4. Συμπεράσματα

Παρά το γεγονός ότι το πεδίο της Τεχνητής Ζωής είναι σχετικά νέο έχει αρκετές εφαρμογές, τόσο επιστημονικές όσο και εμπορικές. Έχει προσελκύσει το ενδιαφέρον των ερευνητών από διάφορες επιστήμες, δυο κύριες από τις οποίες είναι η επιστήμη των υπολογιστών και η εξελικτική βιολογία.

Απο την πλευρά της επιστήμης των υπολογιστών και της Τεχνητής Νοημοσύνης, προσφέρει έναν τρόπο επίτευξης ευφυών μηχανών χωρίς να προϋποθέτει την πλήρη κατανόηση της ανθρώπινης ευφυίας. Ο στόχος αυτός εκτός του ότι δεν είναι ευρέως αποδεκτός ενέχει τον κίνδυνο μιας απροσδόκητης κλιμάκωσης. Οποιαδήποτε τεχνολογία της οποίας τα αποτελέσματα δεν κατανοούμε πλήρως είναι ενδεχομένως επικίνδυνη.

Απο την πλευρά της εξελικτικής βιολογίας, το πεδίο της Τεχνητής Ζωής προσφέρει έναν τρόπο να μελετήσουμε την θεωρία της εξέλιξης των ειδών σε οικοσυστήματα εκτός της Γης. Η δυνατότητα αυτή είναι πρωτόγνωρη για τους ερευνητές της εξέλιξης και μπορεί να είναι μια ευκαιρία να δοθεί απάντηση σε πολλά ανοικτά ερωτήματα.

6. Σχεδιαστικές επιλογές

6.1. Εισαγωγή

Στα επόμενα κεφάλαια θα περιγράψουμε την διαδικασία σχεδίασης ενός προγράμματος το οποίο θα προσομοιώνει ένα απλό οικοσύστημα και την διαδικασία της φυσικής επιλογής σε έναν πληθυσμό οργανισμών του οικοσυστήματος. Στόχος της προσομοίωσης θα είναι η συγκριτική μελέτη της εξέλιξης των οργανισμών του οικοσυστήματος και η παρατήρηση των τυχόν αναδυόμενων συμπεριφορών που θα εκδηλωθούν στο σύστημα.

Για την σχεδίαση του προγράμματος θα βασιστούμε στην θεωρητική μελέτη που παρουσιάσαμε στο πρώτο μέρος της εργασίας και ιδιαίτερα στις βασικές αρχές που αναφέρθηκαν συνοπτικά στο Κεφ. 5.3.3.

6.2. Επιλογή προγραμματιστικού περιβάλλοντος

Στην εργασία αυτή δώσαμε κυρίως έμφαση στην δημιουργία ενός πρωτότυπου προγράμματος που να επιτρέπει την μελέτη των βασικών αρχών του αντικειμένου που εξετάζουμε και τον πειραματισμό με διάφορες ιδέες και δεδομένα. Στόχος ήταν επομένως να αποτελέσει την βάση για ένα πιο ολοκληρωμένο πρόγραμμα και δεν δόθηκε προτεραιότητα στην ταχύτητα, στην εμφάνιση και στην ευκολία εκτέλεσης του προγράμματος σε διάφορα συστήματα.

Χρειαζόμασταν λοιπόν ένα προγραμματιστικό περιβάλλον το οποίο να βοηθάει την μελέτη αυτή και να επιτρέπει εύκολα τροποποιήσεις, διαδοχικές προσθήκες στον κώδικα και γρήγορη γραφική απεικόνιση διάφορων στοιχείων και μεταβλητών. Παράλληλα ζητούμενο ήταν η γλώσσα να είναι υψηλού επιπέδου ώστε να δοθεί βάρος στην σχεδίαση του προγράμματος και όχι στις λεπτομέρειες της υλοποίησης.

Για τους παραπάνω λόγους επιλέξαμε ως προγραμματιστικό περιβάλλον την γλώσσα MATLAB η οποία προσφέρει εύκολη επεξεργασία πινάκων και γρήγορες πράξεις μεταξύ τους, έχει μια πληθώρα επιλογών για την απεικόνιση εικόνων και γραφικών παραστάσεων και έχει εγγενή υποστήριξη για νευρωνικά δίκτυα.

6.3. Βιολογικές αντιστοιχίες

Εφόσον ο στόχος μας είναι η μελέτη της εξέλιξης και της φυσικής επιλογής σε ένα τεχνητό οικοσύστημα είναι σημαντικό να υλοποιήσουμε τις αναγκαίες προϋποθέσεις για την εκδήλωση των φαινομένων αυτών. Παρακάτω θα αναλύσουμε σύντομα τις προϋποθέσεις αυτές.

6.3.1. Υλοποίηση οικοσυστήματος

Η προσομοίωση εξελίσσεται μέσα σε ένα τεχνητό οικοσύστημα. Θέλουμε οι οργανισμοί της προσομοίωσης να έχουν την δυνατότητα της κίνησης οπότε στο οικοσύστημα πρέπει να υπάρχει η έννοια του χώρου. Επιλέξαμε το οικοσύστημα να είναι οριοθετημένο χωρικά και ο χώρος και ο χρόνος να είναι διακριτά μεγέθη. Επίσης αναγκαία είναι η ύπαρξη της δυνατότητας θανάτωσης των οργανισμών και επιτυγχάνεται προσθέτοντας στο οικοσύστημά μας την έννοια της ενέργειας. Επιπλέον ιδιότητες όπως η ύπαρξη τροφής, καιρικών φαινομένων και περιβαλλοντικών κινδύνων δεν υλοποιήθηκαν για να διατηρηθεί η απλότητα του συστήματος.

6.3.2. Υλοποίηση οργανισμών

Η διαδικασία της εξέλιξης απαιτεί έναν πληθυσμό οργανισμών. Οι οργανισμοί αυτοί πρέπει να έχουν τις εξής τρεις βασικές δυνατότητες.

- Δυνατότητα κίνησης ώστε να μπορούν να αλλάξουν την θέση τους στο περιβάλλον. Δεν αποκλείεται η ανάδυση συμπεριφορών μη κινητικής φύσης αλλά οι συμπεριφορές που περιλαμβάνουν την κίνηση γίνονται πιο εύκολα αντιληπτές.
- Δυνατότητα αναπαραγωγής ώστε να μπορούν να μεταφερθούν τα χαρακτηριστικά του κάθε οργανισμού στους απογόνους του και να ενεργήσει στον πληθυσμό η εξέλιξη.
- Τέλος, δυνατότητα θανάτωσης ώστε να διαχωρίζονται τα χαρακτηριστικά που ευνοούν την επιβίωση των ειδών και να ενεργήσει στον πληθυσμό η φυσική επιλογή.

Ο κάθε οργανισμός λοιπόν θα πρέπει να μπορεί να κινηθεί στον χώρο, να αναπαραχθεί και να επιτεθεί σε κάποιον γειτονικό οργανισμό. Η απόφαση του τι θα κάνει κάθε στιγμή θα λαμβάνεται από ένα νευρωνικό δίκτυο – εγκέφαλο. Το νευρωνικό δίκτυο θα δέχεται στην είσοδό του την εσωτερική κατάσταση του οργανισμού καθώς και ερεθισμούς από το περιβάλλον και θα παράγει ως έξοδο την απόφαση για το τι θα κάνει ο οργανισμός.

Η εξελικτική αναπαραγωγή προϋποθέτει να υπάρχει ένας τρόπος κωδικοποιημένης μεταφοράς της πληροφορίας καθώς τα νευρωνικά δίκτυα δεν μπορούν εύκολα να χρησιμοποιηθούν ως γενετική πληροφορία. Τα χαρακτηριστικά ή αλλιώς ο φαινότυπος του κάθε οργανισμού θα κωδικοποιείται λοιπόν σε μια αλληλουχία δυαδικών ψηφίων, τον φαινότυπο. Κατά την αναπαραγωγή θα χρησιμοποιείται τυχαίο μέρος του γονότυπου του κάθε ενός από τους δύο γονείς (crossover). Θα υπάρχει επίσης η δυνατότητα τυχαίας μετάλλαξης κάποιου ψηφίου του γονότυπου του απογόνου (mutation).

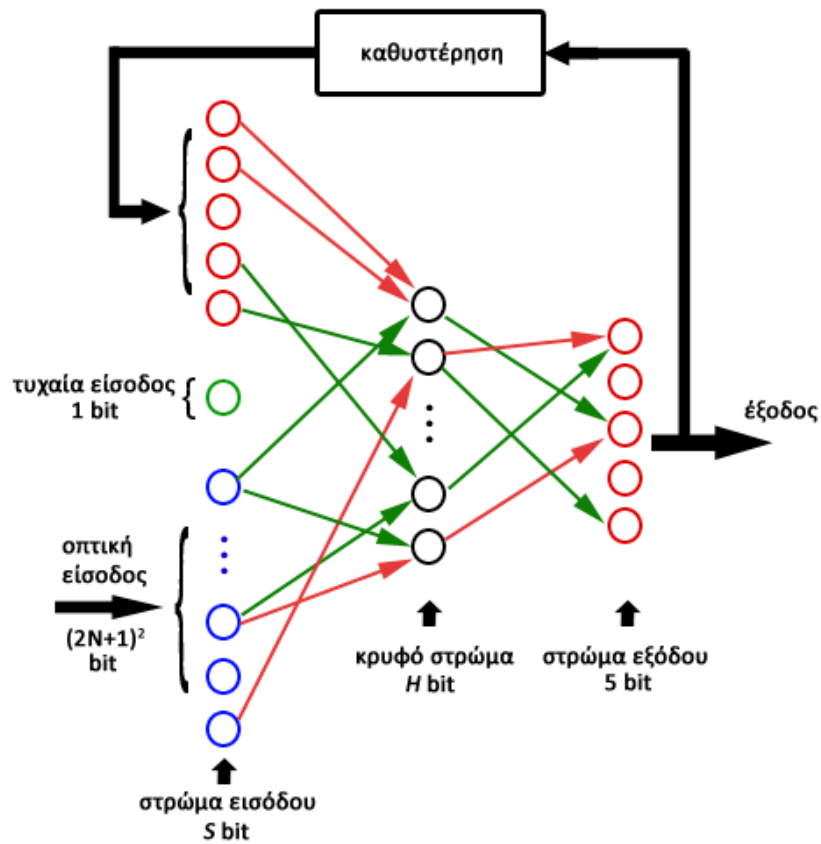
6.3.3. Υλοποίηση νευρωνικού δικτύου

Όπως αναφέραμε παραπάνω η πληροφορία για κάθε οργανισμό θα κωδικοποιείται σε μια αλληλουχία δυαδικών ψηφίων («0» και «1»). Τα ψηφία αυτά κατά την δημιουργία του κάθε οργανισμού θα καθορίζουν τα βάρη του αντίστοιχου νευρωνικού δικτύου. Το δίκτυο θα κρατηθεί όσο το δυνατόν απλούστερο, με βάρη -1 και 1 για συνάψεις εμπέδησης και διέγερσης ανάμεσα σε δύο κόμβους και βάρος 0 για απουσία συνάψεως. Το κατώφλι ενεργοποίησης του κάθε κόμβου θα είναι μηδενικό. Το νευρωνικό δίκτυο θα έχει ένα επίπεδο κόμβων εισόδου, ένα επίπεδο κρυφών κόμβων και ένα επίπεδο κόμβων εξόδου. Ο αριθμός των κόμβων των δύο πρώτων επιπέδων θα είναι παραμετροποιήσιμος.

Το επίπεδο εισόδου του νευρωνικού δικτύου θα δέχεται μια τυχαία είσοδο, μια είσοδο από τον αισθητήρα όρασης του οργανισμού και μια από την εσωτερική κατάσταση του οργανισμού. Ως εσωτερική κατάσταση του οργανισμού θεωρούμε την προηγούμενη έξοδο του δικτύου, οπότε το νευρωνικό δίκτυο θα έχει ανατροφοδότηση με καθυστέρηση μιας χρονικής μονάδας. Η ανάλυση της όρασης των οργανισμών θα είναι παραμετροποιήσιμη – για παράδειγμα αν το παράθυρο όρασης είναι N μονάδες διακριτού χώρου η είσοδος από τον αισθητήρα όρασης θα αποτελείται από $(2N + 1)^2$ ψηφία. Αν υποθέσουμε 1 τυχαίο ψηφίο εισόδου και 5 ψηφία εξόδου ο αριθμός των κόμβων εισόδου του δικτύου θα είναι:

$$S = (2N + 1)^2 + 5 + 1$$

Επιλέξαμε να χρησιμοποιήσουμε 5 κόμβους εξόδου για ευελιξία του συστήματος. Ένα σχέδιο του τελικού νευρωνικού δικτύου φαίνεται στο Σχ. 6.1, όπου H είναι ο παραμετροποιήσιμος αριθμός των κόμβων του κρυφού επιπέδου.



Σχέδιο 6.1, Πρότυπο νευρωνικό δίκτυο προσομοίωσης

Για την αντιστοίχιση των εξόδων του νευρωνικού δικτύου με τις δυνατές ενέργειες των οργανισμών δοκιμάσαμε την σειριακή και την παράλληλη μέθοδο. Στην σειριακή μέθοδο, ο κάθε οργανισμός εκτελεί μια και μόνο ενέργεια ανάλογα με την έξοδο του νευρωνικού δικτύου. Υποθέτοντας ότι έχουμε 5 ψηφία εξόδου η αντιστοίχιση θα μπορούσε να είναι η εξής (Πιν. 6.1):

Bit 1	Bit 2	Ενέργεια
0	0	Κίνηση εμπρός (Bit 3, 4 και 5 καθορίζουν την απόσταση)
0	1	Στροφή (Bit 3 καθορίζει την κατεύθυνση)
1	0	Αναπαραγωγή
1	1	Επίθεση

Πίνακας 6.1, Σειριακή αντιστοίχιση

Στην παράλληλη μέθοδο ο κάθε κόμβος εξόδου αντιστοιχεί σε μια ενέργεια και ο οργανισμός μπορεί να εκτελέσει περισσότερες απο μια ενέργειες κάθε χρονική στιγμή, ανάλογα με το αν είναι ενεργοποιημένος ο αντίστοιχος κόμβος εξόδου. Η αντιστοίχιση που χρησιμοποιήσαμε ήταν η παρακάτω (Πιν. 6.2):

Κόμβος εξόδου	Ενέργεια
1	Κίνηση εμπρός
2	Στροφή αριστερά
3	Στροφή δεξιά
4	Αναπαραγωγή
5	Επίθεση

Πίνακας 6.2, Παράλληλη αντιστοίχιση

Τελικά χρησιμοποιήσαμε την παράλληλη μέθοδο καθώς οδηγεί σε πιο ενδιαφέρουσες συμπεριφορές και βοηθάει στην επισκόπηση της σχέσης μεταξύ του νευρωνικού δικτύου και των παρατηρούμενων συμπεριφορών.

6.4. Επιλογές γραφικού περιβάλλοντος χρήστη

Τα κυρίως στοιχεία της προσομοίωσης των οποίων η εποπτεία μας ενδιαφέρει είναι το οικοσύστημα, οι οργανισμοί που ενεργούν στο οικοσύστημα αυτό και τα νευρωνικά δίκτυα – εγκέφαλοι των οργανισμών αυτών. Το γραφικό περιβάλλον της εφαρμογής μας θα πρέπει λοιπόν να εστιάζει στην απεικόνιση των παραπάνω στοιχείων και παράλληλα να προσφέρει πλήρη έλεγχο της προσομοίωσης. Είναι επίσης ζητούμενο το σύστημα να υπολογίζει και να απεικονίζει ορισμένα στατιστικά στοιχεία για την προσομοίωση.

- Το περιβάλλον – οικοσύστημα και οι οργανισμοί θα απεικονίζονται σε ένα παράθυρο το οποίο θα καταλαμβάνει μεγαλύτερο μέρος της οθόνης. Θα είναι δυνατή η απεικόνιση με χρωματικό κώδικα ανα είδος αντικειμένου και η απεικόνιση με χρωματική ομαδοποίηση ανα γενιά οργανισμών.
 - Σε ένα μικρότερο παράθυρο θα απεικονίζεται το νευρωνικό δίκτυο και κάποια βασικά στοιχεία για το τρέχοντα επιλεγμένο οργανισμό. Η επιλογή οργανισμού θα γίνεται από το παράθυρο του οικοσυστήματος. Θα είναι δυνατή η επιλογή του αν θα απεικονίζονται ή όχι οι συνάψεις εμπέδησης και ενεργοποίησης του νευρωνικού δικτύου.
 - Ένα τρίτο παράθυρο θα περιέχει ένα διάγραμμα στο οποίο θα απεικονίζονται γραφικά και αριθμητικά ο πληθυσμός των οργανισμών, οι δημιουργίες τους και οι θανατώσεις. Στο παράθυρο αυτό θα περιέχονται και πλήκτρα ελέγχου τα οποία θα δίνουν στον χρήστη την δυνατότητα να σταματάει, να ξαναξεκινάει και να αρχικοποιεί την προσομοίωση, να αποθηκεύει και να επαναφέρει μια κατάστασή της και να επηρεάζει την χρονική απεικόνιση και τις υπόλοιπες παραμέτρους της προσομοίωσης.
- Μια αναλυτική απεικόνιση του γραφικού περιβάλλοντος θα δοθεί στο Κεφ. 8.1.

7. Στοιχεία υλοποίησης

7.1. Εισαγωγή και βασική δομή

Στο κεφάλαιο αυτό θα περιγράψουμε με συγκεκριμένο τρόπο την δομή και την λειτουργία της εφαρμογής που υλοποιήσαμε. Λόγω χρήσης της προγραμματιστικής γλώσσας του MATLAB τα περισσότερα αντικείμενα της προσομοίωσης είναι κωδικοποιημένα σε μορφή πινάκων. Τα αντικείμενα και τα στοιχεία της εφαρμογής αναφέρονται αναλυτικά στο Κεφ. 7.3.

Η υλοποίηση της προσομοίωσης έγινε σύμφωνα με το πρότυπο του συναρτησιακού προγραμματισμού – η εφαρμογή αποτελείται από μια κυρίως συνάρτηση και από μια βιβλιοθήκη συναρτήσεων που χρησιμοποιούνται σε κάθε κύκλο της προσομοίωσης όπως η συνάρτηση υλοποίησης νευρωνικού δικτύου ή η συνάρτηση ζευγαρώματος. Οι συναρτήσεις αυτές περιγράφονται στο Κεφ. 7.4.

Η εφαρμογή περικλείεται σε επαναληπτικούς κύκλους («for loops») οι οποίοι αναλαμβάνουν τον έλεγχο των διάφορων μεταβλητών και εκτελούν τα αντίστοιχα κομμάτια της προσομοίωσης. Η ρουτίνα εκτέλεσης φαίνεται συνοπτικά στο Κεφ. 7.2. και αναλυτικότερα στο διάγραμμα ροής του Κεφ. 7.5.

7.2. Ρουτίνα εκτέλεσης

Η ρουτίνα εκτέλεσης του σεναρίου ομαλής λειτουργίας της εφαρμογής φαίνεται παρακάτω.

- (1) Αρχικοποίηση μεταβλητών.
- (2) Δημιουργία πίνακα οικοσυστήματος.
- (3) Δημιουργία πίνακα οργανισμών.
- (4) Σχεδίαση γραφικού περιβάλλοντος (GUI).
- (5) Υπολογισμός εισόδων νευρωνικού δικτύου για κάθε ζωντανό οργανισμό.
- (6) Υπολογισμός αντίστοιχων εξόδων νευρωνικού δικτύου.
- (7) Εκτέλεση δράσης ανάλογα με την έξοδο του νευρωνικού δικτύου:
 - Στροφή,
 - Κίνηση,
 - Αναπαραγωγή,
 - Επίθεση.
- (8) Σχεδίαση κόσμου και γραφικών παραστάσεων, ανανέωση στατιστικών δεδομένων.
- (9) Εάν δεν έχει διακόψει ο χρήστης, επανάληψη από το βήμα (5).

Μια πιο αναλυτική εικόνα της ροής η οποία δεν περιορίζεται στο σενάριο ομαλής λειτουργίας υπάρχει στο Κεφ. 7.5. όπου παραθέτουμε τα διαγράμματα ροής της εφαρμογής.

7.3. Κύρια στοιχεία

Στην ενότητα αυτή θα περιγράψουμε τα κύρια στοιχεία της προσομοίωσης.

7.3.1. Οργανισμοί

Ο κάθε οργανισμός της προσομοίωσης αναπαρίσταται μέσω μιας δομής (struct). Τα πεδία που περιέχει η δομή αυτή είναι τα παρακάτω.

- *id* – το πεδίο αυτό είναι ένας ακέραιος αριθμός που ταυτοποιεί μοναδικά τον κάθε οργανισμό. Ο πρώτος οργανισμός έχει *id* ίσο με 1, και τα *id* των επόμενων οργανισμών αυξάνονται κατά 1 διαδοχικά.
- *DNA* – το πεδίο αυτό περιέχει έναν πίνακα-διάνυσμα με το γενετικό υλικό του πλάσματος. Με βάση το υλικό αυτό δημιουργείται το νευρωνικό δίκτυο που αντιστοιχεί στον «εγκέφαλο» του κάθε πλάσματος. Τα bit του γενετικού υλικού μπορεί να είναι -1, 0 ή 1. Ο πληθυσμός που δημιουργείται κατά την αρχικοποίηση της εφαρμογής έχει τυχαίο πίνακα DNA.
- *Position* – το πεδίο αυτό είναι ένας πίνακας 1x2 που περιέχει την θέση του πλάσματος στον χώρο.
- *Direction* – το πεδίο αυτό είναι ένας πίνακας 1x2 που αντιπροσωπεύει την κατεύθυνση προς την οποία είναι στραμμένο το πλάσμα. Ο πίνακας [-1, -1] για παράδειγμα αντιστοιχεί στην κατεύθυνση κάτω-αριστερά.

7.3.2. Πίνακας οργανισμών

Ο πληθυσμός των οργανισμών αποθηκεύεται σε έναν πίνακα-διάνυσμα μεγέθους ίσο με τον αριθμό των οργανισμών. Σε κάθε χρονική επανάληψη του κυρίως επαναληπτικού κύκλου, τα στοιχεία του πίνακα αυτού προσπελαύνονται διαδοχικά και για κάθε στοιχείο υπολογίζεται η είσοδος και η απόκριση του αντίστοιχου νευρωνικού δικτύου, και η ενέργεια που προκύπτει από την απόκριση αυτή.

7.3.3. Πίνακας οικοσυστήματος

Ο πίνακας οικοσυστήματος είναι ένας δισδιάστατος τετραγωνικός πίνακας διάστασης επιλεγόμενης από το χρήστη. Κάθε κελί του πίνακα αντιστοιχεί σε έναν οριοθετημένο χώρο, και μπορεί να περιέχει είτε έναν οργανισμό, είτε ένα εμπόδιο, είτε να είναι κενό. Οι συνιστώσες i και j του κάθε κελιού ταυτίζονται με την θέση του κάθε αντικειμένου στο χώρο της προσομοίωσης. Η σχεδίαση του κόσμου γίνεται απεικονίζοντας τον πίνακα αυτό.

7.3.4. Πίνακας ταυτοποίησης

Ο πίνακας ταυτοποίησης είναι ένας πίνακας ίδιων διαστάσεων με τον πίνακα οικοσυστήματος ο οποίος περιέχει στις θέσεις όπου υπάρχουν οργανισμοί το πεδίο ταυτοποίησης (id) του αντίστοιχου οργανισμού. Χρησιμεύει στο να μπορούμε να επιλέξουμε εύκολα και με μοναδικό τρόπο έναν οργανισμό αν γνωρίζουμε μόνο την θέση του.

7.3.5. Ουρές αναπαραγωγής και θανάτωσης

Οι ουρές αυτές είναι πίνακες-διανύσματα που περιέχουν σε κάθε στιγμή τους οργανισμούς που περιμένουν είτε να αναπαραχθούν είτε να διαγραφούν. Οι ουρές γεμίζουν κατά την διάρκεια του επαναληπτικού κύκλου που υπολογίζει την απόκριση του νευρωνικού δικτύου του κάθε οργανισμού. Όταν τελειώσει ο τρέχοντας κύκλος, οι οργανισμοί που βρίσκονται στην ουρά αναπαραγωγής αναπαράγονται, οι οργανισμοί που βρίσκονται στην ουρά θανάτωσης διαγράφονται και οι ουρές αδειάζουν.

7.3.6. Νευρωνικό δίκτυο

Ο εγκέφαλος του κάθε οργανισμού αναπαρίσταται από ένα νευρωνικό δίκτυο. Το νευρωνικό δίκτυο έχει έναν αριθμό εισόδων που αντιστοιχούν στην όραση του πλάσματος και έναν αριθμό εισόδων ανατροφοδότησης. Ο αριθμός των εξόδων του καθώς και ο αριθμός του κρυμμένου στρώματος νευρονίων επιλέγονται από τον χρήστη. Τα βάρη των συνάψεων του δικτύου καθορίζονται από το γενετικό υλικό του οργανισμού και μπορούν να πάρουν τις τιμές -1, 0 και 1. Το κάθε νευρόνιο του δικτύου έχει μηδενικό κατώφλι ενεργοποίησης.

7.3.7. Γραφικό περιβάλλον

Η προσομοίωση απεικονίζεται και ελέγχεται μέσω ενός γραφικού περιβάλλοντος (GUI) σχεδιασμένου με τα βασικά στοιχεία της γλώσσας MATLAB. Το γραφικό περιβάλλον αποτελείται από ένα κυρίως παράθυρο («παράθυρο προσομοίωσης») στο οποίο απεικονίζεται γραφικά το οικοσύστημα με τους οργανισμούς του, ένα παράθυρο νευρωνικού δικτύου στο

οποίο φαίνεται η γραφική αναπαράσταση του νευρωνικού δικτύου του επιλεγμένου οργανισμού και ένα παράθυρο ελέγχου το οποίο περιέχει τα στοιχεία ελέγχου της προσομοίωσης (αναπαραγωγή, σταμάτημα, επανεκκίνηση, αποθήκευση και ανάκτηση προσομοίωσης, αλλαγή ταχύτητας προσομοίωσης και αριθμού αρχικών οργανισμών κλπ). Το γραφικό περιβάλλον και η λειτουργία του περιγράφονται πιο αναλυτικά στο Κεφ. 8.

7.4. Συναρτήσεις

7.4.1. Συνάρτηση νευρωνικού δικτύου (nnet)

Η συνάρτηση αυτή παίρνει ως είσοδο τον πίνακα DNA ενός οργανισμού, τις παραμέτρους δομής του νευρωνικού δικτύου που έχουμε επιλέξει και τις εισόδους του νευρωνικού δικτύου. Εσωτερικά υπολογίζει με βάση τα στοιχεία αυτά την αντίστοιχη απόκριση και την επιστρέφει ως έξοδο.

Όπως αναφέραμε στο Κεφ. 6.3.3. επιλέξαμε τα βάρη των συνδέσμων του νευρωνικού δικτύου να παίρνουν τις διακριτές τιμές -1, 0 και 1. Η συνάρτηση ενεργοποίησης των νευρονίων είναι η βηματική – το νευρόνιο έχει έξοδο 1 αν το συνολικό άθροισμα των εισόδων του είναι μεγαλύτερο του μηδενός, αλλιώς έχει έξοδο 0.

7.4.2. Συνάρτηση αναπαραγωγής

Η ουρά αναπαραγωγής διαβάζεται και εκτελούνται οι αντίστοιχες δημιουργίες οργανισμών και ενημερώσεις μεταβλητών μετά το τέλος του τρέχοντος επαναληπτικού κύκλου.

7.4.3. Συνάρτηση θανάτου

Η συνάρτηση θανάτου ελέγχει αν ο οργανισμός που θα διαγράψουμε υπάρχει ήδη στην ουρά θανάτωσης, και αν δεν υπάρχει τον προσθέτει. Όπως και η ουρά αναπαραγωγής, η ουρά θανάτωσης διαβάζεται μετά το τέλος του επαναληπτικού κύκλου και οι οργανισμοί που περιέχει διαγράφονται.

7.4.4. Συνάρτηση γραφικής απεικόνισης νευρωνικού δικτύου

Η συνάρτηση αυτή αναλαμβάνει την απεικόνιση του νευρωνικού δικτύου. Για την απεικόνιση, χρησιμοποιούμε την συνάρτηση `gplot` του MATLAB η οποία δέχεται ως είσοδο τον πίνακα πρόσπτωσης ενός γράφου και ένα διάνυσμα που περιγράφει τις θέσεις σχεδίασης του κάθε κόμβου του πίνακα. Οπότε η συνάρτησή μας αρχικά δημιουργεί με βάση το νευρωνικό δίκτυο τον πίνακα πρόσπτωσης και με βάση τον αριθμό και την τοπολογία των κόμβων του δικτύου το διάνυσμα περιγραφής θέσης.

Τέλος, απεικονίζει ταυτόχρονα με μπλε χρώμα τους κόμβους του νευρωνικού δικτύου, με πράσινο χρώμα τις συνάψεις διέγερσης και με κόκκινο χρώμα τις συνάψεις εμπέδησης. Ανάλογα με την επιλογή του χρήστη, μπορούν να σχεδιαστούν είτε μόνο οι συνάψεις διέγερσης είτε μόνο οι συνάψεις εμπέδησης είτε και οι δυο.

7.4.5. Βοηθητικές συναρτήσεις μετατροπών

Εκτός των βασικών συναρτήσεων χρησιμοποιούμε κάποιες βοηθητικές συναρτήσεις που υλοποιούν λειτουργίες που χρειάστηκαν αλλά δεν υπήρχαν ως έτοιμες συναρτήσεις της MATLAB.

Συνάρτηση βηματικής απόκρισης (stepf)

Η συνάρτηση αυτή επιστρέφει 1 αν η είσοδος της είναι θετική αλλιώς επιστρέφει 0. Όπως αναφέραμε παραπάνω την χρησιμοποιεί η συνάρτηση nnet.

Συνάρτηση συνδυασμού DNA (crossover)

Η συνάρτηση αυτή δέχεται ως είσοδο δυο πίνακες-διανύσματα (όχι αναγκαστικά ίδιου μήκους) και έναν αριθμό m από 0 έως 1 που αντιστοιχεί στην πιθανότητα μετάλλαξης. Επιλέγεται στην τύχη ένα σημείο συνδυασμού N και συνδυάζονται τα πρώτα N στοιχεία του πρώτου πίνακα-DNA, με τα στοιχεία από $N+1$ και μετά του δεύτερου πίνακα-DNA.

Ως αποτέλεσμα έχουμε ένα νέο DNA, ενδεχομένως διαφορετικού μεγέθους από τα αρχικά. Εάν οι αρχικοί πίνακες-DNA έχουν το ίδιο μέγεθος, αυτό διατηρείται στον πίνακα εξόδου.

Στη συνέχεια με πιθανότητα m ένα τυχαίο bit του νέου DNA μεταλλάσσεται, δηλαδή λαμβάνει μια τυχαία τιμή -1, 0 ή 1. Το τελικό DNA αυτό επιστρέφεται στην έξοδο.

Συναρτήσεις μετατροπής πίνακα σε διάνυσμα (matrix2vector, vector2matrix)

Η συνάρτηση matrix2vector μετατρέπει έναν πίνακα διαστάσεων N, M σε διάνυσμα μήκους $N*M$, παραθέτοντας σειριακά τα στοιχεία του πίνακα.

Η συνάρτηση vector2matrix αντίστροφα, δέχεται ως είσοδο ένα διάνυσμα μήκους $N*M$ και τις διαστάσεις N, M και επιστρέφει στην έξοδο έναν πίνακα με τις ζητούμενες διαστάσεις γεμισμένο με τις τιμές του αρχικού διανύσματος.

Χρησιμοποιούνται σε διάφορες περιπτώσεις σε άλλες συναρτήσεις.

7.4.6. Βοηθητικές συναρτήσεις γραφικού περιβάλλοντος (Callback)

Οι συναρτήσεις Callback είναι οι συναρτήσεις που καλούνται όταν αλλάξει κάποιο από τα στοιχεία του γραφικού περιβάλλοντος (όταν για παράδειγμα πατηθεί κάποιο πλήκτρο, ή αλλάξει η τιμή ενός στοιχείου, ή γίνει «κλικ» σε κάποιο σημείο ενός γραφήματος). Οι βασικότερες που χρησιμοποιούμε είναι οι παρακάτω.

Συναρτήσεις Callback ελέγχου ροής (pausebutton, restartbutton, quitbutton)

Οι συναρτήσεις αυτές αλλάζουν απλά την μεταβλητή state η οποία ελέγχει την ροή της εφαρμογής. Έτσι όταν ο χρήστης πατήσει για παράδειγμα το πλήκτρο εξόδου η μεταβλητή state αλλάζει την τιμή της σε “quit” και η εφαρμογή ακολουθεί την αντίστοιχη ροή τερματισμού. Οι πιθανές ροές της εφαρμογής και ο έλεγχος της ροής από την μεταβλητή state φαίνονται καθαρά στο Κεφ. 7.5. όπου παραθέτουμε τα διαγράμματα ροής.

Συναρτήσεις Callback αλλαγής παραμέτρων εμφάνισης (fsbutton, colorbutton, buttongreen, buttonred)

Οι συναρτήσεις αυτές καλούνται από τα στοιχεία του γραφικού περιβάλλοντος που ελέγχουν τις διάφορες παραμέτρους εμφάνισης της προσομοίωσης, και εκτελούν τη διαδικασία που χρειάζεται για την αλλαγή των παραμέτρων αυτών.

Οι παράμετροι εμφάνισης που μπορούν να αλλάξουν είναι η ταχύτητα της γραφικής απεικόνισης της προσομοίωσης (fsbutton), ο χρωματικός πίνακας εμφάνισης των πλασμάτων στο κυρίως παράθυρο (colorbutton) και οι επιλογές εμφάνισης του νευρωνικού δικτύου στο αντίστοιχο παράθυρο (buttongreen, buttonred). Οι αρχικές παράμετροι της προσομοίωσης αλλάζουν είτε από το προγραμματιστικό περιβάλλον είτε από το παράθυρο ρυθμίσεων που περιγράφουμε παρακάτω.

Συναρτήσεις Callback αποθήκευσης και ανάκτησης κατάστασης (savebutton, loadbutton)

Οι συναρτήσεις αυτές εμφανίζουν ένα πλαίσιο αποθήκευσης ή φόρτωσης αρχείου. Ο χρήστης καλείται να διαλέξει το αντίστοιχο αρχείο και η κατάσταση της προσομοίωσης αποθηκεύεται ή ανακτάται από το επιλεγμένο αρχείο. Η αποθήκευση πραγματοποιείται μεταφέροντας στο αρχείο αυτό κάποιες βασικές μεταβλητές που περιγράφουν την κατάσταση της προσομοίωσης. Οι μεταβλητές που δεν είναι χαρακτηριστικές της προσομοίωσης (όπως τα handles του γραφικού περιβάλλοντος) δεν αποθηκεύονται και αρχικοποιούνται εκ νέου μετά την ανάκτηση της κατάστασης.

Συνάρτηση Callback αποθήκευσης εικόνας κυρίως κόσμου (screenshotbutton)

Η συνάρτηση αυτή αποθηκεύει ως εικόνα png την τρέχουσα κατάσταση της προσομοίωσης. Η επιλογή του αρχείου αποθήκευσης γίνεται από το χρήστη μέσω ενός πλαισίου αποθήκευσης που εμφανίζεται.

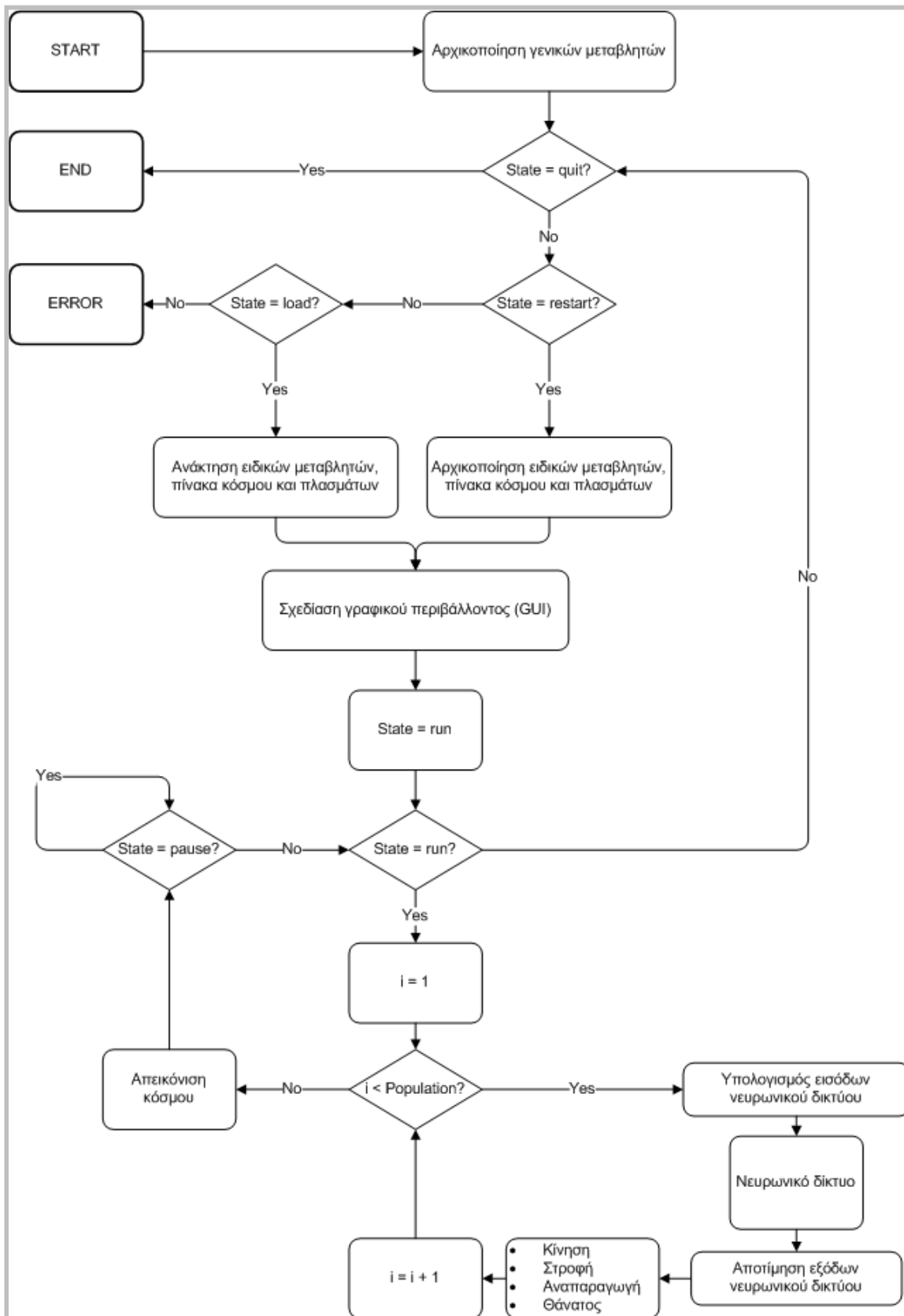
Συνάρτηση Callback επιλογής πλάσματος (worldclick)

Όπως αναφέραμε παραπάνω μια λειτουργία της εφαρμογής είναι η επιλογή ενός από τους οργανισμούς της προσομοίωσης και η γραφική απεικόνιση του νευρωνικού του δικτύου ώστε να μπορεί κανείς ενδεχομένως να μελετήσει τη σχέση της συμπεριφοράς ενός οργανισμού με την τοπολογία του νευρωνικού του δικτύου.

Η συνάρτηση αυτή αναλαμβάνει να ανιχνεύσει το πάτημα του ποντικιού στο κυρίως παράθυρο και την θέση του πατήματος αυτού. Στη συνέχεια κάνει την αντιστοιχία θέσης – οργανισμού (αν υπάρχει βέβαια οργανισμός στην θέση που πατήθηκε) και θέτει τον οργανισμό που επέλεξε ο χρήστης ως ενεργό για την απεικόνιση του νευρωνικού δικτύου.

7.5. Διάγραμμα ροής

Στο Σχ. 7.1 φαίνεται το διάγραμμα ροής της εφαρμογής.



Σχέδιο 7.1, Διάγραμμα ροής κυρίως προγράμματος

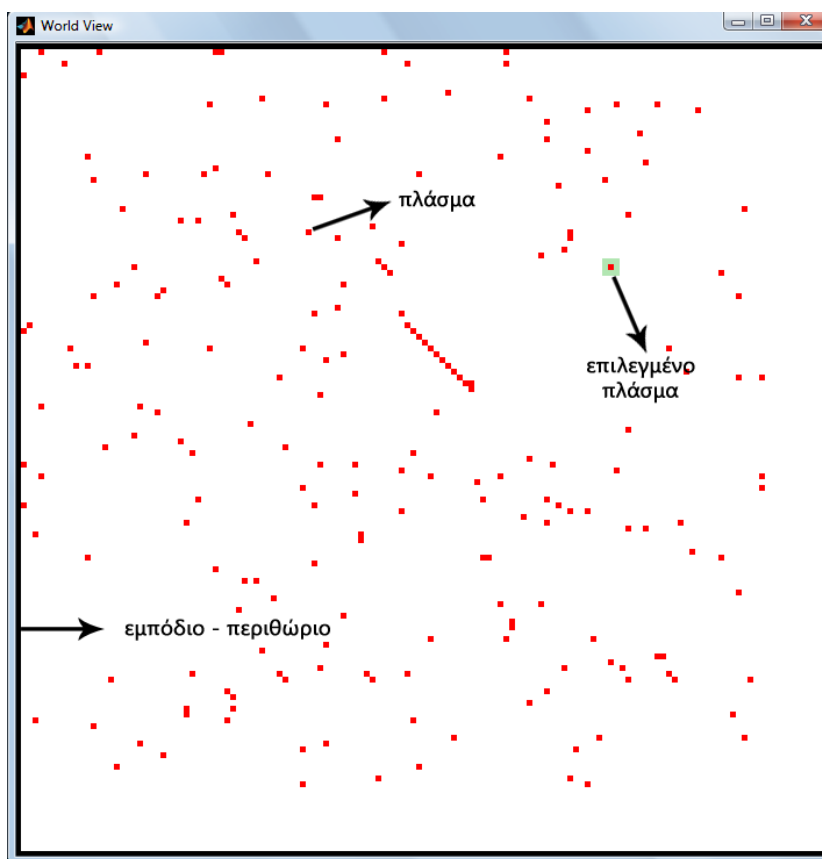
8. Λειτουργία εφαρμογής

8.1. Γραφικό περιβάλλον

Το γραφικό περιβάλλον της εφαρμογής αποτελείται από 3 κυρίως παράθυρα στα οποία είναι καταναμημένη η πληροφορία. Το πρώτο παράθυρο είναι αυτό στο οποίο απεικονίζεται το οικοσύστημα και οι οργανισμοί του, δηλαδή ο «κόσμος» της προσομοίωσης («World View»). Στο δεύτερο παράθυρο βρίσκονται τα πλήκτρα ελέγχου της προσομοίωσης και απεικονίζονται γραφικά κάποια στατιστικά στοιχεία («Controls»). Το τρίτο είναι το παράθυρο στο οποίο σχεδιάζεται το νευρωνικό δίκτυο ενός επιλεγμένου οργανισμού («Neural Network View»). Πατώντας το αντίστοιχο πλήκτρο στο παράθυρο ελέγχου εμφανίζεται και ένα τέταρτο παράθυρο που μας επιτρέπει να αλλάξουμε με γραφικό τρόπο κάποιες βασικές παραμέτρους της προσομοίωσης («Edit starting parameters»).

8.1.1. Παράθυρο προσομοίωσης

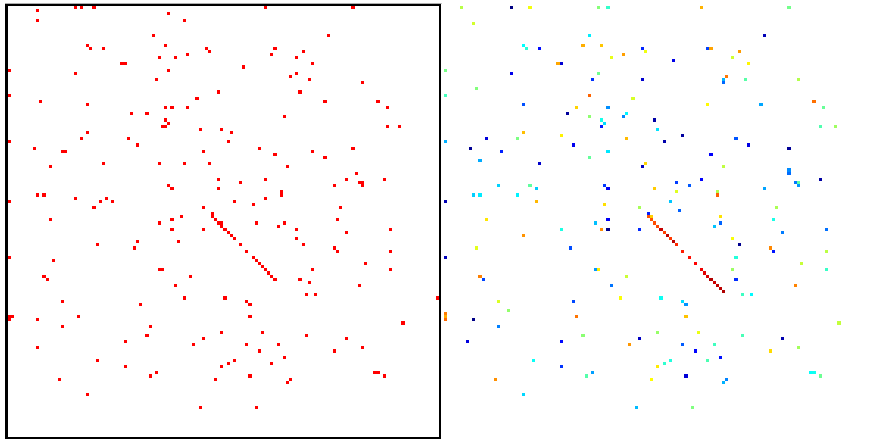
Το παράθυρο προσομοίωσης φαίνεται στο Σχ. 8.1.



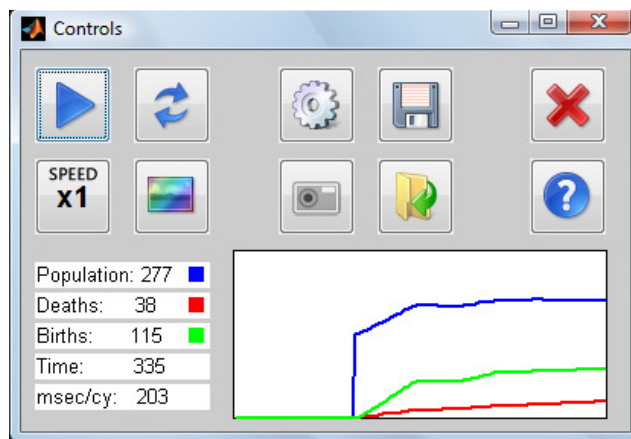
Σχέδιο 8.1, Παράθυρο προσομοίωσης

Στο παράθυρο αυτό απλά απεικονίζεται ο κόσμος (με τα περιθώριά του ή τα εμπόδια) και οι οργανισμοί ενώ κινούνται. Όταν η προσομοίωση είναι σταματημένη, ο χρήστης μπορεί να κάνει «κλικ» σε οποιονδήποτε από τους οργανισμούς ώστε να τον επιλέξει. Ο επιλεγμένος οργανισμός διακρίνεται από ένα πράσινο περιθώριο που σχεδιάζεται γύρω του (Σχ. 8.1). Μια γραφική απεικόνιση του νευρωνικού δικτύου του επιλεγμένου οργανισμού καθώς και ο αριθμός ταυτοποίησής του φαίνονται στο αντίστοιχο παράθυρο.

Ο χρήστης έχει την δυνατότητα να επιλέξει ανάμεσα σε δύο χρωματικές απεικονίσεις. Η πρώτη χρωματίζει τα αντικείμενα ανάλογα με το είδος τους – τα εμπόδια φαίνονται με μαύρο χρώμα και οι οργανισμοί με κόκκινο. Η δεύτερη απεικόνιση εμφανίζει μόνο τους οργανισμούς χρωματίζοντάς τους ανάλογα με τον αριθμό ταυτοποίησής τους. Οργανισμοί με παραπλήσιο αριθμό ταυτοποίησης άρα γεννημένοι σε κοντινές χρονικές στιγμές έχουν παρόμοιο χρώμα. Οι δυο απεικονίσεις φαίνονται στο Σχ. 8.2.



Σχέδιο 8.2, Δυο απεικονίσεις της ίδιας κατάστασης













Σχέδιο 8.3, Παράθυρο ελέγχου

8.1.2. Παράθυρο ελέγχου

Το παράθυρο ελέγχου (Σχ. 8.3) αποτελείται από δυο μέρη. Στο ένα μέρος βρίσκονται τα πλήκτρα ελέγχου της προσομοίωσης ενώ στο άλλο παρουσιάζονται κάποια στατιστικά στοιχεία ως αριθμοί και ως διάγραμμα (Σχ. 8.4).

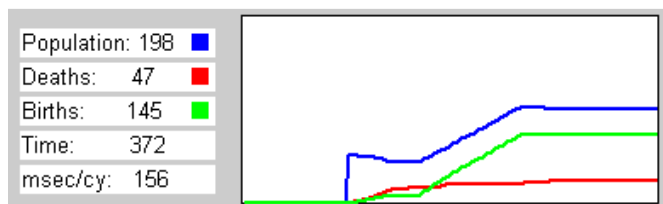
Τα πλήκτρα ελέγχου είναι τα ακόλουθα (Πιν. 8.1.):

	Το πλήκτρο αυτό σταματάει προσωρινά την προσομοίωση. Αν ξαναπατηθεί η προσομοίωση συνεχίζει από το σημείο που σταμάτησε.
	Το πλήκτρο αυτό ξαναρχίζει την προσομοίωση από την αρχή, με τις αρχικοποιήσεις που έχει επιλέξει ο χρήστης και τυχαίο αρχικό πληθυσμό.
	Το πλήκτρο αυτό αποθηκεύει την κατάσταση της προσομοίωσης σε ένα αρχείο .mat που επιλέγεται από τον χρήστη.
	Το πλήκτρο αυτό ανακτά μια αποθηκευμένη προσομοίωση από το αντίστοιχο αρχείο .mat και συνεχίζει την εκτέλεσή της.
	Το πλήκτρο αυτό ελέγχει την ταχύτητα εμφάνισης της προσομοίωσης. Πατώντας το αυξάνει διαδοχικά την ταχύτητα από x1 μέχρι και x5.
	Το πλήκτρο αυτό ελέγχει την χρωματική απεικόνιση της προσομοίωσης. Πατώντας το, εναλλάσσονται οι δυο διαθέσιμες απεικονίσεις.
	Το πλήκτρο αυτό εμφανίζει το παράθυρο ρυθμίσεων (Κεφ. 8.2.3).
	Το πλήκτρο αυτό αποθηκεύει σε μορφή εικόνας .png την εικόνα της προσομοίωσης.
	Το πλήκτρο αυτό εμφανίζει ένα παράθυρο με σύντομες πληροφορίες για την εφαρμογή.
	Το πλήκτρο αυτό τερματίζει την εφαρμογή.

Πίνακας 8.1, Πίνακας πλήκτρων ελέγχου

Τα στατιστικά στοιχεία που εμφανίζονται είναι ο συνολικός αριθμός οργανισμών που περιέχεται στον πληθυσμό («Population»), ο αριθμός θανάτων («Deaths») και ο αριθμός γεννήσεων («Births»). Τα μεγέθη αυτά απεικονίζονται χρονικά και σε μια γραφική παράσταση η οποία προσφέρει πιο εύκολη εποπτεία της σχέσης των μεγεθών και επιτρέπει να αναγνωριστούν σημαντικές στιγμές στο χρόνο όπως για παράδειγμα η αρχή της προσομοίωσης (Σχ. 8.4).

Εμφανίζεται επίσης ο χρόνος της προσομοίωσης μετρούμενος σε αριθμό κύκλων εκτέλεσης. Ως ένδειξη της απόδοσης του συστήματος αναγράφονται τα χιλιοστά του δευτερολέπτου που διήρκεσε η εκτέλεση του τελευταίου κύκλου. Όσο μεγαλύτερος είναι ο χρόνος αυτός τόσο λιγότερο αποκρίσιμο γίνεται το σύστημα.

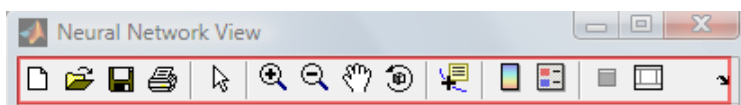


Σχέδιο 8.4, Εμφάνιση στατιστικών στοιχείων

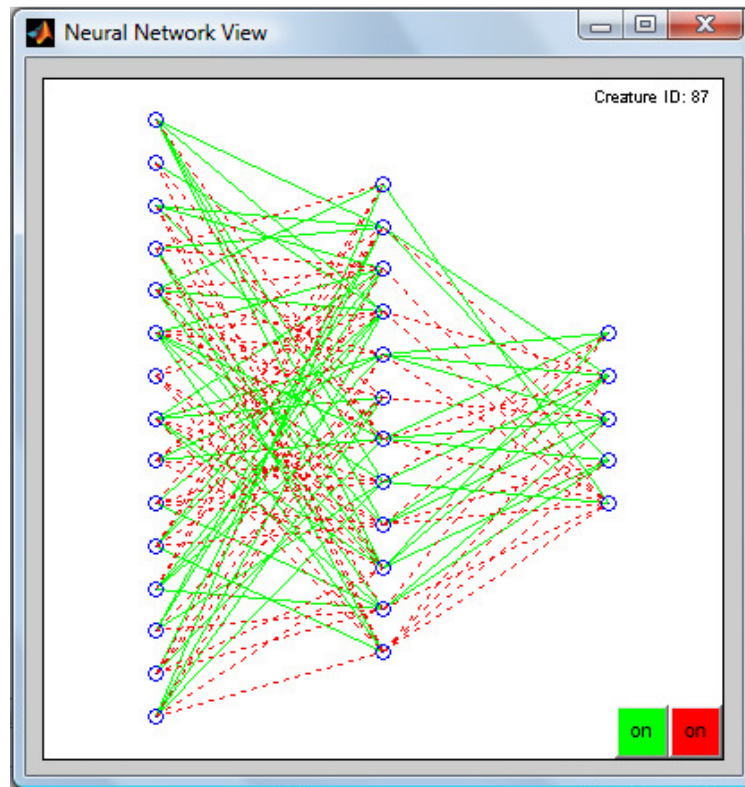
8.1.3. Παράθυρο εμφάνισης νευρωνικού δικτύου

Στην πάνω δεξιά γωνία του παραθύρου (Σχ. 8.6) αναγράφεται ο αριθμός ταυτοποίησης ενός οργανισμού που έχουμε επιλέξει («Creature id») και στο υπόλοιπο παράθυρο σχεδιάζεται το νευρωνικό δίκτυο του οργανισμού αυτού. Οι μπλε κύκλοι αντιστοιχούν στα νευρόνια του δικτύου τα οποία εμφανίζονται ομαδοποιημένα ανά επίπεδο (στρώμα εισόδου, κρυμμένο στρώμα και στρώμα εξόδου). Οι πράσινες γραμμές που συνδέουν δυο νευρόνια είναι οι συνάψεις διέγερσης ενώ οι κόκκινες γραμμές αντιστοιχούν στις συνάψεις εμπέδησης.

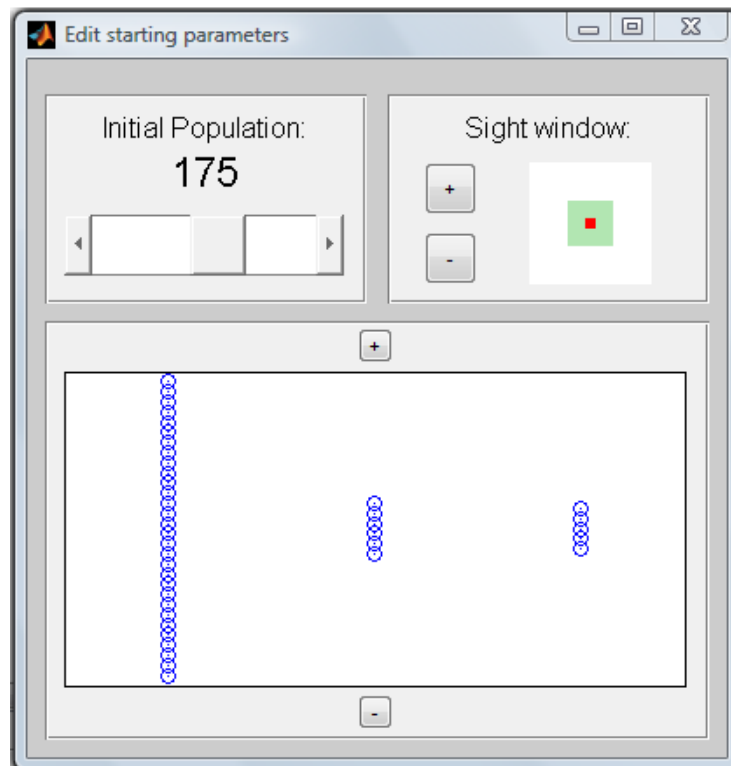
Στο κάτω μέρος του παραθύρου βρίσκονται δυο πλήκτρα κόκκινου και πράσινου χρώματος τα οποία ενεργοποιούν και απενεργοποιούν την εμφάνιση των συνάψεων εμπέδησης και διέγερσης αντίστοιχα. Κάνοντας «κλικ» σε οποιοδήποτε άλλο σημείο του παραθύρου εμφανίζεται μια γραμμή εργαλείων του Matlab (Σχ. 8.5) που επιτρέπει στον χρήστη να επεξεργαστεί την απεικόνιση του νευρωνικού δικτύου. Δίνεται για παράδειγμα η δυνατότητα μεγέθυνσης της εικόνας και εστίασης σε κάποιο συγκεκριμένο σημείο της.



Σχέδιο 8.5, Γραμμή εργαλείων του Matlab



Σχέδιο 8.6, Παράθυρο εμφάνισης νευρωνικού δικτύου



Σχέδιο 8.7, Παράθυρο ρυθμίσεων

8.1.4. Παράθυρο ρυθμίσεων

Όταν ο χρήστης πατήσει το αντίστοιχο πλήκτρο εμφανίζεται το παράθυρο ρυθμίσεων (Σχ. 8.7). Το παράθυρο αυτό επιτρέπει να αλλαχθούν με γραφικό τρόπο οι βασικές παράμετροι της προσομοίωσης οι οποίες είναι: το παράθυρο όρασης («Sight window»), ο αριθμός των νευρώνων στο κρυμμένο στρώμα και ο αρχικός πληθυσμός της προσομοίωσης («Initial population»). Δίνεται επίσης μια γραφική προεπισκόπηση του παραθύρου όρασης και της μορφής του νευρωνικού δικτύου. Οι αλλαγές που κάνει ο χρήστης εφαρμόζονται όταν αρχίσει καινούρια προσομοίωση.

8.2. Πειραματική διαδικασία

Η διαδικασία που ακολουθήσαμε για την εξαγωγή συμπερασμάτων ήταν απλή: αρχικοποιήσαμε τις παραμέτρους της προσομοίωσης με διάφορες τιμές και αφήσαμε το σύστημα να τρέξει παρακολουθώντας την εξέλιξη των κινήσεων των οργανισμών σε βάθος χρόνου. Οι τιμές των αρχικών παραμέτρων που χρησιμοποιήσαμε φαίνονται στον Πίνακα 8.2.

Παράμετρος	Τιμή	Σχόλια
WorldSize	140	Το μέγεθος του πίνακα – κόσμου.
MutationRate	0.05	Η εκατοστιαία πιθανότητα μετάλλαξης σε κάθε αναπαραγωγή.
SWindow	1-3	Το μέγεθος του οπτικού πεδίου του κάθε πλάσματος. Δοκιμάστηκαν οι τιμές 1,2 και 3.
Inter_size	2-10	Το μέγεθος του κρυφού στρώματος νευρώνων. Δοκιμάστηκαν διάφορες τιμές απο 2 έως και 10.
InitialPopulation	100-300	Το αρχικό μέγεθος του πληθυσμού. Δοκιμάστηκαν οι τιμές 100, 200 και 300.

Πίνακας 8.2, Παράμετροι πειράματος

Τα αποτελέσματα της πειραματικής διαδικασίας, οι παρατηρήσεις και τα συμπεράσματα που εξήχθησαν αναλύονται στο Κεφ. 9.

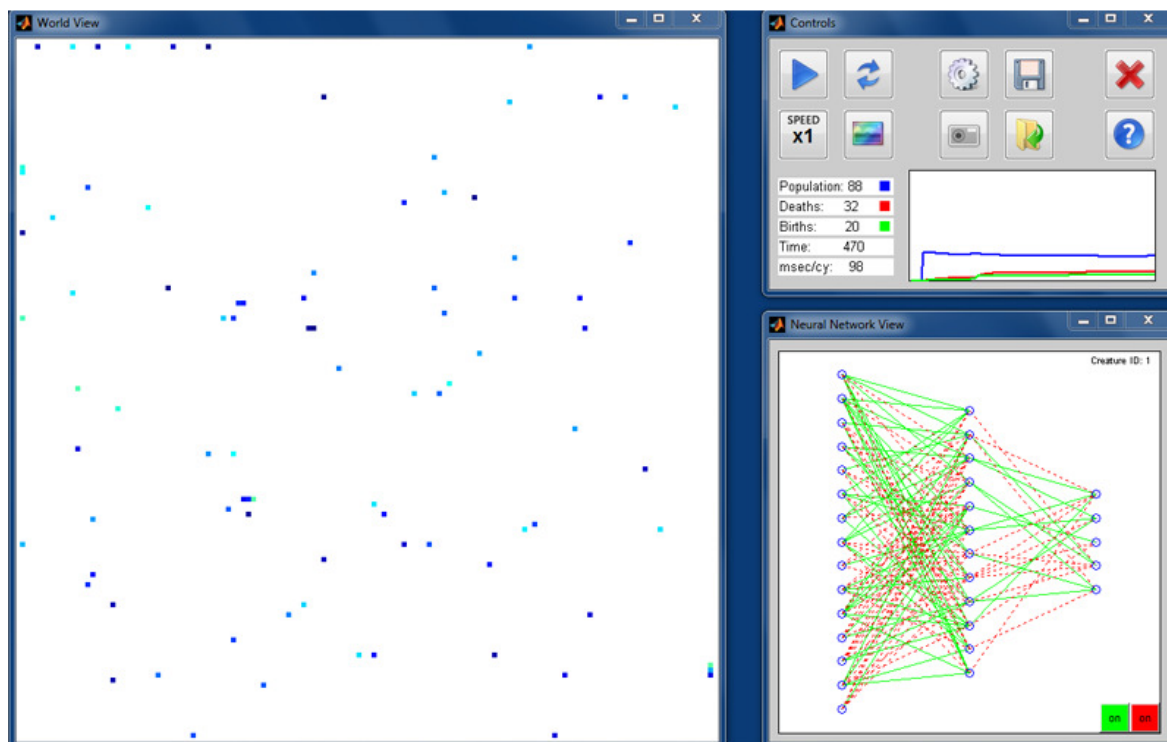
9. Παρατηρήσεις και συμπεράσματα

Σε αυτό το κεφάλαιο θα περιγράψουμε τις παρατηρήσεις που προέκυψαν από την πειραματική διαδικασία καθώς και τα συμπεράσματα που εξάγονται για την λειτουργία και την επιτυχία της προσομοίωσης.

9.1. Παρατηρήσεις

9.1.1. Αλληλεπίδραση οργανισμών

Το πρώτο γεγονός που παρατηρήσαμε ήταν ότι για αριθμό πλασμάτων μικρότερο των 200 η αλληλεπίδραση ανάμεσα στους οργανισμούς ήταν σποραδική αφού οι οργανισμοί ήταν διάσπαρτοι στον χώρο προσομοίωσης και απομακρυσμένοι μεταξύ τους. Τις περισσότερες φορές η προσομοίωση σταματούσε να εξελίσσεται χωρίς αποτελέσματα. Μια τέτοια προσομοίωση φαίνεται στο Σχ. 9.1. Οι στατικές γραφικές παραστάσεις δείχνουν ότι το σύστημα δεν αλλάζει.

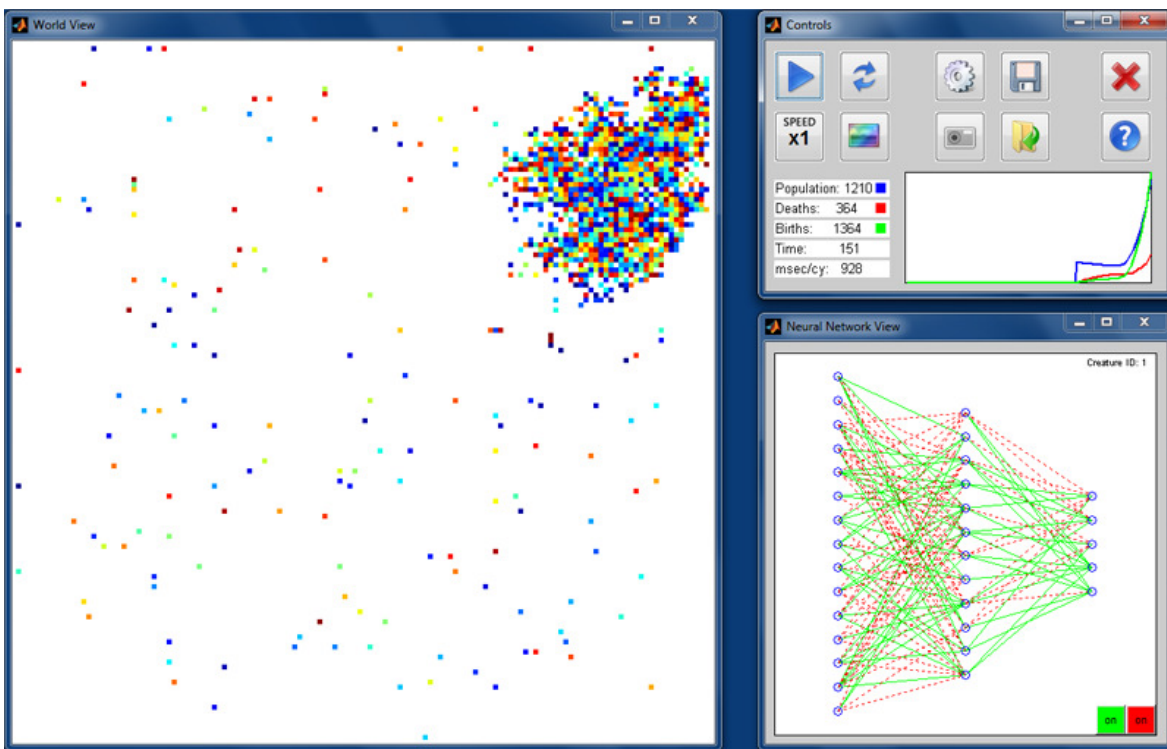


Σχέδιο 9.1, Μια προσομοίωση που δεν εξελίσσεται

Με αυξημένο αριθμό πλασμάτων υπήρχε πλούσια και ουσιώδης αλληλεπίδραση ανάμεσα στους οργανισμούς. Μια πλειοψηφία των οργανισμών μένουν ακίνητοι ή «κολλάνε» στα άκρα του κόσμου αλλά οι υπόλοιποι είναι αρκετοί ώστε να υπάρχει αλληλεπίδραση.

9.1.2. Αναδυόμενες συμπεριφορές

Οι αναδυόμενες συμπεριφορές που παρατηρήσαμε ήταν κυρίως κινητικές και σχετικά χαμηλού επιπέδου. Το σύστημα ευνοεί την συνεχόμενη κυκλική κίνηση και τις συχνές προσπάθειες αναπαραγωγής. Αυτό οδηγεί πολλές φορές σε οργανισμούς οι οποίοι αναπαράγονται ανεξέλεγκτα με αποτέλεσμα να δημιουργούνται «καρκινώματα» και η προσομοίωση να σταματάει να είναι αποκρίσιμη. Ένα τέτοιο παράδειγμα φαίνεται στο Σχ. 9.2. Στην αντίστοιχη γραφική παράσταση παρατηρούμε ότι ο αριθμός του πληθυσμού των οργανισμών ανεβαίνει εκθετικά.



Σχέδιο 9.2, Ανεξέλεγκτος πολλαπλασιασμός οργανισμών

9.1.3. Πολυκυτταρικοί οργανισμοί

Μια ενδιαφέρουσα συμπεριφορά που παρατηρήσαμε ήταν η ανάδυση «πολυκυτταρικών» οργανισμών. Οι οργανισμοί αυτοί αποτελούνταν από ένα σύνολο επιμέρους οργανισμών που χρησιμοποιούσαν ο ένας τον άλλο είτε για κίνηση είτε για πολλαπλασιασμό. Σε πολλές περιπτώσεις οι πολυκυτταρικοί οργανισμοί γεννούσαν και σκότωναν τους οργανισμούς – κύτταρά τους για να μεταφέρονται στον χώρο ή να αυξάνουν το μέγεθός τους. Πολλοί – αλλά όχι όλοι – από τους πολυκυτταρικούς οργανισμούς αυξάνονται ανεξέλεγκτα και όπως είδαμε παραπάνω οδηγούν στο σταμάτημα της προσομοίωσης.

9.2. Συμπεράσματα και προβλήματα

Τα συμπεράσματα που βγήκαν από την προσομοίωση δεν ήταν ενθαρρυντικά. Οι αναδυόμενες συμπεριφορές ήταν χαμηλού επιπέδου και στις περισσότερες περιπτώσεις το σύστημα είτε σταματούσε να εξελίσσεται είτε γινόταν μη αποκρίσιμο λόγω του ανεξέλεγκτου πολλαπλασιασμού κάποιας ομάδας οργανισμών. Παρ' όλα αυτά, βγάλαμε κάποια συμπεράσματα για το είδος των αλλαγών που πρέπει να γίνουν για να έχει καλύτερα αποτελέσματα κάποια μελλοντική έκδοση του προγράμματος. Στη συνέχεια εξηγούνται κάποια από τα προβλήματα και στο Κεφ. 10 προτείνονται αντίστοιχα κάποιες μελλοντικές επεκτάσεις.

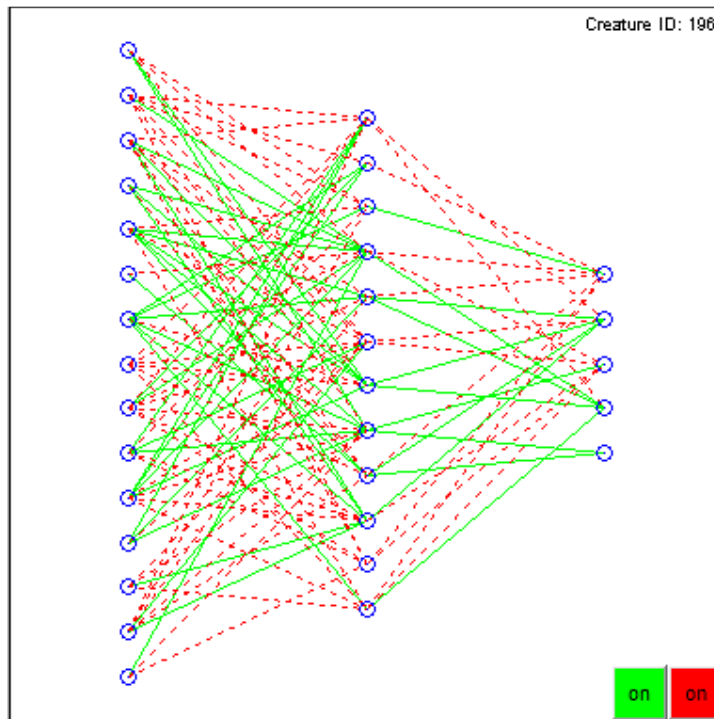
9.2.1. Ανεξέλεγκτος πολλαπλασιασμός

Όπως είδαμε παραπάνω, ένα από τα σημαντικά προβλήματα της προσομοίωσης ήταν ο ανεξέλεγκτος πολλαπλασιασμός κάποιων οργανισμών. Για να λύσουμε το πρόβλημα αυτό δοκιμάσαμε να προσθέσουμε στο σύστημα μια υποτυπώδη έννοια της ενέργειας: για να αναπαραχθούν δυο οργανισμοί χρειάζεται να σπαταλήσουν ένα μέρος της ενέργειάς τους και όταν μάχονται δυο οργανισμοί «καταναλώνει» ο ένας την ενέργεια του άλλου.

Τα αποτελέσματα ήταν ενθαρρυντικά καθώς λύθηκε το πρόβλημα του ανεξέλεγκτου πολλαπλασιασμού. Υλοποιώντας ταυτόχρονα και μια παράλληλη μέθοδο λήψης αποφάσεων (Κεφ. 6.3.3) η απόκριση της προσομοίωσης ήταν αρκετά καλύτερη. Και πάλι όμως οι συμπεριφορές που προέκυψαν ήταν αποκλειστικά κινητικού χαρακτήρα.

9.2.2. Πολύπλοκα νευρωνικά δίκτυα

Ένα σημαντικό πρόβλημα ήταν ότι η απεικόνιση του νευρωνικού δικτύου κάθε οργανισμού είναι τόσο μπερδεμένη που δεν μπορεί να χρησιμοποιηθεί για την εξαγωγή συμπερασμάτων. Ένα παράδειγμα φαίνεται στο Σχ. 9.3.



Σχέδιο 9.3, Ένα ιδιαίτερα μπερδεμένο νευρωνικό δίκτυο

9.2.3. Εύκολη διαταραχή φαινότυπου

Τέλος, ένα από τα σημαντικότερα προβλήματα της προσομοίωσης ήταν ότι μια μικρή αλλαγή του γονοτύπου κάποιου οργανισμού μπορεί να οδηγήσει σε εντελώς διαφορετικό φαινότυπο - νευρωνικό δίκτυο άρα και σε εντελώς διαφορετική συμπεριφορά. Αυτό έχει ως αποτέλεσμα οι γενετικά συμφέρουσες συμπεριφορές να χάνουν εύκολα την αξία τους μέσω της αναπαραγωγής και των μεταλλάξεων. Ένα σωστό σύστημα παραγωγής θα πρέπει να προστατεύει τον φαινότυπο του κάθε οργανισμού από μικρές αλλαγές στον γονότυπο αλλά παράλληλα να είναι αρκετά ευέλικτο και να επιτρέπει ριζικές αλλαγές ώστε να προωθείται η εξέλιξη.

10. Δυνατότητες επέκτασης

Στο κεφάλαιο αυτό θα μελετήσουμε τις δυνατότητες μελλοντικής επέκτασης της εργασίας και θα προτείνουμε κάποιες πιθανές αλλαγές και προσθήκες.

10.1. Βελτιώσεις στην ταχύτητα εκτέλεσης

Ένας σημαντικός περιοριστικός παράγοντας στην εξαγωγή χρήσιμων συμπερασμάτων από την προσομοίωση που τρέξαμε ήταν η υπολογιστική πολυπλοκότητα. Η μελέτη της εξέλιξης και των αναδυόμενων συμπεριφορών έχει νόημα μόνο σε βάθος χρόνου. Στο σύστημά μας, όταν η πολυπλοκότητα ανέβει πάνω από 400-500 παράλληλους οργανισμούς η προσομοίωση γίνεται τόσο αργή που χάνει το νόημά της. Θα είχε ενδιαφέρον να τρέξουμε τον κώδικα σε ένα ισχυρότερο μηχάνημα από αυτό που είχαμε στην διάθεσή μας αλλά η κύρια αιτία της καθυστέρησης είναι ο κώδικας. Ο κώδικας του κύκλου επανάληψης της προσομοίωσης δεν είναι βελτιστοποιημένος με αποτέλεσμα να αυξάνεται δυσανάλογα ο χρόνος επεξεργασίας όταν ο αριθμός των προσομοιούμενων οργανισμών πληθαίνει.

Οπότε μια δυνατότητα μελλοντικής επέκτασης είναι η τροποποίηση του κώδικα με γνώμονα την ταχύτητα, ενδεχομένως σε μια γλώσσα προγραμματισμού με περισσότερες δυνατότητες από την MATLAB και η χρήση πιο δυνατού μηχανήματος ώστε να μπορούν να μελετηθούν πιο σύνθετες συμπεριφορές σε μεγαλύτερο βάθος χρόνου.

10.2. Αλλαγές στο νευρωνικό δίκτυο

Η συμπεριφορά των οργανισμών της προσομοίωσης ελέγχεται από το νευρωνικό δίκτυο που λειτουργεί ως εγκέφαλος τους οπότε η δημιουργία και η λειτουργία του κάθε νευρωνικού δικτύου είναι από τα πιο σημαντικά σημεία της προσομοίωσης. Στα πλαίσια της εργασίας αυτής χρησιμοποιήσαμε ένα στατικό νευρωνικό δίκτυο του οποίου τα βάρη ελέγχονται από το γενετικό υλικό του αντίστοιχου οργανισμού. Υπάρχουν διάφορες δυνατότητες τροποποίησης και βελτίωσης του συστήματος αυτού.

10.2.1. Μη στατικό νευρωνικό δίκτυο

Η πιο σημαντική αλλαγή που μπορεί να γίνει στο σύστημα του νευρωνικού δικτύου είναι η δυναμική του σχεδίαση με βάση κάποιους κανόνες παραγωγής οι οποίοι θα περιέχονται στο γενετικό υλικό του κάθε οργανισμού. Αυτό σημαίνει ότι το γενετικό υλικό θα περιέχει πληροφορίες για την δομή του δικτύου αντί να περιέχει μόνο πληροφορίες για τα βάρη ενός στατικού νευρωνικού δικτύου. Εφόσον δεν θα υπάρχει άνω όριο στους δυνατούς συνδυασμούς

του νευρωνικού δικτύου το σύστημα θα έχει την δυνατότητα μη πεπερασμένης εξελισιμότητας και θα επιτραπούν σταδιακά πιο σύνθετες συμπεριφορές.

Είναι σημαντικό η σχεδίαση του συστήματος κανόνων παραγωγής να γίνει έτσι ώστε μικρές ή μερικές αλλαγές στο γενετικό υλικό να μην οδηγούν σε εντελώς νέα δομή νευρωνικού δικτύου αλλά να τροποποιούν ή να προσθέτουν στην προηγούμενη δομή. Με αυτόν τον τρόπο μια αλλαγή από μετάλλαξη ή συνδυασμό γενετικών υλικών θα διατηρεί ένα μεγάλο μέρος των ιδιοτήτων του αρχικού νευρωνικού δικτύου.

Τα L-systems που περιγράψαμε στο Κεφ. 4.4. είναι συστήματα παράλληλων κανόνων παραγωγής τα οποία προσφέρονται για την δυναμική δημιουργία των νευρωνικών δικτύων. Το γενετικό υλικό του κάθε οργανισμού μπορεί να περιέχει σε κωδικοποιημένη μορφή ένα σύνολο κανόνων γραμματικής οι οποίοι κατά την διαδικασία της δημιουργίας να εφαρμόζονται παράλληλα και επαναληπτικά για την ανάπτυξη του νέου νευρωνικού δικτύου.

10.2.2.Μάθηση νευρωνικού δικτύου

Το νευρωνικό δίκτυο που χρησιμοποιούμε στο σύστημά μας έχει σταθερά, μη μεταβλητά βάρη μετά τη δημιουργία του. Δεν υπάρχει μάθηση κατά τη διάρκεια της ζωής του κάθε οργανισμού και η αλλαγή και προσαρμογή του νευρωνικού δικτύου γίνεται σε βάθος χρόνου ανά γενιά μέσω της διαδικασίας της αναπαραγωγής. Αυτό σιγουρεύει ότι οι προκύπτουσες συμπεριφορές οφείλονται στην εξέλιξη και την κληρονομικότητα και όχι στην προσαρμογή στο περιβάλλον.

Θα μπορούσαμε να πειραματιστούμε προσθέτοντας παράλληλα και κάποιο είδος μάθησης κατά τη διάρκεια της ζωής του κάθε οργανισμού, σιγουρεύοντας όμως με κάποιον τρόπο ότι δεν χάνει το ρόλο της η εξέλιξη ως πρωτεύων μηχανισμός ανάπτυξης.

10.2.3.Περισσότερες επιλογές απεικόνισης

Ένας από τους στόχους της εργασίας αυτής ήταν να επιτρέψει την μελέτη της συσχέτισης ανάμεσα στην συμπεριφορά ενός οργανισμού και του νευρωνικού του δικτύου χρησιμοποιώντας γραφικές απεικονίσεις του δικτύου αυτού. Παρ' όλα αυτά παρατηρήσαμε ότι οι επιλογές εμφάνισης δεν ήταν αρκετές για να έχουμε μια ικανοποιητική εικόνα της συσχέτισης αυτής.

Μια προσθήκη που θα οδηγούσε σε αρκετά απλούστερη εμφάνιση θα ήταν η δυνατότητα ομαδοποίησης κάποιων νευρονίων, για παράδειγμα των νευρονίων που αντιστοιχούν στην όραση. Ομαδοποιώντας τα νευρόνια θα γίνει πιο εύκολο να ανιχνευθεί η σχέση μεταξύ ερεθισμάτων και συμπεριφοράς του κάθε οργανισμού.

Χρήσιμη προσθήκη στις επιλογές εμφάνισης θα ήταν επίσης η απεικόνιση των συναπτικών σχέσεων μεταξύ συγκεκριμένων νευρονίων, επιλεγόμενων από το χρήστη. Οι αλλαγές αυτές στην εμφάνιση του νευρωνικού δικτύου θα πρέπει να μπορούν να γίνονται δυναμικά κατά την διάρκεια της προσομοίωσης.

10.3. Περισσότερα στατιστικά στοιχεία

Ειδικά για μεγάλους πληθυσμούς οργανισμών δεν είναι πάντα εφικτή η οπτική παρατήρηση των διαφόρων ειδών και συμπεριφορών που εμφανίζονται. Μια πιθανή επέκταση της εφαρμογής είναι λοιπόν η προσθήκη περισσότερων στατιστικών στοιχείων και ειδικότερα περισσότερων στατιστικών στοιχείων πληθυσμών.

Ομαδοποιώντας με κάποιο τρόπο τους οργανισμούς που έχουν παρόμοιο γενετικό υλικό άρα ανήκουν στο ίδιο είδος θα είναι δυνατή η μέτρηση και απεικόνιση του πληθυσμού των ομάδων αυτών. Η σχεσιακή ανάλυση των πληθυσμών θα μπορούσε να παράγει χρήσιμα στοιχεία για την πορεία της προσομοίωσης.

Παράρτημα Α – Βιβλιογραφία

1. McCarthy, J. (2007). “What is Artificial Intelligence?”. <http://www-formal.stanford.edu/jmc/whatisai/whatisai.html>
2. Thomason R. (2003). “Logic and Artificial Intelligence”. Stanford Encyclopedia of Philosophy. <http://plato.stanford.edu/entries/logic-ai/>
3. McCarthy, J., Minsky, M. L., Rochester, N., Shannon, C. E. (1955). “A Proposal for the Dartmouth Summer Research project on Artificial Intelligence”. <http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>
4. Turing, A.M. (1950). “Computing machinery and intelligence”. *Mind*, 59, 433-460. <http://loebner.net/Prizet/TuringArticle.html>
5. Minsky, M. (1967). “Computation: Finite and Infinite Machines”. Englewood Cliffs, N.J.: Prentice-Hall. ISBN 0131654497, όπως αναφέρθηκε στο http://en.wikipedia.org/wiki/Artificial_intelligence
6. McCorduck, P. (2004). “Machines Who Think (2nd ed.)”. Natick, MA: A. K. Peters, Ltd., ISBN 1-56881-205-1, όπως αναφέρθηκε στο http://en.wikipedia.org/wiki/Artificial_intelligence
7. Kurzweil, R. (2005). “The Singularity is Near”. Penguin Books. ISBN 0-670-03384-7.
8. Βλαχάβας, Ι., Κεφαλάς, Π., Βασιλειάδης, Ν., Κόκκορας, Φ., Σακελλαρίου, Η. (2005). “Τεχνητή Νοημοσύνη, Β’ Έκδοση” (σελ. 10-12). Εκδόσεις Γαρταγάνη, Θεσσαλονίκη. ISBN 960-7013-35-2.
9. Russell, S. J., Norvig, P. (2003). “Artificial Intelligence: A Modern Approach (2nd ed.)”. Upper Saddle River, NJ: Prentice Hall. ISBN 0-13-790395-2, όπως αναφέρθηκε στο http://en.wikipedia.org/wiki/Artificial_intelligence
10. Kurzweil, R. (2001). “The Law of Accelerating Returns”. <http://www.kurzweilai.net/articles/art0134.html?printable=1>
11. Open Mind Common Sense. commons.media.mit.edu/
12. Cycorp, Inc. <http://www.cyc.com/cyc/cycrandd/overview>
13. Τζαφέστας, Σ. (2002). “Υπολογιστική Νοημοσύνη, Τόμος Α: Μεθοδολογίες” (σελ. 101-102). Αθήνα, 2002.
14. Brooks, R. (1990). “Elephants Don't Play Chess”. <http://people.csail.mit.edu/brooks/papers/elephants.pdf>
15. AARON. <http://www.kurzweilcyberart.com/>
16. Kurzweil, R. (2005). “The Singularity is Near” (σελ. 260). Viking Press, όπως αναφέρθηκε στο http://en.wikipedia.org/wiki/Strong_AI
17. The Singularity Institute for Artificial Intelligence. <http://singinst.org/research/summary>
18. Όπως παραπάνω [8], σελ. 4.
19. Haugeland, J. (1985). “Artificial Intelligence: The Very Idea”. Cambridge, Mass.: MIT Press. ISBN 0-262-08153-9, όπως αναφέρθηκε στο http://en.wikipedia.org/wiki/Good_old_fashioned_artificial_intelligence
20. Blue Brain Project, Frequently Asked Questions. <http://bluebrain.epfl.ch/page18924.html>
21. BEAM Robotics. <http://encyclobeamia.solarbotics.net/articles/beam.html>
22. Όπως παραπάνω [13], σελ. 1-2.
23. Gould, S.J. (2002). “The Structure of Evolutionary Theory”. Cambridge: Belknap Press (Harvard University Press). ISBN 0-674-00613-5, όπως αναφέρθηκε στο <http://en.wikipedia.org/wiki/Evolution>
24. Darwin, C. (1859). “On the Origin of Species (1st ed.)” (σελ. 1). London: John Murray, όπως αναφέρθηκε στο <http://en.wikipedia.org/wiki/Evolution>
25. Kutschera U., Niklas K. (2004). “The modern theory of biological evolution: an expanded synthesis”. *Naturwissenschaften* 91 (6): 255–76, όπως αναφέρθηκε στο <http://en.wikipedia.org/wiki/Evolution>

26. Hurst L. D. (February 2009). "Fundamental concepts in genetics: genetics and the understanding of selection". *Nat. Rev. Genet.* 10 (2): 83–93, όπως αναφέρθηκε στο <http://en.wikipedia.org/wiki/Evolution>
27. Όπως παραπάνω [13], σελ. 298-299.
28. Harwood AJ (1998). "Factors affecting levels of genetic diversity in natural populations". *Philos. Trans. R. Soc. Lond., B, Biol. Sci.* 353 (1366): 177–86, όπως αναφέρθηκε στο <http://en.wikipedia.org/wiki/Evolution>
29. Nei M. (2005). "Selectionism and neutralism in molecular evolution". *Mol. Biol. Evol.* 22 (12): 2318–42, όπως αναφέρθηκε στο <http://en.wikipedia.org/wiki/Evolution>
30. Kilicay H. N., Dagli H. C. (2003). "Emergence and Artificial Life" (σελ. 581). IEEE.
31. Holland J. H. (1997). "Emergence". Addison-Wesley, όπως αναφέρθηκε στο [30]
32. Emergence. <http://en.wikipedia.org/wiki/Emergence>
33. Schneider M. O., Rosa J. L. G. (2005). "BIOANT – Biologically Plausible Computer Simulation of an Environment with Ants". IEEE.
34. Barabasi, A. L., Hawoong J., Reka A. (1999), "The Diameter of the World Wide Web", *Nature* 401: 130-131, όπως αναφέρθηκε στο <http://en.wikipedia.org/wiki/Evolution>
35. Όπως παραπάνω [13], σελ. 293.
36. Wall M. "Introduction to Genetic Algorithms". <http://lancet.mit.edu/~mbwall/presentations/IntroToGAs/index.html>
37. Rennard J. P. (2000) "Introduction to Genetic Algorithm". <http://www.rennard.org/alife/english/gavintrgb.html>
38. Neural Networks. <http://www.statsoft.com/textbook/stneunet.html>
39. Όπως παραπάνω [13], σελ. 3-4.
40. Όπως παραπάνω [13], σελ. 94-97.
41. Stergiou C., Siganos D. (1996). "Neural Networks". http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html
42. Όπως παραπάνω [13], σελ. 101-110.
43. Ghahramani Z. (2004). "Unsupervised learning" (σελ. 3). <http://www.gatsby.ucl.ac.uk/~zoubin/course05/ul.pdf>
44. Cellular Automaton. http://en.wikipedia.org/wiki/Cellular_automaton
45. Wolfram S. (1982) "Cellular Automata as Simple Self-Organizing Systems". <http://www.stephenwolfram.com/publications/articles/ca/82-cellular/>
46. Wolfram S. (1988) "Cellular Automaton Supercomputing". <http://www.stephenwolfram.com/publications/articles/ca/88-cellular/>
47. Ochoa G. (1998). "An Introduction to Lindenmayer Systems". <http://www.cogs.susx.ac.uk/users/gabro/lsys/lsys.html>
48. Prusinkiewicz P., Lindenmayer A. (1990). "The Algorithmic Beauty of Plants". <http://algorithmicbotany.org/papers/#abop>
49. Dickau R. M. (1997). "2D L-Systems". <http://mathforum.org/advanced/robertd/lsys2d.html>
50. Braitenberg, V. (1984). "Vehicles: Experiments in synthetic psychology". Cambridge, MA: MIT Press, όπως αναφέρθηκε στο http://en.wikipedia.org/wiki/Braitenberg_vehicles
51. LaFave S. "Notes on Braitenberg's Vehicles". http://instruct.westvalley.edu/lafave/Vehicles_online.html
52. Christopher G. Langton. (1997). "Artificial Life: An Overview"
53. Christopher G. Langton. (1992). "Artificial Life", σελ. 23-24. http://www.probelog.com/texts/Langton_al.pdf
54. Ray, T. S. 1993. "How I created life in a virtual universe". <http://life.ou.edu/pubs/nathist/>
55. A. D. Channon, "Evolutionary Emergence: The Struggle for Existence in Artificial Biota", σελ. 30-34. University of Southampton, 2001.
56. Tom Barbalet. "Noble Ape Simulation". <http://www.nobleape.com/>
57. Όπως παραπάνω [55], σελ. 26-29.

Παράρτημα Β – Κώδικας

```
function tests
clear all; clc; close all;

ctime = clock;
disp(sprintf(['%d:%02d:%02d]      Initializing,',      ctime(4),      ctime(5),
round(ctime(6))));
state = 'restart';

InitialPopulation = 100;
WorldSize = 140;
out_size = 5;
SWindowDefault = 1;
inter_sizeDefault = 12;
Frameskip = 0;
GreenShow = 1;
RedShow = 1;
ColorMode = 0;
MutationRate = 0.05;

DataGraphWindow = 100;

while(strcmp(state,'quit') == 0)

    if strcmp(state,'restart')

        ctime = clock;
        disp(sprintf(['%d:%02d:%02d] Building world,', ctime(4), ctime(5),
round(ctime(6))));
        viewNNid = 1;
        Population = 0;
        DeathCount = 0;
        BirthCount = 0;
        SWindow = SWindowDefault;
        inter_size = inter_sizeDefault;

        id = 0;
        Time = 0;
        CMap = [1 1 1; 1 0 0; 0 0 0; 0.7 0.9 0.7];
        turn = [-1 -1 0 -1 1 -1 1 0 1 1 0 1 -1 1 -1 0];
        WorldGridBp = ones(WorldSize);
        IDMap = zeros(WorldSize);
        for i=1:WorldSize
            WorldGridBp(i,1) = 3;
            WorldGridBp(1,i) = 3;
            WorldGridBp(i,WorldSize) = 3;
            WorldGridBp(WorldSize,i) = 3;
        end
        WorldGrid = WorldGridBp;

        ctime = clock;
        sC = [1 0; 0 0; 0 0];
        disp(sprintf(['%d:%02d:%02d] Creating initial population,', ctime(4),
ctime(5), round(ctime(6))));
        for i=1:InitialPopulation
            id = id + 1;
            Population = Population + 1;
```

```

        DNA =
round((rand(1,inter_size*((2*SWindow+1)^2+size(matrix2vector(sC),2)+out_size)
)-0.5)*2);
        position = [10+round(rand*(WorldSize-20))
10+round(rand*(WorldSize-20))];
        WorldGrid(position(1),position(2)) = 2;
        IDMap(position(1),position(2)) = id;
        Creatures(i) = struct('id', id, 'DNA', DNA, 'position',
position, 'direction', turn, 'condition', sC);
        end

        iC = Creatures(1).condition;
        SGrid = [zeros(SWindow-1,size(WorldGrid,2)+(SWindow-1)*2);
zeros(size(WorldGrid,1),SWindow-1) WorldGrid zeros(size(WorldGrid,1),SWindow-
1); zeros(SWindow-1,size(WorldGrid,2)+(SWindow-1)*2)];
        iS = SGrid(Creatures(1).position(1)-
1:Creatures(1).position(1)+2*SWindow-1,
Creatures(1).position(2)-
1:Creatures(1).position(2)+2*SWindow-1);

        end

        if strcmp(state,'load')
            [filename, pathname] = uigetfile({'*.mat'}, 'Load simulation state:');
            if filename ~= 0
                load(strcat(pathname,filename));
                ctime = clock;
                disp(sprintf('%d:%02d:%02d] Loading data from %s,', ctime(4),
ctime(5), round(ctime(6)),filename));
            end
        end

        ctime = clock;
        disp(sprintf('%d:%02d:%02d] Creating GUI,', ctime(4), ctime(5),
round(ctime(6))));
        close all;
        Fig1 = figure('Name','World
View','NumberTitle','off','Menubar','None','Toolbar','None','Visible','on','P
osition',[50,50,5*WorldSize,5*WorldSize],'WindowButtonUpFcn',
{@worldclick_Callback});
        Fig2 =
figure('Name','Controls','NumberTitle','off','Menubar','None','Toolbar','None
','Visible','on','Position',[800 500 400 250]);
        MainWindow = axes('Units','Pixels','Parent', Fig1,
'Position',[0,0,5*WorldSize,5*WorldSize]);
        GraphWindow = axes('Units','Pixels','Parent', Fig2,
'Position',[140,10,245,110]);
        Fig3 = figure('Name','Neural Network
View','NumberTitle','off','Menubar','None','Toolbar','None','Visible','on','P
osition',[800 50 400 400],'WindowButtonUpFcn', {@nnclick_Callback});
        NNView = axes('Units','Pixels','Position',[10,10,380,380],
'Visible','Off');
        Fig4 = [];
        Fig5 = [];

        % load images

        imgPlay = imread('img\play.png');
        imgPause = imread('img\pause.png');

```



```

imgRestart = imread('img\restart.png');
imgSave = imread('img\save.png');
imgLoad = imread('img\load.png');
imgClose = imread('img\close.png');
imgFS = imread('img\FS.png');
imgInfo = imread('img\info.png');
imgScreenshot = imread('img\screenshot.png');
imgSettings = imread('img\settings.png');
imgColor0 = imread('img\color0.png');
imgColor1 = imread('img\color1.png');
if ColorMode, imgColor = imgColor1; else imgColor=imgColor0; end;

ButtonRed = uicontrol(Fig3, 'BackgroundColor', 'r',
'Style','pushbutton','String','on','Position',[360 10 30 30], 'Callback',
{@buttonred_Callback});
ButtonGreen = uicontrol(Fig3, 'BackgroundColor', 'g',
'Style','pushbutton','String','on','Position',[330 10 30 30], 'Callback',
{@buttongreen_Callback});
ButtonPause = uicontrol(Fig2,
'Style','pushbutton','CData',imgPause,'Position',[10 190 50 50], 'Callback',
{@pausebutton_Callback});
ButtonFS = uicontrol(Fig2,
'Style','pushbutton','String',sprintf('%x%d',Frameskip+1),'CData',imgFS,'Position',
[10 130 50 50], 'FontSize',14, 'FontWeight','bold', 'Callback',
{@fsbutton_Callback});
ButtonRestart = uicontrol(Fig2,
'Style','pushbutton','CData',imgRestart,'Position',[75 190 50 50],
'Callback', {@restartbutton_Callback});
ButtonSave = uicontrol(Fig2,
'Style','pushbutton','CData',imgSave,'Position',[235 190 50 50], 'Callback',
{@savebutton_Callback});
ButtonLoad = uicontrol(Fig2,
'Style','pushbutton','CData',imgLoad,'Position',[235 130 50 50], 'Callback',
{@loadbutton_Callback});
ButtonQuit = uicontrol(Fig2,
'Style','pushbutton','CData',imgClose,'Position',[335 190 50 50], 'Callback',
{@quitbutton_Callback});
ButtonColor = uicontrol(Fig2,
'Style','pushbutton','CData',imgColor,'Position',[75 130 50 50], 'Callback',
{@colorbutton_Callback});
ButtonSettings = uicontrol(Fig2,
'Style','pushbutton','CData',imgSettings,'Position',[170 190 50 50],
'Callback', {@settingsbutton_Callback});
ButtonInfo = uicontrol(Fig2,
'Style','pushbutton','CData',imgInfo,'Position',[335 130 50 50], 'Callback',
{@infobutton_Callback});
ButtonScreenshot = uicontrol(Fig2,
'Style','pushbutton','CData',imgScreenshot,'Position',[170 130 50 50],
'Callback', {@screenshotbutton_Callback});

NNIDText = uicontrol(Fig3, 'Style','text','BackgroundColor',
'w','FontSize',7,'String','Creature ID: 1','Position',[310 370 80 15]);
PauseText = uicontrol(Fig1, 'Style','text','BackgroundColor',
'w','FontSize',14,'String','Paused','Visible','off','Position',[675 720 80
30]);

Info1Text = uicontrol(Fig2, 'Style','text','BackgroundColor',
'w','FontSize',10,'String','Population:
','HorizontalAlignment','left','Position',[10 95 115 17]);
Info1Box = uicontrol(Fig2, 'Style','text','BackgroundColor',
'b','Position',[111 99 10 10]);

```

```

    Info2Text      =      uicontrol(Fig2,      'Style','text','BackgroundColor',
'w','FontSize',10,'String','Deaths:
','HorizontalAlignment','left','Position',[10 75 115 17]);
    Info2Box      =      uicontrol(Fig2,      'Style','text','BackgroundColor',
'r','Position',[111 79 10 10]);
    Info3Text      =      uicontrol(Fig2,      'Style','text','BackgroundColor',
'w','FontSize',10,'String','Births:
','HorizontalAlignment','left','Position',[10 55 115 17]);
    Info3Box      =      uicontrol(Fig2,      'Style','text','BackgroundColor',
'g','Position',[111 59 10 10]);
    Info4Text      =      uicontrol(Fig2,      'Style','text','BackgroundColor',
'w','FontSize',10,'String','Time:
','HorizontalAlignment','left','Position',[10 35 115 17]);
    Info5Text      =      uicontrol(Fig2,      'Style','text','BackgroundColor',
'w','FontSize',10,'String','msec/cy:
','HorizontalAlignment','left','Position',[10 15 115 17]);
    figure(Fig3);
    viewNN(Creatures(viewNNid).DNA, iS, iC, inter_size, out_size, NNView,
RedShow, GreenShow);
    imshow(WorldGrid, CMap, 'Parent', MainWindow);
    PopulationData = [zeros(1,DataGraphWindow-1) Population];
    DeathData = [zeros(1,DataGraphWindow-1) DeathCount];
    BirthData = [zeros(1,DataGraphWindow-1) BirthCount];
    state = 'run';
    ctime = clock;
    disp(sprintf(['%d:%02d:%02d] Starting simulation.', ctime(4), ctime(5),
round(ctime(6))]);
    while(strcmp(state,'run'))
        Time = Time + 1;
        ctime = clock;
        ClockPoint1 = ctime(6);
        set(Info4Text,'String',sprintf('Time: %10d', Time));
        if mod(Time, 5) == 0
            PopulationData = circshift(PopulationData, [0 -1]);
            PopulationData(DataGraphWindow) = Population;
            DeathData = circshift(DeathData, [0 -1]);
            DeathData(DataGraphWindow) = DeathCount;
            BirthData = circshift(BirthData, [0 -1]);
            BirthData(DataGraphWindow) = BirthCount;
            axes(GraphWindow);
            plot(PopulationData, 'LineWidth', 1.5);
            hold on;
            plot(DeathData,'-r', 'LineWidth', 1.5);
            hold on;
            plot(BirthData,'-g', 'LineWidth', 1.5);
            hold off;
            set(GraphWindow, 'XTick', [], 'YTick', [], 'YLimMode', 'manual',
'YLim', [0 max([400 max(PopulationData) DeathCount BirthCount])]);
        end
    end
    KillMatrix = [];
    ParentMatrix = [];
    set(Info1Text,'String',sprintf('Population: %d', Population));
    for i=1:Population
        %disp(i);
        x1 = Creatures(i).position(1)-1;
        x2 = Creatures(i).position(1)+2*SWindow-1;
        x3 = Creatures(i).position(2)-1;
        x4 = Creatures(i).position(2)+2*SWindow-1;
        SGrid      =      [zeros(SWindow-1,size(WorldGrid,2)+(SWindow-1)*2);
zeros(size(WorldGrid,1),SWindow-1) WorldGrid zeros(size(WorldGrid,1),SWindow-
1); zeros(SWindow-1,size(WorldGrid,2)+(SWindow-1)*2)];
        iS = SGrid(x1:x2, x3:x4);

```

```

o = nnet(iS, iC, inter_size, out_size, Creatures(i).DNA);

iC = [round(rand) o(1); o(2:3); o(4:5)];

% move
if o(1) == 1
    if WorldGrid(Creatures(i).position(1)
Creatures(i).direction(1),Creatures(i).position(2)
Creatures(i).direction(2)) < 2
        WorldGrid(Creatures(i).position(1),
Creatures(i).position(2)) = WorldGridBp(Creatures(i).position(1),
Creatures(i).position(2));
        IDMap(Creatures(i).position(1), Creatures(i).position(2))
= 0;
        Creatures(i).position = Creatures(i).position +
Creatures(i).direction(1:2);
        WorldGrid(Creatures(i).position(1),
Creatures(i).position(2)) = 2;
        IDMap(Creatures(i).position(1), Creatures(i).position(2))
= Creatures(i).id;
    end
end

% turn
if o(2:3) == [0 1]
    Creatures(i).direction = circshift(Creatures(i).direction,[0
-2]);
elseif o(2:3) == [1 0]
    Creatures(i).direction = circshift(Creatures(i).direction,[0
2]);
end

% reproduce
if o(4) == 1
    k = Creatures(i).position(1)+Creatures(i).direction(1);
    l = Creatures(i).position(2)+Creatures(i).direction(2);
    if WorldGrid(k,l)==2
        reproID = IDMap(k,l);
        reproIndex = find([Creatures.id]==reproID);
        if isempty(reproIndex)
            disp('BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB');
        else reproduce(Creatures(i), Creatures(reproIndex));
        end
    end
end

% kill
if o(5) == 1
    for k = (Creatures(i).position(1)
1):2:(Creatures(i).position(1) + 1)
        for l = (Creatures(i).position(2)
1):2:(Creatures(i).position(2) + 1)
            if WorldGrid(k,l)==2
                killID = IDMap(k,l);
                killIndex = find([Creatures.id]==killID);
                if isempty(killIndex)
                    disp('BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB');
                else kill(killIndex);
                end
            end
        end
    end
end

```

```

        end
    end
end
end
end
for c=1:size(ParentMatrix,1)
    Parent1 = ParentMatrix(c,1);
    Parent2 = ParentMatrix(c,2);
    position = 2*Parent1.position - Parent2.position;
    % disp(sprintf('%d %d] x [%d %d] = [%d
    %
    %d]',Parent1.position(1),Parent1.position(2),Parent2.position(1),Parent2.posi
    tion(2),position(1),position(2)));
    if WorldGrid(position(1),position(2)) == 1
    %
        if WorldGrid(position(1),position(2)) == 2
    %
            killID = IDMap(position(1),position(2));
    %
            killIndex = find([Creatures.id]==killID);
    %
            if isempty(killIndex)
    %
                disp('BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB');
    %
            else kill(killIndex);
    %
            end
    %
        end
        Population = Population + 1;
        BirthCount = BirthCount + 1;
        set(Info3Text, 'String', sprintf('Births: %8d', BirthCount));
        DNA = crossover(Parent1.DNA, Parent2.DNA, MutationRate);
        id = id + 1;
        WorldGrid(position(1),position(2)) = 2;
        IDMap(position(1),position(2)) = id;
        Creatures(Population) = struct('id', id, 'DNA', DNA,
'position', position, 'direction', turn, 'condition', sC);
        disp(sprintf('Reproduction: ID %d with ID %d', Parent1.id,
Parent2.id));
    end
end
for c=1:size(KillMatrix,2)
    killIndex = KillMatrix(c);
    disp(sprintf('Death: ID %d', Creatures(killIndex).id));
    WorldGrid(Creatures(killIndex).position(1),
Creatures(killIndex).position(2)) =
WorldGridBp(Creatures(killIndex).position(1),
Creatures(killIndex).position(2));
    IDMap(Creatures(killIndex).position(1),
Creatures(killIndex).position(2)) = 0;
    Creatures = [Creatures(1:killIndex-1)
Creatures(killIndex+1:Population)];
    KillMatrix = KillMatrix - (KillMatrix > killIndex);
    Population = Population - 1;
    DeathCount = DeathCount + 1;
    set(Info2Text, 'String', sprintf('Deaths: %6d', DeathCount));

end
if mod(Time, Frameskip+1) == 0
    if ColorMode == 0
        imshow(WorldGrid, CMap, 'Parent', MainWindow);
    else
        imshow(mod(IDMap,256), [1 1 1; jet(256)],
'Parent',MainWindow);
    end
end
end
pause(0.01);
ctime = clock;
ClockDiff = round((ctime(6) - ClockPoint1)*1000);
set(Info5Text, 'String', sprintf('msec/cy: %5d', ClockDiff));

```

```

        while strcmp(state, 'pause')
            pause(0.1);
        end
    end
end

end

close all;
ctime = clock;
disp(sprintf('%d:%02d:%02d      Quitting.',      ctime(4),      ctime(5),
round(ctime(6))));

function reproduce(Parent1, Parent2)
    if isempty(ParentMatrix) || isempty(find([ParentMatrix.id] ==
Parent1.id))
        ParentMatrix = [ParentMatrix; Parent1 Parent2];
    end
end

function kill(killIndex)
    if isempty(find(KillMatrix == killIndex))
        KillMatrix = [KillMatrix killIndex];
    end
end

% Callbacks

function buttongreen_Callback(hObject, eventdata)
    if GreenShow == 0
        set(ButtonGreen, 'String', 'on');
        GreenShow = 1;
    else
        set(ButtonGreen, 'String', 'off');
        GreenShow = 0;
    end
    viewNN(Creatures(viewNNid).DNA, iS, iC, inter_size, out_size, NNView,
RedShow, GreenShow);
end

function buttonred_Callback(hObject, eventdata)
    if RedShow == 0
        set(ButtonRed, 'String', 'on');
        RedShow = 1;
    else
        set(ButtonRed, 'String', 'off');
        RedShow = 0;
    end
    viewNN(Creatures(viewNNid).DNA, iS, iC, inter_size, out_size, NNView,
RedShow, GreenShow);
end

function pausebutton_Callback(hObject, eventdata)
    if strcmp(state, 'pause')
        state = 'run';
        % set(ButtonPause, 'String', 'Pause');
        set(ButtonPause, 'CData', imgPause);
        set(PauseText, 'Visible', 'off');
    else
        state = 'pause';
        % set(ButtonPause, 'String', 'Continue');
        set(ButtonPause, 'CData', imgPlay);
    end
end

```

```

        set(PauseText, 'Visible', 'on');
    end
end

function restartbutton_Callback(hObject,eventdata)
    disp(sprintf(['%d:%02d:%02d]   Restarting!',   ctime(4),   ctime(5),
round(ctime(6))));
    state = 'restart';
end

function quitbutton_Callback(hObject,eventdata)
    state = 'quit';
end

function infobutton_Callback(hObject,eventdata)
    if ishandle(Fig4)
        close(Fig4);
    else
        Fig4
        =
figure('Name','About','NumberTitle','off','Menubar','None','ToolBar','None','
Visible','on','Position',[850 250 300 150]);
        AboutText = uicontrol(Fig4, 'Style','text','BackgroundColor',
'w','FontSize',10,'String',' ','Position',[10 10 280 130]);
        set(AboutText,'String',textwrap(AboutText, {' ','Artificial Life
testing & simulation program','Version: 4',' ',' ','Papadopoulos-Korfiatis
Alexandros','NTUA, 2009'}));
    end
end

function settingsbutton_Callback(hObject,eventdata)
    if ishandle(Fig5)
        close(Fig5);
    else
        Fig5
        =
figure('Name','Edit   starting
parameters','NumberTitle','off','Menubar','None','ToolBar','None','Visible','
on','Position',[800 50 400 400]);
        Panell = uipanel('Title','','Position',[0.025 0.65 0.46 .3]);
        Text1
        =
uicontrol(Panell,
'Style','text','FontSize',12,'String','Initial Population','Position',[10 85
160 25]);
        Text2
        =
uicontrol(Panell,
'Style','text','FontSize',18,'String',InitialPopulation,'Position',[10 65 160
25]);
        PopSlider
        =
uicontrol(Panell,'Style','slider','BackgroundColor','w','Visible','on','Min',
1,'Max',300,'Value',InitialPopulation,'Position',[10   15   160
35],'SliderStep',[1/30 1/3],'Callback', {@popslider_Callback});
        Panel2 = uipanel('Title','','Position',[0.515 0.65 0.46 .3]);
        Text3
        =
uicontrol(Panel2,
'Style','text','FontSize',12,'String','Sight window:','Position',[10 85 160
25]);
        Box1
        =
axes('Units','Pixels','Parent',   Panel2,
'Position',[80,10,70,70]);
        ButtonPlus
        =
uicontrol(Panel2,
'Style','pushbutton','String','+','Position',[20 50 30 30], 'Callback',
{@plusbutton_Callback});
        ButtonMinus = uicontrol(Panel2, 'Style','pushbutton','String','-
','Position',[20 10 30 30], 'Callback', {@minusbutton_Callback});
        imgSWind = zeros(13);
        imgSWind(7-SWindowDefault:7+SWindowDefault,7-
SWindowDefault:7+SWindowDefault) = 3;
        imgSWind(7,7) = 2;
        imshow(imgSWind,[1 1 1; 1 0 0; 0.7 0.9 0.7],'Parent',Box1);
    end
end

```

```

        Panel3 = uipanel('Title','','Position',[0.025 0.025 0.95 .6]);
        Box2 = axes('Units','Pixels','Parent', Panel3,
'Position',[10,30,355,180]);
        ButtonPlus2 = uicontrol(Panel3,
'Style','pushbutton','String','+', 'Position',[178 215 20 20], 'Callback',
{@plus2button_Callback});
        ButtonMinus2 = uicontrol(Panel3, 'Style','pushbutton','String','-
', 'Position',[178 5 20 20], 'Callback', {@minus2button_Callback});
        plotpreview;
    end
    function plotpreview
        input_size = (2*SWindowDefault+1)^2 + out_size;
        DiagM = diag(ones(1,input_size+inter_sizeDefault+out_size));
        topo2 = [];
        layers2 = [input_size inter_sizeDefault out_size];
        for i=1:size(layers2,2)
            offset2 = (max(layers2)-layers2(i))/2;
            for j=1:layers2(i)
                topo2 = [topo2; i j+offset2];
            end
        end
        gplot(DiagM, topo2, '-ob');
        axis([0.5 size(layers2,2)+0.5 0 max(layers2)+1]);
        set(Box2, 'XTick', [], 'YTick', []);
    end
    function popslider_Callback(hObject,eventdata)
        InitialPopulation = round(get(PopSlider, 'Value'));
        disp(InitialPopulation);
        set(Text2, 'String', InitialPopulation);
    end
    function plusbutton_Callback(hObject,eventdata)
        if SWindowDefault < 5
            SWindowDefault = SWindowDefault + 1;
            imgSWind = zeros(13);
            imgSWind(7-SWindowDefault:7+SWindowDefault, 7-
SWindowDefault:7+SWindowDefault) = 3;
            imgSWind(7,7) = 2;
            imshow(imgSWind,[1 1 1; 1 0 0; 0.7 0.9 0.7], 'Parent', Box1);
        end
        plotpreview;
    end
    function minusbutton_Callback(hObject,eventdata)
        if SWindowDefault > 1
            SWindowDefault = SWindowDefault - 1;
            imgSWind = zeros(13);
            imgSWind(7-SWindowDefault:7+SWindowDefault, 7-
SWindowDefault:7+SWindowDefault) = 3;
            imgSWind(7,7) = 2;
            imshow(imgSWind,[1 1 1; 1 0 0; 0.7 0.9 0.7], 'Parent', Box1);
        end
        plotpreview;
    end
    function plus2button_Callback(hObject,eventdata)
        if inter_sizeDefault < 30
            inter_sizeDefault = inter_sizeDefault + 1;
            plotpreview;
        end
    end
    function minus2button_Callback(hObject,eventdata)
        if inter_sizeDefault > 1
            inter_sizeDefault = inter_sizeDefault - 1;
            plotpreview;
        end
    end
end

```

```

end

function fsbutton_Callback(hObject,eventdata)
    Frameskip = rem(Frameskip + 1, 5);
    set(ButtonFS, 'String', sprintf('%d', Frameskip+1));
end

function worldclick_Callback(hObject,eventdata)
    if strcmp(state, 'pause')
        CP = round(get(Fig1, 'CurrentPoint')./5);
        ClickPoint = [WorldSize - CP(2) CP(1)];
        clickID = IDMap(ClickPoint(1), ClickPoint(2));
        if clickID ~= 0
            clickIndex = find([Creatures.id]==clickID);
            if ColorMode == 0
                HighlightView = WorldGrid;
                HighlightView(Creatures(clickIndex).position(1)-
SWindow:Creatures(clickIndex).position(1)+SWindow,
Creatures(clickIndex).position(2)-
SWindow:Creatures(clickIndex).position(2)+SWindow) = 4;

                HighlightView(Creatures(clickIndex).position(1),Creatures(clickIndex).positio
n(2)) = 2;
                imshow(HighlightView, CMap, 'Parent', MainWindow);
            else
                HighlightView = IDMap;
                HighlightView(Creatures(clickIndex).position(1)-
SWindow:Creatures(clickIndex).position(1)+SWindow,
Creatures(clickIndex).position(2)-
SWindow:Creatures(clickIndex).position(2)+SWindow) = 128;

                HighlightView(Creatures(clickIndex).position(1),Creatures(clickIndex).positio
n(2)) = Creatures(clickIndex).id;
                imshow(mod(HighlightView,256), [1 1 1; jet(256)],
'Parent', MainWindow);
            end
        end

        figure(Fig3);
        viewNNid = clickID;
        viewNN(Creatures(clickIndex).DNA, iS, iC, inter_size,
out_size, NNView, RedShow, GreenShow);
        set(NNIDText, 'String', sprintf('Creature ID: %d',viewNNid));
    end
end
end

function nnclick_Callback(hObject,eventdata)
    if strcmp(get(Fig3, 'Toolbar'), 'none')
        set(Fig3, 'Toolbar', 'figure');
    else
        set(Fig3, 'Toolbar', 'none');
    end
end

function savebutton_Callback(hObject,eventdata)
    [filename, pathname] = uiputfile({'*.mat'}, 'Save simulation state
as:');
    if filename ~= 0

save(strcat(pathname, filename), 'WorldSize', 'inter_size', 'out_size', 'SWindow',
'Frameskip', 'GreenShow', 'RedShow', 'viewNNid', 'InitialPopulation', 'Population'
,

```



```

'id','Time','CMap','turn','WorldGridBp','IDMap','WorldGrid','Creatures','iC',
'iS','sC');
    ctime = clock;
    disp(sprintf(['%d:%02d:%02d] Saved simulation state to %s',
ctime(4), ctime(5), round(ctime(6)), filename));
end
end

function loadbutton_Callback(hObject,eventdata)
    state = 'load';
end

function colorbutton_Callback(hObject,eventdata)
    if ColorMode == 1
        ColorMode = 0;
        set(ButtonColor,'CData',imgColor0);
    else
        ColorMode = 1;
        set(ButtonColor,'CData',imgColor1);
    end
end

function screenshotbutton_Callback(hObject,eventdata)
    [filename, pathname] = uiputfile({'*.png'},'Save screenshot:');
    if filename ~= 0
        if ColorMode == 0
            imwrite(WorldGrid, CMap, strcat(pathname,filename), 'png');
        else
            imwrite(mod(IDMap,256), [1 1 1; jet(256)],
strcat(pathname,filename), 'png');
        end
    end
end

function viewNN(DNA, iS, iC, inter_size, out_size, NNView, RedShow,
GreenShow)

aS = matrix2vector(iS);
aC = matrix2vector(iC);
a = [aS aC];
a_size = size(a,2);
A = vector2matrix(DNA(1:inter_size*a_size),a_size,inter_size);
B =
vector2matrix(DNA(1+inter_size*a_size:inter_size*(a_size+out_size)),inter_size,
out_size);

Z = diag(ones(1,a_size+inter_size+out_size));
M = zeros(a_size+inter_size+out_size);

M(1:size(A,1),1+a_size:a_size+size(A,2)) = A;
M(1+a_size:a_size+size(B,1),1+a_size+inter_size:a_size+inter_size+size(B,2))
= B;

topo = [];
layers = [a_size inter_size out_size];
for i=1:size(layers,2)
    offset = (max(layers)-layers(i))/2;
    for j=1:layers(i)
        topo = [topo; i j+offset];
    end
end
end

```

```

topo = [flipud(topo(1:a_size,:)); flipud(topo(1+a_size:a_size+inter_size,:));
flipud(topo(1+a_size+inter_size:a_size+inter_size+out_size,:))];

if GreenShow == 1
    gplot(M>0, topo, '-g');
    hold on;
end
if RedShow == 1
    gplot(M<0, topo, ':r');
    hold on;
end
gplot(Z, topo, '-ob');
hold off;

axis([0.5 size(layers,2)+0.5 0 max(layers)+1]);
set(NNView, 'XTick', [], 'YTick', []);

end

function o = nnet(iS, iC, inter_size, out_size, DNA)

aS = matrix2vector(iS);
aC = matrix2vector(iC);
a = [aS aC];
a_size = size(a,2);

A = vector2matrix(DNA(1:inter_size*a_size),a_size,inter_size);
B = vector2matrix(DNA(1+inter_size*a_size:inter_size*(a_size+out_size)),inter_size,
out_size);

for j=1:inter_size
    b(j) = 0;
    for i=1:a_size
        b(j) = b(j) + a(i)*A(i,j);
    end
    b(j) = stepf(b(j));
end

for j=1:out_size
    o(j) = 0;
    for i=1:inter_size
        o(j) = o(j) + b(i)*B(i,j);
    end
    o(j) = stepf(o(j));
end
end

function output = stepf(input)
if input > 0
    output = 1;
else
    output = 0;
end
end

function outDNA = crossover(DNA1, DNA2, mutation)

```

```

s1 = size(DNA1, 2);
s2 = size(DNA2, 2);

crossover = rand;
point1 = round(crossover*s1);
point2 = round(crossover*s2);

outDNA = [DNA1(1:point1) DNA2(point2+1:s2)];
if rand < mutation
    mutationpoint = 1+round(rand*(size(outDNA,2)-1));
    outDNA(mutationpoint) = round(2*(rand-0.5));
end

end

function output = vector2matrix(input, s1, s2)

if (s1*s2 == size(input,2))
    for i=1:s1
        for j=1:s2
            output(i,j) = input(j+s2*(i-1));
        end
    end
else
    output=[];
    disp('Error on vector2matrix');
end

end

function output = matrix2vector(input)

for i=1:size(input,1)
    for j=1:size(input,2)
        output(j+size(input,2)*(i-1)) = input(i,j);
    end
end

end

```